Novel Extension of $k - TSP$ Algorithm for Microarray Classification

Marcin Czajkowski and Marek Krętowski

Faculty of Computer Science, Białystok Technical University
Wiejska 45a, 15-351 Białystok, Poland
{mczajk,mkret}@wi.pb.edu.pl

Abstract. This paper presents a new method, referred as $Weight\ k - TSP$, which generates simple and accurate decision rules that can be widely used for classifying gene expression data. The proposed method extends previous approaches: TSP and $k-TSP$ algorithms by considering weight pairwise mRNA comparisons and percentage changes of gene expressions in different classes. Both rankings have been modified as well as decision rules, however the concept of "relative expression reversals" is retained. New solutions to match analyzed datasets more accurately were also included. Experimental validation was performed on several human microarray datasets and obtained results are promising.

1 Introduction

DNA chips provide tens of thousands of genes expression levels in single experiment [6,11]. Recently, microarray technology have been widely used to assist diagnosis and to discriminate cancer samples from normal ones [1,2,4]. Through analysis of gene expressions, some marker genes are found which later on can be used to build medical decision-support systems. However, finding a meaningful and robust classification rule is a real challenge, since in different studies of the same cancer, diverse genes consider to be marked [14]. The analysis of DNA microarrays poses also a large number of statistical problems, in which we can specify two main issues - dimensionality and redundancy.

DNA microarrays produce massive quantities of information which is difficult for analysis and interpretation. Unfortunately, the number of samples (denoted by N) comparing to number of features (genes, P) remains quite small and usually not exceeded one or two hundreds - it is the well known *"small N, large P problem"* [15,16]. Considering some dimensionality reduction (i.e. feature selection) seems to be reasonable, especially when it influences the model complexity [9]. Among other solutions, increasing sample size and joining datasets could be mentioned, nevertheless there are some difficulties in consistency of resulting data [21].

Analyzing many attributes can easily cause a classifier to overfit training data [5]. Hopefully, most of genes are known to be irrelevant for an accurate classification. That is why the gene selection prior the classification not only simplifies

N.T. Nguyen et al. (Eds.): IEA/AIE 2008, LNAI 5027, pp. 456–465, 2008.

calculations and model complexity, but also removes noise and decreases the computation time which is also relevant. Furthermore, experiments have shown that in most of cases the gene selection improves accuracy of the following classification [12].

Many standard classification approaches including statistical learning and pattern recognition methods are applied to microarray data [17]. However, the specificity of gene expression data causes rising new algorithms or extensions of the existing methods. Recently, many methods based on Support Vector Machines (SVM) [19], have been proposed [13,22]. Neural networks [3] and decision trees [8] are also commonly applied in microarray classification. Unfortunately, most of these methods generate very complex diagnostic rules based on many expression values. From a medical point of view they are very difficult to understand and interpret.

In this paper, we propose an extension of $k-TSP$ [18] (k-Top Scoring Pairs), which originates from the TSP algorithm [7]. The presented solution (denoted as $Weight\ k-TSP$) is focused on relative gene expression values and pairwise comparisons between two genes expression levels. By involving only few genes and generating simple decision rules, classification results can be easily analyzed and interpreted. In general, TSP and $k-TSP$ accuracy is relatively high, although these original algorithms, in contrast to $Weight\ k-TSP$, can not be tuned to reflect the specificity of different cancer datasets. It can be expected that this way the classification precision can be improved. The proposed method extends original solutions by considering weight pairwise mRNA comparisons and percentage changes of gene expressions in different classes. Both ranking have been modified as well as decision rules, however the concept of "relative expression reversals" have still retained.

The rest of the paper is organized as follows. In the next section both TSP and $k-TSP$ algorithms are briefly recalled and the $Weight\ k-TSP$ method is presented. In section 3 the proposed approach is experimentally validated on real microarray datasets. The paper is concluded in the last section and possible future works are presented.

