Τρίτη Σειρά Ασκήσεων

Η προθεσμία για την δεύτερη σειρά ασκήσεων είναι Πέμπτη 13 Φεβρουαρίου μέχρι το τέλος της ημέρας. Κάνετε turn-in τον κώδικα σας, με οδηγίες για το πώς τρέχει (αν χρειάζεται), και την αναφορά σας. Η αναφορά θα πρέπει να έχει λεπτομερείς παρατηρήσεις για τα αποτελέσματα σας. Για καθυστερημένες υποβολές ισχύει η πολιτική στην σελίδα του μαθήματος. Λεπτομέρειες για το turn-in στη σελίδα Ασκήσεις του μαθήματος. Αν δεν μπορείτε να κάνετε turnin στείλετε την εργασία σας μέσω email. Θα υπάρξει προφορική εξέταση της Άσκησης. Οι προγραμματιστικές ασκήσεις μπορούν να γίνουν σε ομάδες των δύο ατόμων.

Ερώτηση 1

Στην άσκηση αυτή θα εξασκηθείτε στην εφαρμογή αλγορίθμων κατηγοριοποίησης. Θα χρησιμοποιήσετε τα δεδομένα για τις συνταγές που χρησιμοποιήσατε στην Δεύτερη Άσκηση. Ο στόχος είναι να φτιάξετε ένα classifier που ξεχωρίζει τις διαφορετικές κουζίνες. Η Ερώτηση έχει τα εξής βήματα:

1. Στο πρώτο βήμα θα χρησιμοποιήσετε δεδομένα από τις κουζίνες Italian και Mexican. Θα εξάγετε τα χαρακτηριστικά ακριβώς όπως και στην Άσκηση 2. Θα πειραματιστείτε με πέντε classifiers: Logistic Regression, SVM, Decision Trees, K-NN, και Naïve Bayes. Για την αξιολόγηση θα χρησιμοποιήσετε 5-fold cross validation. Αναφέρετε το μέσο confusion matrix και τις μέσες τιμές για τις μετρικές accuracy, precision, recall και F1-measure.
2. Στο δεύτερο βήμα θα χρησιμοποιήσετε τα ίδια δεδομένα όπως και στο Βήμα 1, αλλά θα εξάγετε τα features διαφορετικά. Αντί να φτιάξετε ένα feature για κάθε υλικό, θα έχετε ένα feature για κάθε λέξη που εμφανίζεται στα υλικά (μια συνταγή θα είναι ένα bag of words). Μπορείτε αν θέλετε να αφαιρέσετε τα stop-words. Κάνετε την ίδια αξιολόγηση όπως στο Βήμα 1, και εξετάστε αν βελτιώνονται ή χειροτερεύουν τα αποτελέσματα.
3. Στο τρίτο βήμα θα εκπαιδεύσετε και θα εφαρμόσετε τους classifiers στα δεδομένα που χρησιμοποιήσατε για clustering στην προηγούμενη σειρά ασκήσεων. Χρησιμοποιήστε τις ίδιες μετρικές αξιολόγησης όπως και στα άλλα βήματα. Πόσο επιτυχής είναι η κατηγοριοποίηση σε σχέση με το clustering στα ίδια δεδομένα?

Παραδώστε την αναφορά σας με τα αποτελέσματα και τις παρατηρήσεις σας.

Ερώτηση 2

Και σε αυτή στην άσκηση θα πειραματιστείτε με αλγορίθμους κατηγοριοποίησης. Ο στόχος είναι να φτιάξετε ένα classifier ο οποίος θα προβλέπει αν ένα sms μήνυμα είναι spam ή όχι, χρησιμοποιώντας το κείμενο του sms μηνύματος.

