Scalable Spectral Clustering with Weighted PageRank

Dimitrios Rafailidis, Eleni Constantinou, and Yannis Manolopoulos

Department of Informatics, Aristotle University of Thessaloniki,
54124 Thessaloniki, Greece
{draf,econst,manolopo}csd.auth.gr

Abstract. In this paper, we propose an accelerated spectral clustering method, using a landmark selection strategy. According to the weighted PageRank algorithm, the most important nodes of the data affinity graph are selected as landmarks. The selected landmarks are provided to a landmark spectral clustering technique to achieve scalable and accurate clustering. In our experiments with 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. Finally, we show that the proposed method outperforms baseline and accelerated spectral clustering methods, in terms of computational cost and clustering accuracy, respectively.

Keywords: Spectral clustering, sparse coding, databases.

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. The key idea is to achieve graph partitioning by performing eigendecomposition of the graph Laplacian matrix. Given a set of d-dimensional data points [1] $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \in \mathbb{R}^d$, 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 [2] $\{V_c\}_{c=1}^{k}$ 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. Ulrike von Luxburg's tutorial [15] includes examples of different Laplacians' constructions. For example, Ratio Cut [5] 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) [22] uses the same objective criterion as Ratio Cut, normalized by the total degree of each cluster, making thus

[1] Following standard notations, we use capital italic letters for matrices (e.g. A), lowercase bold letters for vectors (e.g. $\mathbf{a}$) and calligraphic fonts for sets (e.g. $\mathcal{A}$).
[2] k disjoint data subsets whose union is the whole data set.

Y. Ait Ameur et al. (Eds.): MEDI 2014, LNCS 8748, pp. 289–300, 2014.

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. Finally, the k-means algorithm is applied to generate the clusters.

SC methods have a number of real-world applications such as image segmentation [26], face recognition [4], feature fusion [12], speech recognition [16], 3D shape retrieval [25] and protein sequences clustering [21]. 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 high clustering accuracy.

With respect to the first key factor, baseline SC methods require $O(n^3)$ time 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. Consequently, accelerated methods significantly decrease the high complexity $O(n^3)$ of the baseline SC methods [5,22]. 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.

In this paper, we present 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. With the help of the selected landmarks and a landmark spectral clustering technique we achieve scalable and accurate clustering. In particular, our contribution is summarized as follows:

- High clustering accuracy is achieved by following the proposed landmark selection strategy of weighted PageRank.
- The complexity of the proposed spectral clustering method is preserved low, by following the landmark selection strategy of weighted PageRank and a landmark-based spectral clustering technique.
- In our experiments with two benchmark face and shape image data sets, several landmark selection strategies are examined for scalable spectral clustering.
- Finally, we show that the proposed method outperforms baseline and accelerated spectral clustering methods, in terms of computational cost and clustering accuracy, respectively.

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 [29], firstly performs k-means on the data set with a large number cluster number p 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 [10], Fowlkes et al. applied the Nyström [20] 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 [14], 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 [9] 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 [8].

In [7], Chen and Cai proposed an accelerated SC method with landmark-based representation (LSC). 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 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, by performing the k-means method, LSC-K adds a significant preprocessing cost into LSC. Moreover, the topological properties of the nodes/data points in the affinity graph are ignored. In doing so, the landmark selection strategy of LSC-K has limited clustering accuracy.

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 [7] 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 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 [28], 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:

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

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:

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

where o_b is the number of outgoing links of node b and o_c is the number of outgoing links of node c. Then, the weighted PageRank value $wpr(b)$ for a node $b \in \mathcal{V}$ is calculated 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 [13]. Finally, the p nodes with the highest wpr values are selected as landmarks.

