Locally-Scaled Spectral Clustering using Empty Region Graphs

Carlos D. Correa
Center for Advanced Scientific Computing
Lawrence Livermore National Laboratory
correac@llnl.gov

Peter Lindstrom
Center for Advanced Scientific Computing
Lawrence Livermore National Laboratory
pl@llnl.gov

ABSTRACT

This paper introduces a new method for estimating the local neighborhood and scale of data points to improve the robustness of spectral clustering algorithms. We employ a subset of empty region graphs – the β-skeleton – and non-linear diffusion to define a locally-adapted affinity matrix, which, as we demonstrate, provides higher quality clustering than conventional approaches based on k nearest neighbors or global scale parameters. Moreover, we show that the clustering quality is far less sensitive to the choice of β and other algorithm parameters, and to transformations such as geometric distortion and random perturbation. We summarize the results of an empirical study that applies our method to a number of 2D synthetic data sets, consisting of clusters of arbitrary shape and scale, and to real multi-dimensional classification examples from benchmarks, including image segmentation.

Categories and Subject Descriptors

I.5.3 [**Clustering**]: Algorithms; H.2.8 [**Database applications**]: Data mining

Keywords

Spectral Clustering, Proximity Graphs

1. INTRODUCTION

Clustering is at the core of modern data mining tools. Common techniques, such as those based on K-means or explicit density models, are being replaced by *spectral methods* for clustering, where points are clustered based on a spectral analysis of a matrix of pairwise similarities or affinities, instead of relying on a particular cluster model.

Spectral clustering has been applied successfully in a number of fields, including image segmentation, text mining, and data analysis in general. However, there remain a number of open questions: (1) How to define the neighborhood around data points to estimate a "good" affinity matrix, (2) how to adapt the algorithm to account for variations in local scale or density of the data, and (3) how to automatically select the number of clusters. This paper concerns the former two questions.

The most common approaches to date rely on simple neighborhood definitions, such as k-nearest neighbor (kNN) graphs or ε-graphs, both which involve parameters that govern the graph density. However, clustering results may change dramatically for different values of k or ε. In particular, when the neighborhood graph is too sparse, clusters break up into individual components that cannot be aggregated by spectral clustering, while the spectral method can be unreliable for identifying clusters in an overconnected, dense graph. Alternatively, one can simply connect all points in a fully connected graph and rely on a single scale parameter σ to define the affinity between pairs of points in a weighted graph, which can be thought of as a fuzzy instance of ε-graphs. In either case, the optimal choice of these parameters varies with the dimensionality and across data sets, and more importantly often *within* a data set, as the resulting selection of neighbors or affinities does not adapt to the local density or distribution of points.

In this paper, we exploit *empty region graphs* (ERGs) to construct neighborhoods without requiring a particular choice of the neighborhood extents. In contrast to kNN graphs, which prescribe the number of neighbors without regard to their relative locations, empty region graphs account for the spatial distribution of points to define neighborhoods of varying extent and cardinality, and do in general not suffer from missing or redundant connections. We show that these graphs generally improve the accuracy of spectral clustering algorithms. We also introduce a diffusion-based mechanism that estimates the density based on the average neighborhood size around a point to define affinities. We show that this local scaling algorithm, when combined with empty region neighborhoods, results in a better classification that is robust to noise and geometric transformations of the data points.

We present results on a number of synthetic benchmark data sets, as well as real multi-dimensional classification problems, including image segmentation.

2. RELATED WORK

Spectral Clustering. Spectral clustering is becoming a successful alternative to techniques based on K-means [21] or density models [10], and dates back to Donath and Hoffman [8] and Fiedler [11]. Recently, spectral clustering has found a niche in image segmentation [28], text mining [7] and as a data mining tool in general [26, 17]. Since then, there has been a trend in improving spectral clustering through a detailed analysis of the underlying graph structure [22], the scale and density parameters [34, 1], and the stability [15] and consistency [33] of the algorithm. Most related to our work are the techniques that attempt to estimate the local scale or density to improve spectral clustering of data with varying densities, shapes and levels of noise. Among the first to address this problem for data mining were Zelnik-Manor and Per-

KDD'12, August 12–16, 2012, Beijing, China.
Copyright 2012 ACM 978-1-4503-1462-6 /12/08. ...$10.00.

ona [34], who improve the general algorithm by Ng et al. [26] with local scaling. This approach, although effective even in high dimensions, was shown to be suboptimal for noisy data sets, or for data with clusters of different densities [25]. To alleviate this problem, Nadler and Galun [25] introduce a coherence measure of a set of points belonging to the same cluster. Although not exclusive of spectral methods, the authors show that it alleviates some of the intrinsic limitations of spectral clustering. To deal with noise, Li et al. [19] propose a warping model that maps the data into a new space more suitable for clustering and more resilient to noise. Other methods are able to cluster data consisting of regions of arbitrary shapes, such as density based clustering [27], and, in a similar spirit to Zelnik-Manor and Perona's method, locally scaled density based clustering [1].

In this paper, we address the problem of locally-scaled and noise robust spectral clustering. We take a different approach and identify the problem as early as the selection of the neighborhood graph. Maier et al. suggest that the construction of the graph has a measurable effect on the results of spectral clustering [22]. Inspired by this paper, we turned to alternative neighborhood graphs, namely *empty region graphs* [3], in an effort to obtain better neighborhoods.

Empty Region Graphs. Neighborhood or proximity graphs create a geometric structure that connects two points if they are close in some sense. These graphs have been well studied and include the relative neighborhood graph [16], the Gabriel graph [14], β-skeletons [18], σ-local graphs [2] and Delaunay triangulations [12]. A subset of these, called the empty region graphs, define a neighborhood graph where two points are connected if a geometric region parameterized by those points does not contain any other point [3]. These graphs have been well studied in terms of their geometric properties [6, 3], and have been applied in geographic analysis, pattern recognition and machine learning. Proximity graphs have been applied to clustering as well. Urquhart et al. [32] use the Gabriel graph and the Relative Neighbor graph to improve hierarchical clustering, noting that these graphs result in natural clusters that can be separated depending on the local graph density [32]. Carreira and Zemel apply an ensemble of minimum spanning trees to form neighborhood graphs that are more resilient to noise and varying densities [4]. Choo et al. propose an agglomerate method for hierarchical clustering that merges candidate clusters that belong to the same connected component in the Gabriel graph [5].

In this paper, we propose the use of the one-parameter β-skeleton empty region graph to construct locally-scaled affinity matrices that improve the accuracy of spectral clustering. We show that this approach is more effective when combined with a diffusion step that enhances the block structure of the affinity matrix and, thus, the separability of clusters.

