Ονοματεπώνυμο: Νικόλας Νικολακόπουλος

Α.Μ.: 1115201800133

Ονοματεπώνυμο: Άγγελος Τσούκο

Α.Μ.: 1115201800202

Στο κατάλογο “files” υπάρχουν όλα τα αρχεία εισόδου, τα query files και κάποια output files.

Στον κατάλογο “src” υπάρχει ο κύριος κώδικας για την υλοποίηση της εργασίας. Εντός αυτού του καταλόγου υπάρχουν:

- τα αρχεία inputHandle περιέχει κάποιες συναρτήσεις για την ανάγνωση των ορισμάτων από την γραμμή εντολών και το πέρασμα τους σε δομές(struct userInput / clusterInput). Επίσης περιέχει συνάρτηση για την ανάγνωση του config file και μια συνάρτηση που διαβάζει όλα τα σημεία από ένα αρχείο και τα περνάει σε μια δομή(vector που περιέχει λίστες με στοιχεία).
- τα αρχεία operations που έχουν διάφορες συναρτήσεις κυρίως για μαθηματικές πράξεις πχ euclidean distance , hamming distance , κτλ.
- Τον κατάλογο LSH
- Τον κατάλογο hypercube
- Τον κατάλογο Clustering

Στον κατάλογο LSH:

- Τα αρχεία hash περιέχουν τον κώδικα σχετικά με τις δομές και την υλοποίηση της μεθόδου lsh:

  ° Τα σημεία θεωρούνται ως μια κλάση η οποία και ορίζεται έχοντας ένα πεδίο vector<double>* για την αποθήκευση των συντεταγμένων και ένα πεδίο string για την αποθήκευση του μοναδικού id του σημείου.

° Το hashTable είναι μια κλάση που έχει ένα vector με λίστες σε σημεία και ένα unordered_map για την αποθήκευση των ID(p) = (r1*h1(p)+...+ri*hi(p)) mod M , ώστε να περιορίσουμε τα σημεία που πρέπει να ελεγχθούν μέσα στο bucket του hashtable.

° Η κλάση LSH περιέχει μια δομή με τις παραμέτρους που έχει δώσει ο χρήστης, έναν πίνακα με hash tables και έναν δείκτη σε αντικείμενο κλάσης HashFunctions (θα αναφερθεί αμέσως τι είναι).

• Τα αρχεία hashf περιέχουν τον κώδικα σχετικά με την υλοποίηση των hash functions. Η Κλάση HashFunctions περιέχει ξεχωριστά όλες τις παραμέτρους που χρειάζονται για την λειτουργία της (k, L, …), καθώς και όλα τα τυχαία vectors v και τυχαίους αριθμούς t για κάθε συνάρτηση h και τους τυχαίους αριθμούς r για κάθε συνάρτηση g.

Αντίστοιχα αρχεία υπάρχουν και στον κατάλογο hypercube, μόνο που αντί για πολλά hashTables, στην μέθοδο randomized hypercube projection, χρησιμοποιούμε μόνο 1 hashtable(το hypercube). Ωστόσο, για τη χρήση των hash functions, υπάρχει μια κλάση hashF , η οποία αντιστοιχεί σε κάθε f_i(h_i). Υπενθυμίζουμε ότι η f_i παίρνει μια τιμή που προκύπτει από τη hashfunction h_i (για ένα σημείο p) και επιστρέφει με τυχαίο τρόπο, 0 ή 1. Περιέχει το τυχαίο v array και τον τυχαίο αριθμό t για την h_i hashfunction που αντιστοιχεί στην f_i συνάρτηση, αλλά και ένα map στο οποίο αποθηκεύονται οι τιμές της f_i(h_i(p)), διότι ο τρόπος που παράγονται είναι τυχαίος και για ίδιο h_i(p) θέλουμε να παραχθούν ίδιες τιμές. Η κλάση hashFs περιέχει ένα vector με πολλές τέτοιες hashF, όσες και η διάσταση που δίνεται από τον χρήστη.

Για τον bruteforce αλγόριθμο επιστρέφουμε ένα:

```
std::multimap<double, std::pair<Point *, std::chrono::duration<double>> >
```

το οποίο αντιστοιχεί σε όλα τα K σημεία που είναι κοντά στο σημείο που δίνεται σαν input. Το pair αντιστοιχεί στο σημείο και τον χρόνο που έκανε να βρεθεί, ενώ η double τιμή αντιστοιχεί στην απόσταση του σημείου που βρέθηκε με το query point. Προφανώς ο brute force αλγόριθμος έχει εξαντλητικό χαρακτήρα.

