Pattern Recognition Letters 84 (2016) 78–84

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Efficient evolutionary spectral clustering☆

Rocco Langone[a,*], Marc Van Barel[b], Johan A.K. Suykens[a]

[a] KU Leuven, ESAT-STADIUS, Kasteelpark Arenberg 10, Leuven B-3001, Belgium
[b] KU Leuven, Department of Computer Science, Celestijnenlaan 200A, Leuven B-3001, Belgium

ARTICLE INFO

Article history:
Received 24 February 2016
Available online 26 August 2016

Keywords:
Evolutionary spectral clustering
Incomplete Cholesky decomposition
Linear complexity
Temporal smoothness

ABSTRACT

Evolutionary spectral clustering (ESC) represents a state-of-the-art algorithm for grouping objects evolving over time. It typically outperforms traditional static clustering by producing clustering results that can adapt to data drifts while being robust to short-term noise. A major drawback of ESC is given by its cubic complexity, e.g. $O(N^3)$, and high memory demand, namely $O(N^2)$, that make it unfeasible to handle datasets characterized by a large number N of patterns. In this paper, we propose a solution to this issue by presenting the efficient evolutionary spectral clustering algorithm (E^2SC). First we introduce the notion of a *smoothed graph Laplacian*, then we exploit the incomplete Cholesky decomposition (ICD) to construct an approximation of the this smoothed Laplacian and reduce the size of the related eigenvalue problem from N to m, with $m \ll N$. Furthermore, in contrast to the standard ICD algorithm, a stopping criterion based on the convergence of the cluster assignments after the selection of each pivot is used, which is effective also when there is not a fast decay of the Laplacian spectrum. Overall, the proposed approach scales linearly with respect to the number of input datapoints N and has low memory requirements because only matrices of size $N \times m$ and $m \times m$ are constructed.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Many application scenarios involve clustering objects whose characteristics change over time, due to both a long-term trend and a short-term noisy variation. For example, in traffic jam predictions, where cars equipped with GPS sensors and wireless connections are to be clustered, the coordinate of each car may follow a certain path in the long-term but its estimated coordinate at a given time may vary due to instrumental errors. In similar situations, when the goal is to obtain a clustering result at each time step, which can grasp the concept drift and be insensitive to noise, evolutionary clustering algorithms have been developed, such as [9–15] and many others. In this paper we focus on evolutionary spectral clustering (ESC).

Like spectral clustering [16–19], the ESC algorithm [10] is based on computing the eigenvectors of the Laplacian matrix, which reveals the underlying clustering structure. However, in case of ESC the Laplacian matrix comprises the affinity matrices at both the actual time-step t and the previous time point $t-1$, hence its name of evolutionary Laplacian matrix. This modification allows to obtain clusters that evolve smoothly over time and, as mentioned earlier, is more suitable for clustering evolving objects compared to static clustering. A major issue of the ESC approach is its computational and memory cost. If we denote by N the number of datapoints at a given time[1] instant t, solving the eigenvalue problem has complexity $O(N^3)$, and the $N \times N$ evolutionary Laplacian matrix does not fit into the main memory when N is large.

In this article we propose a solution to this scalability problem by means of the efficient evolutionary spectral clustering (E^2SC) algorithm. The proposed approach takes inspiration from [20] and [21,22], where the Incomplete Cholesky Decomposition (ICD) has been exploited to speed-up (static) spectral clustering and kernel spectral clustering, respectively. The basic ideas behind the E^2SC algorithm can be summarized as follows:

- define a smoothed Laplacian matrix which involve a weighted[2] sum of the current and the past similarity matrices
- solve efficiently the eigenvalue problem involving the smoothed Laplacian by means of the incomplete Cholesky decomposition
- use a stopping criterion based on the convergence of the cluster assignments after the selection of each pivot in place of the classical stopping condition based on the low-rank assumption.

☆ This paper has been recommended for acceptance by S. Wang.

* Corresponding author.
E-mail address: rocco.langone@esat.kuleuven.be (R. Langone).

[1] For the moment, for simplicity, we can assume that the number of nodes N does not vary over time.

[2] Although many choices are possible for the weights, we use an exponentially decaying factor to emphasize more recent history.

http://dx.doi.org/10.1016/j.patrec.2016.08.012
0167-8655/© 2016 Elsevier B.V. All rights reserved.

In fact, the standard stopping condition can be inappropriate if the spectrum of the Laplacian does not have a fast[3] decay [20].

This procedure is more efficient than the standard ESC approach because, although it involves both QR factorization and singular value decomposition, it allows to (i) avoid the construction of the full $N \times N$ affinity matrices (at times t and $t-1$) (ii) avoid computing the solution of a large eigenvalue problem of size $N \times N$. In fact, only an approximated eigenvalue problem of size $m \times m$ must be solved to compute the cluster memberships for the N input datapoints, where m indicates the number of selected pivots. We have observed that usually $m \ll N$ because the cluster assignments after the selection of each pivot tend to have a fast convergence. For instance, in case of the synthetic dataset described in Section 5, $m = 35$ and $N = 10^6$. Finally, in contrast to ESC, the lower computational cost attained by means of the proposed approach allows to consider more than one snapshot in the past in the definition of the temporal smoothness.

The rest of this paper is organized as follows. Section 2 briefly discusses a number of approaches that have been proposed for evolutionary clustering. Section 3 focuses on summarizing the evolutionary spectral clustering technique. In Section 4 the proposed algorithm, i.e. E^{2}SC, is introduced. Section 5 is devoted to presenting the experimental results and finally Section 6 concludes the article.

