Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα 2021-2022

Εργασία 2

Αναζήτηση και συσταδοποίηση διανυσμάτων σε C/C++

Περιεχόμενα

1. Περιγραφή

Nearest Neighbours

LSH και Clustering methods **Loyd's** και **LSH** με **Mean_Vectors**. Εφτιαξα πολλές helper συναρτήσεις και μάζεψα πολύ τον κώδικα. Έφτιαξα το modulo να δουλεύει με τον σωστό τρόπο. Στο lsh γενικά και σε όλες τις άλλες μεθόδους, κάποιες φορές όταν τρέχει δε γίνεται σωστά το hashing και μένει άδειος ο πίνακας, αυτό συμβαίνει μόνο λίγες φορές και αν ξανατρέξει, το πρόβλημα δε θα υπάρχει, δε ξέρω τι το προκαλεί δεν είχα χρόνο να το ψάξω.

LSH_Frechet. Έφτιαξα μια νέα **xreadfile** η οποία διαβάζει τα inputs και σωζει τις τιμές σε μια κλάση. Αυτή διαφέρει στο ότι σώζει και τις τιμές X (τιμές χρόνου 1,2,3,4,5…..) σε έναν μονοδιάστατο πίνακα όπου οι περιττές τιμές είναι οι X και οι άρτιες οι Y , οπότε και ο πίνακας είναι διπλάσιος.Επίσης το xreadfile επιστρέφει την μεγαλύτερη τιμή του αρχείου για το **padding** και το μέγεθος του μεγαλύτερου διανύσματος. Για το hashing καλέι μια συνάρτηση **makegrid** η οποία μέσω ενός τύπου υπολογίζει που θα γινει **snapping** , το δέλτα το υπολογίζω εγώ αλλά επίσης μπορεί να δωθεί και από τη γραμμή εντολων, το **M** για το **padding** το παίρνω από την **xreadfile.** Αφού γίνει το snapping, καλείτε η **ghashfunction** ,που εκτελεί τη διαδικασία του lsh όπως στη προηγούμενη εργασία για να βρεί σε ποιό bucket ανοίκει στον πίνακα κατακερματισμού.
Υπολογίζω εδώ όλες τις αποστάσεις με τη μία και τις σώζω σε έναν πίνακα για να τις έχω έτοιμες όταν κάνω τις συγκρίσεις μέσα στον πίνακα κατακερματισμου, η συνάρτηση του **frechetdistance** είναι πολύ αργή και γι αυτό σώζεται πολυς χρόνος αν κάνω από μία φορά κάθε σύγκριση, οι συναρτήσεις για να τα κάνω όποτε το πρόγραμμα το ζητάει και να μην τα σώσω σε έναν πίνακα υπάρχουν και μπορώ να αλλάξω εύκολα το πρόγραμμα να λείτουργεί έτσι.Για να βρω το **frechetdistance** φτιάχνω έναν δισδιάστατο πίνακα και τον φτιάχνω όπως υπέδειξε η θεωρία , είτε σώζω την ευκλείδια απόσταση των δύο σημείων του δυσδιαστατου χώρου ή παίρνω τη μέγιστη τιμή των προηγούμενων. Και όταν φτιάξω όλον τον πίνακα ξεκινάω από τη κορυφή του και πηγαίνω αριστερά,κάτω ή διαγώνια ακολουθώντας τη μικρότερη διαδρομή και εντομεταξύ κρατάω τον μεγαλύτερο αριθμό που βρίσκω και αυτός είναι η απόσταση.
Μετά παίρνω κάθε **query** curve array , βρίσκω τη θέση του στο hashtable με το snapping και ghashfunction. Όταν βρώ το bucket βρίσκω τους **nearest neighbours** πρώτα ψάχνωντας τα IDs και ελέγχωντας αν μετά το **snapping** βγαίνουν τα ίδια διανύσματα. Μετά αν δε βρώ όσα θέλω, ψάχνω όλα τα διανύσματα του bucket και εδώ χρησιμοποιώ τον πινακα των αποστάσεων που υπολόγισα προηγουμένως. Η συνάρτηση **nearestneighbours** είναι ίδια με τη προηγούμενη άσκηση. Στη συνέχεια τα τυπώνω στο output αρχείο και βρίσκω και τα **in range αντικείμενα** όπως στη προηγούμενη άσκηση χρησιμοποιώντας τον πίνακα με τις frechet αποστάσεις.
Η lsh_frechet τρέχει αργά επειδή αργουν οι υπολογισμοί των frechet αποστάσεων.

