

# Sparse spectral clustering method based on the incomplete Cholesky decomposition

*Katrijn Frederix      Marc Van Barel*

*Report TW552, November 2009*



Katholieke Universiteit Leuven  
Department of Computer Science  
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

# Sparse spectral clustering method based on the incomplete Cholesky decomposition

*Katrijn Frederix      Marc Van Barel*

*Report TW 552, November 2009*

Department of Computer Science, K.U.Leuven

## **Abstract**

A new sparse spectral clustering method using linear algebra techniques is proposed. This method exploits the structure of the Laplacian to construct its approximation, not in terms of a low rank approximation but in terms of capturing the structure of the matrix. The approximation is based on the incomplete Cholesky decomposition with an adapted stopping criterion, it selects a sparse data set which is a good representation of the full data set. With this approximation the eigenvalue problem can be reduced to a smaller problem. To obtain the indicator vectors from the eigenvectors the method proposed by [ Zha et al., Spectral relaxation for  $k$ -means clustering ] is adapted, which computes a pivoted  $LQ$  factorization of the eigenvector matrix. This formulation gives also the possibility to extend the method to out-of-sample points.

**Keywords :** spectral clustering, incomplete Cholesky factorization.

**MSC :** Primary : 62H30, Secondary : 65F15, 68T10, 91C20.

# Sparse spectral clustering method based on the incomplete Cholesky decomposition.

Katrijn Frederix\*, Marc Van Barel\*

November 17, 2009

## Abstract

A new sparse spectral clustering method using linear algebra techniques is proposed. This method exploits the structure of the Laplacian to construct its approximation, not in terms of a low rank approximation but in terms of capturing the structure of the matrix. The approximation is based on the incomplete Cholesky decomposition with an adapted stopping criterion, it selects a sparse data set which is a good representation of the full data set. With this approximation the eigenvalue problem can be reduced to a smaller problem. To obtain the indicator vectors from the eigenvectors the method proposed by [24] is adapted, which computes a pivoted  $LQ$  factorization of the eigenvector matrix. This formulation gives also the possibility to extend the method to out-of-sample points.

## 1 Introduction

Clustering is a widely used technique for partitioning unlabeled data into natural groups, which is a significant problem occurring in applications ranging from computer science, biology to social science or psychology. When clustering is carried out, data points which are related to each other are grouped together but points which are not related to each other are assigned to another group.

Already a wide range of methods exist to cluster unseen data, e.g.  $k$ -means [6, 7, 10], hierarchical clustering [15, 10, 22]. The focus in this paper is on spectral clustering [20, 9, 19, 23] which became in recent years a popular clustering algorithm. In fact, spectral clustering is a relaxation of a graph partitioning problem that is NP-hard and leads to an eigenvalue problem. Compared with other algorithms spectral clustering has the advantage that it is simple to implement and it solves the problem efficiently by standard linear algebra techniques.

A spectral clustering algorithm consists of the following steps. First, the graph Laplacian [9, 17, 18, 23] and the related eigenvalue problem are constructed. Then the  $k$  eigenvectors belonging to the smallest eigenvalues are computed. With these eigenvectors the cluster assignment can be achieved and each data point can be assigned to a cluster. Note that the size of the eigenvalue problem corresponds to the number of data points. This is prohibitive when working with large data sets.

Typically spectral clustering methods were only performed on training data without extensions to new points, apart from recomputing the eigenvectors of a larger system which is not computationally attractive. Recently, two methods are derived for out-of-sample points, one is based on the Nyström method [8] and the other is derived in a weighted kernel PCA framework [1]. These methods make it possible to assign new data points to clusters in an efficient way.

---

\*Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Leuven (Heverlee), Belgium. email: {Katrijn.Frederix,Marc.VanBarel}@cs.kuleuven.be. The research was partially supported by the Research Council K.U.Leuven, project OT/05/40 (Large rank structured matrix computations), CoE EF/05/006 Optimization in Engineering (OPTEC), by the Fund for Scientific Research–Flanders (Belgium), G.0423.05 (RAM: Rational modelling: optimal conditioning and stable algorithms), and by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office, Belgian Network DYSCO (Dynamical Systems, Control, and Optimization). The scientific responsibility rests with its authors.

In this paper, a sparse spectral clustering method is presented, with a possibility for extension to out-of-sample points, based on simple linear algebra techniques. In the first part of the clustering algorithm, the approximation of the graph Laplacian by the incomplete Cholesky decomposition is achieved. This decomposition can lead to a small numerical error if the eigenvalues of the similarity matrix decay rapidly. However, there exist examples for which the eigenvalues of the similarity matrix do not possess this property. In fact, the incomplete Cholesky decomposition selects a limited number of columns and rows of the similarity matrix such that the corresponding sparse data set is a good representation of the full data set. This has nothing to do with a fast decay of eigenvalues but with capturing the structure of the matrix. The stopping criterion of the incomplete Cholesky decomposition will be adjusted such that the approximation captures the structure of the matrix.

With this approximation, the eigenvalue problem is reduced to a smaller eigenvalue problem only based on the information received from the selected sparse data set. In the third part of the clustering algorithm, the obtainment of the indicator vectors, the cluster assignment for each data point can be found by computing a pivoted  $LQ$  decomposition of the eigenvector matrix [24]. This method is adapted such that also the importance of the selected data points is taken into account.

It is also possible to extend the proposed method to out-of-sample (test) points. We will show that this can be achieved by exploiting the information related to the pivoted  $LQ$  decomposition in the cluster assignment.

The paper is organized as follows. Section 2 contains an introduction to spectral clustering. Section 3 proposes the sparse spectral clustering algorithm based on linear algebra techniques. Section 4 gives the numerical results and Section 5 states a conclusion.

## 2 Introduction on spectral clustering

Because of the overwhelming amount of literature on the subject of spectral clustering, we will explain only the main concepts and refer the interested reader to [20, 9, 19, 23] for more information.

As stated in the introduction, spectral clustering is a relaxation of a graph partitioning problem that is NP-hard. Therefore we start with introducing a graph. Represent the data points  $\{\mathbf{x}_i\}_{i=1}^N$  as vertices in an undirected graph and assign a positive weight  $w_{ij}$ , based on a similarity measure, to the edges between  $\mathbf{x}_i$  and  $\mathbf{x}_j$ . From this, a symmetric similarity matrix  $W$  can be constructed,  $W_{ij} = w_{ij}$ . The degree of a vertex, which represents the total number of related weights to a specific node, is defined as  $d_i = \sum_{j=1}^N w_{ij}$  and the related degree matrix as  $D = \text{diag}(d_1, \dots, d_N)$ .

The idea of graph clustering, is to find  $k$  subgraphs such that a minimal number of edges are cut off and that the total weights of these edges are minimal. This is called the mincut problem [21] and it results in minimizing:

$$\text{Cut}(\mathcal{A}_1, \dots, \mathcal{A}_k) := \frac{1}{2} \sum_{i=1}^k W(\mathcal{A}_i, \bar{\mathcal{A}}_i),$$

with  $W(\mathcal{A}, \bar{\mathcal{A}}) := \sum_{i \in \mathcal{A}, j \in \bar{\mathcal{A}}} W_{ij}$  and where  $\bar{\mathcal{A}}$  stands for the complement of  $\mathcal{A}$ . A factor  $\frac{1}{2}$  is added to avoid that the cutted edges are counted twice. In practice, the method does not give satisfactory results. This is shown in Figure 1 where an individual vertex is isolated (partition  $A$ ) instead of the obvious partition  $B$ . To circumvent this, it could be requested that the clusters  $\mathcal{A}_i, i = 1, \dots, k$  are considerably large. This can be achieved in two ways, the first way takes the number of vertices in a set  $\mathcal{A}_i$  into account and the second takes the weights of the edges in consideration:  $\text{vol}(\mathcal{A}_i)$ . So, it results in minimizing one of the two following objective functions

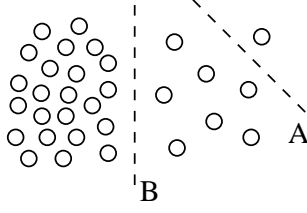


Figure 1: An example where minimum cut gives a bad partition  $A$ ; we should expect partition  $B$ .

