Δεύτερη Σειρά Ασκήσεων

Η προθεσμία για την δεύτερη σειρά ασκήσεων είναι την Πέμπτη 12 Ιανουαρίου 11:55 μ.μ. Παραδώστε Notebooks με τον κώδικα και τις αναφορές. Για την Ερώτηση 1 μπορείτε να παραδώσετε και pdf με την απόδειξη, ή φωτογραφίες από χειρόγραφα. Για καθυστερημένες υποβολές ισχύει η πολιτική στην σελίδα του μαθήματος. Η παράδοση θα γίνει μέσω του ecourse. Λεπτομέρειες στη σελίδα Ασκήσεις του μαθήματος. Η άσκηση είναι **ατομική**.

Ερώτηση 1

Α. 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.

Β. Υποθέστε ότι οι παρατηρήσεις $X = \{x_1, \dots, x_n\}$ έχουν παραχθεί από ένα μείγμα εκθετικών κατανομών, L_1, L_2, με παραμέτρους λ_1, λ_2, και πιθανότητες μίξης (mixture probabilities) π_1, π_2. Θα χρησιμοποιήσετε τον EM αλγόριθμο για να υπολογίσουμε τις παραμέτρους $\Theta = (\lambda_1, \lambda_2, \pi_1, \pi_2)$ του mixture μοντέλου, όπως κάναμε και για την περίπτωση της μίξης από Gaussian κατανομές. Στο Μ βήμα, υποθέτουμε ότι έχουμε τις πιθανότητες ανάθεσης $P(L_k|x_i)$, για $k = 1,2$ και $i = 1, \dots, n$, και θέλουμε να υπολογίσουμε τις παραμέτρους Θ. Δώστε τις εξισώσεις για τα $\lambda_1, \lambda_2, \pi_1, \pi_2$ και τους υπολογισμούς με τους οποίους τις παρήγατε.

Υπόδειξη: Θα σας βοηθήσει να διαβάσετε τις σημειώσεις του Άρη Αναγνωστόπουλου που είναι στην σελίδα του μαθήματος για την περίπτωση της μίξης των Gaussians, τις οποίες παρουσιάσαμε στο μάθημα.

Ερώτηση 2

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

Θα χρησιμοποιήσετε τα δεδομένα από το Yelp που χρησιμοποιήσατε και στην πρώτη σειρά ασκήσεων. Θα εξετάσουμε χρήστες οι οποίοι βαθμολογούν τις επιχειρήσεις στην Φιλαδέλφεια. Μπορείτε να κατεβάσετε τις τριάδες user_id, business_id, stars, από το αρχείο phily_users_businesses_stars.csv στη σελίδα του μαθήματος.

Βήμα 1: Το πρώτο βήμα της άσκησης είναι η επεξεργασία των δεδομένων. Θα κρατήσουμε μόνο χρήστες που έχουν βαθμολογήσει τουλάχιστον 15 επιχειρήσεις και επιχειρήσεις που έχουν βαθμολογηθεί από τουλάχιστον 15 χρήστες. Υλοποιήσετε ένα επαναληπτικό κλάδεμα χρηστών και ταινιών μέχρι να ικανοποιείται αυτή η συνθήκη. Μπορείτε να υλοποιήσετε το κλάδεμα όπως θέλετε (με pandas, ή με απλές δομές). Η υλοποίηση σας θα πρέπει να είναι αποδοτική, άρα θα πρέπει να χρησιμοποιήσετε τις σωστές δομές και μεθόδους. Το αποτέλεσμα είναι 302141 τριάδες με N = 6961 χρήστες και M = 4064 επιχειρήσεις. Σας δίνεται το αποτέλεσμα

αυτής της επαναληπτικής διαδικασίας στο αρχείο phily_users_businesses_stars_pruned.csv στη σελίδα του μαθήματος.

Βήμα 2: Σε αυτό το βήμα θα δημιουργήσετε τα train και test δεδομένα. Από το αποτέλεσμα του προηγούμενου βήματος διαλέξτε τυχαία ένα 10% των τριάδων. Αυτά θα είναι το test set. Τα υπόλοιπα 90% θα είναι το training set. Υλοποιήστε την τυχαία επιλογή όπως προτιμάτε. Για το αποτέλεσμα της επιλογής θα χρησιμοποιήσετε τα έτοιμα αρχεία phily_users_businesses_stars_train.csv και phily_users_businesses_stars_test.csv, ώστε όλοι να δουλεύουμε με τα ίδια δεδομένα.

