You will implement algorithms for clustering vectors in the Euclidean space RK" and of polygonal curves in Euclidean space 32 using the 8 combinations of variations that follow. The metric I1 (MapmaiaP) will be used for the vectors and n pseudometric Onpapeis TiPie nNaripoa (OTNN) for curves.

Initialization

1. Random selection of K points / K curves (simplest)

2. K-means++

Assignment

1. Lloyd's assignment (simplest approach)

2. Assignment by Range search with LSH for vectors / curves (inverse assignment)
Update

1. Partitioning Around Medoids (PAM) 'a la Lloyds

2. Ymodoyiote to Mean Vector / YroAoyiote to DTW centroid Curve

ENTRANCE
1) A text file 1pr. ᾶθτ separated by spaces, colons or commas (5ρᾶε-σερᾶ[aiea, ῖ80-

"vthraGaῖTha or "OPIPIᾶ-ethrᾶ[aitha), which will have the following notation in the case of
polygonal curves:

curves
curve idl mL (x11,y11) (x12,y12) wee (xlm1i,ylml1l)
curve 14N mN (xN1,yN1) (xN2,YN2) eee (xNmN, ylmN)

where (S11, c11) the coordinates ᾰουβῐε tou j onueiou ths curve], where ] « pi and piito number of
points of the curve |.

In NEpintwon Twv StavvovaTwv To input. dat has the following notation:
vectors

item idl X11 X12

item idN XN1 XN2
where Xi] the coordinates of the infinite vector that avamaplota To item i

2) A configuration file of the following format (lines where there is a
value may not be given so the default value is used):

number_of_clusters: <int> // Kof K-means
number_of_grids: <int> //default: 2
number_of_vector_hash_tables: <int> //default: L=3

number_of_vector_hash_functions: <int> // kof LSH for vectors

Ta apxeia input.dat, cluster.conf divovtal Léow MapaLeToWV OTN YPAULN EvtoAwy. H EKTéAEOn
will be done through the command:

$./cluster -i <input file> -c <configuration file> -o <output file> -

complete <optional>

exit

A text file containing the clusters of data produced by each
variant of the algorithm, the execution time in each case as well as the internal
index
agéioAdynons tn¢ ovotadoroinons Silhouette.

In the case of vectors, the output file must follow the following pattern, the
which is repeated for each variant:

Algorithm: IxAxUx
CLUSTER-1 {size: <int>, centroid: <item _id> or array pe TL¢ suvtTetaypéveg tou
centroid
in the case of k-means Update}

CLUSTER-K {size: <int>, centroid: <item _id> or array pe TL¢ suvtTetaypéveg tou
centroid
otnv meptutwon K-means Update }

clustering time: <double> //in seconds

Silhouette: [sl,...,si,...,sK, stotal]

/* si=average s(p) of points in cluster i, stotal=average s(p) of points in dataset
*/

/* Optionally with command line parameter -complete */
CLUSTER-1 {item idA, item idB, ..., item idC}

CLUSTER-K {item idR, item idT, ..., item idZ}

In the case of polygonal curves, the output file must follow the following template, which is repeated for each variation:

Algorithm: IxAxUx

CLUSTER-1 {size: <int>, centroid: <curve_id> or array of onpefa tou centroid in meptutwon DTW medoid curve update}

CLUSTER-K {size: <int>, centroid: <curve id>}

clustering time: <double> //in seconds

Silhouette: [sl,...,si,...,sK, stotal]

/* si=average s(c) of curves in cluster i, stotal=average s(c) of curves in dataset */

/* Optionally with command line parameter -complete */
CLUSTER-1 {curve idA, curve idB, ..., curve 140}

CLUSTER-K {curve idR, curve idT, ..., curve 146}

Additional requirements

1. File (or section in VeDaPITH) that compares algorithms based on results.

2. The program must be well organized with separation of declarations / definitions functions, structures, and data types into logical groups that correspond to separate
header and source code files. The program must be compiled with the
use of the tool piõKke and the existence of a suitable MDKePI6. Its quality is also rated
code (e.g. avoiding PIEPIOI |6õ(s).

3. The deliverable must be adequately documented with full code commentary and the existence of
file (6b(ŋ]e which includes at least: a) title and description of the program, b) list of code / header files and their description, c) its compilation instructions of the program, 6) instructions for using the program and e) full details of the students who use it
they developed.

4. The implementation of the program should be done using a version management system
software and collaboration ((iί or 5NN) [groups of 2 people].

5. Using an appropriate library and performing unit testing.