

# On the Persistence of Clustering Solutions and True Number of Clusters in a Dataset

Amber Srivastava<sup>1a</sup>, Mayank Baranwal<sup>2b</sup>, Srinivasa Salapaka<sup>1c</sup>

<sup>1</sup>University of Illinois at Urbana-Champaign, <sup>2</sup>University of Michigan, Ann Arbor

<sup>a</sup>asrvstv6@illinois.edu, <sup>b</sup>mayankb@umich.edu, <sup>c</sup>salapaka@illinois.edu

## Abstract

Typically clustering algorithms provide clustering solutions with prespecified number of clusters. The lack of a priori knowledge on the true number of underlying clusters in the dataset makes it important to have a metric to compare the clustering solutions with different number of clusters. This article quantifies a notion of *persistence* of clustering solutions that enables comparing solutions with different number of clusters. The persistence relates to the range of data-resolution scales over which a clustering solution persists; it is quantified in terms of the maximum over two-norms of all the associated cluster-covariance matrices. Thus we associate a persistence value for each element in a set of clustering solutions with different number of clusters. We show that the datasets where *natural* clusters are a priori known, the clustering solutions that identify the natural clusters are most persistent - in this way, this notion can be used to identify solutions with *true* number of clusters. Detailed experiments on a variety of standard and synthetic datasets demonstrate that the proposed persistence-based indicator outperforms the existing approaches, such as, gap-statistic method,  $X$ -means,  $G$ -means,  $PG$ -means, dip-means algorithms and information-theoretic method, in accurately identifying the clustering solutions with true number of clusters. Interestingly, our method can be explained in terms of the phase-transition phenomenon in the deterministic annealing algorithm, where the number of distinct cluster centers changes (bifurcates) with respect to an annealing parameter.

## Introduction and Related Work

Many high-impact application areas such as bio-informatics (Andreopoulos et al. 2009), (Di Nuovo and Catania 2008), exploratory data mining (Larose and Larose 2014), combinatorial drug discovery (Sharma, Salapaka, and Beck 2008), data and network aggregation (Yuan, Zhan, and Wang 2014), medical imaging (Huang et al. 2015) and many other information processing fields have fueled significant work on clustering algorithms. Most of these algorithms such as  $k$ -means (Hartigan and Wong 1979),  $k$ -medoids (Park and Jun 2009), and expectation-maximization (Dempster, Laird, and Rubin 1977) require the number of clusters to be prespecified. Despite substantial work on clustering algorithms, there is relatively scant literature on determining the true number

of clusters in a dataset. In this context, it should be noted that there is no single agreed-upon notion of *natural* clusters or *true* number of clusters; typically existing algorithms make assumptions on the datasets (e.g. generated from a mixture of Gaussian distributions) and validate their results on datasets that satisfy the assumptions.

There are various measures developed to characterize the clustering solutions resulting from a clustering algorithm with different number ( $k$ ) of clusters. One of the popular methods for determining the number of clusters is based on computing *gap statistic* (Tibshirani, Walther, and Hastie 2001). It compares the total intracluster variation for different values of  $k$  (number of clusters) with their expected values under null reference distribution of the data. The number of clusters  $k$  is ascribed to the case where the gap is largest. However, as remarked in (Feng and Hamerly 2007), this method works well for finding a small number of clusters, but has difficulty as the true  $k$  increases.

Some of the recent methods that determine the number of clusters under some assumptions on datasets include -  $X$ -means algorithm (Pelleg, Moore, and others 2000), where clustering using  $k$ -means is performed for a range of number  $k$  of clusters, and the value  $k_t := k$  that yields the best Bayesian Information Criterion (BIC) (Kass and Wasserman 1995) score is chosen as an estimate for the true number of clusters. Other related algorithms use criteria such as Akaike information criteria (Akaike 2011) or minimum description length (Rissanen 1978) instead of BIC. The  $X$ -means algorithm works well for well-separated spherical clusters but tends to overfit in the case of non-spherical clusters (Feng and Hamerly 2007).

The information-theoretic approach (Sugar and James 2003) where it estimates the number of true clusters  $k_t$  by detecting a significant *jump* in the modified distortion  $D^\gamma$  vs  $k$  plot; here  $D$  is the clustering distortion objective,  $k$  is the number of clusters,  $\gamma \approx -d/2$ , and the data points are in  $\mathbb{R}^d$ . Although the choice  $\gamma \approx -d/2$  works well for certain datasets, one can find examples where this choice fails (Sugar and James 2003).  $G$ -means (Hamerly and Elkan 2004) algorithm identifies the number of clusters in a dataset under the assumption that each cluster is a Gaussian distribution. It is a hierarchical algorithm that increases the number of clusters  $k$  until the hypothesis that each cluster comes from a single Gaussian distribution is

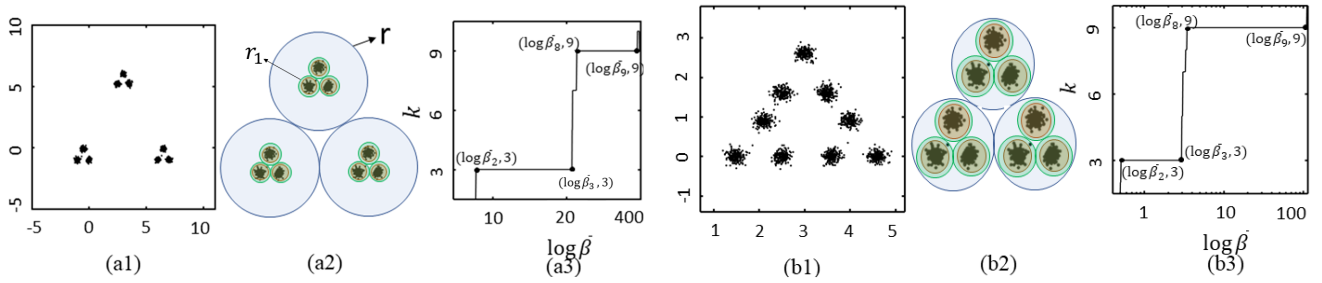


Figure 1: Illustration of a mixture of nine Gaussian distributions arranged in groups of three superclusters. In (a1) the three superclusters are well separated from each other while in (b1) they are closer to each other. Observe that in (a2) for a large range in resolution scales (radii  $r$ ) within the blue annulus, each supercluster appears as a single cluster, and only for a small range of resolution scale depicted by green annulus, each Gaussian distribution is identifiable separately. In other words for a large range of resolution the only the three superclusters are distinguishable from one another while for a smaller range of resolution each Gaussian distribution is identified separately. In (b1) since the three superclusters are closer to each other, the range of resolution scales within the blue annulus gets reduced, thereby indicating existence of three natural clusters in (a1) and nine natural clusters in (b1).

validated; typically done using the Anderson-Darling statistical test (Stephens 1974) on each cluster after projecting it onto a one-dimensional space.  $PG$ -means algorithm (Feng and Hamerly 2007) is an improvement on the  $G$ -means algorithm, where the number of clusters in a Gaussian mixture model is obtained by applying the Kolmogorov-Smirnov (KS) test to the one-dimensional projection of the *entire* dataset;  $PG$ -means also works well when the true clusters are overlapping with each other.

Dip-means (Kalogeratos and Likas 2012) is another method that assumes the dataset is generated from a mixture of unimodal distributions. Here, Hartigan’s dip statistic test (Hartigan and Hartigan 1985) is used to verify the unimodal nature of the admissible cluster. The authors also extend the dip-means algorithm to shape clustering problems by using kernel  $k$ -means (Dhillon, Guan, and Kulis 2004) for clustering. Other alternative approaches in the literature to estimate the true number of clusters are Bayesian  $k$ -means (Kurihara and Welling 2009) that uses Maximization-Expectation to learn a mixture model, a method based on repairing faults in Gaussian mixture models (Sand and Moore 2001) and various stability-based model validation methods (Lange et al. 2003), (Tibshirani and Walther 2005), (Levine and Doman 2001). The main drawbacks in most of the above existing methods stem from the underlying restrictive assumptions on the datasets; accordingly, the algorithms do not perform well when datasets do not meet the assumptions, which is often the case when considering standard non-synthetic datasets. These methods fail to accurately estimate the true number of clusters in most of the standard datasets as illustrated in the experiment section in this paper.

