REGULARISED k-MEANS CLUSTERING FOR DIMENSION REDUCTION APPLIED TO SUPERVISED CLASSIFICATION

Vladimir Nikulin, Geoffrey J. McLachlan
Department of Mathematics, University of Queensland, Brisbane, Australia
v.nikulin@uq.edu.au, gjm@maths.uq.edu.au

Keywords: clustering, cross-validation, gene expression data.

Abstract. Clustering methods provide a powerful tool for the exploratory analysis of high-dimensional, low-sample size data sets, such as gene expression microarray data. Unlike classification and regression, cluster analysis requires no response variable and thus falls into category of unsupervised learning methods. However, there are two major problems: stability of clustering and meaningfulness of centroids as cluster representatives. On the one hand, big clusters impose strong smoothing and possible loss of very essential information. On the other hand, small clusters are, usually, very unstable and noisy. Accordingly, they can not be treated as equal and independent representatives. To address the above problems, we propose regularisation to prevent the creation of super big clusters, and to attract data to existing small clusters. We demonstrate the effectiveness of this approach to the supervised classification of gene expression data.

1 Introduction

The analysis of gene expression data using clustering techniques has an important role to play in the discovery, validation, and understanding of various classes and subclasses of cancer [16]. One feature of microarray studies is the fact that the number of samples collected is relatively small compared to the number of genes per sample which are usually in the thousands. In statistical terms this very large number of predictors compared to a small number of samples or observations makes the classification problem difficult. An efficient way to solve this problem is by using dimension reduction statistical techniques [3].

The SVM-RFE (support vector machine recursive feature elimination) algorithm was proposed in [9] to recursively classify the samples with SVM and select genes according to their weights in the SVM classifiers. However, it was noted in [25] that the original SVM-RFE ranked the genes only once using all samples, and used the top ranked genes in the succeeding cross-validation for the classifier. This is a typical cross-validation (CV) scheme which will generate a biased estimation of errors. In correct CV scheme it is necessary to repeat feature selection for any CV loop which may be very expensive in terms of computational time.

Cluster analysis, an unsupervised learning method [24], is widely used to study the structure of the data when no specific response variable is specified. In contrast to the SVM-RFE, in the case of most of clustering algorithms we can perform feature selection only once.

Recently, several new clustering algorithms (e.g., graph-theoretical clustering, model-based clustering) have been developed with the intention to combine and improve the features of traditional clustering algorithms. However, clustering algorithms are based on different assumptions, and the performance of each clustering algorithm depends on properties of the input dataset. Therefore, the winning clustering algorithm does not exist for all datasets, and the optimization of existing clustering algorithms is still a vibrant research area [4].

The most popular clustering methods are hierarchical and k-means. However, several key issues in hierarchical clustering still need to be addressed. The most serious problem with this method is its lack of robustness to noise, high dimensionality, and outliers.

Hierarchical clustering algorithms are also expensive, both computationally and in terms of space complexity, and thus their applicability for the analysis of large datasets is limited.

The procedure k-means is relatively scalable and efficient when processing large datasets. In addition, k-means can converge to a local optimum in a small number of iterations. But, k-means still has several drawbacks. First, the user has to specify the initial number of clusters and the convergence centroids vary with the initial partitions. One of the characteristics of gene expression clustering is that prior knowledge is not available. Thus, in order to detect the optimal number of clusters, users have to run the algorithm repeatedly with different k values, compare the clustering results, and make a decision about the optimal number of clusters accordingly. For a large gene expression dataset, this extensive fine-tuning process is not practical. We can note here [10] and [14] where it was proposed to use as a stopping criterion normality of the data within any particular cluster. Usually, attempts to estimate the number of Gaussian clusters will lead to a very high value of k [26]. Most simple criteria such as AIC (*Akaike Information Criterion* [1]) and BIC (*Bayesian Information Criterion* [22]) either overestimate or underestimate the number of clusters, which severely limits their practical usability.

A second problem in k-means clustering is its sensitivity to noise and outliers. Gene expression data is noisy and has a significant number of outliers, and this can substantially influence the mean values and thus cluster positions. Finally, k-means often terminates at a local, possibly suboptimal, minimum.

