Multi-Manifold Clustering*

!Optimizer

Sharif Optimization and Applications Laboratory
Department of Mathematical Sciences
Sharif University of Technology

February 2022

This document is the **problem description** for the !Optimizer competition, that is going to be held in Summer 2022. In what follows, the general setup of the problem is explained in Section 1, and some information regarding the format of the input and the output files as well as some extra information about submission rules and the scoring procedure are also presented in Section 2. A description of each one of the rounds of the competition is explained in Section 3, where details of what is going to be provided to competitors, along with what is expected to be submitted by teams during the round are recorded.

This document also contains a very concise introduction to "clustering" and its different formulations as optimization problems in Section[1] 5, with an emphasize on *multi-manifold clustering*. This section goes through some well-known challenges in this field of study and provides some appropriate references to follow.

Problem proposed for the *Optimizer Competition* 2022 on February 19, 2022. Please refer to this problem as:
[SOAL, *Multi-manifold clustering*, Optimizer Competition (2022), Sharif Optimization and Applications Laboratory, Department of Mathematical Sciences, Sharif University of Technology, February 2022, optimizer.math.sharif.edu].
[1]Those who are already familiar with the details of the subject may skip Section 5.

1 !Optimizer Problem Setup

The general setup of the problem for Optimizer2022 competition is to solve the following inverse problem:

- Optimizer2022 **problem (general setup):**

  Given: A dataset that is provided in a geometric setting as a set of vectors (indicating points) in a real vectorspace $\mathbb{R}^d$ (for the format of input files of datasets see Section 2). Note that in each round of the competition you may also receive some additional information about the given dataset, e.g. the number of clusters, some extra information on the shape and dimension of submanifolds of data or some extra information about the existence of outliers or noise within the dataset.

  Query: In each round you are asked to provide a *crisp/hard clustering* of the given dataset along with a partitioning of the dataset in clusters containing the subclusters residing on each submanifold of data. In some rounds you may be asked to provide some extra information about the shape or the dimension of these submanifolds too (if they are not provided to all teams beforehand at the beginning of that round of the competition).

  Output: In each round you are given 4 datasets along with some extra information about the dataset or its parameters (for a list of parameters involved see the next item below) and you are asked to provide a clustering of the dataset that also provide a clustering of the dataset residing on each submanifold, along with the unknown parameters (e.g. the total number of clusters, the number of clusters on each submanifold, etc.). All 4 datasets are provided to all participating teams at the beginning of each round while multi-submissions are also permitted subject to the rules of the competition. The scoring procedure is based on a detailed comparison of the answers with the *ground-truth clustering* which is available to the judge for each dataset and results will be announced on competition's leaderboard while the competition is taking place. For the format of the input/output files and the details of the scoring procedure please see Section 2.

  Parameters: A summary of parameters involved in Optimizer2022 problem is provided in the following list.

  - **The ambient space:** $S = \mathbb{R}^d$
    This is the ambient space of the datapoints.

– **Dataset order and size:** n, N
In what follows n stands for the **order** of the dataset, i.e. the number of datapoints in S, and $N = nd$ is referred to as the **size** of the dataset.

– **Maximum size classification of datasets:**
We use the following names according to the *maximum* size of the datasets. Note that these classes form a hierarchy as

$$\text{Small} \subset \text{Medium} \subset \text{Large} \subset \text{Huge}.$$

  * **Small dataset:** $N \leq x.10^4$.
  * **Medium dataset:** $N \leq x.10^6$.
  * **Large dataset:** $N \leq x.10^8$.
  * **Huge dataset:** $N \leq x.10^{10}$.

– **Total number of clusters:** k
This is the whole number of clusters in the ground truth clustering of the dataset.

– **Submanifolds and their dimensions:** M_i's and d_i's for $1 \leq i \leq m$
The integer m is the number of submanifolds onto which the given dataset reside. In what follows M_i is a submanifold of $\mathbb{R}^d$ of dimension d_i. (For the exception in the case of "Spheres" see Section 2.)

– **Number of clusters on M_i:** k_i's for $1 \leq i \leq m$
This is the the whole number of clusters on M_i existing in the ground truth clustering of the dataset (or you may interpret this number as the number of parts into which you are supposed to partition the dataset existing on M_i). Note that

$$k = \sum_i k_i.$$

– **Number of outliers:** ρ
This is the number of outliers. Recall that an *outlier* is a datapoint which is not in any one of the *ground truth clustering*. A dataset is said to be *clean* if $\rho = 0$.

– **Noisy dataset:**
A *noisy dataset* is a dataset whose *ground truth clustering* do not **exactly** fit to low-dimensional manifolds, however, such a low-dimensional manifold clustering exists by considering small perturbations of datapoints (which is assumed to be the displacement caused by the effect of noise on the coordinates). If necessary, some extra information about the nature of the noise existing in the dataset will be provided to all participants in due time.

2 File Formats, Submissions and Scoring

The !Optimizer competition for 2022 consists of four different rounds, each concentrating on a prefixed setup for multi-manifold clustering in four possibly different scenarios and datasets. Descriptions and details of the questions for each round as well as what is expected to be submitted by participating teams are explained in Section 3. For each one of the rounds, all four datasets will be made available to all teams at the beginning of that specific round of the competition and each team may have as many submissions as they wish for each dataset, until the time for that specific competition round is up (the timetable of competition rounds will be announced before the competition starts in July 2022. For input/output file formats see Section 2.1). During each competition round, only the highest score for each dataset and for each one of the participating teams will be displayed according to their submissions on the competition leader-board that will be updated based on latest submissions in real-time, visible to all participants (for competition's scoring policy see Section 2.3).

2.1 File formats

All required input files will be made available to all participating teams in compressed text formats via the official website according to the announced timeline (in addition to the compressed text format, when necessary, input files may also be made available to participants in some other formats that may facilitate reading data). All submissions (i.e. output files) must be uploaded, in a text format, to the announced portal for the online judge system (details of the timeline for data releases and submission deadlines will be announced before the competition starts in July 2022. For input and output file formats and some samples see below).

The first line of the input file contains integers d, n, m, k, and ρ, respectively, separated by one space, if each one of these parameters are specified as part of the given data for that specific dataset (otherwise there is a symbol "_" in the file, indicating that the corresponding parameter must be evaluated and submitted in the output file as part of the teams response to that specific dataset, according to the description of the round in Section 3). The next line of the input file contains m integers, indicating k_i's (if m is not specified in the input, this line is left empty.). After that, each one of the next n lines contains d real numbers specifying coordinates of the corresponding input vector.