[14, 20]:

$$\text{RatioCut}(\mathcal{A}_1, \dots, \mathcal{A}_k) := \frac{1}{2} \sum_{i=1}^k \frac{W(\mathcal{A}_i, \bar{\mathcal{A}}_i)}{|\mathcal{A}_i|}, \quad (1)$$

$$\text{NCut}(\mathcal{A}_1, \dots, \mathcal{A}_k) := \frac{1}{2} \sum_{i=1}^k \frac{W(\mathcal{A}_i, \bar{\mathcal{A}}_i)}{\text{vol}(\mathcal{A}_i)}. \quad (2)$$

Both objective functions (1)-(2) want to achieve that the clusters are balanced, by there corresponding measures. Rewriting the minimization of (1)-(2) for  $k = 2$  results in [14, 20], respectively:

$$\min_{\mathbf{y}} \frac{\mathbf{y}^T L \mathbf{y}}{\mathbf{y}^T \mathbf{y}} \text{ such that } \mathbf{y} \in \{1, -b\}^N, \mathbf{y}^T \mathbf{1}_N = 0, \quad (3)$$

$$\min_{\mathbf{y}} \frac{\mathbf{y}^T L \mathbf{y}}{\mathbf{y}^T D \mathbf{y}} \text{ such that } \mathbf{y} \in \{1, -b\}^N, \mathbf{y}^T D \mathbf{1}_N = 0, \quad (4)$$

where  $L = D - W$  is the unnormalized graph Laplacian,  $\mathbf{y}$  is the indicator vector and  $b$  is a positive constant that depends on the number of data points assigned to each partition,  $b = \frac{|\mathcal{A}_1|}{|\mathcal{A}_2|}$ , and  $b = \frac{\text{vol}(\mathcal{A}_1)}{\text{vol}(\mathcal{A}_2)}$ , respectively.

As stated before, minimizing (3)-(4) is a NP-hard problem. When the discrete condition of  $\mathbf{y}$  is relaxed such that  $\mathbf{y}$  can also take real values, the minimization of (3)-(4) results in solving the following eigenvalue problems, respectively:

$$L \mathbf{y} = \lambda \mathbf{y}, \quad (5)$$

$$D^{-1} L \mathbf{y} = \lambda \mathbf{y}, \quad (6)$$

with  $L_{rw} = D^{-1} L$  the normalized graph Laplacian.

To obtain the approximated solution of (1)-(2), the eigenvectors of  $L$  and  $L_{rw}$  corresponding to the second smallest eigenvalue (also called the Fiedler vector) are the real valued solution to the problems (1)-(2). The indicator vector can be obtained by binarizing the Fiedler vector.

In general, clustering problems consist of  $k$  clusters instead of two clusters. These also reduce to the eigenvalue problems (5)-(6), but in these cases not only the eigenvector belonging to the second smallest eigenvalue is of interest, but all the eigenvectors corresponding to the  $1, \dots, k$  smallest eigenvalues. Obtaining the indicator vectors is not that simple anymore for  $k > 2$  and often another clustering algorithm is applied to cluster the  $k$  eigenvectors. For instance,  $k$ -means which only works well if the clusters in the new space represented by the eigenvectors are spherical and well-separated.

Previously, we defined the graph Laplacian  $L = D - W$ . In fact, this is the main tool for spectral clustering and has been extensively investigated in spectral graph theory [9]. We will briefly discuss the main properties of three types of graph Laplacians:

$$\begin{array}{ll} L & = D - W & \text{unnormalized graph Laplacian,} \\ L_{rw} & = D^{-1} L = I - D^{-1} W & \text{normalized graph Laplacian,} \\ L_{sym} & = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2} & \text{symmetric normalized graph Laplacian.} \end{array}$$

All three graph Laplacians are symmetric positive semi-definite matrices and they have the basic property that the smallest eigenvalue is 0 and that the corresponding eigenvector is the constant one vector  $\mathbf{1}_N$ , except for  $L_{sym}$  where the eigenvector is a scaled version:  $D^{1/2}\mathbf{1}_N$ . There exists also a relation between the number of connected components and the multiplicity of the eigenvalue 0 of the Laplacian. But first we define the indicator vector  $\mathbf{1}_{\mathcal{A}_1}$  as

$$(\mathbf{1}_{\mathcal{A}_1})_i = \begin{cases} 1, & i \in \mathcal{A}_1, \\ 0, & i \notin \mathcal{A}_1. \end{cases}$$

**Proposition 1** *Let  $G$  be an undirected graph with nonnegative weights. Then the multiplicity  $k$  of the eigenvalue 0 of  $L$ ,  $L_{rw}$  and  $L_{sym}$  equals the number of connected components  $\mathcal{A}_1, \dots, \mathcal{A}_k$  in the graph. For  $L$ ,  $L_{rw}$ : the eigenspace of eigenvalue 0 is spanned by the vectors  $\mathbf{1}_{\mathcal{A}_1}, \dots, \mathbf{1}_{\mathcal{A}_k}$  of those components, and for  $L_{sym}$ : the eigenspace of 0 is spanned by the vectors  $D^{1/2}\mathbf{1}_{\mathcal{A}_i}, i = 1, \dots, k$ .*

**Proof** See [23].

In general, to obtain an approximated solution to the minimization problem, the constant eigenvector  $\mathbf{1}_N$  belonging to the smallest eigenvalue is of no interest. The eigenvectors of interest (for  $k$  clusters) are the eigenvectors corresponding to the  $1, \dots, k$  smallest eigenvalues.

The difference between the unnormalized and normalized Laplacian, is that they both take care of minimizing the between-cluster similarity, but only the normalized takes care of maximizing the within-cluster similarity. That is why the choice often goes to the normalized Laplacian. Between the two normalized Laplacians often the  $L_{rw}$  is elected above  $L_{sym}$  because of the simple fact that the eigenvectors of  $L_{rw}$  are indicator vectors and these of  $L_{sym}$  have to be multiplied with the matrix  $D^{1/2}$ .

For more detailed information of the graph Laplacian and other basic properties the reader is referred to [9, 17, 18, 23]. Also pay attention to the fact that in the literature no unique convention exists about the name graph Laplacian.

In general, a spectral clustering algorithm passes through the following steps: first the graph Laplacian and the related eigenvalue problem are constructed, then the corresponding eigenvalue problem is solved. When the eigenvectors are obtained the cluster assignment can be carried out by computing the indicator vectors. The algorithm in Section 3 will also consist of these three steps. Also an extra step is added where the method is extended to out-of-sample points.

### 3 Clustering algorithm

In this section, a spectral clustering method based on linear algebra techniques is proposed. To acquire the eigenvectors, the incomplete Cholesky (IC) decomposition is used to reduce the eigenvalue problem such that for large data sets the computational burden is not prohibitive anymore. In fact, the incomplete Cholesky decomposition takes care that a sparse set of pivots of the full data set is selected, and the number of pivots is controlled by a new stopping criterion. As stated in [24] the cluster assignment results in computing a pivoted  $LQ$  decomposition of the eigenvector matrix. We will adapt this method such that the importance of the selected data points is taken into account. Additionally this formulation based on the  $LQ$  decomposition can be used to extend the method for out-of-sample points. At the end, an overview of the algorithm is given.

#### 3.1 Cholesky decomposition

A Cholesky decomposition [12] is a decomposition of a symmetric positive definite matrix into the product of a lower triangular matrix and its conjugate transpose:  $A = CC^T$  and is widely used for solving linear systems. When the matrix is positive semi-definite it is possible to compute the incomplete Cholesky decomposition, meaning that some columns of  $C$  are zero. In fact, the incomplete Cholesky decomposition computes a low rank approximation of accuracy  $\eta$  of the matrix in  $\mathcal{O}(r^2N)$  such that  $\|A - CC^T\| < \eta$  with  $C \in \mathbb{R}^{N \times r}$ . As stated in [4], the incomplete

Cholesky decomposition leads to small numerical error and  $r \ll N$  when there is a fast decay of eigenvalues. The efficient computation of the incomplete Cholesky decomposition is shown in Algorithm 1 [5].

---

**Algorithm 1** Incomplete Cholesky decomposition

---

- 1: Put  $i = 1$ ,  $W' = W$ ,  $P = I$ ,  $G_j = W_{jj}$  for  $j = 1, \dots, N$ ,
  - 2: **while**  $\sum_{j=i}^N G_j > \eta$  **do**
  - 3: Find new pivot element  $j^* = \arg \max_{j \in [i, N]} G_j$ .
  - 4: Update permutation  $P$ :  $P_{ii} = P_{j^*j^*} = 0$  and  $P_{ij^*} = P_{j^*,i} = 1$ .
  - 5: Permute elements  $i$  and  $j^*$  in  $W'$ :  $W'_{1:N,i} \leftrightarrow W'_{1:N,j^*}$  and  $W'_{i,1:N} \leftrightarrow W'_{j^*,1:N}$ .
  - 6: Update the already calculated elements of  $C$ :  $C_{i,1:i} \leftrightarrow C_{j^*,1:i}$ .
  - 7: Set  $C_{ii} = \sqrt{W'_{ii}}$ .
  - 8: Calculate  $i^{\text{th}}$  column of  $C$ :  $C_{i+1:n,i} = \frac{1}{C_{ii}} (W'_{i+1:N,i} - \sum_{j=1}^{i-1} C_{i+1:N,j} C_{ij})$ .
  - 9: Update only diagonal elements: for  $j = i + 1, \dots, N$ :  $G_j = G_j - C_{ji}^2$ .
  - 10: Set  $i = i + 1$ .
  - 11: **end while**
- 

In interior point methods of support vector machines (SVM) this factorization is used to decrease the storage requirement and the computational complexity [11, 4], and in independent component analysis (ICA) it is used to approximate contrast functions in an efficient way [3, 5, 13, 2]. All these methods use this factorization as a low rank approximation.