3. BACKGROUND

Our approach combines neighborhoods defined by empty region graphs with density estimation techniques and spectral clustering.

3.1 Spectral Clustering

Spectral clustering refers to a general algorithm where data are clustered into groups based on spectral analysis of a matrix of pairwise affinities or similarities between data points. The intuition is that, based on a similarity graph between points, a good clustering should partition the graph such that points in the same group are similar and points in different groups are dissimilar to each other. The spectral properties of the graph Laplacian helps us partition the graph in that manner, based on one of the properties of the graph Laplacian, which states the graph Laplacian has eigenvalue 0 with multiplicity equal to the number of connected components of the affinity matrix.

A general algorithm for spectral clustering can be implemented as follows [26]: Given a set of n points $S = \{s_1, s_2, \ldots, s_n\}$ in $\mathbb{R}^d$ to be partitioned into K clusters,

- Construct a neighborhood graph (S,E) (e.g. based on k nearest neighbors) over the point set S.
- Define the affinity matrix $A \in \mathbb{R}^{n\times n}$, where

$$A_{ij} = \begin{cases} \exp\left(\frac{-d(s_i,s_j)^2}{\sigma^2}\right) & ij \in E \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

  d is a distance function, commonly Euclidean, and σ is a scale parameter.
- Define D as the diagonal matrix $D_{ii} = \sum_{j=1}^{n} A_{ij}$.
- Define the normalized Laplacian matrix $L = I - D^{-1/2}AD^{-1/2}$.
- Find the K eigenvectors corresponding to the smallest eigenvalues of L, and form the matrix $X \in \mathbb{R}^{n\times K}$ with these eigenvectors as columns.
- Form the matrix Y by normalizing the rows of X, so that $Y_{ij} = X_{ij}/\sqrt{\sum_j X_{ij}^2}$.
- Treat each row of Y as a point in $\mathbb{R}^K$ and cluster via K-means [21].
- Each point s_i is assigned to a given cluster c if the corresponding row i in Y is assigned to cluster c.

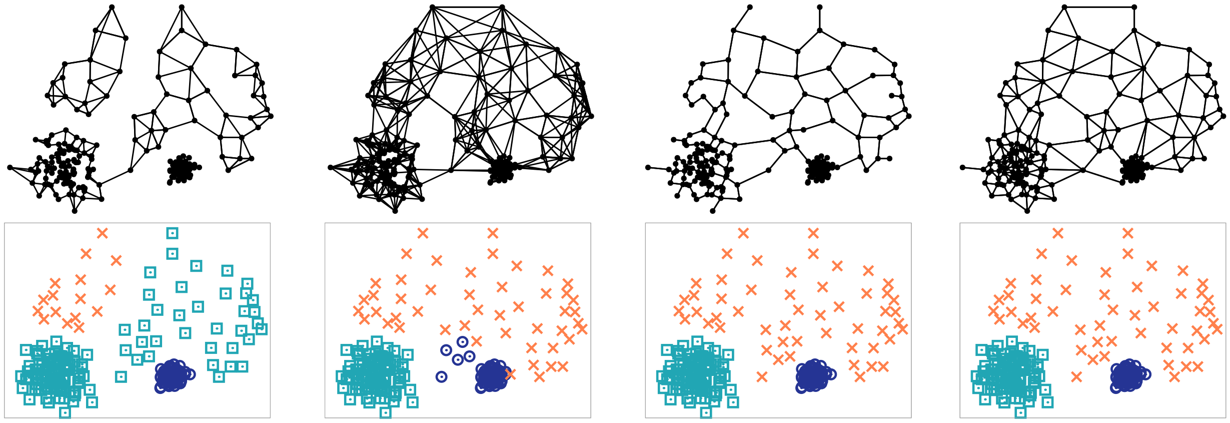
Clearly, the accuracy of the clustering depends, among other factors, on the graph density k and the scale parameter σ. Fig. 1 shows how $k = 3$ and $k = 7$ nearest neighbor graphs with a fixed scale parameter are used to cluster groups of 2D points, two of which define small dense clusters, while the third forms a sparse background. When k is small (Fig. 1(a)), important connections are missing within the background cluster, which is separated into two components. Increasing k helps connect the background cluster (Fig. 1(b)), but adds many redundant edges between clusters that diffuse them and cause the algorithm to misclassify background points. Varying k and σ together may improve the results, though when the point density varies significantly, it may be that no combination of k and σ yields the correct clustering.

To deal with disparate densities, Zelnik-Manor and Perona define a more general affinity that incorporates local scaling [34]. Instead of a single scale parameter, they define the affinity between two points as:

$$A_{ij} = \exp\left(\frac{-d(s_i,s_j)^2}{\sigma_i\sigma_j}\right) \quad (2)$$

where σ_i and σ_j are the local scale parameters estimated for points s_i and s_j, respectively. In the original paper, this parameter is defined as $\sigma_i = d(s_i, s_J^{(i)})$, where $s_J^{(i)}$ is the J^{th} neighbor of s_i. In practice, it was found that a single setting, $J = 7$, gave acceptable results.

Although local scaling tends to improve the clustering, we found that the quality of the results using this approach still depends on finding the right combination of k and J, and that these choices are dependent on the dimensionality of the domain. Furthermore, as we will discuss in Section 5, a single value of J may not correctly cluster data in the presence of noise or under nonlinear geometric transformations.



(a) 3NN (b) 7NN (c) 2-Skeleton (d) 1-Skeleton

Figure 1: Proximity graphs and clusters for kNN and the β-skeleton. The sparsity of 3NN disconnects the "background" cluster, while 7NN adds many spurious connections between clusters, leading to poor separation. By contrast, at only 2.5 edges per point, the 2-skeleton generates the correct clustering via a more judicious choice of edges, as does the 1-skeleton using 3.9 edges per point.

3.2 Empty Region Graphs

In addition to graphs based solely on absolute or relative distances between points, a number of alternative proximity graphs have been proposed, such as the relative neighbor graph (RNG) and the Gabriel graph (GG), as surveyed by Jaromczyk et al. [16]. A family of these, known collectively as the *empty region graphs*, are more representative of the neighborhood of a point and less redundant than kNN, and are more efficient to compute than simplicial tessellations such as the Delaunay triangulation, particularly in high dimensions.

Definition 1. A graph $G(S,R) = (S,E)$ is an empty region graph if for every edge $(p,q) \in E$, a canonical region $R(p,q) \subseteq \mathbb{R}^d$ does not contain any other point in S:

$$pq \in E \iff R(p,q) \cap S = \varnothing \tag{3}$$

where R defines the neighborhood and is called the *empty region*.

