Trabalho apresentado no XLI CNMAC, Unicamp - Campinas - SP, 2022.

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics
Preprint

Hierarchical Similarity Measure for Spectral Clustering

Lucas Siviero Sibemberg[1], Luiz Emilio Allem[2], Carlos Hoppen[3]
IME/UFRGS, Porto Alegre, RS

Abstract. In this paper we propose a novel similarity measure for spectral clustering that incorporates a hierarchical component. The main advantage of this measure is that it produces an algorithm that does not depend on any scaling parameter, making it very easy to apply. Our experiments showed that our algorithm performs better than other spectral clustering methods on synthetic data sets with complex shape and multiple scales.

Keywords. Data Science, Clustering, Spectral Graph Theory, Spectral Clustering, Similarity Measure.

1 Introduction

Nowadays we have a huge amount of information available and it is a hard task to interpret it. Classifying this information into a small number of classes can help us gain valuable insight about our data, and this is widely applied in many fields. This is the aim of clustering algorithms, which seek to split data points into a given number of clusters in a way that data points with similar properties lie the same cluster and dissimilar data points lie in different clusters [9].

Traditional clustering methods, such as single linkage [7] and k-means [4] are easy to describe and to implement, but they often cannot deal with more complex structures. For instance, the performance of single linkage is sensitive to outliers [1] and k-means has been designed to obtain tight clusters in metric spaces, and therefore returns only convex clusters. Spectral clustering refers to a family of clustering algorithms that are based on linear algebra. They rely on similarity measures that are typically used to map the original data points into a new Euclidean space, allowing the algorithms to capture structural similarities beyond the distances between two points. A class of spectral algorithms based on the similarity measure will be presented in Section 2. Compared with traditional methods, they have many fundamental advantages. For example, they are able to find non-convex structures.

As we shall see, the definition of the similarity measure plays a fundamental role in spectral clustering [3]. Assuming that the original data points lie in a Euclidean space, the Gaussian kernel function $e^{-||x_i-x_j||^2/2\sigma^2}$ has been widely used to measure the similarity between two different data points x_i and x_j. It started with the seminal work of Shi and Malik [6] and of Ng *et al.* [5], and has led to good results in different situations.

According to this definition, the Gaussian kernel function depends on a scaling parameter σ, which has to be set by the user, and it is well known that the outcome of spectral clustering is quite sensitive to this parameter [1]. In many applications, the ‘good’ values of σ form a bounded interval, and a bottleneck of the method is to find such a good σ, especially for more complex structures [9]. To illustrate the problem, we applied the spectral clustering algorithm of Ng *et*

[1] lucas.siviero@ufrgs.br
[2] emilio.allem@ufrgs.br
[3] choppen@ufrgs.br

DOI: 10.5540/03.2022.009.01.0263

al. [5] for two values of σ that produce very different classifications, see Figure 1. Some natural definitions of σ, such as the standard deviation and the variance are known to perform poorly in several applications. Ng *et al.* [5] suggest choosing the value of σ that gives the tightest clusters. This criterion may be applied to select a clustering between two competing options, but it does not determine the interval where good values of σ may be found.

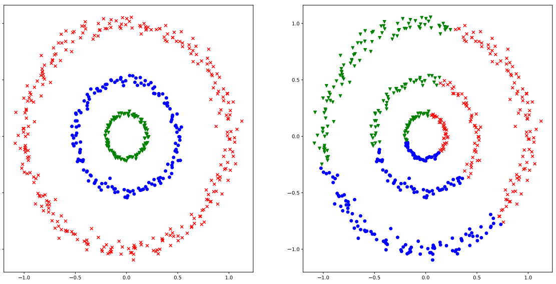

Figure 1: Results of the spectral clustering algorithm by Ng *et al.* [5] for two different scaling parameters in a data set with three circles. On the left $\sigma = 0.075$ and on the right $\sigma = 0.1$. The first is what we call a good value of σ. Source: The authors (2022).

A lot of effort has been made towards finding criteria that leads to good values of σ, but this seems to be a hard problem. Several authors have proposed alternative similarity measures to address this problem [2], as will be discussed in Section 3. To the best of our knowledge, there is no similarity measure in the literature that fully avoids a scaling parameter and, at the same time, defines a clustering algorithm whose performance is as good as the performance of traditional spectral clustering for an appropriate choice of the scaling parameter. The algorithm of this paper closes this gap.

