Time	Group	Submission in Moodle; Mails with subject: [SMD2023]
Th. 12:00–13:00	A	lukas.beiske@udo.edu and tristan.gradetzke@udo.edu
Fr. 08:45–09:45	B	jonas.hackfeld@ruhr-uni-bochum.de and ludwig.neste@udo.edu
Fr. 10:00–11:00	C	stefan.froese@udo.edu and vincent.latko@udo.edu

Exercise 20 *k-NN Classification* **6 p.**

(a) What problems occur when using a k-NN algorithm with attributes that differ greatly in magnitude? How can you solve them?

(b) Why do we call the k-NN a so-called "lazy learner"? What are the runtimes for learning and application phases? How do they compare to other algorithms such as a random forest?

(c) Implement a k-NN algorithm for the classification of events. Follow the class structure given in the attached file `class_structure.py`. The method `predict` should output a numpy array containing the predicted label for each sample. **Procedure:** For each event to be classified:

1) Calculate the distances to all points of the training sample.
2) Determine the k training events with the smallest distance (note: determine only the indices of the events instead of sorting the array itself).

   Hint: The Numpy function `numpy.argsort()` can be useful.
3) Determine the label that occurs most frequently in these events.

(d) Apply your algorithm to the neutrino Monte Carlo of sheet 5. Use the `NeutrinoMC.hdf5` file provided in Moodle.

- Use the attributes `CountHits`, `x` and `y`.
- Set $k = 10$.
- Use 5000 events as a training set.
- The test set shall consist of 20 000 underground and 10 000 signal events.

Determine recall, precision and significance.

(e) What changes if you use `log10(CountHits)` instead of `CountHits`?

(f) What changes if you use $k = 20$ instead of $k = 10$?

Exercise 21 *kMeans by Hand* **4 p.**

Population: (1;4) (1;5) (1;6) (3;3) (3;2) (4;1) (5;1) (6;2) (6;3) (8;4) (8;5) (8;6)

(a) Perform the kMeans algorithm (Euclidean distance measure) by hand to group the points of the population into clusters. Use the randomly chosen cluster centers (3;4), (7;4) and (3;7) as initial values. Calculate the distances only if the cluster center membership is not obvious. Sketch the new cluster centers and the boundaries between the clusters in the prebuilt graphic 1.

(b) Perform 4 more iterations of kMeans. Make a sketch again for each iteration.

(c) After how many iterations does the algorithm converge? Does the result meet your expectations?

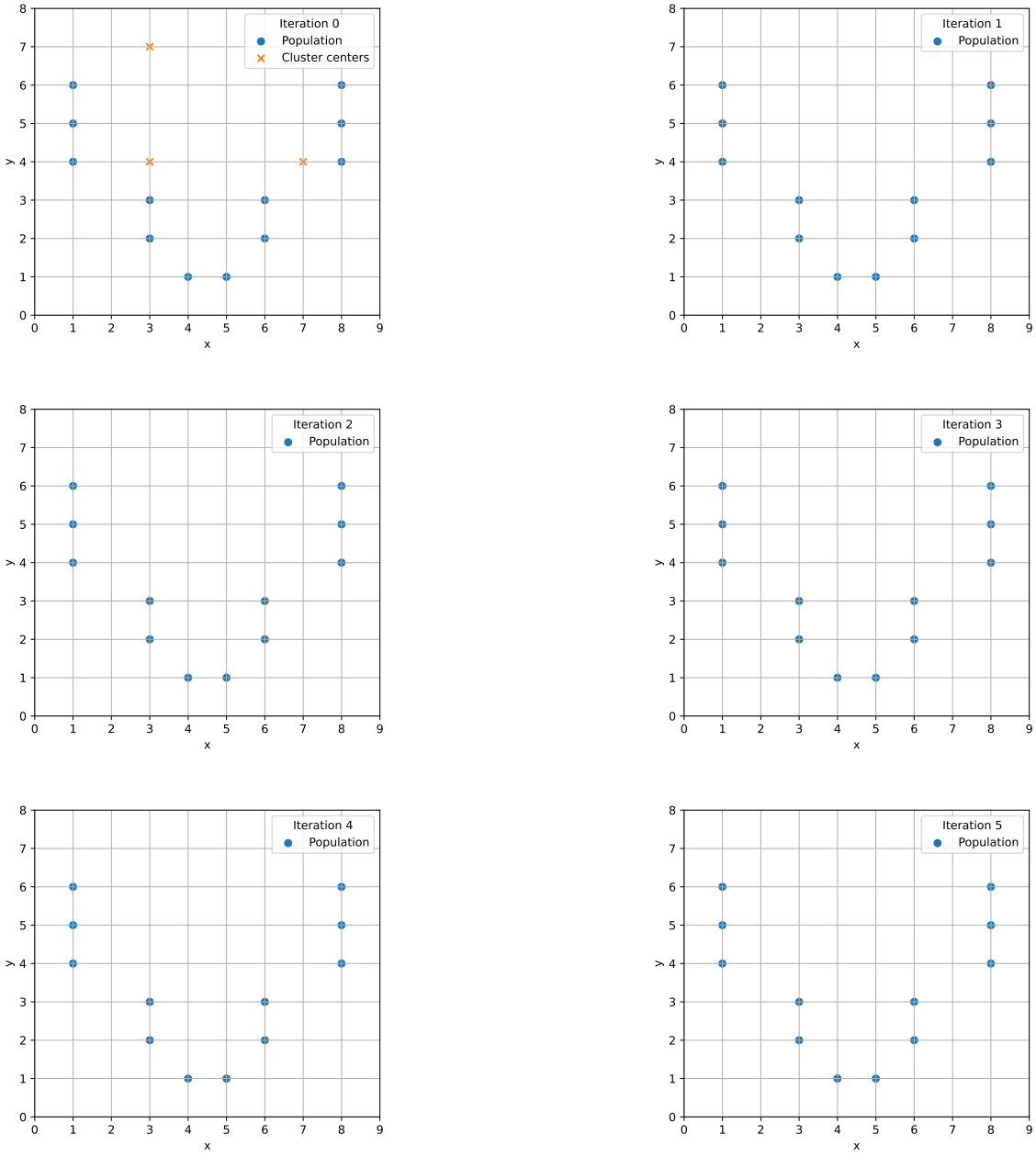

Figure 1: Populations and templates to draw the cluster centers and cluster boundaries for task 21