An approach to test the stability of the clustering solutions has been proposed in [6], [13]. According to this approach, a given data set is repeatedly split into two nondisjoint sets. The sizes of the data sets are free parameters of the model. After clustering both data sets, a predictor is trained on one data set and tested on the other data set. Note also [23], where the random subspace method has been proposed to compute cluster stability scores.

The goal of statistical mixture models, implemented, for example, via the expectation-maximisation (EM) algorithm [17], is to identify or at least estimate unknown parameters (the means and standard deviations) of underlying probability distributions for each cluster in order to maximize the likelihood of the observed data distribution. The EM algorithm is a widely used approach for learning unobserved variables in machine learning. In probabilistic (soft) clustering, the EM algorithm attempts to approximate the observed distributions of values based on mixtures of different distributions in different clusters [16]. The results of EM clustering are different from those computed by k-means clustering. While the latter assigns observations to clusters by trying to maximize the distances between clusters, the EM algorithm computes classification probabilities rather then actual assignments of observations to cluster. In other words, in this method each observation belongs to each cluster with a certain probability. Of course, from the final result it is usually possible to determine the actual assignment of observations to clusters, based on the (largest) posterior probability of cluster membership.

In this paper, we shall focus on the use of cluster analysis to reduce the number of variables. The method is to be demonstrated on the supervised classification of the gene expression data.

2 Threshold-based clustering with merging

Let $(\mathbf{x}_t, y_t), t = 1, \ldots, n$, be a training sample of observations where $\mathbf{x}_t \in \mathbb{R}^m$ is m-dimensional vector of features, and y_t is binary label: $y_t \in \{-1, 1\}$. Boldface letters denote vector-columns, whose components are labelled using a normal typeface. Let us denote by $\mathbf{X} = \{x_{tj}, t = 1, \ldots, n, j = 1, \ldots, m\}$ matrix of explanatory variables.

The aim here is to use clustering to compress matrix $\mathbf{X}$ to a limited number of signatures or clusters without loss of essential information. We can employ well-known

Leader algorithm [11] (pp. 75-76) with two major modifications (1) making centroids flexible and (2) by including "backward" merging operation of the existing clusters which are close enough [18]. We shall refer to this modified algorithm as Algorithm 1. The algorithm requires an update of the matrix of distances between clusters (or centroids) after any transaction. This operation will double the required computation time assuming that the number of clusters remains constant. But, in fact, the backward operation may reduce significantly (subject to the properly selected regulation parameters) the number of clusters or clustering size and, as a consequence, the Algorithm 1 may be even faster comparing with its analogues without merging operation.