We propose a novel similarity measure derived from the Gaussian kernel that incorporates a hierarchical component. Hierarchical approaches have been applied in the early days of clustering. One such approach is the agglomerative hierarchical clustering algorithm known as single linkage [7]: initially, each data point lies in its own cluster, and, at each step, the algorithm merges the clusters that are closest to each other, until all points lie in the same cluster. This produces a tree, or dendrogram, that gives a hierarchy of clusters. According to this method, if we wish to split our original data set into k clusters, it suffices to ignore the k last merging operations. Being a greedy procedure, this strategy does not provide good quality clusterings for many data sets.

Our method is based on the assumption that data points that lie in the same cluster in early steps of the procedure must have high similarity. This is used to replace the fixed scaling parameter σ in the definition of the Gaussian kernel by a factor that depends on the dendrogram for each pair of data points. This gives a similarity measure that has several benefits, for instance, no scaling parameter is required in the similarity measure, it is invariant under translations and expansions, it may be computed easily and, according to our experiments, the spectral clustering framework using this measure performs well in comparison to other traditional methods.

2 A Spectral Clustering Algorithm

In this section, we present a class of spectral algorithms that are based on similarity measures. This is a general framework derived from the work of Ng *et al.* [5]. Given a set of n data points $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$, a similarity measure on this data points may be viewed as a matrix $S = (s_{ij})$ such that s_{ij} is the similarity between x_i and x_j. This framework also uses a positive integer k, the number of clusters, as an input. It proceeds as follows:

(A) Let D be the diagonal matrix with $D_{ii} = \sum_{l=1}^{n} s_{il}$, and consider its normalized Laplacian matrix $\mathcal{L} = D^{-1/2} L D^{-1/2}$, where $L = D - S$.

(B) Compute orthogonal unit eigenvectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k \in \mathbb{R}^n$, where $\mathcal{L}\mathbf{x}_i = \lambda_i \mathbf{x}_i$ and $\lambda_1 \leq \ldots \leq \lambda_k$ are the k smallest eigenvalues of $\mathcal{L}$ (counting multiplicities). Consider the matrix $X = [\mathbf{x}_1 \mathbf{x}_2 \ldots \mathbf{x}_k] \in \mathbb{R}^{n \times k}$ with columns $\mathbf{x}_1, \ldots, \mathbf{x}_k$.

(C) Define the matrix $Y = (y_{ij})$ from $X = (x_{ij})$ so that rows have unit length, i.e., $y_{ij} = x_{ij} / \sqrt{\sum_{j=1}^{n} x_{ij}^2}$.

(D) Split the set $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ of rows of Y into k clusters $C_1, \ldots, C_k$ via k-means. Return $\mathcal{C}$ the partition such that data point i is assigned to cluster ℓ if and only if $\mathbf{y}_i$ lies in C_ℓ.

3 Similarity Measures in the Literature

Experiments showed that the performance of the framework in Section 2 varies a lot according to the choice of the similarity measure [1]. As mentioned in the introduction, a similarity measure based on the Gaussian kernel has been introduced in the early papers about spectral clustering [5, 6]. Given a data set $\mathcal{X} = \{x_1, \ldots, x_n\}$ in a Euclidean space, the similarity between the points x_i and x_j is defined as

$$s_1(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right) \text{ if } i \neq j, \tag{1}$$

where $||x_i - x_j||$ is the Euclidean distance between x_i and x_j and $\sigma > 0$ is a scaling parameter. This similarity and the ones that will be presented below are defined equal to 0, if $i = j$.

The algorithm based on the framework of Section 2 using s_1 as the similarity measure will be called Standard Spectral Clustering and denoted by SC-GK. It often has a good performance in many situations when the parameter σ is well chosen, but there is no fixed σ that works for every data set. Therefore, the value of σ must depend on the data set. Usually, this parameter is set manually [9], in the sense that the algorithm is run for many values of σ and the best solution is chosen by the user. A clear disadvantage is that running the algorithm for many values of σ slows it down. A more methodological drawback is that it is not always clear how to compare solutions obtained for different values of σ. For synthetic data sets, they may be compared to a known optimal solution, but this is not the case for real data sets. Other authors, such as [2], have also pointed out that, because the same scaling parameter is used for all pairs of points, it may not reflect the data distribution accurately, particularly if it contains data points in different scales.