2 A Family of TSP Algorithms

All variants from the family of the TSP methods are applied according to the classical supervised learning framework. In the first step based on the training dataset created from the gene expression profiles with the verified diagnosis, the top scoring pairs are generated. This process is illustrated in Fig. 1. In the next step, the obtained classifier can be applied to a new microarray sample with unknown decision. Only selected genes are analyzed and $TSP-$based prediction is made (Fig. 2).

2.1 TSP

Top Scoring Pairs (TSP) method was presented by Donald Geman [7] and is based on pairwise comparisons of gene expression values. Despite its simplicity

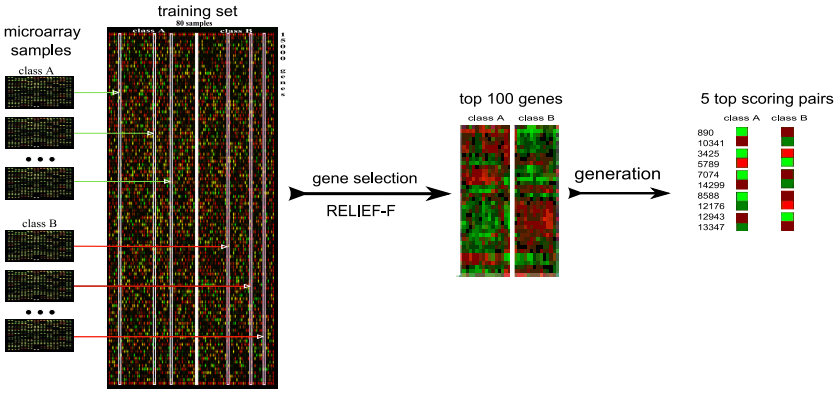

Fig. 1. Building TSP-based decision rules on the training dataset

comparing to other methods, classification rates for TSP are comparable or even exceed other classifiers [7]. Discrimination between two classes depends on finding matching pairs of genes that achieve the highest ranking value called "score".

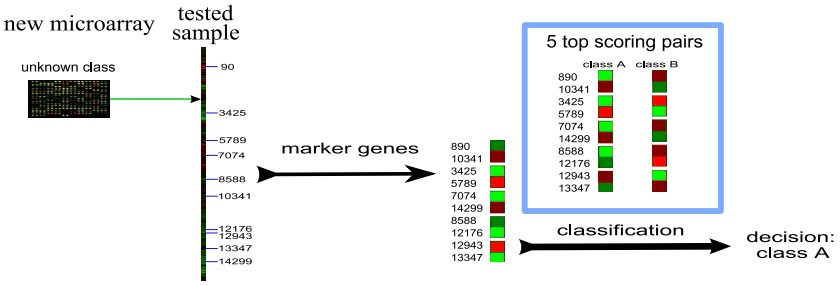

Fig. 2. Testing a new sample with the TSP classifier based on the selected genes

Considering the object containing P genes and N samples participating in the training microarray dataset, a $P \times N$ matrix X can be developed:

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1N} \\ x_{21} & x_{22} & \dots & x_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{P1} & x_{P2} & \dots & x_{PN} \end{pmatrix},$$

in which the expression value of i-th gene from the n-th sample is denoted by x_{ij}. Each row represents observations of a particular gene over N training samples, and each column represents a gene expression profile composed from P genes. Each profile has a true class label denoted $C_m \in C = \{C_1, \dots, C_M\}$. For the

simplicity of calculations it is assumed that there are only two classes ($M = 2$) and profiles with indices from 1 to N_1 ($N_1 < N$) belong to the first class (C_1) and profiles from range $\langle N_1 + 1, N\rangle$ to the second class (C_2).