The idea of this paper, by analogy with [4], is to approximate the similarity matrix  $W$ , which is a positive semi-definite matrix, with the incomplete Cholesky decomposition to reduce the eigenvalue problem. After investigation of the similarity matrices utilized in [4], we noticed that these matrices did not possess eigenvalues decaying fast, i.e. these matrices are not approximately rank deficient. This is shown in Figure 2(b)-(d) where there is no rapid decay of eigenvalues for the two data sets shown in Figure 2(a)-(c), respectively. Hence, it is not possible to approximate efficiently the matrix with a matrix of low rank. Throughout this paper the radial basis function (RBF)  $K(\mathbf{x}, \mathbf{y}) = \exp(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2})$  is taken as similarity measure.

In fact, the incomplete Cholesky decomposition works because it selects in an appropriate manner the rows and the columns such that the structure of the approximation is close to the structure of the original matrix. This is beneficial in spectral clustering because the structure of the eigenvectors plays a crucial role.

The selection procedure of the incomplete Cholesky decomposition will be elaborated for an ideal example with two disjunct clusters. Assume that the points are ordered such that  $W$  is a block diagonal matrix with two blocks

$$W = \begin{bmatrix} W_1 & 0 \\ 0 & W_2 \end{bmatrix}.$$

To obtain a correct clustering a minimum of two columns has to be selected, each column from another cluster. This is exactly what the incomplete Cholesky decomposition does.

Lets have a closer look at Algorithm 1 of the incomplete Cholesky decomposition. First it selects a pivot element  $j^*$  by looking for the location of the maximum of the diagonal  $G$ . For this example the radial basis function (RBF) is taken as similarity measure, resulting in a diagonal consisting of only ones. Hence, the first selected pivot element is  $j^* = 1$ . This column is put as the first column\* of  $C$

$$C = \begin{bmatrix} (W_1)_{1:N/2,1} \\ \mathbf{0} \end{bmatrix}. \quad (7)$$

The diagonal  $G$  is updated as follows:

$$G_i = G_i - C_{i,1}^2 = [\mathbf{1}_N] - \begin{bmatrix} (W_1)_{i,1}^2 \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{1}_{N/2} - (W_1)_{i,1}^2 \\ \mathbf{1}_{N/2} \end{bmatrix}. \quad (8)$$

---

\*The notation is MATLAB like notation (MATLAB is a registered trademark of The MathWorks, Inc.): The colon notation has to be interpreted as follows:  $i : m = [i, i + 1, i + 2, \dots, m]$  and  $A(i : m, j : n) = A_{i:m,j:n}$ .

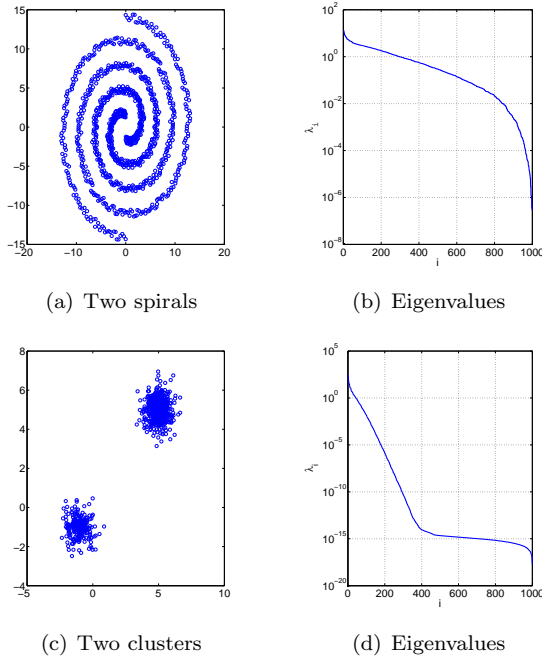


Figure 2: Eigenvalue spectrum: (a) data: two spirals ( $N = 1000$ ), (b) eigenvalue spectrum for two spirals ( $\sigma = 0.45$ ), (c) data: two clusters ( $N = 1000$ ), (d) eigenvalue spectrum for two clusters ( $\sigma = 0.5$ ).

This means that the updated entries of  $G_i$  corresponding to the cluster to which  $j^* = 1$  belongs will have a small value at least smaller than one, and the points related to the other cluster will have still the value one. Hence, an element from the other cluster will be selected as new pivot element  $j^* = \frac{N}{2} + 1$ . The second column of  $C$  becomes

$$C_{1:N,2} = \begin{bmatrix} \mathbf{0} \\ (W_2)_{1:N/2,1} \end{bmatrix}. \quad (9)$$

These two columns in  $C$  are enough to approximate matrix  $W$  in such a way that the structure of matrix  $W$  is captured:

$$W \approx CC^T = \begin{bmatrix} (W_1)_{1:N/2,1}(W_1)_{1:N/2,1}^T & \mathbf{0} \\ \mathbf{0} & (W_2)_{1:N/2,1}(W_2)_{1:N/2,1}^T \end{bmatrix} \quad (10)$$

As you can see, the approximation of  $W$  consists of two blocks. Therefore the structure of the eigenvectors of the Laplacian are similar to the original one.

In practice, the matrix  $W$  is not a block diagonal matrix but a permuted version of it and the anti-diagonal blocks also contain information about the data points. For instance, in the example of the two intermingled spirals as shown in Figure 2(a), the distance between points in different clusters can be smaller than between points in the same cluster. This information is also incorporated in the similarity matrix  $W$ . For this type of examples, it will not work to select two pivots each from a different cluster and solve the corresponding eigenvalue problem, more pivots have to be selected. The incomplete Cholesky decomposition is able to capture this in contrast with other clustering algorithms e.g.  $k$ -means, which can not handle non linear data structures.

In Figure 3 different clusterings are shown based on a different number of selected pivots. The clustering in Figure 3(a) is achieved with the incomplete Cholesky decomposition but not enough pivots ( $r = 70$ ) were selected to obtain a correct clustering. In Figure 3(b) more pivots ( $r = 200$ )

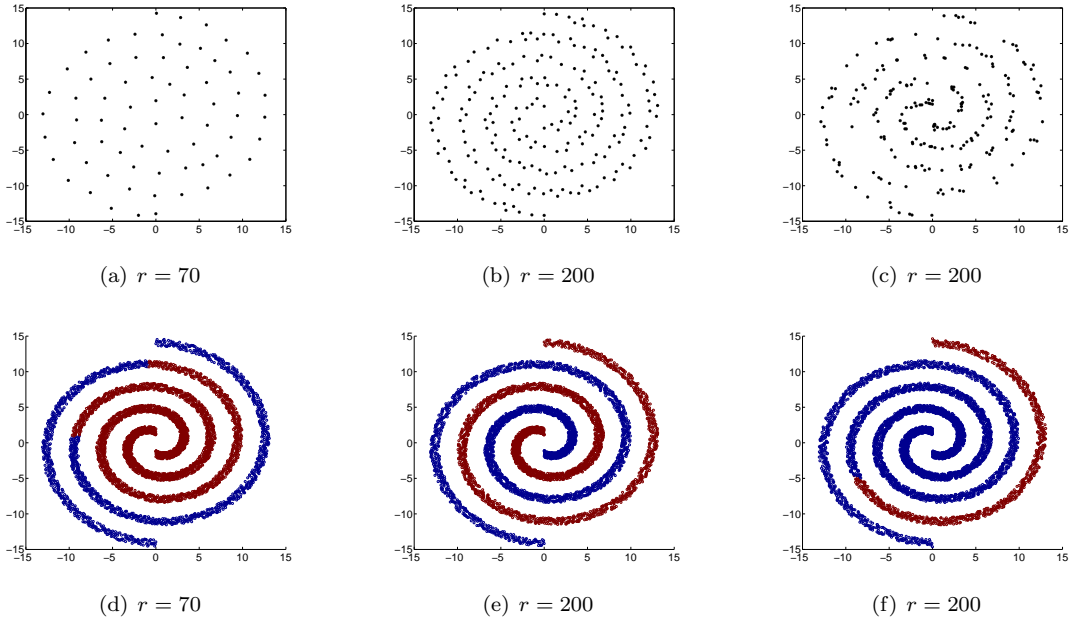


Figure 3: Pivots and resulting clusters ( $N = 1000, \sigma = 0.5$ ) for (a)-(d) Cholesky  $r = 70$ , (b)-(e) Cholesky  $r = 200$ , (c)-(f) Random  $r = 200$ .

are selected and the correct clustering is obtained. The clustering in Figure 3(c) is achieved by selecting the pivots ( $r = 200$ ) arbitrarily and as you can see no correct clustering is obtained. We can conclude that a correct clustering is received when a minimal amount of pivots is selected such that the clusters are visually notable (only in Figure 3(b) this is the case).

