A Multiscale Spectral Method for Learning Number of Clusters

Anna Little
Department of Mathematics
Jacksonville University
Jacksonville, Florida 32211
Email: alittle2@ju.edu

Alicia Byrd
Department of Mathematics
Jacksonville University
Jacksonville, Florida 32211
Email: abyrd1@jacksonville.edu

***Abstract*—We propose a novel multiscale, spectral algorithm for estimating the number of clusters in a data set. Our algorithm computes the eigenvalues of the graph Laplacian iteratively for a large range of values of the scale parameter, and estimates the number of clusters from the maximal eigengap. Thus variation of the scale parameter, which usually confuses the clustering problem, is used to infer the number of clusters in a robust and efficient way. Commute distances are used to transform the distance matrix into a block-diagonal form, allowing the algorithm to succeed on irregularly shaped clusters, and the algorithm is applied to test data sets (both simulated and real-world) for method validation.**

I. Background

The goal of clustering algorithms is to partition a data set into groups so that similar points are assigned to the same group and dissimilar points to different groups. Clustering techniques are extremely useful in data mining and pattern recognition [1], and have been used in a wide variety of applications, including genetics [2], chemistry [3], image segmentation [4], and language processing [5].

Most clustering algorithms require the user to specify the number of clusters k, and developing robust and automated methods for selecting this parameter is an active area of research. Clustering is generally performed iteratively for a range of k values, and the k which optimizes some validity criterion is selected [6].

Various criteria that have been used including the classic Calinski-Harabasz criterion [7], the Silhouette coefficient ([8], [9]), information criteria such as BIC ([10], [11]), and information theoretic criteria such as the gap statistic [12] and the jump statistic [13]. Some multiscale methods include [6], [14], and [15]. More recently, [16] employ a bootstrap method and [17] employ a non-parametric method for estimating the number of clusters.

Various approaches based in spectral clustering have also been suggested, including [18]–[21]; these methods see the data set as a weighted graph and use spectral decompositions to estimate the number of clusters (the analysis in [22] justifies this approach). In [20], [21], a random walk is defined on the data and the number of clusters is estimated by finding the maximal eigenvalue gap of the transition matrix as the random walk diffuses.

In this paper we propose a new multiscale, spectral algorithm for learning the number of clusters. The approach is philosophically similar to [20], [21] but the method proposed for learning the number of clusters is different.

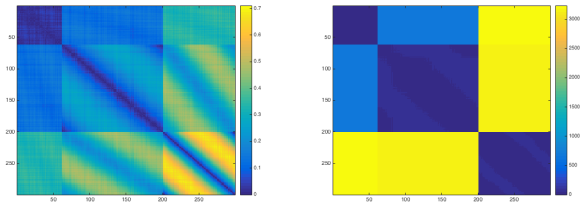


Fig. 1. Euclidean (left) and Euclidean commute (right) distances for the data set in Figure 2h.

II. Algorithm

A. Overview

Given N data points $x_1, x_2, \ldots, x_N$ in $\mathbb{R}^p$ and an appropriate distance function $d(x_i, x_j)$, our goal is to estimate the number of clusters k in the data set.

Define $\lambda_1(\sigma) \leq \lambda_2(\sigma) \leq \ldots \leq \lambda_N(\sigma)$ to be the eigenvalues of L_{sym}, where

$$L_{\text{sym}} = D^{-1/2}(D - W)D^{-1/2} \tag{1}$$

and the weights $W_{ij} = e^{-d(x_i,x_j)^2/2\sigma^2}$ have been defined using the Gaussian kernel and $D_{ii} = \sum_{j=1}^{n} W_{ij}$ is the diagonal degree matrix. In addition, for $1 \leq i \leq N-1$ define $\Delta_i(\sigma) = \lambda_{i+1}(\sigma) - \lambda_i(\sigma)$ to be the i^{th} eigenvalue gap. We compute these eigenvalues iteratively for $\sigma \in [\sigma_{\min}, \sigma_{\max}]$, and define the optimal number of clusters $\hat{k}$ by finding the maximal eigengap across all scales and indices:

$$\hat{k} = \arg\max_i \left(\max_{\sigma \in [\sigma_{\min}, \sigma_{\max}]} \Delta_k(\sigma) \right)$$

We will refer to this method as the Multiscale EigenGap (MEG) algorithm, which is summarized in the following pseudocode. Note that the algorithm outputs not only an estimate of the number of clusters $\hat{k}$ but also a recommended scale $\hat{\sigma}$ for running spectral clustering with the Gaussian kernel.

