Τμήμα Μηχανικών Η/Υ και Πληροφορικής Πανεπιστημίου Ιωαννίνων
MYE047: *Αλγόριθμοι για Δεδομένα Ευρείας Κλίμακας*
Ακαδημαϊκό Έτος 2019-20

Διδάσκων: Σπύρος Κοντογιάννης

2ο Σετ Ασκήσεων: ΚΑΝΟΝΕΣ ΣΥΣΧΕΤΙΣΗΣ & ΡΟΕΣ ΔΕΔΟΜΕΝΩΝ

Ανακοίνωση: Παρασκευή, 5 Μαΐου 2020 **Παράδοση:** Παρασκευή, 5 Ιουνίου 2020

ΤΕΛΕΥΤΑΙΑ ΕΝΗΜΕΡΩΣΗ: Τρίτη, 26 Μαΐου 2020

ΠΕΡΙΓΡΑΦΗ

Στο παρόν συγκεντρώνονται οι πιο συχνές ερωτήσεις φοιτητών σχετικά με την 2η προγραμματιστική ανάθεση για το σπίτι, και οι απαντήσεις τους από τον διδάσκοντα. Το συγκεκριμένο κείμενο θα επικαιροποιείται ανά τακτά χρονικά διαστήματα (δείτε ΗΜΕΡΟΜΗΝΙΑ ΤΕΛΕΥΤΑΙΑΣ ΕΝΗΜΕΡΩΣΗΣ στην κεφαλίδα του κειμένου).

(ΕΡ1) Έχω πρόβλημα μνήμης για την εκτέλεση των αλγορίθμων του ερωτήματος 3β, για καταμέτρηση συχνοτήτων εμφάνισης για ζεύγη αντικειμένων. Τι μπορώ να κάνω?

ΑΠΑΝΤΗΣΗ

Προτείνεται να δοκιμάσετε τους αλγορίθμους κατ' αρχάς στο μικρό σύνολο δεδομένων ratings_10users.csv που είναι ήδη αναρτημένο στη διεύθυνση:

http://www.cse.uoi.gr/~kontog/courses/Algorithms-For-Big-Data/locked-stuff/assignments/lab-2/datasets/ratings_10users.csv

Επίσης, θυμηθείτε ότι ΔΕ ΜΕΤΡΑΜΕ ΤΑΥΤΟΧΡΟΝΑ τις συχνότητες όλων των συνόλων αντικειμένων, σε αυτό το ερώτημα. Άλλωστε, όπως πολλοί από εσάς διαπιστώσατε, είναι πάρα πολύ χρονοβόρο να κάνουμε κάτι τέτοιο. Ασχολούμαστε ΜΟΝΟ με τα ζεύγη αντικειμένων.

Τέλος, για να κατασκευάσουμε διαδοχικά όλα τα ζεύγη που υπάρχουν σε ένα καλάθι Κ ταινιών, εργαζόμαστε ως εξής:

(1) Διατηρούμε τις ταινίες μέσα στα καλάθια μας ταξινομημένες κατ' αύξουσα σειρά (αυτό είναι καλό να εξασφαλιστεί όταν κατασκευάζουμε τα καλάθια μας).
(2) Αρκεί ένα διπλό for-loop ως προς την ταξινομημένη λίστα των ταινιών (ή στο frozenset, αν αποθηκεύουμε ως σύνολο τη λίστα ταινιών) του k-στού καλαθιού:

```
current_basket_size = len(userBaskets[k])
for i in range(0,len(current_basket_size)-1):
    for j in range(i+1,len(current_basket_size))
        …
```

(ΕΡ2) Πώς επεκτείνω την HashedCountersOfPairs για να μετρήσω συχνότητες τριάδων, τετράδων, κλπ. για τις ανάγκες του A-PRIORI?

ΑΠΑΝΤΗΣΗ

Είναι σημαντικό να μη δοκιμάσουμε να κάνουμε σε ένα πέρασμα των καλαθιών ταυτόχρονη καταμέτρηση όλων των συχνών τριάδων, τετράδων, κ.λπ. Θα πρέπει να ακολουθήσουμε τη λογική των διαδοχικών περασμάτων του A-

PRIORI, όπου για κάθε k ≥ 1, καταγράφουμε σε COUNTERS την εμφάνιση ΜΟΝΟ τα (k+1)-σύνολα αντικειμένων που είναι ΥΠΟΨΗΦΙΑ, έχουν δηλαδή προκύψει ως διατεταγμένες κατ' αύξουσα σειρά (k+1)-άδες, από το καρτεσιανό γινόμενο L1 x Lk. Για κάθε υποψήφια (k+1)-άδα, δημιουργούμε (πριν την έναρξη της μέτρησης) στον πίνακα κατακερματισμού πλειάδες της μορφής (item[1] , … , item[k+1] , c=0). Στο τέλος του περάσματος, δημιουργούμε και το σύνολο L_{k+1} που χρειαζόμαστε. Θυμηθείτε ότι θα πρέπει να αποθηκεύετε και τα SUPPORTS (= απόλυτος αριθμός εμφανίσεων) ή τα FREQUENCIES (= ποσοστό εμφανίσεων) των συχνών συνόλων που εντοπίζετε.

