



HAL
open science

An Introduction to Gamma-Convergence for Spectral Clustering

Aditya Challa, Sravan Danda, B S Daya Sagar, Laurent Najman

► **To cite this version:**

Aditya Challa, Sravan Danda, B S Daya Sagar, Laurent Najman. An Introduction to Gamma-Convergence for Spectral Clustering. 2017. hal-01427957v1

HAL Id: hal-01427957

<https://hal.science/hal-01427957v1>

Preprint submitted on 6 Jan 2017 (v1), last revised 27 Jun 2017 (v2)

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.

An Introduction to Gamma-Convergence for Spectral Clustering

Aditya Challa¹, Sravan Danda¹, B.S. Daya Sagar¹, Laurent Najman²

¹ Systems Science and Informatics Unit, Indian Statistical Institute, 8th Mile, Mysore Road, Bangalore, 560059, India.

² Université Paris-Est, LIGM, Equipe A3SI, ESIEE, France.

Abstract. The problem of clustering is to partition the dataset into groups such that elements belonging to the same group are similar and elements belonging to the different groups are dissimilar. The unsupervised nature of the problem makes it widely applicable and also tough to solve objectively. Clustering in the context of image data is referred to as image segmentation. Distance based methods such as K-means fail to detect the non-globular clusters and hence spectral clustering was proposed to overcome this problem. This method detects the non globular structures by projecting the data set into a subspace, in which the usual clustering methods work well. Gamma convergence is the study of asymptotic behavior of minimizers of a family of minimization problems. Such a limit of minimizers is referred to as the gamma limit. Calculating the gamma limit for various variational problems have been proved useful - giving a different algorithm and insights into why existing methods work. In this article, we calculate the gamma limit of the spectral clustering methods, analyze its properties, and compare them with minimum spanning tree based clustering methods and spectral clustering methods.

1 Introduction

The problem of clustering is defined as - given a set of elements $\{x_i\}$, partition the set into non overlapping groups such that elements belonging to the same group are “similar”, and elements belonging to different groups are “dissimilar”. The importance of a solution to the problem of clustering is due to its wide ranging applications [14, 13, 1]. Clustering in image data is also referred to as image segmentation. There exists several methods to solve the problem of clustering [2, 9, 10, 17]. A comprehensive textbook on the subject is [1]. One of the most commonly used clustering methods is the K-means clustering [2, 16]. However, it suffers from the problem of not being able to detect the non-globular structures. Spectral clustering methods were proposed to overcome this problem. Loosely speaking spectral clustering methods embed the data in a lower dimensional subspace, in which usual methods K-means clustering would be able to detect the non globular clusters as well.

Recently, in [5], seeded clustering/segmentation methods in [6, 7, 11, 23] were extended by taking the limit of minimizers. This is referred to as the Γ -limit

[18]. Γ -convergence is a tool to study the asymptotic behavior of families of minimum problems [3]. The aim of Γ -convergence is to replace a family of minimum problems with a single problem whose minima exhibits some interesting properties. For instance in [5] it has been shown that the Γ -limit revealed an new segmentation methods which performs at least as well as graph cuts, random walker and shortest paths, if not better. In this article our aim is to calculate the Γ -limit of the ratio cut spectral clustering method and analyze the solution obtained.

2 Background

Let $\{v_i\}$ be the given set of points in \mathbb{R}^n which we would like to cluster. Taking each of these points as vertices, one can construct a similarity graph $\mathcal{G} = (V, E, W)$ with vertex set V , edge set E , and $W : E \rightarrow \mathbb{R}^+$ denotes weights assigned to each edge. Here \mathbb{R}^+ denotes the set of positive real numbers. With slight abuse of notation, we can write the weights as a $|V| \times |V|$ matrix, with w_{ij} denoting the edge weight between v_i and v_j . The degree of a vertex, d_i is given by

$$d_i = \sum_j w_{ij} \quad (1)$$

Let D be a diagonal matrix, $diag(d_1, d_2, \dots, d_n)$. The *Laplacian* of a graph is then defined by

$$L = D - W \quad (2)$$

We know that the Laplacian is a symmetric positive-semi definite matrix, and hence has non negative real eigenvalues, represented by $0 = \lambda_0 \leq \lambda_1 \leq \dots \leq \lambda_{n-1}$ [15]. The corresponding eigenvectors are denoted by $(e_0, e_1, \dots, e_{n-1})$. Let $A \subseteq V$. Then the vector $\mathbf{1}_A(x)$ is given by

$$\mathbf{1}_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Let p be a real parameter. Let $W^{(p)}$ be the matrix such that $W_{ij}^{(p)} = w_{ij}^p$. Let $D^{(p)}$ denote the matrix as constructed in (1) with weights w_{ij}^p . Let $L^{(p)} = D^{(p)} - W^{(p)}$.

