

Using SVD and SVM methods for selection, classification, clustering and modeling of DNA microarray data

Krzysztof Simek^{a,*}, Krzysztof Fajarewicz^a, Andrzej Świerniak^a, Marek Kimmel^{a,b},
Barbara Jarzab^c, Małgorzata Wiench^c, Joanna Rzeszowska^c

^a *Institute of Automatic Control, Silesian University of Technology, Akademicka 16, 44-101 Gliwice, Poland*

^b *Department of Statistics, Rice University, P.O. Box 1892, Houston, TX 77251, USA*

^c *Centre of Oncology, Maria Skłodowska-Curie Memorial Institute, Wybrzeże Armii Krajowej, 44-101 Gliwice, Poland*

Abstract

DNA microarray technology is the latest and the most advanced tool for parallel measuring of the activity and interactions of thousands of genes. This modern technology promises new insight into mechanisms of living systems, for example only two high-density oligonucleotide microarrays are sufficient to inspect the whole human genome. However, it provides unprecedented amount of data that require application of advanced computational methods. The appropriate choice of data analysis technique depends both on data and on goals of an experiment. In this paper we focus on two promising methods: singular value decomposition and support vector machines. We discuss the possibility of application of these methods for different purposes; particularly for clustering, classification, feature selection and modeling of dynamics of gene expression. We use for testing presented approaches existing data sets, which are widely available via Internet, and one new tumor/normal thyroid microarray data set.

© 2004 Elsevier Ltd. All rights reserved.

Keywords: DNA microarrays; Singular value decomposition; Support vector machines; Clustering; Data mining; Feature selection; Modeling of gene expression data

1. Introduction

DNA microarrays, i.e. microscopic arrays of large sets of nucleotide sequences, are modern tool that is used to obtain information about expression levels of thousands of genes simultaneously. The main advantages of the technology are reproducibility and scalability of obtained data, short time of performing of a single experiment and, of course, the large number of genes, the expressions of which are measured. Microarrays are, in principle and practice, extensions of hybridization-based methods which have been used for decades to identify and quantify nucleic acids in biological samples (e.g. Southern and Northern blots, colony hybridizations, dot blots). In the typical experi-

ment, RNA is isolated from tissues of interest, labeled, either radioactively or fluorescently, and allowed to hybridize to the array. After sufficient time for hybridization and following appropriate washing steps, a digital image of the array is acquired and analyzed to measure the intensity of hybridization for each gene on the array. The hypothesis underlying microarray analysis is that the measured intensities for each arrayed gene represent its expression level.

Currently there are at least two competing types of DNA microarrays: spotted cDNA microarrays (Duggan et al., 1999; Schena et al., 1995; Shalon et al., 1996) developed at Stanford University and oligonucleotide chips (Lipshutz et al., 1999; Lockhart et al., 1996; Schadt et al., 1999) developed by Affymetrix. There are several important differences between these two types of microarrays (Berrar et al., 2003). Spotted microarrays consist of a solid surface onto which miniscule amounts (spots) of single strands of nucleotide sequences are deposited by an automated process called contact spotting (similar to ink-jet printing) in grid-like

*Corresponding author.

E-mail addresses: ksimek@ia.polsl.gliwice.pl (K. Simek), kfajarewicz@ia.polsl.gliwice.pl (K. Fajarewicz), aswierniak@ia.polsl.gliwice.pl (A. Świerniak), kimmel@rice.edu (M. Kimmel), jarzab@io.gliwice.pl (B. Jarzab), wiench@io.gliwice.pl (M. Wiench), jwolny@io.gliwice.pl (J. Rzeszowska).

arrangement. Each spot represents a specific gene and serves as a probe against which sample RNA is hybridized. The single spot has a diameter of approximately 100 μm . In this way 10,000–30,000 probes can be arranged on the single microarray. However, the number of probes do not match the number of genes. For reasons of reproducibility, a gene may be represented by more than one probe. With oligonucleotide chips the probes are synthesized on the array on the basis of the sequences of existing or hypothetical genes using photolithographic technology (similar to the technology used in production of electronic chips). The diameter of each probe spot is approximately 18 μm that allows maximum 500,000 probes per array. *Affymetrix* also makes use of multiple probes to represent the genes. For high-density chips even 22 probes per gene is used, allowing for up to 23,000 genes per chip.

Most microarray experiments investigate relationships between related biological samples based on gene expression measurements performed in different conditions. In many cases several samples, for different tissues, for example normal/tumor tissues or different disease types are studied. Alternatively, development in time of one biological phenomenon is studied, leading to a series of measurements following each other. The aim of the investigation can be formulated in many different ways. The most basic problem is to find genes that exhibit different expression levels under different experimental conditions. Typical studies of this kind include normal versus malignant sample investigation. Slightly different problem is to identify co-regulated genes, i.e. genes whose expression levels vary in a coordinated or correlated way across conditions. Examining the co-regulation is often the first step of the analysis leading to identification of function of novel genes. The functions of genes with highly similar expression patterns can be similar or their expression can be regulated by the same internal (e.g. cell cycle) or external (e.g. exposing to drug or radiation) factor. In experiments of this kind, time-course investigations are usually performed allowing analysis of temporal changes in gene expression. Time-course experiments are also very useful in gene regulatory network identification studies. Other important application of microarray experiments is clinical diagnosis aimed at revealing expression patterns that are characteristic for particular diseases or even for its distinct subtypes.

Microarray experiments provide enormous amount of data that require application of advanced computational methods. The appropriate choice of data analysis technique depends both on data and on the goals of the experiment. However, before analysis can be started the raw data developed from microarrays must be computationally collected, processed and integrated. This phase of data preparation is called pre-processing. Pre-processing is at least three-fold. First, within microarray

normalization is applied to compensate for systematic measurement errors due to array equipment imperfection. Second, the multiple measurements are combined to obtain a single expression level for each gene. Third, data from different microarrays are integrated into a single data matrix. To compensate measurement variation for different arrays an array-to-array normalization is employed.

