



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

**Ανάλυση και οπτικοποίηση χωρικών δεδομένων
βάσει αποτελεσμάτων ερωτήσεων κατάταξης**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Αλέξανδρου Γαλάνη

Επιβλέπων : Βλάχου Ακριβή

Μέλη εξεταστικής επιτροπής : Π. Συμεωνίδης και Θ. Κωστούλας

Σάμος, [Ιούλιος 2021]

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πρόλογος και ευχαριστίες

Θα ήθελα να ευχαριστήσω την κ. Βλάχου για τις ιδέες και την βοήθεια της παρόλες τις δυσκολίες που υπήρξαν, αλλά και την οικογένεια μου που μου έδωσε κουράγιο κατά την διάρκεια της φοίτησης μου στο προπτυχιακό.

© 2020

του

Αλέξανδρου Γαλάνη

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Πίνακας περιεχομένων

Περίληψη	7
Abstract.....	9
1. Εισαγωγή	11
1.1. Αντικείμενο διπλωματικής εργασίας.....	11
1.2. Συνεισφορά και δομή της διπλωματικής.....	12
1.3. Βασικές γνώσεις κατανόησης της διπλωματικής	13
2. Χωρικά δεδομένα.....	14
2.1. Χωρικά Δεδομένα	14
2.2. Μοντέλα γεω-χωρικών δεδομένων	15
2.3. Δομές δεδομένων χαρακτηριστικών	16
2.4. Περιγραφή γεωγραφικών χαρακτηριστικών	16
2.5. Σημασιολογικές σχέσεις γεω-χωρικών αντικειμένων	17
3. Top-k Ερωτήσεις Κατάταξης.....	19
3.1. Εισαγωγή.....	19
3.2. Αλγόριθμος Top-k.....	19
3.3. Top-k χωρικές επερωτήσεις προτιμήσεων	20
3.3.1. Βιβλιογραφική επισκόπηση.....	22
3.3.2. Αξιολόγηση χωρικών ερωτήσεων σε R-δέντρα	24
3.3.3. Χωρικές Ερωτήσεις βασισμένες σε χαρακτηριστικά	25
4. Γράφοι και PageRank	27
4.1. Ορισμοί.....	28
4.1.1. Γράφος.....	28
4.1.2. Μονοπάτια γράφων	28
4.2. Τύποι γράφων.....	29
4.2.1. Τι είναι οι τύποι γράφων	29
4.2.2. Σταθμισμένο γράφημα.....	29
4.3. Κατευθυνόμενος γράφος	29
4.4. PageRank.....	29
4.4.1. Η ιδέα του αλγόριθμου PageRank.....	30
4.4.2. Τρόπος λειτουργίας του αλγόριθμου PageRank	30
4.4.3. Ιστορία του PageRank	31
4.4.4. Παράδειγμα λειτουργίας του PageRank στην πράξη	31

5.	Περιγραφή εφαρμογής	34
5.1.	Top-k αλγόριθμος στην εφαρμογή	34
5.2.	Πώς χρησιμοποιεί τους γράφους η εφαρμογή.....	38
5.3.	Χρήση του αλγόριθμου PageRank από την εφαρμογή	40
5.4.	Οπτικοποίηση δεδομένων	40
6.	Λεπτομέρειες Υλοποίησης	41
6.1.	Δεδομένα εισόδου της εφαρμογής	41
6.2.	Επεξεργασία δεδομένων.....	42
6.3.	Ουρά προτεραιότητας (Priority Queue)	46
6.4.	Βιβλιοθήκες για οπτικοποίηση γράφων	47
6.5.	Πώς λειτουργεί η JGraphT	47
6.5.1.	Σχεδιασμός της βιβλιοθήκης JGraphT	47
6.5.2.	Δομή του γράφου	51
6.5.3.	Υποκείμενη υλοποίηση	52
6.5.4.	Αλγόριθμοι	54
6.6.	Υλοποίηση με τη βιβλιοθήκη JGraphT	54
6.7.	Περιορισμοί και Λεπτομέρειες.....	56
7.	Αποτελέσματα και Οπτική Ανάλυση Δεδομένων	57
7.1.	Αποτελέσματα	57
7.1.1.	Εκτέλεση εφαρμογής (1)	57
7.1.2.	Εκτέλεση εφαρμογής (2)	58
7.1.3.	Εκτέλεση εφαρμογής (3)	59
7.1.4.	Εκτέλεση εφαρμογής (4)	61
7.2.	Τεχνική οπτικοποίησης	62
7.2.1.	Ανάλυση της εικόνας του αποτελέσματος.....	62
8.	Συμπεράσματα	67
	Βιβλιογραφία	69

Λίστα Σχημάτων

Εικόνα 1.	Παραδείγματα από Top-k χωρικές ερωτήσεις προτίμησης.....	21
Εικόνα 2.	Χωρικές επερωτήσεις με χρήση R-δένδρων	24
Εικόνα 3.	Optimal location queries.....	26
Σχήμα 4.	Κατευθυνόμενος γράφος τεσσάρων κόμβων για τον υπολογισμό του PageRank.....	32
Σχήμα 5.	Κατευθυνόμενος γράφος αναπαράστασης των καλύτερων αποτελεσμάτων βάσει του αλγόριθμου Top-k.	39
Εικόνα 6.	Γραμμογράφηση του CSV αρχείου των ενοικιαζόμενων οικημάτων Airbnb.....	41
Εικόνα 7.	Γραμμογράφηση αρχείου επιλογών ενοικιαζόμενων οικημάτων άλλων χρηστών του δικτύου	42
Εικόνα 8.	Περιβάλλον ορισμού βαρών για κάθε κριτήριο αναζήτησης.....	43
Εικόνα 9.	Εκτύπωση των καλύτερων αποτελεσμάτων	46
Εικόνα 10.	Η σχεδίαση του πυρήνα της βιβλιοθήκης JGraphT.....	48
Εικόνα 11.	Η διεπαφή (interface) Graph<V,E>. Όλες οι δοσοληψίες με τους γράφους διεκπεραιώνονται με τη χρήση αυτής της διεπαφής.....	49
Σχήμα 12.	Αποτέλεσμα της οπτικοποίησης των αποτελεσμάτων αναζήτησης ενοικιαζόμενων Airbnb από την εφαρμογή.....	63
Σχήμα 13.	Εστίαση στην περιοχή των χειρότερων αποτελεσμάτων.	64
Σχήμα 14.	Εστίαση στην περιοχή των καλύτερων αποτελεσμάτων.	64
Σχήμα 15.	Αποτελέσματα PageRank στους κόμβους των αποτελεσμάτων αναζήτησης.....	65
Σχήμα 16.	Τα χειρότερα αποτελέσματα του αλγόριθμου PageRank στους κόμβους των αποτελεσμάτων αναζήτησης	65
Σχήμα 17.	Περιβάλλον ορισμού των βαρών στις προτιμήσεις του χρήστη, έναρξης εφαρμογής του αλγόριθμου Top-k και εξαγωγή αποτελεσμάτων αναζήτησης.....	66

Λίστα Πινάκων

Πίνακας 1.	Ορολογία για την περιγραφή γεωγραφικών χαρακτηριστικών	16
Πίνακας 2.	Υπολογισμός PageRank σε διαδοχικές επαναλήψεις, για τον γράφο του Σχήματος 4.....	33
Πίνακας 3.	Ενδεικτικό σύνολο δεδομένων 6 ενοικιαζόμενων διαμερισμάτων Airbnb, για τον υπολογισμό των $k=3$ κορυφαίων προτάσεων αλγορίθμου Top-k	36
Πίνακας 4.	Υπόδειγμα «Πίνακα προτιμήσεων» του χρήστη. Βάσει των βαρών γίνεται η εφαρμογή του αλγορίθμου Top-k.....	37
Πίνακας 5.	Ενδεικτικό σύνολο των $k=3$ κορυφαίων προτάσεων του αλγορίθμου Top-k για τον κάθε ένα από τους χρήστες.....	37
Πίνακας 6.	Οι διαθέσιμες κλάσεις για υλοποίηση γράφων	50

Περίληψη

Ο στόχος της διπλωματικής εργασίας είναι η ανάπτυξη ενός συστήματος για την κατάταξη χωρικών δεδομένων βάσει των προτιμήσεων χρηστών. Συνοπτικά το πρόβλημα που επιχειρεί να επιλύσει το με την εφαρμογή προηγμένων αλγορίθμων, είναι η εύρεση πλήθους σημείων ενδιαφέροντος με χρήση χωρικών δεδομένων και κριτηρίων ενδιαφέροντος. Οι αλγόριθμοι αυτοί εφαρμόζονται σε γειτονικά σημεία ενδιαφέροντος, ώστε να δημιουργείται ένα δίκτυο σημείων, από το οποίο θα προκύψουν οι τελικές επιλογές.

Για να δημιουργηθεί αυτό, προτάθηκε η αρχιτεκτονική διαχωρισμού του συστήματος σε 5 ενότητες: (α) την ενότητα επεξεργασίας δεδομένων και την συλλογή τους από το σύστημα, (β) την κατηγοριοποίηση τους ανάλογα με τις επιλογές του χρήστη/χρηστών με βάση την τοποθεσία του και τις προσωπικές του επιλογές, (γ) την ταξινόμηση των αποτελεσμάτων, (δ) την ενότητα γραφικής απεικόνισης των δεδομένων, και (ε) την επεξεργασία για την απεικόνιση των αποτελεσμάτων.

Λέξεις-Κλειδιά: Top-k, Graph, PageRank, Recommendation systems, Geo-spatial data, Spatial queries, Graph visualization.

Abstract

The objective of my thesis is to develop a program for the classification of spatial data. To develop the software, we split it into 5 pieces, initially processing the data and collecting it from the system, then categorizing it according to the user's / users' choices based on their location and then sorting the results as well as displaying them with innovations modes of processing.

The problem that we are trying to solve with these algorithms we mentioned above, is trying to solve and to find multitude of points of interest with spatial and criteria of interest that apply to neighboring points of interest and to display them in the end.

Keywords: Top-k, Graph, PageRank, Recommendation systems, Geo-spatial data, Spatial queries, Graph visualization.

1. Εισαγωγή

Με την τεράστια αύξηση των χωρικών δεδομένων (χώρο-δεδομένων) στο διαδίκτυο τα τελευταία χρόνια αλλά και με τους ρυθμούς που αυξάνονται, εισερχόμαστε στην εποχή των μεγάλων δεδομένων (big data). Ο όγκος και η ταχύτητα μεταβολής των δεδομένων αυτών δημιουργεί την ανάγκη τα συστήματα επεξεργασίας τους να είναι ευέλικτα και υπολογιστικά ισχυρά. Η εργασία αυτή έχει σκοπό τη δημιουργία μιας προγραμματιστικής διεπαφής, η οποία επιτρέπει την εφαρμογή της σε διαφορετικά συστήματα, ώστε να δίνει λύση στα προβλήματα ταξινόμησης γεωγραφικών δεδομένων σε συνδυασμό με την προσωπική προτίμηση αναζήτησης του κάθε χρήστη ξεχωριστά αλλά και με τις προτιμήσεις ομάδων χρηστών.

Το προτεινόμενο σύστημα χαρακτηρίζεται από τη δυνατότητα που δίνει στο χρήστη να αναπροσαρμόζει τις επιθυμίες του. Σε περίπτωση που το προτεινόμενο αποτέλεσμα υποκειμενικά δεν ικανοποιεί το χρήστη, ή οι προτάσεις άλλων χρηστών δεν καλύπτουν τις υποκειμενικές επιθυμίες του, αυτός μπορεί να αναπροσαρμόσει τις παραμέτρους αναζήτησης, ώστε το νέο αποτέλεσμα να προσεγγίζει καλύτερα την επιθυμία του.

Ο ερευνητικός στόχος της εργασίας είναι η εξεύρεση μεθόδων ανάλυσης και οπτικοποίησης γεω-χωρικών δεδομένων, με χρήση καινοτόμων αλγορίθμων, οι οποίοι εφαρμόζονται σε δεδομένα διαφορετικών κλάσεων. Για παράδειγμα, το σύστημα μπορεί να υποστηρίζει την ταξινόμηση και απεικόνιση σημείων ενδιαφέροντος, όπως εστιατορίων, ξενοδοχείων, κ.ο.κ., με τρόπο τέτοιο ώστε να ανταποκρίνεται στις ανάγκες του συνόλου μιας ομάδας χρηστών, τα μέλη της οποίας κατά κανόνα έχουν διαφορετικές προτιμήσεις.

1.1. Αντικείμενο διπλωματικής εργασίας

Στην διπλωματική αυτή δημιουργήθηκε ένα σύστημα το οποίο δέχεται σαν είσοδο καταγραφές επερωτήσεων (query logs) από μία υπηρεσία γεω-εντοπισμού (Location-based service)¹ (Wikipedia, 2021). Γεωγραφικά δεδομένα σήμερα είναι διαθέσιμα από πολλαπλές πηγές, όπως υπηρεσίες ηλεκτρονικών χαρτών, διαδικτυακές υπηρεσίες (web services), δίκτυα αντικειμένων (Internet Of Things), κ.ο.κ. Τα δεδομένα αυτά προσφέρουν στο σύστημα τη γνώση της γεωγραφικής θέσης των χρηστών καθώς και κάποιες προτιμήσεις τους, οι οποίες είτε έχουν δηλωθεί ρητά είτε προκύπτουν από το προφίλ που διαμορφώνεται από τη χρήση διαφορετικών υπηρεσιών. Δεύτερη είσοδος στο σύστημα είναι ένα σύνολο σημείων ενδιαφέροντος (όπως εστιατόρια, αθλητικοί χώροι, ενοικιαζόμενα διαμερίσματα) τα οποία βρίσκονται σε μια τοποθεσία και

¹ Wikipedia, Location-based service, 2021
https://en.wikipedia.org/wiki/Location-based_service

περιγράφονται εκτός των άλλων και με κάποια ποσοτικά χαρακτηριστικά (τιμή, αξιολόγηση χρηστών, κ.λπ.).

Ο σκοπός του συστήματος είναι να αναλύει την ποιότητα των σημείων ενδιαφέροντας με βάση το πόσο αυτά είναι συναφή με τις προτιμήσεις των χρηστών του Location Based System (LBS) συστήματος.

Η ανάπτυξη του συστήματος θα πραγματοποιηθεί με τη γλώσσα προγραμματισμού Java και την υποστήριξη από βιβλιοθήκες (JGraphT και Priority Queue). Για την ταξινόμηση των αποτελεσμάτων θα χρησιμοποιηθεί η βιβλιοθήκη PageRank και η βιβλιοθήκη JGraphT για την οπτικοποίηση του γράφου των αποτελεσμάτων.

1.2. Συνεισφορά και δομή της διπλωματικής

Η συνεισφορά της διπλωματικής εργασίας στο πεδίο της οπτικοποίησης γεω-χωρικών δεδομένων συνοψίζεται στα εξής:

1. Μελετήθηκαν αλγόριθμοι δημιουργίας δομών ευρετηρίων και αναζήτησης σε αυτές.
2. Υλοποιήθηκαν εφαρμογές με καινοτόμους τρόπους ταξινόμησης.
3. Υλοποιήθηκε αλγόριθμος ταξινόμησης με PageRank.
4. Υλοποιήθηκε αλγόριθμος για την απεικόνιση του γράφου.

Η εργασία ακολουθεί τη δομή που περιγράφεται στη συνέχεια:

- Στο Κεφάλαιο 2 γίνεται αναφορά στο πεδίο των χωρικών δεδομένων, στα μοντέλα που χρησιμοποιούνται για την αναπαράστασή τους και στις αντίστοιχες δομές δεδομένων, ενώ μελετώνται και οι σημασιολογικές σχέσεις μεταξύ των γεωχωρικών αντικειμένων.
- Στο Κεφάλαιο 3 αναλύεται η χρησιμότητα του αλγόριθμου Top-k σε αναζητήσεις δεδομένων και ειδικότερα γεωχωρικών δεδομένων, με έλεγχο των επερωτήσεων από το χρήστη μέσω των προσωπικών του προτιμήσεων.
- Στο Κεφάλαιο 4 περιγράφεται η θεωρία και ο τρόπος λειτουργίας του αλγόριθμου PageRank για την κατάταξη γεωχωρικών σημείων ενδιαφέροντος.
- Στο Κεφάλαιο 5 γίνεται η περιγραφή της εφαρμογής και του τρόπου που ενσωματώθηκαν σε αυτή οι μηχανισμοί αναπαράστασης της πληροφορίας σε γράφους, βαθμονόμησης της πληροφορίας με αλγόριθμο Top-k σε συνδυασμό με τον αλγόριθμο PageRank, αλλά και οπτικοποίησης των αποτελεσμάτων.
- Το Κεφάλαιο 6 αναφέρεται στην τεχνική υλοποίηση του συστήματος με τη χρήση βιβλιοθηκών Java.
- Στο Κεφάλαιο 7 περιλήφθηκαν αποτελέσματα της εκτέλεσης της εφαρμογής, ώστε να καταγραφούν ζητήματα απόδοσης του αλγόριθμου.
- Στο Κεφάλαιο 8 καταγράφονται τα συμπεράσματα που καταγράφηκαν από την

εφαρμογή του συστήματος στα δεδομένα ελέγχου, ενώ γίνονται αναφορές σε δυνατές επεκτάσεις του με πρόσθετη λειτουργικότητα.

1.3. Βασικές γνώσεις κατανόησης της διπλωματικής

Με σκοπό να γίνει μία σύντομη εισαγωγή στις τεχνολογίες που χρησιμοποιήθηκαν στην εργασία και να τεθεί το πλαίσιο των απαιτούμενων γνώσεων για την κατανόησή της, γίνεται στη συνέχεια αναφορά σε αυτές:

1. Βάση δεδομένων: είναι μία συλλογή από *συστηματικά μορφοποιημένα* σχετιζόμενα δεδομένα, στα οποία είναι δυνατή η ανάκτηση δεδομένων μέσω αναζήτησης κατ' απαίτηση.
2. Σχεσιακή βάση δεδομένων: σύμφωνα με τον παραπάνω ορισμό, ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων δημιουργεί μία συλλογή δεδομένων οργανωμένη σε συσχετισμένους πίνακες που παρέχει ταυτόχρονα ένα μηχανισμό για ανάγνωση, εγγραφή, τροποποίηση και πιο πολύπλοκες διαδικασίες αναζήτησης πάνω στα δεδομένα.
3. Χωρικά Δεδομένα: όπως δηλώνει και ο όρος, πρόκειται για δεδομένα τα οποία παρέχουν γεωγραφικές πληροφορίες, όπως για παράδειγμα θέση ενός αντικειμένου στην επιφάνεια της γης, απόσταση σημείων, γειτνίαση ενός αντικειμένου με άλλα, κ.ο.κ.).
4. Query / Queries (επερωτήσεις): με τον όρο αυτό περιγράφονται τα ερωτήματα τα οποία υποβάλλονται σε ένα σύστημα διαχείρισης βάσεων δεδομένων, ώστε να εξαχθούν δεδομένα τα οποία πληρούν τα κριτήρια αναζήτησης ή προτάσεις οι οποίες τροποποιούν τα δεδομένα της αντίστοιχης βάσης δεδομένων.

2. Χωρικά δεδομένα

Τα χωρικά δεδομένα θα μπορούσαν να αναφέρονται σε κάθε κομμάτι πληροφορίας που συσχετίζει ένα αντικείμενο με μια τοποθεσία – από αστέρια στον ουρανό έως τη θέση ενός ανθρώπου σε σχέση με έναν άλλο. Τυπικά, ωστόσο, όταν χρησιμοποιούνται σε ένα γεωγραφικό πλαίσιο, τα χωρικά δεδομένα περιορίζονται στην περιγραφή φαινομένων πάνω ή κοντά στην επιφάνεια της γης. Σε ψηφιακή μορφή, τα δεδομένα είναι η πρωταρχική πληροφορία που χρειάζονται τα συστήματα γεωγραφικών πληροφοριών (GIS), τα οποία αποτελούν τα εργαλεία λογισμικού που χρησιμοποιούνται για ανάλυση χωρικών δεδομένων (Guptill & Morrison, 1999).

Τα χωρικά δεδομένα χρησιμοποιούνται ως ένας περιεκτικός όρος που περιλαμβάνει αντικείμενα από χάρτες γενικού σκοπού και χάρτες των οδών μιας πόλης, έως περισσότερο εξειδικευμένα σύνολα δεδομένων όπως σεισμικά μοντέλα, διασπορά των θραυσμάτων σε αρχαιολογικούς χώρους, ή στατιστικά στοιχεία μετακίνησης πληθυσμών. Πληροφορίες για κτήρια, γέφυρες, δρόμους, ποτάμια, λιβάδια και χώρες αποτελούν χαρακτηριστικά παραδείγματα χωρικών δεδομένων. Γενικά, τα χωρικά δεδομένα αναφέρονται σε κάθε κομμάτι πληροφορίας που περιγράφει μια απόλυτη ή σχετική τοποθεσία. Συνηθέστερα, τα χωρικά δεδομένα συσχετίζονται με πληροφορίες που υπάρχουν σε χάρτες και μπορεί να αναφέρονται σε κλίμακες από 1 μέτρο έως 10 χιλιόμετρα. Τέτοια δεδομένα συχνά αναφέρονται ως γεω-χωρικά δεδομένα και συναντώνται σε εφαρμογές όπως τα Γεωγραφικά Συστήματα Πληροφοριών, Συστήματα καταγραφής υποδομών, Εξ αποστάσεως αίσθηση, Συστήματα γεω-εντοπισμού. Τα χωρικά δεδομένα είναι συλλογές από τοπικές και μη τοπικές πληροφορίες για συγκεκριμένα γεωγραφικά φαινόμενα. Αυτή η συλλογή πληροφοριών διαμορφώνει τον πυρήνα μιας χωρικής βάσης δεδομένων. Μόλις η γεω-χωρική πληροφορία εισαχθεί σε μία βάση δεδομένων, οι σύγχρονοι υπολογιστές παρέχουν τις δυνατότητες διαχείρισης των γεωγραφικών δεδομένων με τρόπους που παλιότερα ήταν αδύνατοι (Laurini & Thompson, 1999).

2.1. Χωρικά Δεδομένα

Γενικά τα χωρικά δεδομένα χρησιμοποιούνται όταν είναι απαραίτητο να γίνει ανάλυση πάνω σε γεωγραφικά δεδομένα, όπως για παράδειγμα δίκτυα καλωδίων ηλεκτρικής ενέργειας ή ακόμα και για δημογραφικά δεδομένα για να τον προσδιορισμό των προτιμήσεων ομάδων πληθυσμού ή τον προσδιορισμό γεωγραφικών σημείων όπου συγκεντρώνονται άτομα διάφορων ηλικιών. Τέτοιου είδους μετρήσεις μπορούν να οδηγήσουν σε συμπεράσματα του τύπου:

Ποιες περιοχές είναι πιο προσιτές σε συγκεκριμένες ηλικιακές ομάδες και για ποιον λόγο;

Βάσει των αποτελεσμάτων μιας τέτοιας ανάλυσης μία επιχείρηση μπορεί να αποφασίσει εάν θα κατασκευάσει εκεί κάποιο κτίριο και ποια θα είναι η χρήση του (εμπορικό κέντρο, αθλητικό κέντρο, χώρος αναψυχής, κ.ο.κ.).

Γενικά, τα χωρικά δεδομένα χρησιμοποιούνται όταν ο παράγοντας ενδιαφέροντος είναι η απόσταση ή ακόμα και γεωγραφικά χαρακτηριστικά, όπως υψόμετρο, κλιματολογικές συνθήκες, κ.ά.

2.2. Μοντέλα γεω-χωρικών δεδομένων

Ένα σύνολο ψηφιακών χωρικών δεδομένων αναπαριστά ένα συγκεκριμένο μοντέλο της γεωγραφικής πραγματικότητας. Αυτά τα μοντέλα περιγράφουν και τις τοπικές και τις μη τοπικές όψεις των φαινομένων που περιγράφουν. Για να κατανοήσουν οι χρήστες τα δεδομένα που προκύπτουν από ένα τέτοιο μοντέλο, πρέπει να κατανοήσουν το εννοιολογικό μοντέλο που περιγράφει τα υποκείμενα δεδομένα.