Notice that the selected pivots are on a certain distance from each other. This depends on the radial basis function parameter  $\sigma$ . A larger  $\sigma$  will result in pivots which are further away from each other, such that the pivots data set is sparse but not a good representation of the full data set. A smaller  $\sigma$  results in points which are closer to each other, such that more pivots are selected to obtain the correct clustering. The selection of the parameter  $\sigma$  will not be discussed further in this paper. It is assumed that an appropriate  $\sigma$  for each problem is available.

### 3.1.1 Stopping criterion

As shown above the incomplete Cholesky decomposition selects the pivots in such a way that a sparse representation of the data set is obtained. The next step is to focus on the stopping criterion of the decomposition, because the common stopping criterion is based on a low rank approximation. We want that the algorithm stops when a minimal number of pivots is selected which ensures a correct clustering.

The stopping criterion will be based on the degree of each node in the approximation of the similarity matrix  $W$ . As defined before the degree of a node  $d_j$  represents all the connections to the other nodes. When matrix  $W$  is approximated, also not the exact degree matrix can be constructed. Therefore an approximation of the degree  $\tilde{d}_j$  will be used in this method. Only the connections of the node to the selected pivots can be taken into account, i.e. it corresponds to  $\text{diag}(\tilde{D}) = CC^T \mathbf{1}_N$ .

In fact, each node must have a certain degree to ensure a good clustering, there must be enough connections to the already selected pivots. And if there is still a node with degree zero, a new pivot should be selected. After selection of a new pivot the degrees of the nodes are updated and based on these values the stopping criterion is verified. From experience, it is found that the

following stopping condition

$$\frac{\min \tilde{d}_j}{\max \tilde{d}_j} > 10^{-3} \quad (11)$$

gives satisfactory results. This will be further discussed in the numerical experiments of Section 4.

### 3.2 Reducing the eigenvalue problem

In this section, the reduction to a smaller eigenvalue problem is proposed, it is similar to the method proposed in [4]. In this paper we are mainly interested in solving the following eigenvalue problem:

$$\hat{L}_{sym} \mathbf{y} = \hat{\lambda} \mathbf{y}.$$

Notice that the eigenvalues of

$$\hat{L}_{sym} = D^{-1/2} W D^{-1/2}, \quad (12)$$

are related to those of  $L_{sym}$  as  $\hat{\lambda} = 1 - \lambda$ . Hence, the largest eigenvalues become important instead of the smallest. As explained in Section 3.1 the incomplete Cholesky decomposition of  $W$  is:  $W \approx CC^T$  with  $C \in \mathbb{R}^{N \times r}$  and  $r$  the number of selected pivots. As we substitute this, together with the related degree matrix  $\tilde{D}$ , in (12):

$$\hat{L}_{sym} \approx \tilde{D}^{-1/2} CC^T \tilde{D}^{-1/2}. \quad (13)$$

To reduce the eigenvalue problem, replace  $\tilde{D}^{-1/2} C$  with its  $QR$  decomposition:  $\tilde{D}^{-1/2} C = QR$  where  $Q \in \mathbb{R}^{N \times r}$  and  $R \in \mathbb{R}^{r \times r}$  and substitute  $R$  with its singular value decomposition  $R = U_R \Sigma_R V_R^T$  where  $U_R, V_R \in \mathbb{R}^{r \times r}$  and  $\Sigma_R \in \mathbb{R}^{r \times r}$ . Equation (13) results in:

$$\begin{aligned} \hat{L}_{sym} &\approx (QR)(QR)^T \\ &\approx Q(U_R \Sigma_R V_R^T)(V_R \Sigma_R U_R^T) Q^T \\ &\approx QU_R (\Sigma_R)^2 U_R^T Q^T. \end{aligned}$$

The columns of the matrix  $\tilde{V} = QU_{R,1:k}$  with  $\tilde{V} \in \mathbb{R}^{N \times k}$  are the  $k$  orthogonal eigenvectors  $\tilde{\mathbf{v}}_j$  with respect to the  $k$  dominant eigenvalues  $(\sigma_{R,j})^2 = \tilde{\lambda}_j$  with  $j = 1, \dots, k$ .

### 3.3 Cluster assignment

To explain the cluster assignment the ideal case of  $k$  connected components, with the  $k$  dominant eigenvalues  $\tilde{\lambda}_1 = \dots = \tilde{\lambda}_k = 1$  and the corresponding eigenvectors  $\tilde{\mathbf{v}}_j, j = 1, \dots, k$ , is considered. According to Proposition 1, the following decomposition holds:

$$\tilde{D}^{-1/2} [\tilde{\mathbf{v}}_1 \dots \tilde{\mathbf{v}}_k] = [\mathbf{1}_{A_1} \dots \mathbf{1}_{A_k}] D_I Q_I,$$

where  $D_I \in \mathbb{R}^{k \times k}$  is the matrix containing the scaling parameters and  $Q_I \in \mathbb{R}^{k \times k}$  an orthogonal matrix. Extracting the indicator vectors from the  $k$  eigenvectors, can be achieved by computing a pivoted  $LQ$  decomposition of the eigenvector matrix  $\tilde{D}^{-1/2} \tilde{V}$ , as proposed in [24]:

$$\tilde{D}^{-1/2} \tilde{V} = PLQ_{\tilde{V}} = P \begin{bmatrix} L_{11} \\ L_{22} \end{bmatrix} Q_{\tilde{V}},$$

with  $P \in \mathbb{R}^{n \times n}$  a permutation matrix,  $L_{11} \in \mathbb{R}^{k \times k}$  lower triangular matrix,  $L_{22} \in \mathbb{R}^{(N-k) \times k}$  and  $Q_{\tilde{V}} \in \mathbb{R}^{k \times k}$  an orthogonal matrix. Put

$$\hat{L} = \begin{bmatrix} L_{11} \\ L_{22} \end{bmatrix} L_{11}^{-1} = \begin{bmatrix} I_k \\ L_{22} L_{11}^{-1} \end{bmatrix}.$$

Then the columns of  $S = P\hat{L}$  are the indicator vectors:

$$S = [\mathbf{1}_{A_1} \dots \mathbf{1}_{A_k}].$$

In fact, the underlying process of the  $LQ$  factorization of a matrix  $A$  can be explained by aid of the Gram-Schmidt process [12]. Select the row  $\mathbf{a}_i$  with maximum residual norm and put  $\tilde{\mathbf{a}}_1 := \mathbf{a}_i$  and  $\mathbf{u}_1 := \mathbf{a}_i$ . Say that the corresponding point  $\mathbf{x}_i$  belongs to cluster  $j$ , i.e. this is the representative of the cluster  $j$ . Calculate for each row ( $l = 1, \dots, N$  and  $l \neq i$ ) the difference between the row and its orthogonal projection onto the row  $\mathbf{u}_1$ :  $\tilde{\mathbf{u}}_l := \mathbf{a}_l - \text{proj}_{\mathbf{u}_1}(\mathbf{a}_l)$ . When a resulting row has small norm the corresponding point belongs to cluster  $j$ , and if the norm is large, it means that the corresponding point belongs to another cluster.

Next pick the row  $i^*$  with largest residual norm and put  $\tilde{\mathbf{a}}_2 := \mathbf{a}_{i^*}$  and  $\mathbf{u}_2 := \tilde{\mathbf{u}}_{i^*}$ , the corresponding point will be the representative of another cluster. Then calculate, for the other rows ( $l \neq i, i^*$ ), the difference between the row and its projection onto the subspace  $\{\mathbf{u}_1, \mathbf{u}_2\}$ :  $\mathbf{a}_l - \text{proj}_{\mathbf{u}_1}(\mathbf{a}_l) - \text{proj}_{\mathbf{u}_2}(\mathbf{a}_l)$ . Pick the row with largest norm and call it  $\mathbf{u}_3$ . This process is continued for  $k$  steps. The pivoted  $LQ$ -factorization of the matrix  $A$  can then be obtained by:

$$L = \begin{bmatrix} \langle \mathbf{e}_1, \tilde{\mathbf{a}}_1 \rangle & 0 & \dots & 0 \\ \langle \mathbf{e}_1, \tilde{\mathbf{a}}_2 \rangle & \langle \mathbf{e}_2, \tilde{\mathbf{a}}_2 \rangle & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \langle \mathbf{e}_1, \tilde{\mathbf{a}}_k \rangle & \langle \mathbf{e}_2, \tilde{\mathbf{a}}_k \rangle & \dots & \langle \mathbf{e}_k, \tilde{\mathbf{a}}_k \rangle \\ \langle \mathbf{e}_1, \tilde{\mathbf{a}}_{k+1} \rangle & \langle \mathbf{e}_2, \tilde{\mathbf{a}}_{k+1} \rangle & \dots & \langle \mathbf{e}_k, \tilde{\mathbf{a}}_{k+1} \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \mathbf{e}_1, \tilde{\mathbf{a}}_N \rangle & \langle \mathbf{e}_2, \tilde{\mathbf{a}}_N \rangle & \dots & \langle \mathbf{e}_k, \tilde{\mathbf{a}}_N \rangle \end{bmatrix}, \quad Q = [\mathbf{e}_1 \ \mathbf{e}_2 \ \dots \ \mathbf{e}_k]^T \quad (14)$$

with  $\mathbf{e}_i = \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|}$ . At the end,  $k$  representative points are received related to the selected rows  $\mathbf{e}_l$  ( $l = 1, \dots, k$ ), and each point represents a cluster. The matrix  $L$  is also multiplied with the inverse of the square  $k \times k$  matrix  $L_{11}$ , consisting of the  $k$  top rows of  $L$ , such that in the resulting matrix  $\hat{L}$  the selected rows  $\mathbf{e}_l$  ( $l = 1, \dots, k$ ) are the actual representatives of the  $k$  clusters. The entries of the matrix  $\hat{L}$  tell how well row  $\tilde{\mathbf{a}}_n$  ( $n = 1, \dots, N$ ) is related to  $\mathbf{e}_l$  ( $l = 1, \dots, k$ ).

In the ideal case, the columns of matrix  $S$  will consist of the cluster indicators

$$S = [\mathbf{1}_{A_1} \ \dots \ \mathbf{1}_{A_k}].$$

In practice, when there are almost  $k$  connected components, the cluster structure is still inherited but the zeros are not zero anymore. The magnitude (in absolute value) of the entries,  $s_{ij}$  of the matrix  $S = P\hat{L} \in \mathbb{R}^{N \times k}$  indicate how well point  $\mathbf{x}_i$  is assigned to cluster  $j$ . Then a point  $\mathbf{x}_i$  can be assigned to a cluster  $j$ : when

$$j = \arg \max_l (|(S)_{i,l}|).$$

Hence, the matrix  $S$  gives also a measure of how good or bad a point belongs to a certain cluster based on the  $k$  representatives. This measure could be used to detect outliers in the data.

Because the proposed algorithm is based on the approximation of the similarity matrix  $W$ , only the correct information of a few points (the selected pivots) is exploited, the information of the other points is approximated. To take this into consideration in the cluster assignment, the eigenvectors  $\tilde{V}$  are scaled with the degree matrix  $\tilde{D}^{1/2}$  such that if the approximated degree of a node is small, this point will not be selected as a representative of a cluster. This corresponds to the operation  $\tilde{D}(\tilde{D}^{-1/2}\tilde{V})$  on the eigenvectors.

### 3.4 Generalization to out-of-sample points

A method using only the selected pivots, is proposed to assign the out-of-sample (test<sup>†</sup>) points  $\{\mathbf{x}_t\}_{t=1}^{N_{test}}$  to a cluster based on the technique explained in Section 3.3. The method runs through

<sup>†</sup>In general, a data set is divided into three sets, validation set (to validate the parameters, e.g. the RBF parameter  $\sigma$ , the number of clusters  $k$ ), training set (to obtain the indicator vectors) and the test set (to test the indicator vectors to new points).

the same steps as the eigenvector computing and the cluster assignment method by using the information received during these steps. In the next formulae, the original steps (at the left) are shown together with the steps necessary for the extended version (at the right).

The first step, of course is to compute the similarity matrix between the out-of-sample points  $\{\mathbf{x}_t\}_{t=1}^{N_{test}}$  and the selected pivots  $\{\tilde{\mathbf{x}}_m\}_{m=1}^r$ :

$$W = CC^T \quad \rightarrow \quad W_{ext} = \begin{bmatrix} C \\ C_{ext} \end{bmatrix} [ C^T C_{ext}^T ],$$

where  $C_{ext} \in \mathbb{R}^{N_{test} \times r}$ . The second step is to compute an extended version of the eigenvectors, instead of recomputing the eigenvector matrix of size  $(N + N_{test}) \times k$  only the bottom  $N_{test}$  rows are computed. This computation is based on information from the original decomposition and  $C_{ext}$ . First the analog of the  $QR$  decomposition of  $\tilde{D}^{-1/2}C$  is applied:

$$\tilde{D}^{-1/2}C = QR \quad \rightarrow \quad Q_{ext} = \tilde{D}_{ext}^{-1/2}C_{ext}R^{-1},$$

with  $Q_{ext} \in \mathbb{R}^{N_{test} \times r}$  and  $\text{diag}(\tilde{D}_{ext}) = C_{ext}C_{ext}^T \mathbf{1}_{N_{test}}$ . In the third step, the information of the eigenvectors  $U_{R,k}$  is applied to obtain the new rows of the eigenvector matrix  $\tilde{V}_{ext} \in \mathbb{R}^{N_{test} \times k}$ .

$$\tilde{V} = QU_{R,k} \quad \rightarrow \quad \tilde{V}_{ext} = Q_{ext}U_{R,k}.$$

When the extension of the eigenvector matrix,  $\tilde{V}_{ext}$  is obtained, these rows have to be transformed in the same way as all previous rows of the eigenvector matrix, as in Section 3.3, to obtain the cluster assignments. This is done by applying  $Q_{\tilde{V}}^T$ .

$$\tilde{D}_{\tilde{V}}\tilde{V} = SQ_{\tilde{V}} \quad \rightarrow \quad S_{ext} = \tilde{D}_{\tilde{V}_{ext}}\tilde{V}_{ext}Q_{\tilde{V}}^T,$$

with  $S_{ext} \in \mathbb{R}^{N_{test} \times k}$ . Then the same cluster assignment criterion can be used to assign the out-of-sample point  $\mathbf{x}_i$  to a cluster: assign point  $\mathbf{x}_i$  to cluster  $j$  when

$$j = \arg \max_l (|(S_{ext})_{i,l}|).$$

### 3.5 Algorithm

An overview of the algorithm is given in Algorithm 2.

---

#### Algorithm 2 Sparse Model for Spectral Clustering Using the Incomplete Cholesky Decomposition

---

- 1: Compute the incomplete Cholesky factor  $C \in \mathbb{R}^{N \times r}$  of the matrix  $W$  such that matrix  $CC^T$  captures the structure of matrix  $W$  and obtain the sparse set  $\mathcal{R} = \{\tilde{\mathbf{x}}_m\}_{m=1}^r$  of pivots.
  - 2: Compute the  $QR$  decomposition of  $\tilde{D}^{-1/2}C = QR$  with  $Q \in \mathbb{R}^{N \times r}$  and  $R \in \mathbb{R}^{r \times r}$ .
  - 3: Compute the singular value decomposition of  $R = U\Sigma V^T$ .
  - 4: Obtain the approximated eigenvectors via:  $\tilde{V} = QU_{R,k}$ .
  - 5: Compute  $LQ$  factorization with row pivoting  $D_{\tilde{V}}\tilde{V} = PLQ_{\tilde{V}}$  and put  $S = P\hat{L}$  with  $\hat{L} = \begin{bmatrix} L_{11}^T & L_{22}^T \end{bmatrix}^T L_{11}^{-1}$ .
  - 6: For all  $i$ , assign point  $\mathbf{x}_i$  to cluster  $j$  when  $j = \arg \max_l (|S_{i,l}|)$ .
  - 7: Compute  $C_{ext} \in \mathbb{R}^{N_{test} \times r}$ .
  - 8: Compute  $Q_{ext} = \tilde{D}_{ext}^{-1/2}C_{ext}R^{-1}$  and  $\tilde{V}_{ext} = Q_{ext}U_R$  and put  $S_{ext} = D_{\tilde{V}_{ext}}\tilde{V}_{ext}Q_{\tilde{V}}^T$ .
  - 9: For all  $t$ , assign point  $x_t$  to cluster  $j$  with  $j = \arg \max_l (|(S_{ext})_{i,l}|)$ .
- 

*Note:* The proposed method has a few similarities with the method proposed by Alzate and Suykens in [4], like the use of the incomplete Cholesky decomposition and the reduction to a

smaller eigenvalue problem. But there are also significant differences. The method proposed by these authors is based on a kernel principal component formulation and leads to the following eigenvalue problem:

$$MW\mathbf{y} = \lambda\mathbf{y} \quad (15)$$

with  $M = D^{-1} - \frac{1}{\mathbf{1}_N^T D^{-1} \mathbf{1}_N} D^{-1} \mathbf{1}_N \mathbf{1}_N^T D^{-1}$  a weighted centering matrix removing the weighted mean from each column of  $W$ . Another difference is the fact that they use the incomplete Cholesky decomposition with the original stopping criterion, this results in an extra parameter in their spectral clustering algorithm, which has to be chosen in a proper way.

