Πανεπιστήμιο Ιωαννίνων,
Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Εξόρυξη Δεδομένων – ΜΥΕ012/ΠΛΕ 059
Χειμερινό Εξάμηνο, 2021

Δεύτερη Σειρά Ασκήσεων

Η προθεσμία για την δεύτερη σειρά ασκήσεων είναι την Κυριακή 20 Φεβρουαρίου 11:55 μ.μ. Παραδώστε Notebooks με τον κώδικα και τις αναφορές. Για την Ερώτηση 2 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

Στην άσκηση αυτή θα εξασκηθείτε στην εφαρμογή αλγορίθμων κατηγοριοποίησης. 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 uncompressed οπότε θα χρειαστείτε χώρο, και πρέπει να το λάβετε υπόψιν σας κατά την επεξεργασία).

Από το αρχείο με τις επιχειρήσεις κρατήστε μόνο τις επιχειρήσεις στην πόλη της Βοστόνης. Φορτώστε τα reviews για αυτές τις επιχειρήσεις, και κρατήστε μόνο τα reviews τα οποία εμφανίστηκαν μέσα στο 2020. Το κάθε review έχει ένα αριθμό από αστέρια (stars). Θα θεωρήσουμε ότι ένα review είναι θετικό αν έχει τέσσερα ή πέντε αστέρια, και αρνητικό αλλιώς. Ο στόχος μας είναι να χρησιμοποιήσουμε το κείμενο του review για να προβλέψουμε αν το review είναι θετικό ή αρνητικό.

Η Ερώτηση έχει δύο βήματα:

1. Στο πρώτο βήμα θα πάρετε την tf-idf αναπαράσταση των reviews, και θα πειραματιστείτε με τρεις classifiers: Logistic Regression, SVM, και K-NN. 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). Σε κάθε fold, θα δημιουργείτε ένα διαφορετικό tf-idf vectorizer με τα train δεδομένα, 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).
   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.
2. Στο δεύτερο βήμα θα χρησιμοποιήσετε τα ίδια δεδομένα όπως και στο Βήμα 1, αλλά θα εξάγετε τα features χρησιμοποιώντας τα word embeddings του Google. Η αναπαράσταση του κειμένου θα είναι η

μέση τιμή των embeddings των λέξεων, όπως δείξαμε στο φροντιστήριο. Κάνετε την ίδια αξιολόγηση όπως στο Βήμα 1, και εξετάστε αν βελτιώνονται ή χειροτερεύουν τα αποτελέσματα. Για το βήμα αυτό μπορείτε αν θέλετε να χρησιμοποιήσετε την μέθοδο cross_validate, εφόσον η αναπαράσταση πλέον δεν εξαρτάται από το υποσύνολο που χρησιμοποιείτε για εκπαίδευση.

Υπόδειξη: Ο στόχος της άσκησης είναι να χρησιμοποιήσετε τα word embeddings του Google. Αν όμως έχετε πρόβλημα να τα φορτώσετε (λόγω περιορισμών στη μνήμη) μπορείτε να εκπαιδεύσετε το δικό σας word embedding μοντέλο.

Παράδοση: Παραδώστε ένα 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 2

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 3

Για την άσκηση αυτή θα ασχοληθείτε με το πρόβλημα του link prediction ή link recommendation, όπου ο στόχος είναι σε ένα κοινωνικό δίκτυο να προβλέψουμε συνδέσεις που θα γίνουν στο μέλλον και να κάνουμε τις αντίστοιχες προτάσεις φιλίας στους χρήστες.

