Ring-like clustering on noisy data

Artificial Intelligence 2023/24 – Main practical assignment

Jose María Moyano Murillo & Agustín Riscos Núñez

1 Introduction

The topic of this assignment is inspired on a real subproblem within the LHCb experiment at CERN, although it has been (obviously) significantly simplified. The RICH (Ring Imaging Cherenkov) detectors try to identify patterns with a circular or elliptic patterns which are caused by the impact of Cherenkov radiation on the sensors: cones of photons emitted by particles resulting from hadrons collisions when such particles travel through a special gas at a speed higher than the speed of light in that gas medium. Nevertheless, we will skip all physical details in this assignment, reformulating the problem as follows:

> *Given a collection of points within a given range (for example, a 100x100 square area), the goal is to find the best set of "rings" modelling the given data in such a way that all points from the cloud fit into one of those rings.*

Of course, we assume that there will be some noise, so that we do not expect a perfect fit, we just try to minimize the error. The term "noise" is used here in a broad sense: it can refer both to "ghost points" that do not belong to any circumference and can be ignored, as well as to small errors in the measurements.

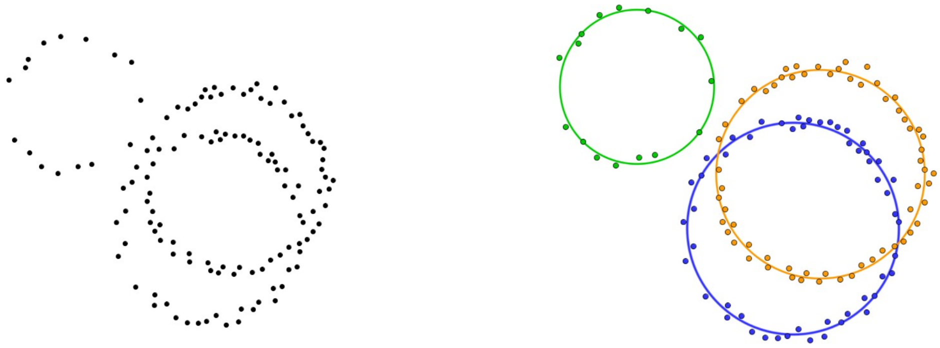

Figure 1. Example of input data (left) and their correct classification (right)

The **main goals** of this assignment are:

1. You are expected to design and implement an adapted clustering algorithm that searches for circumferences, and takes uncertainty explicitly into account. That is, each cluster will be represented by a circumference (e.g. providing its center and radius), while each point can be associated with several (all) clusters, but featuring different "membership degree".
2. You are expected to run a battery of experiments, understand the behaviour of the algorithm, analyze the results, and be able to present them in written and oral form.

The algorithm should follow the following **general scheme**:

```
1   Cluster initialization (set the center and radius of each initial circumference)
2   Loop (until halting condition is met)
        1.1 update membership degree of points to all circumferences
        2.2 update center and radius of all circumferences
3   Assign to each point one cluster, and return the final solution.
```

Pseudocode 1. General scheme of the clustering algorithm to be designed and implemented

Observations about the general scheme:

1 Regarding the initialization, you can choose to use some predefined values (basic version) or to implement some heuristic/random method (advanced version).

2 Regarding the halting condition, you can choose to use a predefined number of iterations (basic version), or you can implement a method that checks if the clusters have converged to a fixed/stable situation (advanced version).

3 Regarding the presentation of the final solution, you can choose to make a complete assignment immediately after the loop halts (each point is associated to the cluster with highest membership degree) and then return for each cluster its centre, radius and list of assigned points (basic version) or you can implement a pre-processing that (a) discards points having a bad fit to all clusters, considering them as "noise" and/or (b) combines several clusters into one if they represent almost identical circumferences (advanced version).

Specific goals:

1. Design and implement a data structure in Python suitable for storing the updated information on each round of the loop of the main algorithm (centre and radius of each cluster, together with the membership degree of each point to all clusters).

   You should describe the details of the design in the report that will be submitted along with the code.

