α) Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα - Εργασία 1

Η εργασία που αναπτύχθηκε αποτελείται από δύο μέρη. Το πρώτο περιέχει τους αλγόριθμους για την εύρεση κοντινότερων γειτόνων και του range search. Έχουν υλοποιηθεί και οι δύο αλγόριθμοι, LSH και τυχαία προβολή στον υπερκύβο. Το αντίστοιχο πρόγραμμα για τον κάθε αλγόριθμο, αφού διαβάσει το αρχείο εισόδου (input file) θα δημιουργήσει τις απαραίτητες δομές που χρειάζονται για την αναζήτηση. Στην συνέχεια για κάθε query point του αρχείου αναζήτησης (query file) θα βρει τους N προσεγγιστικά κοντινότερους γείτονες μέσω του επιλεγμένου αλγορίθμου καθώς και τους τους N πραγματικά κοντινότερους γείτονες μέσω εξαντλητικής αναζήτησης και θα εκτυπώσει τις αποστάσεις τους στο αρχείο εξόδου (output file). Στην συνέχεια εκτελείται range search για το ίδιο query εκτυπώνοντας τα id των διανυσμάτων εντός της επιλεγμένης ακτίνας.

Το δεύτερο μέρος αφορά τον αλγόριθμο KMeans για την συσταδοποίηση αντικειμένων. Η αρχικοποίηση των κεντροειδών πραγματοποιείται με την τεχνική Initialization++. Στην συνέχεια ανάλογα με τον αλγόριθμο ανάθεσης που έχει επιλεχθεί πραγματοποιείται το σχετικό βήμα. Παρέχονται οι επιλογές: Classic (Ανάθεση με Lloyd's), LSH (Αντίστροφη ανάθεση με LSH) και Hypercube (Αντίστροφη ανάθεση με υπερκύβο). Μετά την ανάθεση, ανανεώνεται το κάθε κεντροειδές παίρνοντας το μέσο των αντικειμένων. Αφού ολοκληρωθεί ο αλγόριθμος, εκτυπώνεται το μέγεθος και το κεντροειδές κάθε cluster και στην συνέχεια πραγματοποιείται η μετρική silhouette για την αξιολόγηση του αλγορίθμου. Αφού εκτυπωθεί λοιπόν το μέσο silhouette για κάθε cluster και το συνολικό μέσο, εκτυπώνονται τα id των αντικειμένων για κάθε cluster (σε περίπτωση που ο χρήστης έχει επιλέξει την παράμετρο complete).

Σχεδιαστικές επιλογές:

Παραθέτουμε τις αποφάσεις που πήραμε για κάποια τμήματα των προγραμμάτων και αιτιολόγηση αυτών.

1) Το ID που χρησιμοποιείται για την γρήγορη εύρεση γειτόνων στο LSH έχει υλοποιηθεί αλλά τελικά αποφασίσαμε να μην το αξιοποιήσουμε γιατί σε ορισμένα input δεν πετυχαίναμε ποτέ ίδιο ID, ή βρίσκαμε πολύ σπάνια. Επιλέξαμε να αφήσουμε σχολιασμένη την σχετική γραμμή που αξιοποιούσε το ID σε περίπτωση μελλοντικής ανάπτυξης.

3) Το TableSize του LSH έχει τιμή N/8, και στην αναζήτηση γειτόνων και στο clustering. Δοκιμάσαμε γενικά και άλλες τιμές όπως N/4 ή N/16. Το N/8 έδινε γενικά τα πιο γρήγορα και πιο ικανοποιητικά αποτελέσματα.