In this article we develop a notion of *persistence* of clustering solutions that enables comparing solutions, which result from a clustering algorithm, with different number of clusters. Here we do not make any assumptions on the underlying data distribution. Since a clustering solution requires grouping a set of points in such a way that points in the same cluster are more similar to each other than to those

in other clusters. We characterize *persistence* of a clustering solution as the *range* of resolution scales for which (a) points within each cluster seem indistinguishable, and (b) points in different clusters are distinguishable. For instance, Figure 1(a1) illustrates a dataset containing nine Gaussian clusters which are arranged in groups of three superclusters. If we choose the resolution scale of radius  $r$ , as shown in Figure 1(a2), then the points within each super-cluster is indistinguishable. Therefore, one will conclude that at this resolution level the dataset consists of only three clusters. Also note that in Figure 1(a1), the three super-clusters are persistent for a large range of resolution scales as indicated by the thickness of the blue annulus around each of them. On the other hand, the green annulus around each of the nine Gaussian clusters, is relatively thinner indicating that a clustering solution that identifies all the nine Gaussian clusters is relatively less persistent.

In a later section we quantify this notion of persistence of a clustering solution with  $k$  distinct clusters. In particular, the persistence is characterized in terms of the maximum over two-norms of all the cluster covariance matrices at two successive values of  $k$ . We also show analytically how for a clustering solution that identifies natural clusters, this measure correctly estimates the true number of clusters through a simple illustrative example consisting of spherical clusters with uniform distributions. We also provide extensive experimental results on a variety of standard and synthetic datasets in a later section. The results demonstrate that our method outperforms over the existing algorithms described above. In particular, our method correctly estimates the true number of clusters on 13 of the 14 benchmark datasets tested, whereas the next best method could estimate the true number of clusters only on 7 benchmark instances.

## Persistence of a Clustering Solution and its Quantification

**Intuitive description :** The notion of a cluster can be re-

lated to the resolution scales at which a dataset is viewed. For instance, on one hand the entire dataset can be considered as a single cluster, while on the other hand each data point can also be considered as a cluster by itself. Thus in Figure 1(a1), at low resolution scale (characterized here by a radius greater than the diameter of the entire dataset) no two points of the dataset are distinguishable from each other and the entire dataset is deemed a single cluster. Now consider a clustering solution at a higher resolution scale (for instance resolution characterized by the radius  $r$  in Figure 1(a2)), the datapoints from the three superclusters become distinct from one another and we are able to identify these as three distinct clusters in the dataset. Upon further increasing the resolution scale (for instance resolution characterized by the radius  $r_1$  in Figure 1(a2)), the datapoints sampled from each of the nine Gaussian distributions become distinct from one another and we are able to identify the nine clusters in the dataset. On further increasing the resolution scale, each point by itself will be regarded as a cluster.

We use resolution to capture the notion of persistence of clustering solution. We propose that the clustering solution that persists for large range of resolution is a good indicator of natural clusters and corresponding number estimates true number of clusters. In fact, after quantifying persistence later in this section, we show that for both the datasets in Figure 1(a1) and 1(b1), the persistence is larger for clustering solutions with three and nine clusters, while they are relatively small for clustering solutions with other number of clusters. Moreover, we observe that the three super-clusters are more persistent than the nine clusters for the dataset in Figure 1(a1), while nine clusters are more persistent than the three super-clusters in Figure 1(b1); this is also intuitive from the relative thickness of the blue and green annuli in Figures 1(a2) and 1(b2). Therefore, this suggests that the three super-clusters are *more* natural in Figure 1(a1) while the nine clusters are *more* natural in Figure 1(b1). These inferences agree with our intuition from visual inspection of these datasets and are also corroborated by the measure proposed later in this section.

Let  $\beta_k$  denote the lowest resolution scale at which  $k + 1$  clusters are identifiable in a dataset. We define the persistence of a clustering solution with  $k$  clusters as  $[\log \beta_k - \log \beta_{k-1}]$ . The *true number* of clusters can be estimated in terms of persistence of clustering solution. Accordingly, if the clustering into  $k$  groups is persistent for a long range of resolution scales, without  $k + 1$  clusters becoming evident in the dataset, then  $k$  is a good estimate for the true number of clusters. In other words, for a clustering solution with true number  $k_t$  of natural clusters it takes a large change in the resolution scale for the data points, originally belonging to the same cluster, to become distinguishable enough so as to belong to different clusters. Our hypothesis is that for a clustering solution with  $k_t$  natural clusters  $\log \beta_{k_t} - \log \beta_{k_t-1} > \log \beta_k - \log \beta_{k-1}$  for all  $k \neq k_t$ .

**Quantification of persistence of a clustering solution and resolution scales:** This quantification is substantially motivated by our reinterpretation of the deterministic annealing (DA) algorithm (Rose 1998). The DA algorithm mimics the annealing procedure studied in statistical physics

literature and gives high quality clustering solutions on linearly separable data. We do not describe the DA algorithm in detail here albeit re-interpret the auxiliary cost function (referred to as *free-energy* in (Rose 1998)) in DA to discern a possible use of its annealing parameter as a measure of resolution.

The DA algorithm views the problem of clustering a dataset  $\mathcal{X} = \{x_i : x_i \in \mathbb{R}^d, 1 \leq i \leq N\}$  consisting of  $N$  data points into  $k$  groups of nearly similar entities as an equivalent facility location problem (FLP), where the goal is to allocate a set of facilities  $\mathcal{Y} = \{y_j : y_j \in \mathbb{R}^d, 1 \leq j \leq k\}$  to data points  $\{x_i\}$  such that the cumulative distance between data points and their nearest facilities is minimum. Note that the facility locations are indeed the centroids of individual clusters (for linearly separable data and with squared Euclidean metric) and the FLP viewpoint is critical to many commonly used clustering algorithms, such as  $k$ -means. Thus, a solution to FLP results in clustering of the underlying dataset where the corresponding clusters  $\{\pi_j\}$  are defined by *voronoi partitions*  $\pi_j = \{x_i \in \mathcal{X} : y_j = \arg \min_{\{y_l\}} d(x_i, y_l)\}$ . More precisely, we consider the following optimization problem for FLP

$$D = \min_{\mathcal{Y}} \sum_{i=1}^N p_i \sum_{\{y_j\}} \min_{1 \leq j \leq k} d(x_i, y_j), \quad (1)$$

where  $p_i$  denotes a known relative weight of vector  $x_i$  (e.g.  $p_i = \frac{1}{N}$ ), and  $d(x_i, y_j)$  is a measure of distance between  $x_i$  and  $y_j$  which is usually considered to be the squared Euclidean distance. In data compression literature  $D$  is usually referred to as the distortion function (Gersho and Gray 2012). DA considers the log-sum-exp approximation where it approximates  $\min_{1 \leq j \leq k} d(x_i, y_j)$  by  $-\frac{1}{\beta} \log \sum_{j=1}^k e^{-\beta d(x_i, y_j)}$ , which results in the following smooth optimization problem that approximates (1)

$$F = \min_{\mathcal{Y}} -\frac{1}{\beta} \sum_{i=1}^N p_i \log \sum_{\{y_j\}} e^{-\beta d(x_i, y_j)}, \quad (2)$$

which is parameterized by  $\beta \in \mathbb{R}$ . The parameter  $\beta$  determines the extent of approximation of  $D$  by  $F$ . At larger values of  $\beta \rightarrow \infty$  the approximation function  $F$  tends to converge at the distortion  $D$ . On the other hand, at low values of  $\beta$  ( $\approx 0$ ) approximation  $F$  is considerably distinct from the distortion  $D$ . Here  $F$  is referred to as the *free-energy* function and  $\beta$  as an annealing parameter in the DA algorithm. We obtain the minimum (local) of  $F$  at a given  $\beta$ , by setting the partial derivative  $\frac{\partial F}{\partial y_j}$  to zero, which results in the following centroid-like condition for squared Euclidean distances:

$$y_j = \left( \sum_{i=1}^N p_i p(j|i) x_i \right) / \left( \sum_{i=1}^N p_i p(j|i) \right), \text{ where} \quad (3)$$

$$p(j|i) = \left( e^{-\beta d(x_i, y_j)} \right) / \left( \sum_{y_j \in \mathcal{Y}} e^{-\beta d(x_i, y_j)} \right). \quad (4)$$

We justify the annealing parameter  $\beta$  as a measure of resolution as follows. Note that for any two data points