2. Related work

In the last few years since its first conceptualization in [9], the research in evolutionary clustering has received much attention. Although several approaches have been proposed, the majority of the algorithms does not scale to large problem sizes. Also, unlike the proposed approach, many methods cannot handle a changing number of clusters or datapoints over time, and do not provide a systematic way to choose the number of clusters at each time step. In [8] a probabilistic generative model for analyzing communities and their evolution in dynamic social networks is proposed, which solves the evolutionary clustering problem from a Bayesian perspective and assumes a fixed number of communities over time. An iterative Expectation-Maximization algorithm is used to fit a stochastic block model for generating communities and a probabilistic model based on the Dirichlet distribution for capturing the community evolution. The authors of [7] introduced a novel evolutionary clustering algorithm objective to analyze dynamic multiplex networks, i.e. networks consisting of heterogeneous types of nodes with various interactions occurring between them. The optimization problem is solved through an alternating optimization algorithm, which has an interesting interpretation in terms of iterative latent semantic analysis process but has a high computational cost. In [6] the evolutionary spectral clustering problem has been formulated as a multiobjective optimization problem, whose solution is obtained through a genetic algorithm. This makes the approach computationally expensive and unfeasible for studying networks containing several thousand or millions of nodes. A similar approach, but more general because it can deal with multiplex networks that evolve over time, has been recently introduced in [5]. In [4] a recommender system based on evolutionary clustering is introduced, where two phases are executed: (i) neighborhood computation, which involves clustering the user ratings matrix and computing the neighbor of a particular user or item (ii) prediction, which consists of estimating an unknown rating from the neighborhood that was previously calculated through evolutionary clustering. Within the evolutionary clustering approaches, the algorithms based on spectral clustering are the most closely related to the proposed method. In fact evolutionary spectral clustering, that will be discussed in detail in the next section, has been of inspiration for various other algorithms. The authors of [3] proposed a general framework for evolutionary clustering based on low-rank kernel matrix factorization. At every time step, first a low-rank approximation of the affinity matrix is computed. Next, the factorization in a kernel space yields the clustering. In [11] an evolutionary clustering framework that accurately tracks the time-varying proximities between objects followed by static clustering is presented. The method adaptively estimates the optimal smoothing parameter using shrinkage estimation, which assumes that the observed affinity matrices are a linear combination of true proximity matrices (which are viewed as unobserved states of a dynamic system) and zero-mean noise matrices. In [2] the evolutionary maximum margin clustering has been presented, which at each time step seeks a hyperplane that best separates the current data distribution in a predefined kernel space. By taking into account both the actual data partition cost and the margin change in terms of time, it produces a time-smoothed clustering result by solving a quadratic programming optimization problem. A formulation for evolutionary co-clustering based on the fused Lasso regularization has been proposed in [1], where the optimization problem involved is non-convex, non-smooth and non-separable. To compute the solution efficiently, a two-step procedure that optimizes the objective function iteratively through gradient descent has been devised.

3. Evolutionary spectral clustering

Given a set $\mathcal{G}$ of N nodes, solving the clustering problem means finding a partition $\{\mathcal{G}_1, \ldots, \mathcal{G}_k\}$ of the nodes in $\mathcal{G}$ such that $\mathcal{G} = \cup_{l=1}^{k} \mathcal{G}_l$ and $\mathcal{G}_p \cap \mathcal{G}_q \neq \emptyset$, for $1 \leq p, q \leq k$, $p \neq q$. Moreover, a clustering result (i.e. a partition) can be equivalently expressed by means of an $N \times k$ cluster indicator matrix Z, with $Z_{ij} = \frac{1}{\sqrt{|\mathcal{G}_j|}}$ if node $\mathbf{v}_i \subset \mathcal{G}_j$, where $|\mathcal{G}_j|$ denotes the number of nodes in partition $\mathcal{G}_j$. Spectral clustering allows to address this graph partitioning problem by minimizing the cut size [23], which is the number of edges running between the k connected components of the graph.

In [10] the evolutionary spectral clustering (ESC) algorithm was introduced, which incorporates temporal smoothness in the normalized cut (NC) problem to handle dynamic scenarios. In particular, in the ESC approach one seeks to optimize the cost function $J_{tot} = \eta J_{\text{snap}} + (1-\eta) J_{\text{temp}}$, where J_{snap} refers to the NC objective at time t and J_{temp} measures the cost of applying the partition found at time t to the snapshot at time $t-1$, penalizing then clustering results that disagree with the recent past. Mathematically, the evolutionary normalized cut is defined as:

$$\begin{aligned} &\min_{Z^t} \quad \eta J^t|_{Z^t} + (1-\eta) J^{t-1}|_{Z^t} \\ &\text{subject to} \quad {Z^t}^T Z^t = I^t. \end{aligned} \tag{1}$$

More explicitly, Eq. (1) can be rewritten as:

$$\begin{aligned} &\min_{Z^t} \quad k - \text{Tr}\left[{Z^t}^T (\eta D^{t^{-\frac{1}{2}}} W^t D^{t^{-\frac{1}{2}}}) + (1-\eta) D^{t-1^{-\frac{1}{2}}} W^{t-1} D^{t-1^{-\frac{1}{2}}}) Z^t \right] \\ &\text{subject to} \quad {Z^t}^T Z^t = I^t \end{aligned} \tag{2}$$

where:

[3] We have observed that the smoothed Laplacian is more likely to not have a fast decay spectrum compared to the standard Laplacian, probably because it incorporates the clustering structure at different times. However, we do not have a theoretical proof of this point.

- W^t indicates the similarity matrix[4] at time t
- $D^t = \text{diag}(\mathbf{d}^t)$, with $\mathbf{d}^t = [d_1^t, \ldots, d_N^t]^T$ and $d_i^t = \sum_{j=1}^{N} W_{ij}^t$, denotes the actual graph degree matrix
- Z^t is the current cluster indicator matrix
- I^t denotes the $N^t \times N^t$ identity matrix
- $0 \leq \eta \leq 1$ is the smoothness parameter and reflects the emphasis given by the user to the actual snapshot and the previous data matrix.

