Κ23α - Ανάπτυξη Λογισμικού Για Πληροφοριακά Συστήματα

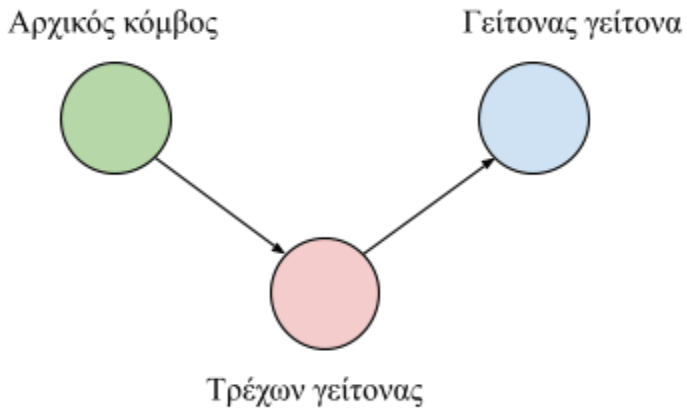
Χειμερινό Εξάμηνο 2023– 2024

Άσκηση 2 - Παράδοση: Δευτέρα 4 Δεκεμβρίου 2023

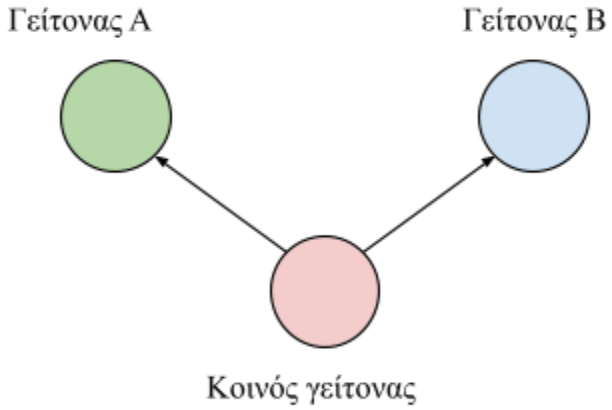
Βελτιστοποιήσεις NNDescennt

Local Join

Μία παρατήρηση αφορά τον τρόπο που βλέπουμε τους υπολογισμούς που πρέπει να κάνουμε. Για κάθε κόμβο, θα πρέπει να υπολογίσουμε την απόσταση από τους γείτονες των γειτόνων, που φαίνεται ως εξής:



Αφού όλοι υπολογισμοί θα πραγματοποιούνται κατά μήκος των δύο ακμών του (κόκκινου) γείτονα, μπορούμε να αντιστρέψουμε την οπτική μας και να εστιάσουμε σε αυτόν. Από την οπτική του κόκκινου κόμβου, ο στόχος είναι να εξετάσουμε αν ο γαλάζιος και ο πράσινος κόμβος θα πρέπει να προστεθούν ο καθένας στη λίστα των K εγγύτερων γειτόνων του άλλου. Με τον τρόπο αυτό αποφεύγουμε τη διπλή επεξεργασία καθώς εξετάζουμε τους γείτονες γειτόνων του γαλάζιου, αλλά και τους γείτονες γειτόνων του πράσινου. Επίσης, η επεξεργασία μετατρέπεται σε αμιγώς τοπική που μπορεί να πραγματοποιηθεί για κάθε κόμβο ανεξάρτητα.



Ανακεφαλαιώνοντας, η κατασκευή του γράφου αρχικά περιλαμβάνει τη δημιουργία ενός συνόλου από "γείτονες" (με άμεσους και αντίστροφους γείτονες) για κάθε κόμβο, και στη συνέχεια την ενημέρωση του γράφου μέσω υπολογισμού όλων των αποστάσεων ανά ζεύγη σε κάθε σύνολο άμεσων γειτόνων. Μετά το πρώτο βήμα του υπολογισμού των αντίστροφων γειτόνων, ο υπολογισμός μπορεί να εκτελεστεί τοπικά για κάθε κόμβο, αποθηκεύοντας προσωρινά τις απαραίτητες τροποποιήσεις για κάθε σύνολο γειτόνων και εκτελώντας τις στο τέλος του βήματος επανάληψης. Η τεχνική αυτή, ενώ δεν διαφοροποιεί τον αλγόριθμο, επιτρέπει την παρακολούθηση για το ποιοι κόμβοι έχουν "νέους" γείτονες και συνεπώς χρειάζονται (οι νέοι γείτονες) υπολογισμούς απόστασης από τους άλλους γείτονες.

