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

2η Προγραμματιστική Εργασία
Κατακερματισμός και αναζήτηση για χρονοσειρές στη C/C++

Η άσκηση πρέπει να υλοποιηθεί σε σύστημα Linux και να υποβληθεί στις Εργασίες του e-class το αργότερο το Σάββατο 18/12 στις 23.59.

Περιγραφή της εργασίας

Α. Θα υλοποιήσετε την αναζήτηση του πλησιέστερου γείτονα μιας χρονοσειράς q εντός ενός συνόλου χρονοσειρών με τις ακόλουθες τεχνικές:

i. Κάθε χρονοσειρά αναπαρίσταται ως διάνυσμα στον ευκλείδειο χώρο $\mathbb{R}^d$, όπου d είναι το πλήθος των χρονικών σημείων που θα επιλεχθούν από το σύνολο των καταγεγραμμένων χρονικών σημείων (για σταθερό σύνολο χρονικών στιγμών). Η τιμή της κάθε συντεταγμένης είναι η αντίστοιχη τιμή της χρονοσειράς. Η απόσταση μεταξύ διανυσμάτων υπολογίζεται βάσει της μετρικής L2. Θα χρησιμοποιηθούν οι αλγόριθμοι LSH και Hypercube της πρώτης εργασίας.

ii. Κάθε χρονοσειρά αναπαρίσταται ως πολυγωνική καμπύλη στο ευκλείδειο επίπεδο $\mathbb{R}^2$. Η απόσταση μεταξύ των χρονοσειρών υπολογίζεται βάσει της μετρικής Discrete Fréchet. Θα χρησιμοποιηθεί ο αλγόριθμος LSH για Discrete Fréchet.

iii. Οι χρονοσειρές υφίστανται προεπεξεργασία φιλτραρίσματος για τη μείωση της πολυπλοκότητάς τους. Κάθε χρονοσειρά αναπαρίσταται ως πολυγωνική καμπύλη στην ευθεία $\mathbb{R}$ απαλείφοντας τη διάσταση του χρόνου. Θα πρέπει να απεικονίσετε τις χρονοσειρές στην ευθεία των πραγματικών αριθμών υπολογίζοντας μια ακολουθία σημείων επί ενός πλέγματος σε αυτήν. Η απόσταση μεταξύ των χρονοσειρών υπολογίζεται βάσει της μετρικής Continuous Fréchet. Θα χρησιμοποιηθεί ο αλγόριθμος LSH για Continuous Fréchet.

Β. Θα υλοποιήσετε αλγορίθμους για τη συσταδοποίηση χρονοσειρών χρησιμοποιώντας τους 5 συνδυασμούς από τις παραλλαγές που ακολουθούν.

Θα χρησιμοποιηθεί η μετρική L2 και η μετρική Discrete Fréchet.

Initialization

1. K-means++

Assignment (Ανάθεση)

1. Lloyd's assignment (simplest approach)
2. Assignment by Range search (LSH Frechet)
3. Assignment by Range search (LSH Vector)
4. Assignment by Range search (Hypercube)

Update

1. Υπολογίστε τη μέση χρονοσειρά ως διάνυσμα (δεν είναι δυνατή η ανάθεση 2)
2. Υπολογίστε τη μέση χρονοσειρά ως καμπύλη (δεν είναι δυνατές οι αναθέσεις 3 και 4)

Ο σχεδιασμός του κώδικα θα πρέπει να επιτρέπει την εύκολη επέκτασή του για διαφορετικές συναρτήσεις απόστασης καθώς και τη μελλοντική χρήση μεμονωμένων συναρτήσεων σε μελλοντικές εργασίες.

ΕΙΣΟΔΟΣ

Α.

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

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

όπου `XIJ` η τιμή κατά τη χρονική στιγμή `J` της χρονοσειράς `I`. Ο ρυθμός δειγματοληψίας επισημαίνεται στο όνομα του αρχείου.

2) Αρχείο κειμένου που περιλαμβάνει το σύνολο των χρονοσειρών για τις οποίες αναζητούμε τον πλησιέστερο γείτονα με την ίδια γραμμογράφηση.

Το πρόγραμμα αρχικά ζητά από τον χρήστη το μονοπάτι του αρχείου dataset. Μετά τη δημιουργία της δομής αναζήτησης, το πρόγραμμα ζητά από τον χρήστη το μονοπάτι του αρχείου αναζήτησης και του αρχείου εξόδου των αποτελεσμάτων, καθώς και τον αλγόριθμο που θα χρησιμοποιηθεί. Στην περίπτωση του αλγορίθμου LSH για καμπύλες προσδιορίζεται και η μετρική για τον υπολογισμό της απόστασης μεταξύ των καμπύλων. Μετά την εκτέλεση του αλγορίθμου και την παραγωγή των αποτελεσμάτων, το πρόγραμμα ζητά από τον χρήστη αν θέλει να τερματίσει το πρόγραμμα ή αν θέλει να επαναλάβει την αναζήτηση για ένα διαφορετικό σύνολο / αρχείο αναζήτησης.