TSP method focuses on gene pair matching (i, j) $(i, j \in \{1, \ldots, P\}, i \neq j)$ for which there is the highest difference in the probability of an event $x_{in} < x_{jn}$ $(n = 1, 2, \ldots, N)$ between class C_1 and C_2. For each pair of genes (i, j) two probabilities are calculated $p_{ij}(C_1)$ and $p_{ij}(C_2)$:

$$p_{ij}(C_1) = \frac{1}{|C_1|} \sum_{n=1}^{N_1} I(x_{in} < x_{jn}) \ , \tag{1}$$

$$p_{ij}(C_2) = \frac{1}{|C_2|} \sum_{n=N_1+1}^{N} I(x_{in} < x_{jn}) \ , \tag{2}$$

where $|C_m|$ denotes a number of profiles from class C_m and $I(x_{in} < x_{jn})$ is the indicator function defined as:

$$I(x_{in} < x_{jn}) = \begin{cases} 1, \text{ if } x_{in} < x_{jn} \\ 0, \text{ if } x_{in} \geq x_{jn} \end{cases} \ . \tag{3}$$

TSP is a rank-based method, so for each pair of genes (i, j) the "score" denoted Δ_{ij} is calculated:

$$\Delta_{ij} = |p_{ij}(C_1) - p_{ij}(C_2)| \ . \tag{4}$$

In the next step of the algorithm pairs with the highest score are chosen. There should be only one top pair in the TSP method, however it is possible that multiple gene pairs achieve the same top score. A secondary ranking based on the rank differences in each class and sample is used to eliminate draws.

For each top-scoring gene pair (i, j) the "average rank difference" in both C_1 and C_2 are computed and defined as:

$$\gamma_{ij}(C_1) = \frac{\sum_{n=1}^{N_1}(x_{in} - x_{jn})}{|C_1|} \ , \tag{5}$$

$$\gamma_{ij}(C_2) = \frac{\sum_{n=N_1+1}^{N}(x_{in} - x_{jn})}{|C_2|} \ . \tag{6}$$

Value of this second rank for each pair of genes (i, j) is defined as:

$$\tau_{ij} = |\gamma_{ij}(C_1) - \gamma_{ij}(C_2)| \ , \tag{7}$$

and the algorithm chooses a pair with the highest score.

The TSP classifier prediction is made by comparing the expression values from two genes (i, j) marked as "top scoring pair" with a test sample (i_{new}, j_{new}). If we observe that $p_{ij}(C_1) \geq p_{ij}(C_2)$ and $x_{inew} < x_{jnew}$, then TSP votes for class C_1, however if $x_{inew} \geq x_{jnew}$ then TSP votes for class C_2. An opposite

situation is when $p_{ij}(C_1) < p_{ij}(C_2)$, cause if $x_{inew} < x_{jnew}$ TSP votes for C_1 and if $x_{inew} \geq x_{jnew}$ TSP chooses C_2. In other words, if $p_{ij}(C_1) \geq p_{ij}(C_2)$ then:

$$y_{new} = h_{TSP}(new) = \begin{cases} C_1, \text{ if } x_{inew} < x_{jnew} \\ C_2, \text{ if } x_{inew} \geq x_{jnew} \end{cases} , \tag{8}$$

where h_{TSP} is a prediction result. Opposite situation is when $p_{ij}(C_1) < p_{ij}(C_2)$.

2.2 k-TSP

A $k-TSP$ classifier proposed by Aik Choon Tan [18] is a simple extension of the original TSP algorithm. The main feature that differ those two methods is the number of top scoring pairs included in final prediction. In the TSP method there can be only one pair of genes and in $k-TSP$ classifier the upper bound denoted as k can be set up before the classification. The parameter k is determined by a cross-validation and in any prediction the $k-TSP$ classifier uses no more than k top scoring disjoint gene pairs that have the highest score. Both primary and secondary rankings (equations (4) and (7)) remain unchanged.

The classification decision is made by comparing the expression values for each pair of genes (i_u, j_u) $(u = 1, \ldots, k)$ with a new test sample. The $k-TSP$ classifier denoted as h_{k-TSP} based on individual classifiers $h_u(new)$ employs a majority voting to obtain the final prediction of y_{new}, however each vote has the same wage:

$$y_{new} = h_{k-TSP}(new) = argmax \sum_{u=1}^{k} I(h_u(new) = C_i) , \tag{9}$$

where $C_i \in C$, and

$$I(h_u(new) = C_i) = \begin{cases} 1, \text{ if } h_u(new) = C_i \\ 0, \text{ otherwise} \end{cases} . \tag{10}$$

Meaning of $h_u(new)$ is the same as in the equation (8).

