

## Κ23γ: Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα Χειμερινό εξάμηνο 2021-22

### 1<sup>η</sup> Προγραμματιστική Εργασία

#### Αναζήτηση και συσταδοποίηση διανυσμάτων στη C/C++

Η άσκηση θα υλοποιηθεί σε σύστημα Linux και θα υποβληθεί στις Εργασίες του e-class το αργότερο την Παρασκευή 12/11 στις 23.59.

#### Περιγραφή της εργασίας

A. Υλοποιήστε τον αλγόριθμο LSH για διανύσματα στον  $d$ -διάστατο χώρο βάσει της μετρικής L2, καθώς και τον αλγόριθμο τυχαίας προβολής στον υπερκύβο για την ίδια μετρική. Το πρόγραμμα θα υλοποιηθεί έτσι ώστε λαμβάνοντας ως είσοδο διάνυσμα  $q$  και ακέραιους  $N$  και  $R$ , να επιστρέφει προσεγγιστικά: α) τον πλησιέστερο γείτονα στο  $q$ , β) τους  $N$  πλησιέστερους γείτονες στο  $q$  και γ) τα διανύσματα εντός ακτίνας  $R$  από το  $q$  (range search). Ο σχεδιασμός του κώδικα θα πρέπει να επιτρέπει την εύκολη επέκτασή του σε διανυσματικούς χώρους με άλλη μετρική, π.χ.,  $p$ -norm, ή διαφορετικούς χώρους.

B. Υλοποίηση αλγορίθμων για τη συσταδοποίηση διανυσμάτων στον χώρο  $\mathbb{R}^d$ . Θα χρησιμοποιηθεί η μετρική L2. (Η αρχικοποίηση πραγματοποιείται με την τεχνική k-Means++ και η ενημέρωση με τον υπολογισμό του μέσου διανύσματος). Οι αλγόριθμοι διαφοροποιούνται στο βήμα της ανάθεσης όπου χρησιμοποιείται (α) ο ακριβής αλγόριθμος Lloyd's, και η αντίστροφη ανάθεση (reverse) μέσω Range search με (β) LSH, ή (γ) Τυχαία προβολή.

#### ΕΙΣΟΔΟΣ

A.

1) Ένα αρχείο κειμένου για την είσοδο του συνόλου δεδομένων (dataset) διαχωρισμένο με στηλοθέτες (tab-separated), με την ακόλουθη γραμμογράφηση:

item_id1	X11 X12	... X1d
.	.	...
item_idN	XN1 XN2	... XNd

όπου  $X_{ij}$  διάνυσματος  $i$  στον  $d$ -διάστατο Ευκλείδειο χώρο. Τα ονόματα (item\_idK) μπορούν να είναι μοναδικοί ακέραιοι ή συμβολοσειρές.

2) Αρχείο κειμένου που περιλαμβάνει το σύνολο αναζήτησης δηλ. των διανυσμάτων  $q$ , και περιέχει τουλάχιστον ένα διάνυσμα με την ίδια γραμμογράφηση.

Το πρόγραμμα αρχικά ζητά από τον χρήστη το μονοπάτι του dataset. Μετά τη δημιουργία της δομής αναζήτησης, το πρόγραμμα ζητά από τον χρήστη το μονοπάτι του αρχείου αναζήτησης και του αρχείου εξόδου. Μετά την εκτέλεση του αλγορίθμου και την παραγωγή των αποτελεσμάτων, το πρόγραμμα ζητά από τον χρήστη αν θέλει να τερματίσει το πρόγραμμα ή να επαναλάβει την αναζήτηση για διαφορετικό σύνολο / αρχείο αναζήτησης.

Για το LSH, μπορούν να δίνονται οι εξής παράμετροι προαιρετικά στη γραμμή εντολών: ακέραιο πλήθος  $k$  των LSH συναρτήσεων  $h_i$  που χρησιμοποιούνται για τον ορισμό των  $g$ , ο ακέραιος αριθμός  $L$  των πινάκων κατακερματισμού, ο ακέραιος αριθμός  $N$  των πλησιέστερων γειτόνων και ο δεκαδικός αριθμός  $R$  της ακτίνας αναζήτησης. Αν τα  $k$ ,  $L$  δεν δίνονται, το πρόγραμμα χρησιμοποιεί default τιμές  $k=4$ ,  $L=5$ ,  $N=1$ ,  $R=10000$ .