2.1 Spectral Clustering

This section briefly reviews the spectral clustering methods. For more details please refer to [22, 24, 19]. As noted before, spectral clustering methods work by embedding the data into a lower dimensional subspace. The three main steps are - 1) Given a set of points $\{v_i\}$ (dataset), construct a similarity graph with each point as a vertex. 2) Construct the Laplacian for the obtained graph and calculate the first k eigenvectors. The value of k is fixed based on the number of clusters one would like to obtain. Let K be the matrix such that the i^{th}

column of K is the i^{th} eigenvector e_{i-1} . 3) Using rows of the matrix K as new representation of the points v_i , use traditional clustering methods such as K-means to obtain the final clusters. Note that as a part of K-means step, the algorithm is run several times with random initialization of seeds.

Why does spectral clustering work? Although the definitive answer to this question still remains open, there exists several analyses which provide insights into this question [24]. One approach is to interpret the spectral clustering in an optimization framework. One of the measures to validate the appropriateness of the clusters is

$$\text{cut}(A_1, A_2, \dots, A_k) = \frac{1}{2} \sum_{i=1}^k W(A_i, \bar{A}_i) \quad (4)$$

where $W(A, B) = \sum_{i \in A; j \in B} w_{ij}$, \bar{A} denotes the complement of A in the vertex set V . $\text{cut}(\cdot, \cdot)$ measures how dissimilar the clusters are by taking the sum of the weights of the edges connecting distinct clusters. In practice minimizing the $\text{cut}(\cdot, \cdot)$ does not give good results, since it generally separates one vertex, and gives degenerate solutions. To solve this, it was proposed to use a slight modification of the above cost function. *Ratio-cut*, [24], is given by

$$\text{Ratio-cut}(A_1, A_2, \dots, A_k) = \frac{1}{2} \sum_{i=1}^k \frac{W(A_i, \bar{A}_i)}{|A_i|} \quad (5)$$

where $|A_i|$ is the cardinality of set A_i . It can be shown that minimizing the *Ratio-cut* for k clusters is approximately equivalent to solving the following optimization problem.

$$\begin{aligned} & \underset{H \in \mathbb{R}^{n \times k}}{\text{minimize}} && \text{Tr}(H^t L H) \\ & \text{subject to} && H^t H = I \end{aligned} \quad (6)$$

Here I is the identity matrix and L is the laplacian as defined in (2). From the *Rayleigh-Ritz* theorem [15] we know that the solution to this optimization problem is obtained by considering the first k eigenvectors of L as columns of H .

2.2 Gamma Convergence

Let $\min\{F_p(x) : x \in X \subset \mathbb{R}^n\}$ be a family of minimum problems. Let x_p^* be a minimum of $F_p(x)$. We are interested in calculating the limit

$$x^* = \lim_{p \rightarrow \infty} \arg \min F_p \quad (7)$$

In other words, we are interested in the limit of a sequence of minimizers of the family $\{F_p(x)\}$. Note that there could be many such sequences. Now, consider a special case where

$$F_p(x) = \sum_{i=1}^n \lambda_i^p Q_i(x) \quad (8)$$

where $1 \geq \lambda_n > \lambda_{n-1} > \dots > \lambda_1 > 0.$, and $Q_i(x)$ are smooth functions. We also assume that there exists a compact set C such that x_p^* belongs to C for all p . It turns out that, in this case, one can find a simple algorithm to calculate the limit of minimizers as described in the following Thm. 1 [18]. Define

$$M_n = \arg \min Q_n(x) \quad x \in C \quad (9)$$

Recursively define for $k = n - 1, n - 2, \dots, 0$

$$M_k = \arg \min Q_k(x) \quad x \in M_{k+1} \quad (10)$$

Theorem 1. *Let $F_p(x)$ be as defined in (8) and x^* be the Γ -limit. Then $x^* \in M_1$.*

Refer to [18] for proof of the above statement. The main consequence of Thm. 1 is that it provides a method to calculate an “approximate” Γ -limit. One starts at the highest scale, optimizes the cost function at this scale, then moves on to the lower scale and repeats the process. Theorem 1 essentially states that one can obtain an approximate solution to the Γ -limit by this process. This gives a generic Alg. 1 to compute an approximate Γ -limit [18]. It is approximate since the theorem only guarantees that both the Γ -limit and the result calculated using the Alg. 1 are in the same set. The question of how good the approximation, needs to be analyzed in the specific case.