Each submission of teams, as an output file, should follow the strict specifications explained below as the output file format (note that any format error by the judge will result in exclusion of the corresponding submission! For samples see Section 2.2).

The output file starts with a line containing the integers n and m, respectively, separated by one space. After this there is a list of m records, each containing the information of the ith manifold for $1 \leq i \leq m$, described as follows.

- The first line of the record for the ith manifold contains parameters d_i, k_i, and t_i, respectively. The parameters d_i, k_i are integers defined in Section 3. The parameter t_i is a word representing the type of the manifold which is either "Shpere" for spherical manifolds, "Affine" for affine subspaces, or "Complex" for more complex manifolds.

- The next part of the record is the geometric specification of the ith manifold that depends on the type of the manifold, specified by t_i. Here are the geometric specifications for different values of t_i:

  - t_i = Affine:
    The specification of an affine d_i-dimensional affine subspace of the ambient space must be provided by $d - d_i + 1$ lines containing the necessary information characterizing $d - d_i$ equations $\langle a_j, x\rangle = b_j$ with $x \in \mathbb{R}^d$ and $1 \leq j \leq d - d_i$ for an **orthonormal** set of vectors $\{a_j \in \mathbb{R}^d \mid 1 \leq j \leq d - d_i\}$, in such a way that, the jth line contains the coordinates of the vector a_j, and proceeding these $d - d_i$ lines, there must be a new line containing $d - d_i$ real numbers b_j, separated by one space. For the case $d = d_i$ just leave a blank line.

  - t_i = Sphere:
    The specifications of a spherical manifold residing in an affine d_i-dimensional subspace, starts with specifications of the affine subspace in $d - d_i + 1$ lines (if $d = d_i$ then leave a blank line), followed by a line containing the coordinates of the center as a point in $\mathbb{R}^d$, separated by one space, as well as and the value of the radius of the sphere (all in one line).

    An exceptional convention: Note that in this competition, for the case of spheres, and to help you to produce reports more easily, you are supposed to report the dimension of an r-dimensional sphere with $r > 0$ as $r + 1$ which is actually the dimension of the corresponding affine ambient space, or equivalently, the dimension of the corresponding ball. Within this setup, and as exceptions, the dimension of a point as a pathological zero dimensional sphere is assumed to be equal to zero and also the dimension of a pair of points, as a zero dimensional sphere, is assumed to be equal to one, referring to the dimension of the corresponding line segment (e.g. see examples in Section 2.2.).

  - t_i = Complex:
    In this case there is no specification for the manifold and the record contains no line for this case.

- Information of k_i clusters follows the specifications of the ith manifold with the following format. Information of the jth cluster (for $1 \leq j \leq k_i$) appears in a separate line, starting with the number of vectors (in the jth cluster) followed by the *index* of input vectors assigned to this cluster. Note that the *index* of an input vector is its order of appearance in the

input file, starting from 1 (i.e. the index of the first vector in the input file).

- The last record is dedicated to the information of outliers. This record starts with an integer indicating the number of outliers ρ, followed by the indices of the input vectors that are determined as outliers, all separated by one space. If you do not find any outlier (i.e. $\rho = 0$), this record just contains a single "0". (Note that each index of datapoints must appear exactly once in the union of your clustering and your outlier set.)

Please refer to Section 2.2 for some examples of input and output files.

2.2 Input/Output samples

Note that sample input and outputs presented here are only provided to show the formatting of the input and outputs. So following sample inputs may not be compatible with requirements of the rounds of the competition and also following sample outputs may represent not good solutions for the problem specified in the corresponding sample input.

Input #1:	Description of input
`2 5 2 _ 0`	$d = 2$, $n = 5$, $m = 2$, k is not specified, and $\rho = 0$.
`_ _`	Values k_i are not specified.
`0 0`	Vector #1:
`0 1`	Vector #2:
`10 10`	Vector #3:
`1 0`	Vector #4:
`1 1`	Vector #5:

Output #1:	Description of output
`5 2`	$n = 5$ and $m = 2$.
`2 1 Affine`	An affine manifold in $\mathbb{R}^2$ having $k_1 = 1$ cluster.
`4 1 2 4 5`	This cluster contains 4 vectors: 1, 2, 4, and 5.
`0 1 Sphere`	A sphere manifold in $\mathbb{R}^0$ having $k_2 = 1$ cluster.
`1 1`	$a_{2,1} = (1, 1)$
`1 -1`	$a_{2,2} = (1, -1)$
`20 0`	$b_{2,1} = 20$ and $b_{2,2} = 0$.
`10 10 0`	Center of sphere is $(10, 10)$, and its radius is 0.
`1 3`	This cluster contains 1 vector: 3.
`0`	There is no outlier ($\rho = 0$).

Input #2:	Description:
2 6 _ _ 1 _ _ 0 0 0 1 10 10 1 0 10 11 11 10	$d = 2$, $n = 6$, m and k are not specified, and $\rho = 1$. Values k_i are not specified. Vector #1: Vector #2: Vector #3: Vector #4: Vector #5: Vector #6:

Output #2:	Description:
6 2 2 3 Complex 1 1 1 2 1 4 2 1 Complex 2 3 5 1 6	$n = 6$ and $m = 2$. A complex manifold in $\mathbb{R}^2$ having $k_1 = 3$ cluster. This cluster contains 1 vectors: 1. This cluster contains 1 vectors: 2. This cluster contains 1 vectors: 4. A complex manifold in $\mathbb{R}^2$ having $k_1 = 1$ cluster. This cluster contains 2 vectors: 3, and 5. There is one outlier ($\rho = 1$): vector 6.

Input #3:	Description of input
3 4 1 1 0 1 1 0 0 2 -1 0 2 1 0 3 0 0	$d = 3$, $n = 4$, $m = 1$, $k = 1$ and $\rho = 0$. $k_1 = 1$. Vector #1: Vector #2: Vector #3: Vector #4:

Output #3:	Description of output
4 1 2 1 Sphere 0 0 1 0 2 0 0 1 4 1 2 3 4 0	$n = 4$ and $m = 1$. A sphere manifold in $\mathbb{R}^2$ having $k_1 = 1$ cluster. $a_{1,1} = (0, 0, 1)$ $b_{1,1} = 0$ Center of sphere is $(2, 0, 0)$, and its radius is 1. This cluster contains 4 vector: 1, 2, 3, 4. There is no outlier ($\rho = 0$).

2.3 Scoring

The scoring procedure in each case contains the following three steps.

