ΕΡΓΑΣΙΑ 3 - Ερωτήσεις κορυφαίων κ και κορυφογραμμής

Νεφέλη-Ελένη Κατσιλέρου

ΑΜ : 4385

Μέρος 1 : Αλγόριθμος A top-k-join

Ξεκινώντας θα εξηγήσω τις μεταβλητές που έχω ορίσει και χρησιμοποιώ στον κώδικα :

- Counter : κρατάει τον αριθμό των επαναλήψεων μεταξύ των processing male και female records.
- Males-hash και females_hash : dictionairies που αποθηκεύουν τις male και female records αντίστοιχα , με κλειδί την ηλικία.
- MaxWeightMen και maxWeightWoman : Κρατάνε τα max weights που έχουν μετρηθεί μέχρι στιγμής για τα male και female records.
- CurrentWeightMen και currenWeightWomen : Αποθηκεύουν τα τρέχοντα weights που επεξεργάζονται για τα male και female records.
- MenViewCount και womenViewCount : Μετρητές για να κρατάνε τον αριθμό των male και female records που έχουν επεξεργαστεί , χρησιμοποιούνται για την αρχικοποίηση των max weights.
- Max_heap : Μέγιστος σορός (που υλοποιείται χρησιμοποιώντας έναν min heap με αρνητικά βάρη) για την αποθήκευση των κορυφαίων k ζευγών με βάση το συνδυασμένο βάρος τους.
- Threshold : Το ελάχιστο όριο συνδυασμένου βάρους για να θεωρηθεί ζευγάρι.
- Results_generated : Μετρητής για τον αριθμό των αποτελεσμάτων μέχρι στιγμής.
- CurrentLine : Αποθηκεύει την τρέχουσα γραμμή που είναι υπό επεξεργασία απο το αρχείο.
- MaleLineGenerator και femaleLineGenerator : Generators για την ανάγνωση γραμμών απο τα ταξινομημένα male και female αρχεία.
- Seen_pairs : Για την παρακολούθηση των ζευγών που έχουν ήδη εμφανιστεί για την αποφυγή επαναλήψεων.

Αρχικά οι συναρτήσεις read_sorted_file1() και read_sorted_file2() διαβάζουν τις γραμμές απο τα αρχεία που δίνω σαν input (males.txt και females.txt). Χρησιμοποιούνται για τη δημιουργία generators που παράγουν γραμμές για επεξεργασία.

Η συνάρτηση extract_fields() εξάγει τα σχετικά πεδία από μια γραμμή του αρχείου εισόδου. Διαχωρίζει τη γραμμή με κόμματα και ανακτά το αναγνωριστικό, την ηλικία και

το βάρος. Ελέγχει επίσης αν το άτομο είναι παντρεμένο ή ανήλικο και παραλείπει τέτοιες εγγραφές όπως ζητείται στην εκφώνηση.