MEG Algorithm

Input:
X : an $N \times p$ set of N data points in $\mathbb{R}^p$
$d(x_i, x_j)$: distance function

Output:
$\hat{k}$: best estimate of number of clusters of X
$\hat{\sigma}$: recommended scale σ for spectral clustering

for $\sigma \in [\sigma_{\min}, \sigma_{\max}]$ **do**
$W_{ij} \leftarrow e^{-d(x_i,x_j)^2/2\sigma^2}$
$D_{ii} \leftarrow \sum_{j=1}^{n} W_{ij}$
$L_{\text{sym}} \leftarrow D^{-1/2}(D - W)D^{-1/2}$
$\lambda_1(\sigma) \leq \ldots \leq \lambda_N(\sigma) \leftarrow$ eigenvalues of L_{sym}
$\Delta_i(\sigma) = \lambda_{i+1}(\sigma) - \lambda_i(\sigma)$ for $1 \leq i \leq N-1$
end for
$\Delta_i \leftarrow \max_{\sigma \in [\sigma_{\min}, \sigma_{\max}]} \Delta_i(\sigma)$ for $1 \leq i \leq N-1$
$\hat{k} \leftarrow \arg\max_i \Delta_i$
$\hat{\sigma} \leftarrow \arg\max_{\sigma \in [\sigma_{\min}, \sigma_{\max}]} \Delta_{\hat{k}}(\sigma)$

B. Distance Function

For the proposed eigengap method to work one must have well-connected and well-separated clusters, so that the weight matrix is approximately block diagonal. For many clusters such as the last three data sets in Figure 2, this is not the case if Euclidean distance is used. However by using a commute distance instead of Euclidean distance, this approach works for irregularly shaped as well as convex clusters.

The commute distance c_{ij} between two vertices x_i and x_j is the expected time it would take for a random walk on the points to travel from x_i to x_j and back again ([23], [24]); it is thus dependent on the collection of all paths between x_i and x_j, and [25] showed that c_{ij} is in fact a distance measure.

In [26] and [25] it is shown that the commute distances for a fully connected graph can be computed from the Moore-Penrose (pseudo) inverse L^+ of the graph Laplacian $L = D - W$. In particular:

$$c_{ij} = \text{vol}(G)(L^+_{ii} - 2L^+_{ij} + L^+_{jj}),$$

where $\text{vol}(G) = \sum_{i=1}^{n} D_{ii}$ is the volume of the graph. Thus as noted in [26], the square root of the commute distance $\sqrt{c_{ij}}$ can be considered a Euclidean distance.

We therefore propose using the MEG algorithm with the Euclidean commute distance $\sqrt{c_{ij}}$ instead of Euclidean distance in $\mathbb{R}^p$ in order to transform the distance matrix into an approximate block diagonal form. We propose using the self-tuning affinity function proposed in [18] in the computation of the commute distances:

$$W_{ij} = e^{-d(x_i,x_j)^2/\sigma_i\sigma_j}, \tag{2}$$

where σ_i is the distance of x_i to it's K^{th} nearest neighbor. Pseudo-code for computing commute distances is given below.

Algorithm for Euclidean Commute Distances

Input:
X : an $N \times p$ set of N data points in $\mathbb{R}^p$
K: number of nearest neighbors for local scaling

Output:
$d(x_i, x_j)$: pairwise Euclidean commute distances

$\sigma_i \leftarrow$ distance of x_i to it's K^{th} nearest neighbor
if $p < 10$ **then**
$W_{ij} \leftarrow e^{-||x_i - x_j||^2/\sigma_i\sigma_j}$
else
$W_{ij} \leftarrow e^{-4||x_i - x_j||^2/\sigma_i\sigma_j}$
end if
$D_{ii} \leftarrow \sum_{j=1}^{n} W_{ij}$
$L \leftarrow D - W$
$L^+ \leftarrow$ Moore-Penrose inverse of L
$d(x_i, x_j) \leftarrow \sqrt{(\sum_{i=1}^{n} D_{ii})(L^+_{ii} - 2L^+_{ij} + L^+_{jj})}$

Although the above procedure leaves unanswered the question of how to choose K, [18] obtained good clustering results using a fixed $K = 7$ for all low and high-dimensional data. We adjusted their recommendation only slightly, using a fixed $K = 6$ for all examples, and replacing σ_i with $0.5\sigma_i$ for high-dimensional data.

Figure 1 shows both the Euclidean and Euclidean commute distances for the data set in Figure 2h; unlike the Euclidean distance matrix, the commute distance matrix of the rings has a block diagonal structure.