Once the final format of the data is achieved exploratory data analysis can start. As the starting point, we assume that for each biological sample assayed a high-quality measurement of the intensity of hybridization for each gene is obtained on the array. The intensity values are ordered into an $n \times m$ expression matrix A . In most applications, the number of genes investigated is much greater than the number of time points or samples assayed, i.e. the case $n > m$ is considered. Each row of matrix of gene expression corresponds to a different gene and each column corresponds to a different sample or time instant at which expression data were measured. The entries of the matrix A are defined by numerical values corresponding to gene expressions. Usually, the first step of exploratory analysis is data transformation. The objective of the transformation is to reduce complexity of the data and to represent the information in a different, more useful format. Statistical operations, like data centering and use of logarithmic transformation, are good examples of such operations.

The available variety of microarray analysis methods ranges from classic statistical or vector algebra approaches, to machine learning techniques and methods from the field of artificial intelligence.

There are many methods of classification of artificial intelligence techniques. One of them is division of all methods into supervised and unsupervised groups of methods. Supervised methods make use of the information about class membership of analyzed samples. In unsupervised methods such information is not utilized. In this paper we intend to present two computational methods, very promising when dealing with gene expression data, belonging to two different, in this sense, categories. They are: singular value decomposition (SVD) which belongs to unsupervised methods and support vector machines (SVM) which is an example of a supervised method. We demonstrate in several examples that these methods are very useful when dealing with gene expression data sets and can be used in many practical applications, such as: clustering, classification, feature selection and modeling of dynamics of gene expression.

The paper is organized as follows. In Section 2 definition and properties of SVD are presented. In Section 3 an SVD based algorithm of selection of genes differentiating groups of samples is addressed. An example of application of the method to original

ontological microarray data is given. Section 4 describes a method of modeling of dynamics of gene expression time-course data. Sections 5 and 6 focuses on SVM method and its application to gene selection.

2. Singular value decomposition

Recently, gene expression data were analyzed using SVD (Alter et al., 2001; Holter et al., 2001; Raychaudhuri et al., 2000; Simek and Kimmel, 2002). SVD is a matrix factorization, known from linear vector algebra, that reveals many important properties of a matrix. It is a standard tool in many areas of physical sciences, and many algorithms in matrix algebra make use of SVD. In gene expression data analysis the principal aim of application of SVD is to detect and extract internal structure existing in the data and corresponding to important relationships between expressions of different genes.

2.1. Mathematical foundation

The singular value decomposition of any $n \times m$ matrix A has the following form (Golub and van Loan, 1996):

$$A = USV^T, \quad (1)$$

where U is an $n \times n$ orthonormal matrix, whose columns are called the left singular vectors of A (gene coefficient vectors), and V is an $m \times m$ orthonormal matrix, whose columns are called the right singular vectors of A (expression level vectors). Matrix S is an $m \times m$ diagonal matrix with the form

$$S = \begin{bmatrix} s_1 & & 0 \\ & \ddots & \\ 0 & & s_m \end{bmatrix}. \quad (2)$$

The diagonal elements of matrix S are listed in a descending order $s_1 \geq s_2 \geq \dots s_m \geq 0$ and called the singular values of A .

Some important mathematical properties of the SVD matrices are presented below.

- (1) Singular values of rectangular matrix A are equal to square root of eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_m$ of matrix $A^T A$.
- (2) Rank of matrix A is equal to the number r of positive singular values: $\text{rank}(A) = r, r \leq m$.
- (3) Euclidean norm of matrix A is equal to the largest singular value: $\|A\|_2 = s_1$.
- (4) First r columns of matrix U form an orthonormal basis for the space spanned by the columns of matrix A .
- (5) First r columns of matrix V form an orthonormal basis for the space spanned by the rows of matrix A .

Sometimes, before applying the SVD, data regularization is performed. The regularized rows and columns of the expression matrix have mean values equal to 0. Because of this operation the rank of matrix A is equal to $r \leq m-1$. Depending on circumstances, polishing can be desirable or not.

2.2. Characteristic modes

Let us denote by X_i the upper r rows of matrix SV^T and define matrix

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_r \end{bmatrix} = \begin{bmatrix} s_1 v_1^T \\ \vdots \\ s_r v_r^T \end{bmatrix}. \quad (3)$$

The orthogonal vectors X_i are called characteristic modes associated with matrix A .

One can easily show that variations of the j th gene expression across analyzed samples, included in the row A_j of matrix A , can be written exactly as a linear combination of the characteristic modes

$$A_j = \sum_{i=1}^r U_{ji} X_i, \quad (4)$$

where the coefficients of the combination are the corresponding entries of matrix U .

Usually not all characteristic modes are needed to reconstruct gene expression patterns with a reasonable accuracy. We may use a truncated expression

$$A_j = \sum_{i=1}^l U_{ji} X_i, \quad l \leq r.$$

The contribution of modes to the gene pattern decreases from the higher order to the lower order modes. The singular values, which represent the magnitudes of the corresponding modes, can be used as measures of relative significance of each characteristic mode in terms of the fraction of overall expression that it captures

$$p_i = \frac{s_i^2}{\sum_{j=1}^r s_j^2}, \quad i = 1, \dots, r. \quad (5)$$

Similar index can be defined for each gene

$$c_k^i = \frac{(U_{ki} s_i)^2}{\sum_{j=1}^r (U_{kj} s_j)^2}. \quad (6)$$

It defines the contribution of the i th mode to the expression pattern of the k th gene.

There are several heuristic methods to estimate the number l of the most significant characteristic modes (Everitt and Dunn, 2001; Jackson, 1991). One of the simplest is to retain just enough modes to capture large percentage of overall expression. Usually values of 70–90% are proposed. The other procedure is to exclude characteristic modes such that the fraction of expression

p_i they capture is less than $(70/r)\%$. Different method is examination of the so-called scree plots for s_i^2 or $\log s_i^2$. Using this method one can usually find a natural border between significant and insignificant singular values (the so-called elbow).

