Genetic Algorithm Optimized Feature Transformation – A Comparison with Different Classifiers

Zhijian Huang[1], Min Pei[1], Erik Goodman[1], Yong Huang[2], and Gaoping Li[3]

[1] Genetic Algorithms Research and Application Group (GARAGe)
Michigan State University, East Lansing, MI
{huangzh1,pei,goodman}@egr.msu.edu

[2] Computer Center, East China Normal University, Shanghai, China
siewl@online.sh.cn

[3] Electrocardiograph Research Lab, Medical College
Fudan University, Shanghai, China
gpli@shmu.edu.cn

Abstract. When using a Genetic Algorithm (GA) to optimize the feature space of pattern classification problems, the performance improvement is not only determined by the data set used, but also depends on the classifier. This work compares the improvements achieved by GA-optimized feature transformations on several simple classifiers. Some traditional feature transformation techniques, such as Principal Components Analysis (PCA) and Linear Discriminant Analysis (LDA) are also tested to see their effects on the GA optimization. The results based on some real-world data and five benchmark data sets from the UCI repository show that the improvements after GA-optimized feature transformation are in reverse ratio with the original classification rate if the classifier is used alone. It is also shown that performing the PCA and LDA transformations on the feature space prior to the GA optimization improved the final result.

1 Introduction

The genetic algorithm (GA) has been tested as an effective search method for high-dimensional complex problems, taking advantage of its capability for sometimes escaping local optima to find optimal or near optimal solutions. In pattern classification, GA is widely used for parameter tuning, feature weighting [1] and prototype selection [2].

Feature extraction and selection is a very important phase for a classification system, because the selection of a feature subset will greatly affect the classification result. GA has recently also been used in the area of feature extraction. The optimization of feature space using GA can be linear [3], or non-linear [4], where in both cases, the GA stochastically, but efficiently, searches in a very high-dimensional data space that makes traditional deterministic search methods run out of time. The GA approach can also be combined with other traditional feature transformation methods.

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2724, pp. 2121–2133, 2003.

Prakash presented the combination of GA with Principal Components Analysis (PCA), where instead of the few largest Principal Components (PCs), a subset of PCs from the whole spectrum was chosen by the GA to get the best performance [5].

In this work, three classifiers – a *k*NN classifier, a Bayes classifier and a Linear Regression classifier – are tested, together with the PCA and LDA transformations. One new, real-world dataset, the Electrocardiogram (ECG) data, and five benchmark datasets from the UCI Machine Learning Repository [6] are used to test the approach.

The paper starts with an introduction to GA approaches in the area of pattern classification in Section 2, followed by our solution designed in Section 3. Section 4 presents the results on ECG data with detailed comparison with regard to both classifier choice and the use of PCA/LDA transformations. Section 5 extends the tests to five benchmark pattern classification test sets, by using the best solution on PCA/LDA combination found in Section 4. Section 6 concludes the paper and Section 7 proposes some possible future work.

2 GA in Pattern Classification

Generally, the GA-based approaches to pattern classification can be divided into two groups:

- Those directly applying GA as part of the classifier.
- Those optimizing parameters in pattern classification.

2.1 Direct Application of GA as Part of the Classifier

When the GA is directly applied as part of the classifier, the main difficulty is how to represent the classifier using the GA chromosome. Bandyopadhyay and Murthy proposed an idea using a GA to perform a direct search on the partitions of an *N*-dimensional feature space, where each partition represents a possible classification rule [7]. In this approach, the decision boundary of the *N*-dimensional feature space is represented by *H* lines. The genetic algorithm is used to find those lines that minimize the misclassification rate of the decision boundary. The number of lines, *H,* turns out to be a parameter similar to the *k* in the *k*NN classifier. More lines (higher *H*) do not necessarily improve the classification rate, due to the effect of over-fitting.

In addition to using lines as space separators, Srikanth et al [8] also gave a novel method clustering and classifying the feature space by ellipsoids.

