Clustering - Classification non-supervisée

Alexandre Gramfort

alexandre.gramfort@inria.fr

Inria - Université Paris-Saclay

Huawei Mathematical Coffee
March 16 2018

Outline

What is clustering?

- One of the most widely used techniques for exploratory data analysis
- Get intuition about data by identifying meaningful groups among the data points
- Knowledge discovery

Examples

- Identify groups of customers for targeted marketing
- Identify groups of similar individuals in a social network
- Identify groups of genes based on their expresssions (phenotypes)

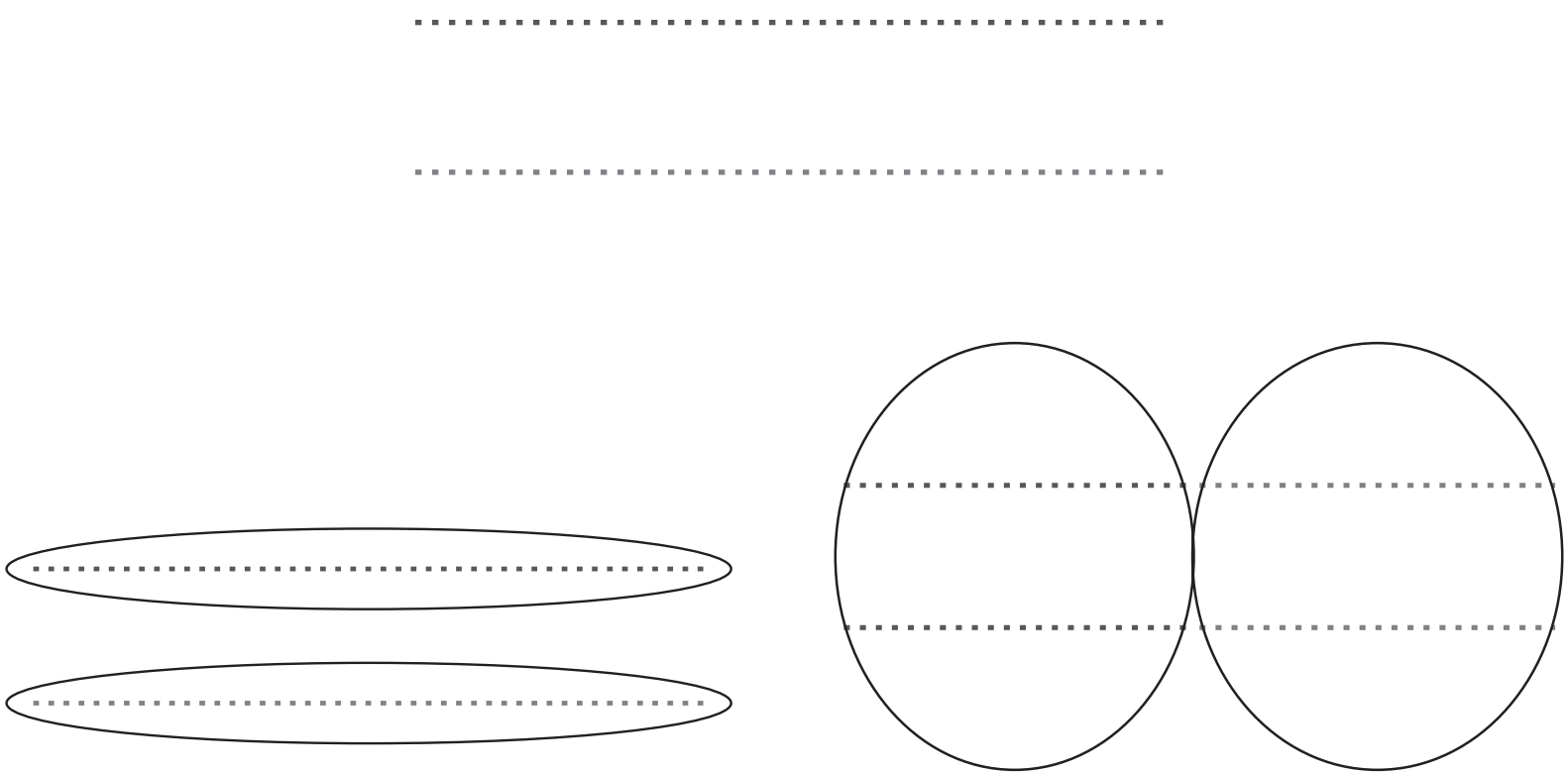
A fuzzy definition

Definition (Clustering)

Task of grouping a set of objects such that similar objects end up in the same group and dissimilar objects are separated into different groups.

