Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα 2021-2022

Εργασία 1

Αναζήτηση και συσταδοποίηση διανυσμάτων σε C/C++

Περιεχόμενα

1. Περιγραφή

lsh.cpp
Το πρόγραμμα δέχεται ένα dataset από διανύσματα μέσω ενός αρχείου και άλλα queries διανύσματα. Αν δεν τα πάρει από τη γραμμή εντολών τότε τα ζητάει μέσω του input.
Αφου φτιάξει δύο vectors κλάσεων με όλα τα διανύσματα κάνει μια lsh λειτουργία κατακερματισμού και εισχωρεί by reference τα διανύσματα σε αυτά με τη g function του lsh, οπότε δημιουργεί πολλούς πίνακες κατακερματισμού και μέσω αυτών δωσμένου ενώς query διανύσματος μπορεί να βρει μέσα απο τα κατάλληλα buckets τους N κοντινότερους γείτονες , προσέχει να μην επιλέξει τους ίδιους γείτονες δύο φορές επειδή σώζει τα ονόματα σε έναν πίνακα και αν δε βρει το αριθμό των κοινών γειτόνων που επιθυμεί ξαναψάχνει αλλά αυτή τη φορά κάθε στοιχείο του bucket ανεξάρτητα από το ID του. Σώζει δύο πίνακες με ονόματα γειτόνων και τις αποστάσεις έτσι ώστε να διαλέξει τους N που είναι οι πιο κοντά.
Επίσης κάνει και range search , στο οποίο ξεκινάει από το bucket με τα κοντινότερα και συνεχίζει να ψάχνει για αποστάσεις μικρότερες του R μέχρι να φτάσει σε έναν κουβά που δεν έχει κανένα στοιχείο που να μας βολεύει. Έτσι ώστε να μαζέψουμε τα περισσότερα στοιχεία σε αυτό το range.

cube.cpp
Το πρόγραμμα δέχεται ένα dataset από διανύσματα μέσω ενός αρχείου και άλλα queries διανύσματα, με ίδιο τρόπο όπως και το lsh.cpp.

Clustering.cpp
Το επόμενο κομμάτι είναι η συσταδοποίηση στην οποία μπορείς να διαλέξεις μεταξύ δύο μεθόδων, lsh και loyd. Το πρόγραμμα αρχίζει παρόμοια με το lsh.cpp διαβάζει τα αρχεία και αρχικοποιεί με το kmeans++ τα κεντροϊδή.
Στη μέθοδο loyd πέρνει ένα ένα τα διανύσματα από τον vector κλασεων και βρίσκει την απόσταση κάθε σημείο με όλα τα κεντροϊδή και η απόσταση που είναι η μικρότερη κερδίζει και μπαίνει το όνομα στον κατάλληλο πίνακα ονομάτων συσταδας για να ξέρω ποιά σημεία είναι σε ποιά συστάδα. Ύστερα για κάθε κεντροιδές παίρνω τα σημεία του και επειδή τα ονόματα τους είναι αριθμοί και με τη σειρά που τα έβαλα στον πίνακα με μια μετατροπή σε αριθμό μπορώ να βρω τα διανύσματα τους οπότε κάνω τον μέσο όρο όλων και βρίσκω ένα καινούργιο διάνυσμα που θα αντικαταστήσει το παλίο κεντροϊδές. Αυτό γίνεται για όλα τα κεντροϊδή και πολλές φορές μέχρι να μην αλλάζουν τα διανύσματα των κεντροϊδών.
Στη lsh μέθοδο κάνει επαναλήψεις ανα 10 και παίρνει τα διανυσματα που είναι στο range [χ,χ+10] μέχρι να βρεί κοινούς αριθμούς μεταξύ των κεντροϊδών

οπότε συνεχίζει να βρίσκει ranges αλλά τώρα αυτά που βρίσκει τα βάζει σε ένα vector<string> και παίρνει μια μια τιμή και την επεξεργάζεται σαν τον loyd. Το range search δεν επιστρέφει όλα τα διανύσματα , δε ξέρω γιατί , οπότε κάποια λείπουν.
Βρίσκει και τη σιλουέττα μέσα απο πολλές επαναλήψεις.
Και το lsh και το clustering τυπώνουν τα αποτελέσματα τους σε ένα output file του οποίου το path δίνεται από τον χρήστη και άμα δεν υπάρχει δημιουργεί ένα καινούργιο.

ghashfunction.cpp/h
readfile
Αυτή η συνάρτη παίρνει σαν όρισμα τον vector<datasetarray> by reference και τον γεμίζει με τα διανύσματα που παίρνει απο το αρχείο με path αυτό που του δίνουμε σαν όρισμα. Η συνάρτηση επιστρέφει δύο ints , το μέγεθος των διανυσμάτων και πόσα ειναι τα διανύσματα.

