Ανάπτυξη Λογισμικού για Πληροφοριακά Συστήματα
Εργασία:2η

Μέλη της ομάδας:
Βουρτζούμη Ουρανία Α.Μ:1152016000024
Κοσμάς Αλέξανδρος Α.Μ:1115201700299
Βαρώνος Διονύσης Α.Μ:1115201600017

UniTesting:

Για το uni testing χρησιμοποιούμε την βιβλιοθήκη acutest.h
Με την εντολή make test δημιουργείται το εκτελέσιμο test το οποίο εκτελείται με την εντολή make test-run. Με την εντολή make test-clean διαγράφονται διαγράφεται το εκτελέσιμο test και όσα .ο αρχεία δημιουργήθηκαν. Το εκτελέσιμο δημιουργείται μέσα στον φάκελο test αλλά όλες οι παραπάνω εντολές δίνονται στον αρχικό φάκελο.

Μεταγλώττιση κώδικα:

Η υλοποίηση μας για το δημιουργεί ένα εκτελέσιμο sigmod το οποίο δημιουργείται με την εντολή make. Με την εντολή make clean διαγράφεται το εκτελέσιμο ολα τα .ο αρχεία

και ό λα τα .csv αρχεί α τα οποί α έ χουν δημιουργηθεί κατά την εκτέ λεση του προγρά μματος.

Εντολή εκτέ λεσης προγρά μματος :

Το εκτελέ σιμο εκτελεί ται με την εντολή :

make run ARGS="- d Directory -w datacsv"

ή απλώ ς

./sigmod - d Directory -w datacsv

Ό που:

Directory:

ο φά κελος που έ χει τους φακέ λους με τα .json αρχεί α δηλαδή ο φά κελος 2013_camera_specs στην συγκεκριμέ νη ά σκηση (ο φά κελος δεν υπά ρχει στον φά κελο μας στο github αλλά θα πρέ πει να υπά ρχει στο directory που τρέ χει το εκτελέ σιμο sigmod)

ΣΗΜΑΝΤΙΚΌ : ως directory θα πρέ πει να δωθει το ό νομα του φακέ λου και ό χι ο φά κελος σαν path . Δηλαδή το σωστό εί ναι 2013_camera_specs και ό χι 2013_camera_specs/

datacsv:

το ό νομα του csv αρχεί ου που περιέ χει τα ταιριασματα των καμερων(πχ sigmod_medium_labelled_dataset.csv ή sigmod_large_labelled_dataset.csv)

Επί σης στην εντολή εκτέ λεσης μπορεί να προστεθεί η προαιρετική σημαί α - ο.
Εαν υπά ρχει η συγκεκριμέ νη σημαί α το πρό γραμμα δοκιμά ζει το μοντέ λο για ό λα τα στοιχεί α του datasetX που δεν έ χουν δοκιμαστεί μεταξύ τους.
Εά ν δεν υπά ρχει η σημαί α το πρό γραμμα τερματί ζει αφού εκτυπώ σει τα αποτελέ σματα του testing.
Ο λό γος που επιλέ ξαμε αυτή την υλοποί ηση εί ναι επειδή ο έ λεγχος ό λων τον στοιχεί ων απαιτεί πολύ χρό νο (τουλά χιστον 30 λεπτά) και θεωρησαμε σημαντικό να μπορεί ο χρή στης να τρέ ξει το πρό γραμμα χωρί ς αυτό ν τον έ λεγχο ώ στε να διαπιστώ σει ό τι ό λες οι υπό λοιπες λειτουργί ες του προγρά μματος εκτελού ντα ομαλά .