- More rigorous definition not so obvious
- Clustering is a transitive relation
- Similarity is not: imagine $x_1, \ldots, x_m$ such that each x_i is very similar to its two neighbors, x_{i-1} and x_{i+1}, but x_1 and x_m are very dissimilar.

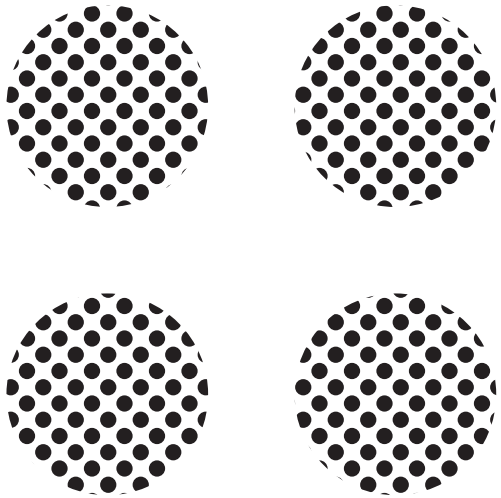
Illustration



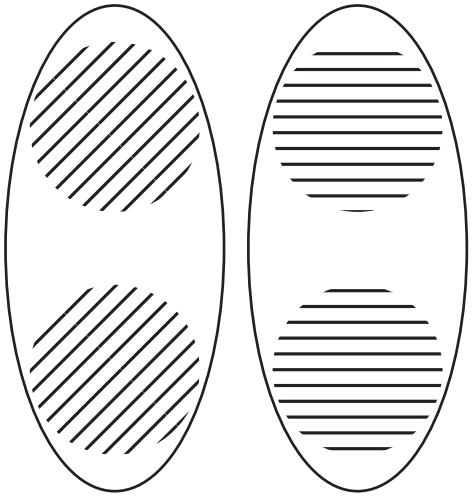
Absence of ground truth

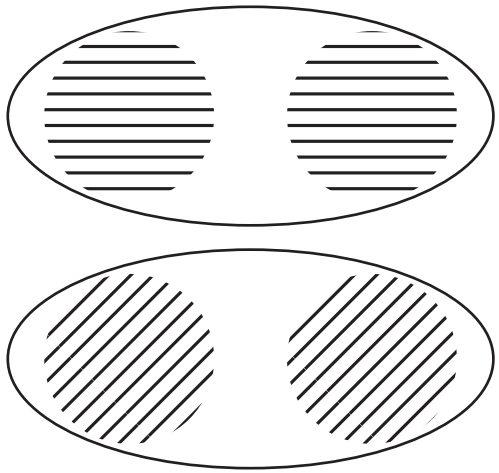
- Clustering is an unsupervised learning problem (learning from unlabeled data).
- For supervised learning the metric of performance is clear
- For clustering there is no clear success evaluation procedure
- For clustering there is no ground truth
- For clustering it is unclear what the correct answer is

Absence of ground truth



Both of these solutions are equally justifiable solutions:



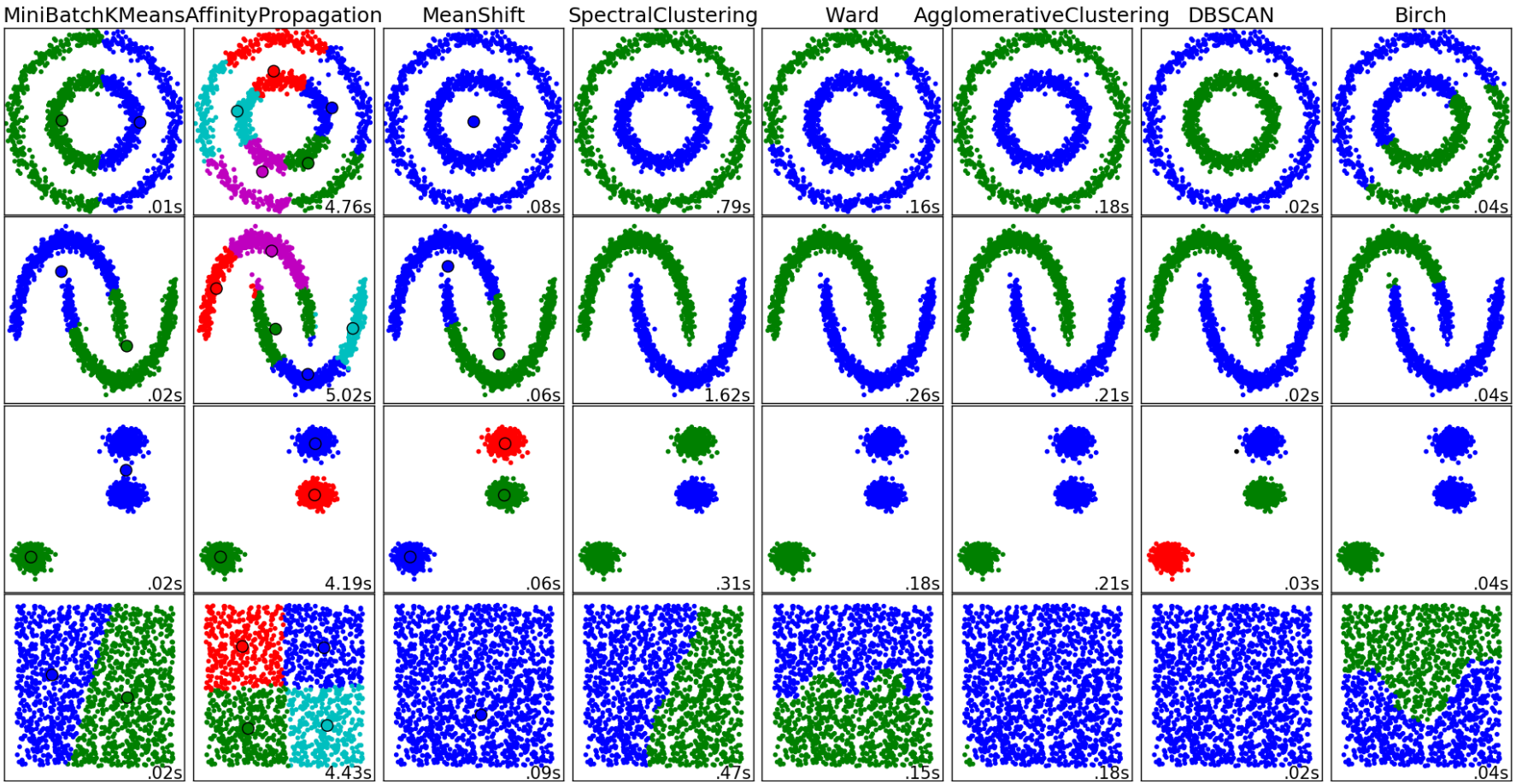


To sum up

Summary

- There may be several very different conceivable clustering solutions for a given data set.
- As a result, there is a wide variety of clustering algorithms that, on some input data, will output very different clusterings.

Zoology of clustering methods



Source: http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

A clustering model

Input

- A set of elements, $\mathcal{X}$, and a distance function over it. That is, a function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ that is symmetric, satisfies $d(x, x) = 0$ for all $x \in \mathcal{X}$, and (often) also satisfies the triangle inequality.
- Alternatively, the function could be a similarity function $s : \mathcal{X} \times \mathcal{X} \to [0, 1]$ that is symmetric and satisfies $s(x, x) = 1$ for all $x \in \mathcal{X}$.
- Also, clustering algorithms typically require:
  - a parameter k (determining the number of required clusters).
  - or a bandwidth / threshold parameter ϵ (determining how close points in a same cluster should be).

A clustering model

Output

- A partition of the domain set $\mathcal{X}$ into subsets:
  - $C = (C_1, \ldots, C_k)$ where $\cup_{i=1}^{k} C_i = \mathcal{X}$ and for all $i \neq j$, $C_i \cap C_j = \emptyset$.
- In some situations the clustering is "soft". The output is a probabilistic assignment to each domain point:
  - $\forall x \in \mathcal{X}$, we get $(p_1(x), \ldots, p_k(x))$, where $p_i(x) = P[x \in C_i]$ is the probability that x belongs to cluster C_i.
- Another possible output is a clustering dendrogram, which is a hierarchical tree of domain subsets, having the singleton sets in its leaves, and the full domain as its root.