Algorithm 1 Generic Algorithm to Compute Γ -limit [18] .

Input: Function $F_p(x) = \sum_{i=1}^n \lambda_i^p Q_i(x)$, where $\lambda_n < \lambda_{n-1} < \dots < \lambda_1$.

Output: x^* : Gamma limit

- 1: $M_n = \arg \min Q_n(x)$ where $x \in C$
 - 2: **for** k from $n - 1$ to 1 **do**
 - 3: Compute $M_k = \arg \min Q_k(x)$ where $x \in M_{k+1}$
 - 4: **end for**
-

3 Gamma Limit of Ratio - Cut

In this section, we calculate the approximate Gamma limit of the Ratio-cut. A few more notation is required. For a given graph \mathcal{G} , let \mathcal{G}_k denote the graph (V, E_k, W_k) . $E_k \subseteq E$ denotes the set of edges whose weight is w_k . We refer to this graph as *scale graph/level graph* at level k . Just as for the original graph, one can construct a laplacian, L_k , for \mathcal{G}_k . A point to observe is that the entries in weight matrices of a level graph are either 1 or 0. We assume that the graph has distinct weights $w_1 < w_2 < \dots < w_j$, where $j < |E|$. Given the notation as above, we have

$$Tr(H^t L H) = \sum_{k=1}^j w_k Tr(H^t L_k H) \quad (11)$$

Thus, we have the following algorithm 2 to calculate the gamma limit. Let \mathcal{P}_k denote the following optimization problem.

$$\begin{aligned} & \underset{H \in \mathbb{R}^{n \times m}}{\text{minimize}} && \text{Tr}(H^t L_k H) \\ & \text{subject to} && H^t H = \mathbf{I} \end{aligned} \tag{12}$$

Algorithm 2 Generic Algorithm to Compute Γ -limit.

Input: A weighted graph, \mathcal{G} , with distinct weights $w_1 < w_2 < \dots < w_j$. Number of clusters to calculate m .

Output: M_1 - Γ -limit

- 1: Set $k = j$, $M_j = \arg \min \mathcal{P}_j$
 - 2: Set $k = k - 1$
 - 3: Construct the scale graph \mathcal{G}_k at level k , and laplacian L_k .
 - 4: Let M_k be the set of solutions for $\arg \min \mathcal{P}_k$ which belong to M_{k+1} .
 - 5: **if** $k = 1$ **then**
 - 6: **return** M_1
 - 7: **else**
 - 8: Goto Step (2)
 - 9: **end if**
-

Observe that the Alg. 2 is a restatement of Alg. 1 in our special case. However Alg. 2 is not implementable. In the rest of this section we give a brief overview of the procedure to get the implementable version and show that this algorithm gives a reasonable approximation. Note that the minimizer of the optimization problem in (12) need not be unique.

The main problem in finding an implementable version for Alg. 2 is to characterize all the solutions for the problem in (12). Let $\lambda_{(m)}$ denote the m^{th} smallest eigenvalue. Let A_k be the matrix $[e_1, e_2, \dots, e_l]$, where e_i is the i^{th} eigenvector, ordered in increasing order of the corresponding eigenvalue. The number of eigenvectors considered are the number of eigenvalues less than or equal to $\lambda_{(m)}$. Let K be the matrix

$$K = \begin{bmatrix} I_{l_1 \times l_1} & 0 \\ 0 & k_{l_2 \times m - l_1} \end{bmatrix} \tag{13}$$

where $K^t K = I$, l_2 is the number of eigenvalues equal to $\lambda_{(m)}$, and $l = l_1 + l_2$. Let X be an orthogonal matrix such that $X^t X = I$. Then,

Proposition 1. *The set of all solutions to the optimization problem in (12) is of the form $A_k K X$.*

Proposition 1 results in an algorithm which is implementable. However, one requires to calculate all the eigenvectors at every stage, which is computationally expensive. Proposition 2 results in an efficient implementation of the algorithm as given in Alg. 3. We refer the result obtained by this algorithm as *Power Recut*.