3.3 Sparse Representation of the Affinity Submatrix

Following the sparse coding strategy of [7], based on the Nadaraya-Watson kernel regression [11], for any data point $\mathbf{x}_i$ its $\widehat{\mathbf{x}}_i$ approximation is calculated as:

$$\widehat{\mathbf{x}}_i = \sum_{j=1}^{p} z_{ji}\mathbf{u}_j \tag{4}$$

where $\mathbf{u}_j$ is the j-th column vector of U and z_{ji} is the ji-th element of Z. Then, to create the sparse representation of the Z affinity sparse matrix, the z_{ji} value is set to 0, if $\mathbf{u}_j$ is not among the $r \leq p$ nearest landmarks. Let $U_{\langle i \rangle} \in \mathbb{R}^{d \times r}$ denote a submatrix of U, composed of r nearest landmarks of $\mathbf{x}_i$. Then each element z_{ji} is computed as:

$$z_{ji} = \frac{\Phi(\mathbf{x}_i, \mathbf{u}_j)}{\sum\limits_{j' \in U_{\langle i \rangle}} \Phi(\mathbf{x}_i, \mathbf{u}_{j'})}, \qquad i \in 1 \ldots n \text{ and } j \in U_{\langle i \rangle} \tag{5}$$

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, where σ controls the local scale of each data point's neighborhood. Therefore, based on (5), the $Z \in \mathbb{R}^{p \times n}$ sparse representation is calculated. Consequently, 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 $\Sigma = diag(\sigma_1, \ldots, \sigma_p)$ and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$ are the singular values of $\widehat{Z}$, $A = [\mathbf{a}_1, \ldots, \mathbf{a}_p] \in \mathbb{R}^{p \times p}$ and $\mathbf{a}_i$'s are called left singular vectors, $B = [\mathbf{b}_1, \ldots, \mathbf{b}_p] \in \mathbb{R}^{n \times p}$ 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 [7] B can be computed as $B = \Sigma^{-1}A^T\widehat{Z}$. 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, i.e. the rows of B.

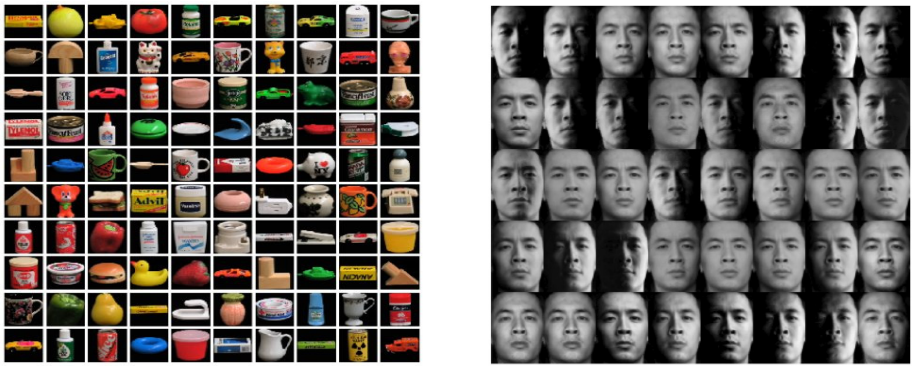
4 Experimental Results

4.1 Data Sets

In our experiments we used two high-dimensional benchmark data sets [3], including a shape image data set of the Columbia University Image Library (COIL100

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

[18]) and a face data set (CMU-PIE [23]) of Carnegie Mellon. 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. 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 [3], which consists of n =11,554 vectors of d =1024 dimensions with 68 clusters, where each cluster represents the face images of each person.



(a) (b)

Fig. 1. (a) The 100 objects of the COIL100 data set; (b) examples of face images of the CMU-PIE data set

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) [2] is defined as:

$$Acc = \frac{\sum_{i=1}^{n} \delta(c_i, map(c_i'))}{n} \tag{6}$$

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 [17]. 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:

$$MI(\mathcal{C}_{gnd}, \mathcal{C}_{alg}) = \sum_{c_i \in \mathcal{C}_{gnd}, c'_j \in \mathcal{C}_{alg}} p(c_i, c'_j) \log \frac{p(c_i, c'_j)}{p(c_i)p(c'_j)} \tag{7}$$

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) [24] is defined as:

$$NMI(\mathcal{C}_{gnd}, \mathcal{C}_{alg}) = \frac{MI(\mathcal{C}_{gnd}, \mathcal{C}_{alg})}{\sqrt{H(\mathcal{C}_{gnd})H(\mathcal{C}_{alg})}} \tag{8}$$

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 i7 2700K at 3.50 GHz, 8GB Ram using Matlab 2011a.