In this section, we describe two additional similarity measures that have been proposed in the literature. Zelnik-Manor and Perona [8] replace σ in (1) by a product $\sigma_i\sigma_j$ that depends on the particular pair of data points under consideration. The parameter σ_i is defined as $\sigma_i = ||x_i - x_{i_\ell}||$ where x_{i_ℓ} denotes the ℓ-th closest data point to x_i. The value $\ell \in \mathbb{N}$ is a parameter chosen by the user, and again plays an important role controlling the size of the neighborhood. In short, the authors defined a similarity measure $s_2(x_i, x_j) = \exp\left(-||x_i - x_j||^2 / \sigma_i\sigma_j\right),\ i \neq j$. The algorithm that uses this similarity measure in the framework of Section 2 is known as Self-Tuning Spectral Clustering (SC-ST). One of the advantages of this method is that, in typical applications, even reasonably small values of ℓ give good results, reducing the number of possibilities.

Zhang *et al.* [9] proposed another variation of the Gaussian kernel similarity measure, which they called the density adaptive similarity measure. In addition to the parameter σ, it uses a parameter $\epsilon > 0$ that defines whether two data points are close to each other or not. For each pair of data points, the number of common-near-neighbors is defined as $\mathrm{CNN}(x_i, x_j) = |\{x \in \mathcal{X} : ||x_i - x|| < \epsilon \text{ and } ||x_j - x|| < \epsilon\}|$. The similarity measure is defined as $s_3(x_i, x_j) = \exp\left(-||x_i - x_j||^2/(2\sigma^2(\mathrm{CNN}(x_i, x_j) + 1))\right),\ i \neq j$. The algorithm that uses this similarity measure in the framework of Section 2 is known as Density Adaptive Spectral Clustering (SC-DA). The experiments in [9] showed that their method was able to amplify the intra-cluster similarity in many situations. However, in addition to σ, this approach also requires the user to fix a parameter ϵ that defines this notion of closeness.

4 Hierarchical Similarity Measure

The two similarity measures described in the previous section have been designed to increase the range of good values of the scaling parameter σ in SC-GK, or to replace σ by another scaling parameter that is easier to set. In this section, we propose a similarity measure that modifies the Gaussian kernel in a way that requires no scaling parameter. Instead, it incorporates information from a hierarchical tree with weights on the vertices, as we now explain.

We start with some terminology. Consider a graph H with weights on its vertices, that is, a triple $H = (V, \omega_V, E)$ with vertex set V, edge set E and a weight function $\omega_V : V \to \mathbb{R}_{\geq 0}$ that assigns a nonnegative weight ω_i to each vertex v_i in V.

Agglomerative clustering is based on a simple idea. Given a data set $\mathcal{X}$, each data point initially lies in its own cluster. Step by step, the algorithm merges two clusters until all points lie in the same cluster. There are many criteria for deciding which clusters to merge [7]. Here, if $\mathcal{C} = \{C_1, \ldots, C_k\}$ is the set of clusters produced up to a certain step, the distance between two clusters A and B is simply $d(A, B) = \min\{||x - y|| : x \in A,\ y \in B\}$. We then merge two clusters C_i and C_j for which $d(C_i, C_j)$ is minimum. This produces the following tree, or dendrogram, that encodes the hierarchy of clusters. Initially, a vertex of weight zero is created for each of the n data points in the data set. In other words, there is a vertex for each cluster at the start of the agglomerative procedure. When two clusters C_i and C_j are merged, a new vertex corresponding to $C_i \cup C_j$ is created, with weight $d(C_i, C_j)$. Two edges are added so that it becomes the parent of the vertices corresponding to C_i and C_j.

