ZESZYTY NAUKOWE POLITECHNIKI BIAŁOSTOCKIEJ 2009
Informatyka – Zeszyt 4

Marcin Czajkowski[1], Marek Krętowski[1]

AN EXTENSION OF TSP-FAMILY ALGORITHMS FOR MICROARRAY CLASSIFICATION

Abstract: Classification of microarray data and generation of simple and efficient decision rules may be successfully performed with Top Scoring Pair algorithms. *TSP*-family methods are based on pairwise comparisons of gene expression values. This paper presents a new method, referred as *Linked TSP* that extends previous approaches $k-TSP$ and *Weight* $k-TSP$ algorithms by linking top pairwise mRNA comparisons of gene expressions in different classes. Opposite to existing *TSP*-family classifiers, the proposed approach creates decision rules involving single genes that most frequently appeared in top scoring pairs. Motivation of this paper is to improve classification accuracy results and to extract simple, readily interpretable rules providing biological insight as to how classification is performed. Experimental validation was performed on several human microarray datasets and obtained results are promising.

Keywords: pairwise classification, decision rules, microarray, gene expression

1. Introduction

DNA chips technology has given rise to the study of functional genomics [3,14]. The entire set of genes of an organism can be microarrayed on an area not greater than 1 cm^2 and enable to analyze hundred of thousands of expression levels simultaneously in a single experiment [7]. Microarray technology make possible comparisons of gene expressions levels and computational analysis allows classification samples by their mRNA expression values.

Nowadays, DNA chips are widely used to assist diagnosis and to discriminate cancer samples from normal ones [2,6]. Extracting accurate and simple decision rules that contain marker genes is of great interest for biomedical applications. 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 [16].

Typical statistical problem that often occurs with microarray analysis is dimensionality and redundancy. In particular, we are faced with the *"small N, large P problem"* [17,18] of statistical learning because the number of samples (denoted by N)

[1] Faculty of Computer Science, Bialystok Technical University, Poland

comparing to number of features/genes (P) remains quite small as N usually does not exceeded one or two hundreds where P is usually several thousands. This may influence the model complexity [11] and cause the classifier to overfit training data. Considering some dimensionality reduction (i.e. feature selection) seems to be reasonable as most of the genes are known to be irrelevant for classification and prediction. Applying gene selection prior classification [15] may simplify calculations, model complexity and often improve accuracy of the following classification.

Recently, many new solutions based on classification approaches including statistical learning and pattern recognition methods are applied to microarray data [19,10]. However most of them generate very complex decision rules that are very difficult or even impossible to understand and interpret. This is a trade-off between credibility and comprehensibility of the classifiers [20].

In this paper, we would like to propose an alternative approach for TSP-family classifiers. The presented solution (denoted as *Linked* TSP) may be applied to original TSP classifier [8] or its extensions: $k-TSP$ [20] and *Weight* $k-TSP$ [5]. In our research we have experimentally observed that some genes, more often to the others, appear in top pairs calculated by one of these TSP algorithms. This may suggest that some genes from the list of top pairs more accurate discriminate cancer samples from normal one. Our method is focused on finding predominatingly genes from calculated top pairs of genes. We believe that these approach will simplify decision rules without reducing classification accuracy or even improve it for some datasets.

The rest of the paper is organized as follows. In the next section TSP-family algorithms are briefly recalled. Section 3 describes proposed solution - *Linked* TSP. In section 4 the presented approach is experimentally validated on real microarray datasets. The paper is concluded in the last section and possible future works are suggested.

2. A Family of TSP Algorithms

TSP-family methods are applied according to the classical supervised learning framework. In the first step the top scoring pairs are generated from the training dataset. This process is illustrated in Fig. 1. In the second step, the obtained classifier can be applied to a new microarray sample with unknown decision class. Only selected genes called "marker genes" are analyzed and used in TSP prediction (Fig. 2). *Linked* TSP classifier uses only first step of TSP-family methods to obtain sorted (decreasingly by significance) list of gene pairs generated by these algorithms.