Σταδιακή αναζήτηση (Incremental Search)

Καθώς εκτελείται ο αλγόριθμος, όλο και λιγότεροι κόμβοι θα προσθαφαιρούνται στον KNN γράφο σε κάθε επανάληψη. Είναι άσκοπο να πραγματοποιείται ένα πλήρες τοπικό join σε κάθε επανάληψη, αφού αρκετά ζευγάρια έχουν ήδη συγκριθεί σε προηγούμενες επαναλήψεις. Μπορεί να χρησιμοποιηθεί η εξής στρατηγική σταδιακής αναζήτησης για να αποφευχθούν περιττοί υπολογισμοί:

- Προσθήκη μιας boolean σημαίας σε κάθε κόμβο στις KNN λίστες. Η σημαία έχει αρχικά την τιμή true όταν ο κόμβος εισέρχεται στη λίστα.
- Στο τοπικό join, δύο κόμβοι συγκρίνονται μόνο αν τουλάχιστον ένας από τους 2 είναι νέος. Μετά τη συμμετοχή του κόμβου σε ένα τοπικό join, η σημαία παίρνει την τιμή false.

Δειγματοληψία

Υπάρχουν ακόμη 2 θέματα με τη μέθοδο NN-Descent. Το πρώτο είναι ότι το κόστος του τοπικού join μπορεί να είναι μεγάλο όταν το K είναι μεγάλο. Ακόμη και αν χρησιμοποιούνται μόνο κόμβοι στο KNN γράφο για τοπικό join, το κόστος κάθε επανάληψης είναι K^2N συγκρίσεις ομοιότητας. Η κατάσταση χειροτερεύει αν συνυπολογίσουμε και τους αντίστροφους γείτονες, καθώς δεν υπάρχει όριο στον αριθμό τους. Το δεύτερο θέμα αφορά κόμβους που είναι συνδεδεμένοι (ως γείτονες γειτόνων) σε περισσότερους από ένα κόμβους. Η δειγματοληψία μπορεί να χρησιμοποιηθεί για να αντιμετωπιστούν και τα 2 προβλήματα.

- Πριν από το τοπικό join, δειγματοληπτούμε pK κόμβους από τους συνολικούς κόμβους που έχουν τη σημαία true και επομένως αναμενόταν να χρησιμοποιηθούν στις συγκρίσεις, $p \in (0, 1]$. Μόνο οι συγκεκριμένοι κόμβοι που επελέγησαν θα σημανθούν ως false στις επόμενες επαναλήψεις.
- Οι αντίστροφες KNN λίστες κατασκευάζονται ξεχωριστά. Οι λίστες αυτές δειγματοληπτούνται με τον ίδιο τρόπο, επομένως η κάθε μία θα έχει το πολύ pK κόμβους.
- Το τοπικό join υπολογίζεται στους επιλεχθέντες κόμβους και μεταξύ των επιλεχθέντων κόμβων και παλιών κόμβων.

Τα αντικείμενα που είναι true, αλλά δεν έχουν επιλεγεί στην τρέχουσα επανάληψη, είναι πιθανό να επιλεγούν στη δειγματοληψία μιας μελλοντικής δειγματοληψίας, αν δεν αντικατασταθούν από καλύτερες προσεγγίσεις.

Τόσο η ακρίβεια, όσο και ο χρόνος αναμένεται να μειωθούν με ρυθμό δειγματοληψίας <1, αν και ο χρόνος αναμένεται να μειωθεί ταχύτερα. Η παράμετρος p μπορεί να χρησιμοποιηθεί για έλεγχο της ισορροπίας ανάμεσα σε ακρίβεια και χρόνο.

