ГОДИШНИК НА СОФИЙСКИЯ УНИВЕРСИТЕТ „СВ. КЛИМЕНТ ОХРИДСКИ“

ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА
Том 104

ANNUAL OF SOFIA UNIVERSITY „ST. KLIMENT OHRIDSKI“

FACULTY OF MATHEMATICS AND INFORMATICS
Volume 104

SPECTRAL CLUSTERING OF MULTIDIMENSIONAL GENETIC DATA

TSVETELIN ZAEVSKI, OGNYAN KOUNCHEV, DEAN PALEJEV, EUGENIA STOIMENOVA

The main purpose of the present paper is to initiate the application of methods from Spectral graph theory to the analysis of multidimensional genetic data, and in particular to the problem of detecting differential expression based on RNA-Seq data. Here we introduce a new algorithm, that is based on the method of Spectral Clustering and integrates an additional information about a priori given relations among the genes.

Keywords: spectral clustering, multidimensional data analysis, RNA-Seq data analysis.

2010 Math. Subject Classification: 62-07, 62H30, 91C20, 92D20.

1. INTRODUCTION

In recent years large amounts of DNA-Seq and RNA-Seq data were produced as a consequence of the advancements of the high-throughput sequencing technologies. One of the most interesting questions that can be answered by analyzing RNA-Seq data is finding differentially expressed genes or transcripts, for which the overall levels in one group of subjects (e.g. patients with a particular disease) is significantly different than the overall levels in another group (e.g. healthy controls). Ever since the first RNA-Seq datasets became available, researchers started developing different methods for analyzing it in order to find differentially expressed

genes and nowadays there are dozens such methods. Further in this article we will show that even for the same dataset, some of these methods produce very different results than the others.

Such differences of the results naturally raise the questions whether some methods are better than others, and more generally how to compare and evaluate such methods.

Comparing the results of differentially expressed genes is difficult because typically researchers are only able to biologically validate some portion of the genes being determined as differentially expressed by the method. In addition, very few or even none of the ones not being determined as differentially expressed are validated as such. There are issues even in cases in which generated or in-silico data is used and therefore we know the true differentially expressed genes, e.g. Soneson and Delorenzi (2013). In these cases the data generation assumes certain distributions, e.g. NB or Poisson, or includes artificially added outliers, which gives and advantage to differential expression methods that assume the respective distributions.

Here we discuss some general methods, in particular Spectral Clustering, for calibrating binary classifications that can also be used to compare and evaluate such classifications. Starting with an initial guess for the clusters (a split of number of points into two groups, i.e. initial classification) and an a priori information about the correlations, the method "moves" some of them between the clusters in order to improve the classification, in a sense that the resulting (calibrated) classification is closer to the "true" classification.

The research is structured in the following way: in section 2 we give a brief review of the spectral clustering algorithm, in section 3 we explain the used methodology and the corresponding results, and finally in section 4 in a short Appendix, we provide a curious relation between the method of Spectral Clustering and kernel PCA.

More details of the present research and the experimental results will be provided in subsequent publication.

2. INTRODUCTION TO THE SPECTRAL CLUSTERING

In the present introduction we provide a short description of the method of Spectral Clustering (SC) and provide some useful references related to recent developments and applications of the method.

The search for clusters is called traditionally clustering, but more recently synonyms were introduced as community detection or modularity maximization, cf. Clauset et al. (2004), Newman (2006), Newman (2008), and Fortunato and Barthelemy (2007). It is one of the main problems in Data Analysis, when studying data which are identified as points not only in a Euclidean space but also in an abstract graph where the weights of the edges may be used to generate a similarity

matrix. The role of the similarity matrix is to reflect the neighborhood relations between data points.

