Πανεπιστήμιο Ιωαννίνων,
Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Εξόρυξη Δεδομένων – ΜΥΕ012/ΠΛΕ 059
Χειμερινό Εξάμηνο, 2020

Δεύτερη Σειρά Ασκήσεων

Η προθεσμία για την δεύτερη σειρά ασκήσεων είναι την Τετάρτη 23 Δεκεμβρίου μέχρι το τέλος της ημέρας. Για τις πρώτη άσκηση συνίσταται θερμά να παραδώσετε κάποιο pdf με τους υπολογισμούς, αλλά αν δεν γίνεται, μπορείτε να παραδώσετε φωτογραφίες των γραπτών απαντήσεων σας. Για τις υπόλοιπες ασκήσεις παραδώσετε το notebook με τον κώδικα και τα αποτελέσματα σας, και την αναφορά με τον σχολιασμό. Παραδώστε το notebook και σε ipynb και σε html μορφή. Για καθυστερημένες υποβολές ισχύει η πολιτική στην σελίδα του μαθήματος. Αν χρησιμοποιήσετε free passes θα πρέπει να το αναφέρετε στην αναφορά. Θα αφαιρεθούν από όλα τα μέλη της ομάδας. Αν κάποιο μέλος της ομάδας έχει χρησιμοποιήσει όλα τα free passes του θα χάσει το ποσοστό που αναλογεί. Λεπτομέρειες για το turn-in, και για το πώς να γράφετε αναφορές είναι στη σελίδα Ασκήσεις του μαθήματος. Οι ασκήσεις μπορούν να γίνουν σε ομάδες μέχρι δύο ατόμων.

Ερώτηση 1

Μία εκθετική κατανομή ορίζεται με συνάρτηση πυκνότητας πιθανότητας $f(x) = \lambda e^{-\lambda x}$, για $x \geq 0$, όπου λ είναι η παράμετρος της κατανομής. Σας δίνεται ένα σύνολο από παρατηρήσεις $X = \{x_1, \dots, x_n\}, x_i \geq 0$, που έχουν παραχθεί από μία εκθετική κατανομή. Χρησιμοποιήστε την Maximum Likelihood Estimation τεχνική που περιγράψαμε στην τάξη για να βρείτε την παράμετρο της κατανομής που ταιριάζει (fits) τα δεδομένα των παρατηρήσεων.

Ερώτηση 2 (Συστήματα συστάσεων)

Ο στόχος αυτής της άσκησης είναι να πειραματιστείτε με αλγορίθμους για συστήματα συστάσεων και να εξασκηθείτε στην διαχείριση πινάκων μέσα από τις βιβλιοθήκες που έχουμε μάθει.

Θα χρησιμοποιήσετε το Yelp dataset που μπορείτε να κατεβάσετε από εδώ. Σε αυτή την άσκηση θα χρησιμοποιήσετε τα αρχεία `yelp_academic_dataset_business.json` και `yelp_academic_dataset_review.json` (το τελευταίο είναι πάνω από 6GB οπότε θα χρειαστείτε χώρο, και πρέπει να το λάβετε υπόψιν σας κατά την επεξεργασία). Η άσκηση θα γίνει σε βήματα:

Βήμα 1: Χρησιμοποιώντας τα παραπάνω αρχεία θα δημιουργήσετε ένα user-business πίνακα με τα ratings των χρηστών για επιχειρήσεις στην πόλη του “Toronto”. Θα κρατήσετε χρήστες και επιχειρήσεις με αρκετά reviews. Συγκεκριμένα, στα τελικά σας δεδομένα θα έχετε ένα σύνολο από χρήστες U, και ένα σύνολο από επιχειρήσεις B, όπου ο κάθε χρήστης στο U θα έχει τουλάχιστον 15 reviews σε επιχειρήσεις στο B, και η κάθε επιχείρηση στο B θα έχει τουλάχιστον 15 reviews από χρήστες στο U.

Βήμα 2: Στο βήμα αυτό θα χρησιμοποιήσετε τα δεδομένα από το Βήμα 1 για να δημιουργήστε έναν αραιό πίνακα R που κρατάει τον πίνακα με τις βαθμολογίες. Το αποτέλεσμα του Βήματος 1 σας δίνεται στο αρχείο pruned_data.csv σε μορφή τριάδων (user id, business id, rating). Μπορείτε να το χρησιμοποιήσετε ακόμη και αν δεν έχετε ολοκληρώσει το Βήμα 1.

Στη συνέχεια αφαιρέσετε τυχαία το 5% των ratings από τον πίνακα R. Αυτά είναι τα ratings που θέλουμε να προβλέψουμε οπότε θα κρατήσετε την θέση τους και την τιμή τους. Ο πίνακας R με τα ratings δεν θα περιέχει πλέον τις τιμές που αφαιρέσατε.