## 4 Numerical experiments

In this section, the proposed method is compared with the method of Alzate and Suykens (AS) [4] on different toy problems. Because the method *AS* depends on an extra parameter  $\eta$ , we also compare the results with an adapted version of the method of *AS*, this method is denoted with method AS (\*) and the number of pivots is given as input such that the proposed method and method AS (\*) can have the same number of pivots. All methods are implemented in MATLAB.

The first experiment shows the sparseness of the proposed method and the idea behind the stopping condition (11) for two different problems. It shows that the method learns what it has to learn in an efficient way. The second experiment shows the results concerning the out-of-sample extensions for three different problems.

In the first experiment, the whole data set will be considered as training set and in the second experiment, the data set will be divided into a training and test set. The results are compared with a known clustering by the Adjusted Rand Index [16] which measures the similarity between two data clusterings, where an ARI equal to one stands for equal clusterings. The simulations are performed ten times, so average results are shown in the figures and tables.

- Experiment 1: Sparseness of proposed method
  - Three Gaussian clouds in  $3D$ : In Figure 4(a) the three clouds are shown. The number of training points is  $N = 6000$  and the RBF parameter  $\sigma$  is fixed to  $\sigma = 3$ . In Figure 4(b) the stopping criterion (in logarithmic scale) for the first 100 selected pivots of the proposed method is shown. It gives the value (11) after each selection of a new pivot. As you can see, the stopping criterion is fulfilled when three pivots are selected, and a correct clustering is obtained.

In this experiment, the influence  $\eta$  ( $10^{-5} \leq \eta \leq 1$ ) of method AS is compared to the proposed method which does not have an extra parameter. In Figure 4(c), the number of selected pivots is shown for the two methods. The proposed method does not depend on the parameter  $\eta$ , so its result stays fixed (dotted line). For the method AS the number of selected pivots decreases when  $\eta$  increases. For  $\eta$  close to one, the same number of pivots is selected as in the proposed method. For the last value of  $\eta$  is the number of selected points 2.2 (not an integer because of the randomization) even smaller than the value of the proposed method. In this case the Adjusted Rand Index does not give 1 but 0.6138, so no correct clustering is obtained. In fact, for three clouds at least three pivots have to be selected to obtain a correct clustering. In Figure 4(d) the computation times in seconds are shown, this decreases also when  $\eta$  increases for method AS but it is higher than the time of the proposed method.
  - Two spirals in  $2D$ : In this experiment two spirals are considered as shown in Figure 5(c). First we consider  $N = 1000$  data points and the RBF kernel parameter is set to  $\sigma = 0.4$ . In Figure 5(a)-(b) the stopping criterion is shown with respect to the number of selected pivots. Figure 5(a) gives a general impression how the stopping condition (11) behaves, Figure 5(b) gives the result in a specific interval [50, 200]. The red dot, indicates that from this point on a correct clustering (ARI=1) is obtained. Notice that at that moment the stopping criterion is not fulfilled. In Figure 5(c) the selected pivots are shown, the

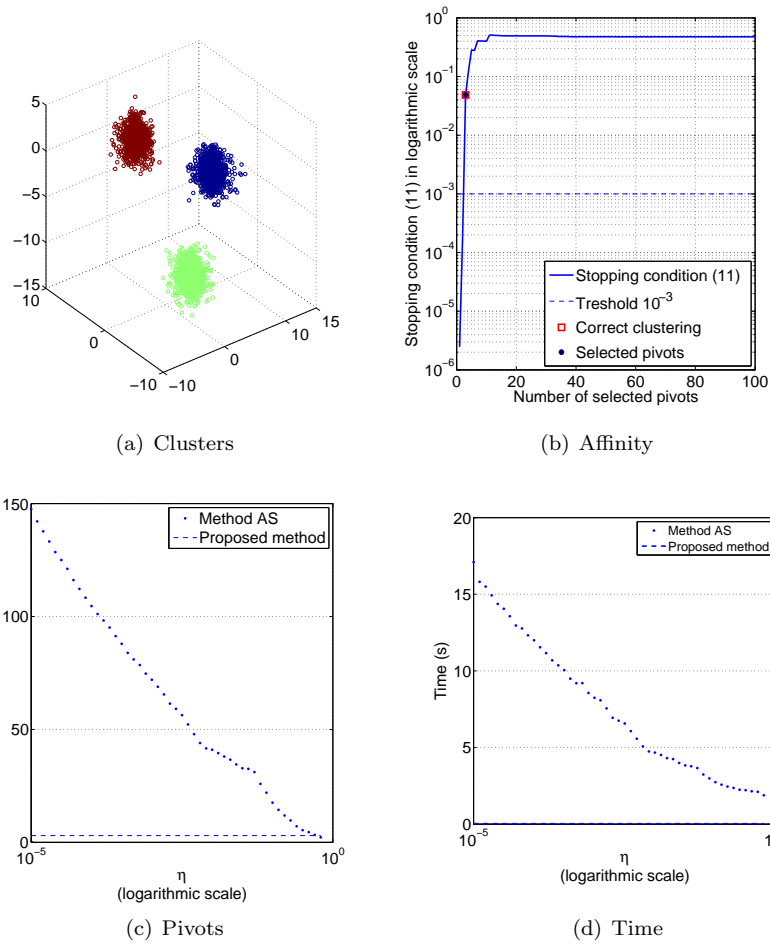


Figure 4: Three Gaussian clouds ( $N = 6000$ ,  $\sigma = 3.8$ ): (a) Clustering obtained with proposed method, (b) Stopping criterion for the first 100 selected pivots, (c) Number of pivots selected, (d) Computation time.

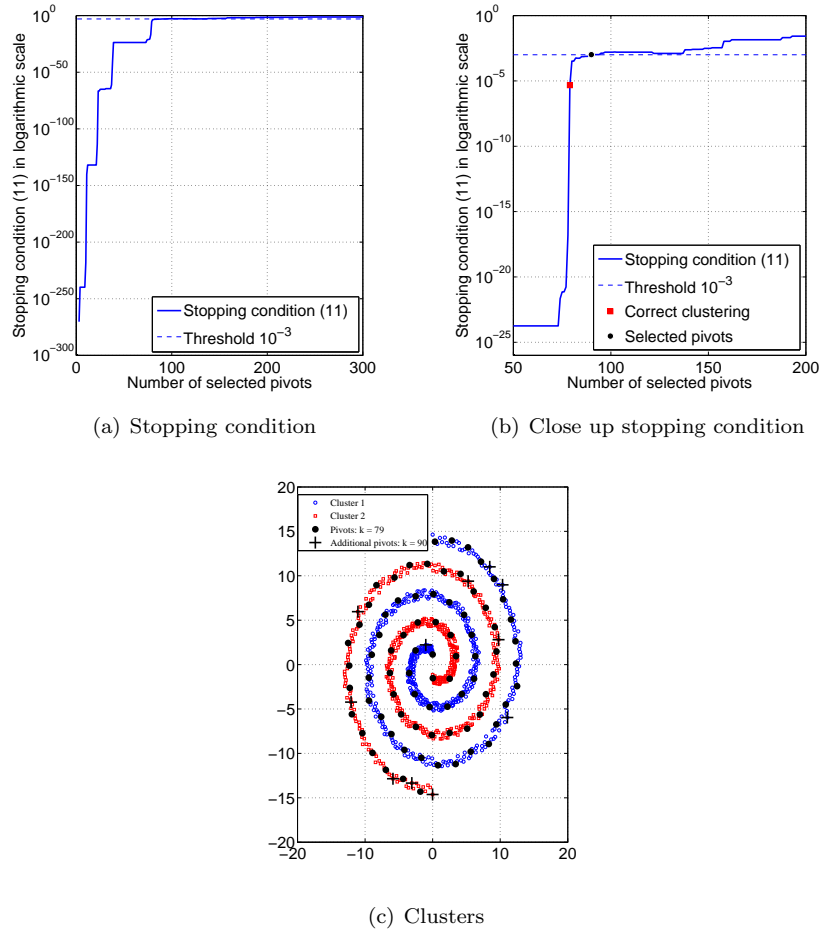


Figure 5: Intertwined spirals ( $N = 1000$ ,  $\sigma = 0.4$ ): (a) Stopping criterion for the first 300 selected pivots, (b) Close up of stopping criterion, (c) Clustering obtained with proposed method with selected pivots.

dots denote the ones that are enough to obtain the correct clustering, the plus signs are the extra pivots which are selected until the stopping condition is fulfilled.

