

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Πτυχιακή Εργασία

**Αναλυτική δεδομένων σε γράφους με αλγορίθμους τυχαίων
περιπάτων**

ΜΠΡΙΦΑΣ ΕΜΜΑΝΟΥΗΛ

Π2013140

Επιβλέπων: Βασίλειος Καρυώτης

Αθήνα 2021

Επιβλέπων

Βασίλειος Καρυώτης

Τριμελής Επιτροπή

Βασίλειος Καρυώτης

Παναγιώτης Κουρουθανάσης

Θεμιστοκλής Έξαρχος

Περίληψη

Οι γράφοι είναι σημαντικοί τύποι δεδομένων σε πολλούς επιστημονικούς τομείς, όπως στην ανάλυση κοινωνικών μέσων δικτύωσης, στη βιοπληροφορική και στην ανάλυση δικτύων υπολογιστών και πληροφοριών. Τα τελευταία χρόνια, το μέγεθος των δεδομένων των γράφων έχει αυξηθεί δραματικά. Για παράδειγμα, ένας τυπικός γράφος κοινωνικών δικτύων μπορεί να περιέχει εκατομμύρια κορυφές και εκατοντάδες εκατομμύρια άκρες. Επιπλέον, αυτοί οι γράφοι μπορεί να εξελίσσονται συνεχώς με την πάροδο του χρόνου. Η επεξεργασία αυτών των δυναμικών δεδομένων μεγάλων γράφων είναι πολύ δύσκολη και χρονοβόρα. Γενικά, οι μεγάλοι γράφοι είναι συνήθως ετερογενείς. Έχουν μη ομοιόμορφες δομές όπου τα άκρα μεταξύ κορυφών σε μια ομάδα είναι πολύ πυκνότερα από τα άκρα που συνδέουν κορυφές σε διαφορετικές ομάδες. Οι αλγόριθμοι ομαδοποίησης γράφου στοχεύουν στην αποκάλυψη αυτής της ετερογένειας και στην εύρεση των υποκείμενων σχέσεων μεταξύ κορυφών. Αυτή η τεχνική είναι κρίσιμη για την κατανόηση των ιδιοτήτων, την πρόβλεψη δυναμικής συμπεριφοράς και τη βελτίωση της οπτικοποίησης δεδομένων μεγάλων γράφων.

Για δεδομένα μεγάλων γραφημάτων, το πρόβλημα γίνεται πιο δύσκολο ή ακόμη και δυσδιάκριτο. Πολύ συχνά, οι άνθρωποι ενδιαφέρονται μόνο να βρουν την ομάδα για στην οποία ανήκει μια δεδομένη κορυφή. Αυτό το πρόβλημα ονομάζεται πρόβλημα τοπικής ομαδοποίησης. Για παράδειγμα, όσον αφορά έναν χρήστη κοινωνικών μέσων δικτύωσης, η εύρεση των στενά συνδεδεμένων φίλων του είναι πιο σημαντική από την εμφάνιση/αποκάλυψη των παγκόσμιων ομάδων χρηστών. Δεν είναι απαραίτητο να εξερευνηθεί ολόκληρη η δομή του γράφου για αυτό το πρόβλημα. Πρόσφατα, οι μέθοδοι τυχαίων περιπάτων έχουν αποκτήσει μεγάλο ενδιαφέρον σε αυτό το τοπικό πρόβλημα ομαδοποίησης γράφων, καθώς μια επανάληψη (περίπατος) που ξεκίνησε από την κορυφή αρχής είναι πιθανότερο να παραμείνει στο σύμπλεγμα όπου ανήκει κιόλας. Συγκριτικά με τις μεθόδους βάσει κριτηρίων, οι μέθοδοι τυχαίων περιπάτων είναι ικανές να εξαγάγουν τοπικές πληροφορίες από ένα μεγάλο γράφων χωρίς τη γνώση ολόκληρων των δεδομένων του γράφου.

Στη συγκεκριμένη Πτυχιακή εργασία γίνεται αρχικά μία βιβλιογραφική ανασκόπηση των αλγόριθμων τυχαίων περιπάτων. Έπειτα επιλέγονται προσεγγίσεις από αυτές οι οποίες θα συγκριθούν μεταξύ τους με προσομοίωση και ανάλυση.

Abstract

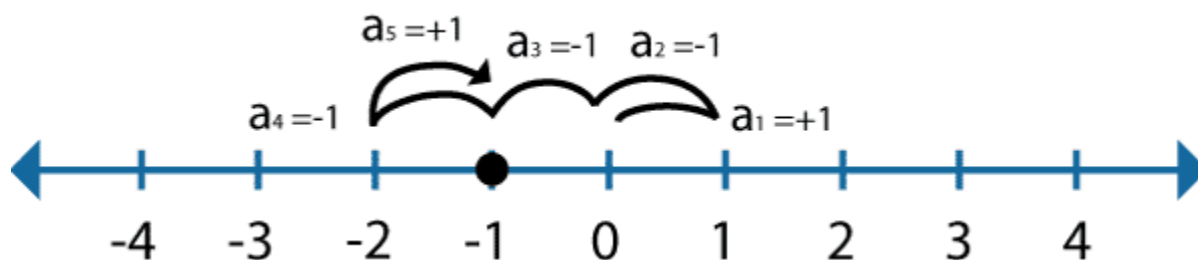
Graphs are important types of data in many scientific fields, such as social media analysis, bioinformatics, and computer and information network analysis. In recent years, the size of graph data has increased dramatically. For example, a typical social network graph can contain millions of nodes and hundreds of millions of edges. In addition, these graphs can evolve continuously over time. Processing this dynamic big graph data is very difficult and time consuming. In general, large graphs are usually heterogeneous. They have non-uniform structures where the ends between vertices in a group are much denser than the ends connecting vertices in different groups. Graph grouping algorithms aim to reveal this heterogeneity and to find the underlying relationships between vertices. This technique is critical to understanding the properties, predicting dynamic behavior, and improving the visualization of large graph data. For large graph data, the problem becomes more difficult or even indistinguishable. Too often, people are only interested in finding the team to which a given node belongs. This problem is called local grouping. For example, for a social media user, finding his / her close friends is more important than showing / revealing global groups of users. It is not necessary to explore the whole structure of the graph for this problem. Recently, random walk methods have gained much interest in this local graph clustering problem, as a repetition (walk) that started from the node is more likely to remain in the cluster to which it already belongs. Compared to criterion-based methods, random walk methods are capable of extracting local information from a large graph without knowing all of the graph data. In this dissertation, a bibliographic review of random walk algorithms is performed. Approaches are then selected from them, which will be compared with each other by simulation and analysis.

Πίνακας Περιεχομένων

Αναλυτική δεδομένων σε γράφους με αλγορίθμους τυχαίων περιπάτων	1
Περίληψη	4
Κεφάλαιο 1. Εισαγωγή.....	7
Κεφάλαιο 2. Αλγόριθμοι τυχαίων περιπάτων	13
2.1 Σύντομη ιστορική αναδρομή και εφαρμογές των αλγορίθμων τυχαίων περιπάτων	13
2.2. Αλγόριθμος Page Rank.....	15
2.3. Αλγόριθμος RWR (Random Walk with Restart).....	17
2.4. Αλγόριθμος B_LIN	21
2.5. Αλγόριθμος LRW (Limited Random Walk)	23
2.6. Αλγόριθμος K_DASH	25
2.7. Αλγόριθμος Node2vec	27
Κεφάλαιο 3. Ανάλυση των επιλεγθέντων δεδομένων	30
3.1. Η γλώσσα Python.....	30
3.2. Μετρικές και χαρακτηριστικά δικτύων	31
3.3. Πληροφορίες για τις επιλεγθέντες βάσεις δεδομένων	39
3.4. Παρουσίαση αποτελεσμάτων και κώδικας.	45
3.4.1. Κώδικας στην Γλώσσα Python	47
3.4.2 Έξοδοι κώδικα και συγκεντρωτικά αποτελέσματα.....	50
Κεφάλαιο 4. Παρατηρήσεις και Συμπεράσματα	54
Βιβλιογραφία	60

Κεφάλαιο 1. Εισαγωγή

Στη μαθηματική επιστήμη, ένας τυχαίος περίπατος (random walk) είναι ένα μαθηματικό αντικείμενο, γνωστό ως στοχαστική ή τυχαία διαδικασία, που περιγράφει μια διαδρομή που αποτελείται από μια διαδοχή τυχαίων βημάτων σε κάποιο μαθηματικό χώρο, όπως οι ακέραιοι. Ένα στοιχειώδες παράδειγμα τυχαίου περιπάτου είναι το τυχαίο «βάδισμα» στην ακέραια αριθμητική γραμμή, $\{Z\}$, το οποίο ξεκινά από το 0 και σε κάθε βήμα κινείται $+1$ ή -1 με ίση πιθανότητα. Άλλα παραδείγματα περιλαμβάνουν τη διαδρομή που ακολουθεί ένα μόριο καθώς ταξιδεύει σε υγρό ή αέριο (Brownian motion), τη διαδρομή που ακολουθεί ένα άγριο ζώο με σκοπό την αναζήτηση τροφής, την τιμή μιας μετοχής με διακυμάνσεις και την οικονομική κατάσταση ενός ανθρώπου που είναι εθισμένος στα τυχερά παιχνίδια. (E. Wirth) Όλα μπορούν να προσεγγιστούν από τυχαία μοντέλα αλγορίθμων τυχαίων περιπάτων, παρόλο που μπορεί να μην είναι τόσο τυχαία στην πραγματικότητα.



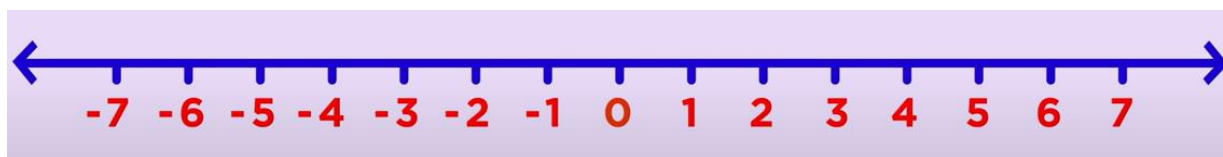
Όπως φαίνεται από αυτά τα παραδείγματα, οι τυχαίοι περίπατοι έχουν εφαρμογές στη μηχανική και σε πολλούς επιστημονικούς τομείς, όπως η οικολογία, η ψυχολογία, η επιστήμη των υπολογιστών, η φυσική, η χημεία, η βιολογία, η οικονομία και η κοινωνιολογία. Οι τυχαίοι περίπατοι εξηγούν τις παρατηρούμενες συμπεριφορές πολλών διαδικασιών σε αυτούς τους τομείς και έτσι χρησιμεύουν ως θεμελιώδες μοντέλο για την καταγεγραμμένη στοχαστική δραστηριότητα. Ως πιο μαθηματική εφαρμογή, η τιμή του π μπορεί να προσεγγιστεί με τη χρήση τυχαίου περιπάτου σε ένα agent-based περιβάλλον μοντελοποίησης, δηλαδή ένα υπολογιστικό μοντέλο για την προσομοίωση των ενεργειών και των αλληλεπιδράσεων αυτόνομων παραγόντων προκειμένου να κατανοηθεί η συμπεριφορά ενός συστήματος και τι διέπει τα αποτελέσματά του. Ο όρος τυχαίος περίπατος εισήχθη για πρώτη φορά από τον Karl Pearson το 1905 (PEARSON, 1905).

Διάφοροι τύποι τυχαίων περιπάτων παρουσιάζουν ενδιαφέρον, οι οποίοι μπορεί να διαφέρουν με

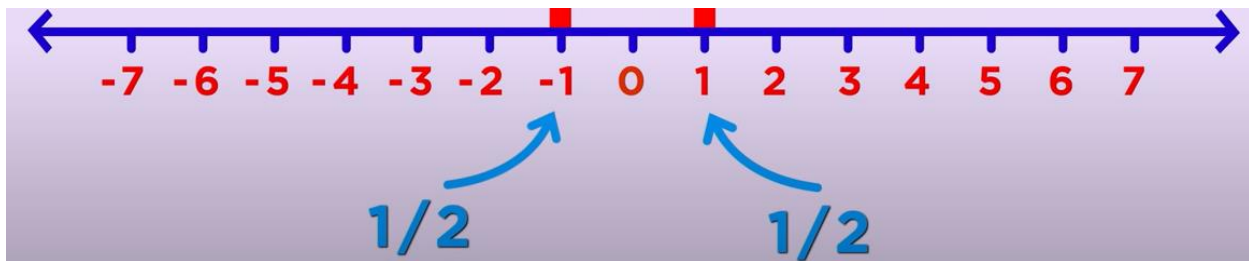
διάφορους τρόπους. Ο ίδιος ο όρος «τυχαίος περίπατος» αναφέρεται συχνότερα σε μια ειδική κατηγορία αλυσίδων Markov, αλλά πολλές από τις εκτελούμενες διαδικασίες που εξαρτώνται από το χρόνο αναφέρονται ως τυχαίες βόλτες, με έναν τροποποιητή να υποδεικνύει τις συγκεκριμένες ιδιότητές τους. Οι τυχαίοι περίπατοι (Markov ή όχι) μπορούν επίσης να πραγματοποιηθούν σε διάφορους χώρους με τους πιο κοινώς μελετημένους αυτούς που περιλαμβάνουν γράφους: στους ακέραιους αριθμούς ή στην γραμμή των πραγματικών αριθμών, στο επίπεδο ή σε διανυσματικά διαστήματα μεγαλύτερων διαστάσεων, σε κυρτές επιφάνειες ή σε επιφάνειες Riemannian πολλαπλών διαστάσεων, καθώς και σε πεπερασμένες ομάδες ή σε ομάδες Lie (ποικίλος διαφοροποιήσιμες).

Για να εξηγηθεί καλύτερα παραστατικά ένας τυχαίος περίπατος, προτού αναλυθεί μαθηματικά, αρκεί να αναφερθεί το παράδειγμα της πιο απλής μορφής του: η αριθμογραμμή ακέραιων αριθμών.

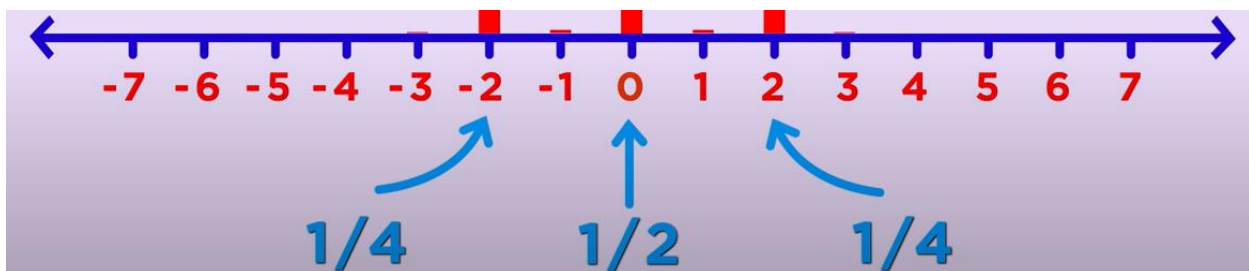
Έστω ότι έχουμε ένα νόμισμα με τη μία πλευρά να υπάρχει το σύμβολο «+» και από την άλλη το σύμβολο «-». Ρίχνοντας το κέρμα συνολικά 100 φορές θα παρακολουθήσουμε τις πιθανότητες που αντιστοιχούν στις κινήσεις πάνω στην αριθμογραμμή. Παρακάτω θα παρουσιάσουμε τις πιθανότητες για τις πρώτες 4 φορές αλλά τα βασικά συμπεράσματα του απλού αυτού παραδείγματος θα παρουσιαστούν γενικά και για τις 100 ρίψεις. Κάθε φορά στην περίπτωση που έρθει το σύμβολο «+» μετακινούμαστε μία θέση ακεραίου δεξιά στην παρακάτω αριθμογραμμή ενώ αν έρθει το σύμβολο «-» μετακινούμαστε μία θέση ακεραίου αριστερά.



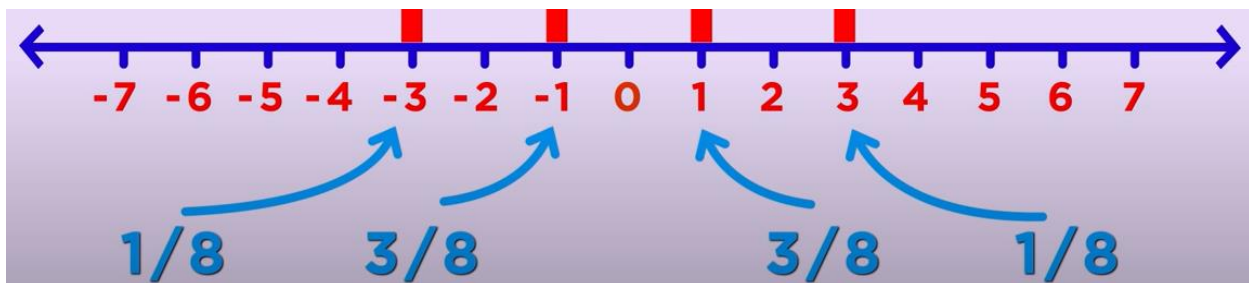
Όπως είναι λογικό ύστερα από ένα γύρισμα του νομίσματος οι πιθανότητες είναι 50% να βρισκόμαστε στη θέση του ακέραιου αριθμού 1 και 50% να βρισκόμαστε στη θέση του ακέραιου αριθμού -1



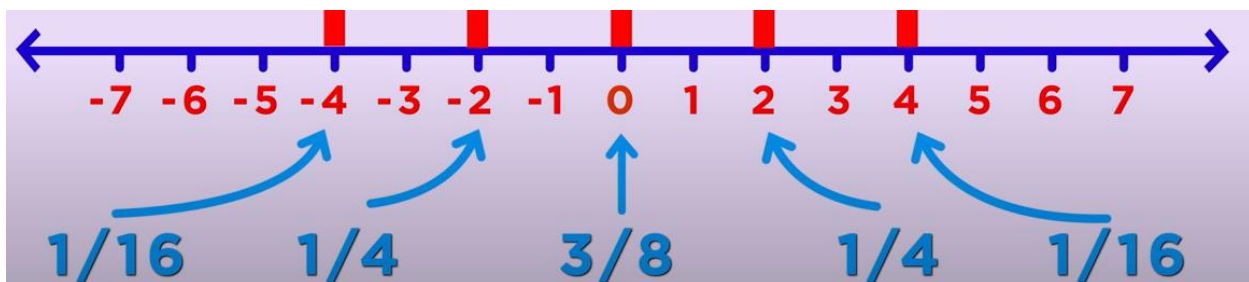
Συνεχίζοντας, γυρνάμε το νόμισμα την δεύτερη φορά οι πιθανότητες διαμορφώνονται αντιστοίχως:



Την τρίτη φορά οι πιθανότητες διαμορφώνονται ως εξής:



Ενώ την τέταρτη φορά οι πιθανότητες διαμορφώνονται ως εξής:



Πριν λοιπόν αναλυθεί μαθηματικά ο αλγόριθμος τυχαίων περιπάτων μπορεί το παραπάνω απλό παράδειγμα να οδηγήσει στα εξής 3 σημαντικά συμπεράσματα:

Στις ρίψεις του νομίσματος που αντιστοιχούν σε μονό αριθμό ρίψης π.χ. (1,3,5,7,9) η κατάληξη του δείκτη της αριθμογραμμής θα είναι πάντα σε μονό αριθμό, ενώ στις ρίψεις του νομίσματος που αντιστοιχούν σε ζυγό αριθμό ρίψης, η κατάληξη θα είναι σε ζυγό αριθμό.