2. Design and implement an auxiliary method such that, **given a set of points as input**, which we assume that belong to the same circular trajectory, **returns as output** an estimation of the **centre and the radius** of such a circumference. You should explain the mathematical details of the chosen method, both in the case of an original idea as well as if you followed some bibliographical or web reference. Experimental results about the performance of this method should be included in the report. You can freely choose to use either a global method, performing the calculation over the whole set of points, or a local method, which calculates the approximation from a sample of points (e.g. taking only three points of the set, as separated as possible). It is allowed to work under the assumption that the points are distributed in a uniform way along the circumference that is being detected (basic version), but we would like you to consider also the possibility that the points are concentrated only on a partial arc (advanced version).

3. Design and implement an auxiliary method that, **given one point and a set of clusters as input**, **returns as output the set** of values of the **membership degree** of the given point to each of the clusters. You can freely choose the formula to calculate the membership degree, as long as the following conditions are satisfied: (a) a very high membership value corresponds to a very small distance between the point and the circumference (and conversely, a very low membership value corresponds to a very large distance); and (b) the set of membership values for each point should be normalized in such a way that the sum of the values is 1.

   Note: the distance between a point P and a circumference c having centre O and radius r is given by the formula:

   $$\mathrm{d}(P,c) = |\, \mathrm{d}(P,O) - r \,|$$

4. Design and implement a modification of the method described in specific goal 2, so that it works receiving as input the whole set of points. That is, we do not assume any more that all points belong to the same circular trajectory, but the contribution of each point to the calculation of the new centre and radius of a cluster will be weighted according to their previously calculated membership degree to that cluster.

5. Design and implement a complete tool to solve the problem formulated in the introduction by means of the ring-clustering technique with uncertainty, according to Pseudocode 1.
6. Write a report containing the project documentation, explaining in detail all your decisions concerning the data representation and the methods design and implementation. The report should be written in the form of a scientific paper (see Section 2.3 for further details).
7. Make a short presentation about the developed code and the obtained results of the experiments during the testing phase (the interviews for the presentations will be scheduled and announced after the submission deadline).

In order for the assignment to be accepted for evaluation, **ALL specific goals** should be accomplished. The code should be original and run with no errors, and it should include the possibility to replicate the experiments that have been carried out and analysed in the report (either with a dedicated script or with detailed instructions). The report should follow the template and guidelines explained in Section 2.3.

2 Assignment description

The methodology to be followed for the correct development of the work is introduced below.

2.1. Algorithms implementation

Both the data structure chosen for the representation and storage of the data, as well as the functions and algorithms to be designed and implemented should preferably be original but may be based on similar structures used in Python libraries available in recognized repositories (which should be conveniently cited). Please, **do not use** the *fuzzy c-means* algorithm.

The code should not be designed specifically for specific data but should be parametric and flexible, allowing new experiments to be carried out easily with other data and different settings (different range of values for the points coordinates, different number of clusters, etc).

The set of points received as input will be provided through a separate file, which can be either *.csv*, or having the desired format restrictions, as long as a README.txt file with the necessary indications is provided (see Evaluation Criteria section).

2.2. Experiments

The experimentation should be approached as a battery of tests with different configurations, executed on examples with known solutions: separate rings, concentric rings, close rings with multiple intersections, etc., analyzing and comparing the performance obtained by launching multiple runs of the algorithm, by:

- repeating several times the execution for each configuration (since, if the initial clusters are chosen randomly, this may affect the result),
- testing for each example dataset with different number of initial clusters,
- adding different noise levels (additional background points that do not belong to any cluster) to each example,
- …

It is expected that a comparative analysis of the results obtained from each experiment, with respect to some numerical metrics (e.g. runtime of each experiment, number of iterations performed until convergence, sum of distances of the points to the circumference for each cluster, number of points classified as noise, etc), will be included in the report. It is preferable to present the results grouped in tables, not just copying all the outputs obtained in the document. The inclusion of a graphical presentation of the results will also be considered as an additional improvement, but it is not a requirement.

2.3. Report

On the course webpage you will find templates with a suggested structure and general format of the report to be presented in different formats (*.tex*, *.odt* and *.doc*). These templates are based on the official format of the IEEE conference proceedings [1]. Anyway, the submitted document must be in PDF format.

