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

1η Προγραμματιστική Εργασία
Κατακερματισμός και αναζήτηση για διανύσματα και πολυγωνικές καμπύλες στη C/C++

Η άσκηση θα υλοποιηθεί σε σύστημα Linux και θα υποβληθεί στις Εργασίες του e-class το αργότερο την Παρασκευή 25/10 στις 23.59.

Περιγραφή της εργασίας

Α. Υλοποιήστε τον αλγόριθμο LSH για διανύσματα στον *d*-διάστατο χώρο βάσει της μετρικής Μανχάταν (L1), καθώς και τον αλγόριθμο τυχαίας προβολής στον υπερκύβο για την ίδια μετρική. Το πρόγραμμα θα υλοποιηθεί έτσι ώστε λαμβάνοντας ως είσοδο διάνυσμα q, να επιστρέφει α) τον κοντινότερο γείτονα στο q και (bonus) τα διανύσματα εντός ακτίνας R από το q. (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).

Β. Υλοποιήστε τον αλγόριθμο LSH και τον αλγόριθμο τυχαίας προβολής για πολυγωνικές καμπύλες όπως παρουσιάστηκαν στο μάθημα για την εύρεση, μέσα σε σύνολο πολυγωνικών καμπύλων, του πλησιέστερου γείτονα καμπύλης q. Κάθε πολυγωνική καμπύλη ορίζεται ως μια ακολουθία σημείων στον ευκλείδειο χώρο $\Re^2$, με διαφορετικό πλήθος σημείων. Το πρόγραμμα θα υποστηρίζει την συνάρτηση απόστασης μεταξύ καμπυλών Dynamic Time warping.

i) Το LSH καμπυλών στις διαλέξεις απεικονίζει καμπύλες σε «καμπύλες πλέγματος» (grid curves), οι οποίες θα αναπαριστώνται ως διανύσματα. Υλοποιήστε την αποθήκευση και αναζήτηση στις καμπύλες πλέγματος α) με τον αλγόριθμο LSH για μετρική Manhattan και β) με τον αλγόριθμο τυχαίας προβολής στον υπερκύβο για την ίδια μετρική.
ii) Η τυχαία προβολή ανάγει επίσης τις καμπύλες σε διανύσματα και εκτελεί αναζήτηση διανυσμάτων. Αυτά προκύπτουν από τα πιθανά ταιριάσματα (traversals) των δεδομένων με καμπύλες μήκους όσο κάθε δυνατό μήκος της καμπύλης ερωτήματος (query). Η αναζήτηση στα διανύσματα που αναπαριστούν τις καμπύλες θα γίνεται επίσης α) με τον αλγόριθμο LSH για μετρική Manhattan και β) με τον αλγόριθμο τυχαίας προβολής στον υπερκύβο για την ίδια μετρική.

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

ΕΙΣΟΔΟΣ

Α.

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 Xij διανύσματος 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. Ο θετικός double R δίνεται στην πρώτη γραμμή. Όταν δίνεται R=0 το πρόγραμμα βρίσκει μόνο τον κοντινότερο γείτονα των διανυσμάτων στο σύνολο δεδομένων.

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. Το αρχείο αναζήτησης έχει την ακόλουθη μορφή:

```
Radius: <double>                    //bonus
item_idS1   X11   X12   ...   X1d
.     .     .     ...   .
item_idSQ   XQ1   XQ2   ...   XQd
```

Τα ονόματα των αντικειμένων στο σύνολο αναζήτησης item_idSj (1<=j<=Q) μπορούν να είναι μοναδικοί ακέραιοι ή συμβολοσειρές.

Για το LSH, 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.

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. Οι default τιμές ειναι: *k*=3, *M*=10, probes=2.

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

```
$./lsh –d <input file> –q <query file> –k <int> -L <int> -o <output file>
$./cube –d <input file> –q <query file> –k <int> -M <int> -probes <int> -o
<output file>
```

Β.

