Landmark Selection for Spectral Clustering based on Weighted PageRank

D. Rafailidis*, E. Constantinou, Y. Manolopoulos

Department of Informatics, Aristotle University, 54124 Thessaloniki, Greece

Abstract

Spectral clustering methods have various real-world applications, such as face recognition, community detection, protein sequences clustering etc. Although spectral clustering methods can detect arbitrary shaped clusters, resulting thus in high clustering accuracy, the heavy computational cost limits their scalability. In this paper, we propose an accelerated spectral clustering method based on landmark selection. According to the Weighted PageRank algorithm, the most important nodes of the data affinity graph are selected as landmarks. Furthermore, the selected landmarks are provided to a landmark spectral clustering technique to achieve scalable and accurate clustering. In our experiments, by using two benchmark face and shape image data sets, we examine several landmark selection strategies for scalable spectral clustering that either ignore or consider the topological properties of the data in the affinity graph. Also, we show that the proposed method outperforms baseline and accelerated spectral clustering methods, in terms of computational cost and clustering accuracy, respectively. Finally, we provide future directions in spectral clustering.

Keywords: Spectral clustering, sparse coding, databases

*Corresponding author
Email addresses: draf@csd.auth.gr (D. Rafailidis), econst@csd.auth.gr (E. Constantinou), manolopo@csd.auth.gr (Y. Manolopoulos)

1. Introduction

Spectral Clustering (SC) comprises several goals by adapting to a wide range of non-Euclidean spaces and detecting non-convex patterns and linearly non-separable clusters. Compared to the baseline k-means algorithm, SC methods differ in how they find optimal partitions through objective functions. For example, given a set of n d-dimensional data points[1] $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \in \mathbb{R}^d$, the k-means algorithm tries to minimize the following objective function $\sum_{j=1}^{k} \sum_{i=1}^{n} ||\mathbf{x}_i^{(j)} - \mathbf{c}_j||^2$, where $||\mathbf{x}_i^{(j)} - \mathbf{c}_j||^2$ is a predefined distance, for example Euclidean, between $\mathbf{x}_i^{(j)}$ and the cluster center $\mathbf{c}_j$. Meanwhile, the key idea in SC is to achieve graph partitioning by performing eigendecomposition of the graph Laplacian matrix [1, 2, 4]. SC methods construct an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, represented by its $W \in \mathbb{R}^{n \times n}$ affinity matrix (or the respective adjacency), where $\mathcal{V}$ and $\mathcal{E}$ are the sets of vertices and edges, respectively. The goal is to find a k-way partitioning to minimize a particular objective. SC methods differ in how they define and construct the Laplacian matrix and thus which eigenvectors are selected to represent the graph partitioning. Luxburg's tutorial [2] includes examples of different Laplacians' constructions. For example, Ratio Cut [1] tries to minimize the total cost of the edges crossing the cluster boundaries, normalized by the size of the k clusters, to encourage balanced cluster sizes. Normalized Cut (NCut) [4] uses the same objective criterion as Ratio Cut, normalized by the total degree of each cluster, making thus the clusters to have similar degrees. The aforementioned baseline SC methods firstly calculate the degree matrix $D = \sum_j W_{ji} \in \mathbb{R}^{n \times n}$, a diagonal matrix whose entries are column (or row, since W is symmetric) sums of W. Then, SC methods use the top-k eigenvectors of the $L = D - W \in \mathbb{R}^{n \times n}$ Laplacian matrix corresponding to the k smallest eigenvalues as the low k-th dimensional representation of the data (k-th dimensional eigenspace). Finally, the k-means algorithm is applied

[1] Following standard notations, we use capital italic letters for matrices (e.g. A), lower-case bold letters for vectors (e.g. $\mathbf{a}$) and calligraphic fonts for sets (e.g. $\mathcal{A}$).

in the k-th dimensional eigenspace to generate the clusters.

SC methods have a number of real-world applications (Fig. 1), such as image segmentation [5], face recognition [6], feature fusion [7], speech recognition [8], 3D shape retrieval [9], document recognition [10], protein sequences clustering [11] and network communities detection [12].

(a) (b) (c)

Figure 1: SC methods have a number of real world applications such as: (a) face recognition, (b) network communities detection, and (c) protein sequences clustering.

1.1. Motivation

However, irrespective of the selected approach, there are two important factors for applying a SC method to a real world application: (a) the scalability of the method to large datasets, and (b) the clustering accuracy.

With respect to the first key factor, baseline SC methods require $O(n^3)$ worst time cost to calculate the eigendecomposition of the corresponding $L \in \mathbb{R}^{n \times n}$ Laplacian matrix. The cubic complexity prohibits the direct application of SC for generating clusters in large-scale data sets. Several accelerated methods have been proposed in the literature trying to reduce the initial problem size of n data points by selecting p ($\ll n$) samples/landmarks of the data set. Accelerated methods, in their approximations perform the eigendecomposition to a highly reduced $L \in \mathbb{R}^{p \times p}$ Laplacian matrix [13, 14, 15]. Consequently, accelerated methods significantly decrease the high complexity $O(n^3)$ of the baseline SC methods [1, 4]. Nevertheless, with respect the clustering accuracy, the accelerated SC methods depend on the sampling strategy that is used to perform the eigendecomposition of the highly reduced matrix.