Σημειώστε ότι η απλή δημιουργία των πλειάδων συγκεκριμένου μεγέθους είναι μεν ακριβή, αλλά (εφόσον για τις περισσότερες περιπτώσεις απλά δεν κάνουμε τίποτε) όχι απαγορευτική. Πχ, συγκρίνετε τους χρόνους εκτέλεσης στα εξής:

```
>> mycounter = 0
>> for i in range(0,200):
        for j in range(i+1,201):
            mycounter += 1
```

```
>> for i in range(0,200):
        for j in range(i+1,201):
        print("the next pair is (",i,",",j,")")
```

(ΕΡ3) Σχετικά με την αναπαράσταση του τριγωνικού μητρώου μετρητών (στο ερώτημα 3β) με τη μορφή διανύσματος, η θέση του κελιού (i,j) ποια είναι ακριβώς?

ΑΠΑΝΤΗΣΗ

Όπως εξήγησα και στη διάλεξη της Παρασκευής 15/5, στη διαφάνεια #28 για τα συχνά σύνολα αντικειμένων υπήρχε ο ΕΣΦΑΛΜΕΝΟΣ τύπος για τον υπολογισμό της θέσης που αντιστοιχεί στο κελί COUNTERS[i,j] του τριγωνικού μητρώου μετρητών COUNTERS, όταν το αποθηκεύουμε με τη μορφή ενός διανύσματος (γραμμή-γραμμή, και μόνο για τις τιμές 1 ≤ i < j ≤ n, που βρίσκονται επάνω από την κύρια διαγώνιο):

$$pos(i , j) = (i - 1)*(n - i / 2) + j - 1$$

Στο βιβλίο (ορθά) αναφέρεται ο τύπος που πρέπει να χρησιμοποιήσετε (έγινε και η σχετική διόρθωση στις διαφάνειες):

$$\mathbf{pos(i , j) = (i - 1)*(n - i / 2) + j - i}$$

ΜΠΟΝΟΥΣ 2ης ΑΝΑΘΕΣΗΣ: Προσπαθήστε να αποδείξετε ότι πράγματι η σωστή θέση για την τιμή COUNTERS[i,j] είναι η pos(i , j) = (i – 1) (n – i / 2) + j – i, όταν θέλουμε τα στοιχεία του μητρώου COUNTERS να εμφανίζονται γραμμή-γραμμή, και μόνο για τα κελιά που βρίσκονται επάνω από την κύρια διαγώνιο του μητρώου.

(ΕΡ4) Σε τι αποσκοπεί μη μεταβλητή min_length?

ΑΠΑΝΤΗΣΗ

Το min_length είναι μια παράμετρος που αφορά το μέγεθος των itemsets που θέλουμε να στοχεύσουμε (για παραγωγή κανόνων συσχέτισης από αυτά).

Πχ, για min_length = 2, μας ενδιαφέρουν όλοι οι κανόνες (κάθε μη τετριμμένος κανόνας περιλαμβάνει τουλάχιστον 2 ταινίες) που περνούν και τα υπόλοιπα κριτήρια. Για min_length = 3, μας ενδιαφέρουν ΜΟΝΟ οι κανόνες που στο στήριγμά τους (= size of itemset) περιλαμβάνουν τουλάχιστον 3 ταινίες. Δηλαδή, εξαιρούνται οι κανόνες που εμπλέκουν μόνο δύο ταινίες.

ΕΠΙΚΑΙΡΟΠΟΙΗΣΗ ΑΠΑΝΤΗΣΗΣ: Πλέον η μεταβλητή εισόδου min_length έχει αντικατασταθεί από τη μεταβλητή εισόδου max_length, η οποία φράσσει ΑΝΩ το μέγεθος των συχνών συνόλων που αναζητάμε. Πχ, για max_length = 4, αναζητάμε συχνά σύνολα ταινιών (άρα, και τους αντίστοιχους κανόνες) που εμπλέκουν ΤΟ ΠΟΛΥ μέχρι 4 ταινίες.