Outline

History

- k-means is certainly the most well known clustering algorithm
- The k-means algorithm is attributed to Lloyd (1957) and was only published in a journal in 1982.
- There is a lot of misunderstanding on the underlying hypothesis
- ... and the limitations
- There is still a lot of research to speed up this algorithm (k-means++ initialization [Arthur et al. 2007], online k-means [Sculley 2010], triangular inequality trick [Elkan ICML 2003], Yinyang k-means [Ding et al. ICML 2015], better initialization [Bachem et al. NIPS 2016]).

Cost minimization clusterings

- Find a partition $C = (C_1, \ldots, C_k)$ of minimal cost
- $G((\mathcal{X}, d), C)$ is the objective to be minimized

Note

- Most of the resulting optimization problems are NP-hard, and some are even NP-hard to approximate.
- Consequently, when people talk about, say, k-means clustering, they often refer to some particular common approximation algorithm rather than the cost function or the corresponding exact solution of the minimization problem.

The k-means objective function

- Data is partitioned into disjoint sets $C_1, \ldots, C_k$ where each C_i is represented by a centroid μ_i.
- We assume that the input set $\mathcal{X}$ is embedded in some larger metric space $(\mathcal{X}', d)$, such as $\mathbb{R}^p$, (so that $\mathcal{X} \subseteq \mathcal{X}'$) and centroids are members of $\mathcal{X}'$.
- k-means objective function measures the squared distance between each point in $\mathcal{X}$ to the centroid of its cluster.

Formally:

$$\mu_i(C_i) = \arg\min_{\mu \in \mathcal{X}'} \sum_{x \in C_i} d(x, \mu)^2$$

$$G_{\text{k-means}}((\mathcal{X}, d), (C_1, \ldots, C_k)) = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i(C_i))^2$$

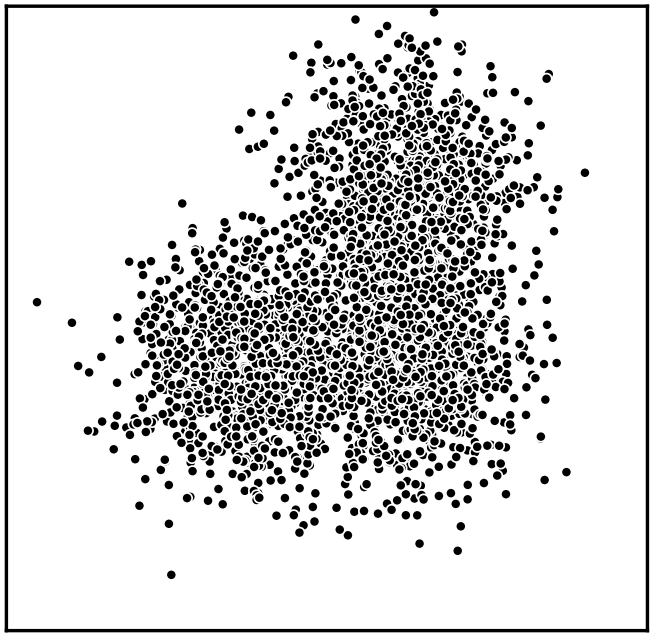
Note: $G_{\text{k-means}}$ is often refered to as *inertia*.

The k-means objective function

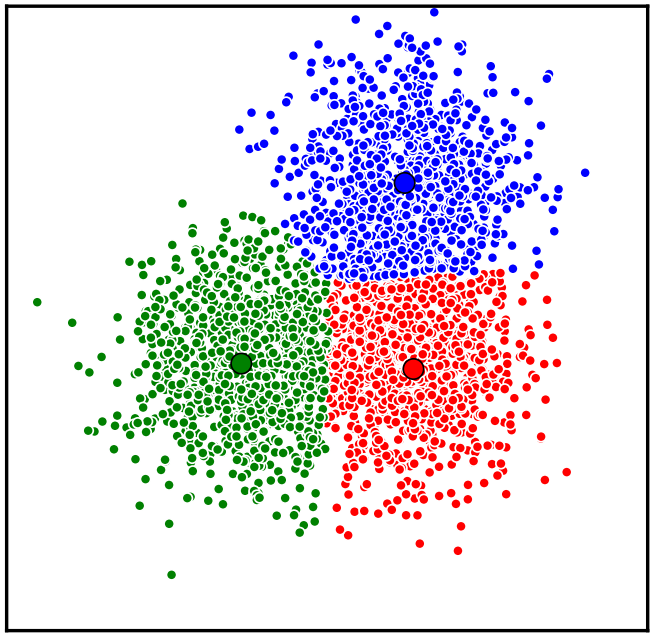
Which can be rewritten:

$$G_{\text{k-means}}((\mathcal{X}, d), (C_1, \ldots, C_k)) = \min_{\mu_1, \ldots \mu_k \in \mathcal{X}'} \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i)^2$$

Samples



KMeans



The k-medoids objective function

Similar to the k-means objective, except that it requires the cluster centroids to be members of the input set:

$$G_{\text{k-medoids}}((\mathcal{X}, d), (C_1, \ldots, C_k)) = \min_{\mu_1, \ldots \mu_k \in \mathcal{X}} \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i)^2$$

The k-median objective function

Similar to the k-medoids objective, except that the "distortion" between a data point and the centroid of its cluster is measured by distance, rather than by the square of the distance:

$$G_{\text{k-median}}((\mathcal{X}, d), (C_1, \ldots, C_k)) = \min_{\mu_1, \ldots \mu_k \in \mathcal{X}} \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i)$$

Example

An example is the facility location problem. Consider the task of locating k fire stations in a city. One can model houses as data points and aim to place the stations so as to minimize the average distance between a house and its closest fire station.

Remarks

- The latter objective functions are center based:

$$G_f((\mathcal{X}, d), (C_1, \ldots, C_k)) = \min_{\mu_1, \ldots \mu_k \in \mathcal{X}'} \sum_{i=1}^{k} \sum_{x \in C_i} f(d(x, \mu_i))$$

- Some objective functions are not center based. For example, the sum of in-cluster distances (SOD)

$$G_{SOD}((\mathcal{X}, d), (C_1, \ldots, C_k)) = \sum_{i=1}^{k} \sum_{x,y \in C_i} d(x, y)$$

k-means algorithm

We describe the algorithm with respect to the Euclidean distance function $d(x, y) = \|x - y\|$.

Algorithm 1 (Vanilla) k-Means algorithm

procedure
Input: $\mathcal{X} \subset \mathbb{R}^n$; Number of clusters k.
Initialize: Randomly choose initial centroids $\mu_1, \ldots, \mu_k$.
Repeat until convergence:

$$\forall i \in [k] \text{ set } C_i = \{x \in \mathcal{X}, i = \arg\min_j \|x - \mu_j\|\}$$

$$\forall i \in [k] \text{ update } \mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

end procedure

k-means algorithm

Theorem (k-means algorithm converges monotonically)

Each iteration of the k-means algorithm does not increase the k-means objective function.

Remark(s)

- No guarantee on the number of iterations to reach convergence.
- There is no nontrivial lower bound on the gap between the value of the k-means objective of the algorithm's output and the minimum possible value of that objective function.
- k-means might converge to a point which is not even a local minimum!
- To improve the results of k-means it is recommended to **repeat the procedure several times with different randomly chosen initial centroids**.

DBSCAN: Density based clustering

- "Density-based spatial clustering of applications with noise" (DBSCAN) is a very popular, simple and powerful algorithm first proposed by Ester et al. 1996 at KDD Conf. ($>$ 11,000 citations).
- DBSCAN is one of the most common clustering algorithms and also most cited in scientific literature.
- In 2014, it was awarded the test of time award at the leading data mining conference, KDD.

