Metodološki zvezki, Vol. 6, No. 1, 2009, 51-67

Item Response Theory Modeling for Microarray Gene Expression Data

Andrej Kastrin[1]

Abstract

The high dimensionality of global gene expression profiles, where number of variables (genes) is very large compared to the number of observations (samples), presents challenges that affect generalizability and applicability of microarray analysis. Latent variable modeling offers a promising approach to deal with high-dimensional microarray data. The latent variable model is based on a few latent variables that capture most of the gene expression information. Here, we describe how to accomplish a reduction in dimension by a latent variable methodology, which can greatly reduce the number of features used to characterize microarray data. We propose a general latent variable framework for prediction of predefined classes of samples using gene expression profiles from microarray experiments. The framework consists of (i) selection of smaller number of genes that are most differentially expressed between samples, (ii) dimension reduction using hierarchical clustering, where each cluster partition is identified as latent variable, (iii) discretization of gene expression matrix, (iv) fitting the Rasch item response model for genes in each cluster partition to estimate the expression of latent variable, and (v) construction of prediction model with latent variables as covariates to study the relationship between latent variables and phenotype. Two different microarray data sets are used to illustrate a general framework of the approach. We show that the predictive performance of our method is comparable to the current best approach based on an all-gene space. The method is general and can be applied to the other high-dimensional data problems.

1 Introduction

With decoding of the human genome and other eukaryotic organisms molecular biology has entered into a new era. High-throughput technologies, such as genomic microarrays can be used to measure the expression levels of essentially all the genes within an entire genome scale simultaneously in a single experiment and can provide information on gene functions and transcriptional networks (Cordero, Botta, and Calogero, 2007). Microarray experiments can lead to a better understanding of the genetic pathways involved in phenotype variations and diagnosis of diseases and therefore better management.

[1] Graduate study in Statistics, University of Ljubljana; University Medical Centre Ljubljana, Institute of Medical Genetics, Šlajmerjeva 3, SI-1000 Ljubljana, Slovenia; Faculty of Information Studies, Sevno 13, SI-8000 Novo mesto, Slovenia; andrej.kastrin@guest.arnes.si

A short version of this work was presented at the Applied Statistics 2008 conference, Ribno, Slovenia, September 21-24, 2008.

The gene expression data can be represented in a matrix notation. Columns are used to represent p genes and rows are used to represent expression levels in n biological samples. From the statistical point of view a gene is a variable and a biological sample is an observation (Roberts, 2008). The major challenge in microarray data analysis is due to their size, where the number of variables p far exceeds the number of observations n, commonly known as the large p, small n problem. This takes it difficult or even impossible to apply directly classical statistical methods and machine learning algorithms to the analysis of microarray data. In such tasks, class prediction is quite different from typical data mining problem, in which every class has a large number of examples (Mitchell, 1997). For example, in linear discriminant analysis, the covariance matrix is singular when $n < p$ (Hastie, Tibshirani, and Friedman, 2001). Another example is regression, where $n < p$ leads to ill-posed problem because the ordinary least squares solution is not unique (Nguyen and Rocke, 2004). In practical applications the common problem with large number of variables over few samples is over-fitting the data. That is, the predicted model can fit the original data well, but may predict poorly for new data. The usual way to handle the problem is to reduce the number of variables by using variable selection methods or projecting them to their lower dimensions. Although statistical analysis dealing with microarray data has been one of the most investigated areas, there are only a few papers discussing this problem. As Fan and Li (2006) claimed, the high-dimensional data analysis will be one of the most important research topics in statistics in the nearest future.

There are two major ways to handle high-dimensional data in the classification framework. The first approach is to eliminate redundant or irrelevant variables and select the most informative subset of differentially expressed genes to improve the generalization performance. The most commonly used procedures of gene selection are based on simple statistical tests (e.g., fold change, t-test, ANOVA, etc.), which are calculated for all genes individually, and genes with the best scores are selected for further processing (Jeffery, Higgins, and Culhane, 2006). The advantages of this approach are its simplicity and interpretability. The standard statistical methods are then applied on this small subset of genes. In biostatistical literature, this approach is often denoted as variable selection or gene filtering. An alternative approach to overcome the problem of dimensionality is application of different multivariate techniques. Cluster analysis has become the most widely used such technique (Do and Choi, 2008).

The aim of cluster analysis is to reveal latent structure and identify interesting patterns in the underlying data. More formally, clustering is the process of grouping objects into a set of disjoint groups or clusters, so that objects within a cluster have high similarity to each other, while objects in separate clusters are more dissimilar (Everitt et al., 2001). In terms of microarray data analysis clustering have been applied to organize gene expression data by grouping genes or samples with similar expression. Coexpressed genes are grouped in clusters based on their expression profiles. Genes with similar expression patterns might share biological function or might be under common regulatory control (Do and Choi, 2008).

Class prediction is a crucial aspect of microarray studies and plays important role in the interpretation of microarray data. With class prediction, the emphasis is on developing a classification function that enables one to predict the class membership of a new sample based on its expression profile. Although cluster analysis is appropriate tool for

exploratory data analysis and discovering patterns in gene expression data, it is not the most suitable for class prediction (Simon, 2003). Since cluster analysis provides categorical groupings on a nominal scale as output, it differs from other multivariate approaches such as principal component analysis that is based on continuous variables (components) and thus directly applicable in prediction modeling. However, hierarchical clustering (HC) forms clusters in a hierarchical fashion resulting in a tree-like dendrogram, indicating which genes are grouped together at various levels of their similarity. Cluster analysis does not provide statistically valid quantitative information about each cluster of genes, suitable for numerical description and discrimination between clusters. More specifically, there is no information about how much one cluster of genes differs from the other cluster of genes on a continuous measurement scale.