2.3 Weight k-TSP

In classification $Weight\ k-TSP$ all rankings have been changed, comparing to TSP and $k-TSP$. Therefore, the selection of top scoring pairs, and the prediction is different than in TSP or $k-TSP$ classifier. The main reason that motivates research on extensions of the $k-TSP$ algorithm is its limitation in finding appropriate top scoring pairs. There are two factors that could cause it. First factor that hampers finding appropriate top scoring pairs is connected to the relatively high computational complexity, which for these methods is $\theta(N * P^2)$. Microarray datasets contain huge amounts of data and the feature selection is usually applied before the actual classification. However, $k-TSP$ sensitivity to the feature selection and small size of datasets may effect rank

calculations and decrease accuracy. This is linked with the second factor which is a small number of features having similar expression values and being opposite to each other in different classes. Depending on data and preprocessing method, expression values can be very different and genes can have values for example from 0 to 10000.

Let us hypothetically assume that for some tested cancer samples two genes are responsible $G1$ and $G2$. Suppose that in healthy samples from training dataset genes the expression levels are in a range $G1 < 0, 50 >$, $G2 < 6000, 9000 >$ and in cancer sample: $G1 < 1000, 2000 >$, $G2 < 2500, 4000 >$. Method $k-TSP$ and TSP will never mark these genes as "top scoring pair" because among all classes $G1$ is smaller then $G2$. It might choose them with other genes, by making more top pairs, but it will not be so accurate and can cause problems in interpretability. A solution for this type of situations could be comparison of percentage changes of gene expression in pairs among different classes.

Considering that S represents average values quotient genes in each pair from K training samples (K determined by a cross-validation). For each pair of genes (i, j) $(i, j \in \{1, \ldots, P\}, i \neq j)$ single element from S can be described as:

$$S_{ij} = \frac{\sum_{m=1}^{K} x_{im}/x_{jm}}{K} \ . \tag{11}$$

$Weight\ k-TSP$ is focused on finding pairs of genes (i, j) that have the highest difference in probability of event $\{x_{in}/x_{jn} < S_{ij}\}$ $(n = 1, 2, \ldots, N)$ between class C_1 and C_2. For each pair of genes (i, j) two probabilities are calculated $p_{ij}(C_1)$ and $p_{ij}(C_2)$:

$$p_{ij}(C_1) = \frac{1}{|C_1|} \sum_{n=1}^{N_1} I(x_{in}/x_{jn} < S_{ij}) \ , \tag{12}$$

$$p_{ij}(C_2) = \frac{1}{|C_2|} \sum_{n=N_1+1}^{N} I(x_{in}/x_{jn} < S_{ij}) \ , \tag{13}$$

where $I(x_{in}/x_{jn} < S_{ij})$ is the indicator function defined as:

$$I(x_{in}/x_{jn} < S_{ij}) = \begin{cases} 1, \text{ if } x_{in}/x_{jn} < S_{ij} \\ 0, \text{ if } x_{in}/x_{jn} \geq S_{ij} \end{cases} \ . \tag{14}$$

Similarly to $k-TSP$, $Weigh\ k-TSP$ is a rank-based method, so for each pair of genes (i, j) the calculated score denoted Δ_{ij}, where:

$$\Delta_{i,j} = |p_{ij}(C_1) - p_{ij}(C_2)| \ . \tag{15}$$

Secondary ranking which used in TSP and $k-TSP$ in case of an equal score has also been modified, since there are exceptions that could discriminate genes with low values of expression. Again, considering the assumption that for some tested cancer two pairs of genes denoted $F1$ and $F2$, having the same primary ranking, are responsible. Suppose in all training dataset samples, genes expression values