Η clustering καλεί την readfile αν -update=Mean_Vector, xreadfile αν -update = Mean_Frechet. Η **kmeansplusplus** είναι ίδια με τη προηγούμενη άσκηση. Τα βήματα για την frechet μέθοδο είναι παρόμοια με τη πρώτη εργασία αλλά, κάνω **frechetdistance** όπου έκανα ευκλείδια απόσταση και για να βρώ τα νέα κεντροειδή χρησιμοποιώ την **meancurve.** Ξεκινάει απο τη **newmeancurve** καλεί για κάθε κεντροειδές καλεί μια συνάρτηση **createbtree** η οποία επιστρέφει ένα καινούργιο float πινακα (κεντροειδές). Η createbtee

είναι ανδρομική συνάρτηση , καλείτε από τον εαυτό της δύο φορές , μία για το δεξιό και μια για το αριστερό κλαδί του “δέντρου”. Όταν φτάσει στο φύλλο , που ξέρω ότι ο κόμβος είναι φύλλο επειδή κρατάω το βάθος του δέντρου σε κάθε αναδρομική κλήση, παίρνω το πρώτο όνομα που είναι αναθετημένο στη συστάδα που φτιάχνω, βρίσκω τη θέση του στον πίνακα που έχω σώσει τα curves με τη συνάρτηση **finddatapos** και έτσι παίρνω το διάνυσμα του και το επιστρέφω, επίσης διαγράφω το όνομα από τον πίνακα για να μην πάρω το ίδιο όνομα απο τον πινακα στο επόμενο φύλλο. Σε κάθε κόμβο του “δέντρου” όταν παιρνω και το δεξι και αριστερό διάνυσμα καλώ την **meancurve** η οποία φτιάχνει τον **frechetdistance** πίνακα και μέσω αυτού βρίσκει το καλύτερο “μονοπάτι” μεταξύ των δύο curves και βρίσκει το μέσο curve αυτών των σημείων. Στο τέλος προσέχω το νέο κεντροειδές που βρήκα να μην ξεπερνάει σε μεγεθος τα υπόλοιπα διανυσματα/curves ή να έχει το μέσο μήκος τους διαγράφοντας στοιχεία αν είναι πολύ μεγάλο (δοκίμασα και **filtering** αλλά διέγραφε πάρα πολλά στοιχεία) και **padding** αν είναι πολύ μικρό. Στο τέλος ελέγχω αν τα κεντροειδή είναι ίδια με τα προηγούμενα που βρήκα και αν είναι τότε τελειώνω το πρόγραμμα και τα τυπώνω στο output.

Αυτά τα έκανα στο τέλος , ήμουν πολύ κουρασμένος και δεν έχω πολύ χρόνο και δύναμη για να λύσω κάποια bug θέματα που έχουν.
Loyd_Frechet. Αυτό δε κάνει καλό clusterıng μαζεύει όλα τα ονόματα σε ένα cluster.
Και το **LSH_Frechet** τρέχει σε ατέρμονους βρόχους επειδή το **rangesearch** δε βρίσκει κανένα curve. Αυτο μπορέι να είναι και το θέμα του lsh που κάποιες φορές δε φτιάχνει καλά το hashtable.

2.Κατάλογος αρχείων

lsh.cpp (main)
clusteringfrechetmain.cpp (main)
class.cpp/h (hashtable)
cliUtils.cpp/h (arguments)
clusteringfrechetfunctions.cpp/h (meancurve)
clusteringfunctions.cpp/h
datasetarray.cpp/h
frechetfunctions.cpp/h
ghashfunction.cpp/h
lshhelper.cpp/h
cube.cpp
cubehash.cpp/h
hypercubefunctions.cpp/h

3.Μεταγλώτιση Και Χρήση Προγράμματος

Όπως πριν υπάρχει makefile, για τους nearestneighbours κάνεις την εντολή
>make lsh
και για το clustering:
>make clustering

και >make clean για διαγραφή των object files
το πργραμμα για το nearestneighbour λεγεται ./lsh και για το clustering ./clustering
algorithm= “lsh” or “frechet”
method= “lsh” or “loyd” or “LSH_Frechet”
update=”Mean_Frechet” και για το Mean_Vector δεν υπάρχει περιορισμός το πρόγραμμα λαμβάνει υπόψη το update αν πατήσει ο χρήστης “Mean_Frechet”
Κάποιες ενδεικτικές εντολές είναι οι εξής:
./clustering -i nasd_input.csv -c cluster.conf -assignment loyd -o out.txt -update Mean_Frechet

./lsh -i nasd_input.csv -q nasd_query.csv -o out.txt -algorithm frechet -metric discrete

./clustering -i nasd_query.csv -c cluster.conf -assignment loyd -o out.txt -update Mean_Vector -silhouette

4.Στοιχεία φοιτητή

Χαράλαμπος Βασιλάκης
A.M.: 1115201500015

Η συνεργάτριά μου τα παρατησε και έφυγε, όχι ότι με βοήθαγε καθόλου ακόμα και στη πρώτη άσκηση.