Στη συνέχεια η συνάρτηση top_k_join() υλοποιεί τον αλγόριθμο top-k-join. Αρχικά ρυθμίζει τους generators για την ανάγνωση των ταξινομημένων αρχείων και αρχικοποιεί τις παγκόσμιες μεταβλητές. Μετά εναλλάσσεται μεταξύ της επεξεργασίας των male και female records με τη χρήση του counter. Χρησιμοποιεί τη συνάρτηση extract_fields() για την εξαγωγή των σχετικών πεδίων (id, weight,age) από την τρέχουσα γραμμή. Ενημερώνει τους πίνακες (males_hash και females_hash) με τις εξαχθείσες εγγραφές καθώς επίσης ενημερώνει και το max weight και το threshold με βάση τα τρέχοντα βάρη που υποβάλλονται σε επεξεργασία. Ελέγχει για ταιριαστές εγγραφές στον πίνακα και υπολογίζει το συνδυασμένο βάρος. Πιο συγκεκριμένα, θα εξηγήσω το μπλοκ του κώδικα που είναι υπεύθυνο για την ένωση male και female records ίδιας ηλικίας. Αρχικά ελέγχω αν υπάρχει κάποιο male record στο hash_table (males_hash) που έχει ίδια ηλικία με το τρέχον female record που είναι υπό επεξεργασία.Εάν δεν υπάρχουν males της ίδιας ηλικίας , η join δεν εκτελείται. Έπειτα επαναληπτικά περνάμε από όλα τα female records που έχουν την συγκεκριμένη ηλικία και για κάθε female record ανακτάται το female_weight και female_record.Η φωλιασμένη for περνάει όλα τα male records που έχουν την ίδια ηλικία και για κάθε male record ανακτά το male_weight και male_record. Για κάθε ζεύγος αρσενικών και θηλυκών εγγραφών με την ίδια ηλικία, τα βάρη τους αθροίζονται για να προκύψει το total_weight.Στη συνέχεια ελέγχουμε αν το συνδυασμένο βάρος (total_weight) είναι μεγαλύτερο ή ίσο με το τρέχον όριο. Το threshold ενημερώνεται δυναμικά με βάση τα μέγιστα βάρη που έχουν προκύψει μέχρι στιγμής. Δημιουργείται ένα ζεύγος από την εγγραφή male_record και την εγγραφή female_record.Το ζεύγος ελέγχεται σε σχέση με το σύνολο seen_pairs για να διασφαλιστεί ότι δεν έχει ήδη υποβληθεί σε επεξεργασία. Εάν δεν έχει ξαναγίνει, προστίθεται στο σύνολο seen_pairs για να το παρακολουθεί. Το ζεύγος, μαζί με το αρνητικό συνδυασμένο βάρος του (-total_weight), ωθείται στον μέγιστο σωρό (max_heap). Η χρήση του αρνητικού βάρους ουσιαστικά επιτρέπει στην υλοποίηση min-heap του heapq να λειτουργεί ως max-heap, εξασφαλίζοντας ότι τα ζεύγη με τα υψηλότερα συνδυασμένα βάρη βρίσκονται στην κορυφή. Ίδια είναι η διαδικασία και του male process.Τέλος εξάγει τα κορυφαία k ζεύγη από το σωρό.

Μέρος 2 : Αλγόριθμος B top-k-join

Οι μεταβλητές που έχω χρησιμοποιήσει είναι οι εξής :

- K : Ο αριθμός των ζευγαριών που πρέπει να ανακτηθούν και εισάγεται από τον χρήστη.
- Males_sorted και females_sorted : Τα ονόματα αρχείων των ταξινομημένων αρχείων εισόδου που περιέχουν ανδρικές και γυναικείες εγγραφές, αντίστοιχα.

- Start_time και end_time : Χρησιμοποιούνται για τη μέτρηση του χρόνου εκτέλεσης του αλγορίθμου.

Αρχικά η συνάρτηση read_sorted_file() διαβάζει μία-μία τις γραμμές του input file και χρησιμοποιεί έναν generator για την απόδοση κάθε γραμμής, επιτρέποντας την αποτελεσματική επανάληψη του αρχείου χωρίς να το φορτώνει εξ ολοκλήρου στη μνήμη.

Η συνάρτηση extract_fields() εξάγει τα σχετικά πεδία (id, instance_weight και age) από μια γραμμή δεδομένων. Διασφαλίζει ότι λαμβάνονται υπόψη μόνο τα άγαμα άτομα ηλικίας 18 ετών και άνω. Επιστρέφει τιμές None εάν η εξαγωγή δεδομένων αποτύχει ή εάν δεν πληρούνται οι προϋποθέσεις.

