

Εργασία 2 στο μάθημα της Ανάπτυξης εφαρμογών για αλγοριθμικά προβλήματα

Ομάδα 70

ονοματεπώνυμο 1: Λακές Αθανάσιο

AM 1: 1115201800090

ονοματεπώνυμο 2: Δαμιανός Δαμιανάκης

AM 2: 1115201800306

Στην εργασία αυτή υλοποιήσαμε τα παρακάτω:

ai, aii, aiii από το Α μέρος

Lloyd, Hyper Reverse, LSH Reverse για τα νέα δεδομένα, Reverse LSH. Έχει υλοποιηθεί ο αλγόριθμος `mean_curve` ωστόσο δεν βγάζει σωστά αποτελέσματα.

Γενικά:

Σαν γενικές πληροφορίες να αναφέρω ότι το πρόγραμμα ακολουθεί την ίδια λογική με την πρώτη εργασία όσον αφορά τις δομές. Δεν έχει προστεθεί η ειδική δομή `curves` και αυτό διότι αναπαριστούμε την έννοια της καμπύλης χωρίς να χρειαστεί να αποθηκεύουμε και τα x . Αυτή η ιδέα υλοποιήθηκε με το σκεπτικό ότι τα X μιας καμπύλης δεν είναι ανάγκη να αποθηκευτούν καθώς δεν χρειάζονται. Οι διαδικασίες του `filtering`, του `minimax`, του `snap` χρησιμοποιούν τα X όπως οι αλγοριθμοί το απαιτούν ωστόσο τελικώς αυτά τα x δεν αποθηκεύονται στις δομές.

Να τονίσω ότι πλέον θα βρείτε μια `main.cpp` στον κατάλογο η οποία με την εντολή `make search` κάνει το `compile` για να τρέξετε έπειτα τα

- `runlsh` (απλό LSH)
- `runhyper` (απλό HYPER)
- `runlshfrechetd` (LSH με Discrete Frechet)
- `runlshfrechetc` (LSH με Continuous Frechet)

LSH:

Όσον αφορά τον lsh αλγόριθμο, θα βρείτε τα αρχεία του στον κατάλογο lsh. Έχουμε διαγράψει αρκετό περιττό κώδικα από την 1η εργασία και πλέον ο κατάλογος περιέχει μονάχα τα αρχεία για το hashTable του LSH καθώς και μια τύπου “main” που λέγεται lsh.cpp. Αυτή η μέθοδος, καλείται από την main.cpp και έχει ως στόχο να αρχικοποιήσει τις δομές και να καλέσει τις κατάλληλες συναρτήσεις προκειμένου να τρέξουν οι 3 διαδικασίες του LSH.

Να αναφέρω σε αυτό το σημείο πως χειριζόμαστε τις καμπύλες χωρίς να διατηρούμε τα X. Αρχικά διαβάζοντας τις συζητήσεις του eclass βγήκε το συμπέρασμα ότι θα πρέπει να αποθηκεύουμε τις original καμπύλες. Αυτό μας οδήγησε στην συνειδητοποίηση ότι εφόσον η original καμπύλη θα ήταν της μορφής (1,y1), (2, y2) ... τα X θα μπορούσαμε να τα προσθέτουμε όποτε θέλουμε χωρίς να τα αποθηκεύουμε καθώς οι τιμές τους είναι το index του y. Έτσι λοιπόν, όταν ένα query ψάχνει για γείτονα και πρέπει να γίνει η διαδικασία του DFD (discrete frechet distance) αυτό που γίνεται είναι ότι, λαμβάνεται η καμπύλη από το hashTable η οποία περιέχει μόνο τα Y, τις προστεθούν τα X από το 1 μέχρι και το τελευταίο index, και έπειτα μπορεί να γίνει η διαδικασία του DFD.

Στο snap επίσης χρησιμοποιούμε τα X. Αρχικοποιούμε το input curve μας με x από 1 μέχρι και το τελευταίο index και του συμπεριφερόμαστε σαν να ήταν ένα κανονικό curve με x, y απλώς στο τέλος, αφού ληφθεί το bucket δεν κρατάμε καθόλου τα X. Να αναφέρω ότι το snap σαν συνάρτηση υλοποιεί και τη διαδικασία του duplicate elimination ενώ ακόμη κανουμε και το λεγόμενο padding όπου χρειάζεται. Τελικώς, μετά από snap, duplicate elimination, pad κανουμε το insert στο hash table δίνοντας την τελική καμπύλη και βρίσκοντας από αυτήν το bucket στο οποίο πρέπει η “καμπύλη” να μπει. Για το query ακολουθούμε την ίδια διαδικασία, κάνοντας snap, duplicate elimination και padding ενώ στο τέλος πάμε με αυτές τις modified καμπύλες να βρούμε το bucket το οποίο θα μας δώσει τους πιθανούς γείτονες.

