Πανεπιστήμιο Ιωαννίνων,
Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Εξόρυξη Δεδομένων – ΜΥΕ012/ΠΛΕ 059
Χειμερινό Εξάμηνο, 2023

Τρίτη Σειρά Ασκήσεων

Η προθεσμία για την δεύτερη σειρά ασκήσεων είναι την Παρασκευή 23 Φεβρουαρίου, 11:55 μ.μ. Παραδώστε Notebooks με τον κώδικα και τις αναφορές σε .ipynb **και** .html μορφή. 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

Για την άσκηση αυτή θα δείξετε την σχέση που υπάρχει μεταξύ του Pagerank διανύσματος με ομοιόμορφο jump vector, και των personalized Pagerank διανυσμάτων. Υπενθυμίζω ότι ο Pagerank αλγόριθμος έχει σαν παράμετρο ένα διάνυσμα $\boldsymbol{v}$ (το jump vector) το οποίο ορίζει μια κατανομή πιθανότητας πάνω στους κόμβους του γραφήματος και η τιμή $\boldsymbol{v}(i)$ καθορίζει την πιθανότητα να επιλέξουμε τον κόμβο i για επανεκκίνηση. Έστω $\boldsymbol{p_u}$ το Pagerank διάνυσμα με ομοιόμορφο jump vector (δηλαδή $\boldsymbol{v^T} = \left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n}\right)$), και $\boldsymbol{p_i}$ το personalized Pagerank διάνυσμα όπου το jump vector δίνει όλη την πιθανότητα στον κόμβο i (δηλαδή, $\boldsymbol{v^T} = (0,0, \dots, 0,1,0, \dots 0)$ με το 1 στην i θέση).

Αποδείξετε ότι το διάνυσμα $\boldsymbol{p_u}$ είναι ο μέσος όρος των διανυσμάτων $\boldsymbol{p_1}, \boldsymbol{p_2}, \dots, \boldsymbol{p_n}$, δηλαδή, $\boldsymbol{p_u} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{p_i}$. Για την απόδειξη θα χρησιμοποιήσετε το γεγονός ότι το Pagerank vector $\boldsymbol{p_v}$ (το Pagerank διάνυσμα με jump vector $\boldsymbol{v}$) μπορεί να γραφτεί σαν γραμμική συνάρτηση του jump vector, δηλαδή $\boldsymbol{p_v^T} = \boldsymbol{v^T Q}$, για κάποιο πίνακα $\boldsymbol{Q}$.

(Υπενθύμιση: Όταν αναφερόμαστε σε διανύσματα υποθέτουμε ότι είναι στήλες. Δηλαδή ένα n-διάστατο διάνυσμα $\boldsymbol{v}$ είναι ένας $n \times 1$ πίνακας. Αν θέλουμε να χρησιμοποιήσουμε το διάνυσμα σαν γραμμή, δηλαδή σαν ένα $1 \times n$ πίνακα θα το συμβολίζουμε ως $\boldsymbol{v^T}$)

Απαντήστε στα εξής ερωτήματα:

1. Χρησιμοποιώντας την σχέση $\boldsymbol{p_v^T} = (1-a)\boldsymbol{p_v^T P} + a\boldsymbol{v^T}$, δώστε την φόρμουλα για τον πίνακα $\boldsymbol{Q}$.
2. Δοθείσας της σχέσης $\boldsymbol{p_v^T} = \boldsymbol{v^T Q}$, τι ισχύει για τις γραμμές του πίνακα $\boldsymbol{Q}$ (ως προς τα personalized Pagerank vectors)?
3. Αποδείξτε ότι $\boldsymbol{p_u} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{p_i}$
4. Στην γενική περίπτωση ενός οποιουδήποτε jump vector $\boldsymbol{v}$ (όχι απαραίτητα το ομοιόμορφο διάνυσμα), πως μπορούμε να εκφράσουμε το $\boldsymbol{p_v}$ σαν συνάρτηση των $\boldsymbol{p_i}$?

Ερώτηση 2

Στην άσκηση αυτή θα εξασκηθείτε στην εφαρμογή αλγορίθμων κατηγοριοποίησης. Θα χρησιμοποιήσετε και πάλι δεδομένα από το amazon reviews dataset, και συγκεκριμένα όλα τα metadata για τα προϊόντα στην κατηγορία "Cell Phones & Accessories". Θα επικεντρωθούμε στις κατηγορίες "Chargers & Power Adapters" και "Car Chargers". Ο στόχος είναι να χρησιμοποιήσετε την περιγραφή του προϊόντος (το πεδίο description) για να κατηγοριοποιήσετε ένα προϊόν στην σωστή κατηγορία.