$\mathbf{x}_1$  and  $\mathbf{x}_2$  in the bounded dataset, the term  $e^{-\beta d(\mathbf{x}_1, \mathbf{y}_j)} \approx e^{-\beta d(\mathbf{x}_2, \mathbf{y}_j)}$  when  $\beta$  is small ( $\beta \approx 0$ ), i.e. points  $\mathbf{x}_1$  and  $\mathbf{x}_2$  are indistinguishable. More precisely, for every  $\epsilon > 0$ , there exists  $\beta > 0$  small enough such that  $|e^{-\beta d(\mathbf{x}_1, \mathbf{y}_j)} - e^{-\beta d(\mathbf{x}_2, \mathbf{y}_j)}| < \epsilon$ . Note that from (3), at small values of  $\beta$  ( $\approx 0$ ) all the facilities  $\{\mathbf{y}_j\}$  are coincident. We can deduce from here that at low  $\beta$  values, no two data points are distinguishable (within  $\epsilon$ ), that is, the optimization problem (2) cannot differentiate between them, and therefore entire dataset will be deemed as a single cluster. In fact, the DA algorithm associates only one resource  $\mathbf{y}_1$  at the centroid of the entire dataset. Now as  $\beta$  increases, two distinct points that originally belonged to the same cluster  $\pi_j$ , become distinguishable for large enough  $\beta$ , i.e.  $e^{-\beta d(\mathbf{x}_1, \mathbf{y}_j)}$  no longer approximates  $e^{-\beta d(\mathbf{x}_2, \mathbf{y}_j)}$ ; thus the problem (2) can differentiate between these data points, and they need not necessarily belong to the same cluster. In the limit  $\beta \rightarrow \infty$ , all the data points are entirely distinct from each other and the optimal solution to (2) is to assign a distinct facility to each point since no two data points are similar enough to be put in the same cluster. Thus  $\beta$  quantifies our intuitive notion of resolution. Equivalently,  $\log \beta$  quantifies the resolution scale.

To quantify *persistence* of a clustering solution in the FLP setup, we need to determine the range of resolution scales  $\log \beta$  over which a clustering solution persists. Note that a clustering solution  $\{\pi_j\}$  to the FLP persists till the set of cluster centers  $\mathcal{Y} = \{\mathbf{y}_j\}$  cease to be a minima of  $F$  as the annealing parameter  $\beta$  (resolution) is increased. Therefore in context of the relaxed problem (2), we need to compute  $\mathcal{Y}$  such that it minimizes  $F$  and find the range of values of  $\log \beta$  for which it remains a minimum, i.e. at every resolution level in this range the optimal centers  $\mathcal{Y}$  must satisfy

$$\frac{d}{d\epsilon} F(\mathcal{Y} + \epsilon\Psi) \Big|_{\epsilon=0} = 0, \quad \text{and} \quad (5)$$

$$H(\mathcal{Y}, \Psi, \beta) := \frac{d^2}{d\epsilon^2} F(\mathcal{Y} + \epsilon\Psi) \Big|_{\epsilon=0} > 0, \quad (6)$$

for all finite perturbations  $\Psi$ . The cluster centers  $\mathcal{Y}$  ceases to be a minimum of  $F$  for a value of  $\beta$  when the Hessian (6) is no longer positive definite, that is when there exists a perturbation  $\Psi$  such that  $H(\mathcal{Y}, \Psi, \beta)$  is no longer positive definite. Now it can be shown that for  $d(x_i, y_j)$  as squared Euclidean distance the Hessian is

$$H(\mathcal{Y}, \Psi, \beta) = \sum_{\mathbf{y}_j} p_i p(j|i) \psi_j^T \left[ I - 2\beta C_{\mathcal{X}|\mathbf{y}_j}^k \right] \psi_j + \sum_{i=1}^N p_i \left[ \sum_{\mathbf{y}_j} p(j|i) (\mathbf{x}_i - \mathbf{y}_j)^T \psi_j \right]^2, \quad (7)$$

$$\text{where } C_{\mathcal{X}|\mathbf{y}_j}^k \triangleq \sum_{i=1}^N p(i|j) (\mathbf{x}_i - \mathbf{y}_j) (\mathbf{x}_i - \mathbf{y}_j)^T \quad (8)$$

is the *cluster covariance matrix* of the posterior distribution  $p(i|j)$  and  $I$  is the identity matrix of appropriate dimensions. From (7), it is not difficult to show that  $H(\mathcal{Y}, \Psi, \beta)$  loses

its positivity only when  $\det \left[ I - 2\beta C_{\mathcal{X}|\mathbf{y}_0}^k \right] = 0$  at some  $\mathbf{y}_0 \in \mathcal{Y}$  (please refer to supplementary material for proof)<sup>1</sup>; therefore the critical value of  $\beta$  beyond which the clustering center  $\mathcal{Y} = \{\mathbf{y}_j \in \mathbb{R}^d : 1 \leq j \leq k\}$  is no longer a minimum is given by

$$\beta_k = 1 / \left( 2\lambda_{\max}(C_{\mathcal{X}|\mathbf{y}_0}^k) \right), \quad (9)$$

where  $\lambda_{\max}(C_{\mathcal{X}|\mathbf{y}_0}^k)$  is the largest eigenvalue of  $C_{\mathcal{X}|\mathbf{y}_0}^k$ . In fact  $\mathbf{y}_0$  is the centroid of that cluster which has the maximum variance, i.e.  $\mathbf{y}_0 = \arg \max_{\{\mathbf{y}_j\}} \lambda_{\max}(C_{\mathcal{X}|\mathbf{y}_j}^k)$ . Therefore beyond the critical value  $\beta_k$  in (9) the number of identifiable cluster increases by one so as to identify a new minimum of  $F$  (Rose 1998). This makes intuitive sense, since we would expect the clusters with biggest variance to split before the others. The spread directions are indicated by the associated eigenvectors of the covariance matrix, with the largest spread along the eigenvector corresponding to the largest eigenvalue. Therefore using this analysis we can quantify persistence of a clustering solution with  $k$  clusters by  $v(k) := \log \beta_k - \log \beta_{k-1}$ , where  $\beta_k$  is the resolution at which the number of distinct clusters increases from  $k-1$  to  $k$ . The *true* number  $k_t$  of clusters can be estimated by  $\arg \max_k v(k)$ .

**Making the persistence independent of the DA algorithm:** Note that, though our quantification of persistence of a clustering solution is motivated from the DA algorithm, we can easily make it algorithm independent by replacing soft associations in (8) with hard associations; thus we re-define the cluster covariance matrix (8) for clustering solution with  $k$  clusters as

$$\bar{C}_{\mathcal{X}|\mathbf{y}_j}^k = \sum_{i=1}^N \nu_{ij} (\mathbf{x}_i - \mathbf{y}_j) (\mathbf{x}_i - \mathbf{y}_j)^T \quad (10)$$

where the posterior distribution term  $p(i|j) \in [0, 1]$  is replaced by  $\nu_{ij} \in \{0, 1\}$  that represents hard-associations between the vector  $\mathbf{x}_i$  and cluster centroid  $\mathbf{y}_j$  such that  $\nu_{ij} = 1$  if vector  $\mathbf{x}_i$  belongs to the cluster  $\pi_j$  and zero otherwise. We formally define the true number of clusters  $k_t$  in a dataset as

$$k_t := \arg \max_k \left[ v(k) := \log \bar{\beta}_k - \log \bar{\beta}_{k-1} \right], \quad (11)$$

$$\text{where } \bar{\beta}_k := \left[ \max_{1 \leq j \leq k} \left[ 2\lambda_{\max}(\bar{C}_{\mathcal{X}|\mathbf{y}_j}^k) \right] \right]^{-1} \quad (12)$$

is the minimum resolution level at which  $k+1$  distinct clusters centroids should be allocated to the dataset.