C. Relationship with Prior Work

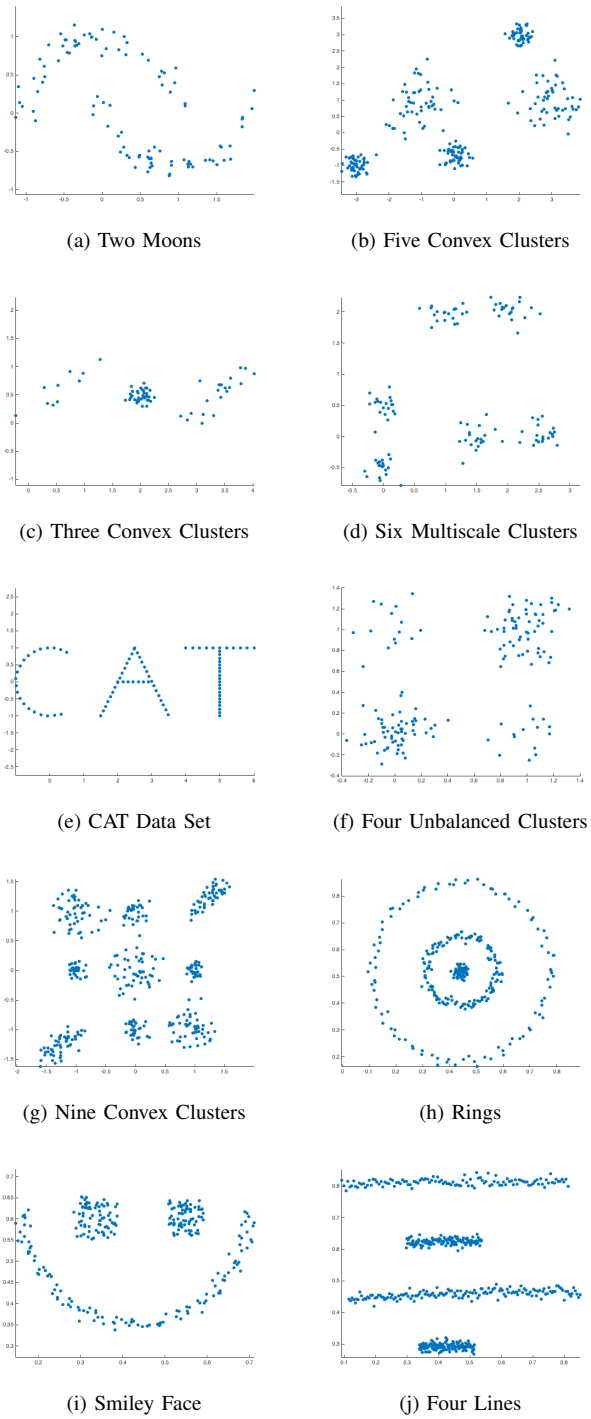
As noted in Section I, this approach has many similarities with the work in [20], [21]. In these papers the authors consider the Markov transition matrix $P = D^{-1}W$ of a random walk defined on the data; P^M gives the transition probabilities after M steps of the random walk. The authors estimate the number of clusters by finding the maximal eigengap across all times M. While in this approach the transition matrix is transformed into a block-diagonal structure by diffusing the random walk, in the MEG-CD algorithm this happens in one step by computing commute distances.

III. Numerical Results

A. Simulated Data

To compare the performance of the MEG algorithm with competing methods, experiments were running on the ten two-dimensional data sets shown in Figure 2. Comparisons were made with the following alternate methods for determining number of clusters:

1) MCLUST-BIC: the EM algorithm proposed in [27] with $\hat{k}$ chosen based on the BIC criteria ([10], [11]); method was run using the R package *mclust*.
2) PAM-AWS: the k-medoids algorithm proposed in [28] with $\hat{k}$ chosen using average width silhouettes ([8], [9]); method was run using the R package *fpc*.
3) AP: the affinity propagation method proposed in [29]. The algorithm was run using the R package *apcluster*.



(a) Two Moons (b) Five Convex Clusters

(c) Three Convex Clusters (d) Six Multiscale Clusters

(e) CAT Data Set (f) Four Unbalanced Clusters

(g) Nine Convex Clusters (h) Rings

(i) Smiley Face (j) Four Lines

Fig. 2. Data Sets used for algorithm testing; (h), (i), and (j) are from [18].