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

```
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.

2) Αρχείο κειμένου που περιλαμβάνει το σύνολο των καμπυλών για τις οποίες αναζητούμε τον πλησιέστερο γείτονα: πρόκειται για το σύνολο αναζήτησης, που περιέχει τουλάχιστον μία καμπύλη (αρχείο αναζήτησης).

Το πρόγραμμα αρχικά ζητά από τον χρήστη το μονοπάτι του αρχείου dataset, τον αλγόριθμο που επιλέγεται για την αναζήτηση καμπυλών (περίπτωση i ή ii) και τον αλγόριθμο αναζήτησης των διανυσμάτων που προκύπτουν από τις καμπύλες (LSH ή Υπερκύβος). Μετά τη δημιουργία της δομής αναζήτησης, το πρόγραμμα ζητά από τον χρήστη το μονοπάτι του αρχείου αναζήτησης και του αρχείου εξόδου των αποτελεσμάτων. 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. Το αρχείο έχει την ακόλουθη μορφή για προβλήματα σε διανυσματικό χώρο, αντιστοίχως με τη μορφή του αρχείου εισόδου:

```
curve_idS1      mS1      (xS11,yS11)  (xS12,yS12)  ...   (xS1mS1,yS1mS1)
    .                        .            .       ...          .
curve_idSN      mSN      (xSN1,ySN1)  (xSN2,ySN2)  ...   (xSNmSN,ySNmSN)
```

Τα ονόματα των καμπύλων στο σύνολο αναζήτησης curve_idSj (1<=*j*<=*Q*) μπορούν να είναι μοναδικοί ακέραιοι ή συμβολοσειρές.

Ως optional παράμετρος μπορεί να δίνεται στη γραμμή εντολών:

- Για το LSH καμπυλών, ο ακέραιος *L* των καμπυλών πλέγματος ανά καμπύλη εισόδου, ενώ αν δεν δίνεται, χρησιμοποιείται default τιμή *L*=4.
- Για τον αλγόριθμο τυχαίας προβολής μπορεί να δίνεται στη γραμμή εντολών ο παράγων προσέγγισης ε, ενώ αν δεν δίνεται, χρησιμοποιείται default τιμή ε=0.5.

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

```
$./curve –d <input file> –q <query file> –k <int> -L <int> -o <output file> -a
{LSH, RandomProjection} –h {LSH, Hypercube}
```

ΕΞΟΔΟΣ

Α.

Αρχείο κειμένου που περιλαμβάνει για κάθε αντικείμενο του συνόλου αναζήτησης με την χρήση των κατάλληλων ετικετών: α) το όνομα του προσεγγιστικά κοντινότερου γείτονα που βρέθηκε και την απόστασή του από το *q*, β) την απόσταση του *q* από τον αληθινά πλησιέστερο γείτονα (μέσω εξαντλητικής αναζήτησης), γ) τον χρόνο εύρεσης του (α) και δ) τον χρόνο εύρεσης του (γ) καθώς και (bonus) τα ονόματα των γειτόνων ακτίνας *R*. Το αρχείο εξόδου ακολουθεί υποχρεωτικά το εξής πρότυπο:

```
Query: itemJ
Nearest neighbor: itemY
distanceLSH: <double>
distanceTrue: <double>
tLSH: <double>
tTrue: <double>
R-near neighbors:       //bonus
itemJ                   //bonus
itemK                   //bonus
. . .
itemW                   //bonus
```

και ούτω καθεξής.

Β.

Αρχείο κειμένου που περιλαμβάνει για κάθε καμπύλη του συνόλου αναζήτησης με την χρήση των κατάλληλων ετικετών: α) το όνομα του πλησιέστερου γείτονα που βρέθηκε από τον LSH / αλγόριθμο τυχαίας προβολής, και την απόστασή του από το *q*. Το αρχείο εξόδου ακολουθεί υποχρεωτικά το παρακάτω πρότυπο:

```
Query: curveJ
Method: {LSH, Projection}
HashFunction: {LSH, Hypercube}
Found Nearest neighbor: curveY
True Nearest neighbor: curveX
distanceFound: <double>
distanceTrue: <double>
```

Συγκρίνετε τα αποτελέσματα των 4 παραλλαγών του αλγορίθμου [LSH for curves / LSH L1, LSH for curves / Hypercube, Random Projection / LSH L1, RandomProjection / Hypercube] ως προς α) το μέγιστο (από όλα τα αντικείμενα του συνόλου αναζήτησης) κλάσμα προσέγγισης = Απόσταση προσεγγιστικά κοντινότερου γείτονα / Απόσταση αληθινά κοντινότερου γείτονα, β) το μέσο χρόνο εύρεσης του προσεγγιστικά κοντινότερου γείτονα. Σχολιάστε τα αποτελέσματα στην τεκμηρίωση του προγράμματος.

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).