Some common ERGs are:

Nearest Neighbor Graph (NNG). This is the directed graph that results from the empty region $R(p,q)$ formed by the open d-ball centered on p with radius $d(p,q)$.

$$pq \in E \iff \forall r \in S,\ d(p,r) \geq d(p,q) \tag{4}$$

Relative Neighborhood Graph (RNG). This graph is defined by a lune-shaped region consisting of the intersection of two d-balls of radius $d(p,q)$, one centered on p and the other centered on q, i.e.,

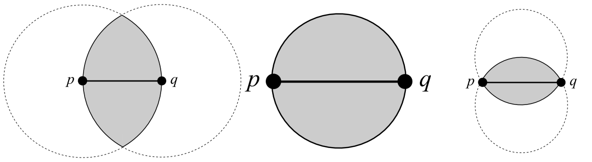
$$pq \in E \iff \forall r \in S,\ \max\{d(p,r), d(q,r)\} \geq d(p,q) \tag{5}$$

Gabriel Graph (GG). This is the graph defined by a d-ball centered at $\frac{1}{2}(p+q)$ with diameter $d(p,q)$, i.e.,

$$pq \in E \iff \forall r \in S,\ \sqrt{d(p,r)^2 + d(q,r)^2} \geq d(p,q) \tag{6}$$

β-Skeleton. The so-called lune-based β-skeleton is a one-parameter generalization of the RNG and GG, defined as follows:

- For $0 < \beta < 1$, the empty region is the intersection of all d-balls with diameter $d(p,q)/\beta$ that have p and q on the boundary.



(a) $1 < \beta \leq 2$ (b) $\beta = 1$ (c) $0 < \beta < 1$

Figure 2: Empty regions parameterized by β.

- For $\beta \geq 1$, the empty region is the intersection of two d-balls with diameter $\beta d(p,q)$ centered at $(1-\frac{\beta}{2})p + \frac{\beta}{2}q$ and $\frac{\beta}{2}p + (1-\frac{\beta}{2})q$.

It follows that $\beta = 2$ gives the RNG, while $\beta = 1$ is the GG. Fig. 2 depicts the geometric regions associated with different values of β. Finally, we note that geometric inclusion of one region within another also implies a partial order of the resulting neighborhood graphs (in terms of their edges), so that:

$$RNG \subseteq GG \subseteq (\beta \leq 1)\text{-skeleton} \tag{7}$$

This observation is key, since it allows us to explore neighborhood graphs of varying density using a single parameter, β, without the problems associated with kNN or ε-graphs. Section 5 provides empirical results that suggest that the clustering is far more stable for different values of β than they are for variations in k and σ.

4. APPROACH

We now motivate the use of empty region graphs for representing the neighborhood around a point, and consequently, for shaping the affinity matrix, and describe a general method for its application in spectral clustering.

4.1 Constructing the Neighborhood Graph

As a first step, we construct the β-skeleton of the data points for a given value $\beta \in (0,2]$. As described above, β-skeletons parameterize the neighborhood graph of a collection of points in a different

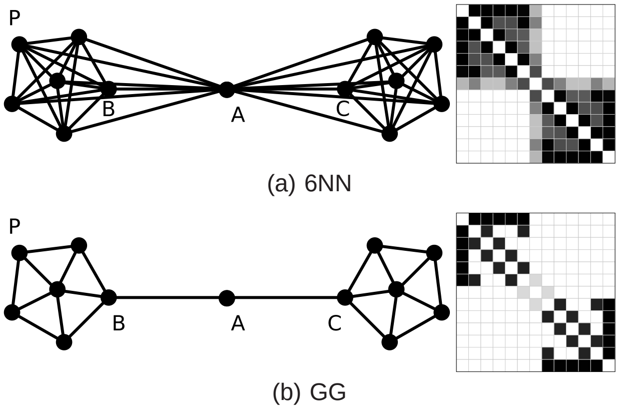


Figure 3: Two proximity graphs and their corresponding affinity matrices (a) 6NN, (b) 1-Skeleton, or Gabriel graph (GG). The bridging node between clusters destroys the block structure of the affinity matrix. While this is pronounced for the 6NN, it only affects two nodes (B and C) for the GG.

way than kNN does. In this paper we show that those graphs lead to better clustering than kNN graphs.

Graphs such as the *k*-nearest neighbor graph are susceptible to short circuiting nearby clusters when extra points lie between the clusters. As shown in Fig. 3(a), separating the two clusters is difficult due to the connecting node. In fact, the corresponding affinity matrix is formed by two blocks that overlap, and a large number of off-diagonal elements.

A β-skeleton alleviates this problem, as illustrated in Fig. 3(b). In this case, the connecting node (which may be due to noise or the presence of a smaller cluster) is joined to each cluster via a single edge. In the resulting affinity matrix, there are three blocks, but the off-diagonal elements are confined to the vicinity of point *A*.

4.2 Local Scaling

Now we show that β-skeletons provide better estimates than kNN of the local scale of a point. This local value defines the scale parameter for the computation of the affinity matrix. Using local scaling allows a pair of points within a high-density cluster to be assigned the same affinity as a pair of points in a low-density cluster when their separation in relation to the local scale is the same.

A natural measure of the local scale around a point include functions of the distance to its neighbors, such as the mean or the median distance. These measures can be brittle in *k*NN graphs, as suggested by Zelnik and Perona [34], and as studied by Maier et al. [22]. Picking the distance to an arbitrary nearest neighbor may prove more effective, but is sensitive to density changes and noise.

Here, we provide an initial estimate of the local scale using the mean or the median distance to a point's neighbors in an empty region graph. These measures make sense for these graphs since they approximate locally the spatial extents associated with each point. To illustrate this, consider the points in Fig. 4(a), connected in a 6NN graph. The dashed circle associated with each point has a radius equal to the average distance of the point to its neighbors, and can be understood as a representation of the local scale. Naturally, over-connecting the points results in artificially large local scales. The larger the local scale, the higher the affinity is of a point with a neighbor enclosed in its respective dashed circle. In Fig. 4(a), there is a high affinity between the points near the boundary of both clusters. The 3NN graph, in contrast, produces smaller local scales and clearly separates the two clusters (Fig. 4(b)). However, finding the appropriate *k* parameter for a kNN graph proves difficult. In contrast, the mean distance to the neighbors in the 1-Skeleton is a good estimator of the local scales required to cluster the points accurately, as shown in Fig. 4(c). Note that the local scales for each cluster is roughly the same as in the 3NN, except for the extremes of the edge connecting the two clusters, where the local scale is larger than expected, and one runs the risk of clustering them together.

