Τρίτη Σειρά Ασκήσεων

Η προθεσμία για την τρίτη σειρά ασκήσεων είναι την Δευτέρα 13 Ιανουαρίου, 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 ομαδοποίησης. Θα χρησιμοποιήσετε τα δεδομένα που χρησιμοποιήσατε στην Δεύτερη Σειρά Ασκήσεων. Σας δίνεται το αρχείο matrix-data.csv με τριπλέτες της μορφής (user id, movie id, rating). Το αρχείο περιέχει όλα τα δεδομένα (training & test). Όλα τα ids ξεκινάνε από το μηδέν.

Ο στόχος μας είναι να κάνουμε ομαδοποίηση των ταινιών βάσει την κοινή συμπεριφορά στα ratings. Φορτώστε τα δεδομένα σε ένα αραιό πίνακα και εφαρμόστε τρεις αλγορίθμους ομαδοποίησης: (1) Τον k-means, (2) Ένα agglomerative clustering αλγόριθμο (3) Τον DBScan αλγόριθμο. Για τις επιλογές (1), (2) για να βρείτε τον σωστό αριθμό από clusters κάνετε ένα γράφημα με δύο Y άξονες, με το error και το silhouette coefficient του k-means αλγόριθμου, όπως είδατε στο μάθημα και το φροντιστήριο. Εξετάστε αριθμό clusters από 2 έως και 15, και επιλέξτε το καλύτερο K. Για τον DBScan χρησιμοποιείστε τον αριθμό των clusters που θα βγάλει και τις default παραμέτρους.

Για να αξιολογήσετε τα clusters θα χρησιμοποιήσετε τα genres των ταινιών που είναι σε κάθε cluster. Μπορείτε να βρείτε τα genres στο αρχείο u.item που έχει πληροφορίες για κάθε ταινία. Προσοχή γιατί τα ids στο αρχείο u.item ξεκινάνε από το 1, σε αντίθεση με το αρχείο matrix_data.csv που ξεκινάνε από το 0. Τα genres μπορείτε να τα βρείτε και στο αρχείο u.genre. Μια ταινία μπορεί να έχει πολλαπλά genres. Πληροφορίες για τα αρχεία μπορείτε να βρείτε στο README αρχείο.

Για κάθε cluster θα υπολογίσετε την συχνότητα των genres και θα βρείτε τα (3 ή 4) πιο συχνά. Χρησιμοποιώντας αυτά προσπαθήστε να καταλάβετε τι αντιπροσωπεύει το κάθε cluster και αξιολογήστε την ομαδοποίηση. Για το DBScan αν έχετε πολλά clusters επικεντρωθείτε στα K μεγαλύτερα, για το K που επιλέξατε.

Παραδώστε ένα Notebook με τον κώδικα σας, τα αποτελέσματα σας και τον σχολιασμό.

Ερώτηση 3

Στην Ερώτηση αυτή θα εξασκηθείτε στην εφαρμογή αλγορίθμων κατηγοριοποίησης. Θα χρησιμοποιήσετε τα δεδομένα από την πρώτη άσκηση, και συγκεκριμένα δεδομένα από το αρχείο Census_Data_cleaned.csv. Θα δουλέψετε με ένα sample με 20,000 δείγματα που σας δίνεται στη σελίδα του μαθήματος στο αρχείο Census_Data_cleaned_sample.csv.

Το class label το οποίο θα προσπαθήσετε να προβλέψετε είναι αν κάποιος έχει ετήσιο εισόδημα πάνω από $50,000. Θα χρησιμοποιήσετε για την πρόβλεψη τα υπόλοιπα δημογραφικά δεδομένα (αφαιρώντας το εισόδημα). Πολλά από τα δεδομένα είναι κατηγορικά. Θα τα μετατρέψετε σε binary features χρησιμοποιώντας τον DictVectorizer που είδατε στο φροντιστήριο.