Algorithm 3 Efficient algorithm to compute Γ -limit for ratio-cut.

Input: A weighted graph, \mathcal{G} , with distinct weights $w_1 < w_2 < \dots < w_j$. Number of clusters, m .

Output: N - A representation of the subspace spanned by the Γ -limit of the minimizers.

- 1: Set $k = j$.
- 2: **while** Number of connected components of $\mathcal{G}_{\geq k}$ is greater than or equal to m **do**
- 3: Set $k = k - 1$ {We refer to this as an MST-Phase}
- 4: **end while**
- 5: Construct N by stacking the vectors $\mathbf{1}_{A_i}/\sqrt{|A_i|}$ in columns, where A_i is a connected component of $\mathcal{G}_{\geq k}$.
- 6: Set $l_1 = 0$ and $l_2 =$ number of connected components in $\mathcal{G}_{\geq k}$
- 7: Consider the graph \mathcal{G}_k and let L_k be the corresponding laplacian.
- 8: Set $C = [N^t L_k N]_{l_2, l_2}$
- 9: Calculate the first eigenvectors of eigenvalue problem whose eigenvalue is less than or equal to $\lambda_{(m)}$.

$$Cx = \lambda x \quad (14)$$

10: Let A be the matrix obtained by stacking the eigenvectors as columns.

11: Construct \hat{A} as

$$\hat{A} = \begin{bmatrix} \mathbf{I} & 0 \\ 0 & A \end{bmatrix} \quad (15)$$

- 12: Update l_1 and l_2 .
 - 13: $N = N \times \hat{A}$
 - 14: Set $k = k - 1$
 - 15: **if** $k = 0$ or number of columns of N is equal to m **then**
 - 16: **return** N
 - 17: **else**
 - 18: Goto Step (7)
 - 19: **end if**
-

Proposition 2. *Given a graph \mathcal{G} , Let $\mathcal{G}_{\geq k}$ denote the graph, whose vertex set is V and edge set E_k containing all the edges whose weight is greater than or equal to w_k . At stage k , if A is a maximal connected component of $\mathcal{G}_{\geq k}$, then $\mathbf{1}_A$ is an eigenvector with eigenvalue 0 of the optimization problem (12).*

The proofs for the propositions and details of simplification from Alg. 2 to the efficient Alg. 3 will be discussed in detail in a full version of the paper. Recall that the Alg. 1 only gives a element of the set to which the Γ -limit actually belongs, and Alg. 3 is an extension of Alg. 1 in the specific case of Ratio-cut. The appropriateness of the solution thus obtained must be proved, which the Prop. 3 gives.

Proposition 3. *Let x be the solution obtained by algorithm 3 and let x^* be a Γ -limit. Then*

$$F_p(x^*) = F_p(x) \quad \text{for all } p \quad (16)$$

Where F_p is as given in (8), where $Q_i(x) = \text{Tr}(x^t L_i x)$.

The above proposition implies that any Γ -limit, and the approximate one calculated have the same cost. In the context of spectral clustering, this implies that the solutions are equivalent and thus the solution obtained from Alg. 3 is a good approximation. The proof of the above proposition will be detailed in later work.

4 Analysis with Experiments

In the rest of the article, we discuss how the algorithm works, its practical implementation and its similarities and dissimilarities with the MST based clustering and spectral clustering methods.

4.1 How the algorithm works?

Recall that the output of spectral clustering is a projection onto a subspace, and thus the algorithm produces a representation of the points in a subspace (denoted by N in algorithm 3) which is a gamma limit. We assume that we need m clusters. In steps 2-4, we progressively add all the edges while there are at least m connected components in the threshold graph $\mathcal{G}_{\geq k}$. Thanks to Prop. 2 we know the first eigenvectors of the laplacian of the threshold graph are the indicator vectors as in (3) where A denotes each of the connected components. We construct the initial representation of the points N by taking the eigenvectors in step 5. Steps 7-14 updates the representation N with respect to the lower weight edges. Note that, once the number of columns of the matrix N is equal to m , we need not update the representation anymore, since any other representation would just be an orthogonal transformation of the points and thus the clustering results would not change. This condition is checked in steps 15-19.