1.2. Contribution and Outline

In [16], we presented an accelerated SC method using a landmark selection strategy based on the Weighted PageRank algorithm. In doing so, the most important nodes in the data affinity graph are selected as landmarks. By incorporating the selected landmarks into a landmark-based SC technique [15], we achieved scalable and accurate clustering. In this article, we extend our previous study by detailing the convergence issues of the Weighted PageRank algorithm, when selecting the landmarks. In addition, we explain the meaning of each parameter in the sparse coding strategy of the landmark-based SC method, and we formally define the proposed algorithm. We experimentally show the effectiveness of the proposed approach against different landmark selections strategies, and the baselines k-means and NCut methods, in terms of clustering accuracy and computational time. Finally, we discuss the main findings of our experimental results, and, we provide interesting future directions.

The rest of the paper is organized as follows. Section 2 summarizes related work. The proposed method is presented in Section 3 and our experimental results on two benchmark image data sets are discussed in Section 4. Finally, we draw the basic conclusions of our study in Section 5.

2. Related Work

Several accelerated SC methods have been proposed in the literature for overcoming the scalability issue. The key idea is to use sampling techniques and consequently to reduce the high complexity of SC in the L Laplacian matrix' eigendecomposition step. The k-means-based approximate SC (KASP) method [13], firstly performs k-means on the data set with a large number of p clusters and then, a baseline SC method is applied on the p cluster centers, with each data point being assigned to the cluster as its nearest center.

In [14], Fowlkes et al. applied the Nyström [17] method to accelerate the eigendecomposition step. Given a random set of p samples, a $W \in \mathbb{R}^{p \times p}$ affinity submatrix is computed and then, the calculated eigenvectors are used to

estimate an approximation of the eigenvectors of the original affinity matrix.

In [18], Kulis et al. followed a kernel approach for graph clustering in a unified framework for graph/vector-based approaches, where they showed that there is a connection between weighted kernel k-means [19] and graph clustering minimization criterion objectives. Establishing the aforementioned connection led to algorithms for locally optimizing graph clustering objectives and thus, improving the clustering accuracy of SC methods. However, weighted kernel k-means is prone to problems of poor local minima and sensitive to the initial centroids selection [20].

In [15], Chen and Cai proposed an accelerated SC method with landmark-based representation (LSC), outperforming the aforementioned accelerated SC methods by using a sparse coding technique. By selecting p landmarks, a $Z \in R^{n \times p}$ affinity submatrix was created, by expressing the pairwise similarities between the p landmarks and the n data points. By using a sparse coding technique, authors significantly reduced the preprocessing cost in $O(p^3 + p^2 n)$ time to compute the eigenvectors. Two variations of LSC are presented: (a) the LSC-R method, based on which the selections of the p landmarks is performed randomly, and (b) the LSC-K method, based on which a preprocessing step is added into LSC for performing k-means for the p landmarks selection. As it was experimentally shown, LSC-K outperformed LSC-R in terms of clustering accuracy. However, the topological properties of the nodes/data points in the affinity graph are ignored in both the LSC-R and LSC-K methods, when selecting the landmarks. In this study, we propose a landmark selection strategy based on the Weighted PageRank algorithm; as we will experimentally show the landmark selection strategy of LSC-K has limited clustering accuracy and adds a significant preprocessing cost into LSC, compared to the proposed landmark selection strategy based on Weighted PageRank (Section 4.3).

Meanwhile, there are several scalable solutions for the baseline k-means algorithm, SC, and PageRank based on massive parallelism through MapReduce

frameworks. For example, Apache Mahout[2] includes an implementation of k-means and SC. In Apache Mahout, the execution method parameter of the implementation specifies whether the sequential or MapReduce method will be used for k-means. The SC implementation uses Stochastic Singular Value Decomposition for dimensionality reduction and applies k-means to perform the final clustering. MLlib[3] is Apache Spark's scalable machine learning library, which implements the k-means clustering algorithm. Two implementations of k-means are provided by Spark: a parallelized variant of k-means, namely k-means++ [21], and streaming k-means to cluster data that arrive in a stream. In addition, Spark offers an SC implementation based on the Power Iteration Clustering algorithm [22]. GraphX[4] is Apache Spark's API for graphs and graph-parallel computation, which includes a MapReduce implementation of the PageRank algorithm. Nonetheless, all the aforementioned methods work on infrastructures with multiple computational nodes, which are beyond of this article's scope. The MapReduce implementation of the proposed method is left for future work.

3. Proposed Method

3.1. Mathematical Formulation

Given (a) a set of d-dimensional data points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \in \mathbb{R}^d$, denoted by a $X \in R^{d \times n}$ matrix, forming thus the data affinity graph $\mathcal{G}$ of nn closest neighbors, and (b) the p landmarks, the goal is to partition the n points into k discrete clusters, with the boundaries of the k clusters lying afar. According to [15] the goal is to design the $W \in \mathbb{R}^{n \times n}$ affinity matrix as $W = \widehat{Z}^T \widehat{Z}$, where $\widehat{Z} \in \mathbb{R}^{p \times n}$ is the p-th dimensional representation of the n data points, expressed as similarities/affinities of the n data points to the p landmarks. The $X \in R^{d \times n}$ matrix can be approximated as $X \approx UZ$, where the columns of

[2] `http://mahout.apache.org/`
[3] `http://spark.apache.org/mllib/`
[4] `http://spark.apache.org/graphx/`

matrix $U \in \mathbb{R}^{d \times p}$ are called basis vectors, i.e. the d-dimensional vectors of the p landmarks. Therefore, the goal is to minimize the approximation error $\min_{U,Z} \|X - UZ\|^2$, where $\|\cdot\|$ denotes the Frobenius norm of a matrix.