1) **Format verification:**

   The format of the submitted file is verified to match what has already been specified in Section 2. In case of any kind of format error, the total score for the submission will be set to **zero**.

2) **Input data verification:**

   Then, the submission is verified to contain and match the information provided in the input file as the input dataset and what is asked for within the specific round. **The total score of a submission that is not compatible with this information is set to be equal to zero**. In particular, any submission must be a hard/crisp clustering of the dataset, i.e. each data point must appear exactly in one of the parts of a clustering (including the outlier set if there is any).

3) **Scoring:**

   In the final stage, the score of a submission is evaluated according to the sum of scores related to *parameter compatibility* and *cluster compatibility* of the submission, compared to the *ground truth clustering* which is available to the judge. In this scoring procedure, which is described explicitly in what follows, the constants ε, σ's, ω_d, ω_I, ω_M, ω_C, ω_k, and ω_O are determined by the judge for each dataset.

   For two given numbers a and b, and two given subsets C and C', let us define

$$s(a,b,\sigma) \stackrel{\text{def}}{=} e^{-\sigma\left(\frac{a-b}{\max(\varepsilon,|a|)}\right)^2},$$

$$s^+(a,b,\sigma) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } b<0, \\ b/a & \text{if } b<a, \\ s(a,b,\sigma) & \text{o.w.} \end{cases}$$

   and

$$J(C,C') \stackrel{\text{def}}{=} \frac{|C\cap C'|}{|C\cup C'|},$$

   which is the Jaccard index for the clusters C and C'. Also, for a given manifold M of dimension d, specific parameters I (see below for the definition) and a ground-truth clustering

$$\mathcal{C} = (C_1, C_2, \ldots, C_k),$$

   the score of a submitted clustering

$$\mathcal{C}' = (C'_1, C'_2, \ldots, C'_{k'})$$

with the submitted dimension d' and specific parameters I', is defined as

$$s(M, M') \stackrel{\text{def}}{=} \omega_{I}\pi_{M,M'} + \omega_{d}s^{+}(d, d', \sigma_{d}) + \omega_{k}s^{+}(k, k', \sigma_{k}) + \omega_{C} \max_{\mathcal{P}} \sum_{(i,j)\in\mathcal{P}} \omega_{C_i} J(C_i, C'_j)$$

in which $\mathcal{P}$ is a maximum pairing (i.e. matching) in $\{1, \ldots, k\} \times \{1, \ldots, k'\}$ and the *parameter-score*, $\pi_{M,M'}$, is defined below.

Now, let

$$\mathcal{C} = (C_{1,1}, C_{1,2}, \ldots, C_{1,k_1}, C_{2,1}, C_{2,2}, \ldots, C_{2,k_2}, \ldots, C_{m,1}, \ldots, C_{m,k_m})$$

be the ground truth clustering in which

$$\mathcal{C}_i = (C_{i,1}, C_{i,2}, \ldots, C_{i,k_i})$$

is the induced clustering on M_i having k_i clusters. Then, the total score of a submitted clustering $\mathcal{C}'$ on a list of manifolds M'_i of size k'_i's for $1 \le i \le m'$ with dimensions d'_i's is defined to be

$$s(\mathcal{C}, \mathcal{C}') \stackrel{\text{def}}{=} \omega_{M} \max_{\mu} \sum_{(i,j)\in\mu} \omega_{M_i}\ s(M_i, M'_j) + \omega_{O}J(O, O'),$$

in which μ is a maximum pairing (i.e. matching) in $\{1, \ldots, m\} \times \{1, \ldots, m'\}$, and O and O' are the sets of points indicated as outliers in the ground truth dataset and in the submitted solution, respectively.

The parameter score $\pi_{M,M'}$ depends on the type t of the manifold M (from the ground truth) and type t' of the manifold M' (submitted to the judge) and is defined as follows:

- $t = t' =$ Affine:
  For an affine subspace determined by the equation $Ax = b$ (where rows of A are chosen to be orthonormal), we define $I \stackrel{\text{def}}{=} A^T[A \quad b]$, where $[A \quad b]$ is a row block matrix. Hence, again, for two given affine subspaces we similarly let

  $$\pi_{M,M'}(I, I') \stackrel{\text{def}}{=} \frac{1}{N} \sum_{(i,j)} s(I_{i,j}, I'_{i,j}, \sigma_{a}),$$

  where N is the number of entries of the matrix I.
- If $t = t' =$ Sphere:
  For a sphere residing on an affine subspace having the parameter matrix I, let $\tilde{I}$ be the $d+1$ dimensional vector whose first coordinate is the radius of the sphere and the rest of the coordinates are the coordinates of the center of the sphere in $S = \mathbb{R}^d$. Then, for two given spheres we define

  $$\pi_{M,M'}(I, \tilde{I}, I', \tilde{I}') \stackrel{\text{def}}{=} \left(\frac{1}{d+1} \sum_{i=1}^{d+1} s(\tilde{I}_i, \tilde{I}'_i, \sigma_{s}) + \frac{1}{N} \sum_{(i,j)} s(I_{i,j}, I'_{i,j}, \sigma_{a}) \right).$$

- $t = t' =$ Complex: $\pi_{M,M'} \stackrel{\text{def}}{=} 1$.
- Otherwise: $\pi_{M,M'} \stackrel{\text{def}}{=} 0$.

3 Competition Rounds

The OPTIMIZER2022 competition consists of 4 rounds each containing 4 datasets to be processed. In what follows, we go through a detailed description of each round of the competition.

1) Warm-up round:

Euclidean low-dimensional clustering

- In this round each dataset is a set of points in $S = \mathbb{R}^d$ with $d \leq 4$ and $k_i = 1$ for $1 \leq i \leq m$. Each team is supposed to provide an Euclidean clustering for each one of the datasets as a set of balls, each with **optimal** radius and dimension d_i, **containing** the clusters.

  In this round, all low-dimensional balls must be represented by the intersection of a four dimensional ball in $\mathbb{R}^d$ and a couple of normal vectors representing the intersecting hyperplanes.