Algorithm 1: Hierarchical Tree

```
Input: $\mathcal{X} = \{x_1, \ldots, x_n\} \subset \mathbb{R}^m$
Output: Hierarchical Tree $H = (V, \omega_V, E)$
Initialize a set $\mathcal{C} = \{C_1, \ldots, C_n\}$ with clusters $C_i = \{x_i\}$, $\forall i \in \{1, \ldots, n\}$ and define a
  weighted graph $H$ with $n$ isolated vertices, one for each cluster in $C$, with weight
  $\omega_{\{x_i\}} = 0$.
for $\ell \in (1, \ldots, n-1)$ do
    $(U, W) = \arg\min\{d(A, B) : A, B \in C\}$
    $d^* = d(U, W)$.
    Redefine $\mathcal{C} = (\mathcal{C} \setminus \{U, W\}) \cup \{U \cup W\}$.
    Add a vertex corresponding to $U \cup W$ to $H$, connect it to $U$ and $W$, and assign the
      weight $\omega_{U \cup W} = d^*$.
end
```

Each pair of points x_i and x_j in the data set is connected by a path p_{ij} in the hierarchical tree constructed by Algorithm 1. Let V_{ij} be the set of vertices on the path p_{ij}. We use it to define a Gaussian kernel correction parameter $\gamma_{ij} = (|p_{ij}| - 2)/(\sum_{x \in V_{ij}} \omega_x + ||x_i - x_j||)$. The hierarchical similarity measure between a pair of points is defined as:

$$s_h(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2}\gamma_{ij}^2\right), \text{ if } i \neq j. \tag{2}$$

This similarity measure may be viewed as replacing $1/\sigma^2$ by γ_{ij}^2 in (1). Spectral Clustering with Hierarchical Similarity (SC-HA) refers to the framework of Section 2 with this similarity measure. This new similarity measure has the following main features:

(1) No scaling parameter needs to be set manually, since γ_{ij} is computed directly from the data set. This makes it more user-friendly.

(2) The spectral clustering algorithm based on this similarity is faster than algorithms that run for several values of the scaling parameter, as the eigenvectors need only be computed once.

(3) The hierarchical similarity is clearly invariant under translations and expansions of the data set. Moreover, given data sets $\mathcal{X} = \{x_1, \ldots, x_n\}$ and $\mathcal{X}' = \{x_1', \ldots, x_n'\}$, where $x_i' = c_1 x_i + c_2$, for some fixed $c_1 \neq 0$, it is easy to see that $s_h(x_i, x_j) = s_h(x_i', x_j')$ for every i, j.

We note that computing the hierarchical tree based on the data set and computing the parameters γ_{ij} is relatively inexpensive for the spectral clustering method. Indeed, it is computationally cheaper than computing the eigenvectors associated with the smallest eigenvalues of the normalized Laplacian matrix (see step (B) of the framework described in Section 2).

5 Experiments on synthetic data sets

We have applied the algorithms described in Sections 3 and 4 to five synthetic data sets that are often used to evaluate the performance of spectral methods. The algorithms were written and run using Python in a personal computer.

When choosing the value of the parameter σ in a spectral clustering algorithm, several authors have suggested to look for σ in a range between 10% and 20% of the total range of the Euclidean distances, see [9]. More precisely, given a data set $\mathcal{X} = \{x_1, \ldots, x_n\}$, we define the vector $d = (d_1, \ldots, d_N)$ containing the distances between each pair of points in $\mathcal{X}$, in ascending order. The values of σ used in our applications of algorithms SC-GK and SC-DA are $\sigma = d_u$ for $u = 1, \ldots, \lceil \frac{N}{5} \rceil$. Regarding to other parameters, when running the algorithm SC-ST, the parameter ℓ is tested for all integers between 2 and 20. For SC-DA, the parameter ϵ is set as $\epsilon = \max_i \min_j ||x_i - x_j||$.

The results for SC-GK, SC-ST, SC-DA and SC-HA are depicted in Figures 2 to 5, respectively, where each figure contains the results of one the methods for all five data sets. For simplicity, we refer to $F_j^{(k)}$ to mean the data set at column j in Figure k. From left to right, the data sets are a circle with two clouds, two circles with noise, two moons, three circles and two clouds with different scales. As usual, the individual objects that form each of the data sets are generated separately and the algorithms are given the task to identify each individual object. They receive the number of objects as an input.

The data sets $F_3^{(k)}$ and $F_4^{(k)}$ are data sets where SC-GK is known to work perfectly. The data set $F_5^{(k)}$ contains data in two different scales, which tends to be very challenging for spectral algorithms. The data set $F_1^{(k)}$ becomes harder as the data points in the clouds are chosen closer to the points on the circle. The data set $F_2^{(k)}$ contains noise, which is challenging for any clustering algorithm.