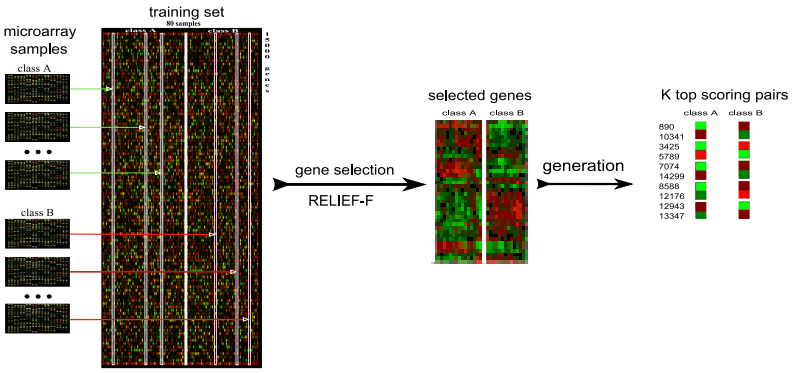

Fig. 1. Building *TSP*-based decision rules on the training dataset

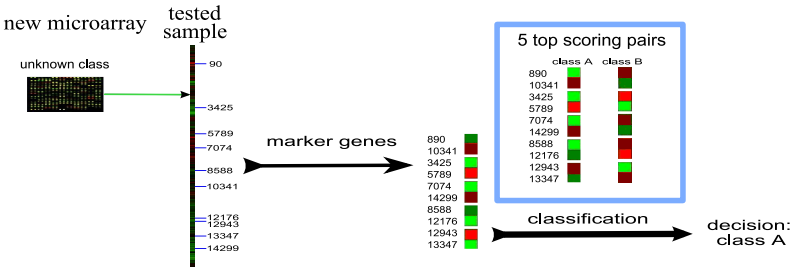

Fig. 2. Testing a new sample with the *TSP* classifier based on the selected genes

2.1 Top Scoring Pair

Top Scoring Pair (*TSP*) method was presented by Donald Geman [8] and is based on pairwise comparisons of gene expression values. Despite its simplicity comparing to other methods, classification rates for *TSP* are comparable or even exceeds other classifiers [8]. Discrimination between two classes depends on finding pairs of genes that achieve the highest ranking value called "score".

Consider a gene expression profile consisting of P genes and N samples participating in the training microarray dataset. These data can be represent as a $P \times N$ matrix X:

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

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 $C_m \in C = \{C_1, \ldots, C_M\}$. 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 p 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-Top Scoring Pairs

A *k-Top Scoring Pairs* $(k-TSP)$ classifier proposed by Aik Choon Tan [20] 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 class prediction 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 partial 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 $C_i \in C = \{C_1, \ldots, C_M\}$, 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* $k-TSP$ proposed by us [5] 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 was its limitation in finding appropriate top scoring pairs. There were 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 connected with the second factor which is a small number of features having similar expression values and being opposite to each other in different classes.

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

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

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.

Similar to $k-TSP$, *Weigh* $k-TSP$ is a rank-based method, so for each pair of genes (i,j) score is calculated and the algorithm chooses the pairs with the highest one. Final prediction is similar to TSP or $k-TSP$ methods and involves voting. However unweighed majority voting was extended by adding weight and mixed decision rules to improve accuracy for different types of datasets.

3. *Linked TSP*

Concept of *Linked* TSP has arisen during our tests of *Weight* $k-TSP$ algorithm. We have observed that some genes, much more often to other ones, join in pairs that achieve high scores in $k-TSP$ and *Weight* $k-TSP$ rankings. This may suggest that these genes more precisely discriminate cancer samples from normal ones.

The presented method can work with any standard TSP-family classifier or different approach that calculates pairs of genes. Proposed algorithm require sorted (decreasingly by significance) list of gene pairs denoted as L from TSP methods. Idea of

approach is to discover list (denoted as G) of most frequently appearing genes from list L. Let the parameter k alike in $k-TSP$ be determined by a cross-validation and stands for the upper bound on the number of top genes to be included in the final *Linked TSP* classifier.

