

Κατακερματισμός και Αναζήτηση για διανύσματα και πολυγωνικές καμπύλες στη C/C++

Ιωάννης Μήτρου
AM : 1115201400108

Μαίρη Ξανθοπούλου
AM : 1115201400300

Από τα ζητούμενα υλοποιήθηκαν όλα, εκτός από το assignment του LSH για τα διανύσματα και τις καμπύλες και της πρώτης update για τις καμπύλες.

1. Αρχεία που παραδίδονται

Μέσα στο παραδοτέο έχουμε 2 φακέλους : Clusters_vectors, Clusters_curves.

Ο πρώτος περιέχει τα απαραίτητα αρχεία για την υλοποίηση των αλγορίθμων συσταδοποίησης διανυσμάτων και ο δεύτερος για την υλοποίηση συσταδοποίησης καμπυλών.

1.1 Αρχεία για τα διανύσματα

cluster.c (όπου βρίσκεται η main), cluster.h, init.c (υλοποίηση των 2 συναρτήσεων init), init.h, assignment.c (υλοποίηση μίας συνάρτησης assignment), assignment.h, update.c (υλοποίηση των 2 συναρτήσεων update), update.h, Makefile (για γρήγορο compile) και ένα test-input και test-config αρχείο.

1.2 Αρχεία για τις καμπύλες

assign.c (υλοποίηση του assignment), assign.h, functions.c (υλοποίηση βοηθητικών functions-LSH-Grids-Manhattan κλπ), functions.h, update.c (υλοποίηση του update), update.h, makefile, structs.c (οι δομές), hash.c (διάβασμα αρχείων και αποθήκευση), hash.h, main.c (υλοποίηση της main).

Επίσης, περιλαμβάνεται test-input και test-config. Σημειώνεται, πως για τα διανύσματα το config διαβάζεται με κενό μετά από : , ενώ στις καμπύλες χωρίς.

2. Εκτέλεση

2.1. Διανύσματα

./cluster -i <inputfile> -c <configfile> -o <outfile> . Συγκεκριμένα : ./cluster -i Data500... -c config.txt -o outfile

2.2. Καμπύλες

./clustering -I <inputfile> -c <configfile> -o <outfile>. Συγκεκριμένα : ./clustering -i input.dat -c cluster.conf -o outfile

3. Αλγόριθμοι/Σχεδιαστικές Επιλογές

3.1. Διανύσματα

Δομές:

-Μία δομή Space που έχει έναν πίνακα με τα σημεία

- Η δομή του σημείου στο χώρο που έχει έναν πίνακα διαστάσεων, το id του σημείου και δείκτη στο centroid του.
- Η δομή του centroid που είναι σαν αυτήν του σημείου μόνο που δεν έχει προφανώς δείκτη σε centroid, αλλά μία μεταβλητή size τύπου int, στην οποία αποθηκεύεται το πλήθος των σημείων που αντιστοιχούν στο συγκεκριμένο κέντρο(cluster).

Μετρική που χρησιμοποιείται:

manhattan (όπως και για τις καμπύλες)

Λογική Υλοποίησης:

Ουσιαστικά η εργασία απαιτεί την εκτέλεση 8 διαφορετικών αλγορίθμων. Αφού στα διανύσματα υλοποιήθηκε ο ένας από τους 2 assignment, οι αλγόριθμοι που εκτελούνται είναι 4. Στο cluster.c, όπου και βρίσκεται η main διαβάζονται αρχικά τα inputfile και configfile και αποθηκεύονται τα διανύσματα του inputfile στη δομή Space. Στη συνέχεια καλούνται όλες οι παραλλαγές των αλγορίθμων με τον εξής τρόπο : 3 εμφωλευμένες for που εκτελούνται 2 φορές η καθεμία. Αυτό ισοδυναμεί με 2^3 επαναλήψεις και συνδυασμούς. Εσωτερικά των for έχουμε μία επανάληψη while για να εκτελούμε τις assign και update πολλαπλές φορές. Η while σταματάει αν βρει ένα flag που έχουμε θέσει θετικό (και όταν ξεπεράσει κάποιο threshold). Για να γίνει το flag θετικό πρέπει μετά την update τα κέντρα να είναι ολόδια, και για να το εξετάσουμε αυτό χρησιμοποιούμε τις βοηθητικές συναρτήσεις set_cluster και is_same. Στο τέλος του cluster.c αποδεσμεύεται η μνήμη που χρησιμοποιήθηκε για τα σημεία και τα clusters!!!

