Analysis of Gene Expression Data Using Evolutionary Multi-Agent System

GREGOR ŠTIGLIC, PETER KOKOL
Laboratory for System Design
University of Maribor
Smetanova 17, 2000 Maribor
SLOVENIA

Abstract: - This paper presents an application of Evolutionary Multi-Agent System (EMAS) to the analysis of gene expression data. Our goal is to find significant classification genes using simple classifiers that can be used by agents when exploring the gene expression database. This way we can get a small subset of significant features (genes) that can help us to identify the clinical state of the patient. The experiments show that agents improve their individual performance through evolution and collaboration with other agents. We present our results on two well-known publicly available gene expression problems.

Key-Words: - Classifier Systems, Genetic Algorithms, Multi-Agent Systems, Bioinformatics.

1 Introduction

Microarrays have become important tool in profiling global gene expression patterns. Their main advantages are: reproducibility and scalability of obtained data, short time of experiment and, of course, the large number of genes, the expression of which is measured. The technique of producing DNA microarrays is improving continuously. The results of improvement are better and more accurate gene expression databases. The problem in analysis of such databases is their multi-dimensionality, where we have large number of features (genes) and only a few instances (samples).

As a possible solution to the problem of classification in gene expression data, we propose a simple nearest neighbor classification method using only two features at a time. To narrow the large search space we employ the system of evolving agents searching for the best classifier. Agents in multi-agent systems are naturally led to building systems that adapt and learn through experience [1]. In our case agents can exchange information about the position of promising classification possibilities in two-dimensional feature space. Based on this fact two types of agents exist. The first type of agents are "static agents" which search in the proximity of the current best solution. The second type of agents are "dynamic agents" (we call them explorers) who are able to explore the search space without constraints. Using this technique we try to optimize current best solution and search for possible new promising points in the search space.

In the next section we describe the multi-agent environment and present basic agent properties and functions. After that a section with the results of multiple simulation cycles on two well-known databases are presented. In the final section we discuss about our method and possible further improvements of the multi-agent system for predictive gene discovery.

2 Evolutionary Multi-Agent System

We know two types of hybrid systems combining evolutionary computation (EC) and multi-agent systems (MAS). In the first case we use evolution to help an agent solving problems using evolutionary techniques. In the other case we try to combine EC and MAS even more closely by using agents as a population of the evolutionary environment. The key idea of our system, which follows the second mentioned case, is that besides interaction mechanisms typical for MAS (such as communication) agents are able to reproduce (generate new agents) and may die (be eliminated from the system) [2]. A decisive factor of the agent's activity is its fitness, expressed by its ability of finding the best combination of genes for classification. Each agent presents a pair of features in the final ensemble of classifying agents and therefore we can keep a high level of diversity in the ensemble [3, 4, 5].

Ho introduced the idea of ensembles of k-Nearest Neighbor (k-NN) classifiers where the variety in the ensemble is generated by selection of different feature subsets for each ensemble in [6]. Since she generates these feature subsets randomly she refers to these different subsets as random subspaces. She points to the ability of ensembles of k-NN classifiers based on different feature subsets to improve on the accuracy of individual k-NN classifiers because of the simplicity and accuracy of the k-NN approach. She shows that an ensemble of k-NN classifiers based on random subsets improves on the accuracies of individual classifiers on a hand-written character recognition problem.

2.1 Genotype and agent behavior

In the evolutionary system we have to follow the basic principles of the evolution theory like selection and inheritance. We can apply those principles to agents in form of:

- Death (elimination of noncontributing agents from the system) and
- Reproduction (production of a new agent from successful parents).

Basic behavior parameters of agent are encoded in genotype and are inherited from its parent(s). Those parameters can be modified using mutation and recombination. Mutation in biology and also in computer science is the local search part of the evolution. Therefore the mutation operator should not significantly change the genotype parameters. This is accomplished allowing smaller changes with higher probabilities and larger ones with less. In our case the mutation modification is distributed using a normal probability function as in research by Oechslein et al. [7].

In our system each agent's genotype consists of the following parameters:

- Exploring capability
- Speed
- Crowd factor
- Type of classifier

Exploring capability defines the level of agent's movement and can range from static to dynamic. Low exploring capability defines an agent as static, which means it will try to search for the best solution near the best-known solution in search space. In other case an agent has a freedom to explore in the undiscovered search space.

Speed defines agent's maximum movement capability when it moves in the search space.

Crowd factor is a parameter that enables control over crowding effect. Our system allows dividing search space in n^2 equal sectors. Crowd factor defines a maximum number of agents in the sector before an agent moves to other sector because of overcrowding.

Type of Classifier represents parameters of used classifier. In our case this is the parameter of k-NN classifier that defines how many neighbors will contribute to the final vote of the test case (see section 2.2).

