ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν
Πανεπιστήμιον Αθηνών

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΑΛΓΟΡΙΘΜΙΚΑ ΠΡΟΒΛΗΜΑΤΑ
ΕΡΓΑΣΙΑ 1η

ΑΝΔΡΙΝΟΠΟΥΛΟΥ ΧΡΙΣΤΙΝΑ ΑΜ:1115201500006

Οκτώβριος 2018

Εισαγωγή

Η παρούσα εργασία υλοποιεί όλα τα ζητούμενα της άσκησης. Ο κώδικας έχει αναπτυχθεί σε γλώσσα προγραμματισμού C++ και έχει γίνει χρήση της STL.

Τα δύο ερωτήματα της άσκησης είναι χωρισμένα στους φακέλους LSH για το πρώτο ερώτημα και HYPERCUBE για το δεύτερο. Υπάρχουν ανάμεσα στους δύο φακέλους κοινά αρχεία, ωστόσο κρίθηκε σκόπιμος ο διαχωρισμός των δύο ερωτημάτων σε ξεχωριστά αρχεία, με σκοπό τη διευκόλυνση της διόρθωσης.

Η μεταγλώττιση και για τα δύο προγράμματα γίνεται με make και οι εντολές για run είναι:
για το 1ο ερώτημα (LSH):
$./LSH_routine –d <input file> –q <query file> –k <int> -L <int> -o <output file>
ή
$./LSH_routine (με τα default ορίσματα που δίνονται)

και για το 2ο ερώτημα(HYPERCUBE):
$./hypercube –d <input file> –q <query file> –k <int> -M <int> -probes <int> -o <output file>
ή
$./hypercube (με τα dedfault ορίσματα που δίνονται)

Στο τέλος κάθε εκτέλεσης ο χρήστης ρωτάται αν επιθυμεί να επανεκτελέσει το πρόγραμμα. Αν πληκτρολογηθεί “Y”, τότε το πρόγραμμα εκτελείται εκ νέου ενώ αν δοθεί οποιοσδήποτε άλλος χαρακτήρας το πρόγραμμα τερματίζει εκεί.

- Στα αρχεία υπάρχουν σχόλια, για την καλύτερη κατανόηση του κώδικα.

- Έγινε χρήση git.

- Υπάρχουν ενδεικτικά αρχεία με τα αποτελέσματα και των δύο προγραμμάτων στο παραδοταίο.

- Τα προγράμματα ανταποκρίνονται και σε μεγάλα datasets.

Περιγραφή

Τα προγράμματα διαβάζουν ένα αρχείο διανυσμάτων και δημιουργούν ένα κατάλληλο dataset. Έπειτα διαβάζουν ένα αρχείο ερωτημάτων και δημιουργούν ανάλογο dataset ερωτημάτων.

Έπειτα το LSH κατασκευάζει με κατάλληλο τρόπο unordered multimap για να εισάγει τα δεδομένα, ενώ το HYPERCUBE κατασκευάζει ένα είδος hashtable με τη βοήθεια των vectors.

Στη συνέχεια για κάθε query ο αλγόριθμος ανατρέχει στις παραπάνω δομές και εκτελεί κατάλληλο KNN. Τα αποτελέσματα (απόσταση, χρόνος, γείτονες) αποθηκεύονται σε αρχείο. Στο ίδιο αρχείο αποθηκεύεται επίσης τα αποτελέσματα από bruteforce εκτέλεση. Στο τέλος του αρχείου αυτού υπάρχει ένας αριθμός (fr) που εκφράζει το Μέγιστο (από όλα τα αντικείμενα του συνόλου αναζήτησης) κλάσμα προσέγγισης = Απόσταση προσεγγιστικά κοντινότερου γείτονα / Απόσταση αληθινά κοντινότερου γείτονα καθώς και ο μέσος χρόνος εύρεσης του προσεγγιστικά κοντινότερου γείτονα (avg tLSH).

Στην περίπτωση του LSH χρησιμοποιήθηκε το h όπως δόθηκε στις διαφάνειες του μαθήματος, ενώ για το hashing έγινε χρήση του default hashing που παρέχεται από το unordered_multimap της C++.

Το σύνολο των μαθηματικών εργαλείων που χρειάστηκαν βρίσκονται στο αρχείο mymath.cpp και στους δύο φακέλους.

Με κατάλληλη παραμετροποίηση (για το LSH: k=4 και L=5 και για τον υπερκύβο: k=3, M=10 και probes=3) και τα ίδια data και queries καταφέρνουμε το μέγιστο κλάσμα προσέγγισης να είναι περίπου ίσο για τις δύο μεθόδους, με ευκλείδια μετρική (fr = 4.62324 και fr = 4.68094). Στην περίπτωση αυτή παρατηρούμε ότι η μέθοδος του hypercube που κάνει προβολή σε μικρότερες διαστάσεις από ότι τα δεδομένα, λειτουργεί εμφανώς πιο γρήγορα από ότι το LSH.