2) Το w έχει την τιμή 700 στο LSH και στο HyperCube. Παρατηρήσαμε με διάφορες εκτελέσεις και ξεχωριστά input αρχεία ότι αυτή η τιμή είναι ιδανική για το w, μιας και δίνει αρκετά ικανοποιητικά αποτελέσματα αλλά και δεν καθυστερεί ιδιαιτέρως την εκτέλεση των αλγορίθμων. Στο clustering ωστόσο, και στα δύο Reverse Assignments, επιλέξαμε την τιμή 100 στο w γιατί παρατηρήσαμε ότι καθυστερούσε αρκετά η εκτέλεση του αλγορίθμου και δεν βελτίωνε ιδιαίτερα τα silhouette scores με τιμές άνω του 100.

3) Τα βήματα της ανάθεσης και ανανέωσης στο clustering εκτελούνται συνολικά για 7 επαναλήψεις. Δοκιμάσαμε και μικρότερους ή μεγαλύτερους αριθμούς επαναλήψεων αλλά ο επιλεγμένος αριθμός έδινε ικανοποιητικά αποτελέσματα σε ιδανικούς χρόνους.

4) Το βήμα της αντίστροφης ανάθεσης και στο LSH και στον υπερκύβο σταματάει άμα ανατεθούν όλα τα στοιχεία του input. Ωστόσο, μπορεί και να διακοπεί όταν σε μία επανάληψη δεν έχει γίνει καμία ανάθεση σε πάνω από τους μισούς clusters. Θεωρήσαμε ότι αυτό είναι ένα ιδανικό κριτήριο διακοπής για το βήμα της ανάθεσης, μιας και βελτίωνε την χρονική του απόδοση.

β) Κατάλογος αρχείων κώδικα/επικεφαλίδων και περιγραφή τους

VectorElement.cpp // Αρχείο κώδικα της κλάσης VectorElement. Η κλάση αυτή χρησιμοποιείται για την αναπαράσταση ενός διανύσματος. Αποτελείται από έναν πίνακα με τις συντεταγμένες του, το id του και άλλες χρήσιμες πληροφορίες για αυτό. Οι κλάσεις αποθηκεύσης (LSHash, HyperCube, Cluster) αποθηκεύουν αντικείμενα της κλάσης VectorElement.
VectorElement.h Αρχείο επικεφαλίδας της κλάσης VectorElement.

LSHMain.cpp // Αποτελεί την main συνάρτηση που χρησιμοποιείται για την εκτέλεση του αλγορίθμου LSH.
LSHash.cpp // Αρχείο κώδικα της κλάσης LSHash. Η κλάση αυτή αποτελεί ένα Local Sensitive Hash table και όλες τις απαραίτητες συναρτήσεις για την εισαγωγή αντικειμένων και αναζήτηση αυτών.
LSHash.h // Αρχείο επικεφαλίδας της κλάσης LSHash.

HyperMain.cpp // Αποτελεί την main συνάρτηση που χρησιμοποιείται για την εκτέλεση του αλγορίθμου τυχαίας προβολής στον υπερκύβο.
HyperCube.cpp // Αρχείο κώδικα της κλάσης HyperCube. Η κλάση αυτή αποτελεί έναν υπερκύβο (υλοποιημένο ως hash table) και όλες τις απαραίτητες συναρτήσεις για την εισαγωγή αντικειμένων και αναζήτηση αυτών.
HyperCube.h // Αρχείο επικεφαλίδας της κλάσης HyperCube.

TableF.cpp // Αρχείο κώδικα της κλάσης TableF. Η κλάση αυτή χρησιμοποιείται για την αποθήκευση τιμών της συνάρτησης f στον υπερκύβο. Αποτελείται από ένα hash table όπου αποθηκεύει όσες τιμές, μιας h, έχουν πάρει 0 και όσες έχουν πάρει 1. Κάθε συνάρτηση f έχει από ένα δικό της TableF.
TableF.h // Αρχείο επικεφαλίδας της κλάσης TableF.