Στα επόμενα βήματα θα υλοποιήσετε και θα τεστάρετε διαφορετικούς αλγόριθμους συστάσεων. Ο στόχος μας είναι για κάθε ζευγάρι χρήστη-επιχείρησης (u, b) στα test δεδομένα να υπολογίσουμε ένα score που είναι όσο πιο κοντά γίνεται στην πραγματική βαθμολογία του χρήστη για την επιχείρηση. Για την αξιολόγηση θα χρησιμοποιήσετε το RMSE (Root Mean Square Error). 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 ενός αλγορίθμου, το RMSE του αλγορίθμου ορίζεται ως

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

Βήμα 3: Οι δύο πρώτοι αλγόριθμοι που θα δοκιμάσουμε είναι ο **User Agerage (UA)** και ο **Business Average (BA)**. Ο πρώτος για κάθε ζευγάρι (u, b) προβλέπει τη μέση βαθμολογία του χρήστη u, ενώ ο δεύτερος τη μέση βαθμολογία της επιχείρησης b. Η μέση βαθμολογία του χρήστη υπολογίζεται μόνο από τις επιχειρήσεις τις οποίες έχει βαθμολογήσει. Αντίστοιχα η μέση βαθμολογία μιας επιχείρησης υπολογίζεται μόνο από τους χρήστες που την έχουν βαθμολογήσει. Για το βήμα αυτό σας προτείνεται να κρατήσετε τα δεδομένα σε dataframes. Για να πάρετε όλους τους βαθμούς της άσκησης η υλοποίηση σας θα πρέπει να γίνει χρησιμοποιώντας εντολές της βιβλιοθήκης pandas, χωρίς να διατρέξετε τις γραμμές του dataframe με for loop. Υπολογίστε το RMSE για τους δύο αλγορίθμους και κάνετε ένα σύντομο σχόλιο για το τι βλέπετε.

Βήμα 4: Ο επόμενος αλγόριθμος που θα δοκιμάσετε είναι ο **Singular Value Decomposition (SVD)**. Για την υλοποίηση του αλγορίθμου θα φορτώσετε το train dataset σε ένα αραιό user-business πίνακα R. Για τον σκοπό αυτό θα χρειαστεί να κάνετε μια αντιστοίχιση των user ids στα νούμερα από 0 εώς N-1 και αντίστοιχα για τα business ids. Στον αλγόριθμο αυτό υπολογίζουμε ένα rank-k R_k approximation του πίνακα R, και για το ζεύγος (u, b) η πρόβλεψη είναι η τιμή $R_k[u, b]$.

Υπολογίστε το Singular Value Decomposition για $K = 100$. Στη συνέχεια για διαφορετικά $k \in [1, K]$ θα υπολογίσετε τις τιμές $R_k[u, b]$ για τα ζευγάρια (u, b). Δεν θα δημιουργήσετε τον πίνακα R_k ο οποίος είναι πυκνός. Η τιμή $R_k[u, b]$ μπορεί να υπολογιστεί ως

$$R_k[u, b] = U_k[u, :]\Sigma_k V_k^T[:, b]$$

Όπου $U_k[u, :]$ είναι η γραμμή u του πίνακα U_k με τα k πρώτα left singular vectors, Σ_k είναι ο διαγώνιος πίνακας με τα k πρώτα singular values και $V_k^T[:, b]$ είναι η στήλη b του ανάστροφου πίνακα V_k με τα k πρώτα right singular vectors.

Υπολογίστε το RMSE για κάθε $k \in [1, K]$ και κάνετε μια γραφική παράσταση του RMSE ως προς το k. Αναφέρετε την τιμή του k που ο αλγόριθμος πετυχαίνει το ελάχιστον RMSE και ποιο είναι αυτό. Τι παρατηρείτε για το RMSE? Δώστε μια πιθανή εξήγηση.

