Εθνικο Μετσοβιο Πολυτεχνειο

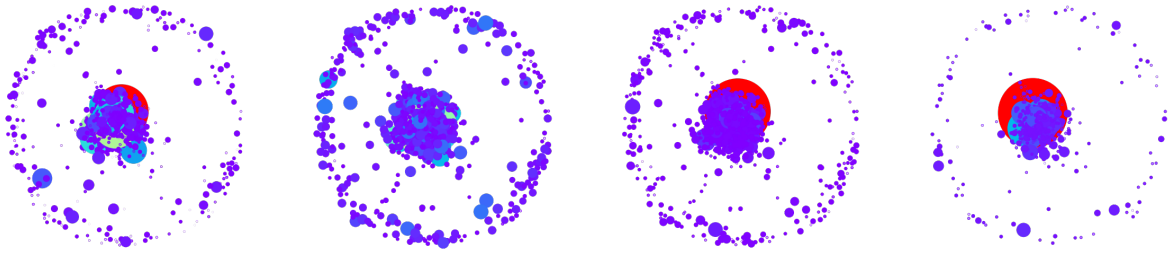
Σχολη Εφαρμοσμενων Μαθηματικων και Φυσικων Επιστημων

Δ.Π.Μ.Σ. Μαθηματικη Προτυποποιηση σε Συγχρονες

Τεχνολογιες και τη Χρηματοοικονομικη

Μεταπτυχιακή Εργασία

Μελέτες επιδραστικότητας χρηστών κοινωνικών δικτύων



Μπιτσικώκος Λοΐζος

Α.Μ.: 09320025

Επιβλέπων: Ιωάννης Κολέτσος, Αν. Καθηγητής Ε.Μ.Π.

Αθήνα

Φεβρουάριος 2022

Ευχαριστίες

Με την εκπλήρωση της παρούσας μεταπτυχιακής εργασίας θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες σε όλους όσους συνέβαλλαν στην εκπόνησή της αλλά και όσους στάθηκαν δίπλα μου σε όλη τη διάρκεια των σπουδών μου.

Ευχαριστώ θερμά τον επιβλέπων καθηγητή μου κ. Ιωάννη Κολέτσο, για την εμπιστοσύνη που μου έδειξε, την καθοδήγηση, τις υποδείξεις του και τη στήριξη.

Επίσης, θα ήθελα να ευχαριστήσω τη συμφοιτήτριά μου και φίλη (από την πρώτη κι όλας μέρα στα έδρανα του Αμφιθεάτρου 4 των Γενικών Εδρών το μακρινό 2012) Παναγιώτα Ισμήνη Χάρκεν Αλεξίου για τις μαθηματικές τις συμβουλές αλλά και όλη τη στήριξη και βοήθεια καθόλη τη διάρκεια των σπουδών μου στο ΔΠΜΣ της Μαθηματικής Προτυποποίησης.

Ακόμα, θερμές ευχαριστίες στο συνάδελφο Δρ. Άγγελο Γιαννόπουλο για την καθοδήγηση, τις συμβουλές, τις πάντα ενδιαφέρουσες ακαδημαϊκές (και όχι μόνο) συζητήσεις αλλά και τις βαθιές γνώσεις του στον τομέα της Επιστήμης των Υπολογιστών και της Μηχανικής Μάθησης.

Τέλος, θα ήθελα να ευχαριστήσω τους γονείς μου Κώστα και Νίκη, την αδερφή μου Αντιγόνη, τη Μάγια και τους φίλους Γεράσιμο και Κώστα για την στήριξη, υπομονή και κατανόηση.

Περιεχόμενα

Περίληψη

Η μοντελοποίηση της λειτουργίας ενός μέσου κοινωνικής δικτύωσης και η εύρεση του πιο επιδραστικού χρήστη του είναι μία από τις πιο ενδιαφέρουσες και πολύπλοκες εφαρμογές της Θεωρίας Γραφημάτων. Το πρόβλημα της ποσοτικοποίησης της έννοιας της επιδραστικότητας έχει τεράστια σημασία για επιχειρήσεις που δραστηριοποιούνται στο χώρο του διαδικτυακού μάρκετινγκ, αποτελώντας αναπόσπαστο κομμάτι του κλάδου της Επιχειρησιακής Έρευνας. Αρχικά, παρουσιάζουμε μία εισαγωγή στην Επιχειρησιακή Έρευνα με έμφαση στη διαδικασία και τα στάδια της Μοντελοποίησης. Στη συνέχεια, κάνουμε μία εισαγωγή στη Θεωρία Γραφημάτων παραθέτοντας εκτενώς ορισμούς και ιδιότητες κατευθυνόμενων και μη κατευθυνόμενων γραφημάτων με αυστηρό μαθηματικό φορμαλισμό. Κατόπιν αναλύεται ο αλγόριθμος PageRank, ένας αλγόριθμος ταξινόμησης ιστοσελίδων σύμφωνα με τη σημαντικότητά τους, ο οποίος βρίσκει εφαρμογές σε κοινωνικά δίκτυα. Επίσης, εισάγεται η έννοια της κεντρικότητας/επιδραστικότητας σε γραφήματα καθώς και κάποια μέτρα κεντρικότητας για υπολογισμούς. Στη συνέχεια, παρουσιάζεται η δομή και λειτουργία του Twitter ενός από τα πιο σύνθετα κοινωνικά δίκτυα για τη μετάδοση πληροφοριών, ενώ μοντελοποιείται με τη βοήθεια της Θεωρίας Γραφημάτων, ορίζοντας πληθώρα μέτρων επιδραστικότητας ειδικά προσαρμοσμένα σε αυτό. Τέλος, εφαρμόζουμε το μοντέλο για το συγκεκριμένο κοινωνικό δίκτυο σε ένα πραγματικό σύνολο δεδομένων από χρήστες και αλληλεπιδράσεις τους, ενώ επιχειρούμε να εντοπίσουμε τόσο τους πιο επιδραστικούς χρήστες όσο και πιθανές συσχετίσεις μεταξύ των υπό μελέτη μέτρων επιδραστικότητας.

Abstract

Modelling the function of an online social network and finding the most influencial user is one of the most intriguing and complex applications of Graph Theory today. The problem of quantifying influence is a difficult and loosely defined task that has tremendous importance in businesses involved in online marketing and therefore is a milestone for Operational Research. As a start, an introduction to Operational Research with an emphasis in Modelling is presented. In addition, Graph Theory is introduced presenting various definitions, examples and properties of directed and undirected graphs with the appropriate mathematical formalism. We continue with an analysis of the PageRank algorithm for classifying websites according to their importance, an algorithm greatly important for classifying social networks users as well. The notion of Centrality/Influence is also analyzed extensively presenting various centrality measures. Continuing, Twitter, an online microblogging social network, is presented and modeled with the use of Graph Theory while some influence measures specifically defined for Twitter are introduced. Lastly, the Twitter model is applied to a real dataset constituted of users and their interactions while we attempt not only to find the most influential users in the dataset but possible correlations between various influence measures.

Κεφάλαιο 1

Εισαγωγή στην Επιχειρησιακή Έρευνα

Στο παρόν κεφάλαιο παρουσιάζονται συνοπτικά ορισμένες βασικές έννοιες της Επιχειρησιακής Έρευνας. Αρχικά, δίνεται ένας ορισμός του συγκεκριμένου ακαδημαϊκού πεδίου, καθώς και ένας λειτουργικός ορισμός για τη μοντελοποίηση, αναπόσπαστο κομμάτι της Επιχειρησιακής Έρευνας, ενώ αναλύεται η διαδικασία μοντελοποίησης και επίλυσης προβλημάτων.

1.1 Εισαγωγικές έννοιες

1.1.1 Ορισμός

Η Επιχειρησιακή Έρευνα (Operations Research συχνά απαντάται και με τη συντομογραφία OR) είναι μία περιοχή των Εφαρμοσμένων Μαθηματικών που βρίσκει εφαρμογή στη λήψη αποφάσεων από οργανισμούς.

Η απαρχή της Επιχειρησιακής Έρευνας ως επιστημονικό πεδίο ξεκινά την περίοδο του Δευτέρου Παγκοσμίου Πολέμου. Κατά τη διάρκεια του πολέμου, υπήρχε η ανάγκη για κατανομή πόρων σε διάφορες στρατιωτικές επιχειρήσεις με τον βέλτιστο δυνατό τρόπο. Σε αυτές τις πρώτες εφαρμογές οφείλει άλλωστε και το ονομά της, καθώς αρχικά η έρευνα αφορούσε στρατιωτικές επιχειρήσεις. Στη συνέχεια, κατόπιν της σημαντικής επιτυχίας των μαθηματικών μεθόδων, ανάλογες μέθοδοι εφαρμόστηκαν σε μία ποικιλία από οργανισμούς (επιχειρήσεις, βιομηχανία και κρατική διοίκηση). Έτσι, πλέον αποκαλείται επιχειρησιακή όχι γιατί αφορά στρατιωτικές επιχειρήσεις αλλά επειδή αναφέρεται σε διαδικασίες ή λειτουργίες εντός οργανισμών. Μετά τη δεκαετία του πενήντα σημειώθηκε σημαντική πρόοδος στην Επιχειρησιακή Έρευνα με την ανάπτυξη και θεμελίωση μεθόδων όπως η SIMPLEX για την επίλυση προβλημάτων γραμμικού προγραμματισμού, ο δυναμικός προγραμματισμός, οι ουρές αναμονής και ο έλεγχος απογραφής (inventory control). Τα επόμενα χρόνια, η χρήση ηλεκτρονικών υπολογιστών και σχετικών λογισμικών επίλυσης προβλημάτων συνέβαλε στην περαιτέρω ανάπτυξη και επιτυχία των μεθόδων της Επιχειρησιακής Έρευνας η οποία συνεχίζεται άλλωστε μέχρι και σήμερα (Hillier & Lieberman, 2001). Καθώς γίνεται χρήση της επιστημονικής μεθόδου για την επίλυση προβλημάτων συχνά αποκαλείται και

διοικητική επιστήμη (management science).

1.1.2 Μοντελοποίηση

Βασικό κομμάτι της Επιχειρησιακής Έρευνας είναι η κατασκευή ενός μαθηματικού μοντέλου που περιγράφει το πρόβλημα προς επίλυση/ανάλυση. Αξίζει να σημειωθεί ότι δεν υπάρχει μία μόνο γενική τεχνική για την επίλυση όλων των μαθηματικών μοντέλων που προκύπτουν στην πράξη (Taha, 2007). Κάθε πρόβλημα μοντελοποιείται και επιλύεται χρησιμοποιώντας πάντα μεθόδους προσαρμοσμένες στην εκάστοτε περίπτωση.[1].

Σημειώνουμε ότι, το μοντέλο που κατασκευάζεται πρέπει να είναι μία ακριβής αναπαράσταση του πραγματικού προβλήματος. Επίσης, υποθέτουμε ότι τα συμπεράσματα (και οι λύσεις) που εξάγονται από το μοντέλο είναι αληθή και στον πραγματικό κόσμο. Η συγκεκριμένη θεώρηση αποτελεί φιλοσοφικά μία ρεαλιστική προσέγγιση ως προς τη διαδικασία μοντελοποίησης υπό την έννοια ότι το μοντέλο που κατασκευάζεται και εν τέλει επιλύεται είναι αληθές ή προσεγγιστικά αληθές στον κόσμο και επομένως οι λύσεις και τα όποια συμπεράσματα είναι επίσης αληθή ή προσεγγιστικά αληθή.[2] Η διαδικασία της μοντελοποίησης είναι το πρώτο βήμα για την επίλυση ενός προβλήματος. Ιδιαίτερη βάση πρέπει να δοθεί ώστε η μοντελοποίηση να γίνει με σωστό και προσεκτικό τρόπο.

Το μοντέλο που κατασκευάζεται πρέπει επίσης να λαμβάνει υπ όψιν όλους τους διαθέσιμους περιορισμούς και συνθήκες που τίθενται στο εκάστοτε πρόβλημα. Όπως επίσης και να καταλήγει στην καλύτερη δυνατή λύση ή όπως συχνά αποκαλείται μία *βέλτιστη* λύση.[3]. Το κομμάτι της βελτιστοποίησης της λύσης του προβλήματος είναι αναπόσπαστο κομμάτι της Επιχειρησιακής Έρευνας.

Συχνά βέβαια, καθώς μελετάμε προβλήματα Επιχειρησιακής Έρευνας αναπόφευκτα από τη μαθηματική σκοπιά τους, παραβλέπουμε ότι στη διαδικασία επίλυσης και λήψης αποφάσεων λαμβάνουν μέρος και άλλες ειδικότητες. Στην πραγματικότητα, η μελέτη ενός προβλήματος Επιχειρησιακής Έρευνας απαιτεί τη συγκρότηση μίας ομάδας που μπορεί να αποτελείται από μαθηματικούς, στατιστικολόγους, οικονομολόγους, ειδικούς στη διοίκηση επιχειρήσεων, την επιστήμη των υπολογιστών, μηχανικούς, φυσικούς, ακόμα και ειδικούς στις κοινωνικές και συμπεριφορικές επιστήμες.

1.2 Διαδικασία Μοντελοποίησης

Μία τυπική διαδικασία μοντελοποίησης ενός προβλήματος Επιχειρησιακή Έρευνα περιλαμβάνει τα εξής βήματα (Winston, 2004):

- Ορισμός προβλήματος και Συλλογή δεδομένων

[1] Μερικές χαρακτηριστικές μέθοδοι που χρησιμοποιούνται είναι: γραμμικός προγραμματισμός, ακέραιος προγραμματισμός, δυναμικός προγραμματισμός, δικτυακός προγραμματισμός, μη γραμμικός προγραμματισμός, ουρές αναμονής και προσομοίωση (Taha, 2007)

[2] Ας μην ξεχνάμε ωστόσο ότι σκοπός της επιχειρησιακής έρευνας είναι η εύρεση λύσεων με απώτερο σκοπό τη λήψη αποφάσεων εντός ενός οργανισμού και από εδώ και στο εξής περιοριζόμαστε στην εφαρμογή μεθόδων χωρίς να ανησυχούμε για τις φιλοσοφικές τους διαστάσεις. Η διοίκηση οργανισμών δεν υποκύπτει άλλωστε στη δικαιοδοσία της φιλοσοφίας.

[3] μία βέλτιση λύση και όχι η βέλτιστη λύση καθώς μπορεί να υπάρξουν πολλαπλές βέλτιστες λύσεις. (Hillier & Lieberman, 2001)

- Κατασκευή μοντέλου
- Τεχνικές επίλυσης
- Έλεγχος μοντέλου
- Εφαρμογή μοντέλου

1.2.1 Ορισμός προβλήματος και συλλογή δεδομένων

Στα περισσότερα θέματα Επιχειρησιακής Έρευνας που αντιμετωπίζονται στην πράξη τα προβλήματα δεν είναι σαφώς ορισμένα εξ αρχής. Έτσι, το πρώτο βήμα σε μία μελέτη Επιχειρησιακής Έρευνας είναι η αναγνώριση και διατύπωση του προβλήματος. Στη διαδικασία ορισμού του προβλήματος συμπεριλαμβάνονται και οι στόχοι (π.χ. μεγιστοποίηση του κέρδους μιας επιχείρησης από μία δραστηριότητα ή αντίστοιχα η ελαχιστοποίηση του κόστους κάποιας δραστηριότητας), οι κατάλληλοι περιορισμοί (π.χ. στην περίπτωση ελαχιστοποίησης κόστους, η ύπαρξη κάποιων πάγιων λειτουργικών εξόδων που δε μπορούν να μειωθούν περαιτέρω), εναλλακτικές δράσεις, χρονοδιαγράμματα (υλοποίηση έργου σε ορισμένη ημερομηνία) αλλά και αλληλεπιδράσεις μεταξύ δραστηριοτήτων (υλοποίηση θεμελίων στην κατασκευή κτηρίου πριν την τοποθέτηση της σκεπής).

Στο στάδιο αυτό είναι απαραίτητη η συλλογή δεδομένων σχετικών με το πρόβλημα. Αρκετός χρόνος δαπανάται στη συλλογή δεδομένων καθώς απαιτούνται για την βαθύτερη κατανόηση του προβλήματος αλλά και για να εισαχθούν στο μοντέλο. Τα δεδομένα πρέπει επίσης να δοθούν στο μοντέλο στην κατάλληλη μορφή και άρα δαπανάται χρόνος για την επεξεργασία τους πέρα από τη συλλογή τους.

1.2.2 Κατασκευή μοντέλου

Ένα μοντέλο αποτελεί μία ιδεαλιστική αναπαράσταση κάποιου φαινομένου ή προβλήματος που μπορεί να εκφραστεί με μαθηματικά σύμβολα και εκφράσεις (Hillier & Lieberman, 2001). Ένα μοντέλο Επιχειρησιακής Έρευνας αποτελείται από τα συστήματα εξισώσεων και τις σχέσεις που περιγράφουν το πρόβλημα.

Έστω n ποσοτικοποιημένες αποφάσεις που αντιστοιχούν σε **μεταβλητές απόφασης** $x_1, \ldots, x_n$ των οποίων τις τιμές πρέπει να προσδιορίσουμε. Έστω επίσης ένα κατάλληλο μέτρο επίδοσης (π.χ. κέρδος ή κόστος) που μπορεί να εκφραστεί ως συνάρτηση των μεταβλητών απόφασης. Η συνάρτηση αυτή αποκαλείται **αντικειμενική συνάρτηση**. Για παράδειγμα, σε περίπτωση που η αντικειμενική συνάρτηση είναι γραμμική έχουμε:

$$f(x_1, \ldots, x_n) = a_1 x_1 + \ldots + a_n x_n \tag{1.1}$$

όπου $a_i \in \mathbb{R}, i = 1, \ldots n$.

Οποιοιδήποτε **περιορισμοί** στις τιμές των $x_i\,, i = 1, \ldots, n$ εκφράζονται ως εξισώσεις ή ανισώσεις. Οι **παράμετροι** του μοντέλου είναι τόσο οι περιορισμοί όσο και η αντικειμενική συνάρτηση. Σκοπός μας είναι η

βελτιστοποίηση της αντικειμενικής συνάρτησης υπό συγκεκριμένους περιορισμούς.

Μέρος της μοντελοποίησης αποτελεί επίσης και η ανάλυση ευαισθησίας, η διερεύνηση δηλαδή κατά πόσον αλλαγή στους περιορισμούς επιφέρει αλλαγές στη βέλτιστη λύση.

Ένα μοντέλο αναγκαστικά περιέχει προσεγγίσεις και απλοποιήσεις επί του πραγματικού προβλήματος. Ιδιαίτερη σημασία πρέπει να δοθεί κατά την κατασκευή του μοντέλου ώστε η πολυπλοκοτότητα του να μην το καθιστά άλυτο. Ένα σωστό μοντέλο πρέπει να παραμένει επιλύσιμο.

Ένα ιδανικό μοντέλο έχει επίσης μεγάλη προβλεπτική ικανότητα. Μπορεί δηλαδή να προσφέρει προβλέψεις για την πραγματική συμπεριφορά του συστήματος ή φαινομένου προς μελέτη. Έτσι, ακόμα και κατά τη διαδικασία κατασκευής του μοντέλου γίνεται προσπάθεια για επαλήθευση και αξιολόγηση του μοντέλου (model validation).

Αξίζει να σημειωθεί ότι σε μοντέλα αυξημένης πολυπλοκότητας μπορεί να υπάρχουν πάνω από μία αντικειμενικές συναρτήσεις. Σημαντικός είναι επομένως ο ορισμός ενός κατάλληλου μέτρου συνολικής απόδοσης του μοντέλου μας.

1.2.3 Τεχνικές επίλυσης

Σημαντικότερη μέθοδος για την εύρεση λύσεων στο μοντέλο μας αποτελεί η χρήση αλγοριθμικών διαδικασιών. Ένας αλγόριθμος, δηλαδή μια συστηματική διαδικασία επίλυσης (Hillier & Lieberman, 2001), εφαρμόζεται με χρήση ηλεκτρονικού υπολογιστή.

Συνήθως στόχος της ανάλυσης είναι εύρεση της βέλτιστης λύσης. Διαφορετικά μοντέλα προβλημάτων Επιχειρησιακής Έρευνας διαθέτουν και διαφορετικά οπλοστάσια αλγορίθμων επίλυσης.

Αξίζει να σημειωθεί ότι συχνά πραγματοποιείται και ανάλυση των αποτελεσμάτων αφού βρεθεί η βέλτιστη λύση. Στο στάδιο αυτό διερωτόμαστε κατά πόσο επηρεάζεται η λύση από πιθανή αλλαγή των υποθέσεων που δόμησαν το μοντέλο μας. Μέρος της είναι επίσης και η ανάλυση ευαισθησίας, στην οποία καθορίζεται το πως επηρεάζουν τη λύση οι τιμές των παραμέτρων του μοντέλου.

1.2.4 Έλεγχος μοντέλου

Στο στάδιο του ελέγχου, δοκιμάζεται η επίλυση του προβλήματος με χρήση του μοντέλου που κατασκευάστηκε. Οποιαδήποτε λάθη ή αστοχίες παρατηρούνται και διορθώνονται. Στο στάδιο αυτό επανακαθορίζεται τόσο το ίδιο το μοντέλο όσο και οι διαδικασίες επίλυσής του.

Η διαδικασία των ελέγχων μπορεί να περιλαμβάνει τη χρήση δεδομένων για τον έλεγχο προσαρμογής του μοντέλου σε παλαιότερα δεδομένα ή μετρήσεις. Έχουμε έτσι και μία εκτίμηση για το πόσο καλά το μοντέλο μας ανταποκρίνεται στην πραγματικότητα. Φυσικά ανακύπτει πάντα το ερώτημα του κατά πόσο μία παρεκβολή (extrapolation) από το παρελθόν στο μέλλον είναι εφικτή. Γι αυτό το λόγο συνίσταται η διαδικασία αυτή να γίνεται με προσοχή και επιφύλαξη.