Η άσκηση αποτελείται από τα εξής βήματα:

Βήμα 1: Σας δίνεται το αρχείο edges.txt το οποίο περιέχει τις ακμές ενός κατευθυνόμενου γραφήματος. Το γράφημα αναπαριστά ένα κοινωνικό δίκτυο και οι ακμές το ποιος χρήστης ακολουθεί ποιον. Οι ακμές είναι διατεταγμένα ζευγάρια από κόμβους, και η κατεύθυνση είναι από τον πρώτο κόμβο προς τον δεύτερο. Από αυτό το γράφημα θα διαλέξετε τυχαία 100 κόμβους που έχουν πάνω από 10 εξερχόμενες ακμές (out_degree > 10). Για κάθε κόμβο θα αφαιρέσετε τυχαία μία από τις εξερχόμενες ακμές του. Μπορείτε να κάνετε τη δειγματοληψία όπως θέλετε (είτε μετατρέποντας τα δεδομένα σε γράφο, είτε με pandas, είτε μα απλή επεξεργασία). Στο τέλος αυτού του βήματος θα πρέπει να έχετε ένα σύνολο S από 100 ακμές, επιλεγμένες με την διαδικασία που περιγράψαμε παραπάνω, και το γράφημα G το οποίο προκύπτει μετά από την αφαίρεση των ακμών.

Βήμα 2: Σας δίνεται το αποτέλεσμα του πρώτου βήματος: Το αρχείο sample.txt περιέχει τις ακμές που αφαιρέσαμε από το δίκτυο και το αρχείο graph.txt τις υπόλοιπες ακμές του γραφήματος G. Θα χρησιμοποιήσετε αυτά τα δεδομένα στα πειράματα σας ώστε να έχετε όλοι τα ίδια αποτελέσματα.

Ο στόχος μας είναι για κάθε ακμή (x, y) στο δείγμα, να κάνουμε συστάσεις φιλίας στον κόμβο x με στόχο να βρούμε τον κόμβο y, της ακμής που αφαιρέσαμε. Για τον κόμβο x οι **υποψήφιοι κόμβοι** $C(x)$ για σύσταση φιλίας είναι όλοι οι κόμβοι του γραφήματος που δεν είναι ήδη γείτονες του x, ή ο ίδιος ο κόμβος x. Οι αλγόριθμοι που θα εξετάσουμε παράγουν μια ταξινόμηση (ranking) του συνόλου $C(x)$ και προτείνουν με βάση αυτή την ταξινόμηση. Μας ενδιαφέρει ο κόμβος y να είναι ψηλά στην ταξινόμηση.

Θα χρησιμοποιήσουμε τις παρακάτω μετρικές αξιολόγησης:

- **Mean Rank (MR):** Η μέση τιμή στις 100 ακμές της θέσης στην ταξινόμηση στην οποία βρίσκουμε τον κόμβο-στόχο y. Η αρίθμηση της θέσης ξεκινάει από το 1. Συγκεκριμένα:

$$MR = \frac{1}{|S|} \sum_{(x,y)\in S} pos_{C(x)}(y)$$

  Το $pos_{C(x)}(y)$ είναι η θέση στην οποία εμφανίζεται ο κόμβος y στην ταξινόμηση του συνόλου $C(x)$ από τον αλγόριθμο μας. Θέλουμε το MR να είναι μικρό.
- **Mean Reciprocal Rank (MRR):** Η μέση τιμή στις 100 ακμές του αντίστροφου της θέσης στην ταξινόμηση στην οποία βρίσκουμε τον κόμβο-στόχο y. Συγκεκριμένα:

$$MMR = \frac{1}{|S|} \sum_{(x,y)\in S} \frac{1}{pos_{C(x)}(y)}$$

  Υπενθυμίζω ότι η αρίθμηση των θέσεων ξεκινάει από το 1. Το MMR είναι μεταξύ 0 και 1, και θέλουμε να είναι όσο πιο μεγάλο γίνεται.

- **Success@k:** Το ποσοστό από τις 100 ακμές που ο κόμβος-στόχος y βρίσκεται στους top-k κόμβους στην ταξινόμηση του συνόλου $C(x)$.