Since optimizing (2) is an NP-hard problem, an approximate solution can be obtained by allowing Z^t to take real values, i.e. $Z^t \in \mathbb{R}^{N^t \times k}$. Like in static spectral clustering, a solution of the relaxed ESC problem is a matrix Z^t whose columns are the eigenvectors associated with the top k eigenvalues of the evolutionary Laplacian matrix:

$$L^t \mathbf{z}_l^t = \lambda_l^t \mathbf{z}_l^t, l = 1, \ldots, k^t. \tag{3}$$

where $L^t = \eta D^{t^{-\frac{1}{2}}} W^t D^{t^{-\frac{1}{2}}} + (1-\eta) D^{t-1^{-\frac{1}{2}}} W^{t-1} D^{t-1^{-\frac{1}{2}}}$. Furthermore, after obtaining Z^t, the final clusters can be obtained by projecting data points into span(Z^t) and then applying the k-means algorithm. Notice that, in case $N^t \neq N^{t-1}$, a pre-processing step is needed to transform W^{t-1} such that it has the same size as W^t. In case $N^t > N^{t-1}$ (that is new objects are present at time t), we can add zero rows and columns in W^{t-1}, on the other hand if $N^t < N^{t-1}$, rows and columns which are not present in W^t can be removed from W^{t-1}.

4. Proposed algorithm

As we have pointed out earlier, the major issue regarding the ESC algorithm is represented by its lack of scalability, due to the cubic complexity of solving eigenvalue problem (3), and to the high memory requirements (i.e. $O(N^2)$) of storing and constructing[5] the similarity matrices W^t and W^{t-1}. In this section, we show how to tackle this problem by means of the efficient evolutionary spectral clustering (E^{2}SC) algorithm.

4.1. Reduced eigenvalue problem of a smoothed Laplacian

The incomplete Cholesky decomposition or ICD [24,25] allows to compute a low rank approximation of accuracy τ of an $N \times N$ matrix A such that $||A - CC^T|| < \tau$, with $C \in \mathbb{R}^{N \times m}$ and $m \ll N$. Basically, the ICD selects the rows and the columns of A, called pivots, such that the rank of the approximation is close to the rank of the original matrix and, as a result, this sparse set of data points is a good representation of the full data set.

In order to exploit the ICD to solve the scalability issue involving the ESC approach, by taking inspiration from [11] we define the following eigenvalue problem involving a smoothed Laplacian[6] matrix L_{sm}:

$$L_{sm}^t \mathbf{g}_l^t = \lambda_l^t \mathbf{g}_l^t, l = 1, \ldots, k^t, \tag{4}$$

where

- $L_{sm}^t = D_{sm}^{t^{-\frac{1}{2}}} W_{sm}^t D_{sm}^{t^{-\frac{1}{2}}}$
- $W_{sm}^t = \eta W^t + (1-\eta) W^{t-1} + \cdots + (1-\eta)^{t-1} W^1$
- $D_{sm}^t = \text{diag}(\mathbf{d}_{sm}^t)$, with $\mathbf{d}_{sm}^t = [d_{sm,1}^t, \ldots, d_{sm,N}^t]^T$ and $d_{sm,i}^t = \sum_{j=1}^{N} W_{sm,ij}^t$.

In order to reduce the size of eigenvalue problem (4), if we replace the similarity matrix W_{sm}^t with its ICD, we obtain that $L_{sm}^t \approx D_{sm}^{t^{-\frac{1}{2}}} CC^T D_{sm}^{t^{-\frac{1}{2}}}$. We can then replace $D_{sm}^{t^{-\frac{1}{2}}} C$ with its QR factorization and substitute R with its singular value decomposition. After some algebraic manipulation we get:

$$L_{sm}^t \approx Q U_R (\Sigma_R{}^2) U_R{}^T Q^T \tag{5}$$

with $Q \in \mathbb{R}^{N^t \times m^t}$, $R \in \mathbb{R}^{m^t \times m^t}$, $R = U_R \Sigma_R V_R^T$ and $U_R, \Sigma_R, V_R \in \mathbb{R}^{m^t \times m^t}$. Notice that now we have to solve an eigenvalue problem of size $m^t \times m^t$ involving matrix RR^T, which can be much smaller than the size $N^t \times N^t$ of the original problem (4). Moreover, the eigenvectors of problem (4) can be estimated as $\hat{\mathbf{g}}_l^t = Q U_{R,l}$ whose related eigenvalues are $\hat{\lambda}_l^t = \sigma_{R,l}^2$. Finally, the cluster assignment for the ith datapoint can be obtained from a pivoted LQ factorization [26] of the matrix $\hat{G}^t = [\hat{\mathbf{g}}_1^t, \ldots, \hat{\mathbf{g}}_{k^t}^t]$

$$j_i = \arg\max_{l=1,\ldots,k^t} (|\hat{Z}_{il}^t|) \tag{6}$$

where $\hat{Z}^t = PLQ_{\hat{G}^t}$, $P \in \mathbb{R}^{N^t \times N^t}$ is a permutation matrix, $L \in \mathbb{R}^{N^t \times k^t}$ is a lower triangular matrix and $Q \in \mathbb{R}^{k^t \times k^t}$ denotes a unitary matrix.

Before getting the cluster memberships as in Eq. (6), two issues must be addressed, namely when to terminate the ICD algorithm and how to select the number of clusters k^t at a given time-step t.