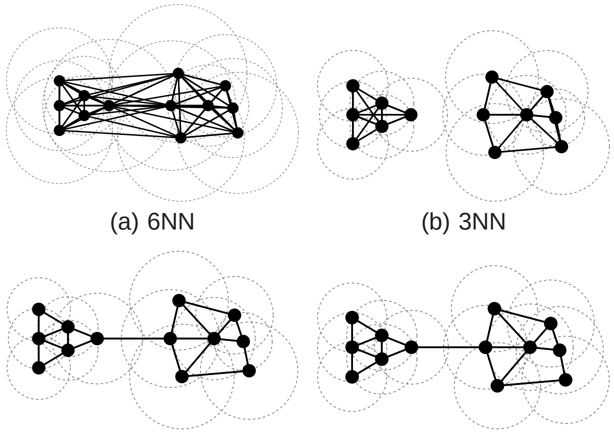


Figure 4: Average distance to neighbors as a measure of local scale. (a) Poor scale estimate due to an overconnected graph. (b) Better scale estimate using a sparser graph. (c) GG further improves the local scale, except for cluster boundary nodes. (d) Diffusion corrects the scale of boundary nodes.

This behavior, the estimation of larger scales for nodes connected via intercluster edges, is the result of using a β-skeleton, which is *well connected*. One might consider the use of the median distance to exclude outliers and solve the problem, but this technique may fail when the neighborhood graph is sparse A more fundamental problem are small "cliques" of a few close points that are embedded in a larger cluster. Their median neighbor distance may be far smaller than the scale suggested by the surrounding density, resulting in artificially small local scales and poor affinity with the rest of the cluster. To avoid this, one must analyze the local scale by looking not only at the immediate neighbors of a point, but at a possibly larger set.

To address this, we introduce a propagation mechanism based on non-linear diffusion, which improves the estimate of the local scale of a point by querying the scale of its neighbors, similar to reachability in density based clustering [27].

4.3 Diffusion-based Scale Refinement

To deal with the local scale of boundary points, one must ensure that a boundary point has a local scale so that points outside the cluster exhibit less affinity than those within the same cluster. In turn, those neighbors in the same cluster should exhibit affinity with their neighbors, and so on. Thus, the local scale of a point is affected by points that may not be immediate neighbors.

To determine the local scale, we use non-linear diffusion, such that the local density of a point (inverse of the local scale) is iteratively blended with the local densities of its neighboring points. Because shorter edges are likely to correspond to points in the same cluster, we use inverse distance weighted kernels, such as Gaussian or inverse polynomials.

The method works as follows: we start from an estimate of the local scale σ_i^0 of a point s_i as the mean or median distance to its neighbors in the neighborhood graph. We iteratively refine the local scale for some iterations $t \in \{1,\dots,T\}$,

$$\sigma_i^{(t)} = \left(\sum_{j \in N(s_i)\cup\{s_i\}} \hat{w}_{ij}^{(t-1)} \frac{1}{\sigma_j^{(t-1)}} \right)^{-1} \tag{8}$$

where $N(s_i)$ is the neighborhood of point s_i (using a β-skeleton), and the weights are normalized kernels, defined as the product of two Gaussians $G(d,\rho) = \exp(-d^2/\rho)$,

$$\hat{w}_{ij}^{(t)} = w_{ij}^{(t)} / \Sigma_k w_{ik}^{(t)} \tag{9}$$

$$w_{ij}^{(t)} = G(d(s_i,s_j),\rho_D) \times G(|\sigma_i^{(t)} - \sigma_j^{(t)}|,\rho_C) \tag{10}$$

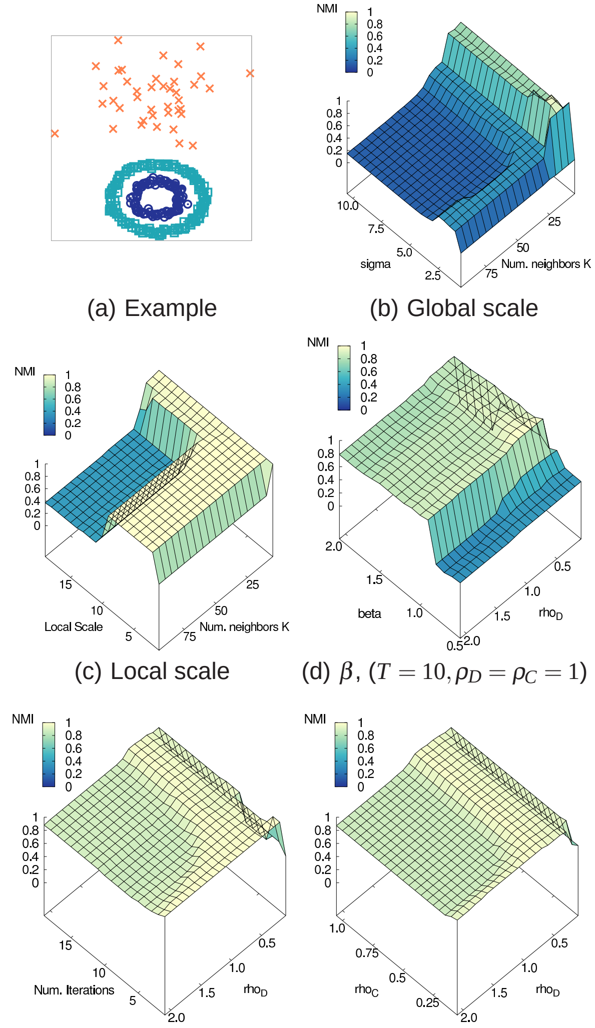
The first kernel blends the scales of nearby points, and ensures that intracluster scales are made more uniform than intercluster scales. The parameter ρ_D, called *diffusivity*, controls the speed at which diffusion propagates the local scales in terms of the distance between points. Diffusivity alone, however, makes the scales converge to a uniform value for the entire graph. We then introduce an additional kernel that penalizes the weight when the difference in scale is high, similar to bilateral filtering in image denoising [30]. The parameter ρ_C, or *conductivity*, controls how fast the diffusion propagates along scale discontinuities.

Note that we diffuse the local density instead of the local scales, similar to equivalent kernel density estimators [9], where the reciprocal of the local scale approximates the density of a point. We found this approach more accurate than blending the local scales.

4.3.1 *Parameter Selection*

Our approach, although designed to eliminate the selection of a global parameter k for the number of neighbors or σ for the global scale, requires the selection of different parameters, namely β, which controls the density of the neighborhood graph, and the diffusion parameters, T, ρ_D and ρ_C. Unlike the exploration of k and σ, we have found that parameterizing the graph density via β allows for more resilience to changes in point density, e.g. by limiting the number of intercluster edges. On the other hand, the diffusion parameters are inter-dependent. A large ρ_D may result in convergence to the same solution as a small ρ_D with only a fraction of the number of iterations T. Moreover, these diffusion parameters are similar to those used in kernel density estimation and diffusion in general, and have been well studied [20].