Η οπτικοποίηση ενός συνόλου χωρικών δεδομένων συνήθως γίνεται με την εφαρμογή ενός από τους εξής τρόπους αναπαράστασης της γεω-χωρικής πληροφορίας: είτε με το μοντέλο των μοντέλων πλέγματος κυψελών (cell matrix ή raster), είτε των μοντέλων διανυσμάτων. Τα μοντέλα πλέγματος κυψελών χρησιμοποιούν συλλογές καθορισμένων γεωμετρικών δομών, για παράδειγμα, κυψέλες γης μήκους πλευράς 50 μέτρων, για να περιγράψουν την τοποθεσία και την έκταση ενός γεωγραφικού φαινομένου. Ψηφιακές φωτογραφίες από αεροπλάνα ή δορυφόρους συλλέγονται σε αυτή τη μορφή. Τα μοντέλα διανύσματος αντίθετα χρησιμοποιούν γραφικά στοιχεία (σημεία, γραμμές και πολύγωνα) για να περιγράψουν την τοποθεσία και την έκταση των υπό μελέτη φαινομένων. Τα περισσότερα ψηφιακά δεδομένα των χαρτών σήμερα βρίσκονται σε αυτή τη μορφή.

Τα χωρικά δεδομένα πρέπει επίσης να συμπεριλαμβάνουν όλες τις μη γεω-χωρικού χαρακτήρα πληροφορίες που χρησιμοποιούνται για την περιγραφή των αντικειμένων σε μία βάση δεδομένων. Στην περίπτωση δεδομένων ψηφιακής εικόνας, η ποσότητα των μη γεω-χωρικών δεδομένων είναι σχετικά μικρή, συνήθως ένα σύνολο τιμών που αναπαριστά χαρακτηριστικά όπως την έκταση της βλάστησης σε μία περιοχή. Στην περίπτωση του διανυσματικού μοντέλου, η ποσότητα των μη γεω-χωρικών δεδομένων μπορεί να είναι πολύ μεγάλη, ακόμα και να ξεπερνάει την ποσότητα των χωρικών δεδομένων. Αυτού του είδους τα χαρακτηριστικά περιλαμβάνουν πληροφορίες για τα διάφορα θέματα που υπάρχουν στη βάση δεδομένων (για παράδειγμα, ο αριθμός των λωρίδων σε μια λεωφόρο) και τη μεταξύ τους σχέση (για παράδειγμα, ότι οι γραμμές του σιδηροδρόμου βρίσκονται σε νησίδα μεταξύ των δύο ρευμάτων της λεωφόρου). Στο πλαίσιο των συστημάτων γεω-χωρικών δεδομένων, τα χαρακτηριστικά αποτυπώνουν το σύνολο των ερμηνειών μας για γεωγραφικά αντικείμενα ή φαινόμενα. Κτήρια, γέφυρες, δρόμοι, ρεύματα, βλάστηση ή και χώρες είναι παραδείγματα χαρακτηριστικών.

2.3. Δομές δεδομένων χαρακτηριστικών

Ο πραγματικός κόσμος της γεωγραφικής ποικιλίας είναι σύνθετος και συχνά ασαφής, αλλά παρ’ όλα αυτά πρέπει να αναπαρίσταται ψηφιακά με έναν σαφή, ντετερμινιστικό τρόπο. Για παράδειγμα, στην περίπτωση που πρόκειται να αναπαρασταθούν δεδομένα της γεωγραφικής πραγματικότητας, η οποία χαρακτηρίζεται από πληροφορίες που υπάρχουν σε χάρτες, αντιλαμβανόμαστε τον κόσμο σαν ένα χώρο ο οποίος δημιουργείται από διαφόρων ειδών χαρακτηριστικά: σημεία, γραμμές και περιοχές. Αυτό το μοντέλο είναι γνωστό ως διανυσματικός χώρος. Τα χαρακτηριστικά έχουν ιδιότητες που τα διακρίνουν το ένα από το άλλο. Κάθε σημείο στο χώρο μπορεί να είναι άδειο ή να περιγράφεται από ένα ή περισσότερα χαρακτηριστικά. Τα χαρακτηριστικά αυτά μπορεί να υπάρχουν καταχωρημένα σε μια βάση δεδομένων ή να ομαδοποιούνται με βάση ένα θέμα ή μια μεταβλητή σε έναν αριθμό στρωμάτων ή κλάσεων (layers ή classes) σε ένα επίπεδο. Η ορολογία που χρησιμοποιείται για να περιγράψουμε αυτά τα χαρακτηριστικά δίνεται στον Πίνακα 1 (Guptill, 2001).

Πίνακας 1. Ορολογία για την περιγραφή γεωγραφικών χαρακτηριστικών

Είδος χαρακτηριστικού	Παράδειγμα	Μοντέλο αποθήκευσης στη βάση δεδομένων
Σημείο	Θέση ενός φωτεινού σηματοδότη	Κόμβος (σημειακές συντεταγμένες)
Γραμμή	Η νησίδα ενός δρόμου	Ευθύγραμμα τμήματα που δηλώνονται με τις συντεταγμένες των δύο άκρων
Επιφάνεια	Μία λίμνη	Πολύγωνο

Οι σχέσεις μεταξύ κόμβων, ακμών και πολυγώνων ονομάζονται τοπολογικές σχέσεις. Με απλά λόγια, οι τοπολογικές σχέσεις περιγράφουν το γεγονός ότι οι ακμές αρχίζουν και τελειώνουν μόνο σε κόμβους, και ότι τα πολύγωνα δημιουργούνται από μια σειρά ακμών. Η τοπολογική περιγραφή κάθε ακμής περιλαμβάνει όχι μόνο τους αρχικούς και τελικούς κόμβους αλλά και τον προσδιορισμό ποιος είναι ο κόμβος αφετηρίας και ποιος ο κόμβος τερματισμού. Με αυτόν τον τρόπο η ακμή είναι κατευθυνόμενη: η πρώτη συντεταγμένη είναι αυτή του αρχικού κόμβου και η τελευταία του τελικού κόμβου. Αυτή η δομή δεδομένων όχι μόνο διευκολύνει την αναπαράσταση των χαρτογραφημένων χαρακτηριστικών, αλλά παρέχει και πληροφορίες που είναι απαραίτητες για να δημιουργηθούν αυτοματοποιημένες μέθοδοι επεξεργασίας και χειρισμού της βάσης δεδομένων (Guptill, 2001).

2.4. Περιγραφή γεωγραφικών χαρακτηριστικών

Τα μοντέλα χωρικών δεδομένων, εκτός από το να παρέχουν δεδομένα γεωγραφικής

φύσης, πρέπει να διαχειρίζονται όλες τις μη γεωγραφικές πληροφορίες που χρησιμοποιούνται για την περιγραφή των χαρακτηριστικών στη βάση δεδομένων. Για να κατανοήσουν οι προγραμματιστές τη φύση των δεδομένων, πρέπει να υπάρχει ένας σαφής προσδιορισμός των χαρακτηριστικών που περιγράφονται στη συγκεκριμένη εφαρμογή. Ο σαφής προσδιορισμός των δεδομένων υπονοεί τον ορισμό των χαρακτηριστικών καθώς και τον ορισμό των ιδιοτήτων και των δυνατών τιμών των ιδιοτήτων τους. Άλλες πληροφορίες που απαιτούνται για την πλήρη κατανόηση είναι οι κανόνες εξαγωγής δεδομένων από τις πρωτογενείς πηγές και οι κανόνες αναπαράστασης της πληροφορίας σε ένα πληροφοριακό σύστημα. Η κατανόηση του τρόπου με τον οποίο αναπαρίσταται ένα γεωγραφικό χαρακτηριστικό, δεν έχει μεγάλη χρησιμότητα αν ο προγραμματιστής δεν αντιληφθεί τη φύση της πληροφορίας που περιγράφεται. Τι πραγματικά εννοούμε με όρους όπως “δέντρο”, “έλος”, “κατοικημένη γη” ή “γέφυρα”; Για παράδειγμα, κάποιος θα μπορούσε να θεωρήσει τη “γέφυρα” ως ένα χαρακτηριστικό. Μια συγκεκριμένη γέφυρα είναι ένα αντικείμενο από ένα σύνολο παρόμοιων δομών πάνω ή κοντά στην επιφάνεια της γης. Ο ορισμός του όρου «γέφυρα» επεκτείνεται περαιτέρω σαν μια δομή που δημιουργήθηκε για να μεταφέρει την κυκλοφορία πάνω από ένα φυσικό εμπόδιο ή ένα ποτάμι. Έτσι, το χαρακτηριστικό «γέφυρα» είναι ένα στιγμιότυπο ενός συνόλου φαινομένων, με συγκεκριμένες λειτουργικές ιδιότητες («να μεταφέρει την κυκλοφορία») και πληροφορίες της τοποθεσίας που βρίσκεται. Ο ορισμός της έννοιας επίσης προσδιορίζει τη σχέση που συνδέει ένα αντικείμενο «γέφυρα» με άλλα αντικείμενα («πάνω από ένα φυσικό εμπόδιο ή ένα ποτάμι»). Οι ορισμοί και οι προδιαγραφές που τίθενται στο σχεδιασμό τέτοιου είδους δεδομένων πρέπει ιδανικά να ανταποκρίνονται στις απαιτήσεις της εκάστοτε εφαρμογής και των χρηστών της.

Για την επίλυση προβλημάτων σχεδιασμού γεω-χωρικών δεδομένων, μπορούμε να δημιουργούμε έναν ορισμό για κάθε χαρακτηριστικό, ο οποίος περιγράφει τα δεδομένα και πώς αυτό μοντελοποιείται. Αυτή η τεχνική χρησιμοποιείται από εθνικούς φορείς στις ΗΠΑ, τη Γαλλία και τη Γερμανία (Konecny, et al., 2016). Οι προσδιορισμοί των χαρακτηριστικών ορίζουν το σύνολο των χαρακτηριστικών το οποίο τελικά θα αναπαρασταθεί σε μια χωρική βάση δεδομένων. Για κάθε τέτοιο χαρακτηριστικό, ορίζεται ένα σύνολο τιμών από τις οποίες θα παίρνει τιμή αυτό το χαρακτηριστικό.

2.5. Σημασιολογικές σχέσεις γεω-χωρικών αντικειμένων

Συγκεκριμένες σχέσεις, που ονομάζονται σημασιολογικές σχέσεις χρησιμοποιούνται για να εκφράσουν αλληλεπιδράσεις που υπάρχουν μεταξύ χαρακτηριστικών. Τέτοιες σχέσεις ορίζουν περιπτώσεις όπου χαρακτηριστικά κάποιων αντικειμένων τέμνονται ή περιπτώσεις όπου τέτοιου είδους χαρακτηριστικά ορίζουν τα όρια μιας περιοχής. Ο σκοπός της μοντελοποίησης όλων αυτών των σχέσεων είναι να αναπαραστήσουν επαρκώς τις αλληλεπιδράσεις μεταξύ αντικειμένων για να υποστηρίξουν τις εφαρμογές γεωγραφικών συστημάτων (Geographical Information Systems – GIS) και τις

απαιτήσεις αυτοματοποιημένων χαρτογραφικών εφαρμογών νέας γενιάς.

3. Top-k Ερωτήσεις Κατάταξης

Τα τελευταία χρόνια, έχει υπάρξει μεγάλο ενδιαφέρον για την ανάπτυξη αποτελεσματικών τεχνικών αναζήτησης και ανάκτησης σε σχεσιακές βάσεις δεδομένων, σε βάσεις δεδομένων εγγράφων και πολυμέσων, σε συστήματα επιστημονικών πληροφοριών κ.ο.κ.

Ένα δημοφιλές παράδειγμα για την αντιμετώπιση αυτού του προβλήματος είναι τα ερωτήματα Top-k, δηλαδή η κατάταξη και η επιστροφή των αποτελεσμάτων με τις k υψηλότερες βαθμολογίες σε σχέση με τα δεδομένα κριτήρια αναζήτησης. Η διάδοση των ερωτημάτων Top-k είναι ραγδαία, με αποτέλεσμα να έχουν αναπτυχθεί πολλές παραλλαγές του προβλήματος βελτίωσης του αλγόριθμου Top-k, ώστε η ταξινόμηση των αποτελεσμάτων να είναι πιο αποδοτική και κοντά στις προσδοκίες του χρήστη (Safe Software, 2020).

3.1. Εισαγωγή

Τα Top-k queries επιστρέφουν μόνο τα δεδομένα που καλύπτουν σε μεγαλύτερο βαθμό τις προτιμήσεις των χρηστών του συστήματος το οποίο χρησιμοποιείται για την έρευνα. Με τη χρήση αυτού του είδους επερωτήσεων επιστρέφονται μόνο τα δεδομένα που είναι αναγκαία στο χρήστη και όχι ογκώδη και δύσκολα στην επεξεργασία τους σύνολα αποτελεσμάτων. Συνεπώς είναι σημαντικό για τον δημιουργό του συστήματος να του προσδώσει την ικανότητα να επιστρέφει τα καλύτερα αποτελέσματα με βάση την καλύτερη αξιολόγηση, μέσα από διαφορετικές παραμέτρους και συναρτήσεις, τα οποία πληρούν τις περισσότερες προτιμήσεις των χρηστών (Wikipedia, 2020).

Πιο συγκεκριμένα, μπορούμε να θεωρήσουμε ένα μεγάλο σύνολο 600.000 εγγράφων, από το οποίο θέλουμε να εντοπίσουμε μόνο k στο πλήθος από αυτά (όπου k ένας μικρός αριθμός), τα οποία θα είναι τα πιο συναφή με την επιθυμία του χρήστη(ή αλλιώς κορυφαία αποτελέσματα). Ένα τέτοιο αποτέλεσμα δίνει τη δυνατότητα στο χρήστη να επεξεργαστεί τα δεδομένα πιο εύκολα, δεδομένου ότι αυτά θα είναι πιο εστιασμένα στις επιθυμίες του. Επίσης επιτυγχάνεται μεγαλύτερη απόδοση του συστήματος ενώ αυτό απαιτεί λιγότερους υπολογιστικούς πόρους.

3.2. Αλγόριθμος Top-k

Τα τελευταία χρόνια, έχει υπάρξει μεγάλο ενδιαφέρον για την ανάπτυξη αποτελεσματικών τεχνικών αναζήτησης και ανάκτησης σε σχεσιακές βάσεις δεδομένων, σε βάσεις δεδομένων εγγράφων και πολυμέσων, σε συστήματα επιστημονικών πληροφοριών κ.ο.κ. Ένα δημοφιλές παράδειγμα για την αντιμετώπιση αυτού του προβλήματος είναι τα ερωτήματα Top-k, δηλαδή η κατάταξη και η επιστροφή των αποτελεσμάτων με τις k υψηλότερες βαθμολογίες σε σχέση με τα

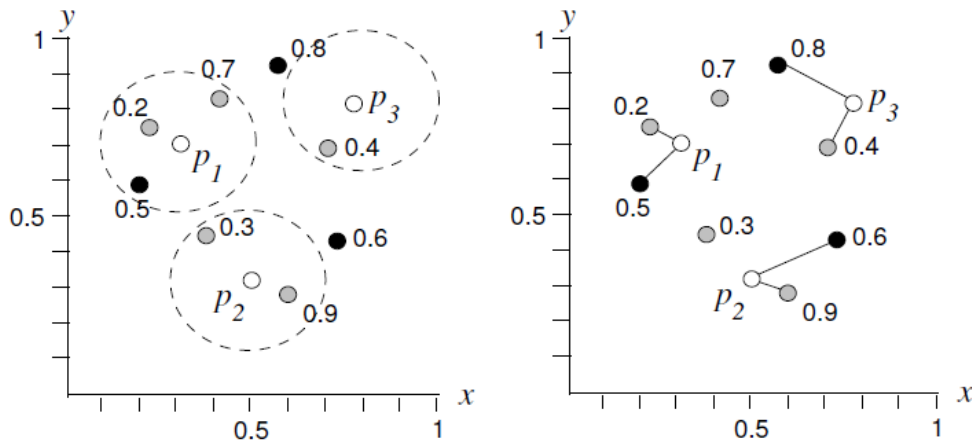
δεδομένα κριτήρια αναζήτησης. Η διάδοση των ερωτημάτων Top-k είναι ραγδαία, με αποτέλεσμα να έχουν αναπτυχθεί πολλές παραλλαγές του προβλήματος βελτίωσης του αλγόριθμου Top-k, ώστε η ταξινόμηση των αποτελεσμάτων να είναι πιο αποδοτική και κοντά στις προσδοκίες του χρήστη (Safe Software, 2020).

3.3. Top-k χωρικές επερωτήσεις προτιμήσεων

Τα χωρικά συστήματα βάσεων δεδομένων διαχειρίζονται μεγάλες συλλογές από γεωγραφικές οντότητες, που εκτός από τις γεω-χωρικές ιδιότητες περιέχουν και μη χωρικές πληροφορίες (ενδεικτικά: όνομα, μέγεθος, τύπο, τιμή κ.λπ.). Η αναζήτηση δεδομένων σε τέτοιου τύπου βάσεις δεδομένων αποκτά μεγάλο ενδιαφέρον, δεδομένου ότι σε αυτές τις ερωτήσεις υπεισέρχονται παράμετροι προτιμήσεων των χρηστών, οι οποίοι τελικά επιλέγουν την καλύτερη (για τους ίδιους) χωρική τοποθεσία με κριτήριο τις ειδικές ιδιότητες της συγκεκριμένης, σε σύγκριση με τις αντίστοιχες ιδιότητες άλλων ομοειδών της (Gunopulos, 2020).

Δεδομένου ενός συνόλου D με αντικείμενα ενδιαφέροντος (όπως για παράδειγμα υποψήφιος τοποθεσίες για διαμονή), μία Top-k χωρική ερώτηση προτίμησης ανακτά από το σύνολο D τα k -αντικείμενα με την υψηλότερη βαθμολογία. Η βαθμολογία ενός αντικειμένου ορίζεται από την ποιότητα των χαρακτηριστικών του (όπως η ποιότητα των εγκαταστάσεων και των παρεχόμενων υπηρεσιών) στη γεω-χωρική του γειτονιά.

Ως παράδειγμα, ένα γραφείο πρακτόρευσης ακινήτων, το οποίο διατηρεί μια βάση δεδομένων με διαμερίσματα διαθέσιμα για ενοικίαση. Ένα πελάτης μπορεί να θέλει να κατατάξει το περιεχόμενο αυτής της βάσης σύμφωνα με την ποιότητα της τοποθεσίας τους, μέγεθος το οποίο υπολογίζεται από ένα σύνολο μη χωρικών χαρακτηριστικών άλλων ιδιοτήτων (π.χ. εάν υπάρχουν κοντά στο ακίνητο εστιατόρια, καφέ, νοσοκομείο, αγορά κ.λπ.). Η χωρική γειτονιά του διαμερίσματος ορίζεται σε ένα συγκεκριμένο εύρος γύρω του. Η ποιότητα είναι ένα μέγεθος υποκειμενικό και καθορίζεται από διάφορες παραμέτρους της ερώτησης. Για παράδειγμα, ένας χρήστης μπορεί να προσδιορίζει την ποιότητα με βάση τα μη χωρικά προτερήματα των γύρω εστιατορίων (όπως εάν σερβίρουν θαλασσινά, το εύρος των τιμών του τιμοκαταλόγου τους, κ.ο.κ.) (Vlachou, et al., 2020), (Σφυρής, 2016).



Εικόνα 1. Παραδείγματα από Top-k χωρικές ερωτήσεις προτίμησης

Τα λευκά σημεία στην Εικόνα 1 δημιουργούν ένα ενδεικτικό σύνολο δεδομένων μιας κλάσης (π.χ. ξενοδοχεία). Στην εικόνα εμφανίζονται επίσης δύο σύνολα ιδιοτήτων. Τα γκρι σημεία αντιστοιχούν σε εστιατόρια και τα μαύρα σημεία σε καφέ. Τα σημεία που αντιστοιχούν σε ιδιότητες παριστάνονται με μη χωρικές τιμές (π.χ. προτεινόμενα από κάποια υπηρεσία αξιολογήσεων, όπως το TripAdvisor.com). Ένας τουρίστας μπορεί να ενδιαφέρεται για το ξενοδοχείο p που μεγιστοποιεί μια βαθμολογία, η οποία συμβολίζεται $\tau(p)$. Η βαθμολογία αυτή ορίζεται ως το άθροισμα των μέγιστων τιμών ποιότητας των εστιατορίων της γειτονιάς και των μέγιστων τιμών ποιότητας των καφέ της γειτονιάς επίσης. Η γειτονιά αναπαρίσταται με το διάστικτο κύκλο με κέντρο τα αντικείμενα της κλάσης p (ξενοδοχεία) και ακτίνα 0.2 χιλιομέτρων. Σύμφωνα με αυτό τον υπολογισμό, η μέγιστη ποιότητα γκρι και μαύρων σημείων μέσα στον κύκλο του p_1 είναι 0,7 και 0,5 αντίστοιχα, άρα η βαθμολογία του p_1 είναι $\tau(p_1)=0,7+0,5=1,2$. Εφόσον δεν υπάρχουν μαύρα σημεία εντός της ακτίνας του p_2 , ο υπολογισμός της βαθμολογίας δίνει $\tau(p_2)=0,9+0=0,9$. Όμοια προκύπτει ότι $\tau(p_3)=0,4+0=0,4$. Βάσει των παραπάνω, το ξενοδοχείο p_1 επιστρέφεται ως κορυφαία πρόταση, βάσει των προτιμήσεων του χρήστη, δεδομένου ότι συγκέντρωσε την υψηλότερη βαθμολογία.

Σαν εναλλακτικό παράδειγμα, η εικόνα 1β παρουσιάζει την περίπτωση όπου η βαθμολογία $\tau(p)$ ενός ξενοδοχείου υπολογίζεται ως το άθροισμα του μέτρου της ποιότητας των πλησιέστερων εστιατορίων και καφέ. Τα κοντινότερα αυτά σημεία συνδέονται με το αντίστοιχο ξενοδοχείο με αντίστοιχες ακμές. Ως εκ τούτου, ο υπολογισμός της βαθμολογίας των τριών ξενοδοχείων p_1 , p_2 , p_3 γίνεται με τη χρήση των τύπων: $\tau(p_1)=0,2+0,5=0,7$, $\tau(p_2)=0,9+0,6=1,5$ και $\tau(p_3)=0,8+0,4=1,2$.

Βάσει του παραπάνω υπολογισμού, το ξενοδοχείο που προτείνεται στο χρήστη σύμφωνα με τις προτιμήσεις του είναι το p_2 .

Παραδοσιακά, έχουν προταθεί δύο τρόποι κατάταξης αντικειμένων: (i) χωρική

κατάταξη, βάσει της οποίας η κατάταξη των αντικειμένων γίνεται με κριτήριο την απόστασή τους από ένα σημείο αναφοράς, και (ii) μη χωρική κατάταξη, η οποία κατατάσσει τα αντικείμενα με τη χρήση μιας συναθροιστικής συνάρτησης σε ένα σύνολο μη χωρικών ιδιοτήτων τους. Οι Top-k ερωτήσεις χωρικών προτιμήσεων ενοποιούν αυτούς τους δύο τύπους κατάταξης αντικειμένων, εφαρμόζοντας συνδυαστικές τεχνικές στους δύο τύπους δεδομένων. Όπως φαίνεται από τα παραδείγματά που αναφέρθηκαν, αυτός ο τύπος επερωτήσεων έχει ένα μεγάλο εύρος εφαρμογών σε συστήματα συστάσεων (recommendation systems) και σε συστήματα υποστήριξης αποφάσεων.

Παρά τη χρησιμότητα των Top-k χωρικών ερωτήσεων προτίμησης, αυτές δεν έχουν μελετηθεί επαρκώς στο παρελθόν. Μια απλή προσέγγιση για τον υπολογισμό χωρικών δεδομένων σε συνδυασμό με ιδιότητες των αντικειμένων που αναζητούνται, είναι να γίνει ο υπολογισμός της βαθμολογίας για όλα τα αντικείμενα του συνόλου D και από το αποτέλεσμα να γίνει η επιλογή k -στοιχείων με τις υψηλότερες βαθμολογίες. Αυτή η μέθοδος, είναι υπολογιστικά ακριβή, υπό την έννοια ότι καταναλώνει μεγάλη υπολογιστική ισχύ για τον υπολογισμό βαθμολογιών σε μεγάλα σύνολα αντικειμένων. Για το λόγο αυτό έχουν προταθεί εναλλακτικοί αλγόριθμοι για τη διεξαγωγή τέτοιων υπολογισμών, οι οποίοι ελαχιστοποιούν τις διαδικασίες εισόδου / εξόδου στον επεξεργαστή του υπολογιστικού συστήματος, ενώ είναι παράλληλα και υπολογιστικά αποτελεσματικές. Τέτοιου είδους τεχνικές δημιουργούν ομάδες αντικειμένων στο χώρο (clusters) και στη συνέχεια εφαρμόζουν σε κάθε ομάδα μεθόδους υπολογισμού της μέγιστης βαθμολογίας για τα αντικείμενα της εκάστοτε ομάδας. Οι τεχνικές αυτές μειώνουν δραστικά το πλήθος των αντικειμένων στα οποία θα γίνει η αναζήτηση και συνεπώς μειώνεται και ο χρόνος απόκρισης του συστήματος.