Θα εξετάσετε τους εξής classifiers: Decision Tree, kNN, Logistic Regression, SVM, και MLP. Για την αξιολόγηση θα κάνετε 5-fold cross validation, και θα χρησιμοποιήσετε τις μετρικές accuracy, precision και recall. Κάνετε ένα πίνακα με όλα τα αποτελέσματα και συγκρίνετε τους classifiers.

Τα αποτελέσματα των classifiers δεν είναι πολύ καλά. Δοκιμάστε μια μέθοδο για να βελτιώσετε την απόδοση τους.

Παραδώστε ένα Notebook με τον κώδικά σας, τα αποτελέσματα και τον σχολιασμό.

Υπόδειξη: Θα σας βοηθήσει η εντολή to_dict('records') για να πάρετε τα features.

Ερώτηση 4

Στην Ερώτηση 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/follow στους χρήστες. Για να αξιολογήσουμε τους αλγορίθμους αφαιρούμε κάποιες υπάρχουσες συνδέσεις και στη συνέχεια ελέγχουμε αν ο αλγόριθμος μπορεί να τις προβλέψει.

Σας δίνεται το αρχείο graph.txt τις ακμές ενός **κατευθυνόμενου** γραφήματος G και το αρχείο test.txt που περιέχει τις ακμές που θέλουμε να προβλέψουμε. Το αρχείο test.txt περιέχει 100 ακμές που αφαιρέθηκαν από 100 διαφορετικούς κόμβους στο αρχικό γράφημα. Συμβολίζουμε ως $S = \{(x_i, y_i)\}$ το σύνολο των ακμών που αφαιρέθηκε. Για την ακμή (x_i, y_i), ο κομβος x_i είναι η αφετηρία της ακμής και ο κόμβος y_i ο προορισμός. Οι κόμβοι αφετηρίας είναι όλοι διακριτοί μεταξύ τους.

Θα χρησιμοποιήσετε τον Personalized Pagerank για την πρόβλεψη των ακμών. Για κάθε ακμή (x_i, y_i) στο συνολο S, τρέξτε ένα Personalized Pagerank αλγόριθμο, με jump probability $\gamma = 0.25$ που κάνει πάντα restart στον κομο x_i. (Προσοχή: το alpha στην συνάρτηση του NetworkX είναι $1 - \gamma$) . Το αποτέλεσμα του αλγορίθμου θα δώσει μια πιθανότητα σε κάθε κόμβο του γραφήματος. Από τους υποψήφιους κόμβους (δηλαδή, κόμβους που δεν είναι ήδη γείτονες του x_i, ή ο ίδιος ο κόμβος x_i), κρατήστε τους k κόμβους με την μεγαλύτερη πιθανότητα, για διαφορετικές τιμές του k, $k = 1, 5, 10, 20, 50, 75, 100$. Έχουμε επιτυχημένη πρόβλεψη αν ο κόμβος y_i είναι μέσα στους top-k κόμβους. Υπολογίστε το ποσοστό επιτυχίας του αλγορίθμου, ως το ποσοστό από τις 100 ακμές που είχαμε επιτυχημένη πρόβλεψη. Δημιουργήστε μια γραφική παράσταση με το ποσοστό επιτυχίας του αλγορίθμου για τα διαφορετικά k.

Επαναλάβετε το πείραμα, για διαφορετικές τιμές του γ, $\gamma = 0.15, 0.25, 0.35, 0.5$. Φτιάξετε μια γραφική παράσταση με καμπύλες για όλα τα γ που να δείχνει το ποσοστό επιτυχίας ως συνάρτηση του k. Ποια τιμή του γ δίνει καλύτερα αποτελέσματα?

Προσπαθήστε να εξηγήσετε τα αποτελέσματα. Παραδώστε ένα notebook με τους υπολογισμούς σας, τα αποτελέσματα, και το κείμενο του σχολιασμού.