4.2. ICD stopping criterion

Regarding the first issue, in this article we do not use the standard stopping criterion based on the assumption that the Laplacian matrix has small rank. In fact, as discussed in [20], in some cases this criterion may not lead to small numerical error, because there is not always a fast decay of the eigenvalues. On the other hand, we only assume that the cluster assignments after the selection of each pivot tend to converge. Therefore, the ICD is stopped when the cluster assignments $\mathbf{j}^s$ at iteration s and $\mathbf{j}^{s-1}$ at iteration $s-1$ (with $\mathbf{j} = [j_1, \ldots, j_N]$) are equal up to a user-defined threshold THR_{stop}, as measured by the normalized mutual information [27] $\text{nmi}^s = \text{NMI}(\mathbf{j}^s, \mathbf{j}^{s-1})$. Moreover, in order to speed-up the procedure, we use the heuristics introduced in [20]. Basically, the convergence of the cluster assignments is checked only when the approximation of the similarity matrix is good enough, that is when $\frac{\min(\tilde{\mathbf{d}})}{\max(\tilde{\mathbf{d}})} > \text{THR}_{\text{deg}}$, where $\tilde{\mathbf{d}} = CC^T \mathbf{1}_{N^t}$. From our experience $\text{THR}_{\text{stop}} = \text{THR}_{\text{deg}} = 10^{-6}$ represents a good choice, which allows to not terminate the ICD algorithm too early (leading to poor clustering performance) but also not too late (which increases the computational complexity). An example of nmi^s as function of the number of selected pivots is shown at the center of Fig. 1 for the synthetic dataset that will be described later.

4.3. Choosing the number of clusters

Concerning the selection of the number of clusters, the eigengap heuristics [28] is utilized to choose the proper k^t, $\forall t$. More in detail, the differences between consecutive eigenvalues is computed, and the number of clusters is selected as the one corresponding to the maximum difference. Notice that, unlike (evolutionary) spectral clustering, we use the small $m \times m$ matrix RR^T to compute the eigengap heuristics, which is much more efficient than computing it on the $N \times N$ Laplacian. An illustration of this procedure is given at the top of Fig. 1.

After computing the cluster assignments for each time-stamp, a tracking procedure is used to match the clusters between consecutive time-steps. In particular, the Hungarian algorithm [11,29] is

[4] Commonly used similarities include the inner product of the feature vectors $W_{ij} = \mathbf{v}_i^T \mathbf{v}_j$, the Gaussian similarity $W_{ij} = \exp(-(\frac{||\mathbf{v}_i - \mathbf{v}_j||_2^2}{\sigma^2})$ and the affinity matrices of graphs.

[5] The construction of W^t and W^{t-1} is not necessary if we are directly given graph affinity matrices, that is when we aim at clustering network data rather than vector data.

[6] Notice that, in contrast to ESC, we can consider all the previous time steps before the actual time point t because of the low computational and memory requirements of the E^{2}SC algorithm.

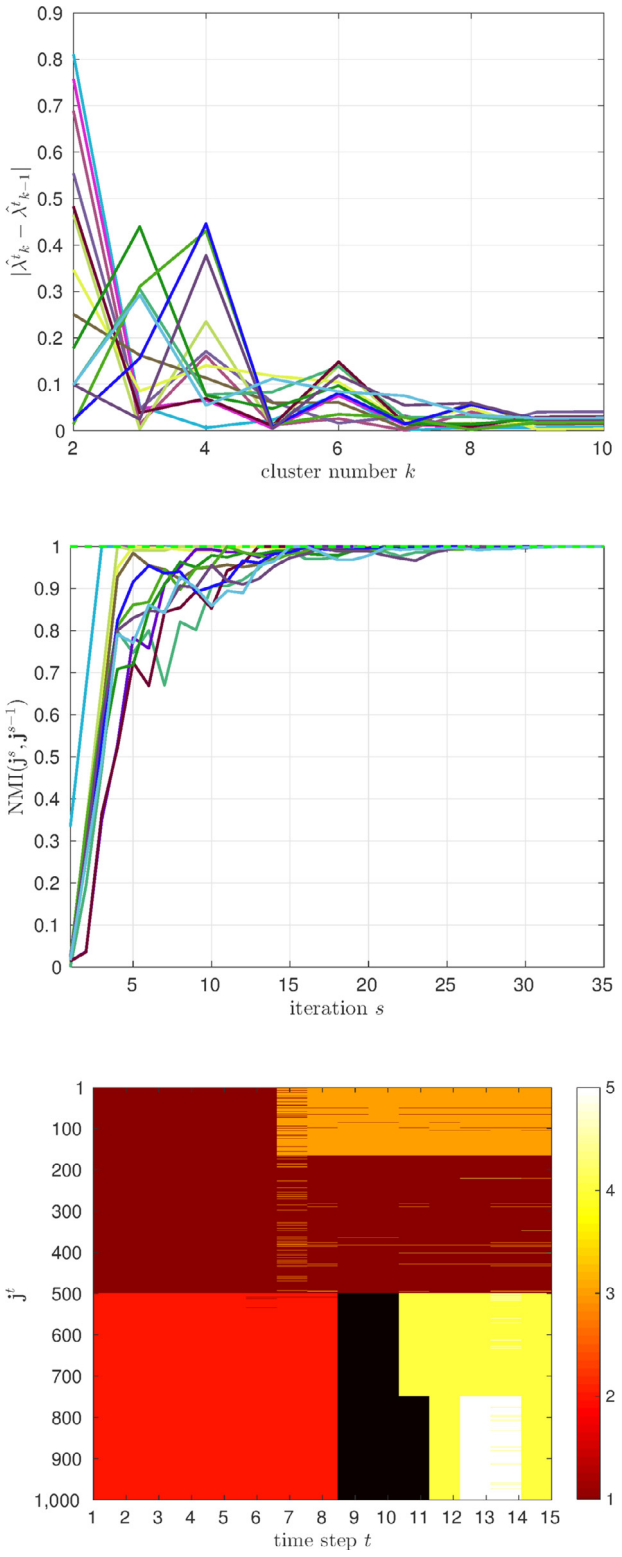


Fig. 1. Synthetic dataset. (Top) Selection of the number of clusters based on the eigengap heuristics, for the different time-steps t. (Center) Convergence of the cluster assignments during the incomplete Cholesky decomposition in terms of the normalized mutual information between consecutive partitions. Each line is related to a different time-stamp t. (Bottom) Heat map showing the evolution of the clusters as discovered by the E^{2}SC algorithm. The proposed approach is able to catch the drifting, splitting, death, appearance and merging events.

