ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Υλοποίηση του Locality Sensitive Hashing

Μάθημα: Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα

Ομάδα:

Ανδρίτσου Γεωργία, 1115201200005

Μπίτου Ιωάννα, 1115201200122

1/11/2016

Περιγραφή Προγράμματος

Γενικά

Κατά την εκκίνηση του προγράμματος , ελέγχονται οι παράμετροι εκτέλεσης ως προς την πληρότητα και την ορθότητά τους, διαβάζεται το αρχείο εισόδου και, ανάλογα με το metric space, η ροή οδηγείται στο κατάλληλο block. Στη συνέχεια, αρχικοποιούνται οι δομές του προγράμματος και γίνεται η εισαγωγή των εκάστοτε στοιχείων. Έπειτα, εκτελούνται επαναληπτικά οι αναζητήσεις (εξαντλητική και lsh) από τα αρχεία αναζήτησης και εκτυπώνονται τα αποτελέσματα στα κατάλληλα δομημένα αρχεία εξόδου. Πριν την ολοκλήρωση του προγράμματος, γίνεται η αποδέσμευση όλων των δομών και το πρόγραμμα τερματίζει χωρίς memory leaks.

Παραδοχές / Σχεδιαστικές Επιλογές

Κατά την υλοποίηση, προσπαθήσαμε να δημιουργήσουμε γενικές, δηλαδή, κοινές για όλες τις μετρικές, συναρτήσεις εισαγωγής, αναζήτησης και διαγραφής, οι οποίες διαχωρίζονται εσωτερικά, όπου κρίνεται απαραίτητο, ανάλογα με το metric space που ζητείται . Με σκοπό την επεκτασιμότητα του προγράμματος, χρησιμοποιήσαμε, επίσης, τόσο κοινά structs όσο και τον τύπο void * για το πέρασμα των παραμέτρων στις κοινές συναρτήσεις. Αξίζει να σημειώσουμε ότι:

- Το path των αρχείων εισόδου, εξόδου και αναζήτησης ζητείται υποχρεωτικά από το command line του προγράμματος. Έτσι, εάν δε δοθεί οποιοδήποτε από αυτά, ο χρήστης καλείται να ξαναεκτελέσει το πρόγραμμα.
- Αφού ολοκληρωθεί η αναζήτηση και η εξαγωγή των αποτελεσμάτων για το πρώτο αρχείο αναζήτησης, ο χρήστης ενημερώνεται με σχετικό μήνυμα για την επιτυχή ολοκλήρωση ή όχι και επιλέγει επαναληπτικά εάν θέλει να ξαναεκτελέσει για διαφορετικό αρχείο αναζήτησης, ονοματίζοντας και το καινούριο αρχείο εξόδου.
- Η εξαντλητική αναζήτηση (brute force) αναζητά τον αληθινό πλησιέστερο γείτονα στον πρώτο hash table της δομής. Η συνάρτηση αυτή γεμίζει μία λίστα με όλους τους αληθινούς κοντινότερους γείτονες.
- Οι συναρτήσεις εύρεσης γειτόνων σε ακτίνα-R και πλησιέστερων γειτόνων lsh γεμίζουν δύο λίστες, οι οποίες περιέχουν όλους τους γείτονες που βρέθηκαν σύμφωνα με τους ελέγχους. Οι παραπάνω λίστες εκτυπώνονται στα αρχεία εξόδου και αποδεσμεύονται άμεσα.
- Το trick που προτάθηκε για την Euclidean και την Cosine μετατράπηκε στον έλεγχο : (στοιχεία που έχουν ελεγχθεί από τη δομή < 6*L), αφού παρατηρήσαμε ότι επιτυγχάνεται μεγαλύτερη ακρίβεια στα αποτελέσματα μεταξύ lsh και brute force, χωρίς ωστόσο να επιβαρύνεται ιδιαίτερα ο χρόνος εκτέλεσης. Στην brute force ο έλεγχος παραλείπεται, αφού επιθυμούμε να ελέγξουμε απαραιτήτως όλα τα στοιχεία του hash table.

- Τα μεγέθη των hash tables έχουν επιλεγεί, βάση θεωρίας αλλά και σύμφωνα με δικούς μας ελέγχους, ως εξής: Για το Hamming: 2^k, για το Euclidean: lines/16+1, για το Cosine: 2^k και για το Matrix, ομοίως, 2^k.
- Το w, δηλαδή το window size έχει γίνει #define 4, έτσι ώστε όποιος έχει γνώση της ακτίνας του αρχείου αναζήτησης, να μπορεί χειροκίνητα να το αυξήσει για να λάβει καλύτερα αποτελέσματα.
- Επιλέχθηκε να γίνεται cunit testing για τις συναρτήσεις υπολογισμού των αποστάσεων και τις hash functions.
- Δημιουργήσαμε το πρόγραμμα check, το οποίο αποτρέπει την εμφάνιση παράλογων αποτελεσμάτων ελάχιστης απόστασης πλησιέστερου γείτονα. Να σημειωθεί, βέβαια, ότι το πρόγραμμα ελέγχει λαμβάνοντας υπ' όψιν μόνο τα συγκεκριμένα queries που μας έχουν δοθεί για κάθε metric space. Εν ολίγοις, ελέγχει αν κάποια απόσταση hamming βγει >23 ή μια απόσταση matrix βγει >10 ή μια απόσταση euclidean /cosine βγει >2. Το πρόγραμμα μεταγλωττίζεται ως εξής: gcc -o check check.o και εκτελείται ./check output.csv output1.csv output2.csv…output.csv –H (για Hamming), ./check output.csv output1.csv output2.csv…output.csv –E (για Euclidean), ./check output.csv output1.csv output2.csv…output.csv –C (για Cosine) και ./check output.csv output1.csv output2.csv…output.csv –M (για DistanceMatrix).