3.2. Landmark Selection based on Weighted PageRank

In the first step of the algorithm, we used the Weighted PageRank algorithm to select the p most important nodes in the affinity graph $\mathcal{G}$. According to [23], the Weighted PageRank algorithm assigns rank values to nodes according to their importance. This importance is assigned in terms of weight values to incoming and outgoing links, in our case links represent the respective content-based relationships, denoted by $w^{in}_{<a,b>}$ and $w^{out}_{<a,b>}$, respectively. $w^{in}_{<a,b>}$ is the weight of link $< a, b >$. It is calculated on the basis of number of incoming links to node b and the number of incoming links to all reference nodes of node a:

$$w^{in}_{<a,b>} = \frac{i_b}{\sum_{c \in \mathcal{R}_a} i_c} \tag{1}$$

where i_b is the number of incoming links of node b, i_c the number of incoming links of node c and $\mathcal{R}_a$ is the reference node set (content-based nearest neighborhood) of node a. Accordingly, $w^{out}_{<a,b>}$ is the weight of link $< a, b >$. It is calculated on the basis of the number of outgoing links of all the reference nodes of node a:

$$w^{out}_{<a,b>} = \frac{o_b}{\sum_{c \in \mathcal{R}_a} o_c} \tag{2}$$

where o_b is the number of outgoing links of node b and o_c is the number of outgoing links of node c.

The original PageRank algorithm presents non-convergence issues for some topologies [24, 25]. For example, consider that the two nodes a and b that point to each other but to no other node, and there is a third node c which points to one of them. This loop will accumulate rank, but never distribute any rank to the first two nodes, as there are no outgoing links. The loop will form a sort of trap, also known as *"rank sink"* [24, 25]. To handle this problem, we approximate the Weighted PageRank value $wpr(b)$ for a node $b \in \mathcal{V}$ via

an iterative process. The computation of $wpr(b)$ requires several iterations to adjust the approximation to the theoretical true value. In each iteration, the $wpr(b)$ value of each node $b \in \mathcal{V}$ is computed as follows:

$$wpr(b) = (1 - damp) + damp \sum_{a \in \mathcal{R}(b)} wpr(a) w^{in}_{<a,b>} w^{out}_{<a,b>} \tag{3}$$

where $damp$ is a dampening factor that is usually set to 0.85 [24, 25]. In each iteration the wpr values for all nodes are reduced based on Eq. (3). Following the implementation of Gephi[5], a widely used toolbox for graphs, the iterative process stops when the following convergence criterion is satisfied for all nodes $b \in \mathcal{V}$:

$$\frac{wpr(b)_{iter-1} - wpr(b)_{iter}}{wpr(b)_{iter}} \leq \epsilon \tag{4}$$

where the fraction is the normalized difference between the previous and the current iteration[6], and ϵ is a predefined convergence threshold. Finally, after the algorithm converges, when $\forall\ b \in \mathcal{V}$ Eq. (4) holds true, the p nodes with the highest wpr values are selected as landmarks.

3.3. Sparse Representation of the Affinity Submatrix

According to the Nadaraya-Watson kernel regression [26], for any data point $\mathbf{x}_i$ its $\widehat{\mathbf{x}}_i$ approximation is calculated as:

$$\widehat{\mathbf{x}}_i = \sum_{j=1}^{p} z_{ji} \mathbf{u}_j \tag{5}$$

Following the sparse coding strategy of [15], we assume that the ji-th element of matrix Z should be larger if $\mathbf{x}_i$ is closer to $\mathbf{u}_j$. To emphasize this assumption, we create the sparse representation of the Z affinity sparse matrix, by selecting the $r < p$ nearest landmarks, instead of the p landmarks ($U \in \mathbb{R}^{d \times p}$), such as, the z_{ji} value is set to 0, if $\mathbf{u}_j$ is not among the r nearest landmarks. In doing so, we promote sparsity in matrix Z. Let $U_{\langle i \rangle} \in \mathbb{R}^{d \times r}$ denote a submatrix of U,

[5] `http://gephi.github.io/`

[6] All wpr values are initialized by $1/n$.

composed of the r nearest landmarks of $\mathbf{x}_i$. Then each element z_{ji} is computed as:

$$z_{ji} = \frac{\Phi(\mathbf{x}_i, \mathbf{u}_j)}{\sum\limits_{j' \in U_{\langle i \rangle}} \Phi(\mathbf{x}_i, \mathbf{u}_{j'})}, \qquad i \in 1 \dots n \text{ and } j \in U_{\langle i \rangle} \tag{6}$$

where $\Phi(\cdot)$ is a kernel function with bandwidth σ. The Gaussian kernel $\Phi(\mathbf{x}_i, \mathbf{u}_j) = \exp(-\|\mathbf{x}_i - \mathbf{u}_j\|/2\sigma^2)$ is one of the most commonly used[7], where σ controls the local scale of each data point's neighborhood. According to the self-tuning strategy of the kernel bandwidth σ in [28], in our implementation, we set $\sigma^2 = \sigma_i * \sigma_j$, $\forall\ \Phi(\mathbf{x}_i, \mathbf{u}_j)$, where σ_i is the average distance of i's neighbors, that is, the non-zero elements in the sparse representation of $Z \in \mathbb{R}^{p \times n}$. For the W affinity matrix it holds that $W = \widehat{Z}^T \widehat{Z}$, where $\widehat{Z} = D^{-1/2} Z$ is the normalized Z by the $D = \sum_j Z_{ji}$ degree matrix.

