Κ23α - Ανάπτυξη Λογισμικού Για Πληροφοριακά Συστήματα

Χειμερινό Εξάμηνο 2023– 2024

Άσκηση 1 – Παράδοση: Δευτέρα 6 Νοεμβρίου 2023

Αναζήτηση εγγύτερου γείτονα (Nearest Neighbor Search)

Η αναζήτηση εγγύτερου γείτονα (nearest neighbor search - NSS), είναι ένα πρόβλημα βελτιστοποίησης εύρεσης του σημείου σε ένα δεδομένο σύνολο που είναι εγγύτερα (ή πιο όμοιο) σε ένα δεδομένο σημείο. Η εγγύτητα συνήθως εκφράζεται ως μια μετρική ανομοιότητας (dissimilarity), δηλαδή όσο λιγότερο όμοια είναι τα αντικείμενα, τόσο μεγαλύτερη η τιμή της ανομοιότητας.

Τυπικά, το πρόβλημα εύρεσης εγγύτερου γείτονα ορίζεται ως εξής: έστω ένα σύνολο *S* σημείων σε ένα χώρο *M* και ένα σημείο επερώτησης $q \in M$. Βρείτε το σημείο στο *S* που είναι πιο κοντά στο *q*. Συνήθως ο *M* είναι μετρικός χώρος και η ανομοιότητα εκφράζεται ως μετρική απόστασης, που είναι ένα συμμετρικό μέγεθος και ικανοποιεί την τριγωνική ανισότητα. Επίσης μπορεί να είναι χώρος *d* διαστάσεων, όπου η ανομοιότητα μετράται χρησιμοποιώντας μεταξύ άλλων την Ευκλείδεια απόσταση μεταξύ των διανυσμάτων.

k-Nearest Neighbors (KNN)

Μια γενίκευση του προβλήματος αναζήτησης του εγγύτερου γείτονα είναι η αναζήτηση των *k* εγγύτερων γειτόνων, όπου χρειάζεται να βρεθούν τα *k* εγγύτερα σημεία (k-Nearest Neighbor Search - KNN). Μία συνήθης τεχνική είναι η χρήση του γράφου εγγύτερων γειτόνων (k-Nearest Neigbors Graph - KNNG). Ο KNNG γράφος για ένα σύνολο αντικειμένων V είναι ένας κατευθυντικός γράφος με ένα σύνολο κορυφών V και μία ακμή από κάθε $v \in V$ στα K πιο όμοια αντικείμενα στο V, δεδομένης μιας μετρικής ομοιότητας.

Η κατασκευή του KNNG μέσω brute-force τεχνικών έχει πολυπλοκότητα $O(n^2)$ και μπορεί πρακτικά να εφαρμοστεί μόνο για μικρά σύνολα δεδομένων. Μπορεί

Προσεγγιστική εύρεση KNN

Επειδή η αναζήτηση της ακριβούς λύσης στο πρόβλημα σε τεράστιους όγκους δεδομένων είναι απαγορευτική σε χρονικό κόστος, έχουν προταθεί και αναζητηθεί προσεγγιστικοί αλγόριθμοι που ανταλλάσσουν την απόλυτη ακρίβεια με σημαντικά καλύτερο χρόνο. Αναλυτικότερα ο αλγόριθμος αναζήτησης θα επιστρέψει ένα σημαντικό ποσοστό από τους εγγύτερους γείτονες (π.χ. 90-95%) σε ένα κλάσμα του χρόνου που χρειάζεται για την εύρεση της ακριβούς λύσης.

Αλγόριθμος NN-descent

Ο αλγόριθμος NN-descent [1] είναι ένας προσεγγιστικός αλγόριθμος κατασκευής του KNNG και εύρεσης των K εγγύτερων γειτόνων. Βασίζεται στην εξής αρχή: ο γείτονας ενός γείτονα είναι αρκετά πιθανό να είναι επίσης γείτονας. Με άλλα λόγια, αν έχουμε μία προσέγγιση των K εγγύτερων γειτόνων σε ένα σημείο, τότε μπορούμε να βελτιώσουμε την προσέγγισή μας, εξερευνώντας για κάθε σημείο τους γείτονες των γειτόνων, όπως ορίζονται από την τρέχουσα προσέγγιση.