3. Gene selection using SVD

In the clustering literature, SVD is sometimes applied to reduce dimensionality of the data set prior to clustering. The idea behind using SVD prior to cluster analysis is that SVD may extract the cluster structure in the data. Since characteristic modes are uncorrelated and ordered, the first few most significant ones, which contain most of the variations in the data, are usually used in cluster analysis.

We aim to investigate the effectiveness of SVD as a preprocessing step to cluster analysis on gene expression data. Our approach differs from that known from the literature, where characteristic modes coefficients (gene coefficient vectors), instead of original variables, are used for clustering. We propose to apply SVD to select a set of original genes and then apply them for clustering samples by one of the standard algorithms.

3.1. Algorithm of gene selection

The gene selection algorithm inspects gene coefficient vectors (columns of matrix U) corresponding to the set of the most significant characteristic modes. Each coefficient is compared to the threshold value (Wall et al., 2001), whose meaning is similar to a 3σ statistical significance cutoff, equal to $Wn^{-1/2}$, where n is the number of genes and W is a weight factor whose recommended value is greater than 3. If the magnitude of the element is greater than the threshold, the corresponding gene is selected to the clustering set. In practice we choose genes having sufficiently big coefficients for the most important characteristic modes, or in other words, genes for which values of index (6) for the most important modes are big enough. Variation of factor W gives possibility of changing a number of selected genes. In the result we obtain set of genes having patterns ‘similar’ to the dominant modes.

3.2. Example of analysis

To illustrate the approach we applied the algorithm to the gene expression data, consisting of 16 tumor/normal thyroid tissue pairs, acquired in Center of Oncology, Gliwice, Poland, using *Affymetrix* Human Genome U133A arrays. About 150 mg of tissue was fragmented and homogenized. Total RNA was extracted and repurified. The quantity and integrity of RNA were checked by spectrophotometry and gel electrophoresis.

RNA was taken for a cDNA synthesis reaction followed by the synthesis of biotin-labeled cRNA. Obtained cRNAs were fragmented and hybridized to U133A arrays. Then washing, staining and scanning of the arrays in a GeneArray scanner was performed.

The aim of our data mining analysis was to identify normal (1–16) and tumor (17–32) samples. Since clustering of original data was not effective, we applied SVD and found characteristic modes for the data. Inspection of the modes, presented in Fig. 1, reveals that the first mode corresponds to the most important trend in the data, which is tumor/normal feature. It allows expecting that basing only on this mode the samples can be split into two proper groups. Applying described procedure we selected 78 out of over 22,000 genes (for $W=4.5$) to be used for clustering. Profiles of the selected genes are presented in Fig. 2. We applied the hierarchical complete-linkage clustering algorithm with

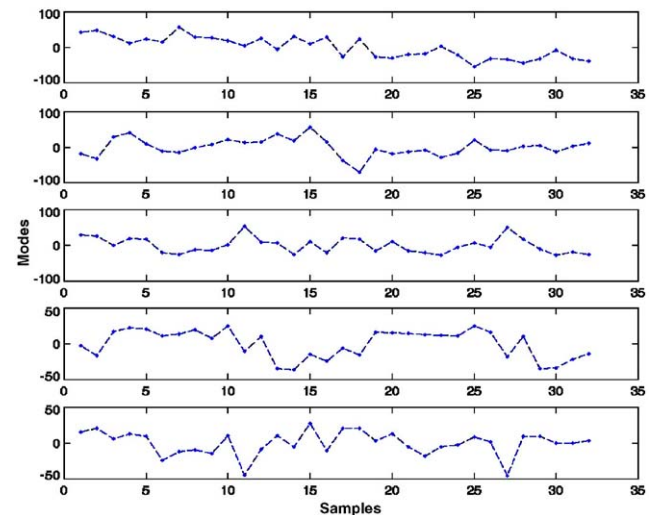


Fig. 1. Five out of 31 characteristic modes for gene expression data.

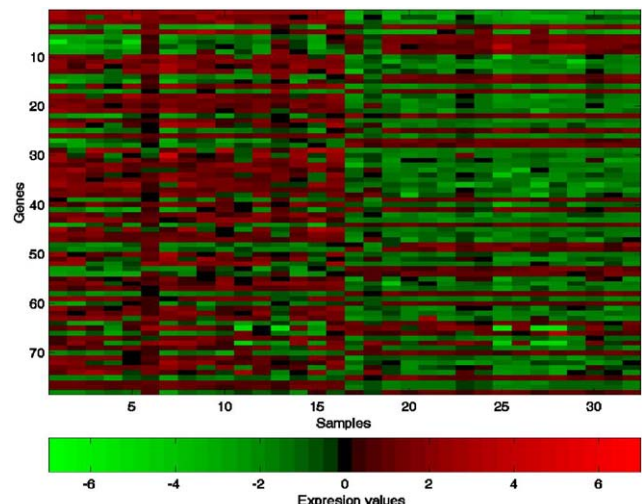


Fig. 2. Expression profile of genes selected by means of SVD.

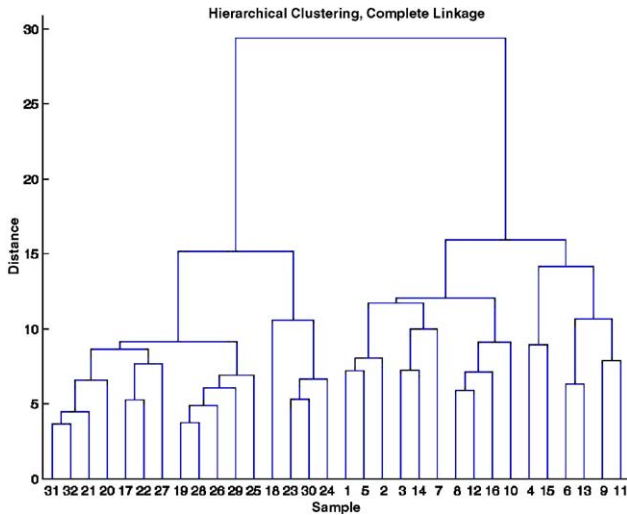


Fig. 3. Results of hierarchical clustering of samples based on selected genes.