To illustrate the sensitivity of our algorithm to these parameters, we conducted an experiment to find the pair of parameters that yields the best clustering of the data set depicted in Fig. 5(a). We compared five scenarios: (1) Global scaling [26], defined in terms of number of neighbors k and global scale σ, (2) local scaling [34], in terms of number of neighbors k and local scale neighbor J, (3) our local scaling using graphs of different density via the parameter β, (4) our diffusion approach for a fixed $\beta = 1$ and $\rho_C = 1$, in terms of diffusivity ρ_D and number of iterations T, and (5) our non-linear diffusion approach, in terms of the diffusivity and conductivity parameters, while keeping a constant number of iterations $T = 10$. Fig. 5 shows the results for this experiment, depicted as surfaces of the normalized mutual information (NMI)—higher is better—for a discretization of their respective parameters. Global scaling is, not surprisingly, the most sensitive approach, and the optimal clustering is only achieved within a narrow interval of values. Local scaling in this case exhibits optimality for a fixed interval of the local scale. However, overshooting this interval produces drastically sub-optimal results. In contrast, exploring the graph density using β instead of k exhibits little sensitivity for $\beta > 0.7$. The same can be said about the diffusion parameters. Typically, we set ρ_D and ρ_C to 1. We believe bandwidth selection algorithms for kernel density estimation can be used towards improving these initial estimates [9].



(a) Example (b) Global scale

(c) Local scale (d) β, $(T = 10, \rho_D = \rho_C = 1)$

(e) Diffusion,$(\rho_C = 1, \beta = 1)$ (f) Conductivity $(T = 10, \beta = 1)$

Figure 5: Sensitivity of clustering to various sets of parameters.

4.4 Constructing the Affinity Matrix

Finally, once we determine the local scale associated with each point, we construct the affinity matrix as

$$A_{ij} = \begin{cases} \exp\left(\frac{-d(s_i,s_j)^2}{\sigma_i^{(T)}\sigma_j^{(T)}}\right) & (i,j) \in G(S,R(\beta)) \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where $R(\beta)$ is an empty region template, parameterized by β. Algorithm 1 summarizes the full algorithm.

4.5 Complexity and Time Analysis

We now describe the computational cost of the key steps in our algorithm.

Construction of ERG. Constructing an ERG can be expensive. A brute-force implementation requires $O(n^3)$ time, which is prohibitively expensive for most practical applications. How-

Algorithm 1: New algorithm for spectral clustering using empty region graphs.

```
input  : Data points $S = \{s_1, \dots, s_n\}$, number of clusters $K$,
         number of iterations $T$, diffusion parameters $\rho_D, \rho_C$
output: Classification $C$

$G \leftarrow \mathrm{ERG}(S)$                                        Construct ERG
for $i \leftarrow 1$ to $n$ do
    $N_i \leftarrow \textsc{Neighbors}(G, s_i)$
    $\sigma_i^{(0)} \leftarrow \textsc{LocalScale}(s_i, N_i)$            Estimate initial local scale

for $t \leftarrow 1$ to $T$ do
    for $i \leftarrow 1$ to $n$ do
        $N_i \leftarrow \textsc{Neighbors}(G, s_i)$
        $\Sigma_i \leftarrow \{\sigma_i^{(t-1)}\}$
        for $j \in N_i$ do
            $\Sigma_i \leftarrow \Sigma_i \cup \{\sigma_j^{(t-1)}\}$
        $\sigma_i^{(t)} \leftarrow \textsc{Diffuse}(\Sigma_i, \rho_D, \rho_C)$        Update local scale
$A \leftarrow \textsc{Affinity}(\{s_i\}, \{\sigma_i\})$                    Compute affinity matrix
$C \leftarrow \textsc{SpectralCluster}(A)$   Cluster using Laplacian of A [26]
```

ever, there are known algorithms that compute the RNG and GG in $O(n^2)$ time [31, 24].

We further reduce the computation cost of obtaining an ERG by restricting the neighbor search to the k_{max} nearest neighbors of a point, where k_{max} is usually a constant factor larger than the k value selected for k nearest neighbor graphs, but $k_{max} \ll n$. The overall complexity of computing such a kNN graph is

$$O(n^2 \log k_{max}),$$

and the additional cost of computing an ERG becomes

$$O(nk_{max}^2) \text{ time.}$$

Diffusion. The diffusion process involves a sequential walk on the neighborhood graph and requires $O(|E|T)$ time, where T is the number of iterations and $|E| \leq nk_{max}$ is the number of edges in the graph.

Solution of Eigenvector Problem. In general, computing the eigenvectors of the Laplacian matrix takes $O(n^3)$ time, but, as described by Song et al. [29], sparse eigensolvers, such as the variants of Lanczos/Arnoldi factorization (e.g., ARPACK), have a cost of

$$O(m^3) + (O(nm) + O(nk_{max}) + O(m-K)) \times (\text{\# restarted Arnoldi})$$

where $m > K$ is the Arnoldi length used to compute the first K eigenvectors of the affinity matrix.

Clustering the spectrum. Finally, using K-means to cluster the eigenvectors has a cost of $O(nK) \times$ (# K-means iterations).

Overall, as suggested by Song et al. [29], as the data size grows larger, the cost of the clustering becomes dependent on the construction of the affinity matrix. Nonetheless, the additional cost of computing an ERG instead of a kNN is relatively small. According to our results, this marginal increase is well worth the benefits of using the β-skeleton instead of k-nearest neighbors.

5. RESULTS

We have validated our algorithm with a number of low dimensional synthetic data sets and a few (higher-dimensional) real classification problems. To compare the quality of these datasets, we measure the normalized mutual information (NMI) between the clustering X and the ground truth classification Y:

$$NMI(X;Y) = \frac{2I(X;Y)}{H(X) + H(Y)} \qquad (12)$$

where $I(X;Y)$ is the mutual information between X and Y and $H(X)$ and $H(Y)$ are their entropies, respectively.