Additional to agent's evolving parameters all agents have to follow some common rules that help them find better solutions in shorter time. Therefore each agent should follow these rules:

- Try to get an information in which areas of search space other agents found useful classification genes
- If the agent is “static” it should search near the best solutions
- Agents with the exploring factor between “Static” and “Dynamic” can search far away from the best solution, but should try to follow the horizontal or vertical line from the best solution (this way an agent is using one of the genes selected by the best classifier so far)

2.2. Fitness Function

The k-nearest neighbors (k-NN) algorithm is a simple but effective classification algorithm. It is widely used in machine learning and has numerous variations [8, 9]. Given a test sample of unknown label, it finds the *k* nearest neighbors in the training set using Euclidean distance (*d*) and assigns the label of the test sample according to the labels of those neighbors. We have used nine different types of classifiers using 1 to 9 nearest neighbors. To ensure a majority in the voting process we used vote weighting where neighbor's vote was weighted by its distance to the test sample. The weight given to each vote was *1/d*.

In all our tests we used leave-one-out cross-validation (LOOCV) for classifier accuracy estimation.

2.3 Datasets

Leukemia data. The original data comes from the research on acute leukemia by Golub et al. [10]. Dataset consists of 38 bone marrow samples from which 27 belong to acute lymphoblastic leukemia (ALL) and 11 to acute myeloid leukemia (AML). Each sample consists of probes for 6817 human genes. Golub et al used this dataset for training. Also 34 samples of testing data were used consisting of 20 ALL and 14 AML samples. Because we used leave-one-out cross-validation, we were able to make tests on three separate databases – training (38 samples), test (34) and combination of both (72).

Central Nervous System data. We used dataset C mentioned in the paper by Pomeroy et al. [11]. It includes gene expression data of 60 similarly treated patients from whom biopsies were obtained before receiving treatment of the medulloblastomas (a specific type of brain tumor). Patients who survived the treatment are labeled as "Class1" the others are labeled as "Class0". The data set contains 60 patient samples, 21 survived the treatment and 39 did not. There are 7129 genes in the dataset.

3 Experimental Results

Our multi-agent system was tested on four databases mentioned in section 2.3. In the preprocessing step we normalized all data to the interval [0, 1]. This is a necessary step when distance based classifier for fitness function computation is used.

Leukemia data. We made three tests on Leukemia database. Firstly we tested our system on training database that was collected by Golub et al. The time complexity of finding a solution with 100% was very low and besides that there were another two solutions that classified all cases correctly (Table 1).

#	Classification Accuracy	Classifier Used	Predictive Genes
1	100.00 %	3-NN	3301, 4847
2	100.00 %	7-NN	2020, 4783
3	100.00 %	1-NN	6225, 4050
4	97.37 %	5-NN	1882, 108
5	97.37 %	6-NN	1544, 4847

Table 1. Best classifying agents on Leukemia train data

The test was continued on the database used as test database in research by Golub containing 34 samples. We have to mention that this database was collected from different sources as the training database. Leave-one-out cross-validation was performed and we got five classifiers with error rate equal to zero (Table 2).

#	Classification Accuracy	Classifier Used	Predictive Genes
1	100.00 %	2-NN	2266, 6225
2	100.00 %	7-NN	1834, 1962
3	100.00 %	8-NN	6855, 6414
4	100.00 %	3-NN	6167, 6855
5	100.00 %	4-NN	2141, 2161

Table 2. Classification results on Leukemia test database

The final task in Leukemia database was applying our multi-agent system to the combined database that consisted of 72 gene expression samples. Even though the combined database samples originated from different sources, we got very promising results. The classifier that misclassified just one out of 72 samples was found (Fig. 1).

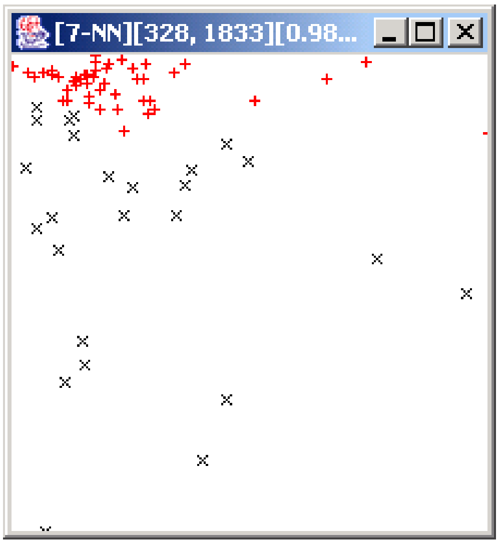

Fig. 1. Best classifier found on combined Leukemia database (72 samples) using LOOCV that classified 71/72 samples correctly. We can see that genes number 328 and 1833 were used in combination with 7-NN classifier.

