

Polyoptimizing Genetic Algorithm for Feature Subset Selection

Ewy Mathe and John Grefenstette*
Bioinformatics and Computational Biology
School of Computational Sciences
George Mason University

Abstract

The analysis of large biological data sets that arise in gene expression or proteomics experiments often involves the selection of a subset of the available features that supports efficient classification. Finding multiple, distinct solutions to the feature subset selection problem may lead to increased biological insights. In this paper we address the problem of finding multiple solutions to the feature subset selection problem using a polyoptimizing genetic algorithm which incorporates a dynamic penalty function. We illustrate the approach on an ovarian cancer classification problem using proteomics data.

1. Introduction

The availability of proteomics data is profoundly affecting the approach to disease diagnosis and treatment [1]. High-throughput analysis of proteins enables development of novel biomarkers for early diagnosis of cancer and other diseases [2], as well as elucidation the underlying biological processes. For example, recent progress has been reported in the early diagnosis of ovarian cancer using proteomics data [3]. In this approach, proteomic patterns may be obtained by mass spectrometry, in which blood serum samples are bombarded with electrons, thereby creating positively charged molecules and ions that are then accelerated through a tube and sorted according to their mass-charge (m/z) ratios. The identity of the charged particles is generally unknown and they may represent entire proteins, protein fragments, or other substances (i.e. sugars, blood lipids) that are present in the serum. In addition, there is variability in the mass spectrometry results from person to person, in addition to variability between disease and healthy persons. Clearly, extracting diagnostic signals from large proteomics data sets is challenging, since biomarkers may consist of subtle patterns hidden in complex mass spectra [4]. The detection of those differences is the key to finding proteomics patterns that will indicate disease.

Previous studies have shown that genetic algorithms are effective methods for selecting features for classification from proteomics data [5, 6]. More generally, genetic algorithm have been applied to feature selection problems in a wide variety of domains, including searching for patterns in microarray data [7], diagnosis based on SNP data [8], breast cancer classification [9-12], classification of hepatic lesions from computed tomography (CT) images [13] and face recognition [14]. These previous approaches generally aim to identify a single subset of features that provides improved classification. In order to strengthen insights into the underlying biological processes, it would be useful to identify the range of features that contribute to the diagnosis. In this paper we address the problem of finding multiple solutions to the feature subset selection problem using a polyoptimizing genetic algorithm. The next section describes the method in more detail. We then illustrate the approach on an ovarian cancer classification problem using proteomics data.

* To whom correspondence should be addressed. Email: jgrefens@gmu.edu

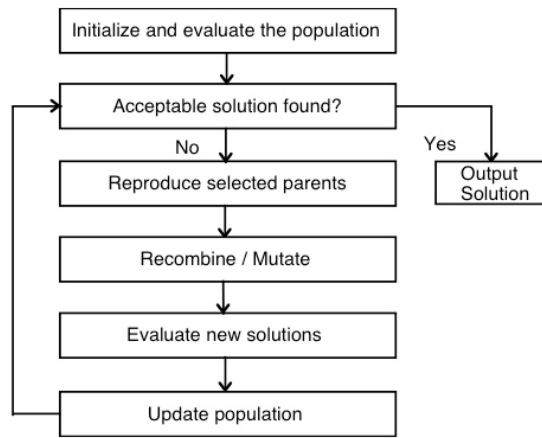


Figure 1. The standard genetic algorithm. A population of candidate solution is generated and evaluated. In each generation, candidates are selectively reproduced, recombined and mutated to form the next generation of candidate solutions.

2. Methods

Genetic Algorithms for Feature Subset Selection

A genetic algorithm is a heuristic search method for finding optimal solutions in complex search spaces, using principles based on population biology and genetics. The algorithm starts with an initial population (i.e. a set of features in this case) and evaluates that population according to a pre-defined evaluation criterion (e.g., how well the feature set discriminates between cancer and normal patients). Evolutionary operators such as selection, recombination and mutation, are then applied to create offspring and the population is updated and re-evaluated, as shown in Figure 1. Figure 2 illustrates the use of a genetic algorithm for feature subset selection using a so-called *wrapper* method, in which each feature subset evaluated by the genetic algorithm is assigned a fitness value based on the accuracy of a classifier built using that subset of features.