Για τον continuous Frechet, εκτελούμε τις διαδικασίες του filtering και του minima maxima (το τελευταίο γίνεται αν δοθεί όρισμα “continuous” στη διαδικασία snapping). Ακόμη, καταφέραμε να συνδέσουμε τον κώδικα μας με τον κώδικα από το repository του github. Συγκεκριμένα, υπάρχει στο αρχείο helpers.cpp η διαδικασία CFD η οποία δεδομένων 2 καμπυλών, χρησιμοποιεί την toRepoCurve η οποία μετατρέπει μια καμπύλης δική μας σε καμπύλη του repository. Έπειτα καλούνται οι απαραίτητες διαδικασίες προκειμένου να βρεθεί η απόσταση των 2 καμπυλών. Να τονίσω ότι κάνουμε συγκρίσεις μεταξύ των filtered καμπυλών, τις οποίες και αποθηκεύουμε στο hashTable. Αυτό γίνεται για να μειωθεί ο χρόνος εκτέλεσης ο οποίος και πάλι είναι μεγάλος για αυτή τη διαδικασία.

Cluster:

Όσον αφορά το 2ο μέρος της εργασίας, έγινε μεγάλη προσπάθεια να υλοποιηθούν τα ζητούμενα ωστόσο το αποτέλεσμα είναι κάπως χαοτικό. Οι LSH*, HyperCube* και Lloyd δουλεύουν όπως και στην 1η εργασία, έχει προστεθεί ο LSH Reverse Frechet ο οποίος ακολουθεί τη διαδικασία του LSH Frechet στο assignment ωστόσο ενώ έγινε προσπάθεια να δουλέψει και το

mean curve τελικά αποδείχθηκε ότι δεν ήταν εφικτό. Παρόλαυτα τις συναρτήσεις μπορείτε να τις βρείτε στο helpers.cpp (meanCurve, meanFrechet, getOptimalTraversal) οι οποίες ακολουθούν τις διαφάνειες αλλά κάπου χάνουν και τελικά το αποτέλεσμα είναι λάθος.

Στον cluster κατάλογο θα βρείτε ένα makeFile τα target του είναι τα:

- make clusterprog για να φτιαχτεί το main πρόγραμμα.
- make runlsh για τον reverse lsh σε clustering
- make runhyper για reverse hyper
- make runclassic για Lloyd
- make runlshfrechet για Reverse LSH με frechet αλλά όχι με mean curve

Ο κατάλογος cluster περιέχει τα main.cpp, cluster.cpp, checkcommandlinecluster.cpp.

Υπάρχουν ακόμη οι κατάλογοι:

- importantStructs, ο οποίος περιέχει κάποιες σημαντικές δομές που χρησιμοποιούνται σε όλο το πρόγραμμα
- helpers, που περιέχει διάφορες helper συναρτήσεις

Κάποια για το τέλος:

Κλείνοντας θα ήθελα να μιλήσω λίγο για την υλοποίηση της εργασίας. Κατανοώ, προφανώς, ότι το γεγονός ότι δεν περιέχει κάποια class curve είναι κάπως άστοχο. Το πρόβλημα είναι ότι έγινε μια λάθος αντίληψη στο ξεκίνημα της εργασίας, για τη σημασία μιας τέτοιας κλάσης με αποτέλεσμα να πορευτούμε χωρίς αυτή και τελικά να είναι αρκετά αργά για να αλλάξουμε όλον τον κώδικα. Κοιτώντας πίσω σίγουρα θα την συμπεριλάμβανα παρολαυτα να αναφέρω ότι σε πρακτικό κομμάτι δεν με επηρέασε καθόλου μέχρι και την mean_curve. Οι διαδικασίες που θα χρησιμοποιούσαν ολοκληρωτικά μια δομή σαν την curve, κοινώς εκείνες που ασχολούνται και με τα x τελικά τα χρησιμοποιούν με πιο straight-forward και απλό τρόπο (δηλαδή πέρνοντας την original και βάζοντας τα x να ακολουθούν το index, τον οποίο και ακολουθούμε για όσες διαδικασίες το απαιτούν) με αποτέλεσμα η έλλειψη μιας τέτοιας κλάσης να μην καθιστά την υλοποίηση ενός τέτοιου αλγορίθμου αδύνατη. Για να είμαι όμως ειλικρινής μας δυσκόλεψε αρκετά όσο προχωρούσαμε και δεν θέλω να θεωρηθεί ότι είχαμε άγνοια για τη σημασία της. Τέλος να αναφέρω πως δεν έχουμε ασχοληθεί με την αποδέσμευση της μνήμης.

Ευχαριστώ.