The challenge now is the integration of such qualitative genomic information derived from cluster analysis into prognostic models that can be applied in clinical settings to improve the accuracy of class prediction in a disease understanding and management. Latent variable modeling, especially item response theory (IRT) provides an appealing framework for such task. By latent variable model we mean any statistical model that relates a set of observer or manifest variables to set of latent variables (van der Linden and Hambleton, 1996). It is assumed that the responses on the manifest variables are the result of an individual's position on the latent variable. The conceptual framework on latent variable modeling originates from psychometrics, starting at the beginning of the 20th century (Fischer and Molenaar, 1995). Utility of these models in biomedical research has only quite recently been recognized (Rabe-Hesketh and Skrondal, 2008). Here, we use HC to describe individual genes in terms of a small number of latent variables. Samples can then be analyzed by summarizing their gene expression patterns in terms of expression patterns of the latent variables. These latent variables provide for dimension reduction and summarize patterns of covariation among the original genes. We assumed that genes with similar expression (i.e., genes in the same cluster) determine one latent variable, and that the item response model can be used to estimate the value of this latent variable.

Our main objective was to extract several latent variables associated with each microarray data set. Next, we were interested in constructing prediction models to evaluate the discrimination power of the latent variables for observed phenotypes. We examined the class prediction model both in the settings with latent variables and when no latent reduction of data was used. To make our evaluation more relevant to practice, the validation of the approach was performed on two different microarray data sets.

2 Methods

In this section we explained the microarray data sets used and the proposed method. The whole algorithm comprises six main steps: (i) preprocessing of the microarray data of p genes and n samples and selection of smaller number of genes p_s that are most differentially expressed between samples; (ii) dimension reduction using cluster analysis to cluster p_s genes into c clusters, which can be identified as latent variables; (iii) discretization of the gene expression matrix $p_s \times n$ into binary variables; (iv) fitting the Rasch item response model for genes in each of the c_i cluster respectively to estimate the expression of latent variable; (v) construction of prediction model with latent variables as covariates

to study the relationship between latent variables and corresponding phenotype.

2.1 Data sets and preprocessing

In order to test the applicability of our method for microarray data in general and to benchmark our approach, two different data sets have been used in our experiments. All data sets were selected from publicly available microarray data.

The first data set analyzes the expression profiles of bone marrow or peripheral blood obtained from acute leukaemia patients and contains probes for 3051 genes on Affymetrix platform (Golub et al., 1999). The data set in publicly available as a part of Bioconductor's multtest package. In the original publication the patients are divided into training and test set; only the training set of 38 samples was used in our experiment. Samples correspond to two different types of leukaemia: acute lymphoblastic (ALL; 27 patients) and myeloid leukaemia (AML; 11 patients). Although the separation between the two conditions is more pronounced that in most other public available microarray experiments, this data set can still be considered as a benchmark. The ALL/AML data set has been extensively used by other scientists for testing different methods of analysis.

The second study examines global changes in gene expression in the blood samples of Huntington's disease patients, compared with normal controls. The data set is publicly available from NCBI GEO repository (GSE1751; Barrett et al., 2007). The data set contains expression profiles for more than 33,000 of the best characterized human genes, measured using Affymetrix technology. In total, global gene expression profiles of 12 symptomatic Huntington's disease patients, 5 presymptomatic individuals carrying the Huntington's disease mutation, and 14 healthy controls are available. Due to uncertain phenotype state, presymptomatic samples were omitted from further processing. This resulted in a data set containing 26 samples. Further information describing this data set is available in Borovecki et al. (2005).

The raw gene expression profiles were log transformed (log_{10}) to stabilize the variance and adjust the extremes, and standardized. In the microarray analysis it is not expected that all the genes are expressed at the biologically meaningful levels. Because of a large number of genes measures, many genes in the original data set are irrelevant to the analysis. Therefore, we used rank product (RP), a nonparametric method, which is based on calculating rank products to identify most differentially expressed genes (Breitling et al., 2004). RP provides a powerful test statistic for defining differentially expressed genes in microarray experiment, because it does not depend on an estimate of the gene-specific measurement variance and is therefore particularly powerful when only a small number of samples are available, as is in our case. For ALL/AML data set we than selected 50 genes that are mostly overexpressed in ALL, and 50 genes that are mostly overexpressed in AML samples. Similarly, we extracted 50 genes that are mostly overexpressed in Huntington's disease patients and 50 genes that are mostly overexpressed in healthy controls. After data preparatory steps both data sets contain 100 most differentially expressed genes. The choice to use only 100 most informative genes in the predictor set was totally arbitrary. The number was seemed likely to be large enough to be robust against noise, and was small enough to be applied for demonstration purposes.