1.2.5 Εφαρμογή μοντέλου

Στο στάδιο αυτό όχι μόνο εφαρμόζεται το μοντέλο αλλά επικοινωνείται κατάλληλα εντός του οργανισμού. Η λύση και οι όποιες εναλλακτικές τους παρουσιάζονται ώστε να παρθούν οι κατάλληλες αποφάσεις. Το σύστημα προς μελέτη συνεχίζει να παρακολουθείται και μετά την εφαρμογή της λύσης ώστε να επιβεβαιωθεί ότι πληρούνται οι προϋποθέσεις και οι στόχοι της λύσης.

Κεφάλαιο 2

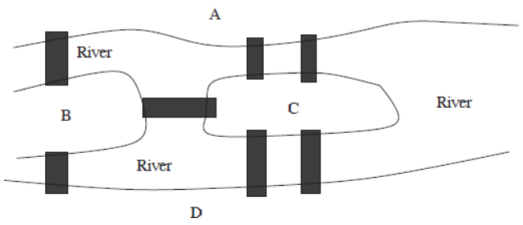
Εισαγωγή στη Θεωρία Γραφημάτων

Στο συγκεκριμένο κεφάλαιο παρουσιάζεται μία εισαγωγή στη Θεωρία Γραφημάτων[1] με έμφαση σε μία συγκεκριμένη κατηγορία γραφημάτων αυτή των δικτύων. Συχνά οι όροι γράφημα/γράφος και δίκτυο θα εναλάσσονται καθιστώντας πάντα σαφές τι εννοούμε με βάση το συγκείμενό του.

2.1 Εισαγωγικοί ορισμοί

2.1.1 Και εγένετω... Θεωρία Γραφημάτων

Η θεωρία γραφημάτων έχει τις απαρχές της στον 18ο αιώνα. Η πόλη Konigsberg διασχίζονταν από τον ποταμό Pregel και ως εκ τούτου περιελάμβανε αρκετά νησιά. Τα νησιά αυτά ενώνονταν από επτά γέφυρες. Το 1736 ο διάσημος μαθηματικός Leonhard Euler διατύπωσε την εξής ερώτηση: Είναι δυνατόν να περπατήσει κανείς την πόλη αρχίζοντας και τελειώνοντας από το ίδιο σημείο ώστε να περάσει από κάθε γέφυρα μία ακριβώς φορά;



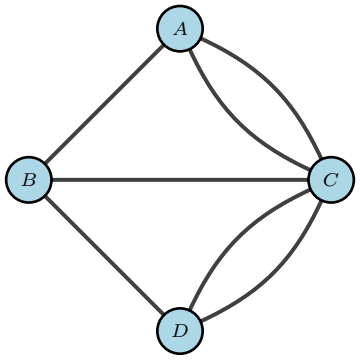
Σχήμα 2.1: Ένας απλός χάρτης του Konigsberg (Bona, 2017).

Ο Euler κατάλαβε πως το σχήμα των νησιών και του ποταμού, αλλά και οι αποστάσεις στο χάρτη δεν επηρεάζουν την απάντηση στην ερώτηση. Οι μόνες σημαντικές πληροφορίες είναι η συνδεσιμότητα, δηλαδή ο αριθμός των γέφυρων μεταξύ των νησιών. Κι έτσι, αντικατέστησε το χάρτη του Σχήματος 2.1 με το απλό

[1] συχνά συναντάται και ως Θεωρία Γράφων

σχεδιάγραμμα του Σχήματος . Τα σημεία αναπαριστούν τα κομμάτια στεριάς και οι γραμμές τις γέφυρες μεταξύ τους. Προφανώς, η διαδρομή που αναζητά το ερώτημα υπάρχει εάν και μόνο αν μπορεί να σχεδιαστεί το γράφημα του Σχήματος χωρίς να σηκώσουμε το μολύβι μας, περνώντας από κάθε γραμμή ακριβώς μία φορά και ξεκινώντας και τελειώνοντας στο ίδιο σημείο. Αυτό ήταν και ο πρώτος γράφος στην ιστορία των μαθηματικών. (Bona, 2017)

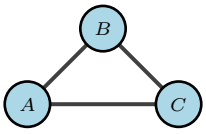
Πολλά συστήματα στον πραγματικό κόσμο μπορούν να αναπαρασταθούν από ένα διάγραμμα σημείων συνδεδεμένα μεταξύ τους με γραμμές. Τα σημεία θα μπορούσαν να συμβολίζουν ανθρώπους ενώ οι γραμμές τις φιλικές σχέσεις μεταξύ τους. Τα σημεία θα μπορούσαν να αναπαριστούν στάσεις τρένων ενώ οι γραμμές τις διαδρομές μεταξύ των σταθμών. Όπως και στο πρόβλημα με τις γέφυρες του Konigsberg αυτό που μας ενδιαφέρει δεν είναι ο τρόπος με τον οποίο συνδέονται τα σημεία μεταξύ τους αλλά το γεγονός της σύνδεσής τους. Καταλήγουμε επομένως σε μία αφηρημένη μαθηματική έννοια του γραφήματος ((Bondy & Murty, 2001).



Σχήμα 2.2: Το γράφημα των γεφυρών του Konigsbers (Bona, 2017)

2.1.2 Ορισμοί

Ένα γράφημα αποτελείται από ένα σύνολο σημείων που αποκαλούνται **κόμβοι** ή **κορυφές**) και ένα σύνολο γραμμών που αποκαλούνται **ακμές**, **τόξα** ή **πλευρές**). [2] Οι ακμές συμβολίζονται χρησιμοποιώντας τα γράμματα των κορυφών τους. Για παράδειγμα στο Σχήμα 2.3 η ακμή που ενώνει τους κόμβους A και B συμβολίζεται ως AB.

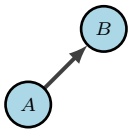


Σχήμα 2.3: Ένα απλό δίκτυο.

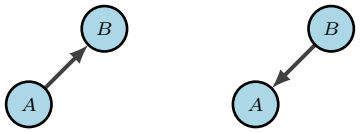
Οι ακμές ενός γράφου μπορεί να υποδεικνύουν μία κάποια ροή. Εάν η ροή σε μία ακμή επιτρέπεται μόνο

[2] η ισοδύναμη αγγλική ορολογία για τους κόμβους είναι nodes και για τις κορυφές vertices, για τα τόξα arcsκαι για τις ακμές/πλευρές edges.

προς μία κατεύθυνση τότε η συγκεκριμένη ακμή ονομάζεται **κατευθυνόμενη**. Στη σχηματική αναπαράσταση ενός γράφου μία κατευθυνόμενη ακμή συμβολίζεται με ένα βέλος (Σχήμα 2.4). Στο συμβολισμό μίας κατευθυνόμενης ακμής σημαντικό ρόλο παίζει η διάταξη. Έτσι, όπως φαίνεται στο Σχήμα 2.5 η ακμή AB υποδηλώνει κατεύθυνση από την κορυφή A στην κορυφή B ενώ αντίθετα η BA υποδηλώνει την αντίθετη κατεύθυνση. Συχνά η κατεύθυνση σε μία κατευθυνόμενη ακμή θα συμβολίζεται και ως $A \to B$.



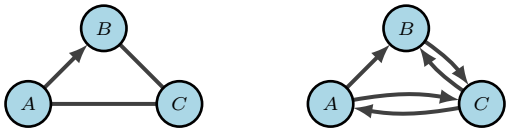
Σχήμα 2.4: Παράδειγμα σχηματικής αναπαράστασης κατευθυνόμενης ακμής.



Σχήμα 2.5: Στα αριστερά παρουσιάζεται η σχηματική αναπαράσταση της κατευθυνόμενης ακμής AB, ενώ στα δεξιά φαίνεται η BA.

Εάν η ροή σε μία ακμή επιτρέπεται προς οποιαδήποτε κατεύθυνση, τότε αποκαλείται **μη κατευθυνόμενη** ακμή. Συχνά, οι μη κατευθυνόμενες ακμές θα αποκαλούνται και **σύνδεσμοι** (links). Σημειώνεται ότι ενώ σε μία μη κατευθυνόμενη ακμή δεν υπάρχει ενδεδειγμένη κατεύθυνση, η ροή δεν πραγματοποιείται και προς τις δύο κατευθύνσεις. Η έλλειψη κατεύθυνσης σημαίνει ακριβέστερα την ελευθερία επιλογής μεταξύ δύο αντίθετων κατευθύνσεων. Παρ' όλα αυτά όπως θα δούμε στη συνέχεια, ταυτόχρονη αντίθετη ροή μεταξύ δύο κορυφών επιτρέπεται με χρήση ενός ζεύγους αντίθετα κατευθυνόμενων ακμών μεταξύ τους.

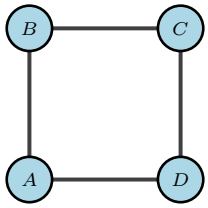
Ένας γράφος που αποτελείται μόνο από κατευθυνόμενες ακμές καλείται **κατευθυνόμενος γράφος** ή **δίκτυο**. Αντίστοιχα ένας γράφος του οποίου όλες οι ακμές είναι μη κατευθυνόμενες, ονομάζεται μη κατευθυνόμενο (για παράδειγμα, το γράφημα του Σχήματος 2.3 είναι μη κατευθυνόμενο). Ένα μικτό γράφημα με κατευθυνόμενες και μη κατευθυνόμενες ακμές μπορεί πάντα να μετασχηματιστεί σε κατευθυνόμενο γράφο-/δίκτυο αντικαθιστώντας τις μη κατευθυνόμενες ακμές του με ζεύγη αντίθετα κατευθυνόμενων.



Σχήμα 2.6: Μετασχηματισμός ενός μεικτού γράφου (αριστερά) σε δίκτυο (δεξιά).

Εάν δύο κορυφές δε συνδέονται με κάποια ακμή, τίθεται το ερώτημα εάν συνδέονται με κάποια ακολουθία ακμών. Ένα **μονοπάτι** (path) μεταξύ δύο κορυφών είναι μία ακολουθία από διακριτές ακμές που συνδέουν τις δύο κορυφές. Για παράδειγμα, ένα μονοπάτι μεταξύ των κόμβων B και D του Σχήματος 2.7 είναι η ακολουθία

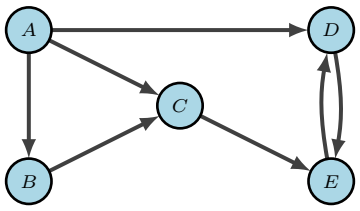
ακμών $BC - CD$ ($B \to C \to D$).



Σχήμα 2.7: Γράφος τεσσάρων κόμβων. Η ακολουθία ακμών $BC - CD$ αποτελεί ένα μονοπάτι μεταξύ των κόμων B και D.

Η έννοια της κατεύθυνσης μπορεί να οριστεί και για μονοπάτια. Ένα **κατευθυνόμενο μονοπάτι** από τον κόμβο i στον κόμβο j είναι η ακολουθία ακμών των οποίων η κατεύθυνση είναι προς τον κόμβο j, υπό την έννοια ότι η ροή από το i στο j επιτρέπεται. Ένα μη κατευθυνόμενο μονοπάτι από τον κόμβο i στον κόμβο j είναι μία ακολουθία ακμών των οποίων η κατεύθυνση μπορεί να είναι είτε προς τον κόμβο j είτε από τον j. Αξίζει να σημειωθεί ότι ένα κατευθυνόμενο μονοπάτι μπορεί να ικανοποιεί τον ορισμό του μη κατευθυνόμενου αλλά όχι το αντίθετο.

Το Σχήμα 2.8 αποτελεί ένα χαρακτηριστικό παράδειγμα γραφήματος με κατευθυνόμενα μονοπάτια (Hillier & Lieberman, 2001). Η ακολουθία ακμών $AB - BC - CE$ ($A \to B \to C \to E$) είναι ένα κατευθυνόμενο μονοπάτι από τον κόμβο B στον κόμβο E, καθώς επιτρέπεται ροή προς τον E σε όλο το μήκος του μονοπατιού. Ωστόσο, το μονοπάτι $BC - AC - AD$ ($B \to C \to A \to D$) δεν είναι ένα κατευθυνόμενο μονοπάτι από τον B στον D καθώς δεν επιτρέπεται ροή από τον κόμβο C στον A με κατεύθυνση προς τον D. Παρ' όλα αυτά το συγκεκριμένο μονοπάτι είναι μη κατευθυνόμενο καθώς η ακολουθία των ακμών $BC - AC - AD$ ενώνει τους κόμβους.

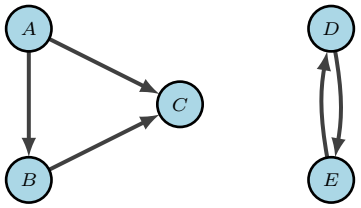


Σχήμα 2.8: Γράφος πέντε κόμβων με κατευθυνόμενο και μη κατευθυνόμενα μονοπάτια και κύκλους.

Ένα μονοπάτι που αρχίζει και τελειώνει στον ίδιο κόμβο ονομάζεται **κύκλος**. Οι κύκλοι μπορούν και αυτοί να έχουν ή να μην έχουν κατεύθυνση. Για παράδειγμα, στο Σχήμα 2.8 το μονοπάτι $DE - ED$ είναι ένα κατευθυνόμενος κύκλος, ενώ ο κύκλος $AB - BC - AC$ είναι μη κατευθυνόμενος. Ωστόσο, αξίζει να σημειωθεί ότι σύμφωνα με τον ορισμό του μονοπατιού ως ακολουθία διακριτών ακμών, το $CD - DC$ του Σχήματος 2.7 δεν είναι κύκλος καθώς τα CD και DC αντιστοιχούν στην ίδια ακμή.

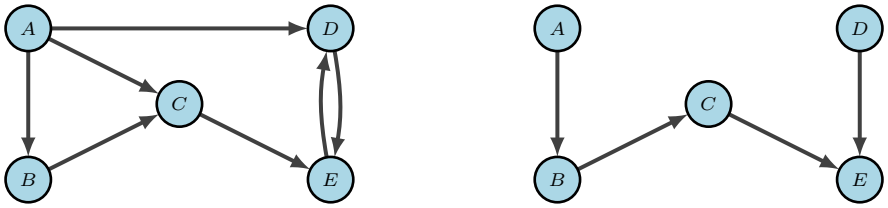
Όπως είναι προφανές από τα παραπάνω, δύο κόμβοι λέμε ότι συνδέονται εάν το γράφημά τους περιέχει τουλάχιστον ένα μη κατευθυνόμενο μονοπάτι μεταξύ τους. Ένα διασυνδεδεμένο (connected) γράφημα είναι

ένα γράφημα στο οποίο κάθε ζεύγος κόμβων είναι συνδεδεμένο μεταξύ τους. Το γράφημα του Σχήματος 2.8 είναι διασυνδεδεμένο, ωστόσο μπορεί να μετατραπεί σε μη διασυνδεδεμένο αφαιρώντας τις ακμές AD και CE όπως φαίνεται στο Σχήμα 2.9.



Σχήμα 2.9: Μη διασυνδεδεμένο γράφημα πέντε κόμβων.

Ένα **ανοιγόμενο δέντρο** (spanning tree) είναι ένα διασυνδεδεμένο γράφημα που δεν περιέχει κάποιον μη κατευθυνόμενο κύκλο. Ένα δέντρο n κόμβων έχει $n-1$ ακμές καθώς αυτός είναι ο ελάχιστος αριθμός ώστε το γράφημα να είναι διασυνδεδεμένο αλλά και ο μέγιστος αριθμός ώστε να μην περιέχει κύκλο. Για παράδειγμα ας πάρουμε τον γράφο του Σχήματος 2.8 αφαιρώντας όλες τις ακμές του. Προσθέτουμε μία ακμή τη φορά ώστε να συνδέσει δύο κόμβους. Κάθε νέα ακμή πρέπει να συνδέει έναν κόμβο που είναι συνδεδεμένος με άλλους με έναν καινούργιο ασύνδετο κόμβο. Η διαδικασία φαίνεται στο Σχήμα 2.10.



Σχήμα 2.10: Κατασκευή ενός ανοιγόμενου δέντρου από τον γράφο του Σχήματος 2.8

Τα αναδυόμενα δέντρα είναι ιδιαίτερα σημαντικά καθώς αποτελούν τη βάση για το πρόβλημα του ελάχιστου αναδυόμενου δέντρου.

Τέλος, όσον αφορά τις **ροές** σε ένα δίκτυο, ορίζουμε ως δυνατότητα κίνησης (arc capacity) τη μέγιστη ποσότητα ροής που μπορεί να περάσει από μία κατευθυνόμενη ακμή. Πραγματοποιούμε και τον εξής διαχωρισμό ως προς τους κόμβους ανάλογα με το εάν παράγουν ροή, απορροφούν ροή ή τίποτα από τα δύο. Ένας **κόμβος προσφοράς** (ή πηγή) έχει την ιδιότητα ότι η ροή που απομακρύνεται από τον κόμβο είναι μεγαλύτερη από τη ροή που κατευθύνεται προς το συγκεκριμένο κόμβο. Αντίθετα σε έναν **κόμβο ζήτησης** η ροή προς τον κόμβο ξεπερνά την ροή από τον κόμβο. Ένας **κόμβος μεταφόρτωσης** (transshipment node) ικανοποιεί την διατήρηση της ροής υπό την έννοια ότι η ροή που εξέρχεται ισούται με την εξερχόμενη ροή.

Στη συνέχεια, επιχειρούμε τη διατύπωση των παραπάνω με πιο αυστηρό μαθηματικό φορμαλισμό.

2.2 Μαθηματικός Φορμαλισμός

Ορισμός 2.2.1. *Ένα **γράφημα/γράφος** είναι ένα ζεύγος συνόλων*

$$G = (V, E)$$

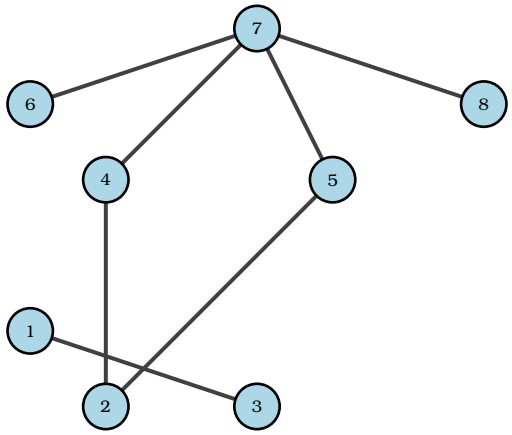
*όπου V το σύνολο των **κόμβων** και E το σύνολο των **ακμών**.*

Τα σύνολα V, E ικανοποιούν την σχέση

$$E \subseteq [V]^2$$

δηλαδή τα στοιχεία του E είναι υποσύνολα δύο στοιχείων του V (Diestel, 2000).

Όπως αναφέρουμε παραπάνω, ένα γράφημα απεικονίζεται με σημεία και γραμμές που τα ενώνουν, όπως φαίνεται στο Σχήμα 2.11. Σημειώνουμε πως αντί για γράμματα οι κόμβοι συμβολίζονται εδώ με αριθμούς, χωρίς βλάβη της γενικότητας.



Σχήμα 2.11: Ένας γράφος πάνω στο $V = \{1, \ldots, 7\}$ με σύνολο ακμών $E = \{\{1,3\},\{2,4\},\{2,5\},\{4,7\},\{5,7\},\{6,7\},\{7,8\}\}$.

Θα λέμε ότι ένας γράφος με σύνολο κόμβων V θα είναι ένας γράφος *πάνω* στο V. Συχνά θα συμβολίζουμε το σύνολο των κόμβων ενός γράφου G με $V(G)$, ενώ το σύνολο των ακμών του $E(G)$. Έτσι, ο γράφος $H = (W, F)$ έχει σύνολο κόμβων $V(H) = W$ και ακμών $E(H) = F$.

Ο αριθμός των κόμβων ενός γραφήματος ονομάζεται *τάξη* του γραφήματος και συμβολίζεται με $|G|$. Ο αριθμός των ακμών του συμβολίζεται με $||G||$. Ένα γράφημα είναι πεπερασμένος ή άπειρος αναλόγως της τάξης του.

Ο κενός γράφος $(0, 0)$ θα συμβολίζεται με 0. Ένας γράφος τάξης 0 ή 1 είναι τετριμμένος. Ένας κόμβος v είναι προσπίπτων στην ακμή e εάν $v \in e$ και τότε η e είναι μία ακμή στον v. Οι δύο κόμβοι που προπίπτουν στην e είναι οι κόμβοι τέλους της ή τα άκρα της και η ακμή ενώνει τα άκρα.

Η ακμή (x, y) γράφεται xy (ή yx καθώς δεν έχει οριστεί ακόμα κατεύθυνση). Εάν $x \in X$ και $y \in Y$ τότε η

xy είναι μία ακμή $X - Y$. Το σύνολο όλων των $X - Y$ ακμών σε ένα σύνολο κόμων ονομάζεται $E(X, Y)$. Το σύνολο όλων των ακμών στο E σε έναν κόμβο v συμβολίζεται με $E(v)$.

Δύο κόμβοι x, y του G είναι γειτονικοί εάν η ακμή xy είναι ακμή του G. Δύο διακριτές ακμές $e \neq f$ είναι γειτονικές εάν έχουν ένα κοινό άκρο. Εάν όλοι οι κόμβοι του G είναι γειτονικοί ανά δύο τότε ο γράφος G είναι πλήρης. Ένας πλήρης γράφος με n κόμβους ονομάζεται K^n γράφος (ειδική περίπτωση γράφος K^3 που ονομάζεται τρίγωνο).

Ζεύγη μη-γειτονικών κόμβων ή ακμών ονομάζονται ανεξάρτητα. Ένα σύνολο κόμβων ή ακμών είναι ανεξάρτητο εάν κανένα ζεύγος στοιχείων του δεν είναι γειτονικό.

