Κ23γ: Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα
Χειμερινό εξάμηνο 2023-24

2η Προγραμματιστική Εργασία
Γραφοθεωρητική αναζήτηση πλησιέστερων γειτόνων στη C/C++

Η άσκηση θα υλοποιηθεί σε σύστημα Linux και θα υποβληθεί στις Εργασίες του e-class το αργότερο την Παρασκευή 1/12 στις 23.59.

Περιγραφή της εργασίας

Α. Υλοποιήστε τον αλγόριθμο αναζήτησης πλησιέστερων γειτόνων GNNS για διανύσματα στον *d*-διάστατο χώρο βάσει της ευκλείδειας μετρικής (L2), το οποίο χρησιμοποιεί τον αλγόριθμο LSH για την κατασκευή του γράφου k-πλησιέστερων γειτόνων (k-NN) κατά τη διαδικασία κατασκευής του ευρετηρίου. Το πρόγραμμα θα υλοποιηθεί έτσι ώστε λαμβάνοντας ως είσοδο διάνυσμα q και ακέραιους N, να επιστρέφει προσεγγιστικά: α) τον πλησιέστερο γείτονα στο q, β) τους N πλησιέστερους γείτονες στο 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 αναζήτησης πλησιέστερων γειτόνων Search-on-Graph για διανύσματα στον *d*-διάστατο χώρο βάσει της ευκλείδειας μετρικής (L2), με τη χρήση μονότονου τυχαιοποιημένου γράφου γειτνίασης ως ευρετηρίου (MRNG). Το πρόγραμμα θα υλοποιηθεί έτσι ώστε λαμβάνοντας ως είσοδο διάνυσμα q και ακέραιους N, να επιστρέφει προσεγγιστικά: α) τον πλησιέστερο γείτονα στο q, β) τους N πλησιέστερους γείτονες στο 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.

Γ. Συγκρίνετε τις επιδόσεις των αλγορίθμων που υλοποιήθηκαν στα ερωτήματα Α. και Β. μεταξύ τους και με τους αλγόριθμους LSH και Hypercube που υλοποιήθηκαν στην 1η εργασία ως προς τον χρόνο αναζήτησης και ως προς το μέγιστο (από όλες τα διανύσματα του συνόλου αναζήτησης) κλάσμα προσέγγισης (= Απόσταση προσεγγιστικά πλησιέστερου γείτονα / Απόσταση αληθινά πλησιέστερου γείτονα) για διαφορετικές τιμές των παραμέτρων των αλγόριθμων. Η σύγκριση θα πρέπει να τεκμηριωθεί αναλυτικά και να περιλαμβάνει συζήτηση των αποτελεσμάτων.

ΕΙΣΟΔΟΣ

Α. και Β.

1) Ένα binary αρχείο `input.dat` για την είσοδο του συνόλου δεδομένων (dataset) με την κάτωθι μορφή:

```
[offset] [type]          [value]          [description]
0000     32 bit integer  0x00000803(2051) magic number
0004     32 bit integer  60000            number of images
0008     32 bit integer  28               number of rows
0012     32 bit integer  28               number of columns
0016     unsigned byte   ??               pixel
0017     unsigned byte   ??               pixel
........
xxxx     unsigned byte   ??               pixel
```

Το αρχείο αυτό αντιστοιχεί στο dataset MNIST που περιέχει εικόνες χειρόγραφων αριθμητικών ψηφίων http://yann.lecun.com/exdb/mnist/. Οι εικόνες είναι 28*28 pixels, καθένα από τα οποία παίρνει ακέραια τιμή από το 0 έως το 255. Το διάνυσμα που αντιστοιχεί σε κάθε εικόνα προκύπτει από τη συνένωση των γραμμών της και έχει διάσταση 784.

2) Δυαδικό αρχείο `query.dat` που περιλαμβάνει το σύνολο αναζήτησης και περιέχει τουλάχιστον μία εικόνα MNIST με την ίδια μορφή.

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 τον GNNS, μπορούν να δίνονται οι εξής παράμετροι προαιρετικά στη γραμμή εντολών: πλήθος k των πλησιέστερων γειτόνων στον γράφο k-NN, ακέραια παράμετρος E των επεκτάσεων, ακέραιος αριθμός R των τυχαίων επανεκκινήσεων και ακέραιος αριθμός N των πλησιέστερων γειτόνων. Αν οι παράμετροι δεν δίνονται, το πρόγραμμα χρησιμοποιεί default τιμές k=50, E=30, R=1 , N=1.

Για τον MRNG, μπορούν να δίνονται οι εξής προαιρετικές παράμετροι στη γραμμή εντολών: πλήθος l της δεξαμενής υποψηφίων και ακέραιος αριθμός N των πλησιέστερων γειτόνων. Οι default τιμές ειναι: l=20, N=1 (l >= N).

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

```
$./graph_search -d <input file> -q <query file> -k <int> -E <int> -R <int> -N <int> -l <int, only for Search-on-Graph> -m <1 for GNNS, 2 for MRNG> -o <output file>
```

Γ. Δεν υπάρχει συγκεκριμένη είσοδος. Θα πρέπει να επαναληφθούν εκτελέσεις των ερωτημάτων Α. και Β. της 2ης εργασίας και του ερωτήματος Α. της 1ης εργασίας για διαφορετικές τιμές των παραμέτρων.

ΕΞΟΔΟΣ

Α και Β. 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 μέσο χρόνο εύρεσης των (α), δ) τον μέσο χρόνο εύρεσης των (β) και ε) το μέγιστο κλάσμα προσέγγισης. Το αρχείο εξόδου ακολουθεί υποχρεωτικά το εξής πρότυπο:

```
GNNS Results or MRNG Results
Query: image_number_in_query_set
Nearest neighbor-1: image_number_in_data_set
distanceApproximate: <double>
distanceTrue: <double>
...
```

```
Nearest neighbor-N: image_number_in_data_set
distanceApproximate: <double>
distanceTrue: <double>
```
και ούτω καθεξής.

```
tAverageApproximate: <double>
tAverageTrue: <double>
MAF: <double> [Maximum Approximation Factor]
```

Το image_number προσδιορίζεται βάσει της σειράς με την οποία εμφανίζεται η εικόνα στα εκάστοτε αρχεία.

Γ. Αναφορά η οποία περιλαμβάνει τεκμηρίωση της εκτέλεσης των πειραμάτων σύγκρισης και συζήτηση των αποτελεσμάτων.

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).