Η Ερώτηση έχει τα εξής βήματα:

1. Στο πρώτο βήμα θα κάνετε την προεπεξεργασία των δεδομένων για να δημιουργήσετε τα δεδομένα που θα χρησιμοποιήσετε για την κατηγοριοποίηση. Το αποσυμπιεσμένο αρχείο είναι πολύ μεγάλο, οπότε συνίσταται να κάνετε ένα πέρασμα στο αρχείο αντί να το φορτώσετε σε κάποιο DataFrame. Μπορείτε να χρησιμοποιήσετε την βιβλιοθήκη json για να κάνετε parse τις γραμμές του αρχείου. Θα κρατήσετε μόνο τα προϊόντα που είναι στις κατηγορίες που μας ενδιαφέρουν (η τελευταία κατηγορία στη λίστα του πεδίου "category") και έχουν μη κενή περιγραφή. Η τελική περιγραφή του προϊόντος θα είναι η συνένωση όλων των Strings στην λίστα του πεδίου "description". Αν κάποιο asin εμφανίζεται πολλές φορές θα κρατήσετε την πρώτη εμφάνιση του. Το τελικό dataset θα αποτελείται από εγγραφές που θα αποτελούνται από ένα String με την περιγραφή του προϊόντος, και ένα String με την κατηγορία του προϊόντος.
2. Στη συνέχεια, θα πειραματιστείτε με τρεις classifiers: Logistic Regression, SVM, και Multi-Layer Perceptron. Για την αξιολόγηση θα χρησιμοποιήσετε 5-fold cross validation. Θα κάνετε shuffle τα δεδομένα και θα χρησιμοποιήσετε την μέθοδο KFold για να πάρετε τα 5 train-test υποσύνολα (ή μπορείτε να κάνετε μόνοι σας το σπάσιμο). Για την εξαγωγή features θα χρησιμοποιήσετε την tf-idf αναπαράσταση των περιγραφών. Σε κάθε fold, θα δημιουργείτε ένα διαφορετικό tf-idf vectorizer για τα train δεδομένα του fold, θα κάνετε train τους classifiers, και θα τους τεστάρετε στα test δεδομένα.
   Αναφέρετε το μέσο confusion matrix από τα 5 folds, και τις μέσες τιμές για τις μετρικές accuracy, precision, recall και F1-measure (για τις τρεις τελευταίες ανά κλάση). Μπορείτε να πειραματιστείτε με διάφορες παραλλαγές του tf-idf vectorizer (π.χ., συγκεκριμένο αριθμό από features, κλπ) και διαφορετικά settings για τον MLP classifier. Σχολιάστε τα αποτελέσματα.
   Για τον Logistic Regression classifier στο τελευταίο fold, βρείτε τις 20 λέξεις που ο classifier δίνει το μεγαλύτερο θετικό βάρος και τις 20 λέξεις με το μικρότερο αρνητικό βάρος. Σχολιάστε τις λέξεις που είναι σημαντικές για την κατηγοριοποίηση.
3. Στο τρίτο βήμα θα χρησιμοποιήσετε τα ίδια δεδομένα όπως και στο Βήμα 1, αλλά θα εξάγετε τα features χρησιμοποιώντας embeddings, χρησιμοποιώντας το Doc2Vec model το οποίο θα εκπαιδεύεστε εσείς. Κάνετε την ίδια αξιολόγηση όπως στο Βήμα 1. Δηλαδή θα κάνετε k-fold cross validation με k=5, και θα χρησιμοποιήσετε ακριβώς τα ίδια folds όπως και πριν, και θα μετρήσετε τις ίδιες μετρικές. Για καθένα από τα training sets θα εκπαιδεύσετε ένα διαφορετικό embedding. Συγκρίνετε τα αποτελέσματα σε αυτή την προσέγγιση με αυτά με την tf-idf αναπαράσταση.
4. Χρησιμοποιήστε τα υπάρχοντα Glove word embeddings για να εκπαιδεύσετε ένα classifier. Η αναπαράσταση του κειμένου στην περίπτωση αυτή θα είναι η μέση τιμή των embeddings των λέξεων, όπως δείξαμε στο

φροντιστήριο. Κάνετε την ίδια αξιολόγηση όπως και στα προηγούμενα βήματα και συγκρίνετε με τις άλλες προσεγγίσεις.