Έστω δύο γραφήματα $G = (V, E)$ και $G' = (V', E')$. Λέμε ότι τα G, G' είναι ισομοφρικά και γράφουμε $G \simeq G'$, εάν υπάρχει μία ένα-προς-ένα αντιστοιχία (bijection) $\phi : V \to V'$ με $xy \in E \Leftrightarrow \phi(x)\phi(y) \in E'$ για κάθε $x, y \in V$. Η απεικόνιση ϕ καλείται ομοιομορφισμός, ενώ στην περίπτωση που $G = G'$ ονομάζεται ομομορφισμός (automorphism). Μία απεικόνιση που παίρνει γραφήματα σαν όρισμα ονομάζεται αναλλοίωτο εάν αντιστοιχεί ίσες τιμές σε ισομορφικούς γράφους. Ο αριθμός κόμβων και ο αριθμός ακμών είναι δύο αναλλοίωτα.

2.2.1 Πράξεις και σχέσεις μεταξύ γραφημάτων

Καθώς τα γραφήματα αποτελούνται από σύνολα, ορίζονται πράξεις μεταξύ γραφημάτων όμοιες με πράξεις μεταξύ συνόλων. Έτσι έχουμε:

$$G \cup G' := (V \cup V', E \cup E')$$
$$G \cap G' := (V \cap V', E \cap E')$$

Επίσης, εάν $G \cap G' = \emptyset$ τότε οι G, G' είναι ξένοι μεταξύ τους.

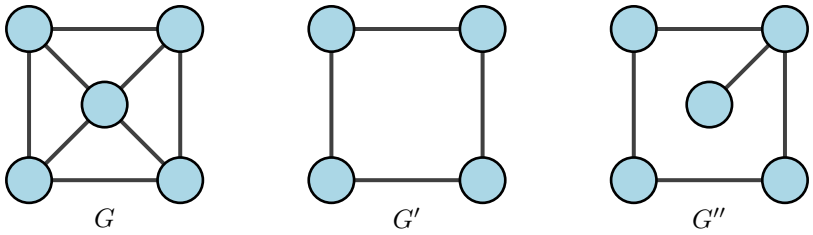
Εάν $V' \subseteq V$ και $E' \subseteq E$ τότε ο G' είναι *υπογράφος* του G (και αντίστοιχα G *υπεργράφος* του G'). Συμβολίζουμε με $G' \subseteq G$.

Εάν $G' \subseteq G$ και G' περιέχει όλες τις ακμές $xy \in E$ με $x, y \in E'$ τότε το G' είναι *επαγώμενο υπογράφημα* του G (induced subgraph). Θα λέμε ότι το V' επάγει ή επεκτείνει το G' στο G και θα γράφουμε $G[V'] := G'$.

Εάν $U \subseteq V$ είναι οποιοδήποτε σύνολο κόμβων τότε το $G[U]$ συμβολίζει το γράφημα πάνω στο U του οποίου οι ακμές είναι ακριβώς οι ακμές του G με άκρα στο U. Εάν H είναι ένα υπογράφημα του G (όχι απαραίτητα επαγώμενο) αντί για $G[V(H)]$ γράφουμε $G[E]$. Τέλος, εάν $G' \subseteq G$ είναι επεκτεινόμενο υπογράφημα του G εάν το V' επεκτείνει όλο το G, δηλαδή $V' = V$.

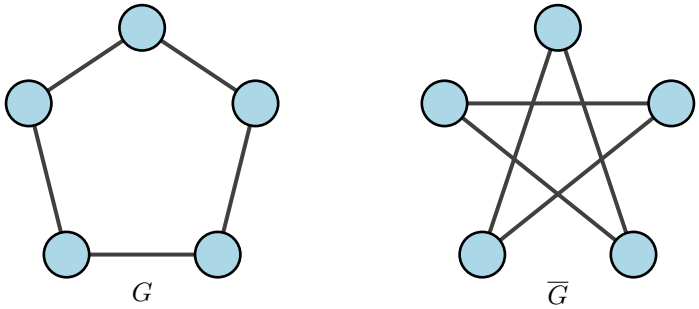
Εάν U είναι οποιοδήποτε σύνολο κόμβων (συνήθως στο G) γράφουμε $G - U$ εννοώντας $G[V \setminus U]$. Έτσι, παίρνουμε το $G - U$ διαγράφοντας όλους τους κόμβους που ανήκουν στο $U \cap V$ καθώς και τις αντίστοιχες ακμές.

Εάν G και G' είναι ξένοι ματαξύ τους, ορίζουμε ως $G * G'$ το γράφημα που παίρνουμε από το $G \cup G'$ ενώνοντας όλους τους κόμβους του G με όλους τους κόμβους του G'.



Σχήμα 2.12: Ο γράφος G έχει τα υπογραφήματα G' και G''. Το G' είναι επαγώμενο υπογράφημα του G ενώ το G'' όχι.

Το συμπλήρωμα $\overline{G}$ του G είναι ένας γράφος στο V με σύνολο ακμών $[V]^2 \setminus E$ (Diestel, 2000).



Σχήμα 2.13: Γράφος ισομορφικός στο συμπλήρωμά του.

2.2.2 Ο βαθμός (degree) κάθε κόμβου

Έστω $G = (V, E)$ ένα μη-κενό γράφημα. Το σύνολο των γειτονικών κόμβων του κόμβου v συμβολίζεται ως $N_G(v)$ ή $N(v)$.

Εν γένει, για $U \subseteq V$ οι γείτονες του $V \setminus U$ των κόμβων του U ονομάζονται γείτονες του U και το σύνολό τους συμβολίζεται ως $N(U)$.

Ο βαθμός $deg_G(v) = deg(v)$ ενός κόμβου v είναι ο αριθμός $|E(v)|$ των ακμών στον κόμβο v και επομένως ισούται με τον αριθμό των γειτόνων του v.

Εάν ο βαθμός ενός κόμβου είναι 0 τότε ο κόμβος ονομάζεται απομονωμένος.

Σε ένα δίκτυο, δηλαδή ένα κατευθυνόμενο γράφημα ορίζουμε τόσο τον βαθμό inlink όσο και τον βαθμό outlink δηλαδή τον αριθμό των ακμών που κατευθύνονται προς τον κόμβο και τον αριθμό τον ακμών που απομακρύνονται από τον κομβο αντίστοιχα. Συμβολίζουμε τις δύο ποσότητες με $deg^{in}(v)$ και $deg^{out}(v)$ αντίστοιχα.

Ορίζουμε επίσης τον μέγιστο και ελάχιστο βαθμό του γραφήματος σύμφωνα με τις σχέσεις:

$$\delta(G) := \min\{deg(v)|v \in V\}$$
$$\Delta(G) := \max\{deg(v)|v \in V\}$$

Εάν όλοι οι κόμβοι του του γράφου G έχουν τον ίδιο βαθμό k τότε ο γράφος G ονομάζεται $k-$κανονικός ή κανονικός.

Ορίζουμε ως μέσο βαθμό του γράφου:

$$deg(G) := \frac{1}{|V|}\sum_{v\in V} deg(v)$$

για τον οποίο ισχύει:

$$\delta(G) \leq deg(G) \leq \Delta(G)$$

Ο μέσος βαθμός του γράφου ποσοτικοποιεί ολικά (globally) την τοπική μέτρηση των βαθμών των κόμβων. Συχνά χρησιμοποιείται εναλλακτικά και η ποσότητα:

$$\epsilon(G) := \frac{|E|}{|V|}$$

για την οποία ισχύει (Diestel, 2000):

$$\begin{aligned} |E| &= \frac{1}{2}\sum_{v\in V} deg(v) \\ &= \frac{1}{2} deg(G) \cdot |V| \\ \Rightarrow \quad \epsilon(G) &= \frac{deg(G)}{2} \end{aligned}$$

2.2.3 Μονοπάτια και κύκλοι

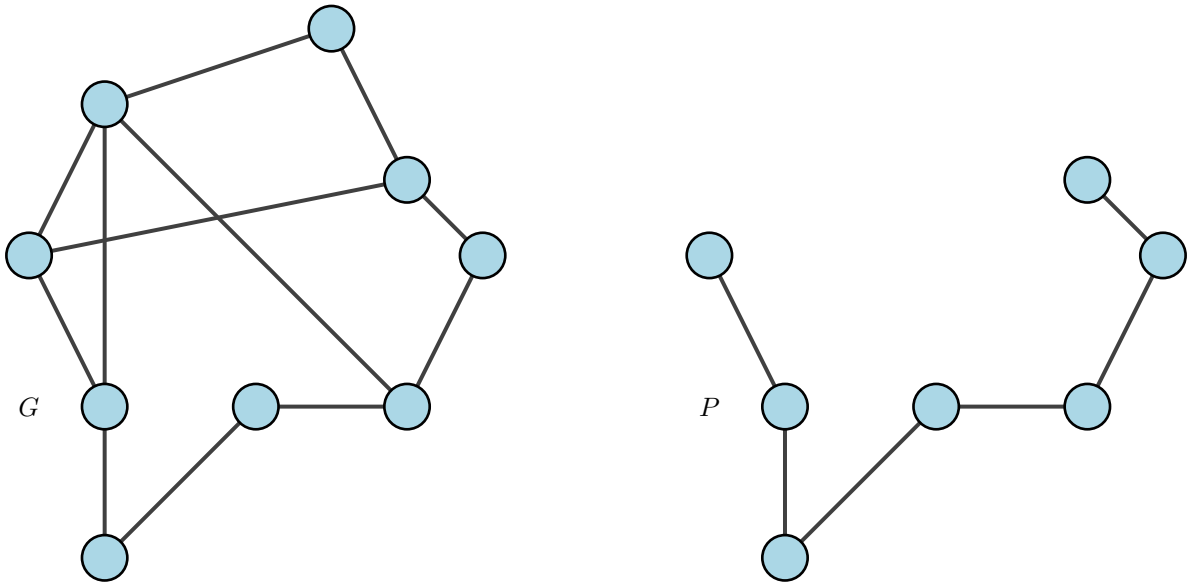
Ορισμός 2.2.2. *Ένα **μονοπάτι** είναι ένας μη κενός γράφος $P(V, E)$ της μορφής*

$$V = \{x_0, x1, \ldots, x_k\} \qquad E = \{x_0x_1, x_1x_2, \ldots, x_{k-1}x_k\}$$

όπου τα x_i είναι διακριτά.

Οι κόμβοι x_0 και x_k συνδέονται μέσω του P και καλούνται άκρα του μονοπατιού, ενώ οι υπόλοιποι κόμβοι

($x_1, \ldots, x_{k-1}$) καλούνται εσωτερικοί κόμβοι του P. Ο αριθμός των ακμών ενός μονοπατιού είναι το μήκος τους. Μονοπάτια μήκους k συμβολίζονται ως P^k.



Σχήμα 2.14: Ένα μονοπάτι P^7 στο δίκτυο G.

Συχνά αναφερόμαστε στο μονοπάτι ως την ακολουθία των ακμών του $P = x_0x_1 \ldots x_k$ ή $P = x_k \ldots x_0$. Για $0 \leq i \leq j \leq k$ γράφουμε:

$$Px_i := x_0 \ldots x_i$$
$$x_iP := x_i \ldots x_k$$
$$x_iPx_j := x_i \ldots x_j$$

$$\mathring{P} := x_1 \ldots x_{k-1}$$
$$P\mathring{x}_i := x_0 \ldots x_{i-1}$$
$$\mathring{x}_iP := x_{i+1} \ldots x_k$$
$$\mathring{x}_iP\mathring{x}_j := x_{i+1} \ldots x_{j-1}$$

όμοια διαχειριζόμαστε τις πράξεις μεταξύ μονοπατιών εάν είναι ξανά μονοπάτια.

Δοθέντων δύο συνόλων κόμβων A, B αποκαλούμε το μονοπάτι P, $A - B$ μονοπάτι εάν:

$$V(P) \cap A = \{x_0\}$$
$$V(P) \cap B = \{x_k\}$$

Δύο μονοπάτια είναι ανεξάρτητα εάν κανένα τους δεν περιέχει κάποιον εσωτερικό κόμβο του άλλου.

Εάν $P = x_0 \dots x_{k-1}$ είναι ένα μονοπάτι και $k \geq 3$ τότε ο γράφος $C := P + x_{k-1}x_0$ ονομάζεται κύκλος. Όπως και στα μονοπάτια συμβολίζουμε τους κύκλους με την (κυκλική) ακολουθία κόμβων τους. Έτσι, ο κύκλος C μπορεί να γραφεί ως $x_0 \dots x_{k-1}x_0$. Το μήκος ενός κύκλου είνια ο αριθμός των ακμών (ή κόμβων) του. Ένας κύκλος μήκους k ονομάζεται $k-$κύκλος και συμβολίζεται C^k.

2.3 Συνδεσιμότητα

Ένας μη κενός γράφος καλείται συνδεδεμένος εάν οποιοιδήποτε δύο κόμβοι του συνδέονται με ένα μονοπάτι στο G. Εάν $U \subseteq V(G)$ και $G[U]$ συνδεδεμένος, καλούμε και το ίδιο το U συνδεδεμένο στο G.

Πρόταση 2.3.1. *Οι κόμβοι ενός συνδεδεμένου γράφου G μπορούν να απαριθμηθούν έτσι ώστε $G_i := G[v_1, \dots, v_i]$ να είναι συνδεδεμένος για κάθε i (Diestel, 2000).*

Έστω $G = (V, E)$, ο μέγιστος συνδεδεμένος υπογράφος του G ονομάζεται *component* του G. Ο κενός γράφος δεν έχει component.

Λέμε ότι ένα σύνολο $X \subseteq V \cup E$ *διαχωρίζει* δύο σύνολα $A, B \subseteq V$ στο G εάν κάθε μονοπάτι $A - B$ περιέχει έναν κόμβο ή ακμή από το X. Επομένως, $A \cap B \subseteq X$.

Εν γένει, λέμε ότι το X *διαχωρίζει* το G και ότι το X είναι *διαχωρίζον* σύνολο στο G, εάν το X διαχωρίζει δύο κόμβους του $G - X$ στο G.

Ένας κόμβος που διαχωρίζει δύο άλλους κόμβους του ίδιου component ονομάζεται *cut vertex* και η ακμή που διαχωρίζει τα άκρα ονομάζεται *γέφυρα*. Έτσι, οι γέφυρες είναι οι ακμές που δεν περιέχονται σε κανέναν κύκλο.

Ορισμός 2.3.1. *Ο γράφος G καλείται $k-$συνδεδεμένος με $k \in \mathbb{N}$ εάν $|G| > k$ και $G - X$ είναι συνδεδεμένος $\forall X \subseteq V$ με $|X| < k$.*

Έτσι, σε έναν $k-$συνδεδεμένο γράφο δεν υπάρχει ζεύγος κόμβων που διαχωρίζεται από λιγότερους από k κόμβους. Επίσης, κάθε μη κενός γράφος είναι $0-$συνδεδεμένος και οι $1-$συνδεδεμένοι γράφοι είναι οι μη τετριμμένη συνδεδεμένοι γράφοι.

Ορισμός 2.3.2. *Ο μέγιστος ακέραιος k τ.ω. ο γράφος G να είναι $k-$συνδεδεμένος ονομάζεται **συνδεσιμότητα** του G και συμβολίζετι $\kappa(G)$*

Σημειώνουμε ότι $\kappa(G) = 0$ αν και μόνο αν το G είναι ασύνδετο ή K^1 και $\kappa(K^n) = n - 1$, $\forall n \geq 1$.

Ορισμός 2.3.3. *Εάν $|G| > 1$ και $G - F$ συνδεδεμένο $\forall F \subseteq E$ με λιγότερες από l ακμές τότε το γράφημα G ονομάζεται $l-$edge-connected.*

Ορισμός 2.3.4. *Το μέγιστο $l \in \mathbb{Z}$ τ.ω. το G να είναι $l - edge - connected$ ονομάζεται **edge-connectivity** του G και συμβολίζεται $\lambda(G)$.*

Για κάθε μη τετριμμένο γράφο G ισχύει:

$$\kappa(G) \leq \lambda(G) \leq \delta(G)$$

Επομένως (Diestel, 2000):

- Υψηλή συνδεσιμότητα απαιτεί μεγάλο έλαχιστο βαθμό
- Μεγάλος ελάχιστος βαθμός δεν σημαίνει απαραίτητα υψηλή συνδεσιμότητα, ούτε και υψηλή συνδεσιμότητα ακμών.
- μεγάλος ελάχιστος βαθμός συνεπάγεται ύπαρξη υψηλά συνδεδεμένου υπογράφου

Θεώρημα 2.3.1. *Κάθε γράφος μέσο βαθμού τουλάχιστον $4k$ έχει έναν $k-$συνδεδεμένο υπογράφο.*

2.4 Δέντρα και δάση

Ορισμός 2.4.1. *Ένας **ακυκλικός** γράφος (δηλαδή ένας γράφος που δεν περιέχει κύκλους) ονομάζεται **δάσος**. Ένα συνδεδεμένο δάσος ονομάζεται **δέντρο**.*

Σύμφωνα με τους παραπάνω ορισμούς ένα δάσος είναι ένα γράφημα με components δέντρα.

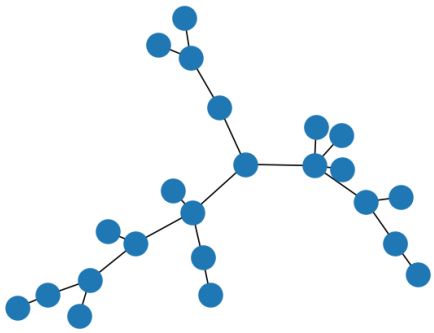
Ορισμός 2.4.2. *Οι κόμβοι βαθμού 1 ενός δέντρου ονομάζονται **φύλλα**.*

Κάθε μη τετριμμένο δέντρο έχει τουλάχιστον δύο φύλλα (π.χ. τα άκρα του μεγαλύτερου μονοπατιού).

Αξίζει να σημειωθεί ότι αφαιρώντας ένα φύλλο από ένα δέντρο το εναπομένον γράφημα εξακολουθεί να είναι δέντρο.

Θεώρημα 2.4.1. *Τα ακόλουθα είναι ισοδύναμα για ένα γράφο T:*

1. *Το T είναι δέντρο.*
2. *κάθε ζέυγος κόμβων του T ενώνονται από μοναδικό μονοπάτι στο T.*
3. *Το T είναι minimally connected δηλαδή το T είναι συνδεδμενο αλλά το $T - e$ είναι μη συνδεδμένο $\forall$ ακμή $e \in T$.*



Σχήμα 2.15: Παράδειγμα δέντρου.

4. *Το T είναι maximally acyclic δηλαδή το T δεν περιέχει κύκλου ωστόσο το $T + xy$ έχει κύκλους $\forall$ ζεύγο non-adjacent κόμβων $x, y \in T$.*

Με τη βοήθεια του παραπάνω θεωρήματος, αποδεικνύεται ότι κάθε συνδεδεμένος γράφος περιέχει spanning tree. (Diestel, 2000).

Πρόταση 2.4.1. *Οι κόμβοι ενός δέντρου μπορούν να απαριθμηθούν: $v_1, \ldots, v_n$ έτσι ώστε κάθε v_i, $i \geq 2$ να έχει μοναδικό γείτονα στο $\{v_1, \ldots, v_{i-1}\}$.*

Πρόταση 2.4.2. *Ένας συνδεδεμένος γράφος με n κόμβους είναι δέντρο εάν και μόνο αν έχει $n - 1$ ακμές.*

Πρόταση 2.4.3. *Εάν T δέντρο και G οποιοσδήποτε γράφος με $\delta(G) \geq |T| - 1$ τότε $T \subseteq G$, δηλαδή ο G έχει υπογράφο ισομορφικό στο T.*

Επιλέγουμε έναν κόμβο του δέντρου που ονομάζουμε *ρίζα*. Ένα δέντρο με συγκεκριμένη ρίζα ονομάζεται rooted tree. Επιλέγοντας μία ρίζα r σε ένα δέντρο T επιβάλουμε διάταξη στο $V(T)$ επιλέγοντας $x \leq y$ εάν $x \in rTy$. Αυτή είναι και *τάξη-δέντρου* στο $V(T)$ που σχετίζεται με τα T και r. Σημειώνουμε ότι:

- Το r είναι το τελευταίο στοιχείο σε αυτή τη διάταξη.
- Κάθε φύλλο $x \neq r$ του T είναι μέγιστο.
- Τα άκρα κάθε ακμής του T είναι συγκρίσιμα.
- Κάθε σύνολο της μορφής $\{x | x \leq y\}$, όπου y είναι οποιοσδήποτε σταθερός κόμβος, είναι *αλυσίδα* δηλαδή ένα σύνολο από ανα δύο συγκρίσιμα στοιχεία.

Ορισμός 2.4.3. *Ένα rooted tree που περιέχεται στο γράφο G ονομάζεται* ***κανονικό*** *στο G εάν τα άκρα κάθε $T-$μονοπατιού στο G είναι συγκρίσιμα στη διάταξη-δέντρου του T.7.*

Εάν το T καλύπτει (spans) το G τότε 2 κόμβοι του T πρέπει να είναι συγκρίσιμοι όταν είναι προσπίπτωτονες (adjacent) στο G.

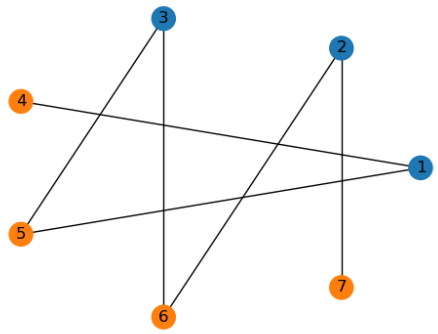
Τα κανονικά spanning δεντρα συχνά ονομάζονται και depth-first search treeskαι αποτελούν σημαντικά εργαλεία στη θεωρία γραφημάτων (Diestel, 2000).