Βήμα 3: Υλοποιήστε τον αλγόριθμο **User-Based Collaborative Filtering (UCF)**. Ο αλγόριθμος έχει μια παράμετρο k, που είναι ο αριθμός των όμοιων χρηστών που κοιτάει. Για να υπολογίσετε την τιμή ενός κελιού (u, b) υπολογίστε το σύνολο $N_k(u, b)$ με τους k πιο όμοιους χρήστες με τον χρήστη u οι οποίοι έχουν βαθμολογήσει την επιχείρηση b. Στη συνέχεια χρησιμοποιήστε την εξής εξίσωση για την πρόβλεψη σας:

$$p(u,b) = \frac{\sum_{u' \in N_k(u,b)} s(u,u') r(u',b)}{\sum_{u' \in N_k(u,b)} s(u,u')}$$

Στην εξίσωση $s(u, u')$ είναι η ομοιότητα μεταξύ των χρηστών u και u'. Για την υλοποίηση σας θα χρησιμοποιήσετε το cosine similarity.

Η υλοποίηση έχει τα εξής βήματα:

- Υπολογίστε τον πίνακα με τις ομοιότητες μεταξύ των χρηστών (χρησιμοποιήστε έτοιμη βιβλιοθήκη γι αυτό είναι πολύ πιο γρήγορη)
- Για κάθε ζευγάρι (u, b) το οποίο έχετε αφαιρέσει:
    1. Βρείτε τους χρήστες που έχουν βαθμολογήσει το b
    2. Πάρετε την ομοιότητα αυτών των χρηστών με τον χρήστη u και κρατήστε τους k πιο όμοιους χρήστες
    3. Φτιάξετε δύο διανύσματα, ένα με τις ομοιότητες και ένα με τις βαθμολογίες για τους k πιο όμοιους χρήστες
    4. Υπολογίστε την βαθμολογία με την παραπάνω εξίσωση. Ο υπολογισμός μπορεί να γίνει με πράξεις διανυσμάτων.

Για να πάρετε όλους του βαθμούς η υλοποίηση των βημάτων 1-4 θα πρέπει να γίνει χρησιμοποιώντας μόνο μεθόδους διαχείρισης πινάκων και διανυσμάτων της numpy ή scipy.

Βήμα 4: Υλοποιήστε τον αλγόριθμο **Item-Based Collaborative Filtering (ICF)**. Ο αλγόριθμος είναι ουσιαστικά ο ίδιος με αυτόν στο Βήμα 3, απλά δουλεύετε με τον ανάστροφο πίνακα και ανταλλάσσετε χρήστες και επιχειρήσεις και ανάποδα. Παρακάτω είναι η περιγραφή του για πληρότητα:

Ο αλγόριθμος έχει μια παράμετρο k , που είναι ο αριθμός των όμοιων επιχειρήσεων που θα κοιτάξει. Για να υπολογίσετε την τιμή ενός κελιού (u, b) υπολογίστε το σύνολο $N_k(b, u)$ με τις k πιο όμοιες επιχειρήσεις ως προς την b από αυτές που έχει βαθμολογήσει ο χρήστης u. Στη συνέχεια χρησιμοποιήστε την εξής εξίσωση για την πρόβλεψη σας:

$$p(u,b) = \frac{\sum_{b' \in N_k(b,u)} s(b,b') r(u,b')}{\sum_{b' \in N_k(b,u)} s(b,b')}$$

Στην εξίσωση $s(b, b')$ είναι η ομοιότητα μεταξύ των επιχειρήσεων b και b'. Για την υλοποίηση σας θα χρησιμοποιήσετε το cosine similarity.

Η υλοποίηση έχει τα εξής βήματα:

- Υπολογίστε τον πίνακα με τις ομοιότητες μεταξύ των επιχειρήσεων (χρησιμοποιήστε έτοιμη βιβλιοθήκη γι αυτό είναι πολύ πιο γρήγορη)
- Για κάθε ζευγάρι (u, b) το οποίο έχετε αφαιρέσει:
    5. Βρείτε τις επιχειρήσεις που έχει βαθμολογήσει ο u
    6. Πάρετε την ομοιότητα αυτών των επιχειρήσεων με την επιχείρηση b και κρατήστε τις k πιο όμοιες επιχειρήσεις
    7. Φτιάξετε δύο διανύσματα, ένα με τις ομοιότητες και ένα με τις βαθμολογίες για τις k πιο όμοιες επιχειρήσεις
    8. Υπολογίστε την βαθμολογία με την παραπάνω εξίσωση. Ο υπολογισμός μπορεί να γίνει με πράξεις διανυσμάτων.