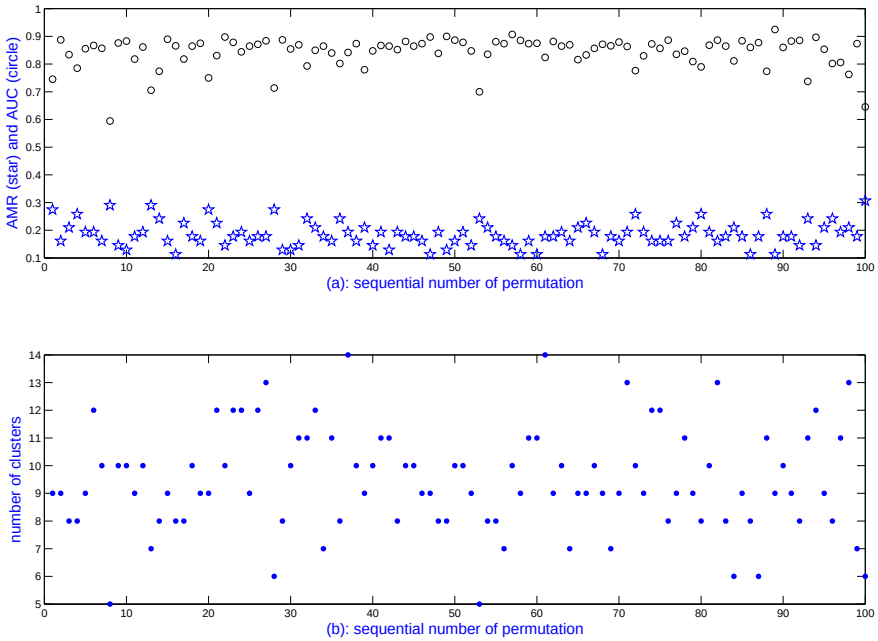

Figure 1: Algorithm 1: colon data; (a) AMRs (blue stars) and AUCs (black circles), (b) numbers of clusters (vertical axes) against 100 random permutations (horizontal axes, see for details Section 7.1).

Algorithm 1

1: Select forward and backward threshold parameters $H_F, H_B, H_F \geq H_B$, and distance Φ;

2: initialize $j := 1$, number of clusters $k := 1$, the first cluster with centroid $\mathbf{q}_k := \mathbf{x}_j$ as a first element in the training dataset;

3: $j := j + 1$, obtain a sequential data-instance $\mathbf{x}_j$ and compute

$$\begin{cases} D = \min\limits_{c=1..\tau} \Phi(\mathbf{x}_j, \mathbf{q}_c); \\ j = \operatorname{argmin}\limits_{c=1..\tau} \Phi(\mathbf{x}_j, \mathbf{q}_c); \end{cases}$$

4: if $D \leq H_F$, then assign $\mathbf{x}_j$ to the cluster c and recompute $\mathbf{q}_c$ as a sample average;

5: if $D > H_F$, then create a new cluster with centroid $\mathbf{q}_{k+1} := \mathbf{x}_j$, $k := k + 1$;

6: if $k \geq 2$, compute triangle matrix of distances between centroids, find minimal distance d_{min} and corresponding centroids;

7: merge 2 nearest clusters if $d_{min} < H_B, k := k - 1$;

8: repeat steps 3-7, until no instances are left in the training set.

As an outcome above algorithm produces matrix Q of k centroids, where we can expect that k is much smaller comparing with m, subject to the properly selected forward and backward regulation parameters H_F and H_B. All further analysis will be based on the matrix Q as a replacement of the original matrix $\mathbf{X}$.

3 Regularised k-means clustering

Stability in cluster analysis is strongly dependent on the data set, especially, on how well separated and how homogeneous the clusters are. Stability is a very important aspect in cluster analysis. Stability means that a meaningful valid cluster should not disappear easily if the data set is changed in a non-essential way [12]. On the one hand, big clusters impose strong smoothing and possible loss of very essential information. On the other hand, small clusters are, usually, very unstable and noisy. Accordingly, they can not be treated as equal and independent representatives.

The target of the following below regularisation is to prevent creation of super big clusters, and to attract data to existing small clusters.

