Κ23γ: Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα - Χειμερινό εξάμηνο 2016-17
3η Προγραμματιστική Εργασία

Η εργασία έχει 2 σκέλη. Πρέπει να υλοποιηθεί σε σύστημα Linux και να υποβληθεί στις Εργασίες του e-class το αργότερο την Παρασκευή 13/01/2016 στις 23.59.

Υλοποίηση αλγορίθμων υπόδειξης (Recommendation)

Έστω σύνολο αντικειμένων `I={i1, i2, …, iM}`, σύνολο χρηστών `U={u1, u2, …, uN}`. Θα υλοποιήσετε τις παρακάτω δύο μεθόδους Recommendation, όπου κάθε χρήστης αναπαρίσταται από το διάνυσμα διάστασης Μ των κανονικοποιημένων αξιολογήσεων των αντικειμένων. Οι αξιολογήσεις κανονικοποιούνται βάσει της μέσης αξιολόγησης κάθε χρήστη `u: R'(u,i)=R(u,i)−R(u)` όταν χρησιμοποιείται η ευκλείδεια μετρική και η ομοιότητα συνημιτόνου. Όταν χρησιμοποιείται η μετρική Hamming εφαρμόζεται η προσέγγιση της αποκοπής (rounding / cut-off).

Α. Μέθοδος NN-LSH Recommendation

1. Εύρεση P πλησιέστερων γειτόνων κάθε χρήστη με χρήση του range LSH και binary-repeated range search (βλ. διαφάνειες: 3b.recommend.pdf).
2. Αξιολογήσεις των μη αξιολογημένων αντικειμένων με χρήση του σταθμισμένου αθροίσματος `R’(u,i) = z*Σsim(u,υ)*R’(υ,i)`, όπου `υ` οι `P` γείτονες του `u`.
3. Υπόδειξη των 5 καλύτερων αντικειμένων από το `I` ανά χρήστη στο `U`, εξ όσων δεν είχαν αξιολογηθεί.

Β. Μέθοδος συσταδοποίησης (Clustering) Recommendation

1. Συσταδοποίηση των χρηστών σε `k` συστάδες (clusters) με αλγόριθμο της επιλογής σας. Βελτιστοποίηση του k βάσει της αξιολόγησης Silhouette: να συγκρίνετε το βέλτιστο `k` με το `N/P`.
2. Ομοίως με το (Α2), όπου `υ` οι χρήστες στην ίδια συστάδα με τον `u`.
3. Ομοίως με το (Α3).

Θα υλοποιήσετε κάθε μέθοδο με τρεις μετρικές: την Ευκλείδεια μετρική, τη μετρική Hamming και την ομοιότητα συνημίτονου (Cosine similarity) που είναι ψευδομετρική. Οι έξι αλγόριθμοι να αξιολογηθούν βάσει του μέσου απόλυτου σφάλματος με τη μέθοδο 10-fold cross-validation. Σε κάθε μέθοδο να συγκρίνετε την απόδοση των τριών μετρικών.

ΕΙΣΟΔΟΣ

Αρχείο κειμένου `input.dat` διαχωρισμένο με στηλοθέτες (tab-separated), το οποίο θα έχει την ακόλουθη γραμμογράφηση:

```
P: <Integer>
1    915  1
1    952  1
1    964  5
1    979  1
2    93   5
2    127  3
2    170  5
2    255  5
2    292  5
```

Στην πρώτη γραμμή ορίζεται ο αριθμός `P` των κοντινότερων γειτόνων που θα χρησιμοποιηθούν (default 20). Η πρώτη στήλη αντιστοιχεί στο `userId`, η δεύτερη στήλη στο `itemId` (αμφότερες σε αύξουσα ακολουθία) και η τρίτη στήλη στη βαθμολογία που έχει δώσει ο χρήστης για το αντικείμενο που κυμαίνεται από 1 έως 5. Το αρχείο εισόδου ακολουθεί το πρότυπο του dataset R3 -Yahoo! Music ratings for User Selected and Randomly Selected songs.