Στη συνέχεια η συνάρτηση alternative_top_k_join() εφαρμόζει τον αλγόριθμο top-k join για την εύρεση των κορυφαίων k ζευγών με τα υψηλότερα συνδυασμένα βάρη.Χρησιμοποιεί δύο λεξικά, males_hash και females_hash, για να αποθηκεύει εγγραφές ανδρών και γυναικών με ευρετήριο την ηλικία. Χρησιμοποιεί ένα min-heap για να παρακολουθεί τα κορυφαία k ζεύγη με τα υψηλότερα συνδυασμένα βάρη. Πιο συγκεκριμένα , τα male records διαβάζονται πρώτα και δημιουργείται ένας hash table (males_hash), όπου κάθε κλειδί είναι μια ηλικία και η τιμή είναι μια λίστα πλειάδων που περιέχει το βάρος και το αναγνωριστικό των male records αυτής της ηλικίας. Στη συνέχεια διαβάζονται τα female records και για κάθε female record, ο αλγόριθμος ελέγχει αν υπάρχουν male records της ίδιας ηλικίας στο hash table. Εάν βρεθούν ταιριαστές εγγραφές ανδρών, υπολογίζεται το συνδυασμένο βάρος κάθε ζεύγους ανδρών-γυναικών. Αυτά τα ζεύγη στη συνέχεια σπρώχνονται σε έναν ελάχιστο σωρό, εάν είναι μεταξύ των κορυφαίων k ζευγών με τα υψηλότερα συνδυασμένα βάρη. Μετά την επεξεργασία όλων των εγγραφών, τα κορυφαία k ζεύγη εξάγονται από τον ελάχιστο σωρό. Τα ζεύγη αυτά ταξινομούνται κατά φθίνουσα σειρά με βάση τα συνδυασμένα βάρη τους και επιστρέφονται ως αποτέλεσμα.

<u>Μέρος 3 : Γραπτό μέρος</u>

Το part1 δίνει τα εξής αποτελέσματα :

Για K=1 , t = 0,004449 sec

Για K=2 , t = 0,007272 sec

Για K=5 , t = 0,023799 sec

Για K=10 , t = 0,123272 sec

Για K=20 , t = 0,756557 sec

Για K=50 , t = 6,519514 sec

Για K=100 , t = 29,143992 sec

Το part2 δινει τα εξής αποτελέσματα :

Για K=1 , t = 2,828081 sec

Για K=2 , t = 2,812684 sec

Για K=5 , t = 2,824650 sec

Για K=10 , t = 2,819679 sec

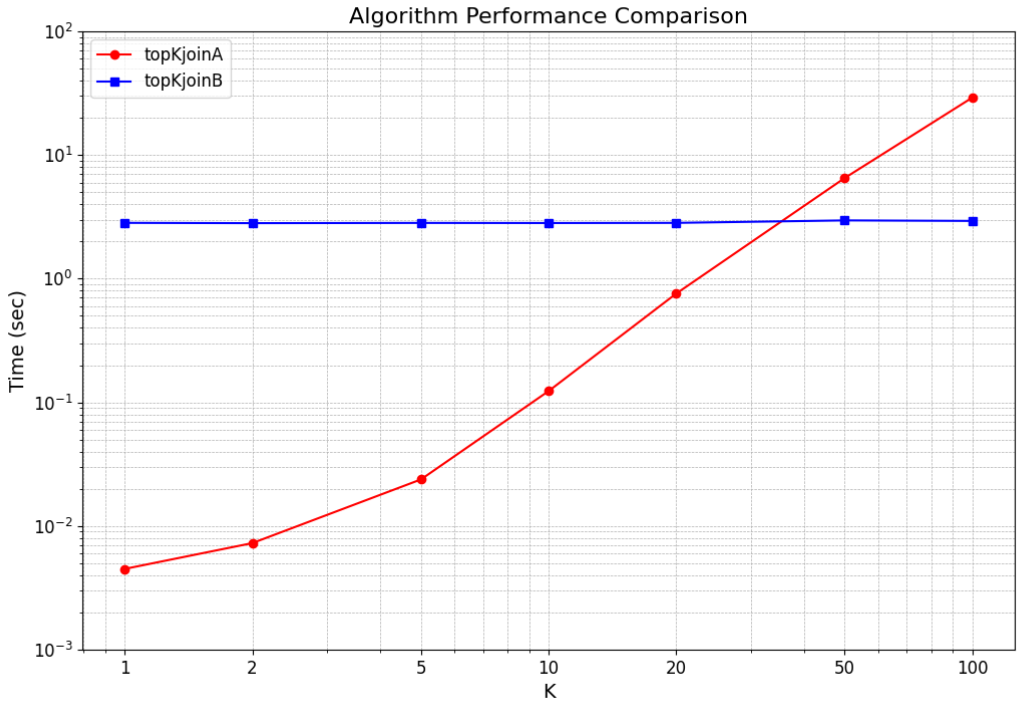
Για K=20 , t = 2,286865 sec

Για K=50 , t = 2,959532 sec

Για K=100 , t = 2,931836 sec

Παρακάτω παρατίθεται το διάγραμμα το οποίο δείχνει το χρόνο εκτέλεσης κάθε αλγορίθμου για καθεμία απο τις τιμές του K.