Euclidean distance metrics to identify normal and tumor samples. Hierarchical clustering has the advantage that it is simple and the results can be easily visualized and interpreted. It has become one of the most commonly used techniques in the field. The results are presented in Fig. 3. The analysis proves effectiveness of the proposed exploratory algorithm.

4. Dynamic model for characteristic modes

Since for time-course studies characteristic modes are functions of time, we try to find a discrete-time dynamical model of temporal changes of the modes. We assume the simplest linear model in which the expression values at a given time are linear combinations of the values at a previous time instant.

4.1. Problem formulation

Let us denote by $Y(t_j)$ the expression level of all characteristic modes at time points t_j , when gene expression was measured. Matrix of characteristic modes (3) can now be rewritten as

$$X = [Y(t_1), Y(t_2), \dots, Y(t_m)] \quad (7)$$

and the dynamical model can be written in the form of a linear equation:

$$Y(t + \Delta t) = MY(t), \quad (8)$$

where M is a $q \times q$ translation matrix, q is a number of characteristic modes considered in the model, $q \leq m$ and Δt a time step for the dynamical model.

For equally spaced measurements Δt can be found from the expression $\Delta t = t_{i+1} - t_i$ and $t_i = i\Delta t$. For unequally spaced measurements Δt is defined as max-

imal time interval such that each measurement time is an integer multiple of Δt , i.e., $t_i = n_i \Delta t$.

Since, as mentioned earlier, time-series data often can be represented by the most significant modes only and a part of characteristic modes can be excluded, one can try to build reduced order model taking into account only small number of variables. In this case the dimension of vector $Y(t_i)$ is $q = l$ but the form of the dynamical model (8) is not changed.

To obtain the model we find matrix M based on the knowledge of temporal patterns of characteristic modes. The optimization problem consists of minimization of the performance index of the form

$$J = \frac{\sum_j \|Y(t_j) - Z(t_j)\|^2}{\sum_j \|Y(t_j)\|^2}, \quad (9)$$

where $Z(t)$ is a time variable described by linear discrete equation

$$Z(t_1 + k\Delta t) = M^k Y(t_1), \quad k = 1, 2, \dots, n_m$$

with initial condition $Z(t_1) = Y(t_1)$. Since the measurements $Y(t_j)$ are given, the problem consists of finding the q^2 entries of matrix M , which minimize J .

4.2. Methods of solution

(1) *Equally spaced measurements and $q = r$* : For equally spaced measurements and $q = r$ (i.e. full-order model), the solution of the problem leads to the solution of a linear system of algebraic equations

$$Y = \tilde{Y}\tilde{M}, \quad (10)$$

where \tilde{Y} is a square $r^2 \times r^2$ matrix and \tilde{M} is a vector containing transposed rows of matrix M .

Solving the equation one obtains optimal elements of matrix M . Assuming that matrix \tilde{Y} is nonsingular, the equation has one unique solution and the value of the index (9) is equal to 0. Standard MATLAB procedures are used to solve the problem.

(2) *Equally spaced measurements and $q < r$* : In the case of equally spaced measurements and $q < r$ (i.e. reduced-order model) optimization problem may be brought to the solution of the equation similar to (10), but now matrix \tilde{Y} is an $r q \times q^2$ rectangular matrix. The resulting translation matrix M is the solution in the least-squares sense to the overdetermined system of equations of type (10). Obtained fitting is not ideal. Again Standard MATLAB procedures can be applied.

(3) *Unequally spaced measurements and $q \leq r$* : For unequally spaced measurements and the general case $q \leq r$, it is necessary to minimize the goodness of the fit index J , as defined in (9). In Alter et al., 2001 the authors used simulated annealing, while we use a standard Gauss–Newton algorithm (see Branch and Grace, 1996 and references therein for details) as provided in MATLAB, with very good results. The problem is

strongly nonlinear and in general very hard to solve, especially for meaningful differences in measurements time intervals. Since the applied optimization algorithm is very sensitive to the choice of the initial guess of the solution, we apply an original two-step optimization procedure. In most cases appropriate tuning of parameters of optimization is required to obtain a precise solution.

4.3. Example of analysis

To illustrate the considerations we use publicly available data on yeast *cdc-15* synchronized cell cycle, described in Spellman et al., (1998). In yeast culture synchronized by *cdc-15* over 6000 genes were monitored over approximately 2.5 cell cycle periods. We chose data set consisting of almost 800 genes, classified to be cell cycle regulated, and 12 measurements at 20 min intervals, beginning at $t_1 = 10$ min. The analysis consists of two parts. In the first part we built a dynamical model for the original data. In the second part, to test the reconstruction properties of dynamical system fitting, we deleted some data, i.e. two columns corresponding to measurements at times $t = 70, 150$, obtaining a modified data sets with unequally spaced measurements.

Analysis of singular values and corresponding coefficients of relative significance, given in Table 1, reveals in both cases that first two characteristic modes capture roughly 70% of the overall variability of the expression. It means that the temporal pattern of gene expression can be described by the use of two characteristic modes with reasonable accuracy.

For original data the solution matrix M is unique and the dynamical model provides an exact reconstruction of the characteristic modes. In Fig. 4 characteristic modes of the data sets are presented. It is easy to notice that the distortion of the data, i.e., deleting two columns, equivalent to 16% missing data, did not change shapes

of the original characteristic modes. It proves robustness of SVD.

Fig. 6 shows reconstruction of characteristic modes with the use of the full dynamical model. For distorted data set the reconstruction at the retained measurement points is very precise. It means that optimization procedure provides accurate solutions and that the obtained dynamical model can be used to recover missing data with reasonable fidelity.

As shown in Figs. 5 and 7, which present reconstruction of the first two characteristic modes by using reduced dynamical models in both cases, the main features of expression patterns are reproduced quite well. It shows that influence of the high-order modes on dominant ones is weak and the dominant modes could be reconstructed basing on a reduced order model (Figs. 5–7).