In Figure 6(a)-(b), the number of selected pivots and the computation times are shown for increasing  $N$  and  $\eta = 0.7$ . As you can see, the proposed method selects less pivots, and the computation times reduce significantly. Also in comparison with method AS (\*) the reduction time is significant. Note that the full approach does not fit into memory. In Table 1, the number of selected pivots is shown for the method of AS and the proposed method. Also the degree of sparseness is indicated.

- Experiment 2: Out-of-sample extensions

- Three Gaussian clouds in  $2D$ : In this experiment, the effect of the number of training points will be investigated on an almost ideal problem (clusters well separable) and on a non-ideal problem (clusters hard to separate), see Figure 7(a) and Figure 8(a). The number of data points is  $N = 900$ , the RBF parameter  $\sigma$  is set to 0.8 and 0.5 respectively, and  $\eta$  is taken 0.5 in both cases. The number of training points varies from  $N_{train} = 20, \dots, 880$  with steps of 20, and the remaining data points are attributed to the test set.

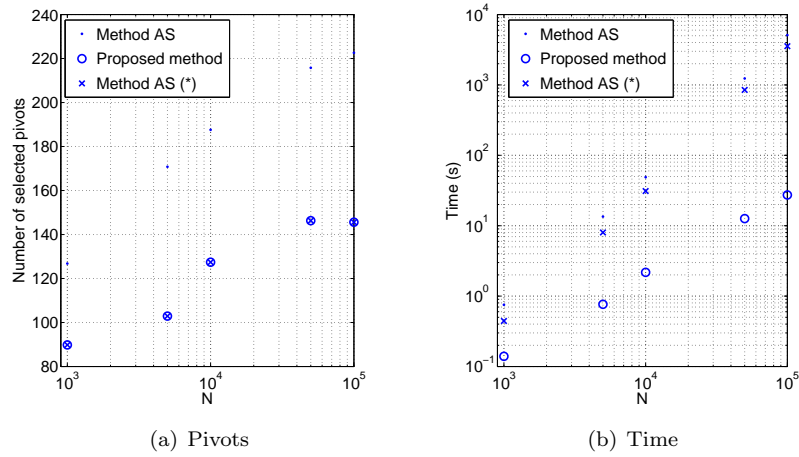


Figure 6: Two spirals ( $\sigma = 0.4$ ,  $\eta = 0.7$ ): (a) Number of selected pivots with respect to  $N$ , (b) Computation times with respect to  $N$ .

Size $N$	Pivots Method AS	Pivots proposed method
1000	129 (87.1%)	94 (90.6%)
5000	169 (96.6%)	100 (98.0%)
10000	186 (98.1%)	121 (98.8%)
50000	216 (99.6%)	143 (99.7%)
100000	224 (99.8%)	144 (99.9%)

Table 1: Two spirals: Number of selected pivots  $r$  with respect to  $N$  for method AS and proposed method. The percentage indicates the degree of sparseness.

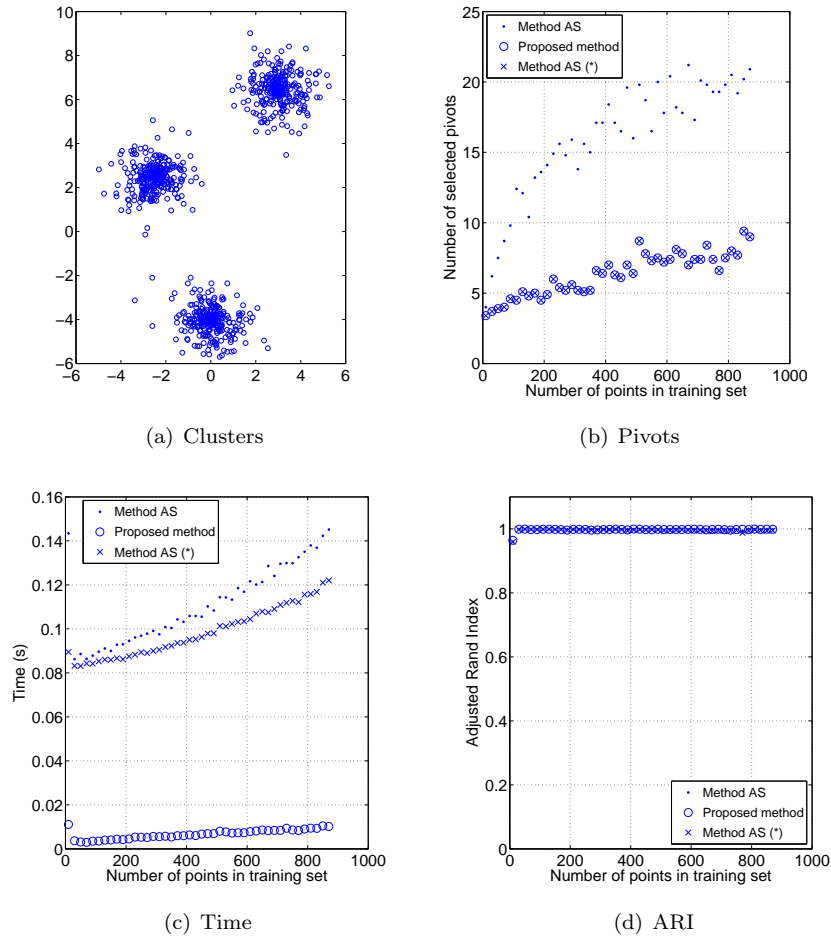
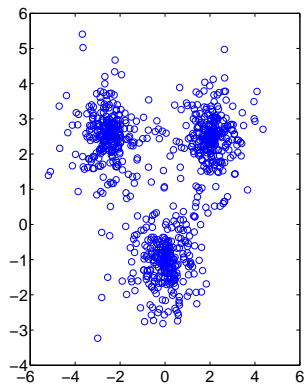


Figure 7: Three Gaussian clouds in 2D ( $N = 900$ ,  $\sigma = 0.8$ ,  $\eta = 0.5$ ): (a) Clusters, (b) Number of selected pivots in training set, (c) Computation time, (d) Adjusted rand index.

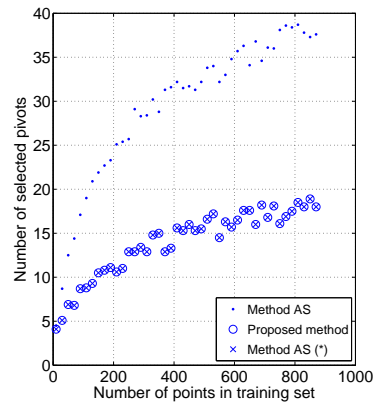
Figure 7(b)-(d) shows the number of selected pivots, the computation times in seconds and the adjusted rand index for increasing number of training points. Figure 7(b) shows that if more training points are selected the number of selected pivots increases. The increase is stronger for method AS than for the proposed method. Maybe  $\eta$  should be varied when  $N_{train}$  is increasing. In Figure 7(c) you can see that if the number of pivots increases also the computation times increase. Figure 7(d) shows the adjusted rand index. As you can see the proposed method gives directly a correct clustering and the method AS needs one step more to obtain a correct clustering.

In Figure 8(a), the three clouds are not well defined, they are visually distinctable but there are several points which are hard to assign to a specific cluster. Figure 8(b) shows that if more training points are selected more pivots will be selected. This gives also an increase in the computation times (Figure 8(c)). Figure 8(d) shows the ARI, the ARI for method AS are slightly better than for the proposed method. But probably this is because the method AS selects more pivots than the proposed method. If we compare the proposed method with method AS (\*), we see that in most cases the proposed method obtains a slightly better adjusted rand index.

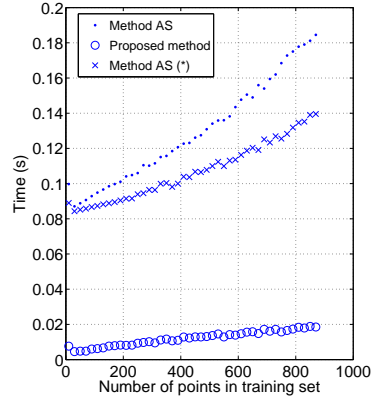
- $k$  Gaussian clouds in 2D with  $k = 2, \dots, 10$ : In this experiment the effect of an increasing number of clusters for a fixed number of training data is investigated. The dataset



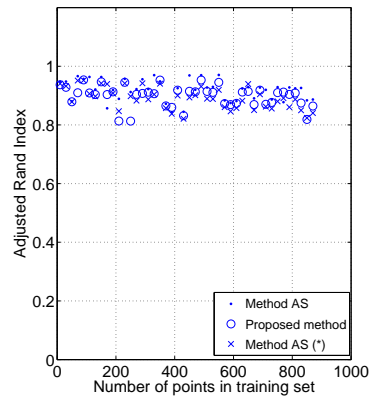
(a) Clusters



(b) Pivots



(c) Time



(d) ARI

Figure 8: Three Gaussian clouds in  $2D$  ( $N = 900$ ,  $\sigma = 0.5$ ,  $\eta = 0.5$ ): (a) Clusters, (b) Number of selected pivots in training set, (c) Computation time, (d) Adjusted rand index.