**Remark on Scalability and use of  $\log$  in  $v(k)$ :** Note that our proposed method is scalable with respect to size ( $N$ ) of the dataset as it requires computation of the largest eigenvalue of a  $d \times d$  covariance matrix  $\bar{C}_{\mathcal{X}|\mathbf{y}_j}^k$  which can be computed in  $O(d^2)$ , where  $d$  is the dimension of the feature-space. Our notion of persistence captures, by what factor one should scale resolution to obtain more clusters. This factor is simply expressed as a difference using  $\log$  for better visualization. Also the range of resolutions is typically very

<sup>1</sup>arXiv:1811.00102

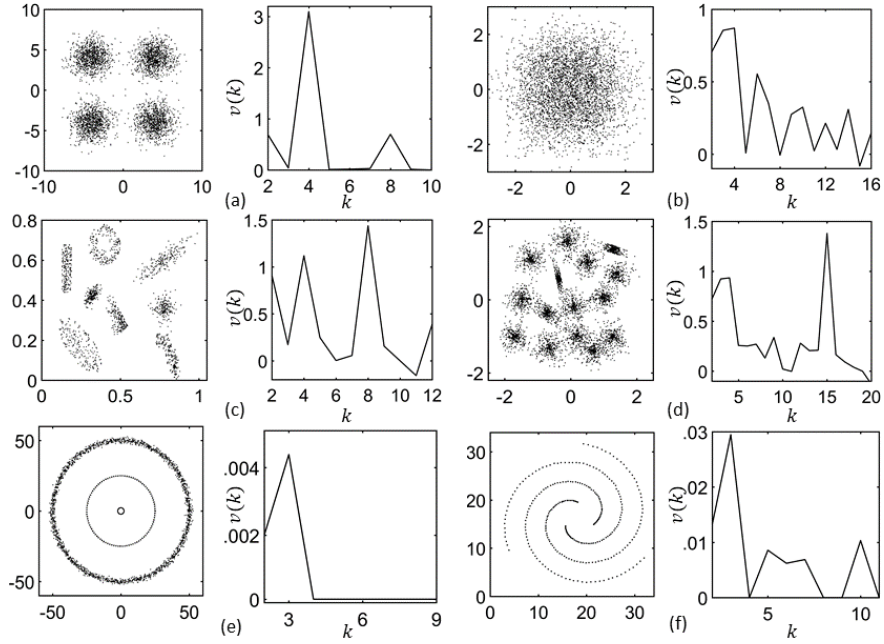


Figure 2: Evaluation of our method on a variety of synthetic datasets - (a) Low variance Gaussian,  $k_t = 4$ , (b) High variance Gaussian,  $k_t = 4$ , (c) ComboSetting,  $k_t = 8$ , (d) Synthetic-15 (S15),  $k_t = 15$ , (e) Concentric rings,  $k_t = 3$ , and (f) Spirals,  $k_t = 3$ . Our method predicts the correct number of clusters in each of these scenarios.

large (each data point is a highest resolution cluster to the entire dataset being the lowest resolution cluster); therefore log function discriminates this range better.

---

**Algorithm 1**  $\text{main}(\mathcal{X}, k_{\max})$

---

1. Initialize  $k = 1$ .
  2. Run a clustering algorithm on  $\mathcal{X}$  with  $k$  clusters and compute  $\bar{\beta}_k$  using (12).
  3.  $k \leftarrow k + 1$ . Go to step 2. Stop if  $k = k_{\max} + 1$ .
  4. Choose  $k_t$  using (11).
- 

Figures 1(a3) and 1(b3) illustrates our method for determining number of true clusters on two datasets considered in Figures 1(a1) and 1(b1) respectively. Observe that the quantities  $\log \bar{\beta}_3 - \log \bar{\beta}_2 \gg \log \bar{\beta}_k - \log \bar{\beta}_{k-1}$  and  $\log \bar{\beta}_9 - \log \bar{\beta}_8 \gg \log \bar{\beta}_k - \log \bar{\beta}_{k-1}$  for all  $k \neq 3$  and  $k \neq 9$  in both the figures. Further, we observe in the Figure 1(a3) that  $\log \bar{\beta}_3 - \log \bar{\beta}_2 > \log \bar{\beta}_9 - \log \bar{\beta}_8$  which indicates  $k_t = 3$  (using (10)) for the dataset in Figure 1(a1). Similarly we observe in Figure 1(b3) that  $\log \bar{\beta}_9 - \log \bar{\beta}_8 > \log \bar{\beta}_3 - \log \bar{\beta}_2$  which indicates  $k_t = 9$  for the dataset in Figure 1(b1). Again, these inferences agree with our intuition from visual inspection of these datasets.

**Extending to nonlinearly separable data:** The method proposed above works well for linearly separable data. However, many problems such as shape clustering or clustering with pairwise distances often consist of data points that are not separable linearly. We use kernel trick (Dhillon, Guan,

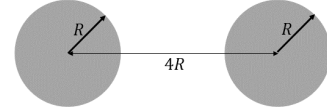


Figure 3: Illustrates two circular clusters of radius  $R$  with uniformly distributed data-points.

and Kulis 2004) to overcome this issue. Accordingly, we map data points  $\mathbf{x}_i$  to an abstract higher-dimensional space (where data-points are linearly separable) through a suitably chosen kernel function  $\phi(\cdot)$ . While an explicit representation of kernel function  $\phi(\cdot)$  is unknown, the inner-products  $\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$  are known as elements of a kernel matrix  $\mathcal{K}$  (Dhillon, Guan, and Kulis 2004). Thus, in order to identify the clustering solution with the correct number of clusters in a nonlinearly separable dataset, one must evaluate the largest eigenvalue of the kernel data covariance matrix  $\bar{C}_{\phi(\mathcal{X})|\mathcal{Y}_j}^k$  defined as:

$$\bar{C}_{\phi(\mathcal{X})|\mathcal{Y}_j}^k = \sum_{\mathbf{x}_i \in \pi_j} (\phi(\mathbf{x}_i) - \mathbf{y}_j) (\phi(\mathbf{x}_i) - \mathbf{y}_j)^T, \quad (13)$$

where  $\pi_j$  denotes the  $j$ -th cluster. Computation of eigenvalues of  $\bar{C}_{\phi(\mathcal{X})|\mathcal{Y}_j}^k$  is not straightforward (since  $\phi$  is unknown), however, we make use of the following lemma in order to obtain the spectral values of  $\bar{C}_{\phi(\mathcal{X})|\mathcal{Y}_j}^k$ .

**Lemma 1** *Let  $\bar{C}_{\phi(\mathcal{X})|\mathcal{Y}_j}^k$  be the cluster covariance matrix as defined in (13). Then  $\bar{C}_{\phi(\mathcal{X})|\mathcal{Y}_j}^k$  and  $A = [A_{kl}]$  share the same non-zero eigenvalues, where  $A_{kl} =$*

Table 1: Comparing algorithms on a variety of standard and synthetic datasets

Algorithm	Low Variance	High Variance	Combo-Setting	Wisconsin	Yeast	Glass	Leaves
	$k_t = 4$	$k_t = 4$	$k_t = 8$	$k_t = 2$	$k_t = 10$	$k_t = 6$	$k_t = 100$
gap-statistic	<b>4</b>	1	27	12	49	39	117
$X$ -means	1	1	25	6	47	1	219
$G$ -means	<b>4</b>	68	33	83	56	15	66
$PG$ -means	<b>4</b>	3	12	6	3	7	Error
dip-means	<b>4</b>	1	<b>8</b>	11	1	1	3
Info. Th.	<b>4</b>	3	<b>8</b>	<b>2</b>	2	<b>6</b>	99
kernel dip-means	-	-	-	-	-	-	-
Our Method	<b>4</b>	<b>4</b>	<b>8</b>	<b>2</b>	<b>10</b>	<b>6</b>	<b>100</b>

Algorithm	Wine	Iris	Banknote	Thyroid	Birch1	Concentric Rings	Spirals
	$k_t = 3$	$k_t = 3$	$k_t = 2$	$k_t = 3$	$k_t = 100$	$k_t = 3$	$k_t = 3$
gap-statistic	<b>3</b>	<b>3</b>	58	17	Error	n/a	n/a
$X$ -means	40	12	230	1	1	n/a	n/a
$G$ -means	2	4	57	7	1953	n/a	n/a
$PG$ -means	1	2	35	<b>3</b>	32	n/a	n/a
dip-means	1	2	4	1	Error	n/a	n/a
Info. Th.	<b>3</b>	2	5	<b>3</b>	<b>100</b>	n/a	n/a
kernel dip-means	-	-	-	-	-	<b>3</b>	1
Our method	<b>3</b>	2	<b>2</b>	<b>3</b>	<b>100</b>	<b>3</b>	<b>3</b>

$$(\phi(\mathbf{x}_k) - \mathbf{y}_j)^T (\phi(\mathbf{x}_i) - \mathbf{y}_j).$$

**Proof:** Please refer to the supplementary material<sup>1</sup> for proof of the above lemma.

Note that from (3), the cluster center  $\mathbf{y}_j = \frac{1}{|\pi_j|} \sum_{\mathbf{x}_i \in \pi_j} \phi(\mathbf{x}_i)$ . Thus elements of  $A$  are known in terms of the elements of kernel matrix, and hence the spectral values of  $\bar{C}_{\phi(\mathcal{X})|\mathcal{Y}_j}^k$  can be easily obtained.

