HAL
open science

A heuristic for the automatic parameterization of the spectral clustering algorithm

Pierrick Bruneau

To cite this version:

Pierrick Bruneau. A heuristic for the automatic parameterization of the spectral clustering algorithm. 2013. hal-00868416

HAL Id: hal-00868416
https://hal.science/hal-00868416

Preprint submitted on 1 Oct 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A heuristic for the automatic parameterization of the spectral clustering algorithm

Pierrick Bruneau

CRP - Gabriel Lippmann
41, rue du Brill, L-4422 Belvaux (Luxembourg)

Abstract

Finding the optimal number of groups in the context of a clustering algorithm is a known as a difficult problem. In this article, we describe and evaluate a heuristic thereof for the spectral clustering algorithm. Our method is deterministic, and remarkable by its low computational burden. We show its effectiveness in most cases. Some limits are identified though, and serve to the formulation of perspectives to this work.

1 Introduction

Clustering a set of objects in a pre-defined number of groups is often difficult, according to the chosen model and criterion. The optimal choice of the number of groups itself (disambiguated as the variable k in the remainder) is maybe even more complex. The generally accepted Occam's Razor principle, under which the number of clusters is as small as acceptable, forbids the exhaustiveness of the clustering structure, no compromise between these antagonistic objectives being valid *a priori*.

In practice, this parameter often has to be set manually by the practitioner, even with recent data analysis software packages. For an exploratory approach, with k likely to be unknown, a heuristic is desirable.

In this article, we restrict to the spectral clustering algorithm, and propose a new simple, cost-effective way of estimating k from the spectrum of the Laplacian that characterizes this algorithm. Bartlett's test for the equality of variances is used since long for determining the number of factors to retain in the context of a Principal Component Analysis (PCA) [3]. We show that a pretty straightforward adaptation is possible for the estimation of k in the context of the spectral clustering algorithm.

First, we recall the state of the art about spectral clustering, and k estimation methods in this context. We then describe our method, coming up with a simple algorithm. The efficiency of the method is illustrated by experiments using synthetic and real data from the literature. A critical view of our results allows us to formulate some perspectives, given in conclusion.

2 Fundamentals of spectral clustering

The basics of spectral clustering emerge from the graph theory literature. This technique was popularized by [6] and [5]. Given a collection of N elements, represented by a symmetric pairwise similarity matrix[1] $\mathbf{S}$, the spectral clustering algorithm in k groups of elements can be stated as follows:

- Compute the diagonal matrix $\mathbf{D}$, with $\mathbf{D}_{nn} = \sum_{n'=1}^{N} \mathbf{S}_{nn'}$;
- Compute the Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{S}$;
- Eigen-decomposition of $\mathbf{L}$;
- With the k minor eigenvectors (i.e. associated to the k smallest eigenvalues) as columns, form the matrix $\mathbf{Y}$;
- Run the k-means algorithm on the rows of $\mathbf{Y}$, obtaining the labels of the respective elements of $\mathbf{S}$. ;

Algorithm 1: The spectral clustering algorithm

Variants of this algorithm mostly differ from the Laplacian used. The default one, unnormalized, induces some practical difficulties (e.g. dependency on the data domain and distribution) [7]. The following normalizations were thus proposed in the literature:

$$\text{symmetric version [5]: } \mathbf{L}_{\text{sym}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}}, \tag{1}$$

$$\textit{random walk} \text{ version [6]: } \mathbf{L}_{\text{rw}} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{S}, \tag{2}$$

with $\mathbf{I}$ the $N \times N$ identity matrix. Let us note that the multiplicity of the eigenvalue 0 in decomposing these Laplacians can be interpreted as the number of connected components of the underlying graph [7], i.e. the number of clusters formed by its vertices. Another notable normalization variant is $\mathbf{L}_{\text{alt}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}}$ [8]. A recent R implementation actually grounds upon the latter [4]. The inspection of equation (1) shows that $\mathbf{L}_{\text{sym}} = \mathbf{I} - \mathbf{L}_{\text{alt}}$. As a consequence, the algorithm 1 is adapted to $\mathbf{L}_{\text{alt}}$ by considering the major eigenvectors, and linking k to the multiplicity of the eigenvalue 1.

3 State of the art on determining the number of clusters

The link between the parameter k of the algorithm 1, and the multiplicity of the eigenvalue 0 in the spectrum of the normalized Laplacian is strictly valid only for connected components. However, graphs considered here may contain several components weakly connected with each other, but not completely disjoint: for example, similarities computed by a Radial Basis Function (RBF) never equal exactly 0, thus always inducing a single connected component for the whole data set. The goal of the algorithm is then precisely to find this structure.

[1] These similarities can also be interpreted as edge weights, of *kernel* values, without loss of generality. The diagonal of the matrix is conventionally 0.