3.4. Clusters' Generation

Let the Singular Value Decomposition (SVD) of $\widehat{Z} = A \Sigma B^T$, where $\Sigma = diag(\sigma_1, \dots, \sigma_p)$ and $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$ are the singular values of $\widehat{Z}$, $A = [\mathbf{a}_1, \dots, \mathbf{a}_p] \in \mathbb{R}^{p \times p}$ and $\mathbf{a}_i$'s are called left singular vectors, $B = [\mathbf{b}_1, \dots, \mathbf{b}_p] \in \mathbb{R}^{n \times p}$ and $\mathbf{b}_i$'s are called right singular eigenvectors. It is easy to verify that B are the eigenvectors of matrix $\widehat{Z}^T \widehat{Z}$ and A are the eigenvectors of matrix $\widehat{Z} \widehat{Z}^T$. Since the size of $\widehat{Z} \widehat{Z}^T$ is $p \times p$, we can compute A in $O(p^3)$ and then according to [15] B can be computed as

$$B = \Sigma^{-1} A^T \widehat{Z}. \tag{7}$$

The overall time is $O(p^3 + p^2 n)$, which is a significant reduction from $O(n^3)$ since $p \ll n$. To obtain the final k clusters the traditional k-means method is applied to the n right singular eigenvectors, $\mathbf{b}_i$'s, that is, the rows of B, thus requiring a $O(nk^2)$ cost. An overview of the proposed method is presented in Algorithm 1.

[7] Also, other types of kernel functions could be used, such as linear and polynomial, thoroughly examined in [27] for machine learning methods.

ALGORITHM 1: Spectral clustering based on Weighted PageRank

Input: (1) Affinity graph $\mathcal{G}=<\mathcal{V},\mathcal{E}>$, with $|\mathcal{V}|=n$; (2) number of landmarks p; (3) number of nearest landmarks r; (4) convergence threshold ϵ

Output: k clusters.

Initialize $wpr(b) \leftarrow 1/n$, $\forall\, b \in \mathcal{V}$;
repeat
  for *b=1:n* **do**
    Calculate $wpr(b)$ based on Eq. (3);
  end
until *Convergence based on Eq. (4) and threshold* ϵ;
Select p landmarks based on the highest wpr values;
Calculate matrix Z based on Eq. (6), the selected landmarks p, and the matrix U with the r nearest landmarks;
Generate the normalized matrix $\widehat{Z} = D^{-1/2}Z$, with $D = \sum_j Z_{ji}$;
Compute $A = [\mathbf{a}_1, \ldots, \mathbf{a}_p] \in \mathbb{R}^{p\times p}$, the first k eigenvectors of $\widehat{Z}\widehat{Z}^T$;
Compute $B \in \mathbb{R}^{n\times p}$ based on Eq. (7);
Generate the final k clusters, by applying k-means in B, with each row of B being a data point;
return the final clusters k;

4. Experimental Results

4.1. Data Sets

In our experiments we used two high-dimensional benchmark data sets[8], including a shape image data set of the Columbia University Image Library (COIL100 [29]) and a face data set of Carnegie Mellon (CMU-PIE [30]). COIL100 contains 100 objects, where the images of each object were taken five degrees apart as the object is rotated on a turnable view, generating thus for each object 72 shape images. The size of each image is 32 × 32 pixels, with 256 grey levels per pixel. Thus, each image is represented by a d=1024-dimensional vector.

[8]All data sets were downloaded in the .mat format, publicly available at `http://www.cad.zju.edu.cn/home/dengcai/Data/data.html`

Therefore, COIL100 consists of n=7,200 vectors of d=1,024 dimensions with k=100 clusters, where each cluster represents the shape images of each object. Additionally, CMU-PIE is a database of 41,365 face images of 68 people, each person under 13 different poses, 43 different illumination conditions and with 4 different expressions. We used the face evaluation data set of [31], which consists of n=11,554 vectors of d=1024 dimensions with 68 clusters, where each cluster represents the face images of each person.

4.2. Evaluation Protocol

In our experiments, the performance was measured in terms of: (a) clustering accuracy, (b) Normalized Mutual Information, and (c) preprocessing cost.

The clustering accuracy (*Acc*) [32] is defined as:

$$Acc = \frac{\sum_{i=1}^{n} \delta(c_i, map(c'_i))}{n} \tag{8}$$

where c_i is the true class label and c'_i is the cluster label of $\mathbf{x}_i$ obtained from the clustering algorithm, $\delta(\cdot)$ is the delta function and $map(\cdot)$ is the best mapping function. The $map(\cdot)$ function matches the true class labels and the best mapping is solved by using the Kuhn-Munkres algorithm [33]. The *Acc* values range from 0 to 1, where a larger *Acc* indicates a better performance.

Let $\mathcal{C}_{gnd}$ denote the set of clusters obtained from the ground truth and $\mathcal{C}_{alg}$ obtained from a given clustering algorithm. Their Mutual Information $MI(\mathcal{C}_{gnd}, \mathcal{C}_{alg})$ is defined as:

$$MI(\mathcal{C}_{gnd}, \mathcal{C}_{alg}) = \sum_{c_i \in \mathcal{C}_{gnd}, c'_j \in \mathcal{C}_{alg}} p(c_i, c'_j) \log \frac{p(c_i, c'_j)}{p(c_i)p(c'_j)} \tag{9}$$

where $p(c_i)$ and $p(c'_j)$ are the respective probabilities that an arbitrary sample of the data set belongs to the clusters c_i and c_j, respectively. $p(c_i, c'_j)$ is the joint probability that the sample belongs to c_i and c'_j. Then, the Normalized Mutual Information (NMI) [34] is defined as:

$$NMI(\mathcal{C}_{gnd}, \mathcal{C}_{alg}) = \frac{MI(\mathcal{C}_{gnd}, \mathcal{C}_{alg})}{\sqrt{H(\mathcal{C}_{gnd})H(\mathcal{C}_{alg})}} \tag{10}$$

where function $H(\mathcal{X}) = -\sum_{c_i \in \mathcal{X}} p(c_i) \log pc_i$ is the entropy of the $\mathcal{X}$ clusters. It is easy to check that $NMI(\mathcal{C}_{gnd}, \mathcal{C}_{alg})$ ranges from 0 to 1, with NMI=1 if the two sets of clusters are identical and NMI=0 if the two data sets are independent. In our experimental results, *Acc* and NMI are expressed as a percentage.