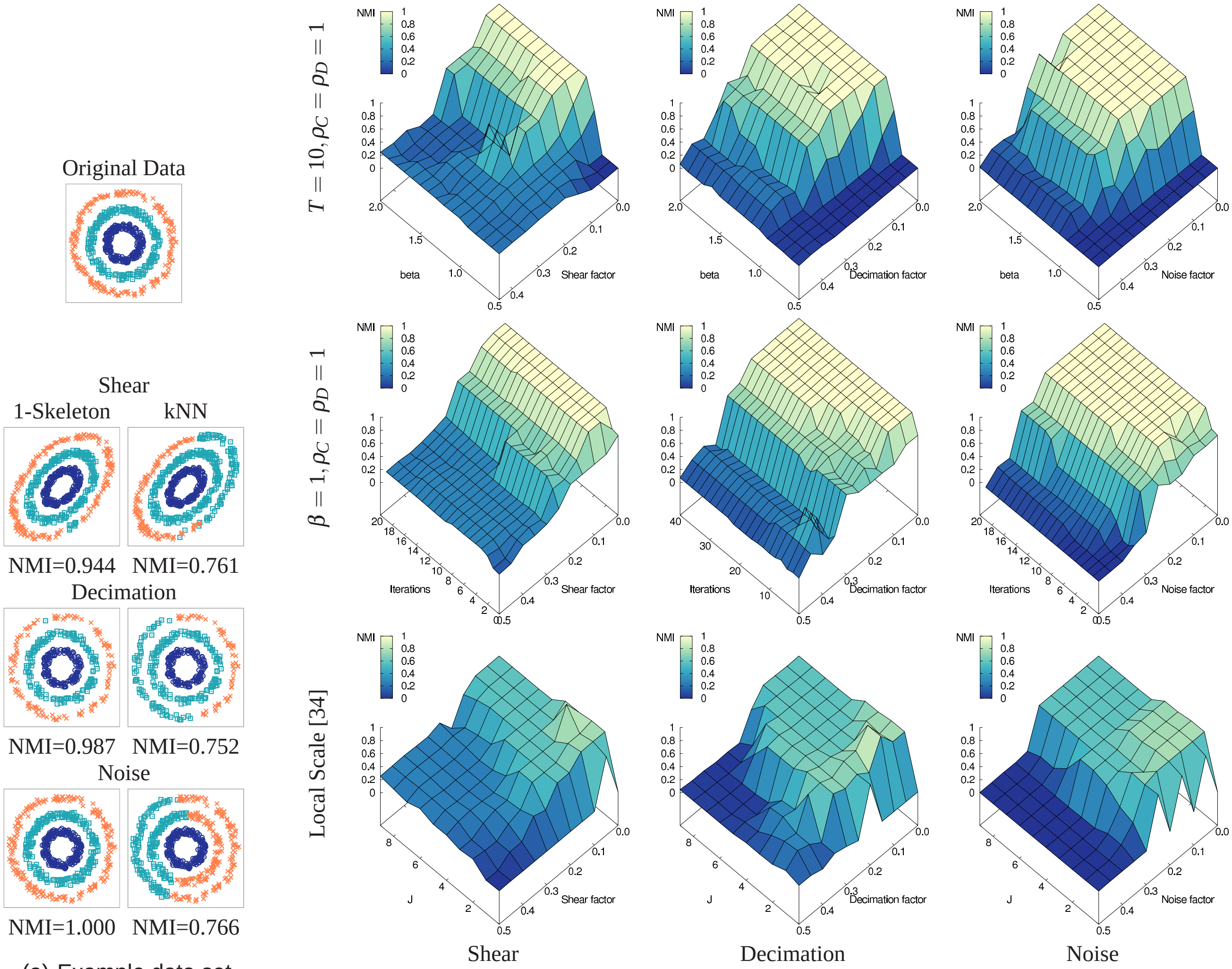
5.1 Sensitivity to Transformations

To illustrate the benefits of using the β-skeleton, we analyzed the results of our algorithm compared to traditional approaches using k-Nearest Neighbors (kNN) when applied to transformations of a synthetic data set. We compare three types of transformations: geometric distortion (shear transformation), decimation (where we remove $0.05Dn$ points, for a decimation factor $D \in \{1, \dots, 10\}$, and noise (where we perturb the data with Gaussian noise of increasing standard deviation). Fig. 6(a) compares the result of clustering a 2D data set consisting of three concentric rings under these three transformations. Fig. 6(b) shows quality surfaces for the different transformations and different values of the main parameters for our method (β and T) vs. the number of neighbors J used to define the local scale in kNN-based methods (as a reference, Zelnik and Perona's method uses $J = 7$). The quality of the clustering increases as the color of the surface approaches yellow. Notice how the optimal value for kNN methods (bottom) varies depending on the degree of distortion, decimation or noise. For our method, the quality converges as we increase the number of iterations and is less sensitive to the transformation itself.

A similar analysis was performed for other data sets, as shown in Fig. 7. Here, we show the mean, minimum and maximum quality for different values of β using our approach (blue), and for different values of k (global scale—green) and J (local scale—red) for kNN methods. In general, we observed a higher quality for empty region graphs and a lower variance when compared to globally-scaled clustering. Particularly for the first two rows, our approach is also better than locally-scaled clustering based on kNN. Note an important exception, the fourth row, consisting of small clusters surrounded by a random noisy background. β-skeletons do not cluster data as well on this data set, and in other similar cases. Locally-scaled kNN based methods are able to segment these because a small number of neighbors is able to keep them disconnected, while ERGs are always well-connected.

5.2 UCI Benchmarks

We validated our approach on multi-dimensional data sets from synthetic generators and the UCI Machine Learning repository [13]. The results are summarized in Table 1. We compare our approach using the β-skeleton (the parameter β chosen and number of iterations are reported next to each, for $\rho_D = 0.1$ and $\rho_C = 1$) with conventional clustering using kNN and (1) global scaling, (2) local scaling [34], and (3) DBSCAN, a density-based approach [27]. In all these cases, we report the best results obtained after exhaustively exploring their key parameters (k and σ for global-kNN and DBSCAN, k for local-kNN and β and T for ours). As seen in the table, it was possible to find a suitable neighborhood graph using the β parameter that produces better results than exhaustive search of the parameters of other conventional algorithms. Although the density of the graph increases with the number of dimensions, the exploration of the β parameter allows us to get the necessary density to cluster the higher-dimensional data sets. Similar to the results in Fig. 7, we also observed that quality is less sensitive to β than to values of K.



(a) Example data set

(b) Clustering quality as a function of transformation severity and algorithm parameter

Figure 6: (a) Results of clustering data undergoing transformations (geometric distortion, decimation and noise). (b) Clustering quality in terms of the transformation severity and algorithm parameter. From top to bottom: β and T in our algorithm, and the local scale given by J in [34].