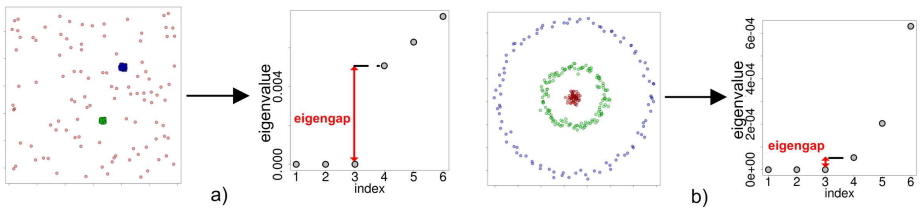


Figure 1: Minor eigenvalues profile for *synth2* and *synth1* (see Section 5 for a description).

In the remainder, to handle the variability of data sets from both domain and distribution points of view, we use the RBF variant proposed in [4]. The latter adapts the radius of the function to each element according to the median of the K nearest neighbors. As advocated by the authors, we retained $K = 5$ for our experiments, including the spectra shown in Figure 1.

Figure 1a shows that the profile of minor eigenvalues can inform us on the probable optimal value for k. Intuitively, only the latest eigenvalue equals exactly 0, $k-1$ other are *approximately* equal to 0, and the rest is *significantly* different from 0: the best value for k then emerges through the absolute difference between the kth and the $(k+1)^{\text{th}}$ eigenvalue, named *eigengap* [7].

Most existing work determines the likely eigengap empirically, either by comparing candidates to an arbitrary threshold, or by analyzing the variation rate of the eigenvalue profile *via* the scree test of Cattell [1]. Figure 1b illustrates that even for rather simple data sets, applying this test in an automated setting can be problematic. An iterative optimization procedure was also proposed, but remains complex, both from the conceptual and computational points of view [8]. We propose a simple and efficient alternative, by adapting Bartlett's test for equal variances to the spectral clustering case. Like the scree test, it was originally employed for determining the number of factors to retain in the context of a PCA [3].

4 Method description

Considering a sample of N elements described by p variables, the PCA computes the q factors representative of the sample covariance matrix, using the implicit hypothesis that uni-dimensional samples generated by any of the $k = p - q$ remaining factors must have an identically small variance. This null hypothesis can be tested by a χ^2 test. The following test statistic weas proposed in [3]:

$$-\left(N - 1 - q - \frac{k^2+1}{3k} + \sum_{i=1}^{q} \frac{\bar{\lambda}_k^2}{(\lambda_i - \bar{\lambda}_k)^2}\right) \ln(V_q) \sim \chi^2_{\frac{(k+2)(k-1)}{2}}, \qquad (3)$$

with λ_i the ith eigenvalue taken in decreasing order (as conventional with PCA), $\bar{\lambda}_k$ the mean of the k smallest eigenvalues, and $V_q = \prod_{i=q+1}^{p} (k\lambda_i / \sum_{j=q+1}^{p} \lambda_j)$.

Algorithm 2 is then a simple way to find the smallest acceptable value for q. This algorithm is $O(p^2)$. As the eigen-decomposition is cubic, the computational overhead is then modest.

Data: The vector of p eigenvalues, a risk level α, e.g. 5%
Result: The smallest acceptable q
$q \leftarrow 0$;
repeat
  $q \leftarrow q + 1$;
  $s \leftarrow$ statistic from Equation (3) ;
  /* $q < p - 1$ because Equation (3) is defined for $k > 1$ */
until $q = p - 2$ *or* $P_{\chi^2}(X < s) \leq 1 - \alpha$;
/* The minimal q that does not lead to rejecting the null hypothesis is obtained */

Algorithm 2: A simple algorithm for determining the number of PCA factors

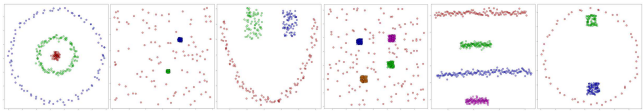
The determination of k for the spectral clustering algorithm is analogous to that of estimating the number of PCA factors to be retained: instead of looking for the q major eigenvalues of a covariance matrix, we focus on the k smallest eigenvalues of a Laplacian (see Section 2). The algorithm 2 simply has to be adapted to the search of the largest acceptable value for $k = N - q$ (indeed, $N = p$ in a Laplacian). In the context of clustering, $k << N$: therefore it is more efficient to have the search starting at $k = 2$, i.e. initialize q to $p-2$ in algorithm 2, and decrement at each iteration, with an adapted stopping criterion.

Furthermore, we empirically noticed that with $k << N$, the set of eigenvalues $\{\lambda_i\}_{i \leq q}$ of the normalized Laplacian is very close to 1 in average: this allows to approximate $\sum_{i=1}^{q} \bar{\lambda}_k^2/(\lambda_i - \bar{\lambda}_k)^2$ by $q\bar{\lambda}_k^2/(1-\bar{\lambda}_k)^2$ in Equation (3), leading to a criterion depending only on the k minor eigenvalues. Interlacing Algorithms 1 and 2, an incremental extraction of the eigenvalue from the smallest can then be stopped early, as soon as acceptable. AS $k << N$, we obtain a spectral clustering algorithm in $O(N^2)$, that includes the automatic determination of k.