Στις 100 ρίψεις του νομίσματος η κατανομή που θα σχηματιστεί θα είναι μορφής καμπάνας, και

Γενικότερα για N ρίψεις νομίσματος τα περισσότερα αποτελέσματα θα συγκεντρώνονται μεταξύ των αριθμών $-\sqrt{N}$ και $+\sqrt{N}$

Η τεχνική τυχαίων περιπάτων είναι μια μέθοδος τυχαίας δειγματοληψίας στην οποία ο αριθμός των βημάτων μεταξύ των δειγμάτων καθορίζεται από τυχαίους αριθμούς, οι οποίοι συνήθως αντλούνται από πίνακες τυχαίων αριθμών. Όταν λαμβάνονται διάφορα σύνολα δειγμάτων, είναι καλύτερο να ξεκινάει η διαδικασία κάθε συνόλου από το ίδιο σημείο. Εάν τούτο είναι αδύνατο, είναι σημαντικό το εύρος των αριθμών από το οποίο γίνεται μια επιλογή να επιτρέπει σε κάθε δείγμα της μελέτης να έχει ίσες πιθανότητες να επιλεγεί κάθε φορά. Στην ουσία είναι μία στοχαστική διαδικασία ειδικής μορφής που μπορεί να ερμηνευθεί ως μοντέλο που περιγράφει την κίνηση ενός σωματιδίου σε ένα συγκεκριμένο χώρο κατά την δράση κάποιου τυχαίου μηχανισμού. Ο χώρος κατάστασης είναι συνήθως ένας d -διάστατος ευκλείδειος χώρος ή το αναπόσπαστο πλέγμα σε αυτόν. Ο τυχαίος μηχανισμός μπορεί να είναι διαφόρων ειδών. Οι πιο συχνές τυχαίοι περίπατοι είναι αυτοί που δημιουργούνται από άθροισμα ανεξάρτητων τυχαίων μεταβλητών ή από αλυσίδες Markov. Δεν υπάρχει ακριβής γενικά αποδεκτός ορισμός ενός τυχαίου περιπάτου.

Οι τροχιές ενός τυχαίου περιπάτου στην περίπτωση $d = 1$ περιγράφονται από την αρχική θέση $S_0=0$ και την ακολουθία των αθροισμάτων

$$S_n = X_1 + \dots + X_n, n = 1, 2, \dots$$

όπου το X_n είναι ανεξάρτητα και ακολουθούν την κατανομή Bernoulli

$$P\{X_i = 1\} = p, P\{X_i = -1\} = q = 1 - p, p \in (0,1).$$

Η αξία του S_n μπορεί να ερμηνευθεί ως το κέρδος ενός από τους δύο παίκτες μετά από n παιχνίδια σε καθένα από τα οποία αυτός ο παίκτης κερδίζει ένα δολάριο με πιθανότητα p και το χάνει με πιθανότητα $1 - p$. Εάν το παιχνίδι αποτελείται από το πέταγμα ενός αμερόληπτου νομίσματος,

τότε θεωρείται ότι $p = 1/2$. Υπό την προϋπόθεση ότι το αρχικό κεφάλαιο του πρώτου παίκτη είναι b και αυτό του δεύτερου παίκτη είναι a , το παιχνίδι θα τελειώσει όταν το κινούμενο σωματίδιο (με συντεταγμένες S_1, S_2) αγγίζει πρώτα ένα από τα επίπεδα a ή $-b$. Αυτή τη στιγμή, ένας από τους παίκτες έχει καταστραφεί. Αυτό είναι το κλασικό πρόβλημα καταστροφής, στο οποίο τα όρια στα σημεία a και $-b$ μπορούν να θεωρηθούν απορροφητικά (randomwalk).

Εάν $a = \infty$, τότε κάποιος αντιμετωπίζει το πρόβλημα τυχαίου περιπάτου με ένα όριο. Εάν $a = b = \infty$, τότε κάποιος αντιμετωπίζει έναν απεριόριστο τυχαίο περίπατο. Οι τυχαίοι περίπατοι μελετώνται συνήθως χρησιμοποιώντας τη συσκευή διακριτών αλυσίδων Markov και, ειδικότερα, διερευνώντας τις αντίστοιχες εξισώσεις πεπερασμένων διαφορών. Για παράδειγμα, στο πρόβλημα της καταστροφής, έστω ότι u_k είναι η πιθανότητα να καταστραφεί ο πρώτος παίκτης, δεδομένου ότι το αρχικό του κεφάλαιο είναι ίσο με k , $0 \leq k \leq a + b$, όπου το συνολικό κεφάλαιο και των δύο παικτών είναι σταθερό και ίσο με $a + b$. Στη συνέχεια, με τον τύπο συνολικής πιθανότητας, προκύπτει ότι το u_k ικανοποιεί την εξίσωση

$$u_k = pu_{k+1} + 1 + qu_{k-1}, 0 < k < a + b,$$

και οι οριακές συνθήκες $u_a + b = 0$, $u_0 = 1$.

$$u_b = \frac{\left(\frac{q}{p}\right)^a + b - \left(\frac{q}{p}\right)^b}{\left(\frac{q}{p}\right)^a + b - 1} \text{ όπου } p \neq q,$$

$$u_b = \frac{a}{a + b} \text{ όπου } p = q = \frac{1}{2}$$

Η δεύτερη σχέση δείχνει ότι ακόμη και ένα "δίκαιο" παιχνίδι (στο οποίο και οι δύο παίκτες έχουν τις ίδιες πιθανότητες) οδηγεί σε καταστροφή με πιθανότητα κοντά στο 1, υπό την προϋπόθεση ότι το κεφάλαιο a του δεύτερου παίκτη είναι μεγάλο σε σύγκριση με το b ($u_b = 1$ όταν $b < \infty$, $a = \infty$).

Ένας τυχαίος περίπατος είναι μια τυχαία διαδικασία στο μαθηματικό χώρο. Περιγράφει μια διαδρομή που αποτελείται από μια διαδοχή τυχαίων βήματα στον μαθηματικό χώρο. Οι τυχαίες βόλτες μπορούν να χρησιμοποιηθούν για την ανάλυση και προσομοίωση της τυχαιότητας των

δεδομένων και υπολογίστε τη συσχέτιση μεταξύ αντικειμένων, η οποία είναι χρήσιμα για την επίλυση πρακτικών προβλημάτων. Γίνεται γρήγορα και αποτελεί βασικό όργανο στους τομείς της επιστήμης των υπολογιστών, της φυσικής, χημείας, βιολογίας, οικονομίας κ.λπ. Στο μαθηματικό χώρο, ένα απλό τυχαίο μοντέλο περιπάτου είναι ένας τυχαίος περίπατος σε ένα κανονικό πλέγμα, στον οποίο ένα σημείο μπορεί να περπατήσει σε άλλη θέση σε κάθε βήμα σύμφωνα με μία ορισμένη κατανομή πιθανότητας. Όταν εφαρμόζεται σε ένα συγκεκριμένο δίκτυο, η πιθανότητα μετάβασης μεταξύ των κόμβων είναι θετική σχετικά με τη δύναμη συσχέτισής τους. Δηλαδή, όσο ισχυρότεροι είναι τόσο μεγαλύτερη είναι η πιθανότητα μετάβασης. Μετά από αρκετά βήματα, μπορούμε να αποκτήσουμε μια τυχαία διαδρομή που μπορεί να περιγράψει τη δομή του δικτύου.

Κεφάλαιο 2. Αλγόριθμοι τυχαίων περιπάτων

Οι τυχαίοι περίπατοι σε πεπερασμένα δίκτυα είναι ένα αγαπημένο ερευνητικό θέμα εδώ και δεκαετίες. Ίσως πιο σημαντικά, οι τυχαίοι περίπατοι αποτελούν βασική τεχνική για τη δημιουργία αλγορίθμων για την εξαγωγή χρήσιμων πληροφοριών από δεδομένα δικτύου. Τέτοιες εφαρμογές τυχαίων περιπάτων περιλαμβάνουν ανίχνευση κοινότητας, κατάταξη κόμβων και ακμών, μείωση διαστάσεων δεδομένων, δειγματοληψία, κ.α. Πολλές θεωρητικές, υπολογιστικές και αλγοριθμικές μελέτες έχουν χρησιμοποιήσει απλούς τυχαίους περιπάτους σε μη σταθμισμένα δίκτυα, κάτι που εξ ορισμού υπαγορεύει ότι ένας agent μετακινείται σε έναν από τους γείτονές του με την ίδια πιθανότητα σε κάθε χρονικό βήμα.

2.1 Σύντομη ιστορική αναδρομή και εφαρμογές των αλγορίθμων τυχαίων περιπάτων

Το 1863, ο Γάλλος μαθηματικός Jules Regnault έγινε χρηματιστής και δημοσίευσε ένα βιβλίο με τίτλο "Calcul des Chances et Philosophie de la Bourse" ή "The Study of Chance and the Philosophy of Exchange". Το έργο του Regnault θεωρείται μία από τις πρώτες προσπάθειες για τη χρήση προηγμένων μαθηματικών στην ανάλυση του χρηματιστηρίου.

Επηρεασμένος από το έργο του Regnault, ο Louis Bachelier, ένας άλλος Γάλλος μαθηματικός, δημοσίευσε μια εργασία με τίτλο «Théorie de la Spéculation» ή «Theory of Speculation». Αυτό το έργο πιστώνεται με τη θέσπιση των βασικών κανόνων που θα είναι καθοριστικής σημασίας για τη χρήση των μαθηματικών και των στατιστικών μεθόδων στο χρηματιστήριο.

Το 1964, ο Αμερικανός οικονομικός οικονομολόγος Paul Cootner δημοσίευσε ένα βιβλίο με τίτλο «The Random Character of Stock Market Prices». Θεωρείται κλασικό κείμενο στον τομέα των χρηματοοικονομικών, ενέπνευσε άλλα έργα, όπως το "A Random Walk Down Wall Street" του Burton Malkiel και το "Random Walks in Stock Market Τιμές" της Eugene Farma.

Δεδομένου ότι η θεωρία τυχαίου περιπάτου υποστηρίζει ότι είναι αδύνατο να προβλεφθεί η κίνηση των τιμών των μετοχών, είναι επίσης αδύνατο για έναν επενδυτή χρηματιστηριακών αγορών να ξεπεράσει ή να "νικήσει" την αγορά μακροπρόθεσμα. Αυτό σημαίνει ότι είναι αδύνατο για έναν επενδυτή να ξεπεράσει την αγορά χωρίς να αναλάβει μεγάλα ποσά πρόσθετου κινδύνου.

Το 1988, η θεωρία τυχαίου περιπάτου δοκιμάστηκε στον περίφημο διαγωνισμό επένδυσης με ρίψη

βέλους. Σχεδιασμένος από τη Wall Street Journal, αυτός ο διαγωνισμός έφερε αντιμέτωπους επαγγελματίες επενδυτές που εργάζονταν έξω από το Χρηματιστήριο της Νέας Υόρκης ενάντια σε εικονικούς επενδυτές. Οι εικονικοί επενδυτές αποτελούνταν από το προσωπικό της Wall Street Journal που επέλεξε μετοχές ρίχνοντας βελάκια σε έναν πίνακα. Το πείραμα, με τίτλο «The Wall Street Journal Dartboard Contest», κέρδισε μεγάλη δημοσιότητα και την προσοχή των ΜΜΕ. Από 100 διαγωνισμούς, οι επαγγελματίες επενδυτές κέρδισαν 61, ενώ τα ομοιώματα «ρίψης βέλους» κέρδισαν 39. Μία από τις κύριες κριτικές της θεωρίας τυχαίου περιπάτου είναι ότι το χρηματιστήριο αποτελείται από μεγάλο αριθμό επενδυτών και ο χρόνος που ξοδεύει κάθε επενδυτής στην αγορά είναι διαφορετικός. Συνεπώς, είναι πιθανό βραχυπρόθεσμα να εμφανιστούν τάσεις στις τιμές των κινητών αξιών και ένας έξυπνος επενδυτής μπορεί να ξεπεράσει την αγορά αγοράζοντας στρατηγικά μετοχές όταν η τιμή είναι χαμηλή και πουλώντας μετοχές όταν η τιμή είναι υψηλή σε σύντομο χρονικό διάστημα. Άλλοι κριτικοί υποστηρίζουν ότι ολόκληρη η βάση της θεωρίας τυχαίων περιπάτων είναι ελαττωματική και ότι οι τιμές των μετοχών ακολουθούν μοτίβα ή τάσεις, ακόμη και μακροπρόθεσμα. Υποστηρίζουν ότι επειδή η τιμή ενός τίτλου επηρεάζεται από έναν εξαιρετικά μεγάλο αριθμό παραγόντων, μπορεί να είναι αδύνατο να διακρίνουμε το μοτίβο ή την τάση που ακολουθείται από την τιμή της συγκεκριμένης αξίας. Ωστόσο, μόνο και μόνο επειδή ένα μοτίβο δεν μπορεί να προσδιοριστεί με σαφήνεια, αυτό δεν σημαίνει ότι ένα μοτίβο δεν υπάρχει.

Παρόλη τη μεγάλη δημοσιότητα που έλαβε η λογική των τυχαίων περιπάτων στο τομέα των οικονομικών και πιο συγκεκριμένα του χρηματιστηρίου όπως αναφέρθηκε παραπάνω, υπάρχουν πολλοί τομείς για τους οποίους εργαλειακά αποτελεί αναπόσπαστο κομμάτι στις μέρες μας. Στις θετικές επιστήμες είναι το απλούστερο μοντέλο για τη μελέτη πολυμερών (corporate finance institute , n.d.). Σε άλλους τομείς των μαθηματικών, ο τυχαίος περίπατος χρησιμοποιείται για τον υπολογισμό λύσεων στις εξισώσεις Laplace, για την εκτίμηση του αρμονικού εξισώσεων και για διάφορες κατασκευές που βοηθούν στην ανάλυση και την συνδυαστική λογική. Η λογική των τυχαίων περιπάτων μπορεί να υιοθετηθεί και να χρησιμοποιηθεί από οποιονδήποτε επιστημονικό κλάδο εφόσον ο χρήστης είναι γνώριμος με την λογική τους και η χρήση τους ως εργαλείο είναι εύλογη όπως στην έρευνά των James L. Cornette and Bruce S. Lieberman (2004) στον τομέα της εξελικτικής Βιολογίας με τίτλο «Random walks in the history of life». Στην επιστήμη των υπολογιστών, οι τυχαίοι περίπατοι χρησιμοποιούνται για την εκτίμηση των χαρακτηριστικών ενός ιστότοπου όπως το μέγεθός του.

2.2. Αλγόριθμος Page Rank

Η κατασκευή μιας μηχανής αναζήτησης είναι ένα δύσκολο έργο. Οι μηχανές αναζήτησης ευρετηριάζουν δεκάδες έως εκατοντάδες εκατομμύρια ιστοσελίδες που περιλαμβάνουν συγκρίσιμο αριθμό διακριτών όρων. Απαντούν σε δεκάδες εκατομμύρια ερωτήματα κάθε μέρα. Παρά τη σημασία των μηχανών αναζήτησης μεγάλης κλίμακας στο διαδίκτυο, έχουν γίνει πολύ λίγες ακαδημαϊκές έρευνες σε αυτές. Επιπλέον, λόγω της ταχείας προόδου στην τεχνολογία και τον πολλαπλασιασμό του διαδικτύου, η δημιουργία μιας μηχανής αναζήτησης ιστού σήμερα είναι πολύ διαφορετική από πριν από τρία χρόνια. Εκτός από τα προβλήματα κλιμάκωσης των παραδοσιακών τεχνικών αναζήτησης σε δεδομένα αυτού του μεγέθους, υπάρχουν νέες τεχνικές προκλήσεις που σχετίζονται με τη χρήση των πρόσθετων πληροφοριών που υπάρχουν στο υπερκείμενο για την παραγωγή καλύτερων αποτελεσμάτων αναζήτησης (Page). Το Google έχει τον πιο γνωστό αλγόριθμο κατάταξης: τον αλγόριθμο Page Rank ο οποίος παρέχει, όπως ισχυρίζεται, κορυφαία θέση κατάταξης σε σελίδες που είναι σχετικές. Ο αλγόριθμος Page Rank χρησιμοποιήθηκε και βελτιώθηκε από τους Lawrence Page και Sergey Brin. Ο αλγόριθμος Page Rank περιγράφει τη δημοτικότητα μιας σελίδας ή ενός ιστότοπου. Αυτός ο αλγόριθμος εξαρτάται από την ανάλυση συνδέσμων στην οποία η κατάταξη μια σελίδας αποφασίζεται βάση των εξερχόμενων και εισερχόμενων συνδέσμων. Αυτό σημαίνει ότι βασίζεται πλήρως στον σύνδεσμο του «www» και η Google χρησιμοποιεί αυτόν τον αλγόριθμο για αναζήτηση των ιστοσελίδων με βάση τον αριθμό των υπερσυνδέσμων όπως εξερχόμενων και εισερχόμενων .

Εισερχόμενοι σύνδεσμοι: Οι εισερχόμενοι σύνδεσμοι είναι οι σύνδεσμοι που έρχονται από άλλους ιστοτόπους στην σελίδα μας, είναι επίσης γνωστοί ως “backlinks”. Η Google θεωρεί σχετικούς συνδέσμους μόνο αυτούς που κατευθύνονται στη σελίδα μας αλλά δεν μπορούμε να ελέγξουμε αυτές τις σελίδες. Αν το περιεχόμενο της σελίδας είναι μοναδικό και πλούσιο τότε υπάρχει μεγάλη πιθανότητα να υπάρχουν εισερχόμενοι σύνδεσμοι.

Εξερχόμενοι σύνδεσμοι: είναι οι σύνδεσμοι που δείχνουν σε άλλες σελίδες πληροφορίες από την ιστοσελίδα μας και ως εκ τούτου έχουμε τον έλεγχο αυτών των συνδέσμων.

Σε περίπτωση που μία σελίδα έχει υψηλό rank, αν άλλες σελίδες με υψηλό rank συνδέονται σε αυτή μας το δείχνει η παρακάτω εξίσωση:

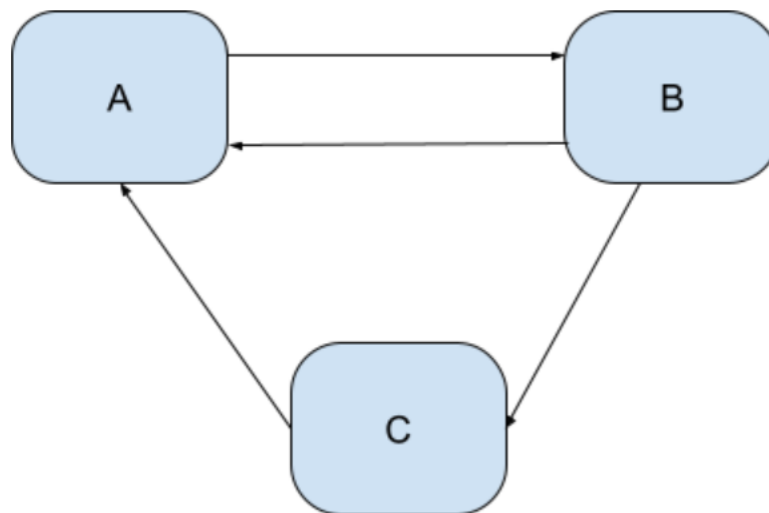
$$PR(A) = (1 - d) + dPR(T_i)/C(T_i) + \dots + PR(T_n)/C(T_n)), \text{ όπου}$$

- **A** ορίζεται ως η σελίδα της οποίας το page rank είναι $PR(A)$
- Ως **$PR(T_i)$** ορίζεται το page rank των σελίδων T_i , οι οποίες συνδέονται με την σελίδα A
- Ως **$C(T_i)$** ορίζεται ο αριθμός των εξερχόμενων συνδέσμων που εξέρχονται από την σελίδα T_i
- Ως **d** ορίζεται μια παράμετρος μεταξύ 0 και 1 συνήθως 0.85. Μερικές φορές δεν πηγαίνει σε άλλες σελίδες μέσω συνδέσμων αλλά κάνει άλμα σε άλλες σελίδες τυχαία.
- **(1-d)** είναι η πιθανότητα να κάνει άλμα σε μια σελίδα τυχαία. Κάθε σελίδα έχει ένα ελάχιστο page rank ίσο με (1-d)

Για να υπολογιστεί το page rank μιας οποιασδήποτε σελίδας πρέπει να γνωρίζουμε το page rank κάθε σελίδας που συνδέεται με αυτήν καθώς και τον αριθμό των εξερχόμενων συνδέσμων αυτής.