for these pairs are in range $F1 < 0, 1000 >$, $F2 < 3000, 9000 >$. Pair $F1$ may create more interesting decision rules and improve prediction than $F2$ because genes have changed their expression values from low level to high. In pair denoted $F2$, genes expression levels have stayed in high or very high range among all samples and should not be classified as a top scoring pair. However TSP and $k-TSP$ algorithm would choose $F2$ pair because of its rankings that compares only value differences. Proposed modifications do not discriminate genes with low values of expression and would mark $F1$ as a top scoring pair. To eliminate problem of very small values (near 0) to dividend divisor - average S was added, so all values belong to range $\langle 0, 1\rangle$:

$$\gamma_{ij}(C_m) = \frac{\sum_{n \in C_m} \frac{x_{in}/x_{jn}}{S_{ij}+x_{in}/x_{jn}}}{|C_m|} \ , \tag{16}$$

where $|C_m|$ denote number of profiles in C_m. Value of this second rank for each par of genes (i, j) is defined as:

$$\tau_{ij} = |\gamma_{ij}(C_1) - \gamma_{ij}(C_2)| \ . \tag{17}$$

Similarly to previous solution the algorithm chooses pair with the largest score.

Final prediction is similar to presented earlier in TSP and $k-TSP$ algorithm and is based on voting. However unweighed majority voting was extended by adding weight and mixed decision rules, which for some datasets improved accuracy. The equations (9) and (10) for unweighed majority voting are still valid in $Weight\ k-TSP$, however there were changes in $h_u(new)$.
If $p_{ij}(C_1) \geq p_{ij}(C_2)$ then:

$$h_u(new) = \begin{cases} C_1, \text{ if } x_{inew}/x_{jnew} < S_{ij} \\ C_2, \text{ if } x_{inew}/x_{jnew} \geq S_{ij} \end{cases} \ , \tag{18}$$

where x_{inew} denote an expression level of i-th gene from sample named new. Opposite situation is when $p_{ij}(C_1) < p_{ij}(C_2)$. In wage voting - equation (9) also has changed.
If $p_{ij}(C_1) > p_{ij}(C_2)$ then:

$$I_{wg}(h_u(new) = C_i) = \begin{cases} \frac{S_{ij}}{S_{ij}+x_{inew}/x_{jnew}}, \text{ if } h_u(new) = C_i \\ 0, \text{ otherwise} \end{cases} \ . \tag{19}$$

If $p_{ij}(C_1) \leq p_{ij}(C_2)$, wage changes:

$$I_{wg}(h_u(new) = C_i) = \begin{cases} \frac{x_{inew}/x_{jnew}}{S_{ij}+x_{inew}/x_{jnew}}, \text{ if } h_u(new) = C_i \\ 0, \text{ otherwise} \end{cases} \ . \tag{20}$$

3 Experimental Results

3.1 Setup

Performance of $Weight\ k-TSP$ classifier was investigated on public available microarray datasets described in Table 1. All datasets come from Kent Ridge

Table 1. Binary class gene expression datasets

Datasets	**Genes**	**Class 1**	**Class 2**
Leukemia	7129	47(all)	25(aml)
Breast Cancer	24481	51(n)	46(r)
CNS	7129	39(f)	21(s)
Colon Tumor	2000	22(n)	40(t)
Prostate Cancer	12600	59(n)	77(t)
Lung Cancer	12533	31(mpm)	150(adca)
Lymphoma	4026	23(a)	24(g)

Table 2. Classifiers accuracy results tested with WEKA software

	Datasets							
Classifiers	ALL-AML	BC	CNS	CT	PC	LC	DLBCL	**Average**
Naive Bayes	97.05	78.94	76.66	74.19	23.52	99.32	97.87	78,22
RBF Network	97.05	78.94	75.00	85.48	79.41	98.65	95.74	**87,18**
SMO	94.11	68.42	81.66	88.70	26.47	99.32	97.87	79,5
AdaBoostMI	91.17	63.15	73.33	79.03	44.11	81.87	91.48	74,87
Bagging	94.11	63.15	76.66	82.25	41.17	95.97	85.10	76,91
J48	91.17	47.36	75.00	79.03	29.41	77.18	80.85	68,57
Random Forest	73.52	89.47	76.66	85.48	73.52	96.64	97.87	84,73
JRip	91.17	52.63	70.00	72.58	29.41	97.98	78.72	70,35