Στο αρχείο init.c βρίσκεται η υλοποίηση των 2 init. Η απλή init (init1) διαλέγει στην τύχη από τον χώρο των σημείων k σημεία για αρχικά κέντρα. Η πιο πολύπλοκη init (init2-kmeans++) φτιάχνει τον πίνακα αθροιστικών πιθανοτήτων, από τον οποίο, με τυχαίο sampling, προσδιορίζει το επόμενο κέντρο, μέχρι να συμπληρωθεί ο επιθυμητός αριθμός, k.

Στο αρχείο assignment.c έχουμε υλοποιήσει απλά την απλή assignment. Αυτή διατρέχει όλα τα σημεία του χώρου και για καθένα βρίσκει την minimum απόσταση από κάποιο κέντρο και θέτει τον δείκτη του σημείου να δείχνει σε αυτό το κέντρο.

Στο αρχείο update.c βρίσκεται η υλοποίηση των 2 update. Η απλή update (update1) βρίσκει τα καινούρια κέντρα στο χώρο και όχι αναγκαστικά πάνω στα υπάρχοντα και αυτό το κάνει με το να βρει για κάθε διάσταση το άθροισμα των αποστάσεων και να διαιρέσει με το πλήθος. Από την άλλη, η update2(PAM) ουσιαστικά βελτιώνει την πρώτη πρόβλεψη που κάνει η Init για το συγκεκριμένο cluster, διαλέγοντας ένα άλλο από τα υπάρχοντα σημεία ΜΕΣΑ στο συγκεκριμένο cluster. Πιο συγκεκριμένα, για κάθε cluster, βρίσκει για κάθε σημείο το άθροισμα των αποστάσεων όλων των άλλων σημείων στο cluster από αυτό και μετά από όλα αυτά τα αθροίσματα, διαλέγει το σημείο με το μικρότερο και αυτό γίνεται το καινούριο cluster. Αυτό επαναλαμβάνεται για κάθε cluster. Όσον αφορά την υλοποίηση, για να το κάνουμε αυτό χρησιμοποιούμε έναν πίνακα για τις αποστάσεις και έναν πίνακα που κρατάει το index του σημείου για κάθε άθροισμα αποστάσεων. Έτσι μπορούμε για κάθε cluster να βρούμε το minimum άθροισμα αποστάσεων και ποιο σημείο (σε ποιο index στον πίνακα των σημείων του χώρου) είναι.

3.2. Καμπύλες

Δομές:

Οι δομές που χρησιμοποιούνται είναι παρόμοιες με αυτές των διανυσμάτων με κάποιες επιπλέον για τα hashtables και grids.

Λογική Υλοποίησης:

Η λογική υλοποίησης στις καμπύλες των init, assignment, update είναι παρόμοια με τα διανύσματα, μόνο που για να υπολογίσουμε την απόσταση καμπύλων χρησιμοποιούμε DTW. Στην main παίρνουμε τα data από το input file και από το configuration file και τα αποθηκεύουμε. Μετά, όπως στην 1η εργασία, ξεκινάει η διαδικασία να δημιουργήσουμε τα L hash tables και τα grids.