4) GapStatistic: the information theoretic approach in [12]. PAM was used as the clustering method; the algorithm was run using the R package *cluster*.
5) DBSCAN: the density-based approach in [30]. Note this method required manual selection of a parameter which was done according to the heuristic proposed in the paper, but results were sensitive to this parameter. The algorithm was run using the R package *fpc*.
6) Bootstrap: the bootstrap approach in [16] with k-means as the clustering method. The algorithm was run using the R package *fpc*.
7) Self-tuning: the spectral method proposed in [18]. Matlab code was downloaded from http://www.vision.caltech.edu/lihi/Demos/SelfTuningClustering.html.
8) RandomWalk: the spectral approach in [21]. No available code was found so the algorithm was implemented in Matlab.

TABLE I. ALGORITHM COMPARISONS FOR SIMULATED DATA

METHOD:	(a) Two Moons	(b) Five Convex Clusters	(c) Three Convex Clusters	(d) Six Multiscale Clusters	(e) CAT Data Set	(f) Four Unbalanced Clusters	(g) Nine Convex Clusters	(h) Rings	(i) Smiley Face	(j) Four Lines
MCLUST-BIC	6	**5**	**3**	**6**	8	1	**9**	14	6	5
PAM-AWS	**2**	**5**	**3**	**6**	**3**	11	**9**	16	7	9
AP	8	**5**	5	**6**	6	7	**9**	18	10	14
GapStatistic	**2**	**5**	1	**6**	1	1	1	1	7	1
DBSCAN	3	7	2	**6**	**3**	1	**9**	5	4	5
Bootstrap	**2**	2	10	3	**3**	10	10	10	2	8
Self-tuning	9	**5**	**3**	4	**3**	**4**	3	**3**	**3**	**4**
RandomWalk	**2**	7	6	5	**3**	**4**	**9**	**3**	**3**	**4**
MEG-ED	**2**	**5**	**3**	**6**	**3**	**4**	**9**	8	**3**	6
MEG-CD	**2**	**5**	**3**	**6**	**3**	**4**	**9**	**3**	**3**	**4**

Note all R packages referenced above are available at http://cran.r-project.org/web/packages [31]. Results are shown in Table I, with correct answers in bold. Note that all of the non-spectral methods fail on the last three data sets. For the first seven data sets consisting of more well-separated clusters, the PAM-AWS was the best non-spectral method. The RandomWalk and Self-tuning methods performed well on the irregularly shaped clusters, but each failed on three other examples. MEG-ED found the correct number of clusters on all data sets except (h) and (j), while MEG-CD was the only algorithm to find the correct number of clusters for all ten data sets. Although MEG-CD is clearly superior for irregularly shaped clusters, MEG-ED is faster and often more robust in the presence of noise and small clusters than MEG-CD.

B. Real-world Data

We also discuss the performance of the MEG algorithm on two real-world data sets available from UCI's Machine Learning Repository [32].

The first is the Vehicle Silhouette data set consisting of 846 images of four distinct vehicles, taken at various angles and elevations; there are 18 attributes for each image. MEG-ED and MEG-DC were applied to the standardized data. MEG-ED correctly returned $\hat{k} = 4$, while MEG-CD returned $\hat{k} = 3$.

The second is the Wine data set containing 178 samples of three different wine types, each derived from a distinct cultivar; there are 13 recorded attributes. Once again the data was standardized; both MEG-ED and MEG-CD correctly returned

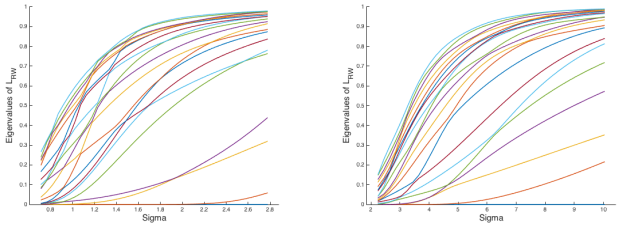


Fig. 3. Eigenvalues of the normalized graph Laplacian as a function of scale σ for the Vehicle Silhouettes data set (left) and Wine data set (right).

$\hat{k} = 3$ clusters. See Figure 3 for a plot of the eigenvalues as a function of σ for both data sets.

IV. Conclusion

This paper introduces a new multiscale spectral method for estimating the number of clusters in a data set. The variation of results as one changes the scale parameter σ, which usually confuses the clustering problem, is used to infer the number of clusters. The algorithm computes the eigenvalues of the normalized Laplacian iteratively for a range of σ values, and selects $\hat{k}$ by finding the maximal eigengap across all indices and scales. MEG-CD outperformed all competing methods in Section III-A on the test bank of simulated data sets, demonstrating robustness to irregular shapes, and MEG-ED outperformed all methods on convex data sets. In summary, the algorithm is computationally efficient, estimating k without solving the clustering problem, and is robust for noisy, unbalanced, irregularly shaped clusters.

