Πανεπιστήμιο Ιωαννίνων,
Τμήμα Πληροφορικής

Εξόρυξη Δεδομένων – ΠΛΕ 059
Χειμερινό Εξάμηνο, 2012

Δεύτερη Σειρά Ασκήσεων

Η σειρά πρέπει να παραδοθεί στην αρχή του μαθήματος της 20ης Νοεμβρίου. Για καθυστερημένες υποβολές ισχύει η πολιτική στην σελίδα του μαθήματος. Οι λεπτομέρειες για το turn-in θα αναρτηθούν στη σελίδα του μαθήματος.

Άσκηση 1 (Απόσταση και ομοιότητα)

1. Ας υποθέσουμε ότι x και y είναι δύο διανύσματα με μήκος (L_2 norm) ίσο με 1. Ποια είναι η σχέση μεταξύ της Ευκλείδειας απόστασης $d(x, y)$ και της cosine ομοιότητας $cos(x, y)$ των δύο διανυσμάτων?
2. Ας υποθέσουμε ότι $X = (x_1, x_2, \dots, x_n)$ και $Y = (y_1, y_2, \dots., y_n)$ είναι δύο διανύσματα μεγέθους n. Το sample Pearson correlation coefficient ορίζεται ως:
$$\rho = \frac{\sum_{i=1}^{n}(x_i - \mu_X)(y_i - \mu_Y)}{\sqrt{\sum_{i=1}^{n}(x_i - \mu_X)^2}\sqrt{\sum_{i=1}^{n}(y_i - \mu_Y)^2}}$$
όπου μ_X, μ_Y είναι η μέση τιμή του διανύσματος X και Y αντίστοιχα.
Περιγράψτε πως μπορούμε να χρησιμοποιήσουμε το cosine similarity για να υπολογίσουμε το sample Pearson correlation coefficient.
3. Έστω $U = \{u_1, \dots, u_n\}$ ένα σύνολο από n αντικείμενα, και R μία διάταξη (ιεράρχηση -- ranking) των αντικειμένων. Ορίστε μια «λογική» συνάρτηση απόστασης $d(R_1, R_2)$ μεταξύ δύο διατάξεων των στοιχείων στο U, η οποία να ορίζει μια μετρική. Αποδείξτε ότι η συνάρτηση σας ορίζει μία μετρική. Η συνάρτηση σας θα πρέπει να είναι τέτοια ώστε να παίρνει την μέγιστη τιμή όταν η μία διάταξη είναι η αντίστροφη της άλλης.
4. Έστω $X = (x_1, x_2, \dots, x_n)$ ένα διάνυσμα πραγματικών αριθμών. Προς χάριν απλότητας υποθέστε ότι οι συνιστώσες του διανύσματος είναι σε φθίνουσα σειρά ως προς την απόλυτη τιμή τους, δηλαδή $|x_1| \geq |x_2| \geq \cdots \geq |x_n|$. Το p-norm του διανύσματος X ορίζεται ως $\|X\|_p = (|x_1|^p + \cdots + |x_n|^p)^{1/p}$. Αποδείξτε ότι όταν $p \to \infty$, $\|X\|_p \to |x_1|$.

Άσκηση 2 (Min-Hashing)

Κάνετε την **Άσκηση 3.3.2** από το βιβλίο Mining Massive Datasets των Anand Rajaraman and Jeff Ullman. Παραδώστε τα ενδιάμεσα αποτελέσματα του υπολογισμού της Min-Hashing υπογραφής (όπως κάναμε στην τάξη και όπως κάνουν στο βιβλίο), την τελική υπογραφή και με τις τέσσερεις συναρτήσεις, και την εκτιμώμενη και πραγματική ομοιότητα μεταξύ όλων των ζευγών από στήλες.