2.2 Hierarchical clustering

Among variety of algorithms available for clustering microarray data we choose HC to detect similarity relationships in gene expression, because it was the simplest method that provided a partition of our data into clusters. Moreover, HC is a widely used method to group genes sharing similar expression levels under different experimental conditions in microarray experiments (Do and Choi, 2008). Two key steps are involved in HC analysis. The first is the measurement of the object dissimilarity, and the second is to group the objects based upon the resulted distances. The final result of the algorithm is a binary tree of clusters called a dendrogram, which shows how the clusters are related to each other. Different aggregation methods can be used for the construction of the dendogram generally leading to different tree topologies and to various cluster definitions. In this study we used Euclidean distance as a measure of similarity between variables and Ward's minimum variance algorithm to group genes into clusters. The Euclidean distance between a pair of variables $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$ is defined as

$$d(X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \tag{2.1}$$

The dissimilarity matrix was used as an input in the Ward clustering algorithm (Ward, 1963). Empirical studies have shown that Ward's method is widely used for clustering microarray data (Belacel, Wang, and Cuperlovic-Culf, 2006). It is based on the consecutive merger of two groups into a new group. Since we performed our analysis on the basis of dissimilarity matrix, the Ward clustering algorithm computes the distance between groups according to Lance-Williams recurrence formula. We refer the reader to Everitt et al. (2001) for details.

The key parameter to determine in proposed approach is the number of clusters c. Parameter c can be obtained by cutting the dendrogram at a certain level to obtain the desired number of clusters. There is no standard algorithm for choosing a cut-off point for dendrogram. The choice is often made by visual inspection. The value of this parameter was selected in the range of $c = 2, 3, 4, 5$ for both experiments. If this value is selected too high, to our experiences the biological meaning of the latent variables is difficult to assign.

2.3 Discretization

The gene expression values obtained from microarray experiment are continuous real numbers. To apply the proposed item response model to the microarray data, gene expression levels need to be binarized. In order to represent microarray data in a binary relation, discretization of the continuous gene expression values into a finite set of values is required. Intuitively, we would like to discretize the gene expression values in such a way that two close values share the same attribute.

Among available algorithms for discretizing continuous variable we used equal frequency discretization (EFD) in our application (Dougherty, Kohavi, and Sahami, 1995). EFD divides the range of observed continuous values for a variable into k bins so that

each bin contains n/k (possibly duplicated) adjacent values. k is a user-supplied parameter. In our settings the Rasch model requires binary discretization, so we set $k = 2$. To denote gene expression levels, we use two discrete values, 0 and 1, ranging from low to high expression levels.

2.4 Item response model

The Rasch model, sometimes referred to as the one parameter logistic model (1PL), is a simple latent variable model where binary responses on each of J items are related to a single continuous latent variable U, which is assumed to have the same effect on all item responses (van der Linden and Hambleton, 1996). In terms of latent trait theory, we can define each gene in a latent variable as an item, and each biological sample as a person. In this manner the expression level is defined as the response of a given biological sample to a given gene.

For a given sample, using expression profiles of genes in a given cluster, we estimated the expression of a latent variable by fitting a Rasch model. Assume that we have I observed variables and J samples. In this setting, the observed variables are genes. U_{ij} represents the continuous unobserved expression of the sample j. The response of sample j to the gene i is $U_{ij} = 1$ if gene expression level in sample j is high and $U_{ij} = 0$ if gene expression level in sample j is low. The Rasch model predicts the probability of high gene expression level for sample j on gene i (i.e., $P(U_{ij} = 1)$), as follows:

$$P(U_{ij} = 1|\theta) = \frac{1}{1 + exp\{-(\theta - \delta_i)\}} \tag{2.2}$$

for $i = 1, \ldots, I$, and $j = 1, \ldots, J$, where θ is the sample parameter that expresses the latent variable of the sample j that is measured by the I genes. δ_i is the gene-specific parameter, which expresses the attractiveness of the gene i. Genes with large values of δ_i have lower proportions of samples with high gene expression responses. The gene parameters can be estimated based on the conditional likelihood. Details on the conditional likelihood estimation of the item parameters can be found in van der Linden and Hambleton (1996).

The final step of the item response modeling is to derive latent scores from the gene responses for each sample. Latent scores are summary measures of the posterior distribution $p(z_j|u_j)$, where z_j denotes the vector of latent scores and u_j the vector of manifest gene responses (van der Linden and Hambleton, 1996). Latent scores are usually calculated as the mode of the posterior distribution for each sample:

$$\hat{z}_j = arg \max_z \{p(x_j|z_j; \theta)p(z_j)\}. \tag{2.3}$$

Latent scores are then be used as predictor variables for latent variables in machine learning setup.

2.5 Predictive modeling

To provide a basis for predictive modeling (i.e., for discriminating between patients with ALL/AML and between patients with Huntington's disease and healthy controls), we used two popular machine learning algorithms, namely support vector machines (SVM) classifier and decision trees (DT).

SVM classifier is a very popular machine learning technique (Noble, 2006). SVM has shown excellent empirical performance on many classification problems across different domains. As suggested by a large body of literature to date, SVM can be considered 'best of class' algorithms for classification of microarray data (Wu et al., 2007). SVM provide a framework for supervised learning which is robust against overfitting and capable of handling non-separable cases. A SVM model creates a separating hyperplane, a higher-dimensional generalization, so that it optimally discriminates between two classes. During a minimization procedure, the hyperplane is tuned so that the SVM model generalization error is minimized, thus achieving an optimal solution to the classification problem (Vapnik, 1999). The proposed prediction method was implemented using LibSVM library[2]. SVM require only a small number of labeled training data as input. This training data is than used to learn parameters of the categorization model. In the validation phase, the effectiveness of the model is tested on previously unseen examples.

DT is the most popular inductive learning algorithm (Mitchell, 1997). The nodes of the tree correspond to attribute tests, the links to attribute values and the leaves to the classes. To induce a DT, the most important attribute is selected and placed at the root; one branch is made for each possible attribute value. The process is repeated recursively for each subset until all instances at a node have the same classification, in which case a leaf is created. To prevent over-training DT are typically pruned. DT are interesting because we can see what features were selected for the top levels of the trees. In our application we use Quinlans C4.5 algorithm (Quinlan, 1993) to generate decision trees.