Bio-medical Dataset Repository and are related to studies of human cancer, including: leukemia, breast cancer, central nervous system, colon tumor, prostate cancer, lung cancer and lymphoma. Comparison $Weight\ k-TSP$ with several popular classifiers like: SMO, NB, RBF Network, ADA, Bagging, J48, Random Forest and JRip was performed on WEKA [20] - data mining software. $Weight\ k-TSP$ and TSP-family classifiers were implemented and tested on the MLP software. Before classification, gene selection Relief-F [10] with neighbours k=10 and sample size 100 was applied.

Typical normalization was also performed: $a'_i = \frac{(a_i - a_{i_{min}})}{(a_{i_{max}} - a_{i_{min}})}$, where a_i is an expression level of i-th gene, $a_{i_{max}}$ is maximal and $a_{i_{min}}$ is a minimal value in dataset. For all classifiers default parameters, proposed in WEKA and recommend in [18]; typically 10 runs and 10 folds in crossvalidation were used.

3.2 Outcome

Results of experiments are shown in Table 1 and Table 2. Table 2 summarizes the results for 8 different classifiers on 7 binary classification problems, tested on WEKA software. Based on the results: the RBF Network (87,18) and Random Forest (84,73) yield the best accuracy averaged over the 7 problems and SVM (79,50) outperformed other methods in 4 of the 7 cases. However in terms of

Table 3. Classifiers accuracy results tested with MLP software

Classifiers	Datasets							Average
	ALL-AML	BC	CNS	CT	PC	LC	DLBCL	
TSP	85,29	73,68	57,49	79,59	76,47	97,98	86,30	79,54
std. dev.	1,49	0,00	4,17	2,13	0,00	0,00	2,86	1,69
k-TSP	92,05	78,94	**59,33**	84,66	82,35	98,25	94,35	84,28
std. dev.	2,79	0,00	5,16	2,86	0,00	0,56	2,13	1,76
Weight TSP	91,17	73,68	58,33	84,73	82,35	84,63	86,40	81,68
std. dev.	0,00	0,00	5,77	2,54	0,00	1,49	2,78	1,64
Weight k-TSP	**93,52**	**84,21**	58,50	**87,61**	**89,11**	**98,38**	**97,95**	**87,04**
std. dev.	1,86	1,49	2,65	1,72	3,41	0,56	1,34	1,83

efficiency and simplicity $Weight\ k-TSP$ and TSP-family methods are superior. Table 3 summarizes the results for TSP-family classifiers on the 7 binary classification problems - implemented and tested on MLP software.

To appropriately compare $Weight\ k-TSP$ to TSP method, $Weight\ TSP$ was implemented. It can be observed that $Weight\ TSP$ and $Weight\ k-TSP$ accuracy is slightly increased in almost every tested database and in terms of efficiency and simplicity it is similar to TSP-family classifiers.

4 Conclusion and Future Works

This paper presents extension of TSP and $k-TSP$ classifier called $Weight\ k-TSP$. Main concept of this method is weight pairwise mRNA comparisons and percentage changes in gene expression in different classes. By concentrating in $Weight\ k-TSP$ on a relative gene expression changes between tested samples and by building novel prediction rules - the increase of the average classification accuracy was observed. In terms of efficiency and simplicity $Weight\ k-TSP$ is similar to TSP-family methods however it is superior to other classifiers like SVM and Naive Bayes.