Στο βήμα αυτό θα εξετάσουμε τον Personalized Pagerank (PPR) αλγόριθμο. Για κάθε ακμή $(x, y) \in S$ θα τρέξετε ένα Personalized Pagerank με restart κόμβο τον κόμβο x. Ο αλγόριθμος θα δώσει μια πιθανότητα σε κάθε κόμβο του γραφήματος. Θα ταξινομήσετε το σύνολο $C(x)$ σε φθίνουσα σειρά με βάση αυτές τις πιθανότητες.

Χρησιμοποιήστε τις μετρικές MR και MMR για να αποφασίσετε την τιμή του alpha, για τον PPR αλγόριθμο. Δημιουργήστε δύο γραφικές παραστάσεις με τις τιμές των MR και MMR για alpha = 0.1, 0.15, 0.2, 0.25, 0.3, 0.4, 0.5, 0.6, 0.7, 0.75, 0.8, 0.15, 0.9 και επίσης τυπώστε τις τιμές. Σχολιάστε:

1. Τις διαφορές μεταξύ MR και MMR. Ποια είναι ποιοτικά η διαφορά των δύο μετρικών?
2. Την επίδοση του αλγορίθμου για διαφορετικές τιμές του alpha. Εξηγείστε τι συμβαίνει όταν αυξάνουμε το alpha, λαμβάνοντας υπόψη αυτά που έχουμε συζητήσει στην διάλεξη για το πως μοιράζει την πιθανότητα ο Personalized Pagerank αλγόριθμος.

Σημείωση: Το alpha στην υλοποίηση του NetworkX δεν είναι το jump probability αλλά η πιθανότητα ο τυχαίος περίπατος να μεταβεί σε ένα τυχαία επιλεγμένο γείτονα. Άρα το jump probability είναι 1-alpha για το alpha που χρησιμοποιεί η υλοποίηση του NetworkX

Βήμα 3: Στο βήμα αυτό θα υλοποιήσετε δύο ακόμη αλγορίθμους και θα συγκρίνετε με τον Personalized Pagerank (PPR). Θα υλοποιήσετε τους εξής αλγορίθμους:

- Common Neighbors (CN): Για κάθε ακμή $(x, y) \in S$, για κάθε υποψήφιο κόμβο $z \in C(x)$, υπολογίστε τον αριθμό των κοινών γειτόνων (εξερχόμενων γειτόνων – out-neighbors) που έχουν οι κόμβοι x, z και ταξινομήστε τους κόμβους σε φθίνουσα σειρά.
- Neighbor Jaccard Similarity (NJS): Για κάθε ακμή $(x, y) \in S$, για κάθε υποψήφιο κόμβο $z \in C(x)$, υπολογίστε το Jaccard similarity των γειτόνων (εξερχόμενων γειτόνων – out-neighbors) των κόμβων x, z και ταξινομήστε τους κόμβους σε φθίνουσα σειρά.

Υπολογίστε τις μετρικές MR και MMR για τους δύο αυτούς αλγόριθμους και συγκρίνεται με τον PPR. Επίσης, δημιουργείστε μια γραφική παράσταση με τα ποσοστά επιτυχίας των αλγορίθμων PPR (για το καλύτερο alpha από το προηγούμενο βήμα), CN, NJS για $k = 1, 5, 10, 20, 50, 75, 100$ (τυπώστε επίσης τις τιμές). Συγκρίνετε τους αλγορίθμους και αναφέρετε τα συμπεράσματα σας.

Βήμα 4: Στα προηγούμενα πειράματα χρησιμοποιήσατε ένα κατευθυνόμενο γράφημα. Στο βήμα αυτό θα μετατρέψετε το γράφημα σε μη κατευθυνόμενο και θα τρέξετε ξανά τους αλγόριθμους από τα προηγούμενα βήματα. Χρησιμοποιήστε και τις τρεις μετρικές για να συγκρίνετε τις επιδόσεις των αλγορίθμων στο κατευθυνόμενο και μη κατευθυνόμενο γράφημα. Τι παρατηρείτε ως προς την επίδοση των αλγορίθμων και τις τιμές του alpha? Σχολιάστε και εξηγήστε τα αποτελέσματα σας.

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