All experiments were performed on a Windows 7 PC with Intel core i5-2430M CPU @ 2.4 GHz with 8GB RAM, using Matlab 2010a.

4.3. Comparison against Landmark Selection Strategies

In this first set of experiments, we evaluate the following landmark selection strategies:

- **Random**: is the LSC-R method of [15], with the nodes being randomly selected as landmarks, irrespective of their topological features in the affinity graph.
- **k-means**: is the LSC-K method of [15], where the k-means algorithm is used to determine the landmarks. For p landmarks, k=p centroids of the clusters are selected as landmarks.
- **Degree centrality**: is defined as the number of links incident upon a node. Nodes with the highest degree centrality are selected as landmarks.
- **Betweenness centrality**: is a measure of a node's centrality in a graph [35]. It is equal to the number of shortest paths from all vertices to all others that pass through that node, expressing how each node controls the flow within the graph. Nodes with the highest betweenness centrality are selected as landmarks.
- **PageRank**: is the widely known Google's PageRank measure [36], which estimates the importance of a node in the graph. To consider the weights of the links we used the Weighted PageRank algorithm of [23]. Nodes with the highest PageRank values are selected as landmarks, as described in Section 3.2.

All centrality measures were extracted in the Gephi toolbox. With respect to the computational cost, degree centrality has the less complexity, i.e. 0.01 and 0.02 seconds for the COIL100 and CMU-PIE data sets, whereas betweenness centrality requires 52.6 and 222.47 seconds, respectively. The computational cost of betweenness centrality is high, as it requires the calculation of all-to-all paths in the graph. In the Weighted PageRank algorithm, we set the Gephi's default convergence threshold ϵ=10^{-3}. According to Eq. (4), for the n nodes in each iteration, we have n normalized differences between the previous and current iterations. In Figure 2, we present the convergence rate of Weighted PageRank. On the y-axis, we report the maximum normalized difference of Eq. (4) for each iteration. The Weighted PageRank algorithm converges in 12 and 14 iterations and needs 0.83 and 1.56 seconds for the COIL100 and CMU-PIE data sets, respectively. The landmark selection strategy using the centroids of the k-means clustering (LSC-K) depends on the number of landmarks. Therefore, for $p = 5, 10, 15, 20\%$ landmarks, expressed as a percentage of the data set size n, k-means requires 1.39, 1.61, 3.09 and 3.46 seconds for COIL100 and 3.15, 6.37, 8.78 and 12.06 seconds for the CMU-PIE data set, making the computational cost of the landmark selection strategy of LSC-K high, compared to the proposed Weighted PageRank selection algorithm.

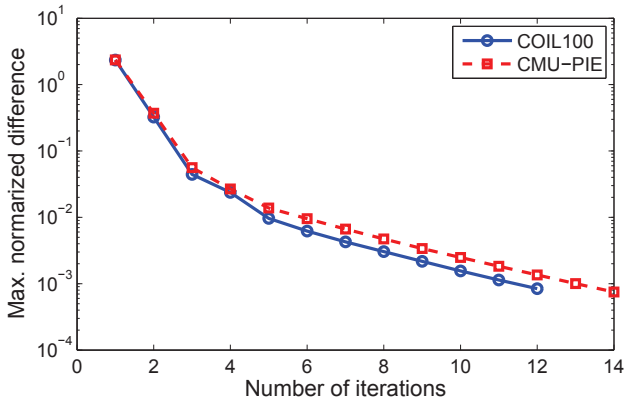

Figure 2: Weighted PageRank convergence rate: maximum normalized difference from the previous iteration versus the number of iterations (convergence threshold ϵ=10^{-3}).

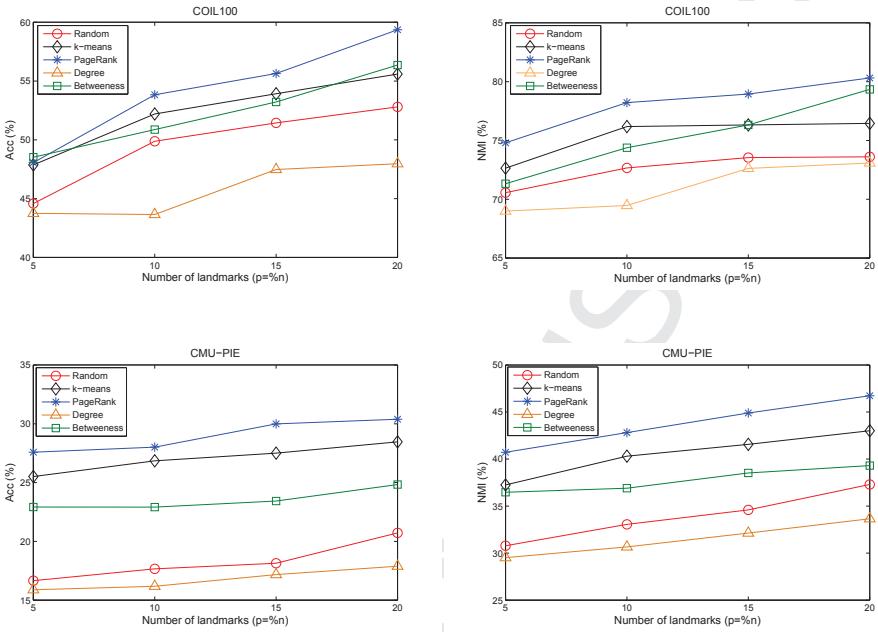

Figure 3: Landmark selection strategies for Landmark Spectral Clustering (LSC).

