Ανάπτυξη Λογισμικού για Αλγοριθμικά Προγράμματα

3η Προγραμματιστική Εργασία

Υλοποίηση αλγορίθμων υπόδειξης (Recommendation)

Συσταδοποίηση μοριακών διαμορφώσεων

Βασίλειος Βρουλιώτης, 1115201300025, sdi1300025@di.uoa.gr
Παναγιώτης Φιλιάνος, 1115201300193, sdi1300193@di.uoa.gr
Για οποιαδήποτε απορία μη διστάσετε να επικοινωνήσετε.

Περιγραφή:
Το πρόγραμμα που μεταγλωττίζεται παρέχει όλη τη λειτουργικότητα που περιλαμβάνεται στην περιγραφή του από την εκφώνηση. Παρακάτω παρέχεται μία σύντομη ματιά στις δομές που χρησιμοποιήθηκαν ανά την περίπτωση καθώς και των σχεδιαστικών επιλογών μας. Επιπλέον μπορούν να βρεθούν οδηγίες για την μεταγλώττιση.
Ταυτόχρονα περιλαμβάνεται μία σύντομη ανάλυση των αποτελεσμάτων που προέκυψαν κατά τη συσταδοποίηση των μοριακών διαμορφώσεων.

Κατάλογος συναρτήσεων (FunctionsDescription.txt):
Μαζί με τα αρχεία περιλαμβάνεται και πλήρης οδηγός κάθε υλοποιημένης συνάρτησης FunctionsDescription.txt
Έτσι, το πρόγραμμα ανάγεται σε εργαλείο βιβλιοθήκης και μπορεί να χρησιμοποιηθεί και σε άλλα projects. Περισσότερες λεπτομέρειες για τη διεπαφή θα βρει κανείς στα αρχεία επικεφαλίδας.
Απευθυνθείτε εκεί και στα σχόλια του κώδικα ώστε να επιλύσετε οποιαδήποτε απορία.

Σχεδιαστικές επιλογές:
Οι παρακάτω σχεδιαστικές επιλογές αφορούν το σύνολο των προγραμμάτων:

Βασικές δομές εκείνες των clusterAssign, centroids και clusterTable. Η λειτουργία τους περιέχεται παρακάτω:

- clusterAssign: Ένας πίνακας που περιέχει για κάθε στοιχείο ποιό είναι το κοντινότερο κεντροειδές, το δεύτερο κοντινότερο και σε ποιό βρίσκεται, ανά πάσα στιγμή.
- centroids: Ένας πίνακας που περιέχει τα κεντροειδή ανά πάσα στιγμή.
- clusterTable: Ένας πίνακας λιστών, όπου κάθε λίστα (αποτελούμενη απο ClusterNodes) αποτελεί ένα, επικαιροποιημένο ανά πάσα στιγμή, cluster.

Ταυτόχρονα έχουν επιλεγεί σχετικά σπάταλες (από άποψη χώρου) δομές. Ο στόχος αυτού είναι να βελτιστοποιηθεί το πρόγραμμα από άποψη χρόνου. Περισσότερες λεπτομέρειες στην παρακάτω επεξήγηση:

Όσον αφορά στους αλγορίθμους υπόδειξης:

- user_rating_table : Ένας πίνακας που περιλαμβάνει όλες τις μετρήσεις για κάθε χρήστη. Οι γραμμές είναι οι χρήστες και οι στήλες αντιστοιχούν στη μέτρηση που έχει δώσει για το αντίστοιχο στοιχείο.
- user_general_rating_table: Ένας πίνακας που περιλαμβάνει τον μέσο όρο των μετρήσεων για κάθε χρήστη.
- NN_table: Περιλαμβάνει για κάθε χρήστη τους κοντινότερους γείτονες ανάλογα με τον εκάστοτε αλγόριθμο.

Σε καθε περίπτωση, στους παραπάνω αλγορίθμους έχει γίνει υλοποίηση **και με λίστες**, ώστε να δοθεί στον developer και η δυνατότητα αξιολόγησης τους.