Παράδειγμα παρουσίασης λειτουργίας του Page Rank

Έστω τρεις ιστοσελίδες A,B και C.



Η σελίδα A περιέχει 1 εξερχόμενο σύνδεσμο προς την κατεύθυνση της σελίδας B

Η σελίδα B περιέχει 2 εξερχόμενους συνδέσμους προς την κατεύθυνση της σελίδας A και C

Η σελίδα C περιέχει 1 εξερχόμενο σύνδεσμο προς την κατεύθυνση της σελίδας A

Το αρχικό page rank κάθε σελίδας ορίζεται ως μονάδα

Το page rank για κάθε σελίδα υπολογίζεται από τις παρακάτω εξισώσεις.

$$PR(A) = 0.2 + 0.4 PR(B) + 0.8 PR(C)$$

$$PR(B) = 0.2 + 0.8 PR(A)$$

$$PR(C) = 0.2 + 0.4 PR(B)$$

Τα αποτελέσματα των παραπάνω εξισώσεων είναι:

$$PR(A) = 1.2$$

$$PR(B) = 1.0$$

$$PR(C) = 0.66$$

2.3. Αλγόριθμος RWR (Random Walk with Restart)

Ο αλγόριθμος Random Walk with Restart (RWR) είναι ένας αλγόριθμος που δίνει την εγγύτητα μεταξύ δύο κόμβων στον γράφο. Αρχικά προτάθηκε για το έργο της τμηματοποίησης εικόνας.

Μας δίνεται ένα συνδεδεμένος γράφος με πιθανότητες μετάβασης p_{ij} που υποδηλώνουν την πιθανότητα μετάβασης από κατάσταση-κόμβο j σε i . Σε έναν αλγόριθμο τυχαίου περιπάτου σε κάθε στάδιο, μπορούμε να μεταβούμε σε οποιονδήποτε από τους γειτονικούς κόμβους. Τελικά, ο στόχος είναι να υπολογίσουμε την πιο πιθανή τοποθεσία που πρόκειται να βρεθούμε.

Η κύρια διαφορά στον RWR είναι ότι σε κάθε στάδιο υπάρχει πιθανότητα να μπορούμε να κάνουμε επανεκκίνηση, δηλαδή να πάμε στον αρχικό κόμβο.

Συνήθως, πρέπει να βρούμε μια λύση στην εξίσωση:

$$\mathbf{r} = c\mathbf{W}\mathbf{r} + (1 - c)\mathbf{e}, \text{ όπου:}$$

το c είναι ένας αριθμός στο εύρος $(0,1)$,

το $\mathbf{W}$ είναι ο πίνακας των πιθανοτήτων μετάβασης όπου το $W[i, j]$ υποδηλώνει την πιθανότητα μετάβασης από το j στο i ,

το $\mathbf{e}$ είναι το αρχικό διάνυσμα δηλαδή $e[i] = 1$, εάν i είναι ο κόμβος εκκίνησης αλλιώς ισούται με το 0 και

το $\mathbf{r}$ είναι ένας φορέας στήλης όπου $r[i]$ υποδηλώνει την πιθανότητα να βρίσκεται στον κόμβο i .

Θα μπορούσε κάποιος να συσχετίσει τον τυχαίο περίπατο με ένα μεθυσμένο άτομο που περπατά σε ένα πεδίο όπου μπορεί να κάνει ένα βήμα προς κάθε κατεύθυνση με μια σταθερή πιθανότητα, ενώ το RWR μπορεί να θεωρηθεί ως το ίδιο μεθυσμένο άτομο σε ένα πεδίο όπου υπάρχει πεπερασμένη πιθανότητα στο κάθε βήμα που μπορεί να αγγίζει ένα «κλειδί τηλεμεταφοράς» που θα τον τηλεμεταφέρει πίσω στην αρχική θέση του πεδίου.

Ας υποθέσουμε ότι βρισκόμαστε ακριβώς στον κόμβο εκκίνησης e . Στη συνέχεια, η πιθανότητα r να βρίσκεστε σε οποιονδήποτε κόμβο i στην επόμενη κίνηση θα δοθεί από τον τύπο:

$$\mathbf{r}_0[i] = \mathbf{W}[i] \cdot \mathbf{e}, \text{ όπου:}$$

το $\mathbf{W}[i]$ αντιπροσωπεύει την i οστή σειρά του πίνακα $\mathbf{W}$, δηλαδή τις πιθανότητες μετάβασης στον κόμβο i από όλους τους κόμβους.

Στην επόμενη κίνηση η πιθανότητα να βρισκόμαστε στον κόμβο i θα είναι:

$$\mathbf{r}_1[i] = \mathbf{W}[i] \cdot \mathbf{r}_0$$

Λαμβάνοντας τώρα υπόψη την επανεκκίνηση:

$$\mathbf{r}_k[i] = c \cdot \mathbf{W}[i] \cdot \mathbf{r}_{k-1} + (1 - c)\mathbf{e}$$

Επομένως, ο στόχος είναι να βρούμε την τιμή του $\mathbf{r}_k$ έτσι ώστε οι περαιτέρω προσομοιώσεις να μην επηρεάζουν τις πιθανότητες. Υποθέτουμε λοιπόν ότι το $\mathbf{r}$ έχει συγκλίνει δηλ. $\mathbf{R}_{k-1} = \mathbf{r}_k$. Έτσι μπορούμε να γράψουμε την παραπάνω εξίσωση ως εξής:

$$\mathbf{r} = c\mathbf{W}\mathbf{r} + (1 - c)\mathbf{e}$$

Οι παραπάνω εξισώσεις μπορεί να μοιάζουν πολύ με την εξίσωση του PageRank. Στην πραγματικότητα, η μόνη διαφορά μεταξύ του PageRank και του RWR είναι ότι γενικά στην κατάταξη σελίδας η πιθανότητα μετάβασης σε άλλη σελίδα θεωρείται ομοιόμορφη, δηλαδή

αντικαθιστούμε το e με ένα διάνυσμα στήλης όπου όλα τα στοιχεία της είναι μονάδα. Σε σύγκριση με το PageRank, το RWR δεν αναφέρει στον τρόπο κατασκευής του πίνακα W .

Παραδοσιακά, δύο λύσεις έχουν χρησιμοποιηθεί για την επίλυση προβλημάτων που σχετίζονται με το RWR.

α. Επανάληψη έως τη σύγκλιση

Η λύση περιλαμβάνει τον υπολογισμό της τιμής του διανύσματος r χρησιμοποιώντας επαναληπτικά τον παρακάτω τύπο μέχρι να φτάσουμε στο σημείο σύγκλισης ή σε συγκεκριμένο αριθμό επαναλήψεων:

$$r_k[i] = c \cdot W[i] \cdot r_{k-1} + (1 - c) e$$

Επομένως, η λύση δεν απαιτεί πρότερους υπολογισμούς και συνήθως φτάνουμε σε σύγκλιση εντός ενός πολύ μικρού αριθμού επαναλήψεων.

Για παράδειγμα, στην πρότυπη εργασία για το PageRank (Page), αναφέρθηκε ότι ο αλγόριθμος PageRank για ένα δίκτυο που αποτελείται από 322 εκατομμύρια συνδέσμους (in-edges και out-edges), συγκλίνει εντός ενός ανεκτού ορίου 52 επαναλήψεων. Ωστόσο, ο αριθμός μπορεί να είναι μεγαλύτερος για ένα γράφο με χαμηλή συνδεσιμότητα και μεγάλο αριθμό κόμβων. Ο απαιτούμενος χρόνος έχει γραμμική σχέση με τον αριθμό των επαναλήψεων και τετραγωνική σχέση με τον αριθμό των κόμβων στον γράφο (λόγω του πολλαπλασιασμού πίνακα-διανύσματος).

β. Χρησιμοποιώντας τον αντίστροφο του πίνακα

Η εξίσωση που προέκυψε παραπάνω μπορεί να ξαναγραφτεί βάζοντας τον όρο με r στο αριστερό μέρος της εξίσωσης. Έτσι παίρνουμε:

$$(I - cW)r = (1 - c)e$$

που μας δίνει:

$$r = (1 - c)(I - cW)^{-1}e$$

Σε αυτή τη μέθοδο, το χρονοβόρο βήμα και ο εντατικός υπολογισμός είναι να βρούμε τον αντίστροφο του πίνακα που έχει πολυπλοκότητα $O(n^3)$ χρησιμοποιώντας την απαλοιφή Gauss. Η συνολική πολυπλοκότητα καθορίζεται από την πολυπλοκότητα του αντίστροφου πίνακα καθώς ο

πολλαπλασιασμός πίνακα-διανύσματος έχει την πολυπλοκότητα $O(n^2)$. Ένα άλλο σημαντικό πράγμα που πρέπει να σημειωθεί είναι ότι εδώ με τον υπολογισμό του πίνακα $(I - cW)^{-1}$ μία φορά μπορούμε να πάρουμε τις απαιτούμενες πιθανότητες για οποιονδήποτε κόμβο εκκίνησης για το δεδομένο γράφο. Αυτό είναι ιδιαίτερα χρήσιμο εάν πρέπει να εκτελέσουμε αρκετές τυχαίες διαδρομές στον ίδιο γράφο με διαφορετικούς κόμβους εκκίνησης (π.χ. σε εργασία τμηματοποίησης εικόνας χρησιμοποιώντας τον αλγόριθμο τυχαίου περιπάτου). Ωστόσο, απαιτεί την αποθήκευση του πίνακα $(I - cW)^{-1}$ που απαιτεί χώρο της τάξης $O(n^2)$.

Υπάρχουν επίσης ορισμένες βελτιστοποιήσεις που μπορούν να μειώσουν το χώρο που απαιτείται για την αποθήκευση του αντίστροφου πίνακα, υποθέτοντας ότι οι περισσότερες από τις τιμές στον πίνακα συσχετίζονται (Tong et al., 2006)

Ένα σημαντικό πράγμα που πρέπει να παρατηρηθεί είναι ότι η αρχική τιμή του διανύσματος r δεν επηρεάζει τις πιθανότητες που έχουμε για τη σύγκλιση όσο ο γράφος είναι διασυνδεδεμένος. Αυτό φαίνεται από τον τύπο που παίρνουμε χρησιμοποιώντας τη μέθοδο αντίστροφου πίνακα για RWR.

Οι πιο συχνές εφαρμογές του RWR είναι οι εξής:

I. Τμηματοποίηση εικόνας

Αρχικά ο αλγόριθμος RWR προτάθηκε για τμηματοποίηση εικόνας. Η εικόνα μετατρέπεται σε γράφο όπου το βάρος της ακμής μεταξύ των γειτονικών pixel αντιπροσωπεύει την ομοιότητα μεταξύ των δύο pixel. Ο τρόπος που λειτουργεί είναι ότι απελευθερώνουμε πολλούς τυχαίους περιπάτους από ορισμένες θέσεις στην εικόνα (επιλέγονται από τον χρήστη ή χρησιμοποιώντας μια αυτοματοποιημένη διαδικασία). Αυτό μας δίνει μια βαθμολογία ως προς το πόσο κοντά βρίσκεται κάθε pixel στο δεδομένο pixel. Στα pixel/κόμβους για τους οποίους ο αλγόριθμος αποδίδει υψηλότερη τιμή για μια δεδομένη τυχαία διαδρομή, διορίζονται στην ίδια ομάδα με τον κόμβο εκκίνησης αυτής της τυχαίας διαδρομής.

II. Αυτόματη λεζάντα εικόνας

Το RWR μπορεί να χρησιμοποιηθεί για την αυτόματη δημιουργία λεζάντων για μια εικόνα. Ο

τρόπος με τον οποίο λειτουργεί είναι ότι πρώτα δημιουργείται ένας γράφος χρησιμοποιώντας το σύνολο εικόνων που έχουν λεζάντα και των εικόνων που επιθυμούμε να τοποθετηθεί λεζάντα. Στη συνέχεια, για κάθε λέξη της λεζάντας εκτιμά την εγγύτητα με την εικόνα χρησιμοποιώντας τον RWR. Ωστόσο, λόγω της αύξησης των τεχνικών στην βαθιά μάθηση, αυτή η μέθοδος δεν χρησιμοποιείται πολύ στις μέρες μας.

III. Κοινοτική ανίχνευση

Οι κοινοτικές μέθοδοι ανίχνευσης συγκεντρώνουν γενικά τους κόμβους του γράφου βάση κάποιας συνάρτησης που ερμηνεύει την απόσταση. Δεδομένου ότι το RWR δίνει την εγγύτητα των κόμβων στον γράφο, μπορεί να χρησιμοποιηθεί ως απόσταση μεταξύ ενός ζεύγους κόμβων στο γράφημα.

IV. Συστήματα συστάσεων

Το RWR δίνει την εγγύτητα διαφορετικών κόμβων στο γράφο. Αυτή η ιδιότητα είναι φυσικά κατάλληλη για συστήματα σύστασης. Το Twitter χρησιμοποιεί μια παρόμοια μέθοδο για να προτείνει στους χρήστες να ακολουθούν άλλους χρήστες.

2.4. Αλγόριθμος B_LIN

Ο καθορισμός της βαθμολογίας συνάφειας μεταξύ δύο κόμβων είναι ένα από τα θεμελιώδη δομικά στοιχεία στην εξόρυξη δεδομένων από γράφους. Μια πολύ επιτυχημένη τεχνική βασίζεται σε τυχαίους περιπάτους με επανεκκίνηση (RWR). Ο RWR δέχεται αυξανόμενο ενδιαφέρον τόσο στην εφαρμογή του όσο και στη θεωρία. Μια σημαντική ερευνητική πρόκληση είναι η ταχύτητα του αλγορίθμου, ειδικά σε γράφους μεγάλων διαστάσεων.

Το B_LIN (Pan, Random walk with restart: fast solutions and applications, 2007), εκμεταλλεύεται δύο ιδιότητες που μοιράζονται πολλοί πραγματικοί γράφοι: (α) την μπλοκ, κοινοτική δομή και (β) τις γραμμικές συσχετίσεις μεταξύ σειρών και στηλών του πίνακα γειτνίασης. Η προτεινόμενη μέθοδος εξισορροπεί προσεκτικά το κόστος προεπεξεργασίας εκτός σύνδεσης (τόσο το κόστος CPU όσο και το κόστος αποθήκευσης), με την ποιότητα απόκρισης (σε σχέση με την ακρίβεια και τον χρόνο απόκρισης). Σε σύγκριση με τη μέθοδο PreCompute, απαιτεί μόνο προ-υπολογισμό και αποθήκευση της προσέγγισης χαμηλής τάξης ενός μεγάλου αλλά αραιού πίνακα και την αντιστροφή ορισμένων πινάκων μικρού μεγέθους. Σε σύγκριση με τη μέθοδο OnTheFly,

χρειάζεται μόνο λίγες πράξεις πολλαπλασιασμού πίνακα - διανύσματος με διαδικασία απόκρισης αυτήν ενός on-line δικτύου.

Παρακάτω δίνεται παράδειγμα χρήσης του αλγόριθμου B_LIN από την έρευνα «Random walk with restart: fast solutions and applications», Tong et. al. 2007.

B_LIN
Είσοδος: Ο σταθμισμένος κανονικοποιημένος πίνακας W και το αρχικό διάνυσμα e_i
Έξοδος: Το διάνυσμα κατάταξης r_i
Στάδιο προ-υπολογισμού (Off-Line):
1. Χώρισμα του γραφήματος σε k κομμάτια
2. Ανάλυση του πίνακα W σε δύο πίνακες $\mathbf{W}=\mathbf{W}_1+\mathbf{W}_2$ σύμφωνα με το αποτέλεσμα του χωρίσματος, όπου ο $\mathbf{W}_1$ περιέχει όλους τους «within-partition» συνδέσμους και ο $\mathbf{W}_2$ περιέχει όλους τους cross-partition συνδέσμους
3. Έστω $\mathbf{W}_1$, το ισοτό κομμάτι, ορίζουμε το $\mathbf{W}_1$ ως την εξίσωση (1)
4. Υπολογίζουμε και αποθηκεύουμε $Q_{1,i}^{-1} = (I - c\tilde{W}_{1,i})^{-1}$ για κάθε κομμάτι i
5. Εφαρμόζομαι low-rank προσέγγιση για τον $\mathbf{W}_2=USV$
6. Ορίζουμε το Q_1^{-1} ως εξίσωση (2). Υπολογίζουμε και αποθηκεύουμε το:
$\tilde{A} = S^{-1} - cVQ_1^{-1}U)^{-1}$
Query Stage (On-Line):
1. Έξοδος $\vec{r}_i = (1 - c)(Q_1^{-1}\vec{e}_i + cQ_1^{-1} - U\tilde{A}VQ_1^{-1}\vec{e}_i)$

$$\tilde{W}_1 = \begin{pmatrix} \tilde{W}_{1,1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \tilde{W}_{1,2} & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots \\ \mathbf{0} & \dots & \mathbf{0} & \tilde{W}_{1,k} \end{pmatrix} \quad (1)$$

$$Q_1^{-1} = \begin{pmatrix} Q_{1,1}^{-1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & Q_{1,2}^{-1} & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots \\ \mathbf{0} & \dots & \mathbf{0} & Q_{1,k}^{-1} \end{pmatrix} \quad (2)$$

2.5. Αλγόριθμος LRW (Limited Random Walk)

Ένας αλγόριθμος ομαδοποίησης γράφων με βάση τους τυχαίους περιπάτους είναι ο αλγόριθμος περιορισμένης τυχαίας διαδρομής (LRW). Πρώτα απ' όλα, ο αλγόριθμος LRW δεν βασίζεται σε καμία συνάρτηση προσαρμογής clustering (fitness function). Η μέθοδος μπορεί να αντιμετωπίσει αποτελεσματικά την υπολογιστική πολυπλοκότητα. Ως μοναδική ιδιότητα μεταξύ πολλών μεθόδων ομαδοποίησης γράφων, το LRW μπορεί να προσαρμοστεί τόσο σε ολική όσο και σε τοπική ομαδοποίηση γράφων με αποτελεσματικό τρόπο.