We demonstrate the efficacy of our proposed metric using simulations on synthetic and standard datasets in the experiments section. Additionally, we analytically solve for the persistence of clustering solutions for an example problem as shown in Figure 3. The Figure illustrates two equally sized circular clusters ( $k_t = 2$ ) with uniform distribution. One can easily compute the cluster covariance matrices for clustering solutions (as given by  $k$ -means) at various  $k$ 's and show that  $v(2) > \eta \forall \eta \in \{v(3), v(4), v(5), v(6)\}$ ; thereby implying that for the dataset in Figure 3,  $k = 2$  is a *more* natural choice of the number of clusters than  $k = 3, 4, 5$  and  $6$ . Please refer to the supplementary material<sup>1</sup> for the proof.

## Experiments

In this section we employ the notion of persistence in estimating the true number of clusters in synthetic as well as standard datasets from the literature (Dheeru and Karra Taniskidou 2017), (Fränti and Sieranoja 2018) and provide comparisons with the gap-statistic method (Tibshirani, Walther, and Hastie 2001), dip-means (Kalogeratos and Likas 2012),  $X$ -means (Pelleg, Moore, and others 2000),

$G$ -means (Hamerly and Elkan 2004),  $PG$ -means (Feng and Hamerly 2007) algorithms and the information theoretic approach (Sugar and James 2003). We observe that our proposed method outperforms the existing methods on various standard datasets as well as on synthetic datasets with acute overlap between two clusters.

As the method proposed in this paper evaluates a clustering solution for its persistence, any suitable clustering algorithm can be employed to obtain these clustering solutions. Note that irrespective of the criteria or metric used, a stable and good clustering solution at each  $k$  is a prerequisite to correctly estimating the true number of clusters in a dataset. Since the  $k$ -means algorithm is ubiquitous in the data science literature, in all our simulations on linearly separable datasets we use  $k$ -means algorithm with multiple runs to determine the clustering solution at each value of  $k$ . For the purpose of estimating the true number of shape clusters we use the spectral clustering algorithm (Ng, Jordan, and Weiss 2002) to obtain the clustering solutions at various  $k$ 's. Also note that for all the simulations demonstrated in this section we normalize every dataset to mean zero and standard deviation one.

Figure 2 demonstrates multiple instances of synthetic datasets, and the corresponding plots of  $v(k)$  versus  $k$ , where  $k$  is the choice for the number of clusters. Figure 2a illustrates a mixture of four well separated Gaussian distributions. The corresponding plot of  $v(k)$  versus  $k$  shows a clear peak at  $k = 4$ . Figure 2b illustrates another mixture of four Gaussian distributions with very high variances as compared to the Figure 2a. As can be seen, the sampled data is highly



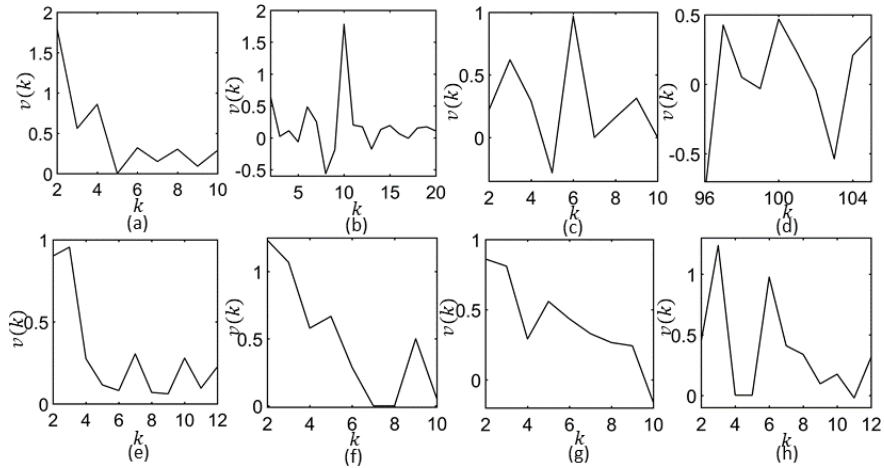


Figure 4: Illustration of performance of our method on high-dimensional datasets - (a) Wisconsin ( $d = 9$ ,  $N = 681$ ,  $k_t = 2$ ), (b) Yeast ( $d = 8$ ,  $N = 1484$ ,  $k_t = 10$ ), (c) Glass ( $d = 9$ ,  $N = 214$ ,  $k_t = 6$ ) (d) Leaves ( $d = 64$ ,  $N = 1600$ ,  $k_t = 100$ ) (e) Wine ( $d = 13$ ,  $N = 178$ ,  $k_t = 3$ ) (f) Iris ( $d = 4$ ,  $N = 150$ ,  $k_t = 3$ ) (g) Banknote Authentication ( $d = 4$ ,  $N = 1372$ ,  $k_t = 2$ ) (h) Thyroid ( $d = 5$ ,  $N = 215$ ,  $k_t = 3$ ). Note that, except for the Iris dataset,  $v(k_t)$  is maximum in all the plots.

overlapping and it seems that the entire dataset is sampled from a single distribution. As seen in the corresponding  $v(k)$  versus  $k$  plot in the Figure 2b, our method exhibits a maximum value of  $v(k)$  at  $k = 4$ . On the other hand, the algorithms such as  $G$ -means,  $PG$ -means and dip-means fail to estimate the true number of clusters in this case (as shown in Table 1), even though this dataset satisfies the assumptions required by these algorithms. In Figure 2c, the dataset is a mixture of well separated eight non-uniform clusters. The corresponding plot of  $v(k)$  versus  $k$  determines two distinguishable peaks at  $k = 4$  and  $k = 8$ , although the peak is larger at  $k = 8$  and denotes the true number of clusters in this dataset.

Similar conclusions are observed for the linearly separable S15 dataset (a mixture of 15 Gaussian distribution (Fränti and Virmajoki 2006)) in Figure 2(d), and other standard high-dimensional datasets - Wisconsin ( $d = 9$ ,  $N = 681$ ,  $k_t = 2$ ), Yeast ( $d = 8$ ,  $N = 1484$ ,  $k_t = 10$ ), Glass ( $d = 9$ ,  $N = 214$ ,  $k_t = 6$ ), Leaves ( $d = 64$ ,  $N = 1600$ ,  $k_t = 100$ ), Wine ( $d = 13$ ,  $N = 178$ ,  $k_t = 3$ ), Iris ( $d = 4$ ,  $N = 150$ ,  $k_t = 3$ ), Banknote Authentication ( $d = 4$ ,  $N = 1372$ ,  $k_t = 2$ ), Thyroid ( $d = 5$ ,  $N = 215$ ,  $k_t = 3$ ), shown in Figure 4, where our proposed method estimates the true number of clusters appropriately. In particular, note that our proposed metric estimates the true number of clusters for a very high-dimensional leaves ( $d = 64$ ,  $k_t = 100$ ) dataset and a large Birch1 ( $N = 100,000$ ,  $k_t = 100$ ) dataset. All the other methods, except Information Theoretic approach on Birch1 dataset, fail to determine the true number of clusters in these two datasets. Table 1 compares our method to the gap-statistic,  $X$ -means,  $G$ -means,  $PG$ -means, dip-means, kernel dip-means algorithms and information theoretic approach of determining the true number of clusters in the dataset. We observe that our method estimates the correct value of  $k_t$  even for the high-variance Gaussian distribution Figure 2(b), yeast dataset, banknote authentication dataset

and a high-dimensional Leaves dataset ( $d = 64$ ) where all the other methods fail to correctly estimate the true number of clusters. As previously noted, the proposed metric is scalable with respect to the size of the datasets and involves eigenvalue computation of a  $d \times d$  cluster covariance matrix, where  $d$  is the dimension of datapoints in the dataset. For the Iris dataset most of the methods, including ours, estimate the true number of clusters to be 2. This is because the two of three clusters in the Iris dataset have significant overlap with each other. However, the gap-statistic method outperforms all the others and estimates correctly the true number of clusters in the Iris dataset.