3.3.1. Βιβλιογραφική επισκόπηση

Η κατάταξη αντικειμένων αποτελεί μια απαραίτητη διαδικασία σε διάφορες εφαρμογές ανάκτησης δεδομένων. Σε σχεσιακές βάσεις δεδομένων, οι πλειάδες που επιστρέφονται από μία επερώτηση πρέπει να ταξινομηθούν βάσει μιας συναθροιστικής συνάρτησης στις τιμές των ιδιοτήτων τους (Bruno, et al., 2002). Στο παράδειγμα του μεσιτικού γραφείου, έστω μια βάση δεδομένων ακινήτων, η οποία περιέχει πληροφορίες για διαμερίσματα διαθέσιμα προς ενοικίαση. Ένας ενδιαφερόμενος ενδέχεται να θέλει να προβάλλει τα 10 διαμερίσματα με τα μεγαλύτερα μεγέθη και τις χαμηλότερες τιμές. Η βαθμολογία του κάθε διαμερίσματος σε αυτήν την περίπτωση εκφράζεται από το άθροισμα δύο ξεχωριστών βαθμών: το μέγεθος και την τιμή, αφού οι τιμές τους κανονικοποιηθούν ώστε να ανήκουν σε κοινό εύρος (έστω μεταξύ 0 και 1, όπου το 1 σημαίνει την υψηλότερη προτίμηση, η οποία αντιστοιχεί στο ακίνητο με το μέγιστο δυνατό μέγεθος και την ελάχιστη δυνατή τιμή).

Μία άλλη δημοφιλής εφαρμογή κατάταξης αντικειμένων είναι η κατάταξη εγγράφων με βάση τη σχετικότητα των λέξεων-κλειδιών (όρων) που περιέχονται σε μια ερώτηση

του χρήστη. Οι ερωτήσεις αυτές εκφράζονται ως σύνολα όρων, συνήθως σε φυσική γλώσσα. Το πρόβλημα αυτό έχει αποτελέσει ερευνητικό ζήτημα στο πεδίο της ανάκτησης πληροφοριών (Information Retrieval) για πάνω από δύο δεκαετίες (Baeza-Yates & Ribeiro-Neto, 1999). Η συνάρτηση κατάταξης των εγγράφων υπολογίζει μια συνάθροιση της σχετικότητας των όρων των ερωτήσεων του χρήστη με το έγγραφο. Η συνάθροιση αυτή συχνά εμπλουτίζεται με κάποια γενικότερη βαθμολόγηση των εγγράφων με βάση τη δημοφιλία τους.

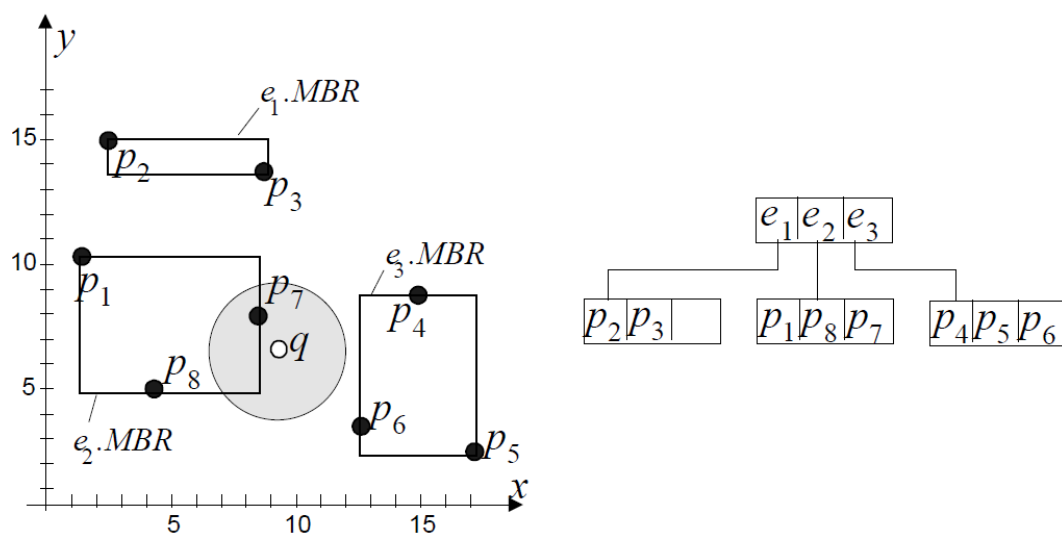
Στις χωρικές βάσεις δεδομένων, η κατάταξη ανάγεται συχνά στην ανάκτηση των κοντινότερων γειτόνων ενός κόμβου (Nearest neighbor). Δεδομένης μιας επερώτησης με βάση την τοποθεσία, συχνά το ενδιαφέρον εστιάζεται στην ανάκτηση του συνόλου των κοντινότερων αντικειμένων, τα οποία πληρούν μια συνθήκη (για παράδειγμα να ανήκουν στην κλάση «εστιατόρια»). Υποθέτοντας ότι το σύνολο αντικειμένων ενδιαφέροντος δεικτοδοτείται με μια ιεραρχική μέθοδο χωρικής πρόσβασης (όπως μία δομή R-δένδρου), είναι δυνατό να εφαρμοστούν κάποια όρια στην απόσταση των κόμβων ενώ γίνεται η διάσχιση των κόμβων, ώστε η αναζήτηση να περιορίζεται σε συγκεκριμένα γεωγραφικά όρια (Hjaltason & Samet, 1999). Οι Tao et al. (Tao, et al., 2014) εισηγήθηκαν τη μοντελοποίηση Top-k ερωτήσεων ως (σταθμισμένες) ερωτήσεις εύρεσης του κοντινότερου γείτονα, στον πολυδιάστατο χώρο που ορίζεται από τις ιδιότητες των αντικειμένων, όπου κάθε απάντηση στην επερώτηση προκύπτει λαμβάνοντας τη μέγιστη τιμή κάθε διάστασης.

Η χρήση πολυδιάστατων δεικτών (multi-dimensional indexes) δεν είναι πάντοτε εφικτή για την ανάκτηση Top-k αποτελεσμάτων. Ένας λόγος είναι ότι τέτοιου είδους δείκτες είναι ασταθείς σε πολυδιάστατους χώρους (Weber, et al., 1998). Επίσης, οι επερωτήσεις Top-k ενδέχεται να χρησιμοποιούν ένα αυθαίρετο σύνολο ιδιοτήτων (π.χ. μέγεθος, τιμή, απόσταση από την παραλία, αριθμός υπνοδωματίων, όροφος κ.λπ.) και οι απαραίτητοι δείκτες ενδέχεται να μην είναι διαθέσιμοι για όλους τους πιθανούς συνδυασμούς ιδιοτήτων. Σε τέτοιες περιπτώσεις η δημιουργία και η διατήρηση των δεικτών είναι υπολογιστικά και χρονικά πολύ ακριβή διαδικασία, ικανή να επιβραδύνει χαρακτηριστικά την αναζήτηση. Μία τάση στην έρευνα των Top-k ερωτήσεων έχει εστιάσει στην αποτελεσματική συγχώνευση κατατάξεων αντικειμένων τα οποία συγκεντρώνονται από διαφορετικές (κατανεμημένες) πηγές δεδομένων. Το κίνητρο αυτών των μεθόδων είναι η ελαχιστοποίηση του αριθμού των προσβάσεων στις πληροφορίες κατάταξης των αντικειμένων που αξιολογούνται στην ερώτηση, έως ότου ανακαλυφθούν τα αντικείμενα με τις Top-k συνολικά καλύτερες βαθμονομήσεις. Για να γίνει αυτό εφικτό, οι αλγόριθμοι διατηρούν τα ανώτερα και χαμηλότερα όρια των βαθμονομήσεων των αντικειμένων που έχουν βρεθεί μέχρι το τρέχον σημείο της επεξεργασίας της επερώτησης ενώ διασχίζουμε την ταξινομημένη λίστα, ώστε να αξιολογούνται αντικείμενα μόνο μέσα σε αυτό το εύρος.

3.3.2. Αξιολόγηση χωρικών ερωτήσεων σε R-δέντρα

Η πιο δημοφιλής μέθοδος χωρικής πρόσβασης είναι το R-δένδρο, το οποίο χρησιμοποιείται για την ευρετηρίαση των ελάχιστων ορθογωνίων που οριοθετούν το εκάστοτε αντικείμενο (Minimum Bounding Rectangles - MBRs). Η Εικόνα 2 απεικονίζει μια συλλογή $R=p_1, \dots, p_8$ χωρικών αντικειμένων (πχ. σημεία) και μία δομή R-δέντρου η οποία τα δεικτοδοτεί.

Τα R-δέντρα μπορούν να επεξεργαστούν αποτελεσματικά τους κυριότερους τύπους χωρικών ερωτήσεων, συμπεριλαμβανομένων ερωτήσεων χωρικού εύρους, ερωτήσεις κοντινότερου γείτονα, και χωρικές συζεύξεις (joins). Δεδομένης μιας χωρικής περιοχής W , μία ερώτηση χωρικής περιοχής θα επιστρέψει εκείνα τα αντικείμενα του συνόλου R τα οποία ανήκουν στην τομή με το σύνολο W . Ως παράδειγμα αναφέρουμε μία επερώτηση εύρους, η οποία αναζητά όλα τα αντικείμενα τα οποία βρίσκονται σε απόσταση 3 μονάδων από το αντικείμενο q . Τα αντικείμενα αυτά αντιστοιχούν στη σκιασμένη περιοχή της Εικόνας 2.



Εικόνα 2. Χωρικές επερωτήσεις με χρήση R-δένδρων

Ξεκινώντας από τη ρίζα του δέντρου, η ερώτηση εξελίσσεται ακολουθώντας αναδρομικά τους κλάδους του δέντρου, έχοντας MBRs που διατέμνουν την περιοχή της επερώτησης. Για παράδειγμα, το αντικείμενο e_1 δεν διατέμνει την περιοχή της επερώτησης, και έτσι το υποδένδρο που ξεκινά από το αντικείμενο e_1 δεν μπορεί να περιέχει κανένα αποτέλεσμα της επερώτησης. Αντίθετα, το αντικείμενο e_2 ακολουθείται από τον αλγόριθμο αναζήτησης και η βαθμολογία στον αντίστοιχο κόμβο εξετάζεται με αλγόριθμο αναδρομής, ώστε να προκύψει το αντικείμενο p_7 ως αποτέλεσμα της επερώτησης.

Μία επερώτηση κοντινότερου γείτονα (Nearest Neighbor - NN) δέχεται ως είσοδο ένα

αντικείμενο ερώτησης q και επιστρέφει το κοντινότερο αντικείμενο στο q , το οποίο ανήκει στο σύνολο R . Για παράδειγμα, ο κοντινότερος γείτονας του q στην Εικόνα 2 είναι το p_7 . Μια δημοφιλής γενίκευση είναι η επερώτηση k -NN, που επιστρέφει τα k κοντινότερα αντικείμενα σε q , όπου k θετικός ακέραιος. Η επεξεργασία των επερωτήσεων NN και k -NN γίνεται αποτελεσματικά χρησιμοποιώντας τον αλγόριθμο Best-First (BF) με δεδομένο ότι το σύνολο R δεικτοδοτείται με τη χρήση R -δένδρου (Gunopulos, 2020).

Ο αλγόριθμος ξεκινά αρχικοποιώντας μια ουρά προτεραιότητας PQ με τις ρίζες των αντικειμένων. Η οργάνωση του R -δένδρου γίνεται με βάση την ελάχιστη απόσταση των MBRs τους από το αντικείμενο q . Για να βρεθεί το NN του q στην Εικόνα 2, ο αλγόριθμος BF εισάγει στην ουρά προτεραιότητας PQ τα αντικείμενα e_1, e_2, e_3 και τις αποστάσεις τους από το q . Τότε η κοντινότερη είσοδος e_2 ανακτάται από την ουρά προτεραιότητας PQ και τα αντικείμενα p_1, p_7, p_8 εισάγονται στην ουρά προτεραιότητας PQ . Το αμέσως κοντινότερο στοιχείο της PQ είναι το p_7 , το οποίο είναι ο κοντινότερος γείτονας του q . Σε όρους εισόδου / εξόδου (I/O), ο αλγόριθμος BF αποδεικνύεται ότι δεν είναι χειρότερος από οποιονδήποτε NN αλγόριθμο που εφαρμόζεται στο ίδιο δέντρο R .

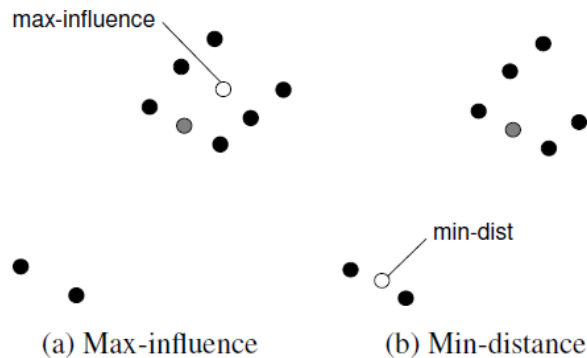
Μία λιγότερο αποτελεσματική προσέγγιση είναι ο depth-first αλγόριθμος, ο οποίος διασχίζει το δέντρο δίνοντας κατά βάθος. Όπως αναφέρεται φαίνεται στο (Hjaltason & Samet, 1999), ο αλγόριθμος DF ενδέχεται να καταναλώνει περισσότερο I/O από τον BF. Ωστόσο, ο DF απαιτεί συγκεκριμένη ποσότητα μνήμης και μόνο μία διαδρομή των κλάδων του δένδρου παραμένει στη μνήμη κατά την αναζήτηση.

Το γενικευμένο R -δέντρο (aggregate R -tree) είναι μια παραλλαγή του R -δένδρου, όπου κάθε κόμβος ο οποίος δεν είναι φύλλο του δένδρου λειτουργεί ως μετρητής για κάποιες τιμές των ιδιοτήτων όλων των κόμβων στο υποδέντρο του. Ως παράδειγμα, το δένδρο που απεικονίζεται στην Εικόνα 2 μπορεί να αναβαθμιστεί σε ένα MAX γενικευμένο R -δέντρο στο σύνολο των κόμβων, αν τα αντικείμενα e_1, e_2, e_3 περιέχουν τις μέγιστες τιμές του μέτρου των συνόλων $\{p_2, p_3\}$, $\{p_1, p_8, p_7\}$, $\{p_4, p_5, p_6\}$ αντίστοιχα. Ας υποθέσουμε ότι οι τιμές μέτρων των p_4, p_5, p_6 είναι 0.2, 0.1, 0.4 αντίστοιχα. Σε αυτήν την περίπτωση το συνολικό μέτρο του αντικειμένου e_3 θα ήταν ο μέγιστος του συνόλου των τιμών $\{0.2, 0.1, 0.4\} = 0.4$.

3.3.3. Χωρικές Ερωτήσεις βασισμένες σε χαρακτηριστικά

Μία διαδεδομένη Top- k επερώτηση είναι αυτή του εντοπισμού των k ιστοσελίδων με τη μεγαλύτερη επιρροή, οι οποίες μελετώνται στα (Du, et al., 2005), (Zhang, et al., 2006). Ο στόχος είναι η εύρεση της τοποθεσίας στον χώρο (ο οποίος δεν προσδιορίζεται σε ένα κλειστό σύνολο ιστοσελίδων), η οποία ελαχιστοποιεί το αποτέλεσμα της συνάρτησης υπολογισμού που χρησιμοποιείται. Στις εικόνες 3α και

3b, τα σημεία τα οποία χαρακτηρίζουν ιδιότητες απεικονίζονται με μαύρους κύκλους ενώ οι ιστοσελίδες απεικονίζονται ως γκρι σημεία.



Εικόνα 3. Optimal location queries

Ας υποθέσουμε ότι όλα τα σημεία που απεικονίζουν ιδιότητες έχουν την ίδια ποιότητα. Η ερώτηση μέγιστης επιρροής στη βέλτιστη απόσταση (Du, et al., 2005), βρίσκει μία τοποθεσία (την οποία στη συνέχεια θα εισάγει στο υπάρχον σύνολο ιστοσελίδων) με τη μεγαλύτερη επιρροή (όπως ορίζεται στο (Xia, et al., 2005)), ενώ η επερώτηση βέλτιστης τοποθεσίας στην ελάχιστη απόσταση (Zhang, et al., 2006) αναζητά την τοποθεσία που ελαχιστοποιεί τη μέση απόσταση κάθε ιδιότητας από την κοντινότερη ιστοσελίδα. Οι βέλτιστες θέσεις και για τις δύο ερωτήσεις σημειώνονται με άσπρα σημεία στις Εικόνες 3a και 3b αντίστοιχα.

Ο (Xia, et al., 2005) έλυσε το πρόβλημα εύρεσης των Top-k θέσεων (π.χ. εστιατόρια) με βάση την επιρροή τους σε διάφορα σύνολα αντικειμένων (όπως στην επιλογή ενοικιαζόμενου διαμερίσματος). Ως παράδειγμα, η Εικόνα 3 δείχνει ένα σύνολο θέσεων (λευκά σημεία) και ένα σύνολο χαρακτηριστικών (μαύρα σημεία) με αντίστοιχα βάρη, τέτοια που κάθε διακεκομμένη γραμμή συνδέει μία ιδιότητα με το κοντινότερο σημείο. Η επιρροή μιας θέσης p_i ορίζεται από το άθροισμα των βαρών των ιδιοτήτων που έχουν το p_i ως πλησιέστερο σημείο ενδιαφέροντος. Για παράδειγμα, η βαθμολογία του p_1 είναι $0,9 + 0,5 = 1,4$. Ομοίως, οι βαθμολογίες των p_2 και p_3 είναι 1,5 και 1,2 αντίστοιχα. Έτσι, το p_2 επιστρέφεται ως η Top-1 θέση με τη μεγαλύτερη επιρροή.

Οι τεχνικές που προτείνουν οι (Xia, et al., 2005), (Du, et al., 2005) και (Zhang, et al., 2006) είναι συγκεκριμένες και εφαρμόζονται σε συγκεκριμένους τύπους ερωτήσεων, όπως αυτούς που περιγράφηκαν παραπάνω και δεν μπορούν να εφαρμοστούν στην κλάση των Top-k ερωτήσεων χωρικών προτιμήσεων, σαν αυτές που διαχειρίζεται το προτεινόμενο σύστημα αυτής της εργασίας. Επίσης, σχετίζονται με ένα σύνολο δεδομένων μίας μόνο ιδιότητας, ενώ οι επερωτήσεις που διαχειρίζεται η προτεινόμενη εφαρμογή λαμβάνουν υπόψη πολλαπλά σύνολα δεδομένων ιδιοτήτων των υπό εξέταση αντικειμένων.

4. Γράφοι και PageRank

Ένας **γράφος** είναι μια αναπαράσταση ενός συνόλου στοιχείων που μπορεί να συνδέονται μεταξύ τους ανά ζεύγη. Η σύνδεση των στοιχείων γίνεται με γραμμές, οι οποίες μπορεί να δηλώνουν την κατεύθυνσή τους, ανάλογα με τον γράφο. Τα συνδεδεμένα στοιχεία συχνά αναπαρίστανται με μαθηματικούς ορισμούς και ονομάζονται «κορυφές». Οι γραμμές ή τόξα που συνδέουν τις κορυφές, ονομάζονται «ακμές».

Στο πεδίο του προγραμματισμού, ένας γράφος αναπαρίσταται σαν ένας χάρτης από κύκλους και τόξα που αναπαριστούν κόμβους και ακμές, οι οποίες δείχνουν την σχέση των κόμβων μεταξύ τους. Σε πολλές περιπτώσεις στις ακμές σημειώνονται κατευθύνσεις όπως και βάρη. Ενδέχεται σε έναν γράφο να εμφανίζονται κόμβοι που δεν συνδέονται με κανέναν άλλο κόμβο. Γενικά υπάρχουν πολλά είδη γράφων με πολλαπλές ιδιότητες. Στη συνέχεια θα περιγραφούν ορισμένοι τύποι γράφων οι οποίοι παρουσιάζουν ενδιαφέρον για την εργασία αυτή και θα αναλυθούν, ώστε να περιγραφεί καθαρά ο λόγος που επιλέχθηκαν για την απεικόνιση των αποτελεσμάτων αναζήτησης που παρουσιάζει το σύστημα στον χρήστη.

Με τη χρήση γράφων είναι δυνατή η απεικόνιση σύνθετων και περίπλοκων καταστάσεων, οι οποίες εξαρτώνται από μεγάλα πλήθη εννοιών και χρειάζονται έναν αριθμό από διαδικασίες για να μπορέσουμε να τα αναλύσουμε σωστά.

Ένα χαρακτηριστικό παράδειγμα για την καλύτερη επεξήγηση της χρησιμότητας των γράφων είναι το ακόλουθο. Έστω ότι βρισκόμαστε σε μια έκθεση ζωγραφικής ενός γνωστού ζωγράφου. Στην έκθεση βρίσκεται ακόμη η οικογένεια του ζωγράφου αλλά και επισκέπτες, οι οποίοι ήρθαν για να δουν τα έργα του από κοντά. Θεωρούμε ότι το κάθε άτομο ξεχωριστά είναι ένας κόμβος. Έστω ότι ο κάθε κόμβος δείχνει σε αυτόν που γνωρίζει, τότε λογικά οι περισσότεροι άνθρωποι ή κόμβοι που βρίσκονται στο χώρο της έκθεσης λογικά γνωρίζουν το ζωγάφο (ο οποίος είναι το κοινώς γνωστό πρόσωπο) και επομένως θα δείχνουν προς τον κόμβο του ατόμου αυτού. Ο ζωγράφος όμως κατά πάσα πιθανότητα γνωρίζει μόνο την οικογένειά του και συνεπώς ο κόμβος του θα συνδέεται μόνο με τους κόμβους της οικογένειάς του.

Συνεπώς σε αυτόν τον γράφο μόνο όσοι γνωρίζουν τον ζωγάφο έχουν και ακμές προς αυτό το άτομο, σε αντίθεση με τον ζωγάφο που δεν δείχνει στους περισσότερους επειδή δεν τους γνωρίζει. Προφανώς αυτό ισχύει επειδή δεν είναι δυνατό ένας κόμβος να γνωρίζει όλους τους κόμβους οι οποίοι τον γνωρίζουν και το αντίθετο.

Ένας τέτοιου είδους γράφος ονομάζεται «κατευθυνόμενος», επειδή υπάρχουν οι κόμβοι οι οποίοι συνδέονται με βέλη (ακμές). Ο γράφος αυτός μπορεί να επεκταθεί με την προσθήκη βαρών στις ακμές, ώστε να προκύπτει και ένας βαθμός γνωριμίας κάθε ατόμου με ένα γειτονικό του. Ένας τέτοιος γράφος ονομάζεται «σταθμισμένος».

4.1. Ορισμοί

4.1.1. Γράφος

Ένας **γράφος** περιγράφεται ως ένα διατεταγμένο ζεύγος:

$$G = (V, E)$$

Το ζεύγος αυτό αποτελείται από ένα σύνολο V των κορυφών ή κόμβων και ένα σύνολο E , το οποίο είναι το σύνολο από ακμές ή τόξα, τα οποία συνδέουν δύο στοιχεία του συνόλου V . Συνεπώς, εάν μια ακμή συνδέεται με δύο κόμβους χωρίς να δείχνει την κατεύθυνση, η σχέση απεικονίζεται ως μη ταξινομημένο ζεύγος των κόμβων σε σχέση με τη συγκεκριμένη ακμή.

Όταν δύο κόμβοι συνδέονται μεταξύ τους με μια ή περισσότερες ακμές τότε λέγονται γειτονικοί ή γειτνιάζοντες κόμβοι.