Για την τυχαία προβολή στον υπερκύβο, μπορούν να δίνονται οι εξής προαιρετικές παράμετροι στη γραμμή εντολών: η διάσταση στην οποία προβάλλονται τα σημεία  $k$  ( $=d'$ ), το μέγιστο επιτρεπόμενο πλήθος υποψήφιων σημείων που θα ελεγχθούν  $M$ , το μέγιστο επιτρεπόμενο πλήθος κορυφών του υπερκύβου που θα ελεγχθούν (probes), ο ακέραιος αριθμός  $N$  των πλησιέστερων γειτόνων και ο δεκαδικός αριθμός  $R$  της ακτίνας αναζήτησης. Οι default τιμές είναι:  $k=14$ ,  $M=10$ , probes=2,  $N=1$ ,  $R=10000$ .

Τα αρχεία εισόδου και αναζήτησης θα μπορούν να δίνονται και μέσω παραμέτρων στη γραμμή εντολών.

Οπότε η εκτέλεση θα γίνεται μέσω της εντολής:

```
$ ./lsh -i <input file> -q <query file> -k <int> -L <int> -y <output file> -y
<number of nearest> -R <radius>

$ ./cube -i <input file> -q <query file> -k <int> -M <int> -probes <int> -y
<output file> -y <number of nearest> -R <radius>
```

B.

1) Το αρχείο της περίπτωσης A. 1

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 of K-medians
number_of_vector_hash_tables: <int> // default: L = 3
number_of_vector_hash_functions: <int> // k of LSH for vectors, default: 4
max_number_M_hypercube: <int> // M of Hypercube, default: 10
number_of_hypercube_dimensions: <int> number_of_probes: // k of Hypercube, default: 3
<int> // probes of Hypercube, default: 2
```

Τα αρχεία input.txt, 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> -m <method: Classic OR LSH or Hypercube>
```

## ΕΞΟΔΟΣ

A. Αρχείο κειμένου που περιλαμβάνει για κάθε εικόνα του συνόλου αναζήτησης με την χρήση των κατάλληλων ετικετών: α) τον αριθμό του  $N$ -οστού προσεγγιστικά πλησιέστερου γείτονα που βρέθηκε και την απόστασή του από το  $q$ , β) την απόσταση του  $q$  από τον αληθινά  $N$ -οστό πλησιέστερο γείτονα (μέσω εξαντλητικής αναζήτησης), γ) τον χρόνο εύρεσης των (α), δ) τον χρόνο εύρεσης των (β) και ε) τους γείτονες εντός ακτίνας  $R$ . Το αρχείο εξόδου ακολουθεί υποχρεωτικά το εξής πρότυπο:

```
Query: query_id
Nearest neighbor-1: item_id
distanceLSH: <double> [or distanceHypercube respectively]
```

distanceTrue: <double>

...

Nearest neighbor-N: item\_id

distanceLSH: <double> [or distanceHypercube respectively]

distanceTrue: <double>

tLSH: <double>

tTrue: <double>

R-near neighbors:

item\_id\_A

item\_id\_B

. . .

item\_id\_Z

και ούτω καθεξής.

B. Ένα αρχείο κειμένου το οποίο περιλαμβάνει τις συστάδες των εικόνων 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: Lloyds OR Range Search LSH OR Range Search Hypercube

CLUSTER-1 {size: <int>, centroid: array with centroid} coordinates cent

. . . . .

CLUSTER-γ {size: <int>, centroid: table with the coordinates of centroid}

clustering\_time: <double> // in seconds

Silhouette: [s1, ..., si, ..., sγ, 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 {centroid, item\_id\_A, ..., item\_idX}

. . . . .

CLUSTER-γ {centroid, item\_idR, ..., item\_idZ}

## **Επιπρόσθετες απαιτήσεις**

- 1) The program must be well organized by separating the declarations / definitions of the functions, structures and types of data into logical groups corresponding to separate header and source code files. The quality of the code is also graded (eg avoiding memory leaks). The compilation of the program must be done using the make tool and the availability of a suitable Makefile.
- 2) The delivery must be sufficiently documented with full commentary of the code and the existence of a readme file which includes at least: a) title and description of the program, b) list of code files / headings and descriptions, c) instructions of the program, d) instructions for use of the program and e) complete details of the students who developed it.
- 3) The implementation of the program should be done using the publication management system software and collaboration (Git).