Για την άσκηση αυτή δημιουργήθηκε ένας διαγωνισμός στο Kaggle για το μάθημα (εδώ είναι ο σύνδεσμος για τον διαγωνισμό). Δημιουργήστε ένα account με το email του πανεπιστημίου. Θα σας δοθεί πρόσβαση στον

διαγωνισμό μέσω του link και θα μπορέστε να καταθέσετε μια λύση για τον διαγωνισμό. Εκεί σας δίνονται τα training data και τα test data.

Υπάρχει μία κατάταξη στην οποία μπορείτε να δείτε την θέση σας σε σχέση με άλλες λύσεις. Μια καλή θέση θα ενισχύσει τον βαθμό σας. Το σημαντικό είναι να πετύχετε ένα καλό σκορ στο F1-measure για την κλάση spam, σε σχέση με τις υπόλοιπες λύσεις. Μπορείτε να χρησιμοποιήσετε όποιο αλγόριθμο θέλετε.

Στην αναφορά περιγράψετε τα χαρακτηριστικά που δημιουργήσατε και το μοντέλο που χρησιμοποιήσατε, και αναφέρετε το όνομα σας στο Kaggle, και το σκορ σας στον διαγωνισμό. Επίσης, αναφέρετε και πειράματα που κάνατε και δεν δούλεψαν, ή πως βελτιώσατε μια λύση που δεν απέδιδε καλά (περιλάβετε και αποτελέσματα από τα πειράματα).

Ερώτηση 3

Για την άσκηση αυτή θα δείξετε την σχέση που υπάρχει μεταξύ του pagerank διανύσματος με ομοιόμορφο jump vector, και των personalized pagerank διανυσμάτων. Έστω $\boldsymbol{p_u}$ το pagerank διάνυσμα με ομοιόμορφο jump vector (δηλαδή $\boldsymbol{v} = \left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n}\right)$), και $\boldsymbol{p_i}$ το personalized pagerank διάνυσμα όπου το jump vector δίνει όλη την πιθανότητα στον κόμβο i (δηλαδή, $\boldsymbol{v} = (0,0, \dots, 0,1,0, \dots 0)$ με το 1 στην i θέση).

Πρέπει να 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 μπορεί να γραφτεί σαν γραμμική συνάρτηση του jump vector, δηλαδή $\boldsymbol{p_v} = \boldsymbol{Q}\boldsymbol{v}$, για κάποιο πίνακα $\boldsymbol{Q}$.

Απαντήστε στα εξής ερωτήματα:

1. Χρησιμοποιώντας την σχέση $\boldsymbol{p_v} = (1-a)\boldsymbol{p_v}\boldsymbol{P} + a\boldsymbol{v}$, δώστε την φόρμουλα για τον πίνακα $\boldsymbol{Q}$.
2. Δοθείσας της σχέσης $\boldsymbol{p_v} = \boldsymbol{Q}\boldsymbol{v}$, τι μπορείτε να δείξετε για τις γραμμές του πίνακα $\boldsymbol{Q}$?
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 4

Σε αυτή την άσκηση θα υλοποιήσετε τον αλγόριθμο για value propagation σε γράφο τον οποίο περιγράψαμε στην τάξη.

Θα χρησιμοποιήσετε ένα γράφο ο οποίος αναπαριστά τα links μεταξύ διαφορετικών political blogs, από την εκλογική καμπάνια του 2004 στις ΗΠΑ. Το gml αρχείο του γράφου θα το βρείτε εδώ. Ο γράφος είναι κατευθυνόμενος. Για κάθε κόμβο έχετε το id του (ένα νούμερο), ένα label (το URL του blog), την πηγή, και το value που παίρνει τιμές 0 και 1, αν το blog είναι αριστερό/προοδευτικό, ή δεξιό/συντηρητικό. Για να φορτώσετε τα δεδομένα προσδιορίστε label=id στην εντολή read_gml, ώστε να πάρετε αριθμητικά ids τα οποία είναι πιο εύκολα να χειριστείτε. Ο γράφος που θα δημιουργηθεί είναι ένας directed multi-graph (κάποιες ακμές επαναλαμβάνονται). Μετατρέψτε τον σε απλό γράφο καλώντας την DiGraph εντολή πάνω του.