Άσκηση 3 (Min-Hashing και LSH)

Σε αυτή την άσκηση θα υλοποιήσετε το Min-Hashing και το Locality Sensitive Hashing. Θα τεστάρετε την υλοποίηση σας στο MovieLens 100k dataset το οποίο αποτελείται από 943 χρήστες που έχουν βαθμολογήσει 1682 ταινίες. Μπορείτε να κατεβάσετε τα δεδομένα από τη σελίδα http://www.grouplens.org/node/73 . Διαβάστε το Readme αρχείο για λεπτομέρειες και επεξεργαστείτε

τα δεδομένα όπως χρειάζεστε. Για την άσκηση μας ενδιαφέρει μόνο το *σύνολο από ταινίες* που βαθμολόγησε ο χρήστης και όχι οι βαθμοί. Θέλουμε να υπολογίσουμε τη Jaccard ομοιότητα μεταξύ των χρηστών.

Υπολογίστε την ακριβή Jaccard ομοιότητα μεταξύ όλων των ζευγαριών χρηστών, και τυπώστε τα ζεύγη με ομοιότητα τουλάχιστον 0.5. Μετά υπολογίστε τα min-hash signatures για όλους τους χρήστες και την προσεγγιστική Jaccard ομοιότητα όπως περιγράψαμε στην τάξη. Χρησιμοποιήστε 50, 100, και 200 hash functions. Για κάθε τιμή, τυπώστε τα ζεύγη με προσεγγιστική ομοιότητα τουλάχιστον 0.5, και τον αριθμό των false positives και false negatives. Για τα false positives και false negatives αναφέρετε τη μέση τιμή από 5 διαφορετικά τρεξίματα.

Μετά, σπάσετε τον πίνακα με τα signatures σε b bands με r hash functions ανά band, όπως περιγράψαμε στην τάξη, και υλοποιήστε το Locality Sensitive Hashing. Ο στόχος είναι να βρούμε υποψήφια ζεύγη με ομοιότητα τουλάχιστον 0.6. Πειραματιστείτε με r=5, b =10 για τον πίνακα με τα 50 hash functions, r=5, b=20 για τον πίνακα με τα 100 hash functions, r=5, b= 40, και r=10, b=20 για τον πίνακα με τα 200 hash functions. Αναφέρετε τον αριθμό των false positives και false negatives παίρνοντας την μέση τιμή σε 5 τρεξίματα. Πως αλλάζουν αυτά τα νούμερα αν έχουμε όριο ομοιότητας 0.8? Ποιο είναι το κατώφλι της σιγμοειδούς συνάρτησης ?

Κάνετε turn in τον κώδικα σας, και τα αρχεία εξόδου για τα διαφορετικά τρεξίματα. Επίσης κάνετε turn-in ένα αρχείο με τις μέσες τιμές και μια αναφορά με τις παρατηρήσεις σας.

Τεχνικές Λεπτομέρειες

1. Για το pre-processing των δεδομένων, οι παρακάτω unix εντολές μπορεί να σας φανούν χρήσιμες:
    a. cut: επιτρέπει να πάρουμε συγκεκριμένες κολώνες από ένα αρχείο με διαχωριζόμενες τιμές
    b. sort: ταξινομεί τις γραμμές ενός αρχείου σε αλφαβητική σειρά . –n for αριθμητική σειρά
    c. uniq: αφαιρεί συνεχόμενες γραμμές που είναι ίδιες.
2. Χρησιμοποιείστε την παρακάτω hash function για τα signatures:
    Διαλέξτε ένα αρκετά μεγάλο πρώτο αριθμό R (π.χ.., R = 131071);
    Διαλέξτε a,b,c, τυχαίους αριθμούς στο διάστημα [0,R]
    h(x) = (a*(x>>4) +b*x +c)%R;
3. Για την υλοποίηση του LSH δεν χρειάζεται να υλοποιήσετε δικό σας hash table ή list. Χρησιμοποιείστε υπάρχουσες υλοποιήσεις που έρχονται με τη γλώσσα που χρησιμοποιείτε (π.χ., στην C++ υπάρχει η STL υλοποίηση, κοιτάξτε την σελίδα του μαθήματος του αντικειμενοστρεφή προγραμματισμού για λεπτομέρειες)
4. Αν και δεν είναι αποτελεσματικό, προς χάριν απλότητας μπορείτε να δημιουργήσετε τον πλήρη 0/1 πίνακα χρηστών και ταινιών. Θα πάρετε bonus βαθμούς για μια πιο αποτελεσματική υλοποίηση.