Unlike the usual methods for data clustering and graph partitions, as e.g. $k-$means, the method of Spectral Clustering is based on a completely different view on partition of graphs. In principle, SC may be applied to graphs where one has a naturally defined similarity matrix; in particular, if the data may be embedded into an Euclidean space, then we may use various approaches to defining a similarity matrix. Hence, we may apply the SC to very abstract situations.

Whereas the standard approach to clustering, as the method of $k-$means emphasizes upon the "compactness" of the data points, the SC makes the point on the "connectivity" or the "modularity" of the data points. The method of SC may be considered as a method for partitioning of graphs. Assume that the vertices of an undirected graph are enumerated as $x_j \in V$ (the set of vertices) and the *similarity* between them is defined by a weight matrix

$$W = (w_{ij})_{i,j}$$

with coefficients

$$w_{ij} := \omega(x_i, x_j) \geq 0$$

where the function ω regulates the size of the neigbourhoods. Then the set of edges is defined as those couples $E_{ij} := (x_i, x_j) \in E$ for which $w_{ij} > 0$. The main idea of the graph partitioning is to subdivide it into groups of vertices, so that edges E_{ij} for which x_i and x_j belong to the same group have large weights w_{ij}, while edges E_{ij} with x_i and x_j in different groups have small weight w_{ij}.

The simplest example would be if we consider a graph consisting of points $x_j \in \mathbb{R}^n$. One may take a weight function of the form

$$w_{ij} := g(x_i - x_j)$$

in particular, the Gaussian one

$$w_{ij} := \exp\left(\frac{\|x_i - x_j\|^2}{2\sigma^2}\right).$$

A standard method for clustering is the Min-Cut: For every two sets A and B we define the "strength of interaction" as

$$\operatorname{cut}(A, B) := \sum_{i \in A, j \in B} w_{ij}$$

Now the intuitive idea of the method of Min-Cut is based on minimizing the weight of edges connecting vertices in A to vertices in B. This very intuitive algorithm takes $O(|V|\,|E|)$ time for the calculations, where $|V|$ denotes the set of elements in

the set V. However, it is not a very successful algorithm as it often isolates vertices. It is essentially improved by the method of Normalized-Cut defined as

$$\mathrm{Ncut}\,(A,B) := \mathrm{cut}\,(A,B)\left(\frac{1}{\|A\|}+\frac{1}{\|B\|}\right)$$

where for every subset A in the graph we have put

$$\|A\| := \sum_{i\in A} d_i$$

and d_i is the degree of the vertex i. The method of Normalized Cut is based on the minimization of $\mathrm{Ncut}\,(A,B)$, i.e. the weights of edges connecting vertices in A to vertices in B, while keeping the sizes of A and B very similar. However it is NP-hard to solve.

An interesting approach to understanding the idea of the SC method is by first introducing the Normalized Cut. A main observation is that if we are given two sets A and B and define now the vector $f = (f_j)_j$ by putting

$$f_j := \begin{cases} \frac{1}{\|A\|} & \text{for } j \in A \\ \frac{-1}{\|B\|} & \text{for } j \in B \end{cases}$$

then we have

$$f^T L f = \sum_{i,j} w_{ij}\,(f_i - f_j)^2 = \sum_{i,j} w_{ij}\left(\frac{1}{\|A\|}+\frac{1}{\|B\|}\right)^2$$

and

$$f^T D f = \sum_{i,j} d_i f_i^2 = \frac{1}{\|A\|}+\frac{1}{\|B\|}$$

Here we see that the important notions appear in a natural way: the diagonal matrix D has its diagonal given by the vector $(d_j)_j$ and L is the *unnormalized Laplacian* matrix defined by

$$L := D - W.$$

We see easily that

$$\mathrm{Ncut}\,(A,B) = \frac{f^T L f}{f^T D f}$$

hence

$$\min_{A,B} \mathrm{Ncut}\,(A,B) = \min_{A,B} \frac{f^T L f}{f^T D f}$$

where the minimum is taken over the sets A, B. Obviously, we may apply a relaxation by considering only those f for which $f^T D 1 = 0$. Hence, we obtain the solutions to the above problems as a solution to the generalized eigenvalue problem

$$Lf = \lambda D f.$$

For details, we refer to Chung (1997) and von Luxburg (2007).