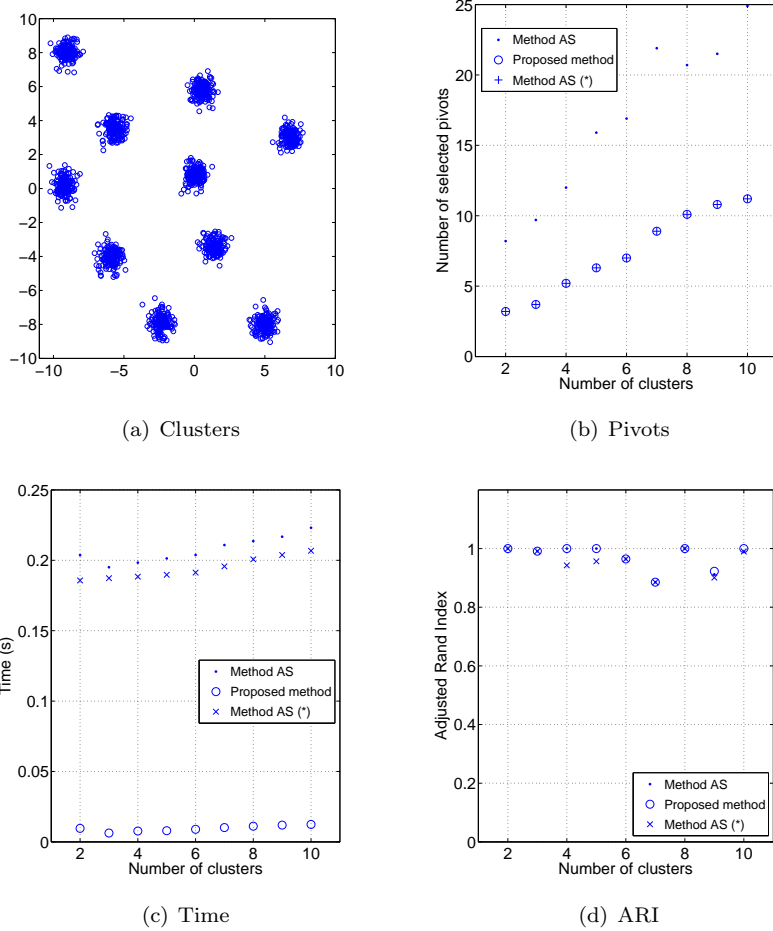


Figure 9:  $k$  clusters with  $k = 2, \dots, 10$  in  $2D$  space with varying number of training points ( $N = 2000$ ,  $\sigma = 0.5$ ,  $\eta = 0.5$ ): (a) Ten clusters (b) Number of selected pivots in training set, (c) Computation time, (d) Adjusted rand index.

contains  $N = 2000$  data points, one fifth will be used for training and the remaining points for testing. The RBF parameter  $\sigma$  is fixed to  $\sigma = 0.5$ , and  $\eta = 0.5$ .

Figure 9(a) shows the data points for ten clusters. Figure 9(b)-(d), shows the number of selected pivots, the computation time and the Adjusted Rand Index for an increasing number of clusters. The proposed method selects less pivots than method AS. The adjusted rand index indicates that the proposed method is comparable with method AS, and performs slightly better than method AS (\*) for an increasing number of clusters.

- Three concentric rings in a  $2D$  space: In this experiment, a nonlinear problem is investigated where the data points have few members and the rings have a multiscale nature. In Figure 10 the concentric rings are shown. Table 2 shows the results for an optimal and a non-optimal  $\sigma$ : 0.1 and 0.2 respectively. The data set consists of  $N = 1400$  points, 600 points are used for training ( $N_{train}$ ) and the other 800 for testing ( $N_{test}$ ). The methods have a similar behavior, because almost the same amount of pivots are selected. The proposed method gives a slightly better result.

	Optimal $\sigma = 0.1$			Non-optimal $\sigma = 0.2$		
	Pivots	Time (s)	ARI	Pivots	Time (s)	ARI
Method AS	93	0.5203	0.8526	45	0.2109	0.4778
Proposed method	87	0.2242	0.8693	42	0.0429	0.4852
Method AS (*)	87	0.4698	0.8547	42	0.2036	0.4828

Table 2: Three concentric rings in a  $2D$  space: Results for an optimal  $\sigma = 0.1$  and a non-optimal  $\sigma = 0.2$ , ( $N = 1400$ ,  $\eta = 0.65$ ).

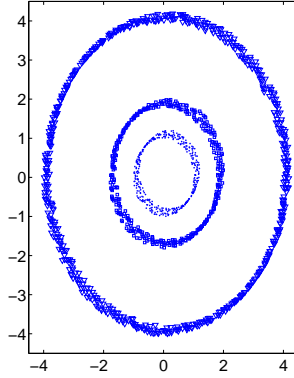


Figure 10: Three concentric rings in  $2D$  space for an optimal  $\sigma = 0.1$  and a non-optimal  $\sigma = 0.2$ , ( $N = 1400$ ,  $\eta = 0.65$ ).

## 5 Conclusion

A sparse spectral clustering method is presented which is based on linear algebra techniques. In the algorithm, the incomplete Cholesky decomposition, the LQ factorization with row pivoting, and the singular value decomposition are applied to obtain the cluster assignment efficiently. In fact, the data set is represented with only a sparse set of pivots, and based on this information the indicator vectors are derived. To acquire this, an adapted stopping criterion for the incomplete Cholesky decomposition is proposed such that no extra parameter is necessary in the algorithm. The proposed method is also extended to out-of-sample points. In the numerical simulations it is shown that the method presented achieves good results compared to method AS [4], especially when looking at the computational complexity.

## References

- [1] C. Alzate and J. Suykens. A weighted kernel PCA formulation with out-of-sample extensions for spectral clustering methods. In *Proceedings of the 2006 International Joint Conference on Neural Networks*, pages 138–144, 2006.
- [2] C. Alzate and J. Suykens. ICA through an LS-SVM based kernel CCA measure for independence. In *Proceedings of the 2007 International Joint Conference on Neural Networks*, pages 2920–2925, 2007.
- [3] C. Alzate and J. Suykens. A regularized kernel CCA contrast function for ICA. *Neural Networks*, pages 170–181, 2008.
- [4] C. Alzate and J. Suykens. Sparse kernel models for spectral clustering using the incomplete Cholesky decomposition. In *Proceedings of the 2008 International Joint Conference on Neural Networks*, pages 3555–3562, June 2008.
- [5] Francis R. Bach and Michael I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.
- [6] G. H. Ball and D. J. Hall. ISODATA: A novel method of data analysis and pattern recognition. Technical report, Stanford Research Institute, Menlo Park, CA, 1965.
- [7] G. H. Ball and D. J. Hall. A clustering technique for summarizing multi-variate data. *Behavioral Science*, 12:153–156, 1967.
- [8] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, J. Le Roux, and M. Ouimet. Out-of-sample extensions for LLE, isomap, MDS, eigenmaps, and spectral clustering. *Advances in Neural Information Processing Systems*, 2004.
- [9] F. Chung. *Spectral graph theory*. American Mathematical Society, 1997.
- [10] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley, New York, 2001.
- [11] L. Finesso and K. Scheinberg. Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264, 2001.
- [12] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, USA, third edition, 1996.
- [13] A. Gretton, R. Herbrich, A. Smola, O. Bousquet, and B. Schölkopf. Kernel methods for measuring independence. *Journal of Machine Learning Research*, 6:2075–2129, 2005.
- [14] L. Hagen and A. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design*, 11(9):1074–1085, 1992.
- [15] D. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*. MIT Press, Cambridge, MA, 2001.
- [16] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, pages 193–218, 1985.
- [17] B. Mohar. The Laplacian spectrum of graphs. In *Graph Theory, Combinatorics, and Applications*, volume 2, pages 871–898, 1991.
- [18] B. Mohar. Some applications of Laplace eigenvalues of graphs. In G. Hahn and G. Sabidussi, editors, *Graph Symmetry: Algebraic Methods and Applications*, volume 497, pages 225–275, Dordrecht, 1997. Kluwer,.

- [19] Andrew Y. Ng, Michael I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 14, 2002.
- [20] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [21] M. Stoer and F. Wagner. A simple min-cut algorithm. *Journal Association for Computing Machinery*, 44(4):585–591, 1997.
- [22] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, San Diego, CA, 2003.
- [23] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, 2007.
- [24] H. Zha, C. Ding, M. Gu, X. He, and H. Simon. Spectral relaxation for k-means clustering. *Advances in Neural Information Processing Systems*, pages 1057–1064, 2002.