The paper should be at least 6 pages long, and the general structure of the paper should be as follows: First, an introduction explaining the main objective, including a brief background on the subject of the work and the methods used (mention bibliographic references); then describe the structure of the work, the design decisions that were made during its development (also mentioning elements that were initially considered but subsequently discarded, if any), as well as the methodology followed to implement it (never include code in the report, but pseudocode); then detail the experiments carried out, analyzing the results obtained. Finally, the document must include a conclusions section, and a bibliography that includes not only the references cited in the introduction section, but also any documents consulted during the realization of the work (including web references to pages or repositories).

2.4. Optional improvements

Although not being mandatory, the inclusion of a user-friendly graphical interface or menu that facilitates experimentation or the presentation of results will be considered in the evaluation. Other improvements or additions to the paper beyond the minimum requirements mentioned in the list of specific objectives may also receive additional points.

For example: implementing several different methods for calculating the new value of the center and radius of a cluster and comparing their efficiency, designing and implementing an automatic example generator that, given a set of circles, generates a set of points approximately distributed on those trajectories, etc.

2.5. Submission and defense

It is necessary to meet the requirements described in the Introduction and Objectives section (each and every one of them, at least in its basic version) for the work to be evaluated. A single compressed *.zip* file must be submitted to the corresponding task in the course webpage, including:

- **A folder containing the source code.** Within this folder, there must be a README.txt file that summarizes the structure of the source code and indicates how to use the interface (if implemented), or at least how to test the implemented functions, including usage examples. It should also indicate how to reproduce the experiments performed. It is important the coherence of this file with the defense.
- **The document/article in PDF format.** It must be at least 6 pages long. It must include all the consulted bibliography (books, articles, technical reports, web pages, source codes, slides, etc.) in the references section and mention them throughout the document.

On the day of the defense (that will be announced to each student with enough time), a small presentation (PDF, PowerPoint, or similar) of 10 minutes should be made. This presentation will follow roughly the same structure of the paper, but special emphasis should be placed on the results obtained and their critical analysis. A laptop (the student's own), slides and/or a blackboard may be used. During the next 10 minutes of the defense, the professor will ask questions about the work, which may include both the document and the source code.

3 Evaluation criteria

The evaluation will be based on the following criteria, with the maximum score corresponding to the advanced version (if the basic version described in Section 1 is chosen, the scores for the first three sections will be reduced by 0.25 each):

- **The source code (0.75 points)**: clarity of the code and good programming style, correctness and efficiency of implementation, and quality of comments will be evaluated. The legibility and clarity of the README.txt file will also be evaluated. Any work with code copied directly from other colleagues or from the Internet will not be evaluated. The work may make partial use of existing Python libraries, but only original code developed by the student will be evaluated in this section.
- **Ease of use and experimentation (0.75 points):** The number of experiments performed and the quality and interest of the data used will be evaluated: size and complexity of the examples, variety of tests performed (varying the initialization parameters and/or stop criteria, etc.) and the clarity with which the results obtained are presented and analyzed in the report (statistics, tables and/or graphs). Usability is also evaluated when reproducing the same or new experiments, both in terms of input (e.g., ease of introducing instances and parameters) and output (ability to provide not only the final result, but also details of the intermediate steps of the loop).
- **The scientific report (0.75 points):**
  - Appropriate use of language and the overall report style will be evaluated (e.g., using the suggested template).
  - The preliminary research work done will be evaluated as long as it is clear that the research is understood.
  - Clarity of explanations, reasoning of decisions, and especially the analysis and presentation of results in the Experiments and Conclusions sections will generally be evaluated.
  - Similarly, the paper will not be evaluated if any copying is detected. The references section should include all relevant sources consulted.
- **The presentation and defense (0.75 points):** The clarity of the presentation and the good explanation of the contents of the work will be evaluated, as well as, in particular, the answers to the questions asked by the professor.
- **Improvements:** They will be evaluated with up to 0.5 extra points, without exceeding the maximum total of 3 points in any case.

IMPORTANT NOTE: Any plagiarism, code sharing, or use of material that is not original and does not properly cite the source will automatically result in a zero grade for the course for all students involved. Therefore, these students will not receive any grades for the current call. This is without prejudice to any disciplinary action that may be taken.

4 References

[1] IEEE (2019). IEEE Template. https://www.ieee.org/conferences_events/conferences/publishing/templates.html. (Last accesed: 19/03/2024)