As described earlier, our technique extends to non-linearly separable data too. Figure 2e and 2f illustrate two non-linearly separable datasets. We use the spectral clustering algorithm (Ng, Jordan, and Weiss 2002) and determine the similarity graph using the Gaussian similarity function,  $s(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / (2\sigma^2))$ , where the  $\sigma$  parameter is set to be 0.01 and 0.08 for (e) concentric rings, and (f) spirals, respectively. Table 1 shows comparison with the kernel dip-means (Kalogeratos and Likas 2012) algorithm on these two non-linearly separable datasets. Note that the latter fails to correctly estimate the true number for clusters on the spiral dataset.

## Conclusion

In this paper we study the persistence of a clustering solution both qualitatively and quantitatively. We use this persistence of clustering solutions to propose a simple yet effective approach for estimating the true number of clusters in a dataset. The key idea used here is to map the distinctiveness of number of clusters to phase-transition phenomenon occurring in the deterministic annealing algorithm. Moreover, we extend the results on linearly separable data to clustering of shapes (nonlinearly separable data) using kernel embedding. The proposed method does not make any assumptions

on the underlying distributions that generate the dataset. Our simulations demonstrate the efficacy of this uncomplicated approach and experimental results shows that our method outperforms many other existing methods in literature.

## References

- Akaike, H. 2011. Akaike information criterion. In *International encyclopedia of statistical science*. Springer. 25–25.
- Andreopoulos, B.; An, A.; Wang, X.; and Schroeder, M. 2009. A roadmap of clustering algorithms: finding a match for a biomedical application. *Briefings in Bioinformatics* 10(3):297–314.
- Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)* 1–38.
- Dheeru, D., and Karra Taniskidou, E. 2017. UCI machine learning repository.
- Dhillon, I. S.; Guan, Y.; and Kulis, B. 2004. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 551–556. ACM.
- Di Nuovo, A. G., and Catania, V. 2008. An evolutionary fuzzy c-means approach for clustering of bio-informatics databases. In *Fuzzy Systems, 2008. FUZZ-IEEE 2008. (IEEE World Congress on Computational Intelligence). IEEE International Conference on*, 2077–2082. IEEE.
- Feng, Y., and Hamerly, G. 2007. Pg-means: learning the number of clusters in data. In *Advances in neural information processing systems*, 393–400.
- Fränti, P., and Sieranoja, S. 2018. K-means properties on six clustering benchmark datasets.
- Fränti, P., and Virtajoki, O. 2006. Iterative shrinking method for clustering problems. *Pattern Recognition* 39(5):761–765.
- Gersho, A., and Gray, R. M. 2012. *Vector quantization and signal compression*, volume 159. Springer Science & Business Media.
- Hamerly, G., and Elkan, C. 2004. Learning the k in k-means. In *Advances in neural information processing systems*, 281–288.
- Hartigan, J. A., and Hartigan, P. M. 1985. The dip test of unimodality. *The Annals of Statistics* 70–84.
- Hartigan, J. A., and Wong, M. A. 1979. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28(1):100–108.
- Huang, C.-W.; Lin, K.-P.; Wu, M.-C.; Hung, K.-C.; Liu, G.-S.; and Jen, C.-H. 2015. Intuitionistic fuzzy c-means clustering algorithm with neighborhood attraction in segmenting medical image. *Soft Computing* 19(2):459–470.
- Kalogeratos, A., and Likas, A. 2012. Dip-means: an incremental clustering method for estimating the number of clusters. In *Advances in neural information processing systems*, 2393–2401.
- Kass, R. E., and Wasserman, L. 1995. A reference bayesian test for nested hypotheses and its relationship to the schwarz criterion. *Journal of the american statistical association* 90(431):928–934.
- Kurihara, K., and Welling, M. 2009. Bayesian k-means as a maximization-expectation algorithm. *Neural computation* 21(4):1145–1172.
- Lange, T.; Braun, M. L.; Roth, V.; and Buhmann, J. M. 2003. Stability-based model selection. In *Advances in neural information processing systems*, 633–642.
- Larose, D. T., and Larose, C. D. 2014. *Discovering knowledge in data: an introduction to data mining*. John Wiley & Sons.
- Levine, E., and Domany, E. 2001. Resampling method for unsupervised estimation of cluster validity. *Neural computation* 13(11):2573–2593.
- Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2002. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, 849–856.
- Park, H.-S., and Jun, C.-H. 2009. A simple and fast algorithm for k-medoids clustering. *Expert systems with applications* 36(2):3336–3341.
- Pelleg, D.; Moore, A. W.; et al. 2000. X-means: Extending k-means with efficient estimation of the number of clusters. In *Icml*, volume 1, 727–734.
- Rissanen, J. 1978. Modeling by shortest data description. *Automatica* 14(5):465–471.
- Rose, K. 1998. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of the IEEE* 86(11):2210–2239.
- Sand, P., and Moore, A. W. 2001. Repairing faulty mixture models using density estimation. In *ICML*, 457–464. Citeseer.
- Sharma, P.; Salapaka, S.; and Beck, C. 2008. A scalable approach to combinatorial library design for drug discovery. *Journal of chemical information and modeling* 48(1):27–41.
- Stephens, M. A. 1974. Edf statistics for goodness of fit and some comparisons. *Journal of the American statistical Association* 69(347):730–737.
- Sugar, C. A., and James, G. M. 2003. Finding the number of clusters in a dataset: An information-theoretic approach. *Journal of the American Statistical Association* 98(463):750–763.
- Tibshirani, R., and Walther, G. 2005. Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics* 14(3):511–528.
- Tibshirani, R.; Walther, G.; and Hastie, T. 2001. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63(2):411–423.
- Yuan, F.; Zhan, Y.; and Wang, Y. 2014. Data density correlation degree clustering method for data aggregation in wsn. *IEEE Sensors Journal* 14(4):1089–1098.



## Supplementary material: On the number of clusters in data

### Additional Experimental Results

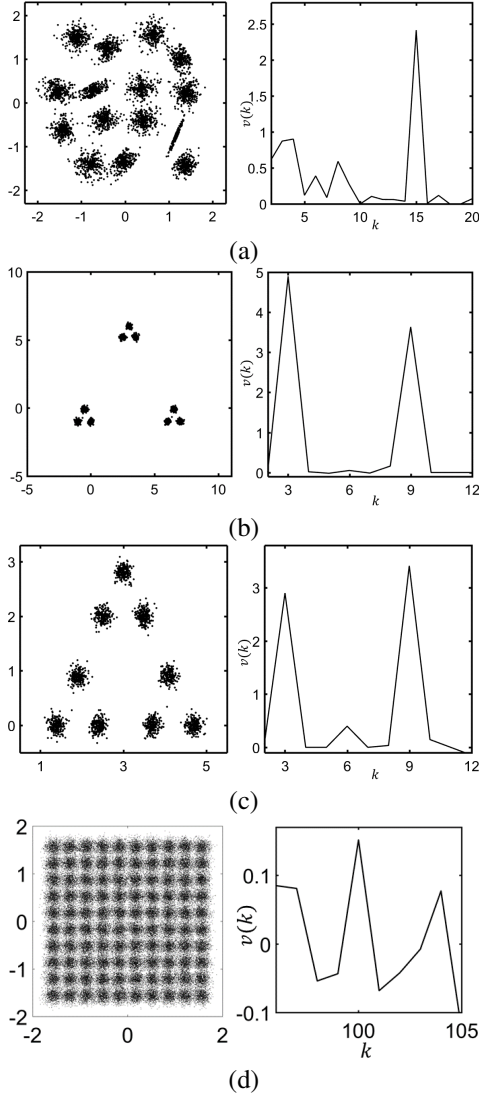


Figure 5: (a) Synthetic 2-d data with  $N = 5000$  vectors and  $k_t = 15$  Gaussian clusters (Fränti and Virtajoki 2006) with different degree of overlapping. The corresponding  $v(k)$  versus  $k$  plot for both the cases show a peak at  $k = 15$ . (b) The  $v(k)$  versus  $k$  plot indicates 3 to be true number of clusters. (c) The  $v(k)$  versus  $k$  plot indicates 9 to be true number of clusters. (d) The  $v(k)$  versus  $k$  plot indicates  $k_t = 100$  for Birch1 dataset.