2.2 Optimizing Parameters in Pattern Classification by GA

However, most of the approaches using GA in pattern classification do not design the classifier using GA. Instead, GA is used to estimate the parameters of the pattern classification system, which can be categorized into the following four classes:

GA-Optimized Feature Selection and Extraction. Feature selection and extraction are the most widely used applications of GA in pattern classification. The classification rate is affected indirectly when different weights are applied to the features. A genetic algorithm is used to find a set of optimal feature weights that can improve the classification performance on training samples. Before GA-optimized feature extraction and selection, traditional feature extraction techniques such as the Principal Components Analysis (PCA) can be applied [5], while after that, a classifier should be used to calculate the fitness function for the GA. The most commonly used classifier is the *k*-Nearest Neighbor classifier [9], [1].

GA-Optimized Prototype Selection. In supervised pattern classification, the reference set or training samples are critical for the classification of testing samples. A genetic algorithm can be also used in the selection of prototypes in case-based classification [2], [10]. In this approach, a subset of the most typical samples is chosen to form a prototype, on which the classification for testing and validation samples is based.

GA-Optimized Classifier. GA can be used to optimize the input weight or topology of a Neural Network (NN) [4]. It is intuitive to give weights for each connection in a NN. By evolving the weights using GA, it is possible to throw away some connections of the neural network if their weights are too small, thus improving the topology of the NN, too.

GA-Optimized Classifier Combination. The combination of classifiers, sometimes called Bagging and Boosting [11], may also be optimized by Genetic Algorithm. Kuncheva and Jain [12], in their design of the Classifier Fusion system, not only selected the features, but also selected the types of the individual classifiers using a genetic algorithm.

3 GA-Optimized Feature Transformation Algorithm

This section first reviews the two models for feature extraction and feature selection in pattern classification. Then a GA-optimized feature weighting and selection algorithm based on the wrapper model [13] is proposed, outlining the structure of the experiment in this paper.

3.1 The Filter Model and the Wrapper Model

For the problem of feature extraction and selection in pattern classification, two models play important roles. The *filter model* chooses features by heuristically determined “goodness”, or knowledge, while the *wrapper model* does this by the feedback of the classifier evaluation, or experience. Fig. 1 illustrates the differences between these two models.

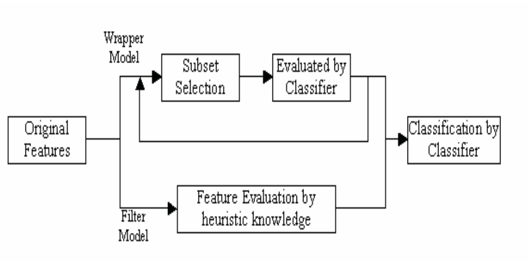

Fig. 1. Comparison of filter model and wrapper model

Research has shown that the wrapper model performs better than the filter model, comparing the predictive power on unseen data [14]. Some widely used feature extraction approaches, such as Principal Components Analysis (PCA), belong to the Filter model because they rank the features by their intrinsic properties: the eigenvalues of the covariance matrix. Most recently developed feature selection or extraction techniques are categorized to be Wrapper models, taking into consideration the classification results of a particular classifier. For example, the GFNC (Genetically Found, Neurally Computed) approach by Firpi [4] uses a GA and a Neural Network to perform non-linear feature extraction with the feedback from a *k*NN classifier as the fitness function for the Genetic Algorithm. A modified PCA approach proposed by Prakash also uses genetic algorithms to search for the optimal feature subset with the feedback result of the classifier [5].

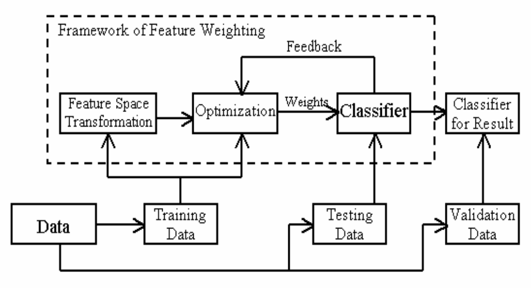

Fig. 2. Classification system with feature weighting

3.2 GA-Optimized Pattern Classification with Feature Weighting

Consider the wrapper model introduced above; in any of the pattern classification systems with weighted features, there are five components that need to be determined as illustrated in Fig. 2.

