Ανάπτυξη Λογισμικού για Πληροφοριακά Συστήματα

Project3

Περιγραφή προγράμματος

Το project αυτό αποτελεί επέκταση του δεύτερου μέρους (project2). Προσθέτει λοιπόν στο πρόγραμμα τις παρακάτω λειτουργίες:

Η διαδικασία training του μοντέλου μηχανικής μάθησης έχει τροποποιηθεί με σκοπό την καλύτερη εκμάθηση, και ως αποτέλεσμα την παροχή καλύτερων προβλέψεων. Η νέα διαδικασία που ακολουθείται είναι η εξής:

1. Το μοντέλο εκπαιδεύεται με το 60% του Dataset W. Η διαδικασία αυτή εκτελείται σε batches, το μέγεθος τον οποίων ορίζεται από τον χρήστη.
2. Δημιουργείται ένα νέο hashTable αποτελούμενο από το παραπάνω τμήμα του Dataset, το οποίο θα χρησιμοποιηθεί για την επίλυση τυχών transitivity issues.
3. Διατρέχονται όλα τα ζευγάρια του Dataset, και για κάθε ένα από αυτά υπολογίζεται η πρόβλεψη του μοντέλου.
4. Αν η πρόβλεψη (p) είναι: $p < threshold$ ή $p > (1 – threshold)$ το ζευγάρι τοποθετείται σε ένα δέντρο, χρησιμοποιώντας ως κλειδί την βεβαιότητα της πρόβλεψης του μοντέλου γι' αυτό.
5. Όταν ολοκληρωθεί η διαδικασία του βήματος 4, τα ζευγάρια εξάγονται από το δέντρο με inorder traversal, αν δεν υπάρχει transitivity issue, προσθέτονται στο νέο hashTable και έπειτα σε ένα νέο επαυξημένο training set.
6. Η παραπάνω διαδικασία επαναλαμβάνεται 3 φορές.

Μόλις ολοκληρωθεί η εκπαίδευση του μοντέλου, γίνεται δοκιμή του χρησιμοποιώντας ένα 20% του Dataset W ως test-set. Κατά την διαδικασία αυτή αποθηκεύονται διάφορες μετρικές (recall, f1, precision), οι οποίες στην συνέχεια εκτυπώνονται.

Σημειώσεις

- Το threshold είναι μια μεταβλητή η οποία αρχικοποιείται σε 0.10 και σε κάθε επανάληψη της διαδικασίας αυξάνεται κατά 0.10.
- Το μοντέλο δηλαδή εκτελεί 3 φορές training πάνω στα δεδομένα, έχοντας κάθε φορά μεγαλύτερο training set.
- Έχουμε τροποποιήσει με τέτοιο τρόπο την παραπάνω διαδικασία ώστε κάθε ζευγάρι με αποτέλεσμα 1, να "περνάει" από τον classifier 12 φορές καθώς τα δεδομένα είναι unbalanced ως προς το 1.
- Τελικά γίνονται 3 διαφορετικά train του μοντέλου:
  Το πρώτο είναι με το αρχικό Dataset.
  Το 2ο με το επαυξημένο που δημιουργήθηκε για threshold = 0.10.
  Το 3ο με το επαυξημένο που δημιουργήθηκε για threshold = 0.20.
- Η εκπαίδευση του μοντέλου σε αντίθεση με το 2ο παραδοτέο, γίνεται πλέον με την τεχνική του Batch-Training. Για την ενημέρωση δηλαδή κάθε βάρους απαιτείται ένα batch ζευγαριών, το μέγεθος του οποίου ορίζεται από τον χρήστη.

Ενσωμάτωση Threads

Η διαδικασία εκπαίδευσης του μοντέλου πραγματοποιείται με την χρήση Threads. Συγκεκριμένα, έχει υλοποιηθεί ένα thread-pool το οποίο μεταφέρει στα threads συναρτήσεις με σκοπό να τις εκτελέσουν. Για την ομαλή λειτουργία του pool έχουν υλοποιηθεί οι ακόλουθες δομές:

I. Job: Μια δομή η οποία περιέχει την συνάρτηση που θα εκτελεστεί, καθώς και τα ορίσματα της
II. Queue: Μια ουρά (LIFO) , στις οποία τοποθετούνται Jobs
III. Job Scheduler μια δομή η οποία περιέχει:
    1. Μια ουρά διεργασιών
    2. Όλα τα απαραίτητα mutex, cond_variables για την λειτουργία των Threads
    3. Ορισμένους μετρητές.