One issue with the gamma limit is that the property of non-trivial clusters for the Ratio-cut is not preserved in the limit. In the above algorithm, in practice, we get a lot of outliers and thus results in small clusters. To avoid this, in steps 2-4, instead of calculating the number of connected components, we calculate the number of connected components whose size is greater than a given parameter *threshComp*. At the moment the algorithm jumps out of the while loop, all the components which have less than *threshComp* number of vertices are ignored. Recall that after the representation of the points is obtained, one has to perform K-means clustering to get the final partition. At this stage the ignored “small” clusters can either be returned as a different cluster, or combined with one of the larger clusters at random. An important note about the exposition is that - we stick to the philosophy that theory is developed to be as general as possible, while the experiments are conducted with slightly modified theory based on practical considerations such as “small” clusters described above.

A simple application of the algorithm is illustrated in Fig. 1, where one can see that the algorithm correctly detects the object (flower in this case). Since small clusters are ignored, for the applications of image segmentation we do not get closed contours for the segments. Thus, for image segmentation the

clusters are post-processed with an operator. Observe in Fig. 1(b) a few parts of the flower are missing. In this case, simple operator such as an opening works well, which gives the result as in Fig. 1(c). In general connected operators [21] preserve contours and are better for the post-processing of the image. Another important property of the Power Rcut clustering method is that, it results in smooth contours compared to the spectral clustering method. This is illustrated in Fig. 1 (d) - (f).



Fig. 1. (a) Original Image (Flower) (b) Power Rcut segment result (c) Power Rcut segment result post processing with opening (d) Original Image (Cameraman) (e) Contours obtained by Power R cut and (f) Contours obtained by Normalized spectral clustering.

4.2 Relation to MST-clustering

MST (Maximum Spanning Tree) based clustering is one of the earliest graph based clustering approach [25, 12, 20, 4]. There exists several variations of the method. We consider here the simplest method - (a) Construct an MST (b) Iteratively remove the least weight edges until we get the required number of clusters. One of the most useful properties of spectral clustering is its ability to detect non-convex clusters in the data. This property is shared by the MST based clustering methods as well.

However, the problem with MST based clustering is breaking ties between edges of equal weight, which it does arbitrarily. Spectral clustering on the other

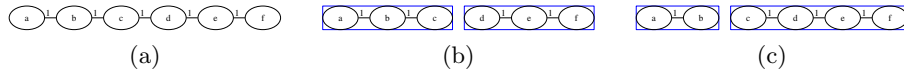


Fig. 2. (a) Basic Graph (b) Power Rcut Clustering (c) MST based clustering. Observe that Power Rcut splits the graph into two equal halves, where as MST based clustering does not.

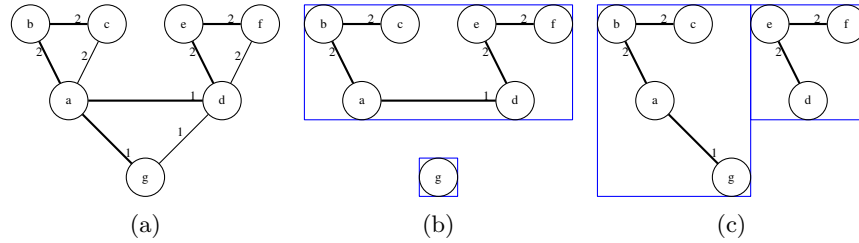


Fig. 3. (a) Basic Graph and MST edges highlighted in bold (b) Clustering with MST (c) Power Rcut Clustering. Observe that Power Rcut clustering tries to split the graph into equal parts.

hand ensures a single clustering (up to the arbitrariness of k-means step). In this sense, Power Rcut can be considered to be a method between these two clustering methods. Power Rcut follows a similar procedure as MST based clustering, and it breaks ties with spectral clustering on a subgraph. For example consider the graph in Fig. 2(a). Power Rcut segments the graph into two equal clusters, Fig. 2(b). The same behavior is also exhibited by spectral clustering. MST based clustering on the other hand segments the graph into non-equal parts since it breaks the ties arbitrarily, Fig. 2(c).