Πρόταση 2.4.4. *Κάθε συνδεδεμένος γράφος περιέχει έναν κανονικό spanning tree με οποιουδήποτε κόμβο ως ρίζα. (Diestel, 2000)*

2.5 Bipartite graphs

Ορισμός 2.5.1. *Έστω $r \geq 2$, $r \in \mathbb{Z}$. Ένα γράφος $G = (V, E)$ ονομάζεται **r-partite** εάν το V χωρίζεται σε r κλάσεις διαμέρισης τ.ω. κάθε ακμή να έχει άκρα σε διαφορετικές κλάσεις.*

Σημειώνουμε ότι κόμβοι στην ίδια κλάση διαμέρισης δεν πρέπει να είναι adjacent. Επίσης, αντί για 2-partite θα λέμε bipartite (Σχήμα 2.16).



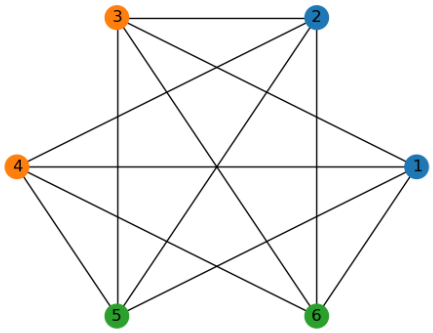
Σχήμα 2.16: Παράδειγμα bipartite γραφήματος.

Ορισμός 2.5.2. *Ένα r-partite γράφημα με κάθε δύο κόμβους από διαφορετικές κλάσεις να είναι adjacent ονομάζεται πλήρες.*

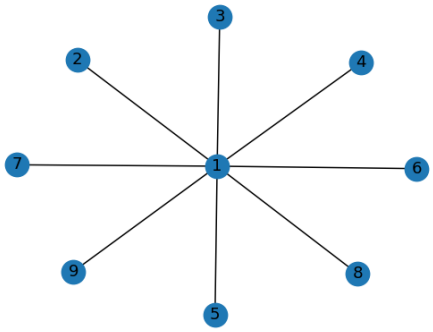
Το πλήρες r-partite γράφημα $\bar{K^{n_1}} * \ldots * \bar{K^{n_r}}$ συμβολίζεται $K_{n_1,\ldots,n_r}$. Επίσης εάν $n_1 = \ldots == n_r = s$ γράφουμε K_s^r, δηλαδή το K_s^r είναι το πλήρες r-partite γράφημα στο οποίο κάθε κλάση διαμέρισης περιέχει ακριβώς s κόμβους (Σχήμα 2.17).

Μια ειδική κατηγορία τέτοιων γράφων είναι οι $K_{1,n}$ που ονομάζονται και *άστρα* (Σχήμα 2.18).

Πρόταση 2.5.1. *Ένας γράφος είναι bipartite αν και μόνο αν δεν περιέχει κανέναν περιττό κύκλο. (Diestel, 2000).*



Σχήμα 2.17: Ένας partite K_2^3 γράφος.



Σχήμα 2.18: Ένα άστρο $K_{1,9}$.

2.6 Στοιχεία γραμμική άλγεβρας

Ορισμός 2.6.1. *Έστω $G = (V, E)$ ένας γράφος με n κόμβους και m ακμές:*

$$V = \{v_1, \ldots, v_n\}$$
$$E = \{e_1, \ldots, e_m\}$$

Ο χώρος κόμβων (vertex space) $\mathcal{V}(G)$ του G είναι ο διανυσματικός χώρος πάνω στο πεδίο $\mathbb{F}_2 = \{0, 1\}$ όλων των συναρτήσεων $V \to \mathbb{F}_2$.

Κάθε στοιχείο του $\mathcal{V}(G)$ αντιστοιχεί φυσικά σε ένα υποσύνολο του V, το σύνολο των κόμβων στους οποίους αντιστοιχεί την τιμή 1, ενώ κάθε υποσύνολο του V αναπαρίσταται μοναδικά στο $\mathcal{V}(G)$ από την δείκτρια συνάρτηση.

Μπορούμε να σκεφτόμαστε το $\mathcal{V}(G)$ ως το δυναμοσύνολο του V ως διανυσματικό χώρο για τον οποίο έχουμε:

- *το άθροισμα $U + U'$ δύο συνόλων κόμβων $U, U' \subseteq V$ είναι η συμμετρική διαφορά τους*
- *$U = -U$ για κάθε $U \subseteq V$*
- *Το μηδενικό στοιχειο στο $\mathcal{V}(G)$ είναι το κενό σύνολο κόμβων $\emptyset$.*
- *Εφόσον $\{\{v_1\}, \ldots, \{v_n\}\}$ είναι μία βάση στο $\mathcal{V}(G)$ (η κανονική βάση) τότε $\dim(\mathcal{V}(G)) = n$*

Ομοίως, συναρτήσεις $E \to \mathbb{F}_2$ ορίζουν τον χώρο ακμών (edge space) $\mathcal{E}(G)$ του G, τα στοιχεία του οποίου, είναι υποσύνολα του E. Έχουμε τις εξής ιδιότητες:

- *το διανυσματικο άθροισμα δίνει τη συμμετρική διαφορά*
- *το $\emptyset \subseteq E$ είναι το μηδενικό στοιχείο*
- *$F = -F$ για κάθε $F \subseteq E$*
- *$\{\{e_1\}, \ldots, \{e_m\}\}$ είναι η κανονική βάση του $\mathcal{E}(G)$ και $\dim(\mathcal{E}(G)) = m$.*

Έστω δύο σύνολα ακμών $F, F' \in \mathcal{E}(G)$ και $\lambda_1, \ldots, \lambda_m$ και $\lambda'_1, \ldots, \lambda'_m$ οι συντελεστές τους ως προς την κανονική βάση αντίστοιχα, θα γράφουμε:

$$\langle F, F' \rangle = \lambda_1 \lambda'_1 + \ldots + \lambda_m \lambda'_m$$

σημειώνουμε ότι $\langle F, F' \rangle = 0$ ακόμα και αν $F = F' \neq \emptyset$. Πράγματι, $\langle F, F' \rangle = 0$ αν και μόνο αν F, F' έχουν άρτιο το πλήθος κοινών ακμών.

Δοθέντος ενός υπόχωρου $\mathcal{F}$ του $\mathcal{E}(G)$:

$$\mathcal{F}^{\perp} := \{D \in \mathcal{E}(G) \ \langle F, D \rangle = 0 \, \forall F \in \mathcal{F}\}$$

Ο $\mathcal{F}^{\perp}$ είναι επίσης υπόχωρος του $\mathcal{E}(G)$ και έχουμε:

$$\dim(\mathcal{F}) + \dim(\mathcal{F}^{\perp}) = m$$

Ο χώρος κύκλων $\mathcal{C} = \mathcal{C}(G0$ είναι ο υπόχωρος του $\mathcal{E}(G)$ που παράγεται από όλους τους κύκλους στο G (τα σύνολα ακμών τους). Η διάσταση του $\mathcal{C}(G)$ ονομάζεται κυκλοματικός αριθμός του G (cyclomatic number).

Ορισμός 2.6.2. *Οι επαγώμενοι κύκλοι στο G παράγουν το σύνολο του κυκλικού χώρου.*

Πρόταση 2.6.1. *Ένα σύνολο ακμών $F \subseteq E$ βρίσκεται εντός του $\mathcal{C}(G)$ αν και μόνο αν κάθε κόμβος του (V, F) έχει άρτιο βαθμό.*

Ορισμός 2.6.3. *Εάν $\{V_1, V_2\}$ μία διαμέριση του V, το σύνολο $E(V_1, V_2)$ όλων των ακμών του G που κάνουν cross αυτή τη διαμέριση ονομάζεται cut .*

Για $V_1 = \{v\}$ αυτό το cut συμβολίζεται με $E(v)$.

*Ο **πίνακας πρόσπτωσης** (incidence matrix) $B = (b_{ij})_{n\times m}$ ένος γράφου $G = (V, E)$ με $V = \{v_1, \ldots, v_n\}$ και $E =_1, \ldots, e_n\}$ ορίζεται στον $\mathbb{F}_{\not=}$ ως:*

$$b_{ij} = \begin{cases} 1 & \text{, αν } v_i \in e_j \\ 0 & \text{, διαφορετικά} \end{cases}$$

Ορισμός 2.6.4. *Ο πίνακας γειτνίασης (adjacency matrix) $A = (a_{ij})_{n\times n}$ ενός γραφήματος G ορίζεται ως:*

$$a_{ij} = \begin{cases} 1 & \text{, αν } v_i v_j \in E \\ 0 & \text{, διαφορετικά} \end{cases}$$

Εάν D πραγματικός διαγώνιος πίνακας $n \times n$ με $d_{ii} = d(v_i)$ και $d_{ij} = 0$ διαφορετικά, τότε:

$$BB^T = A + D$$

2.7 Ειδικές κατηγορίες γράφων

Υπεργράφος Ένας υπεργράφος είναι ένα ζεύγος (V, E) από ξένα μεταξύ τους σύνολα όπου τα στοιχεία του E είναι μη κενά υποσύνολα (οποιασδήποτε cardinality) του V.

Κατευθυνόμενος γράφος Ένας κατευθυνόμενος γράφος είναι ένα ζεύγος (V, E) ξένων μεταξύ τους συνόλων (κόμβων και ακμών) μαζί με δύο απεικονίσεις:

$$init : E \to V$$
$$ter :: E \to V$$

που αντιστοιχούν σε κάθε ακμή e, έναν αρχικό κόμβο $init(e)$ και έναν τελικό κόμβο $ter(e)$. Η ακμή e λέγεται ότι έχει κατεύθυνση από τον κόμβο $init(e)$ στον $ter(e)$. Εάν $init(e) = ter(e)$ τότε η ακμή e ονομάζεται loop . (Diestel, 2000)

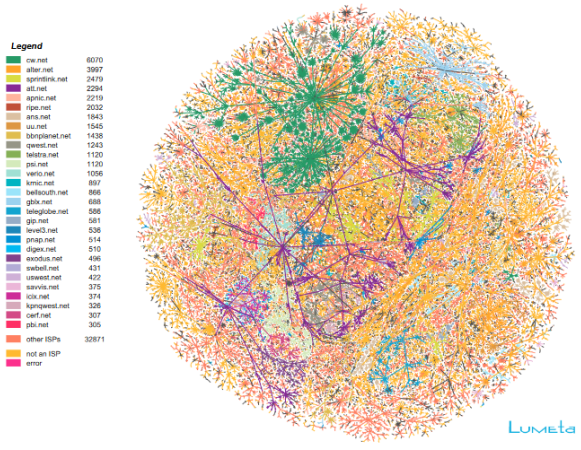
Ένας κατευθυνόμενος γράφος συχνά αποκαλείται και δίκτυο (network).

Κεφάλαιο 3

Αλγόριθμος PageRank

3.1 Εισαγωγή

Ο αλγόριθμος PageRank αποτελεί έναν ευρέως διαδεδομένο αλγόριθμο μέτρησης της επιδραστικότητας των κόμβων δικτύων. Ο αλγόριθμος πρωτοεμφανίστηκε το 1998 στην ιστορική πλέον δημοσίευση των Sergey Brin και Larry Page ιδρυτών της εταιρείας Google (Langeville & Meyer, 2006). Στον συγκεκριμένο αλγόριθμο βασίζεται η ίδια η λειτουργία της μηχανής αναζήτησης Google. Αξίζει να σημειωθεί ότι παράλληλα με τους Brin και Page ο Jon Kleiberg σήμερα καθηγητής στο Πανεπιστήμιο Cornell παρουσίασε τη δική του εκδοχή ενός αλγορίθμου μέτρησης της επιδραστικότητας των κόμβων του Διαδικτύου που ονόμασε HITS (Hypertext Induced Topic Search).



Σχήμα 3.1: Ένα μέρος του στιγμιαίου γράφου του Διαδικτύου. (Dodge & Kitchin, 2001).

3.2 Βασική ιδέα

Αρχικά θεωρούμε το χώρο του Διαδικτύου ως δίκτυο με τις ιστοσελίδες να αποτελούν τους κόμβους του δικτύου. Εάν μία ιστοσελίδα περιέχει ένα σύνδεσμο προς μία άλλη τότε θεωρούμε πως οι δύο κόμβοι συνδέονται λαμβάνοντας υπ όψιν και την κατεύθυνηση της σύνδεσης. Αξίζει να σημειωθεί ότι η συγκεκριμένη παραδοχή δίνει μία στατική εικόνα του Διαδικτύου θεωρώντας πως το σύνολο του Διαδικτύου μπορεί να περιγραφεί επαρκώς από ένα στιγμιότυπό του. Ανανεώνοντας φυσικά το στιγμιότυπο μπορούμε να έχουμε και μια ακριβέστερη αναπαράστασή του.

Η βασική ιδέα πίσω από τη σύλληψη του αλγορίθμου PageRank είναι ιδιαίτερα απλή γι αυτό και ο ίδιος ο αλγόριθμος έχει κάποιου τύπου κομψότητα [1]. Η ιδιοφυής σύλληψη των Brin και Page συνοψίζεται στην εξής πρόταση:

Μία ιστοσελίδα είναι σημαντική εάν άλλες σημαντικές ιστοσελίδες δείχνουν σε αυτή.

Μια άλλη σημαντική οντότητα του αλγορίθμου είναι αυτή του **τυχαίου περιηγητή** (random surfer). Θεωρούμε έναν χρήστη ο οποίος σερφάρει στο Διαδίκτυο. Ο χρήστης ανοίγει μία ιστοσελίδα και επιλέγει τυχαία να κάνει κλικ σε κάποιο σύνδεσμο/λινκ που βρίσκεται στη συγκεκριμένη ιστοσελίδα. Ο χρήστης επαναλαμβάνει τη συγκεκριμένη διαδικασία συνεχώς. Στην πραγματικότητα, η συγκεκριμένη οντότητα αποτελεί έναν τυχαίο περιπατητή επί του γράφου του Διαδικτύου.

Έτσι, θεωρούμε το γράφημα του Διαδικτύου $G = (V, E)$ όπου V το σύνολο των κόμβων/ιστοσελίδων και E το σύνολο των συνδέσεων μεταξύ των ιστοσελίδων. Σκοπός μας είναι η εύρεση της επιδραστικότητας κάθε κόμβου/ιστοσελίδας. Για το σκοπό αυτό μία τιμή σε κάθε κόμβο του δικτύου το οποίου καλούμε PageRank score.

Έστω $r(i)$ το PageRank score του κόμβου i, θα έχουμε:

$$r(i) = \sum_{j \in B_i} \frac{r(j)}{deg^{out}(j)}$$

όπου:

- B_i: το σύνολο των ιστοσελίδων που δείχνουν την ιστοσελίδα i, δηλαδή το σύνολο των κόμβων που ενώνονται με μία κατευθυνόμενη ακμή με τον i:
$$B_i = \{k : ki \in E\}$$

- $deg^{out}(j)$: ο αριθμός των ακμών που κατευθύνονται προς τον κόμβο j δηλαδή ο βαθμός outlink του κόμβου j.

[1] τα μαθηματικά άλλωστε δεν αρκεί να είναι ορθά πρέπει να είναι και... όμορφα!

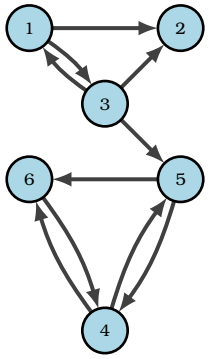
3.3 Υπολογιστική διαδικασία

Ο υπολογισμός των PageRank score πραγματοποιείται μέσω μίας επαναληπτικής διαδικασίας. Για την εκκίνηση της υπολογιστικής διαδικασίας θεωρούμε ότι όλοι οι κόμβοι έχουν αρχικά ίδια τιμή PageRank ίση με $1/n$ όπου n το πλήθος των κόμβων. Έτσι μία απλή υπολογιστική διαδικασία υπολογισμού των τιμών του αλγορίθμου είναι η εξής :

$$\begin{cases} r_{k+1}(i) = \sum_{j \in B_i} \frac{r_k(j)}{deg^{out}(j)} \\ r_0(i) = 1/n \end{cases} \tag{3.1}$$

εφαρμόζοντας τον υπολογισμό $\forall i$.

Για παράδειγμα, εφαρμόζουμε τον παραπάνω αλγόριθμο στο δίκτυο του Σχήματος 3.2.



Σχήμα 3.2: Δίκτυο $G = (V, E)$ με $V = \{1, 2, 3, 4, 5, 6\}$ και $E = \{(1, 2), (1, 3), (3, 1), (3, 2), (3, 5), (4, 5), (4, 6), (5, 4), (5, 6), (6, 4)\}$

i	$deg^{out}(i)$	B_i
1	$deg^{out}(1) = 2$	$B_1 = \{3\}$
2	$deg^{out}(2) = 0$	$B_2 = \{1, 3\}$
3	$deg^{out}(3) = 3$	$B_3 = \{1\}$
4	$deg^{out}(4) = 2$	$B_4 = \{5, 6\}$
5	$deg^{out}(5) = 2$	$B_5 = \{3, 4\}$
6	$deg^{out}(6) = 1$	$B_6 = \{4, 5\}$

Πίνακας 3.1: Αριθμός outbound links και σύνολα ιστοσελίδων που δείχνουν κάθε κόμβο για το γράφημα του Σχήματος 3.2.

Για $k = 0$:

$$r_0(i) = 1/6, \qquad \forall i = 1, \ldots, 6$$

Για $k = 1$:

$$r_1(1) = \sum_{j \in B_1} \frac{r_0(j)}{deg^{out}(j)} = \frac{r_0(3)}{deg^{out}(3)} = \frac{1/6}{3} = \frac{1}{18} \simeq 0.0555556$$

$$r_1(2) = \sum_{j \in B_2} \frac{r_0(j)}{deg^{out}(j)} = \frac{r_0(1)}{deg^{out}(1)} + \frac{r_0(3)}{deg^{out}(3)} = \frac{1/6}{2} + \frac{1/6}{3} = \frac{1}{12} + \frac{1}{18} = \frac{5}{36} \simeq 0.138889$$

$$r_1(3) = \sum_{j \in B_3} \frac{r_0(j)}{deg^{out}(j)} = \frac{r_0(1)}{deg^{out}(1)} = \frac{1/6}{2} = \frac{1}{12} \simeq 0.0833333$$

$$r_1(4) = \sum_{j \in B_4} \frac{r_0(j)}{deg^{out}(j)} = \frac{r_0(5)}{deg^{out}(5)} + \frac{r_0(6)}{deg^{out}(6)} = \frac{1/6}{2} + \frac{1/6}{1} = \frac{1}{12} + \frac{1}{6} = \frac{1}{4} = 0.25$$

$$r_1(5) = \sum_{j \in B_5} \frac{r_0(j)}{deg^{out}(j)} = \frac{r_0(3)}{deg^{out}(3)} + \frac{r_0(4)}{deg^{out}(4)} = \frac{1/6}{3} + \frac{1/6}{2} = \frac{1}{18} + \frac{1}{12} = \frac{5}{36} \simeq 0.138889$$

$$r_1(6) = \sum_{j \in B_6} \frac{r_0(j)}{deg^{out}(j)} = \frac{r_0(4)}{deg^{out}(4)} + \frac{r_0(5)}{deg^{out}(5)} = \frac{1/6}{2} + \frac{1/6}{2} = \frac{1}{12} + \frac{1}{12} = \frac{1}{6} \simeq 0.166667$$

Για $k = 2$:

$$r_2(1) = \sum_{j \in B_1} \frac{r_1(j)}{deg^{out}(j)} = \frac{r_1(3)}{deg^{out}(3)} = \frac{1/12}{3} = \frac{1}{36} \simeq 0.0277778$$

$$r_2(2) = \sum_{j \in B_2} \frac{r_1(j)}{deg^{out}(j)} = \frac{r_1(1)}{deg^{out}(1)} + \frac{r_1(3)}{deg^{out}(3)} = \frac{1/18}{2} + \frac{1/12}{3} = \frac{1}{36} + \frac{1}{36} = \frac{1}{18} \simeq 0.0555556$$

$$r_2(3) = \sum_{j \in B_3} \frac{r_1(j)}{deg^{out}(j)} = \frac{r_1(1)}{deg^{out}(1)} = \frac{1/18}{2} = \frac{1}{36} \simeq 0.0277778$$

$$r_2(4) = \sum_{j \in B_4} \frac{r_1(j)}{deg^{out}(j)} = \frac{r_1(5)}{deg^{out}(5)} + \frac{r_1(6)}{deg^{out}(6)} = \frac{5/36}{2} + \frac{1/6}{1} = \frac{5}{74} + \frac{1}{6} = \frac{26}{111} \simeq 0.234234$$

$$r_2(5) = \sum_{j \in B_5} \frac{r_1(j)}{deg^{out}(j)} = \frac{r_1(3)}{deg^{out}(3)} + \frac{r_1(4)}{deg^{out}(4)} = \frac{1/12}{3} + \frac{1/4}{2} = \frac{1}{36} + \frac{1}{8} = \frac{11}{72} \simeq 0.152778$$

$$r_2(6) = \sum_{j \in B_6} \frac{r_1(j)}{deg^{out}(j)} = \frac{r_1(4)}{deg^{out}(4)} + \frac{r_1(5)}{deg^{out}(5)} = \frac{1/4}{2} + \frac{5/36}{2} = \frac{1}{8} + \frac{5}{74} = \frac{41}{296} \simeq 0.138514$$

Έχοντας υπολογίσει ήδη μόνο δύο βήματα του αλγορίθμου αναδεικνύεται μία σημαντική πτυχή του. Παρατηρούμε ότι οι κόμβοι 4, 5, 6 παρουσιάζουν σημαντικά μεγαλύτερο PageRank score σε σχέση με τους 1, 2, 3 των οποίων οι τιμές φαίνεται να τείνουν στο 0. Πράγματι, παρατηρώντας το Σχήμα 3.2 παρατηρούμε ότι εάν ένας τυχαίος περιπατητής ταξιδεύει εντός του γράφου G, τότε κάποια στιγμή αναπόφευκτα θα παγιδευτεί εντός του υπογράφου των κόμβων 4, 5 και 6. Έτσι, αναπόφευκτα οι κόμβοι 4, 5, 6 θα έχουν μεγάλη τιμή PageRank όταν ο αριθμός των επαναλήψεων είναι μεγάλος. Κάτι τέτοιο είναι αρκετά συχνό στο Διαδίκτυο και υπογράφοι ή

κόμβοι με παρόμοια συμπεριφορά ονομάζονται **καταβόθρες** (**rank sinks**). Οι καταβόθρες αυτές δημιουργούν προβλήματα στην κατάταξη της επιδραστικότητας των κόμβων του δικτύου καθώς μονοπωλούν στη συσσώρευση τιμών PageRank . Ωστόσο, υπάρχει τρόπος να ξεπεράσουμε το συγκεκριμένο πρόβλημα όπως και αναλύεται στη συνέχεια.