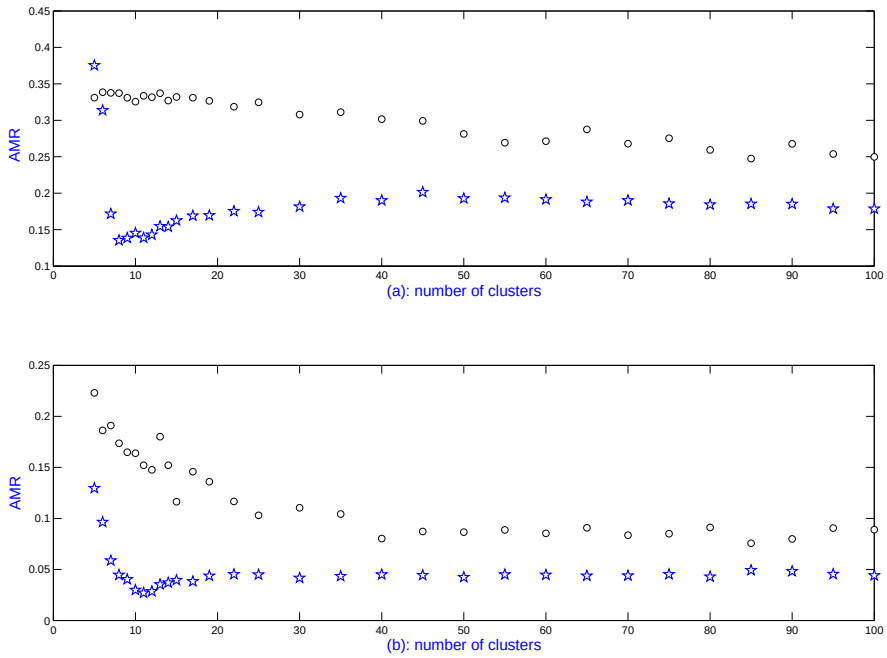

Figure 2: Algorithm 2: (a) colon data: AMRs (vertical axes) as a function of the numbers of clusters (horizontal axes): $\alpha = 0$ (black circles, no regularisation), $\alpha = 0.1$ (blue stars); (b) leukaemia data: $\alpha = 0$ (black circles, no regularisation), $\alpha = 0.1$ (blue stars).

Algorithm 2

1: Select number of clusters k, distance Φ and regulation parameter α;

2: split randomly available genes into k subsets (clusters) with approximately the same size;

3: compute an average (centroid) $\mathbf{q}_c$ for any cluster c;

4: compute maximum distance L between genes and centroids;

5: redistribute genes according to

$$\Phi(\mathbf{x}_j, \mathbf{q}_c) + R_c,$$

where regularisation term $R_c = \frac{\alpha \cdot L \cdot \#c}{m}$, $\#c$ is the size of cluster c at the current time, m is the total number of genes;

6: recompute centroids;

7: repeat steps 5-6, until convergence (that means stable behavior of the target function).

Remark 3.1 *There may be a situation in the Step 3 of the above algorithm that some of the clusters are empty. In this situation, the coordinates of the corresponding centroids were generated using standard uniform random numbers generator. On the one hand, we want to penalize the system for having empty clusters. On the other hand, we hope that randomly generated centroid will attract new data to the cluster.*

3.1 On the differences between Algorithms 1 and 2

Let us consider a toy example where the elements of the dataset $\mathbf{X}$ are distributed nearly uniformly around some n-dimensional area. In this case $\mathbf{X}$ may be regarded as inseparable in the sense of the k-means clustering, and it will be logical to split $\mathbf{X}$ into several approximately equal (in the sense of number of internal data) subsets using the Algorithm 1. Note that in this particular example the Algorithm 2, used without proper regularisation, may produce as an outcome one super big cluster.

As a second toy example, let us consider the case with three clusters C_1, C_2 and C_3, where all three clusters are well separable by the Algorithm 2. Suppose that the clusters C_1 and C_2 are very important, but small in size spatially. Also, we will assume that the clusters C_1 and C_2 are close. Third cluster C_3 is very large spatially and the information containing in C_3 is not significant. Now, let us consider performance of the Algorithm 1 as a function of the forward threshold parameter H_F. In order to separate clusters C_1 and C_2 the parameter H_F must be small. As an outcome, we will have large number of clusters (see last line of the following below Table 1, leukaemia case).

Note that we don't need regularisation in the case of the Algorithm 1, because the forward and backward threshold parameters (as a natural part of the algorithm) are playing the role of efficient regulators.

4 Regularised linear regression model

In supervised classification algorithms, a classifier is trained with all the labelled training data and used to predict the class labels of unseen test data. In other words, the label y_t may be hidden, and the task is to estimate it using vector of features. Let us consider the most simple linear decision function

$$u_t = u(\mathbf{x}_t) = \sum_{j=1}^{\ell} w_j \cdot x_{tj} + b,$$

where w_i are weight coefficients and b is a bias term.

Remark 4.1 *In order to simplify the notations, we will use the same symbols for the secondry features, which were produced as an outcome of clustering. It is assumed that* $\ell \ll m$.

Let us consider the most basic quadratic minimization model [21] with the following target function:

$$L(\mathbf{w}) = \Omega(\mu, n, \mathbf{w}) + \sum_{t=1}^{n} (y_t - u_t)^2, \tag{1}$$

where $\Omega(\mu, n, \mathbf{w}) = \mu \cdot n \cdot \|\mathbf{w}\|^2$ is a regularization term with ridge parameter μ.