Όσον αφορά στους αλγόριθμους συσταδοποίησης μοριακών διαμορφώσεων:

- cRMSD:
    - comformation_X_consec: Περιλαμβάνει όλα τα άτομα μίας διαμόρφωσης συνεχόμενα. Όμοια για comformation_Y_consec.
    - comformation_X_consec_rotated: Ο παραπάνω πίνακας αφού έχει περάσει από πίνακα περιστροφής
    - returned_table: Πολλαπλασιασμός των πινάκων X,Y.
    - x_q_minus_y: Ο πίνακας XQ-Y
    - q_rotation_table: Ο πίνακας $Q = UV^T$
    - u_table: Ο πίνακας U που προκύπτει από SVD: $X^T*Y = U\Sigma V^T$
    - singular: Ο πίνακας Σ που προκύπτει από SVD: $X^T*Y = U\Sigma V^T$
    - v_t_table: Ο πίνακας V που προκύπτει από SVD: $X^T*Y = U\Sigma V^T$
- Δημιουργία του χαρακτηριστικού διανύσματος μέσω αποστάσεων:
    - all_conformation_table: Πίνακας που περιλαμβάνει το σύνολο των δεδομένων. Είναι τριών διαστάσεων. Η πρώτη διάσταση περιλαμβάνει τις διαμορφώσεις. Η δεύτερη τα άτομα, ενώ η τρίτη μία από τις τρείς συντεταγμένες (x,y,z) του συγκεκριμένου ατόμου.
    - all_first_conf_distances: Πίνακας που περιλαμβάνει πια ζευγάρια επιλέχθηκαν από την πρώτη διαμόρφωση. Δες παρακάτω για τον τρόπο indexing των ζευγαριών αποστάσεων.
    - distance_pairs: Λαμβάνοντας υπ' όψην πια r ζευγάρια έχουν επιλεγεί από την πρώτη διαμόρφωση, ο πίνακας αυτός περιέχει, για κάθε εκάστοτε διαμόρφωση, τις αποστάσεις μεταξύ των αντίστοιχων ατόμων για όλα τα ζευγάρια.
    - Συγκεκριμένα για το indexing των ζευγαριών: Το index ξεκινάει από το 0. Αυτό αντιστοιχεί στην απόσταση μεταξύ του πρώτου και του δεύτερου ατόμου. Στη συνέχεια στο index 1 αντιστοιχεί η απόσταση μεταξύ του πρώτου και του τρίτου ατόμου. Ακολουθώντας όμοια την λογική, index N-1 (με N ο αριθμός των ατόμων σε κάθε διαμόρφωση) έχει η απόσταση μεταξύ του πρώτου και του τελευταίου ατόμου. Έτσι index N θα έχει η απόσταση μεταξύ του δεύτερου και του τρίτου ατόμου κ.ο.κ.

Κάθε κύκλος Assign - Update γίνεται 3 φορές. Ο αριθμός αυτός προκύπτει εμπειρικά και δίνει καλή σχέση χρόνου - τιμών. Αυτό μπορεί να αλλάξει στο

αρχείο ListsFunctions.cpp και από τα αντίστοιχα αρχεία που περιλαμβάνουν assign - update κύκλους.

Κατά την εκτέλεση των LSH αλγορίθμων δίνεται το default (default, k = 4, L = 5).

Σε όλον τον κώδικα υπάρχει συνεπής χρήση των delete. Το ίδιο ισχύει και για τις συναρτήσεις Unit Testing.
Για τη διασφάλιση του παραπάνω έγινε εκτενής χρήση του εργαλείου valgrind ώστε να εκμηδενιστούν τα memory leaks καθώς και να διασφαλιστεί η πλήρως ορθή λειτουργία του προγράμματος.

Ο κώδικας είναι κατάλληλα διαμερισμένος σε επιμέρους αρχεία. Αυτό διαπιστώνεται και με ένα make count στο working directory.

Σε κάθε περίπτωση ο κώδικας είναι πλήρως τεκμηριωμένος και οποιαδήποτε απορία μπορεί να επιλυθεί εκεί.

Unit Testing:
Υλοποιείται Unit Testing για δύο ενδεικτικές κλάσεις: Node και ListData.
Αυτές είναι ενδεικτικές και περιλαμβάνουν τη λογική πίσω από το unit testing.
Ταυτόχρονα, σε σχέση με τις προηγούμενες εργασίες διατηρήθηκαν οι ίδιες κλάσεις, καθώς έγινε πλήρης χρήση των ήδη υπάρχοντων εργαλείων έχοντας τη διάθεση να δημιουργηθεί μία δυναμική διεπαφή, επίλυσης clustering και hashing προβλημάτων.
Στη σελίδα του **github** (https://github.com/Destinyplyr/MoleculesAndRecommendation)θα προστεθούν unit tests για όλες τις υπόλοιπες κλάσεις.