3.4 Αναπαράσταση πινάκων

Ιδιαίτερα χρήσιμο είναι να περάσουμε σε μία αναπαράσταση του αλγορίθμου με πίνακες. Ορίζουμε τον πίνακα $\mathbf{H}$ έναν $n \times n$ hyperlink πίνακα κανονικοποιημένο ως προς τις γραμμές με στοιχεία:

$$H_{ij} = \begin{cases} 1/deg^{out}(i) & , (i,j) \in E \\ 0 & , (i,j) \notin E \end{cases} \tag{3.2}$$

Αξίζει να σημειωθεί ότι ο πίνακας $\mathbf{H}$ έχει την ίδια δομή με τον πίνακα γειτνίασης (adjacency matrix). Ωστόσο τα μη μηδενικά στοιχεία του είναι πιθανότητες.

Για παράδειγμα ο πίνακας $\mathbf{H}$ του δικτύου του Σχήματος 3.2 είναι ο εξής:

$$\mathbf{H} = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Επίσης ορίζουμε ένα $1 \times n$ διάνυσμα $\boldsymbol{\pi}^T$ τα στοιχεία του οποίου είναι η τιμή PageRank κάθε κόμβου. Το αντίστοιχο διάνυσμα στην k επανάληψη ορίζεται ως $\boldsymbol{\pi}^{(k)T}$. Έτσι ο αναδρομικός τύπος του αλγορίθμου γράφεται ως:

$$\begin{aligned} (3.1) \Rightarrow & r_{k+1}(i) = \sum_{j \in B_i} \frac{r_k(j)}{deg^{out}(j)} \\ \Rightarrow & r_{k+1}(i) = \sum_{j=1}^{n} r_k(j) H_{ij} \\ \Rightarrow & \boldsymbol{\pi}^{(k+1)T} = \boldsymbol{\pi}^{(k)T} \mathbf{H} \end{aligned} \tag{3.3}$$

και σύμφωνα με την (3.1) το αρχικό διάνυσμα είναι το:

$$\boldsymbol{\pi}^{(0)T} = \frac{1}{n} \mathbf{e}^T \tag{3.4}$$

όπου $\mathbf{e}^T$: το $1 \times n$ διάνυσμα με στοιχεία μονάδες.

Τόσο η δομή του πίνακα $\mathbf{H}$ όσο και η επιλογή για το αρχικό διάνυσμα δεν είναι μονόδρομος. Όπως θα δούμε στη συνέχεια, οδηγούν μάλιστα σε ορισμένα σημαντικά προβλήματα.

3.4.1 Παρατηρήσεις

Η εξίσωση (3.3) οδηγεί σε κάποια πολύ σημαντικά συμπεράσματα (Langville & Meyer, 2006) :

- Κάθε επανάληψη είναι ένας πολλαπλασιασμός πίνακα-διανύσματος είναι δηλαδή ένας $\mathcal{O}(n^2)$ υπολογισμός.

- Ο πίνακας $\mathbf{H}$ είναι αραιός (sparse). Οι αραιοί πίνακες είναι ιδιαίτερα σημαντικοί. Αρχικά απαιτούν σημαντικά λιγότερη μνήμη για την αποθήκευσή τους καθώς για την περιγραφή τους απαιτείται μόνο η τιμή των μη μηδενικών στοιχείων και η θέση τους. Επίσης, ένας πολλαπλασιασμός πίνακα-διανύσματος όπως αυτός της (3.3) όπου ο πίνακας είναι αραιός, απαιτεί πολύ λιγότερο χρόνο υπολογισμού από $\mathcal{O}(n^2)$. Ο υπολογισμός είναι της τάξης του $\mathcal{O}(nnz(\mathbf{H}))$ όπου $nnz(\mathbf{H})$ είναι ο αριθμός των μη μηδενικών στοιχείων του πίνακα $\mathbf{H}$. Στην πράξη, ο υπολογισμός πέφτει από $\mathcal{O}(n^2)$ σε $\mathcal{O}(n)$.

- Ο αλγόριθμος αποτελεί ουσιαστικά εφαρμογή της μεθόδου των δυνάμεων για τον πίνακα $\mathbf{H}$. Η υπολογιστική διαδικασία αποτελεί επομένως simple linear stationary process και το πρόβλημα ανάγεται στον υπολογισμό της κύριας ιδιοτιμής (principal eigenvalue) του πίνακα.

- Ο πίνακας $\mathbf{H}$ μοιάζει αρκετά με έναν στοχαστικό πίνακα πιθανοτήτων μετάβασης (stochastic transition probability) μίας μαρκοβιανής αλυσίδας. Ο πίνακας είναι συγκεκριμένα υποστοχαστικός (substochastic).

Στο σημείο αυτό τίθενται ορισμένα ερωτήματα ως προς την υπολογιστική διαδικασία :

- η υπολογιστική διαδικασία παρουσιάζει σύγκλιση ;

- ποιες προϋποθέσεις πρέπει να ικανοποιεί ο πίνακας $\mathbf{H}$ ώστε η διαδικασία συγκλίνει ;

- συγκλίνει μοναδικά ;

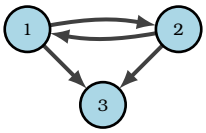
- εξαρτάται η σύγκλιση από την επιλογή του αρχικού διανύσματος $\boldsymbol{\pi}^{(0)T}$;

- εάν συγκλίνει, σε πόσο μεγάλο αριθμό επαναλήψεων ;

Προσπαθώντας να απαντήσουμε τα παραπάνω ερωτήματα σύντομα θα καταλάβουμε ότι τα πράγματα δεν είναι τόσο ρόδινα όσο φαίνονται.

3.4.2 Ορισμένα προβλήματα

Επιλογή αρχικού διανύσματος Στη συνέχεια παρουσιάζονται ορισμένα προβλήματα του αλγορίθμου όπως έχει περιγραφεί ως τώρα. Η επιλογή για το αρχικό διάνυσμα σύμφωνα με την (3.4) αναδεικνύει το πρόβλημα των **καταβόθρων** (**rank sinks**). Καταβόθρες μπορεί να είναι μεμονωμένοι κόμβοι αλλά και σύνολα κόμβων που συγκεντρώνουν όλο και περισσότερο PageRank score σε κάθε επανάληψη μονοπωλώντας στους υπολογισμούς. Για παράδειγμα, στο Σχήμα 3.3 ο κόμβος 3 αποτελεί καταβόθρα, ενώ στο Σχήμα 3.2 το υπογράφημα των κόμβων $4, 5, 6$ αποτελεί καταβόθρα όπως φαίνεται άλλωστε ήδη από τη δεύτερη επανάληψη του αλγορίθμου που υπολογίστηκε παραπάνω.



Σχήμα 3.3: Ένας απλός γράφος με καταβόθρα (rank sink).

Για την βαθύτερη κατανόηση του προβλήματος των καταβόθρων υλοποιούμε την απλή αυτή εκδοχή του αλγορίθμου των εξισώσεων (3.4), (3.3) σε ένα `Jupyter notebook` [2] χρησιμοποιώντας τη γλώσσα Python[3]. Αρχικά, εισάγουμε το δίκτυο του Σχήματος 3.2 χρησιμοποιώντας τη βιβλιοθήκη (module) διαχείρισης γραφημάτων `networkx` ενώ στη συνέχεια θα χρησιμοποιήσουμε τόσο το module `matplotlib` για τη σχεδίαση γραφικών παραστάσεων όσο και το module αριθμητικών υπολογισμών `numpy`:

```
import networkx as nx
import matplotlib.pyplot as plt
import numpy as np
```

```
G = nx.DiGraph()  # a directed graph object
for i in range(6): G.add_node(i + 1)
G.add_edge(1, 2)
G.add_edge(1, 3)
G.add_edge(3, 1)
G.add_edge(3, 2)
G.add_edge(4, 5)
G.add_edge(4, 6)
G.add_edge(5, 4)
G.add_edge(5, 6)
G.add_edge(6, 4)
G.add_edge(3, 5)
```

Ο πίνακας γειτνίασης (adjacency matrix) υπολογίζεται χρησιμοποιώντας την αντίστοιχη συνάρτηση του `networkx` και αποθηκεύεται στη μνήμη υπό μορφή αραιού πίνακα (sparse matrix):

[2] Το Project Jupyter αποτελεί μία προσπάθεια για ανάπτυξη λογισμικού ανοικτής πηγής και διαδραστικό προγραμματισμό για πληθώρα γλωσσών προγραμματσμού με έμφαση στις γλώσσες Julia, Python και R.

[3] Το σύνολο του κώδικα είναι διαθέσιμο στη σελίδα `https://github.com/lbitsiko/twitter-influence-master-thesis`

```
adj_matrix = nx.adjacency_matrix(G)
adj_matrix
```

```
<6x6 sparse matrix of type '<class 'numpy.intc'>'
with 10 stored elements in Compressed Sparse Row format>
```

έτσι ο πίνακας γειτνίασης είναι:

```
adj_matrix.todense()
```

```
matrix([[0, 1, 1, 0, 0, 0],
        [0, 0, 0, 0, 0, 0],
        [1, 1, 0, 0, 1, 0],
        [0, 0, 0, 0, 1, 1],
        [0, 0, 0, 1, 0, 1],
        [0, 0, 0, 1, 0, 0]], dtype=int32)
```

Χρησιμοποιώντας τον πίνακα γειτνίασης υπολογίζουμε τον $\mathbf{H}$ αντικαθιστώντας τα μη μηδενικά στοιχεία του σύμφωνα με την (3.2):

```
H = nx.adjacency_matrix(G).astype(float)
out_degree_for_nodes_of_G = G.out_degree
for i, j in zip(H.nonzero()[0], H.nonzero()[1]):
    try:
        H[i, j] = 1. / out_degree_for_nodes_of_G[i + 1]
    except ZeroDivisionError:
        print("non zero elements appear to be zero")
H
```

```
<6x6 sparse matrix of type '<class 'numpy.float64'>'
with 10 stored elements in Compressed Sparse Row format>
```

```
    H.todense()
```

```
matrix([[0.        , 0.5       , 0.5       , 0.        , 0.        ,
         0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        ],
        [0.33333333, 0.33333333, 0.        , 0.        , 0.33333333,
         0.        ],
```

```
        [0.        , 0.        , 0.        , 0.        , 0.5       ,
         0.5       ],
        [0.        , 0.        , 0.        , 0.5       , 0.        ,
         0.5       ],
        [0.        , 0.        , 0.        , 1.        , 0.        ,
         0.        ]])
```

Εκτελούμε ενδεικτικά 10 επαναλήψεις του αλγορίθμου:

```
pi0 = H.shape[0]*[1/H.shape[0]] # elements 1/n
pi0 = np.array(pi0)
print('iteration: 0\n', pi0)
print('\n')
for i in range(10):
    pi_new = np.matmul(pi0,H.todense())
    print(f'iteration: {i+1}\n', pi_new)
    print('\n')
    pi0 = pi_new
```

```
iteration: 0
 [0.16666667 0.16666667 0.16666667 0.16666667 0.16666667 0.16666667]


iteration: 1
 [[0.05555556 0.13888889 0.08333333 0.25       0.13888889 0.16666667]]


iteration: 2
 [[0.02777778 0.05555556 0.02777778 0.23611111 0.15277778 0.19444444]]


iteration: 3
 [[0.00925926 0.02314815 0.01388889 0.27083333 0.12731481 0.19444444]]


iteration: 4
 [[0.00462963 0.00925926 0.00462963 0.25810185 0.1400463  0.19907407]]
```

```
iteration: 5
 [[0.00154321 0.00385802 0.00231481 0.26909722 0.13059414 0.19907407]]


iteration: 6
 [[0.0007716  0.00154321 0.0007716  0.26437114 0.13532022 0.19984568]]


iteration: 7
 [[2.57201646e-04 6.43004115e-04 3.85802469e-04 2.67505787e-01
  1.32442773e-01 1.99845679e-01]]


iteration: 8
 [[1.28600823e-04 2.57201646e-04 1.28600823e-04 2.66067065e-01
  1.33881494e-01 1.99974280e-01]]


iteration: 9
 [[4.28669410e-05 1.07167353e-04 6.43004115e-05 2.66915027e-01
  1.33076400e-01 1.99974280e-01]]


iteration: 10
 [[2.14334705e-05 4.28669410e-05 2.14334705e-05 2.66512480e-01
  1.33478947e-01 1.99995713e-01]]
```

Επιβεβαιώνουμε έτσι ότι οι κόμβοι 1,2,3 λαμβάνουν μηδενικό PageRank score ενώ το PageRank score των

4,5,6 φαίνεται να συγκλίνει στις ακόλουθες τιμές:

$$r(4) = 4/15 \simeq 0.267$$
$$r(5) = 2/15 \simeq 0.133$$
$$r(6) = 1/5 \simeq 0.20$$

Το συγκεκριμένο παράδειγμα αναδεικνύει ένα ακόμα πρόβλημα του αλγορίθμου που οφείλεται στην ύπαρξη καταβόθρων. Όσο οι κόμβοι καταβόθρες συσσωρεύουν τιμές PageRank, κάποιοι κόμβοι μπορεί να καταλήξουν να έχουν μηδενικές τιμές. Έτσι, η μέτρηση της επιδραστικότητας κάθε κόμβου γίνεται ιδιαίτερα δύσκολη. Ιδανικά θα θέλαμε το διάνυσμα $\boldsymbol{\pi}^T$ να είναι αυστηρά θετικό.

Ένα άλλο σημαντικό πρόβλημα είναι αυτό των κύκλων. Η ύπαρξη κύκλων εντός του γράφου μπορεί να οδηγήσει σε μη σύγκλιση του αλγορίθμοου και ατέρμονους βρόγχους. Για παράδειγμα ας θεωρήσουμε το γράφημα του Σχήματος 3.4. Παίρνοντας ως αρχικό διάνυσμα το:

$$\boldsymbol{\pi}^{(0)T} = \begin{pmatrix} 0 & 1 \end{pmatrix}$$
$$\Rightarrow \boldsymbol{\pi}^{(1)T} = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

ο αλγόριθμος δεν παρουσιάζει σύγκλιση καθώς σε κάθε βήμα το διάνυσμα παίρνει είτε την αρχική τιμή (εάν k άρτιο) είτε την $\boldsymbol{\pi}^{(1)T}$ (εάν k περιττό).

Σχήμα 3.4: Ένας γράφος με κύκλο.

Το συγκεκριμένο αποτέλεσμα επαληθεύεται χρησιμοποιώντας την απλή υλοποίηση του αλγορίθμου σε Python:

```
# Cycles
G = nx.DiGraph()  # a directed graph object
G.add_node(1)
G.add_node(2)
G.add_edge(1, 2)
G.add_edge(2, 1)

H = nx.adjacency_matrix(G).astype(float)
out_degree_for_nodes_of_G = G.out_degree
for i, j in zip(H.nonzero()[0], H.nonzero()[1]):
    try:
```

```
        H[i, j] = 1. / out_degree_for_nodes_of_G[i + 1]
    except ZeroDivisionError:
        print("non zero elements appear to be zero")

pi0 = np.array([0, 1.])
print('iteration: 0\n', pi0)
print('\n')
for i in range(6):
    pi_new = np.matmul(pi0,H.todense())
    print(f'iteration: {i+1}\n', pi_new)
    print('\n')
    pi0 = pi_new
```

```
iteration: 0
 [0. 1.]


iteration: 1
 [[1. 0.]]


iteration: 2
 [[0. 1.]]


iteration: 3
 [[1. 0.]]


iteration: 4
 [[0. 1.]]


iteration: 5
 [[1. 0.]]
```

```
iteration: 6
 [[0. 1.]]
```

Για καλή μας τύχη, τα παραπάνω προβλήματα διορθώνονται τροποποιώντας ελαφρώς τον αλγόριθμο.

3.5 Βελτιώσεις

Αρχικά θα θέλαμε να εφαρμόσουμε τη μέθοδο των δυνάμεων σε ένα πίνακα πιθανοτήτων μετάβασης μίας μαρκοβιανής αλυσίδας (transition probability matrix). Γνωρίζουμε ότι, η μέθοδος των δυνάμεων εφαρμοζόμενη σε έναν μαρκοβιανό πίνακα συγκλίνει για κάθε επιλογή αρχικού διανύσματος σε ένα μοναδικό θετικό διάνυσμα που καλείται στάσιμο διάνυσμα (stationary vector) αρκεί ο πίνακας να είναι στοχαστικός, μη αναγωγήσιμος (irreducible) και απεριοδικός (aperiodic) [4].). Επομένως, για να συγκλίνει ο αλγόριθμος αρκεί να τροποποιήσουμε κατάλληλα τον πίνακα $\mathbf{H}$.

3.5.1 Προσαρμογή στοχαστικότητας

Για τη μετατροπή του πίνακα $\mathbf{H}$ σε στοχαστικό αντικαθιστούμε τις μηδενικές γραμμές $\mathbf{0}^T$ με $1/n\mathbf{e}^T$. Έτσι ο τυχαίος περιπατητής που εισέρχεται σε έναν κόμβο καταβόθρα μπορεί να υπερπηδήσει ισοπίθανα σε οποιοδήποτε κόμβο του γραφήματος. Για παράδειγμα, ο στοχαστικός πίνακας του γραφήματος του Σχήματος 3.2 είναι:

$$\mathbf{S} = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Η μετατροπή του $\mathbf{H}$ στον στοχαστικό πίνακα $\mathbf{S}$ είναι ιδιαίτερα απλή, καθώς απαιτείται μόνο μία rank-one ενημέρωση του πίνακα:

$$\mathbf{S} = \mathbf{H} + \mathbf{a}(1/n\mathbf{e}^T) \tag{3.5}$$

[4] Ένας μη αναγωγήσιμος απεριοδικός πίνακας είναι primitive

όπου **a** ένα διάνυσμα με στοιχεία:

$$a_i = \begin{cases} 1 & \text{, εάν ο κόμβος } i \text{ είναι καταβόθρα} \\ 0 & \text{, διαφορετικά} \end{cases}$$

Μία rank one ενημέρωση υλοποιείται ιδιαίτερα εύκολα στην Python. Συνεχίζουμε την υλοποίηση για τον γράφο του Σχήματος 3.2 έχοντας εισάγει τον πίνακα $\mathbf{H}$ όπως περιγράφηκε παραπάνω.

Το διάνυσμα **a** υλοποιείται χρησιμοποιώντας τη συνάρτηση `np.zeros` που αρχικοποιεί ένα διάνυσμα με μηδενικές τιμές:

```
alphas = np.zeros(H.shape[0])
zero_indexes = np.where(H.getnnz(1)==0)
for i in zero_indexes[0]:
    alphas[i] = 1.0
alphas
```

```
array([0., 1., 0., 0., 0., 0.])
```

αντίστοιχα το διάνυσμα $1/n\mathbf{e}^T$ υλοποιείται χρησιμοποιώντας τη συνάρτηση `np.ones` που αρχικοποιεί ένα διάνυσμα με μονάδες:

```
epsilons = np.ones(H.shape[0])/H.shape[0]
epsilons
```

```
array([0.16666667, 0.16666667, 0.16666667, 0.16666667, 0.16666667,
       0.16666667])
```

και ο στοχαστικός πίνακας $\mathbf{S}$ υπολογίζεται :

```
H + np.dot(alphas[:,None],epsilons[None,:])
```

```
matrix([[0.        , 0.5       , 0.5       , 0.        , 0.        ,
         0.        ],
        [0.16666667, 0.16666667, 0.16666667, 0.16666667, 0.16666667,
         0.16666667],
        [0.33333333, 0.33333333, 0.        , 0.        , 0.33333333,
         0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.5       ,
         0.5       ],
        [0.        , 0.        , 0.        , 0.5       , 0.        ,
         0.5       ],
```

```
[0.        , 0.        , 0.        , 1.        , 0.        ,
 0.        ]])
```

3.5.2 Προσαρμογή primitivity

Για να είναι ο πίνακας primitive πρέπει να είναι μη αναγωγήσιμος και απεριοδικός. Ας φανταστούμε τον τυχαίο περιπατητή πάνω στο γράφημα. Ο περιπατητής κινείται στο γράφημα σύμφωνα με τον πίνακα της εξίσωσης (3.5) τις περισσότερες φορές. Κάποια στιγμή ωστόσο «τηλεμεταφέρεται» σε κάποιο άλλο κόμβο. Έτσι, με μια μεγάλη σχετικά πιθανότητα c κινείται σύμφωνα με την (3.5) ενώ με πιθανότητα $1-c$ επιλέγει τυχαία να τηλεμεταφερθεί σε ένα οποιοδήποτε κόμβο. Επομένως, εισάγουμε έναν νέο πίνακα που αποκαλείται **πίνακας Google** :

$$\mathbf{G} = c\,\mathbf{S} + (1-c)\mathbf{E} \tag{3.6}$$

όπου:

- $c \in [0,1]$ η πιθανότητα ο τυχαίος περιπατητής να συνεχίσει τη διαδικασία hyperlink
- $\mathbf{E} = \frac{1}{n}\mathbf{e}\mathbf{e}^T$ ο πίνακας τηλεμεταφοράς

Εάν $c = 0.8$ τότε στο 80% των περιπτώσεων ο τυχαίος περιπατητής συνεχίζει να κινείται στη δομή του γράφου σύμφωνα με τον στοχαστικό πίνακα $\mathbf{S}$ ενώ στο 40% των περιπτώσεων τηλεμεταφέρεται σε ένα κόμβο.