Another four classifiers also proved very successful and achieved high accuracy rates (Table 3). Observing predictive genes used for classification we can spot a gene number 4847 that was used by agent 2 and 5. This points out to the high importance of the mentioned gene in classification of leukemia type.

#	Classification Accuracy	Classifier Used	Predictive Genes
1	98.61 %	7-NN	329, 1833
2	97.22 %	8-NN	4847, 257
3	95.83 %	4-NN	3252, 2146
4	95.83 %	1-NN	6041, 5106
5	95.83 %	7-NN	4847, 5062

Table 3. Performance of best five agents on Leukemia combined database

Central Nervous System data. When testing this database we could only compare our results to the original article by Pomeroy et al. where they achieved classification of 47 out of 60 samples correctly. This is probably due to the fact that this database is relatively new and is therefore not as widespread as Leukemia problem database. We were able to find three classifiers that classified 51 samples correctly and another two who did just slightly worse (Table 4). This way our system proved that searching through the whole search space can give very promising results using only two genes.

#	Classification Accuracy	Classifier Used	Predictive Genes
1	85.00 % (51/60)	7-NN	X91103_at [4785], HG1067-HT1067_r_at [6774]
2	85.00 % (51/60)	9-NN	M36089_at [1990], M55998_s_at [6322]
3	85.00 % (51/60)	5-NN	HG2797-HT2906_s_at [5790], U28055_at [5401]
4	83.33 % (50/60)	5-NN	Z33642_at [5123], D13900_at [218]
5	83.33 % (50/60)	3-NN	D29956_at [348], M33680_at [1962]

Table 4. Five best performing agents on Central Nervous System dataset including names of the genes used

4 Conclusion

Our aim was to prove that we can find good solutions to classification of gene expression data using very simple classifiers. Many currently used approaches in gene expression classification rely upon rank-based gene selection schemes that usually use statistic measures of correlation between samples of different classes. While they are good at identifying genes that are strongly correlated to the target class, rank-based methods tend to ignore correlations between genes. We show that it is possible to find promising nearest neighbor classifiers using just two genes. Another benefit here is the accuracy of metrics (in our case Euclidean) that become less sensitive as the dimensionality increases [12]. On the other hand we still have to be aware that we are searching in a huge search space and that there could very well be other relevant genes that are not employed in the final predictors.

In the future research we plan to emphasize the importance of composing a set of the best classifiers in an ensemble and classifying gene expression samples this way.

References:

[1] P.J. Modi and W.M. Shen, Collaborative Multiagent Learning for Classification Tasks, *Proceedings of the Fifth International Conference on Autonomous Agents*, 2001, pp. 37-38.

[2] K. Socha and M. Kisiel-Dorohinicki, Agent-based Evolutionary Multiobjective Optimisation, *Proceedings of CEC'02 - Congress on Evolutionary Computation*, Vol.1, 2002, pp. 109-114.

[3] G. Valentini and F. Masulli, Ensembles of learning machines, *Neural Nets WIRN Vietri*, Vol. 2486, 2002, pp. 3-19.

[4] L. Kuncheva, That Elusive Diversity in Classifier Ensembles, *Proceedings of Pattern Recognition and Image Analysis – IbPRIA*, Vol. 2652, 2003, pp. 1126-1138.

[5] P. Cunningham and J. Carney, Diversity versus Quality in Classification Ensembles Based on Feature Selection, *Proceedings of 11th European Conference on Machine Learning*, Vol. 1810, 2000, 109-116.

[6] T.K. Ho, Nearest Neighbours in Random Subspaces, Proceedings of 2nd International Workshop on Statistical Techniques in Pattern Recognition, 1998, pp. 640-648.

[7] C. Oechslein, A. Hörnlein and F. Klügl, Evolutionary Optimization of Societies in Simulated Multi-Agent Systems, *Modelling Artificial Societies and Hybrid Organizations*, 2000.

[8] R. Duda, *Pattern classification*, Wiley, 2001.

[9] C. Yeang, Molecular Classification of Multiple Tumor Types, *Bioinformatics*, Vol. 17, 2001, pp. 316-322.

[10] T.R. Golub et al., Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring, *Science*, Vol. 286, 1999, 531-537.

[11] S.L. Pomeroy et al., Prediction of Central Nervous System Embryonal Tumour Outcome Based on Gene Expression, *Letters to Nature, Nature*, Vol. 415, 2002, pp. 436-442.

[12] A.D. Keller, M. Schummer, L. Hood and W.L. Ruzzo, *Bayesian Classification of DNA Array Expression Data*, Technical Report, Department of Computer Science and Engineering, University of Washington, Seattle, 2000.