4.3 Results

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

- **Random**: nodes are randomly selected as landmarks, irrespective of their topological features in the affinity graph.
- **k-means**: 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 [1]. 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 [13], which estimates the importance of a node in the graph. To consider the weights of the links we used the weighted PageRank algorithm of [28]. Nodes with the highest PageRank values are selected as landmarks, as described in Section 3.2.

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, since it requires the calculation of all-to-all paths in the graph. Weighted PageRank 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 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.

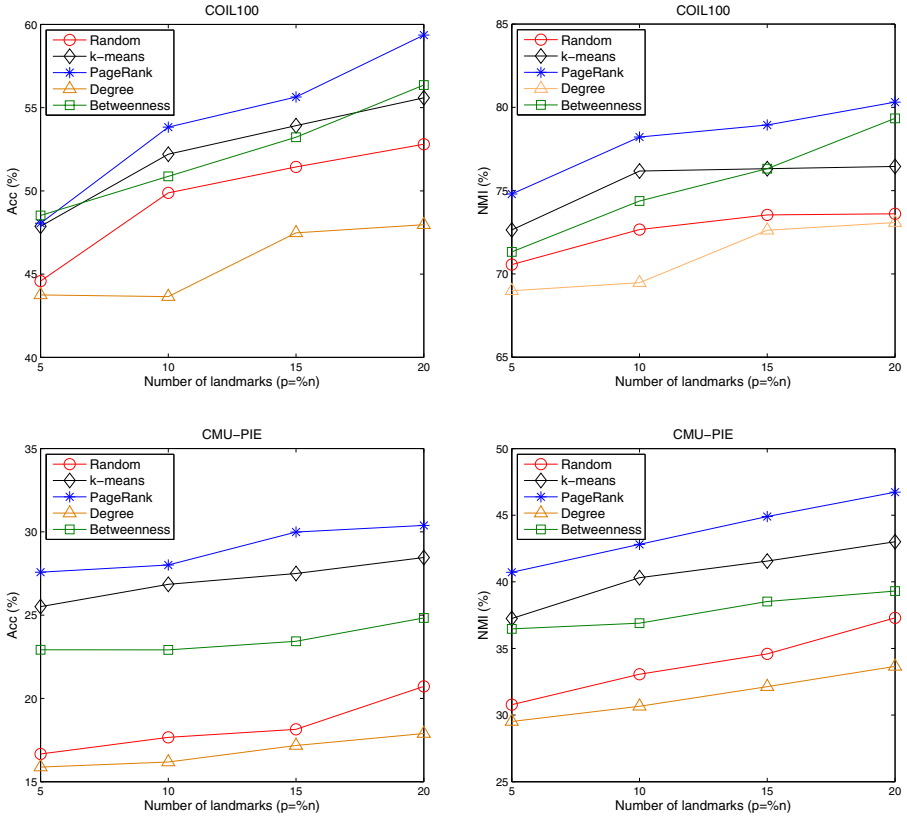


Fig. 2. Landmark selection strategies for Landmark Spectral Clustering (LSC)

In Fig. 2, 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 PageRank clearly outperforms the competitive strategies. This happens because 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