In the first step, we seek for genes g ($g \in G = \{g_1, \ldots, g_k\}$) that most frequently appears in the list L_T ($L_T \in L$). List denoted as L_T contains top T pairs of genes from L that have the highest TSP ranking. For each gene in L_T ranking is calculated based on a appearing frequency in the rest $T-1$ pairs. After finding gene with the highest ranking score denoted as g_1, all pairs from list L_T containing this gene are removed and first step is repeated until k top genes are found.

Next step (which is optional) uses permutation of G list to remove irrelevant genes. All top genes from list G are used to test training sample - at first individually and later in double and triple size sets. At each step the worst gene or set of genes is removed. To prevent classifier over-fitting, internal 10-folds cross-validation is performed and to ensure stable results - average score of 10 runs is calculated.

Finally, algorithm determine (m-times by internal cross-validation) final number of top k genes from list G that will be used in prediction - similar to TSP method. Concept of choosing top genes is presented in Algorithm 1.

Algorithm 1 Calculate the list of $G = \{g_1, \ldots, g_k\}$ top genes

Require: Maximum number of genes to search: $k \geq 0$
Ensure: *Linked TSP* classifier
  for cross-validation - repeat m times **do**
    Make an ordered list L of all of the gene pairs from highest to lowest score using TSP methods.
    for i = 1 to k **do**
      Make a list L_T that contains top T gene pairs from the L list
      for each gene in L_T **do**
        Calculate the number of pairs that involve this gene
      end for
      Add the most common gene g_i to the G list
      Remove every pair from L that involves g_i
      Compute the error rate for the classifier based on genes in list G
    end for
  end for
  Select the value of K whose average classification rate over m loops is optimal.
  [optional] Remove genes from list G that have the lowest accuracy through internal permutation
  Return *Linked TSP* classifier

List of genes denoted as G that will be used in *Linked TSP* classifier is usually similar to the ones obtained from TSP-family classifiers. Often the top genes from

each pairs that were used in TSP prediction also built *Linked TSP* classifier. However, no pairs of genes and the concept of "*relative expression reversals*"[8][20] in *Linked TSP* prediction model required new method to classify data.

We would like to propose rank prediction that will be composed of 3 simple steps. Let's assume that prediction model require k genes denoted as $g_1, g_2, \ldots, g_k$ from list G. Genes are marked separately in each classes $C_m \in C = \{C_1, \ldots, C_M\}$, where M denotes alike in TSP number of classes. Ranking for each class is presented in the equation (12).

$$Rank(C_m) = \sum_{i=1}^{k} (I_1(g_i) + I_2(g_i) + I_3(g_i)), \tag{12}$$

where:

$$I_1(g_i) = \begin{cases} \tau_1, \ where\ g_{min_i}(C_m) \leq g_i \leq g_{max_i}(C_m) \\ 0, \ otherwise \end{cases}$$

$$I_2(g_i) = \begin{cases} \tau_2, \ where\ (g_{min_i}(C_m) \leq g_{min_i}(C \setminus C_m)\ and\ g_i \leq g_{min_i}(C_m)) \\ \qquad or\ (g_{max_i}(C_m) \geq g_{max_i}(C \setminus C_m)\ and\ g_i \geq g_{max_i}(C_m)) \\ 0, \ otherwise \end{cases}$$

$$I_3(g_i) = \begin{cases} \tau_3, \ where\ |g_i - \bar{g}_i(C_m)| < |g_i - \bar{g}_i(C \setminus C_m)| \\ 0, \ otherwise \end{cases},$$

where: $g_{min_i}(C_m)$, $g_{max_i}(C_m)$, $\bar{g}_i(C_m)$ denote minimum, maximum and average value of expression level of i-gene in training dataset that was chosen for prediction, in class C_m.
Score achieved by genes from list G determines prediction result. Tested sample is classified to the class that has the highest average score through all K genes. Ranking prediction may be adjusted to analyzed dataset by parameter τ in each step.

4. Experimental Results

Performance of *Linked TSP* classifier was investigated on public available microarray datasets described in Table 1. We have comprised accuracy and size of *Linked TSP* method with TSP-family algorithms. In addition, other popular classifiers were analyzed and all results were enclosed in Tables from 2 to 5.