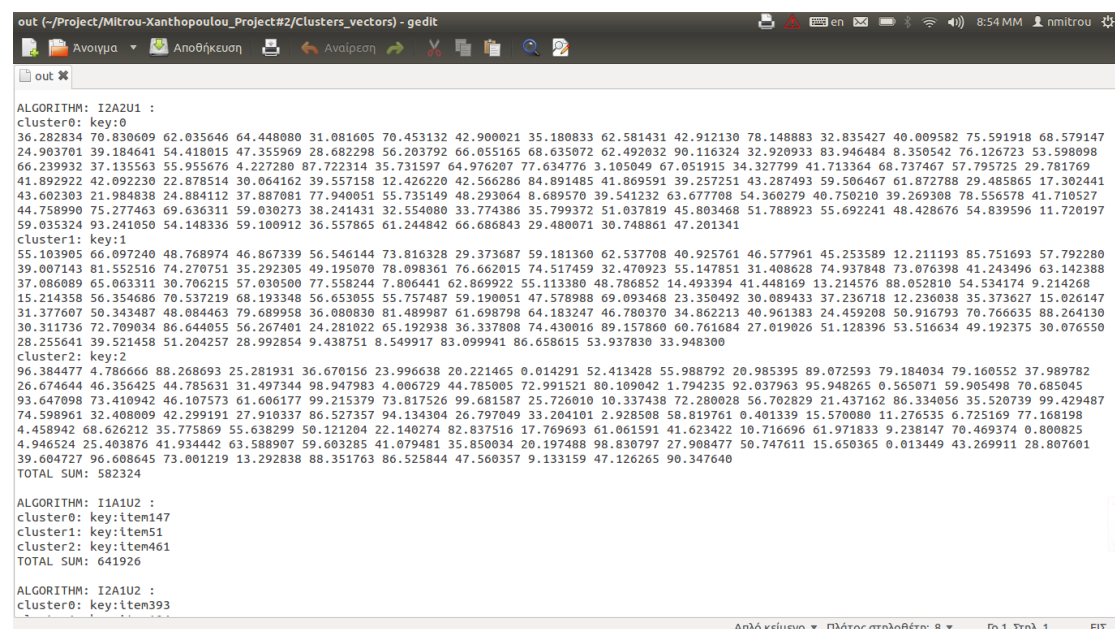
Έπειτα τρέχουν όλοι οι συνδυασμοί από τους αλγορίθμους init 1 , init 2 , assign 1 και update 1.

4. Αποτελέσματα

Τα αποτελέσματα στην εκτέλεση του προγράμματος των διανυσμάτων και αυτού των καμπυλών είναι της μορφής που ζητείται χωρίς να περιλαμβάνεται ο χρόνος και το silhouette. Στα διανύσματα, για να δείχνουμε περίπου πόσο καλός είναι ο αλγόριθμος εκτυπώνουμε το συνολικό άθροισμα των αποστάσεων των σημείων από τα κέντρα τους (objective function).