Remark 4.2 *The target of the regularization term with parameter μ is to reduce the difference between training and test results. Value of μ may be optimized using cross-validation as discussed in [15]. We used in our experiments $\mu = 0.01$.*

4.1 Gradient-based optimisation

The direction of the steepest decent is defined by the gradient vector

$$g(\mathbf{w}) = \{g_j(\mathbf{w}), j = 1, \ldots, \ell\},$$

where

$$g_j(\mathbf{w}) = \frac{\partial L(\mathbf{w})}{\partial w_j} = 2\mu \cdot n \cdot w_j - 2\sum_{t=1}^{n} x_{tj}\,(y_t - u_t)\,.$$

Initial values of the linear coefficients w_i and the bias parameter b may be arbitrary. Then, we recompute the coefficients

$$\mathbf{w}^{(i+1)} = \mathbf{w}^{(i)} + \delta_i \cdot g(\mathbf{w}^{(i)}),\ b^{(i+1)} = b^{(i)} + \frac{1}{n}\sum_{t=1}^{n}(y_t - u_t),$$

where i is a sequential number of iteration. Minimizing (1) we find the size of the step according to the formula

$$\delta = \frac{L_1 - L_2 - \mu \cdot n \sum_{j=1}^{\ell} w_j g_j}{\sum_{t=1}^{n} s_t^2 + \mu \cdot n \sum_{j=1}^{\ell} g_j^2}, \tag{2}$$

where

$$L_1 = \sum_{t=1}^{n} s_t y_t, \quad L_2 = \sum_{t=1}^{n} s_t u_t, \quad s_t = \sum_{j=1}^{\ell} x_{tj} g_j.$$

5 Support vector machines

In difference to the linear regression model, the target of the support vector machines is not to approximate, but to separate the patterns. As an output SVMs create a decision boundary separating the patterns. This boundary is based on the most relevant data-instances (the so-called support vectors).

Good performance of a pattern classifier is achieved when the number of adjustable parameters is matched to the size of the training set. Using above idea as a motivation and according to the Lagrangian method we can transform original classification problem: minimize $\|\mathbf{w}\|^2$, subject to $y_t \cdot u_t \geq \phi > 0$, into the dual space. The objective of the SVM model is to maximize

$$L(\mathbf{v}) = \sum_{t=1}^{n} v_t \left[\phi - \frac{y_t}{2}\sum_{j=1}^{n} v_j y_j K(\mathbf{x}_t, \mathbf{x}_j)\right] \tag{3}$$

subject to the following conditions

$$\sum_{t=1}^{n} v_t y_t = 0,\ v_t \geq 0,$$

where

$$K_{tj} = K(\mathbf{x}_t, \mathbf{x}_j) = \langle \mathbf{x}_t, \mathbf{x}_j \rangle = \sum_{v=1}^{\ell} x_{tv} x_{jv} \tag{4}$$

are elements of the kernel matrix. Performance of the SVM algorithm depends essentially on the regulation parameter ϕ. We used in our experiments value $\phi = 1$.

Remark 5.1 *Similar to Section 4.1, the target function (3) may be maximised using gradient based optimisation as it was discussed in [19].*

6 Data

Colon dataset[1] represents a matrix of 62 tissue samples (40 negative and 22 positive) and 2000 genes. The microarray matrix for this set thus has $m = 2000$ rows and $n = 62$ columns.

Leukaemia dataset[2] was originally provided in [8], which contains the expression levels of 7129 genes of 72 patients, among them, 47 patients suffer from acute lymphoblastic leukaemia (ALL) and 25 patients suffer from the acute myeloid leukaemia (AML).

We applied to the data double normalisation. First, we normalised each column to have means zero and unit standard deviation. Then, we applied the same normalisation to each row.

7 Experiments

7.1 Colon data

The performance of the Algoritm 1 depends essentially on the sequential order of the data. Accordingly, we used 100 random permutations of the indexes $\{1, ..., 2000\}$. In line with ensemble machine learning techniques [20], we can consider a classifier which is based on any single permutation as a base or weak learner. Ensembles are often capable of greater prediction accuracy than any of their individual members. As a consequence of the diversity between individual base-learners, an ensemble does not suffer from overfitting.