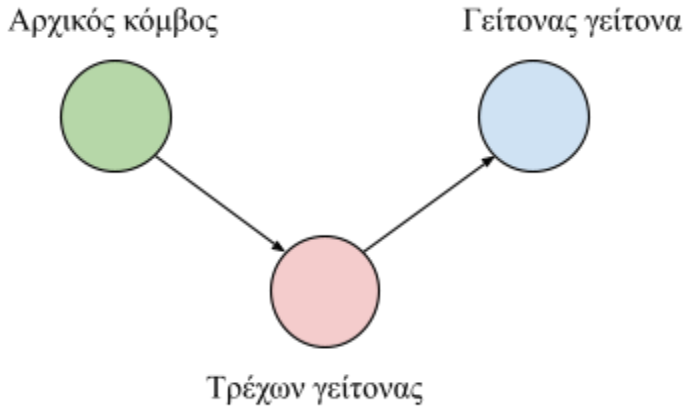
Ο βασικός αλγόριθμος κατασκευής του γράφου θα έχει ως εξής:

1. Υλοποίησε έναν τυχαίο γράφο (σύνδεσε κάθε κόμβο με K τυχαίους κόμβους
2. Για κάθε κόμβο:
   Μέτρησε την απόσταση από τον κόμβο στους γείτονες των γειτόνων
   Αν κάποιοι γείτονες γειτόνων είναι εγγύτεροι, ενημέρωσε τον γράφο ανάλογα και διατήρησε μόνο τους K εγγύτερους γείτονες
3. Αν τροποποιήθηκε ο γράφος πήγαινε στο βήμα 2, αλλιώς τερμάτισε

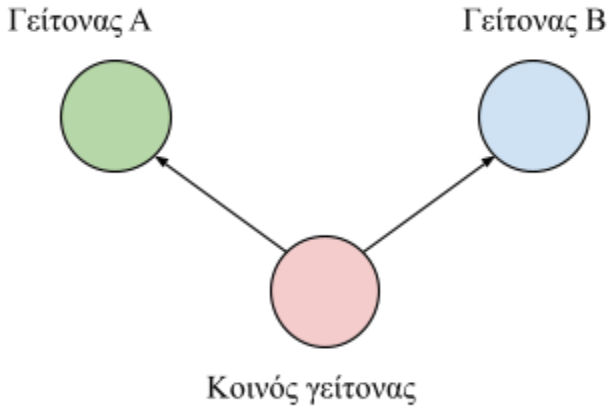
Επειδή η κατασκευή αφορά μη κατευθυντικό γράφο, θα πρέπει όταν αναζητούμε τους γείτονες ενός κόμβου να λαμβάνουμε υπόψη όχι μόνο τους K κόμβους προς τους οποίους ο κόμβος έχει ακμές, αλλά και όλους τους άλλους κόμβους που έχουν επιλέξει τον εν λόγω κόμβο ως έναν από τους K γείτονες τους (αντίστροφοι εγγύτεροι γείτονες - reverse nearest neighbors). Ενώ είναι εύκολη η παρακολούθηση και ενημέρωση των K εγγύτερων γειτόνων για κάθε κόμβο, είναι σημαντικά δυσκολότερη η παρακολούθηση των αντίστροφων εγγύτερων γειτόνων για κάθε κόμβο και η ενημέρωσή τους καθώς ο γράφος τροποποιείται.

Για τον λόγο αυτό, μπορούμε να υπολογίσουμε το συνδυασμένο σύνολο των γειτόνων και αντίστροφων γειτόνων για κάθε κόμβο μια φορά στην αρχή κάθε επανάληψης και να χρησιμοποιούμε αυτή την πληροφορία καθώς ενημερώνουμε τον κόμβο στο παρασκήνιο. (Εναλλακτικά, η δομή του γράφου διατηρείται σταθερή για όλους τους υπολογισμούς και στη συνέχεια τροποποιείται συνολικά στο τέλος).

Μία άλλη 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.

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