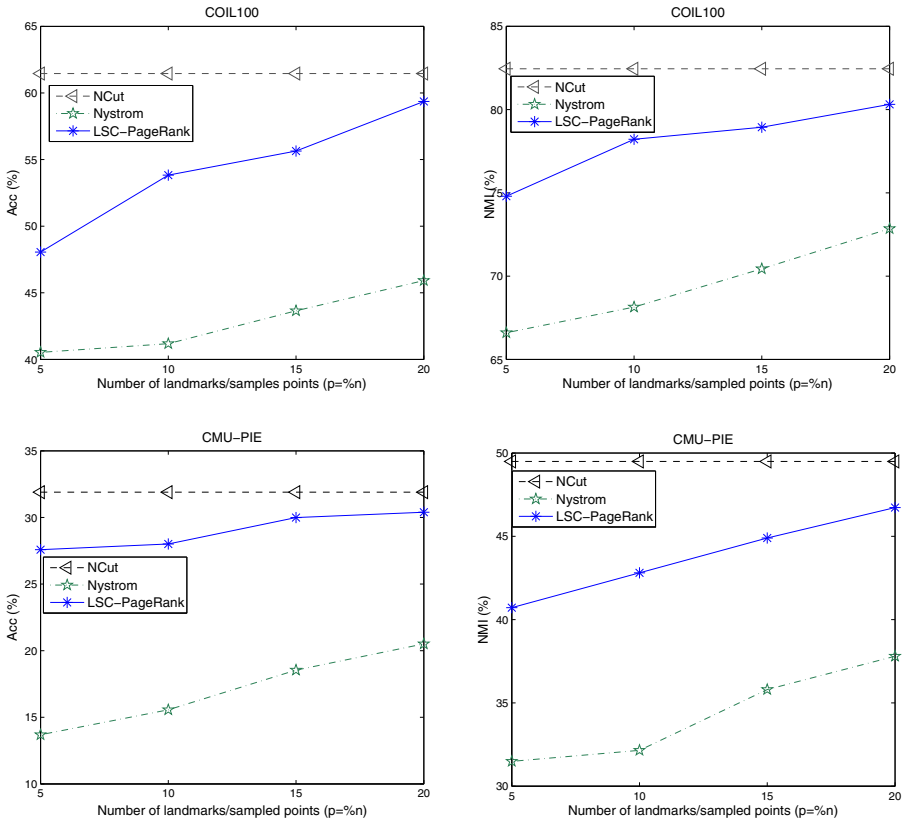


Fig. 3. Comparison of the proposed LSC-PageRank method against (a) the baseline NCut and (b) the accelerated Nyström spectral clustering methods.

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.

The proposed LSC-PageRank method is compared against the baseline NCut method [4] [22] and the accelerated Nyström spectral clustering method with orthogonalization [5] [10]. According to the experimental results of Fig. 3, 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. In Table 1, the computational cost of each examined method is presented. The baseline NCut method has high preprocessing cost $O(n^3)$, due to the eigendecomposition of the Laplacian matrix $L \in \mathbb{R}^{n \times n}$, whereas the accelerated Nyström method and the proposed

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

Table 1. CPU-time (sec) of the baseline NCut, the proposed LSC-PageRank and the accelerated Nyström spectral clustering methods.

	COIL100			**CMU-PIE**		
	NCut	LSC-PageRank	Nyström	NCut	LSC-PageRank	Nyström
$p = 5\%$	361.69	3.99	2.05	1,388.1	5.55	4.16
$p = 10\%$	361.69	4.99	5.38	1,388.1	6.99	17.14
$p = 15\%$	361.69	5.28	13.43	1,388.1	7.66	50.57
$p = 20\%$	361.69	6.54	28.45	1,388.1	9.08	115.27

LSC-PageRank method have a low computational overhead, by performing the eigendecomposition to a highly reduced matrix.

Summarizing, the proposed landmark selection strategy of weighted PageRank improves the clustering accuracy of LSC, by adding a low computational cost, in contrast to the rest of selection strategies that either ignore or consider the topological features of the nodes in the affinity graph. Additionally, the proposed LSC-PageRank method significantly outperforms the baseline NCut and the accelerated Nyström spectral clustering, in terms of preprocessing cost and clustering accuracy, respectively.

5 Conclusion

In this paper we presented an efficient method for accurate and scalable spectral clustering. In particular, we propose 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.

In real-world applications continuously and efficiently updates are required, over the data sets evolution. Recently, several incremental strategies [19] 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.

Additionally, several semi-supervised spectral clustering methods [8,27] 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.

Finally, modern web databases require a significantly large preprocessing cost for spectral clustering in billions of data. For instance, the work of Chen et al [6] introduced a parallel spectral clustering in distributed systems. Towards this aim, in our future work we plan to design the proposed method for distributed databases.