Πρόωρος τερματισμός

Το κριτήριο φυσικού τερματισμού είναι όταν ο KNN γράφος δεν μπορεί πλέον να βελτιωθεί. Στην πράξη, αριθμός των ενημερώσεων του KNNG μειώνεται δραστικά μετά από κάθε επανάληψη. Στις τελευταίες επαναλήψεις, είναι πιθανόν να μην πραγματοποιείται υπολογιστική εργασία, αλλά κυρίως διαχείριση των εσωτερικών δομών.

Μπορούν να χρησιμοποιηθούν τα ακόλουθα κριτήρια πρόωρου τερματισμού για να τερματίσει ο αλγόριθμός, όταν οι επιπλέον επαναλήψεις δεν αναμένεται να επιφέρουν σημαντική βελτίωση στην ακρίβεια: μετράται ο αριθμός των ενημερώσεων στις KNN λίστες σε κάθε επανάληψη και τερματίζεται ο αλγόριθμος όταν πέσει κάτω από δKN, όπου δ μία παράμετρος ακρίβειας (περίπου όσο το κλάσμα των αληθινών KNN που επιτρέπεται να απωλεσθεί λόγω πρόωρου τερματισμού). Μπορείτε να χρησιμοποιήσετε διαφορετικές τιμές του δ, ξεκινώντας από 0,001.

Περαιτέρω βελτιστοποιήσεις

Μπορείτε να πειραματιστείτε με περαιτέρω βελτιστοποιήσεις που περιγράφονται στο [3].

Προδιαγραφές κώδικα

Ο κώδικας που θα υλοποιηθεί θα πρέπει να είναι μία βιβλιοθήκη που να παρέχει τις εξής δυνατότητες

- χειρισμός δεδομένων αυθαίρετου αριθμού διαστάσεων
- χειρισμός δεδομένων αυθαίρετου αριθμού αντικειμένων
- δυνατότητα εύρεσης των k εγγύτερων γειτόνων για ένα ή για όλα τα μέλη του συνόλου
- δυνατότητα χρήσης εναλλακτικής μετρικής ομοιότητας (π.χ. ευκλείδεια, Manhattan απόσταση κ.α.)

Παράδοση εργασίας

Η εργασία είναι ομαδική, **2 ή 3 ατόμων**.
Γλώσσα υλοποίησης: C / C++ χωρίς χρήση stl.
Περιβάλλον υλοποίησης: Linux (gcc > 9.4+).

Παραδοτέα: Η παράδοση της εργασίας θα γίνει με βάση το τελευταιο commit πριν την προθεσμία υποβολής στο git repository σας. **Η χρήση git είναι υποχρεωτική**.
Στο αρχείο README.md θα αναφέρονται τα εξής:

- Ονοματεπώνυμα και ΑΜ των μελών της ομάδας
- Αναφορά στο ποιο μέλος της ομάδας ασχολήθηκε με ποιο αντικείμενο

Επιπλέον, εκτός από τον πηγαίο κώδικα, θα παραδώσετε μια σύντομη αναφορά, με τις σχεδιαστικές σας επιλογές καθώς και να εφαρμόσετε ελέγχους ως προς την ορθότητα του λογισμικού με τη χρήση ανάλογων βιβλιοθηκών (Software testing). Η ορθότητα τυχόν μεταβολών θα ελέγχεται με αυτοματοποιημένο τρόπο σε κάθε commit/push μέσω github actions.

Αναφορές

1. Wei Dong, Charikar Moses, and Kai Li. 2011. Efficient k-nearest neighbor graph construction for generic similarity measures. In Proceedings of the 20th international conference on World wide web (WWW '11). Association for Computing Machinery, New York, NY, USA, 577–586. https://doi.org/10.1145/1963405.1963487
2. Περιγραφή υλοποίησης του αλγορίθμου σε python για το project PyNNDescent https://pynndescent.readthedocs.io/en/latest/how_pynndescent_works.html
3. Dan Kluser and Jonas Bokstaller and Samuel Rutz and Tobias Buner. Fast Single-Core K-Nearest Neighbor Graph Computation, https://doi.org/10.48550/arXiv.2112.06630