5.3 Image Segmentation

Finally, we applied our method to image segmentation, where each pixel is defines a five dimensional point of its pixel coordinates x and y, and the color intensities in the $L^*u^*v^*$ color space. In all these results, the number of clusters is picked manually. Fig. 8 shows three examples from the Berkeley Segmentation Dataset and Benchmark [23]. Note how ERG based methods produce visually better clusters, while methods such as local scaling result in over-segmentation. Notably, Fig. 8(c) demonstrates the importance of good neighborhood graphs for local scaling. Attempting to separate this image into two segments proves difficult with a global scale using kNN, and users must resort to over-segmentation (4 clusters instead of 2). In our case (leftmost panels), the Gabriel graph is able to segment the airplane using both 2 and 4 clusters.

6. SUMMARY AND CONCLUSION

A fundamental problem in clustering is defining a neighborhood graph that defines how similar two data points are. This paper introduces a more natural parameterization of the neighborhood density based on the β-skeleton. We showed that using β to define the affinity between points provides higher quality than the prevalent approach of picking k nearest neighbors or finding a global scale parameter. We have also shown that our clustering results are far less sensitive to the choice of β and the diffusion parameters (diffusivity and conductivity), and to data transformations such as perturbations and geometric distortion.

Our diffusion-based local scaling approach proves effective for clustering irregular data sets and estimating the correct local scale at each point, despite the fact that the empty region graphs we explore (supersets of the relative neighbor graph), are well-connected. We must point out that we introduce additional parameters to tune the diffusion mechanism, but these are well known in other domains where diffusion is applied, and there are methods for choosing them. From a practical standpoint, one can define heuristics to explore the parameter space efficiently. As a rule of thumb, one defines the diffusivity parameter as the smallest scale one wants to preserve in the data set, if that information is known, for example,

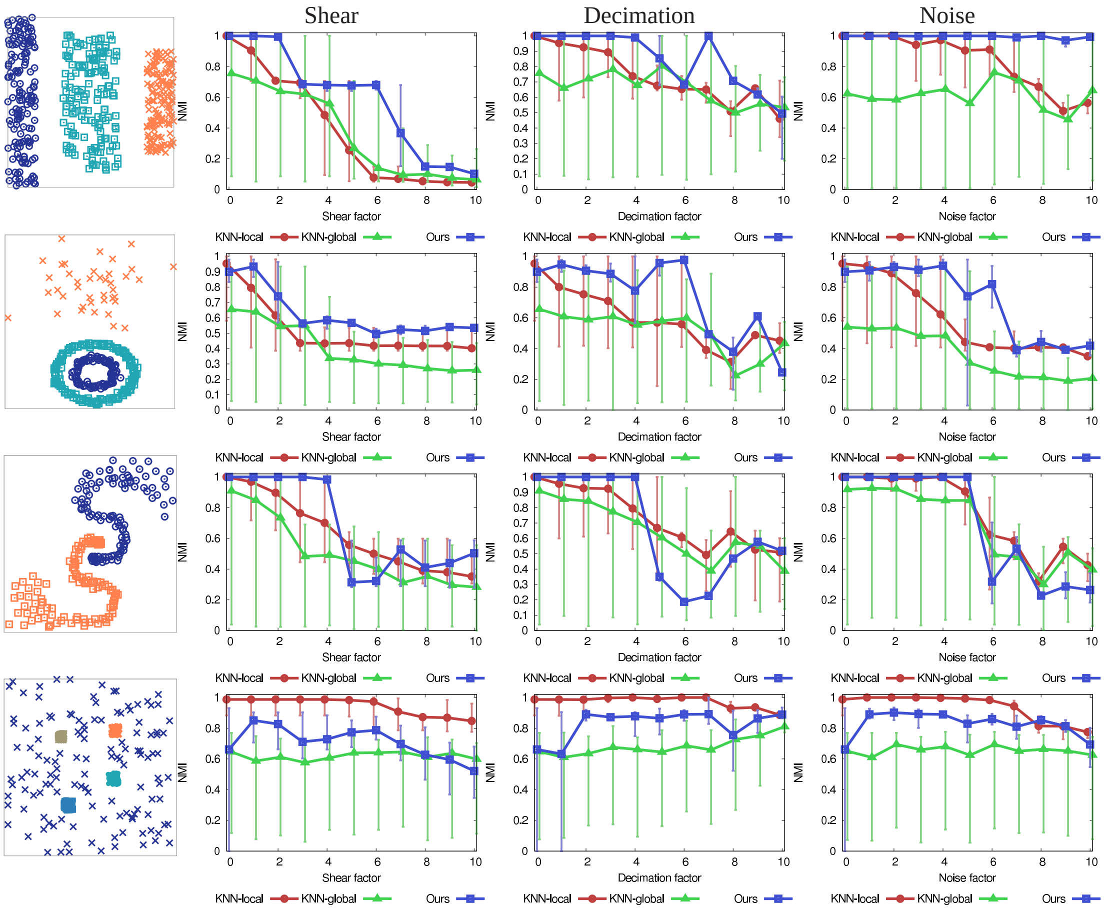


Figure 7: Average clustering quality (NMI – higher is better) of various 2D data sets as a function of transformation severity for our algorithm (blue), globally- (green) and locally-scaled (red) clustering based on kNN. Each curve shows the mean, maximum and minimum NMI while varying $\beta \in [0.8, 2.0]$ (blue), $k \in [2, 20]$ (green) or local scale $J \in [1, 20]$ (red). The β-skeleton provides higher quality than global scaling, and far less variance. Compared to local scaling, our algorithm performed better except in the last row.

when interested in the presence (or absence) of data features of a given size (e.g., in medical image segmentation).

In our experiments, we often apply the 1-Skeleton as an initial candidate for generating the affinity matrix. Over-segmentation may be an indication that the graph is too sparse and one might explore other graphs with $\beta < 1$. Conversely, when one suspects that the result is under-segmented, one may explore sparse graphs, with $1 < \beta \leq 2$. Our approach can be extended in a number of ways to retrieve the number of clusters automatically, as suggested by approaches like [34, 1]. We believe our approach, proving to be resilient to noise and other types of data perturbations, is a step forward towards robust spectral clustering, and may have broader applications where a neighborhood graph is required.

Acknowledgements

Prepared by LLNL under Contract DE-AC52-07NA27344.
LLNL-CONF-513768.

7. REFERENCES

[1] E. Biçici and D. Yuret. Locally scaled density based clustering. In *Proc. 8th int. conf. on Adaptive and Natural Computing Algorithms, Part I*, ICANNGA '07, pages 739–748, 2007.

[2] P. Bose, S. Collette, S. Langerman, A. Maheshwari, P. Morin, and M. Smid. Sigma-local graphs. *J. of Discrete Algorithms*, 8:15–23, 2010.

[3] J. Cardinal, S. Collette, and S. Langerman. Empty region graphs. *Comput. Geom. Theory Appl.*, 42:183–195, 2009.