used to perform a one-to-one cluster matching based on a maximum weight matching between consecutive partitionings (with weights corresponding to the number of common objects between clusters). This allows to handle the arbitrariness of the assigned labels and therefore to follow the evolution of the clusters over time.

The complete clustering algorithm proposed in this paper, named E^{2}SC, is summarized in Algorithm 1. The related Matlab package can be downloaded from: http://www.esat.kuleuven.be/stadius/ADB/langone/EESClab.php.

Algorithm 1: E^{2}SC algorithm.

Data: Data matrices $\{X^t = [\mathbf{x}^t_1, \ldots, \mathbf{x}^t_{N^t}]\}_{t=1}^T$ with $\mathbf{x}^t_i \in \mathbb{R}^d$ or graph affinity matrices $\{W^t\}_{t=1}^T \in \mathbb{R}^{N^t \times N^t}$, thresholds THR_{stop} and THR_{deg}, maximum number of clusters to search for *maxk*.

Result: Selected number of clusters k^t, vector of cluster assignments $\mathbf{j}^t$.

for $t = 2 : T$ **do**

  /* *Pre-processing* */

  Transform matrices $X^1, \ldots, X^{t-1}$ in case of vector data or matrices $W^1, \ldots, W^{t-1}$ in case of network data such that $X^1, \ldots, X^{t-1} \in \mathbb{R}^{N^t \times d}$ or $W^1, \ldots, W^{t-1} \in \mathbb{R}^{N^t \times N^t}$

  /* *Settings* */

  $s = 1$

  $P = I_{N^t}$

  $C = 0_{N^t \times d_C}$ /* *Ex.* $d_C = 200$ */

  $W = \tilde{W}$ /* *Ex.* $\tilde{W} = \mathbf{0}_{N^t}$ */

  $\mathbf{h}_r = W_{rr}, r = 1, \ldots, N^t$

  $\mathbf{j}^1 = \mathbf{1}_{N^t}$

  $\eta = 0.9$. /* *default* */

  /* *Start ICD* */

  while $|nmi^s - 1| < THR_{stop}$ **do**

    Find new pivot element $r^\star = \arg\max_{r \in [s, N^t]} \mathbf{h}_r$

    Update permutation matrix P such that $P_{ss} = P_{r^\star r^\star} = 0$ and $P_{sr^\star} = P_{r^\star s} = 1$

    Permute elements s and $r^\star$ in $\tilde{W}$ as $\tilde{W}_{1:N^t,s} \leftrightarrow \tilde{W}_{1:N^t,r^\star}$ and $\tilde{W}_{s,1:N^t} \leftrightarrow \tilde{W}_{r^\star,1:N^t}$

    Update the element of C as $C_{s,1:s} = C_{r^\star,1:s}$

    Set $C_{ss} = \sqrt{\tilde{W}_{ss}}$

    Calculate s^{th} column of C as $C_{s+1:N^t,s} = \frac{1}{C_{ss}}(\tilde{W}_{s+1:N^t,s} - \sum_{r=1}^{s-1} C_{s+1:N^t,r} C_{sr})$

    Calculate $r_{\text{deg}} = \frac{\min(\tilde{\mathbf{d}})}{\max(\tilde{\mathbf{d}})}$

    if $r_{deg} > THR_{deg}$ **then**

      Compute QR decomposition of $\tilde{D}^{-\frac{1}{2}} C$

      Compute the singular value decomposition of R as $R = U \Sigma V^T$

      Obtain the approximated eigenvectors via $\hat{G} = QU_{R,1:\text{maxk}}$

      /* *Select current number of clusters* */

      Compute differences between consecutive eigenvalues and store them in vector $\mathbf{d}_{\tilde{\lambda}}$

      Set the current number of clusters: $k^s = \text{argmax}(\mathbf{d}_{\tilde{\lambda}}) + 1$

      /* *Check stopping condition* */

      Set $\hat{G} = \hat{G}_{1:k^s}$

      Compute LQ factorization with row pivoting as $D_{\hat{G}} \hat{G} = PLQ_{\hat{G}}$

      Put $\hat{Z} = P\hat{L}$, with $\hat{L} = [L_{11}^T L_{22}^T] L_{11}^{-1}$, being $L_{11} \in k^s \times k^s$ a lower triangular matrix

      Compute cluster assignment for point $\mathbf{x}_i$ according to Eq. (6), where $k = k^s$

      Store current assignments for the N datapoints in vector $\mathbf{j}^s$

      Compute $\text{nmi}^s = \text{NMI}(\mathbf{j}^{s-1}, \mathbf{j}^s)$

    end

    $\mathbf{h}_r = \mathbf{h}_r - C_{rs}^2, r = s+1, \ldots, N^t$

    $s = s + 1$

  end

  $k^t = k^s$

  $\mathbf{j}^t = \mathbf{j}^s$

end

/* *Post-processing* */

Match memberships between consecutive time-steps by means of the Hungarian algorithm.