The algorithm SC-GK found the expected clustering in 3 out of 5 situations (see Figure 2). For the data sets $F_1^{(2)}$ and $F_5^{(2)}$, no σ in our domain has led to the correct result. Regarding the former, choosing a single value for σ makes it hard to distinguish between clusters that are close

DOI: 10.5540/03.2022.009.01.0263

to each other. Regarding the latter, it is well known that data sets whose clusters have multiple sizes and local densities are difficult for SC-GK [9].

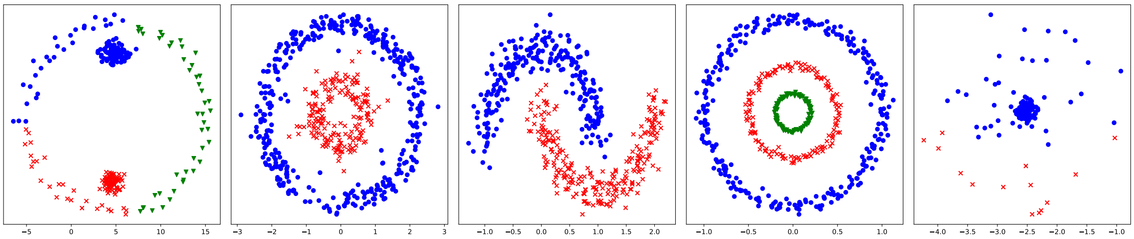

Figure 2: Results for each data set using SC-GK [5]. From left to right, the best results were achieved for $\sigma = 0.9$, $\sigma = 0.1$, $\sigma = 0.05$, $\sigma = 0.075$ and $\sigma = 0.5$, respectively. Source: The authors (2022).

The algorithm SC-DA obtained the correct clustering in four of the data sets, as depicted in Figure 3. We would like to point out that, except for $F_5^{(3)}$, the range of good values of σ increased, as expected.

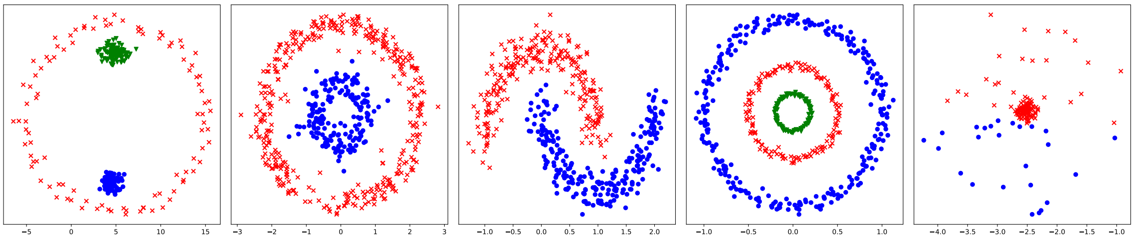

Figure 3: Results for each data set using SC-DA [9]. From left to right, we used $\sigma = 0.45$, $\sigma = 0.1$, $\sigma = 0.1$, $\sigma = 0.075$ and $\sigma = 1.5$, respectively. Source: The authors (2022).

The algorithm SC-ST found the correct clustering in three of the data sets, as presented in Figure 4. It seems to be more sensitive to noise, as it failed for $F_2^{(4)}$ and $F_3^{(4)}$.

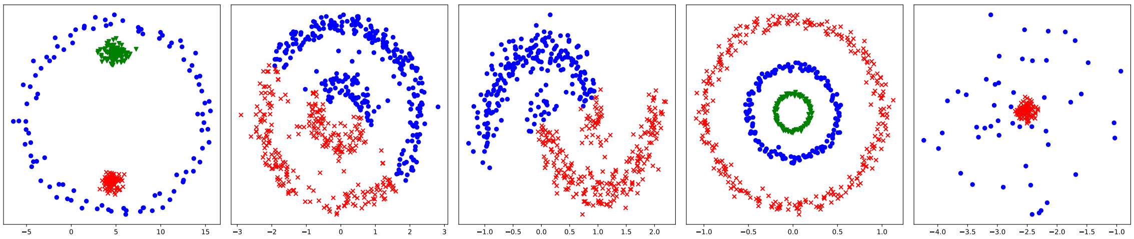

Figure 4: Results in each data set using SC-ST [8]. On the top, from left to right, we used $\ell = 7$, $\ell = 7$, $\ell = 7$, $\ell = 7$ and $\ell = 3$, respectively. Source: The authors (2022).