### Proof of Lemma1

**Proof:** Let  $(\lambda, \mathbf{u})$  be any eigenvector-eigenvalue pair of  $A$ .

Here  $\mathbf{u} \triangleq [u_1, u_2, \dots, u_N]$ . Then, by definition:

$$\sum_{l=1}^N A_{kl} u_l = \lambda u_k$$

$$\Rightarrow (\phi(\mathbf{x}_k) - \mathbf{y})^T \sum_{l=1}^N (\phi(\mathbf{x}_l) - \mathbf{y}) u_l = \lambda u_k \quad (14)$$

Multiplying (14) by  $(\phi(\mathbf{x}_k) - \mathbf{y})$  and then summing over  $k$  yields:

$$\sum_{k=1}^N (\phi(\mathbf{x}_k) - \mathbf{y}) (\phi(\mathbf{x}_k) - \mathbf{y})^T \underbrace{\sum_{l=1}^N (\phi(\mathbf{x}_l) - \mathbf{y}) u_l}_{\tilde{\mathbf{u}}}$$

$$= \lambda \underbrace{\sum_{k=1}^N (\phi(\mathbf{x}_k) - \mathbf{y})}_{\tilde{\mathbf{u}}} u_k,$$

$$\Rightarrow C_{\phi(\mathcal{X})|\mathcal{Y}_j}^k \tilde{\mathbf{u}} = \lambda \tilde{\mathbf{u}} \quad (15)$$

### Proof for $H(\mathcal{Y}, \Psi, \beta) = 0$ only when $\det(I - 2\beta C_{\mathcal{X}|\mathcal{Y}_0}^k) = 0$

We claim that  $H(\mathcal{Y}, \Psi, \beta)$  is non-negative for all finite perturbation  $\Psi$  if and only if the matrix  $[I - 2\beta C_{\mathcal{X}|\mathcal{Y}_0}^k]$  is positive definite. The 'if' part is straightforward since the second term in the expression is non-negative. For the 'only if' part we show that when  $[I - 2\beta C_{\mathcal{X}|\mathcal{Y}_0}^k]$  is negative semi-definite, there exists a finite perturbation  $\Psi$  such that the second term in  $H(\mathcal{Y}, \Psi, \beta)$  becomes zero thereby making the entire term negative. Let us assume that there exists a  $\mathbf{y}_0 \in \mathcal{Y}$  with positive probability such that the matrix  $[I - 2\beta C_{\mathcal{X}|\mathcal{Y}_0}^k]$  is negative semi-definite. The perturbation  $\Psi$  be such that  $\Psi_{\mathbf{y}} = 0 \forall \mathbf{y} \neq \mathbf{y}_0$  and  $\sum_{\mathbf{y} \in \mathcal{Y}: \mathbf{y} = \mathbf{y}_0} \Psi_{\mathbf{y}} = 0$ . Then the second term in  $H(\mathcal{Y}, \Psi, \beta)$  becomes zero. Thus whenever the first term in  $H(\mathcal{Y}, \Psi, \beta)$  is non-positive we can construct a perturbation such that the second term vanishes. Hence the positivity of the  $H(\mathcal{Y}, \Psi, \beta)$  for all perturbations  $\Psi$  depends solely on the positive definiteness of  $[I - 2\beta C_{\mathcal{X}|\mathcal{Y}_0}^k]$ . The  $\mathcal{Y}$  is no longer a minimum of  $F$  when  $H(\mathcal{Y}, \Psi, \beta) = 0$  which happens when the matrix  $[I - 2\beta C_{\mathcal{X}|\mathcal{Y}_0}^k]$  loses its positive definiteness; i.e.

$$\det[I - 2\beta C_{\mathcal{X}|\mathcal{Y}_0}^k] = 0$$

The proof is as given in (Rose 1998).

### Proof for $v(2) > v(k) \forall k \in \{3, 4, 5, 6\}$

Figure 6 illustrates an optimal clustering solution at various values of  $k$ . We assume that each circle contains  $N$  data-points. We have rotated the original dataset in Figure 3 by 90 degrees for the ease of presentation; however, this will not change the corresponding problem and its solution. Note that Figure 6 shows one of the optimal clusterings at each value of  $k$ ; even if some other optimal clustering is chosen,

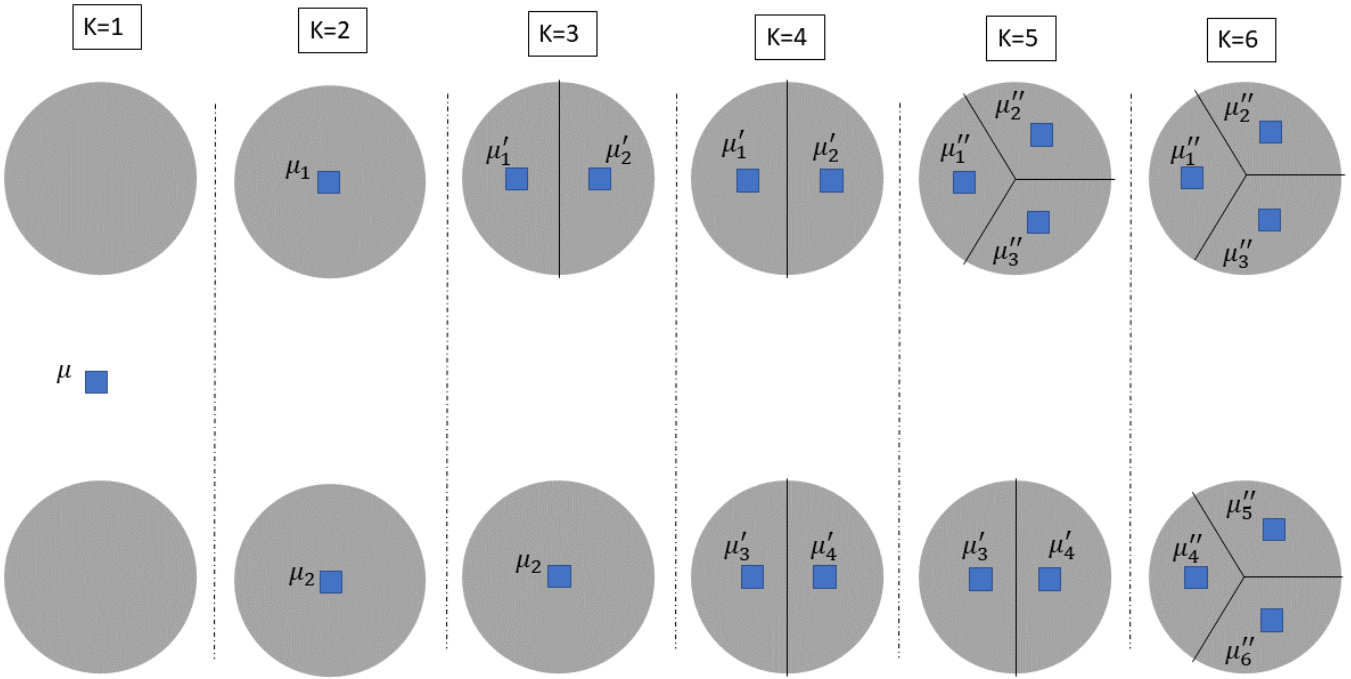


Figure 6: The Figure illustrates the optimal clustering at each value of  $k \in \{1, 2, 3, 4, 5, 6\}$  for the dataset in the Figure 3

due to symmetry, the cluster covariance matrix will remain unchanged. To evaluate  $\beta_k \forall k \in \{2, 3, 4, 5, 6\}$ , we compute the relevant covariance matrices  $\bar{C}_{\mathcal{X}|\mu}^1$ ,  $\bar{C}_{\mathcal{X}|\mu_1}^2$ ,  $\bar{C}_{\mathcal{X}|\mu_2}^3$ ,  $\bar{C}_{\mathcal{X}|\mu'_1}^4$ ,  $\bar{C}_{\mathcal{X}|\mu'_3}^5$  and  $\bar{C}_{\mathcal{X}|\mu''_1}^6$ .