2.6 Classification performance evaluation metrics

To substantiate the generalizability of the prediction models based on machine learning algorithms, a statistical validation of classifiers is necessary. We used several classification performance metrics. Predictive accuracy (*Acc*) is the overall correctness of the prediction and was calculated as the sum of correct classifications (*TP* and *TN*) divided by the total number of classifications ($TP + FP + FN + TN$). In this formula, *TP* is the number of true positives (correctly predicted as positive samples); *FP* denotes the number of false positives (incorrectly predicted as positive samples); *FN* refers to the number of false negatives (incorrectly predicted as negative samples), and *TN* is the number of true negatives (correctly predicted as negative samples).

Besides prediction accuracy, the performance measures of sensitivity and specificity were used to evaluate the results of the classification algorithm. Sensitivity (*Sen*) and specificity (*Spe*) were calculated as:

$$Sen = \frac{TP}{TP + FN} \quad \text{and} \quad Spe = \frac{TN}{TN + FP}. \tag{2.4}$$

[2]http://www.csie.ntu.edu.tw/∼cjlin/libsvm/

Note that neither sensitivity nor specificity, took individually, constitute an exhaustive measure. A single value that summarizes both measures into a better one is Matthews correlation coefficient (*MCC*), which is also known as phi coefficient. It can be computed as follows:

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}}. \tag{2.5}$$

The range of *Acc*, *Sen*, and *Spe* measures falls between 0 and 1, with 1 indicating the best quality of the classifier. A *MCC* of $+1$ represents a perfect prediction, 0 an average random prediction, and -1 an inverse prediction.

We used 10-fold cross-validation to evaluate the performance of the classifiers. Cross-validation is especially attractive in applications with relatively limited amounts of data as all observations are used as both training and test data. Cross-validation is a commonly used procedure for evaluating the quality of statistical models. In brief, in 10-fold cross-validations, the available data are partitioned into 10 disjoint subsets of approximately equal size. A set of 10 classifiers is then constructed; each classifier is trained on a different combination of 9 subsets and tested on the remaining subset. In order to qualify and compare the performance of each classifier, we calculated accuracy, precision, and recall measure. Values of evaluation measures were averaged over runs for further reporting. The average test performance of the 10 classifiers generally provides a good estimate of the generalization performance of a single classifier trained on the entire data set.

2.7 Statistical comparison among classifiers

When comparing two classifiers, it is important to assess whether the observed difference in classification performance is statistically significant or simply due to chance. We assessed significance of differences in classification performance using McNemar's test (Dietterich, 1998). We constructed the contingency table assuming there are two classifiers A and B, where n_{11} is the frequency of cases misclassified by both classifiers; n_{21} is the frequency of cases misclassified by classifier B but not A; n_{12} is the frequency of cases misclassified by algorithm A but not B; n_{22} are correctly classified cases by both classifiers. McNemar's test statistics (z_0) is computed as follows:

$$z_0 = \frac{n_{21} - n_{12}}{\sqrt{n_{21} + n_{12}}}. \tag{2.6}$$

Square of McNemar's statistics (z_0^2) follows chi-square distribution with 1 degree-of-freedom. The null hypothesis assumes that the performance of two different classifiers is the same (i.e., $n_{12} = n_{21}$). We also calculate corresponding p-value. Small p-value suggest that null hypothesis is unlikely to be true, therefore we may reject the null hypothesis if the probability that $z_0^2 \geq 3.84$ is less than 0.05.

2.8 Software

The computations were carried out in the R software environment for statistical computing and graphics (R Development Core Team, 2008). HC was performed using generic

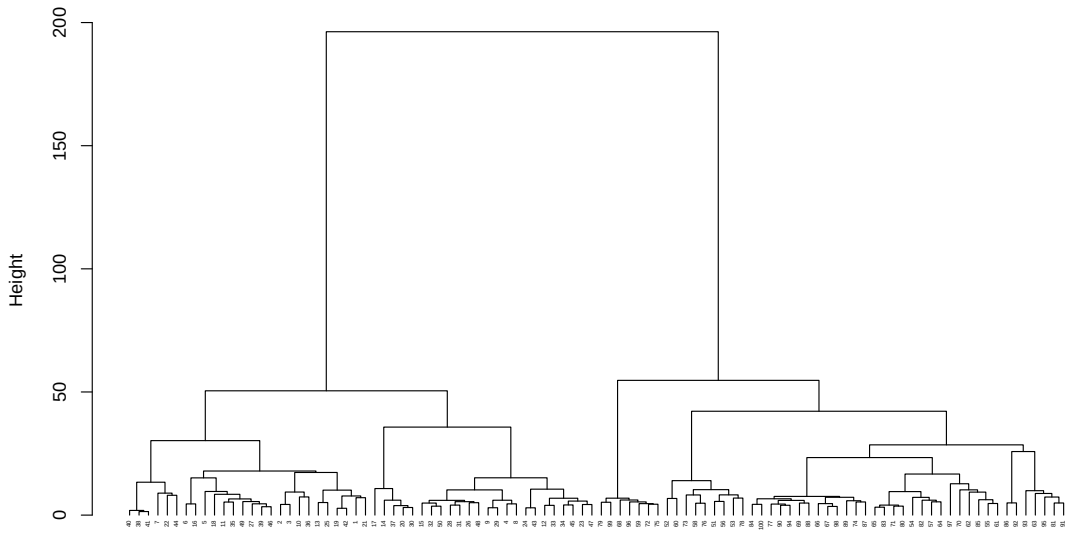

Figure 1: Dendrogram visualization of hierarchical clustering on selected genes for Golub data set.