TSP-family classifiers and several different machine learning methods was compared on 7 gene expression datasets involving human cancers. From results, $Weight\ k-TSP$ perform approximately the same as the RBF Network classifier and it was more accuracy then other methods on these data, however $Weight\ k-TSP$ provides simple decision rules usually involving few genes which have clear biological connections to adequate cancer types.

Furthermore, many possible improvement for $Weight\ k-TSP$ still exist. One direction of current research is to use discrete data of "present", "absent" or "marginal" gene expression values. Tests for finding marker genes that occurred most often in top pairs are also performed and superficial results are very promising.

Acknowledgments. This work was supported by the grant W/WI/5/05 from Białystok Technical University.

References

1. Alizadeh, A.A.: Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature 403, 503–511 (2000)
2. Bittner, M., Meltzer, P., Chen, Y.: Molecular classification of cutaneous malignant melanoma by gene expression profiling. Nature 406, 536–540 (2000)
3. Cho, H.S., Kim, T.S., Wee, J.W.: cDNA Microarray Data Based Classification of Cancers Using Neural Networks and Genetic Algorithms. Nanotech 1 (2003)
4. Dhanasekaran, S.M.: Delineation of prognostic biomarkers in prostate cancer. Nature 412, 822–826 (2001)
5. Duda, R.O., Heart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. J. Wiley, New York (2001)
6. Duggan, D.J., Bittner, M., Chen, Y., Meltzer, P., Trent, J.M.: Expression profiling using cDNA microarrays. Nature Genetics Supplement 21, 10–14 (1999)
7. Geman, D., d'Avignon, C., Naiman, D.Q., Winslow, R.L.: Classifying Gene Expression Profiles from Pairwise mRNA Comparisons. Statistical Applications in Genetics and Molecular Biology 3(1), 19 (2007)
8. Grześ, M., Kręłowski, M.: Decision Tree Approach to Microarray Data Analysis. Biocybernetics and Biomedical Engineering 27(3), 29–42 (2007)
9. Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning. Springer, Heidelberg (2001)
10. Kononenko, I.: Estimating Attributes: Analysis and Extensions of RELIEF. In: European Conference on Machine Learning, pp. 171–182 (1994)
11. Lipschutz, R.J., Fodor, S.P.A., Gingeras, T.R., Lockhart, D.J.: High density synthetic oligonucleotide arrays. Nature Genetics Supplement 21, 20–24 (1999)
12. Lu, Y., Han, J.: Cancer classification using gene expression data. Information Systems 28(4), 243–268 (2003)
13. Mao, Y., Zhou, X., Pi, D., Sun, Y., Wong, S.T.C.: Multiclass Cancer Classification by Using Fuzzy Support Vector Machine and Binary Decision Tree With Gene Selection. J. Biomed. Biotechnol., 160–171 (2005)
14. Nelson, P.S.: Predicting prostate cancer behavior using transcript profiles. Journal of Urology 172, S28–S32 (2004)
15. Sebastiani, P., Gussoni, E., Kohane, I.S., Ramoni, M.F.: Statistical challenges in functional genomics. Statistical Science 18(1), 33–70 (2003)
16. Simon, R., Radmacher, M.D., Dobbin, K., McShane, L.M.: Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification. Journal of the National Cancer Institute 95, 14–18 (2003)
17. Speed, T.: Statistical Analysis of Gene Expression Microarray Data. Chapman & Hall/CRC, New York (2003)
18. Tan, A.C., Naiman, D.Q., Xu, L., Winslow, R.L., Geman, D.: Simple decision rules for classifying human cancers from gene expression profiles. Bioinformatics 21, 3896–3904 (2005)
19. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)
20. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
21. Tan, X.L., Naiman, A.C., Geman, D.Q., Windslow, D., Robust, R.L.: prostate cancer marker genes emerge from direct integration of inter-study microarray data. Bioinformatics 21(20), 3905–3911 (2005)
22. Zhang, C., Li, P., Rajendran, A., Deng, Y., Chen, D.: Parallelization of multicategory support vector machines (PMC-SVM) from classifying microarray data. BMC Bioinformatics (2006)