To apply the genetic algorithm to the feature selection problem, an appropriate representation of the search space must be specified. In this study, the population consists of lists of features that are to be included in the classifier. For instance, two members of the population might look like the following:

(10 123 456 798 835 888 923)
 (29 378 456 645 798 912 988)

where the first list specifies that features 10, 123, and so on, should be used in the classification process. In order to introduce new candidate feature sets, recombination and mutation operators are applied. Recombination involves the swapping features between feature sets, while a mutation involves a random change in one or more features in one set (Fig. 3).

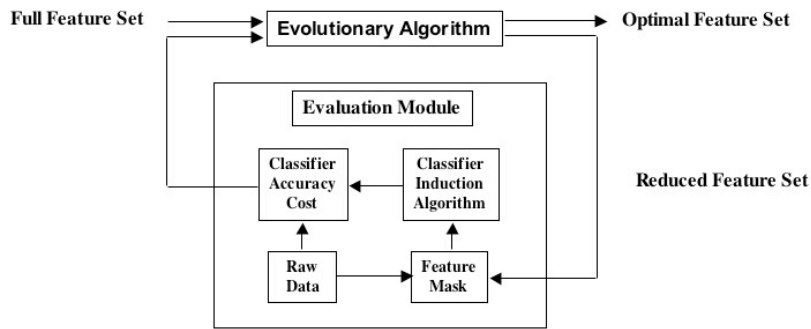


Figure 2. Application to genetic algorithms to feature subset selection using a wrapper model. The fitness of a feature subset is determined by the accuracy of a classifier built using those features.

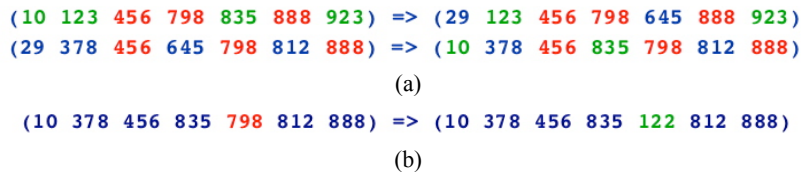


Figure 3. Genetic operators on feature lists. (a) Recombination of two feature lists: all features in common are inherited by both offspring, and distinct features are randomly assigned to either offspring. (b) Mutation of feature list involves making a substitution of a randomly selected feature for one feature in the list.

For each generation, the population of candidate features is evaluated according to the survival of the fittest competition. This means that the number of offspring assigned to each feature set is proportional to the accuracy of that set of features when used to train a classifier to discriminate between disease and non disease samples. The accuracy is calculated using a 5-fold cross validation of a kNN nearest neighbor classifier.

The k-nearest neighbor classifier evaluates the k closest training samples, based on their Euclidean distance, from a test sample. For instance, if a feature set containing 10 features is being considered, then each sample is represented by a point in 10 dimensions. The majority class of the k closest points in Euclidean distance to the test sample predicts the class of the test sample (i.e. whether cancer or normal). The prediction of the test sample is then compared to its actual class, and an accuracy measure of all test samples is calculated. Those feature sets that have high accuracy will generate more offspring when the population of the genetic algorithm is subsequently updated.

Polyoptimizing Genetic Algorithms

This study investigates the effectiveness of a poly-optimizing genetic algorithm (POGA) in identifying alternative sets of features that have high prediction accuracy. The POGA extends the genetic algorithm paradigm to the problem of finding multiple distinct solutions to complex optimization problems. In a POGA, the fitness assigned to a candidate solution is determined in part by the set of points visited previously by the algorithm. By means of a dynamic penalty function, the fitness of a point may be

reduced if it is sufficiently similar to previously visited points. As a result, the algorithm can be expected to shift its focus to alternative regions of the search space, looking for alternative solutions as it proceeds. All acceptable solutions may be recorded for further analysis.