Η διαδικασία της τυχαίας πορείας που ακολουθεί ο αλγόριθμος είναι ισοδύναμη με μια διακριτή διαδικασία στάσιμης αλυσίδας Markov. Κάθε κορυφή αντιστοιχεί σε μια κατάσταση στην αλυσίδα Markov και κάθε άκρο δείχνει μια πιθανή μετάβαση μεταξύ των δύο καταστάσεων. Ο πίνακας μετάβασης Markov [P] μπορεί να ληφθεί κανονικοποιώντας τον πίνακα γειτνίασης ώστε κάθε στήλη να έχει άθροισμα έως 1, π.χ.

$$P_{ij} = \frac{A_{kj}}{\sum_{k=1}^n A_{kj}} \quad (1)$$

ή

$$P = AD^{-1} \quad (2)$$

Μπορούν επίσης να χρησιμοποιηθούν και άλλες μορφές του πίνακα μετάβασης P, για παράδειγμα ο «τεμπέλης» τυχαίος πίνακας χρησιμοποιεί τον πίνακα μετάβασης

$$P = 1/2 (I + AD^{-1}), \text{ όπου}$$

I είναι ο πίνακας ταυτότητας. Δεδομένου του μεταβατικού πίνακα P, μπορούμε να υπολογίσουμε $x(t+1)$ από $x(t)$ χρησιμοποιώντας την εξίσωση:

$$x^{(t+1)} = Px^{(t)}.$$

Μία κλειστή διαδρομή είναι ένας περίπατος σε ένα γράφο όπου η τελική κορυφή είναι ίδια με την αρχική κορυφή. Η περίοδος μιας κορυφής ορίζεται ως ο μέγιστος κοινός διαιρέτης των μηκών όλων των κλειστών διαδρομών που ξεκινούν από αυτήν την κορυφή. Λέμε ότι ένα γράφημα είναι περιοδικό εάν όλες οι κορυφές του έχουν περιόδους 1.

Για ένα μη κατευθυνόμενο, συνδεδεμένο και περιοδικό γράφο, υπάρχει μια κατάσταση ισορροπίας π , έτσι ώστε $\pi = P\pi$. Αυτή η κατάσταση είναι μοναδική και άσχετη με το σημείο εκκίνησης. Με επανάληψη της εξίσωσης (3), το $x(t)$ συγκλίνει σε π .

Καθορίζουμε πρώτα τον πίνακα μετάβασης P . Αναθέτουμε την ίδια πιθανότητα στη μετάβαση ώστε ο agent που βαδίζει να παραμένει στην τρέχουσα κορυφή και για τη μετάβαση του σε οποιαδήποτε γειτονική κορυφή. Προσθέτουμε έναν πίνακα ταυτοτήτων στον πίνακα γειτνίασης και στη συνέχεια κανονικοποιείται το αποτέλεσμα ώστε κάθε άθροισμα στήλης να είναι 1. Ο πίνακας μετάβασης μπορεί να γραφτεί ως εξής:

$$P = (I + A)(I + D)^{-1}. \quad (4)$$

Σε σύγκριση με τον πίνακα μετάβασης στην εξίσωση (2), αυτή η διαδικασία είναι παρόμοια με την προσθήκη αυτό-επαναλήψεων στο γράφο, αλλά ταυτόχρονα αυξάνοντας τον βαθμό κάθε κορυφής κατά 1 αντί για 2. Αυτή η τροποποίηση διορθώνει το πρόβλημα περιοδικότητας που μπορεί να έχει ο γράφος. Βελτιώνει σημαντικά τη σταθερότητα και την ακρίβεια του αλγορίθμου στην ομαδοποίηση γραφών. Σε κάθε βήμα, ο φορέας πιθανότητας $x(t)$ υπολογίζεται χρησιμοποιώντας την εξίσωση (3). Σημειώνεται ότι, γενικά, τα στοιχεία στο $x(t)$ που βρίσκονται γύρω από την κορυφή της αρχικής κορυφής είναι μη μηδενικά και τα υπόλοιπα είναι μηδενικά. Επομένως, δεν χρειαζόμαστε τον πλήρη πίνακα μετάβασης για τον υπολογισμό του διανύσματος πιθανότητας για το επόμενο βήμα. Ξεκινώντας από την αρχική κορυφή, ένας τυχαίος περίπατος θα εξερευνήσει τελικά ολόκληρο το γράφο. Για να αποκαλυφθεί μια τοπική δομή γραφών, μπορούν να χρησιμοποιηθούν διαφορετικές τεχνικές για τον περιορισμό του πεδίου των περιπάτων. Οι Harel et. al. 2001 καθορίζουν τον αριθμό των βημάτων από μια προκαθορισμένη σταθερά. Οι Xin et al. 2016 χρησιμοποιούν μια στοχαστική μέθοδο για να προσδιορίσουν εάν

πρέπει να συνεχιστεί ένας περίπατος και να ορίσουν τον μέγιστο αριθμό βημάτων σε 6 σύμφωνα με την αρχή των «έξι βαθμούς διαχωρισμού» .

Η διαδικασία περιορισμένου τυχαίου περιπάτου (LRW) σε σύγκριση με την κανονική διαδικασία τυχαίου περιπάτου περιλαμβάνει διαδικασίες στρογγυλοποίησης και κανονικοποίησης σε κάθε βήμα του περιπάτου. Αυτές οι μη γραμμικές λειτουργίες περιορίζουν τον agent να περπατά γύρω από τη γειτονιά της αρχικής κορυφής, ειδικά εάν υπάρχει ένα σαφές όριο ομαδοποίησης γράφων.

Ο αλγόριθμος MCL (Markov clustering algorithm) προσομοιώνει τη ροή μέσα σε ένα γράφο. Χρησιμοποιεί τη λειτουργία στρογγυλοποίησης και κανονικοποίησης για να ενισχύσει τη ροή μέσα σε ένα σύμπλεγμα και να μειώσει τη ροή μεταξύ των ομαδοποιήσεων. Η διαδικασία MCL είναι μια ανομοιογενής αλυσίδα Markov στην οποία ο μεταβατικός πίνακας μεταβάλλεται με την πάροδο του χρόνου. Ο αλγόριθμος MCL ξεκινά τον τυχαίο περίπατο από όλες τις κορυφές ταυτόχρονα - υπάρχουν n agents που «περπατούν» στο γράφο ταυτόχρονα. Ο περίπατος μπορεί να συνεχιστεί μόνο όταν όλοι οι agents έχουν ολοκληρώσει ένα βήμα περιπάτου και ο πίνακας πιθανότητας αποτελέσματος έχει διογκωθεί και κανονικοποιηθεί. Σε αντίθεση με τον αλγόριθμο MCL, η διαδικασία LRW είναι μια χρονικά ομοιογενής αλυσίδα Markov. Ξεκινάμε με μία τυχαία διαδρομή από την μία κορυφή και κάνουμε την στρογγυλοποίηση στις τιμές πιθανότητας αυτού του agent. Αυτός ο σχεδιασμός έχει πολλά πλεονεκτήματα. Πρώτον, αποφεύγει περιττούς περιπάτους αφού η δομή του γράφου εξελίσσεται γύρω από την κορυφή η οποία μπορεί να ευρεθεί σε μόλις μία διαδρομή. Δεύτερον, η διαδικασία είναι κατάλληλη για τα τοπικά προβλήματα ομαδοποίησης επειδή δεν απαιτεί ολόκληρα τα δεδομένα γράφου. Τρίτον, εάν απαιτούνται πολλαπλοί περίπατοι, κάθε διαδικασία περιπάτου μπορεί να εκτελεστεί ανεξάρτητα. Έτσι ο αλγόριθμος είναι πλήρως παραλληλοποιήσιμος.

2.6. Αλγόριθμος K_DASH

Μία από τις πιο επιτυχημένες τεχνικές που είναι γνωστές στις ακαδημαϊκές κοινότητες βασίζεται σε τυχαίο περίπατο με επανεκκίνηση (RWR) που αναλύθηκε νωρίτερα. Αυτό συμβαίνει επειδή η εγγύτητα που ορίζεται από το αλγόριθμο RWR αποφέρει τα ακόλουθα οφέλη: καταγράφει την ολική δομή του γραφου και, καταγράφει σχέσεις πολλαπλών όψεων μεταξύ δύο κόμβων σε αντίθεση με τις παραδοσιακές αποστάσεις γραφημάτων.

Ωστόσο, ο υπολογισμός των αποστάσεων από το RWR είναι υπολογιστικά «ακριβός».

Εξετάζοντας ένα τυχαίο σωματίδιο που ξεκινά από τον κόμβο αναζήτησης q , το σωματίδιο αυτό μετακινείται επαναληπτικά στους γειτονικούς κόμβους με πιθανότητα ανάλογη των βαρών των κορυφών. Επιπλέον, σε κάθε βήμα υπάρχει η πιθανότητα επανεκκίνησης από τον κόμβο q . Η πιθανότητα κατάληξης σε κάποιον κόμβο αλλάζει με την πάροδο του χρόνου κατά τη διάρκεια επαναλήψεων εφαρμόζοντας ανακυκλικά τις παραπάνω διαδικασίες. Ως αποτέλεσμα, μπορεί να επιτευχθεί η πιθανότητα μιας σταθερής κατάστασης. Η απόσταση του κόμβου u σε σχέση με τον κόμβο q ορίζεται ως η πιθανότητα αυτής της σταθερής κατάστασης με την οποία το σωματίδιο παραμένει στον κόμβο u . Αν και το RWR δέχεται συνεχώς αυξανόμενο ενδιαφέρον από πολλές εφαρμογές, ο υπερβολικός χρόνος ανάγκης CPU έχει οδηγήσει στην εισαγωγή άλλων προσεγγίσεων. Αυτές οι προσεγγίσεις έχουν το πλεονέκτημα της ταχύτητας σε βάρος της ακρίβειας. Ωστόσο, οι κατά προσέγγιση αλγόριθμοι δεν έχουν υιοθετηθεί βέλτιστα από αυτές τις προσεγγίσεις.. Αυτό συμβαίνει επειδή είναι δύσκολο για τους αλγόριθμους αυτούς να βελτιώσουν την ποιότητα των πραγματικών εφαρμογών. Επομένως, μία πρόταση είναι να αντιμετωπιστεί το πρόβλημα ως εξής :

Ορίζεται ο κόμβος ερωτήματος q και ο απαιτούμενος αριθμός κόμβων απάντησης K . Οι κόμβοι top- K είναι οι κόμβοι με τις μικρότερες αποστάσεις, υπολογιζόμενες με ακρίβεια, σε σχέση με τον κόμβο q .

Η προσέγγισή στην εύρεση των κόμβων (top- k) στον αλγόριθμο RWR είναι η πρώτη λύση που επιτυγχάνει ταυτόχρονα ακρίβεια και αποτελεσματικότητα. Η νέα αυτή μέθοδος που ονομάζεται K-dash μπορεί να εντοπίσει αποτελεσματικά τους κόμβους top- k στον αλγόριθμο RWR προκειμένου να μειωθεί το κόστος αναζήτησης με τον εξής τρόπο: χρησιμοποιούνται αραιοί πίνακες για τον υπολογισμό της ακριβούς απόστασης ενός επιλεγμένου κόμβου και, γίνεται «κλάδεμα» των κόμβων μεγάλης απόστασης εκτιμώντας, και όχι υπολογίζοντας ακριβώς, τις αποστάσεις αυτές.

Η μέθοδος K-dash έχει τα ακόλουθα ελκυστικά χαρακτηριστικά με βάση τις παραπάνω ιδέες:

- **Ακρίβεια:** Η μέθοδος K-dash δεν θυσιάζει την ακρίβεια παρότι εκμεταλλεύεται μια προσέγγιση βάσει εκτιμήσεων για να κλαδεύονται οι απίθανοι κόμβοι. Επιστρέφει, κατά προσέγγιση, κόμβους top- k χωρίς σφάλμα σε αντίθεση με τις προηγούμενες προσεγγίσεις.
- **Αποδοτικότητα:** Η μέθοδος K-dash απαιτεί πρακτικά χρόνο $O(n + m)$ όπου n και m είναι ο αριθμός κόμβων και ακμών, αντίστοιχα. Συγκριτικά, οι λύσεις που βασίζονται σε

υπάρχοντες κατά προσέγγιση αλγόριθμους είναι «ακριβές» καθώς χρειάζονται $O(n^2)$ ώρα για να βρουν τους κόμβους απάντησης. Σημειώνεται ότι πρακτικά $m \ll n^2$, M. Newman 2006.

- **Επιδεξιότητα:** Η μέθοδος K-dash χρειάζεται πρακτικά χώρο $O(n + m)$ ενώ οι προηγούμενες προσεγγίσεις χρειάζονται $O(n^2)$ χώρο. Ο απαιτούμενος χώρος μνήμης της μεθόδου K-dash είναι μικρότερος από αυτό των προηγούμενων προσεγγίσεων.
- **Αποδεσμευμένη από παραμέτρους:** Οι προηγούμενες προσεγγίσεις απαιτούν προσεκτική ρύθμιση της εσωτερικής παραμέτρου (H. Tong 2006), δεδομένο που επηρεάζει τα αποτελέσματα αναζήτησης. Η μέθοδος K-dash, ωστόσο, είναι εντελώς αυτόματη. Αυτό σημαίνει ότι δεν απαιτεί από το χρήστη να ορίσει τυχόν εσωτερικές παραμέτρους.

Με τον τρόπο αυτό, Η μέθοδος K-dash θα επιτρέψει στο μέλλον να αναπτυχθούν περισσότερες εφαρμογές βασισμένες στον αλγόριθμο RWR.

2.7. Αλγόριθμος Node2vec

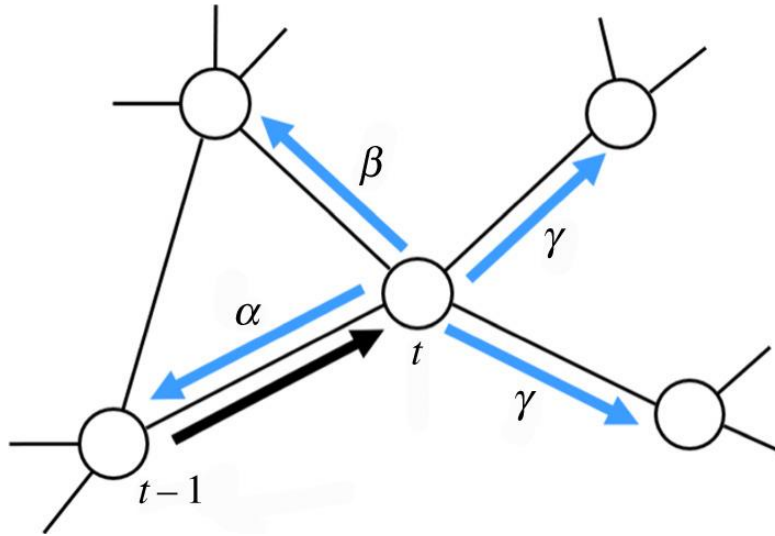
Πολλές θεωρητικές, υπολογιστικές και αλγοριθμικές μελέτες έχουν χρησιμοποιήσει απλούς τυχαίους περιπάτους σε μη σταθμισμένα δίκτυα, κάτι που εξ ορισμού υπαγορεύει ότι ένας agent μετακινείται σε έναν από τους γείτονές του με την ίδια πιθανότητα σε κάθε χρονικό βήμα. Ωστόσο, υπάρχουν επίσης διάφοροι άλλοι τύποι τυχαίων περιπάτων, πολλοί από τους οποίους έχουν τροφοδοτηθεί με αλγόριθμους τυχαίων περιπάτων. Οι τυχαίοι περίπατοι που αναπτύχθηκαν για το αλγοριθμικό πλαίσιο που ονομάζεται node2vec είναι ένας τέτοιος τυχαίος περίπατος. Σε αντίθεση με τους απλούς τυχαίους περιπάτους, οι μεταβάσεις των τυχαίων βημάτων στο node2vec δεν εξαρτώνται μόνο από το βαθμό του κόμβου που επισκέπτεται αυτήν τη στιγμή ή την παραλλαγή του με βάρη ακμής, αλλά και από τη δομή του τοπικού δικτύου και του κόμβου τελευταίας επίσκεψης. Οι Grover και Leskovec πρότειναν το node2vec για κλιμάκωση της εκμάθησης χαρακτηριστικών σε δίκτυα, το οποίο μπορεί να χρησιμοποιηθεί σε εργασίες όπως η ανίχνευση κοινότητας, η ταξινόμηση με πολλές ετικέτες και η πρόβλεψη συνδέσμων. Στο node2vec, μπορεί κανείς να συντονίσει το βάρος της τοπικής και της καθολικής αναζήτησης του δικτύου διαμορφώνοντας τις τιμές παραμέτρων. Το node2vec έχει βρει εφαρμογές, για παράδειγμα, για την πρόβλεψη γονιδίων που σχετίζονται με τη νόσο του Πάρκινσον. (Masuda, 2020) Μέχρι σήμερα, δεν είναι πολλά γνωστά για τη συμπεριφορά των τυχαίων περιπάτων node2vec.

Σημειώνεται εδώ, ότι μεταξύ των διαφόρων ιδιοτήτων των τυχαίων περιπάτων, η στατική πιθανότητα παίζει βασικό ρόλο στην κατάταξη των κόμβων και ο χρόνος χαλάρωσης επηρεάζει, για παράδειγμα, το ποσοστό σύγκλισης των αλγορίθμων τυχαίας πορείας και την ποιότητα της δομής της κοινότητας.

Δεδομένου ενός κόμβου πηγής u , προσομοιώνουμε ένα τυχαίο περίπατο με σταθερό μήκος L . Αφήνεται το c_i να δηλώσει τον ισοτό κόμβο στον περίπατο, ξεκινώντας με $c_0 = u$. Οι κόμβοι c_i δημιουργούνται από την ακόλουθη κατανομή:

$$P(c_i = x \mid c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{Z} & \text{if } (v, x) \in E \\ 0 & \text{otherwise} \end{cases}$$

Η μεταβλητή α αντιπροσωπεύει την τάση για τυχαίο περπάτημα προς τα πίσω, η β το βάρος της επίτευξης ένωσης με ένα κοινό γείτονα του κόμβου που επισκέπτεται αυτήν τη στιγμή και του κόμβου που επισκέφτηκε στο τελευταίο βήμα, και η γ το βάρος της εξερεύνησης σύνδεσης με οποιονδήποτε άλλο γείτονα. Μια μεγάλη τιμή β συνεπάγεται μία κατά προσέγγιση δειγματοληψία πλάτους και μια μεγάλη τιμή γ υποδηλώνει μια κατά προσέγγιση δειγματοληψία βάθους. Εάν $\alpha = \beta = \gamma \neq 0$, η τυχαία βόλτα `node2vec` μειώνεται σε μια απλή τυχαία βόλτα. Εάν $\alpha = 0$ και $\beta = \gamma \neq 0$, ο τυχαίος περίπατος `node2vec` είναι ένας τυχαίος περίπατος χωρίς οπισθοδρόμηση. Πιθανές μεταβάσεις σε ένα βήμα του τυχαίου περιπάτου `node2vec` παρουσιάζονται σχηματικά στο παρακάτω σχήμα.