The spectral properties of the Laplacian are closely related to the topological properties of the graph, as the following classical result shows.

Proposition 1. *The matrix L is symmetric and positive semi-definite; as such it has n non-negative, real eigenvalues, and the smallest one satisfies $\lambda_1 = 0$; the corresponding eigenvector has all elements equal to 1. If G is an undirected graph with nonnegative weights $w_{ij} \geq 0$, then the multiplicity k of the eigenvalue λ_1 is equal to the number of connected components of G.*

We see that the eigenvalue λ_1 gives the basic information about the clustering of the graph into disconnected components, hence, it is very natural to ask for a deeper knowledge of the cluster structure by inspecting the next eigenvalues. Thus, these thought follow naturally the historical steps undertaken in 1973 in the paper of Donath and Hoffman (1973) and the paper of Fiedler (1973), who considered the second eigenvalue.

2.1. SPECTRAL CLUSTERING ALGORITHM

The general scheme of the SC algorithm is given by the following steps (see e.g. von Luxburg (2007)):

1. Let $W \in R^{n\times n}$ be the similarity matrix with elements w_{ij}. Let also put $d_i = \sum_{j=1,...,n} w_{ij}$ and define D as the diagonal matrix having diagonal elements d_i. Let us assume that the number of clusters is k.

   Compute the Laplacian matrix by putting $L = D - W$.

2. Compute the first k eigenvectors $u_1, ..., u_k$ of L.

   Let $U \in R^{n\times k}$ be the matrix constructed from the vectors $u_1, ..., u_k$ as columns.

3. Let $y_i \in R^k$, for $i = 1, ..., n$, be the corresponding i-th row of U.

   Cluster the vectors $y_i, i = 1, ..., n$, in k clusters, $C_1, C_2, ..., C_k$, using the k-means algorithm.

4. The clusters, $A_1, A_2, ..., A_k$, of the initial data are recomputed by $A_i = \{j, y_j \in C_i\}$.

The success of the SC method is usually illustrated by a relatively simple toy example, with data points located on the real axis, cf. von Luxburg (2007), p. 399. This toy data set consists of a random sample of 200 points $x_1, ..., x_{200} \in \mathbb{R}$ drawn according to a mixture of four Gaussians. Since the main applications which we intend are in the area of genetic analysis using gene expressions which are at

least two-dimensional, we will be more interested in demonstrating the power of the SC method for simulated two-dimensional data. In Figure 1 below we have examples of two-dimensional graphs which are generated in a way very analogous to the one-dimensional. In Figure 1a we have generated a random sample of 20000 points in the plane drawn according to a mixture of four two-dimensional Gaussians which are located on four ellipses (5000 points on each one). The semiaxes of the ellipses are $(1, 1)$, $(2, 3)$, $(5, 4)$, and $(30, 5)$, respectively. In Figure 1b these are seven mixtures with total of 35000 points located in seven ellipses (again 5000 points on each one). In addition to the ellipses in Figure 1a we generate three other with semiaxes $(2, 3)$, $(7, 2)$, and $(6, 1)$, respectively. After the deterministic generation of each point, we move it on random distance in every axes (normally distributed with parameters $(0, 0.1)$).

In fact, in Figure 1 one sees the result of the application of SC - it provides a perfect clustering.

3. ENHANCEMENT OF INITIAL CLUSTERING BY INCORPORATING A PRIORI INFORMATION

The purpose of this section is to introduce our methodology for enhancing an already available clustering, which is based on the appropriate usage of additional information in the form of a priori given correlations between the elements. We present the performance of our method on simulated data.