1. **Dataset R11:**
   - **Description:**
     You are given a small Euclidean clean dataset in $S = \mathbb{R}^3$ along with the number of clusters k, for which $k_i = 1$ for all $1 \leq i \leq m$.
   - **Task:**
     You are supposed to provide a clustering $\mathcal{C}$, along with a set of balls in $\mathbb{R}^3$ (determined by center and radius) that cover the clusters.
2. **Dataset R12:**
   - **Description:**
     You are given a small Euclidean clean dataset in $S = \mathbb{R}^3$ along with the number of clusters k, for which $k_i = 1$ for all $1 \leq i \leq m$. Clusters reside on lower dimensional balls whose dimensions are all the same and equal to $d' \leq 3$.
   - **Task:**
     You are supposed to provide a clustering $\mathcal{C}$, the number d', along with a set of k low-dimensional balls of dimension d' (determined by their center, radius and normal vectors representing the intersecting hyperplanes), that cover the clusters.
3. **Dataset R13:**

- **Description:**
  You are given a medium Euclidean dataset in $S = \mathbb{R}^4$ containing a small set of outliers, along with the number of clusters k, for which $k_i = 1$ for all $1 \leq i \leq m$. Clusters reside on lower dimensional (possibly intersecting) balls whose dimensions d'_i may vary, i.e. $1 \leq d'_i \leq 4$.
- **Task:**
  You are supposed to provide a clustering $\mathcal{C}$, a set Ω of outliers, the numbers d'_i for $1 \leq i \leq k$, along with a set of k low-dimensional balls, each of dimension d'_i (determined by center, radius and normal vectors representing the intersecting hyperplanes), that cover the clusters.

4. **Dataset R14:**

- **Description:**
  You are given a large Euclidean noisy dataset in $S = \mathbb{R}^4$ along with the number of clusters k, for which $k_i = 1$ for all $1 \leq i \leq m$. Clusters reside on lower dimensional (possibly intersecting) balls whose dimensions d'_i may vary, i.e. $0 \leq d'_i \leq 4$.
- **Task:**
  You are supposed to provide a clustering $\mathcal{C}$, the numbers d'_i for $1 \leq i \leq k$, along with a set of k low-dimensional balls, each of dimension d'_i (determined by center, radius and normal vectors representing the intersecting hyperplanes), that cover the clusters.

2) Novice optimizer round:

low-dimensional Euclidean multi-clustering on spheres

- **General description:** In this round each dataset is a set of points in $S = \mathbb{R}^d$ for some $d < 500$, where datapoints are located **on** a set of balls (i.e. submanifolds are spheres) while there may be more than one cluster on each sphere.
- **General task:** In general, teams are supposed to find the number of balls m and their specific parameters (i.e. their center and radius), the dimension of each one of the balls d_i for $1 \leq i \leq m$, the number of clusters on each ball, k_i, along with the corresponding clustering. Data may contain outliers or noise as described in the specifications of each dataset. The output must contain all information already specified in output file format in Section 2.1.

1. **Dataset R21:**

- **Description:**
  You are given a small dataset in S, along with the total number of clusters k and the number of spheres m.

2. **Dataset R22:**

   - **Description:**
     You are given a small dataset in S, along with the total number of clusters k.

3. **Dataset R23:**

   - **Description:**
     You are given a medium dataset in S, containing a small set of outliers, as well as the total number of clusters, k.

4. **Dataset R24:**

   - **Description:**
     You are given a large noisy dataset in S.

3) Expert optimizer round:

Multi-manifold geometric multi-clustering with extra information

- **General description:** In this round all datasets are large or smaller as a set of points in $S = \mathbb{R}^d$ for some $d < 1000$, where datapoints are located **on** some (possibly intersecting) submanifolds of dimension **one** or **two**. It is probable that the number of clusters on each submanifold be greater than one (i.e. $k_i \geq 1$).

- **General task:** In general, teams are supposed to find the number of submanifolds and their dimensions, i.e. d_i for $1 \leq i \leq m$, the number of clusters on each submanifold, k_i, along with the corresponding clustering. Data may contain outliers or noise as described in the specifications of each dataset. The output must contain all information already specified in output file format in Section 2.1.

1. **Dataset R31:**

   - **Description:**
     You are given a medium dataset in S, along with the total number of submanifolds, m, and the total number of clusters, k.

2. **Dataset R32:**

   - **Description:**
     You are given a medium dataset in S, along with the total number of submanifolds, m.

3. **Dataset R33:**

   - **Description:**
     You are given a large dataset in S, containing a small set of outliers.

4. **Dataset R34:**

   - **Description:**
     You are given a large noisy dataset in S.

4) The !Optimizer round:

High-dimensional complex multi-manifold multi-clustering

- **General description:** In this round you are given four datasets, **R41, R42, R43** and **R44** which are all noisy, and large or huge in size, as a set of points in $S = \mathbb{R}^d$ for some $d < 10000$, where datapoints are located **on** some (possibly intersecting) submanifolds of lower dimension. It is probable that the number of clusters on each submanifold be greater than one (i.e. $k_i \geq 1$).

- **General task:** In general, teams are supposed to find the number of submanifolds and their dimensions, i.e. d_i for $1 \leq i \leq m$, the number of clusters on each submanifold, k_i, along with the corresponding clustering. The output must contain all information already specified in output file format in Section 2.1.

4 Supporting participating teams

During the preparation phase of the competition, SOAL will do its best to organize lectures and workshops related to *clustering*. Also, SOAL will provide all teams with a *precompetition*, called Optimizer2022 playground, as a warmup with small size toy-examples of data that may help all participating teams to get familiar with the competition platform on Quera as well as testing their programs, in particular input/output routines. The Optimizer2022 playground will eventually be released with all the original data used in Optimizer2022, after the competition is over[2].

[2]For supports of Optimizer2021 you may visit the archives of http://optimizer.math.sharif.edu/. Also for the Optimizer2021 playground go to https://quera.org/accounts/login?next=/contest/add_to_contest/0EKk3VjVD16XfrC/.

5 Motivation and Background

Clustering is among the most fundamental problems in data science and artificial intelligence, generally described as efficient ways of looking for partitions (i.e. groupings) of a dataset into clusters (i.e. parts), each containing objects with maximum similarity (within the cluster), while the dissimilarity between the clusters are also maximized at the same time. This very basic problem may be classified within the category of unsupervised learning procedures, while there already exists a vast literature on this fundamental subject and its application (e.g. see [3, 5, 36, 43, 44, 46, 52, 55]). Let us review the most challenging aspects of the clustering problem in what follows.

Although usually a clustering problem is formulated as an optimization problem (see what follows), one of the main difficulties in dealing with this problem is due to lack of universal definitions for the main concepts involved, such as the definitions of "a cluster", "similarity" or "outliers". This controversy is rooted in different intrinsic aspects of the problem itself and most probably can not be removed. For instance, the data itself may appear in many different forms or presentations such as points in $\mathbb{R}^d$, or just as a set of conceptual data points along with a similarity measure between them that may be presented as a similarity graph. On the other hand, the whole problem and its solution are strongly influenced by *scaling*, where this property is an actual obstacle against deriving a universal and well-behaved formulation with an acceptable performance in different kinds of applications. *Hierarchical clustering* setups are proposed to circumvent this difficulty through providing a refinement of the clustering structure in different levels of scaling. These challenges that naturally are faced in relation to the most basic aspects of the problem, and are deeply related to its definition, have given rise to a variety of formulations which may vary in relation to "presentation of data", "similarity measure or metric" or the "cost function" to be optimized.