Output προγρά μματος:

Το πρό γραμμα κατά την εκτέ λεση του δημιουργεί έ να νέ ο αρχεί ο με ό νομα Same.csv που περιέ χει ό λες τις θετικέ ς συσχετισεις που δημιουργή θηκαν απο τις κλικες μέ σω του 60% των δεδομέ νων του dataset W και έ να αρχεί ο με ό νομα Different.csv με τις αντί στοιχες αρνητικέ ς συσχετισεις.
Στην συνέ χεια εκτυπώ νει το Success rate του μοντέ λου κατά την διαδικασί α του testing το οποί ο γί νεται με το επό μενο 20% του datasetw.
Τέ λος εκτυπώ νει τις συσχετισεις που έ χουν ό ριο πιθανό τας 0.001(δηλαδή τις αρνητικέ ς με πιθανό τητα τεριασματος μικρό τερο του 0.001 και τις θετικέ ς με πιθανό τητα τεριασματος μεγαλύ τερη του 0.999)οι προέ ρχονται απο τα

αποτελέσματα που μας δίνει το μοντέλο όταν το εκτελούμε για κάθε στοιχείο (δηλαδή κάθε μέρα) του datasetX με όλα τα υπόλοιπα. (εαν ο χρηστης δωσει την σημαια -ο στην εντολή εκτελεσης)
Κατά την διάρκεια του προγράμματος εκτυπώνωνται κάποιες προτάσεις που μας ενημερώνουν πως το πρόγραμμα μόλις τελείωσε κάποια συγκεκριμένη διαδικασία (πχ όταν διαβάσει όλα τα δεδομένα του datasetX)

Ροή προγράμματος :

Το πρόγραμμα δημιουργεί δημιουργεί μια δομή Hash στην οποία αποθηκευονται τα στοιχεία του κάθε json αρχείου απο το datasetX και μια δομή LHash η οποία αντιπροσωπευει το vocabulary όλων των json.

Για κά θε αρχεί ο json αποθηκεύ εται το id της κά μερα μέ σα στην δομή Hash και στην συνέ χεια διαβά ζουμε το json λέ ξη - λέ ξη ό που κά θε λέ ξη αποθηκεύ εται στο λεξιλό γιο και σε μια δομή WHash που αποθηκεύ εται στην αντί στοιχη θέ ση που αποθηκευτικε το id του json μέ σα στην δομή Hash.

Για κά θε νέ α λέ ξη που βρί σκει το λεξιλό γιο την αποθηκεύ ει και για κά θε λέ ξη που έ χει ή δη αυξά νει κατά έ να την μεταβλητή wordperj που αντιστοιχεί στην συγκεκριμέ να λέ ξη και αντιπροσοπευει τον αριθμό των json στα οποί α βρέ θηκε αυτή η λέ ξη.

Αντί στοιχα η δομή WHash αποθηκεύ ει κά θε νέ α λέ ξη του json και για κά θε ή δη υπά ρχουσα αυξά νει κατά έ να την μεταβλητή τά δε που αντιστοιχεί στο πό σες φορέ ς βρέ θηκε η λέ ξη αυτή σε αυτό το json.

Μό λις διαβαστουν ό λες οι λέ ξεις του json το WHash υπολογί ζει το tf της κά θε λέ ξης του και στην συνέ χεια για κά θε μί α απ τις λέ ξεις του δί νει το tf στο λεξιλό γιο το οποί ο το προσθέ τει στην μεταβλητή tfcount της αντί στοιχης λέ ξης και με αυτό ν τον τρό πο κρατά ει το ά θροισμα των tf της κά θε λέ ξης για ό λα τα json.

Αφού τελειώ σει η παραπά νω διαδικασί α για ό λα τα αρχεί α του datasetX η δομή LHash υπολογί ζει το μέ σο tf-idf της κά θε λέ ξης και το αποθηκεύ ει στην μεταβλητή isf και αποθηκεύ ει το idf της λέ ξης στην μεταβλητή tfcount.

Στην συνέ χεια το λεξιλό γιο ταξινομει τις λέ ξεις με βά ση το μεγαλύ τερο μέ σο tf-idf και κρατά ει τις 1000 πρώ τες.

Στην συνέχεια για κάθε στοιχείο της δομής Hash μια δομη Hvector η οποία είναι ένας πίνακας κατακερματισμου που αντιπροσοπευει ενα spar array.
Δηλαδή για κάθε μια λέξη απο τις 1000 σημαντικές του λεξιλογίου εαν υπάρχει στην κάμερα αποθηκεύεται η θέση της και η τιμή tf*idf της αντίστοιχης λέξης για την συγκεκριμένη κάμερα.
Με αυτόν τον τρόπο δεν αποθηκεύουμε στο vector μας τις τιμές που ισουνται με 0 και είναι περιττή πληροφορία.