Based on the experimental trials, we used constant forward and backward threshold parameters: $H_F = 45, H_B = 10$. Note that Algorithms 1 and 2 were used with Manhatten and Euclidean distances. As an outcome, the Algorithm 1 produces matrices of centroids. The corresponding clustering sizes are given in the Fig. 1(c). Average number of clusters was 9.44 ranging from 5 to 14.

We used two evaluation criterions: 1) average misclassification rate (AMR) and 1) area under receiver operation curve (AUC), see Fig. 1(a).

By definition,

$$AMR = \frac{1}{n} \sum_{t=1}^{n} \delta(f_t, y_t),$$

where f_t is prediction of the label y_t,

$$\delta(f_t, y_t) = \begin{cases} 1 & \text{if } f_t \neq y_t; \\ 0, & \text{if } f_t = y_t. \end{cases}$$

We performed LOO (leave-one-out) cross-validation to explore the classification potential of our method. This means that we set aside the ith observation and fit the classifier by considering remaining $(n-1)$ data points. Taking into account small number of centroids, we used linear regression as a base-learner. We shall denote such a scheme as

$$RP\{100\}\left[Alg.1 + LOO\{RLR\}\right]. \tag{5}$$

The final decision function was computed as a sample average of base-learners. The corresponding AMR was 0.1612.

[1] http://microarray.princeton.edu/oncology/affydata/index.html

[2] http://www.broad.mit.edu/cgi-bin/cancer/publications/

Table 1: Some selected results (in terms of AMR), where the LOO Scheme 1 corresponds to (5) and Scheme 2 corresponds to (6). Columns "N" show number of the used clusters, and average number of the used clusters in the case of Algorithm 1.

LOO	Step 1	Step 2	Colon	N	Leukaemia	N
1	Alg. 2	SVM	0.1361	9	0.0139	7
2	Alg. 2	SVM	0.1524	9	0.0235	7
1	Alg. 2	RLR	0.1590	11	0.0174	15
1	Alg. 1	SVM	0.1612	9.44	0.0187	653

Fig. 2 represents an average of 100 independent procedures, each of which may be described as follows. Firstly, we used the regularised k-means Algorithm 2 in order to compress data to the selected number of clusters where initial allocation of genes to clusters was drawn at random. Components of the corresponding centroids were used as representatives of tissues. Then, we conducted evaluation using LOO method with linear SVM as a base learner. In line with (5), we can denote such scheme as $RP\{100\}\left[Alg.2 + LOO\{SVM\}\right]$. Fig. 2(a) demonstrates AMR as a decreasing function of number of clusters (black circles) in the case without regularisation. We conducted experiment with up to 100 clusters and observed AMR was above 0.24.

The structure of the graph was changed dramatically when we applied regularisation with $\alpha = 0.01$. Initially, we observed rapid decline of the AMR to the point $k = 15, AMR = 0.1558$. Then, AMR grows slowly to the level above 0.21 (overfitting may be regarded as the most likely explanation for such growth), and in the case if number of clusters was more than 60 AMR became stable. Further improvement was obtained with $\alpha = 0.1$, see Fig. 2(a) - blue stars. The lowest point corresponds to the number of clusters k=9 with AMR = 0.1361.

Remark 7.1 *The property represented by the Fig. 2 may be used for the selection of the number of clusters.*

Remark 7.2 *In the Fig. 1 - 2, we have plotted the AMRs estimated using LOO cross-validation under assumption that the choice of clusters will be the same during the n validation trials as chosen on the basis of the full data set. However, there will be a selection bias in these estimates as the clusters should be reformed as a natural part of any validation trial; see, for example, [2]. But, since the labels y_t of the training data were not used in the clustering process, the selection bias should not be of a practical importance.*

The validation scheme

$$RP\{100\}\left[LOO\{Alg.2 + SVM\}\right]. \tag{6}$$

requires a lot more computational time comparing with the model (5). Nevertheless, we tested the model (6) in application to the fixed number of clusters $k = 9$ with regulation parameter $\alpha = 0.1$ and observed AMR = 0.1524.