Χρόνοι εκτέλεσης:
Οι χρόνοι εκτέλεσης για μεγάλα αρχεία εισόδου μπορεί να χρειάζονται μία επιπλέον δόση υπομονής για τους αλγορίθμους υπόδειξης. Επομένως περιλαμβάνονται μικρότερα αρχεία για να γίνει δοκιμή του προγράμματος.

Οδηγίες Μεταγλώττισης/ Εκτέλεσης:
Στα αρχεία του προγράμματος, περιλαμβάνεται Makefile. Η μεταγλώττιση γίνεται με την εντολή:

```
>make
>make all
```

στο directory που βρίσκονται τα αρχεία.
Σε περίπτωση που δεν προκύπτουν τα επιθυμητά αποτελέσματα, κάνουμε reset το build του εκτελέσιμου αρχείου μας, ως εξής:

```
>make clean
```

και στη συνέχεια:

```
>make
```

Οι εντολές της μεταγλώττισης βρίσκονται μέσα στο Makefile.

Για το πρόγραμμα των αλγορίθμων υπόδειξης:
Μέσω της εκτέλεσης του, παράγεται εκτέλεσιμο recommendation.

Για εκτέλεση δίνουμε:

```
>./recommendation -d yahoo_music_small.dat -o out.txt
```

Προαιρετικά, μπορούμε να προσθέσουμε παραμέτρους:

```
-validate
```

θέλοντας να εκτελέσουμε και 10-fold cross validation.
Οποιοσδήποτε συνδυασμός των παραπάνω είναι αποδεκτός.

Για το πρόγραμμα συσταδοποίησης των μοριακών διαμορφώσεων:
Μέσω της εκτέλεσης του, παράγεται εκτέλεσιμο molecules.

Για εκτέλεση δίνουμε:

```
>./molecules
```

Οδηγίες Χρήσης του Προγράμματος :
Ο χρήστης προμηθεύει το πρόγραμμα με τα ζητούμενα αρχεία και το αποτέλεσμα προκύπτει στο out.txt (ή conform.dat και experim.dat για την περίπτωση του προγράμματος molecules)
Ταυτόχρονα τυπώνονται ορισμένα διαγνωστικά μηνύματα στην οθόνη κατά την εκτέλεση.
Οποιαδήποτε πληροφορία που θα μπορούσε να ήταν χρήσιμη κατά την εκτέλεση του προγράμματος έχει σχολιαστεί στα πηγαία αρχεία και υπάρχει η δυνατότητα να δοθεί ανά πάσα στιγμή.

Μετρήσεις της συσταδοποίησης διαμορφώσεων:

Ύστερα από σειρά μετρήσεων, οι οποίες παρουσιάζονται στο experimental_data.txt, προκύπτουν τα εξής μέσα αποτελέσματα:

---

r: N
T: min distance pairs
μέσο k: 7
μέση Silhouette: 0,1723886
μέσο Clustering time: 0,0318714

---

r: N
T: max distance pairs
μέσο k: 8,2
μέση Silhouette: 0,1606012
μέσο Clustering time: 0,0303332

---

r: N
T: random distance pairs
μέσο k: 6,6
μέση Silhouette: 0,1169376
μέσο Clustering time: 0,0306738

---

r: $N^{1.5}$
T: min distance pairs
μέσο k: 8,2
μέση Silhouette: 0,1514448
μέσο Clustering time: 0,0301718

---

r: $N^{1.5}$
T: max distance pairs
μέσο k: 8
μέση Silhouette: 0,1935868
μέσο Clustering time: 0,0302436

---

r: $N^{1.5}$
T: random distance pairs
μέσο k: 8,2
μέση Silhouette: 0,1941828
μέσο Clustering time: 0,0302436

---

r: All pairs
T: -
μέσο k: 8,2
μέση Silhouette: 0,1665744
μέσο Clustering time: 0,0307548

Αναλύοντας τα παραπάνω δεδομένα καταλήγουμε πως σημαντική βελτίωση στα αποτελέσματα εμφανίζεται για r: $N^{1.5}$ και T ώστε τα ζευγάρια που επιλέγονται να μην είναι μικρής απόστασης.
Ταυτόχρονα με την παραπάνω βελτίωση δεν παρατηρείται και αντίστοιχη χρονική απαίτηση, πράγμα σημαντικό για την αποτελεσματικότητα του αλγόριθμου.
Τέλος, συμπεράνουμε πως η αύξηση των cluster οδηγεί και σε αύξηση του αντίστοιχου silhouette (επισημαίνουμε πως κενά clusters έχουν silhouette = 0).