- The dataset used.
- The feature space transformation to be applied on the original feature space, known as the feature extraction phase in traditional pattern classification systems.

- The optimization algorithm which searches for the best weighting for the features;
- The classifier used to get the feedback, or fitness, for the GA, of the feature weighting. (The Induction Algorithm in the Wrapper Model [13])
- The classifier to calculate the final result for the classification problem based on the newly weighted features (the classifier for the result).

From Fig. 2, we can see that among these components, the *feature space transformation*, the *optimization* and the *classifier* are the plug-in procedures for feature weighting optimization. The feature-weighting framework is the plug-in procedure in traditional classification systems that transforms the feature space, weights each feature, evaluates and optimizes the weight attached to each feature.

In this paper, we can replace each of the components listed above by the specific choices we made, as follows:

- *Data*: ECG data and five other data sets from UCI repository.
- *Feature Space Transformation*: PCA, LDA and their combinations.
- *Optimization Algorithm*: Genetic algorithm or none. When implemented with different classifiers: feature weighting for *k*NN classifier, and feature selection for other classifiers. The reason for using feature weighting for the *k*NN classifier is because of its distance metric, that will affect the classification result by changing its weight [1], while for the Bayes classifier and the linear regression classifier, feature weighting has no further effect on the training error when compared with feature selection [15].
- *Classifier* (induction algorithm): A *k*NN classifier, a Bayes classifier and a linear regression classifier.
- *Classifier for Result*: Same classifier as used for the induction algorithm.

4 Test Results for ECG Data

The test results for ECG data, with various settings regarding the feature space transformation, using the GA or not, and using various classifiers, are presented and compared in this section. We will first discuss the experimental settings, and then move on to the results.

4.1 Experimental Settings

The ECG data used in this paper is directly extracted from the 12-lead, 10-second digital signal acquired from Shanghai Zhongshan Hospital, containing 618 normal ECG cases and 618 abnormal cases. The abnormal cases contain three different kinds of diseases with roughly the same number of cases of each. Altogether, 23 features, including 21 morphological features and 2 frequency domain features, are extracted from the original signal.

For a non-GA-optimized classifier, the data is partitioned into training samples and validation samples; for a GA-optimized algorithm, the data is partitioned into training data, testing data and validation data, in an n-fold cross-validation manner. If not specifically indicated, here the *n* is set to be 10. Table 1 lists the details of the data partitioning.

A simple genetic algorithm (SGA) is used here to do the optimization. The crossover rate is 30% and the mutation rate is set to 0.03 per bit. The program runs for 200 generations with a population size of 50 individuals (80 individuals for kNN feature weighting). When there has been no improvement within the last 100 generations, the evolution is halted.

Classifiers: A 5-nearest neighbor classifier is used. A Bayes Plug-In classifier with its parameters estimated by Maximum Likelihood Estimation (MLE) is implemented, and a linear regression classifier uses the simple multivariate regression combined with a threshold decision to predict the class labels.

With the kNN classifier, the feature weighting is allowed to range among 1024 different values between 0.0 and 10.0, with minimum changes of about 0.01, as determined by the GA, by setting the chromosome for each feature to be 10 binary digits. With the Bayesian classifier and linear regression classifier, only feature selection was tested.