`hclust` function. Discretization of numerical gene expression values according to equal width heuristics was performed by `disc` function in minet package. For latent variable modeling the bundle of functions in ltm package was used. Rasch parameters were estimated by conditional maximum likelihood algorithm. RWeka, an interface to Weka machine learning suite (Witten and Frank, 2005), was used for building SVM classifier. We used SVM implementation in the LibSVM software library with polynomial kernel. For all other parameters of the SVM algorithm, the default values set in Weka were used. For decision trees modeling we used C4.5 learning scheme implemented as J48 class in Weka. Statistical evaluation was done using `mcnemar.test` function in R.

3 Results

In this section, we report the results from two experiments that study the usefulness of the latent variable approach to predict disease status. First, we examined the cluster solution of HC in order to define appropriate set of latent variables. Next, we examined the classification performance over two prediction models (SVM and DT) where latent variables were used as predictors. For meaningful and well-grounded evaluation, we directly compared the classification results based on latent variable approach to classical approach, where full data set (all-genes model), without latent variable induction, was used.

The visualizations of the HC for both data sets are presented in Figure 1 and Figure 2. When performing HC, we needed to define threshold (i.e., appropriate height) to cut the tree of clusters into individual clusters on the dendrogram, which determines the number of clusters. However, due to exploratory orientation of our study, we decided to examine the behaviour of classification performance by setting the number of clusters $c = 2, 3, 4, 5$, respectively. This is very important, because we were interested in how the number of clusters, could influence the results.

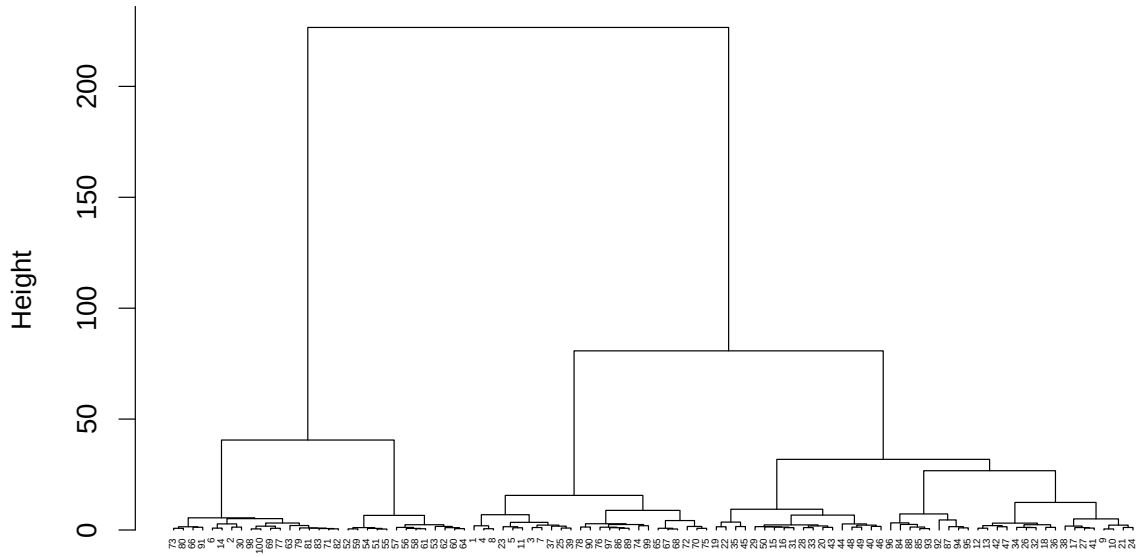

Figure 2: Dendrogram visualization of hierarchical clustering on selected genes for Huntington's disease data set.

Table 1: Mean values and 95% confidence intervals of estimated latent variable scores for Golub data set.

		ALL			AML			
Model	LV	M	CI_l	CI_u	M	CI_l	CI_u	p
1	A	-0.41	-0.60	-0.21	1.46	1.15	1.78	< 0.01
	B	0.34	0.17	0.50	-1.43	-1.63	-1.23	< 0.01
2	A	-0.41	-0.60	-0.21	1.46	1.15	1.78	< 0.01
	B	0.42	0.25	0.59	-1.40	-1.60	-1.19	< 0.01
	C	0.37	0.17	0.58	-0.96	-1.13	-0.79	< 0.01
3	A	-0.47	-0.65	-0.29	1.30	0.99	1.60	< 0.01
	B	-0.41	-0.64	-0.18	1.09	0.80	1.38	< 0.01
	C	0.42	0.25	0.59	-1.40	-1.60	-1.19	< 0.01
	D	0.37	0.17	0.58	-0.96	-1.13	-0.79	< 0.01
4	A	-0.47	-0.65	-0.29	1.30	0.99	1.60	< 0.01
	B	-0.41	-0.64	-0.18	1.09	0.80	1.38	< 0.01
	C	-0.07	-0.30	0.16	-0.83	-0.99	-0.68	< 0.01
	D	0.58	0.41	0.75	-1.27	-1.47	-1.06	< 0.01
	E	0.37	0.17	0.58	-0.96	-1.13	-0.79	< 0.01

Legend: ALL - acute lymphoblastic leukaemia; AML - acute myeloid leukaemia; LV - latent variable; M - Mean latent score; CI_l - lower confidence limit; CI_u - upper confidence limit; p - level of statistical significance of the Welch t-test.