Table 1
Pivots RCV15t dataset. Some of the pivots that have been selected via the ICD for weeks 1, 14, 28. A possible interpretation of the related cluster is provided (column Category). In general, the selected pivots seem good representative of the category they belong to.

Week ID	Cluster number	Pivots	Category
1	1	Kenyan	Religion
	2	Driver, Gerald	Sport
	3	Queen, Middlesbrough	Politics
	4	Colera, meningitis, smoke, virus, reaction, serum, therapy, capsule, compound ...	Health
	5	Kmph, time, state, storm, week, north, hurricane, coast, forecast, rain ...	Weather
14	1	Mouloudia, Ismail	Religion
	2	Game, 408	Sport
	3	Privacy	Politics
	4	Polio, euthanasia, inflammation, pharmacy, breakdown, Ronaldo, Benfica, Pedro, Klinsmann ...	Health + Sport (names)
	5	Cyberspace, unwieldy	Science (technology)
28	1	Vyborg	Weather (location)
	2	Socceraffican	Sport
	3	Turnout	Politics
	4	Parkinson, efficacy, potent, euthanasia, mammography, fruit, psychiatry, serum ...	Health
	5	Supercold	Weather

Table 2
Performance evaluation. The proposed method, the ESC and the AFFECT algorithms are contrasted in terms of average Davies Bouldin index (the lower the better) and the Calinski–Harabasz criterion (the higher the better) across the whole time period.

Dataset	Algorithm	DB	CH
Synthetic dataset	E^{2}SC	0.51	$\mathbf{3.77*10^3}$
	AFFECT [11]	0.57	$3.34*10^3$
	ESC [10]	**0.50**	$3.75*10^3$
RCV15t	E^{2}SC	**17.64**	**1.054**
	AFFECT [11]	32.19	1.031
	ESC [10]	18.40	1.036

5. Experimental results

In this section the experimental results are discussed. In particular, the behavior of the proposed approach is evaluated on the following two datasets:

- *Synthetic dataset:* This experiment has been performed to test the ability of the proposed method to discover the main events characterizing the clusters dynamics, that is drifting, splitting, death, appearance and merging of clusters. The dataset comprises $T = 15$ data matrices $X^1, \ldots, X^{15}$, with $X^t = [x_1, \ldots, x_{N^t}]$, $x_i^t \in \mathbb{R}^2$, and $N^1 = 10^6$. From time step 1 to time step 5 a mixture of 2 Gaussian distribution moves, from $t = 5$ to $t = 8$ one component of the mixture splits into two parts, from $t = 9$ to $t = 10$ the other component disappears, at $t = 11$ a new Gaussian distribution appears, at $t = 12$ another Gaussian distribution is created, from $t = 13$ to $t = 15$ two clusters move towards each other and merge.
- *RCV15t dataset:* It is a subset of the Reuters RCV1 corpus containing 10,116 news articles related to a 7 month period, formatted as TF-IDF documents [30]. Each article is annotated with a single ground truth topical label (health, religion, science, sport, weather), which are present across the entire time period of the corpus. From the TF-IDF representation we created a word-word graph W^t for each time-stamp t, where the weight of the edges in the network W^t is proportional to occurrence of 2 words together in all the news articles for that time step. There is a total of $T = 28$ time step graphs, each one representing one week period.

In case of the synthetic dataset, the Gaussian kernel, i.e. $W(\mathbf{x}_i, \mathbf{x}_r) = \exp(-\frac{||\mathbf{x}_i - \mathbf{x}_r||_2^2}{2\sigma^2})$, has been used to build the similarity matrices $W^1, \ldots, W^{15}$. Furthermore, the parameter σ has been

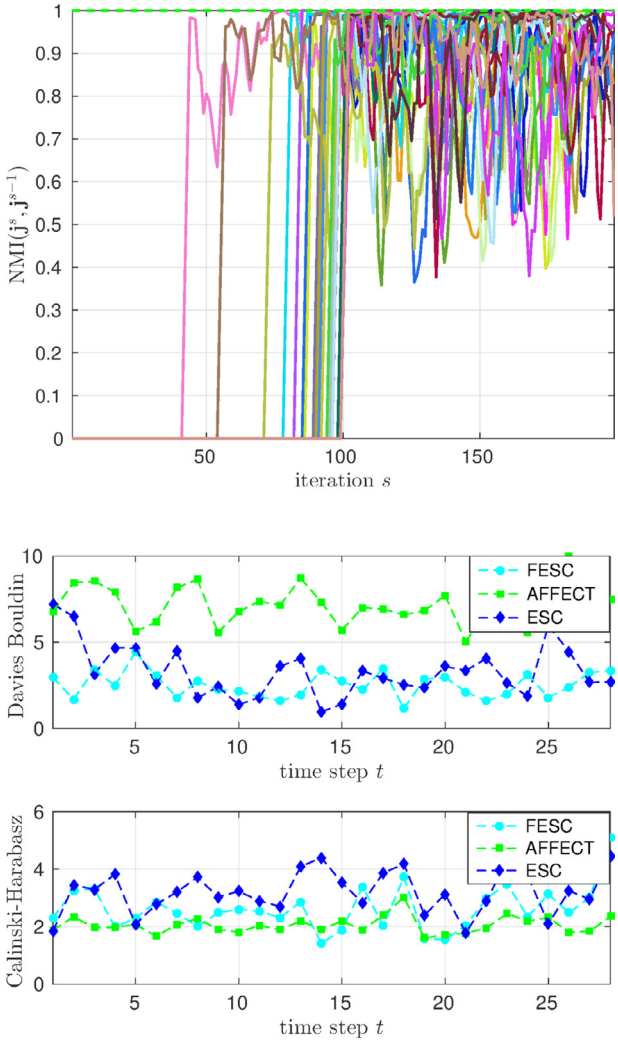


Fig. 2. RCV1 dataset. (Top) Convergence of the cluster assignments during the incomplete Cholesky decomposition as measured by the normalized mutual information between consecutive partitionings. Each line is associated to a different time-stamp t. In 8 out of 28 timesteps the algorithm stops for having reached the maximum number of iterations, without having converged yet (however, if we set $\text{THR}_{\text{stop}} = \text{THR}_{\text{deg}} = 10^{-5}$ in all the time-steps the algorithm converges). (Bottom) *(Top)* Trend of the Davies Bouldin index (the lower the better) *(Bottom)* Behavior of the Calinski–Harabasz index (the higher the better).