Table 1. Summary of Data Partitioning

Experiments	Training	Testing	Va-lid
Non-GA	40%	N/A	60%
GA (*n*-fold)	$40\%\times\frac{n-1}{n}$	$60\%\times\frac{n-1}{n}$	$\frac{1}{n}$

Table 2. Result of *k*NN classifier (*k* = 5)

Settings	*k*NN (*k* = 5) Results of Classification Rate in %				
Fea. Trans	Non-GA	GA	Trn	Improve	P-Value
None	73.09	74.54	78.29	1.45±3.23	0.3008
PCA	74.10	77.16	79.39	3.06±1.59	**0.0043**
LDA	73.09	77.00	78.72	3.91±2.24	**0.0065**
Both	72.41	75.38	79.82	2.97±2.78	**0.0404**
Overall P value:					**0.0000**

4.2 Results and Conclusions

Tables 2-4 list the results of GA-optimized feature extraction using a kNN classifier (*k*=5), Bayes Plug-In classifier and linear regression classifier. The improvement after GA optimization is represented by the average improvement, with a two-tailed t-test with a 95% confidence interval. A P value indicating the probability of the Null Hypothesis (the improvement is 0) is also given, among which results having a 95% significant improvement are in bold font.

In addition to the row-wise statistics, an overall improvement significance level based on the improvement percentage is calculated for each classifier, which is a P value on all the improvement ratio values of that classifier. This significance indicator can be considered as a final summary of the GA improvement based on a particular classifier and is listed at the bottom of each table.

Table 3. Result of Bayes classifier

Set-tings	Bayes Results of Classification Rate in %				
Fea. Trans	Non-GA	GA	Trn	Improve	P-Val
None	71.92	74.19	76.51	2.27±3.69	0.1912
PCA	72.53	74.27	77.23	1.75±1.95	0.0730
LDA	72.95	75.47	77.23	2.52±3.11	0.0968
Both	71.85	76.99	78.33	5.14±2.54	**0.0014**
Overall P value:					**0.0000**

Table 4. Result of Linear Regression classifier

Set-tings	Linear Regression Results, Rate in %				
Fea. Trans	Non-GA	GA	Trn	Improve	P-Value
None	77.68	76.49	79.08	-1.19±3.23	NA
PCA	76.48	78.02	79.87	1.53±3.09	0.25
LDA	78.44	77.42	77.36	-1.02±3.15	NA
Both	77.59	77.41	77.84	-0.18±2.79	NA
Overall P value:					NA

Conclusions:

1. For the ECG data, the utility of GA-optimized feature weighting and selection depends on the classifier used. The GA-kNN feature weighting and GA-Bayes feature selection yield significant improvement, with three rows showing a significance level of more than 90%, and the fourth showing less improvement, but in the same direction. As a result, the overall significance based on the improvement ratio is 99.99%. In contrast, the GA-optimized linear regression classifier does not show improved performance, and the inconsistency of change direction makes it likely that any systematic improvement due to the GA-optimized weights, if it exists, is very small.

2. The PCA and/or LDA transformation is useful, in combination with GA optimization. As we can see from the kNN and Bayes classifiers, although applying the PCA and/or LDA transformation on the non-GA optimized classifiers yields no major progress, their combination with GA yields significant improvement as well as better final classification rates. In some sense, the PCA and LDA transformations can help the GA to break the “barrier” of the optimum classification rate.

3. The more accurate the original classifier is, the less improvement GA optimization yields. From the data, we can see that the linear regression classifier is the most powerful classifier if used alone, and also the least improved classifier after GA optimization. Fig. 3 illustrates this conclusion.

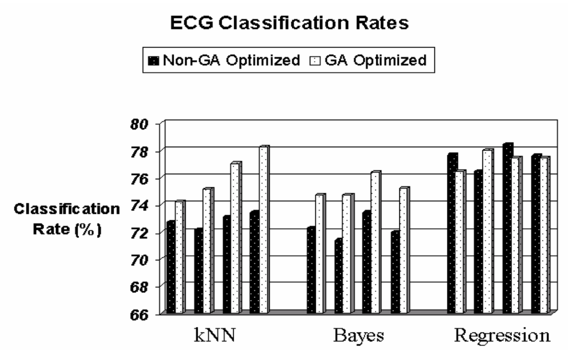

Fig 3. Summary of the ECG classification rate

4.3 Results of GA Search

For small numbers of features (here we set the standard: ≤ 15) in feature selection, it is possible to apply an exhaustive search in the whole feature space, thus providing the possibility to determine whether the GA can find the best solution or not. At the same time, some information about the usefulness of features can be traced from the terrain graph of the whole feature space.