Table 2: Mean values and 95% confidence intervals of estimated latent variable scores for Huntington's disease data set.

		HD			Ctrl			
Model	LV	M	CI_l	CI_u	M	CI_l	CI_u	p
1	A	0.18	−0.14	0.50	−0.15	−0.35	0.05	0.10
	B	0.14	−0.27	0.56	−0.12	−0.35	0.11	0.29
2	A	−0.52	−0.75	−0.29	0.44	0.15	0.74	< 0.01
	B	0.14	−0.27	0.56	−0.12	−0.35	0.11	0.29
	C	0.64	0.25	1.03	−0.55	−0.73	−0.37	< 0.01
3	A	−0.52	−0.75	−0.29	0.44	0.15	0.74	< 0.01
	B	0.18	−0.22	0.57	−0.15	−0.65	0.34	0.32
	C	0.64	0.25	1.03	−0.55	−0.73	−0.37	< 0.01
	D	0.02	−0.46	0.50	0.03	−0.45	0.51	0.98
4	A	−0.52	−0.75	−0.29	0.44	0.15	0.74	< 0.01
	B	0.18	−0.22	0.57	−0.15	−0.65	0.34	0.32
	C	−0.02	−0.06	0.03	0.02	−0.02	0.05	0.25
	D	0.85	0.58	1.12	−0.72	−0.89	−0.55	< 0.01
	E	0.02	−0.46	0.50	0.03	−0.45	0.51	0.98

Legend: HD - Huntington's disease; Ctrl - control group; LV - latent variable; M - Mean latent score; CI_l - lower confidence limit; CI_u - upper confidence limit; p - level of statistical significance of the Welch t-test.

Next, we fit the Rasch model for a set of genes in each of the c clusters respectively. Table 1 and Table 2 summarized descriptive statistics of estimated latent variable scores for different number of extracted latent variables for each data set. In all experimental settings, the latent variables concerning different clusters were not statistically significant correlated. To indicate the differences between experimental conditions the results are presented separately for each experimental group. The Welch t-test was computed to compare mean latent scores between experimental conditions within a given latent variable. In the case of Golub data set all latent variables showed statistical significant differences between ALL and AML type of disease. On the other hand, the differences on Huntington's disease data set are less discriminative. Only two latent variables (A and C) indicate statistically significant differences between patients with Huntington's disease and healthy controls.

The evaluated scores on latent variables were used as predictors in the machine learning setup. The output of the prediction model is a list of predicted classes for all the samples. The comparison of our method with two classifiers with respect to the classification accuracy, sensitivity, specificity, and *MCC* is summarized in Table 3 and Table 4. For each parameter calculation, 10 runs were performed and the averages were calculated.

SVM were able to classify the ALL/AML data without loss of performance. According to classification results the optimal number of latent variables was found to be $c = 2$ since it gave the highest classification performances while retaining the simplicity of the solution. Overall classification results based on DT modeling are lower in comparison to SVM results.

Table 3: Classification results for Golub data set.

	SVM					DT				
	c					c				
	2	3	4	5	Full	2	3	4	5	Full
Acc	1.00	1.00	1.00	1.00	1.00	0.97	0.97	0.97	0.97	0.95
Sen	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.96
Spe	1.00	1.00	1.00	1.00	1.00	0.91	0.91	0.91	0.91	0.91
MCC	1.00	1.00	1.00	1.00	1.00	0.94	0.94	0.94	0.94	0.87

Legend: SVM - support vector machines; DT - decision trees; c - number of latent variables in prediction model; Full - predictor model with all selected genes; *Acc* - accuracy; *Sen* - sensitivity; *Spe* - specificity; *MCC* - Matthews correlation coefficient.

Table 4: Classification results for Huntington's disease data set.

	SVM					DT				
	c					c				
	2	3	4	5	Full	2	3	4	5	Full
Acc	0.73	0.92	0.96	1.00	1.00	0.46	0.88	0.88	0.96	0.96
Sen	0.75	0.92	0.92	1.00	1.00	0.00	0.92	0.92	1.00	1.00
Spe	0.71	0.93	1.00	1.00	1.00	0.86	0.86	0.86	0.93	0.93
MCC	0.46	0.85	0.92	1.00	1.00	−0.27	0.77	0.77	0.93	0.93

Legend: SVM - support vector machines; DT - decision trees; c - number of latent variables in prediction model; Full - predictor model with all selected genes; *Acc* - accuracy; *Sen* - sensitivity; *Spe* - specificity; *MCC* - Matthews correlation coefficient.

Classification results for Huntington's disease data set for SVM and DT algorithms, respectively, indicated that both algorithms had difficulty learning task when the number of latent variables is low ($c = 2$); the SVM scored a classification accuracy of $Acc = 0.73$, while the DT prediction model scored with accuracy of $Acc = 0.46$. The performance measures increases as the number of latent variables increases. The overall performance of SVM was better that with DT algorithm.

Next, the performance results of classification prior to latent variable induction are also summarized in Table 3 and Table 4. Although the numbers appears to be high, due to a larger set of predictors it is not unusual that the classification accuracy would be 100%. To evaluate the statistical significance in performance between classifiers with all-genes as predictors and latent variable models, we used McNemar's test as described previously in the Methods section. McNemar's test showed that only two latent models with two latent variables as predictors in the case of Huntington's disease data set were outperformed by the all-genes model at $p < 0.01$, regardless the prediction method (i.e., SVM or DT) used.