3.1. ALGORITHM

Suppose we are given data which is already clustered using some method. For simplicity we shall use two clusters, the sets I and NI with significantly different sizes – the smaller one I will be considered to be containing the significant elements (important genes), and for this reason will be called "important set", while NI will be bigger and will contain the not important elements. If we have in addition some information for the relations between the elements of the graph, especially in the form of correlations, we will use it to improve the initial clustering. Our aim is (1) to incorporate in a proper way the a priori correlation information by means of defining an appropriate similarity matrix (2) to keep as many elements as possible in the set I and (3) to move to the set I those elements in NI which have a high value of similarity w.r.t. any element in I.

In Statistical data analysis, the correlation matrix is an important statistical technique which measures the relation between two variables. For our methodology we develop a model for which we need to know (1) which elements are important, i.e. the set I, and (2) a "good enough" correlation matrix, which will be used for creating a similarity matrix. The proposed SC algorithm will enhance every initially given clustering defined by initial sets I and NI in two respects: First, it

will add some new elements to the set I, which have a high correlation with the elements in I. Second, it will remove from the set I some elements which were thought initially to be important, because of their large correlation with the set NI.

For simplicity sake we will explain our methodology on an example which appears in the analysis of expression levels of genes for RNA-Seq data. The most important problem when analyzing RNA-Seq data, is to find differentially expressed genes or transcripts, for which the overall levels in one group of subjects (e.g. patients with a particular disease) is significantly different than the overall levels in another group (e.g. healthy controls). On the other hand, and important ingredient of this difficult problem is a matrix with historically available correlations between the genes which is however not positive-definite. It is important to find an appropriate way to incorporate this a priori information in the algorithm. In the present example, we assume that the number of expressed genes is 8824.[1]

Let the number of all subjects studied be n and n_1 be the disease patients, while $n_2 = n - n_1$ be the number of the healthy controls. Thus the graph G we have to study is the subset of all 8824 points in the euclidean space $\mathbb{R}^n$. To simplify this setting, we calculate the average of the expression levels for each of the 8824 genes for the n_1 subjects, and on the other hand, calculate the average of the expression levels for each of the 8824 genes for the n_2 subjects. Thus we obtain 8824 points in the real plane $\mathbb{R}^2$ which reduces the problem to a clustering problem in the plane. Although this situation seems to be too simplified, it remains very non-trivial. It still makes deep sense to identify which are the important genes since the intuitive expectation is that the averaged gene expression levels for the disease patients would be in principle different from the averaged gene expression levels for the healthy controls. Such identification of the important genes in the plane would be very helpful to solve the genuine clustering problem in $\mathbb{R}^n$.

Our algorithm runs as follows:

1. *Generation of simulated clustering*

   First, we generate a simulated clustering given by a partition of the graph G given by $G = I \cup NI$. We generate the set I_{500} by selecting randomly 500 (respectively, the set I_{1000} with 1000) points in the plane $\mathbb{R}^2$ – normally distributed with expectation one and standard deviation 0.1. This will be defined as the important set I, and it is visualized in the top right corner in Figure 2. We generate in a similar way the set of not important elements NI (NI_{500} with 8324 points, and respectively NI_{1000} with 7824 points) – the center of their normal distribution is -1. This set is placed in the bottom left corner in Figure 2.

[1]Here the number 8824 is not accidentally chosen, but is the number of genes with average expressions at least 8 in the widely-used dataset by Bottomly et al. (2011).

2. *Correlation matrix*

   We generate a correlation matrix which would mimic the historically available correlations between the genes. We generate a correlation matrix C by using an algorithm described in Numpacharoen and Atsawarungruangkit (2012), modified by an introduction of a beta distribution. We provide two experimental settings by generating two correlation matrices, C_1 and C_2:

   (a) The matrix C_1 is generated by using a beta distribution $Beta(2, 5)$ with parameters 2 and 5

   (b) The matrix C_2 is generated in a similar way by the beta distribution $Beta(2, 2)$.