Σχηματικά ο τυχαίος περίπατος *node2vec*. Υποθέτουμε ότι το δίκτυο είναι μη σταθμισμένο. Η πιθανότητα μετάβασης σε έναν από τους τέσσερις γείτονες τη στιγμή t σε αυτό το παράδειγμα δίνεται από την σχέση $(\alpha + \beta + 2\gamma)$, $\beta/(\alpha + \beta + 2\gamma)$ ή $\gamma/(\alpha + \beta + 2\gamma)$

Κεφάλαιο 3. Ανάλυση των επιλεχθέντων δεδομένων

3.1. Η γλώσσα Python

Η Python είναι διερμηνευόμενη (interpreted), γενικού σκοπού (general-purpose) και υψηλού επιπέδου, γλώσσα προγραμματισμού. Ανήκει στις γλώσσες προστακτικού προγραμματισμού (Imperative programming) και υποστηρίζει τόσο το διαδικαστικό (procedural programming) όσο και το αντικειμενοστραφές (object-oriented programming) προγραμματιστικό υπόδειγμα. Δημιουργήθηκε από τον Ολλανδό Γκίντο βαν Ρόσσουμ (Guido van Rossum) στο ερευνητικό κέντρο Centrum Wiskunde & Informatica (CWI) το 1989 και κυκλοφόρησε για πρώτη φορά το 1991. Ο κύριος στόχος της είναι η αναγνωσιμότητα του κώδικά της και η ευκολία χρήσης της. Το συντακτικό της επιτρέπει στους προγραμματιστές να εκφράσουν έννοιες σε λιγότερες γραμμές κώδικα από ότι θα ήταν δυνατόν σε γλώσσες όπως η C++ ή η Java. Διακρίνεται λόγω του ότι έχει πολλές βιβλιοθήκες που διευκολύνουν ιδιαίτερα αρκετές συνηθισμένες εργασίες και για την ταχύτητα εκμάθησής της. Μειονεκτεί στο ότι επειδή είναι διερμηνευόμενη είναι πιο αργή από τις μεταγλωττιζόμενες (compiled) γλώσσες όπως η C και η C++. Για αυτόν τον λόγο δεν είναι κατάλληλη για γραφή λειτουργικών συστημάτων. Οι διερμηνευτές της Python είναι διαθέσιμοι για εγκατάσταση σε πολλά λειτουργικά συστήματα, επιτρέποντας στην Python την εκτέλεση κώδικα σε ευρεία γκάμα συστημάτων. Η Python αναπτύσσεται ως ανοιχτό λογισμικό (open source) και η διαχείρισή της γίνεται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation.

Στα μέσα της δεκαετίας του 2000, υπήρξε μια εκθετική αύξηση του όγκου δεδομένων που οι εταιρείες αποθήκευαν, παρακολουθούσαν και διαχειριζόντουσαν. Επιπλέον, από το 2005 και μετά, οι εταιρείες κοινωνικών μέσων δικτύωσης όπως το Facebook, το Instagram, το YouTube και το Twitter είχαν επίσης γίνει δημοφιλείς. Λόγω της ευκολίας χρήσης της Python, αυτές οι νέες εταιρείες την βρήκαν ως την πιο οικονομική λύση για την πρόωμη ανάπτυξη, με αποτέλεσμα να γίνει η γλώσσα επιλογής για τη διαχείριση δεδομένων κοινωνικών μέσων δικτύωσης. Μετά την οικονομική κρίση του 2008, , αυτές οι εταιρίες επέλεξαν να αντικαταστήσουν στοιχεία της «ανθρώπινης» παρέμβασης στις ροές εργασίας τους με ισοδύναμα τεχνητής νοημοσύνης, με αποτέλεσμα καθαρότερους τρόπους για τη διαχείριση των τεράστιων όγκων δεδομένων που παράγει καθημερινά ο χρηματοοικονομικός κόσμος.

Με τα δεδομένα να πολλαπλασιάζονται, η μηχανική μάθηση και η επιστήμη των δεδομένων έγινε σύντομα η νέα «καυτή» καριέρα για προγραμματιστές. Μαζί με αυτήν τη μετατόπιση της ζήτησης από τον προγραμματισμό λογισμικού στην επιστήμη των δεδομένων προέκυψε η ανάγκη ενοποίησης εργαλείων. Την προσπάθεια που ξεκίνησε στην έρευνα και τη χρηματοδότηση υπερασπίστηκαν μεγάλες εταιρείες τεχνολογίας όπως η Google και το Facebook, συμπεριλαμβανομένων των παρακάτω 3 βιβλιοθηκών Python:

Pandas: ένα έργο που ξεκίνησε από τον Wes McKinney το 2008 για την εκτέλεση ποσοτικής ανάλυσης με οικονομικά δεδομένα.

Scikit-learn: ξεκίνησε το 2007 από τον David Cournapeau, που επιτρέπει στους προγραμματιστές να χρησιμοποιούν αλγόριθμους μηχανικής μάθησης και επιστήμης δεδομένων.

Matplotlib: ξεκίνησε το 2008 από τον John D. Hunter για να μεταφέρει πολλές από τις λειτουργίες γραφικών παραστάσεων της MATLAB στην Python.

Αναπτύχθηκαν εργαλεία, συμβάσεις κωδικοποίησης, βέλτιστες πρακτικές, αλγόριθμοι και κλιμακωτές λύσεις χρησιμοποιώντας το Python, συμπεριλαμβανομένων των παρακάτω παραδειγμάτων:

TensorFlow: ένα έργο που ξεκίνησε η Google το 2015 για να επιτρέψει καλύτερη οπτικοποίηση δεδομένων στη ροή εργασίας της επιστήμης δεδομένων και της μηχανικής μάθησης.

Seaborn: ξεκίνησε από τον Michael Waskom το 2012 για να βελτιώσει τα εργαλεία οπτικοποίησης δεδομένων στην επιστημονική κοινότητα υπολογιστών.

NetworkX: Το NetworkX είναι ένα πακέτο της Python για τη δημιουργία, τον χειρισμό και τη μελέτη της δομής, της δυναμικής και των λειτουργιών πολύπλοκων δικτύων. Παρέχει εργαλεία για τη μελέτη της δομής και της δυναμικής των κοινωνικών, βιολογικών και υποδομών δικτύων. Επίσης παρέχει μια τυπική διεπαφή προγραμματισμού και υλοποίηση γράφων που είναι κατάλληλη για πολλές εφαρμογές, ένα περιβάλλον ταχείας ανάπτυξης για συνεργατικά, διεπιστημονικά έργα, διασύνδεση με υπάρχοντες αριθμητικούς αλγόριθμους και κώδικα γραμμένους σε C, C++ και FORTRAN. Επίσης τη δυνατότητα ανώδυνης εργασίας με μεγάλα μη τυποποιημένα σύνολα δεδομένων. Με το NetworkX μπορεί κάποιος να φορτώσει και να αποθηκεύσει δίκτυα σε τυπικές και μη τυποποιημένες μορφές δεδομένων, να δημιουργήσει πολλούς τύπους τυχαίων και κλασικών δικτύων, να αναλύσει τη δομή του δικτύου, να

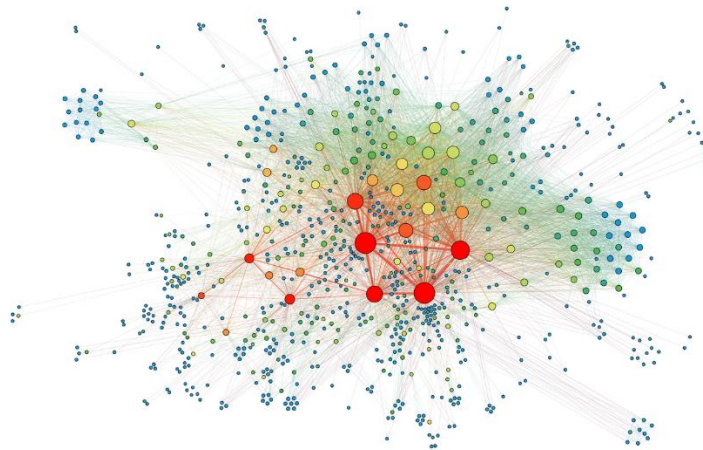
δημιουργήσει μοντέλα δικτύου, να σχεδιάσει νέους αλγόριθμους δικτύου, να σχεδιάσει δίκτυα και πολλά άλλα.

3.2. Μετρικές και χαρακτηριστικά δικτύων

Γενικά για την ανάλυση ενός δικτύου είναι σημαντικό να παρατηρηθεί πώς μοιάζει αυτό. Το σύνολο δεδομένων είναι μια αφηρημένη αναπαράσταση αυτού που υποθέτει κάποιος ότι είναι συνδέσεις μεταξύ οντοτήτων. Το δίκτυο είναι η συγκεκριμένη υλοποίηση αυτών των υποθέσεων. Το δίκτυο, τουλάχιστον σε αυτό το πλαίσιο, είναι ο τρόπος με τον οποίο ο υπολογιστής διαβάζει τις συνδέσεις που κωδικοποιεί κάποιος σε ένα σύνολο δεδομένων. Ένα δίκτυο έχει μια τοπολογία, ή ένα συνδετικό σχήμα, που θα μπορούσε να είναι συγκεντρωτικό ή αποκεντρωμένο, πυκνό ή αραιό, κυκλικό ή γραμμικό.

Το σχήμα και οι βασικές ιδιότητες του δικτύου θα βοηθήσουν να ελεγχθεί με τι ακριβώς εργάζεται και ποιες αναλύσεις φαίνονται εύλογες. Γνωρίζοντας ήδη τον αριθμό των κόμβων και των άκρων, πρέπει να είναι γνωστό το πώς «μοιάζει» το δίκτυο. Αν οι κόμβοι συγκεντρώνονται μεταξύ τους ή είναι εξίσου απλωμένοι, αν υπάρχουν πολύπλοκες δομές ή αν όλοι οι κόμβοι είναι διατεταγμένοι σε ευθεία γραμμή.

Υπάρχουν πολλοί τρόποι για την απεικόνιση ενός δικτύου, όπως μία διάταξη που κατευθύνεται από δυνάμεις (Force-directed graph drawing (Wikipedia, n.d.)) (Εικόνα 1.), ο οποίος τρόπος είναι από τους πιο συνηθισμένους. Οι συγκεκριμένοι γράφοι προσπαθούν να βρουν τη βέλτιστη τοποθέτηση για τους κόμβους τους με υπολογισμούς που βασίζονται στην τάση των ελατηρίων από τον νόμο του Hook, ο οποίος σε μικρότερους γράφους συχνά δημιουργεί καθαρές και ευανάγνωστες απεικονίσεις. Η παρακάτω απεικόνιση δείχνει ότι υπάρχουν κάποια μεγάλα στοιχεία συνδεδεμένων κόμβων (στο κέντρο) και πολλά μικρά εξαρτημένα με συνδέσεις γύρω από τις άκρες. Αυτή είναι μια αρκετά κοινή δομή δικτύου. Γνωρίζοντας ότι υπάρχουν πολλά στοιχεία στο δίκτυο, ο περιορισμός στους υπολογισμούς που χρειάζεται να εκτελεστούν προάγεται μέσω αυτής της απεικόνισης, γεγονός που την καθιστά πολύ χρήσιμη. Εμφανίζοντας τον αριθμό των συνδέσεων ως το μέγεθος των κόμβων, η απεικόνιση δείχνει επίσης ότι υπάρχουν μερικοί κόμβοι με πολλές συνδέσεις που διατηρούν τα κεντρικά στοιχεία δεμένα μεταξύ τους. Αυτοί οι μεγάλοι κόμβοι είναι γνωστοί ως hubs και στην περίπτωση που εμφανίζονται καθαρά, δίνει μια ιδέα για το τι να περιμένει κάποιος όσο αφορά την κεντρικότητα ενός δικτύου (centrality).



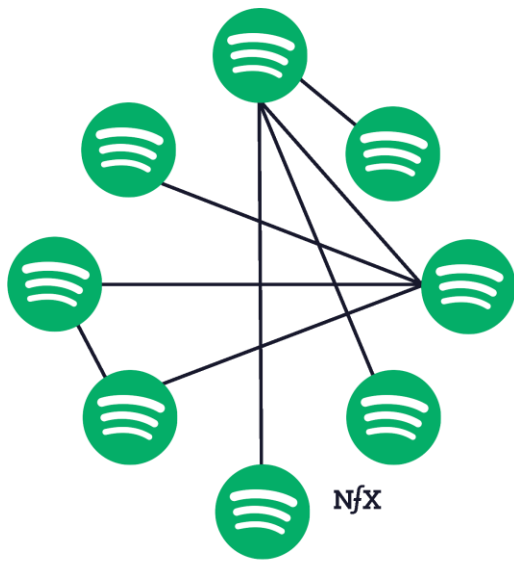
Εικόνα 1. Force-directed graph drawing

Οι οπτικοποιήσεις, ωστόσο, έχουν όρια στην ποσότητα των πληροφοριών που μπορούν να μας προσφέρουν. Όσο περισσότερα δίκτυα εξετάζονται, τόσο περισσότερο συνειδητοποιεί κάποιος ότι τα περισσότερα από αυτά μοιάζουν αρκετά μεταξύ τους και είναι δύσκολο να ξεχωρίσει το ένα από το άλλο. Οι ποσοτικές μετρήσεις είναι αυτές που επιτρέπουν να διαφοροποιούνται τα δίκτυα, και να μαθαίνει κάποιος σημαντικές πληροφορίες όπως: για τις τοπολογίες τους ή τους συνδυασμούς των κόμβων και άκρων.

Πυκνότητα δικτύου (Density)

Μια καλή μετρική για αρχή είναι η πυκνότητα δικτύου (density). Η πυκνότητα του δικτύου είναι ο λόγος των πραγματικών ακμών του δικτύου προς όλες τις πιθανές ακμές του δικτύου. Σε ένα μη κατευθυνόμενο (undirected) δίκτυο, θα μπορούσε να υπάρχει μία μόνο ακμή μεταξύ δύο κόμβων, αλλά μόνο μερικές από αυτές τις πιθανές ακμές είναι πραγματικά παρούσες. Η πυκνότητα ουσιαστικά του δικτύου δίνει μια γρήγορη αίσθηση του πόσο στενά δεμένο είναι το δίκτυο.

Η έξοδος της πυκνότητας ενός δικτύου είναι ένας αριθμός. Σε κλίμακα από το 0 έως το 1, όσο ο αριθμός πλησιάζει το 1 τόσο πιο πυκνό είναι το δίκτυο. Δηλαδή το 0 θα σήμαινε ότι δεν υπάρχουν καθόλου συνδέσεις και ένα το 1 θα έδειχνε ότι υπάρχουν όλες οι πιθανές ακμές (ένα τέλεια συνδεδεμένο δίκτυο).



Low Density



High Density

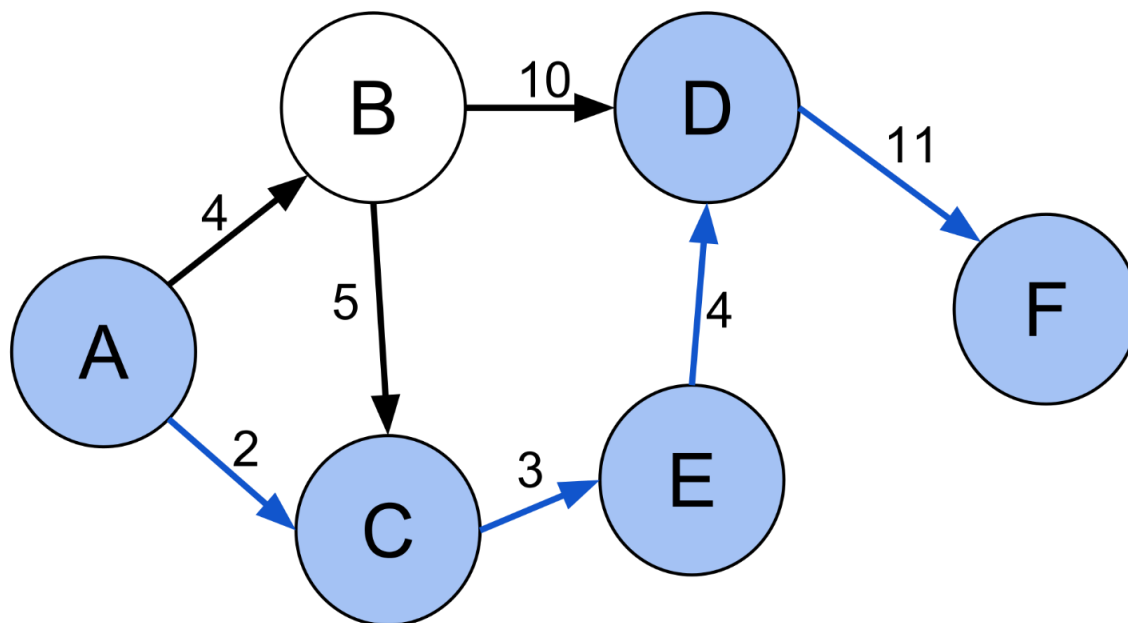
Συντομότερο μονοπάτι (Shortest path)

Η μετρική για ένα συντομότερο μονοπάτι (shortest path) είναι λίγο πιο περίπλοκη. Υπολογίζει τη συντομότερη δυνατή σειρά κόμβων και άκρων που βρίσκονται μεταξύ δύο κόμβων, κάτι που είναι δύσκολο να δει κανείς σε απεικονίσεις μεγάλου δικτύου. Για παράδειγμα σε ένα δίκτυο κοινωνικής δικτύωσης αυτή η μετρική είναι ουσιαστικά η εύρεση φίλων. Αν ένα συγγενής μου γνωρίζει κάποιον που δεν γνωρίζω, τότε ο συγγενής μου αυτός είναι ο συντομότερος δρόμος μεταξύ εμού και του ατόμου αυτού.

Για να υπολογιστεί μια συντομότερη διαδρομή, θα χρειαστεί να εισαχθούν πολλές μεταβλητές εισόδου (πληροφορίες που δίνετε σε μια συνάρτηση της γλώσσας Python). Δηλαδή ολόκληρος ο γράφος, ο κόμβος προέλευσης και ο κόμβος στόχος. Ανάλογα με το μέγεθος του δικτύου, αυτό θα χρειαστεί λίγο χρόνο για να υπολογιστεί, αφού η Python βρίσκει πρώτα όλες τις πιθανές διαδρομές και στη συνέχεια επιλέγει τη συντομότερη. Η έξοδος της εντολής `shortest_path` θα είναι μια λίστα με τους κόμβους που περιλαμβάνει την «πηγή», τον «στόχο» και τους κόμβους μεταξύ τους. Γνωρίζοντας βάση του βαθμού κεντρικότητας (degree centrality) αν ένας κόμβος έχει πολλές συνδέσεις, θα μπορούσαμε να υποθέσουμε ότι αρκετά συντομότερα μονοπάτια περνούν από αυτόν, χρησιμοποιώντας τον ως διαμεσολαβητή.

Η Python περιλαμβάνει πολλά εργαλεία που υπολογίζουν τα συντομότερα μονοπάτια. Υπάρχουν

συναρτήσεις για τα μήκη των συντομότερων μονοπατιών, για όλες τα συντομότερα μονοπάτια και για το αν υπάρχει ή όχι ένα συντομότερο μονοπάτι για ένα συγκεκριμένο ζητούμενο.



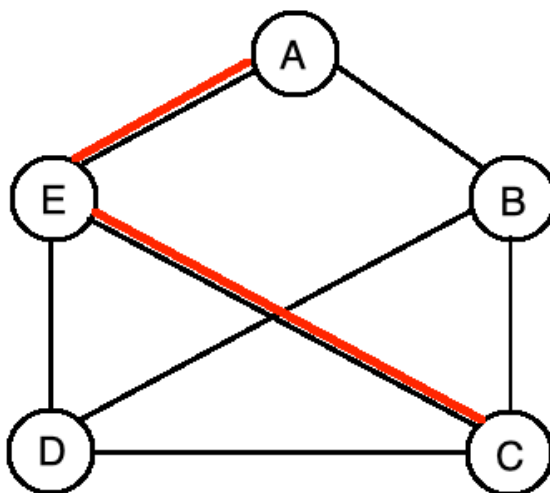
Διάμετρος (Diameter)

Υπάρχουν πολλές μετρικές δικτύου που προέρχονται από συντομότερα μήκη μονοπατιών. Μία τέτοια μετρική είναι η διάμετρος (diameter), η οποία είναι αντιπροσωπεύει το μονοπάτι με το μεγαλύτερο μήκος από όλα τα συντομότερα μονοπάτια. Μετά τον υπολογισμό όλων των συντομότερων μονοπατιών μεταξύ κάθε πιθανού ζεύγους κόμβων στο δίκτυο, η διάμετρος είναι το μήκος της διαδρομής μεταξύ των δύο κόμβων που απέχουν περισσότερο. Αυτή η μετρική έχει σχεδιαστεί για να δώσει μια αίσθηση του συνολικού μεγέθους του δικτύου, δηλαδή την απόσταση από το ένα άκρο του δικτύου στο άλλο.

