Κ23γ: Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα
Χειμερινό εξάμηνο 2019-20

2η Προγραμματιστική Εργασία
Υλοποίηση των αλγορίθμων συσταδοποίησης K-means / K-medoids για διανύσματα και πολυγωνικές καμπύλες στη γλώσσα C/C++

Η άσκηση πρέπει να υλοποιηθεί σε σύστημα Linux και να υποβληθεί στις Εργασίες του e-class το αργότερο την Παρασκευή 6/12 στις 23.59.

Περιγραφή της άσκησης

Θα υλοποιήσετε αλγορίθμους για τη συσταδοποίηση διανυσμάτων στον ευκλείδειο χώρο $\Re^d$ και πολυγωνικών καμπύλων στον ευκλείδειο χώρο $\Re^2$ χρησιμοποιώντας τους 8 συνδυασμούς από τις παραλλαγές που ακολουθούν. Θα χρησιμοποιηθεί η μετρική L1 (Manhattan) για τα διανύσματα και η ψευδομετρική Dynamic Time Warping (DTW) για τις καμπύλες.

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. Υπολογίστε το Mean Vector / Υπολογίστε το DTW centroid Curve

ΕΙΣΟΔΟΣ

1) Ένα αρχείο κειμένου `input.dat` διαχωρισμένο με κενά, στηλοθέτες ή κόμματα (space-separated, tab-separated ή comma-separated), το οποίο θα έχει την ακόλουθη γραμμογράφηση στην περίπτωση των πολυγωνικών καμπυλών:

```
curves
curve_id1      m1      (x11,y11)  (x12,y12)  ...   (x1m1,y1m1)
    .                      .          .      ...        .
curve_idN      mN      (xN1,yN1)  (xN2,yN2)  ...   (xNmN,y1mN)
```

όπου `(xij,yij)` οι συντεταγμένες double του j σημείου της καμπύλης i, όπου j ≤ mi και mi το πλήθος των σημείων της καμπύλης i.

Στην περίπτωση των διανυσμάτων το `input.dat` έχει την ακόλουθη γραμμογράφηση:

```
vectors

item_id1      X11     X12       ...
.          .          .    ...
item_idN      XN1     XN2       ...
```

όπου **Xij** οι συντεταγμένες double του διανύσματος που αναπαριστά το item i

2) Ένα αρχείο ρύθμισης παραμέτρων cluster.conf με την ακόλουθη μορφή (γραμμές όπου υπάρχει default τιμή μπορούν να μην δινονται οπότε χρησιμοποιείται η default τιμή):

number_of_clusters: <int> // K of K-means
number_of_grids: <int> //default: 2
number_of_vector_hash_tables: <int> //default: L=3
number_of_vector_hash_functions: <int> // k of LSH for vectors

Τα αρχεία `input.dat, cluster.conf` δίνονται μέσω παραμέτρων στη γραμμή εντολών. Η εκτέλεση θα γίνεται μέσω της εντολής:
`$./cluster –i <input file> –c <configuration file> -o <output file> -complete <optional>`

ΕΞΟΔΟΣ

Ένα αρχείο κειμένου το οποίο περιλαμβάνει τις συστάδες των δεδομένων που παρήχθησαν από κάθε παραλλαγή του αλγορίθμου, τον χρόνο εκτέλεσης σε κάθε περίπτωση καθώς και τον δείκτη εσωτερικής αξιολόγησης της συσταδοποίησης **Silhouette**.

Στην περίπτωση των διανυσμάτων, το αρχείο εξόδου ακολουθεί υποχρεωτικά το παρακάτω πρότυπο, το οποίο επαναλαμβάνεται για κάθε παραλλαγή:

```
Algorithm: IxAxUx
CLUSTER-1 {size: <int>, centroid: <item_id> ή πίνακας με τις συντεταγμένες του centroid στην περίπτωση k-means Update}
.      .      .      .      .      .      .
CLUSTER-K {size: <int>, centroid: <item_id> ή πίνακας με τις συντεταγμένες του centroid στην περίπτωση K-means Update }
clustering_time: <double> //in seconds
Silhouette: [s1,...,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}
```

Στην περίπτωση των πολυγωνικών καμπυλών, το αρχείο εξόδου ακολουθεί υποχρεωτικά το παρακάτω πρότυπο, το οποίο επαναλαμβάνεται για κάθε παραλλαγή:

```
Algorithm: IxAxUx
CLUSTER-1 {size: <int>, centroid: <curve_id> ή πίνακας με τα σημεία του centroid στην περίπτωση DTW medoid curve update}
CLUSTER-K {size: <int>, centroid: <curve_id>}
clustering_time: <double> //in seconds
Silhouette: [s1,...,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_idC}
.      .      .      .      .      .      .

CLUSTER-K {curve_idR, curve_idT, ..., curve_idZ}
```

Επιπρόσθετες απαιτήσεις

1. Αρχείο (ή ενότητα στο Readme) που συγκρίνει τους αλγορίθμους με βάση τα αποτελέσματα.

2. Το πρόγραμμα πρέπει να είναι καλά οργανωμένο με χωρισμό των δηλώσεων / ορισμών των συναρτήσεων, των δομών και των τύπων δεδομένων σε λογικές ομάδες που αντιστοιχούν σε ξεχωριστά αρχεία επικεφαλίδων και πηγαίου κώδικα. Η μεταγλώττιση του προγράμματος πρέπει να γίνεται με τη χρήση του εργαλείου make και την ύπαρξη κατάλληλου Makefile. Βαθμολογείται και η ποιότητα του κώδικα (π.χ. αποφυγή memory leaks).

3. Το παραδοτέο πρέπει να είναι επαρκώς τεκμηριωμένο με πλήρη σχολιασμό του κώδικα και την ύπαρξη αρχείου readme το οποίο περιλαμβάνει κατ' ελάχιστο: α) τίτλο και περιγραφή του προγράμματος, β) κατάλογο των αρχείων κώδικα / επικεφαλίδων και περιγραφή τους, γ) οδηγίες μεταγλώττισης του προγράμματος, δ) οδηγίες χρήσης του προγράμματος και ε) πλήρη στοιχεία των φοιτητών που το ανέπτυξαν.

4. Η υλοποίηση του προγράμματος θα πρέπει να γίνει με τη χρήση συστήματος διαχείρισης εκδόσεων λογισμικού και συνεργασίας (Git ή SVN) [ομάδες 2 ατόμων].

5. Χρήση κατάλληλης βιβλιοθήκης και εκτέλεση ελέγχων μονάδων λογισμικού (unit testing).