

Project-K23 Μέρος 2ο

Υλοποίηση των αλγορίθμων συσταδοποίησης K-means/K-medoids για διανύσματα και πολυγωνικές καμπύλες στη γλώσσα C++.

Τα αρχεία του προγράμματος είναι:

- **structs.h**: περιλαμβάνει τη δήλωση των κλάσεων `Vector_Item`, `Bucket`, `Hypercube_vertices`, `Point`, `Curve`
- **cluster.cpp**: περιλαμβάνει την υλοποίηση της `main` όπου ελέγχουμε το `infile` αρχείο αν έχει `vectores` ή `curves` και αντίστοιχα κάνουμε ένα `loop` για τις 8 περιπτώσεις του `clustering`.
- **cluster_funct.cpp**: περιλαμβάνει την υλοποίηση των συναρτήσεων που χρησιμοποιήσαμε για το `clustering` των `vectors` σύμφωνα με τους αλγορίθμους που μας δώθηκαν στο μάθημα.
- **cluster_curve_funct**: περιλαμβάνει την υλοποίηση των συναρτήσεων που χρησιμοποιήσαμε για το `clustering` των `curves` σύμφωνα με τους αλγορίθμους που μας δώθηκαν στο μάθημα.
- **funct.cpp**: περιλαμβάνει τις συναρτήσεις που χρησιμοποιούνται στη `main` και αρχικοποιούν τους `vectors` με το `dataset` και άλλες βοηθητικές συναρτήσεις
- **VectorItem.cpp** : περιλαμβάνει την υλοποίηση των συναρτήσεων της κλάσης `Vector_Item`
- **hash.cpp**: περιλαμβάνει την υλοποίηση των συναρτήσεων που χρησιμοποιήσαμε για το `hashing` σύμφωνα με τύπους που μας δώθηκαν στο μάθημα
- **bucket.cpp**: περιλαμβάνει τη υλοποίηση των συναρτήσεων για το `Bucket` που χρησιμοποιήσαμε στα `hash tables`
- **Point.cpp**: περιλαμβάνει την υλοποίηση των συναρτήσεων της κλάσης `Point`
- **Curve.cpp**: περιλαμβάνει την υλοποίηση των συναρτήσεων της κλάσης `Curve`
- **Cluster.cpp**: περιλαμβάνει την υλοποίηση των συναρτήσεων της κλάσης `Cluster`
- **curve_funct.cpp**: περιλαμβάνει την υλοποίηση των συναρτήσεων που χρησιμοποιούνται στις `main` για τους αλγορίθμους των καμπυλών γιατί είχαμε διαφορετική επεξεργασία του `dataset`
- **Makefile** : περιλαμβάνει την υλοποίηση του `makefile` για τα 4 εκτελέσιμα και την λειτουργία `make clean`
- οι δηλώσεις των συναρτήσεων βρίσκονται στα αρχεία `.h`

Μεταγλώττιση:

make

Εκτέλεση:

```
./cluster -i DataVectors_5_500x100.csv -c cluster.conf -o output_cluster.out -complete 1
```

Αξιολόγηση:

Για την αξιολόγηση των αλγορίθμων υλοποιήσαμε τη μέθοδο Silhouette τόσο για τα διανύσματα όσο και για τις καμπύλες. Τα αποτελέσματα που μας δώθηκαν σε dataset 455 διανυσμάτων και 1600 καμπυλών με 5 clusters ήταν τα εξής:

I1A1U1: 0.17 - 0.78s

I1A1U2: 0.92 - 0.099s

I1A2U1: 0.36 - 2.19s

I1A2U2: 0.2 - 0.277s

I2A1U1: 0.89 - 1.65s

I2A1U2: 0.92 - 1.13s

I2A2U1: 0.36 - 3.24s

I2A2U2: 0.44 - 1.373s

Άρα παρατηρούμε ότι για το συγκεκριμένο παράδειγμα είναι πιο αποδοτικός ο αλγόριθμος με την τυχαία επιλογή κέντρων, την ανάθεση των σημείων σειριακά στο κοντινότερο κέντρο και το k-means update τόσο στο χρόνο εκτέλεσης όσο και στην ακρίβεια της συσταδοποίησης.

Παρατηρήσαμε όμως ότι αυτό αλλάζει ανάλογα με το dataset και τον αριθμό των clusters που δίνουμε κάθε φορά.