The algorithm proposed in this paper, SC-HA, had a good performance for all five data sets, as depicted in Figure 5. It successfully dealt with noise ($F_2^{(5)}$), multiple scales ($F_5^{(5)}$) and more complex shapes ($F_1^{(5)}$). In $F_1^{(5)}$ exactly two points in the data set (out of 300) have been added to the wrong cluster and in $F_3^{(5)}$ a single point (out of 500) has been classified incorrectly.

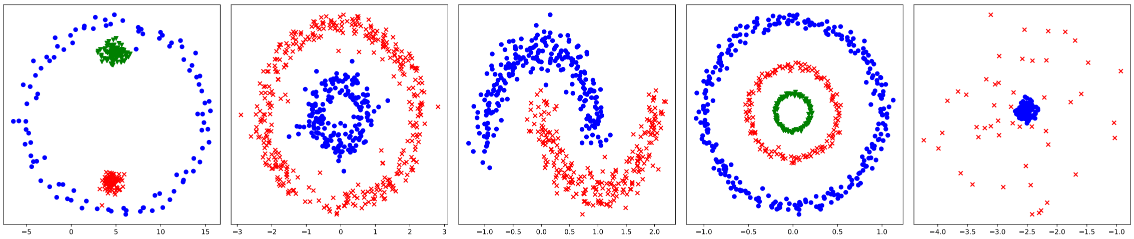

Figure 5: Results in each data set using SC-HA. Source: The authors (2022).

6 Conclusion

Spectral clustering algorithms are widely used for their practical performance [2]. In this paper, we propose a similarity measure for spectral clustering that incorporates a hierarchical component to the Gaussian kernel similarity measure. The spectral algorithms in the literature typically use a scaling parameter that has to be set by the user. Finding a good value of this parameter is often challenging, and may be a daunting task for users that do not have a lot of experience with the inner workings of clustering algorithms. Our method does not require such a scaling parameter. In comparison with other traditional spectral algorithms, our approach performed well on synthetic data sets, being able to find the correct clustering for data sets with complex shape and multiple scales. Even though we have not addressed this in this paper, it turns out that this approach has shown competitive results on real data sets extracted from machine learning repositories. Thus, we consider this a promising approach that warrants further investigation.

References

[1] H. Chang and D. Yeung. “Robust path-based spectral clustering”. In: **Pattern Recognition** 41.1 (2008), pp. 191–203. DOI: 10.1016/j.patcog.2007.04.010.

[2] H. Jia, S. Ding, X. Xu, and R. Nie. “The latest research progress on spectral clustering”. In: **Neur. Comput. and Appl.** 24.7 (2014), pp. 1477–1486. DOI: 10.1007/s00521-013-1439-2.

[3] U. von Luxburg. “A Tutorial on Spectral Clustering”. In: **Statistics and computing** 17.4 (2007), pp. 395–416. DOI: 10.1007/s11222-007-9033-z.

[4] J. Macqueen. “Some methods for classification and analysis of multivariate observations”. In: **5-th Berkeley Symposium on Mathematical Statistics and Probability** (1967), pp. 281–297.

[5] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. “On Spectral Clustering: Analysis and an Algorithm”. In: MIT Press, 2001, pp. 849–856. DOI: 10.5555/2980539.2980649.

[6] J. Shi and J. Malik. “Normalized cuts and image segmentation”. In: **IEEE Trans. Pattern Anal. Machine Intell.** 22.8 (2000), pp. 888–905. DOI: 10.1109/34.868688.

[7] R. Sibson. “SLINK: An optimally efficient algorithm for the single-link cluster method”. In: **The Computer Journal** 16.1 (1973), pp. 30–34.

[8] L. Zelnik-Manor and P. Perona. “Self-Tuning Spectral Clustering”. In: **Advances in Neural Information Processing Systems (NIPS)** 17 (2004). DOI: 10.5555/2976040.2976241.

[9] X. Zhang, J. Li, and H. Yu. “Local density adaptive similarity measurement for spectral clustering”. In: **Pattern Recognition Letters** 32 (2011), pp. 352–358. DOI: 10.1016/j.patrec.2010.09.014.