Let $g: X \rightarrow R$ be the objective function that maps structures in search space X into the reals. Let a distance metric $d: X \rightarrow R$ be defined over the search space X , such that $d(x_i, x_j)$ is the distance between search points x_i and x_j . The distance metric is defined for each problem and is used to determine whether alternative solutions are sufficiently distinct to meet the user's needs. A *polyoptimization* problem may be defined by the quintuple $\langle X, g, d, \delta, \epsilon \rangle$ where the goal is to find a maximal set

$$H = \{ x_i \in X \mid g(x_i) \geq \epsilon \text{ and } d(x_i, x_j) \geq \delta \text{ for all } x_j \in H, j \neq i \} \quad (1)$$

That is, it is desired to find as many distinct, acceptable solutions as possible, where a solution is acceptable if its objective function value is at least ϵ , and two solutions are distinct if the distance between them is at least δ .

Given this operational definition of the problem, a natural efficiency metric for poly-optimization algorithms is the hit rate $h(t)$ defined as the size of set H_t of acceptable, distinct solutions found by time t , where t refers to the number of points from X evaluated so far.

In a POGA, the fitness of each point evaluated during the search will be transformed by a dynamic penalty function. The fitness function is used by the genetic algorithm to decide which members of the current population will be selected to generate offspring in the next generation. In a POGA, the fitness function at time t is:

$$f(x, t) = g(x) - p(x, t) \quad (2)$$

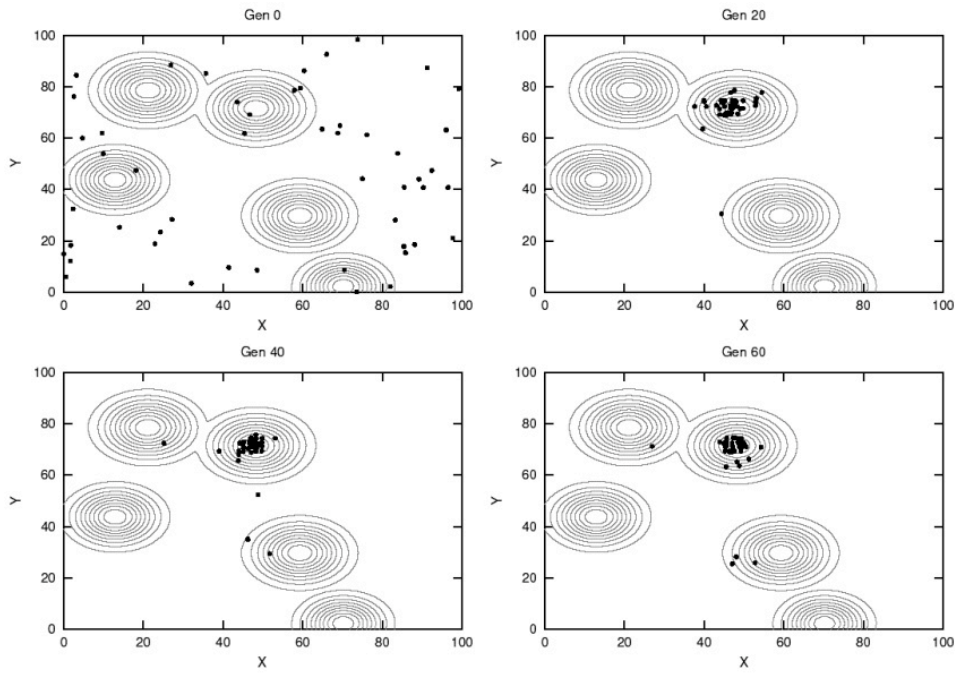
where $g(x)$ is the objective function and $p(x, t)$ is the penalty function at time t . The penalty function $p(x, t)$ is defined in terms of a pairwise penalty function $r(u, v)$ that computes the penalty to be applied to point u based on the previously visited point v . More specifically, the pair-wise penalty function for a point in the search space (i.e., a set of features) is defined as:

$$p(x_t, t) = \sum r(x_t, x_i) \quad (3)$$

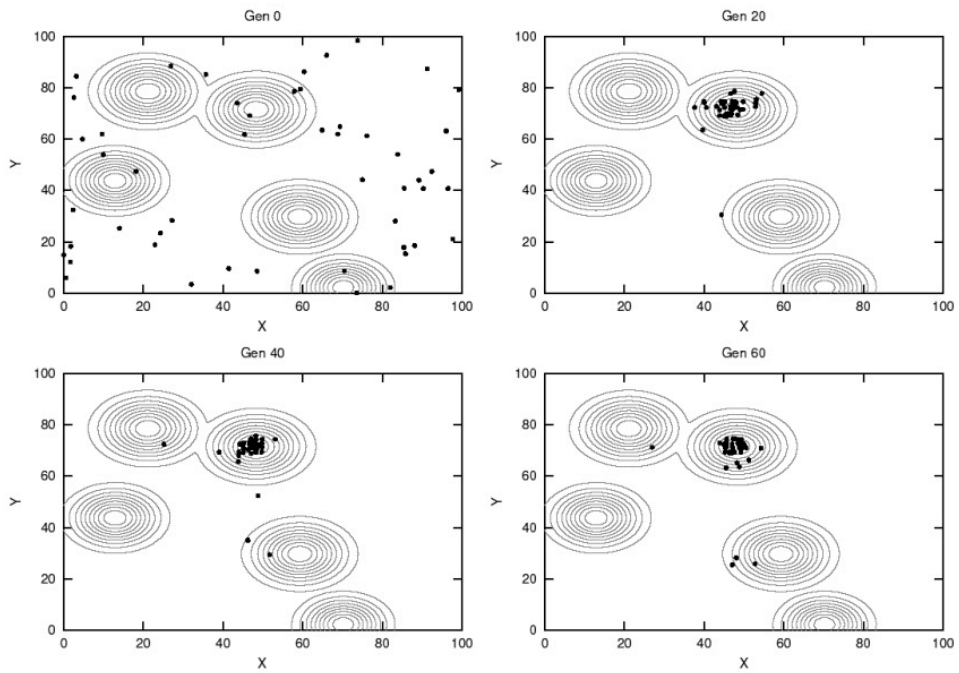
where

$$\begin{aligned} r(u, v) &= \text{penalty applied to point } u \text{ based on previously visited point } v \\ &= a (1 - (d(u, v) / \delta) 2) \\ d(u, v) &= \text{distance metric on domain} \\ a &= \text{penalty for "direct hit", i.e., revisiting a previously evaluated point} \end{aligned}$$

Applying the penalty function generally decreases the fitness of a region of the search space once it has been explored. Consequently, the algorithm starts searching in other areas for alternative solutions. As an illustration, a comparison between solutions to an artificial test function found by a standard genetic algorithm and by a POGA is depicted in Figure 4.



(4a)



(4b)

Figure 4: Comparing a standard genetic algorithm (a) and the poly-optimizing genetic algorithm (b) on an artificial test function with multiple optimal solutions. The genetic algorithm converges to a single solution while POGA finds multiple distinct solutions.