[4] M. A. Carreira-Perpiñán and R. S. Zemel. Proximity graphs for clustering and manifold learning. In *Advances in Neural Information Processing Systems*, pages 225–232. MIT Press, 2005.

[5] J. Choo, R. Jiamthapthaksin, C.-S. Chen, O. U. Celepcikay, C. Giusti, and C. F. Eick. Mosaic: A proximity graph approach for agglomerative clustering. In *DaWaK*, volume 4654 of *Lecture Notes in Computer Science*, pages 231–240. Springer, 2007.

[6] R. J. Cimikowski. Properties of some euclidean proximity graphs. *Pattern Recogn. Lett.*, 13:417–423, 1992.

[7] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proc. of ACM SIGKDD int. conf. on Knowledge discovery and data mining*, pages 269–274, 2001.

[8] W. Donath and A. Hoffman. Lower bounds for partitioning of graphs. *IBM J. Res. Dev.*, 17:420–425, 1973.

[9] V. A. Epanechnikov. Non-parametric estimation of a multivariate probability density. *Theory of Probability and Its Applications*, 14, 1969.

[10] M. Ester, H. peter Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[11] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 1973.

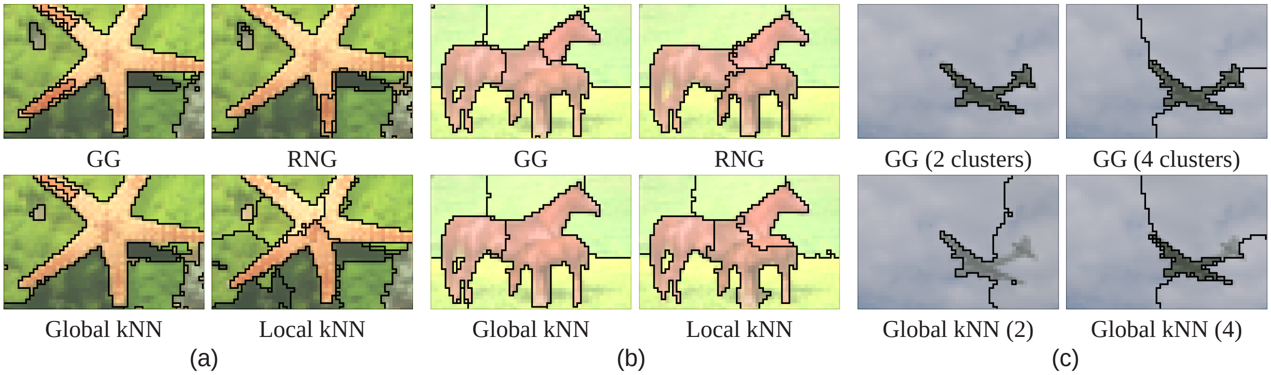


Figure 8: Image Segmentation in $L^*u^*v^*$. ERG based methods are able to adapt to local densities and produce better segmentations. In (c), finding the appropriate k and σ proves difficult for background-foreground segmentation and the user resorts to over-segmentation. In contrast, our approach produces good clusters even for over- segmented cases.

Table 1: Summary of experiments on multi-dimensional data.

Data set	Dimensions	Num. Clusters	Ours (β, T)	Global (kNN)	Local (kNN)	DBSCAN
Gaussians	3	3	**0.982** (1.0, 1)	0.822	**0.982** (J=7)	**0.982**
Noisy Rings	3	2	**1.000** (2.0, 5)	0.983	0.910	0.957
Ellipsoids	3	3	**0.975** (1.0, 20)	0.958	0.960	0.670
Iris	4	3	**0.843** (1.5, 2)	0.833	0.790	0.731
Ellipsoids5D	5	5	**0.816** (1.5, 70)	0.728	0.746	0.537
Auto-mpg	5	5	**0.713** (1.9, 2)	0.648	0.635	0.471
E.coli	7	8	**0.675** (1.0, 42)	0.671	0.620	0.437
Breast	9	2	**0.782** (1.7,8)	0.741	0.735	0.695
Glass	9	6	**0.466** (0.99,8)	0.355	0.366	0.459
Wine	13	3	**0.947** (1.9, 46)	0.938	0.866	0.527

[12] S. Fortune. Voronoi diagrams and delaunay triangulations. In J. E. Goodman and J. O'Rourke, editors, *Handbook of discrete and computational geometry*, pages 377–388. 1997.

[13] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[14] R. K. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18(3):259–278, 1969.

[15] L. Huang, D. Yan, M. I. Jordan, and N. Taft. Spectral clustering with perturbed data. In *Neural Information Processing Systems*, pages 705–712, 2008.

[16] J. Jaromczyk and G. Toussaint. Relative neighborhood graphs and their relatives. *Proc. of the IEEE*, 80(9):1502 –1517, 1992.

[17] R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. *J. The ACM*, 51:497–515, 2004.

[18] D. Kirkpatrick and J. Radke. A framework for computational morphology. *CG*, 85:217–248, 1985.

[19] Z. Li, J. Liu, S. Chen, and X. Tang. Noise robust spectral clustering. In *IEEE int. conf. on Computer Vision, 2007*, pages 1 –8, 2007.

[20] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.

[21] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. L. Cam and J. Neyman, editors, *Proc. of the 5th Berkeley Symp. on Mathematical Statistics and Probability*, volume 1, pages 281–297, 1967.

[22] M. Maier, U. von Luxburg, and M. Hein. How the result of graph clustering methods depends on the construction of the graph. *CoRR*, abs/1102.2075, 2011. informal publication.

[23] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[24] D. L. Millman and V. Verma. A slow algorithm for computing the gabriel graph with double precision. In *Proc. of the 23nd Canadian Conf. on Computational Geometry*, pages 485–487, 2011.

[25] B. Nadler and M. Galun. Fundamental limitations of spectral clustering. In *Neural Information Processing Systems*, pages 1017–1024, 2006.

[26] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856. MIT Press, 2001.

[27] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Min. Knowl. Discov.*, 2:169–194, 1998.

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

[29] Y. Song, W.-Y. Chen, H. Bai, C.-J. Lin, and E. Y. Chang. Parallel spectral clustering. In *ECML PKDD '08*, pages 374–389, 2008.

[30] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proc. of the Sixth int. conf. on Computer Vision*, ICCV '98, pages 839–, 1998.

[31] G. T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, 12:261–268, 1980.

[32] R. Urquhart. Graph theoretical clustering based on limited neighbourhood sets. *Pattern Recognition*, 15(3):173 – 187, 1982.

[33] U. von Luxburg, M. Belkin, and O. Bousquet. Consistency of spectral clustering. *Annals of Statistics*, 36:555–586, 2008.

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