For the different landmark selection strategies, we varied the number of r nearest landmarks (Section 3.3) from 2 to 10, where we concluded to 6 and 4 for COIL100 and CMU-PIE, respectively, with the exceptional case of r=3 for random (LSC-R) and k-means (LSC-K) in CMU-PIE. In Fig. 3, we present the experimental results of the Landmark Spectral Clustering (LSC) method (Sections 3.3 and 3.4), for the different landmark selection strategies, where Weighted PageRank clearly outperforms the competitive strategies. This happens because Weighted PageRank identifies the most important nodes of the affinity graph, improving thus the clustering accuracy of LSC. The landmark selection strategy based on the Degree centrality reduces the clustering accuracy, making LSC prone to problems of poor local minima. Therefore, the proposed landmark selection strategy of Weighted PageRank achieves high clustering accuracy, by adding a low preprocessing cost to LSC, in contrast to the rest of landmark selection strategies.

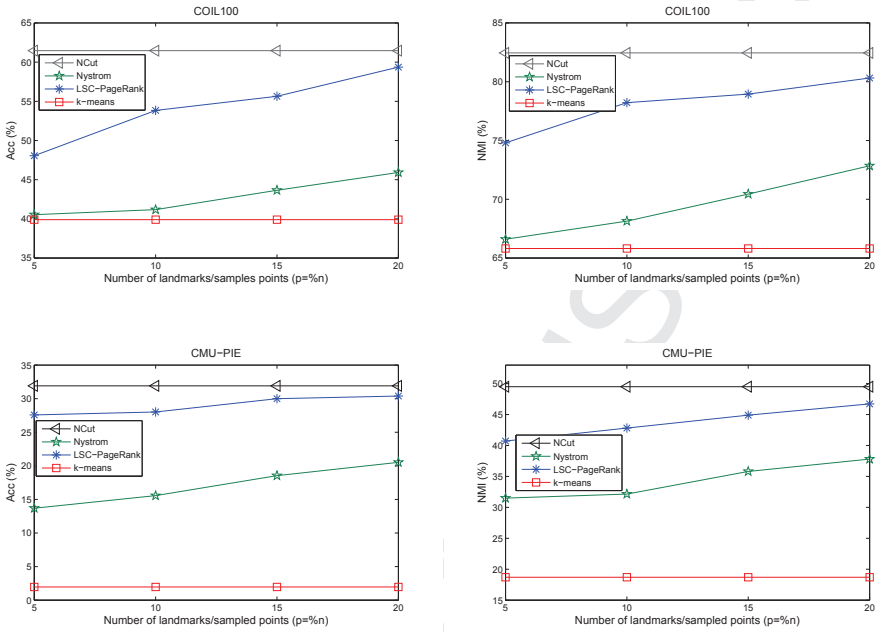

Figure 4: Comparison of the proposed LSC-PageRank method against: (a) the baseline NCut, (b) the accelerated Nyström spectral clustering methods, and (c) the baseline k-means algorithm.

4.4. Comparison against Baselines

The proposed LSC-PageRank method is compared against the baseline NCut method [9] [4], the accelerated Nyström spectral clustering method with orthogonalization [10] [14], and the k-means algorithm over the original data. The reason for using orthogonalization in the Nyström spectral clustering method is that increases the clustering accuracy, as it was experimentally shown in [14]. According to the experimental results of Fig. 4, LSC-PageRank achieves high clustering accuracy, comparable to the clustering accuracy of the baseline NCut method (in the case of p=20% landmarks), while significantly outperforming the accelerated Nyström method for all number of landmarks/sampled points variations. The k-means algorithm has limited clustering accuracy, by perform-

[9] `http://vision.ucsd.edu/~sagarwal/clustering.html`
[10] `alumni.cs.ucsb.edu/~wychen/sc.html`

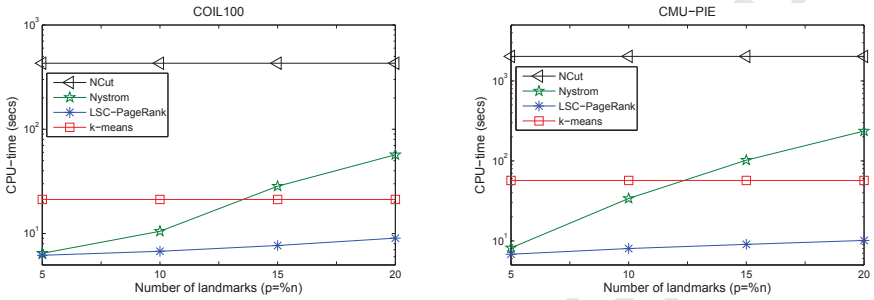

Figure 5: CPU-time (sec) of the baseline NCut, the proposed LSC-PageRank the accelerated Nyström spectral clustering methods and k-means.

ing clustering in the original d-dimensional space, and not in the k-dimensional eigenspace as LSC-PageRank does.