Table 1. Kent Ridge Bio-medical gene expression datasets

	Datasets	Abbreviation	Attributes	Training Set	Testing Set
1	Breast Cancer	BC	24481	34/44	12/7
2	Central Nervous System	CNS	7129	21/39	-
3	Colon Tumor	CT	6500	40/22	-
4	DLBCL Standford	DS	4026	24/23	-
5	DLBCL vs Follicular Lymphoma	DF	6817	58/19	-
6	DLBCL NIH	DN	7399	88/72	30/50
7	Leukemia ALL vs AML	LA	7129	27/11	20/14
8	Lung Cancer Brigham	LCB	12533	16/16	15/134
9	Lung Cancer University of Michigan	LCM	7129	86/10	-
10	Lung Cancer - Totonto, Ontario	LCT	2880	24/15	-
11	Prostate Cancer	PC	12600	52/50	27/8

4.1 Datasets

Datasets came from Kent Ridge Bio-medical Dataset Repository [12] and are related to studies of human cancer, including: leukemia, colon tumor, prostate cancer, lung cancer, breast cancer etc. Typical 10-folds crossvalidation was applied for datasets that were not arbitrarily divided into the training and the testing sets. To ensure stable results for all datasets average score of 10 runs is shown. All data was not transformed, no standardization or normalization was performed.

4.2 Setup

Comparison of *Linked TSP* was performed with original $k-TSP$ and *Weight* $k-TSP$ algorithms. Maximum number of gene pairs k used in all prediction models was default (equal 10) through all datasets. *Linked TSP* method has this number doubled because it calculates single not pairs of genes. Default values for the prediction rankings were set decreasingly: $\tau_1 = 1$, $\tau_2 = 0.5$, $\tau_3 = 0.1$ and the list L_T default size equal 100 top pairs.

All classifications were preceded by a step known as feature selection where a subset of relevant features is identified. We decided to use popular for microarray analysis method Relief-F [13] with default number of neighbors (equal 10) and 1000 features subset size.

Linked TSP accuracy was also compared to other popular classifiers that generates comprehensible decision rules. Comparison *Linked TSP* to other classifiers was performed with:

– Several popular decision trees:

1. AD Tree (AD) - alternating decision tree
2. BF Tree (BF) - best-first decision tree classifier
3. J48 Tree (J48) - pruned C4.5 decision tree
4. Random Tree (RT) - algorithm constructing a tree that considers K randomly chosen attributes at each node
5. Simple Cart (CT) - CART algorithm that implements minimal cost-complexity pruning

– Rule classifier:
  6. JRip (JR) - rule learner - Repeated Incremental Pruning to Produce Error Reduction (RIPPER)
– Ensemble decision trees:
  7. Bagging (BG) - reducing variance meta classifier
  8. Adaboost (ADA) - boosting algorithm using Adaboost M1 method

The main software package used in the comparison experiment for these classifiers is Weka [22]. Classifiers were employed with default parameters through all datasets. Experimental results on tested datasets are described in Tables 2 and 5.

4.3 Outcome

Table 2 enclose *Linked TSP* comparison results to *TSP*-family classifiers. Since first step *Linked TSP* algorithm may involve list of best gene pairs calculated from different *TSP* methods we would like to present results separately. Let *Linked* $k-TSP$ denote results of *Linked TSP* classifier built on $k-TSP$ method and *Linked Weight* $k-TSP$ represents *Linked TSP* built on *Weight* $k-TSP$ method.

Results enclosed in Table 2 reveal that *Linked TSP* based on $k-TSP$ yield the best averaged accuracy (78.59) over 11 classification problems. *Linked TSP* based on *Weight* $k-TSP$ achieved second score and also improve *Weight* $k-TSP$ classifiers. We can observe that in 7 over 11 datasets *Linked TSP* has the highest accuracy. We believe that achieved results are promising and proposed approach may compete and be an alternative for $k-TSP$ or *Weight* $k-TSP$ methods. However in our opinion *Linked TSP* can not replace other *TSP* classifiers because each method generates significant rules that can capture interactions in datasets from different aspects. In our experiments around 50% identical genes were used by all three classifiers however the different prediction model influence the final score.