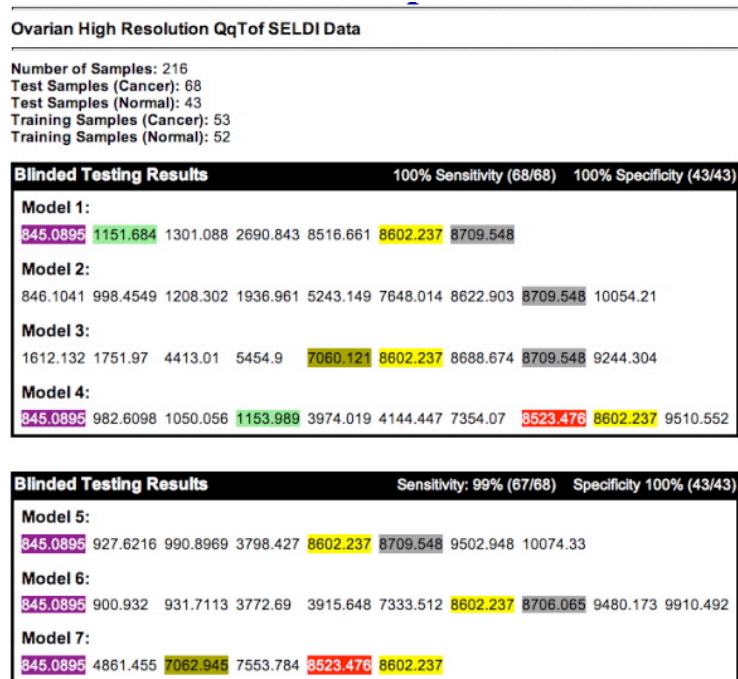


Figure 5. Previous results obtained from genetic algorithms on proteomics data. See <http://ncifdaproteomics.com/> for further details.

Proteomics Data

We evaluated the ability of the POGA to find non-redundant feature sets that have a high accuracy in discriminating between cancer and non-cancer samples of proteomics data obtained from Dr. Lance Liotta at NCI (personal communication). The data consists of mass spectroscopy values (mass/charge) for 216 blood serum samples. Of those samples, 95 are classified as normal (~44%) and 121 are classified as cancer (~56%). For each sample, there are 349,905 mass/charge values. To reduce the number of possible features per sample to 7,105, a linear growing window binning algorithm was applied as provided by Dr. Donald Johann (personal communication) at NCI. The data was further subdivided into a training set comprising 105 samples (53 cancer, 52 normal) and a test set with 110 samples (67 cancer, 43 normal).

Previous work using genetic algorithms has shown success in identifying sets of features that accurately discriminate between ovarian cancer and non-ovarian cancer samples. Figure 5 illustrates previously reported results that indicate the value of using genetics algorithms for finding sets of features. In addition to the high sensitivity and specificity values, it is important to note that there are some features that occur in multiple high-accuracy sets (i.e. 845.0895) and that there are pairs (i.e. 845.0895 and 8602.237) or triplets (i.e. 845.0895, 8602.237, and 8709.548) of features that occur in optimal sets. Such cooperativity between features within sets may have some biological significance worth pursuing. In addition, there may be some further patterns within the optimal feature sets that may reveal new information. For this reason, rather than finding the best feature set, searching for as many non-redundant features sets as possible may be valuable.

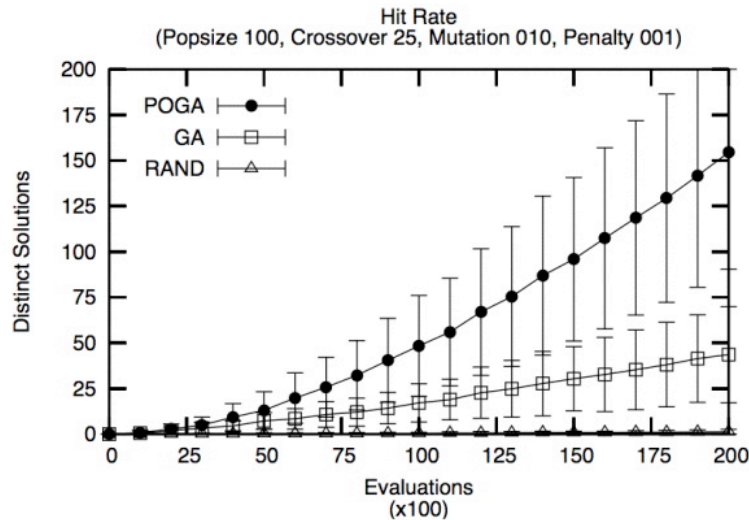


Figure 6: Comparing the amount of non-redundant solutions generated by POGA, standard genetic algorithm, and random search.

We used the following parameters to test the effectiveness of the POGA in finding distinct, acceptable models for this data set.:

- population size = 100 feature sets
- generations = 200
- crossover rate = 0.25
- mutation rate = 0.01
- penalty functions: $\delta = 0.7, a = 0.001$

Each feature set consists of 10 features (mass/charge values) and the evaluation function consists of a kNN classifier with $k=5$, evaluated by five-fold cross validation. Acceptable feature sets (i.e., members of the hit set) have at least a 95% accuracy and are at least 70% distinct compared to all previously found acceptable models. For comparison purposes, the search for distinct and accurate solutions was also performed using a standard genetic algorithm and a random search. Each algorithms was runs 20 times on the data set.