Σημείωση: Σε κανένα σημείο δεν θα πρέπει να δημιουργήσετε ένα πυκνό πίνακα $N \times M$. Αν η βαθμολογία που υπολογίζει ο αλγόριθμος είναι μεγαλύτερη από 5 ή μικρότερη από μηδέν κάνετε "clip" την βαθμολογία σε 5 και 0 αντίστοιχα.

Βήμα 5: Στη συνέχεια θα υλοποιήσετε τον **User-Based Collaborative Filtering (UCF).** Για το σκοπό αυτό θα δημιουργήσετε πάλι ένα αραιό user-business πίνακα R όπου τώρα όμως το κελί $R[u, b]$ θα κρατάει την βαθμολογία του χρήστη u στην επιχείρηση b, μείον την μέση βαθμολογία του u, όπως την υπολογίσατε στον UA αλγόριθμο. Η δημιουργία των δεδομένων για τον πίνακα μπορεί να γίνει σε pandas ή όπως αλλιώς θέλετε.

Ο UCF αλγόριθμος έχει μια παράμετρο k, που είναι ο αριθμός των όμοιων χρηστών που κοιτάει. Για να υπολογίσετε την τιμή για το ζευγάρι (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:

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

Στην εξίσωση $\overline{r}_u$ είναι η μέση τιμή των βαθμολογιών του u και $s(u, u')$ είναι η ομοιότητα μεταξύ των χρηστών u και u'. Για την υλοποίηση σας θα χρησιμοποιήσετε το cosine similarity.

Η βασική ιδέα της υλοποίησης είναι η ακόλουθη:

Για κάθε ζευγάρι (u, b) στα test δεδομένα:

1. Βρείτε τους χρήστες που έχουν βαθμολογήσει την επιχείρηση b.
2. Υπολογίστε την ομοιότητα αυτών των χρηστών με τον χρήστη u και κρατήστε τους k πιο όμοιους χρήστες. Αν υπάρχουν λιγότεροι από k χρήστες που έχουν βαθμολογήσει επιχείρηση b, χρησιμοποιήστε τους όλους.
3. Χρησιμοποιείστε τα διανύσματα με τις ομοιότητες και τις βαθμολογίες για τους k πιο όμοιους χρήστες, και υπολογίστε την βαθμολογία με την παραπάνω εξίσωση.

Όπως και στο Βήμα 3, αν η βαθμολογία είναι πάνω από 5 ή κάτω από 0 θα την κάνετε "clip" στο 5 και το 0 αντίστοιχα. Επίσης προσέξτε την περίπτωση που μετά την αφαίρεση της μέσης τιμής όλες οι βαθμολογίες γίνουν μηδέν. Στην περίπτωση αυτή θα επιστρέφετε απλά τη μέση βαθμολογία του χρήστη.

Δημιουργήστε μια συνάρτηση η οποία για ένα ζευγάρι (u, b), για μια λίστα από τις τιμές για το k υπολογίζει τις βαθμολογίες για το (u, b) για όλες τις τιμές του k. Για αρχή, μπορείτε να υπολογίσετε μόνο για ένα συγκεκριμένο k, και μετά να την επεκτείνετε για μια λίστα με τιμές για το k. Η σωστή υλοποίηση της συνάρτησης αυτής είναι ένα μεγάλο κομμάτι αυτής της υποερώτησης. Για να πάρετε όλους τους βαθμούς της άσκησης, η συνάρτηση αυτή πρέπει να υλοποιηθεί χρησιμοποιώντας μόνο μεθόδους διαχείρισης πινάκων και διανυσμάτων της numpy ή scipy. Σε κανένα σημείο του προγράμματος δεν πρέπει να υπολογίσετε κάποιο πυκνό πίνακα $N \times M$, και σε

κανένα σημείο δεν πρέπει να χρησιμοποιήσετε for loop για να διατρέξετε τους χρήστες ή τα businesses στα training data.