Το διάγραμμα το έφτιαξα με το img.py αρχείο (θα υπάρχει στο zip)



Αλγόριθμος topKjoinA:

Σε χαμηλότερες τιμές του K (συγκεκριμένα, K=1,2,5) ο αλγόριθμος topKjoinA αποδίδει πολύ αποτελεσματικά, με χρόνους εκτέλεσης της τάξης των χιλιοστών του δευτερολέπτου (0,0045, 0,0073 και 0,0238 δευτερόλεπτα, αντίστοιχα).

Καθώς το K αυξάνεται, ο χρόνος εκτέλεσης του topKjoinA αυξάνεται σημαντικά. Στο

K =10, ο χρόνος εκτέλεσης εκτοξεύεται στα 0,1233 δευτερόλεπτα και συνεχίζει να αυξάνεται απότομα. Για πολύ υψηλές τιμές του K (K=50 και K=100), ο χρόνος εκτέλεσης γίνεται πολύ μεγαλύτερος, φτάνοντας τα 6,52 δευτερόλεπτα και 29,14 δευτερόλεπτα, αντίστοιχα. Αυτό υποδηλώνει ότι η απόδοση της topKjoinA υποβαθμίζεται ραγδαία με την αύξηση των K, υποδεικνύοντας ότι μπορεί να έχει μη γραμμική πολυπλοκότητα που αυξάνεται γρήγορα με το K.

Αλγόριθμος topKjoinB:

Σε αντίθεση με τον topKjoinA, ο αλγόριθμος topKjoinB παρουσιάζει αξιοσημείωτη συνέπεια στο χρόνο εκτέλεσής του σε διαφορετικές τιμές των K. Για όλες τις τιμές του K, ο χρόνος εκτέλεσης παραμένει στο εύρος περίπου 2,81 έως 2,96 δευτερόλεπτα. Αν και υπάρχουν μικρές διακυμάνσεις στους χρόνους εκτέλεσης (π.χ. μια μικρή πτώση στο K=2 και K=10, και μια μικρή αύξηση σε K=50), οι μεταβολές αυτές είναι ελάχιστες και δεν παρουσιάζουν σημαντική τάση καθώς K αυξάνεται. Η σταθερή απόδοση του topKjoinB υποδηλώνει ότι κλιμακώνεται καλά με το K και είναι πιθανότατα βελτιστοποιημένη για να χειρίζεται διαφορετικές τιμές του K χωρίς σημαντική υποβάθμιση των επιδόσεων. Αυτό θα μπορούσε να σημαίνει μια πιο αποτελεσματική αλγοριθμική προσέγγιση ή καλύτερο χειρισμό μεγαλύτερων συνόλων δεδομένων.

Για χαμηλές τιμές του K, το topKjoinA είναι πολύ ταχύτερο από το topKjoinB. Αυτό δείχνει ότι η topKjoinA μπορεί να είναι προτιμότερη για σενάρια όπου K είναι μικρό και απαιτούνται υψηλές επιδόσεις.

Όπως το K αυξάνεται, το topKjoinB γίνεται πιο πλεονεκτικό λόγω της σταθερής απόδοσής του. Η απόδοση του topKjoinA υποβαθμίζεται σημαντικά με μεγαλύτερα K, καθιστώντας το λιγότερο κατάλληλο για εφαρμογές όπου το K μπορεί να είναι μεγάλο.

Συμπερασματικά , το topKjoinA είναι κατάλληλο για σενάρια με μικρές τιμές για το K όπου η γρήγορη εκτέλεση είναι σημαντική. Ενώ το topKjoinB είναι ιδανικό για εφαρμογές όπου το K μπορεί να μεταβάλλεται σημαντικά ή να είναι μεγάλο, λόγω της σταθερής και προβλέψιμης απόδοσής του.