Detailed discussion of the properties of the proposed dynamical models is presented in Simek (2003).

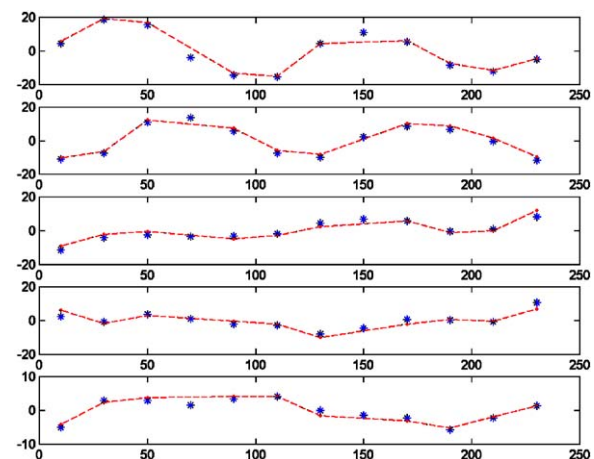


Fig. 4. Characteristic modes for the original data (stars) and for the modified data set.

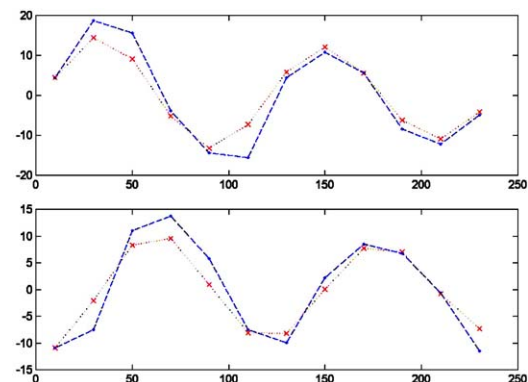


Fig. 5. Reconstruction of the first two characteristic modes for the original data. The dots correspond to characteristic modes, the crosses correspond to reconstructed variables basing on the reduced dynamical model of order 2.

Table 1
Singular values and corresponding p_i index for analyzed data sets

Part 1		Part 2	
s_i	p_i	s_i	p_i
38.59	0.43	36.96	0.46
30.50	0.27	27.02	0.27
18.75	0.10	17.41	0.10
15.41	0.07	14.38	0.07
10.97	0.03	10.80	0.04
9.97	0.03	9.27	0.03
8.48	0.02	7.56	0.02
7.62	0.02	6.84	0.02
7.22	0.01	6.10	0.01
6.56	0.01	—	—
5.65	0.01	—	—

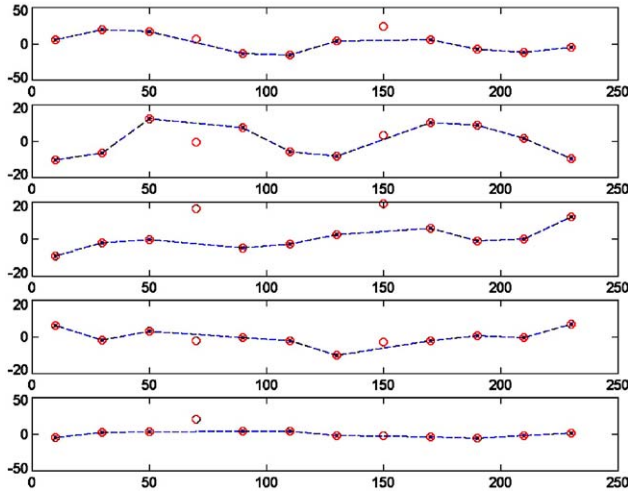


Fig. 6. Reconstruction of the characteristic modes for the modified data set. Dots correspond to the characteristic modes, crosses correspond to the reconstructed characteristic modes (full dynamical model), circles show approximation of the temporal pattern resulting from the dynamical model.

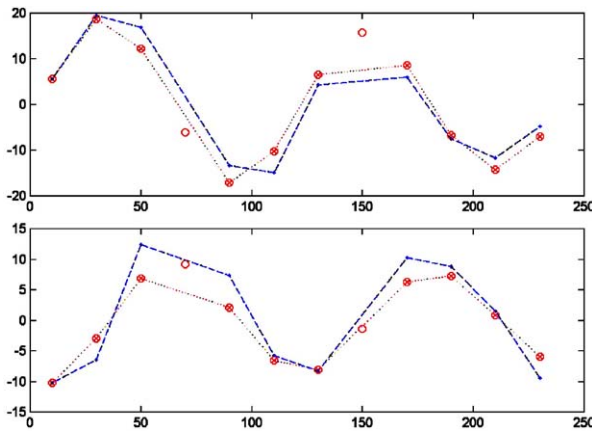


Fig. 7. Reconstruction of the first two characteristic modes for the modified data set. Dots correspond to the characteristic modes, crosses correspond to the reconstructed modes (the second-order dynamical model), circles show approximation of the temporal pattern resulting from running the model for each time $t=n \Delta t$.

5. Support vector machines

SVM method (Christianini and Shawe-Taylor, 2000; Haykin, 1999; Vapnik, 1995) is one of the prospective tools of analysis for gene expression data coming from DNA microarrays. This is due to the fact that SVM method is particularly suitable to cope with “rare” data sets, i.e. when number of samples is much less than the number of features (genes). Moreover, as it will be shown in next section, it is possible to use the SVM technique for gene selection.

Let us consider a matrix A of dimension $n \times m$ containing gene expression data. The matrix A is composed of m column vectors $x_i \in R^n$; $i = 1, 2, \dots, m$.

Each vector represents one and only one class ω_1 or ω_2 (for example tumor or normal tissue). In standard linear classification problem we are looking for a weight vector $w \in R^n$ and scalar bias b of a linear classifying (discriminant) function

$$f(x) = w^T x + b \quad (11)$$

which satisfies the following set of inequalities:

$$\begin{aligned} w^T x_i + b &> 0 & \text{for } x_i \in \omega_1, \\ w^T x_i + b &< 0 & \text{for } x_i \in \omega_2. \end{aligned} \quad (12)$$

When the training set is linearly separable then there exists such a function.