Για να πάρετε όλους του βαθμούς η υλοποίηση των βημάτων 1-4 θα πρέπει να γίνει χρησιμοποιώντας μόνο μεθόδους διαχείρισης πινάκων και διανυσμάτων της numpy ή scipy.

Βήμα 5: Εφαρμόστε το **Singular Value Decomposition (SVD)** στον πίνακα R, και κρατήστε τα k μεγαλύτερα singular vectors για να πάρετε ένα rank-k πίνακα R_k. Στη συνέχεια χρησιμοποιήστε την τιμή $p(u, b) = R_k(u, b)$ για την πρόβλεψη σας. Αν η τιμή γίνει μικρότερη του 0, ή μεγαλύτερη του 5, στρογγυλοποιείστε στο 0 ή το 5.

Βήμα 6: Αξιολογήστε τους αλγορίθμους σας. Για την αξιολόγηση θα χρησιμοποιήσετε την RMSE (Root Mean Square Error) μετρική. Αν $r_1, r_2, \dots, r_n$ είναι τα ratings που θέλουμε να προβλέψουμε, και $p_1, p_2, \dots, p_n$είναι οι προβλέψεις του αλγορίθμου, το RMSE του αλγορίθμου ορίζεται ως

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - p_i)^2}$$

Δημιουργήστε γραφικές παραστάσεις με το RMSE για διαφορετικές τιμές του k για όλους τους αλγορίθμους. Για τον αλγόριθμο UCF χρησιμοποιείστε τις τιμές [1,5,10,20,50,100,200,500,1000]. Για τον αλγόριθμο ICF χρησιμοποιείστε τις τιμές [1,5,10,20,40,50,60,70,80,100]. Για τον αλγόριθμο SVD χρησιμοποιείστε τις τιμές [1,5,10,20,30,40,50,75,100]. Βρείτε το k με το χαμηλότερο error. Μπορείτε να «ζουμάρετε» σε κάποιο διάστημα για να εξερευνήσετε επιπλέον τιμές του k.

Θα συγκρίνετε επίσης με τα παρακάτω απλά “baselines”:

1. **User Average (UA):** Χρησιμοποιήστε την μέση τιμή $\overline{r(u)}$ των ratings του u για την πρόβλεψη.
2. **Business Average (BA):** Χρησιμοποιήστε την μέση τιμή $\overline{r(b)}$ των ratings της επιχείρησης b για την πρόβλεψη.

Φτιάξετε ένα πίνακα που να έχει όλους τους αλγορίθμους μαζί, και το καλύτερο error που επιτυγχάνει ο κάθε αλγόριθμος, και σχολιάστε τα αποτελέσματα.

Παραδώσετε ένα notebook με τον κώδικα σας για κάθε βήμα και μια αναφορά με τις παρατηρήσεις σας για τα αποτελέσματα.

Bonus: Υλοποιήσετε και τεστάρετε και την παραλλαγή του UCF που προβλέπει τις αποκλίσεις από την μέση τιμή. Στην περίπτωση αυτή , θα χρησιμοποιήστε την εξής εξίσωση για την πρόβλεψη σας:

$$p(u,b) = \overline{r(u)} + \frac{\sum_{u' \in N_k(u,b)} s(u,u')(r(u',b) - \overline{r(u')})}{\sum_{u' \in N_k(u,b)} s(u,u')}$$

Για την ομοιότητα χρησιμοποιήστε το correlation coefficient (το cosine similarity, μετά την αφαίρεση της μέσης τιμής από την κάθε γραμμή). Αν η τιμή γίνει μικρότερη του 0, ή μεγαλύτερη του 5, στρογγυλοποιείστε στο 0 ή το 5.

Υποδείξεις

- Για να διαβάσετε τα αρχεία ανοίξτε τα με τις παραμέτρους encoding = 'utf8', errors = 'ignore'.
- Το αρχείο με τα reviews είναι πολύ μεγάλο και άρα δεν μπορείτε να το φορτώσετε στην μνήμη. Επίσης, ο αριθμός των user/business ζευγών είναι μεγάλος και για να τον διαχειριστείτε θα πρέπει να δημιουργήσετε κατάλληλες δομές.
- Θα σας είναι χρήσιμη η μέθοδος nonzero των αραιών πινάκων
- Οι πράξεις μεταξύ πινάκων και διανυσμάτων με τη βιβλιοθήκη numpy μερικές φορές επιστρέφουν διανύσματα που μπορεί να έχουν διαφορετική μορφή απ ότι θα θέλατε οπότε θα πρέπει να είσαστε προσεκτικοί. Η μέθοδος reshape μπορεί να σας βοηθήσει.

