ΑΜ : 1115201400024
Δημήτριος Γάγγας

Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα

Project 3
Recommendation

Υλοποιήθηκε το Recommendation επιτυχώς
σε C++11 με STL library σε λειτουργικό Linux.

- **[Optimality Note]** Σχεδόν όλες οι παράμετροι στις συναρτήσεις γίνονται pass by reference για εξοικονόμηση χώρου και βελτιστοποίηση της ταχύτητας.

- Ελέγχθηκαν επιτυχώς με Valgrind .Συνεπώς δεν υπάρχουν memory leaks.

- Χρησιμοποιήθηκε το git και το gitkraken για οπτικοποίηση των εκάστοτε αλλαγών αλλά και για την ευκολία επιστροφής σε προηγούμενα στάδια ανάπτυξης του κώδικα.

- Χρησιμοποιήθηκε ο κώδικας για το lsh, hypercube καθως και του Clustering που είχαν υλοποιηθεί επιτυχώς.

- Περιέχεται makefile που μεταγλωττίζεται με make.

- Έχουν υλοποιηθεί όλες οι απαιτήσεις της εκφώνησης για τα ορίσματα με όποια σειρά και να δοθούν τα flags.
  $./recommendation –d <input file> -o <output file>

- Επίσης μπορεί να δοθεί και το flag -validate με το οποίο τρέχει το cross-validation για καθε ενα απο τα 4 ερωτήματα και μας δινει το εκάστοτε average MAE.

- Για τον καλύτερο δυνατό έλεγχο των αρχείων δημιουργήθηκαν φάκελοι.

- Περιέχεται στο φάκελο **demos** και ένα ενδεικτικές εκτελέσεις με αποτέλεσματα που έβγαλε το πρόγραμμα.

- Περιέχεται και ενα εκτελέσιμο μέσα στο φάκελο με ονομασια **recommendation**.

Αρχεία .cpp/.h που δημιουργήθηκαν :

- AbstractLSH_CLUSTER.cpp/.h
- ClusterAPI.cpp/.h
- ClusteringProxSearching.cpp/.h
- Cross_Validation.cpp/.h

Δημιουργήθηκαν οι εξης φάκελοι:

- **Readme** στον οποίο περιέχονται τα readme και των προηγούμενων ασκήσεων.
- **Inputs** στον οποίο βρίσκονται τα input files
- **Configs** στον οποίο υπάρχουν τα configuration files για τις εξης οντότητες:
    - *Lsh1.conf* : για το lsh των πραγματικών Users (uj)
    - *Lsh2.conf* : για το lsh των εικονικών Users(cj)
    - *Cluster1.conf* : για το clustering των Tweets και τη δημιουργια των εικονικών Users-vectors(cj)
    - *Cluster2.conf* : για το clustering των πραγματικών Users (uj)
    - *Cluster3.conf* : για το clustering των εικονικών Users (cj)
- .**git** με όλα τα commits που γίναν.
- **Outputs** στον οποίο βρίσκονται (Αν επιθυμεί ο χρήστης απο τα config files)
  τα outputs των cluster απο project2 αν δοθούν οι συνδυασμοί:
  Output: yes | Output_file_name: ./outputs/Out[*] | Silhouette: yes.

*__Σημαντικη λεπτομέρεια__ : Στα cluster[*].conf αρχεία δόθηκε η δυνατότητα να γινεται ξεχωριστα το configuration του καθε cluster.
Επίσης, προστέθηκαν οι εξης δυνατότητες:

1. Να δίνεται ο μέγιστος αριθμός επαναλήψεων που θα τρέξει το clustering
2. Να μπορεί ο χρήστης να εκτυπώσει τα αποτελέσματα της 2ης εργασίας καθως επίσης να έχει τη δυνατότητα να επιλέξει το αρχείο εξόδου(output).
3. Η δυνατότητα επιλογής Silhouette και εκτύπωσης της
4. Καθώς και η δυνατότητα επιλογής ολοκληρωμένης εκτύπωσης μέσω της επιλογής complete: yes

Η εύρεση του (τοπικού) βέλτιστου k (αριθμός συστάδων) για καθε cluster:

1. [*Cluster1.conf*]:Για το clustering των tweets και τη δημιουργια των εικονικών χρηστών
   Χρησιμοποιήθηκαν οι καλύτεροι δυνατοί παράμετροι απο τη 2η εργασία
   [K = 200 , I1A1U2 , euclidean]. ----> Mean Silhouette ~0.4

2. [*Cluster2.conf*] : Για το clustering των u χρησιμοποιήθηκαν οι εξής παράμετροι
   [K = 150 , I1A1U1 , euclidean]. ----> Mean Silhouette ~0.4-0.5

3. [*Cluster3.conf*] : Για το clustering των c χρησιμοποιήθηκαν οι εξής παράμετροι
   [K = 15 , I1A1U1 , euclidean]. ----> Mean Silhouette ~0.3-0,4

Λεπτομέρειες υλοποίησης

Ακολουθήθηκε ο 2ος τυπος των διαφανειων σελίδα 3 :
R(u,i)= R(u)+ z* Σsim(u,υ)*(R(υ,i)-R(υ))

Συνεπώς, η κανονικοποίηση γίνεται on the fly στην αποτίμηση του κάθε νομίσματος.

Τα δεδομένα φυλασσονται σε **unordered_map** για την δυνατότητα που δίνει εύρεσης κλειδιού σε constant time.
Η συσχέτιση Users -Tweets αναπαριστάται σε **multimap**.
Στο userTweetsSentimScore_umap βρίσκονται οι συσχετίσεις User-CryptoScore (Αντίστοιχα για cj)
Στα crypto για τα οποία δεν έχει μιλήσει ο χρήστης υπάρχουν inf τιμές.
Στο userTweetsSentimScoreWithoutInfsAndZeroVectors_umap αντικαθιστόνται τα inf στα cryptoScore απο τους μ.ο του εκάστοτε χρήστη αλλα και γίνονται discard τα μηδενικά διανύσματα.
Τα διανύσματα που βρίσκονται στο παραπάνω umap εισάγονται στο lsh και Cluster.

Επιπλέον:

1. Δημιουργήθηκε η **class** AbstractLshCluster η οποία κληρωνομεί εναν τις κλάσεις Lsh και ClusterProxSearching με αποτέλεσμα για lsh και Clustering να μην χρειάζονται να γραφούν 2 ξεχωριστες συναρτήσεις.
2. Δημιουργήθηκαν 2 functor για το Recommendations Coins.Ενα για το τα uj και ενα για τα cj.
3. Τέλος, δημιουργήθηκαν οι 2 συναρτήσεις για το cross-Validation.
   a. 10-fold cross validation για την περιπτωση των uj
   b. Επιλέγονται επαναληπτικά (10 φορες) καποια γνωστά νομίσματα(μπορούμε να ορίσουμε αριθμό) και γίνεται πρόγνωση για αυτά βάσει της σύγκρισης με τους εικονικούς χρήστες.