Τρέξτε τον αλγόριθμο για $k = 1,2,3,5,8,10,12,15,20,25,30,40,50$, και φτιάξετε μια γραφική παράσταση με το RMSE ως συνάρτηση του k. Δώστε την τιμή η οποία δίνει τα καλύτερα αποτελέσματα, και σχολιάστε τα αποτελέσαμτα. Μπορείτε να δοκιμάσετε περισσότερες τιμές αν θέλετε. Αν έχετε υλοποιήσει τη μέθοδο που υπολογίζει score για όλες τις τιμές του k, ένα τρέξιμο του αλγορίθμου θα πάρει 2-3 λεπτά. Είναι πιο αργό αν καλείτε τη μέθοδο για διαφορετικό k κάθε φορά. Λάβετε υπόψη σας το υπολογιστικό κόστος του αλγορίθμου στην υλοποίηση σας. Αν θέλετε να κάνετε πολλά εξερευνητικά πειράματα χρησιμοποιήστε κάποιο sample των test δεδομένων. Τα αποτελέσματα που θα αναφέρετε θα πρέπει να είναι στο σύνολο των δεδομένων.

Βήμα 6: Υλοποιήστε τον αλγόριθμο **Item-Based Collaborative Filtering (ICF)**. Ο αλγόριθμος είναι πολύ παρόμοιος με αυτόν στο Βήμα 3, απλά δουλεύετε με στήλες αντί για γραμμές.

Θα χρησιμοποιήσετε τον πίνακα R που δημιουργήσατε στο Βήμα 3. 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:

$$p(u, j) = \overline{r}_u + \frac{\sum_{b' \in N_k(b,u)} s(b, b') R[u, b']}{\sum_{b' \in N_k(b,u)} s(b, b')}$$

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

Η βασική ιδέα της υλοποίησης είναι η ακόλουθη:

Για κάθε ζευγάρι (u, b) στα test:

1. Βρείτε τις επιχειρήσεις που έχει βαθμολογήσει ο χρήστης u.
2. Υπολογίστε την ομοιότητα αυτών των επιχειρήσεων με την επιχείρηση b και κρατήστε τις k πιο όμοιες. Αν υπάρχουν λιγότερες από k επιχειρήσεις που έχει βαθμολογήσει ο u, χρησιμοποιήστε τις όλες.
3. Χρησιμοποιείστε τα διανύσματα με τις ομοιότητες και τις βαθμολογίες για τις k πιο όμοιες επιχειρήσεις, και υπολογίστε την βαθμολογία με την παραπάνω εξίσωση.

Όπως και πριν θα υλοποιήσετε μια μέθοδο που για ένα ζευγάρι θα υπολογίζει όλες τις βαθμολογίες για όλες τις τιμές του k. Η υλοποίηση σας θα γίνει χρησιμοποιώντας μόνο μεθόδους διαχείρισης πινάκων και διανυσμάτων της numpy ή scipy. Δεν πρέπει ποτέ να διατρέξετε τις επιχειρήσεις ή τους χρήστες στα training δεδομένα, και δεν πρέπει να υλοποιήσετε ένα πυκνό πίνακα $N \times M$.

Τρέξτε τον αλγόριθμο για $k = 1,2,3,5,8,10,12,15,20,25,30,40,50$ και φτιάξετε μια γραφική παράσταση με το RMSE ως συνάρτηση του k. Σχολιάστε το αποτέλεσμα και την τιμή η οποία δίνει τα καλύτερα αποτελέσματα. Μπορείτε να δοκιμάσετε περισσότερες τιμές αν θέλετε.

Βήμα 7: Αφού ολοκληρώσετε τα Βήματα 1-6, κάνετε μια συγκριτική αξιολόγηση των αλγορίθμων. Φτιάξετε ένα πίνακα που να έχει όλους τους αλγορίθμους, και το καλύτερο error που επιτυγχάνει ο κάθε αλγόριθμος, και σχολιάστε τα αποτελέσματα. Στο σχολιασμό σας αναφέρετε και τους χρόνους εκτέλεσης για τους αλγορίθμους (για να υπολογίσετε όλες τις βαθμολογίες για όλες τις τιμές του k). Μπορείτε να επαναλλάβετε κάποιο από τον σχολιασμό σε προηγούμενα βήματα.

Παραδώσετε ένα notebook με τον κώδικα σας και την αναφορά με τις παρατηρήσεις σας για τα αποτελέσματα. Στο notebook το κάθε βήμα θα πρέπει να έχει δική του επικεφαλίδα.