Worth to notice in Table 2 is the number of genes used in classification. It is significantly smaller in *Linked TSP* (6.57 and 11.38) to the $k-TSP$ (14.21) and *Weight* $k-TSP$ (14.85) methods. Therefore presented solution simplify decision rules and uses only significant genes in classification and prediction.

Table 2. Comparison of *Linked TSP* accuracy and size with original $k-TSP$ and *Weight* $k-TSP$ classifiers. The highest classifiers accuracy for each dataset was bolded.

Datasets	**Classifiers**							
	k-TSP		Weight k-TSP		Linked k-TSP		Linked Weight k-TSP	
	accuracy	size	accuracy	size	accuracy	size	accuracy	size
1. BC	74.73	17.20	47.36	18.00	**88.42**	8.10	51.57	12.20
2. CNS	59.49	17.96	51.16	17.84	55.66	7.96	**65.33**	18.37
3. CT	76.83	10.40	85.47	5.08	82.19	8.37	**86.73**	18.13
4. DS	78.90	14.91	62.20	17.68	65.65	8.37	**86.95**	15.52
5. DF	**91.44**	17.44	81.41	15,71	87.80	8.44	87,76	13.11
6. DN	56.37	17.60	52.25	17.60	**70.00**	9.00	51.62	20.00
7. LA	**94.11**	18.00	93.23	17.60	91.17	3.00	91.17	2.00
8. LCB	77.18	2.00	96.71	16.80	**100.00**	3.00	91.94	2.00
9. LCM	95.62	15.48	98.53	15.34	97.26	3.71	**99.26**	2.00
10. LCT	73.41	12.44	**89.50**	4.76	67.00	8.28	74.50	13.27
11. PC	63.66	12.88	**73.66**	16.92	59.33	4.08	59.33	8.57
Average score	76.52	14.21	75.59	14.85	**78.59**	6.57	76.92	11.38

Table 3. Marker genes used in Lung Cancer Brigham dataset classification

Classifiers	k-TSP	Weight k-TSP	Linked k-TSP	Linked Weight k-TSP
Accuracy	77.18%	96.71%	100.00%	91.94%
Genes	31393_r_at **33499_s_at**	more than 20 genes	37947_at **33499_s_at_at** 36528	37013_at 35236_g_at

In one of our experiments we tested Lung Cancer dataset (*LCB*) from "Brigham and Women's Hospital" Harvard Medical School. We managed to achieve perfect accuracy with only 3 genes to other classifiers results (90-99%) described in [9] that used 6 features and more. In Table 3 we have enclosed genes that build tested classifiers and we have bolded identical ones. We may observe that using more features in this case increased classifiers accuracy. However involving too many genes in decision model makes the method harder to understand by human experts.

Higher number of genes used in prediction not always cause increase of an accuracy. Different Lung Cancer dataset (*LCM*) from University of Michigan [1] may be a good example. Number of genes that build classification model in *Linked TSP* was more than 5 times smaller to *TSP* methods although accuracy results slightly increased. In Table 4 we have compared genes used by *Linked* $k-TSP$ and *Linked Weight* $k-TSP$ classifiers. Higher number of genes to classifiers size is caused by crossvalidation of training dataset as no tested set was provided. Similar genes that

Table 4. Marker genes used in Lung Cancer University of Michigan dataset classification

Classifiers	k-TSP	Weight k-TSP	Linked k-TSP	Linked Weight k-TSP
Accuracy	95.62%	98.53%	97.26%	99.26%
Genes	more than 40 genes	more than 40 genes	**J03600_at** **M24486_s_at** **X64177_f_at** **U87964_at** U60061_at Y09216_at	**J02871_s_at** **M24486_s_at** **X64177_f_at** **U87964_at** - -

build *Linked TSP* classifiers despite different algorithms suggest that those genes can be considered as marked. Original TSP methods used over 40 different genes in prediction. They contained *Linked TSP* genes however many irrelevant features were also enclosed making the classifiers results much more difficult to analyze and interpret by human experts.