Ωστόσο, η εκτέλεση αυτής της εντολής σε ένα γράφο μπορεί οδηγήσει σε σφάλμα που θα λέει ότι ο γράφος «δεν είναι συνδεδεμένος». Αυτό σημαίνει απλώς ότι ο γράφος έχει περισσότερα από ένα στοιχεία. Επειδή υπάρχουν μερικοί κόμβοι που δεν έχουν καθόλου μονοπάτια προς άλλους, είναι αδύνατο να βρεθούν όλα τα συντομότερα μονοπάτια. Σε αυτή τη περίπτωση μπορεί να το διορθωθεί το σφάλμα διαπιστώνοντας πρώτα εάν ο γράφος είναι «συνδεδεμένος» και, εάν δεν είναι συνδεδεμένος, βρίσκοντας το μεγαλύτερο στοιχείο και υπολογίζοντας τη διάμετρο μόνο σε

αυτό το στοιχείο.

Δεδομένου ότι η διάμετρος προήλθε το μεγαλύτερο στοιχείο, μπορούμε να υποθέσουμε ότι δεν υπάρχει μεγαλύτερη διάμετρος. Επομένως, είναι μια καλή βάση για τη διάμετρο ολόκληρου του γράφου. Αν για παράδειγμα η διάμετρος ενός δικτύου, του προήλθε από το μεγαλύτερο στοιχείο αυτού, είναι 8 σε αντίθεση με την πυκνότητα που κλιμακώνεται από 0 έως 1, είναι δύσκολο να γνωρίζουμε μόνο από αυτόν τον αριθμό αν το 8 είναι μεγάλη ή μικρή διάμετρος. Για να εκτιμηθεί λοιπόν αν αυτός ο αριθμός αντιπροσωπεύει μία μεγάλη ή μικρή διάμετρο θα πρέπει να γίνει σύγκριση με δίκτυα παρόμοιου μεγέθους και σχήματος.

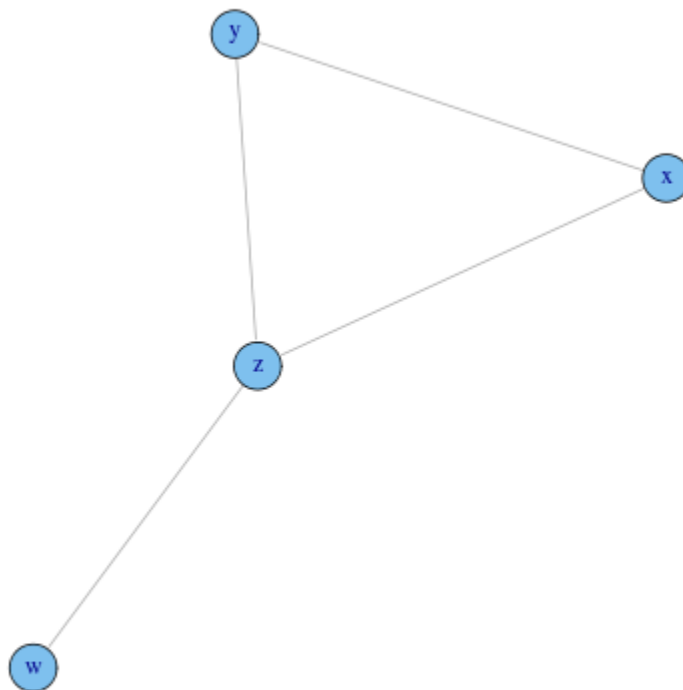


Τριαδικό κλείσιμο (Triadic closure) και Μεταβατικότητα (Transitivity)

Ο τελικός διαρθρωτικός υπολογισμός που γίνεται σε ένα δίκτυο αφορά την έννοια του τριαδικού κλεισίματος (triadic closure). Σε συνέχεια του παραπάνω παραδείγματος, σε ένα δίκτυο κοινωνικής δικτύωσης το τριαδικό κλείσιμο υποθέτει ότι εάν δύο άνθρωποι γνωρίζουν το ίδιο άτομο, είναι πιθανό να γνωρίζονται μεταξύ τους, ολοκληρώνοντας ένα τρίγωνο στην απεικόνιση τριών ακμών που συνδέονται. Ο αριθμός αυτών των κλειστών τριγώνων στο δίκτυο μπορεί να χρησιμοποιηθεί για την εύρεση ομάδων και κοινοτήτων ατόμων που όλοι γνωρίζονται αρκετά

καλά. Ένας τρόπος μέτρησης του τριαδικού κλεισίματος ονομάζεται συντελεστής ομαδοποίησης (clustering tendency) λόγω αυτής της τάσης που υπάρχει για ομαδοποίηση, αλλά η μετρική του δικτύου είναι γνωστή ως μεταβατικότητα (transitivity). Η μεταβατικότητα είναι η αναλογία όλων των τριγώνων έναντι όλων των πιθανών τριγώνων. Ένα πιθανό τρίγωνο υπάρχει όταν ένα κόμβος συνδέεται με άλλους δύο. Έτσι, η μεταβατικότητα, όπως και η πυκνότητα, εκφράζει πόσο διασυνδεδεμένο είναι ένα γράφημα ως προς την αναλογία πραγματικών έναντι πιθανών συνδέσεων. Μετρικές όπως η μεταβατικότητα και η πυκνότητα αφορούν πιθανότητες και όχι βεβαιότητες.

Επίσης, όπως και η πυκνότητα, έτσι και η μεταβατικότητα διακυμαίνεται από 0 έως 1. Αν ένας γράφος δεν είναι πολύ πυκνός, υπάρχουν λιγότερα πιθανά τρίγωνα για αρχή, τα οποία μπορεί να οδηγήσουν σε ελαφρώς υψηλότερη μεταβατικότητα. Δηλαδή, κόμβοι που έχουν ήδη πολλές συνδέσεις είναι πιθανό να είναι μέρος αυτών των κλειστών τριγώνων.



Κεντρικότητα (Centrality)

Αφού εντοπιστούν ορισμένα βασικά χαρακτηριστικά ολόκληρης της δομής του δικτύου, το επόμενο καλό βήμα είναι να βρεθούν οι κόμβοι οι οποίοι είναι οι πιο σημαντικοί στο δίκτυο. Επειδή υπάρχουν πολλοί τρόποι προσέγγισης της ερώτησης «Ποιοι κόμβοι είναι οι πιο σημαντικοί;» υπάρχουν πολλοί διαφορετικοί τρόποι υπολογισμού της κεντρικότητας. Οι πιο συνηθισμένοι τρόποι μέτρησης της κεντρικότητας είναι τρεις:

α) βαθμός κεντρικότητας (degree centrality): Ο βαθμός κεντρικότητας ενός κόμβου είναι το άθροισμα των άκρων του. Εάν ένας κόμβος έχει τρεις γραμμές που εκτείνονται από αυτόν σε άλλους κόμβους, ο βαθμός του είναι τρεις. Αν έχει πέντε άκρες, ο βαθμός του είναι πέντε. Είναι πραγματικά τόσο απλό.

$$C_D(j) = \sum_{i=1}^n A_{ij}$$

β) η κεντρικότητα μεταξύ κόμβων (betweenness centrality): Η κεντρικότητα μεταξύ κόμβων είναι λίγο διαφορετική από τις άλλες δύο, καθώς δεν ενδιαφέρεται για τον αριθμό των άκρων που έχει ένας κόμβος ή το σύνολο των κόμβων. Η κεντρικότητα μεταξύ κόμβων εξετάζει όλες τις συντομότερες διαδρομές που διέρχονται από έναν συγκεκριμένο κόμβο. Για να γίνει αυτό, πρέπει πρώτα να υπολογίσει κάθε πιθανή συντομότερη διαδρομή στο δίκτυο. Η κεντρικότητα μεταξύ κόμβων, η οποία εκφράζεται επίσης σε κλίμακα 0 έως 1, είναι αρκετά καλή στην εύρεση κόμβων που συνδέουν δύο διαφορετικά διαφορετικά μέρη ενός δικτύου.

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

γ) η ιδιοδιανυσματική κεντρικότητα (eigenvector centrality): Η ιδιοδιανυσματική κεντρικότητα είναι ένα είδος επέκτασης του βαθμού κεντρικότητας καθώς εξετάζει έναν συνδυασμό των άκρων ενός κόμβου και των άκρων των γειτονικών κόμβων αυτού του κόμβου. Η ιδιοδιανυσματική κεντρικότητα λαμβάνει υπόψιν αν ένα στοιχείο είναι κόμβος, αλλά επίσης

ενδιαφέρεται με πόσους κόμβους είναι συνδεδεμένος αυτός. Λαμβάνει τιμές από το 0 έως το 1 και, όσο πιο κοντά στο ένα, τόσο μεγαλύτερη είναι η κεντρικότητα. Η ιδιοδιανυσματική κεντρικότητα είναι χρήσιμη για την κατανόηση των κόμβων που μπορούν να λάβουν και να μεταδώσουν πληροφορίες από πολλούς άλλους κόμβους γρήγορα. Ο αλγόριθμος PageRank χρησιμοποιεί μια επέκταση αυτού του τύπου για να αποφασίσει ποιες ιστοσελίδες θα φτάσουν στην κορυφή των αποτελεσμάτων αναζήτησης.

$$x_v = \frac{1}{\lambda} \sum_{t \in M(v)} x_t = \frac{1}{\lambda} \sum_{t \in G} a_{v,t} x_t$$

Στην παρούσα διπλωματική εργασία επιλέχθηκαν 4 βάσεις δεδομένων με ποικίλα χαρακτηριστικά. Τα χαρακτηριστικά των βάσεων αναφέρονται παρακάτω. Η συλλογιστική πορεία αφού επιλέχθηκαν οι βάσεις ήταν μέσω της Python να γίνει η προσπάθεια ομαδοποίησης των δεδομένων κάθε βάσης με δύο διαφορετικούς αλγορίθμους (Pagerank, Node2vec) αλλά και προσδιορισμός των προαναφερθέντων χαρακτηριστικών. Επίσης μετρήθηκε η ικανότητα ομαδοποίησης και ο χρόνος ολοκλήρωσης ως μετρικές για την σύγκριση των 2 αλγορίθμων μεταξύ τους. Σημειώνεται εδώ ότι οι βάσεις δεδομένων χωρίζονται σε δύο κατηγορίες βάση των σχέσεων των δεδομένων και της διαδρομής μεταξύ τους. Στα κατευθυνόμενα δίκτυα (directed), οι σχέσεις δεν χρειάζεται να είναι αμοιβαίες (σε μία βάση δεδομένων κοινωνικής δικτύωσης π.χ. το άτομο A μπορεί να στείλει ένα γράμμα στον B χωρίς να το πάρει πίσω), ενώ στα μη κατευθυνόμενα (undirected) δίκτυα οι συνδέσεις είναι πάντα αμοιβαίες ή συμμετρικές. Σε μη σταθμισμένα δίκτυα (unweighted), το μήκος μιας διαδρομής είναι ο αριθμός των κορυφών που διασχίζονται. Στα σταθμισμένα δίκτυα (weighted), το μήκος μιας διαδρομής είναι το άθροισμα των βαρών των κορυφών που διασχίζονται.

3.3. Πληροφορίες για τις επιλεγμένες βάσεις δεδομένων

1. **Wiki-Vote (πηγή: snap.stanford.edu):** Η Wikipedia είναι μια δωρεάν εγκυκλοπαίδεια που γράφτηκε από εθελοντές σε όλο τον κόσμο. Ένα μικρό μέρος των συνεργατών της Wikipedia είναι διαχειριστές, οι οποίοι είναι χρήστες με πρόσβαση σε επιπλέον τεχνικές δυνατότητες που βοηθούν στη συντήρηση. Προκειμένου ένας χρήστης να γίνει

διαχειριστής, εκδίδεται ένα Αίτημα διαχειριστή (RfA) και η κοινότητα της Wikipedia μέσω δημόσιας συζήτησης ή ψήφου αποφασίζει ποιον θα προωθήσει στη διοίκηση. Χρησιμοποιώντας την πιο πρόσφατη πλήρη ένδειξη ιστορικού επεξεργασίας της σελίδας Wikipedia (από τις 3 Ιανουαρίου 2008) η συγκεκριμένη βάση δεδομένων εξήγαγε όλες τις εκλογές που αφορούν διαχειριστές και τα δεδομένα ιστορικού ψηφοφορίας. Το αποτέλεσμα έδωσε 2.794 εκλογές με 103.663 συνολικές ψήφους και 7.066 χρήστες που συμμετείχαν στις εκλογές (είτε ψήφισαν είτε ψηφίστηκαν). Από αυτές τις 1.235 εκλογές κατέληξαν σε επιτυχημένη προβολή, ενώ 1.559 εκλογές δεν οδήγησαν στην προαγωγή. Περίπου οι μισές ψήφοι στο σύνολο δεδομένων είναι από υπάρχοντες διαχειριστές, ενώ οι άλλες μισές προέρχονται από απλούς χρήστες της Wikipedia. Το δίκτυο περιέχει όλα τα δεδομένα ψήφου της Wikipedia από την έναρξη της Wikipedia έως τον Ιανουάριο του 2008. Οι κόμβοι στο δίκτυο αντιπροσωπεύουν τους χρήστες wikipedia και οι κατευθυνόμενες ακμές από τον κόμβο i στον κόμβο j αντιπροσωπεύουν αυτόν τον χρήστη i που ψήφισε το χρήστη j .

Πίνακας στατιστικών στοιχείων της βάσης δεδομένων

Nodes	7115
Edges	103689
Nodes in largest WCC	7066 (0.993)
Edges in largest WCC	103663 (1.000)
Nodes in largest SCC	1300 (0.183)
Edges in largest SCC	39456 (0.381)
Average clustering coefficient	0.1409
Number of triangles	608389
Fraction of closed triangles	0.04564
Diameter (longest shortest path)	7
90-percentile effective diameter	3.8

Σημειώνεται ότι η βάση δεδομένων χαρακτηρίζεται ως *directed-unweighted*.

2. **ENZYMES_g1 (Cheminformatics) (πηγή: networkrepository.com):** Αυτή η βάση δεδομένων επιτρέπει στους χρήστες να εξερευνήσουν και να αναλύσουν διάφορα υποσύνολα κόμβων και τα σημαντικά ενδιαφέροντα στατιστικά στοιχεία και ιδιότητες τους για να αποκτήσουν πληροφορίες στα δεδομένα του γράφου. Η δομή των συνδέσμων του ENZYMES_g1 που περιέχει δεδομένα χημικοπληροφορικής δίνει την δυνατότητα στο χρήστη να κάνει συγκρίσεις με εκατοντάδες άλλα σύνολα δεδομένων δικτύου σε πολλές διαφορετικές κατηγορίες και τομείς.

Πίνακας στατιστικών στοιχείων της βάσης δεδομένων

Nodes	37
Edges	168
Density	0.2523
Maximum degree	7
Minimum degree	0
Average degree	4.54
Assortativity	-0.02
Number of triangles	149
Average number of triangles	4.03
Maximum number of triangles	6.5
Average clustering coefficient	0.523
Fraction of closed triangles	0.482
Maximum k-core	4
Maximum k-truss	2
Upper bound of Chromatic number	5

Σημειώνεται ότι η βάση δεδομένων χαρακτηρίζεται ως undirected-unweighted.

3. **Socfb-American75 (Facebook Networks)** (πηγή: networkrepository.com): Η βάση δεδομένων αποτελεί ένα δίκτυο κοινωνικής δικτύωσης που εξάγεται από το Facebook και αποτελείται από άτομα (κόμβους) με άκρα που αντιπροσωπεύουν δεσμούς φιλίας.

Πίνακας στατιστικών στοιχείων της βάσης δεδομένων

Nodes	6.4K
Edges	217.7K
Density	0.0106763
Maximum degree	930
Minimum degree	1
Average degree	68
Assortativity	0.0660225
Number of triangles	4.4M
Average number of triangles	689
Maximum number of triangles	17.4K
Average clustering coefficient	0.240179
Fraction of closed triangles	0.163545
Maximum k-core	57
Lower bound of Maximum Clique	9

Σημειώνεται ότι η βάση δεδομένων χαρακτηρίζεται ως undirected-unweighted.

4. **bio-CE-CX (Biological Networks)** (πηγή: networkrepository.com): Τα εργαλεία βιοπληροφορικής για την ιεράρχηση των υποψηφίων συσχετισμών γονιδίου-φαινοτύπου διευκολύνουν τη γενετική έρευνα περίπλοκων χαρακτηριστικών στα ζώα. Η συγκεκριμένη

βάση δεδομένων προέρχεται από το WormNet v3 το οποίο είναι ένας νέος διακομιστής πρόβλεψης με σημαντικές ενημερώσεις στο βασικό του δίκτυο γονιδίων καθώς και μεθόδους ανάλυσης δικτύου. Το νέο βασικό δίκτυο γονιδίων έδειξε ουσιαστικά βελτιωμένες προβλέψεις των φαινοτύπων RNAi που συλλέχθηκαν από το WormBase220.

Πίνακας στατιστικών στοιχείων της βάσης δεδομένων

Nodes	15.2K
Edges	246K
Density	0.00212112
Maximum degree	375
Minimum degree	1
Average degree	32
Assortativity	0.343624
Number of triangles	6.7M
Average number of triangles	442
Maximum number of triangles	14.5K
Average clustering coefficient	0.211165
Fraction of closed triangles	0.287461
Maximum k-core	79
Lower bound of Maximum Clique	43

Σημειώνεται ότι η βάση δεδομένων χαρακτηρίζεται ως undirected-weighted

Στην παρούσα Πτυχιακή εργασία εξετάστηκαν ποικίλες ιδιότητες και χαρακτηριστικά των παραπάνω βάσεων δεδομένων. Θεωρήθηκε εύλογο οι βάσεις δεδομένων να διαφέρουν στα 4 βασικά χαρακτηριστικά που διέπουν την μορφή του γράφου (**undirected, unweighted, weighted,**

directed) αλλά να έχουν και ποικιλία όσο αφορά το μέγεθος τους. Στην συνέχεια έγινε μία ανάλυση πάνω στους τυχαίους περίπατους σε γράφους με τις βιβλιοθήκες του **NetworkX (Pagerank)**, του **Node2vec** και του **Scikit-learn** ενώ ταυτόχρονα υπολογίστηκαν χαρακτηριστικά όπως η **κεντρικότητα (centrality)**, η **πυκνότητα δικτύου (density)**, **τριαδικό κλείσιμο (triadic closure)** και η **μεταβατικότητα (transitivity)**. Επίσης όσο αφορά τον αλγόριθμο Pagerank συμπληρωματικά χρησιμοποιήθηκε ο αλγόριθμος **Jenks natural breaks optimization** (wikipedia_jenks, n.d.) για την ομαδοποίηση (clustering) των δεδομένων. Αντίστοιχα οι μετρικές που χρησιμοποιήθηκαν ήταν ο χρόνος υλοποίησης του αλγορίθμου καθώς και μία μετρική για αλγόριθμους ομαδοποίησης (rand index adjusted for chance).

NetworkX

Το NetworkX προσφέρει δεκάδες λειτουργίες και μετρικές που μπορούν να χρησιμοποιηθούν σε διάφορους συνδυασμούς και με σωστή χρήση της Python εφαρμόζονται αυτές οι μετρικές με σχεδόν απεριόριστους τρόπους. Μια γλώσσα προγραμματισμού όπως η Python δίνει την ευκαιρία σε πολλούς επιστημονικούς κλάδους την ευελιξία να εξερευνήσουν υπολογιστικά δίκτυα δεδομένων με τρόπους που δεν μπορούν να προσφέρουν άλλες εφαρμογές/γλώσσες διεπαφής, επιτρέποντάς το συνδυασμό αλλά και την σύγκριση των στατιστικών αποτελεσμάτων του δικτύου με άλλα χαρακτηριστικά των δεδομένων.