    Κατά την εκτέλεση του, δημιουργεί τα Threads που έχει ζητήσει ο χρήστης, αναθέτοντας τους μια συνάρτηση διαχείρησης εργασιών (thread-function)

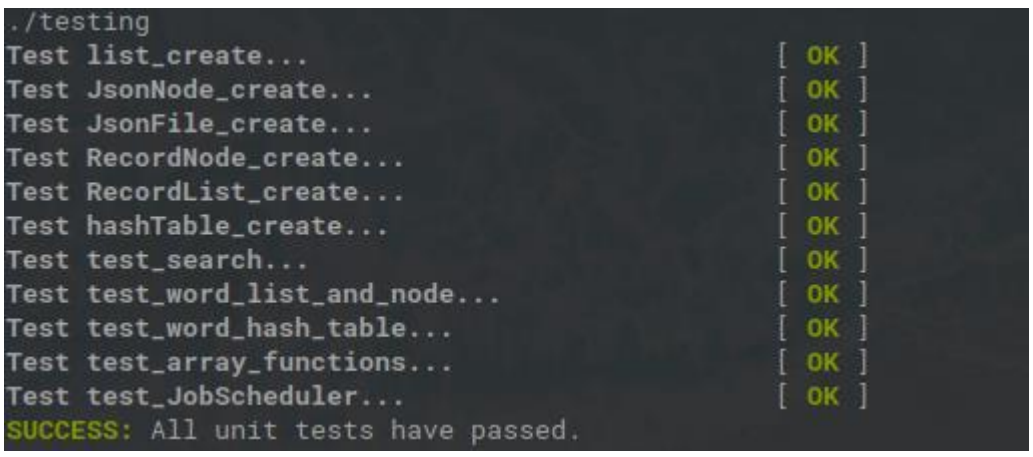
Η συνάρτηση που εκτελεί το train (CreateTrainAndTest), δημιουργεί τον Job Scheduler, ο οποίος με την σειρά του δημιουργεί τα Threads. Τα νήματα, βρίσκονται σε αναμονή μέχρι να προστεθεί κάποια διεργασία στην ουρά. Για κάθε βάρος του Classifier, τα Threads μοιράζονται το Batch, και εκτελούν ξεχωριστά τις μαθηματικές διαδικασίες που απαιτούνται για τον υπολογισμό του συγκεκριμένου βάρους. Όταν ένα από τα Threads ολοκληρώσει την δουλεία του, ενημερώνει με κατάλληλο τρόπο μια μεταβλητή. Το τελευταίο νήμα που θα ολοκληρώσει, ενημερώνει την μεταβλητή, καθώς και τον Job Scheduler ότι η διαδικασία έχει ολοκληρωθεί. Στο σημείο αυτό, ο Job Scheduler γεμίζει ξανά την ουρά με τις απαραίτητες διεργασίες για τον υπολογισμό του επόμενου βάρους. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να ολοκληρωθεί το train. Τα Threads μπαίνουν σε αναμονή έως το επόμενο train.

Βελτιώσεις Δεύτερου Μέρους

1. Η αναπαράσταση των διανυσμάτων των αρχείων έχει αντικατασταθεί από sparce-matrix (σε αντίθεση με τον απλό πίνακα που υπήρχε πριν). Ως αποτέλεσμα, η διαδικασία του training απαιτεί σημαντικά λιγότερο χρόνο, ενώ οι δομές καταλαμβάνουν λιγότερο χώρο.
2. Επίλυση warnings από την χρήση της συνάρτησης snprintf() και των test.
3. Υλοποίηση συνάρτησης επίλυσης Transitivity.

Unit Testing

Έχει γίνει unit testing των συναρτήσεων της εργασίας χρησιμοποιώντας την βιβλιοθήκη acutest.h, η οποία και έχει συμπεριληφθεί στα αρχεία της εργασίας. Ο έλεγχος έχει επίσης ενσωματωθεί με το github μέσω της λειτουργείας github actions, και πραγματοποιείται κάθε φορά που γίνεται push στο repository. Τα αποτελέσματα του είναι τα ακόλουθα:



Compiling - Εκτέλεση
Το compile του κώδικα γίνεται με την εντολή «make» καθώς έχει υλοποιηθεί makefile. Γίνεται separate compilation των αρχείων για να παραχθεί το εκτελέσιμο πρόγραμμα. Η εκτέλεση γίνεται με τον ακόλουθο τρόπο: make run: Ο βασικός τρόπος εκτέλεσης του προγράμματος. Εκτελεί την εντολή:
./project3 -x Datasets -c sigmod_large_labelled_dataset.csv

Υπάρχουν επίσης αρκετές επιλογές οι οποίες τρέχουν το πρόγραμμα με χρήση valgrind η gdb όπως: make valgrind, make valgrind_extreme.

Κουμερτάς Σταύρος 1115201600231
Κοροβέσης Παναγιώτης 1115201700056
Κοροβέσης Χαράλαμπος 1115201700057