Οι κόμβοι που ανήκουν σε μια ακμή ονομάζονται τελικές κορυφές της ακμής. Ένας κόμβος μπορεί να υπάρχει σε έναν γράφο και να μην υπάρχει καμία ακμή που να έρχεται σε αυτόν ή να φεύγει από αυτόν.

Η τάξη ενός γράφου ισούται με το πλήθος των κορυφών, ενώ το μέγεθος ενός γράφου ισούται με το πλήθος των ακμών του. Ο βαθμός ενός κόμβου ισούται με τον αριθμό των ακμών που εισέρχονται ή εξέρχονται από αυτόν. Μια ακμή η οποία αρχίζει και καταλήγει στον ίδιο κόμβο δημιουργώντας έναν βρόχο, συνυπολογίζεται δύο φορές, τόσο ως εισερχόμενη όσο και ως εξερχόμενη.

4.1.2. Μονοπάτια γράφων

Ένα **μονοπάτι** από κόμβο σε κόμβο ενός γράφου λέγεται ακολουθία κόμβων, όπου ο κάθε κόμβος στην ακολουθία συνδέεται με τον επόμενο του μέσω ακμών. Το μήκος ενός μονοπατιού σε έναν γράφο ορίζεται από το πλήθος των κόμβων που συνδέει, ή αλλιώς είναι ίσο με το πλήθος των ακμών, προσθέτοντας μία μονάδα $(n+1)$. Συνεπώς, εάν ένας γράφος δεν έχει ακμές σε ένα μονοπάτι του, το μονοπάτι δεν υπάρχει ή αλλιώς είναι ένα μονοπάτι μηδενικού μήκους. Στην περίπτωση κατά την οποία ένα μονοπάτι είναι κατευθυνόμενο, όταν επισκεπτόμαστε έναν κόμβο δεν μπορούμε να επιστρέψουμε στον προηγούμενό του, αλλά μόνο να συνεχίσουμε στον επόμενο μέχρι το τέλος του μονοπατιού. Αντίθετα, στην περίπτωση του μη κατευθυνόμενου γράφου, μπορούμε να μεταβούμε στον επόμενο κόμβο αλλά και στον προηγούμενο. Συνεπώς και οι δύο κατευθύνσεις είναι δυνατές σε αυτό το είδος γράφων.

4.2. Τύποι γράφων

4.2.1. Τι είναι οι τύποι γράφων

Υπάρχουν πολλοί τύποι γράφων, οι οποίοι υποστηρίζουν διάφορες διαδικασίες. Στο πλαίσιο της διπλωματικής εργασίας θα εστιάσουμε σε δύο τύπους γράφων, τον κατευθυνόμενο γράφο αλλά και τον σταθμισμένο γράφο, ο οποίος ουσιαστικά είναι ο κατευθυνόμενος γράφος με βάρη. Η χρήση και των δύο γράφων επεξηγείται στη συνέχεια αναλυτικά.

4.2.2. Σταθμισμένο γράφημα

Ένας γράφος ονομάζεται σταθμισμένος, εάν σε κάθε ακμή του αντιστοιχίζεται ένας αριθμός – ο οποίος ονομάζεται «βάρος», το οποίο έχει κάποια σημασία σε σχέση με το εκάστοτε πρόβλημα (κόστος, απόσταση, κλπ.). Για παράδειγμα το βάρος θα μπορούσε να δηλώνει την απόσταση από έναν κόμβο σε έναν άλλον ή το χρόνο που θα χρειαστεί για να φτάσει ένα αντικείμενο από τον ένα κόμβο σε κάποιον άλλο.

Μία κλάση προβλημάτων η οποία επιλύεται με χρήση σταθμισμένων γράφων, είναι τα προβλήματα βέλτιστης δρομολόγησης.

4.3. Κατευθυνόμενος γράφος

Ένας κατευθυνόμενος γράφος είναι ένα ζευγάρι

$$G = (V, A) \text{ ή } G = (V, E)$$

V είναι ένα σύνολο που περιέχει στοιχεία τα οποία πρακτικά είναι οι κόμβοι και A ή E τα σύνολα των ακμών, οι οποίες συνδέουν ζεύγη κόμβων κατά διατεταγμένο τρόπο και ονομάζονται κατευθυνόμενες ακμές ή τόξα.

Επίσης, όταν υπάρχει μια κατευθυνόμενη ακμή $a=(x, y)$ αυτό σημαίνει ότι στον γράφο υπάρχουν 2 κόμβοι, ο x και ο y , με την ακμή να ξεκινά από τον x και να καταλήγει στον y . Η ακμή αυτή δείχνει από τον x προς τον y αλλά όχι το αντίθετο. Η ύπαρξη της αντίθετης ακμής προϋποθέτει την ύπαρξη ενός ζεύγους $b=(y, x)$.

Ένας κατευθυνόμενος γράφος G είναι συμμετρικός όταν για κάθε ακμή που ξεκινάει από έναν κόμβο με προορισμό έναν άλλο κόμβο, υπάρχει και μια ακόμα ακμή με την αντίστροφη ακριβώς κατεύθυνση.

4.4. PageRank

Ο **PageRank** είναι ο πρώτος αλγόριθμος που χρησιμοποιήθηκε για ταξινόμηση σελίδων αποτελεσμάτων αναζήτησης. Είναι ο πιο γνωστός αλγόριθμος της Google, ο

οποίος δημιουργήθηκε από τους Larry Page και Sergey Brin, με σκοπό να κατατάξει τις σελίδες στο διαδίκτυο με το να τις αξιολογεί με βάση την ποιότητα της κάθε σελίδας.

Η βασική λογική του PageRank είναι να συμπεριλαμβάνει στη συνάρτηση υπολογισμού της μετρικής αυτής, τόσο το πλήθος των backlinks (εξερχόμενων συνδέσεων από την εκάστοτε σελίδα) όσο και το πλήθος των inbound links (εισερχόμενες συνδέσεις προς την εν λόγω σελίδα).

Προφανώς η Google χρησιμοποίησε και άλλους αλγορίθμους, αλλά ο βασικός για πολλά χρόνια ήταν ο PageRank. Ένας από τους λόγους που ο αλγόριθμος αυτός χρησιμοποιείται μέχρι και σήμερα σε διάφορα συστήματα, είναι ότι είναι υπολογιστικά αποτελεσματικός (Mind the SEO, 2020).

Η Google εγκατέλειψε τον αλγόριθμο PageRank στη μηχανή αναζήτησής της το 2014, επειδή επινοήθηκαν τρόποι οι οποίοι τον υποχρέωναν σε εξαγωγή εσφαλμένων αποτελεσμάτων (Southern, 2016). Εταιρείες βάσισαν τις επιχειρηματικές τους κινήσεις στην κατάχρηση του αλγορίθμου PageRank. Ο αλγόριθμος όμως εξακολουθεί να έχει εφαρμογές σε ιδιωτικά δίκτυα και ιδιωτικές σελίδες, όπου η αποτελεσματικότητά του τον καθιστά την πρωταρχική επιλογή (Sullivan, 2016).

4.4.1. Η ιδέα του αλγορίθμου PageRank

Η ιδέα του PageRank ξεκίνησε από διαφορά ακαδημαϊκά έγγραφα που θεωρούνταν πιο έγκυρα ανάλογα με το πόσες περισσότερες πηγές δήλωνε κάθε ένα. Βασισμένοι σε αυτή την ιδέα, οι συνιδρυτές της Google χρησιμοποίησαν τον ίδιο αλγόριθμο πάνω στις ιστοσελίδες με βάση τις εισερχόμενες συνδέσεις ως δείκτη του κατά ποσό έγκυρες είναι και συνεπώς πόσο υψηλότερη θέση θα κατέχουν στην κατάταξη των αποτελεσμάτων αναζήτησης.

4.4.2. Τρόπος λειτουργίας του αλγορίθμου PageRank

Ο PageRank λειτουργεί χρησιμοποιώντας τις συνδέσεις από την μια σελίδα στην άλλη. Όσο περισσότερες εισερχόμενες και εξερχόμενες συνδέσεις έχει μια σελίδα τόσο καλύτερα βαθμολογείται η σελίδα. Όσο πιο πολλές συνδέσεις έχει μια σελίδα η οποία με τη σειρά της δείχνει σε μια δική μας σελίδα, αυτό το χαρακτηριστικό αυξάνει τη βαθμολογία της δικής μας σελίδας. Το συμπέρασμα είναι ότι σημασία στην κατάταξη έχει και η σπουδαιότητα της κάθε σελίδας.

Είναι επίσης γνωστό ότι η κλίμακα που χρησιμοποιεί ο PageRank για να βαθμολογεί την κάθε σελίδα είναι από το 0 μέχρι το 10. Όσο πιο μεγάλο βαθμό έχει η κάθε ιστοσελίδα τόσο υψηλότερα θα βρίσκεται στην κατάταξη των αποτελεσμάτων αναζήτησης (Search Engine Result Pages).

4.4.3. Ιστορία του PageRank

Όπως αναφέρθηκε ήδη, ο αλγόριθμος PageRank φτιάχτηκε από τους Larry Page και Sergey Brin στο πανεπιστήμιο του Στάνφορντ το 1996 καθώς διεξαγόταν έρευνα πάνω στην ανακάλυψη ενός είδους μηχανής αναζήτησης.

Ο Sergey Brin σκέφτηκε ότι ο καλύτερος τρόπος για να ταξινομήσει πληροφορίες είναι να λαμβάνεται υπόψη το πόσο γνωστός είναι ένας σύνδεσμος, και ότι το πλήθος των συνδέσεων που έχει θα ήταν μια καλή μετρική για να κατατάξει μια σελίδα πιο υψηλά στα αποτελέσματα αναζήτησης.

Στην δημιουργία του αλγόριθμου βοήθησαν και ο Scott Hassan όπως και ο Alan Steremberg που ήταν βασικοί παράγοντες για την ανάπτυξη του, αλλά και η Google. Κατά το 1998 δημοσιεύτηκε το πρωτότυπο της μηχανής αναζήτησης της Google και λίγο αργότερα ιδρύθηκε η εταιρία Google Inc.

Η λέξη PageRank είναι εμπορικό σήμα της Google και πήρε το όνομά της από το όνομα του Larry Page. Απέκτησε δίπλωμα ευρεσιτεχνίας των ΗΠΑ, αν και το δίπλωμα ανήκει στο πανεπιστήμιο του Stanford και όχι στην εταιρία.

Η Google αν και δεν πήρε το δίπλωμα ευρεσιτεχνίας, έχει τα αποκλειστικά δικαιώματα από το πανεπιστήμιο, μετά από την εκχώρηση 1,8 εκατομμυρίων μετοχών της Google στο πανεπιστήμιο, το οποίο τις πούλησε το 2005 για 336 εκατομμύρια δολάρια.

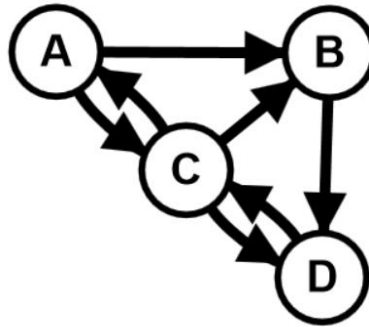
Ο PageRank δεν χρησιμοποιείται άμεσα από την Google, Η εταιρεία έκανε την τελευταία ενημέρωση του αλγόριθμου το 2014. Ο λόγος είναι ότι υπήρχε συνεχής κατάχρηση από πολλές ομάδες που έβρισκαν τρόπους να τροποποιούν τα αποτελέσματα με τρόπο μη θεμιτό, και προφανώς τα κέρδη ήταν πολύ μεγάλα για να σταματήσουν να το κάνουν.

4.4.4. Παράδειγμα λειτουργίας του PageRank στην πράξη

Η μετρική PageRank υπολογίζεται με την παρακάτω συνάρτηση:

$$PR_{t+1}(P_i) = \sum_{P_j} \frac{PR(P_j)}{C(P_j)}$$

Εξετάζοντας τον κατευθυνόμενο γράφο με 4 κόμβους A,B,C,D και τις αντίστοιχες ακμές (Σχήμα 4), θα εκτελέσουμε τον αλγόριθμο PageRank βάσει του Πίνακα 2, ώστε να προκύψει το αποτέλεσμα PageRank για κάθε ένα κόμβο. Προκειμένου να πιστοποιηθεί η ορθότητα της εκτέλεσης του αλγόριθμου σε κάθε επανάληψη, ο αλγόριθμος θα πρέπει να δίνει την τιμή 1 για όλους τους κόμβους.



Σχήμα 4. Κατευθυνόμενος γράφος τεσσάρων κόμβων
για τον υπολογισμό του PageRank

1. Στην πρώτη επανάληψη (Iteration 0) υπολογίζουμε τον αρχικό κόμβο δια το πλήθος των υπολοίπων κόμβων, για κάθε έναν κόμβο, συνεπώς $1/4$ σε όλα τα κελιά του Iteration 0.
2. Στη δεύτερη επανάληψη (Iteration 1) δεχόμαστε την τιμή της προηγούμενης επανάληψης και προσθέτουμε από αυτούς που δείχνουν στον κάθε κόμβο δια το πλήθος των κόμβων που δείχνουν αυτοί οι κόμβοι συνεπώς ο A θα έχει την τιμή $(1/4)/3 = 1/12$ επειδή το C δείχνει και στους άλλους 3, ο B παίρνει από τον A και από τον C, αλλά ο A δείχνει και στον A αλλά και στον B, άρα $(1/4)/2 +$ από τον C που δείχνει και στον A και στον B αλλά και στον D $(1/4)/3$ άρα $(1/4)/2 + (1/4)/3 = 2,5/12$, για τον C τον δείχνουν οι A και D άρα $(1/4)/2 + (1/4)/1 = 4,5/12$ και τέλος για τον D μόνο ο B και ο C τον δείχνουν οπότε $(1/4)/3 + (1/4)/1$ άρα $4/12$.
3. Στην τρίτη επανάληψη (Iteration 2) δεχόμαστε την τιμή από την προηγούμενη τιμή του κόμβου δια το πλήθος των εξερχόμενων ακμών, συνεπώς τα αποτελέσματα είναι τα εξής: για το $A = (4,5/12)/3 = 1,5/12$, για το $B = (1/12)/2 + (4,5/12)/3 = 2/12$, για το $C = (1/12)/2 + (4/12)/1 = 4,5/12$ και τέλος για το $D = (2,5/12)/1 + (4,5/12)/3 = 4/12$.
4. Τέλος τα αποτελέσματα του PageRank προκύπτουν με βάση ποιος κόμβος έλαβε το μεγαλύτερο και καταλήγουμε με πρώτο το A, δεύτερο το B, τρίτο το D και τελευταίο το C.

Αποδεικνύεται ότι όλοι οι υπολογισμοί είναι σωστοί δεδομένου ότι όλες οι επαναλήψεις (iterations) ξεχωριστά δίνουν αθροιστικά αποτέλεσμα ίσο με το 1.

Πίνακας 2. Υπολογισμός PageRank σε διαδοχικές επαναλήψεις, για τον γράφο του Σχήματος 4

	Iteration 0	Iteration 1	Iteration 2	PageRank
A	1/4	1/12	1.5/12	1
B	1/4	2.5/12	2/12	2
C	1/4	4.5/12	4.5/12	4
D	1/4	4/12	4/12	3

5. Περιγραφή εφαρμογής

Η εφαρμογή που αναπτύχθηκε στο πλαίσιο της εργασίας αυτής υλοποιεί τον αλγόριθμο Top-k, ώστε να επιστρέψει στο χρήστη που τη χρησιμοποιεί ένα μικρό σύνολο αποτελεσμάτων, τα οποία έχουν μεγαλύτερη συσχέτιση με τους όρους αναζήτησης αλλά και τα αντίστοιχα βάρη των όρων αυτών.

Η εφαρμογή δέχεται ως είσοδο ένα σύνολο δεδομένων (στη συγκεκριμένη περίπτωση ένα σύνολο ενοικιαζόμενων οικημάτων Airbnb) και ένα σύνολο κριτηρίων επιλογής που θέτει ο ενδιαφερόμενος χρήστης. Σε κάθε κριτήριο ο χρήστης αποδίδει ένα συντελεστή βαρύτητας, ο οποίος επί της ουσίας εκφράζει το πόσο σημαντικός είναι ο συγκεκριμένος παράγοντας για την επιλογή του.

Το σύστημα επεξεργάζεται τα δεδομένα του και χρησιμοποιώντας τεχνικές PageRank βαθμολογεί τους κόμβους των αποτελεσμάτων αναζήτησης, ώστε να καταλήξει στους k πλέον «σημαντικούς» κόμβους (αποτελέσματα), τους οποίους τελικά παρουσιάζει στον χρήστη.

5.1. Top-k αλγόριθμος στην εφαρμογή

Ο αλγόριθμος Top-k έχει σαν σκοπό από ένα μεγάλο σύνολο αποτελεσμάτων να εντοπίσει ένα σημαντικά μικρότερο πλήθος k στοιχείων. Τα k αυτά στοιχεία είναι λίγα σε σχέση με τα αρχικά ογκώδη δεδομένα, ώστε η περαιτέρω επεξεργασία τους να μπορεί να γίνει πιο εύκολα, ενώ ταυτόχρονα το υπολογιστικό σύστημα απαιτεί λιγότερους πόρους για να τα διαχειριστεί.

Το αποτέλεσμα της εφαρμογής αλγόριθμου Top-k περιγράφεται γλαφυρά στο παρακάτω παράδειγμα. Έστω ένας χρήστης ο οποίος αναζητά στο διαδίκτυο αξιοθέατα στην Ελλάδα και το σύστημα επιστρέφει πολλές χιλιάδες δεδομένα, προκειμένου να επιλέξει αυτά που τον ενδιαφέρουν περισσότερο. Η εφαρμογή ενός Top-k αλγόριθμου για $k=10$, θα οδηγούσε στον εντοπισμό 10 σημείων ενδιαφέροντος, έτσι ώστε αφενός ο χρήστης να μπορεί να αντλήσει γρήγορα την πληροφορία χωρίς να μπει στη διαδικασία ατέρμονου φιλτραρίσματος των αποτελεσμάτων (πράγμα που για παράδειγμα θα μπορούσε να οδηγήσει σε παραπάνω χρεώσεις λόγω χρήσης Internet σε μία φορητή συσκευή), και αφετέρου δε θα επιβάρυνε τη συσκευή με τη χρήση πρόσθετων υπολογιστικών πόρων που χρειάζονται σε άλλες εργασίες του συστήματος (Vlachou, et al., 2020). Στο πλαίσιο της εργασίας αυτής επιλέχθηκε ένα σύνολο δεδομένων ενοικιαζόμενων διαμερισμάτων της πλατφόρμας Airbnb στη Νέα Υόρκη. Στον Πίνακα 3 που παρατίθεται στη συνέχεια, καταγράφονται τα χαρακτηριστικά των ενοικιαζόμενων διαμερισμάτων: όνομα, τιμές ενοικίασης, απόσταση από το μετρό,

διαθεσιμότητα².

Προκειμένου το σύστημα να είναι πρακτικό στη χρήση του και από χρήστες οι οποίοι δεν είναι έμπειροι σε θέματα τεχνολογίας, εισάχθηκε ο φορμαλισμός ενός «πίνακα προτιμήσεων», ο οποίος παρατίθεται στη συνέχεια (Πίνακας 3). Στον πίνακα αυτό ο χρήστης εισάγει τις προτιμήσεις του υπό τύπο βαρών για το κάθε χαρακτηριστικό του ενοικιαζόμενου διαμερίσματος.

Έστω ότι ζητείται από το σύστημα να επιστρέψει τα $k=3$ καλύτερα στοιχεία από αυτόν τον συνολικό πίνακα δεδομένων, ο οποίος περιέχει 6 εγγραφές. Αν χρησιμοποιηθεί η συνάρτηση:

$$F_{\text{ατόμου}}(x) = (\text{βάρος τιμής}) * (\text{τιμή}) + (\text{βάρος απόσταση από το μετρό}) * (\text{απόσταση από το μετρό}) + (\text{βάρος μη διαθεσιμότητας}) * (\text{κατειλημμένο}) \quad (1)$$

σε κάθε ένα από τα αποτελέσματα της αναζήτησης, θα προκύψει ένας βαθμός (score). Στο συγκεκριμένο παράδειγμα, όσο μικρότερος είναι αυτός ο βαθμός, τόσο το καλύτερο το αποτέλεσμα.

Έστω ότι εφαρμόζουμε τον τύπο αυτό για τις προτιμήσεις δύο υποτιθέμενων χρηστών, του Αλέξανδρου και του Γιάννη. Επομένως, εφαρμόζοντας τη συνάρτηση (1) προκύπτουν τα ακόλουθα αποτελέσματα βάσει του αλγόριθμου Top-k για τον Αλέξανδρο και τον Γιάννη³:

Για το χρήστη Αλέξανδρο:

$$F_{\text{Αλέξανδρος}} (\text{Clean \& quiet apt home by the park}) = 0.7 * 165 + 0.2 * 280 + 0.1 * 0 \\ = \underline{171.5}$$

$$F_{\text{Αλέξανδρος}} (\text{Skylit Midtown Castle}) = 0.7*225 + 0.2*400 + 0.1*10 \\ = \underline{238.5}$$

$$F_{\text{Αλέξανδρος}} (\text{THE VILLAGE OF HARLEM.NEW YORK}) = 0.7*150 + 0.2*300+ 0.1*100 \\ = \underline{175}$$

$$F_{\text{Αλέξανδρος}} (\text{Cozy Entire Floor of Brownstone}) = 0.7 * 89 + 0.2* 350+ 0.1 * 120 \\ = \underline{144.3}$$

$$F_{\text{Αλέξανδρος}} (\text{Entire Apt: Spacious Studio/Loft by central park}) = 0.7 * 230 + 0.2 * 225+ 0.1 * 25 \\ = \underline{208.5}$$

² Ορισμένα δεδομένα δεν ισχύουν στην πραγματικότητα και έχουν δημιουργηθεί καθαρά για το συγκεκριμένο παράδειγμα. Ένα τέτοιο παράδειγμα είναι το πεδίο «Απόσταση από το Μετρό», του οποίου οι τιμές είναι εικονικές

³ Wikipedia, Partial sorting, 2020
https://en.wikipedia.org/wiki/Partial_sorting

$$F_{\text{Αλέξανδρος}} (\text{Large Cozy 1 BR Apartment In Midtown East}) = 0.7 \cdot 180 + 0.2 \cdot 340 + 0.1 \cdot 200 \\ = \underline{\underline{214}}$$

Για το χρήστη Γιάννη:

$$F_{\text{Γιάννης}} (\text{Clean \& quiet apt home by the park}) = 0.5 \cdot 165 + 0.3 \cdot 280 + 0.2 \cdot 0 \\ = \underline{\underline{166.5}}$$

$$F_{\text{Γιάννης}} (\text{Skylit Midtown Castle}) = 0.5 \cdot 225 + 0.3 \cdot 400 + 0.2 \cdot 10 \\ = \underline{\underline{234.5}}$$

$$F_{\text{Γιάννης}} (\text{THE VILLAGE OF HARLEM.NEW YORK}) = 0.5 \cdot 150 + 0.3 \cdot 300 + 0.2 \cdot 100 \\ = \underline{\underline{185}}$$

$$F_{\text{Γιάννης}} (\text{Cozy Entire Floor of Brownstone}) = 0.5 \cdot 89 + 0.3 \cdot 350 + 0.2 \cdot 120 \\ = \underline{\underline{173.5}}$$

$$F_{\text{Γιάννης}} (\text{Entire Apt: Spacious Studio/Loft by central park}) = 0.5 \cdot 230 + 0.3 \cdot 225 + 0.2 \cdot 25 \\ = \underline{\underline{187.5}}$$

$$F_{\text{Γιάννης}} (\text{Large Cozy 1 BR Apartment In Midtown East}) = 0.5 \cdot 180 + 0.3 \cdot 340 + 0.2 \cdot 200 \\ = \underline{\underline{232}}$$

**Πίνακας 3. Ενδεικτικό σύνολο δεδομένων 6 ενοικιαζόμενων διαμερισμάτων
Airbnb, για τον υπολογισμό των k=3 κορυφαίων προτάσεων
αλγορίθμου Top-k**

Όνομα	Τιμή	Απόσταση από το Μετρό (m)	Κατειλημμένο
Clean & quiet apt home by the park	165	280	0
Skylit Midtown Castle	225	400	10
THE VILLAGE OF HARLEM NEW YORK	150	300	100
Cozy Entire Floor of Brownstone	89	350	120
Entire Apt: Spacious Studio/Loft by central park	230	225	25
Large Cozy 1 BR Apartment in Midtown East	180	340	200

Πίνακας 4. Υπόδειγμα «Πίνακα προτιμήσεων» του χρήστη. Βάσει των βαρών γίνεται η εφαρμογή του αλγορίθμου Top-k

	Βάρη		
Χρήστης	Τιμή	Απόσταση από το Μετρό	Μη-διαθεσιμότητα
Αλέξανδρος	0,7	0,2	0,1
Γιώργος	0,6	0,3	0,1
Γιάννης	0,5	0,3	0,2
Μιχάλης	0,4	0,4	0,2
Δημήτρης	0,3	0,5	0,2
Κώστας	0,2	0,5	0,3

Βάσει των παραπάνω υπολογισμών προκύπτει ότι τα κορυφαία k που θα πρέπει να προταθούν σε κάθε χρήστη είναι (Πίνακας 5):

Πίνακας 5. Ενδεικτικό σύνολο των k=3 κορυφαίων προτάσεων του αλγορίθμου Top-k για τον κάθε ένα από τους χρήστες

Χρήστης	1ο Καλύτερο (Top-1)	2ο καλύτερο (Top-2)	3ο καλύτερο (Top-3)
Αλέξανδρος	Cozy Entire Floor of Brownstone	Clean & quiet apt home by the park	THE VILLAGE OF HARLEM NEW YORK
Γιώργος	Cozy Entire Floor of Brownstone	THE VILLAGE OF HARLEM NEW YORK	Entire Apt: Spacious Studio/Loft by central park
Γιάννης	Clean & quiet apt home by the park	Cozy Entire Floor of Brownstone	THE VILLAGE OF HARLEM NEW YORK
Μιχάλης	Clean & quiet apt home by the park	Entire Apt: Spacious Studio/Loft by central park	Cozy Entire Floor of Brownstone
Δημήτρης	Entire Apt: Spacious Studio/Loft by central park	Clean & quiet apt home by the park	THE VILLAGE OF HARLEM NEW YORK
Κώστας	Entire Apt: Spacious Studio/Loft by central park	Clean & quiet apt home by the park	THE VILLAGE OF HARLEM NEW YORK

Σύμφωνα με τον Πίνακα 5 τα κορυφαία στοιχεία k για τον χρήστη Αλέξανδρο είναι το

(Cozy Entire Floor of Brownstone), το (Clean & quiet apt home by the park) και το (THE VILLAGE OF HARLEM NEW YORK) σε αντίθεση με τον Γιάννη που είναι τα (Clean & quiet apt home by the park), το (Cozy Entire Floor of Brownstone) και το (THE VILLAGE OF HARLEM NEW YORK). Συνεπώς αυτά τα 3 στοιχεία θα πρέπει το σύστημα να παρουσιάσει τελικά στον αντίστοιχο χρήστη σαν τις καλύτερες προτάσεις Airbnb, με βάση τις προτιμήσεις του.