Even in a very simple Euclidean setup, in which one usually let clusters to be a collection of separated Euclidean balls of points (or vectors) in $\mathbb{R}^d$, it is quite obvious that the clustering problem becomes more and more challenging when the balls tend to become closer to each other or even intersect. Hence, in general, a measure of *well-separatedness* is strongly related to the difficulty as well as the performance of the outcome of a typical clustering algorithm. This may become even more challenging when one notes that clusters of a real problem are usually not ball-shaped, while in real applications they are actually low-dimensional *manifolds* residing in the original ambient space of the dataset.

This scenario suggests that a *multi-manifold clustering* setup, as a problem in which one is supposed to rediscover the actual clusters of points on each low-dimensional manifold of data, when the original cumulative data is given as a set of points in $\mathbb{R}^d$, is among the most general setups of a clustering problem (e.g. see [2, 6, 13, 22, 31, 32, 38, 47, 50, 51, 56, 57, 58]). In this regard, the most important sources of difficulty for such an inverse problem are as follows:

- Sensitivity to scaling.

- Shape-complexity of the submanifolds of clusters and the clusters themselves (in particular the effect of curvature on similarity measures!).
- Well-separatedness of the clusters (in particular clustering problem on intersections of submanifolds).
- Not being aware of the number of clusters.
- Not being aware of the dimension of clusters (i.e. the submanifolds).
- Hardness of processing large datasets (even quadratic-time algorithms are inefficient for large datasets!).
- High-Dimensional datasets (curse of dimensionality!).
- Existence of outliers and their distribution.
- Existence of noise and its nature.

Regardless of the algorithmic/practical importance of the clustering problem which is mainly related to its applications in computer science and AI, it is quite interesting and also fundamental to consider the theoretical importance of the problem, as a discrete version of the *isoperimetry problem*, which is among the most fundamental problems in *geometry*, with strong ties to the geometry of metric measure spaces as well as partial differential equations. Not only this bidirectional interconnection has already been fruitful for both sides of this fascinating area of research, but also it has given rise to very interesting and fundamental new questions, with profound impacts on both theoretical and practical sides of the subject. Among these very interesting open questions, it is quite instructive to mention the *discrete curvature problem*, as one of these very basic questions, seeking for a nice and well-defined definition for the concept of "curvature" in the discrete setting, with deep applications in clustering, learning theory and manifold optimization as well as important theoretical consequences in theory of discrete metric measure spaces and discrete dynamics.

Clustering algorithms may also be divided into *crisp* (i.e. *hard*) or *soft* versions in general. Note that, in the crisp/hard case, each datapoint is supposed to be assigned to at most one of the clusters, while in soft methods, fuzzy assignment of a datapoint to more than one cluster with non-crisp degrees of membership is acceptable.

Important assumption!
In this competition, all clustering problems are of crisp type and the desired outputs are assumed to be of hard type as a subpartition of datapoints!

In the sequel, we provide a very concise list of the most basic methods of tackling the clustering problem just to provide an overview to the subject and some general references to be followed for the details.

- **K-means method and its variants**
  K-means is among the oldest clustering center-based methods, when the

problem is given in an Euclidean setting with datapoints in $\mathbb{R}^d$. The method tries to find a couple of optimal balls covering clusters of datapoints, by considering the following optimization problem

$$kmeans \stackrel{\text{def}}{=} \underset{\mathcal{D}(B(x_1,r_1),\ldots,B(x_k,r_k))}{\arg\min} \sum_{i=1}^{k} \sum_{p\in B(x_i,r_i)} \|p - x_i\|^2.$$

in which, $B(x,r) \subset \mathbb{R}^d$ is the ball of radius r centered at $x \in \mathbb{R}^d$, and $\mathcal{D}(B(x_1,r_1),\ldots,B(x_k,r_k))$ is a collection of k such disjoint balls.

There also exist other center based methods as K-median and K-center variants of this approach. On the other hand, a variety of algorithms have been already proposed to deal with this optimization problem, in particular, some variants that also handle outliers using *robust* versions of the cost function, applying regularizers (e.g. see [4, 10, 11, 14, 20, 21, 23, 24, 26, 34, 59]).

- **Spectral clustering**
  Spectral clustering is one of the most fundamental bridges between the practical and theoretical aspects of the clustering problem and is widely used for medium-size datasets. In this setting the whole data is encoded in a *similarity* matrix (resp. graph) whose rows/columns (resp. vertices) are indexed by the datapoints and each entry of the matrix (resp. edge-weight) is assumed to represent the measure of similarity between the two data entries. The spectral clustering method is based on the fundamental fact that there is a close relationship between the eigenstructure of the corresponding Laplacian matrix and the concentration of datapoints (see the comments on isoperimetry below). The main part of the procedure is based on the observation (proved in [15, 27, 35, 40, 49, 54]) that the normalized row vectors corresponding to the first k eigenvectors of the Laplacian provide a transformation of the dataset of size n into the unit ball of $\mathbb{R}^k$ which are almost in an Euclidean configuration. This fascinating phenomenon naturally provides a very straight forward *quadratic-time* transformation method to reconfigure any dataset into an Euclidean setting, having a smaller dimension, and then apply very simple clustering procedures as variants of the K-means method. Unfortunately, the method is too time-consuming for large datasets and can not be applied as the main clustering procedure in these cases.

- **Graph-based isoperimetric clustering**
  Isoperimetry is one of the most fundamental concepts in geometry that generalizes all preceding methods in a unified setting.