7.2 Leukaemia data

Based on the experimental trials, we used Algorithm 1 with constant forward and backward threshold parameters: $H_F = 52, H_B = 35$. Again (as in the case of colon), we conducted experiments against 100 random permutations. The average number of clusters was 867.4, ranging from 827 to 924. The mean AMR was 0.0561, with range from 0.0278 to 0.0833.

The figures in Fig. 2(b) were obtained using an identical procedure (as in the case of the colon data, see Fig. 2(a)) applied to the leukaemia data, and illustrates very similar structures. The best result AMR = 0.0253, see Fig. 2(b) - blue stars, corresponds to $\alpha = 0.1, k = 11$.

7.2.1 Additional preprocessing steps

We followed the preprocessing steps of [7]: (1) thresholding: floor of 1 and ceiling of 20000; (2) filtering: exclusion of genes with max / min ≤ 2 and (max - min) ≤ 100, where max and min refer respectively to the maximum and minimum expression levels of a particular gene across a tissue sample. This left us with 1896 genes. In addition, the natural logarithm of the expression levels was taken. Finally, we applied double normalisation as described in Section 6.

After above preprocessing we observed significant improvement in quality of classification. The corresponding results are presented in the Table 1, and are competitive comparing with previous publications [5], [12], where the best reported result for colon set is AMR = 0.113, and AMR = 0.0139 for leukaemia set.

Remark 7.3 *We conducted similar studies in application to the colon data. Firstly, we observed the following statistical characteristics:* $min = 5.82, max = 20903, 4.38 \leq max/min \leq 1258.6$. *Then, we took natural logarithm of the expression levels. Based on our experimental results we can not report any improvement in the quality of classification.*

7.3 Computation time

A Linux computer with speed 3.2GHz, RAM 16GB, was used for most of the computations. The time for the scheme (5) in the case of colon (left column of the Fig. 2) was about 5 hours. The only one case with 9 clusters, scheme (6), took about 4 hours. Similar computations for leukaemia data took about 15 hours (4 hours) for the scheme (5), and 9 hours (3 hours) for the scheme (6), where times for the reduced set with 1896 genes were given in brackets.

8 Concluding remarks

Microarray data analysis is challenging the traditional machine learning techniques due to the availability of a limited number of training instances and the existence of large number of genes, together with the inherent various uncertainties. In many cases machine learning techniques rely too much on the gene selection, which may cause selection bias. Generally, feature selection may be classified into two categories based on whether the criterion depends on the learning algorithm used to construct the prediction rule. If the criterion is independent of the prediction rule, the method is said to follow a filter approach, and if the criterion depends on the rule, the method is said to follow a wrapper approach [2].

The objective of this study is to develop a filtering machine learning approach and produce a robust classification for microarray data. The proposed regularized k-means algorithm represents a very important component of the classification system. The results that we obtained on two real datasets confirm the potential of our approach.

References

[1] H. Akaike. “On the likelihood of a time series model.” *The Statistician*, vol. 27, pp. 217-235, 1978.

[2] C. Ambroise and G. McLachlan. “Selection bias in gene extraction on the basis of microarray gene expression data.” *Proceedings of the National Academy of Sciences USA*, vol. 99, pp. 6562-6566, 2002.

[3] A. Antoniadis, S. Lambert-Lacroix and F. Leblanc. “Effective dimension reduction methods for tumor classification using gene expression data.” *Bioinformatics*, vol. 19, no. 5, pp. 563-570, 2003.

[4] N. Belacel, Q. Wang and M. Cuperlovic-Culf. “Clustering methods for microarray gene expression data.” *A Journal of Integrative Biology*, vol. 10, no. 4, pp. 507–531, 2006.

[5] M. Dettling and P. Buhlmann. “Boosting for tumor classification with gene expression data.” *Bioinformatics*, vol. 19, no. 9, pp. 1061-1069, 2003.