We have observed that genes from list G that built *Linked TSP* more often occurred in tested decision trees and the rest of classifiers to selected ones from $k-TSP$ or *Weight* $k-TSP$. This may confirm that *Linked TSP* prediction model involve only predominatingly genes from TSP pairs. Relying on experimental results we may conclude that *Linked TSP* simplify decision rules without reducing classification accuracy and even improving it for some datasets.

In our research we also investigate performance 8 different classifiers on datasets from Table 1. In our research we were focused on the "white box" methods rather the "black box" algorithms and this is the reason why methods like Support Vector Machine (SVM) [**?**] or neutral networks [4] were not included in our analysis. Comparison tests were performed with methods that like TSP generate simple and comprehensible decision rules. Results for those classifiers are enclosed in Table 5. If we compare them with ones from Table 2 we may observe that TSP-family classifiers achieve relatively higher average accuracy through all datasets. Even methods, that generate more complex to analyze and interpret decision tree ensembles like Bagging or Boosting also achieved slightly lower score.

5. Conclusion and Future Works

This paper presents extension of TSP-family classifiers called *Linked TSP*. We believe it is an interesting approach that may compete with TSP-family methods. General improvement of *Linked TSP* method did not exceed 2% although for some analyzed datasets idea of linking top pairwise mRNA comparisons of gene expressions

Table 5. Comparison classifiers accuracy

Dataset/Classifier	1. AD	2. BF	3. J48	4. RT	5. CT	6. JR	7. BG	8. ADA
1. BC	42.10	47.36	52.63	36.84	68.42	73.68	63.15	57.89
2. CNS	63.33	71.66	56.66	63.33	73.33	65.00	71.66	75.00
3. CT	74.19	75.80	85.48	70.96	75.80	74.19	79.03	79.03
4. DS	95.74	80.85	87.23	68.08	82.97	74.46	87.23	89.36
5. DF	88.31	79.22	79.22	81.81	83.11	77.92	85.71	90.90
6. DN	50.00	60.00	57.50	53.75	62.50	61.25	58.75	65.00
7. LA	91.17	91.17	91.17	55.88	91.17	94.11	94.11	91.17
8. LCB	81.87	89.65	81.87	77.18	81.87	95.97	82.55	81.87
9. LCM	96.87	96.87	98.95	91.66	96.87	93.75	97.91	96.87
10. LCT	69.23	61.53	58.97	53.84	58.97	64.10	61.53	69.23
11. PC	38.23	44.11	29.41	47.05	44.11	32.35	41.17	41.17
Average score	71.90	73.04	72.05	65.02	75.65	74.30	76.68	**76.72**

increased accuracy for over 10%. The size of classification model was significantly smaller (almost 40%) therefore *Linked TSP* generates more adequate and comprehensible decision rules. However, for some tested datasets original TSP was more accurate that is why the best TSP method can not be indicated. It is worth to notice that all TSP classifiers used similar set of top genes in decision model. This may suggest that each algorithm generates significant rules that capture interactions in datasets from different aspects.

Classification results comparison through tested datasets reveal that TSP-family classifiers are good alternative to decision trees and other classification rules. We believe that there is still place for improvement TSP-family classifiers. Merging the $k-TSP$, *Weight* $k-TSP$ and *Linked* TSP predictive power in a single algorithm might significantly increase accuracy and provide efficient decision rules with clear biological connections to adequate cancer type.

References

[1] Beer D.G.: Gene-expression Profiles Predict Survival of Patients with Lung Adenocarcinoma. Nature Medicine, 8(8):816-823, 2002.

[2] Bittner M., Meltzer P., Chen Y.: Molecular classification of cutaneous malignant melanoma by gene expression profiling. Nature, 406, 536-540, 2000.

[3] Brown P.O., Botstein D.: Exploring the new world of the genome with DNA microarrays. Nature Genet 21. 33-37, 1999.