Rand index adjusted for chance.

Ο δείκτης Rand υπολογίζει ένα μέτρο ομοιότητας μεταξύ δύο ομαδοποιήσεων λαμβάνοντας υπόψη όλα τα ζεύγη δειγμάτων και καταμετρώντας ζεύγη που έχουν εκχωρηθεί στα ίδια ή διαφορετικά συμπλέγματα στις προβλεπόμενες και αληθινές ομαδοποιήσεις.

$$ARI = (RI - Expected_RI) / (max(RI) - Expected_RI)$$

Αν και ο Δείκτης Rand μπορεί να αποδώσει μόνο μια τιμή μεταξύ 0 και +1, ο προσαρμοσμένος δείκτης Rand μπορεί να αποδώσει αρνητικές τιμές εάν ο δείκτης είναι μικρότερος από τον αναμενόμενο δείκτη.

Έτσι, ο προσαρμοσμένος δείκτης Rand έχει μια τιμή ARI κοντά στο 0 για τυχαίες ομαδοποίηση και 1.0 όταν οι ομαδοποιήσεις είναι ίδιες.

Jenks natural breaks optimization

Η μέθοδος βελτιστοποίησης Jenks, που ονομάζεται επίσης μέθοδος ταξινόμησης Jenks natural breaks, είναι μια μέθοδος ομαδοποίησης δεδομένων που έχει σχεδιαστεί για να καθορίσει την καλύτερη διάταξη των τιμών σε διαφορετικές κατηγορίες. Αυτό επιδιώκεται να ελαχιστοποιήσει τη μέση απόκλιση κάθε κλάσης από τη μέση κλάση, ενώ μεγιστοποιεί την απόκλιση κάθε κλάσης από τα μέσα των άλλων κλάσεων. Με άλλα λόγια, η μέθοδος επιδιώκει να μειώσει τη διακύμανση εντός των κλάσεων και να μεγιστοποιήσει τη διακύμανση μεταξύ των κλάσεων.

3.4. Παρουσίαση αποτελεσμάτων και κώδικας.

Ο κώδικας για την ανάλυση των δεδομένων στους γράφους των προαναφερθέντων βάσεων δεδομένων συντάχθηκε στην γλώσσα Python και προστέθηκαν σχολιασμοί για κάθε βήμα που χρειάζονταν επεξήγηση.

Συνάρτηση `import_graphs`

- Διαβάζει τα δεδομένα από το δίσκο στη μνήμη. Τα δεδομένα αποθηκεύονται στην μνήμη ως αντικείμενα από την network βιβλιοθήκης.
- Είσοδος: None
- Έξοδος: λεξικό με κλειδιά [0,3] και δεδομένα τα network αντικείμενα με τους γράφους

Συνάρτηση `page_rank`

- Υπολογίζει το page rank κάθε κόμβου για κάθε Γράφο και εφαρμόζει ομαδοποίηση με τον αλγόριθμο Jenks Natural Breals
- Είσοδος: networkx αντικείμενο γράφου, τον αριθμό των κλάσεων που θα ομαδοποιηθεί ο Γράφος
- Έξοδος: numpy διάνυσμα δύο διαστάσεων. Η πρώτη διάσταση έχει τις ετικέτες κλάσεων. Η δεύτερη διάσταση θα έχει το χρόνο που έγινε η όλη διαδικασία.

Συνάρτηση `node2vec`

- Υπολογίζει τα embeddings με τον node2vec αλγόριθμο και εκτελεί ομαδοποίηση με τον αλγόριθμο Kmeans

- Είσοδος: `networkx` αντικείμενο γράφου, ένα ακαριαίο που είναι ο δείκτης του γράφου, τον αριθμό των κλάσεων που θα ομαδοποιηθεί ο Γράφος
- Έξοδος: `numpy` διάνυσμα δύο διαστάσεων. Η πρώτη διάσταση έχει τις ετικέτες κλάσεων. Η δεύτερη διάσταση θα έχει το χρόνο που έγινε η όλη διαδικασία.

Συνάρτηση `baseline`

- Κατασκευή ενός μοντέλου σύγκρισης. Πραγματοποιείται ομαδοποίηση με τον τον αλγόριθμο `spectral clustering`
- Είσοδος: `networkx` αντικείμενο γράφου, τον αριθμό των κλάσεων που θα ομαδοποιηθεί ο Γράφος
- Έξοδος: `numpy` διάνυσμα δύο διαστάσεων. Η πρώτη διάσταση έχει τις ετικέτες κλάσεων. Η δεύτερη διάσταση θα έχει το χρόνο που έγινε η όλη διαδικασία.

Συνάρτηση `display_metric`

- Βοηθητική συνάρτηση που τυπώνει μετρικές για γράφους
- Είσοδος: `networkx` αντικείμενο γράφου, το όνομα της μετρικής
- Έξοδος: `None`

Συνάρτηση `centrality`

- Τυπώνει την κεντρικότητα των γράφων.
- Είσοδος: `networkx` αντικείμενο γράφου.
- Έξοδος: `None`

Συνάρτηση `main`

- Καλεί όλες τις συναρτήσεις και τις εκτελεί
- Είσοδος: `None`
- Έξοδος: `None`

3.4.1. Κώδικας στην Γλώσσα Python

```
import networkx as nx
import os
import numpy as np
from jenkspy import JenksNaturalBreaks
import time
from operator import itemgetter
from node2vec import Node2Vec
from sklearn.cluster import KMeans, SpectralClustering
from sklearn import metrics

def import_graphs() -> dict:
    """
    Import the graphs in memory
    :return: dictionary; Keys=[0,1,2,3]; values = nx.graph
    """

    # list all the files in the directory
    files = os.listdir(DIR)
    graph = {}
    for num, i in enumerate(files):
        graph[num] = nx.read_edgelist(DIR + i, nodetype=int, data= (('weight',
float),), create_using=nx.DiGraph())
    return graph

def page_rank(graph: nx.graph, nb_class: int = 10) -> (np.array, float):
    """
    Computes the page rank of each node of a graph and performs clustering with
    Jenks Natural Breaks algorithm
    :param graph: nx.graph (directed or undirected)
    :param nb_class: number of cluster. Default = 10
    :return: Labels of each node, elapsed time
    """
    start_time = time.time()
    # compute pagerank
    pr = nx.pagerank(graph)
    pr_array = np.array(list(pr.values()))
    # cluster with Jenks Natural Breaks
    jnb = JenksNaturalBreaks(nb_class=nb_class)
    jnb.fit(pr_array)
    end_time = time.time()
    # get the cluster labels of the nodes.
    labels = jnb.labels_
    # Elapsed time
    elapsed = end_time - start_time
    return labels, elapsed

def node2vec(graph: nx.graph, name: int, nb_class: int = 10) -> (np.array,
float):
```

```

"""
https://stackoverflow.com/questions/62902871/how-can-i-cluster-a-graph-g-created-in-networkx
Computes node embedding with node2vec algorithm and perform Kmeans clustering
:param name: integer to name the embedding file
:param graph: nx.graph (directed or undirected)
:param nb_class: number of cluster. Default = 10
:return: Labels of each node, elapsed time
"""
start_time = time.time()
if 'embedding_{}.emb'.format(name) not in os.listdir(os.getcwd()):
    embedding = Node2Vec(graph, dimensions=2, walk_length=20, num_walks=10,
workers=4)
    model = embedding.fit(window=nb_class, min_count=1)
    model.wv.save_word2vec_format("embedding_{}.emb".format(name))
# load the embedding of the nodes of the graph
embedding = np.loadtxt("embedding_{}.emb".format(name), skiprows=1)
# sort the embeddings based on index
embedding = embedding[embedding[:, 0].argsort()]
# remove the index from embeddings
embedding = embedding[:, 1:]
# apply kmeans
kmeans = KMeans(n_clusters=10, random_state=0).fit(embedding)
end_time = time.time()
# get the cluster labels of the nodes.
labels = kmeans.labels_
# Elapsed time
elapsed = end_time - start_time
return labels, elapsed

def baseline(graph: nx.graph, nb_class: int = 10) -> (np.array, float):
    """
    https://stackoverflow.com/questions/46258657/spectral-clustering-a-graph-in-python
    A baseline model. Performs clustering with spectral clustering algorithm
    :param graph: nx.graph (directed or undirected)
    :param nb_class: number of cluster. Default = 10
    :return: Labels of each node, elapsed time
    """
    start_time = time.time()
    adj_mat = nx.to_numpy_matrix(graph)
    # Cluster
    sc = SpectralClustering(nb_class, affinity='precomputed', n_init=100)
    sc.fit(adj_mat)
    # get the cluster labels of the nodes.
    labels = sc.labels_
    end_time = time.time()
    # Elapsed time
    elapsed = end_time - start_time
    return labels, elapsed

def display_metric(graph: dict, metric: str):
    """
    Helper function that prints graph metrics

```

```

:param metric:
:param graph:
:return: None
"""
# Get files directory
files = os.listdir(DIR)
names = [files[i].split('.')[0] for i in range(len(files))]
print(metric)
print(15 * '-')
# Display metric
for i in graph:
    print(f'{names[i]}: {getattr(nx, metric)(graph[i]):.5f}')

def centrality(graph: dict):
    """
    Display centrality measure for a graph
    :return:
    """
    # Get files directory
    files = os.listdir(DIR)
    names = [files[i].split('.')[0] for i in range(len(files))]
    for i in graph:
        # Display dataset names
        print(names[i])
        degree_dict = dict(graph[i].degree(graph[i].nodes()))
        print(sorted(degree_dict.items(), key=itemgetter(1), reverse=True)[:20])
        print(15 * '-')

def main():
    # import graphs
    g = import_graphs()
    # Display metrics
    display_metric(g, 'density')
    display_metric(g, 'transitivity')
    centrality(g)
    baseline_ = []
    page_rank_ = []
    node2vec_ = []
    # Compute baseline, page rank and node2vec for every dataset
    for i in range(len(g)):
        baseline_.append((baseline(g[i], nb_class=5)))
        page_rank_.append((page_rank(g[i], nb_class=5)))
        node2vec_.append((node2vec(g[i], i, nb_class=5)))

    models = [page_rank_, node2vec_]
    # Display for every dataset and every models the elapsed time and rand index
    # adjusted for chance
    for model in models:
        for i in range(4):
            print(metrics.adjusted_rand_score(baseline_[i][0], model[i][0]))
            print(f'Time elapsed {model[i][1]}')
        print(15 * '-')

if __name__ == '__main__':

```

```
DIR = os.getcwd() + '\\data\\'
main()
```

3.4.2 Έξοδοι κώδικα και συγκεντρωτικά αποτελέσματα

Υπολογισμός πυκνότητας δικτύου για κάθε βάση δεδομένων

Density

Για την πυκνότητα των παρακάτω δικτύων παρατηρούμε ότι στις βάσεις δεδομένων με μικρότερο μέγεθος οι συνδέσεις είναι περισσότερες καθώς η έξοδος της πυκνότητας ενός δικτύου είναι ένας αριθμός και σε κλίμακα από το 0 έως το 1, όσο ο αριθμός πλησιάζει το 1 τόσο πιο πυκνό είναι το δίκτυο.

bio-CE-CX: 0.00106

ENZYMES_g1: 0.12613

socfb-American75: 0.00534

Wiki-Vote: 0.00205

Υπολογισμός μεταβατικότητας δικτύου για κάθε βάση δεδομένων

Transitivity

Στην περίπτωση των παραπάνω βάσεων δεδομένων όπως παρατηρούμε ότι οι κόμβοι που σχηματίζουν κλειστά τρίγωνα συμμετέχουν και σε άλλα κλειστά τρίγωνα ως ακμές. Για αυτό ενώ παρατηρούμε την ίδια τάση με την πυκνότητα η δείκτες είναι υψηλότεροι (πιο κοντά στην μονάδα).

bio-CE-CX: 0.14241

ENZYMES_g1: 0.50476

socfb-American75: 0.08249

Wiki-Vote: 0.05285

Υπολογισμός κεντρικότητας δικτύου για κάθε βάση δεδομένων

centrality

bio-CE-CX

[(69, 375), (2335, 373), (829, 352), (1401, 347), (407, 335), (2002, 330), (720, 329), (2853, 321), (808, 314), (820, 314), (748, 312), (781, 312), (343, 304), (1439, 303), (441, 301), (769, 301), (1380, 299), (1403, 299), (723, 297), (801, 296)]

ENZYMES_g1

[(30, 14), (4, 12), (29, 12), (21, 12), (34, 12), (35, 12), (2, 10), (3, 10), (28, 10), (6, 10), (8, 10), (10, 10), (13, 10), (27, 10), (26, 10), (22, 10), (31, 10), (36, 10), (25, 8), (5, 8)]

socfb-American75

[(224, 930), (4046, 781), (2442, 554), (1065, 534), (4485, 497), (4960, 461), (2637, 455), (5232, 421), (4780, 415), (6319, 414), (3048, 401), (2321, 394), (175, 388), (2782, 384), (1006, 377), (5554, 373), (2231, 372), (7, 363), (4742, 363), (4141, 361)]

Wiki-Vote

[(2565, 1167), (1549, 832), (766, 773), (11, 743), (1166, 743), (457, 732), (2688, 618), (1374, 551), (1151, 543), (5524, 538), (3352, 537), (2485, 484), (2328, 481), (4037, 472), (3456, 464), (737, 463), (2871, 444), (5079, 436), (2651, 431), (1608, 428)]

Computing transition probabilities: 100% ██████████ 37/37 [00:00<00:00, 2368.61it/s]

Generating walks (CPU: 3): 100% ██████████ 2/2 [00:00<00:00, 8.53it/s]

Generating walks (CPU: 4): 100% ██████████ 2/2 [00:00<00:00, 7.53it/s]

Generating walks (CPU: 1): 100% ██████████ 3/3 [00:00<00:00, 9.14it/s]

Generating walks (CPU: 2): 100% ██████████ 3/3 [00:00<00:00, 9.14it/s]

Computing transition probabilities: 100% ██████████ 6386/6386 [01:17<00:00, 82.24it/s]

Generating walks (CPU: 1): 100% ██████████ 3/3 [00:11<00:00, 3.80s/it]

Generating walks (CPU: 2): 100% ██████████ 3/3 [00:10<00:00, 3.34s/it]

Generating walks (CPU: 3): 100% ██████████ 2/2 [00:06<00:00, 3.45s/it]

Generating walks (CPU: 4): 100% ██████████ 2/2 [00:05<00:00, 2.51s/it]

Computing transition probabilities: 100% ██████████ 7115/7115 [00:38<00:00, 186.55it/s]

Generating walks (CPU: 1): 100%  3/3 [00:07<00:00, 2.60s/it]

Generating walks (CPU: 2): 100%  3/3 [00:07<00:00, 2.59s/it]

Generating walks (CPU: 3): 100%  2/2 [00:05<00:00, 2.64s/it]

Generating walks (CPU: 4): 100%  2/2 [00:03<00:00, 1.96s/it]

Υπολογισμός χρόνος ολοκλήρωσης και δείκτης του Rand index adjusted for chance για την ομαδοποίηση για τον αλγόριθμο Pagerank

bio-CE-CX

Rand index adjusted for chance -0.006696067647227666

Time elapsed 10.484376192092896 sec

ENZYMES_g1

Rand index adjusted for chance -0.0040330374054555525

Time elapsed 0.01562643051147461 sec

socfb-American75

Rand index adjusted for chance -0.004021920231611532

Time elapsed 11.625556945800781 sec

Wiki-Vote

Rand index adjusted for chance -0.0031618185981275256

Time elapsed 3.1718719005584717 sec

Υπολογισμός χρόνος ολοκλήρωσης και δείκτης του Rand index adjusted for chance για την ομαδοποίηση για τον αλγόριθμο Node2vec

bio-CE-CX

Rand index adjusted for chance 0.0007430033252510298

Time elapsed sec 372.5954394340515

ENZYMES_g1

Rand index adjusted for chance 0.22763603597590482

Time elapsed 1.260068655014038 sec

socfb-American75

Rand index adjusted for chance -0.00025084066859536414

Time elapsed 280.0178029537201 sec

Wiki-Vote

Rand index adjusted for chance -8.077736718461118e-05

Time elapsed 131.7734773159027 sec

Κεφάλαιο 4. Παρατηρήσεις και Συμπεράσματα

Με βάση τα αποτελέσματα του κώδικα μπορούμε αρχικά να καταλήξουμε σε κάποιες παρατηρήσεις που αφορούν τα βασικά χαρακτηριστικά των γράφων λόγω των διαφορών που υπάρχουν στα κριτήρια με τα οποία επιλέχθηκαν οι βάσεις δεδομένων.

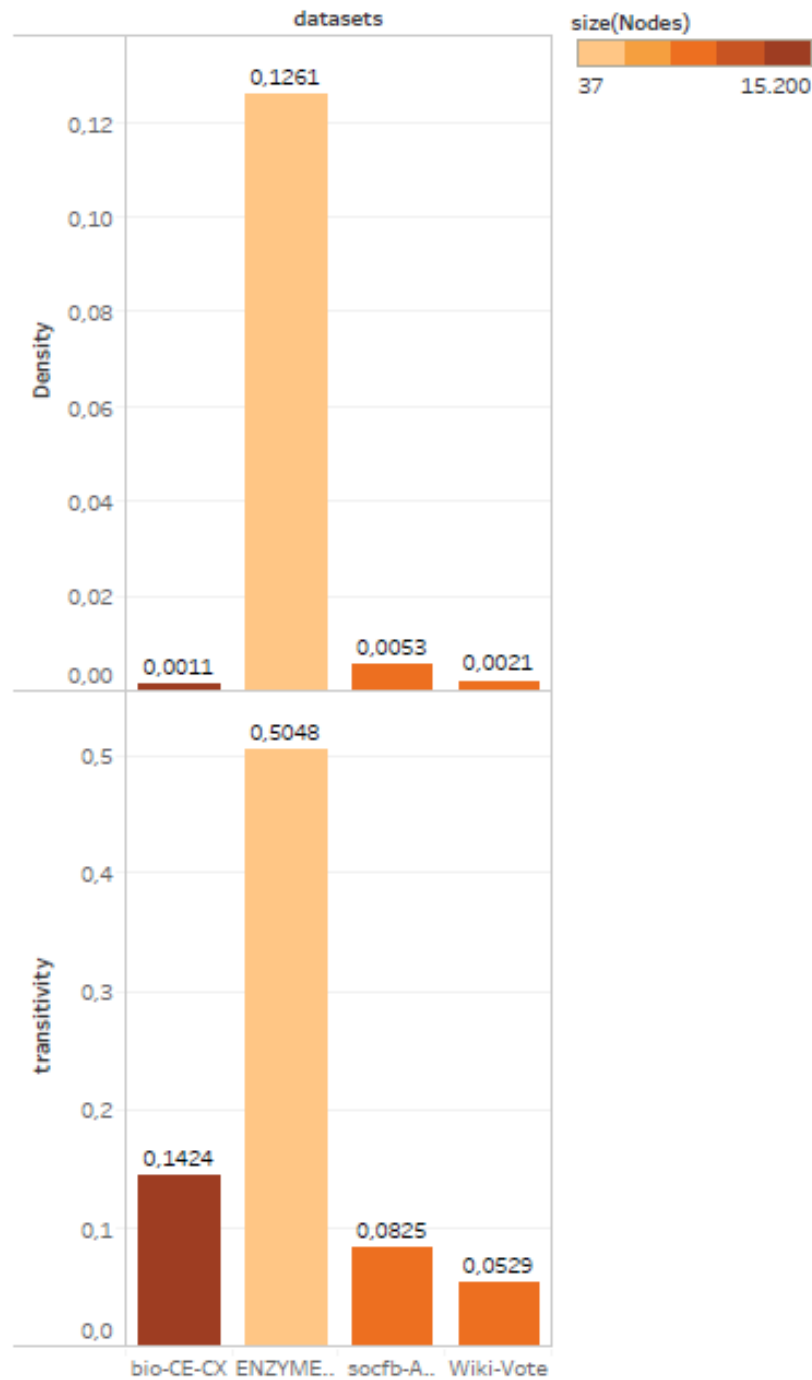
Όσο αφορά την πυκνότητα (density) των γράφων παρατηρούμε ότι στις βάσεις δεδομένων με μικρότερο μέγεθος οι συνδέσεις είναι περισσότερες καθώς η έξοδος της πυκνότητας ενός δικτύου είναι ένας αριθμός και σε κλίμακα από το 0 έως το 1, όσο ο αριθμός πλησιάζει το 1 τόσο πιο πυκνό είναι το δίκτυο. Όσο το μέγεθος του γράφου μεγαλώνει τόσο λιγότερο συνδεδεμένο είναι το δίκτυο γεγονός το οποίο συγκεκριμένα στα undirected δίκτυα είναι αναμενόμενο δηλαδή, μόνο μερικές τις πιθανές ακμές είναι πραγματικά παρούσες. Αυτό συμβαίνει διαισθητικά καθώς όσο μεγαλύτερος είναι ο γράφος τόσο μειώνονται οι πιθανότητες να είναι απολύτως συνδεδεμένος.