3.6 Παρατηρήσεις για τον Google matrix

Ο πίνακας Google όπως ορίστηκε στην εξίσωση (3.6) έχει τις εξής ιδιότητες:

1. Ο πίνακας $\mathbf{G}$ είναι στοχαστικός, καθώς οι $\mathbf{S}$ και $\mathbf{E}$ είναι στοχαστικοί.
2. Είναι μη αναγωγήσιμος.
3. Είναι απεριοδικός, καθώς $G_{ii} > 0\,\forall i$
4. Είναι primitive καθώς $\mathbf{G}^k > 0$ και επομένως υπάρχει μοναδικό θετικό διάνυσμα $\boldsymbol{\pi}^T$ έτσι ώστε η μέθοδος των δυνάμεων να συγκλίνει.

5. Είναι πυκνός. Ωστόσο, μπορεί να γραφεί ως μία rank-one ενημέρωση του αραιού πίνακα $\mathbf{H}$:

$$\begin{aligned}\mathbf{G} &= c\,\mathbf{S} + (1-c)\mathbf{E} \\ &= c\left(\mathbf{H} + \mathbf{a}\frac{1}{n}\mathbf{e}^T\right) + (1-c)\frac{1}{n}\mathbf{e}\mathbf{e}^T \\ &= c\mathbf{H} + c\mathbf{a}\frac{1}{n}\mathbf{e}^T + (1-c)\frac{1}{n}\mathbf{e}\mathbf{e}^T \\ &= c\mathbf{H} + (c\mathbf{a} + (1-c)\mathbf{e})\,\frac{1}{n}\mathbf{e}^T \end{aligned} \tag{3.7}$$

3.7 Τροποποιημένος αλγόριθμος PageRank

Έτσι, είμαστε σε θέση πλέον να εφαρμόσουμε τη μέθοδο των δυνάμεων στον πίνακα $\mathbf{G}$:

$$\boldsymbol{\pi}^{(k+1)T} = \boldsymbol{\pi}^{(k)T}\mathbf{G}$$

ο οποίος είναι και ο αλγόριθμος PageRank.

Επομένως η τιμή Pagerank κάθε κόμβου είναι ουσιαστικά η εκάστοτε συνιστώσα του ιδιοδιανύσματος που αντιστοιχεί στη μέγιστη ιδιοτιμή του πίνακα $\mathbf{G}$.

3.8 Εφαρμογή αλγορίθμου

Ως απλή εφαρμογή ανατρέχουμε ξανά στο γράφημα του Σχήματος 3.2. Αφού εισάγουμε το γράφημα χρησιμοποιώντας το module `networkx`, υπολογίσουμε τον πίνακα γειτνίασης $\mathbf{H}$ καθώς και τα διανύσματα $\mathbf{a}$, $\mathbf{e}$, εισάγουμε στη συνέχεια τη σταθερά c δίνοντάς της την αυθαίρετη τιμή 0.85:

```
c = 0.85
```

Υπολογίζουμε το δεξιό διάνυσμα $c\mathbf{a} + (1-c)\mathbf{e}$ σύμφωνα με την εξίσωση (3.7)

```
right_vector = c * alphas + (1.0 - c) * epsilons
right_vector
```

```
array([0.15, 1.  , 0.15, 0.15, 0.15, 0.15])
```

Έτσι σύμφωνα με την εξίσωση (3.7) ο πίνκας Google είναι:

```
one_over_n_epsilons = epsilons/H.shape[0]
google_matr = c*H + np.dot(right_vector[:,None],one_over_n_epsilons[None,:])
google_matr
```

```
matrix([[0.025     , 0.45      , 0.45      , 0.025     , 0.025     ,
         0.025     ],
        [0.16666667, 0.16666667, 0.16666667, 0.16666667, 0.16666667,
         0.16666667],
        [0.30833333, 0.30833333, 0.025     , 0.025     , 0.30833333,
         0.025     ],
        [0.025     , 0.025     , 0.025     , 0.025     , 0.45      ,
         0.45      ],
        [0.025     , 0.025     , 0.025     , 0.45      , 0.025     ,
         0.45      ],
        [0.025     , 0.025     , 0.025     , 0.875     , 0.025     ,
         0.025     ]])
```

Πράγματι, σύμφωνα με την υλοποίηση για τον υπολογισμό του πίνακα του `networkx`:

```
nx.google_matrix(G)
```

```
matrix([[0.025     , 0.45      , 0.45      , 0.025     , 0.025     ,
         0.025     ],
        [0.16666667, 0.16666667, 0.16666667, 0.16666667, 0.16666667,
         0.16666667],
        [0.30833333, 0.30833333, 0.025     , 0.025     , 0.30833333,
         0.025     ],
        [0.025     , 0.025     , 0.025     , 0.025     , 0.45      ,
         0.45      ],
        [0.025     , 0.025     , 0.025     , 0.45      , 0.025     ,
         0.45      ],
        [0.025     , 0.025     , 0.025     , 0.875     , 0.025     ,
         0.025     ]])
```

Στη συνέχεια εφαρμόζουμε την απλή υλοποίηση που παρουσιάσαμε παραπάνω για τη μέθοδο των δυνάμεων για δέκα επαναλήψεις:

```
pi0 = google_matr.shape[0]*[1/google_matr.shape[0]] # elements 1/n
pi0 = np.array(pi0)
print('iteration: 0\n', pi0)
print('\n')
for i in range(10):
    pi_new = np.matmul(pi0, google_matr)
```

```
    print(f'iteration: {i+1}\n', pi_new)
    print('\n')
    pi0 = pi_new
```

```
iteration: 0
 [0.16666667 0.16666667 0.16666667 0.16666667 0.16666667 0.16666667]


iteration: 1
 [[0.09583333 0.16666667 0.11944444 0.26111111 0.16666667 0.19027778]]


iteration: 2
 [[0.0824537  0.12318287 0.08934028 0.28118056 0.19342593 0.23041667]]


iteration: 3
 [[0.06776399 0.10280681 0.07749373 0.32051109 0.18726572 0.24415866]]


iteration: 4
 [[0.06152086 0.09032055 0.06836399 0.32668709 0.19773807 0.25536944]]


iteration: 5
 [[0.05716521 0.08331157 0.06394177 0.33889812 0.19600722 0.2606761 ]]


iteration: 6
 [[0.05491931 0.07921452 0.06109769 0.34168023 0.19895101 0.26413724]]


iteration: 7
 [[0.05353307 0.07687377 0.05956276 0.34529289 0.19874717 0.26599033]]
```

```
iteration: 8
 [[0.05276657 0.07551812 0.05864201 0.34644978 0.19951605 0.26710748]]


iteration: 9
 [[0.05231364 0.07473943 0.05812419 0.34753408 0.19955479 0.26773388]]


iteration: 10
 [[0.05205661 0.0742899  0.05782138 0.34797267 0.19975859 0.26810085]]
```

Παρατηρούμε ότι ήδη με μόλις 10 επαναλήψεις οι τιμές PageRank που δίνει η απλή υλοποίησή μας είναι αρκετά κοντά στις τιμές που υπολογίζει η υλοποίηση του `networkx` όπως φαίνεται παρακάτω:

```
nx.pagerank(G)
```

```
{1: 0.05170556259095014,
 2: 0.07368068204240268,
 3: 0.05741336396912545,
 4: 0.34870204607252414,
 5: 0.19990341577794055,
 6: 0.26859492954705705}
```

3.9 Ένα μεγαλύτερο παράδειγμα

Εφαρμόζουμε τον αλγόριθμο PageRank στο σύνολο δεδομένων indochina-2004 (Rossi & Ahmed, 2015).Το συγκεκριμένο δίκτυο αποτελεί ένα κομμάτι στιγμιότυπου του Διαδικτύου της μορφής $G = (V, E)$. Μερικά χαρακτηριστικά του δικτύου παρουσιάζονται στον Πίνακα 3.2.

Πλήθος κόμβων $\|V\|$	Πλήθος ακμών $\|E\|$	Μέγιστος βαθμός $\max\{\deg(i)\}$	Ελάχιστος βαθμός $\min\{\deg(i)\}$
11400	47600	199	1

Πίνακας 3.2: Χαρακτηριστικά του δικτύου indochina-2004 (Rossi & Ahmed, 2015).

Εισάγουμε το γράφημα σε ένα `Jupyter notebook` χρησιμοποιώντας τις εντολές:

```
import networkx as nx
import matplotlib.pyplot as plt
import pandas as pd
```

```
df = pd.read_table('./data_sets/web-indochina-2004/web-indochina-2004 - Copy.mtx', sep = ' ')#,
    names = ['A','B'], skiprows= 2)
df
```

```
      A      B
0     551    1
1     11338  1
2     70     2
3     71     2
4     414    2
...   ...    ...
47601 11354  11352
47602 11355  11352
47603 11354  11353
47604 11355  11353
47605 11355  11354

47606 rows × 2 columns
```

```
G = nx.from_pandas_edgelist(df, source = 'A', target = 'B')
```

Το γράφημα περιέχει 11358 κόμβους:

```
len(G.nodes)
```

```
11358
```

οι οποίοι συνδέονται με 47606 ακμές:

```
len(G.edges)
```

```
47606
```

Υπολογίζουμε τον πίνακα γειτνίασης και τον πίνακα Google (Σχήμα 3.5)

```
H = nx.adjacency_matrix(G)
H
```

```
<11358x11358 sparse matrix of type '<class 'numpy.intc'>'
with 95212 stored elements in Compressed Sparse Row format>
```

```
H.todense()
```

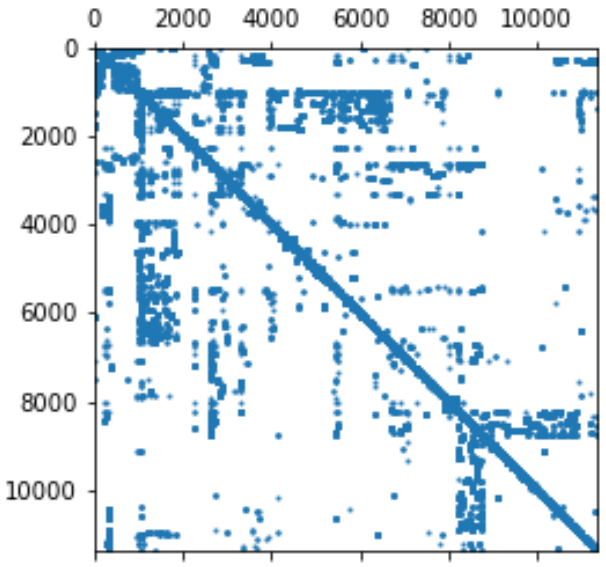
```
matrix([[0, 1, 1, ..., 0, 0, 0],
        [1, 0, 1, ..., 0, 0, 0],
        [1, 1, 0, ..., 0, 0, 0],
        ...,
        [0, 0, 0, ..., 0, 0, 0],
        [0, 0, 0, ..., 0, 0, 0],
        [0, 0, 0, ..., 0, 0, 0]], dtype=int32)
```

```
plt.spy(H, markersize = 1)
```

```
nx.google_matrix(G)
```

```
matrix([[1.32065504e-05, 8.34653988e-03, 8.34653988e-03, ...,
         1.32065504e-05, 1.32065504e-05, 1.32065504e-05],
        [4.25013207e-01, 1.32065504e-05, 4.25013207e-01, ...,
         1.32065504e-05, 1.32065504e-05, 1.32065504e-05],
        [7.95713178e-03, 7.95713178e-03, 1.32065504e-05, ...,
         1.32065504e-05, 1.32065504e-05, 1.32065504e-05],
        ...,
        [1.32065504e-05, 1.32065504e-05, 1.32065504e-05, ...,
         1.32065504e-05, 1.32065504e-05, 1.32065504e-05],
        [1.32065504e-05, 1.32065504e-05, 1.32065504e-05, ...,
```

```
 1.32065504e-05, 1.32065504e-05, 1.32065504e-05],
[1.32065504e-05, 1.32065504e-05, 1.32065504e-05, ...,
 1.32065504e-05, 1.32065504e-05, 1.32065504e-05]])
```



Σχήμα 3.5: Γράφημα του αραιού πίνακα γειτνίασης για τον γράφο web-indochine-2004.

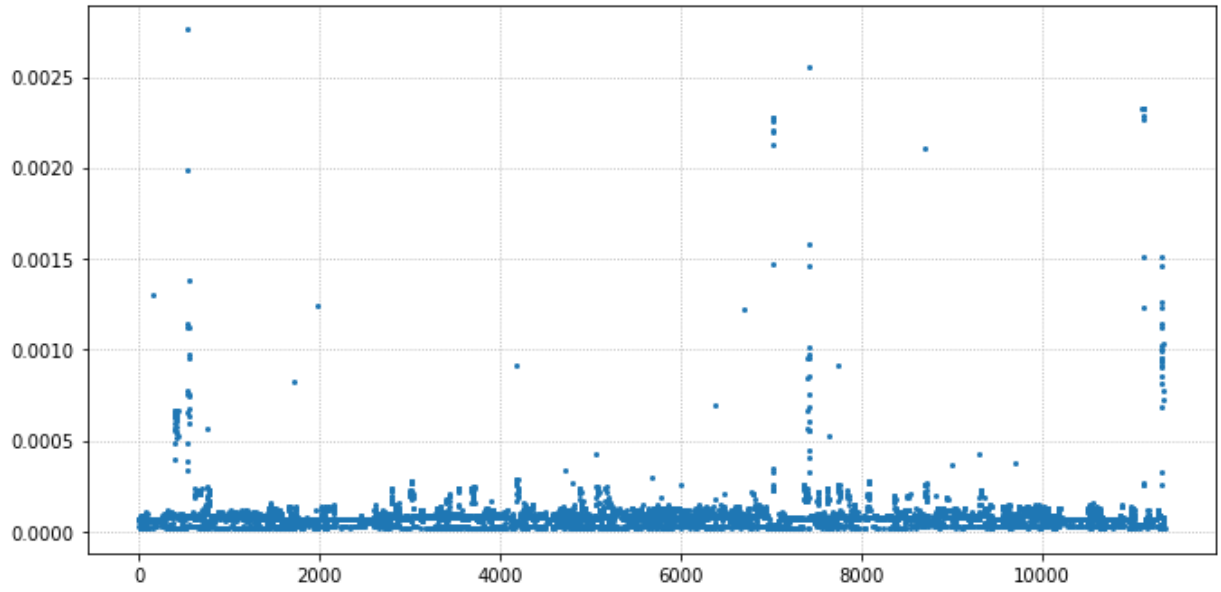
Εφαρμόζουμε τον αλγόριθμο PageRank στον γράφο:

```
pagerank_scores_dict =nx.pagerank(G)
pagerank_scores_dict
```

```
{551: 0.0009793893281335537,
 1: 3.008735025195669e-05,
 11338: 0.001034087693227333,
 70: 5.667430142232539e-05,
 2: 5.667430142232539e-05,
 71: 5.667430142232539e-05,
 414: 0.0006358905556296455,
 4: 6.700442668215593e-05,
 3: 6.700442668215593e-05,
 ...
```

και σχεδιάζουμε τις τιμές PageRank (Σχήμα 3.6)

```
fig, ax = plt.subplots(1, 1, figsize=(1.5*6.4, 4.8), tight_layout=True)
vals = [pagerank_scores_dict[key] for key in pagerank_scores_dict.keys()]

ax.plot(pagerank_scores_dict.keys(),vals, 'o', markersize = 2)
ax.grid(ls = ':')
```

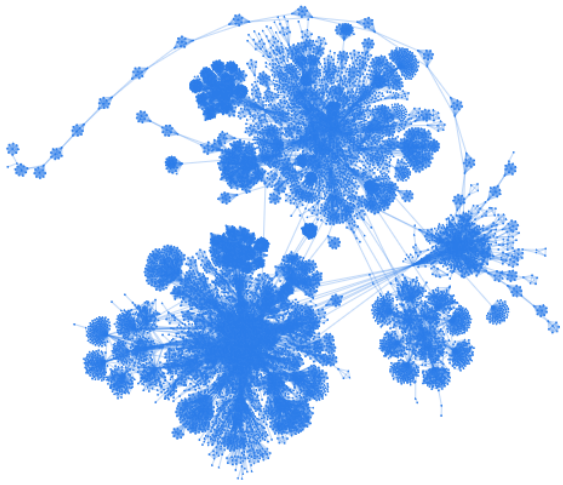


Σχήμα 3.6: Τιμές PageRank για κάθε κόμβο του γράφου web-indochine-2004.

Παρατηρούμε ότι κάποιοι κόμβοι $(545, 7429, 11129, \ldots)$ παρουσιάζουν ιδιαίτερα υψηλές τιμές PageRank καθιστώντας τους ιδιαίτερα επιδραστικούς στο δίκτυο.

Σχεδιάζουμε τον γράφο χρησιμοποιώντας το πακέτο pyvis (Σχήμα 3.7) εκτελώντας τις εξής εντολές:

```
from pyvis.network import Network
net = Network(notebook=True)
net.from_nx(G)
net.show("example.html")
```



Σχήμα 3.7: Το γράφημα του δικτύου web-indochina-2004 σχεδιασμένο με το πακέτο pyvis.

Κεφάλαιο 4

Ανάλυση κοινωνικών δικτύων - Μέτρα κεντρικότητας

Η ανάλυση κοινωνικών δικτύων είναι μια ερευνητική περιοχή κοινωνιολογικού, μαθηματικού και υπολογιστικού ενδιαφέροντος. Η κοινωνιολογική θεωρητική βάση της τίθεται από τους Georg Simmel και Emile Durkheim, ενώ οι πρώτες αναλυτικές μέθοδοι εισάγονται τη δεκαετία του 30 από τους Jacob Moreno και Helen Jengings. Ωστόσο, το ερευνητικό πεδίο αρχίζει να παίρνει τη μορφή που έχει σήμερα τη δεκαετία του 70 με την εισαγωγή χρήσης ηλεκτρονικών υπολογιστών (Freeman, 2004).

Η ύπαρξη και εδραίωση διαδικτυακών κοινωνικών δικτύων [1] όπως οι πλατφόρμες Facebook, Instagram, Twitter, TikTok αναζωπύρωσαν το ενδιαφέρον για τη μελέτη της ανάλυσης κοινωνικών δικτύων. Μάλιστα, η γνώση της επιδραστικότητας/επιρροής των χρηστών σε τέτοια κοινωνικά δίκτυα και η δυνατότητα πρόβλεψής της είναι ιδιαίτερα χρήσιμη σε ορισμένες εφαρμογές όπως για παράδειγμα στο viral marketing (Riquelme & Gonzalez-Cantergiani, 2016)αλλά και παροχή προτάσεων σε χρήστες για νέες επαφές και συστήματα προτάσεων βασισμένες στα ενδιαφέροντα των χρηστών (Aggarwal, 2011).

Η επιδραστικότητα είναι μία ποσότητα εγγενώς συνδεδεμένη με την δομή του δικτύου. Με τον όρο επιδραστικότητα εννοούμε το πόσο σημαντικός ή κεντρικός είναι ένας κόμβος για το δίκτυο. Γι αυτόν ακριβώς το λόγο συχνά απαντάται και με τον όρο κεντρικότητα (centrality). Όπως είναι φανερό, η επιδραστικότητα είναι ιδιότητα των κόμβων του δικτύου. Ωστόσο, ο ορισμός της είναι αρκετά χαλαρός. Επομένως, μπορούν να οριστούν πολλαπλά μέτρα επιδραστικότητας (influence or centrality measures).

[1] Από εδώ και στο εξής όταν θα αναφερόμαστε σε κάποιο κοινωνικό δίκτυο, θα εννοούμε ότι είναι διαδικτυακό.

4.1 Κεντρικότητα βαθμού

Το απλούστερο μέτρο κεντρικότητας είναι ο βαθμός κάθε κόμβου ($\deg(i)$), το οποίο αποκαλείται *κεντρικότητα βαθμού*. Προφανώς σε ένα δίκτυο (κατευθυνόμενο γράφημα) ορίζονται δύο κεντρικότητες βαθμού η inlink και η outlink ($\deg^{in}(i)$ και $\deg^{out}(i)$ αντίστοιχα). Μια απλή ερμηνεία της κεντρικότητας βαθμού είναι ότι δίνουμε στον κόμβο μία μονάδα για κάθε γειτονικό κόμβο.

4.2 Κεντρικότητα ιδιοδιανύσματος

Η κεντρικότητα ιδιοδιανύσματος βασίζεται στην παραδοχή ότι όλοι οι γειτονικοί κόμβοι του δικτύου δεν είναι ισοδύναμοι. Για παράδειγμα, η σημαντικότητα ενός κόμβου αυξάνεται εάν γειτονεύει με άλλες σημαντικούς κόμβους. Έτσι, αντί να δίνουμε μία μονάδα σε κάθε κόμβο για κάθε γείτονα, δίνουμε μία τιμή ανάλογη του αθροίσματος των τιμών των γειτόνων του. (Newman, 2010)

Ξεκινάμε δίνοντας σε κάθε κόμβο την τιμή 1:

$$x_i^{(0)} = 1, \qquad \forall i$$

και υπολογίζουμε μία βελτιωμένη τιμή για την κεντρικότητα:

$$x_i^{(k+1)} = \sum_j A_{ij} x_j^{(k)}$$

όπου A_{ij} τα στοιχεία του πίνακα γειτνίασης.