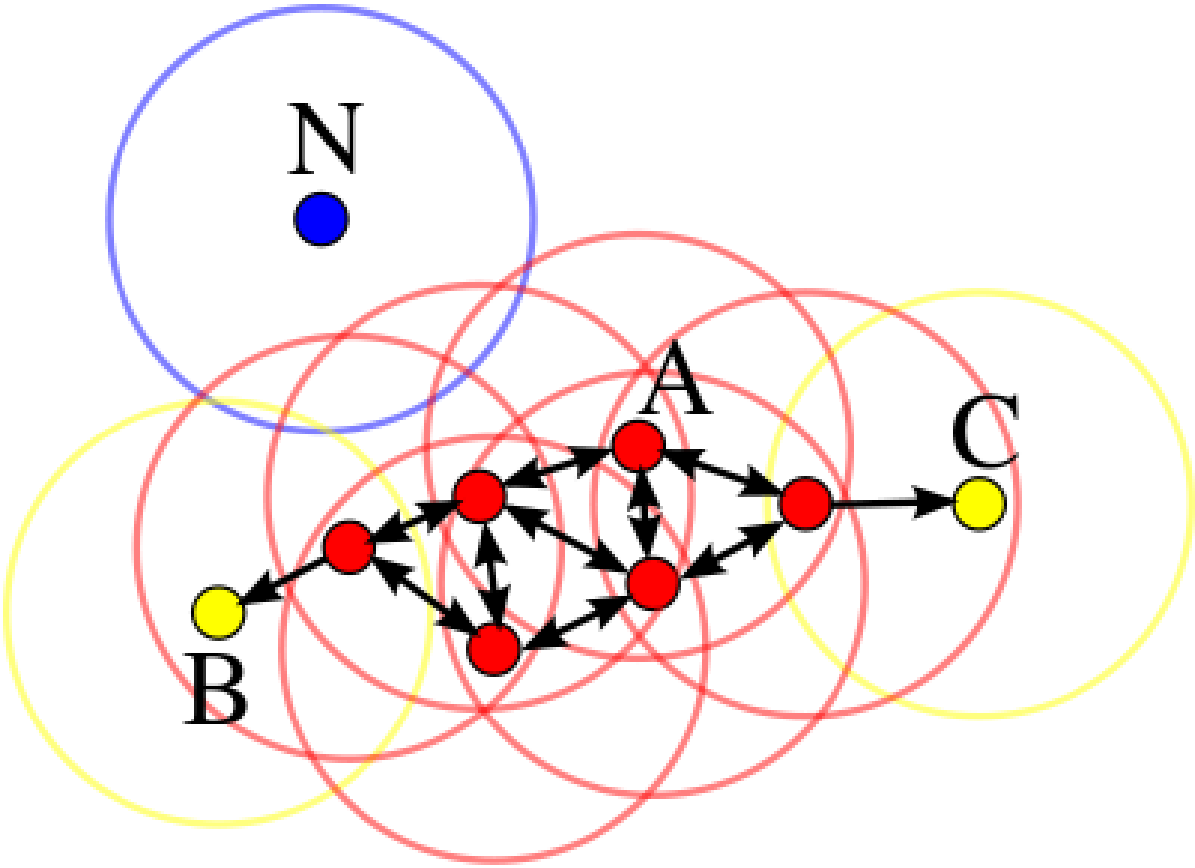
DBSCAN Algorithm

- 2 parameters: ϵ and the minimum number of points required to form a dense region q.
- Start with an arbitrary starting point not yet visited. Retrieve its ϵ-neighborhood. If it contains sufficiently many points, a cluster is started. Otherwise, the point is labeled as noise.[1]
- If a point is found to be a dense part of a cluster, its ϵ-neighborhood is also part of that cluster. All points that are found within the ϵ-neighborhood are added, so is their own ϵ-neighborhood when they are also dense.
- Process continues until the density-connected cluster is completely found.
- Start again with a new point, until all points have been visited.

[1] A point marked as noise might later be found in a sufficiently sized ϵ-environment of a different point and hence be made part of a cluster.

DBSCAN Illustration

With q=4 in 2D:



Red: core points, Yellow: non core but in cluster, Blue: noise

Source: https://en.wikipedia.org/wiki/DBSCAN

Algorithm 2 DBSCAN

procedure DBSCAN($\mathcal{X}$, ϵ, q)
Initialize: $C = 0$.
    for each point x in $\mathcal{X}$ **do**
        if x is visited **then**
            continue to next point.
        end if
        mark x as visited.
        neighbors = getNeighbors(x, ϵ)
        if $|\text{neighbors}| < q$ **then**
            mark x as noise.
        else
            C = next cluster
            expandCluster(x, neighbors, C, ϵ, q)
        end if
    end for
Output: All produced clusters.
end procedure

```
1: procedure expandCluster(x, neighbors, C, ϵ, q)
2:     add x to C
3:     for each y in neighbors do
4:         if y is not visited then
5:             mark y as visited
6:             neighbors_y = regionQuery(y, ϵ)
7:             if |neighbors_y| ≥ q then
8:                 neighbors = neighbors joined with neighbors_y
9:             end if
10:        end if
11:        if y is not yet member of any cluster then
12:            add y to cluster C
13:        end if
14:    end for
15: end procedure
16: procedure regionQuery(x, ϵ)
17:    Output: all points within x's ϵ-neighborhood (including x)
18: end procedure
```

