Κ23γ: Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα
Χειμερινό εξάμηνο 2016-17
3η Προγραμματιστική Εργασία
Υλοποίηση αλγορίθμων Υπόδειξης (Recommendation) & Συσταδοποίηση Μοριακών Διαμορφώσεων

Εκπονήθηκε από τους φοιτητές:
- Βασίλειος Δρέττας με ΑΜ: 1115201300042
- Κυριακή Ράπτη με ΑΜ: 1115201100105

Η άσκηση που υλοποιήθηκε περιλαμβάνει δυο κομμάτια:
- Το πρώτο κομμάτι μας ζητούσε να υλοποιήσουμε δυο μεθόδους Recommendation: την NN-LSH Recommendation Και την Clustering Recommendation για ένα σύνολο αντικειμένων και χρηστών.
- Το δεύτερο κομμάτι μας ζητούσε να εκτελέσουμε τους αλγορίθμους για τον υπολογισμό απόστασης cRMSD και dRMSD των μοριακών διαμορφώσεων

Επιγραμματικά η εκτέλεση του προγράμματος είναι η εξής:

-Στο πρώτο μέρος:
- Υλοποίηση της μεθόδου NN-LSH Recommendation/Clustering Recommendation
- Κανονικοποίηση των αξιολογήσεων
- Αξιολόγηση των μη αξιολογημένων αντικειμένων από τον χρήστη(χρησιμοποιειται ο τυπος $R(u,i)= R(u)+ z*\Sigma sim(u,υ)*(R(υ,i)-R(υ))$

- Υπόδειξη των 5 καλύτερων αντικειμένων με βάση τις αξιολογήσεις
- Για την ευρεση των γειτονικων χρηστων υλοποιηθηκαν δυο μεθοδοι. Μια που παιρνει τους χρηστες και τους ταξινομει και μια που υλοποιει το binary search που παρουσιαστηκε στο μαθημα. Παρατηρησαμε πιο γρηγορα αποτελεσματα με την απλη ταξινομηση οποτε κρατησαμε αυτην την μεθοδο.
- Στην cutoff παρατηρησαμε καλυτερα αποτελεσματα αμα θεταμε μηδεν οσα δν εχουν βαθμολογια και 1 οσα εχουν. Οποτε αλλαξαμε το οριο

Το πρώτο μέρος υλοποιήθηκε για τρεις μετρικές: Euclidean, Cosine και Hamming.
Για την αξιολόγηση των αποτελεσμάτων υλοποιήθηκε ο αλγόριθμος του 10-fold cross validation.

-Στο δεύτερο μέρος:

- Υλοποίηση του αλγορίθμου υπολογισμού απόστασης c-RMSD για ένα σύνολο μοριακών διαμορφώσεων (για την υλοποίηση χρησιμοποιήθηκε η εξωτερική μαθηματική βιβλιοθήκη Eigen)
- Συσταδοποίηση των μοριακών διαμορφώσεων και εκτέλεση του c-RMSD (για την εύρεση του κατάλληλου k , χρησιμοποιείται η μέθοδος Silhouette)
- Υλοποίηση του d-RMSD αλγορίθμου για κάθε μοριακή διαμόρφωση, η κατασκευή ενός διανύσματος μεγέθους r μεταξύ των ζευγών σημείων της διαμόρφωσης. (και εδώ γίνεται χρήση της Eigen) (για την εύρεση του κατάλληλου k , χρησιμοποιείται η μέθοδος Silhouette)

Τα αρχεία που υλοποιήθηκαν για αυτή την άσκηση είναι τα εξής(εδώ συμπεριλαμβάνω μόνο τα καινούργια αρχεία που υλοποιήθηκαν αποκλειστικά για το part 3):