Το αρχείο `input.dat` δίνεται μέσω παραμέτρου στη γραμμή εντολών. Η εκτέλεση θα γίνεται μέσω της εντολής:

```
$./recommendation –d <input file> -o <output file>
```

Η αξιολόγηση των μεθόδων πραγματοποιείται μέσω της εντολής:

```
$./recommendation –d <input file> -o <output file> -validate
```

ΕΞΟΔΟΣ

Η εκτέλεση του προγράμματος παράγει ένα αρχείο κειμένου, το οποίο θα περιλαμβάνει τα αντικείμενα που θα υποδειχθούν σε κάθε χρήστη για κάθε αλγόριθμο. Οι χρήστες και τα αντικείμενα ταυτοποιούνται μέσω ακέραιων αριθμών.

```
<Algorithm> (π.χ. Hamming LSH)
<u1>  <item11>    <item12>    <item13>    <item14>    <item15>
<u2>  <item21>    <item22>    <item23>    <item24>    <item25>
.....................................................
<uN>  <itemN1>    <itemN2>    <itemN3>    <itemN4>    <itemN5>
Execution Time: <milliseconds>
```

Η εκτέλεση του προγράμματος για την αξιολόγηση των αλγόριθμων εκτυπώνει για κάθε αλγόριθμο:

```
<Algorithm> MAE: <Double>
```

Συσταδοποίηση μοριακών διαμορφώσεων

Δίνεται σύνολο μοριακών διαμορφώσεων (conformations). Κάθε διαμόρφωση αποτελείται από μία συγκεκριμένη ακολουθία `N` σημείων στον τριδιάστατο Ευκλείδειο χώρο.

Η είσοδος είναι αρχείο κειμένου `input.dat`. Οι δύο πρώτες γραμμές περιέχουν το πλήθος διαμορφώσεων και το `N`. Το υπόλοιπο αρχείο είναι διαχωρισμένο με στηλοθέτες σε 3 στήλες που περιέχουν τις `N` τριάδες συντεταγμένων: `x y z` της 1ης διαμόρφωσης, έπειτα `N` τριάδες της 2ης διαμόρφωσης κοκ, δηλ. συνολικά `numConform*N` γραμμές μετά τις 2 πρώτες:

```
numConform: <Integer>
N: <Integer>
-32.5       91.2        11.7
12.8        -18.3       79.1
...
```

A. Ζητείται (1) η υλοποίηση της συνάρτησης απόστασης c-RMSD, με την χρήση εξωτερικής βιβλιοθήκης γραμμικής άλγεβρας, π.χ. LAPACKE / GNU Scientific Library / Eigen, (2) η συσταδοποίηση σε `k` συστάδες (clusters), για δοσμένο `k`, με αλγόριθμο της επιλογής σας και χρήση της c-RMSD, (3) η εύρεση του κατάλληλου `k` μέσω του αλγόριθμου silhouette.

Η έξοδος είναι αρχείο κειμένου `conform.dat`, το οποίο εκφράζει τη συσταδοποίηση με την βέλτιστη τιμή silhouette, διαχωρισμένο με στηλοθέτες με την ακόλουθη γραμμογράφηση:

```
k: <Integer>
s: <real in (-1,1)>
1     9     11    12 ….
2     3     17    [η 2η συστάδα περιέχει 3 στοιχεία]
```

όπου η 1η γραμμή περιέχει το k, η 2η την τιμή του silhouette, και οι επόμενες k γραμμές περιέχουν τους δείκτες των στοιχείων των αντίστοιχων συστάδων σε αύξουσα σειρά.

B. Ζητείται:
(1) η κατασκευή, για κάθε διαμόρφωση, διανύσματος r αποστάσεων μεταξύ ζευγών σημείων της διαμόρφωσης, για δοσμένο r μικρότερο ή ίσο του N(N-1)/2, όπου η επιλογή των ζευγών γίνεται με βάση τη μεταβλητή T που δηλώνει τις r μικρότερες, r μεγαλύτερες ή r τυχαίες αποστάσεις (τα r ζεύγη επιλέγονται με βάση τις αποστάσεις τους στην πρώτη διαμόρφωση),
(2) η συσταδοποίηση με χρήση Ευκλείδειας μετρικής στα διανύσματα του (1) για δοσμένα r, T, με αλγόριθμο της επιλογής σας (τύπου k-means), και η εύρεση του βέλτιστου k μέσω

του αλγόριθμου silhouette,
(3) η καταγραφή της σχέσης των k, r, της τιμής silhouette και του χρόνου συσταδοποίησης για r = N, N^1.5, N(N-1)/2 και όλες τις τιμές T, με εκτέλεση του (2).

Η έξοδος αφορά το (3): πρόκειται για αρχείο κειμένου `experim.dat`, με 7 γραμμές διότι για r=N(N-1)/2 υπάρχει μοναδική επιλογή και το T αχρηστεύεται. Κάθε γραμμή περιέχει τα r, T, k, τιμή silhouette, χρόνο συσταδοποίησης, σε 5 στήλες. Στο README, εξάγετε συμπέρασμα για το βέλτιστο r και T (ως προς την τιμή silhouette) σχετικά με τον χρόνο συσταδοποίησης.

Επιπρόσθετες απαιτήσεις

Όπως στις προηγούμενες εργασίες