  Within the discrete setting that models a dataset, let us consider a *weighted graph* $G = (V, E)$ with the weight functions $c : E \to \mathbb{R}_+$ and $\pi : V \to \mathbb{R}_+$ on its sets of edges and vertices, respectively. Also, define $\mathcal{D}_{k,\rho}(V)$ to be the set of all k-tuples $\mathcal{A} = (A_1, A_2 \ldots, A_k)$ of nonempty disjoint subsets

$A_i \subseteq V$, called *k-subpartitions*, such that

$$\sum_{1\leq i\leq k} |A_i| \geq |V| - \rho.$$

The *classical* isoperimetry problem is a generalization of the *minimum cut* problem which is defined as follows. Given any subset $A \subseteq V$, the *normalized cut* value corresponding to A is defined as

$$\varphi(A) \stackrel{\text{def}}{=} \frac{c(\delta(A))}{\pi(A)},$$

in which $\delta(A) \stackrel{\text{def}}{=} \{e = uv \in E \mid u \in A \ \& \ v \notin A\}$ is the set of boundary edges of A and the weight of a set is the sum of the weights of its members. Then, in this setting, one may define the vector

$$\Phi_{G,k}(\mathcal{A}) \stackrel{\text{def}}{=} [\varphi(A_1), \varphi(A_2), \ldots, \varphi(A_k)] \in \mathbb{R}^k,$$

along with the cost function,

$$\Psi_{G,p}(\mathcal{A}) \stackrel{\text{def}}{=} \|\Phi_{G,k}(\mathcal{A})\|_p,$$

whose minimization over $\mathcal{D}_{k,\rho}(V)$ may be interpreted as a model for the robust clustering problem (with outliers) in which G is the similarity graph of the dataset.

A fascinating feature of this formulation is the fact that the whole setting may be generalized to a real-relaxed optimization problem in which *normalized cuts* are interpreted as *normalized energies* of characteristic functions of subsets of V. To see this, note that one may interpret the normalized cut value $\varphi(A)$ as

$$\frac{c(\delta(A))}{w(A)} = \frac{\|\nabla\chi_A\|_1}{\|\chi_A\|_1} = \frac{\|\nabla\chi_A\|_2^2}{\|\chi_A\|_2^2} = \frac{\langle\Delta(\chi_A), \chi_A\rangle}{\langle\chi_A, \chi_A\rangle},$$

in which χ_A stands for the characteristic function of the set A, the symbol ∇ refers to the gradient operator and $\Delta \stackrel{\text{def}}{=} \nabla^*\nabla$ is the natural Laplacian operator acting on the set of real functions on V (e.g. see [16, 18] for the definitions). This generalized setting may be formalized as follows and has many interesting consequences.

Again, define the generalized normalized cut as

$$\varphi_{\alpha,\beta}(f) \stackrel{\text{def}}{=} \frac{\|\nabla f\|_\alpha}{\|f\|_\beta},$$

for norms $\|.\|_\alpha$ and $\|.\|_\beta$, that may be interpreted as a *normalized energy* of a function f. Then one may similarly define the cost function

$$\tilde{\Psi}_{G,p}(\mathcal{F}) \stackrel{\text{def}}{=} \|\tilde{\Phi}_{G,k}(\mathcal{F})\|_p,$$

with

$$\tilde{\Phi}_{G,k}(\mathcal{F}) \stackrel{\text{def}}{=} [\varphi_{\alpha,\beta}(f_1), \varphi_{\alpha,\beta}(f_2), \ldots, \varphi_{\alpha,\beta}(f_k)] \in \mathbb{R}^k,$$

where

$$\mathcal{F} \stackrel{\text{def}}{=} \{f_1, f_2, \ldots, f_k\}.$$

A minimization of $\tilde{\Psi}_{G,p}(\mathcal{F})$ when $\mathcal{F}$ varies within the collection of k-tuples of nonnegative functions with mutually disjoint supports, is known as the (generalized) isoperimetry problem. It is known that when $\|.\|_\alpha$ and $\|.\|_\beta$ are equal to the L_1 norm $\|.\|_1$ and $p \in \{1, \infty\}$, the solution of the isoperimetry problem is equivalent to the classical case when the minimization is over the k-subpartitions. Also, it is quite interesting that the cost function of this classical case is essentially equal to a shift of the K-means cost function for a suitably defined kernel (e.g. see [16] and references therein). On the other hand, another relaxation of the generalized isoperimetry problem to arbitrary real (i.e. not necessarily nonnegative) functions and the $\|.\|_2$ norm, naturally gives rise to an eigenvalue problem for the Laplacian kernel (e.g. compare to Courant-Fischer-Weyl min-max principle and Cheeger-type inequalities), showing that the generalized isoperimetry problem actually covers both K-means and spectral approaches to the clustering problem (e.g. see [16, 33] for more on this).

Isoperimetry, in general, is an NP-hard problem and very hard to solve, however, a very surprising phenomenon is the fact that the *classical* max-isoperimetry (i.e. $p = \infty$) problem is solvable on trees in $O(n \log n)$ time, which provides a very efficient approach to clustering of large datasets using in conjunction to the fact that the minimum spanning tree of the corresponding metric graph (i.e. the dissimilarity graph) captures the main geometric properties of the dataset (in particular its clustering structure) [1, 17, 19, 28].

- **Density-based clustering**
  Density-based clustering is an approach to manage non-Euclidean datasets using the idea of considering the density of datapoints in an r-neighbourhood of a point. Intuitively, the larger the density, the more probable is the case that the point is within a cluster along with its neighbourhood. Hence, in density-based clustering one usually fixes a density threshold, say δ, along with a radius r, and then one computes the density of points in the r-neighbourhood of each datapoint. After that there are many different strategies to use this r-density profile of the data to obtain a clustering, in all of which, central-points that achieve a density larger than δ play a crucial role. A classical clustering algorithm of this type is DBSCAN introduced in 1998, while a variety of density-based algorithms have been introduced for many different applications ever since (e.g. see [7, 45]).

- **Model-based clustering**
  In model-based clustering, one usually fixes a probabilistic/statistical model as a parametrized class of distributions and then one tries to fit the model

(i.e. an optimal choice of parameters) to a given dataset by minimizing a cost function that intuitively behaves as a generalized distance. There are a variety of models and cost functions in the literature and one ought to note that this approach usually is tuned to produce soft clusters as mixtures of well-known distributions (e.g. the Gaussian distribution). A typical and popular scenario of this type is the variational Bayesian approach in which the cost function is the Kullback-Leibler divergence (i.e. the relative entropy) where the class of distributions (i.e. the model) may be chosen to be of exponential family type or even more complex graphical models. Recent popularity of these variational approaches is mainly due to the fact that one may use Bethe approximations of the Gibbs variational principle that usually will give rise to an expectation-maximization iterative setup which is quite fast and applicable to large datasets. For more on this approach see e.g. [8, 9, 25, 29, 39, 48, 60].

- **Miscellany**
  There are a variety of other approaches to clustering problem, among which *hierarchical-clustering, diffusion-based clustering* and *functional data clustering* are some of the most recent and important ones. For more on these topics one may refer to the general survey articles mentioned at the beginning of this section as well as [12, 14, 30, 37, 41, 42, 53].