(ΕΡ5) Σχετικά με την υλοποίηση του sampledApriori (ΒΗΜΑ 5 της ανάθεσης), ζητείται να παρουσιάζουμε τις αλλαγές σε κάθε βήμα που αλλάζει το δείγμα, ή στο τέλος της επεξεργασίας της ροής?

ΑΠΑΝΤΗΣΗ

Για το ζητούμενο στο ΒΗΜΑ 5 της ανάθεσης, θα πρέπει να γίνει ο εξής συνδυασμός του myApriori που υλοποιήσατε με μια διαδικασία τυχαίας δειγματοληψίας καλαθιών από τη ροή των αξιολογήσεων (δηλαδή, γραμμών από το ratings αρχείο που χρησιμοποιείτε) με βάση τον αλγόριθμο reservoirSampling που μελετήσαμε στην τάξη για δειγματοληψία σταθερού μήκους.

Καθώς εξελίσσεται η ροή (δηλαδή, καθώς εμφανίζονται γραμμή-γραμμή οι αξιολογήσεις των χρηστών), ο reservoirSampling αποφασίζει αν θα συμπεριλάβει στο δείγμα έναν ΚΑΙΝΟΥΡΓΙΟ χρήστη current_user, που εμφανίζεται στην επόμενη αξιολόγηση current_assessment, στο δείγμα (για ΠΑΛΙΟΥΣ χρήστες η δειγματοληψία δε χρειάζεται να κάνει κάτι).

- Αν όχι, τότε δε γίνεται τίποτε (δεν αλλάζουν τα καλάθια που βρίσκονται εντός του δείγματος).
- Αν ναι, θα πρέπει να γίνει διαγραφή κάποιου από τα υπάρχοντα καλάθια στο δείγμα, και προσθήκη ενός νέου ζεύγους: SampledBaskets[current_user] = frozenset()
-

Αν ο current_user (ασχέτως αν είναι παλιός ή καινούργιος χρήστης) υπάρχει στο δείγμα, τότε θα πρέπει να γίνει επικαιροποίηση του καλαθιού του με την ταινία current_movie που αφορά η τρέχουσα αξιολόγηση current_assessment: SampledBaskets[current_user].union(frozenset({current_movie}))

Θα πρέπει να δίνεται εξ αρχής η δυνατότητα είτε να αφήσουμε τον αλγόριθμο να επεξεργαστεί ολόκληρη τη ροή (χωρίς να διακόψουμε τη δειγματοληψία), ή να διακόψουμε τη ροή με πάτημα ενός προεπιλεγμένου πλήκτρου. Μετά την ολοκλήρωση (ή τη διακοπή) ανάγνωσης της ροής, εκτελείται ο myApriori μόνο στα καλάθια που τελικά περιλαμβάνονται στο δείγμα. Θα πρέπει επίσης να δίνεται ΕΠΙΛΟΓΗ εκτέλεσης (πλήρους) δεύτερης σάρωσης της ροής, προκειμένου να μετρηθούν οι πραγματικές συχνότητες των συνόλων ταινιών που βρέθηκαν συχνά στο δείγμα, προκειμένου να απορριφθούν τα FALSE-POSITIVES.

(ΕΡ6) Πώς σχετίζεται η κλιμάκωση lift(A→B) με το ενδιαφέρον interest(A→B) που είδαμε στο μάθημα?

ΑΠΑΝΤΗΣΗ

Για να συγκρίνουμε την εμπιστοσύνη confidence(A→B) με την αρχική πεποίθηση Pr(B) = frequency(B) ότι τα αντικείμενα του συνόλου B θα υπάρχουν σε ένα τυχαία επιλεγμένο καλάθι της συλλογής μας, υπάρχουν συνήθως δυο τρόποι:

- Να μελετήσουμε τη διαφορά των δύο ποσοτήτων, δηλαδή το ενδιαφέρον: interest(A→B) = confidene(A→B) – Pr(B). Θετική τιμή του ενδιαφέροντος υπονοεί θετική συσχέτιση, ενώ αρνητική τιμή του ενδιαφέροντος υπονοεί αρνητική συσχέτιση.
- Να μελετήσουμε τον λόγο των δύο ποσοτήτων, δηλαδή την κλιμάκωση: lift(A→B) = confidene(A→B) / Pr(B). Τιμή της κλιμάκωσης μεγαλύτερη του 1 υπονοεί θετική συσχέτιση, ενώ τιμή μικρότερη της μονάδας υπονοεί αρνητική συσχέτιση.

ΜΠΟΝΟΥΣ 2ης ΑΝΑΘΕΣΗΣ: Να εκφραστεί η τιμή interest(A→B) ως συνάρτηση της εμπιστοσύνης και της κλιμάκωσης του συγκεκριμένου κανόνα.