5 Experimental Results

Our method is implemented as a R package, *speccalt*[2], i.e. an *alternative* to the *specc* function from the *kernlab* R package. The interface is minimal, and only requires a similarity matrix as input; k is optional, and automatically estimated if absent. We used $\mathbf{L}_{\text{alt}}$ as the Laplacian for the clustering algorithm, as suggested in [4, 8], as this leads to more numerical stability in practice. However, Algorithm 2 is still based upon $\mathbf{L}_{\text{rw}}$[3]; those two distinct Laplacians have thus to be independently decomposed. Yet complexities announced in Section 4 remain valid, and then only change by a constant factor.

[2] `http://cran.r-project.org/web/packages/speccalt/index.html`.
[3] or indistinctly $\mathbf{L}_{\text{sym}}$, both Laplacians sharing the same spectrum.



Data sets (ground truth)	Our method	State of the art	corrected Rand index
synth1(3)	**3**	4 ± 0,00	0,88 ± 0,18
synth2(3)	**3**	5 ± 0,00	0,97 ± 0,12
synth3(3)	**3**	**3** ± 0,00	0,90 ± 0,21
synth4(5)	**5**	**5** ± 0,00	0,76 ± 0,18
synth5(4)	**4**	**4** ± 0,00	0,89 ± 0,21
synth6(3)	2	4 ± 0,00	0,58 ± 0,00
iris(3)	2	4 ± 0,00	0,54 ± 0,00
isolet(5)	2	20 ± 0,00	0,39 ± 0,00

Figure 2: *Top*: Synthetic data sets from [8]. The ground truth is indicated by glyph colors. *Bottom*: Synthesis of the experimental results. Means and standard deviations of 20 independently computed Rand indexes are reported. The same procedure is applied with the method from [8].

For the evaluation, we retrieved the 6 synthetic samples introduced in [8] (see Figure 2), and used two well-known UCI data sets, *iris* (150 elements, 4 features) and the *isolet* vowels (1500 elements, 617 features). The synthetic samples are named from *synth1* to *synth6*, according to their position from right to left in Figure 2. For these experiments, we used our implementation of the spectral clustering algorithm with the automatic estimation of k. This estimate, along with the corrected Rand index [2, section 7.2.4], are recorded for erach data set. As a comparison, we also reported the respective estimates obtained by the method in [8][4]. As Algorithm 1 is sensitive to local minima through its dependence on k-means, the corrected Rand index is averaged from 20 independent executions for each data set. The same procedure was performed for the method in [8], as an account of its iterative nature. This was not needed for our method, as Algorithm 2 is deterministic. Those results are presented in Figure 2.

We first notice that our heuristic performs better than the method used as a reference. The result is satisfactory for the first data sets, but less with *isolet*, *synth6*, and *iris* (2 inferred clusters, against respectively 5, 3 et 3 according to the ground truth). This is somehow reflected by a clear degradation of the respective Rand indexes. The ground truth of *isolet* is not characterized by clear decision frontiers, which leads our method to inferring the minimal number of clusters. The cases of *synth6* and *iris* are more subtle: by following exactly

[4] We used the Matlab code available at `http://www.vision.caltech.edu/lihi/Demos/SelfTuningClustering.html`.

Algorithm 2, we would have found respectively 62 and 29 clusters. Actually, our method does not penalize an excessive count of clusters, or their being very small: each point in the loosely populated circle in *synth6* is attributed to its own cluster. The almost discrete nature of *iris* (i.e. all its values have at most one decimal) also seems problematic. To handle this, our implementation of Algorithm 2 explicitly bounds k from above by 20. If $1-\alpha$ is not reached for any k within the bounds, the threshold is lowered to the largest quantile measured for $k \in [2, 20]$. For a fair comparison, the method in [8] is parameterized likewise.

6 Conclusion

In this article, we proposed a simple and effective method, with low computational cost, that automatically estimates k in the context of the spectral clustering algorithm, as demonstrated by our experiments. However, we also highlighted some limits to the approach, by its exclusive focus to the characterization of manifolds in data.

The spectral clustering algorithm uses k-means as an intermediate step: the latter, equivalent to a mixture of isotropic Gaussians estimated by an EM algorithm, opens a possibility of combining our method to a Bayesian estimation of k, for example by deriving a prior distribution from our heuristic.

References

[1] R. B. Cattell. The scree test for the number of factors. *Multivariate Behavioral Research*, 1(2):245–276, 1966.

[2] A. D. Gordon. *Classification*. Chapman and Hall, 1999.

[3] A. T. James. Test of equality of the latent roots of the covariance matrix. *Multivariate Analysis, Volume 2*, 1969.

[4] A. Karatzoglou, A. Smola, and K. Hornik. *kernlab (R package)*, 2013.

[5] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *NIPS*, 2001.

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

[7] U. von Luxburg. A tutorial on spectral clustering. Technical report, Max Planck Institute for Biological Cybernetics, 2006.

[8] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. *NIPS*, 2004.