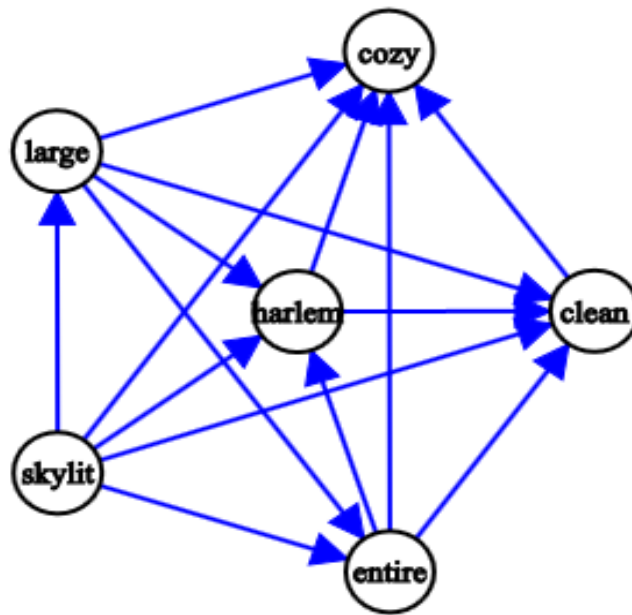
Η εφαρμογή μοντελοποιεί τα αποτελέσματα της αναζήτησης χρησιμοποιώντας κατευθυνόμενο γράφο, με σκοπό στους «καλύτερους» κόμβους να καταλήγουν ακμές από άλλους κόμβους, οι οποίοι όμως δεν είναι τόσο «καλοί» σε σχέση με τις επιθυμίες του χρήστη. Αυτό σημαίνει ότι ένα προς ενοικίαση διαμέρισμα δείχνει προς ένα καλύτερό του (σε σχέση με την επιθυμία και τα κριτήρια του χρήστη).

5.2. Πώς χρησιμοποιεί τους γράφους η εφαρμογή

Η εφαρμογή χρησιμοποιεί κατευθυνόμενους γράφους ώστε οι ακμές που φτάνουν σε έναν κόμβο να προέρχονται από κόμβους οι οποίοι βρίσκονται χαμηλότερα στην προτίμηση του χρήστη.

Στη συνέχεια παρατίθεται η απεικόνιση ενός κατευθυνόμενου γράφου για τον χρήστη Αλέξανδρο, στον οποίο αναπαρίσταται το «καλύτερο» Airbnb για τον ίδιο. Στον ίδιο γράφο αποτυπώνεται το γεγονός ότι ο κάθε κόμβος δείχνει στον καλύτερό του, με κατευθυνόμενες ακμές (βάσει των συγκεκριμένων συνόλων προτιμήσεων), όπως μπορούμε να δούμε στο παρακάτω σχήμα.

Πιο συγκεκριμένα το πρώτο βήμα για να φτιάξουμε τον γράφο είναι αρχικά να εισάγουμε όλους τους κόμβους που μας ενδιαφέρουν. Στην προκειμένη περίπτωση ο κάθε κόμβος είναι ένα Airbnb από το παραπάνω παράδειγμα του πίνακα 5. Στην συνέχεια θα πρέπει ο κάθε κόμβος να έχει κατευθυνόμενες ακμές προς τους κόμβους (Airbnb) που είναι καλύτερα από τον ίδιο. Ο “cozy” είναι ο καλύτερος κόμβος οπότε δεν θα υπάρχει καμία ακμή από αυτόν προς κάποιον άλλο κόμβο, σε αντίθεση με τον “harlem” που είναι ο τρίτος καλύτερος και συνεπώς θα πρέπει να εξέρχεται από τον ίδιο μια κατευθυνόμενη ακμή προς τον top-1 (“cozy”) και top-2 (“clean”) κόμβο, και οι υπόλοιποι κόμβοι (“large”, “skylit”, “entire”) να δείχνουν σε αυτόν με τις ανάλογες ακμές.



Σχήμα 5. Κατευθυνόμενος γράφος αναπαράστασης των καλύτερων αποτελεσμάτων βάσει του αλγόριθμου Top-k.⁴

Οι γράφοι χρησιμοποιούνται στην εφαρμογή για την αναπαράσταση των αποτελεσμάτων της αναζήτησης. Όταν ολοκληρωθεί ο Top-k αλγόριθμος στο πρόγραμμα, πρέπει να δημιουργηθεί ένας κατευθυνόμενος γράφος. Ο γράφος αυτός συνδέει τους καλύτερους k κόμβους για τον χρήστη, μόλις τελειώσουν όλες οι επαναλήψεις της εκτέλεσης του αλγόριθμου Top-k με διαφορετικές προτιμήσεις και με βάση την τοποθεσία του χρήστη. Στη συνέχεια δημιουργούνται αντίστοιχοι γράφοι και για τους πλέον «σημαντικούς» χρήστες. Η σημαντικότητα του χρήστη είναι ένα μέγεθος το οποίο λαμβάνεται από την εφαρμογή μέσω των πληροφοριών του προφίλ του κάθε χρήστη της εφαρμογής. Τα προφίλ των χρηστών αυτών, φορτώνονται στην εφαρμογή από ένα αρχείο τύπου CSV.

Στην συνέχεια οι κατευθυνόμενοι αυτοί γράφοι συγχωνεύονται ώστε οι κοινοί κόμβοι να εμφανίζονται μία και μοναδική φορά στον συγχωνευμένο γράφο. Αυτή η διαδικασία θα επαναληφθεί μερικές φορές, έως ότου ενσωματωθούν όλοι οι γράφοι που έχουν δημιουργηθεί για κάθε έναν χρήστη σε έναν ενιαίο, ο οποίος αργότερα θα αναλυθεί υπό το πρίσμα της οπτικοποίησης αλλά και υπό το πρίσμα του PageRank.

⁴ Ο γράφος δημιουργήθηκε με το εργαλείο https://csacademy.com/app/graph_editor/ tool

5.3. Χρήση του αλγόριθμου PageRank από την εφαρμογή

Όπως αναφέρθηκε προηγουμένως, η εφαρμογή συνθέτει έναν μεγάλο κατευθυνόμενο γράφο από επιμέρους μικρότερους γράφους αποτελεσμάτων. Ο γράφος αυτός περιλαμβάνει τους καλύτερους κόμβους (στην συγκεκριμένη περίπτωση διαμερίσματα Airbnb) σύμφωνα με τις προτιμήσεις και την τοποθεσία που βρίσκεται ο εκάστοτε χρήστης. Η διαδικασία περιλαμβάνει την επεξεργασία του γράφου, ώστε να προκύψει το αποτέλεσμα του PageRank γι’ αυτόν. Στο τέλος της διαδικασίας γίνεται ταξινόμηση κατά φθίνουσα σειρά, ώστε στην κορυφή της λίστας των αποτελεσμάτων να εμφανιστούν τα καλύτερα αποτελέσματα. Ο τρόπος υπολογισμού του PageRank ακολουθεί αυτόν που περιγράφηκε στο παράδειγμα της παραγράφου 4.4.4.

5.4. Οπτικοποίηση δεδομένων

Ο σκοπός της οπτικοποίησης είναι να προβάλλει δεδομένα σε μορφή εικόνας, δεδομένου ότι ο άνθρωπος πολλές φορές αντιλαμβάνεται καλύτερα τα δεδομένα σε μορφή εικόνας, παρά σε μορφή ενός συνόλου δομημένων δεδομένων ενός πίνακα. Επιπρόσθετα, η κατανόηση των αποτελεσμάτων ενός αλγορίθμου είμαι πιο αποδοτική με τη χρήση οπτικοποιημένων δεδομένων. Για το σκοπό αυτό η εφαρμογή που αναπτύχθηκε ενσωμάτωσε εργαλεία και βιβλιοθήκες λογισμικού, ώστε να είναι εφικτή η αναπαράσταση των αποτελεσμάτων σε μορφή εικόνας και δίνοντάς τους μία μορφή αντιληπτή και κατανοητή στον άνθρωπο⁵.

⁵ Why spatial data is important, Safe Software
<https://www.safe.com/why/spatial-data/>

6. Λεπτομέρειες Υλοποίησης

6.1. Δεδομένα εισόδου της εφαρμογής

Τα δεδομένα εισόδου της εφαρμογής αποτελούν δύο αρχεία μορφής CSV:

1. Το πρώτο αρχείο περιέχει τα δεδομένα των αντικειμένων πάνω στα οποία θα γίνει η αναζήτηση. Στην προκειμένη περίπτωση τα αντικείμενα αυτά είναι ενοικιαζόμενα οικήματα της πλατφόρμας Airbnb. Για τα αντικείμενα αυτά τηρούνται εκτός από ένα μοναδικό αναγνωριστικό, η ονομασία τους, οι γεωγραφικές τους συντεταγμένες (γεωγραφικό μήκος και πλάτος), οι αξιολογήσεις τους και κάποιες πληροφορίες σχετικές με την ενοικίαση (όπως η τιμή ανά ημέρα και η διαθεσιμότητα). Η γραμμογράφηση του αρχείου των αντικειμένων φαίνεται στην Εικόνα 6.

	ID Airbnb	Longitude	Latitude	Price	Reviews count	Availability					
▲	A	B	C	D	E	F	G	H	I	J	K
1	2539		40.64749		-73.9724		149		9		365
2	2595		40.75362		-73.9838		225		45		355
3	3647		40.80902		-73.9419		150		0		365
4	3831		40.68514		-73.9598		89		270		194
5	5022		40.79851		-73.944		80		9		0
6	5099		40.74767		-73.975		200		74		129
7	5121		40.68688		-73.956		60		49		0
8	5178		40.76489		-73.9849		79		430		220
9	5203		40.80178		-73.9672		79		118		0
10	5238		40.71344		-73.9904		150		160		188

Εικόνα 6. Γραμμογράφηση του CSV αρχείου
των ενοικιαζόμενων οικημάτων Airbnb

2. Το δεύτερο αρχείο μορφής CSV περιέχει τις επιλογές ενοικιαζόμενων οικημάτων, οι οποίες έχουν γίνει από σημαντικούς χρήστες του δικτύου, βάσει των κριτηρίων που έθεσε ο καθένας απ' αυτούς. Αυτό σημαίνει ότι το αρχείο αυτό περιέχει την επιλογή ενός σημαντικού χρήστη και το κριτήριο (ή τα κριτήρια) βάσει των οποίων έγινε η επιλογή αυτή. Για την καλύτερη επίδοση της εφαρμογής, είναι σημαντικό να υπάρχει διαθέσιμο ένα δείγμα σημαντικών χρηστών με τις αντίστοιχες επιλογές τους, ώστε το τελικό αποτέλεσμα να προκύψει από το συνυπολογισμό των απόψεων χρηστών, οι οποίοι έκαναν στο παρελθόν αντίστοιχες επιλογές. Η γραμμογράφηση του αρχείου αυτού φαίνεται στην Εικόνα 7.

	Longitude	Latitude	Price weight	Reviews weight	Distance weight
▲	A	B	C	D	E
1	40.7375	-73.9799	0.875	0.099	0.026
2	40.722	-73.9882	0.062	0.836	0.102
3	40.7277	-73.9895	0.373	0.018	0.609
4	40.7457	-74.0261	0.645	0.007	0.348
5	40.8449	-73.9371	0.826	0.065	0.109
6	40.5896	-73.7384	0.773	0.199	0.028
7	40.7644	-73.9958	0.067	0.029	0.904
8	40.7275	-74.0386	0.968	0.007	0.025
9	40.7556	-73.9818	0.961	0.002	0.037
10	40.7215	-74.0045	0.343	0.229	0.428
11	40.6917	-73.9425	0.125	0.617	0.258
12	40.7557	-73.9923	0.967	0.02	0.013

Εικόνα 7. Γραμμογράφηση αρχείου επιλογών ενοικιαζόμενων οικημάτων άλλων χρηστών του δικτύου

Στη συνέχεια περιγράφεται ο τρόπος ενσωμάτωσης του αλγόριθμου Top-k στην εφαρμογή καθώς και η εφαρμογή του αλγόριθμου Page Rank.

6.2. Επεξεργασία δεδομένων

Η ροή επεξεργασίας των δεδομένων εισόδου της εφαρμογής περιγράφεται στη συνέχεια.

Βήμα 1

Η εφαρμογή διαβάζει το αρχείο CSV των ενοικιαζόμενων οικημάτων και δημιουργεί για κάθε γραμμή του αρχείου ένα αντικείμενο της κλάσης Hotel. Η δημιουργία των αντικειμένων αυτών διευκολύνει την επεξεργασία του μεγάλου όγκου δεδομένων. Παράλληλα, κατά την επεξεργασία του αρχείου ενοικιαζόμενων και τη δημιουργία των αντικειμένων γίνεται ο υπολογισμός των ελάχιστων και μέγιστων τιμών για τις μεταβλητές «γεωγραφικό μήκος» (longitude) και «γεωγραφικό πλάτος» (latitude). Επίσης για κάθε ακίνητο υπολογίζεται το πλήθος των αξιολογήσεων καθώς και ο βαθμός με τον οποίο έχει αξιολογηθεί το ακίνητο.

Βήμα 2

Στο επόμενο βήμα της διαδικασίας επεξεργασίας των δεδομένων εισαγωγής, διαβάζεται το δεύτερο CSV αρχείο, το οποίο περιέχει τους πιο σημαντικούς χρήστες καθώς και τις προτιμήσεις τους. Δεδομένου ότι το πλήθος τους είναι ακαθόριστο, η αποθήκευση κάθε αντικειμένου γίνεται σε μία δομή αποθήκευσης ArrayList της γλώσσας Java. Κάθε αντικείμενο του ArrayList ανήκει στην κλάση Preferences της

εφαρμογής.

Βήμα 3

Τα δύο προηγούμενα βήματα αποτελούν τη φάση της προετοιμασίας της εφαρμογής ώστε να είναι σε θέση να δεχτεί τις προτιμήσεις του χρήστη με μορφή κριτηρίων για την αναζήτηση των αντικειμένων Airbnb τα οποία ανταποκρίνονται περισσότερο στις επιθυμίες του. Τα κριτήρια που μπορεί ο χρήστης να δώσει είναι:

1. Η τιμή
2. Η αξιολόγηση
3. Η απόσταση

Για τις παραμέτρους αυτές ο χρήστης χρησιμοποιεί μία κλίμακα από το 0 έως το 10. Ο αντίστοιχος αριθμός της κλίμακας εκφράζει το βάρος της αντίστοιχης παραμέτρου στην αναζήτηση ή διαφορετικά η τιμή 0 σημαίνει ότι ο χρήστης δεν ενδιαφέρεται καθόλου για το συγκεκριμένο χαρακτηριστικό, ενώ η τιμή 10 ότι ενδιαφέρεται πάρα πολύ για αυτό. Στην Εικόνα 8 φαίνεται το περιβάλλον της εφαρμογής, όπου ο χρήστης εισάγει τις προτιμήσεις του (βάρη) για κάθε παράμετρο.

```
run:
Reads a file
Give me how much do you care about price:
Enter number from 0 to 10: 2
Give me how much do you care about reviews:
Enter number from 0 to 10: 4
Give me how much do you care about distance:
Enter number from 0 to 10: 4
Enter your location: Give me x axe
40.64749
Give me y axe
-73.97237
```

Εικόνα 8. Περιβάλλον ορισμού βαρών για κάθε κριτήριο αναζήτησης

Ο χρήστης εισάγει επίσης την τοποθεσία του, σε μορφή γεωγραφικών συντεταγμένων (latitude, longitude). Βάσει της τοποθεσίας αυτής θα γίνει ο αντίστοιχος υπολογισμός οικημάτων με συνυπολογισμό του αντίστοιχου συντελεστή βαρύτητας.

Τέλος ο χρήστης ορίζει το πλήθος των κόμβων που επιθυμεί να συμπεριληφθούν στο αποτέλεσμα. Ένα ζήτημα που προκύπτει στο σημείο αυτό είναι το ότι το πρόγραμμα θα πρέπει (σε μία επόμενη έκδοσή του) να ορίζει μόνο του το όριο αυτό, βάσει της

πολυπλοκότητας των υπολογισμών που πρόκειται να κάνει κάθε φορά. Για μεγάλα σύνολα δεδομένων, η πολυπλοκότητα του αλγόριθμου κατασκευής του γράφου αυξάνεται πολύ, ενώ και η βιβλιοθήκη που χρησιμοποιείται για την οπτικοποίησή του δεν είναι σε θέση να απεικονίσει γράφους με περισσότερους κόμβους από ένα άνω όριο.

Κατά την εισαγωγή των δεδομένων αυτών γίνονται οι απαραίτητοι έλεγχοι, ώστε ο χρήστης να μην εισάγει μη έγκυρες τιμές στην εφαρμογή.

Βήμα 4

Η εφαρμογή στη συνέχεια αρχικοποιεί έναν κατευθυνόμενο γράφο κλάσης `DefaultDirectedGraph` της βιβλιοθήκης `JGraphT`. Αρχικά, στο γράφο αυτό δεν ορίζονται βάρη στις ακμές που συνδέουν τις κορυφές μεταξύ τους. Για κάθε στοιχείο που έχει εισαχθεί στο `ArrayList<Preferences>` χρησιμοποιούμε τον αλγόριθμο `Top-k`, στον οποίο διαβιβάζουμε ως είσοδο το πλήθος των κόμβων που επιθυμεί ο χρήστης να συμπεριληφθεί στο αποτέλεσμα, τις ιδιότητες του αντίστοιχου αντικειμένου κλάσης `Preferences` και τις οριακές τιμές των γεωγραφικών συντεταγμένων (ελάχιστα και μέγιστα γεωγραφικά μήκη και πλάτη), όπως υπολογίστηκαν στο Βήμα 1.

Για την υλοποίηση του αλγόριθμου `Top-k` χρησιμοποιήθηκε ουρά προτεραιότητας (`PriorityQueue`), ώστε τα αποτελέσματα τα οποία προκύπτουν από την εφαρμογή των κριτηρίων στο σύνολο δεδομένων των ενοικιαζόμενων `Airbnb` να ταξινομούνται βάσει της σημαντικότητάς τους σε σχέση με τις προτιμήσεις του χρήστη.

Προκειμένου να γίνεται η σύγκριση δύο αντικειμένων, έγινε υπερκάλυψη (`override`) της μεθόδου “`compare`” της κλάσης `PriorityQueue`, ώστε η σύγκριση δύο αντικειμένων να γίνεται βάσει της βαθμολογίας που συγκεντρώνει το ενοικιαζόμενο οίκημα σύμφωνα με τα βάρη που εισήγαγε ο χρήστης. Ο υπολογισμός της βαθμολογίας αυτής γίνεται σύμφωνα με τον ακόλουθο τύπο:

$$\begin{aligned} \text{Total Rating} = & [(\text{βάρος απόστασης}) * (\text{απόσταση χρήστη-Airbnb}) / \\ & (\text{μέγιστη δυνατή απόσταση})] \\ & + [(\text{βάρος τιμής}) * [(\text{μέγιστη τιμή}) - (\text{τιμή του Airbnb})] / \\ & [(\text{μέγιστη τιμή}) - (\text{ελάχιστη τιμή})]] \\ & - [(\text{βάρος αξιολογήσεων}) * [(\text{αξιολογήσεις του Airbnb}) / \\ & [(\text{μέγιστη αξιολόγηση Airbnb}) \\ & - (\text{ελάχιστη αξιολογήσεων})]]] \end{aligned}$$

Ο υπολογισμός της απόστασης του χρήστη από το ενοικιαζόμενο οίκημα γίνεται μέσω κατάλληλης μεθόδου, η οποία δέχεται ως ορίσματα τις συντεταγμένες του χρήστη και του εκάστοτε ενοικιαζόμενου `Airbnb`. Ο υπολογισμός λαμβάνει υπόψη του την καμπυλότητα της γης, ώστε η απόσταση να προσδιορίζεται με ακρίβεια.

Βήμα 5

Η εφαρμογή προσδιορίζει τα k καλύτερα ενοικιαζόμενα, όπου k ο αριθμός των κόμβων που επέλεξε για τα αποτελέσματα ο χρήστης. Για καθένα από αυτά τα αντικείμενα ελέγχεται ο κατευθυνόμενος γράφος του Βήματος 4. Εάν το αντικείμενο δεν υπάρχει σαν κόμβος στο γράφο, τότε εισάγεται σε αυτόν και συνδέεται με τους επόμενούς του κόμβους σύμφωνα με τη σειρά προτεραιότητας. Στην περίπτωση που το αντικείμενο βρίσκεται ήδη στον γράφο ως κόμβος, τότε συνδέεται με τους γειτονικούς του κόμβους, προσθέτοντας ακμές με βάρος ανάλογα της ουράς προτεραιότητας. Η επεξεργασία αυτή αφορά τα ενοικιαζόμενα που επέλεξαν οι πλέον σημαντικοί χρήστες της εφαρμογής (Βήμα 2).

Στη συνέχεια η ίδια διαδικασία επαναλαμβάνεται για τις επιλογές του χρήστη (βάρη που έχει δώσει), με αποτέλεσμα να ολοκληρωθεί ο αλγόριθμος Top- k και να έχει ολοκληρωθεί και ο κατευθυνόμενος γράφος.