i) Για το LSH διανυσμάτων μπορούν να δίνονται οι εξής παράμετροι προαιρετικά στη γραμμή εντολών: ακέραιο πλήθος k των LSH συναρτήσεων hi που χρησιμοποιούνται για τον ορισμό των g, ο ακέραιος αριθμός L των πινάκων κατακερματισμού. Αν τα k, L δεν δίνονται, το πρόγραμμα χρησιμοποιεί default τιμές k=4, L=5.

ii) Για την τυχαία προβολή στον υπερκύβο, μπορούν να δίνονται οι εξής προαιρετικές παράμετροι στη γραμμή εντολών: η διάσταση στην οποία προβάλλονται τα διανύσματα *k (=d')*, το μέγιστο επιτρεπόμενο πλήθος υποψήφιων διανυσμάτων που θα ελεγχθούν *M* και το μέγιστο επιτρεπόμενο πλήθος κορυφών του υπερκύβου που θα ελεγχθούν (probes) Οι default τιμές ειναι: *k*=14, *M*=10, probes=2.

iii) Για το LSH καμπυλών μπορεί να δίνεται στη γραμμή εντολών η παράμετρος δ.

Τα αρχεία εισόδου και αναζήτησης θα μπορούν να δίνονται και μέσω παραμέτρων στη γραμμή εντολών, οπότε η εκτέλεση θα γίνεται μέσω της εντολής:

i)

```
$./search –i <input file> –q <query file> –k <int> -L <int> -M <int> -probes <int> -o <output file> -algorithm <LSH or Hypercube or Frechet> -metric <discrete or continuous | only for –algorithm Frechet> -delta <double>
```

Β.

1. Αρχείο εισόδου συνόλου δεδομένων Α.1

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-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>     // k of Hypercube, default: 3
number_of_probes: <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> -update <Mean Frechet or Mean Vector> –assignment <Classic or LSH or Hypercube or LSH_Frechet> -complete <optional> -silhouette <optional>
```

ΕΞΟΔΟΣ

Α.

Αρχείο κειμένου που περιλαμβάνει για κάθε χρονοσειρά του συνόλου αναζήτησης με τη χρήση των κατάλληλων ετικετών: α) το όνομα του προσεγγιστικά πλησιέστερου γείτονα που βρέθηκε από τον εκάστοτε αλγόριθμο και την απόστασή του από το *q*, β) το μέγιστο (από όλες τις χρονοσειρές του συνόλου αναζήτησης) κλάσμα προσέγγισης = Απόσταση προσεγγιστικά πλησιέστερου γείτονα / Απόσταση αληθινά πλησιέστερου γείτονα, γ) ο μέσος χρόνος εύρεσης του προσεγγιστικά πλησιέστερου γείτονα, δ) ο μέσος χρόνος εύρεσης του πραγματικά πλησιέστερου γείτονα

Το αρχείο εξόδου ακολουθεί υποχρεωτικά το παρακάτω πρότυπο :.

```
Query: itemJ
Algorithm: {LSH_Vector, Hypercube, LSH_Frechet_Continuous, LSH_Frechet_Discrete}
Approximate Nearest neighbor: itemY
True Nearest neighbor: itemX
distanceApproximate: <double>
distanceTrue: <double>
κ.ο.κ.

tApproximateAverage: <double>
tTrueAverage: <double>
MAF: <double> [Maximum Approximation Factor]
```

Β.

Ένα αρχείο κειμένου το οποίο περιλαμβάνει τις συστάδες των δεδομένων που παρήχθησαν από την εκάστοτε παραλλαγή του αλγορίθμου, τον χρόνο εκτέλεσης σε κάθε περίπτωση καθώς και, προαιρετικά, τον δείκτη εσωτερικής αξιολόγησης της συσταδοποίησης **Silhouette**. Το αρχείο εξόδου ακολουθεί υποχρεωτικά το παρακάτω πρότυπο:

```
Algorithm: AxUx
CLUSTER-1 {size: <int>, centroid: πίνακας με τις συντεταγμένες / σημεία του
centroid}
.     .     .     .     .     .     .
CLUSTER-k {size: <int>, centroid: πίνακας με τις συντεταγμένες / σημεία του
centroid}
clustering_time: <double> //in seconds

/* Optionally with command line parameter –silhouette */
Silhouette: [s1,...,si,...,sk, stotal]
/* si=average si in cluster i, stotal=average si in dataset */

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

CLUSTER-k {centroid, 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. Βαθμολογείται και η ποιότητα του κώδικα (π.χ. αποφυγή memory leaks). Η μεταγλώττιση του προγράμματος πρέπει να γίνεται με τη χρήση του εργαλείου make και την ύπαρξη κατάλληλου Makefile.
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 και ε) πλήρη στοιχεία των φοιτητών που το ανέπτυξαν.

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