

1η Προγραμματιστική Εργασία

Αναζήτηση και συσταδοποίηση διανυσμάτων στη C/C++

Ονοματεπώνυμο	ΑΜ
Κόλιας Γεώργιος	1115201600070
Κατρακάζας Γρηγόριος	1115201600063

Makefile

main_lsh.cpp

// main lsh

main_cube.cpp

// main cube

main_clustering.cpp

// main clustering

math.cpp

// μαθηματικές συναρτήσεις

math.h

ReadMNIST.cpp

// διαβασμα αρχείων, αποθήκευση των δεδομένων σε <vector<vector >> και επιστροφή του size του.

ReadMNIST.h

hashtable.cpp

// HashTableCreate

HashTableDestroy

HashTableInsert

HashTableFindNN -> *και για lsh και για cube* -> Βρίσκει τον approximate κοντινότερο, νιοστό και σε ακτίνα R γείτονα

hashtable.h

lsh_function.cpp

// συνάρτησεις
h -> hash function
g -> υπολογίζει το g
manhattan distance
find_W -> υπολογισμός W
true_nearest_neighbour -> Βρίσκει τον πραγματικό κοντινότερο γείτονα
true_nearest_neighbourNR -> Βρίσκει τον πραγματικό νιοστό κοντινότερο γείτονα και σε ακτίνα R
lsh_nearest_neighborNR-> Καλεί την HashTableFindNN που επιστρέφει τον approximate κοντινότερο, νιοστό και σε ακτίνα R γείτονα για κάθε hashtable , και μετά βρίσκει συνολικά τον κοντινότερο, νιοστό και σε ακτίνα R
lsh_nearest_neighborR -> ίδια με την απο πάνω αλλά για το cube

header.h
ApproximateNNStatistics.h
TrueNNStatistics.h

ComparePairs.h
// δομή ουράς

Clustering.cpp
//συναρτήσεις
ClusteringInitialization -> επιλέγει τα κέντρα - clusters (το 1ο τυχαία , και τα υπολοιπα με βάσει το αλγόριθμο kmeans++)
ClusteringAssignmentLLoyd -> ανάθεση κάθε εικόνας στο κοντινότερο κέντρο σύμφωνα με το LSH (βρίσκει επαναληπτικά εικόνες μέσα σε ένα R)
ClusteringAssignmentLSH -> ανάθεση κάθε εικόνας στο κοντινότερο κέντρο σύμφωνα με τον αλγόριθμο LLoyd
ClusteringAssignmentCUBE -> ανάθεση κάθε εικόνας στο κοντινότερο κέντρο σύμφωνα με το Cube (βρίσκει επαναληπτικά εικόνες μέσα σε ένα R)
ClusteringUpdate -> επανυπολογισμός καινούργιου κέντρου σε κάθε cluster ώστε να έχουμε το ελαχιστο αθροισμα αποστασεων της κάθε εικόνας απο το κέντρο
ClusteringPrint -> εκτυπωση κεντρων και χρόνου
ClusteringSilhouette-> υπολογισμος μέσου όρου silhouette όλων των εικόνων
ClusteringFinishCondition -> τέλος επαναλήψεων αν το καινούργιο κεντρο απέχει πολύ λιγο απο το παλιο

Clustering.h

Compile: > make

- g++ -o lsh main_lsh.cpp lsh_function.cpp math.cpp hashtable.cpp ReadMNIST.cpp
- g++ -o cube main_cube.cpp lsh_function.cpp math.cpp hashtable.cpp ReadMNIST.cpp
- g++ -o clustering main_clustering.cpp clustering.cpp lsh_function.cpp math.cpp hashtable.cpp ReadMNIST.cpp

Παράδειγμα εκτελέσεων

- ./lsh -d train-images.idx3-ubyte -q t10k-images.idx3-ubyte -k 4 -L 5 -o results_lsh -N 5 -R 10000
- ./cube -d train-images.idx3-ubyte -q t10k-images.idx3-ubyte -k 4 -M 5000 -probes 5000 -o results_cube -N 5 -R 10000

Αυτή είναι η εκτέλεση για το cube.

Για 30 πρώτα έχουμε το αρχείο results_cube.txt

- ./clustering -i t10k-images.idx3-ubyte -c clustering.conf -o results_clustering -m Classic

Αυτή είναι η εκτέλεση για το clustering και έχουμε το αρχείο results_clustering.txt.

Περιγραφή εργασίας

Έχουν υλοποιηθεί όλα τα ζητούμενα. Έχουμε επισυνάψει τα παρακάτω αρχεία με βάση της παραπάνω εκτελέσεις:

- results_lsh.txt
- results_cube.txt

Τα οποία έχουν τα αποτελέσματα των πρώτων 30 queries

- results_clustering.txt

Το οποίο περιέχει τα αποτελέσματα το clustering