   The main difference between them is that C_1 has a relatively low large values.

3. *Similarity matrix*

   (a) Let us note that there are some elements in the cluster NI, which have a very low correlations with the others (less than 0.03). The spectral clustering algorithm can not decide correctly if such element is important or not. For this reason, we state that such elements are closer to the elements in NI, by assigning higher correlation levels.

   (b) We will modify the correlation matrix C by introducing the so-called level of significance l. The meaning of this parameter is to increase the role of the correlations which are higher than l. Here we use a power function of the form

$$f(x) := \begin{cases} x^p & \text{for } x < l \\ x^{1/p} & \text{for } x \geq l \end{cases}$$

   for an appropriate integer number p. One may use also different functions f which have similar "amplification behavior". We replace the matrix C with elements $c_{i,j}$ by the matrix C' with elements $c'_{i,j} = f(c_{i,j})$. We carry out experiments with different significance levels l. Since the maximal correlation of the first correlation matrix C_1 is 0.8807, it makes sense to make experiments with three different values $l = \{0.6, 0.7, 0.8\}$. For the same reason, for the second matrix C_2 we make experiments with five values $l = \{0.6, 0.7, 0.8, 0.9, 0.98\}$.

   (c) The core of our algorithm is the definition of a proper similarity matrix which takes into account the correlation matrix C. We define the similarity matrix W by putting:

      i.

$$w_{i,j} := \exp\left[-\frac{d(x_i, x_j)^2}{2\sigma^2}\right]$$

for the elements x_i, x_j in I; this is the Gaussian similarity coefficient which preserves the geometrical closeness of the elements, as here $d(x_i, x_j)$ denotes the euclidean distance.

ii. for taking into account the a priori given correlations C' we put

$$w_{i,j} := c'_{i,j}$$

for the rest of the pairs (x_i, x_j).

3.2. RESULTS

The results for the model with clustering sets I_{500} and NI_{500} are presented in Figure 2 and Tables 1, 2. The Figure representing the model with clustering sets I_{1000} and NI_{1000} looks similar, and we do not provide it here. The set of important elements after clustering are colored in red. The new important elements are the red points in the bottom left corner. As we can expect, their number varies for different levels of significance l – these elements are more for smaller values. This can be easily viewed in Figure 2. The initially accepted for important elements in I, which after clustering are changed to not important, can not be seen clearly in Figure 2 since they are only few, however one can observe their number in the fourth column of Tables 1 and 2. These tables contain the following values:

1. The first column contains the values of the parameters – in the brackets are the parameters of the beta distribution used for the generation of the correlation matrix; the other parameter is the level of significance l.

2. The second column contains the number of the expected important elements before the clustering – respectively 500 and 1000.

3. The third column contains the number of those expected important elements which are important again after the clustering.

4. The fourth column contains the number of those expected important elements which are NOT important after the clustering (column 2 - column 3).

5. The fifth column contains the number of the expected not important elements before the clustering –respectively 8324 and 7824.

6. The sixth column contains the number of those expected not important elements which have moved to the important set after the clustering.

7. The last column contains the number of those expected not important elements which are again not important after the clustering. (column 5 - column 6).

Also, it is reasonable to expect that the total number of the important elements after clustering varies for different levels of significance – they are more for lower values of l. Table 1 shows that for beta distribution $Beta(2,2)$ they vary, respectively in the following ranges:

1. between 499 and 3434, for the model with clustering given by the sets I_{500} and NI_{500} ,
2. between 1000 and 3753, for the model with clustering given by the sets I_{1000} and NI_{1000}.

The same observation is true when the correlation matrix C is generated using a beta distribution $Beta(2,5)$ – we can see in the Table 2 that the number of important elements varies in the following ranges:

1. between 515 and 844, for the model with clustering given by the sets I_{500} and NI_{500} ,
2. between 1010 and 1322, for the model with clustering given by the sets I_{1000} and NI_{1000}.