Σε μορφή πίνακα:

$$\mathbf{x}^{(k)} = \mathbf{A}\mathbf{x}^{k-1}$$
$$\Rightarrow \mathbf{x}^{(k)} = \mathbf{A}^{(k)}\mathbf{x}^{(\mathbf{0})}$$

Αναλύοντας την προσέγγιση στο πρώτο βήμα ως προς τα ιδιοδιανύσματα $\mathbf{v}_i$ του πίνακα γειτνίασης έχουμε:

$$\mathbf{x}^{(0)} = \sum_i c_i \mathbf{v}_i$$

όπου c_i κατάλληλες σταθερές.

Έτσι:

$$\begin{aligned}\mathbf{x}^{(k+1)} &= \mathbf{A}^{(k)} \sum_i c_i \mathbf{v}_i \\ &= \sum_i c_i \mathbf{A}^{k)} \mathbf{v}_i \\ &= \sum_i c_i \lambda_i^k \mathbf{v}_i \\ &= \lambda_1^k \sum_i c_i \left(\frac{\lambda_i}{\lambda_1}\right)^k \mathbf{v}_i\end{aligned}$$

όπου: λ_i οι ιδιοτιμές του πίνακα $\mathbf{A}$ με μέγιστη ιδιοτιμή λ_1.

Καθώς $\lambda_i/\lambda_1 < 1 \, \forall i \neq 1$, όταν $k \to \infty$:

$$\lim_{k\to\infty} x^{(k)} = c_1 \lambda_1 \mathbf{v}_1$$

επομένως σε μεγάλο αριθμό επαναλήψεων το διάνυσμα κεντρικότητας $\mathbf{x}$ είναι ανάλογο του κύριου ιδιοδιανύσματος του πίνακα γειτνίασης.

Έτσι η κεντρικότητα ικανοποιεί την εξής σχέση (Bonacich, 1987):

$$\mathbf{A}\mathbf{x} = \lambda_1 \mathbf{x} \tag{4.1}$$

και επομένως:

$$x_i = \frac{1}{\lambda_1} \sum_j A_{ij} x_j$$

και άρα η κεντρικότητα μπορεί να έχει υψηλή τιμή είτε επειδή ο κόμβος έχει πολλούς γείτονες είτε επειδή έχει σημαντικούς γείτονες ή και συνδυασμός των δύο. Έτσι, ένας σημαντικός χρήστης σε ένα κοινωνικό δίκτυο δεν είναι μόνο αυτός που γνωρίζει πολλούς άλλους χρήστες αλλά και αυτός που γνωρίζει λίγους αλλά σημαντικούς ανθρώπους.

Οι κεντρικότητες ιδιοδιανύσματος κάθε κόμβου είναι μη μηδενικές καθώς εάν το $\mathbf{x}(0)$ έχει μη μηδενικά στοιχεία τότε και το $\mathbf{A}\mathbf{x}(0)$ έχει επίσης μη μηδενικά στοιχεία.

Αξίζει να σημειωθεί ότι οι κεντρικότητες που προκύπτουν από την (4.1) δεν είναι κανονικοποιημένες. Κάτι τέτοιο δεν είναι ιδιαίτερα σημαντικό πρόβλημα καθώς μας ενδιαφέρει ποιος κόμβος έχει υψηλότερη και ποιος χαμηλότερη τιμή κεντρικότητας. Ωστόσο, θα μπορούσαμε να κανονικοποιήσουμε τις κεντρικότητες απαιτώντας:

$$\sum_i \lambda_i = n$$

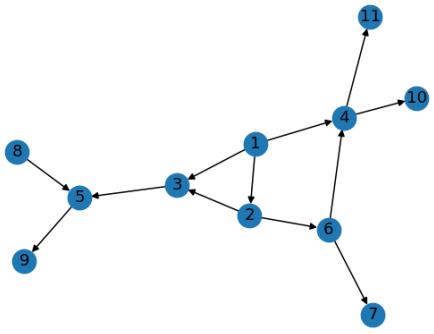
Η κεντρικότητα ιδιοδιανύσματος μπορεί να υπολογιστεί τόσο για δίκτυα όσα και για μη κατευθυνόμενους

γράφους, είναι ωστόσο, καλύτερη μετρική για μη κατευθυνόμενους γράφους.

Οι κατευθυνόμενοι γράφοι οδηγούν σε ορισμένα προβλήματα:

- Ο πίνακας γειτνίασης $\mathbf{A}$ είναι μη συμμετρικός. Έτσι, έχουμε δύο κύρια ιδιοδιανύσματα, το δεξιό και το αριστερό ιδιοδιάνυσμα. Το συγκεκριμένο προβλήμα μπορεί να διορθωθεί επιλέγοντας το δεξιό ιδιοδιάνυσμα καθώς η σημαντικότητα κάθε κόμβου εντοπίζεται περισσότερο στο να τον δείχνουν άλλοι κόμβοι παρά στο να δείχνει προς άλλους.

- Ένα επίσης δυνητικό πρόβλημα είναι ένας κόμβος να έχει μόνο ακμές που απομακρύνονται από αυτόν (outgoing edges) όπως για παράδειγμα ο κόμβος 1 του Σχήματος 4.1. Σε αυτή την περίπτωση ο κόμβος 1 παρουσιάζει μηδενική τιμή κεντρικότητας. Κάτι τέτοιο δεν είναι απαραίτητα πρόβλημα καθώς ο κόμβος 1 μπορεί πράγματι να μην είναι σημαντικός. Ωστόσο, ο κόμβος 2 παρουσιάζει επίσης μηδενική κεντρικότητα καθώς έχει μόνο μία ακμή που δείχνει τον κόμβο η οποία ξεκινά από τον 1. Έτσι, η συγκεκριμένη συμπεριφορά της κεντρικότητα ιδιοδιανύσματος παρασύρει και κόμβους που δείχνονται από κόμβους με μηδενική κεντρικότητα (Σχήμα 4.2).

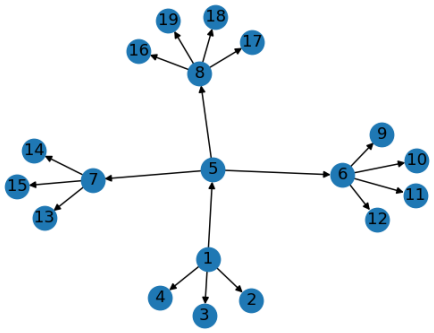


Σχήμα 4.1: Ένας κατευθυνόμενος γράφος. Παρατηρούμε ότι ο κόμβος 1 δεν έχει καμία ακμή προς την κατεύθυνσή το κάτι που του προσδίσει μηδενική τιμή κεντρικότητας. Ωστόσο, το γεγονός αυτό συμπαρασύρει μαζί του τον κόμβου 2 ο οποίος δείχνεται μοναδικά από τον 1.

Έτσι, συμπεραίνουμε ότι μόνο κόμβοι σε ισχυρά συνδεδεμένα component με πλήθος κόμβων τουλάχιστον 2 έχουν μη μηδενική κεντρικότητα. Όμως, εν γένει, θέλουμε υψηλή τιμή $\deg^{in}$ να συνεπάγεται και υψηλή τιμή κεντρικότητας. Κάτι τέτοιο καθιστά την συγκερκιμένη μετρική άχρηστη για ακυκλικά δίκτυα (Newman, 2010).

4.3 Κεντρικότητα Katz

Η κεντρικότητα Katz αποτελεί παραλλαγή της κεντρικότητας ιδιοδιανύσματος και δίνει λύση στα προβλήματα της τελευταίας δίνοντας εξ αρχής μία τιμή σε κάθε κόμβο ανεξαρτήτως της θέσης του στο δίκτυο αλλά και της



Σχήμα 4.2: Ένας κατευθυνόμενος γράφος. Παρατηρούμε ότι το γεγονός ότι ο κόμβος 1 παρουσιάζει μηδενική κεντρικότητα συμπρασύρει όλους τους κόμβους του γραφήματος.

θέσης των γειτόνων του. Έτσι:

$$x_i = a \sum_j A_{ij} x_j + b \tag{4.2}$$

όπου $a, b > 0$ σταθερές.

Σε μορφή πίνακων:

$$\begin{aligned} (4.2) \Rightarrow \qquad \mathbf{x} &= a\mathbf{A}\mathbf{x} + b\mathbf{e} \\ \mathbf{x} - a\mathbf{A}\mathbf{x} &= b\mathbf{e} \\ (\mathbf{I} - a\mathbf{A})\mathbf{x} &= b\mathbf{e} \\ \mathbf{x} &= (\mathbf{I} - a\mathbf{A})^{-1} b\mathbf{e} \end{aligned} \tag{4.3}$$

όπου $\mathbf{e}$ ένα διάνυσμα $n \times 1$ με στοιχεία μονάδες.

Συχνά επιλέγουμε $b = 1$ (Katz, 1953), έτσι ώστε:

$$\mathbf{x} = (\mathbf{I} - a\mathbf{A})^{-1} \mathbf{e} \tag{4.4}$$

Τίθεται επομένως το ερώτημα του πως επιλέγεται η σταθερά a. Προφανώς για $a \to 0$ παίρνουμε κεντρικότητα ίση με $b = 1$ για κάθε κόμβο. Όσο αυξάνουμε την τιμή του a παρουσιάζεται απόκλιση της κεντρικότητας κάθε κόμβου με κρίσιμη τιμή τη μέγιστη ιδιοτιμή του πίνακα $\mathbf{A}$ όπως προκύπτει από την χαρακτηριστική εξίσωση:

$$\det\left(\mathbf{A} - a^{-1}\mathbf{I}\right) = 0$$

δηλαδή:

$$\lambda_1 = \frac{1}{a}$$
$$a = \frac{1}{\lambda_1}$$

Καθώς επιθυμούμε σύγκλιση της μεθόδου επιλέγουμε τιμές για τη σταθερά a που ικανοποιούν τη συνθήκη:

$$a < \frac{1}{\lambda_1}$$

Παραμένει ωστόσο σχετική ελευθερία στην επιλογή του a. Κάποιοι ερευνητές επιλέγουν τιμές κοντά στο $1/\lambda_1$ δίνοντας μέγιστο βάρος στον όρο της κεντρικότητας ιδιοδιανύσματος και ελάχιστο στο σταθερό όρο (Newman, 2010).

Ο άμεσος υπολογισμός της κεντρικότητας από την εξίσωση (4.4) παρουσιάζει μεγάλο υπολογιστικό κόστος καθώς απαιτεί αντιστροφή πίνακα. Αντίθετα για τον υπολογισμό εφαρμόζουμε μία επαναληπτική διαδικασία της μορφής:

$$\mathbf{x}' = a\mathbf{A}\mathbf{x} + b\mathbf{e}$$

βασιζόμενοι στην αρχική διανυσματική εξίσωση (4.3).

Εφαρμόζοντας την επαναληπτική μέθοδο, το διάνυσμα $\mathbf{x}$ συγκλίνει στο ιδιοδιάνυσμα. Καθώς ο πίνακας $\mathbf{A}$ έχει m μη μηδενικά στοιχεία απαιτούνται m πολλαπλασιασμοί καθιστώντας τον υπολογισμό της τάξης $\mathcal{O}(rm)$ όπου r ο αριθμός των επαναλήψεων που απαιτούνται για σύγκλιση της μεθόδου. Ο ακριβής αριθμός r δεν μπορεί να υπολογιστεί καθώς εξαρτάται τόσο από το ίδιο το δίκτυο όσο και από την επιλογή του a.

Αξίζει να σημειωθεί ότι η κεντρικότητα Katz μπορεί να εφαρμοστεί και σε μη κατευθυνόμενους γράφους.

Η κεντρικότητα Katz μπορεί να γενικευθεί δίνοντας διαφορετικά σταθερά βάρη σε κάθε κόμβο τροποποιώντας κατάλληλα την (4.2):

$$x_i = a\sum_j A_{ij}x_j + b_i \tag{4.5}$$

Για παράδειγμα, σε ένα κοινωνικό δίκτυο η σημαντικότητα κάθε κόμβου (ατόμου) μπορεί να εξαρτάται από παράγοντες που δε σχετίζονται με το δίκτυο όπως η ηλικία ή το εισόδημα (Newman, 2010). Τέτοιου τύπου αλληλεπιδράσεις μπορούν να ληφθούν υπ'όψιν από τους όρους b_i.

Σε διανυσματική μορφή:

$$\begin{aligned}
(4.5) \Rightarrow \mathbf{x} &= a\mathbf{A}\mathbf{x} + \mathbf{b} \\
\mathbf{x} - a\mathbf{A}\mathbf{x} &= \mathbf{b} \\
(I - a\mathbf{A})\,\mathbf{x} &= \mathbf{b} \\
\mathbf{x} &= (I - a\mathbf{A})^{-1}\,\mathbf{b}
\end{aligned}$$

Η κεντρικότητα Katz παρουσιάζει ένα ανεπιθύμητο χαρακτηριστικό. Εάν ένας κόμβος με υψηλή κεντρικότητα Katz δείχνει πολλούς άλλους κόμβους τότε και αυτοί με τη σειρά τους λαμβάνουν υψηλή τιμή κεντρικότητας. Κάτι τέτοιο δεν είναι πάντα επιθυμητό. Για παράδειγμα το γεγονός ότι μία ιστοσελίδα δείχνεται από το Google το οποίο είναι αδιαμφισβήτητα σημαντική ιστοσελίδα, δεν την κάνει απαραίτητα σημαντική.

4.4 Κεντρικότητα PageRank

Ο αλγόριθμος PageRank αναλύθηκε εκτενώς στο αντίστοιχο κεφάλαιο, ωστόσο επαναλαμβάνουμε εδώ ορισμένα χαρακτηριστικά συνδέοντάς τον με τα υπόλοιπα μέτρα κεντρικότητας.

Η κεντρικότητα PageRank διορθώνει το πρόβλημα της κεντρικότητας Katz διαιρώντας την κεντρικότητα κάθε κόμβου με το $\deg^{out}$ του. Έτσι, κάθε κόμβος δίνει ένα μικρό ποσό κεντρικότητας σε γειτονικούς του κόμβους ακόμα και αν η δική του κεντρικότητα είναι υψηλή (Newman, 2010). Έτσι, η κεντρικότητα PageRank γράφεται:

$$x_i = a \sum_j A_{ij} \frac{x_j}{\deg^{out}(j)} + b \tag{4.6}$$

Η συγκεκριμένη προσέγγιση ως προς τη σημαντικότητα των κόμβων παρουσιάζει προβλήματα σε κόμβους με $\deg^{out}(i) = 0$. Επιλέγοντας τεχνητά $\deg^{out}(i) = 1$ για όλους τους κόμβους με μηδενικό $\deg^{out}$ το πρόβλημα αίρεται.

Σε διανυσματική μορφή έχουμε:

$$(4.6) \Rightarrow \mathbf{x} = a\mathbf{A}\mathbf{D}^{-1}\mathbf{x} + b\mathbf{e} \tag{4.7}$$

όπου $\mathbf{D}$ ο διαγώνιος πίνακας με διαγώνια στοιχεία:

$$D_{ii} = \max\left\{\deg^{out}(i), 1\right\}$$

επιλύοντας ως προς $\mathbf{x}$:

$$\begin{aligned}(4.7) \Rightarrow \mathbf{x} &= \left(\mathbf{I} - a\mathbf{A}\mathbf{D}^{-1}\right)^{-1}\mathbf{e} \\ &= \left(\mathbf{D}\mathbf{D}^{-1} - a\mathbf{A}\mathbf{D}^{-1}\right)^{-1}\mathbf{e} \\ &= \mathbf{D}\left(\mathbf{D} - a\mathbf{A}\right)^{-1}\mathbf{e} \qquad (4.8)\end{aligned}$$

Όπως και στην περίπτωση της κεντρικότητας Katz, η εξίσωση (4.8) περιέχει την ελεύθερη παράμετρο a. Κατάναλογία η τιμή της θα πρέπει να είναι μικρότερη της αντίστροφης ιδιοτιμής του πίνακα $\mathbf{A}\mathbf{D}^{-1}$ η οποία είναι ίση με μονάδα (από Θεώρημα Perron - Frobenius). Έτσι στην περίπτωση ενός μη κατευθυνόμενου γράφου:

$$a < 1$$

Σε περίπτωση που ο γράφος είναι κατευθυνόμενος τιμές κοντά στη μονάδα είναι αρκετά καλές προσεγγίσεις (Newman, 2010).

Όπως έχουμε αναφέρει η τιμή που επιλέγεται από το Google είναι $a = 0.85$.

Γενικεύοντας την (4.6):

$$x_i = a\sum_j A_{ij}\frac{x_j}{\deg^{out}(j)} + b_i$$
$$\mathbf{x} = \mathbf{D}(\mathbf{D} - a\mathbf{A})^{-1}\mathbf{b}$$

καταλήγουμε έτσι σε μία εξατομικευμένη γενίκευση της κεντρικότητας PageRank.

4.5 Κεντρικότητα Closeness

Η κεντρικότητα closeness αποτελεί ένα εντελώς διαφορετικό μέτρο σημαντικότητας η οποία μετράει τη μέση απόσταση μεταξύ κόμβων. Βασικό κομμάτι της συγκεκριμένης κεντρικότητας είναι το γεωδαισιακό μονοπάτι, δηλαδή το μικρότερο μονοπάτι μεταξύ δύο κόμβων. Έστω d_{ij} το γεωδαισιακό μονοπάτι μεταξύ των κόμβων i και j, δηλαδή το πλήθος των ακμών κατά μήκος του μονοπατιού. Η μέση γεωδαισιακή απόσταση από το i στο j σε όλους τους κόμβους του δικτύου είναι:

$$l_i = \frac{1}{n}\sum_j d_{ij}$$

Η συγκεκριμένη ποσότητα λαμβάνει χαμηλές τιμές για κόμβους που διαχωρίζονται από άλλους με μικρές γεωδαισιακές αποστάσεις. Τέτοιοι κόμβοι έχουν καλύτερη πρόσβαση σε πληροφορίες που διαχέονται στο δίκτυο και άρα μεγαλύτερη επιρροή σε άλλους κόμβους.

Συχνά στην άθροιση αγνοούνται οι όροι $i = j$:

$$l_i = \frac{1}{n-1} \sum_{j \neq i} d_{ij}$$

καθώς η επιδραστικότητα ενός κόμβου στον εαυτό του δεν έχει ιδιαίτερη σημασία. Άλλωστε $d_{ii} = 0$ και ο όρος δε συνεισφέρει στο άθροισμα.

Η μέση γεωδαισιακή απόσταση διαφέρει από τα μέτρα κεντρικότητας που είδαμε στις προηγούμενες ενότητες. Δίνει χαμηλές τιμές για κεντρικούς κόμβους και υψηλές τιμές για λιγότερο κεντρικούς κόμβους. Έτσι, εναλλακτικά ορίζουμε την κεντρικότητα closeness ως:

$$C_i = \frac{1}{l_i} = \frac{n}{\sum\limits_j d_{ij}} \tag{4.9}$$

Η κεντρικότητα closeness παρουσιάζει ορισμένα προβλήματα (Newman, 2010):

- Οι τιμές της έχουν σχετικά μικρό εύρος καθώς τα d_{ij} τείνουν να είμαι μικρά και αυξάνονται λογαριθμικά με το μέγεθος του δικτύου. Έτσι, δύσκολα μπορεί κανείς να διακρίνει μεταξύ κεντρικών και λιγότερων κεντρικών κόμβων καταφεύγοντας συχνά στην εξέταση δεκαδικών ψηφίων για την ταξινόμηση των κόμβων.

- Η γεωδαισιακή απόσταση μεταξύ δύο κόμβων που βρίσκονται σε διαφορετικά component είναι άπειρη και επομένως $C_i = 0$.

Ένας τρόπος αντιμετώπισης του δεύτερου προβλήματος είναι η άθροιση κόμβων που ανήκουν στο ίδιο component . Κάτι τέτοιο οδηγεί βέβαια με τη σειρά του σε νέα προβλήματα. Οι αποστάσεις εντός του ίδιου component είναι μικρές, ειδικά αν το component είναι μικρό λαμβάνοντας μικρές τιμές l_i και υψηλές C_i από κόμβους σε μεγαλύτερα components. Η συγκεκριμένη συμπεριφορά δεν είναι επιθυμητή καθώς οι κόμβοι σε μικρά components θεωρούνται λιγότερο κεντρικοί.

Έτσι, ορίζουμε ξανά την κεντρικότητα closeness ως εξής:

$$C'_i = \frac{1}{n-1} \sum_{j \neq i} \frac{1}{d_{ij}} \tag{4.10}$$

Η εξίσωση (4.10) παρουσιάζει ορισμένες σημαντικές ιδιότητες:

- οι κόμβοι i και j που βρίσκονται σε διαφοερικά component για τους οποίους ισχύει $d_{ij} = \infty$ δε συνεισφέρουν στο άθροισμα.

- Η συγκεκριμένη μετρική δίνει βάρος σε κόμβους κοντά στο i

Ωστόσο, η εξίσωση (4.10) σπάνια χρησιμοποιείται (Newman, 2010).

Μία ποσότητα που σχετίζεται άμεσα με την κεντρικότητα clonseness είναι η μέση γεωδαισιακή απόσταση μεταξύ κόμβων, η οποία παίζει ιδιαίτερα σημαντικό ρόλο σε φαινόμενα small-world.

Έστω δίκτυο με ένα component, τότε η μέση απόσταση μεταξύ ζευγών κόμβων:

$$l = \frac{1}{n^2} \sum_{ij} d_{ij} = \frac{1}{n} \sum_{i} l_i$$

Ωστόσο, εμφανίζεται ξανά το πρόβλημα που αναφέρθηκε παραπάνω σχετικά με τον απειρισμό των d_{ij}. Έτσι, αθροίζουμε μόνο μονοπάτια μεταξύ κόμβων του ίδιου component. Έστω $\{\mathcal{C}_m\}$ το σύνολο όλων των component του δικτύου, τότε:

$$k = \frac{\sum\limits_{m} \sum\limits_{ij \in \mathcal{C}_m} d_{ij}}{\sum\limits_{m} n_m^2}$$

όπου n_m το πλήθος των κόμβων στο $\mathcal{C}_m$.