In some cases, the global optimum was found. However, in Fig. 4, although the GA found quite a good result, it was not the global optimum. But since the classification rate for the validation samples is related to but not linearly dependent on the training rate obtained by the GA, such a near-optimal result seems to produce good performance for a pattern classification problem.

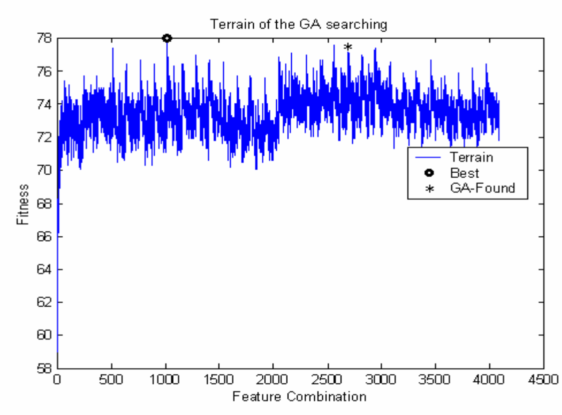

Fig. 4. The GA search space (Best not found)

5 Test Results for Other Data

Five datasets from UCI repository were tested to further validate our result from the previous section. A brief introduction to the data sets is given first, followed by the results and discussion.

5.1 The Testing Datasets

- WDBC: The Wisconsin Diagnostic Breast Cancer [16] data contains 30 features expanded from the original 10 features by taking mean, standard error and extreme value of the originally measured features. The dataset has 357 benign and 212 malignant samples, with highest reported classification accuracies around 97%.
- LIVER: The BUPA Liver Disorders data has 6 numerical features on 345 instances, with 2 classes.
- PIMA: The Pima Indians Diabetes Database has 8 features on female diabetes patients, classified to be positive and negative. The total of 768 samples has 500 negative and 268 positive samples.
- SONAR: The Sonar data compares mines versus rocks based on their reflected sonar signals. With the 111 metal patterns and 97 rock patterns, each of them has 60 feature values [17].
- IONO: The Ionosphere database from Johns Hopkins University [18] contains 351 instances of radar signals returned from the ionosphere, with 34 features. It contrasts "Good" and "Bad" radar returns that show or not show evidence of some types of structure in the ionosphere.

5.2 Results for Benchmark Datasets

The results presented here are all based on both PCA and LDA transformation, which were shown to be useful in GA-optimized pattern classification in Section 4.

Table 5. *k*NN classifier (Benchmark Datasets)

DATA	Non-GA	GA	Train	Improve	P-Value
WDBC	91.38	94.19	92.24	2.81±2.93	0.0571
LIVER	66.36	66.64	72.89	0.28±6.28	0.9126
PIMA	70.44	72.65	76.97	2.21±2.90	0.1076
SONAR	69.28	74.97	64.06	5.69±5.91	0.0563
IONO	79.19	80.32	69.98	1.13±3.13	0.3986
Overall P value:					**0.0001**

Table 6. Bayes classifier (Benchmark Datasets)

DATA	Non-GA	GA	Train	Improve	P-Value
WDBC	94.78	94.71	96.66	-0.07±2.44	NA
LIVER	56.96	64.00	67.40	7.04±6.57	**0.0334**
PIMA	72.91	74.74	76.30	1.84±3.73	0.2864
SONAR	46.83	68.27	76.37	21.45±10.77	**0.0015**
IONO	64.88	90.64	94.17	25.76±11.52	**0.0000**
Overall P value:					**0.0000**

From the results shown in Table 5—7, we can see that:

1. More than half of the row-wise results show an improvement with significance above 90%. For some settings, such as Bayes classifier for SONAR data, the original classification rate is very low, but GA can make up for this and yield a decent result. The effect of GA optimization here is to reach a fairly good result, if not the best, when the original settings of the classifier are not very good.