We can see immediately that the number of the important elements when we use beta distribution $Beta(2,5)$ are significantly less than the corresponding number in the model with beta distribution $Beta(2,2)$. This is true because the high levels in the $(2,2)$-correlation matrix are significantly more than those in the $(2,5)$-matrix.

We will only briefly explain the idea of our algorithm. Let us assume that we have after clustering a set of important elements I. On the other hand, let $I_1 \subset I$ be the set of those important elements, which before we perform clustering are not expected to be important. And finally, let NI be the set of not important elements after clustering. Now, the logic of our algorithm becomes clear from the following inequalities:

$$\min_i \left\{ \max_j \{|C(m_i, m_{1,j})|\} \right\} > l, \qquad m_i \in I, m_{1,j} \in I_1 \tag{1}$$

$$\max_i \left\{ \max_j \{|C(m_i, n_j)|\} \right\} < l, \qquad m_i \in I, n_j \in NI \tag{2}$$

where l is the level of significance and C is the corresponding correlation matrix introduced above. This means, that

1. For every important element, for which we initially thought that it is not important, there exists at least one important element such that the correlation between them is larger than the level of significance l.
2. For every not important element, there is no important one such that the correlation between them is larger than the level of significance l.

4. APPENDIX ON SPECTRAL CLUSTERING AND KERNEL PCA

For a reader more used to the traditional methods for dimensionality reduction in data analysis, we provide below a short comment about the relation between the method of Spectral Clustering and the so-called kernel Principal Component Analysis (PCA). This has been observed apparently for the first time by Bengio et al. (2003), where the authors show how both methods are special cases of a more general learning problem, that of learning the principal eigenfunctions of a kernel. An essential role is played by the fact that the smallest eigenvectors of graph Laplacians can also be interpreted as the largest eigenvectors of kernel matrices.

Before defining *kernel PCA*, let us remind that PCA is a basis transformation to diagonalize an estimate of the covariance matrix of the data. Given N points in d dimensions PCA essentially projects the data points onto p, directions ($p < d$) which capture the maximum variance of the data. These directions correspond to the eigenvectors of the covariance matrix of the training data points. Intuitively PCA fits an ellipsoid in d dimensions and uses the projections of the data points on the first p major axes of the ellipsoid. The "classic" PCA approach is a linear projection technique that works well if the data is linearly separable. However, in the case of linearly inseparable data, a nonlinear technique is required if the task is to reduce the dimensionality of a dataset. An here we come to the *Kernel PCA*. It is another unsupervised learning method that was proposed earlier and that is based on the simple idea of performing PCA in the feature space of a kernel by Schoelkopf, Smola and Muller in 1998. Schölkopf et al. (1997) propose the use of integral operator kernel functions, for computing principal components in high dimensional feature spaces, related to input space by some nonlinear map.

The basic idea of kernel PCA to deal with linearly inseparable data is to project it onto a (much) higher dimensional space where it becomes linearly separable. Let ϕ be this nonlinear mapping function so that a sample x can be mapped as $x \to \phi(x)$. The term "kernel" represents a function that calculates the dot product of the images of the samples x under ϕ, namely,

$$\kappa(x_i, x_j) = \phi(x_i)\phi(x_j)^T.$$

In other words, the function ϕ maps the original d-dimensional features into a larger, k-dimensional feature space by creating nonlinear combinations of the original features. Often, the mathematical definition of the Gaussian basis kernel function is written and implemented as

$$\kappa(x_i, x_j) = \exp(-\gamma||xi - xj||^2)$$

where $\gamma = 1/2\sigma^2$ is a free parameter that is to be optimized.

5. REFERENCES

Yoshua Bengio, Pascal Vincent, Jean-Franois Paiement, Olivier Delalleau, Marie Ouimet, and Nicolas Le Roux. Spectral clustering and kernel pca are learning eigenfunctions. Technical report, CIRANO, 2003.