Μολις δημιουργείται το κάθε vector διαγράφεται η δομή WHash αυτής της θέσης.

Στην συνέχεια δημιουργούνται οι κλικες βάση το 60% των θετικών και το 60% των αρνητικών συσχετήσεων του dataW και οι οι αρνητικές συσχετισεις μεταξύ των κλικων που δεν τεριαζουν.

Επίσης δημιουργούνται τα αρχεία Testing.csv και Validation.csv που το κάθε ένα περιέχει 20% αρνητικων και θετικών συσχετησεων και θα χριεαστουν στην συνέχεια για να πάρουμε τα δεδομένα στις αντίστοιχες διαδικασίες.

Αφού δημιουργηθουν οι κλικες και οι αρνητικές συσχετισεις το πρόγραμμα δημιουργεί τα αρχεία Same.csv και Different.csv που σε αυτά αποθηκευονται όλες οι θετικές

συσχετισεις βά ση των κλικων της δομή ς και ό λες οι αρνητικέ ς συσχετισεις αντί στοιχα.

Στην συνέ χεια μεσω της συνά ρτησης Training δημιουργού με μια δομή model η οποί α αντιπροσοπευει το μοντέ λο μας. Το μοντέ λο μας εκπαιδευεται με τα δεδομέ να των αρχεί ων Same.csv και Different.csv και πιο συγκεκριμέ να για τις διπλά σιες αρνητικέ ς συσχετησεις απο τις θετικέ ς.

Η διαδικασία της εκπαίδευσης είναι ως εξής :

Για κά θε ζευγά ρι των αρχεί ων Same.csv και Different.csv υπολογί ζουμε το concatenation των vectors των αντί στοιχων καμερων και την τιμή της πρό βλεψης του μοντέ λου μεί ον την σωστή τιμή της συσχέ τισης.

Στην συνέ χεια για κά θε θέ ση του concatenation υπολογί ζουμε το γινό μενο της τιμή ς της συγκεκριμέ νης θέ σηςεπι της παραπά νω τιμή ς.

Αυτό το γινό μενο το αφαιρού με κά θε φορά απο το βά ρος της αντί στοιχης θέ σης

Αυτή η διαδικασί α επαναλαμβά νεται 3 φορέ ς και μό λις τελειώ σει η συνά ρτηση επιστρέ φει το μοντέ λο

Στην συνεχεια μεσω της συνά ρτησης Testing για κά θε ζευγά ρι του αρχεί ου Testing.csv υπολογί ζουμε την πρό βλεψη

του εκπαιδευμενου μοντέ λου και αφού αυτό γί νει για ό λα τα ζευγά ρια το πρό γραμμα εκτυπώ νει το success rate του μοντέ λου δηλαδή το ποσοστό των σωστων προβλέ ψεων

Στην συνέ χεια εαν ο χρηστης δωσει την σημαια -ο στην εντολή εκτελεσης το πρό γραμα εκτυπώ νει τις συσχετισεις που έ χουν ό ριο πιθανό τας 0.001(δηλαδή τις αρνητικέ ς με πιθανό τητα τεριασματος μικρό τερο του 0.001 και τις θετικέ ς με πιθανό τητα τεριασματος μεγαλύ τερη του 0.999)οι οποί ες προέ ρχονται απο τα αποτελέ σματα που μας δί νει το μοντέ λο ό ταν το εκτελού με για κά θε στοιχεί ο (δηλαδή κά μερα) του datasetX με ό λα τα υπό λοιπα.

Τέ λος αποδεσμευονται ό λες οι δομέ ς και το πρό γραμμα τερματί ζει.

Δομέ ς :

Η δομή Hash εί ναι η δομή που αποθηκευονται τα δεδομέ να του κά θε json αρχεί ου.

Εί ναι έ νας δυναμικό ς πί νακας κατακερματισμου με bucket-list ό που ως κλειδί χρησιμοποιεί το id κά θε json (πχ www.ebay.com//567)και κά νει rehash κά θε φορά που φτά νει 80% πληρό τητα.