For the simplicity let us introduce a set of desired responses (target outputs): $\{d_i\}_{i=1}^m$

$$d_i = \begin{cases} +1 & \text{when } x_i \in \omega_1, \\ -1 & \text{when } x_i \in \omega_2. \end{cases} \quad (13)$$

Discriminant function (11) determines, in an n -dimensional input space, a hyperplane P called a *decision surface*. The equation of this surface is as follows:

$$w^T x + b = 0. \quad (14)$$

For the linearly separable case there are infinite number of “good” discriminant hyperplanes, i.e., satisfying inequalities (12), but only one is optimal in SVM sense. Optimal hyperplane P^o satisfies inequalities (12), but also maximizes a *margin of separation* γ which indicates the Euclidean distance ρ between hyperplane P and the closest vector. Hence, the problem can be stated mathematically as follows:

Problem. Find optimal w^o and b^o that maximize

$$\gamma = \min_i \rho(P, x_i), \quad i = 1, 2, \dots, m \quad (15)$$

subject to constraints (12).

Vectors, for which $\rho(\cdot)$ takes minimal value, are called *support vectors*.

It can be shown that this problem can be transformed into a quadratic programming task. Moreover, introducing so-called slack variables makes possible to deal with non-separable data sets. In practice, a dual problem to quadratic programming task is solved and optimal values of Lagrange multipliers (each corresponding to one training vector) are calculated.

A linear SVM is a special case of a more general nonlinear SVM constructed by introducing an additional set of nonlinear functions (nonlinear kernel) (Christianini and Shawe-Taylor, 2000; Haykin, 1999; Vapnik, 1995).

6. SVM in gene selection

One of the benefits of the SVM method, besides its optimality is presented before sense, is the fact that it gives unique solution. This method combined with the so-called leave-one-out cross-validation, which also gives unique evaluation of generalization ability, allowed us to formulate a method of features (genes) selection called recursive feature replacement (RFR) (Fujarewicz and Wiench, 2003).

6.1. Evaluation of gene subset generalization ability

In this subsection we describe two methods which we use to evaluate classification quality of particular gene subset. Both are based on the result of leave-on-out cross-validation but they use different formulas for evaluating gene subset generalization ability.

The fact, that is worth recalling here, is that the recognition system is constructed not to separate perfectly the training set. The primary goal is rather to find the feature set (gene set in our application), the form of the classifying function and the learning algorithm, for which the samples not being used during learning phase are classified correctly. In other words, the learning machine should be characterized by a good generalization ability.

In general, in the leave-one-out cross-validation method one vector x_k is removed from the training set and remaining vectors serve during learning phase. After this it is checked how the removed vector is classified. In our approach SVM technique is used for finding linear classification rule. The leave-one-out cross-validation method, when SVM method is used, can be stated formally as follows:

1. Remove one vector x_k from the training set.
2. For remaining vectors calculate w^o and b^o using SVM method.
3. For the removed vector x_k calculate the function

$$f_{\text{norm}}(x_k) = \frac{d_k}{\|w^o\|} \cdot (w^{o\top} x_k + b^o) \quad (16)$$

4. Repeat steps 1–3 for every $k = 1, 2, \dots, m$.

In formula (16) d_k is the target output (13) and $\|\cdot\|$ denotes the Euclidean norm. Thanks to division by the norm of w^o the absolute value of (16) is equal to the Euclidean distance between decision surface and the vector x_k . This is because after this normalization the norm of the gradient of the function (16) is equal to 1. The positive value of (16) indicates that the vector x_k is correctly classified.

As mentioned above, we use two different performance indices based on all values of (16) calculated for all samples.

The former is a simple percentage index which takes into account how many samples are correctly classified in leave-one-out cross-validation

$$J_{cv1} = \frac{N_{\text{corr}}}{N} 100\%, \quad (17)$$

where N_{corr} is a number of positive values of (16).

The latter is based only on the worst (minimal) value among all values of (16)

$$J_{cv2} = \frac{1}{\sqrt{n}} \min_k f_{\text{norm}}(x_k). \quad (18)$$

In formula (18) the results are divided by square root of n in order to make results comparable for training sets with different numbers of genes n . High values of both (17) and (18) indicate good generalization ability. If the performance index (18) is positive then all samples during leave-one-out cross-validation are classified correctly.

Note that the cross-validation method evaluates the generalization ability of the whole recognition system. Since in our approach the form of the discriminant function and the learning algorithm are fixed, the outcome of the cross-validation method presented here depends only on the way of selecting the gene set. Moreover, for fixed gene subset this outcome is unique because both: the method of cross-validation and the SVM technique, give unique results.

Let us denote by Ω the set of numbers of all measured genes $\Omega = \{1, 2, \dots, m\}$, and by $\Omega^* \subset \Omega$ any of its subset. The symbols

$$J_{cv1}(\Omega^*), \quad (19)$$

$$J_{cv2}(\Omega^*) \quad (20)$$

will denote the results of evaluating the generalization ability (17) and (18) of the subset of genes Ω^* .

6.2. The recursive feature replacement method of gene selection

As mentioned in introduction, due to high computational cost, it is impossible to examine all subsets of thousands of genes the expressions of which are measured using microarrays. Therefore, we proposed (Fujarewicz and Wiench, 2003) a new heuristic and iterative algorithm. In this algorithm the subset of genes Ω^* is modified in successive iterations so that the value of the performance index increases. Since the performance index (17) takes only discrete values (0%, $1/N 100\%$, $2/N 100\%$, ..., 100%) we use second performance index (18) which has real value.