This is because, Power Rcut takes into consideration the sizes of the cluster while breaking ties. Consider another synthetic example in Fig. 3(a). An MST is highlighted in bold edges in Fig. 3(a). Since MST based clustering would break the ties randomly, it could result in clusters as in 3(b). Power Rcut clustering on the other would definitely not give clusters as in 3(b). An example Power Rcut clusters is given in 3(c).

Proposition 4. *If at a threshold t , we have that $\mathcal{G}_{\geq t}$ has exactly m clusters, then MST clustering and Power Rcut clustering results in the same clusters.*

Power Rcut clustering and MST based clustering are in fact closely related as suggested by the Prop. 4. The proof follows from the following observation - for any two vertices belonging to the same connected component has the same value in the embedded space, and hence belong to the same cluster. Since, we require m clusters, and there are m connected components, the k-means step results in each of these m components as a cluster.

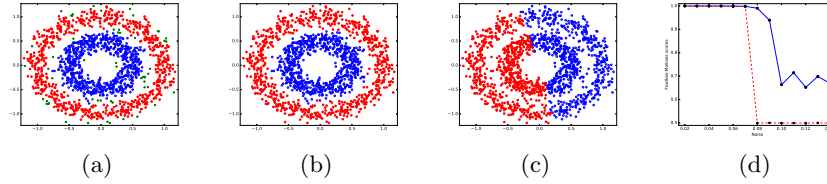


Fig. 4. Data is sampled from two concentric circles with noise. The clusters obtained are represented by red dots and blue circles. (a) Power Rcut clustering. The green stars indicate the “noise” clusters obtained. (b) Power Rcut clustering obtained by adjusting the noise clusters to the nearest cluster. (c) Spectral clustering (d) Fowlkes-Mallows scores for varying noise. The dashed line indicates spectral clustering score and continuous line indicate Power Rcut score. The higher the score, the better the procedure.

4.3 Relation to Spectral Clustering

The Power Rcut solution can also be interpreted as being obtained by spectral clustering on every level graph \mathcal{G}_k . Since, the first few eigenvectors are the indicator of the connected components, this gives a heuristic explanation for steps 2-4 in algorithm 3. This points out the similarity between Power Rcut solution and spectral clustering.

In low noise conditions, spectral clustering and Power Rcut clustering results are similar. However, as noise level increases, spectral clustering will not be able to identify the regions anymore. In Fig. 4(a),(b) and (c) data points are sampled from two concentric circles with noise. Fig. 4(c) shows the results obtained by spectral clustering are shown. Notice that the structure of the two circles is not preserved. Fig. 4(a) show the results obtained with Power Rcut. To generate the results of Power Rcut, as a post processing step, we assign each of the points the “small” clusters to the closest cluster, which results in Fig. 4(b) . Although not perfect, Power Rcut results preserve some structure. For a quantitative view we calculate the Fowlkes-Mallows(FW) scores, given by the formula

$$Score = \frac{TP}{\sqrt{(TP + FP)}\sqrt{(TP + FN)}} \quad (17)$$

Where TP is true positives, FP is false positives and FN is false negatives. The FW scores for various noise levels is plotted in Fig. 4(d). Note that Power Rcut performs better in high noise scenarios compared to spectral clustering.

5 Conclusion And Future Work

In this article we outlined the basics of spectral clustering and discussed the concept of Γ -convergence. Important results to calculate the Γ -limit of Ratio-cut were outlined and the algorithm to calculate the gamma limit was obtained.

The correctness of the algorithm was shown via a proposition. The similarities and dissimilarities between Power Rcut, MST clustering and Spectral clustering are analyzed stating the result that Power Rcut clustering is a specific kind of MST clustering. Power Rcut clustering was shown to be superior to MST clustering in dealing with ramp effects of the image, and superior to spectral clustering in noisy scenarios.

Note that none of the steps in spectral clustering methods indicate directly as to why spectral clustering methods could obtain non-convex clusters. This has been an open question for a long time. We believe that, the fact that gamma limit of spectral clustering is a method close to MST clustering provides an insight into this question. In particular, this provides a bridge between spectral clustering and MST clustering and allows us to dive into the question of why spectral methods work.