Daniel Bottomly, Nicole A. Walter, Jessica Ezzell E. Hunter, Priscila Darakjian, Sunita Kawane, Kari J. Buck, Robert P. Searles, Michael Mooney, Shannon K. McWeeney, and Robert Hitzemann. Evaluating gene expression in C57BL/6J and DBA/2J mouse striatum using RNA-Seq and microarrays. *PLOS One*, 6(3):e17820+, March 2011. ISSN 1932-6203.

F.R.K. Chung. *Spectral Graph Theory*. Number 92 in CBMS Regional Conference Series. Conference Board of the Mathematical Sciences, 1997. ISBN 9780821889367. URL https://books.google.bg/books?id=YUc38_MCuhAC.

Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70:066111, Dec 2004. doi: 10.1103/PhysRevE.70.066111. URL https://link.aps.org/doi/10.1103/PhysRevE.70.066111.

W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM J. Res. Dev.*, 17(5):420–425, September 1973. ISSN 0018-8646. doi: 10.1147/rd.175.0420. URL http://dx.doi.org/10.1147/rd.175.0420.

Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2):298–305, 1973. URL http://eudml.org/doc/12723.

Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006. doi: 10.1073/pnas.0601602103. URL http://www.pnas.org/content/103/23/8577.abstract.

M.E.J. Newman. Mathematics of networks. In Steven N. Durlauf and Lawrence E. Blume, editors, *The New Palgrave Dictionary of Economics*. Palgrave Macmillan, Basingstoke, 2008.

K. Numpacharoen and A. Atsawarungruangkit. Generating correlation matrices based on the boundaries of their coefficients. *PLOS ONE*, 7(11):1–7, 11 2012. doi: 10.1371/journal.pone.0048902. URL https://doi.org/10.1371/journal.pone.0048902.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. *Kernel principal component analysis*, pages 583–588. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997. URL https://doi.org/10.1007/BFb0020217.

Charlotte Soneson and Mauro Delorenzi. A comparison of methods for differential expression analysis of rna-seq data. *BMC Bioinformatics*, 14(1):91, Mar 2013. ISSN 1471-2105. doi: 10.1186/1471-2105-14-91. URL https://doi.org/10.1186/1471-2105-14-91.

Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4): 395–416, Dec 2007. doi: 10.1007/s11222-007-9033-z. URL https://doi.org/10.1007/s11222-007-9033-z.

ACKNOWLEDGEMENT. The second and third of the authors were supported by Project I02/19, while the first- and the fourth-named authors were supported by project DH02-13 with Bulgarian NSF.

A. TABLES AND FIGURES

Table 1: Clustering results for data with 500 initially important elements

parameters	Expected important			Expected not important		
	Total	Imp.	Not imp.	Total	Imp.	Not imp.
(2,2) 0.6	500	500	0	8324	2934	5390
(2,2) 0.7	500	490	10	8324	1778	6546
(2,2) 0.8	500	499	1	8324	850	7474
(2,2) 0.9	500	487	13	8324	239	8085
(2,2) 0.98	500	489	11	8324	10	8314
(2,5) 0.6	500	500	0	8324	344	7980
(2,5) 0.7	500	499	1	8324	80	8244
(2,5) 0.8	500	500	0	8324	15	8309

Table 2: Clustering results for data with 1000 initially important elements

parameters	Expected important			Expected not important		
	Total	Imp.	Not imp.	Total	Imp.	Not imp.
(2,2) 0.6	1000	999	1	7824	2754	5070
(2,2) 0.7	1000	996	4	7824	1669	6155
(2,2) 0.8	1000	999	1	7824	801	7023
(2,2) 0.9	1000	990	10	7824	255	7599
(2,2) 0.98	1000	991	9	7824	9	7815
(2,5) 0.6	1000	1000	0	7824	322	7502
(2,5) 0.7	1000	996	4	7824	74	7750
(2,5) 0.8	1000	997	3	7824	13	7811