The algorithm

1. Read initial subset $\Omega^* \subset \Omega$.
2. Find the single gene of the number $k \in \Omega^*$ that maximizes $J_{cv2}(\Omega^* \setminus \{k\})$.
3. Find the single gene of the number $l \in \Omega \setminus \Omega^*$ that maximizes $J_{cv2}(\Omega^* \cup \{l\})$.
4. If $J_{cv2}((\Omega^* \setminus \{k\}) \cup \{l\}) > J_{cv2}(\Omega^*)$, then $\Omega^* := (\Omega^* \setminus \{k\}) \cup \{l\}$, go to step 2.
5. Stop.

Let p denote the number of genes in the subset Ω^* . Note that p does not change in successive iterations of the algorithm. So, if we want to find optimal subsets for all $p = 1, 2, \dots, n$ the algorithm has to be run for every $p = 2, 3, \dots, n-1$ (for $p = 1$ no recurrence is needed and for $p = n$ the optimal set is the set Ω). In practice, we are interested in finding the subset for which maximal value of the performance index J_{cv2} is achieved and it is not necessary to find all optimal subsets. In most cases the final value $p_{\max} = 100$ is sufficient. Moreover, prior to using the RFR algorithm a pre-selection may be performed and the number n is much less than the number of all genes which expressions are measured. Typically, calculations on a PC computer, for $p_{\max} = 100$ and the number of genes after pre-selection stage $n = 300$, takes few hours (it depends on m). As a starting gene subset Ω^* for given p we choose p -element subset of the best genes obtained during the pre-selection stage. Hence, it is desirable to apply a pre-selection algorithm which sorts genes. If not, or if several different methods are used for pre-selection, it is better to sort it. We find the recurrent feature elimination (RFE) (Guyon et al., 1999) very useful for this purpose.

In papers (Fujarewicz et al., 2002; Fujarewicz and Wiench, 2003) we compared the RFR algorithm to other methods of gene selections such as: the Sebestyen criterion (Sebestyen, 1962), the correlation coefficient proposed in the article by Golub et al. (1999), the method proposed by Szabo et al. (2002), and the RFE method (Guyon et al., 1999), which is also based on SVM technique. Among all investigated methods two methods distinctly outperformed the rest. These methods are RFR and RFE algorithms. They are comparable for larger number of genes but for smaller gene subsets the RFR method is better than the RFE method—gives smaller gene subset which gives no misclassifications during the leave-one-out cross-validation.

6.3. Example of analysis

To illustrate the approach, let us apply the RFR algorithm to the gene expression data, consisting of 16 tumor/normal thyroid tissue pairs used in Section 3.2.

The pre-selection stage was performed using two different methods of data selection: the Sebestyen

criterion and the correlation coefficient proposed in Golub et al., (1999). Each method selected 250 genes and the sum of these two gene subsets was sorted using the RFE method. The result of application of the RFR method to the data set is presented in Fig. 8. The maximum value of the performance index (20) is about 0.774 and it was achieved for the gene subset consisting of 20 genes. For the comparison, values of the performance index obtained the RFE method are presented in the same figure.

Now, let us focus on the following problem: how does the performance index depend on the number of samples m ? Let us remove randomly a half of samples (8 normal and 8 tumor tissues) and let us perform RFE algorithm for both, whole and truncated sets. The result is presented in Fig. 9. It can be observed that the performance index is better for smaller set of samples. This phenomenon can be easily explained when we

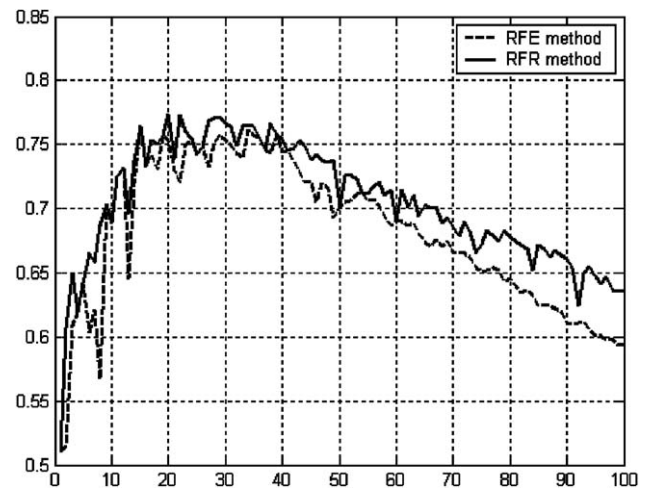


Fig. 8. Comparison of the performance index J_{cv2} versus number of genes for gene subsets obtained using RFR and RFE methods.

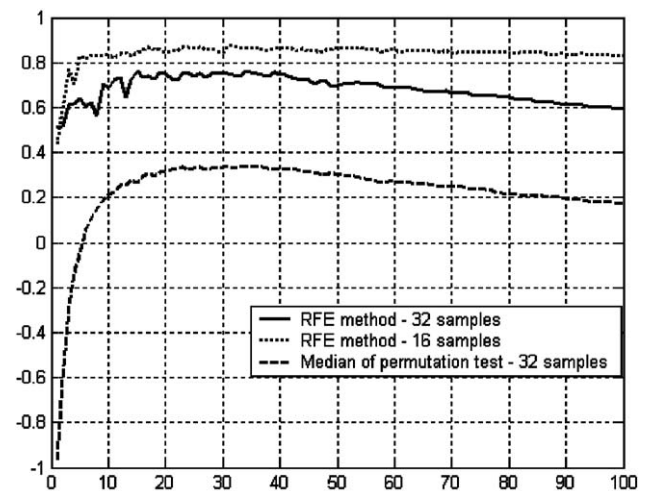


Fig. 9. Comparison of the performance index J_{cv2} for gene subsets obtained using RFR and RFE methods.

imagine only two-element data set. In such a case each differently expressed gene perfectly classifies this training set. When the data set is small the probability of finding a good discriminant gene is greater. But on the other hand, it may be only a case. So it is very important to remember this fact and to check whether the result is not obtained by chance.

It can be verified for example by using a permutation test. We performed 100 permutations of the whole (32 element) training set and the median of the performance index obtained by using the RFE method¹ is presented in Fig. 9. It can be observed that the result of permutation test is considerably worse and it indicates that our result (gene subsets) was not achieved by chance.