Αφού φορτώσετε τα δεδομένα θα κρατήσετε την μεγαλύτερη ισχυρά συνεκτική συνιστώσα (strongly connected component) του γράφου. Για κάθε μία από τις τιμές 0/1 του value, θα κρατήσετε τυχαία ένα b% των τιμών, όπου το b θα πάρει τιμές από 10% έως 90%, ανά 10%. Χρησιμοποιώντας αυτές τις τιμές κάνετε value propagation στον υπόλοιπο γράφο, και υπολογίσετε μια τιμή για κάθε άλλο κόμβο. Ο στόχος είναι να χρησιμοποιήσετε τις τιμές που υπολογίσατε για να προβλέψετε τις τιμές που κρύψατε. Προβλέψετε 1 αν η τιμή είναι μεγαλύτερη από 0.5, και 0 διαφορετικά. Υπολογίσετε την ακρίβεια των προβλέψεων (το ποσοστό των σωστών προβλέψεων).

Επαναλάβετε το πείραμα 10 φορές και υπολογίσετε την μέση ακρίβεια για κάθε πείραμα. Φτιάξετε ένα γράφημα το οποίο να δείχνει την μέση ακρίβεια, με confidence intervals, ως συνάρτηση της τιμής του b. Σχολιάστε αν το value propagation μπορεί επιτυχώς να προβλέψει τις τιμές που λείπουν.

Στο δεύτερο κομμάτι της άσκησης, αντί να διαλέγετε τους κόμβους που θα κρατήσουν τις τιμές τους τυχαία, τρέξετε τον pagerank αλγόριθμο και κρατήστε τους b% πρώτους κόμβους. Υπολογίσετε και πάλι την ακρίβεια και φτιάξετε μια γραφική παράσταση.

Συγκρίνετε τις δύο μεθόδους. Ποια από τις δύο φαίνεται να δουλεύει καλύτερα? Για να γίνει σαφής η σύγκριση βάλετε τις δύο γραφικές παραστάσεις σε ένα γράφημα. Σχολιάστε τα αποτελέσματα σας.

Παραδώστε τον κώδικα και την αναφορά με τις παρατηρήσεις σας.

Ερώτηση 5 (bonus)

Στο Kaggle υπάρχει ένας ενεργός διαγωνισμός για την πρόβλεψη αν ένα tweet αφορά μια φυσική καταστροφή ή όχι (εδώ το link του διαγωνισμού). Χρησιμοποιώντας τον λογαριασμό που δημιουργήσατε στην Ερώτηση 2, υποβάλετε μια λύση στον διαγωνισμό. Ο στόχος δεν είναι να κερδίσετε τον διαγωνισμό (αν και η θέση σας στον διαγωνισμό θα ενισχύσει το βαθμό σας), αλλά να δουλέψετε σε ένα πραγματικό πρόβλημα που δεν έχει γνωστή λύση.

Δημιουργήστε μια αναφορά που θα περιέχει τα παρακάτω:

a. Μια περιγραφή της λύσης που υλοποιήσατε. Τι χαρακτηριστικά χρησιμοποιήσατε? Ποιες τεχνικές? Μια σύντομη περιγραφή της λογικής πίσω από τις επιλογές σας.
b. Τα αποτελέσματα σας στο Kaggle test dataset.
c. Ένα σχολιασμό στα παραπάνω: Τι δούλεψε και τι δεν δούλεψε τόσο καλά? Τι καταλάβατε για τα δεδομένα και το πρόβλημα?

Παραδώσετε το κώδικα σας και την αναφορά. Αναφέρετε το όνομα σας στο Kaggle, και την θέση σας στην κατάταξη όταν κάνατε την υποβολή.