References

1. Brandes, U.: A faster algorithm for betweenness centrality. Journal of Mathematical Sociology 25(2), 163–177 (2001)
2. Cai, D., He, X., Han, J.: Document clustering using locality preserving indexing. IEEE Transactions on Knowledge and Data 17(12), 1624–1637 (2005)
3. Cai, D., He, X., Han, J.: Efficient kernel discriminant analysis via spectral regression. In: Proceedings of the 7th IEEE International Conference on Data Mining (ICDM), Omaha, NE, pp. 427–432 (2007)
4. Cevikalp, H., Triggs, B.: Face recognition based on image sets. In: Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA (2010)
5. Chan, P., Schlag, M., Zien, J.: Spectral k-way ratio cut partitioning. IEEE Transactions on CAD-Integrated Circuit and Systems 13, 1088–1096 (1994)
6. Chen, W.Y., Song, Y., Bai, H., Lin, C.J., Chang, E.Y.: Parallel spectral clustering in distributed systems. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(3), 568–586 (2011)
7. Chen, X., Chai, D.: Large-Scale spectral clustering with landmark-based representation. In: Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI), San Francisco, CA, pp. 313–318 (2011)
8. Chen, W., Feng, G.: Spectral clustering: a semi-supervised approach. Neurocomputing 77, 229–242 (2012)
9. Dhillon, I., Guan, Y., Kulis, B.: Kernel k-means, spectral clustering and normalized cuts. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Seattle, WA, pp. 551–556 (2004)
10. Fowlkes, C., Belongie, S., Chung, F., Malik, J.: Spectral grouping using the nyström method. IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004)
11. Härdle, W.: Applied non-parametric regression. Cambridge University Press (1992)
12. Huang, H.-C., Chuang, Y.-Y., Chen, C.S.: Affinity aggregation for spectral clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, pp. 773–780 (2012)
13. Kleinberg, J.: Authoritative sources in a hyper-linked environment. Journal of the ACM 46(5), 604–632 (1999)
14. Kulis, B., Basu, S., Dhillon, I., Mooney, R.: Semi-supervised graph clustering: a kernel approach. Journal of Machine Learning 74, 1–22 (2009)
15. Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17(4), 395–416 (2007)

16. Iso, K.: Speaker clustering using vector quantization and spectral clustering. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Dallas, TX, pp. 4986–4989 (2010)
17. Munkres, J.: Algorithms for the assignment and transportation problems. Journal of the Society for Industrial and Applied Mathematics 5(1), 32–38 (1957)
18. Nene, S.A., Nayar, S.K., Murase, H.: Columbia object image library. Department of Computer Science, Columbia University, New York, Technical Report CUCS-005-96 (1996)
19. Ning, H., Xu, W., Chi, Y., Gong, Y., Huang, T.S.: Incremental spectral clustering by efficiently updating the eigen-system. Pattern Recognition 43(1), 113–127 (2010)
20. Nyström, E.J.: Über die praktische Auflösung von Integralgleichungen mit Anwendungen auf Randwertaufgaben. Acta Mathematica 54, 185–204 (1930)
21. Paccanaro, A., Chennubhotla, C., Casbon, J.A., Saqi, M.A.S.: Spectral clustering of protein sequences. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN), Portland, OR, pp. 3083–3088 (2003)
22. Shi, J., Makil, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
23. Shim, T., Baker, S.: The CMU pose, illumination and expression database. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(12), 1615–1617 (2003)
24. Strehl, A., Gosh, J.: Cluster ensembles: a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning 3, 583–617 (2002)
25. Tatsuma, A., Aono, M.: Multi-Fourier spectra descriptor and augmentation with spectral clustering for 3D shape retrieval. Visual Computer 25(8), 785–804 (2009)
26. Tung, F., Wong, A., Clausi, D.A.: Enabling scalable spectral clustering for image segmentation. Pattern Recognition 43(12), 4069–4076 (2010)
27. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained k-means clustering with background knowledge. In: Proceedings of the 18th International Conference on Machine Learning (ICML), Williamstown, MA (2001)
28. Xing, W., Ghorbani, A.: Weighted PageRank algorithm. In: Proceedings of the 2nd Annual Conference on Communication Networks and Services Research (CNSR), Fredericton, Canada, pp. 305–314 (2004)
29. Yan, D., Huang, L., Jordan, M.I.: Fast approximate spectral clustering. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Paris, France (2009)