References

[1] M. Alimi, A. Daneshgar, and M. Foroughmand-Araabi, *Mean isoperimetry with control on outliers: Exact and approximation algorithms*, https://arxiv.org/pdf/1807.05125.pdf.

[2] A. Babaeian, *Multiple manifold clustering using curvature constrained path*, (2018), https://arxiv.org/abs/1812.02327.

[3] A. M. Bagirov, N. Karmitsa, and S. Taheri, *Partitional clustering via nonsmooth optimization: Clustering via optimization*, Unsupervised and Semi-Supervised Learning, Springer International Publishing, 2020.

[4] S. Bandyapadhyay and K. Varadarajan, *On Variants of k-means Clustering*, 32nd International Symposium on Computational Geometry (SoCG 2016) (Dagstuhl, Germany) (Sándor Fekete and Anna Lubiw, eds.), Leibniz International Proceedings in Informatics (LIPIcs), vol. 51, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, https://arxiv.org/pdf/1512.02985.pdf, pp. 14:1–14:15.

[5] A. C. Benabdellah, A. Benghabrit, and I. Bouhaddou, *A survey of clustering algorithms for an industrial context*, Procedia Computer Science **148** (2019), 291–302, The second international conference on intelligent computing in data sciences, ICDS2018.

[6] F. Besold and V. Spokoiny, *Adaptive manifold clustering*, Weierstraß-Institut für Angewandte Analysis und Stochastik Leibniz-Institut im Forschungsverbund Berlin, Preprint (2020), no. 2800, http://www.wias-berlin.de/preprint/2800/wias_preprints_2800.pdf.

[7] P. Bhattacharjee and P. Mitra, *A survey of density based clustering algorithms*, Frontiers of Computer Science **15** (2021), 1–27.

[8] C. Biernacki and C. Maugis, *High-dimensional clustering*, Choix de modèles et agrégation, Sous la direction de J-J. DROESBEKE, G. SAPORTA, C. THOMAS-AGNAN Edition: Technip., September 2017.

[9] C. Bouveyron, G. Celeux, T. B. Murphy, and A. E. Raftery, *Model-based clustering and classification for data science: With applications in r*, Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, 2019.

[10] S. Brodinová, P. Filzmoser, T. Ortner, C. Breiteneder, and M. Rohm, *Robust and sparse k-means clustering for high-dimensional data*, Adv. Data Anal. Classif. **13** (2019), no. 4, 905–932.

[11] C. Brunet-Saumard, E. Genetay, and A. Saumard, *K-bmom: A robust lloyd-type clustering algorithm based on bootstrap median-of-means*, Comput. Stat. Data Anal. **167** (2022), 107370, https://arxiv.org/pdf/2002.03899.pdf.

[12] F. Centofanti, A. Lepore, and B. Palumbo, *Sparse and smooth functional data clustering*, https://arxiv.org/pdf/2103.15224.pdf.

[13] D. Chen, J. Lv, and Y. Zhang, *Unsupervised multi-manifold clustering by learning deep representation*, The Workshops of the The Thirty-First AAAI Conference on Artificial Intelligence, Saturday, February 4-9, 2017, San Francisco, California, USA, AAAI Workshops, vol. WS-17, AAAI Press, 2017.

[14] X. Chen and Y. Yang, *Diffusion k-means clustering on manifolds: provable exact recovery via semidefinite relaxations*, https://arxiv.org/pdf/1903.04416.pdf.

[15] R. R. Coifman and S. Lafon, *Diffusion maps*, Applied and Computational Harmonic Analysis **21** (2006), no. 1, 5–30, Special Issue: Diffusion Maps and Wavelets.

[16] A. Daneshgar, H. Hajiabolhassan, and R. Javadi, *On the isoperimetric spectrum of graphs and its approximations*, Journal of Combinatorial Theory Ser. B **100** (2010), no. 4, 390–412.

[17] A. Daneshgar and R. Javadi, *On the complexity of isoperimetric problems on trees*, Discrete Applied Mathematics **160** (2012), no. 1-2, 116–131.

[18] A. Daneshgar, R. Javadi, and L. Miclo, *On nodal domains and higher-order cheeger inequalities of finite reversible markov processes*, Stochastic Processes and their Applications **122** (2012), no. 4, 1748–1776.

[19] A. Daneshgar, R. Javadi, and S. B. Shariatrazavi, *Clustering and outlier detection using isoperimetric number of trees*, Pattern Recognition **46** (2013), no. 12, 3371–3382.

[20] A. Deshpande, P. Kacham, and R. Pratap, *Robust k-means++*, Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), vol. 124, 2020.

[21] O. Dorabiala, J. N. Kutz, and A. Y. Aravkin, *Robust trimmed k-means*, https://arxiv.org/pdf/2108.07186.pdf.

[22] E. Elhamifar and R. Vidal, *Sparse manifold clustering and embedding*, Proceedings of the 24th International Conference on Neural Information Processing Systems (Red Hook, NY, USA), NIPS'11, Curran Associates Inc., 2011, p. 55–63.

[23] Z. Friggstad, K. Khodamoradi, M. Rezapour, and M.R. Salavatipour, *Approximation schemes for clustering with outliers*, ACM Trans. Algorithms **15** (2019), no. 2.

[24] Z. Friggstad, M. Rezapour, and M. R. Salavatipour, *Local search yields a ptas for k-means in doubling metrics*, SIAM Journal on Computing **48** (2019), no. 2, 452–480.

[25] S. Frühwirth-Schnatter, G. Celeux, and C. P. Robert, *Handbook of mixture analysis*, Chapman & Hall/CRC Handbooks of Modern Statistical Methods, Taylor & Francis Group, 2020.

[26] A. Georgogiannis, *Robust k-means: a theoretical revisit*, Advances in Neural Information Processing Systems (D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, eds.), vol. 29, Curran Associates, Inc., 2016.

[27] B. Ghojogh, A. Ghodsi, F. Karray, and M. Crowley, *Laplacian-based dimensionality reduction including spectral clustering, Laplacian eigenmap, locality preserving projection, graph embedding, and diffusion map: Tutorial and survey*, https://arxiv.org/pdf/2106.02154.pdf.

[28] L. Grady and E. L. Schwartz, *Isoperimetric graph partitioning for image segmentation*, IEEE Trans. Pattern Anal. Mach. Intell. **28** (2006), 469–475.

[29] F. Hirschberger, D. Forster, and J. Lucke, *A variational em acceleration for efficient clustering at very large scales*, IEEE Transactions on Pattern Analysis and Machine Intelligence (2021), 1–1.