Ghashfunction
Αυτή πάιρνει τα προ υπολογισμένα v,r,t από την κλάση του πινακα κατακερματισμού που του δίνουμε για να φτιάξουμε τη g σύμφωνα με τη θεωρία του lsh. Προσέχει το g μην είναι αρνητικό και επιστρέφει δύο αριθμούς που είναι το g και το ID.

Innerproduct
εσωτερικός πολλαπλασιασμός διανυσμάτων

calcEuclideanDist
ευκλείδια απόσταση

rangesearch
Όπως είπα παραπάνω αυτή ειναι που βρίσκει όλους τους γείτονες σε απόσταση R από ένα δωσμένο query. Θυμάται μέσω του namebank να μην επιλέξει δύο φορές ίδια διανύσματα και μέσω boolean flags βρίσκω πότε το πρόγραμμα δε βρίσκει παραπάνω γείτονες οπότε σταματάει και επιστρέφει τον πίνακα με τα ονόματα που έχει βρεί.

Datasetarray.h/cpp

Είναι η κλάση όπου σώζουμε αρχικά τα διανύσματα. Έχει το όνομα, το μέγεθος και το ίδιο το διάνυσμα σε vector<float>.

Class.h/cpp

Εδώ υπάρχει η κλάση του πίνακα κατακερματισμού. Περιέγχει τους τυχαίους πινακες και αριθμο p,v,r αντιστοιχα που αρχικοποιούνται για κάθε πίνακα κατακερματισμού. Τον αριθμό των κουβάδων και μία λίστα απο hashlist structs. Τα hashlist έχει το ID και το datasetarray με το διάνυσμα που θέλουμε, by reference. Η printhashtable μπορεί να χρησιμοποιηθεί για να τυπωθούν στο terminal τα περιεγχόμενα κάθε bucket.

Το nordist φτιάχνει ένα τυχαίο διανυσμα από τη κανονική κατανομή.

Clusteringfunctions.h/cpp

readconfig
Διαβάζει το conf file για τη συσταδοποίηση. Το διαβάζει σαν strings. Βρίσκει που είναι το space και ξέρει ότι μέτα από το κενό έχει τους αριθμους που θέλουμε σε κάθε γραμμή.

Kmeansplusplus
Παίρνει ένα εντελώς τυχαίο διάνυσμα από το datasetarray και το κάνει κεντροϊδές. Ύστερα κάνει τη τεχνική kmeans++ φτιάχνει τα Dp^2 φτιάνει τον πίνακα P[10000] και διαλέγει έναν άλλο αριθμό τυχαία οπότε πάιρνει το διάνυσμα που αντιστοιχεί στο datasetarray. Δεν επιστρέφει τίποτα επειδή τα centroids παιρνάνε σαν refernece.

cubehash.h/cpp

Η cubehash είναι μία κλάση που περιέχει τις συναρτήσεις ffunction και hashvalue. Η ffunction επιστρέφει για το σημείο που της δίνεται σαν όρισμα την τιμή 0/1 που του αντιστοιχεί και επίσης κρατάει τις τιμές που έχει ήδη αντιστοιχίσει στο map valuemapping, με σκοπό να αντιστοιχεί πάντα τις ίδιες τιμές στα αποτελέσματα της h. Η hashvalue καλεί την ffunction k φορές, μία για κάθε διάσταση. Δημιουργεί αρχικά έναν πίνακα από 0 και 1 και ύστερα τα ενώνει σε ένα string από 0/1, το οποίο αντιστοιχεί σε μία από τις κορυφές του κύβου. Καλούμε την hashvalue μια φορά για κάθε σημείο, για να γεμίσουμε το hashtable του κύβου.

2.Μεταγλώτιση

Υπάρχει makefile οπότε για να φτιάξετε το πρόγραμμα για το lsh αρκεί να πατήσετε
>make lsh

Για να φτιάξετε το εκτελέσιμο για το clustering
>make clustering

Για να φτιάξετε το εκτελέσιμο για το cube
>make cube

3.Χρήση προγράμματος

Υπάρχει makefile οπότε αρκεί να δώσετε τις εντολές make lsh, make clustering ή make cube.

Τρέχει με τα ορίσματα της εκφώνησης ή τα ζητάει από τη γραμμή εντολων και τρέχει τα υπόλοιπα με τις default τιμές.
Για να δώσει καλά αποτελέσματα η lsh πρέπει να δώσετε μεγάλο L και κ.
Οι μέθοδοι στο clustering είναι η “lsh” και η “loyd”

4.Στοιχεία φοιτητών

Χαράλαμπος Βασιλάκης
A.M: 1115201500015
Αχολήθηκε με το lsh και το clustering

Ειρήνη Αράπκουλε
A.M.: 1115201500011
Ασχολήθηκε με το hypercube, Makefile και βοήθησε στο lsh συγκεκριμένα έκανε τις συναρτήσεις hFunction,innerproduct,calcEuclideanDist και έβαλε τις τυχαίες μεταβλητές p, v, r στην κλάση hashclass.