1. cRMSDConform.h: το αρχείο περιλαμβάνει την κλάση CRMSDConform που χρησιμοποιείται για την υλοποίηση του αλγορίθμου cRMSD και τον υπολογισμό της απόστασης. Γίνεται χρήση της μαθηματικής βιβλιοθήκης Eigen και στη διαχείριση των δεδομένων καθώς έχουμε ορίσει τα δεδομένα με βάση τους τύπους που περιέχει η Eigen.
2. cRMSDConform.h:: ορισμός της κλάσης CRMSDConform
3. dRMSD.h: περιλαμβάνει την κλάση DRMSD όπου χρησιμοποιείται για την υλοποίηση του αλγορίθμου DRMSD (το Β ζητούμενο του δεύτερου μέρους)
4. dRMSD.cpp:ορισμός της κλάσης DRMSD
5. kFolds.h: περιλαμβάνει την κλάση KFolds που έχει σκοπό να χωρίζει το input σε k ομάδες όπου οι k-1 αξιοποιούνται στο training set και η k ομάδα χρησιμοποιείται για το validation.
6. kFolds.cpp: ορισμός της κλάσης KFolds
7. main.cpp: η main για τους αλγοριθμους recommendation
8. mainConformations.cpp: η main για τη συσταδοποιηση μοριακων διαμορφωσεων
9. recommendation.h: υλοποιεί τη μέθοδο του recommendation όπως αναγράφεται και στην εργασία χρησιμοποιώντας μια από τις δυο μεθόδους: NN-LSH Recommendation ή Clustering Recommendation
10. recommendation.cpp: ορισμός των κλάσεων που εμπεριέχονται στο .h
11. shape.h: περιέχει την κλάση Shape η οποία αξιοποιείται στο μέρος της εργασίας με τις μοριακές διαμορφώσεις για τη δομή του input

12. shape.cpp: ορισμός της κλάσης Shape

Συμπερασματα:
Recommendation

Παρατηρησαμε οτι απο ολες τις μετρικες την πιο καλη κατανομη στα hashtables και στα clusters ειχε η ευκλιδεια μετρικη και ως αποτελεσμα ηταν και η πιο γρηγορη στις περισσοτερες περιπτωσεις.

Οσον αφορα την MAE και οι τρεις μετρικες παρουσιασαν παρομοια αποτελεσματα και δν μπορουμε να διακρινουμε καποια απο τις τρεις ως καλυτερη.

Επισης στην μεθοδο clustering οταν το k προσεγγιζει το N/P (N το πληθος των χρηστων, P το πληθος των γειτονων) εχουμε καλυτερα αποτελεσματα σε σχεση με αλλα k (σχετικα με τον δεικτη silhouette).

Συσταδοποίηση μοριακών διαμορφώσεων
Καλυτερα αποτελεσματα (στον δεικτη silhouette) παρατηρησαμε οταν επιλεγαμε τις μικροτερες r αποστασεις. Οι χρονοι και στις τρεις επιλογες ηταν οι ιδιοι το οποιο ειναι λογικο αφου δεν κανουμε καποια παραπανω διαδικασια σε καποια απο τις τρεις επιλογες.

Εντολές Μεταγλώττισης:

- make (μόνο με αυτή την εντολή το σύστημα θα βρει το makefile – εφόσον έχουμε μόνο ένα- και θα εκτελέσει όλες τις εντολές μεταγλώττισης που εμπεριέχονται στο αρχείο)

Οδηγίες Χρήσης:

Εγκατάσταση βιβλιοθήκης Eigen:

Unzip το αρχειο eigen-eigen-f562a193118d.tar.gz και τοποθετηση του στον φακελο /usr/local/include/

Οδηγίες Εκτελεσης:

Συσταδοποίηση μοριακών διαμορφώσεων(part καθ. Εμιρη)

./conformation –i <input file> -o <output file> -a <drmsd or crmsd> -r <n,nsqrtn,all> -T <low,random,high> -all

input file: Αρχειο εισοδου
output file: Αρχειο εξοδου
-a <drmsd,crmsd>: Αλγοριθμος εκτελεσης (πχ. -a drmsd ή -a crmsd)
-r <n,nsqrtn,all>: πληθος r αποστασεων, μονο για το drmsd (πχ. -r n) (default τιμη το n)

-T <low,random,high>: Τυπος r αποστασεων (μικροτερες,τυχαιες,μεγαλυτερες), μονο για το drmsd (default τιμη low)
-all: τρεχουν ολοι οι συνδιασμοι για το drmsd (n και low, n και random etc)

Recommendation(part καθ. Χαμοδρακα)
./recommendation –d <input file> -o <output file> -validate -m <clustering,lsh> -t <Hamming, Cosine, Euclidean>

Οι παραμετροι validate, -m και -t ειναι προαιρετικοι
Αμα εκτελεστει χωρις τα -m -t τοτε ελενχονται οι οι συνδιασμοι

-m <clustering,lsh>: μεθοδος recommendation (πχ -m lsh)
-t <Hamming, Cosine, Euclidean>: μετρικη που θα χρησιμοποιηθει(πχ -t Euclidean)

Παραδειγμα εκτελεσης ./recommendation -d yahoo_music_small.dat -o output -m lsh -t Cosine