4.1. Διανύσματα

Screenshot



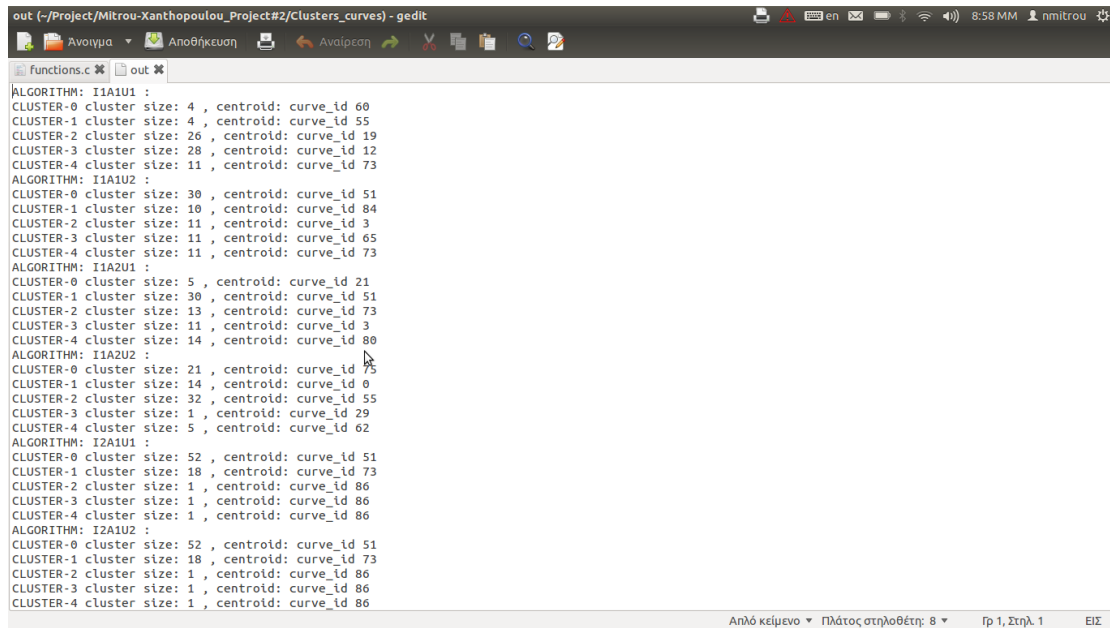
```
out (-/Project/Mitrou-Xanthopoulos_Project#2/Clusters_vectors) - gedit
out *
ALGORITHM: I2A2U1 :
cluster0: key:0
36.282834 70.830609 62.035646 64.448080 31.081605 70.453132 42.900021 35.180833 62.581431 42.912130 78.148883 32.835427 40.009582 75.591918 68.579147
24.903701 39.184641 54.418015 47.355969 28.682298 56.203792 66.055165 68.635072 62.492032 90.116324 32.920933 83.946484 8.350542 76.126723 53.598098
66.239932 37.135563 55.955676 4.227280 87.722314 35.731597 64.976207 77.634776 3.105049 67.051915 34.327799 41.713364 68.737467 57.795725 29.781769
41.892922 42.092230 22.878514 30.064162 39.557158 12.426220 42.566286 84.891485 41.869591 39.257251 43.287493 59.506467 61.872788 29.485865 17.302441
43.602303 21.984838 24.804112 37.887081 77.940051 55.735149 48.293064 8.689570 39.541232 63.677708 54.360279 40.750210 39.269308 78.556578 41.710527
44.758990 75.277463 69.636311 59.030273 38.241431 32.554080 33.774386 35.799372 51.037819 45.803468 51.788923 55.692241 48.428676 54.839596 11.720197
59.835324 93.241050 54.148336 59.100912 36.557865 61.244842 66.686843 29.480071 30.748861 47.201341
cluster1: key:1
55.103905 66.097240 48.768974 46.867339 56.546144 73.816328 29.373687 59.181360 62.537708 40.925761 46.577961 45.253589 12.211193 85.751693 57.792280
39.007143 81.552516 74.270751 35.292305 49.195070 78.098361 76.662015 74.517459 32.470923 55.147851 31.408628 74.937848 73.076398 41.243496 63.142388
37.086089 65.063311 30.706215 57.030500 77.558244 7.806441 62.869922 55.113380 48.786852 14.493394 41.448169 13.214576 88.052810 54.534174 9.214268
15.214358 56.354686 70.537219 68.193348 56.653055 55.757487 59.190051 47.578988 69.093468 23.350492 30.089433 37.236718 12.236038 35.373627 15.026147
31.377607 50.343487 48.084463 79.689958 36.080830 81.489987 61.698798 64.183247 46.780370 34.862213 40.961383 24.459208 50.916793 70.766635 88.264130
30.311736 72.709034 86.644055 56.267401 24.281022 65.192938 36.337808 74.430016 89.157860 60.761684 27.019026 51.128396 53.516634 49.192375 30.076550
28.255641 39.521458 51.204257 28.992854 9.438751 8.549917 83.099941 86.658615 53.937830 33.948300
cluster2: key:2
96.384477 4.786666 88.268693 25.281931 36.670156 23.996638 20.221465 0.014291 52.413428 55.988792 20.985395 89.072593 79.184034 79.160552 37.989782
26.674644 46.356425 44.785631 31.497344 98.947983 4.006729 44.785005 72.991521 80.109042 1.794235 92.037963 95.948265 0.565071 59.905498 70.685045
93.470998 32.410802 46.107573 61.606177 99.215379 73.817526 99.681587 25.726010 10.337438 72.280028 56.702829 21.437162 86.334056 35.520739 99.429487
74.598961 32.408094 42.299191 27.910337 86.527357 94.134304 26.797049 33.204101 2.928508 58.819761 0.401339 15.570080 11.276535 6.725169 77.168198
4.458942 68.626212 35.775869 55.638299 50.121204 22.140274 82.837516 17.769693 61.061591 41.623422 10.716696 61.971833 9.238147 70.469374 0.800825
4.946524 25.403876 41.934442 63.588907 59.603285 41.079481 35.850034 20.197488 98.830797 27.908477 50.747611 15.650365 0.013449 43.269911 28.807601
39.604727 96.608645 73.001219 13.292838 88.351763 86.525844 47.560357 9.133159 47.126265 90.347640
TOTAL SUM: 582324

ALGORITHM: I1A1U2 :
cluster0: key:item147
cluster1: key:item51
cluster2: key:item461
TOTAL SUM: 641926

ALGORITHM: I2A1U2 :
cluster0: key:item393
```