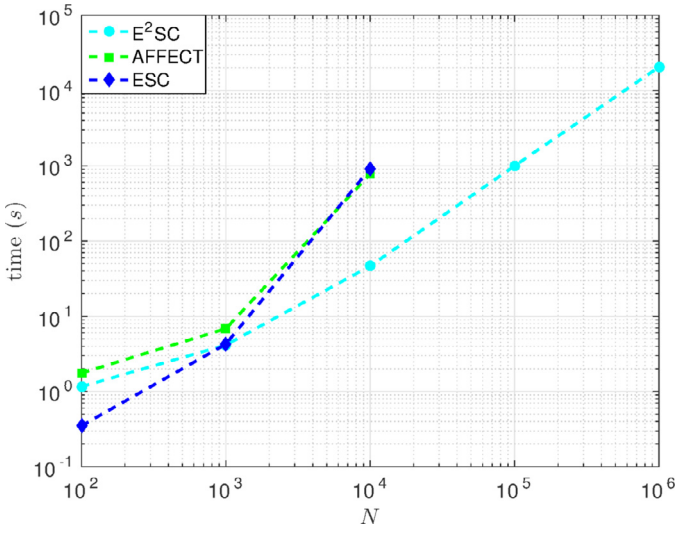


Fig. 3. Computational complexity. Scalability of the proposed algorithm, the ESC and the AFFECT methods with the number of datapoints N, where $N^1 = N^2 = \cdots = N^{15} = N$, and $N = \{10^2, 10^3, 10^4, 10^5, 10^6\}$. The Synthetic dataset has been used to perform this analysis, and the runtime refers to the total time needed to cluster the $T = 15$ data matrices of size $N \times 2$. The complexity of E^{2}SC is $O(N)$, which makes the method suitable for handling large-scale clustering problems. In contrast, the other algorithms have complexity $O(N^3)$. Furthermore, because of their high memory requirements, they cannot be used when $N > 10^4$.

chosen based on the Silverman[7] rule of thumb [31]. For the RCV15t dataset the cosine similarity matrix $W(\mathbf{x}_i, \mathbf{x}_r) = \frac{\mathbf{x}_i \cdot \mathbf{x}_r}{||\mathbf{x}_i|| \cdot \mathbf{x}_r}$ has been used, which is known to be a meaningful measure for text datasets. In both experiments, the forgetting factor and thresholds are set to their default values, i.e. $\eta = 0.9$, $\text{THR}_{\text{deg}} = \text{THR}_{\text{stop}} = 10^{-6}$.

The results given by the E^{2}SC algorithm are depicted at the bottom of Fig. 1 in case of the synthetic dataset. It can be noticed how the proposed approach is able to correctly model the main events characterizing the evolution of the Gaussian distributions over time. This is confirmed by the average values of the Davies Bouldin (DB) index [32] and the Calinski–Harabasz (CH) criterion [33] reported in Table 2, where a comparison with two state-of-the art evolutionary clustering algorithms, namely ESC and AFFECT [11] is also shown. Concerning the analysis of the RCV15t dataset, the performances of the proposed approach, the ESC and the AFFECT algorithms are illustrated at the bottom of Fig. 2 and reported in Table 2. As in case of the synthetic dataset, E^{2}SC is competitive with other state-of-the-art algorithms for what concerns the cluster quality.

Regarding the computational complexity, the proposed method outperforms ESC and AFFECT approaches. As it is illustrated in Fig. 3, the E^{2}SC runtime scales linearly with respect to the number of datapoints[8] N , i.e. E^{2}SC has complexity $O(N)$. On the contrary, the ESC and AFFECT methods have complexity $O(N^3)$, because the eigen-decomposition of the full $N \times N$ similarity matrices are involved at each time t.

6. Conclusions

In this paper we have presented an evolutionary spectral clustering method which has linear complexity. The new algorithm, named E^{2}SC, makes use of the incomplete Cholesky decomposition (ICD) to reduce the size of the eigenvalue problem involving a smoothed graph Laplacian. Moreover, unlike the standard ICD algorithm, a stopping criterion based on the convergence of the cluster assignments after the selection of each pivot is used, which is effective also when the Laplacian spectrum does not present a fast decay.

[7] The issue concerning the selection of the bandwidth parameter is outside of the scope of the paper.

[8] In this synthetic example for each time step from 1 to 15, the number of datapoints remains unchanged, that is $N^1 = N^2 = \cdots = N^{15} = N$, with N varying from 10^2 to 10^6.

Acknowledgment

EU: The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC AdG A-DATADRIVE-B (290923). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information. Research Council KU Leuven: CoE PFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants Flemish Government: FWO: Projects: G.0377.12 (Structured systems), G.088114N (Tensor based data similarity); PhD/Postdoc grant iMinds Medical Information Technologies SBO 2015 IWT: POM II SBO 100031 Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012–2017). The research was partially supported by the Research Council KU Leuven, Project OT/10/038 (Multi-parameter model order reduction and its applications), PF/10/002 Optimization in Engineering Centre (OPTEC), by the Fund for Scientific Research – Flanders (Belgium), G.0828.14N (Multivariate polynomial and rational interpolation and approximation), 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.