Βήμα 6

Στο επόμενο βήμα της διαδικασίας γίνεται η οπτικοποίηση του γράφου. Η εφαρμογή ελέγχει τυχόν IOExceptions τα οποία μπορούν να συμβούν στη βιβλιοθήκη JGraphT, σε περίπτωση που οι πόροι του συστήματος δεν επαρκούν. Η δημιουργία της οπτικής αναπαράστασης του γράφου γίνεται μέσω της κλάσης ImageVisualization. Ο κατασκευαστής της κλάσης αυτής καλεί μία μέθοδο, η οποία αρχικοποιεί έναν JGraphXAdapter της βιβλιοθήκης JGraphT και έναν mxCircleLayout, ώστε το αποτέλεσμα να είναι οι κόμβοι να διατάσσονται σε κύκλο. Το αποτέλεσμα είναι η δημιουργία μίας εικόνας σε μορφή PNG, η οποία μπορεί να αποθηκευτεί σε αρχείο.

Βήμα 7

Το τελευταίο βήμα της επεξεργασίας αφορά στον υπολογισμό του PageRank για κάθε κόμβο του γράφου. Με σκοπό να τηρείται η βαθμολογία του PageRank με μεγάλη ακρίβεια, επιλέχθηκε η χρήση μεταβλητής τύπου double για την καταχώρηση του αποτελέσματος του αλγόριθμου. Τα αποτελέσματα του PageRank ταξινομούνται με έναν απλό αλγόριθμο ταξινόμησης και εκτυπώνονται στην κονσόλα του συστήματος (Εικόνα 9). Η εκτύπωση γίνεται με φθίνουσα σειρά της βαθμολογίας που υπολόγισε ο PageRank.

```
=====
Calculating the score
=====

Sorting PageRank
1)Airbnb Id:4152752      PageRank Value: 0.06585037843593036
2)Airbnb Id:36487245     PageRank Value: 0.06540736554553091
3)Airbnb Id:22386804     PageRank Value: 0.06488184979958027
4)Airbnb Id:28736904     PageRank Value: 0.06425905566511884
5)Airbnb Id:18288828     PageRank Value: 0.06352158935191876
6)Airbnb Id:34796261     PageRank Value: 0.06264897622656979
7)Airbnb Id:27929993     PageRank Value: 0.061617116566731106
8)Airbnb Id:15777411     PageRank Value: 0.06039764524363903
9)Airbnb Id:35034543     PageRank Value: 0.05895717837761416
10)Airbnb Id:35406411    PageRank Value: 0.057256427019137225
11)Airbnb Id:15778177    PageRank Value: 0.05524915438785315
12)Airbnb Id:14183466    PageRank Value: 0.05288094906043624
13)Airbnb Id:33590374    PageRank Value: 0.05008778162597955
14)Airbnb Id:35186445    PageRank Value: 0.046794306595483615
15)Airbnb Id:9451492     PageRank Value: 0.04291186460838476
16)Airbnb Id:35034773    PageRank Value: 0.03833613204536847
17)Airbnb Id:28701957    PageRank Value: 0.032944355822951354
18)Airbnb Id:34814184    PageRank Value: 0.026592100164413393
19)Airbnb Id:15009618    PageRank Value: 0.019109419222948527
```

Εικόνα 9. Εκτύπωση των καλύτερων αποτελεσμάτων

Για πρακτικούς λόγους, η εφαρμογή υπολογίζει το χρόνο που απαιτείται για τον εκάστοτε υπολογισμό. Αυτό γίνεται μεταξύ άλλων και για την αξιολόγηση του συστήματος σε διάφορες συνθήκες λειτουργίας (πλήθος αντικειμένων ή συνδυασμούς παραμέτρων), ώστε να γίνουν βελτιώσεις στην αλγοριθμική του λειτουργία αλλά και την υλοποίηση.

Ένα συμπέρασμα που προκύπτει από την ανάλυση των χρόνων αυτών είναι ότι ο περισσότερος χρόνος δαπανάται στην οπτικοποίηση των αποτελεσμάτων και μετά στον υπολογισμό του Top-k. Οι χρόνοι για τις υπόλοιπες επεξεργασίες του προγράμματος είναι μηδαμινά, δεδομένου ότι το αρχείο των αντικειμένων Airbnb περιείχε 50.000 εγγραφές.

6.3. Ουρά προτεραιότητας (Priority Queue)

Η ουρά προτεραιότητας είναι μια επέκταση της δομής δεδομένων «ουρά» (queue) με τις ακόλουθες ιδιότητες:

- Κάθε στοιχείο έχει προτεραιότητα που σχετίζεται με αυτό.
- Ένα στοιχείο με υψηλή προτεραιότητα αποβάλλεται πριν από ένα στοιχείο με χαμηλή προτεραιότητα.
- Εάν δύο στοιχεία έχουν την ίδια προτεραιότητα, εξυπηρετούνται σύμφωνα με τη σειρά τους στην ουρά.

Σε μια ουρά προτεραιότητας, ένα στοιχείο με υψηλή προτεραιότητα εξυπηρετείται πριν από ένα στοιχείο με χαμηλή προτεραιότητα. Σε μερικές υλοποιήσεις, εάν δύο στοιχεία

έχουν την ίδια προτεραιότητα, εξυπηρετούνται σύμφωνα με τη σειρά με την οποία καταγράφηκαν, ενώ σε άλλες υλοποιήσεις η σειρά στοιχείων με την ίδια προτεραιότητα είναι απροσδιόριστη.

Η εφαρμογή χρησιμοποιεί τη βιβλιοθήκη *priority queue* της Java, την οποία υλοποιεί μέσα στην κλάση *TopK*. Στην κλάση αυτή υλοποιείται μία δομή της *priority queue*, της οποίας ο ρόλος είναι να εισάγει στη δομή αυτή τα αποτελέσματα του κάθε χρήστη ταξινομημένα.

6.4. Βιβλιοθήκες για οπτικοποίηση γράφων

Παρά τη δημοφιλία της Java ως γλώσσας προγραμματισμού ακαδημαϊκά και βιομηχανικά, ο αριθμός των πακέτων γράφων γραμμένων σε Java είναι πολύ περιορισμένος. Αυτήν τη στιγμή, υπάρχουν μόνο δύο βιώσιμες εναλλακτικές σε *JGraphT*: *Java Universal Network/Graph Framework (JUNG)* (O'Madadhain, et al., 2003) και μια ενότητα διαχείρισης γράφων ενσωματωμένη στη βιβλιοθήκη *Google Guava*. Η *JUNG* βασίζεται σε μία δομή δεδομένων γράφων, υλοποιεί βασικούς αλγόριθμους όπως συντομότερου μονοπατιού και κεντρικότητας (*centrality*), και μία ενότητα σχεδίασης γράφων. Η *Google Guava*, περιλαμβάνει έναν αριθμό δομών δεδομένων για αναπαράσταση γράφων. Τέτοιες δομές, στον ορολογία του *Guava* είναι οι “*Graph*”, “*ValueGraph*” and “*Network*”. Μεταξύ αυτών των τριών, η κλάση “*Network*” είναι η πιο γενική και έχει μεγάλες ομοιότητες με τη βιβλιοθήκη γράφων *JGraphT*. Για να παράσχει αλγοριθμική υποστήριξη στην *Guava*, η *JGraphT* περιλαμβάνει κλάσεις – προσαρμογείς, οι οποίες επιτρέπουν στους χρήστες να ενεργοποιήσουν όλους τους αλγόριθμους της *JGraphT* σε δομές δεδομένων γράφων *Guava*.

Τα πακέτα λογισμικού για ανάλυση δικτύου μπορούν να κατηγοριοποιηθούν ευρέως σε: (1) πακέτα για δομές δεδομένων και αποθήκευση, συμπεριλαμβανομένων των βάσεων δεδομένων για μεγάλης κλίμακας δίκτυα, (2) αλγοριθμικά πακέτα για ανάλυση δικτύου, προορισμένα κυρίως για λειτουργίες ανάλυσης δεδομένων από τα δεδομένα δικτύου και (3) πακέτα για οπτικοποιήσεις γράφων, τα οποία παράγουν εποπτικές, κατανοητές από τον άνθρωπο, οπτικές αναπαραστάσεις τους.

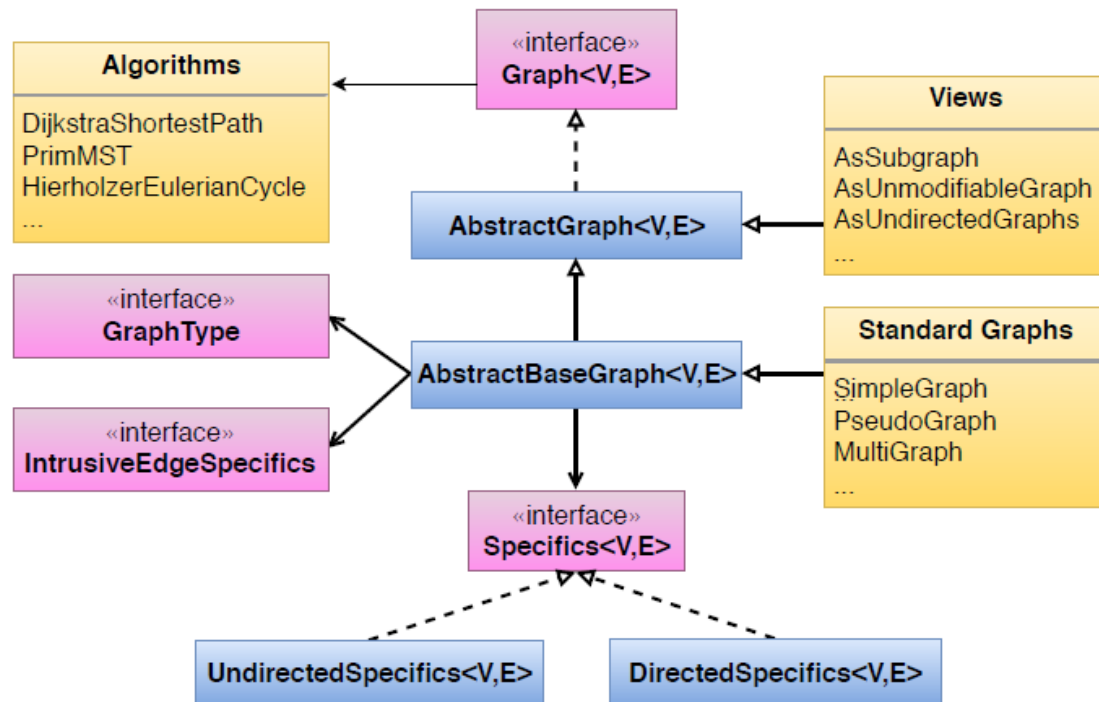
6.5. Πώς λειτουργεί η JGraphT

6.5.1. Σχεδιασμός της βιβλιοθήκης JGraphT

Η *JGraphT* είναι σχεδιασμένη με έμφαση στην ευελιξία και την απόδοση των υπολογισμών της. Στην ενότητα αυτή γίνεται αναφορά στη σχεδίαση της *JGraphT* καθώς και στους συμβιβασμούς που έγιναν στη σχεδίαση της βιβλιοθήκης.

Η *JGraphT* είναι χτισμένη γύρω από μια κεντρική διεπαφή *Graph <V, E>* (Εικόνα 10).

Αυτή η διεπαφή παρέχει βασικές λειτουργίες για την κατασκευή ενός γράφου, καθώς και τελεστές για την πρόσβαση σε στοιχεία του γράφου (Εικόνα 11). Όλες οι αλληλεπιδράσεις με τον γράφο διεκπεραιώνονται μέσω αυτής της διεπαφής: κάθε προκαθορισμένη κλάση γράφου στη βιβλιοθήκη υλοποιεί τη διεπαφή (interface), και όλοι οι αλγόριθμοι της JGraphT δέχονται σαν όρισμα ένα αντικείμενο της κλάσης Graph.



Εικόνα 10. Η σχεδίαση του πυρήνα της βιβλιοθήκης JGraphT
Πηγή: (Michail, et al., 2020)

```
public interface Graph<V,E> {  
    GraphType getType();  
  
    V addVertex();  
    boolean removeVertex(V v);  
    E addEdge(V sourceVertex, V targetVertex);  
    boolean removeEdge(E e);  
  
    Set<V> vertexSet();  
    Set<E> edgeSet();  
  
    V getEdgeSource(E e);  
    V getEdgeTarget(E e);  
    E getEdge(V sourceVertex, V targetVertex);  
  
    double getEdgeWeight(E e);  
    void setEdgeWeight(E e, double weight);  
  
    Set<E> edgesOf(V v);  
  
    /* More methods omitted */  
}
```

Εικόνα 11. Η διεπαφή (interface) $Graph<V,E>$. Όλες οι δοσοληψίες με τους γράφους διεκπεραιώνονται με τη χρήση αυτής της διεπαφής
Πηγή: (Michail, et al., 2020)

Η διεπαφή παίρνει δύο γενικές παραμέτρους $<V>$ και $<E>$ που καθορίζουν τον τύπο των αντικειμένων που χρησιμοποιούνται αντίστοιχα σαν κορυφές και ακμές του γράφου. Ο JGraphT επιτρέπει στο χρήστη να χρησιμοποιήσει οποιονδήποτε τύπο αντικειμένου ως ακμή ή κορυφή. Στην απλούστερη μορφή τους, οι κορυφές ενός γράφου αναπαριστώνται από ακέραιους αριθμούς (integers) ή συμβολοσειρές (Strings), ενώ οι ακμές αναπαριστώνται από μια κλάση, με το όνομα “DefaultEdge”. Ένα χαρακτηριστικό παράδειγμα είναι η μοντελοποίηση ενός δρόμου σε μορφή γράφου, όπου οι κορυφές αναπαριστούν διασταυρώσεις και οι ακμές αποτελούν τα τμήματα δρόμου. Τυπικά, μία υλοποίηση θα μπορούσε να χρησιμοποιεί μία κλάση “Intersection”, η οποία θα αποθήκευε τις γεωγραφικές συντεταγμένες της αντίστοιχης διασταύρωσης, καθώς και μια κλάση “RoadSegment”, η οποία καταγράφει πληροφορίες όπως ο αριθμός των λωρίδων του δρόμου, η ταχύτητα οδήγησης, το μήκος, το σχήμα και ίσως το όνομα του τμήματος. Η δυνατότητα που δίνει η βιβλιοθήκη JGraphT για τη χρήση ενός αντικειμένου ως κορυφή ή ακμή, κάνει τη βιβλιοθήκη JGraphT εύκολη στην προσαρμογή στην εκάστοτε εφαρμογή, ενώ

ταυτόχρονα οι βασικές δομές δεδομένων που υποστηρίζει μπορούν να εκφράσουν κάθε είδος σχέσης ή αλληλεπίδρασης μεταξύ οποιουδήποτε τύπου αντικειμένου χρησιμοποιείται για τη μοντελοποίηση του γράφου.

Πίνακας 6. Οι διαθέσιμες κλάσεις για υλοποίηση γράφων

Class Name	Edges	Self-loops	Multiple edges	Weighted
SimpleGraph	undirected	no	no	no
Multigraph	undirected	no	yes	no
Pseudograph	undirected	yes	yes	no
DefaultUndirectedGraph	undirected	yes	no	no
SimpleWeightedGraph	undirected	no	no	yes
WeightedMultigraph	undirected	no	yes	yes
WeightedPseudograph	undirected	yes	yes	yes
DefaultUndirectedWeightedGraph	undirected	yes	no	yes
SimpleDirectedGraph	directed	no	no	no
DirectedMultigraph	directed	no	yes	no
DirectedPseudograph	directed	yes	yes	no
DefaultDirectedGraph	directed	yes	no	no
SimpleDirectedWeightedGraph	directed	no	no	yes
DirectedWeightedMultigraph	directed	no	yes	yes
DirectedWeightedPseudograph	directed	yes	yes	yes
DefaultDirectedWeightedGraph	directed	yes	no	yes

Η JGraphT παρέχει υλοποιήσεις κοινών τύπων γράφων, όπως απλοί γράφοι, σύνθετοι γράφοι, ψευδογράφοι κ.λπ. Κάθε ένας από αυτούς τους τύπους γράφων μπορεί να είναι κατευθυνόμενος ή μη κατευθυνόμενος, να περιλαμβάνει βάρη ή όχι. Ο Πίνακας 6 περιλαμβάνει επισκόπηση διαφορετικών τύπων γράφων που υποστηρίζονται από τη JGraphT. Δεδομένου ότι κάθε γράφος υλοποιεί τη διεπαφή “Graph”, οι μέθοδοι της κλάσης που προσδιορίζονται από τη διεπαφή συμπεριφέρονται διαφορετικά ανάλογα με τον τύπο του γράφου. Η μέθοδος degreeOf (V vertex), για παράδειγμα, στην περίπτωση που ο γράφος είναι μη κατευθυνόμενος επιστρέφει το πλήθος των ακμών που καταλήγουν σε έναν κόμβο (σε περίπτωση που μία ακμή ξεκινά και καταλήγει στον ίδιο κόμβο, υπολογίζεται δύο φορές). Όμοια, η μέθοδος inDegreeOf (V vertex) όταν εφαρμόζεται σε έναν κατευθυνόμενο γράφο επιστρέφει τον αριθμό των κατευθυνόμενων ακμών οι οποίες ξεκινούν από έναν κόμβο, ενώ στην περίπτωση μη κατευθυνόμενων γράφων επιστρέφει το πλήθος των ακμών που καταλήγουν σε έναν κόμβο.

Για να δημιουργηθεί ένα αντικείμενο μιας κλάσης απλού γράφου, μπορεί να

χρησιμοποιηθεί ο παρακάτω κώδικας:

```
Graph<Integer, DefaultEdge> graph = new SimpleGraph <> (DefaultEdge.class);
```

Η υλοποίηση ενός γράφου είναι μια διαδικασία αρκετά σύνθετη, ειδικά για χρήστες οι οποίοι δεν έχουν το θεωρητικό υπόβαθρο της θεωρίας γράφων. Η JGraphT υποστηρίζει για το λόγο αυτό μια απλοποιημένη και γενική υλοποίηση. Για παράδειγμα, ένας ψευδογράφος ο οποίος έχει πολλαπλές ακμές και κύκλους μπορεί να χρησιμοποιηθεί για την αναπαράσταση ενός απλού γράφου που δεν υποστηρίζει αυτά τα χαρακτηριστικά. Μια τέτοια τεχνική βέβαια έχει το μειονέκτημα της μειωμένης απόδοσης, δεδομένου ότι ψευδογράφοι τυπικά καταλαμβάνουν περισσότερο χώρο, και οι μέθοδοι αυτών των γράφων λειτουργούν πιο αργά σε σχέση με τις πιο ειδικές μορφές γράφων. Για να διευθετηθεί το θέμα αυτό, και για να απλοποιηθεί η διαδικασία επιλογής του κατάλληλου τύπου γράφου, η JGraphT επιτρέπει στο χρήστη να δημιουργήσει γράφους μέσα από ένα μοντέλο “builder” (Gamma, et al., 1993). Αξιοποιώντας αυτό το μοντέλο, η βιβλιοθήκη προσδιορίζει την πιο κατάλληλη υλοποίηση για τον εκάστοτε γράφο.

Η βιβλιοθήκη GraphType επιτρέπει την ερώτηση ενός γράφου, ώστε να αποκαλύψει τα χαρακτηριστικά του. Η GraphType περιλαμβάνει τους απαραίτητους τύπους πληροφοριών, ορίζοντας αν ο γράφος είναι κατευθυνόμενος ή μη κατευθυνόμενος, ζυγισμένος ή μη ζυγισμένος, και αν επιτρέπει αυτοαναφορικές ακμές (self-loops), πολλαπλές ακμές, κ.λπ.

6.5.2. Δομή του γράφου

Η δομική σχεδίαση του JGraphT, όπως απεικονίζεται στην Εικόνα 10, διαχωρίζει τη διεπαφή `Graph<V,E>`, από τις δομές δεδομένων που χρησιμοποιούνται για την αποθήκευση του γράφου. Η διεπαφή `Graph`, τοποθετείται στην κορυφή της ιεραρχίας κλάσεων, και ορίζει όλες τις υψηλού επιπέδου λειτουργίες του γράφου. Η κλάση `AbstractGraph` προσφέρει μια ελάχιστη υλοποίηση της διεπαφής `Graph`, χωρίς να ορίζει ρητά τις δομές δεδομένων για αποθήκευση και δεικτοδότηση, δεδομένου ότι αυτές τις λειτουργίες τις διαχειρίζεται ο εκάστοτε γράφος. Ο εκάστοτε γράφος επεκτείνει την κλάση `AbstractGraph`, καταγράφει το `GraphType`, και υλοποιεί δομές δεδομένων ώστε να είναι δυνατή η αποθήκευσή του. Η Εικόνα 11 απεικονίζει το προεπιλεγμένο backend, το οποίο έχει υλοποιηθεί από την κλάση `AbstractBaseGraph`. Οι περισσότερες από τις προεπιλεγμένες κλάσεις γράφων που αναφέρονται στον Πίνακα 6 είναι υπο-κλάσεις της κλάσης `AbstractBaseGraph`.

Για να αναπτυχθεί ένας νέος τύπος γράφου, ή για να τροποποιηθεί μία υποκείμενη υλοποίηση ενός υπάρχοντος γράφου, ο προγραμματιστής τυπικά εφαρμόζει λειτουργίες αρχικοποίησης (`instantiate`), επικάλυψης (`override`) ή επέκτασης (`extend`) κάποιων κλάσεων που απεικονίζονται στην Εικόνα 10. Για παράδειγμα, οι Views των

γράφων μπορούν να οριστούν επεκτείνοντας την κλάση `AbstractGraph`. Όλες οι λειτουργίες που ενεργοποιούνται σε μια `view` ανατίθενται στον γράφο που δημιουργεί την προβολή. Συνεπώς, οι `views` προσφέρουν ένα φυσικό τρόπο για να μοντελοποιούμε υπογράφους (με τη χρήση της κλάσης `AsSubgraph`). Μπορούν ακόμη να χρησιμοποιηθούν για μεταχειριστούν έναν κατευθυνόμενο γράφο σαν μη κατευθυνόμενο (`AsUndirectedGraph`), για να προσθέσουν βάρη σε έναν μη σταθμισμένο γράφο (`AsWeightedGraph`) ή για να καταστήσουν έναν γράφο μη τροποποιήσιμο (`AsUnModifiableGraph`). Εκτός από τις προβολές, είναι δυνατό να ορίσουμε κλάσεις προσαρμογών (`adapter classes`) επεκτείνοντας την κλάση `AbstractGraph`. Μέσω αυτών των προσαρμογών ένας χρήστης μπορεί να ενεργοποιήσει οποιονδήποτε αλγόριθμο επεξεργασίας γράφων σε άλλες δομές αναπαράστασης γράφων.

6.5.3. Υποκείμενη υλοποίηση

Η υποκείμενη υλοποίηση και αποθήκευση δεδομένων ενός γράφου, ανεξάρτητα από το αν ο τύπος του γράφου είναι προκαθορισμένος ή έχει οριστεί από τον χρήστη, μπορεί να προσαρμοστεί στις ανάγκες του προγραμματιστή. Υπάρχουν πολλές εφαρμογές γράφων, στις οποίες απαιτούνται εξειδικευμένες γνώσεις του χρήστη, ώστε να γίνει η καλύτερη επιλογή των απαραίτητων δομών δεδομένων. Συναφείς γνώσεις οι οποίες μετουσιώνονται σε αντίστοιχες υλοποιήσεις είναι: ο τύπος δεδομένων που αναπαριστώνται, η πυκνότητα του γράφου (αραιός (`sparse`) ή πυκνός (`dense`) γράφος), το μέγεθος του γράφου (πλήθος ακμών / κορυφών), ο διαθέσιμος χώρος αποθήκευσης, οι απαιτήσεις απόδοσης των λειτουργιών του γράφου, και το είδος των λειτουργιών που χρησιμοποιείται πιο συχνά. Παρόμοιες αποφάσεις λαμβάνονται για τη δημιουργία ενός πίνακα γειτνίασης (`adjacency matrix`) ώστε να γίνεται η αναζήτηση γειτονικών κορυφών ή όταν επιλέγουμε τις δομές που χρησιμοποιούνται για να αναπαραστήσουν έναν πίνακα επίπτωσης (`incidence matrix`). Αν για παράδειγμα οι κορυφές περιλαμβάνουν ακέραιους αριθμούς και μόνο, ο πίνακας επίπτωσης μπορεί να αναπαρασταθεί με έναν πίνακα (`array`) δύο διαστάσεων, ενώ στην περίπτωση που κάθε κορυφή περιέχει αντικείμενα κάποιας κλάσης τότε η λύση είναι η δημιουργία ενός πίνακα κατακερματισμού.