DBSCAN Pros

- No need to specify the number of clusters in the data a priori, as opposed to k-means.
- It can find arbitrarily shaped clusters. It can even find a cluster completely surrounded by (but not connected to) a different cluster.
- Due to the q parameter, the so-called single-link effect (different clusters being connected by a thin line of points) is reduced.
- It has a notion of noise, and is robust to outliers.

DBSCAN Cons

- It is not entirely deterministic (output depends on the order of the points).
- It still needs to specify a distance measure (like k-means or spectral clustering).
- It can not cluster data sets with a large difference in densities as the $q - \epsilon$ combination cannot then be chosen appropriately for all clusters.

Beyond DBSCAN

- Ordering points to identify the clustering structure (OPTICS) [Ankerst et al. ACM SIGMOD 1999] which can detect clusters in data of varying density. [2]
- Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [Campello et al. 2013, McInnes et al. 2017][3].
  - It performs DBSCAN over varying ϵ values and finds the most stable clustering.
  - Like OPTICS it allows to find clusters of varying densities.
  - It is more robust to parameter selection.

[2] Close to Local Outlier Factor (LOF) algorithm for anomaly detection.

[3] `https://github.com/scikit-learn-contrib/hdbscan`

Outline

Food for thoughts

An Impossibility Theorem for Clustering

Jon Kleinberg
Department of Computer Science
Cornell University
Ithaca NY 14853

Abstract

Although the study of *clustering* is centered around an intuitively compelling goal, it has been very difficult to develop a unified framework for reasoning about it at a technical level, and profoundly diverse approaches to clustering abound in the research community. Here we suggest a formal perspective on the difficulty in finding such a unification, in the form of an *impossibility theorem*: for a set of three simple properties, we show that there is no clustering function satisfying all three. Relaxations of these properties expose some of the interesting (and unavoidable) trade-offs at work in well-studied clustering techniques such as single-linkage, sum-of-pairs, k-means, and k-median.

[Kleinberg "An Impossibility Theorem for Clustering", NIPS 2002]

References I

1. Lloyd, S. P. (1957). "Least square quantization in PCM". Bell Telephone Laboratories Paper. Published in journal much later: Lloyd., S. P. (1982). "Least squares quantization in PCM". IEEE Transactions on Information Theory 28 (2): 129–137
2. Elkan, C. (2003). "Using the triangle inequality to accelerate k-means" (PDF). Proceedings of the Twentieth International Conference on Machine Learning (ICML).
3. Arthur, D. and Vassilvitskii, S. (2007). "k-means++: the advantages of careful seeding". Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics Philadelphia, PA, USA. pp. 1027–1035.
4. Ding Y., et al. (2015) "Yinyang K-Means: A Drop-In Replacement of the Classic K-Means with Consistent Speedup", Proceedings of The 32nd International Conference on Machine Learning (ICML), pp. 579–587.

References II

5. Ester M., Kriegel H.-P., Sander J., and Xu X. (1996), "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining (KDD), pp. 226–231.
6. R. Campello, D. Moulavi, and J. Sander (2013), "Density-Based Clustering Based on Hierarchical Density Estimates" In: Advances in Knowledge Discovery and Data Mining, Springer, pp 160-172. 2013
7. McInnes L, Healy J. (2017), "Accelerated Hierarchical Density Based Clustering" In: 2017 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, pp 33-42.
8. Sculley D. (2010), "Web-Scale K-Means Clustering" In: Proceeding WWW '10 Proceedings of the 19th international conference on World wide web, pp 1177-1178
9. Ankerst M, Breunig M M, Kriegel H-P, Sander J (1999). "OPTICS: Ordering Points To Identify the Clustering Structure". ACM SIGMOD international conference.

References III

10 Kleinberg J M (2002), "An Impossibility Theorem for Clustering", Advances in Neural Information Processing Systems 15, pp 463-470