Ερώτηση 3 (Clustering)

Στην ερώτηση αυτή θα εξασκηθείτε με την εφαρμογή αλγορίθμων ομαδοποίησης (clustering). Θα χρησιμοποιήσετε και πάλι τα δεδομένα από το Yelp και συγκεκριμένα τις επιχειρήσεις από το Τορόντο. Ο στόχος μας είναι να τις ομαδοποιήσουμε χρησιμοποιώντας το κείμενο των κριτικών που δέχονται. Για να αξιολογήσουμε το clustering θα χρησιμοποιήσουμε την κατηγορία της επιχείρησης.

Από το αρχείο `yelp_academic_dataset_business.json` κρατήστε τις επιχειρήσεις από την πόλη του Τορόντο που έχουν την κατηγορία “Beauty & Spas”, “Shopping” και “Bars” μέσα στις κατηγορίες τους (για επιχειρήσεις που έχουν και τις τρεις κατηγορίες, αναθέσετε την επιχείρηση σε μία κατηγορία με παραπάνω σειρά προτεραιότητας). Κρατήστε τις επιχειρήσεις που έχουν τουλάχιστον 10 reviews. Για κάθε επιχείρηση στην λίστα σας, πάρετε όλα τα reviews για την επιχείρηση από το αρχείο yelp_academic_dataset_review.json και ενώστε τα σε ένα μεγάλο κείμενο για την επιχείρηση. Χρησιμοποιείστε τα κείμενα που δημιουργήσατε για τις επιχειρήσεις για να πάρετε την tf-idf αναπαράσταση των επιχειρήσεων (χρησιμοποιήσετε την έτοιμη βιβλιοθήκη της python - μπορείτε επίσης να κάνετε επιπλέον επιλογές για τις παραμέτρους της βιβλιοθήκης). Χρησιμοποιώντας αυτά τα δεδομένα θα κάνετε τα εξής πειράματα:

1. Κάνετε cluster τις επιχειρήσεις χρησιμοποιώντας k-means και agglomerative clustering με τρία (3) clusters, ο αριθμός των κατηγοριών που επιλέξαμε. Για το agglomerative clustering δοκιμάστε single-link, complete-link, average και ward. Εξετάστε αν τα clusters που βρίσκετε αντιστοιχούν στις κατηγορίες χρησιμοποιώντας τον πίνακα σύγχυσης και τις μετρικές precision και recall και ανά cluster και συνολικά.

Για τον υπολογισμό του precision/recall αντιστοιχίσετε το κάθε cluster σε μία κλάση όπως κάναμε στο φροντιστήριο. Σχολιάστε τα αποτελέσματα.

2. Είναι πιθανό τα κείμενα να μην γίνονται clustered με τρόπο που συμφωνεί με τις κατηγορίες. Χρησιμοποιήστε τον k-means αλγόριθμο και δημιουργήστε το silhouette plot για να αποφασίσετε για τον αριθμό των clusters. Γι αυτή την τιμή του k τρέξτε τον k-means και δημιουργείστε και πάλι το confusion matrix. Σχολιάστε τα αποτελέσματα σας.
3. Στο κομμάτι αυτό θα κάνετε μια χειρωνακτική αξιολόγηση των αποτελεσμάτων του k-means. Στο clustering που κάνατε στο (1) βλέπουμε ότι έχουμε κάποια λάθη γιατί κάποιες επιχειρήσεις από το Beauty & Spa, πάνε στο cluster που αντιστοιχεί στο Shopping. Σχεδιάσετε ένα πείραμα για να εξετάσετε αυτές τις επιχειρήσεις και να εξηγήσετε αυτά τα λάθη (π.χ., μπορείτε να δείτε τις σημαντικές λέξεις γι αυτές τις επιχειρήσεις, να δείτε τις κατηγορίες τους, να εξετάσετε κάποιες επιλεγμένες τυχαία, να εξετάσετε κάποιες με βάσει το πόσο κοντά είναι στο centroid του cluster, κλπ). Αντίστοιχα για το clustering που κάνατε στο (2) θα δείτε ότι το κύριο αποτέλεσμα των επιπλέον clusters είναι ότι κάποια από τα αρχικά clusters φαινεται να σπάνε στα δύο. Χρησιμοποιείστε τα centroids των clusters για να εξερευνήσετε γιατί συμβαίνει αυτό.

Παραδώσετε ένα notebook με τον κώδικα σας και την αναφορά με τον σχολιασμό και ανάλυση των αποτελεσμάτων σας.