References

[1] R. Li, W. Zhang, Y. Zhao, Z. Zhu, S. Ji, Sparsity learning formulations for mining time-varying data, IEEE Trans. Knowl. Data Eng. 27 (5) (2015) 1411–1423.
[2] X. Fan, L. Zhu, L. Cao, X. Cui, Y.-S. Ong, Maximum margin clustering on evolutionary data, in: Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM), 2012, pp. 625–634.
[3] L. Wang, M. Rege, M. Dong, Y. Ding, Low-rank kernel matrix factorization for large-scale evolutionary clustering, IEEE Trans. Knowl. Data Eng. 24 (6) (2012) 1036–1050.
[4] C. Rana, S.K. Jain, An evolutionary clustering algorithm based on temporal features for dynamic recommender systems, Swarm Evolut. Comput. 14 (2014) 21–30.
[5] A. Amelio, C. Pizzuti, Evolutionary clustering for mining and tracking dynamic multilayer networks, Comput. Intell. (2016), doi:10.1111/coin.12074.
[6] F. Folino, C. Pizzuti, An evolutionary multiobjective approach for community discovery in dynamic networks, IEEE Trans. Knowl. Data Eng. 26 (8) (2014) 1838–1852.
[7] L. Tang, X. Wang, H. Liu, Community detection via heterogeneous interaction analysis, Data Min. Knowl. Discov. 25 (1) (2012) 1–33.
[8] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, B.L. Tseng, Analyzing communities and their evolutions in dynamic social networks, ACM Trans. Knowl. Discov. Data 3 (2) (2009) 1–31.
[9] D. Chakrabarti, R. Kumar, A. Tomkins, Evolutionary clustering, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006, pp. 554–560.
[10] Y. Chi, X. Song, D. Zhou, K. Hino, B.L. Tseng, Evolutionary spectral clustering by incorporating temporal smoothness, Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007, pp. 153–162.
[11] K. Xu, M. Kliger, A. Hero, Adaptive evolutionary clustering, Data Min. Knowl. Discov. 28 (2) (2015) 304–336.
[12] R. Langone, C. Alzate, J.A.K. Suykens, Kernel spectral clustering with memory effect, Phys. A: Stat. Mech. Appl. 392 (10) (2013) 2588–2606.
[13] R. Langone, R. Mall, J.A.K. Suykens, Clustering data over time using kernel spectral clustering with memory, Proceedings of Symposium Series on Computational Intelligence – Computational Intelligence and Data Mining (SSCI CIDM), 2014.
[14] J. Zhang, Y. Song, G. Chen, C. Zhang, On-line evolutionary exponential family mixture, in: Proceedings of the 21st International Joint Conference on Artifical Intelligence (IJCAI), 2009, pp. 1610–1615.
[15] L. Tang, H. Liu, J. Zhang, Z. Nazeri, Community evolution in dynamic multi-mode networks, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 677–685.

[16] F.R.K. Chung, Spectral Graph Theory, American Mathematical Society, 1997.
[17] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, in: Advances in Neural Information Processing Systems 14 (NIPS), 2002, pp. 849–856.
[18] U. von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416.
[19] H. Jia, S. Ding, X. Xu, R. Nie, The latest research progress on spectral clustering, Neural Comput. Appl. 24 (7–8) (2014) 1477–1486.
[20] K. Frederix, M.V. Barel, Sparse spectral clustering method based on the incomplete Cholesky decomposition, J. Comput. Appl. Math. 237 (1) (2013) 145–161.
[21] C. Alzate, J.A.K. Suykens, Sparse kernel models for spectral clustering using the incomplete Cholesky decomposition, in: Proceedings of the 2008 International Joint Conference on Neural Networks (IJCNN), 2008, pp. 3555–3562.
[22] M. Novak, C. Alzate, R. Langone, J.A.K. Suykens, in: Fast Kernel Spectral Clustering Based on Incomplete Cholesky Factorization for Large Scale Data Analysis. Internal Report 14-119, ESAT-SISTA, KU Leuven, Leuven, Belgium, 2014, pp. 1–44.
[23] M. Stoer, F. Wagner, A simple min-cut algorithm, J. ACM 44 (4) (1997) 585–591.
[24] G.H. Golub, C.F.V. Loan, Matrix Computations, The Johns Hopkins University Press, 1996.
[25] F.R. Bach, M.I. Jordan, Kernel independent component analysis, J. Mach. Learn. Res. 3 (2002) 1–48.
[26] H. Zha, C. Ding, M. Gu, X. He, H. Simon, Spectral relaxation for k-means clustering, in: Advances in Neural Information Processing Systems 14 (NIPS), 2002.
[27] A. Strehl, J. Ghosh, Cluster ensembles – a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (2002) 583–617.
[28] C. Davis, The rotation of eigenvectors by a perturbation, J. Math. Anal. Appl. 6 (2) (1963) 159–173.
[29] H.W. Kuhn, The hungarian method for the assignment problem, Naval Res. Logist. Q. 2 (1–2) (1955) 83–97.
[30] S. Robertson, Understanding inverse document frequency: on theoretical arguments for IDF, J. Doc. 60 (5) (2004) 503–520.
[31] B.W. Silverman, Density Estimation for Statistics and Data Analysis, Chapman & Hall, 1986.
[32] D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Trans. Pattern Anal. Mach. Intell. 1 (2) (1979) 224–227.
[33] T. Caliński, J. Harabasz, A dendrite method for cluster analysis, Commun. Stat. 3 (1) (1974) 1–27.