Κ23γ: Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα
Χειμερινό εξάμηνο 2017-18

2η Προγραμματιστική Εργασία
Υλοποίηση των αλγορίθμων συσταδοποίησης K-means / K-medoids στη γλώσσα C/C++

Η άσκηση πρέπει να υλοποιηθεί σε σύστημα Linux και να υποβληθεί στις Εργασίες του e-class το αργότερο την Παρασκευή 30/11 στις 23.59.

Περιγραφή της άσκησης

Θα υλοποιήσετε αλγορίθμους για τη συσταδοποίηση διανυσμάτων στον χώρο $\mathfrak{R}^d$, χρησιμοποιώντας τους 12 συνδυασμούς από τις παραλλαγές που ακολουθούν. Θα χρησιμοποιηθούν οι αποστάσεις Euclidean και Cosine.

Initialization
1. Random selection of k points (simplest)
2. K-means++

Assignment
1. Lloyd's assignment
2. Assignment by Range search with LSH (εργασία 1)
3. Assignment by Range search with Hypercube (εργασία 1)

Update
1. K-means
2. Partitioning Around Medoids (PAM) improved like Lloyd's

ΕΙΣΟΔΟΣ

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

```
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 δίνονται οπότε χρησιμοποιείται η default τιμή):

```
number_of_clusters: <int>              // k
number_of_hash_functions: <int>        //default:4
number_of_hash_tables: <int>           //default:L=5
```

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> -d <metric>
```

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**. Το αρχείο εξόδου ακολουθεί υποχρεωτικά το παρακάτω πρότυπο, το οποίο επαναλαμβάνεται για κάθε παραλλαγή:

```
Algorithm: IxAxUx
Metric: Euclidean ή Cosine
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 */

/* Additionally with command line parameter –complete */
CLUSTER-1 {item_idA, item_idB, ..., item_idC}
.      .      .      .      .      .      .

CLUSTER-k {item_idR, item_idT, ..., item_idZ}
```

Επιπρόσθετες απαιτήσεις

1. 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).

2. 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, ε) πλήρη στοιχεία του φοιτητή που το ανέπτυξε.

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

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

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