Όσο αφορά την μεταβατικότητα των γράφων (transitivity) παρατηρούμε την ίδια κατάσταση με την πυκνότητα. Η μεταβατικότητα είναι η αναλογία όλων των τριγώνων έναντι όλων των πιθανών τριγώνων. Ένα πιθανό τρίγωνο υπάρχει όταν ένα κόμβος συνδέεται με άλλους δύο. Έτσι, η μεταβατικότητα, όπως και η πυκνότητα, εκφράζει πόσο διασυνδεδεμένο είναι ένα γράφημα ως προς την αναλογία πραγματικών έναντι πιθανών συνδέσεων.

Αν ένας γράφος δεν είναι πολύ πυκνός, υπάρχουν λιγότερα πιθανά τρίγωνα για αρχή, τα οποία μπορεί να οδηγήσουν σε ελαφρώς υψηλότερη μεταβατικότητα. Δηλαδή, κόμβοι που έχουν ήδη πολλές συνδέσεις είναι πιθανό να είναι μέρος αυτών των κλειστών τριγώνων. Στην περίπτωσή των παραπάνω βάσεων δεδομένων όπως παρατηρούμε και στο διάγραμμα μπορούμε με σιγουριά να φτάσουμε στην παρατήρηση ότι οι κόμβοι που σχηματίζουν κλειστά τρίγωνα συμμετέχουν και σε άλλα κλειστά τρίγωνα ως ακμές. Για αυτό ενώ παρατηρούμε την ίδια τάση με την πυκνότητα η δείκτες είναι υψηλότεροι (πιο κοντά στην μονάδα).

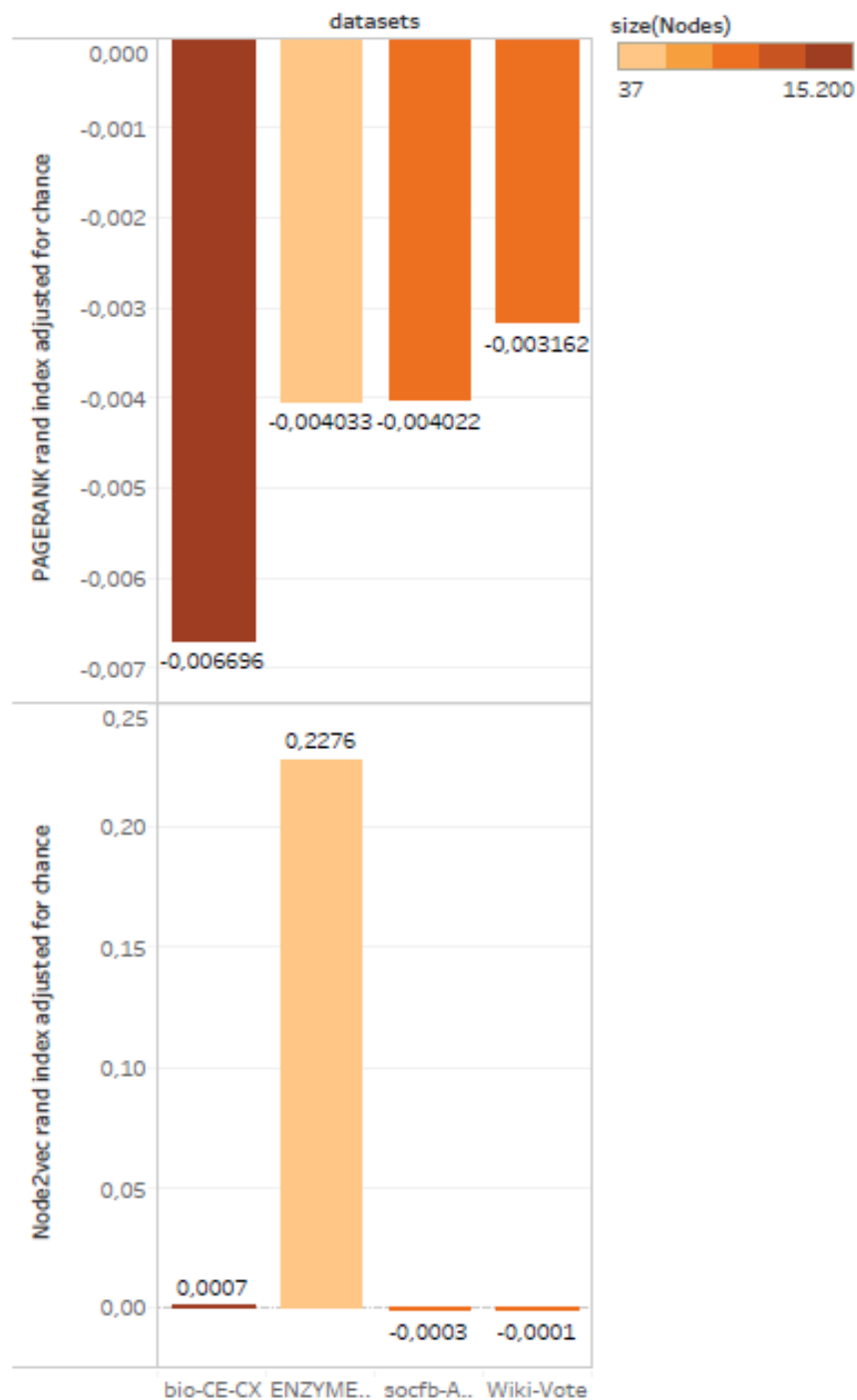
Τονίζεται σε αυτό το σημείο ότι μια υψηλή πυκνότητα δεν συνάδει απαραίτητα με υψηλή μεταβατικότητα (π.χ. σε μονοδιάστατους γράφους) και επίσης η μεταβατικότητα και η πυκνότητα αφορούν πιθανότητες και όχι βεβαιότητες.

Density, Transitivity over Datasets



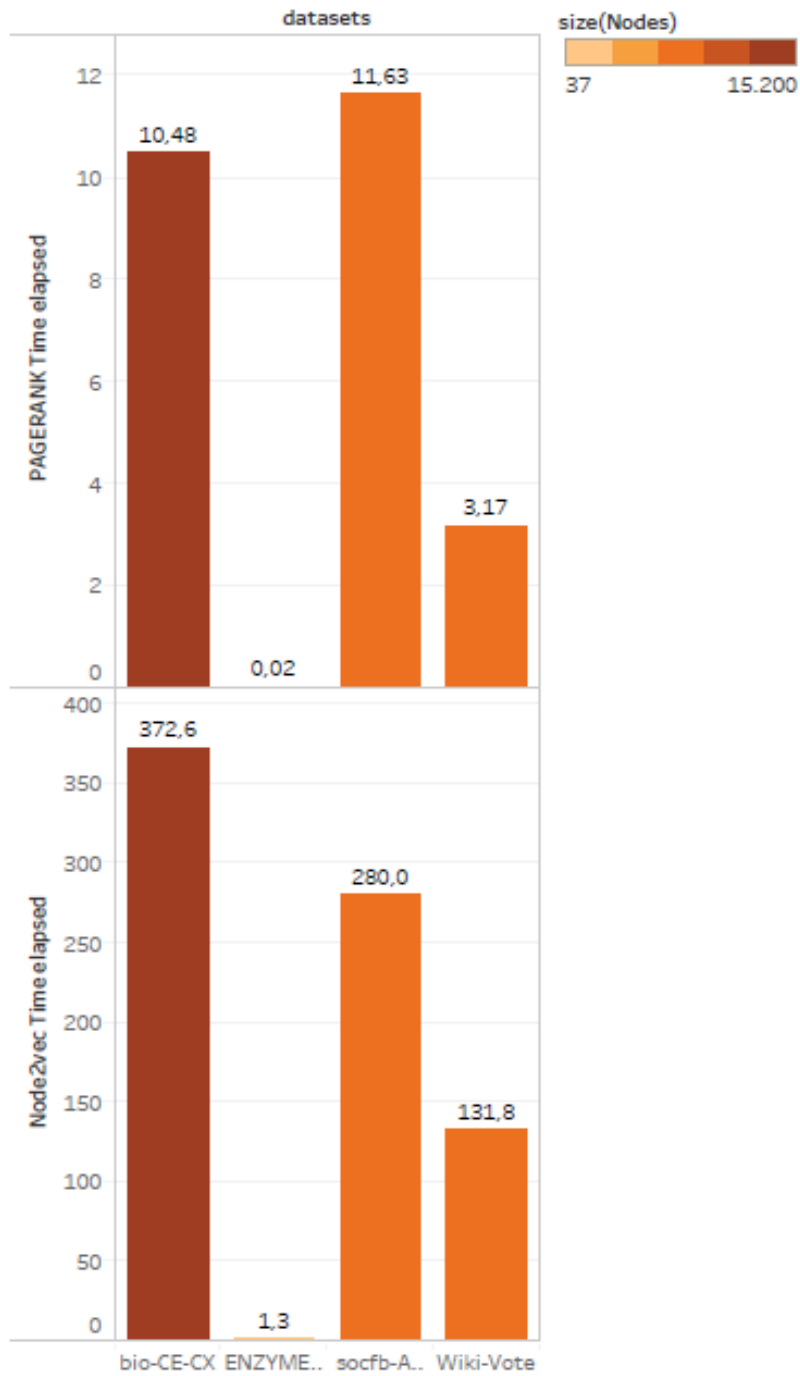
Density as an attribute and transitivity as an attribute for each datasets. Color shows sum of size(Nodes).

Rand Index Adjusted for change over Datasets



Sum of PAGERANK rand index adjusted for chance and sum of Node2vec rand index adjusted for chance for each datasets. Color shows sum of size(Nodes).

Time elapsed over Datasets



Sum of PAGERANK Time elapsed and sum of Node2vec Time elapsed for each datasets. Color shows sum of size(Nodes).

Για να υπολογιστεί η μετρική του rand index adjusted for chance, δεδομένου ότι δεν υπάρχει πραγματική αλήθεια σε ένα πρόβλημα ομαδοποίησης, το baseline που χρησιμοποιήθηκε ως πραγματική αλήθεια για να συγκριθεί η απόδοση του Pagerank και του Node2vec ήταν το spectral clustering από την βιβλιοθήκη **Scikit-learn**. Και οι δύο αλγόριθμοι απέδωσαν τιμές κοντά στο 0. Αυτό σημαίνει ότι ο διαχωρισμός που έγινε (ομαδοποίηση) δεν ήταν ο καλύτερος. Συγκριτικά μεταξύ τους ο αλγόριθμος Node2vec έδειξε καλύτερα αποτελέσματα καθώς κάποιες τιμές της μετρικής ήταν θετικές, ενώ του αλγορίθμου Pagerank ήταν όλες αρνητικές. Στον αλγόριθμο Pagerank το χειρότερο αποτέλεσμα στη μετρική του rand index adjusted for chance ήταν στην βάση δεδομένων **bio-CE-CX** η οποία και ήταν η μεγαλύτερη σε μέγεθος ενώ στις υπόλοιπες κινήθηκαν σε ίδια επίπεδα ανεξαρτήτως του μεγέθους τους. Στον αλγόριθμο Node2vec το καλύτερο αποτέλεσμα στη μετρική του rand index adjusted for chance ήταν στην βάση δεδομένων **ENZYMES_g1** η οποία ήταν και μικρότερη σε μέγεθος ενώ στις υπόλοιπες τα αποτελέσματα κινήθηκαν στα ίδια επίπεδα ανεξαρτήτως του μεγέθους τους. Αυτό παρατηρείται και στα διαγράμματα.

Όσο αφορά τον χρόνο ολοκλήρωσης του τυχαίου περιπάτου μαζί με την ταξινόμηση μεταξύ του Pagerank και του Node2vec όπως φαίνεται και στα διαγράμματα παρατηρούμε ότι όσο μεγαλύτερη σε μέγεθος η βάση δεδομένων τόσο περισσότερος χρόνος χρειάζεται για την ολοκλήρωση. Το μέγεθος λοιπόν της βάσης δεδομένων είναι το κυρίαρχο κριτήριο για τον χρόνο ολοκλήρωσης του τυχαίου περιπάτου μαζί με την ταξινόμηση. Αυτό που είναι όμως ξεκάθαρο είναι ότι ο αλγόριθμος Pagerank είναι κατά μία τάξη μεγέθους πιο γρήγορος από τον αλγόριθμο Node2vec. Στην περίπτωση που οι βάσεις δεδομένων που επιλέγονται σε μία έρευνα είναι πολύ περισσότερες σε πλήθος αλλά και πολύ μεγαλύτερες σε μέγεθος από τις επιλεχθέντες αυτής της εργασίας, όπου ο χρόνος ολοκλήρωσης είναι σημαντικό οικονομικό και υπολογιστικό κριτήριο ο Pagerank φαίνεται ως μονόδρομος.

Γενικότερα το πρόβλημα της ομαδοποίησης σε μεγάλους γράφους μέσω τυχαίων περιπάτων ή όχι είναι παρόμοιο με το πρόβλημα του partition και ως εκ τούτου ο τρόπος επίλυσης τους είναι μέσω προσεγγίσεων (heuristics), δηλαδή δεν υπάρχει βέλτιστη λύση. Οι αλγόριθμοι που επιχειρούν την επίλυση λειτουργούν με αποδεκτά και μη όρια στα οποία ένας ερευνητής θέτει τους στόχους του κάθε φορά. Στο πρόβλημα που αντιμετώπισε η παρούσα Πτυχιακή Εργασία οι αλγόριθμοι δεν

είχαν πολύ καλά αποτελέσματα διότι προσπαθούσαν να λύσουν αυτό το δύσκολο υπολογιστικά πρόβλημα με διαφορετικές προσεγγίσεις. Και οι δύο αλγόριθμοι που εφαρμόστηκαν χρησιμοποιούν τυχαίους περιπάτους. Ο Page Rank για να ταξινομήσει τους κόμβους ελέγχοντας πόσοι σύνδεσμοι τον ακολουθούν από άλλους κόμβους και άρα πόσο δημοφιλής είναι. Ο Node2Vec χρησιμοποιεί τυχαίους περιπάτους έτσι ώστε να μπορέσει να χαρτογραφήσει τον γράφο και τελικά να δημιουργήσει αναπαραστάσεις "embeddings" του γράφου σε ένα διάνυσμα το οποίο θα περιέχει όλη την πληροφορία του γράφου αυτού. Αυτό δεν σημαίνει ότι κάποιος άλλος αλγόριθμος στις συγκεκριμένες βάσεις δεδομένων θα απέδιδε αναγκαστικά καλύτερα ούτε ότι προσεγγιστικά δεν θα μπορούσε κάποιος αλγόριθμος να πλησιάσει περισσότερο. Ένα δυναμικό πρόβλημα όπως αυτό της ομαδοποίησης σε μεγάλους γράφους είναι μια διαδικασία δοκιμής και αποτυχίας και ένα από τα πιο ανοιχτά πεδία έρευνας αυτή την στιγμή στο πεδίο της Πληροφορικής. Αυτό που μπορούμε με σιγουριά να πούμε είναι ότι οι αλγόριθμοι τυχαίων περιπάτων έχουν δείξει και δείχνουν τα πιο ελπιδοφόρα αποτελέσματα στην προσέγγιση του συγκεκριμένου προβλήματος για αυτό και επιλέγονται σαν εργαλεία.

Bibliography

- (n.d.). Retrieved from https://encyclopediaofmath.org/wiki/Random_walk?fbclid=IwAR3VR8_CWoEigJ598yXAvpxl9rbqRvZFLzXPIQzYM7Ak8Y5HPX-5JFQ32IY
- (n.d.). Retrieved from <https://towardsdatascience.com/random-walks-with-restart-explained-77c3fe216bca>
- (n.d.). Retrieved from <https://programminghistorian.org/en/lessons/exploring-and-analyzing-network-data-with-python>
- (n.d.). Retrieved from <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-016-0060-5#:~:text=Random%20walk%20on%20a%20graph,vertex%20and%20moves%20to%20it>
- (n.d.). Retrieved from https://encyclopediaofmath.org/wiki/Random_walk
- (n.d.). Retrieved from corporate finance institute : https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/what-is-the-random-walk-theory/?fbclid=IwAR2YK_AD8j9DRvLhAxvyhYT3913pqnmuvgzplZDoD9ytBL0AWEIMdbTeX_s
- (n.d.). Retrieved from https://ec.europa.eu/jrc/sites/default/files/booklet-a4_innovation_radar.pdf
- (n.d.). Retrieved from <https://www.viima.com/blog/innovation-management>
- E. Wirth, G. S. (n.d.). MEASURE LANDSCAPE DIVERSITY WITH LOGICAL SCOUT AGENTS.
- Leskovec, A. G. (2016). node2vec: Scalable Feature Learning for Networks.
- Masuda, L. M. (2020). Analysis of node2vec random walks on networks.
- Orbanz, B. B.-R. (2018). RANDOM WALK MODELS OF NETWORK FORMATION AND.
- Page, S. B. (n.d.). The Anatomy of a Large-Scale Hypertextual Web Search Engine.
- Pan, H. T.-Y. (2007). Random walk with restart: fast solutions and applications.
- Pan, H. T.-Y. (2008). Random walk with restart: fast solutions and applications.
- PEARSON, K. (1905). The Problem of the Random Walk. *Nature*.
- Tong, & H.-Y. (2006). Fast Random Walk with Restart and Its Applications.
- Wikipedia. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Force-directed_graph_drawing?fbclid=IwAR1z4TONRu4Rxu3izNu0R0Gy4Xjftw6wQJwpPHcrzXyuivtN5ozl37cFSUY

wikipedia_jenks. (n.d.). Retrieved from

https://en.wikipedia.org/wiki/Jenks_natural_breaks_optimization?fbclid=IwAR3EljguxrWjH_tX54GIDexGswEuAPwiAx7fD2utc77JiAgtD8G1ahcUGs4

Yasuhiro Fujiwara*, M. N. (2012). Fast and Exact Top-k Search for Random Walk with Restart.