Ο λεπτομερής έλεγχος της μεθόδου αποθήκευσης των δεδομένων ενός γράφου γίνεται προσαρμόζοντας τον backend γράφο. Ο `JGraphT` παρέχει δύο προκαθορισμένα backend που καλύπτουν τις πιο κοινές περιπτώσεις χρήσης: το default backend και το αραιό (`sparse`) backend. Επιπρόσθετα, ο χρήστης μπορεί να υλοποιήσει προσαρμοσμένα backend, απλά δημιουργώντας μια νέα υπο-κλάση της κλάσης `AbstractGraph`. Όταν χρησιμοποιούμε τα προκαθορισμένα backend, όλες οι λειτουργίες που καλούνται στους γράφους της `JGraphT` εκτελούνται με ντετερμινιστικό τρόπο. Αυτή η συμπεριφορά γίνεται αντιληπτή από τη χρήση των δομών δεδομένων που έχουν μια προβλέψιμη σειρά επανάληψης όπως οι λίστες,

καθώς και σαν sets και maps, δομές που μπορούν να υλοποιηθούν με την εφαρμογή διπλά συνδεδεμένων λιστών (κλάσεις `LinkedHashSet` και `LinkedHashMap`).

Το default backend, το οποίο υλοποιείται στην κλάση `AbstractBaseGraph` (Εικόνα 11), είναι σχεδιασμένο για να προσφέρει ισορροπία μεταξύ απόδοσης και κατανάλωσης μνήμης. Εφόσον οι κορυφές και οι ακμές μπορούν να μοντελοποιηθούν από αντικείμενα, το προκαθορισμένο backend βασίζεται κατά κύριο λόγο στους πίνακες κατακερματισμού (hash tables) για την αποθήκευση των κορυφών και των ακμών, και για την υλοποίηση λιστών γειτνίασης. Συνεπώς, βασικές λειτουργίες όπως η διαγραφή ή προσθήκη κορυφών ή ακμών μπορούν να εκτελεστούν σε προκαθορισμένο σταθερό χρόνο ($O(1)$). Παρέχεται επίσης ένας (προαιρετικός) μηχανισμός ευρετηρίασης, ο οποίος δεικτοδοτεί τα δύο άκρα μιας ακμής για να επιτρέψει γρήγορη εύρεση των κορυφών. Αυτός ο μηχανισμός ευρετηρίασης παρέχει ισορροπία μεταξύ απόδοσης και κατανάλωσης μνήμης. Ο προγραμματιστής μπορεί να προσαρμόσει περαιτέρω τον τρόπο που θα υλοποιηθούν οι πίνακες γειτνίασης και επίπτωσης και πώς υλοποιούνται οι αναζητήσεις ακμών (μαζί με τον τρόπο που αποθηκεύονται τα βάρη των ακμών), χρησιμοποιώντας τις διεπαφές `Specifics` και `IntrusiveEdgesSpecifics`. Για παράδειγμα, οι πίνακες κατακερματισμού της Java (Java hash tables) οι οποίοι μπορούν να χρησιμοποιηθούν για την αποθήκευση των πινάκων γειτνίασης και επίπτωσης, είναι δυνατό να αντικατασταθούν από εξειδικευμένες κλάσεις οι οποίες περιέχονται σε βιβλιοθήκες όπως η `Fastutil` (`fastutil`, 2020) ή η `Eclipse Collections` (`Eclipse Collections`, 2018), ώστε να περιοριστεί η μνήμη που απαιτείται για την αποθήκευση ενός γράφου.

Το προεπιλεγμένο backend είναι συμβατό με γενικού σκοπού γράφους και υποστηρίζει αποτελεσματικά λειτουργίες επεξεργασίας όπως είναι η προσθήκη ή διαγραφή μιας κορυφής ή ακμής. Καλύτερη απόδοση όμως επιτυγχάνεται με μια λιγότερο δυναμική υλοποίηση, όπου ο γράφος κατασκευάζεται με μια μοναδική μαζική λειτουργία δημιουργίας κόμβων και ακμών.

Μια τέτοια υλοποίηση είναι γνωστή με τον όρο `write-once read-many` και είναι συνηθισμένη όταν εκτελούμε σύνθετους αλγόριθμους σε γράφους, που συνήθως περιλαμβάνουν τη φόρτωση ενός γράφου στη μνήμη από μια εξωτερική μονάδα αποθήκευσης.

Για να αντιμετωπίζει αποτελεσματικά μια τέτοια περίπτωση χρήσης, η βιβλιοθήκη `JGraphT` παρέχει ένα εξειδικευμένο `sparse backend` γράφο, ο οποίος υλοποιεί την κλάση `Graph <Integer, Integer>` και σύμφωνα με τον ορισμό της απαιτεί όλες οι κορυφές και ακμές να αναπαριστώνται από ακεραίους. Στην περίπτωση που μελετούμε εδώ, θεωρούμε ότι κορυφές αριθμούνται σειριακά με ακεραίους αριθμούς $0, \dots, n-1$ ενώ οι ακμές αριθμούνται $0, \dots, m-1$ όπου n και m αντιπροσωπεύουν τον αριθμό κορυφών και ακμών στο γράφο. Η τελευταία υπόθεση καθιστά τις λειτουργίες επεξεργασίας λιγότερο αποτελεσματικές, εφόσον η προσθήκη ή μετακίνηση μιας

κορυφής (αντίστοιχα και ακμής) πιθανόν να προκαλεί την επαναρίθμηση όλων των άλλων κορυφών και ακμών. Για να περιοριστούν οι απαιτήσεις αποθηκευτικού χώρου ενός γράφου, το sparse backend αποθηκεύει τον incidence matrix σε μορφή Compressed-Sparse-Rows (CSR) (Saad, 2003), εκμεταλλευόμενο το γεγονός πως οι περισσότεροι γράφοι στον πραγματικό κόσμο είναι αραιοί. Οι γράφοι που αποθηκεύονται με αυτή τη δομή υποστηρίζουν και αυτοαναφορικές ακμές (self-loops) και πολλαπλές ακμές. Μία επισκόπηση των αραιών γράφων και των ιδιοτήτων τους υπάρχει στον Πίνακα 6. Για να υλοποιήσουμε μη κατευθυνόμενους γράφους, το sparse backend αναπαριστά τον incidence matrix μέσω ενός πίνακα boolean τιμών, ενώ στην περίπτωση των κατευθυνόμενων γράφων χρησιμοποιούνται δύο πίνακες boolean τιμών (μια για κάθε κατεύθυνση). Τα βάρη των ακμών, σε περιπτώσεις σταθμισμένων γράφων, αποθηκεύονται σε έναν απλό πίνακα μήκους m .

6.5.4. Αλγόριθμοι

Η βιβλιοθήκη JGraphT υλοποιεί πλήθος αλγορίθμων, όπως αυτοί περιγράφονται στη θεωρία των γράφων. Οι αλγόριθμοι αυτοί είναι:

- Connectivity
- LCA (Least Common Ancestor)
- Cycles
- Shortest paths
- Node Centrality
- Spanning Trees and Spanners
- Recognizing Graphs
- Matchings
- Cuts and Flows
- Isomorphism
- Coloring
- Cliques
- Vertex Cover
- Tours

6.6. Υλοποίηση με τη βιβλιοθήκη JGraphT

Η βιβλιοθήκη JGraphT της Java προσφέρει τις κλάσεις οι οποίες διευκολύνουν το χειρισμό γράφων από μία εφαρμογή γραμμένη στη γλώσσα Java. Η βιβλιοθήκη JGraphT είναι περιέχει μια δομή, η οποία προσομοιώνει έναν γράφο ως ένα σύνολο κορυφών που συνδέονται με ένα σύνολο ακμών.

Τα αντικείμενα (objects) των κορυφών είναι στιγμιότυπα της κλάσης `java.net.URI`. Η JGraphT δεν παρέχει μόνο μια κλάση κορυφών αλλά δίνει την δυνατότητα στον προγραμματιστή να επιλέξει τις κλάσεις που χρειάζεται για να αναπαραστήσει κορυφές

και ακμές ανάλογα με τις προδιαγραφές και κάποιους μικρούς περιορισμούς.

Η JGraphT δίνει επίσης την δυνατότητα στον προγραμματιστή να επιλέξει την επιθυμητή κλάση για αναπαράσταση των ακμών. Εάν δεν υπάρχει ανάγκη για κάποια ειδική κλάση αναπαράστασης ακμών, μπορεί να χρησιμοποιηθεί η κλάση DefaultEdge που παρέχεται από την βιβλιοθήκη JGraphT.

Στην εφαρμογή χρησιμοποιήθηκε σαν δομή του γράφου η κλάση DefaultDirectedGraph, λόγω του ότι δίνει δυνατότητες όπως:

- Υποστήριξη κατευθυνόμενων γράφων.
- Δημιουργία κύκλων στον γράφο.
- Αποτροπή δημιουργίας ακμών με αρχή και προορισμό τον ίδιο κόμβο.
- Υποστήριξη ακμών χωρίς βάρη.

Οι ιδιότητες αυτές οδήγησαν στην επιλογή της συγκεκριμένης κλάσης, η οποία δεν απαιτεί επιπλέον προγραμματισμό και την υλοποίηση πρόσθετων κλάσεων.

Η υλοποίηση του αλγόριθμου της διπλωματικής εργασίας θα μπορούσε να γίνει από κάποιον τρίτο με τη χρήση μιας διαφορετικής κλάσης της βιβλιοθήκης JGraphT. Στη συγκεκριμένη υλοποίηση επιλέχθηκε η κλάση DefaultDirectedGraph, η οποία προσφέρει έναν απλό, εύκολο και γρήγορο τρόπο δημιουργίας ενός κατευθυνόμενου γράφου, ο οποίος μπορεί στη συνέχεια να οπτικοποιηθεί με τη χρήση της ίδιας βιβλιοθήκης.

Ο κατασκευαστής (constructor) του γράφου δέχεται ως παράμετρο την κλάση ακμής, έτσι ώστε να μπορεί να δημιουργήσει νέα αντικείμενα ακμών κάθε φορά που η μέθοδος addEdge καλείται να συνδέσει 2 κορυφές.

Ο αλγόριθμος εισάγει επαναληπτικά στο γράφο τα αποτελέσματα που προκύπτουν από την εφαρμογή των προτιμήσεων των χρηστών, ώστε να προκύψουν τα Top-k αποτελέσματα από την ουρά προτεραιότητας που δημιουργήθηκε με τα πρωταρχικά αποτελέσματα της αναζήτησης. Οι γράφοι αυτοί έχουν τη μορφή αλυσίδας (1 αλυσίδα ανά 1 κριτήριο προτίμησης) και προκύπτουν από τα Top-k αποτελέσματα. Οι αλυσίδες αυτές στη συνέχεια συγχωνεύονται στον γράφο (Σχήμα 12), κόμβοι του οποίου είναι τα ενοικιαζόμενα Airbnb και κάθε ακμή που φεύγει από έναν κόμβο δείχνει στον κόμβο ο οποίος είναι καλύτερος από αυτόν, βάσει του κριτηρίου.

Μόλις ολοκληρωθεί η διαδικασία και ο γράφος ολοκληρωθεί με τους κόμβους και τις ακμές, ξεκινά η φάση της οπτικοποίησης, ώστε να είναι δυνατό να εξαχθούν συμπεράσματα με διασθητικό τρόπο. Για να υλοποιηθεί αυτό, χρησιμοποιούμε έναν προσαρμογέα (adapter), ο οποίος κάνει τον γράφο συμβατό με τη βιβλιοθήκη JGraphT, ώστε να παραχθεί η γραφική αναπαράσταση του γράφου.

Η τελική φάση περιλαμβάνει την εφαρμογή του αλγόριθμου PageRank στον γράφο. Βάσει του πλήθους εισερχόμενων και εξερχόμενων ακμών, υπολογίζεται ο PageRank. Αποτέλεσμα του αλγόριθμου είναι μια λίστα σε φθίνουσα ταξινόμηση, η οποία ξεκινά από τους καλύτερους κόμβους, προς τους όχι και τόσο κάλους, σύμφωνα με την βαθμολογία για τον κάθε ένα ξεχωριστά.

6.7. Περιορισμοί και Λεπτομέρειες

Προκειμένου η εφαρμογή να μπορεί να λειτουργήσει σε κάθε υπολογιστή και στο μέλλον και ίσως και σε φορητές συσκευές (κινητά τηλέφωνα ή tablets) και χωρίς την ανάγκη σύνδεσης στο internet, επιλέχθηκε να μη χρησιμοποιηθεί η λύση του Maven, η οποία κατεβάζει αυτόματα τις βιβλιοθήκες που είναι απαραίτητες για τη λειτουργία του προγράμματος. Οι αναγκαίες βιβλιοθήκες διατίθενται μαζί με την εφαρμογή και είναι διαθέσιμες σε συγκεκριμένο φάκελο της εφαρμογής, ώστε να μην είναι απαραίτητο το κατέβασμά τους κάθε φορά που εγκαθίσταται η εφαρμογή σε μία συσκευή.

Λόγω των αρκετών υπολογιστικών πόρων που είναι απαραίτητοι για τους υπολογισμούς των γράφων και του PageRank, έχει τεθεί περιορισμός στο μέγεθος των γράφων που παράγονται. Συνεπώς οι γράφοι των αποτελεσμάτων δε μπορούν να έχουν περισσότερους από 4 κόμβους. Θα ήταν δυνατό να επιτρέπεται στην εφαρμογή να παράγει ακόμα μεγαλύτερους γράφους, αλλά επειδή αυτό απαιτεί αρκετούς πόρους για την επεξεργασία και τη δημιουργία της εικόνας, αποφασίστηκε να τεθεί ο περιορισμός. Επιπρόσθετα, ο περιορισμός δρα προς όφελος της δυνατότητας η εφαρμογή να εκτελείται και σε συσκευές με χαμηλότερες υπολογιστικές δυνατότητες, χωρίς να χάνεται κάποια λειτουργικότητα.

Η εφαρμογή δέχεται σαν είσοδο δύο διαφορετικά αρχεία μορφής CSV. Το πρώτο αρχείο περιλαμβάνει το dataset των ενοικιαζόμενων διαμερισμάτων Airbnb και το δεύτερο περιέχει τις προτιμήσεις από κάποιους υποτιθέμενους σημαντικούς χρήστες. Ο τρόπος με τον οποίο έχει υλοποιηθεί η εφαρμογή, επιτρέπει την εύκολη ολοκλήρωσή της με άλλες πηγές δεδομένων, όπως για παράδειγμα από ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων ή από αρχεία μορφής JSON. Μία τέτοια δυνατότητα θα προσφέρει την πιο εύκολη ενημέρωση των σημείων ενδιαφέροντος (ενδεικτικά Airbnb, εστιατόρια, ξενοδοχεία, κ.ο.κ.).

7. Αποτελέσματα και Οπτική Ανάλυση Δεδομένων

Η φύση των δεδομένων ορισμένες φορές επιβάλλει την οπτική τους αναπαράσταση, ώστε αυτά να αναλυθούν με αποδοτικό τρόπο. Επίσης μέσα από την οπτικοποίηση προκύπτουν ευκολίες όπως ο γεωγραφικός εντοπισμός ενός σημείου, αλλά και η αναγνώριση των δομών τους όταν προκύπτει ένα μεγάλο σύνολο δεδομένων.

Για τους λόγους αυτούς κρίθηκε απαραίτητο η διπλωματική αυτή εργασία να υλοποιήσει μηχανισμούς οπτικοποίησης των αποτελεσμάτων των επερωτήσεων των χρηστών, ώστε αυτά να είναι πιο κατανοητά και ευκολότερο να προσδιοριστούν.

7.1. Αποτελέσματα

Στο κεφάλαιο αυτό παρατίθενται μετρήσεις περιπτώσεων εφαρμογής, σύμφωνα με τις παραμέτρους που αναφέρονται πριν την κάθε εκτέλεση.

7.1.1. Εκτέλεση εφαρμογής (1)

Εκτέλεση χωρίς άλλους σημαντικούς χρήστες με τις εξής προτιμήσεις (βάρη):

- price: 2
- reviews: 8
- distance: 5 x: 40.76404 y: -73.98933

```
run:
Reads a file
Give me how much do you care about price:
Enter number from 0 to 10: 2
Give me how much do you care about reviews:
Enter number from 0 to 10: 8
Give me how much do you care about distance:
Enter number from 0 to 10: 5
Enter your location: Give me x axe
40.76404
Give me y axe
-73.98933
Give me how much chain results you want(20 is recommended)
more than that may break the program or take way to many resources and time
20

=====
Calculating TopK and making the graph
Finished TopK and Graph
=====

Graph size:20
Making the Image
The image is ready!!!

=====
Making the pagerank
=====

Calculating the score
=====

Sorting PageRank
1)Airbnb Id:4152752      PageRank Value: 0.06585037843593036
2)Airbnb Id:36487245     PageRank Value: 0.06540736554553091
3)Airbnb Id:22386804     PageRank Value: 0.06488184979958027
4)Airbnb Id:28736904     PageRank Value: 0.06425905566511884
5)Airbnb Id:18288828     PageRank Value: 0.06352158935191876
6)Airbnb Id:34796261     PageRank Value: 0.06264897622656979
7)Airbnb Id:27929993     PageRank Value: 0.061617116566731106
8)Airbnb Id:15777411     PageRank Value: 0.06039764524363903
9)Airbnb Id:35034543     PageRank Value: 0.05895717837761416
10)Airbnb Id:35406411    PageRank Value: 0.057256427019137225
11)Airbnb Id:15778177    PageRank Value: 0.05524915438785315
12)Airbnb Id:14183466    PageRank Value: 0.05288094906043624
13)Airbnb Id:33590374    PageRank Value: 0.05008778162597955
14)Airbnb Id:35186445    PageRank Value: 0.046794306595483615
15)Airbnb Id:9451492     PageRank Value: 0.04291186460838476
16)Airbnb Id:35034773    PageRank Value: 0.03833613204536847
17)Airbnb Id:28701957    PageRank Value: 0.032944355822951354
18)Airbnb Id:34814184    PageRank Value: 0.026592100164413393
19)Airbnb Id:15009618    PageRank Value: 0.019109419222948527

-
20)Airbnb Id:28306661    PageRank Value: 0.01029635423441042

=====

Time it took to read the hotels's file and find the max/min: 0.0 seconds
Time it took to read the preferences's file: 0.0 seconds
Time it took to make the graph and topk: 0.0 seconds
Time it took to make the image: 0.0 seconds
Time it took to make the PageRank: 0.0 seconds
Time it took to sort the PageRank: 0.0 seconds
BUILD SUCCESSFUL (total time: 39 seconds)
```

7.1.2. Εκτέλεση εφαρμογής (2)

Εκτέλεση χωρίς άλλους σημαντικούς χρήστες με τις εξής προτιμήσεις (βάρη):

- price: 8
- reviews: 2

- distance: 1 x: 40.76404 y: -73.98933

```
run:
Reads a file
Give me how much do you care about price:
Enter number from 0 to 10: 8
Give me how much do you care about reviews:
Enter number from 0 to 10: 2
Give me how much do you care about distance:
Enter number from 0 to 10: 1
Enter your location: Give me x axe
40.76404
Give me y axe
-73.98933
Give me how much chain results you want(20 is recommended)
more than that may break the program or take way to many resources and time
20

=====
Calculating TopK and making the graph
Finished TopK and Graph
=====
Graph size:20
Making the Image
The image is ready!!!
=====
Making the pagerank
=====
Calculating the score
=====
Sorting PageRank
1)Airbnb Id:4152752      PageRank Value: 0.06585037843593036
2)Airbnb Id:28736904    PageRank Value: 0.06540736554553091
3)Airbnb Id:36487245    PageRank Value: 0.06488184979958027
4)Airbnb Id:21701058    PageRank Value: 0.06425905566511884
5)Airbnb Id:18288828    PageRank Value: 0.06352158935191876
6)Airbnb Id:22386804    PageRank Value: 0.06264897622656979
7)Airbnb Id:27929993    PageRank Value: 0.061617116566731106
8)Airbnb Id:34796261    PageRank Value: 0.06039764524363903
9)Airbnb Id:15777411    PageRank Value: 0.05895717837761416
10)Airbnb Id:9451492    PageRank Value: 0.057256427019137225
11)Airbnb Id:15778177   PageRank Value: 0.05524915438785315
12)Airbnb Id:35034543   PageRank Value: 0.05288094906043624
13)Airbnb Id:33590374   PageRank Value: 0.05008778162597955
14)Airbnb Id:35406411   PageRank Value: 0.046794306595483615
15)Airbnb Id:14183466   PageRank Value: 0.04291186460838476
16)Airbnb Id:35186445   PageRank Value: 0.03833613204536847
17)Airbnb Id:2359340    PageRank Value: 0.032944355822951354
18)Airbnb Id:35034773   PageRank Value: 0.026592100164413393
19)Airbnb Id:15009618   PageRank Value: 0.019109419222948527

20)Airbnb Id:28701957   PageRank Value: 0.01029635423441042
=====
Time it took to read the hotels's file and find the max/min: 0.0 seconds
Time it took to read the preferences's file: 0.0 seconds
Time it took to make the graph and topk: 0.0 seconds
Time it took to make the image: 0.0 seconds
Time it took to make the PageRank: 0.0 seconds
Time it took to sort the PageRank: 0.0 seconds
BUILD SUCCESSFUL (total time: 35 seconds)
```

7.1.3. Εκτέλεση εφαρμογής (3)

Εκτέλεση με άλλους σημαντικούς χρήστες με τις εξής προτιμήσεις (βάρη):

- price: 8
- reviews: 6

- distance: 4 x: 40.76404 y: -73.98933

```
run:
Reads a file
Give me how much do you care about price:
Enter number from 0 to 10: 8
Give me how much do you care about reviews:
Enter number from 0 to 10: 6
Give me how much do you care about distance:
Enter number from 0 to 10: 4
Enter your location: Give me x axe
40.76404
Give me y axe
-73.98933
Give me how much chain results you want(20 is recommended)
more than that may break the program or take way to many resources and time
3
=====
Calculating TopK and making the graph
Finished TopK and Graph
=====
Graph size:249
Making the Image
The image is ready!!!
=====
Making the pagerank
=====
Calculating the score
=====
Sorting PageRank
1)Airbnb Id:33397385      PageRank Value: 0.009419933126967619
2)Airbnb Id:13894339      PageRank Value: 0.008569803330416667
3)Airbnb Id:31340283      PageRank Value: 0.008294395325602659
4)Airbnb Id:1066769       PageRank Value: 0.008093394985463251
5)Airbnb Id:3799598       PageRank Value: 0.00802196648561497
6)Airbnb Id:12999167      PageRank Value: 0.007731752679787528
7)Airbnb Id:17524441      PageRank Value: 0.007258785528413097
8)Airbnb Id:5270589       PageRank Value: 0.0072569868816288845
9)Airbnb Id:14068646      PageRank Value: 0.007256986881628884
10)Airbnb Id:36001911     PageRank Value: 0.006944868508374594
11)Airbnb Id:35252509     PageRank Value: 0.006766272941334789
12)Airbnb Id:525120       PageRank Value: 0.006680424464278582
13)Airbnb Id:19745952     PageRank Value: 0.006680424464278582
14)Airbnb Id:19226897     PageRank Value: 0.006538424525713302
15)Airbnb Id:9334086      PageRank Value: 0.006011263331814162
16)Airbnb Id:24879207     PageRank Value: 0.005761553659553612
17)Airbnb Id:35199464     PageRank Value: 0.005761553659553612
18)Airbnb Id:20953661     PageRank Value: 0.0055603807345436925
19)Airbnb Id:2440173      PageRank Value: 0.005453456492547007
20)Airbnb Id:32186503     PageRank Value: 0.005453456492547007

241)Airbnb Id:11596767    PageRank Value: 0.0021218037191972575
242)Airbnb Id:15447090    PageRank Value: 0.0021218037191972575
243)Airbnb Id:11558167    PageRank Value: 0.0021218037191972575
244)Airbnb Id:8664721     PageRank Value: 0.0021218037191972575
245)Airbnb Id:424767      PageRank Value: 0.0021218037191972575
246)Airbnb Id:6141401     PageRank Value: 0.0021218037191972575
247)Airbnb Id:4152752     PageRank Value: 0.0021218037191972575
248)Airbnb Id:36487245    PageRank Value: 0.0021218037191972575
249)Airbnb Id:28736904    PageRank Value: 0.0021218037191972575
=====
Time it took to read the hotels's file and find the max/min: 0.0 seconds
Time it took to read the preferences's file: 0.0 seconds
Time it took to make the graph and topk: 6.0 seconds
Time it took to make the image: 18.0 seconds
Time it took to make the PageRank: 0.0 seconds
Time it took to sort the PageRank: 0.0 seconds
BUILD SUCCESSFUL (total time: 48 seconds)
```