Αρχεία Κώδικα / Κεφαλίδων

lsh.c: Περιέχει τη main.

Hash.c: Περιέχει τις συναρτήσεις που αφορούν τους hash πίνακες, καθώς και τις συναρτήσεις για αρχικοποίηση των hash functions, τις ίδιες τις hash functions , τις συναρτήσεις για αναζήτηση σε επίπεδο πίνακα και την brute force αναζήτηση.

chain.c: Περιέχει τις συναρτήσεις που αφορούν τις λίστες (chains) , όπως εισαγωγή, αναζήτηση γειτόνων σε ακτίνα R και αναζήτηση πλησιέστερου γείτονα.

distances.c: Περιέχει τις συναρτήσεις υπολογισμού αποστάσεων.

nnrlist.c: Περιέχει συναρτήσεις για εισαγωγή σε λίστα , εκτύπωση λίστας σε αρχείο, συνδυασμό λιστών όπου η μία γράφεται στο τέλος της άλλης και καταστροφή λίστας.

hash.h : Περιέχει τα πρότυπα των συναρτήσεων του hash.c καθώς και τη δομή των hash tables και τη δομή των g hash functions.

chain.c: Περιέχει τα πρότυπα των συναρτήσεων του chain.c καθώς και τη δομή των λιστών (chains) των πινάκων.

nnrlist.c: Περιέχει τα πρότυπα των συναρτήσεων και τη δομή των λιστών που περιέχουν τους γείτονες σε ακτίνα R, όλους τους πλησιέστερους γείτονες με την ίδια ακτίνα και όλους τους πραγματικά πλησιέστερους γείτονες.

distances.h: Περιέχει τα πρότυπα των συναρτήσεων υπολογισμού αποστάσεων.

check.c : Περιέχει main που ελέγχει αν τα αποτελέσματα στο αρχείο εξόδου είναι σωστά, μόνο για τα δοθέντα αρχεία αναζήτησης.

Εντός του φακέλου Tests υπάρχουν δύο φάκελοι, ο DistancesTest και ο HashTest.

Οι φάκελοι αυτοί περιέχουν αρχεία για cunit testing των συναρτήσεων για υπολογισμό αποστάσεων και των hash functions.

Οδηγίες Μεταγλώττισης Προγράμματος

Τα αρχεία κώδικα συνοδεύονται από ένα Makefile , συνεπώς ένα make αρκεί.

Αναλυτικά, οι εντολές για μεταγλώττιση είναι:

gcc -g -c lsh.c

gcc -g -c chain.c

gcc -g -c hash.c

gcc -g -c distances.c

gcc -g -c nnrlist.c

gcc -g -o lsh lsh.o chain.o hash.o distances.o nnrlist.o -lm

Οδηγίες Χρήσης Προγράμματος

Η εκτέλεση του προγράμματος γίνεται μέσω της γραμμής ./lsh –d <input file> –q <query file> –k <int> -L <int> -o <output file>. Αρχικά το πρόγραμμα ελέγχει έναν ο αριθμός των ορισμάτων υπερβαίνει τον μέγιστο αριθμό που μπορεί να δοθεί ή αν είναι περιττός και ενημερώνει τον χρήστη με μήνυμα λάθους.

Τα ορίσματα μπορούν να δοθούν με οποιαδήποτε σειρά. Επίσης τα ορίσματα k, L είναι προαιρετικά οπότε η εκτέλεση μπορεί να γίνει μέσω της γραμμής ./lsh –d <input file> –q <query file> -o <output file>. Οι default τιμές των k και L είναι 4 και 5 αντίστοιχα. Σε περίπτωση που δοθούν αρνητικοί αριθμοί το πρόγραμμα ζητά από τον χρήστη να εισάγει εκ νέου τιμή. Στις περιπτώσεις Distance Matrix, Euclidean και Cosine εάν ο χρήστης έχει εισάγει k μεγαλύτερο του 10 ή L μεγαλύτερο 30 το πρόγραμμα ζητά από τον χρήστη να εισάγει νέο αριθμό έως ότου δώσει αριθμό εντός τον ορίων.

Μετά την εισαγωγή των στοιχείων στη δομή και την αναζήτηση των γειτόνων κάθε στοιχείου του αρχείου αναζήτησης που δόθηκε στην γραμμή εντολών γίνεται ερώτηση στον χρήστη αν επιθυμεί να συνεχίσει την αναζήτηση για ένα άλλο αρχείο αναζήτησης. Εάν πληκτρολογήσει Y ζητείται το νέο αρχείο αναζήτησης καθώς και το νέο αρχείο αποτελεσμάτων. Η διαδικασία αυτή συνεχίζεται έως ότου ο χρήστης πληκτρολογήσει N.

Πηγές

1. http://stackoverflow.com/questions/16870485/how-can-i-read-an-input-string-of-unknown-length

2. http://stackoverflow.com/questions/4003232/how-to-code-a-modulo-operator-in-c-c-obj-c-that-handles-negative-numbers

3. https://phoxis.org/2013/05/04/generating-random-numbers-from-normal-distribution-in-c/

Προδιαγραφές Μηχανημάτων

1. Επεξεργαστής: Intel i7-3610QM CPU @ 2.30GHz 2.30 GHz , Oracle VM VirtualBox με Ubuntu (64-bit).

2. Επεξεργαστής: Intel Atom CPU N2600 @ 1.60GHz x 4, Ubuntu 16.04 LTS (32-bit).