[30] J. Jacques and C. Preda, *Functional data clustering: a survey*, Advances in Data Analysis and Classification **8** (2014), 231–255.

[31] A. Khan and P. Maji, *Multi-manifold optimization for multi-view subspace clustering*, IEEE Transactions on Neural Networks and Learning Systems (2021), 1–13.

[32] O. Koyejo and J. Ghosh, *Mipps: A generative model for multi-manifold clustering*, Manifold Learning and Its Applications, Papers from the 2009 AAAI Fall Symposium, Arlington, Virginia, USA, November 5-7, 2009, AAAI Technical Report, vol. FS-09-04, AAAI, 2009.

[33] J. R. Lee, S. Oveis Gharan, and L. Trevisan, *Multiway spectral partitioning and higher-order Cheeger inequalities*, Journal of the ACM **61** (2014), no. 6, 37:1–37:30.

[34] Y. Li, Y. Zhang, Q. Tang, W. Huang, Y. Jiang, and S. Xia, *t-k-means: A robust and stable k-means variant*, ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 3120–3124.

[35] U. V. Luxburg, *A tutorial on spectral clustering*, Statistics and computing **17** (2007), no. 4, 395–416.

[36] U. V. Luxburg, R. C. Williamson, and I. Guyon, *Clustering: Science or art?*, Unsupervised and Transfer Learning - Workshop held at ICML 2011, Bellevue, Washington, USA, July 2, 2011 (I. Guyon, G. Dror, V. Lemaire, G. W. Taylor, and D. L. Silver, eds.), JMLR Proceedings, vol. 27, JMLR.org, 2012, pp. 65–80.

[37] B. A. Manghiuc and H. Sun, *Hierarchical clustering: $o(1)$-approximation for well-clustered graphs*, 35th Conference on Neural Information Processing Systems (NeurIPS 2021), 2021.

[38] J. M. Martínez-Martínez, P. Escandell-Montero, J. D. Martín-Guerrero, J. Vila-Francés, and E. Soria-Olivas, *Manisons: A new visualization tool for manifold clustering*, 21st European Symposium on Artificial Neural Networks, ESANN 2013, Bruges, Belgium, April 24-26, 2013, 2013.

[39] P. D. McNicholas, *Model-based clustering*, J. Classif. **33** (2016), no. 3, 331–373.

[40] A. Y. Ng, M. I. Jordan, and Y. Weiss, *On spectral clustering: Analysis and an algorithm*, Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada] (T. G. Dietterich, S. Becker, and Z. Ghahramani, eds.), MIT Press, 2001, pp. 849–856.

[41] M. Rahgoshay and M. R. Salavatipour, *Hierarchical clustering: New bounds and objective*, https://arxiv.org/pdf/2111.06863.pdf.

[42] C. K. Reddy and B. Vinzamuri, *A survey of partitional and hierarchical clustering algorithms*, Data Clustering: Algorithms and Applications, 2013.

[43] M. Z. Rodriguez, C. H. Comin, D. Casanova, O. M. Bruno, D. R. Amancio, F. A. Rodrigues, and L. da Fontoura Costa, *Clustering algorithms: A comparative approach*, PLoS ONE **14** (2019).

[44] R. Scitovski, K. Sabo, F. Martínez-Álvarez, and Š. Ungar, *Cluster analysis and applications*, Springer International Publishing, 2021.

[45] P. Singh and P. A. Meshram, *Survey of density based clustering algorithms and its variants*, 2017 International Conference on Inventive Computing and Informatics (ICICI), 2017, pp. 920–926.

[46] M. C. Thrun and A. Ultsch, *Clustering benchmark datasets exploiting the fundamental clustering problems*, Data in Brief **30** (2020), 105501.

[47] N. G. Trillos, P. He, and C. Li, *Large sample spectral analysis of graph-based multi-manifold clustering*, (2021), https://arxiv.org/abs/2107.13610.

[48] D. Q. Vu, D. R. Hunter, and M. Schweinberger, *Model-based clustering of large networks*, The Annals of Applied Statistics **7** (2013), 1010–1039.

[49] H. Wang, Y. Zhang, M. Chen, and T. Yang, *Spectral clustering with smooth tiny clusters*, https://arxiv.org/pdf/2009.04674.pdf.

[50] Y. Wang, Y. Jiang, Y. W, and Z. H. Zhou, *Multi-manifold clustering*, PRICAI, 2010, http://129.211.169.156/publication/pricai10.pdf.

[51] Y. Wang, Y. Jiang, Y. Wu, and Z. H. Zhou, *Spectral clustering on multiple manifolds*, IEEE Trans. Neural Networks **22** (2011), no. 7, 1149–1161.

[52] M. Wegmann, D. Zipperling, J. Hillenbrand, and J. Fleischer, *A review of systematic selection of clustering algorithms and their evaluation*, https://arxiv.org/ftp/arxiv/papers/2106/2106.12792.pdf.

[53] Wikipedia, *Hierarchical clustering*, https://en.wikipedia.org/wiki/Hierarchical_clustering.

[54] ———, *Spectral clustering*, https://en.wikipedia.org/wiki/Spectral_clustering.

[55] D. Xu and Y. Tian, *A comprehensive survey of clustering algorithms*, Annals of Data Science **2** (2015), 165–193.

[56] X. Ye and J. Zhao, *Multi-manifold clustering: A graph-constrained deep nonparametric method*, Pattern Recognition **93** (2019), 215–227.

[57] X. Ye, J. Zhao, and Y. Chen, *A nonparametric model for multi-manifold clustering with mixture of gaussians and graph consistency*, Entropy **20** (2018), no. 11, 830.

[58] J. Zhang, M. Pechenizkiy, Y. Pei, and J. Efremova, *A robust density-based clustering algorithm for multi-manifold structure*, Proceedings of the 31st Annual ACM Symposium on Applied Computing (New York, NY, USA), SAC '16, Association for Computing Machinery, 2016, p. 832–838.

[59] Z. Zhang, K. L. Lange, and J. Xu, *Simple and scalable sparse k-means clustering via feature ranking*, https://arxiv.org/pdf/2002.08541.pdf.

[60] I. M. Ziko, J. Yuan, E. Granger, and I. B. Ayed, *Variational fair clustering*, Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, AAAI Press, 2021, pp. 11202–11209.

!Optimizer

Sharif Optimization and Applications Laboratory,
Department of Mathematical Sciences,
Sharif University of Technology,
Tehran, Iran.
optimizer.math.sharif.edu