Ομοίως, στον KNN επιστρέφουμε τέτοια δομή. Αξιοσημείωτο είναι ότι χρησιμοποιούμε ένα explored_set με τα σημεία που έχουμε επισκεφτεί, ώστε στα επόμενα hash tables να μην ελέγξουμε τα ίδια σημεία.

Στο ranged search απλά η αναζήτηση γίνεται με βάση την ακτίνα με το ίδιο σκεπτικό με το KNN. Ωστόσο, χρησιμοποιώ priority queue με βάση το item id για να υπάρχει μια αύξουσα σειρά και να είναι πιο διακριτά τα αποτελέσματα.

Εδώ πρέπει να σημειωθεί ότι αυτές οι συναρτήσεις μετασχηματίστηκαν σε KNN_amplified kai rangedSearchAmplified για τις ανάγκες του reversed assignment στο clustering. Οι αλλαγές που έγιναν έχουν να κάνουν με την εισαγωγή του barrier σε κάθε bucket και την μεταφορά των σημείων στο τέλος του bucket εάν έμπαιναν στο cluster, ώστε να μην ελεγχθούν πάλι σε επόμενη επανάληψη. Το ίδιο έγινε και στο hypercube.

Η συνάρτηση QueryFile στην LSH κλάση απλά καλεί τις συναρτήσεις KNN, ranged search και εκτυπώνει τις λύσεις στο output file.

Με τον ίδιο τρόπο υλοποιείται και το hypercube με την διαφορά να έγκειται στο hashfunction και την αποθήκευση των δεδομένων σε 1 μόνο hashtable. Στις συναρτήσεις αναζήτησης κοντινών σημείων (KNN, ranged search) πρέπει να παραχθούν τα keys που αντιστοιχούν σε κοντινές κορυφές του υπερκύβου. Αυτό γίνεται με το να υπολογίσουμε τα κλειδιά με hamming distance = 1 από ένα αρχικό κόμβο και να τα προσθέσουμε σε μια ουρά. Έπειτα κάνουμε pop από την ουρά και συνεχίουμε πιο βαθιά με το ίδιο σκεπτικό.

Στον κατάλογο clustering:

- Τα αρχεία “assignment” περιέχουν τη συνάρτηση “lloyd” για την ανάθεση με τη μέθοδο του Lloyd καθώς και τη συνάρτηση “assignment”, η οποία επιλέγει τον κατάλληλο αλγόριθμο ανάθεσης, ανάλογα με τη μέθοδο που έχει δοθεί στη γραμμή εντολών.
- Τα αρχεία “initialization” περιέχουν συναρτήσεις για την αρχικοποίση των clusters, καθώς και για την ενημέρωσή τους. Επίσης, έχουν συναρτήσεις για τη διαγραφή των δομών των clusters και των σημείων του dataset.
- Τα αρχεία “reverse” περιέχουν κώδικα για την ανάθεση των σημείων σε clusters με τις μεθόδους “Range Search LSH” και “Range Search Hypercube”.

Για το reverse assignment ακολουθείται πιστά ο αλγόριθμος που υπάρχει στις διαφάνειες. Κατά το assignment μπορεί να υπάρξουν σημεία που δεν έχουν ανατεθεί πουθενά ή υπάρχουν σε 2 ή περισσότερα clusters και υπάρχουν ακόμα τα barriers στην δομή που αποθηκεύονται τα σημεία μέσα στα buckets. Για αυτό καλώ τις συναρτήσεις unassignedPoints για να επιλύσουν αυτά τα προβλήματα με bruteforce μέθοδο.

Οδηγίες μεταγλώττισης

- Για τη διαγραφή όλων των εκτελέσιμων αρχείων γράφουμε:

  make clean

- Για τη μεταγλώττιση του “lsh.cpp” και των υπολοίπων απαραίτητων αρχείων γράφουμε:

make lsh

- Για τη μεταγλώττιση του “hypercube.cpp” και των υπολοίπων απαραίτητων αρχείων γράφουμε:

make cube

- Για τη μεταγλώττιση του “cluster.cpp” και των υπολοίπων απαραίτητων αρχείων γράφουμε:

make cluster

Πρωτού ξαναμεταγλωττίσουμε ένα πρόγραμμα πρέπει οπωσδήποτε να κάνουμε “make clean”.

Οδηγίες χρήσης

Για τα input file, query file, configuration file και output file πρέπει να δίνεται και το μονοπάτι από τον τρέχοντα κατάλογο.

Οι μέθοδοι για το clustering πρέπει να δίνονται με συγκεκριμένο τρόπο:

- “-m Classic” ή “-m classic” για το Lloyd’s
- “-m LSH” ή “-m lsh” για το Range Search LSH
- “-m Hypercube” ή “-m hypercube” για το Range Search Hypercube