7.1.4. Εκτέλεση εφαρμογής (4)

Εκτέλεση με άλλους σημαντικούς χρήστες με τις εξής προτιμήσεις (βάρη):

- price: 2
- reviews: 4
- distance: 10 x: 40.76404 y: -73.98933

```
run:
Reads a file
Give me how much do you care about price:
Enter number from 0 to 10: 2
Give me how much do you care about reviews:
Enter number from 0 to 10: 4
Give me how much do you care about distance:
Enter number from 0 to 10: 10
Enter your location: Give me x axe
40.76404
Give me y axe
-73.98933
Give me how much chain results you want(20 is recommended)
more than that may break the program or take way to many resources and time
3
=====
Calculating TopK and making the graph
Finished TopK and Graph
=====
Graph size:249
Making the Image
The image is ready!!!
=====
Making the pagerank
=====
Calculating the score
=====
Sorting PageRank
1)Airbnb Id:33397385      PageRank Value: 0.009419933126967619
2)Airbnb Id:13894339      PageRank Value: 0.008569803330416667
3)Airbnb Id:31340283      PageRank Value: 0.008294395325602659
4)Airbnb Id:1066769       PageRank Value: 0.008093394985463251
5)Airbnb Id:3799598       PageRank Value: 0.00802196648561497
6)Airbnb Id:12999167      PageRank Value: 0.007731752679787528
7)Airbnb Id:17524441      PageRank Value: 0.007258785528413097
8)Airbnb Id:5270589       PageRank Value: 0.0072569868816288845
9)Airbnb Id:14068646      PageRank Value: 0.007256986881628884
10)Airbnb Id:36001911      PageRank Value: 0.006944868508374594
11)Airbnb Id:35252509      PageRank Value: 0.006766272941334789
12)Airbnb Id:525120       PageRank Value: 0.006680424464278582
13)Airbnb Id:19745952      PageRank Value: 0.006680424464278582
14)Airbnb Id:19226897      PageRank Value: 0.006538424525713302
15)Airbnb Id:9334086       PageRank Value: 0.006011263331814162
16)Airbnb Id:24879207      PageRank Value: 0.005761553659553612
17)Airbnb Id:35199464      PageRank Value: 0.005761553659553612
18)Airbnb Id:20953661      PageRank Value: 0.0055603807345436925
19)Airbnb Id:2440173      PageRank Value: 0.005453456492547007
20)Airbnb Id:32186503      PageRank Value: 0.005453456492547007
```

```
246)Airbnb Id:6141401          PageRank Value: 0.0021218037191972575
247)Airbnb Id:36487245        PageRank Value: 0.0021218037191972575
248)Airbnb Id:34796261        PageRank Value: 0.0021218037191972575
249)Airbnb Id:22386804        PageRank Value: 0.0021218037191972575
=====
Time it took to read the hotels's file and find the max/min: 0.0 seconds
Time it took to read the preferences's file: 0.0 seconds
Time it took to make the graph and topk: 5.0 seconds
Time it took to make the image: 17.0 seconds
Time it took to make the PageRank: 0.0 seconds
Time it took to sort the PageRank: 0.0 seconds
BUILD SUCCESSFUL (total time: 52 seconds)
```

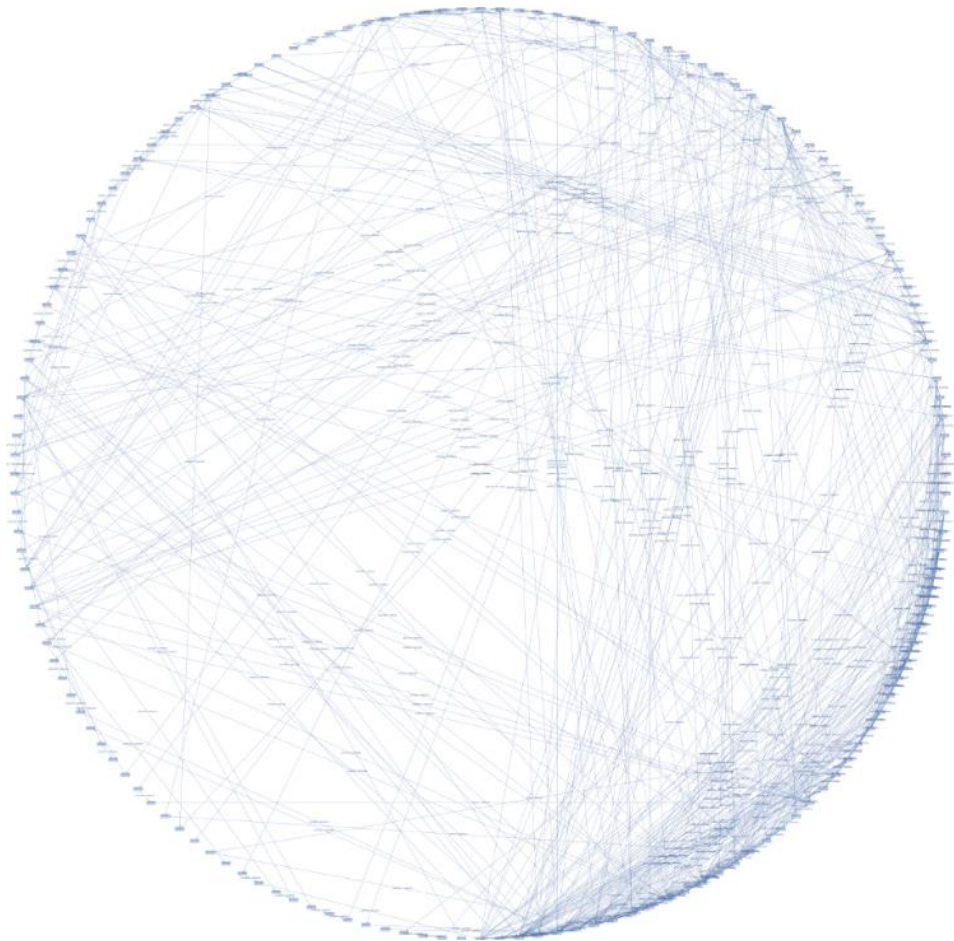
7.2. Τεχνική οπτικοποίησης

Σήμερα είναι διαθέσιμες πολλές μέθοδοι και τεχνικές για την οπτικοποίηση δεδομένων. Στην περίπτωση των αποτελεσμάτων του αλγόριθμου Top-k της εφαρμογής, επιλέχθηκε η οπτικοποίηση του γράφου που δημιουργείται μετά την ολοκλήρωση της συγχώνευσης των γράφων των επιμέρους αποτελεσμάτων αναζήτησης. Στο σημείο αυτό προκύπτουν τα καλύτερα αποτελέσματα και ουσιαστικά προκύπτει η ουσιώδης πληροφορία που είναι απαραίτητη για την ανάλυση του γράφου, ώστε να υπολογιστεί σε αυτόν ο PageRank του κάθε αντικειμένου.

Η οπτικοποίηση έγινε με τη χρήση της βιβλιοθήκης JGraphT της Java και ενσωματώθηκαν σημαντικές λειτουργίες στον κώδικα της εφαρμογής, ώστε να προκύψει το γραφικό αποτέλεσμα.

7.2.1. Ανάλυση της εικόνας του αποτελέσματος

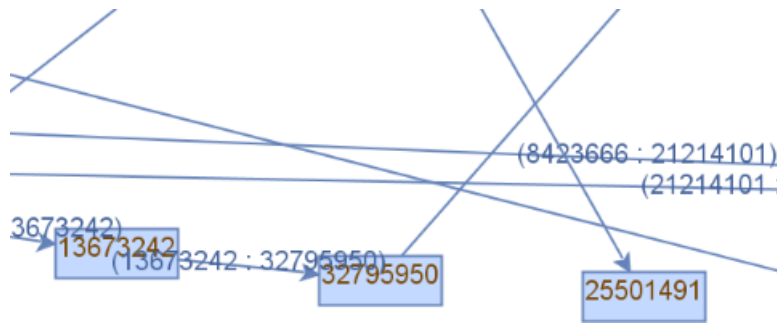
Στο Σχήμα 12 βλέπουμε έναν κύκλο με πολλές γραμμές που διέρχονται από ένα σημείο τις περιφέρειας του κύκλου σε ένα άλλο σημείο τις περιφέρειας.



Σχήμα 12. Αποτέλεσμα της οπτικοποίησης των αποτελεσμάτων αναζήτησης ενοικιαζόμενων Airbnb από την εφαρμογή

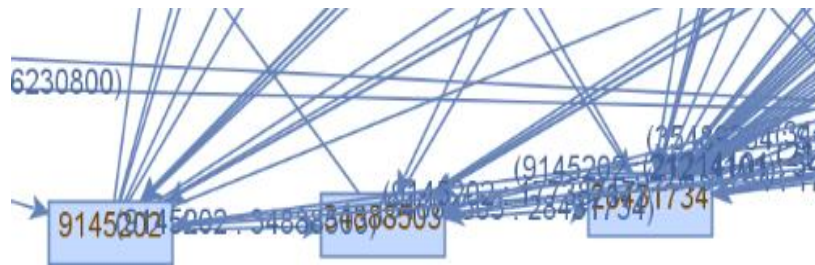
Στο Σχήμα 12 οι κόμβοι αποτελούν ένα ενοικιαζόμενο ακίνητο Airbnb. Οι κόμβοι διατάσσονται στην περιφέρεια ενός κύκλου. Ο κάθε ένας κόμβος αναπαριστά ένα Airbnb το οποίο ουσιαστικά δείχνει στο καλύτερο του, σύμφωνα με τους αλγόριθμους που χρησιμοποιήθηκαν.

Πιο συγκεκριμένα, εστιάζοντας σε ένα σημείο του κύκλου (κάτω και αριστερά στο Σχήμα 12) παρατηρούμε ότι στο κάτω σημείο και αριστερά οι χειρότεροι κόμβοι δεν εμπεριέχουν πολλά inbounds όπως και πολλά backlinks (οι αριθμοί μέσα στα παραλληλόγραμμα είναι τα ID του κάθε Airbnb και όσο πιο αριστερά μετατοπιζόμαστε, τόσο καλύτερο αποτέλεσμα προκύπτει με βάση τον αλγόριθμο PageRank). Αυτό επίσης είναι μία παρατήρηση η οποία προκύπτει και από την ανάλυση της εικόνας, με το ανθρώπινο μάτι, καταδεικνύοντας έτσι τη σημασία της οπτικοποίησης.



Σχήμα 13. Εστίαση στην περιοχή των χειρότερων αποτελεσμάτων.
Οι χειρότεροι κόμβοι δεν έχουν πολλές inbound ακμές.

Αντίθετα, εστιάζοντας στην κάτω δεξιά πλευρά του Σχήματος 12, οι εκεί κόμβοι έχουν και πολλές inbounds ακμές αλλά και backlinks.



Σχήμα 14. Εστίαση στην περιοχή των καλύτερων αποτελεσμάτων.
Οι καλύτεροι κόμβοι έχουν πολλές inbound ακμές

Παρατηρώντας το Σχήμα 14 αντιλαμβανόμαστε ότι ο δεξιός κόμβος ασχέτως του γεγονότος ότι έχει περισσότερες ακμές που καταλήγουν σε αυτόν, δεν είναι ο καλύτερος επειδή έχει σημασία και η ποιότητα των άλλων κόμβων ή στην συγκεκριμένη περίπτωση των Airbnb που συνδέονται με αυτόν .

Το συμπέρασμα της παρατήρησης του Σχήματος 12 είναι ότι όσο πιο αριστερά μετατοπιζόμαστε στον κύκλο, τόσο μεγαλύτερη βαθμολογία αποκόμισε ο αντίστοιχος κόμβος από τον αλγόριθμο PageRank.

Από τα αποτελέσματα του PageRank διαπιστώνουμε ότι αυτά είναι ακριβώς τα ίδια με τα αποτελέσματα που βλέπουμε στον γράφο.

Sorting PageRank

1) Id:9145202	PageRank Value: 0.022047128597861614
2) Id:34888503	PageRank Value: 0.01721370184415007
3) Id:28431734	PageRank Value: 0.017196993742262197
4) Id:17738717	PageRank Value: 0.016670587852567982
5) Id:21214101	PageRank Value: 0.015026662082197193

Σχήμα 15. Αποτελέσματα PageRank στους κόμβους των αποτελεσμάτων αναζήτησης

Τα αποτελέσματα του Σχήματος 15 διαπιστώνουμε ότι είναι όντως τα καλύτερα. Στο Σχήμα 16 που ακολουθεί, διαπιστώνουμε ποια είναι τα χειρότερα αποτελέσματα, ώστε να τα διασταυρώσουμε με τις θέσεις τους στον κύκλο (Σχήμα 12).

148) Id:36230800	PageRank Value: 0.002065761277216904
149) Id:14462592	PageRank Value: 0.002065761277216904
150) Id:13673242	PageRank Value: 0.0019485323584008684
151) Id:32795950	PageRank Value: 0.0019128539403859453
152) Id:25501491	PageRank Value: 0.0017787750964289042

Σχήμα 16. Τα χειρότερα αποτελέσματα του αλγόριθμου PageRank στους κόμβους των αποτελεσμάτων αναζήτησης

Τα αποτελέσματα αυτά προέκυψαν εφαρμόζοντας τον πίνακα προτιμήσεων του χρήστη. Η τοποθεσία του χρήστη επιλέχθηκε τυχαία.

```
Reads a file
Give me how much do you care about FoodRating
Enter number from 0 to 10: 2
Give me how much do you care about PlaceRating
Enter number from 0 to 10: 8
Give me how much do you care about ServiceRating
Enter number from 0 to 10: 5
Enter your location: Give me x axe
456456
Give me y axe
645645
Give me how much results you want
20
=====
Calculating TopK and making the graph
Finished TopK and Graph
=====
Graph size:152
Making the Image
The image is ready!!!
=====
Making the pagerank
=====
Calculating the score
=====
Sorting PageRank
1) Id:9145202          PageRank Value: 0.022047128597861614
2) Id:34888503        PageRank Value: 0.01721370184415007
3) Id:28431734        PageRank Value: 0.017196993742262197
```

Σχήμα 17. Περιβάλλον ορισμού των βαρών στις προτιμήσεις του χρήστη, έναρξης εφαρμογής του αλγόριθμου Top-k και εξαγωγή αποτελεσμάτων αναζήτησης

8. Συμπεράσματα

Η διαδικασία παραγωγής συστάσεων προς τον χρήστη μέσα από ένα σύνολο δεδομένων όπου σε αυτόν παρουσιάζονται οι καλύτερες δυνατές προτάσεις αποτελεί ζωτικό παράγοντα της επιτυχίας ενός δικτύου σήμερα. Η πληθώρα της πληροφορίας που είναι διαθέσιμη, αλλά και οι σχέσεις μεταξύ των πληροφοριών, δημιουργούν ένα πολύπλοκο πλαίσιο για την αναγνώριση των καλύτερων και εγγύτερων προτάσεων προς την επιθυμία του χρήστη. Ο στόχος του συστήματος είναι λοιπόν να προσδιοριστούν λίγα και συναφή προς την επιθυμία του χρήστη αντικείμενα, τα οποία υποτίθεται πως καλύπτουν καλύτερα την εκάστοτε ανάγκη του, είτε πρόκειται για μία ενημερωτική σελίδα, είτε πρόκειται για ένα ξενοδοχείο στο οποίο θέλει να διαμείνει.

Το πρόβλημα των Top-k συστάσεων εμφανίζει μεγάλο ερευνητικό ενδιαφέρον, με αποτέλεσμα να έχουν προταθεί πολλοί αλγόριθμοι επίλυσης του. Μια από τις προσεγγίσεις αποτελεί η χρήση γράφων. Οι αλγόριθμοι που βασίζονται σε γράφους αποδεικνύονται – σύμφωνα και με τα αποτελέσματα της υλοποίησης του συγκεκριμένου έργου – ικανοί να αντιμετωπίσουν ιδιαίτερα αποδοτικά το πρόβλημα της εξαγωγής των πλέον κατάλληλων προτάσεων, διατηρώντας το πλήθος τους μικρό, ώστε ο χρήστης να έχει άμεσα τις πιο χρήσιμες ή επιθυμητές γι’ αυτόν συστάσεις. Το βασικό χαρακτηριστικό τους είναι ότι επιτρέπουν την αλληλεπίδραση του χρήστη με τον αλγόριθμο μέσω των ιδιοτήτων των δεδομένων και των βαρών που αυτός αποδίδει σε κάθε χαρακτηριστικό. Ο αλγόριθμος έχει την ικανότητα ακόμα και σε περιπτώσεις που δεν υπάρχει άμεση σύνδεση μεταξύ κάποιων πληροφοριών, να μεταδίδει πληροφορία μέσω των ακμών του γραφήματος που συνδέουν τους κόμβους των αποτελεσμάτων μιας αναζήτησης. Ο δημοφιλέστερος αλγόριθμος επίλυσης του προβλήματος Top-k με χρήση γράφων είναι ο PageRank, ο οποίος με επιτυχία χρησιμοποιήθηκε από την Google στη μηχανή αναζήτησής της. Ο αλγόριθμος αυτός υλοποιήθηκε στην εφαρμογή ώστε από τον γράφο των αποτελεσμάτων που δημιουργείται από μία επρώτηση του συστήματος να εξάγονται εκείνοι οι κόμβοι με την καλύτερη ποιότητα συνδέσεων. Αυτός ο μηχανισμός βελτιώνει σημαντικά τη συνάφεια των αποτελεσμάτων με τις προσδοκίες του χρήστη, δεδομένου ότι λαμβάνεται υπόψη και ο παράγοντας της «ποιότητας» των γράφων από τους οποίους καταλήγουν ακμές σε έναν κόμβο του αποτελέσματος.

Ένα σημαντικό χαρακτηριστικό που ενσωματώθηκε στην εφαρμογή, είναι η οπτικοποίηση των αποτελεσμάτων με τη χρήση συνεκτικών κατευθυνόμενων γράφων, στους οποίους τα καλύτερα αποτελέσματα προσδιορίζονται με τρόπο γραφικό, σύμφωνα με την πυκνότητα των ακμών που καταλήγουν στους προτεινόμενους κόμβους.

Ένα συμπέρασμα που επίσης προέκυψε ήταν η απόδοση του αλγόριθμου. Οι επερωτήσεις εξυπηρετούνται σε πολύ μικρό χρονικό διάστημα, της τάξης των 6 sec

(για τον προσδιορισμό των Top-k αποτελεσμάτων όπως και για την δημιουργία του γράφου) και της τάξης των 18 sec για την παραγωγή των πιο σύνθετων γράφων.

Η εφαρμογή που αναπτύχθηκε μπορεί να επεκταθεί σχετικά εύκολα, ώστε να δέχεται σαν είσοδο δεδομένα από πολλαπλές πηγές δεδομένων (βάσεις δεδομένων, διαδικτυακές υπηρεσίες), και να εξυπηρετεί διαφορετικές ανάγκες αναζητήσεων κάθε φορά.

Βιβλιογραφία

- Baeza-Yates, R. A. & Ribeiro-Neto, B. A., 1999. *Modern Information Retrieval*. s.l.:ACM Press / Addison-Wesley.
- Bruno, N., Gravano, L. & Marian, A., 2002. *Evaluating Top-k Queries over Web-accessible Databases*. s.l., ICDE.
- Du, Y., Zhang, D. & Xia, T., 2005. *The Optimal-Location Query*. s.l., SSTD, 2005.
- Eclipse Collections, 2018. *Eclipse Collections*. [Online]
Available at: <https://www.eclipse.org/collections/>
[Accessed January 2021].
- fastutil, 2020. *fastutil: Fast & compact type-specific collections for Java*. [Online]
Available at: <http://fastutil.di.unimi.it/>
[Accessed January 2021].
- Gamma, E., Helm, R., Johnson, R. & Vlissides, J., 1993. *Design patterns: Abstraction and reuse of object-oriented design*. s.l., European Conference on Object-Oriented Programming.
- Gunopulos, D., 2020. *Top-K Query Processing*, Αθήνα: Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών, Τμήμα Πληροφορικής. Διαλέξεις μαθήματος).
- Guptill, S. C., 2001. Spatial Data. In: N. J. Smelser & P. B. Baltes, eds. *International Encyclopedia of the Social & Behavioral Sciences*. s.l.:Elsevier Ltd.
- Guptill, S. C. & Morrison, J. L., 1999. *Elements of Spatial Data Quality*. Oxford, UK: Pergamon.
- Hjaltason, G. R. & Samet, H., 1999. Distance Browsing in Spatial Databases. *TODS*, 24(2), p. 265–318.
- Konecny, G., Breitkopf, U., Radtke, A. & Lee, K., 2016. *The status of topographic mapping in the world*, s.l.: Institute of Photogrammetry and GeoInformation, Leibniz University Hannover, Germany.
- Laurini, R. & Thompson, D., 1999. *Fundamentals of Spatial Information Systems*. London: Academic Press.
- Michail, D., Kinable, J., Naveh, B. & Sichi, J., 2020. JGraphT—A Java Library for Graph Data Structures and Algorithms. *ACM Transactions on Mathematical Software*.

- Mind the SEO, 2020. *Πώς λειτουργεί η αναζήτηση της Google;*. [Online]
[Accessed Οκτώβριος 2020].
- O'Madadhain, J., Fisher, D., White, S. & Boey, Y., 2003. *The jung (java universal network/graph) framework*, Irvine, California: University of California.
- Saad, Y., 2003. *Iterative methods for sparse linear systems*. 2nd edition ed.
Philadelphia, PA: Society for Industrial and Applied Mathematics SIAM.
- Safe Software, 2020. *Why Spatial Data is Important*. [Online]
[Accessed Οκτώβριος 2020].
- Southern, M., 2016. *Google PageRank Officially Shuts its Doors to the Public*.
[Online]
Available at: <https://www.searchenginejournal.com/google-pagerank-official-shuts-doors-public/161874/#close>
[Accessed Οκτώβριος 2020].
- Sullivan, D., 2016. *RIP Google PageRank score: A retrospective on how it ruined the web*. [Online]
Available at: <https://searchengineland.com/rip-google-pagerank-retrospective-244286>
[Accessed Σεπτέμβριος 2020].
- Tao, Y., Hristidis, V., Papadias, D. & Papakonstantinou, Y., 2014. Branch-and-Bound Processing of Ranked Queries. *Information Systems*.
- Vlachou, A., Doulkeridis, C., Kotidis, Y. & Nørnvåg, K., 2020. *Reverse Top-k Queries*,
Αθήνα: Οικονομικό Πανεπιστήμιο Αθηνών.
- Vlachou, A., Doulkeridis, C., Kotidis, Y. & Nørnvåg, K., 2020. *Reverse Top-k Queries*,
s.l.: s.n.
- Weber, R., Schek, H. J. & Blott, S., 1998. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. *VLDB*.
- Wikipedia, 2020. *Partial sorting*. [Online]
[Accessed Ιανουάριος 2021].
- Wikipedia, 2021. *Location-based service*. [Online]
Available at: https://en.wikipedia.org/wiki/Location-based_service
[Accessed Νοέμβριος 2020].
- Xia, T., Zhang, D., Kanoulas, E. & Du, Y., 2005. On Computing Top-t Most Influential Spatial Sites. *VLDB*.

Zhang, D., Du, Y., Xia, T. & Tao, Y., 2006. Progressive Computation of The Min-Dist Optimal-Location Query. *VLDB*.

Σφυρής, Γ., 2016. *Υπολογισμός ανάστροφων ερωτημάτων κατάταξης σε κατανεμημένο περιβάλλον*, Πειραιάς: Πανεπιστήμιο Πειραιώς - Μεταπτυχιακή Διπλωματική Εργασία.