In Figure 5, we present the computational cost of each examined method, by varying the number of landmarks p. The baseline NCut method has high preprocessing cost $O(n^3)$, due to the eigendecomposition of the Laplacian matrix $L \in \mathbb{R}^{n \times n}$, while the accelerated Nyström method and the proposed LSC-PageRank method have low computational overheads $O(p^3 + pn)$ and $O(p^3 + p^2 n)$, by performing the eigendecomposition to highly reduced matrices. All SC methods have a common $O(nk^2)$ cost to generate the final clusters in the k-th dimensional eigenspace. LSC-PageRank has lower computational cost than the Nyström method for the following reasons (i) the Weighted PageRanks adds a low cost to the original LSC method; (ii) the Nyström method has higher computation cost than the LSC method [15], with LSC exploiting a sparse coding technique. In addition, the k-means algorithm has higher computational cost than the accelerated methods, especially for a small number of landmarks, as k-means requires a $O(nkd)$ cost to perform clustering in the d-dimensional original space.

Summarizing, the proposed landmark selection strategy of Weighted PageRank, namely LSC-PageRank, outperforms all the competitive landmark selection strategies by achieving the highest clustering accuracy and having one of the lowest preprocessing costs in the LSC-based strategy. This happens because the

landmark selection strategy based on Weighted PageRank identifies the most important nodes in the data affinity graph in low computational time. Both the clustering accuracy and the low preprocessing cost are important factors in accelerated SC methods. In addition, the proposed LSC-PageRank method significantly outperforms the baseline NCut and the accelerated Nyström SC method, in terms of preprocessing cost and clustering accuracy, respectively, while LSC-PageRank achieves comparable clustering accuracy with NCut, especially for a larger number of landmarks.

5. Conclusion

In this paper we presented an efficient method for accurate and scalable spectral clustering. In particular, we proposed a landmark selection strategy based on the Weighted PageRank algorithm for selecting the most representative nodes in the data affinity graph. As we experimentally showed, the proposed method outperforms state-of-the-art landmark selection strategies that either ignore or consider the topological properties of the nodes in the affinity graph. Finally, by following a landmark spectral clustering method we showed that the proposed method significantly outperforms competitive methods of baseline and accelerated spectral clustering in terms of preprocessing cost and clustering accuracy, respectively. For future work we will consider the following three main directions.

Semi-supervised SC: Several semi-supervised spectral clustering methods [20, 37, 38] have been proposed in the literature to improve the clustering accuracy, by adding must-link and cannot link constraints to the affinity graph. Nevertheless, irrespective of the final constructed affinity graph, where the constraints have been embedded to, the eigendecomposition of the respective Laplacian matrix $L \in \mathbb{R}^{n \times n}$ is still performed, preserving thus the high complexity of the baseline spectral clustering methods. However, the influence of must-link and cannot link constraints to the affinity graph must be further examined, since the most important nodes may vary, modifying thus the proposed

landmark strategy of Weighted PageRank.

Incremental SC on Evolving Graphs: In real-world applications continuously and efficiently updates are required, over the data sets evolution. Recently, several incremental strategies [39, 40, 41] have been proposed in the literature, able to handle not only insertion/deletion of data points but also similarity changes between existing points. In our future research we plan to examine the incremental strategy of the proposed method, by efficiently updating the constructed eigenspace, as well as including an incremental PageRank computation algorithm on evolving graphs [42].

Parallel SC: Finally, modern web databases require a significantly large preprocessing cost for spectral clustering in billions of data. For instance, several works [28, 43] introduced a parallel spectral clustering in distributed databases. Towards this aim, in our future work we plan to design the proposed method for distributed databases in the context of SC methods in Big Data [44].

References

[1] P. Chan, M. Schlag, J. Zien, Spectral k-way ratio-cut partitioning and clustering, IEEE Transactions on CAD-Integrated Circuit and Systems 13 (9) (1994) 1088–1096.

[2] U. Luxburg, A tutorial on spectral clustering, Statistics and Computing 17 (4) (2007) 395–416.

[3] A. Y. NG, M. I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, in: Proceedings in Advances in Neural Information Processing Systems (NIPS), 2002.

[4] J. Shi, J. Makil, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.

[5] F. Tung, A. Wong, D. Clausi, Enabling scalable spectral clustering for image segmentation, Pattern Recognition 43 (12) (2010) 4069–4076.

[6] H. Cevikalp, B. Triggs, Face recognition based on image sets., in: Proceedings 23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 2567–2573.

[7] H.-C. Huang, Y.-Y. Chuang, C. Chen, Affinity aggregation for spectral clustering, in: Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 773–780.

[8] K. Iso, Speaker clustering using vector quantization and spectral clustering, in: Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2010, pp. 4986–4989.

[9] A. Tatsuma, M. Aono, Multi-fourier spectra descriptor and augmentation with spectral clustering for 3d shape retrieval, Visual Computer 25 (8) (2009) 785–804.

[10] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.

[11] A. Paccanaro, C. Chennubhotla, J. Casbon, M. Saqi, Spectral clustering of protein sequences, in: Proceedings International Joint Conference on Neural Networks (IJCNN), 2003, pp. 3083–3088.

[12] J. Qiu, J. Peng, Y. Zhai, Network community detection based on spectral clustering, in: Proceedings International Conference on Machine Learning and Cybernetics (ICMLC), 2014, pp. 648–652.