Acknowledgment

The authors thank Jacksonville University for the 2015 Faculty Research Grant awarded to support this research.

References

[1] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," *Pattern recognition*, vol. 41, no. 1, pp. 176–190, 2008.

[2] C. Huttenhower, A. Flamholz, J. Landis, S. Sahi, C. Myers, K. Olszewski, M. Hibbs, N. Siemers, O. Troyanskaya, and H. Coller, "Nearest neighbor networks: clustering expression data based on gene neighborhoods," *Bmc Bioinformatics*, vol. 8, no. 1, p. 250, 2007.

[3] D. L. Massart and L. Kaufman, *The interpretation of analytical chemical data by th e use of cluster analysis*. Wiley, 1983.

[4] A. K. Jain and P. J. Flynn, *Image segmentation using clustering*. IEEE Press, Piscataway, NJ, 1996.

[5] A. Ushioda and J. Kawasaki, "Hierarchical clustering of words and application to nlp tasks," in *Proceedings of the Fourth Workshop on Very Large Corpora*, 1996, pp. 28–41.

[6] E. Nakamura and N. Kehtarnavaz, "Determining number of clusters and prototype locations via multi-scale clustering," *Pattern Recognition Letters*, vol. 19, no. 14, pp. 1265–1283, 1998.

[7] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[8] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[9] R. Lletı, M. C. Ortiz, L. A. Sarabia, and M. S. Sánchez, "Selecting variables for k-means cluster analysis by using a genetic algorithm that optimises the silhouettes," *Analytica Chimica Acta*, vol. 515, no. 1, pp. 87–100, 2004.

[10] C. Fraley and A. E. Raftery, "How many clusters? which clustering method? answers via model-based cluster analysis," *The computer journal*, vol. 41, no. 8, pp. 578–588, 1998.

[11] G. Schwarz *et al.*, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[12] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[13] C. A. Sugar and G. M. James, "Finding the number of clusters in a dataset," *Journal of the American Statistical Association*, vol. 98, no. 463, 2003.

[14] R. Kothari and D. Pitts, "On finding the number of clusters," *Pattern Recognition Letters*, vol. 20, no. 4, pp. 405–416, 1999.

[15] M. Herbin, N. Bonnet, and P. Vautrot, "Estimation of the number of clusters and influence zones," *Pattern Recognition Letters*, vol. 22, no. 14, pp. 1557–1568, 2001.

[16] Y. Fang and J. Wang, "Selection of the number of clusters via the bootstrap method," *Computational Statistics & Data Analysis*, vol. 56, no. 3, pp. 468–477, 2012.

[17] A. Fujita, D. Y. Takahashi, and A. G. Patriota, "A non-parametric method to estimate the number of clusters," *Computational Statistics & Data Analysis*, vol. 73, pp. 27–39, 2014.

[18] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Advances in neural information processing systems*, 2004, pp. 1601–1608.

[19] G. Sanguinetti, J. Laidler, and N. D. Lawrence, "Automatic determination of the number of clusters using spectral algorithms," in *Machine Learning for Signal Processing, 2005 IEEE Workshop on*. IEEE, 2005, pp. 55–60.

[20] A. Azran and Z. Ghahramani, "Spectral methods for automatic multiscale data clustering," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1. IEEE, 2006, pp. 190–197.

[21] ——, "A new approach to data driven clustering," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 57–64.

[22] A. Y. Ng, M. Jordan, Y. Weiss *et al.*, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.

[23] L. Lovász, "Random walks on graphs: A survey," *Combinatorics, Paul erdos is eighty*, vol. 2, no. 1, pp. 1–46, 1993.

[24] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[25] D. J. Klein and M. Randić, "Resistance distance," *Journal of Mathematical Chemistry*, vol. 12, no. 1, pp. 81–95, 1993.

[26] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, "A novel way of computing dissimilarities between nodes of a graph, with application to collaborative filtering and subspace projection of the graph nodes," Technical Report, Tech. Rep., 2006.

[27] C. Fraley and A. Raftery, "Mclust: Software for model-based cluster and discriminant analysis," *Department of Statistics, University of Washington: Technical Report*, no. 342, 1998.

[28] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009, vol. 344.

[29] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *science*, vol. 315, no. 5814, pp. 972–976, 2007.

[30] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[31] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2014. [Online]. Available: http://www.R-project.org/

[32] K. Bache and M. Lichman, "UCI Machine Learning Repository Irvine," http://archive.ics.uci.edu/ml, 2013, university of California, Irvine, School of Information and Computer Sciences.