According to a 10-fold cross-validation results, the SVM model overall outperformed the DT model.

4 Discussion

Traditional approaches to microarray data analysis focus on identifying individual marker genes, which are associated with a phenotype of interest, and on using them to build classification and prediction models for samples whose phenotype may be unknown. Herein we introduce a latent variable approach to reduce the dimensionality of microarray gene expression data to a small number of latent variables, which could reduce noise while still capturing the main biological properties of the original data. The results presented in this paper illustrate that latent variable approach provides an effective and reliable method for dimensional reduction of microarray data.

The method, as introduced here, has a number of advantages over classical approaches to the analysis of microarray data. Latent variable approach reduces dimensionality and summarizes the most noticeable features of a data set with coherent patterns shared by multiple coexpressed genes. Results indicate that class prediction using latent variables as covariates compares favourably with standard prediction model, where all individual genes are used in the predictor set. Slightly better classification performance in the case of Golub data set confirm the fact that the biological separation between the two conditions (ALL/AML) is more pronounced in Golub data set than in Huntington's disease data set (De Smet et al., 2004).

In contrast to approaches using principal component analysis or singular value decomposition, it yields a comprehensive representation of the original data. Moreover, latent variables in proposed approach are easier to interpret and analyze by domain experts. Classifiers build in latent variable, rather than in all-gene space, are more robust, reproducible, and generalizable across platforms and laboratories because the latent space projection can reduce noise and technology-based variation more than simple normalization (Tamayo et al., 2007).

Recently several machine learning approaches to class prediction in the domain of microarray analysis have been proposed (Ressom et al., 2008). Decision to use SVM and DT classifiers was totally arbitrary, based on a list of top 10 algorithms in machine learning (Wu et al., 2007). In this application, we observed that SVM outperforms DT, although the differences are not statistically significant. Although this is an important observation, it is consistent with previous empirical findings (e.g., Wang, 2005). Observed differences between algorithms can be explained by the capability of SVM to handle high-dimensional data based on Vapnik's statistical learning theory (Vapnik, 1999). In contrast, DT utilizes tree nodes to select only a small number of latent variables for classification.

The data used to derive the IRT scales required extensive transformations. IRT scaling assumes a dichotomous response of a gene expression level. Although our results indicate that there were no loss of information due to discretization step in our procedure, it is still the issue, if it is reasonable to consider gene expression discretely. Referring to the work of Sheng, Moreau, and De Moor (2003), who demonstrated the effectiveness of the Gibbs sampling to the biclustering of discretized microarray data, we argue that discretization may improve the robustness and generalizability of the prediction algorithm with regard to the high noise level in the microarray data. Following Hartemink (2001) the discretization of continuous gene expression levels is preferred for four reasons: (i) discretization offers the benefit of allowing the majority of the qualitative relationships between variables to be modelled while at the same time reducing the dimensionality of the problem;

(ii) gene transcription occurs in one of a small number of states (low/high, off/low/high, low/medium/high, off/low/medium/high, etc.); (iii) the mechanisms of cellular regulatory networks can be reasonably approximated by primarily qualitative statements describing the relationships between states of genes; (iv) discretization, as a general rule, introduces a measure of robustness against error. It is a worthwhile future project to study the performance of our method on different levels of discretization, using polytomous IRT models (e.g., partial credit model) that could model more than two states of gene expression.

The major dilemma coupled with class prediction studies is the measurement of the performance of the classifier. The classifier cannot be evaluated accurately when sample size is very low. Moreover, feature selection and feature extraction steps should be an integral part of the classifier, and as such they must be a part of the evaluation procedure that is used to estimate the prediction performance (Asyali et al., 2006). This is especially important issue because HC provided a basis for predictive modeling in our framework. HC is known to be quite sensitive to sampling error (Everitt et al., 2001). To address these issues, the future work should concern the application of cross-validation scheme to estimate the appropriate number of latent variables and re-randomization experimental design to stabilize performance measures. We should also concern fit analysis, which was not introduced in this work.

However, the proposed approach not only decreases the dimensionality of microarray data, but can also provide a powerful knowledge-based approach to the analysis of microarray data. This method is of general applicability, not limited to analyzing microarray gene expression data.

Acknowledgment

The author would like to acknowledge the support of the Junior Research Fellowship granted by Slovenian Research Agency. The author is also grateful to both reviewers for their useful suggestions and comments.

References

[1] Asyali, M.H., Colak, D., Demirkaya, O., and Inan, M.S. (2006): Gene expression profile classification: A review. *Current Bioinformatics*, **1**, 55-73.

[2] Barrett, T., Troup, D.B., Wilhite, S.E., Ledoux, P., Rudnev, D., Evangelista, C., Kim, I.F., Soboleva, A., Tomashevsky, M., and Edgar, R. (2007): NCBI GEO: mining tens of millions of expression profiles database and tools update. *Nucleic Acids Research*, **35**, D760-D765.

[3] Belacel, N., Wang, Q., and Cuperlovic-Culf, M. (2006): Clustering methods for microarray gene expression data. *OMICS: A Journal of Integrative Biology*, **10**, 507-531.

[4] Borovecki, F., Lovrecic, L., Zhou, J., Jeong, H., Then, F., Rosas, H.D., Hersch, S.M., Hogarth, P., Bouzou, B., Jensen, R.V., and Krainc, D. (2005): Genome-wide

expression profiling of human blood reveals biomarkers for Huntington's disease. *Proceedings of the National Academy of Sciences*, **102**, 11023-11028.