- Computing  $\bar{C}_{\mathcal{X}|\mu_1}^2$ .

$$\begin{aligned} \bar{C}_{\mathcal{X}|\mu_1}^2 &= \sum_{x \in C_{\mu_1}} (x - \mu_1)(x - \mu_1)^T \\ &\approx \int_{r, \theta} (x - \mu_1)(x - \mu_1)^T \rho r dr d\theta, \end{aligned}$$

where  $x = [r \cos \theta \quad r \sin \theta]^T$

$$= \int_0^{2\pi} \int_0^R \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} [r \cos \theta \quad r \sin \theta] \rho r dr d\theta$$

we assume  $\mu_1 = [0 \quad 0]^T$  without loss of generality

$$= \int_0^{2\pi} \int_0^R \begin{bmatrix} r^2 \cos^2 \theta & r^2 \cos \theta \sin \theta \\ r^2 \sin \theta \cos \theta & r^2 \sin^2 \theta \end{bmatrix} \rho r dr d\theta$$

$$= \begin{bmatrix} (\rho\pi R^2)R^2/4 & 0 \\ 0 & (\rho\pi R^2)R^2/4 \end{bmatrix}$$

$(\rho\pi R^2)$  denotes the number of data points  $N$

$$= \frac{NR^2}{4} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\Rightarrow \lambda_{\max}(\bar{C}_{\mathcal{X}|\mu_1}^2) = NR^2/4$$

- Computing  $\bar{C}_{\mathcal{X}|\mu_2}^3$ . Due to symmetry between the cluster  $\pi_{\mu_1}$  and  $\pi_{\mu_2}$  we can see that  $\bar{C}_{\mathcal{X}|\mu_1}^2 = \bar{C}_{\mathcal{X}|\mu_2}^3$ . Hence

$$\Rightarrow \lambda_{\max}(\bar{C}_{\mathcal{X}|\mu_2}^3) = NR^2/4$$

In Figure 6, we assume WLOG that  $\pi_{\mu_1}$  splits and  $\pi_{\mu_2}$  remains intact between  $k = 2$  and  $k = 3$ .

- Computing  $\bar{C}_{\mathcal{X}|\mu}^1$

$$\begin{aligned} \bar{C}_{\mathcal{X}|\mu}^1 &= \sum_{x \in C_{\mu}} (x - \mu)(x - \mu)^T \\ &= \sum_{x \in C_{\mu_1}} (x - \mu)(x - \mu)^T + \sum_{x \in C_{\mu_2}} (x - \mu)(x - \mu)^T \\ &= \sum_{x \in C_{\mu_1}} (x - \mu_1 + (\mu_1 - \mu))(x - \mu_1 + (\mu_1 - \mu))^T \\ &\quad + \sum_{x \in C_{\mu_2}} (x - \mu_2 + (\mu_2 - \mu))(x - \mu_2 + (\mu_2 - \mu))^T \\ &= \bar{C}_{\mathcal{X}|\mu_1}^2 + \bar{C}_{\mathcal{X}|\mu_2}^2 + 2N \left( \frac{\mu_1 - \mu_2}{2} \right) \left( \frac{\mu_1 - \mu_2}{2} \right)^T \end{aligned}$$

using the definition of  $\bar{C}_{\mathcal{X}|\mu_1}^2$  and  $\bar{C}_{\mathcal{X}|\mu_2}^2$

$$\text{and } \mu = \frac{\mu_1 + \mu_2}{2}$$

note that  $\mu_1 - \mu_2 = [0 \quad 4R]^T$

$$= \begin{bmatrix} NR^2/2 & 0 \\ 0 & NR^2/2 + 8NR^2 \end{bmatrix}$$

$$= NR^2 \begin{bmatrix} 1 & 0 \\ 0 & 8.5 \end{bmatrix}$$

$$\Rightarrow \lambda_{\max}(\bar{C}_{\mathcal{X}|\mu}^1) = 8.5NR^2$$

- Computing  $\bar{C}_{\mathcal{X}|\mu'_1}^4$ . Since  $\pi_{\mu'_1}$  is a semi-circle, we can easily locate the centroid  $\mu'_1$  in it. Assuming that the origin  $(0, 0)$  is at the center of the full circle corresponding to

this semi-circle, hence we have that  $\mu'_1 = [0 \quad \frac{4R}{3\pi}]^T$ .

$$\begin{aligned} \bar{C}_{\mathcal{X}|\mu'_1}^4 &= \sum_{x \in \mathcal{C}_{\mu'_1}} (x - \mu'_1)(x - \mu'_1)^T \\ &\approx \int_0^R \int_0^\pi \begin{bmatrix} r \cos \theta & \\ r \sin \theta - \frac{4R}{3\pi} & \end{bmatrix} \begin{bmatrix} r \cos \theta & \\ r \sin \theta - \frac{4R}{3\pi} & \end{bmatrix}^T \rho r dr d\theta \\ &= NR^2 \begin{bmatrix} \frac{1}{8} & 0 \\ 0 & \frac{1}{8} + \frac{8}{9\pi^2} - \frac{16}{9\pi^2} \end{bmatrix} \\ &\Rightarrow \lambda_{\max}(\bar{C}_{\mathcal{X}|\mu'_1}^4) = 0.125NR^2 \end{aligned}$$

- Computing  $\bar{C}_{\mathcal{X}|\mu'_3}^5$ . Note that due to symmetry  $\bar{C}_{\mathcal{X}|\mu'_1}^4 = \bar{C}_{\mathcal{X}|\mu'_3}^5$ .

$$\Rightarrow \lambda_{\max}(\bar{C}_{\mathcal{X}|\mu'_3}^5) = 0.125NR^2$$

- Computing  $\bar{C}_{\mathcal{X}|\mu''_1}^6$ . Can be calculated as done above. We get

$$\begin{aligned} \bar{C}_{\mathcal{X}|\mu''_1}^6 &= NR^2 \begin{bmatrix} 0.0165 & 0 \\ 0 & 0.049 \end{bmatrix} \\ &\Rightarrow \lambda_{\max}(\bar{C}_{\mathcal{X}|\mu''_1}^6) = 0.049NR^2 \end{aligned}$$

Hence we have that  $\bar{\beta}_1 = \frac{1}{17NR^2}$ ,  $\bar{\beta}_2 = \frac{1}{2NR^2/4}$ ,  $\bar{\beta}_3 = \frac{1}{2NR^2/4}$ ,  $\bar{\beta}_4 = \frac{1}{0.125NR^2}$ ,  $\bar{\beta}_5 = \frac{1}{0.125NR^2}$ ,  $\bar{\beta}_6 = \frac{\pi}{0.98NR^2}$

$$\begin{aligned} v(2) &= \log \bar{\beta}_2 - \log \bar{\beta}_1 = 3.53 \\ v(3) &= \log \bar{\beta}_3 - \log \bar{\beta}_2 = 0 \\ v(4) &= \log \bar{\beta}_4 - \log \bar{\beta}_3 = 0.69 \\ v(5) &= \log \bar{\beta}_5 - \log \bar{\beta}_4 = 0 \\ v(6) &= \log \bar{\beta}_6 - \log \bar{\beta}_5 = 0.91 \end{aligned}$$

Hence we have that  $v(2) > v(k) \forall k \in \{3, 4, 5, 6\}$  as summarized in the Figure 7

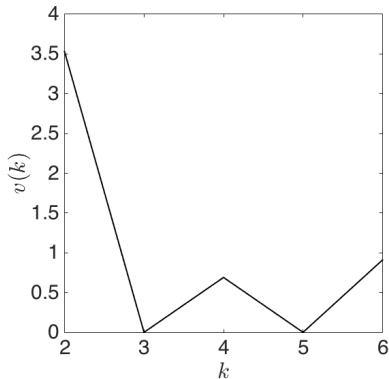


Figure 7: Illustrates the  $v(k)$  vs  $k$  plot for the analytically calculated  $v(k)$  at various  $k$ 's for the dataset in Figure 3. As shown,  $k = 2$  is a more natural number of cluster than  $k \in \{3, 4, 5, 6\}$ .

## References to algorithmic implementation of various other metrics

- Gap-statistic method: MATLAB in-built class *cluster-evaluation.GapEvaluation* class.
- X-meas, G-means, dip-means, kernel dip-means: <http://kalogeratos.com/psite/material/dip-means/>
- PG-means: Implementation provided by the author of this method (Feng and Hamerly 2007).
- Information theoretic method: Self-implemented on MATLAB.