Figure 1: SC succeeds to separate all ellipses in the Gaussian mix example

(a) (b)

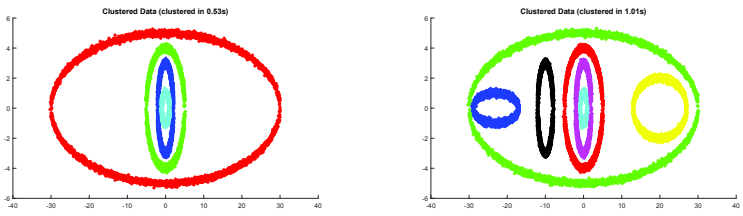
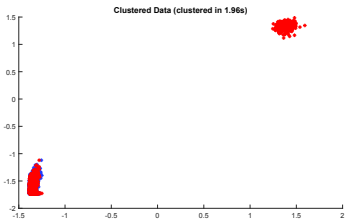
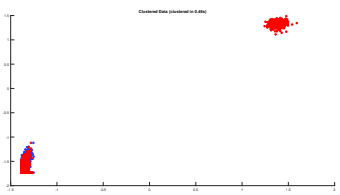


Figure 2: Clustering with 500 initial important elements

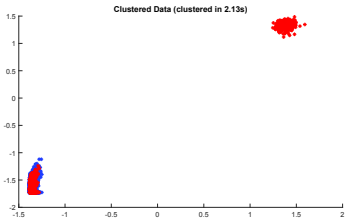
(a) Parameters: *Beta(2,2), l = 0.6*



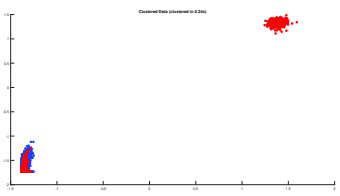
(b) Parameters: *Beta(2,2), l = 0.7*



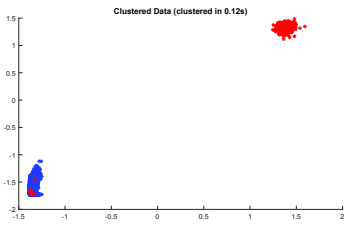
(c) Parameters: *Beta(2,2), l = 0.8*



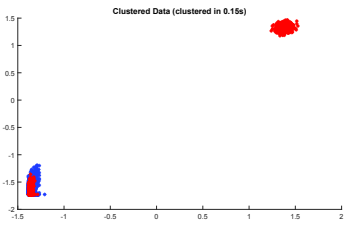
(d) Parameters: *Beta(2,2), l = 0.9*



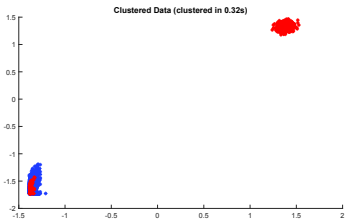
(e) Parameters: *Beta(2,2), l = 0.98*



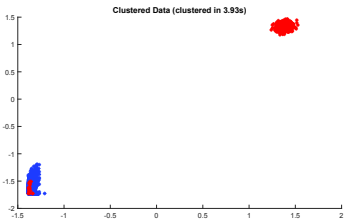
(f) Parameters: *Beta(2,5), l = 0.6*



(g) Parameters: *Beta(2,5), l = 0.7*



(h) Parameters: *Beta(2,5), l = 0.8*



Received on November 2, 2017

Tsvetelin Zaevski, Ognyan Kounchev, Dean Palejev, Evgenia Stoimenova

Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
Acad. G. Bonchev st., bl. 8, BG-1113 Sofia
BULGARIA

E-mails: t_s_zaevski@math.bas.bg
okounchev@gmail.com
palejev@math.bas.bg
jenistoimenova@gmail.com