[4] Cho H.S., Kim T.S., Wee J.W.: cDNA Microarray Data Based Classification of Cancers Using Neural Networks and Genetic Algorithms. Nanotech, vol. 1, 2003.

[5] Czajkowski M., Krętowski M.: Novel extension of k-TSP algorithm for microarray classification. Lecture Notes in Artificial Intelligence, vol. 5027:456-465, 2008.

[6] Dhanasekaran S.M.: Delineation of prognostic biomarkers in prostate cancer. Nature, 412, 822-826, 2001.

[7] Duggan D.J., Bittner M., Chen Y., Meltzer P., Trent J.M.: Expression profiling using cDNA microarrays. Nature Genetics Supplement, 21, 10-14, 1999.

[8] 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.

[9] Gordon J.G.: Translation of Microarray Data into Clinically Relevant Cancer Diagnostic Tests Using Gege Expression Ratios in Lung Cancer And Mesothelioma. Cancer Research, 62:4963-4967, 2002.

[10] Grześ M., Krętowski M.: Decision Tree Approach to Microarray Data Analysis. Biocybernetics and Biomedical Engineering, vol. 27(3), 29-42, 2007.

[11] Hastie T., Tibshirani R., Friedman J.H.: The Elements of Statistical Learning. Springer-Verlag, 2001.

[12] Kent Ridge Bio-medical Dataset Repository: http://datam.i2r.a-star.edu.sg/datasets/index.html

[13] Kononenko I.: Estimating Attributes: Analysis and Extensions of RELIEF. In: European Conference on Machine Learning, 171-182, 1994.

[14] Lockhart D.J., Winzeler E.A.: Genomics, gene expression and DNA arrays. Nature 405, 827-836, 2000.

[15] Lu Y., Han J.: Cancer classification using gene expression data. Information Systems, 28(4), pp. 243-268, 2003.

[16] Nelson P.S.: Predicting prostate cancer behavior using transcript profiles. Journal of Urology, 172, 28-32, 2004.

[17] Sebastiani P., Gussoni E., Kohane I.S., Ramoni M.F.: Statistical challenges in functional genomics. Statistical Science, 18(1), 33-70, 2003.

[18] 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.

[19] Speed T.: Statistical Analysis of Gene Expression Microarray Data. Chapman & Hall/CRC, New York, 2003.

[20] Tan A.C., Naiman D.Q., Xu L., Winslow R.L. and Geman D.: Simple decision rules for classifying human cancers from gene expression profiles. Bioinformatics, vol. 21, 3896-3904, 2005.
[21] Vapnik, V.N.: Statistical Learning Theory. Wiley, New York, 1998.
[22] Witten I.H., Frank E.: Data Mining: Practical machine learning tools and techniques. 2nd edn. Morgan Kaufmann, San Francisco, 2005.

ROZSZERZENIE METOD Z RODZINY TSP W KLASYFIKACJI MIKROMACIERZY DNA

Streszczenie Klasyfikacja danych mikromacierzowych a także późniejsza interpretacja reguł decyzyjnych może być skutecznie przeprowadzona za pomocą metod z rodziny Top Scoring Pair, polegających na analizie par genów o przeciwstawych poziomach ekspresji w różnych klasach. W poniższym artykule zaprezentowano nową metodę: *Linked TSP*, która rozszerza działanie klasyfikatorów $k-TSP$ i *Weight* $k-TSP$. W przeciwieństwie do algorytmów z rodziny TSP proponowane rozwiązanie tworzy reguły decyzyjne zbudowane z pojedynczych genów co znacznie ułatwia ich późniejszą interpretacje medyczną. W algorytmie wykorzystywane są pary genów uzyskane z algorytmów TSP z których następnie, wybierane są pojedyncze, najczęściej powtarzające się geny. Testy algorytmu *Linked TSP* przeprowadzone zostały na rzeczywistych zbiorach danych pacjentów a uzyskane wyniki są obiecujące.

Słowa kluczowe: klasyfikacja par genów zależnych, analiza mikromacierzy, reguły decyzyjne, ekspresja genów

Artykuł zrealizowano w ramach pracy badawczej W/WI/5/08.