3. Results

The results using the training set averaged over twenty runs clearly demonstrate that the POGA consistently finds the significantly more non-redundant models than either the standard genetic algorithm or random search (Fig 6). The low number of distinct solutions found by random search indicates that non-redundant solutions are relatively rare in the search space of all feature subsets. The small number of non-redundant solutions found by the standard GA indicates that the algorithm tends to converge to specific regions of the search space. The number of non-redundant solutions found by POGA appears to be accelerating as the search proceeds, indicating that it has not converged to a single region of the search space. The error bars are considered to be due to the variability in the initial population.

	Total Models (Mean)	Non-Redundant Models (Mean)	Non-Redundant Models (%)
RAND	1.3	1.3	100%
GA	7575	43.6	0.6%
POGA	4133	154.5	3.7%

Table 1. Total number of acceptable solutions and number of non-redundant solutions found by random search, standard GA and POGA.

As can be seen from Table 1, the random search method finds very few acceptable models yet all of them are non-redundant, as expected. On the other hand, the standard genetic algorithm finds the largest number of acceptable models but only 0.6% of them are non-redundant. This indicates a convergence to a region containing very similar acceptable solutions. While POGA finds fewer acceptable models than the standard GA, a larger percentage of those (3.7%) are non-redundant. Given the goal of find as many distinct subset sets of features as possible, the POGA offers the best solutions among the three algorithms.

We collected all the feature subsets that yielded classifiers with 100% accuracy on the training set, and tested the accuracy of those classifiers on the test set (Table 2). The results show an average sensitivity of 71% and an average specificity of 83% indicating that the models tend to overfit the training set. While this clearly indicates a limitation with evaluation metric used in this study, it does not diminish the ability of the POGA to identify non-redundant feature subsets that satisfy the given evaluation metric.

Model	Sensitivity	Specificity
1	72% (48/67)	88% (38/43)
2	73% (49/67)	95% (41/43)
3	78% (52/67)	81% (35/43)
4	81% (54/67)	79% (34/43)
5	69% (46/67)	84% (36/43)
6	69% (46/67)	79% (34/43)
7	64% (43/67)	79% (34/43)
8	73% (49/67)	79% (34/43)
9	69% (46/67)	84% (36/43)
10	66% (44/67)	84% (36/43)
11	69% (46/67)	81% (35/43)

Table 2. Test-set classification accuracy of the distinct models found by the POGA that exhibited 100% accuracy on training set.

4. Discussion

This study investigated the effectiveness of the polyoptimizing genetic algorithms (POGA) on the problem of identifying non-redundant solutions to the feature subset selection problem. It was shown that the POGA identifies significantly more distinct feature subsets than either a standard GA or a random search, on a cancer diagnosis problem using proteomics data. The small number of distinct solutions found by random search indicates that non-redundant acceptable solutions are sparse in the search space. Further analysis in identifying the common features found in the non-redundant solutions found by POGA would be of interest. In addition, it would clearly be very valuable to investigate the biological significance of the components that are described in the mass spectrometry values. The underlying kNN classifier is suspected of overfitting the training data, and other classifiers need to be investigated. In general though, the goal to find a large number of distinct feature sets is clearly achieved using POGA.

There are numerous other problems in computational biology for which the POGA may be useful. For example, POGA could be applied for RNA structure prediction where the algorithm would search for alternative conformations that are energetically favorable. In phylogenetic analysis, POGA might be useful in finding alternative phylogenetic models for given sets of data. Further work will be aimed at characterizing the difficulty of various classes of problems with respect to polyoptimization, and to exploring and analyzing other penalty functions.

In summary, the polyoptimizing genetic algorithm appears to be a promising method for the selection of feature subsets to identify patterns in proteomics data that discriminate between cancer and non-cancer samples. Non-redundant feature subsets are particularly useful because they better represent the range of possible diagnostic features. Finally, it is clear that the classifier algorithm (kNN) needs further refinement to obtain more robust results.