[13] D. Yan, L. Huang, M. Jordan, Fast approximate spectral clustering, in: Proceedings 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2009, pp. 907–916.

[14] C. Fowlkes, S. Belongie, F. Chung, J. Malik, Spectral grouping using the nystrom method, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2) (2004) 214–225.

[15] X. Chen, D. Chai, Large-scale spectral clustering with landmark-based representation, in: Proceedings 25th AAAI Conference on Artificial Intelligence (AAAI), 2011, pp. 313–318.

[16] D. Rafailidis, E. Constantinou, Y. Manolopoulos, Scalable spectral clustering with weighted pagerank, in: Proceedings 4th International Conference on Model and Data Engineering (MEDI), 2014, pp. 289–300.

[17] E. Nystrom, Uber die praktische auflosung von integralgleichungen mit anwendungen auf randwertaufgaben, Acta Mathematica 54 (1930) 185–204.

[18] B. Kulis, S. Basu, I. Dhillon, R. Mooney, Semi-supervised graph clustering: a kernel approach, Journal of Machine Learning 74 (1) (2009) 1–22.

[19] I. Dhillon, Y. Guan, B. Kulis, Kernel k-means, spectral clustering and normalized cuts, in: Proceedings 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2004, pp. 551–556.

[20] W. Chen, G. Feng, Spectral clustering: a semi-supervised approach, Neurocomputing 77 (1) (2012) 229–242.

[21] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, S. Vassilvitskii, Scalable k-means++, PVLDB 5 (7) (2012) 622–633.

[22] F. Lin, W. W. Cohen, Power iteration clustering, in: Proceedings of the 27th International Conference on Machine Learning (ICML), 2010, pp. 655–662.

[23] W. Xing, A. Ghorbani, Weighted pagerank algorithm, in: Proceedings 2nd Annual Conference on Communication Networks and Services Research (CNSR), 2004, pp. 305–314.

[24] L. Page, S. Brin, R. Motwani, T. Winograd, The pagerank citation ranking: Bringing order to the web, in: Technical report, Stanford Digital Libraries SIDL-WP-1999-0120, 1999.

[25] C. Ridings, M. Shishigin, Pagerank uncovered, in: Technical report, 2002.

[26] W. Härdle, Applied non-parametric regression, Cambridge University Press, 1992.

[27] T. Hofmann, B. Schölkopf, A. J. Smola, Kernel methods in machine learning, The Annals of Statistics 36 (3) (2008) 1171–1220.

[28] W. Chen, Y. Song, H. Bai, C. Lin, E. Chang, Parallel spectral clustering in distributed systems, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (3) (2011) 568–586.

[29] S. A. Nene, S. K. Nayar, H. Murase, Columbia object image library, Tech. Rep. CUCS-005-96, Department of Computer Science, Columbia University (1996).

[30] T. Shim, S. Baker, The cmu pose, illumination and expression database, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (12) (2003) 1615–1617.

[31] D. Cai, X. HE, J. Han, Efficient kernel discriminant analysis via spectral regression., in: Proceedings 7th IEEE International Conference on Data Mining (ICDM), 2007, pp. 427–432.

[32] D. Cai, X. He, J. Han, S. Member, Document clustering using locality preserving indexing, IEEE Transactions on Knowledge and Data Engineering 17 (12) (2005) 1624–1637.

[33] J. Munkres, Algorithms for the assignment and transportation problems, Journal of the Society for Industrial and Applied Mathematics 5 (1) (1957) 32–38.

[34] A. Strehl, J. Gosh, Cluster ensembles: a knowledge reuse framework for combining multiple partitions, Journal of Machine Learning 3 (2002) 583–617.

[35] U. Brandes, A faster algorithm for betweenness centrality, Journal of Mathematical Sociology 25 (2) (2001) 163–177.

[36] J. Kleinberg, Authoritative sources in a hyper-linked environment, Journal of the ACM 46 (5) (1999) 604–632.

[37] K. Wagstaff, C. Cardie, S. Rogers, S. Schroedl, Constrained k-means clustering with background knowledge, in: Proceedings 18th International Conference on Machine Learning (ICML), 2001, pp. 577–584.

[38] X. Wang, B. Qian, I. Davidson, On constrained spectral clustering and its applications, Data Mining and Knowledge Discovery 28 (1) (2014) 1–30.

[39] Y. Chi, X. Song, D. Zhou, K. Hino, B. L. Tseng, On evolutionary spectral clustering, ACM Transactions on Knowledge Discovery from Data 3 (4).

[40] R. Langone, O. M. Agudelo, B. D. Moor, J. A. K. Suykens, Incremental kernel spectral clustering for online learning of non-stationary data, Neurocomputing 139 (2014) 246–260.

[41] H. Ning, W. Xu, Y. Chi, Y. Gong, T. Huang, Incremental spectral clustering by efficiently updating the eigen-system, Pattern Recognition 43 (1) (2010) 113–127.

[42] P. K. Desikan, N. Pathak, J. Srivastava, V. Kumar, Incremental page rank computation on evolving graphs, in: Proceedings 14th International Conference on World Wide Web (WWW), Special interest tracks and posters, 2005, pp. 1094–1095.

[43] U. Kang, B. Meeder, E. E. Papalexakis, C. Faloutsos, Heigen: Spectral analysis for billion-scale graphs, IEEE Transactions on Knowledge and Data Engineering 26 (2) (2014) 350–362.

[44] R. Mall, R. Langone, J. A. Suykens, Kernel spectral clustering for big data networks, Entropy 15 (5) (2013) 1567–1586.