[5] Breitling, R., Armengaud, P., Amtmann, A., and Herzyk, P. (2004): Rank products: a simple, yet powerful, new method to detect differentially regulated genes in replicated microarray experiments. *FEBS Letters*, **573**, 83-92.

[6] Cordero, F., Botta, M., and Calogero, R.A. (2008): Microarray data analysis and mining approaches. *Briefings in Functional Genomics & Proteomics*, **6**, 265-281.

[7] De Smet, F., Moreau, Y., Engelen, K., Timmerman, D., Vergote, I., De Moor, B. (2004): Balancing false positives and false negatives for the detection of differential expression in malignancies. *British Journal of Cancer*, **91**, 1160-1165.

[8] Dietterich, T.G. (1998): Approximate Statistical Test For Comparing Supervised Classification Learning Algorithms. *Neural Computation*, **10**, 1895-1923.

[9] Do, J.H. and Choi, D.K. (2008): Clustering Approaches to Identifying Gene Expression Patterns from DNA Microarray Data. *Molecules and Cells*, **25**, 279-288.

[10] Dougherty, J., Kohavi, R., and Sahami, M. (1995): Supervised and unsupervised discretization of continuous features. In A. Prieditis and S.J. Russell (Eds): *Proceedings of the Twelfth International Conference on Machine Learning*, 194-202. Tahoe City, CA: Morgan Kaufman.

[11] Everitt, B.S., Landau, S., and Leese, M. (2001): *Cluster Analysis*. London: Arnold Publishers.

[12] Fan, J. and Li, R. (2006): Statistical challenges with high dimensionality: feature selection in knowledge discovery. In M. Sanz-Sole, J. Soria, J. L. Varona, and J. Verdera (Eds): *Proceedings of the International Congress of Mathematicians*, 595-622. Madrid: European Mathematical Society Publishing House.

[13] Fischer, G.H. and Molenaar, I.W. (Eds) (1995): *Rasch Models: Foundations, Recent Developments, and Applications*. Berlin: Springer.

[14] Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., and Lander, E. (1999): Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, **286**, 531-537.

[15] Hartemink, A.J. (2001): *Principled Computational Methods for the Validation and Discovery of Genetic Regulatory Networks*. Ph.D thesis. Boston, MA: Massachusetts Institute of Technology.

[16] Hastie, T., Tibshirani, R., and Friedman, J.H. (2001): *The Elements of Statistical Learning*. New York, NY: Springer.

[17] Jeffery, I.B., Higgins, D.G., and Culhane, A.C. (2006): Comparison and evaluation of methods for generating differentially expressed gene lists from microarray data. *BMC Bioinformatics*, **7**, 359.

[18] van der Linden, W.J. and Hambleton, R.K. (Eds) (1996): *Handbook of Modern Item Response Theory*. Berlin: Springer.

[19] Mitchell, T. (1997): *Machine Learning*. New York, NY: McGraw-Hill.

[20] Nguyen, D.V. and Rocke, D.M. (2004): On partial least squares dimension reduction for microarray-based classification: a simulation study. *Computational Statistics & Data Analysis*, **46**, 407-425.

[21] Noble, W.S. (2006): What is a support vector machine? *Nature Biotechnology*, **24**, 1565-1567.

[22] Quinlan, R.J. (1993): *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.

[23] R Development Core Team (2008): *R: A Language and Environment for Statistical Computing*. Vienna: R Foundation for Statistical Computing.

[24] Rabe-Hesketh, S. and Skrondal, A. (2008): Classical latent variable models for medical research. *Statistical Methods in Medical Research*, **17**, 5-32.

[25] Ressom, H.W., Varghese, R.S., Zhang, Z., Xuan, J., and Clarke, R. (2008): Classification algorithms for phenotype prediction in genomics and proteomics. *Frontiers in Bioscience*, **13**, 691-708.

[26] Roberts, P.C. (2008): Gene expression microarray data analysis demystified. *Biotechnology Annual Review*, **14**, 29-61.

[27] Sheng, Q., Moreau, Y., and De Moor, B. (2003): Biclustering microarray data by Gibbs sampling. *Bioinformatics*, **19**, 196-205.

[28] Simon, R. (2003): Diagnostic and prognostic prediction using gene expression profiles in high-dimensional microarray data. *British Journal of Cancer*, **89**, 1599-1604.

[29] Tamayo, P., Scanfeld, D., Ebert, B.L., Gillette, M.A., Roberts, C.W., and Mesirov, J.P. (2007): Metagene projection for cross-platform, cross-species characterization of global transcriptional states. *Proceedings of the National Academy of Sciences*, **104**, 5959-5964.

[30] Vapnik, V.N. (1999): *The Nature of Statistical Learning Theory*. New York, NY: Springer.

[31] Wang Y., Tetko I.V., Hall, M.A., Frank, E., Facius, A., Mayer, K.F., and Mewes, H.W. (2005): Gene selection from microarray data for cancer classification - a machine learning approach. *Computational Biology and Chemistry*, **29**, 37-46.

[32] Ward, J.H. (1963): Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, **58**, 236-244.

[33] Witten, I.H. and Frank, E. (2005): *Data Mining: Practical Machine Learning Tools and Techniques*. San Francisco, CA: Morgan Kaufmann.

[34] Wu,X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., Zhou, Z.H., Steinbach, M., Hand, D.J., and Steinberg, D. (2008): Top 10 algorithms in data mining. *Knowledge and Information Systems*, **14**, 1-37.