Table 7. Regression classifier (Benchmark Datasets)

DATA	Non-GA	GA	Train	Improve	P-Value
WDBC	94.43	95.26	95.49	0.83±2.50	0.4737
LIVER	66.04	67.23	66.33	1.19±6.21	0.6752
PIMA	76.57	76.69	77.55	0.11±3.14	0.9364
SONAR	63.65	72.44	74.78	8.79±9.91	0.0757
IONO	85.12	82.61	84.98	-2.51±5.78	NA
Overall P value:					**0.1364**

2. The linear regression classifier has the highest classification rate among the three. After GA-optimized feature weighting and selection, the gaps in performance among the various classifiers became smaller, as shown in Fig. 5.

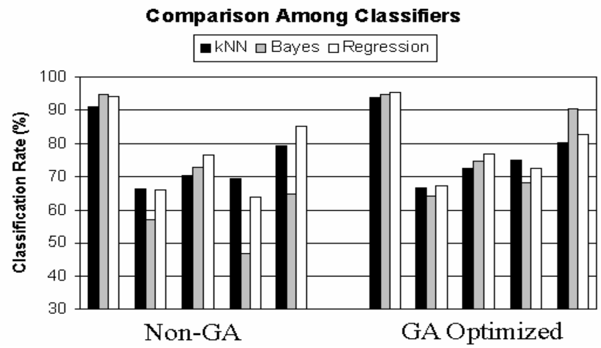

Fig. 5. Comparison, before and after GA optimization

6 Conclusions

The utility of GA-optimized feature weighting and selection depends on both the classifier and the data set. Especially in those cases when a particular classifier has a better classification rate than some other classifiers, the potential improvement from GA optimization of the better classifier seems to be quite limited in comparison with its performance improvement of the poorer classifiers.

Clearly, to evaluate a new approach involving optimization on feature space, it is necessary to test on different classifiers, and the improvement in the best classifier will be the most convincing evidence of the utility of that method.

In this work, the Genetic Algorithm shows powerful searching ability in high-dimensionality feature spaces. By comparing it with an exhaustive search algorithm on small-scale problems, it was determined that the GA found the optimal or a nearly optimal solution with a computational complexity of $O(n)$.

The results from Sections 4 and 5 indicate that over-fitting exists in various approaches. While the training performances can be significantly improved, the improvements on the validation samples lag behind in every case.

The tests run in Section 4 show that the PCA and LDA transformations are very useful in pattern classification. The significance levels of GA optimization are greatly improved for the kNN and Bayes classifiers, though the absolute values after PCA and LDA transformation without GA optimization do not differ much from the non-PCA and non-LDA cases. The point is, however, that GA-optimized feature extraction and selection extend the utilities of those traditional feature transformation techniques.

7 Future Works

To solve the problem of over-fitting, one possible approach is to evaluate the solution not only by the classification rate on training data, but also to consider the margin between classes and the boundary, because a larger margin means a more general and more robust classification boundary. Support Vector Machines (SVM) are classification systems that separate the training patterns by maximizing the margins between support vectors (those nearest patterns) and the decision boundary in a high-dimensional space. Work by Gartner and Flach [19] that used SVM rather than a GA to optimize the feature space yielded a statistically significant result on Bayes classifiers.

Another possible improvement may be non-linear feature construction using GA [4]. Non-linear feature construction can generate more artificial features so that the GA can search for more hidden patterns. However, the problem of over-fitting still theoretically exists.

This work was supported in part by the National Science Foundation of China (NSFC) under Grant 39970210.