Neighbours.cpp // Αρχείο κώδικα της κλάσης neighboursInfo. Η κλάση αυτή χρησιμοποιείται για την συγκέντρωση N γειτόνων για ένα query. Αποτελείται από έναν πίνακα με τα υποψήφια id καθώς και έναν πίνακα με τις αποστάσεις που συγκεντρώθηκαν. Κάθε LSHash ή ενα HyperCube έχει έναν πίνακα μεγέθους query rows, όπου κάθε θέση του αντιστοιχεί σε ένα αντικείμενο neighboursInfo, για κάθε query.
Neighbours.h // Αρχείο επικεφαλίδας της κλάσης neighboursInfo.

IdDistancePair.cpp // Αρχείο κώδικα της κλάσης idDistancePair. Η κλάση αυτή περιέχει το id και την απόσταση ενός υποψήφιου γείτονα. Χρησιμοποιείται στις main συναρτήσεις για την κατάλληλη οργάνωση των δεδομένων που θέλουμε στο output.
IdDistancePair.h // Αρχείο επικεφαλίδας της κλάσης idDistancePair.

ClusterMain.cpp // Αποτελεί την main συνάρτηση που χρησιμοποιείται για την εκτέλεση του αλγορίθμου KMeans.
KMeans.cpp // Αρχείο κώδικα της κλάσης KMeans. Η κλάση αυτή περιέχει όλα τα απαραίτητα δεδομένα για την εκτέλεση του αλγορίθμου KMeans όπως αριθμό και πίνακα από clusters, HyperCube αντικείμενο και πίνακα από LSHash αντικείμενα για την αντίστροφη ανάθεση κ.τ.λ. Οι μέθοδοι της κλάσης αυτής αντιστοιχούν στα βήματα του αλγορίθμου, όπως αρχικοποίηση κεντροϊδων, ανάθεση αντικειμένων, ανανέωση κ.τ.λ.
KMeans.h // Αρχείο επικεφαλίδας της κλάσης KMeans.

Cluster.cpp // Αρχείο κώδικα της κλάσης Cluster. Η κλάση αυτή περιέχει δεδομένα από ένα cluster όπως ένα VectorElement αντικείμενο που αναπαριστά το κεντροειδές του, λίστα από VectorElement για τα στοιχεία που ανήκουν στον cluster κ.τ.λ.
Cluster.h // Αρχείο επικεφαλίδας της κλάσης Cluster.

Helpers.cpp // Επιπλέον βοηθητικές συναρτήσεις, όπως και για χρήση debugging κ.τ.λ.
Helpers.h // Δηλώσεις των σχετικών συναρτήσεων.

γ) Το πρόγραμμα μεταγλωττίζεται με την χρήση της εντολής: make

Επίσης παρέχεται η δυνατότητα του να αφαιρέσει κανείς τα object και executable αρχεία που παράγονται από το Makefile με την χρήση της εντολής: make clean

Αν επιθυμεί κανείς μπορεί να μεταγλωττίσει και τα επιμέρους προγράμματα ως εξής:
Για το LSH: g++ -o lsh LSHMain.cpp LSHash.cpp Neighbours.cpp IdDistancePair.cpp Helpers.cpp VectorElement.cpp

Για το HyperCube: g++ -o cube HyperMain.cpp HyperCube.cpp TableF.cpp Neighbours.cpp IdDistancePair.cpp Helpers.cpp VectorElement.cpp

Για το Clustering: g++ -o cluster ClusterMain.cpp KMeans.cpp Cluster.cpp VectorElement.cpp LSHash.cpp HyperCube.cpp TableF.cpp Neighbours.cpp Helpers.cpp IdDistancePair.cpp

δ) Οδηγίες χρήσης του προγράμματος

1) Ο αλγόριθμος LSH για την εύρεση κοντινότερων γειτόνων μπορεί να κληθεί με το εκτελέσιμο: ./lsh

Παρέχονται οι επιλογές του να εκτελέσει κανείς τον αλγόριθμο χωρίς ορίσματα (όπου εκεί θα χρησιμοποιηθούν οι default παράμετροι αλλά θα ζητηθεί από τον χρήστη να θέσει τα ονόματα των αρχείων) ή με την χρήση των παραμέτρων που αναφέρει η εκφώνηση. Ακολουθούν κάποιες ενδεικτικές εκτελέσεις:

Με όλες τις παραμέτρους
./lsh -i input_small_id -q query_small_id -k 10 -L 5 -o myLogFile.txt -N 7 -R 100
./lsh -q query_small_id -i input_small_id -o myLogFile.txt -N 7 -R 100 -k 10 -L 5
./lsh -i input_small_id -q query_small_id -k 10 -L 5 -o myLogFile.txt -N 5 -R 250 (μπορείτε να δείτε κάποια αποτελέσματα από αυτήν την εκτέλεση στο αρχείο: lsh_results.txt από το παραδοτέο)

Χωρίς όλες τις παραμέτρους
./lsh -q query_small_id -i input_small_id -o myLogFile.txt

Χωρίς καμία παράμετρο
./lsh

2) Ο αλγόριθμος τυχαίας προβολής στον υπερκύβο για την εύρεση κοντινότερων γειτόνων μπορεί να κληθεί με το εκτελέσιμο: ./cube

Παρέχονται οι επιλογές του να εκτελέσει κανείς τον αλγόριθμο χωρίς ορίσματα (όπου εκεί θα χρησιμοποιηθούν οι default παράμετροι αλλά θα ζητηθεί από τον χρήστη να θέσει τα ονόματα των αρχείων) ή με την χρήση των παραμέτρων που αναφέρει η εκφώνηση. Ακολουθούν κάποιες ενδεικτικές εκτελέσεις:

Με όλες τις παραμέτρους
./cube -i input_small_id -q query_small_id -k 5 -M 6 -probes 7 -o myLogFile.txt -N 3 -R 3000
./cube -i input_small_id -o myLogFile.txt -q query_small_id -k 5 -M 6 -probes 7 -N 3 -R 3000
./cube -i input_small_id -q query_small_id -k 5 -M 5000 -probes 5 -o myLogFile.txt -N 5 -R 250 (μπορείτε να δείτε κάποια αποτελέσματα από αυτήν την εκτέλεση στο αρχείο: cube_results.txt από το παραδοτέο)

Χωρίς όλες τις παραμέτρους
./cube -q query_small_id -i input_small_id -o myLogFile.txt

Χωρίς καμία παράμετρο
./cube

3) Ο αλγόριθμος τυχαίας προβολής στον υπερκύβο για την εύρεση κοντινότερων γειτόνων μπορεί να κληθεί με το εκτελέσιμο: ./cluster

Παρέχονται οι επιλογές του να εκτελέσει κανείς τον αλγόριθμο χωρίς ορίσματα (όπου εκεί θα χρησιμοποιηθούν οι default παράμετροι αλλά θα ζητηθεί από τον χρήστη να θέσει τα ονόματα των αρχείων) ή με την χρήση των παραμέτρων που αναφέρει η εκφώνηση. Ακολουθούν κάποιες ενδεικτικές εκτελέσεις:

./cluster -i input_small_id -c cluster.conf -o myLogFile.txt -complete true -m Classic
./cluster -i input_small_id -c cluster.conf -o myLogFile.txt -complete true -m LSH
./cluster -i input_small_id -c cluster.conf -o myLogFile.txt -complete true -m Hypercube

Για κάθε μία από τις παραπάνω εκτελέσεις παρέχονται ενδεικτικά output files στο παραδοτέο (cluster_classic.txt, cluster_lsh.txt, cluster_cube.txt)

ε) Ονοματεπώνυμα και ΑΜ:
Παντελεήμων Μαλέκας 1115201600268
Θεοφάνης Μπιρμπίλης 1115201600110

Η εργασία αναπτύχθηκε με χρήση Git και είναι διαθέσιμη στο repository: https://github.com/pantmal/Algorithmic-Problems-Project-1. Επίσης χρησιμοποίηθηκε και το extension Live Share του VS Code για την ταυτόχρονη συγγραφή κώδικα.