Υποδείξεις

Οι παρακάτω συναρτήσεις μπορεί να σας φανούν χρήσιμες:

- 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. Οι μέθοδοι reshape και flatten μπορεί να σας βοηθήσουν.
- Η μέθοδος nonzero σας δίνει τις μη μηδενικές τιμές σε ένα numpy διάνυσμα ή πίνακα.
- Η εντολή argsort της numpy σας δίνει την σειρά των indices ενός διανύσμτος όπως προκύπτει μετά την ταξινόμηση των τιμών του.
- Για το RMSE μπορείτε να χρησιμοποιήσετε τη μέθοδο sklearn.mean_squared_error.

Ερώτηση 3

Στην ερώτηση αυτή θα εξασκηθείτε με αλγορίθμους για ομαδοποίηση (clustering). Θα χρησιμοποιήσετε και πάλι τα δεδομένα για τις επιχειρήσεις στην Philadelphia, και συγκεκριμένα επιχειρήσεις που είναι εστιατόρια. Για την ομαδοποίηση θα χρησιμοποιήσετε το κείμενο των reviews για τις επιχειρήσεις. Για τη αξιολόγηση θα ερευνήσουμε αν τα clusters που βρίσκουμε συμφωνούν με την κατηγορία της επιχείρησης.

Σας δίνεται το αρχείο philly_businesses.csv στη σελίδα ασκήσεων του μαθήματος με τις επιχειρήσεις στην Philadelphia. Από το αρχείο αυτό κρατήστε μόνο τις επιχειρήσεις που εμφανίζεται το String Restaurant μέσα στο πεδίο categories, και επίσης ένα από τα παρακάτω Strings: Japanese, Italian, Burgers. Αφαιρέστε τις επιχειρήσεις που εμφανίζονται σε παραπάνω από μία από τις τρεις παραπάνω κατηγορίες. Θα πρέπει να μείνετε με 951 επιχειρήσεις. Κάθε επιχείρηση έχει πλέον ένα μοναδικό category από τα Japanese, Italian, Burgers.

Τραβήξτε όλα τα reviews που έχουν γίνει γι αυτές τις επιχειρήσεις, και συνενώστε τα σε ένα μεγάλο String. Χρησιμοποιήστε τον tf-idf vectorizer και πάρετε ένα διάνυσμα για κάθε επιχείρηση (αφαιρέστε τα stop-words και σας προτείνεται να βάλετε κάποια κατώφλια στον αριθμό των features, στο document frequency των λέξεων, και στον αριθμό των εμφανίσεων μιας λέξης). Αυτά είναι τα δεδομένα που θέλουμε να κάνουμε cluster.

Θα κάνετε τα εξής βήματα

1. Εφαρμόστε τον k-means αλγόριθμο και τον agglomerative clustering αλγόριθμο για όλες τις διαφορετικές επιλογές της παραμέτρου linkage, για αριθμό clusters $k = 3$. Εκτυπώστε τον πίνακα σύγχυσης μεταξύ των cluster labels και των categories. Τι παρατηρείτε για την απόδοση των αλγορίθμων?

2. Για τον k-means αλγόριθμο, πάρετε τα κέντρα των clusters και για κάθε κέντρο τυπώστε τις 10 λέξεις με τις μεγαλύτερες τιμές. Τι παρατηρείτε και τι συμπέρασμα βγάζετε?
3. Για τον k-means αλγόριθμο και τον καλύτερο agglomerative αλγόριθμο από το Βήμα 1, υπολογίστε το precision, recall, F1-measure για κάθε cluster και συνολικά. Τι παρατηρείτε?
4. Για τον k-means αλγόριθμο, δοκιμάστε τιμές του k από 2 εώς 10 και δημιουργείστε το συνδυασμένο διάγραμμα με το SSE και το Silhouette Coefficient για να αποφασίσετε ποιος είναι ο "σωστός" αριθμός από clusters. Σχολιάστε το γράφημα και την απόφαση σας.
   Στη συνέχεια δημιουργείστε τον πίνακα σύγχυσης για την τιμή του k που επιλέξατε και τυπώστε πάλι τις 10 πιο σημαντικές λέξεις από τα κέντρα των clusters. Τι παρατηρείτε?

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

Υποδείξεις:

- Μπορεί να σας φανεί χρήσιμη η μέθοδος agg για pandas groupby
- Μπορείτε να χρησιμοποιήσετε την συνάρτηση που φτιάξαμε στην τάξη για να αντιστοιχίσετε τα cluster με τα category labels.