References

1. M. Pei, E. D. Goodman, W. F. Punch. "Feature Extraction Using Genetic Algorithms", *Proceeding of International Symposium on Intelligent Data Engineering and Learning'98* (IDEAL'98) Hong Kong, Oct. 1998.
2. David B. Skalak. “Using a Genetic Algorithm to Learn Prototypes for Case Retrieval and Classification”, *Proceedings of the AAAI-93 Case-Based Reasoning Workshop,* pages 64–69, Washington, D.C., American Association for Artificial Intelligence, Menlo Park, CA, 1994.
3. W.F. Punch, E.D. Goodman, Min Pei, Lai Chia-Shun, P. Hovland and R. Enbody. “Further Research on Feature Selection and Classification Using Genetic Algorithms”, In *5th International Conference on Genetic Algorithms*, Champaign IL, pages 557–564, 1993
4. Hiram A. Firpi Cruz. “Genetically Found, Neurally Computed Artificial Features With Applications to Epileptic Seizure Detection and Prediction”, Master’s Thesis, University of Puerto Rico, Mayaguez, 2001.
5. M. Prakash, M. Narasimha Murty. "A Genetic Approach for Selection of (Near-) Optimal Subsets of Principal Components for Discrimination", *Pattern Recognition Letters,* Vol. 16, pages 781–787, 1995.
6. Blake, C.L. & Merz, C.J. UCI Repository of machine learning databases [http://www.ics.uci.edu /~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science (1998).
7. S. Bandyopadhyay, C.A. Murthy. “Pattern Classification Using Genetic Algorithms”, *Pattern Recognition Letters,* Vol. 16, pages 801–808, 1995.
8. R. Srikanth, R. George, N. Warsi, D. Prabhu, F.E.Petry, B.P.Buckles. “A Variable-Length Genetic Algorithm for Clustering and Classification”, *Pattern Recognition Letters*, Vol. 16, pages 789–800, 1995.
9. W. Siedlecki, J. Sklansky. “A Note on Genetic Algorithms for Large-Scale Feature Selection”, *Pattern Recognition Letters*, Vol. 10, pages 335–347, 1989.
10. Ludmila I. Kuncheva. “Editing for the *k*-Nearest Neighbors Rule by a Genetic Algorithm”, *Pattern Recognition Letters*, Vol. 16, pages 809–814, 1995.
11. Richard O. Duda, Peter E. Hart, David G. Stock. “Pattern Classification”, Second Edition, Wiley 2001.
12. Ludmila I. Kuncheva and Lakhmi C. Jain. “Designing Classifier Fusion Systems by Genetic Algorithms”, *IEEE Transactions on Evolutionary Computation*, Vol. 4, No. 4, September 2000.
13. Ron Kohavi, George John. “The Wrapper Approach”, *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Edited by Hiroshi Motoda, Huan Liu, Kluwer Academic Publishers, July 1998.
14. George H. John, Ron Kohavi, Karl Pfleger. "Irrelevant Features and the Subset Selection Problem", *Proceedings of the Eleventh International Conference of Machine Learning*, pages 121–129, Morgan Kaufmann Publishers, San Francisco, CA, 1994.
15. Zhijian Huang. “Genetic Algorithm Optimized Feature Extraction and Selection for ECG Pattern Classification”, Master’s Thesis, Michigan State University, 2002.
16. O.L. Mangasarian, W.N. Street and W.H. Wolberg. “Breast Cancer Diagnosis and Prognosis via Linear Programming”, *Operations Research*, 43(4), pages 570–577, July-August 1995.
17. Gorman and T. J. Sejnowski. “Learned Classification of Sonar Targets Using Massively Parallel Network”, *IEEE Transactions on Acoustic, Speech and Signal Processing*, 36 (7), pages 1135–1140, 1988.

18. V.G. Sigilito, S.P. Wing, L.V. Hutton and K.B. Baker. “Classification of Radar Returns from the Ionosphere Using Neural Networks”, *Johns Hopkins APL Technical Digest*, Vol. 10, pages 262–266, 1989.
19. Thomas Gartner, Peter A. Flach. “WBC_{SVM}: Weighted Bayesian Classification based on Support Vector Machine”, *Eighteenth International Conference on Machine Learning (ICML-2001),* pages 156–161. Morgan Kaufmann, 2001.