Κάνετε μια σύνοψη των αποτελεσμάτων των διαφορετικών αλγορίθμων και διαφορετικών προσεγγίσεων για την αναπαράσταση των περιγραφών και εξάγετε ένα τελικό συμπέρασμα.

Bonus: Χρησιμοποιείστε επιπλέον πεδία από τα χαρακτηριστικά των προϊόντων για να βελτιώσετε την ποιότητα της κατηγοριοποίησης.

Παράδοση: Παραδώστε το notebook με τους υπολογισμούς σας, τα αποτελέσματα, και το κείμενο του σχολιασμού. Στο notebook θα πρέπει να είναι σαφή τα διαφορετικά βήματα της άσκησης.

Ερώτηση 3

Σε αυτή την ερώτηση θα χρησιμοποιήσετε τον Personalized Pagerank αλγόριθμο για να προβλέψετε αν ένας χρήστης θα αγοράσει ένα προϊόν. Θα χρησιμοποιήσετε δεδομένα παρόμοια με αυτά στην Άσκηση 2, σχετικά με χρήστες που έχουν αγοράσει κινητά τηλέφωνα.

Σας δίνονται δύο αρχεία: (1) Το αρχείο graph_train_data.txt το οποίο περιέχει ζεύγη (reviewerID, asin) χωρισμένα με κενό. Τα ζεύγη αντιστοιχούν σε χρήστες οι οποίοι έχουν αγοράσει τα προϊόντα. (2) Το αρχείο graph_test_data.txt επίσης της ίδιας μορφής, με τα ζεύγη (reviewerID, asin) που θέλουμε να προβλέψουμε. Στα test δεδομένα υπάρχει μία εγγραφή για κάθε χρήστη από τα train δεδομένα με ένα προϊόν το οποίο έχουμε «κρύψει».

Χρησιμοποιώντας τα δεδομένα στο train αρχείο δημιουργείστε ένα (διμερές) γράφημα με ακμές μεταξύ reviewerID και asin αν ο χρήστης έχει αγοράσει αυτό το αντικείμενο. Στη συνέχεια για κάθε reviewerID-asin ζευγάρι (r, a) στα test δεδομένα τρέξετε ένα Personalized Pagerank αλγόριθμο που έχει σαν σημείο επανεκκίνησης τον κόμβο r. Ταξινομήστε τα προϊόντα (**μόνο** τα asins) σε φθίνουσα σειρά βάσει την πιθανότητα που παράγει ο Personalized Pagerank, αφαιρώντας τα προϊόντα τα οποία ήδη έχει αγοράσει ο χρήστης (τους γείτονες του στο διμερές γράφημα). Καταχωρείστε την θέση στο ranking που εμφανίζεται το asin a που μας ενδιαφέρει (η πρώτη θέση είναι η 1). Θέλουμε το a να είναι όσο πιο ψηλά γινεται στο ranking.

Για να αξιολογήσετε τον αλγόριθμο θα κοιτάξετε δύο μετρικές. Η πρώτη είναι το Mean Reciprocal Rank (**MRR**), το οποίο ορίζεται ως η μέση τιμή (υπολογισμένη από όλα τα ζεύγη στα test δεδομένα), του ανάστροφου της θέσης στην οποία βρήκαμε το προϊόν που ψάχνουμε. Μαθηματικά, αν p_i είναι η θέση στην οποία βρήκαμε το προϊόν a_i, για το ζεύγος (r_i, a_i), $MRR = \frac{1}{n}\sum_i \frac{1}{p_i}$ όπου n είναι ο αριθμός των ζευγαριών στα test δεδομένα.

Επίσης θα υπολογίσετε το hit-ratio στις K πρώτες θέσεις (**HR@K**), που είναι το ποσοστό των ζευγαριών στα test δεδομένα, όπου το προϊόν βρέθηκε μέσα στις K πρώτες θέσεις του ranking. Υπολογίστε το HR@K για K = 1,…,20 και κάνετε μια γραφική παράσταση του HR@K ως προς το K.

Σχολιάστε την απόδοση του αλγορίθμου.

Bonus: Τροποποιείστε τον αλγόριθμο UCF που υλοποιήσατε στην Δεύτερη Άσκηση ώστε να δουλεύει με binary δεδομένα, και παράγετε ένα σκορ για κάθε προϊόν που δεν έχει δει ένας χρήστης. Χρησιμοποιείστε αυτό το σκορ για να κάνετε rank τα προϊόντα και συγκρίνετε με τον Personalized Pagerank χρησιμοποιώντας τις μετρικές που είδαμε παραπάνω.