Acknowledgements. Thanks to Dr. Lance Liotta for providing the proteomics data, and to Dr. Don Johann for advice on pre-processing the data set.

References

- [1] B. Domon and S. Broder, "Implications of new proteomics strategies for biology and medicine," *J Proteome Res*, vol. 3, pp. 253-60, 2004.
- [2] E. M. Posadas, B. Davidson, and E. C. Kohn, "Proteomics and ovarian cancer: implications for diagnosis and treatment: a critical review of the recent literature," *Curr Opin Oncol*, vol. 16, pp. 478-84, 2004.
- [3] E. F. Petricoin, A. M. Ardekani, B. A. Hitt, P. J. Levine, V. A. Fusaro, S. M. Steinberg, G. B. Mills, C. Simone, D. A. Fishman, E. C. Kohn, and L. A. Liotta, "Use of proteomic patterns in serum to identify ovarian cancer," *Lancet*, vol. 359, pp. 572-7, 2002.
- [4] W. Pusch, M. T. Flocco, S. M. Leung, H. Thiele, and M. Kostrzewa, "Mass spectrometry-based clinical proteomics," *Pharmacogenomics*, vol. 4, pp. 463-76, 2003.
- [5] L. Li, H. Tang, Z. Wu, J. Gong, M. Gruidl, J. Zou, M. Tockman, and R. A. Clark, "Data mining techniques for cancer detection using serum proteomic profiling," *Artif Intell Med*, vol. 32, pp. 71-83, 2004.
- [6] Q. N. Van, J. R. Klose, D. A. Lucas, D. A. Prieto, B. Luke, J. Collins, S. K. Burt, G. N. Chmurny, H. J. Issaq, T. P. Conrads, T. D. Veenstra, and S. K. Keay,

- "The use of urine proteomic and metabonomic patterns for the diagnosis of interstitial cystitis and bacterial cystitis," *Dis Markers*, vol. 19, pp. 169-83, 2003.
- [7] B. K. Lavine, C. E. Davidson, and W. S. Rayens, "Machine learning based pattern recognition applied to microarray data," *Comb Chem High Throughput Screen*, vol. 7, pp. 115-31, 2004.
- [8] S. C. Shah and A. Kusiak, "Data mining and genetic algorithm based gene/SNP selection," *Artif Intell Med*, vol. 31, pp. 183-96, 2004.
- [9] R. Jain and J. Mazumdar, "A genetic algorithm based nearest neighbor classification to breast cancer diagnosis," *Australas Phys Eng Sci Med*, vol. 26, pp. 6-11, 2003.
- [10] H. P. Chan, B. Sahiner, K. L. Lam, N. Petrick, M. A. Helvie, M. M. Goodsitt, and D. D. Adler, "Computerized analysis of mammographic microcalcifications in morphological and texture feature spaces," *Med Phys*, vol. 25, pp. 2007-19, 1998.
- [11] B. Sahiner, H. P. Chan, D. Wei, N. Petrick, M. A. Helvie, D. D. Adler, and M. M. Goodsitt, "Image feature selection by a genetic algorithm: application to classification of mass and normal breast tissue," *Med Phys*, vol. 23, pp. 1671-84, 1996.
- [12] B. Zheng, Y. H. Chang, X. H. Wang, W. F. Good, and D. Gur, "Feature selection for computerized mass detection in digitized mammograms by using a genetic algorithm," *Acad Radiol*, vol. 6, pp. 327-32, 1999.
- [13] M. Gletsos, S. G. Mougiakakou, G. K. Matsopoulos, K. S. Nikita, A. S. Nikita, and D. Kelekis, "A computer-aided diagnostic system to characterize CT focal liver lesions: design and optimization of a neural network classifier," *IEEE Trans Inf Technol Biomed*, vol. 7, pp. 153-62, 2003.
- [14] J. Bala, J. De Jong, J. Huang, H. Vafaie, and H. Wechsler, "Using Learning to Facilitate the Evolution of Features for Recognizing Visual Concepts," *Evolutionary Computation*, vol. 4, pp. 297-311, 1996.