Note that the algorithm starts with combining the edges of highest weight until the number of clusters are obtained. This can be interpreted as a greedy method of “combining all the points which definitely belong to the same cluster”, thus reducing the size of the dataset and allowing for faster computation. This in theory can also be done in parallel. This can allow spectral clustering to be applicable in the case of large datasets and also have efficient implementations. The error bounds and the exact algorithm to do this are a subject of future research.

In [8], the authors study in depth hierarchies and their equivalence with MST. Observe that the Alg. 3 is inherently hierarchical. The question of how Alg. 3 is related to the concepts in [8] is also a subject of future research.

6 Acknowledgements

AC and SD would like to thank Indian Statistical Institute. This work has been partly funded by ANR-15-CE40-0006 CoMeDiC and ANR-14-CE27-0001 GRAPHSIP research grants. BSDS would like to acknowledge the support received from the Science and Engineering Research Board (SERB) of the Department of Science and Technology (DST) with the grant number EMR/2015/000853, and the Indian Space Research Organization (ISRO) with the grant number ISRO/SSPO/Ch-1/2016-17.

References

1. Aggarwal, C.C., Reddy, C.K.: Data Clustering: Algorithms and Applications. Chapman & Hall/CRC, 1st edn. (2013)
2. Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. pp. 1027–1035. Society for Industrial and Applied Mathematics (2007)
3. Braides, A.: Gamma-convergence for Beginners, vol. 22. Clarendon Press (2002)
4. Chowdhury, N., Murthy, C.: Minimal spanning tree based clustering technique: Relationship with bayes classifier. Pattern Recognition 30(11), 1919–1929 (1997)

5. Couprie, C., Grady, L., Najman, L., Talbot, H.: Power watershed: A unifying graph-based optimization framework. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33(7), 1384–1399 (2011)
6. Cousty, J., Bertrand, G., Najman, L., Couprie, M.: Watershed cuts: Minimum spanning forests and the drop of water principle. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31(8), 1362–1374 (2009)
7. Cousty, J., Bertrand, G., Najman, L., Couprie, M.: Watershed cuts: Thinnings, shortest path forests, and topological watersheds. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(5), 925–939 (2010)
8. Cousty, J., Najman, L., Kenmochi, Y., Guimares, S.: Hierarchical segmentations with graphs: quasi-flat zones, minimum spanning trees, and saliency maps. Research report, LIGM (Jul 2016), <https://hal.archives-ouvertes.fr/hal-01344727>
9. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)* pp. 1–38 (1977)
10. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Kdd*. vol. 96, pp. 226–231 (1996)
11. Grady, L.: Random walks for image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28(11), 1768–1783 (2006)
12. Grygorash, O., Zhou, Y., Jorgensen, Z.: Minimum spanning tree based clustering algorithms. In: 2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06). pp. 73–81. IEEE (2006)
13. Jain, A.K.: Data clustering: 50 years beyond k-means. *Pattern recognition letters* 31(8), 651–666 (2010)
14. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. *ACM computing surveys (CSUR)* 31(3), 264–323 (1999)
15. Lütkepohl, H., Lutkepohl, H.: *Handbook of Matrices*. Wiley, 1 edn. (1997)
16. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. vol. 1, pp. 281–297. Oakland, CA, USA. (1967)
17. McLachlan, G., Peel, D.: *Finite mixture models*. John Wiley & Sons (2004)
18. Najman, L.: Extending the powerwatershed framework thanks to γ -convergence. Tech. rep., Université Paris-Est, LIGM, ESIEE Paris (2017), <https://hal-upec-upem.archives-ouvertes.fr/hal-XXXXXX>
19. Ng, A.Y., Jordan, M.I., Weiss, Y., et al.: On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems* 2, 849–856 (2002)
20. Peter, S.J.: Minimum spanning tree-based structural similarity clustering for image mining with local region outliers. *International Journal of Computer Applications* (0975–8887) Volume (2010)
21. Serra, J.: A lattice approach to image segmentation. *Journal of Mathematical Imaging and Vision* 24(1), 83–130 (2006)
22. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence* 22(8), 888–905 (2000)
23. Sinop, A.K., Grady, L.: A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In: *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. pp. 1–8. IEEE (2007)
24. Von Luxburg, U.: A tutorial on spectral clustering. *Statistics and computing* 17(4), 395–416 (2007)
25. Zahn, C.T.: Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on computers* 100(1), 68–86 (1971)