7. Conclusion

In this paper we investigated a possibility of using two computational methods: SVD and SVM to analyze gene expression data. From the point of view of artificial intelligence theory these two methods belong to different class of computational techniques: the SVM method is an example of supervised method while the SVD method belongs to class of unsupervised methods.

We discuss the possibility of application of these methods for different purposes such as: clustering, classification, feature selection and modeling of dynamics of gene expression.

The paper is partly a survey of our previous works where we applied SVD and SVM methods, and algorithms based on these methods, to analyze the gene expression data. Here we applied these methods to new microarray data set containing expression profiles for tumor/normal thyroid tissues. Using this data set, we indicate prospects of application of investigated methods. A possibility of using the SVD method for gene selection was examined. We also demonstrated how the number of samples in the training set influences the result of the leave-one-out cross-validation. In addition we suggest permutation tests as a tool to validate the quality of obtained discriminant gene subsets to avoid the risk connected with fact that the number of features (genes) is much greater than the number samples.

Acknowledgements

This work was supported by the Polish Committee of Scientific Research (KBN) partially under grant

4T11F01824 and partially under grant PBZ KBN-040/P04/08.

References

- Alter, O., Brown, P.O., Botstein, D., 2001. Processing and modeling genome-wide expression data using singular value decomposition. *Proceedings of SPIE* 4266, 171–186.
- Berrar, E.P., Dubitzki, W., Granzow, M. (Eds.), 2003. *Practical Approach to Microarray Data Analysis*, Kluwer Academic Press, London.
- Branch, M.A., Grace, A., 1996. *Matlab Optimization Toolbox. User's Guide*. The Math Works Inc., Natick, MA, USA.
- Christianini, N., Shawe-Taylor, J., 2000. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, Cambridge.
- Duggan, D.J., Bittner, M., Chen, Y., Meltzer, P., Trent, J., 1999. Expression profiling using cDNA microarrays. *Nature Genetics* 21, 10–14.
- Everitt, B.S., Dunn, G., 2001. *Applied Multivariate Data Analysis*. Oxford University Press, New York.
- Fujarewicz, K., Wiench, M., 2003. Selecting differentially expressed genes for colon tumor classification. *International Journal of Applied Mathematics and Computer Science* 13 (3), 101–110.
- Fujarewicz, K., Kimmel, M., Rzeszowska-Wolny, J., Świerniak, A., 2002. A note on classification of gene expression data using support vector machines. *Journal of Biological Systems* 10 (4), 1–14.
- Golub, G.H., van Loan, C.F., 1996. *Matrix Computations*. Johns Hopkins University Press, Baltimore.
- Golub, T.R., Slonim, T.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Downing, J.R., Caliguri, M.A., Bloomfield, C.D., Lander, E.S., 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* 286, 531–537.
- Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 1999. Gene selection for cancer classification using support vector machines. *Machine Learning* 64, 389–422.
- Haykin, S., 1999. *Neural networks—a comprehensive foundation*. Prentice-Hall Int. Inc., Englewood Cliffs, NJ.
- Holter, N.S., Mitra, M., Maritan, A., Cieplak, M., Fedoroff, N.V., Banavar, J.R., 2001. Dynamic modeling of gene expression data. *Proceedings of the National Academy of Science USA* 98, 1693–1698.
- Jackson, J.E., 1991. *A User's Guide to Principal Components*. Wiley, New York.
- Lipshutz, R.J., Fodor, S.P.A., Gingeras, T.R., Lockhart, D.J., 1999. High density synthetic oligonucleotide arrays. *Nature Genetics* 21, 20–24.
- Lockhart, D.J., Dong, H., Byrne, M.C., Follettie, M.T., Gallo, M.V., Chee, M.S., Mittmann, M., Wang, C., Kobayashi, M., Horton, H., Brown, E.L., 1996. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology* 14, 1675–1680.
- Raychaudhuri, S., Stuart, J.M., Altman, R., 2000. Principal components analysis to summarize microarray experiments: application to sporulation time series. In: Altman, R.B., Lauderdale, K., Dunker, A.K., Hunter, L., Klein, T.E. (Eds.), *Proceedings of Pac. Symposium Biocomput.* 2000. World Scientific, Singapore, pp. 455–466.
- Schadt, E.E., Cheng, L., Cheng, S., Wong, W.H., 1999. Analyzing high-density oligonucleotide gene expression array data. *Journal of Cellular Biochemistry* 80, 192–202.
- Schena, M., et al., 1995. Quantitative monitoring of gene expression patterns with a cDNA microarray. *Science* 270, 467–470.

¹Fig. 9 presents results of RFE instead of RFR method because RFE method is less time consuming and it was easier to perform the permutation test.

- Sebestyen, G.S., 1962. Decision making processes in pattern recognition. Macmillan, New York.
- Shalon, D., Smith, S.J., Brown, P.O., 1996. A DNA micro-array system for analyzing complex DNA samples using two-color fluorescent probe hybridization. *Genome Research* 6, 639–645.
- Simek, K., 2003. Properties of singular value decomposition based dynamical model of gene expression data. *International Journal of applied mathematics and computer science* 13 (8), 337–346.
- Simek, K., Kimmel, M., 2002. A note on estimation of dynamics of multiple gene expression based on singular value decomposition. *Mathematical Biosciences* 182, 183–199.
- Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., Futcher, B., 1998. Comprehensive identification of cell cycle-regulated genes of the yeast *saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell* 9, 3273–3297.
- Szabo, A.K., Boucher, W.L., Carroll, L.B., Klebanov, A.D., Tsodikov, A.Y., Yakovlev, 2002. Variable selection and pattern recognition with gene expression data generated by the microarray technology. *Mathematical Biosciences* 176, 71–98.
- Vapnik, V., 1995. *The Nature of Statistical Learning Theory*. Springer, New York.
- Wall, M.E., Dyck, P.A., Brettin, T.S., 2001. SVDMAN-singular value decomposition analysis of microarray data. *Bioinformatics* 17, 566–568.