[6] S. Dudoit and J. Fridlyand. “A prediction-based resampling method for estimating the number of clusters in a dataset.” *Genome Biology*, vol. 3, no. 7, 2002.

[7] S. Dudoit, J. Fridlyand and T. Speed. “Comparison of discrimination methods for the classification of tumors using gene expression data.” *Journal of Americal Statistical Association*, vol. 97, no. 457, pp. 77-87, 2002.

[8] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh and J. Downing. “Molecular classification of cancer: class discovery and class prediction by gene expression monitoring.” *Science*, 286, pp. 531-537, 1999.

[9] I. Guyon, J. Weston, S. Barnhill and V. Vapnik. “Gene selection for cancer classification using support vector machines.” *Machine Learning*, vol. 46, pp. 389-422, 2002.

[10] G. Hamerly and C. Elkan. “Learning the k in k-means.” *16th Conference on Neural Information Processing Systems (NIPS)*, 2003.

[11] J. Hartigan. “Clustering Algorithms.” John Wiley and Sons, New York, 1975.

[12] C. Hennig. “Cluster-wise assessment of cluster stability.” *Computational Statistics and Data Analysis*, vol. 52, pp. 258-271, 2007.

[13] T. Lange, Y. Roth, M. Braun and J. Buhmann. “Stability-based validation of clustering solutions.” *Neural Computation*, vol. 16, pp. 1299-1323, 2004.

[14] Y. Liu, D. Hayes, A. Nobel and J. Marron. “Statistical significance of clustering for high-dimension, low-sample size data.” *Journal of Americal Statistical Association*, vol. 103, no. 483, pp. 1281-1293, 2008.

[15] C. De Mol, S. Mosci, M. Traskine and A. Verri. “A regularised method for selecting nested groups of relevant genes from microarray data.” *Journal of Computational Biology*, vol. 16, no. 5, pp. 677-690, 2009.

[16] G. McLachlan, R. Bean and D. Peel. “A mixture model-based approach to the clustering of microarray expression data.” *Bioinformatics*, vol. 18, no. 3, pp. 413-422, 2002.

[17] G. McLachlan and S.K. Ng. “The EM algorithm.” *In The Top-Ten Algorithms in Data Mining, X. Wu and V. Kumar (Eds.). Boca Raton, Florida: Chapman and Hall/CRC*, pp. 93-115, 2009.

[18] V. Nikulin. “Weighted threshold-based clustering for intrusion detection systems.” *International Journal of Computational Intelligence and Applications*, vol. 6, no. 1, pp. 1-19, 2006.

[19] V. Nikulin. “Learning with mean-variance filtering, SVM and gradient-based optimization.” *International Joint Conference on Neural Networks, Vancouver, IEEE*, pp. 4195-4202, 2006.

[20] Y. Peng. “A novel ensemble machine learning for robust microarray data classification.” *Computers in Biology and Medicine*, vol. 36, pp. 553-573, 2006.

[21] M. Segal, K. Dahlquist and B. Conklin. “Regression approaches for microarray data analysis.” *Journal of Computational Biology*, vol. 10, no. 6, pp. 961-980, 2003.

[22] G. Schwarz. “Estimating the dimension of a model.” *The Annals of Statistics*, vol. 6, no. 2, pp. 461-464, 1978.

[23] M. Smolkin and D. Ghosh. “Cluster stability scores for microarray data in cancer studies.” *BMC Bioinformatics*, vol. 4, no. 36, 2003.

[24] G. Tseng and W. Wong. “Tight clustering: a resampling-based approach for identifying stable and tight patterns in data.” *Biometrics*, vol. 61, pp. 10-16, 2005.

[25] X. Zhang, X. Lu, Q. Shi, X. Xu, H. Leung, L. Harris, J. Iglehart, A. Miron, J. Liu and W. Wong. “Recursive SVM feature selection and sample classification for mass-spectrometry and microarray data.” *BMC Bioinformatics*, vol. 7, no. 197, 2006.

[26] S. Zhong and J. Ghosh. “A unified framework for model-based clustering.” *Journal of Machine Learning Research*, vol. 4, pp. 1001-1037, 2003.