Για την υλοποί ηση του bucket-list χρησιμοποιεί ται η δομή NList που σε κά θε κό μβο της αποθηκεύ εται:

το id του json στην μεταβλητή camera τύ που char*

οι λέ ξεις του json αρχεί ου στην μεταβλητή spear που εί ναι τύ που Whash*

το αντί στοιχο vector του json που εί ναι τύ που HVector*

έ νας δεί κτης σε δομη CList που αντιστοιχεί στην κλικα την οποί α ανή κει η camera.

Η δομή CList εί ναι μια συνδεδεμέ νη λί στα που για την υλοποί ηση των κλικων.

Σε κά θε θέ ση της αποθηκεύ ει:

το ό νομα της κά μερας

έ ναν δεί κτη σε NList που αντιστοιχεί με τον κό μβο NList που εί ναι αποθηκευμέ να τα στοιχεί α της κά μερας.

Ο πρώ τος κό μβος της κά θε CList δεν αποθηκεύ ει δεδομέ να μιας κά μερας αλλά μια λί στα TList(συνδεδεμέ νη λί στα που αποθηκεύ ει δεί κτες σε CList) που αποθηκεύ ει τις κλικες με τις οποί ες δεν τεροιαζει η κλικα.

Η δομή WHash εί ναι μια δομή Πί νακα κατακερματισμου χωρί ς bucket-list που αποφεύ γει τα collision πηγαί νοντας στην επό μενη διαθέ σιμη κενή θέ ση και κά νει rehash ό ταν έ χει 80% πληρό τητα.

Η δομή αυτή χρησιμοποιεί ται για να αποθηκεύ ει τις λέ ξης του κά θε json αρχεί ου με κλειδί την κά θε λέ ξη.
Σε κά θε bucket αποθηκεύ ει μια λέ ξη και το tf αυτή ς της λέ ξης για το συγκεκριμέ νο json.

Η δομή Hvector εί ναι μια δομή πί νακα κατακερματισμου χωρί ς bucket-list που αποφεύ γει τα collision πηγαί νοντας στην επό μενη διαθέ σιμη κενή θέ ση και κά νει rehash ό ταν έ χει 80% πληρό τητα.
Η δομή αυτή αναπαριστά το vector της κά θε κά μερας
Σε κά θε bucket αποθηκεύ ει την θέ ση και την tf*idf της αντί στοιχης θέ σεις για κά θε λέ ξη απο τις 1000 πιο σημαντικέ ς του λεφιλογί ου ενα υπαρχει στην κά μερα.

Η δομή LHash εί ναι μια δομή Πί νακα κατακερματισμου χωρί ς bucket-list που αποφεύ γει τα collision πηγαί νοντας στην επό μενη διαθέ σιμη κενή θέ ση και κά νει rehash ό ταν έ χει 80% πληρό τητα.
Η δομή αυτή χρησιμοποιεί ται για την υλοποί ηση του λεξιλογιου ό λων των json με κλειδί την κά θε λέ ξη.
Σε κά θε bucket αποθηκεύ εται η κά θε λέ ξη το idf και το μέ σω tf-idf της κά θε λέ ξης

Η δή λωση ό λων των δομων λιστων της δομή ς Hvector και της δομή ς WHash και τα πρό τυπα των συναρτή σεων για την διαχεί ριση τους βρί σκονται στο αρχεί ο list.h και οι υλοποιή σεις των συναρτή σεων στο αρχεί ο list.c

Η δή λωση ό λων των δομων Hash και WHash και τα πρό τυπα των συναρτή σεων για την διαχεί ριση τους βρί σκονται στο

αρχεί ο hash.h και οι υλοποιή σεις των συναρτή σεων στο αρχεί ο hash.c

Η δή λωση της δομή ς Model και τα πρό τυπα των συναρτή σεων για την διαχεί ριση της βρί σκονται στο αρχεί ο logistic.h και οι υλοποιή σεις των συναρτή σεων στο αρχεί ο logistic.c