4.2. Καμπύλες

Screenshot



```
out (~/Project/Mitrou-Xanthopoulos_Project#2/Clusters_curves) - gedit
functions.c out
ALGORITHM: I1A1U1 :
CLUSTER-0 cluster size: 4 , centroid: curve_id 60
CLUSTER-1 cluster size: 4 , centroid: curve_id 55
CLUSTER-2 cluster size: 26 , centroid: curve_id 19
CLUSTER-3 cluster size: 28 , centroid: curve_id 12
CLUSTER-4 cluster size: 11 , centroid: curve_id 73
ALGORITHM: I1A1U2 :
CLUSTER-0 cluster size: 30 , centroid: curve_id 51
CLUSTER-1 cluster size: 10 , centroid: curve_id 84
CLUSTER-2 cluster size: 11 , centroid: curve_id 3
CLUSTER-3 cluster size: 11 , centroid: curve_id 65
CLUSTER-4 cluster size: 11 , centroid: curve_id 73
ALGORITHM: I1A2U1 :
CLUSTER-0 cluster size: 5 , centroid: curve_id 21
CLUSTER-1 cluster size: 30 , centroid: curve_id 51
CLUSTER-2 cluster size: 13 , centroid: curve_id 73
CLUSTER-3 cluster size: 11 , centroid: curve_id 3
CLUSTER-4 cluster size: 14 , centroid: curve_id 80
ALGORITHM: I1A2U2 :
CLUSTER-0 cluster size: 21 , centroid: curve_id 75
CLUSTER-1 cluster size: 14 , centroid: curve_id 0
CLUSTER-2 cluster size: 32 , centroid: curve_id 55
CLUSTER-3 cluster size: 1 , centroid: curve_id 29
CLUSTER-4 cluster size: 5 , centroid: curve_id 62
ALGORITHM: I2A1U1 :
CLUSTER-0 cluster size: 52 , centroid: curve_id 51
CLUSTER-1 cluster size: 18 , centroid: curve_id 73
CLUSTER-2 cluster size: 1 , centroid: curve_id 86
CLUSTER-3 cluster size: 1 , centroid: curve_id 86
CLUSTER-4 cluster size: 1 , centroid: curve_id 86
ALGORITHM: I2A1U2 :
CLUSTER-0 cluster size: 52 , centroid: curve_id 51
CLUSTER-1 cluster size: 18 , centroid: curve_id 73
CLUSTER-2 cluster size: 1 , centroid: curve_id 86
CLUSTER-3 cluster size: 1 , centroid: curve_id 86
CLUSTER-4 cluster size: 1 , centroid: curve_id 86
Απλό κείμενο ▾ Πλάτος στηλοθέτη: 8 ▾ Γρ 1, Σηλ. 1 ΕΙΣ
```