Εναλλακτικά:

$$\frac{1}{l'} = \frac{1}{n(n-1)} \sum_{i \neq j} \frac{1}{d_{ij}} = \frac{1}{n} \sum_{i} C'_i$$

$$\Rightarrow \quad l' = \frac{n}{\sum\limits_{i} C'_i}$$

4.6 Κεντρικότητα Betweenness

Η κεντρικότητα betweeness αποδίδεται στον Freeman (Freeman, 1977) ωστόσο εμφανίζεται νωρίτερα σε μία τεχνική έκθεση από τον Anthonisse (Newman, 2010). Είναι ένα μέτρο του κατά πόσο ένας κόμβος βρίσκεται σε μονοπάτια μεταξύ κόμβων.

Έστω δίκτυο $G = (V, E)$. Θεωρούμε ότι κάποια ποσότητα μεταδίδεται από κόμβο σε κόμβο κατά μήκος των ακμών του δικτύου. Για παράδειγμα σε ένα κοινωνικό δίκτυο η ποσότητα προς μελέτη θα μπορούσε να είναι μηνύματα, νέα, φήμες και εν γένει πληροφορίες. Υποθέτουμε ότι δύο κόμβοι ανταλλάσσουν πληροφορίες ισοπίθανα ενώ οι πληροφορίες ταξιδεύουν εντός της δομής του δικτύου κατά μήκος των συντομότερων διαδρομών (ή διαλέγουν μία διαδρομή τυχαία εάν υπάρχουν περισσότερες τέτοιες διαδρομές).

Η κεντρικότητα betweeness αναδεικνύεται μέσω της απάντησης στην εξής ερώτηση: Εάν περιμένουμε αρκετό χρόνο ώστε πολλές πληροφορίες να περνούν από όλα τα ζεύγη κόμβων, πόσες πληροφορίες κατά μέσο όρο έχουν περάσει από κάθε κόμβο του δικτύου;

Η απάντηση είναι απλή, τα μηνύματα περνούν μέσα από κάθε μονοπάτι με τον ίδιο ρυθμό και επομένως, ο αριθμός μηνυμάτων που περνούν από κάθε κόμβο είναι ανάλογος του αριθμού των γεωδαισιακών μονοπατιών που περιέχουν τον κόμβο. Αυτός ακριβώς είναι και ο ορισμός της κεντρικότητας betweeness.

Κόμβοι με υψηλή τιμή betweeness ασκούν σημαντική επιρροή στο δίκτυο καθώς ελέγχουν τις πληροφορίες

που μεταδίδονται εντός του. Εάν ένας τέτοιος κόμβος αφαιρεθεί από το δίκτυο τότε η επικοινωνία μεταξύ των εναπομεινάντων κόμβων διαταράσσεται σημαντικά.

Για λόγους απλότητας θεωρούμε αρχικά έναν μη κατευθυνόμενο γράφο με το πολύ ένα γεωδαισιακό μονοπάτι μεταξύ κάθε ζεύγους κόμβων (η περίπτωση 0 αφορά σε κόμβους διαφορετικών component).

Έστω το σύνολο όλων των γεωδαισιακών μονοπατιών του γραφήματος μεταξύ δύο κόμβων, τότε η κεντρικότητα betweeness για έναν κόμβο i, x_i, είναι ο αριθμός αυτών των μονοπατιών που περνούν από το i:

$$x_i = \sum_{st} n_{st}^i \tag{4.11}$$

όπου:

$$n_{st}^i = \begin{cases} 1 & \text{, αν } i \in st \\ 0 & \text{, αν } i \notin st \text{ ή δεν υπάρχει τέτοιο μονοπάτι} \end{cases}$$

Σημειώνουμε ότι κάθε μονοπάτι αθροίζεται δύο φορές καθώς ο γράφος είναι μη κατευθυνόμενος, κάτι που δεν επηρεάζει τη διάταξη για τους κόμβους καθώς ενδιαφερόμαστε για σχετικές και όχι απόλυτες τιμές κεντρικότητας. Επίσης, η άθροιση περιλαμβάνει και μονοπάτια από κάθε κόμβο στον εαυτό του, αφαίρεση των οποίων δεν επηρεάζει πάλι την διάταξη.

Μία γενίκευση της (4.11) αφορά την περίπτωση που το γράφημα δεν περιέχει μόνο το πολύ ένα γεωδαισιακό μονοπάτι μεταξύ κάθε ζεύγους κόμβων. Ο τρόπους που γενικεύουμε είναι δίνοντας βάρη σε κάθε κόμβο ανάλογα του αντιστρόφου του πλήθους των μονοπατιών δηλαδή:

$$x_i = \sum_{st} \frac{n_{st}^i}{g_{st}} \tag{4.12}$$

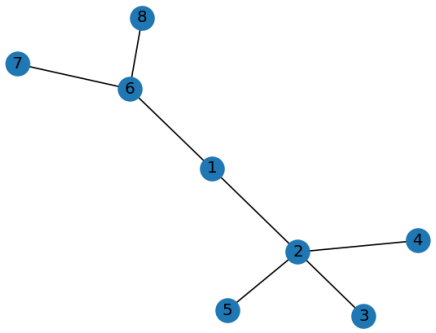
όπου:

- n_{st}^i: το πλήθος των γεωδαισιακών μονοπατιών από τον s στον t που περνούν από τον i.
- g_{st}: ο αριθμός των γεωδαισιακών μονοπατιών από το s στο t.

Σημειώνουμε ότι στη σχέση (4.12) υιοθετούμε τη σύμβαση $n_{st}^i/g_{st} = 0$ εάν $n_{st}^i = 0$ ή g_{st}.

Η κεντρικότητα betweeness διαφέρει σημαντικά από τα υπόλοιπα μέτρα κεντρικότητας που μελετήσαμε στις προηγούμενες ενότητες. Αντί να μετρά το πόσο καλά συνδεδεμένος είναι ένας κόμβος, μετρά το πόσο μεσολαβεί μεταξύ άλλων κόμβων. Χαρακτηριστικό παράδειγμα είναι ο κόμβος 1 στο γράφημα του Σχήματος 4.3. Ο συγκεκριμένος κόμβος παρουσιάζει υψηλή τιμή betweeness ενώ ο βαθμός του είναι μόλις 2.

Η κεντρικότητα betweeness παρουσιάζει σχετικά μεγάλο εύρος. Η μέγιστη τιμή της παρουσιάζεται σε άστρα. Για παράδειγμα σε ένα άστρο με n κόμβους, ο κεντρικός κόμβος βρίσκεται σε όλα τα n^2 το πλήθος συντομότερα μονοπάτια μεταξύ όλων των περιφερειακών $n-1$ κόμβων, ενώ δε μεσολαβεί στα $n-1$ μονοπάτια από κάθε



Σχήμα 4.3: Ο κόμβος 1 παρουσιάζει υψηλή τιμή betweeness ενώ ο βαθμός του είναι μικρός.

περιφερειακό κόμβο στον εαυτό του, δίνοντας κεντρικότητα:

$$x_1 = n^2 - (n-1) = n^2 - n + 1$$

Η ελάχιστη τιμή της είναι $2n-1$ καθώς κατ' ελάχιστο κάθε κόμβος βρίσκεται σε κάθε μονοπάτι που αρχίζει και τελειώνει με αυτόν, συγκεκριμένα έχουμε:

- $n-1$ μονοπάτια από τον κόμβο προς άλλους κόμβους
- $n-1$ μονοπάτια από άλλους κόμβους προς τον κόμβο
- 1 μονοπάτι από τον κόμβο στον εαυτό του

αθροίζοντας:

$$n-1+n-1+1 = 2n-1$$

Κάτι τέτοιο συμβαίνει όταν το γράφημα περιέχει κάποιο φύλλο.

Έτσι, το πηλίκο της μέγιστης και ελάχιστης τιμής betweeness είναι:

$$\frac{\max x}{\min x} = \frac{n^2 - n + 1}{2n-1} \simeq \frac{n}{2}$$

Μπορούμε να κανονικοποιήσουμε την (4.12) διαιρώντας με n^2, δηλαδή τον συνολικό αριθμό ζευγών κόμβων με διάταξη (Newman, 2010):

$$x_i = \frac{1}{n^2} \sum_{st} \frac{n_{st}^i}{g_{st}} \tag{4.13}$$

Στην παραπάνω ανάλυση ορίσαμε το betweeness με τις πληροφορίες να ρέουν μέσω των συντομότερων μονοπατιών. Στην πραγματικότητα, ωστόσο, πληροφορία ρέει και μέσω άλλων μονοπατιών. Γι αυτό το λόγο,

εναλλακτικά μπορούμε να ορίσουμε την flow betweeness που βασίζεται στην ιδέα της μέγιστης ροής. Φαντα-ζόμαστε τις ακμές του δικτύου ως σωλήνες και τις πληροφορίες ως ρευστό που ρέει εντός τους. Θα θέλαμε να γνωρίζουμε τη μέγιστη ροή μεταξύ ενός κόμβου πηγή s και ενός κόμβου target t. Η flow betweeness ορίζεται επίσης σύμφωνα με την εξίσωση (4.11):

$$x_i = \sum_{st} n_{st}^i$$

όπου n_{st}^i είναι η ποσότητα ροής μέσω του κόμβου i όταν μεταδίδεται μέγιστη ροή μεταξύ των s και t ή ισοδύναμα ο αριθμός των ανεξάρτητων μονοπατιών (Newman, 2010).

Επίσης, ένα άλλο μέτρο betweeness είναι η random walk betweeness όπου θεωρούμε έναν τυχαίο περίπατο που ξεκινά από το s και συνεχίζει έως ότου καταλήξει στο t:

$$x_i = \sum_{st} n_{st}^i$$

όπου n_{st}^i ο αριθμός των φορών που ο τυχαίος περιπατητής περνά από το i κατά μέσο όρο σε μεγάλο αριθμό επαναλήψεων (Newman, 2010).

Κεφάλαιο 5

Twitter

Το Twitter είναι ένα κοινωνικό δίκτυο που προσφέρει υπηρεσίες ιστολογίου τύπου microblogging[1]. Οι χρήστες του στέλνουν, διαβάζουν και μοιράζονται μηνύματα 140 χαρακτήρων που ονομάζονται tweet (τουίτ). Ένα tweet δεν περιέχει μόνο κείμενο αλλά και εικόνες, συνδέσμους (URL), αναφορές σε άλλους χρήστες εντός της πλατφόρμας (με την επισήμανση @) και λέξεις κλειδιά γνωστά ως hashtags (με την επισήμανση #). Κεντρικό ρόλο στην πλατφόρμα παίζουν τα δημοφιλή θέματα (trending topics) δηλαδή ένας όρος (μία ή περισσότερες λέξεις) που εμφανίζεται σε μεγάλο αριθμό tweet για μία συγκεκριμένη χρονική περίοδο σε κάποιο συγκεκριμένο μέρος. Το μέρος και το χρονικό διάστημα καθορίζονται από τα μεταδεδομένα του tweet (Riquelme & Gonzalez-Cantergiani, 2016).

5.1 Σχέσεις μεταξύ χρηστών

Το Twitter μπορεί να μοντελοποιηθεί ως ένας ιδιαίτερα πολύπλοκος κατευθυνόμενος γράφος. Η πολυπλοκότητα οφείλεται στην ποικιλία αλληλεπιδράσεων εντός του. Μερικές χαρακτηριστικές αλληλεπιδράσεις είναι οι σχέσεις μεταξύ χρηστών. Ένας χρήστης Α μπορεί να ακολουθεί έναν χρήστη Β και τότε ο Α ονομάζεται ακόλουθος (follower) του Β. Αντίστοιχα ο Β είναι ακολουθούμενος (followee) του Α. Ένας άλλος τύπος αλληλεπίδρασης είναι αυτός χρήστη με tweet , ένας χρήστης μπορεί να διαμοιραστεί κάποιο tweet άλλου χρήση σε μία δράση που ονομάζεται retweet. Εν γένει, υπάρχουν 4 ειδών σχέσεις που παρουσιάζονται εκτενώς στον Πίνακα 5.1 (Riquelme & Gonzalez-Cantergiani, 2016):

- χρήστη-προς-χρήστη
- χρήστη-προς-tweet
- tweet-προς-tweet

[1]Η έννοια του microblogging αναφέρεται σε σύντομες τακτικές δημοσιεύσεις σε αντίθεση με ένα συμβατικό ιστολόγιο

- tweet-προς-χρήστη

	χρήστης	tweet
χρήστης	ακολουθεί ή ακολουθείτε από αναφέρει (@) απαντά σε (reply) κάνει retweet σε	κοινοποιεί (post) κάνει retweet κάνει like απαντά σε (reply)
tweet	κοινοποιείται από (post) γίνεται retweet από γίνεται like από απαντάται από	απαντά/απαντάται από (reply) κάνει/γίνεται retweet από

Πίνακας 5.1: Δυνατές αλληλεπιδράσεις μεταξύ χρηστών και δημοσιεύσεων (tweet) στο Twitter(Riquelme & Gonzalez-Cantergiani, 2016).

5.2 Μοντελοποίηση με θεωρία γραφημάτων

Μοντελοποιούμε το Twitter ως δίκτυο (κατευθυνόμενο γράφημα) $G = (V, E)$. Ωστόσο, τα σύνολα V, E δεν ορίζονται μονοσήμαντα ακριβώς εξαιτίας της πολυπλοκότητας των σχέσεων μεταξύ χρηστών και tweet (Πίνακας 5.1).

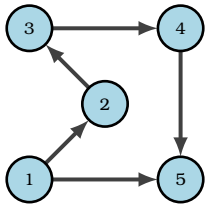
Απλή προσέγγιση Μία σχετικά απλή προσέγγιση είναι να θεωρήσουμε ως κόμβους αποκλειστικά τους χρήστες και ως ακμές τις σχέσεις ακολουθίας μεταξύ τους. Για παράδειγμα εάν $i, j \in V$ τότε η ακμή ij υπονοεί ότι ο χρήστης i ακολουθεί τον χρήστη j (αντίστοιχα ο j ακολουθείται από τον i). Στη συγκεκριμένη προσέγγιση οι αλληλεπιδράσεις μεταξύ των χρηστών όπως τα retweet, reply και like είναι μέρος της υποκείμενης δυναμικής του δικτύου (Riquelme & Gonzalez-Cantergiani, 2016). Ένας τέτοιος γράφος συμβολίζεται με G_1.

Πολύπλοκη προσέγγιση Μία πιο πολύπλοκη προσέγγιση είναι να θεωρήσουμε το ίδιο σύνολο κόμβων (δηλαδή οι κόμβοι είναι οι χρήστες του κοινωνικού δικτύου). Ωστόσο, κάθε ακμή μοντελοποιεί τις διαφορετικές αλληλεπιδράσεις μεταξύ των χρηστών (Πίνακας 5.1). Ο συγκεκριμένος γράφος θα συμβολίζεται με G_2. Η συγκεκριμένη προσέγγιση αποτελεί μία επέκταση του γράφου G_1 δημιουργώντας επιπλέον κατευθυνόμενους γράφους με το ίδιο σύνολο κόμβων και διαφορετικές ακμές, οι οποίες περιλαμβάνουν συχνά βάρη. Οι ακμές με βάρη συμβολίζουν σχέσεις μεταξύ χρηστών και tweet άλλων χρηστών. Τα βάρη με τη σειρά τους αναπαριστούν τον όγκο ή την συχνότητα των διάφορων αλληλεπιδράσεων. Για παράδειγμα, μία ακμή τύπου mention βάρους w μεταξύ δύο χρηστών Α και Β υποδεικνύει ότι ο χρήστης Α έχει αναφέρει (mention - @) τον χρήστη Β σε w το πλήθος tweet του.

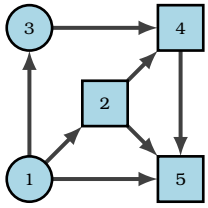
Συνδυασμός προσεγγίσεων Θα μπορούσαμε και να συνδυάσουμε τις δύο παραπάνω προσεγγίσεις ορίζοντας δύο σύνολα κόμβων V_1 και V_2 όπου V_1 το σύνολο των χρηστών και V_2 είναι το σύνολο όλων των tweet που

δημοσίευσαν οι χρήστες. Έτσι, οι ακμές μπορούν να αντιστοιχούν σε όλες τις πιθανές σχέσεις του Πίνακα 5.1. Συμβολίζουμε με G_3. Όπως και στα γραφήματα τύπου G_2 τα βάρη των ακμών αναπαριστούν τον όγκο ή την συχνότητα των διάφορων αλληλεπιδράσεων

Αξίζει να σημειωθεί ότι οι συγκεκριμένες προσεγγίσεις αποτελούν μόνο μία προσπάθεια μοντελοποίησης του Twitter. Για παράδειγμα, θα μπορούσε κάλλιστα να μοντελοποιηθεί ως multigraph (Jabeur, Tamine & Boughanem, 2012) αλλά και Μαρκοβιανή αλυσίδα (Simmie, Vigliotti & Hankin, 2014), edge-colored graphs, multilayer network (Omodei, De Domenico & Arenas, 2015).



Σχήμα 5.1: Γράφος μοντελοποίησης των σχέσεων μεταξύ χρηστών της μορφής G_1 ή G_2.



Σχήμα 5.2: Γράφος μοντελοποίησης των σχέσεων μεταξύ χρηστών της μορφής G_3. Οι κύκλοι αντιστοιχούν σε κόμβους του V_1 (χρήστες) ενώ τα τετράγωνα σε κόμβους του συνόλου V_2 (tweet).

5.3 Μετρήσιμες ποσότητες στο Twitter

Οι μετρήσιμες ποσότητες είναι μαθηματικές εκφράσεις που παίρνουν αριθμητικές τιμές και δίνουν σημαντικές πληροφορίες για το δίκτυο. Θυμίζουν αρκετά φυσικές ποσότητες. Οι μετρήσιμες ποσότητες χρησιμοποιούνται για τον ορισμό κάποιου μέτρου επιδραστικότητας/κεντρικότητας. Στον Πίνακα 5.2 παρουσιάζονται μερικές χρήσιμες μετρήσιμες ποσότητες για το Twitter.

Αξίζει να σημειωθεί ότι κάποιες από αυτές τις μετρήσιμες ποσότητες παρουσιάζουν συσχέτιση. Για παράδειγμα, εάν η μετρική $RT2$ έχει υψηλή τιμή τότε και η $RT3$ έχει σχετικά υψηλή τιμή (Ye Wu, 2010). Ωστόσο, η $M4$ παραμένει χαμηλή εάν η $M3$ είναι υψηλή (Riquelme & Gonzalez-Cantergiani, 2016).

Συμβολισμός	Μετρήσιμες ποσότητες
$OT1$	πλήθος πρωτότυπων tweet που δημοσιέυονται από το χρήστη
$OT2$	πλήθος URL που διαμοιράζονται από τα πρωτότυπα tweet
$OT3$	πλήθος hashtag στα πρωτότυπα tweet
$RP1$	πλήθος απαντήσεων (reply) δημοσιευμένα από το χρήστη
$RP2$	πλήθος πρωτότυπαων tweet δημοσιευμένα από το χρήστη και έχοντας απαντηθεί από άλλους χρήστες
$RP3$	πλήθος χρηστών που έχουν απαντήσει στα tweet του χρήστη
$RT1$	πλήθος retweet του χρήστη
$RT2$	πλήθος πρωτότυπων tweet δημοσιευμένα από το χρήστη και retweeted από άλλους
$RT3$	πλήθος χρηστών που έχουν κάνει retweet σε tweet του χρήστη
$FT1$	πλήθος tweet άλλων χρηστών που έχουν γίνει like από το χρήστη
$FT2$	πλήθος tweet του χρήστη που έχουν γίνει like από άλλους χρήστες
$FT3$	πλήθος χρηστών που έχουν κάνει like σε tweet του χρήστη
$M1$	πλήθος mention προς άλλους χρήστες από τον χρήστη
$M2$	πλήθος χρηστών που έχουν γίνει mentionαπό τον χρήστη
$M3$	πλήθος mention από άλλους χρήστες προς τον χρήστη
$M4$	πλήθος χρηστών που έχουν κάνει mention τον χρήστη
$F1$	πλήθος follower
$F2$	πλήθος θεματικά ενεργών ακόλουθων
$F3$	πλήθος ακολουθούμενων
$F4$	πλήθος θεματικά ενεργών ακολουθούμενων
$F5$	πλήθος ακόλουθων που κάνουν tweet πάνω σε ένα θέμα μετά τον συγγραφέα
$F6$	πλήθος ακολουθούμενων που κάνουν tweet πάνω σε ένα θέμα πριν τον συγγραφέα

Πίνακας 5.2: Οι χαρακτηριστικότερες μετρήσιμες ποσότητες στο Twitter (Riquelme & Gonzalez-Cantergiani, 2016).

5.4 Επιδραστικότητα χρηστών

Όπως έχει τονιστεί σε προηγούμενες ενότητες, η μέτρηση της επιδραστικότητας ενός χρήστη δεν είναι ένα καλώς ορισμένο πρόβλημα, καθώς μπορούν να δοθούν πολλοί διαφορετικοί ορισμοί για το πότε ένας χρήστης είναι επιδραστικός. Κατά κύριο λόγω τα μέτρα επιδραστικότητας στο Twitter χωρίζονται σε τρεις κατηγορίες: μέτρα δραστηριότητας, μέτρα δημοτικότητας και μέτρα επιρροής (Riquelme & Gonzalez-Cantergiani, 2016).

5.4.1 Μέτρα δραστηριότητας

Ένας χρήστης θεωρείται δραστήριος όταν η συμμετοχή του στο κοινωνικό δίκτυο είναι συνεχής και συχνή σε συγκεκριμένο χρόνο ανεξάρτητα από την προσοχή που λαμβάνει από άλλους χρήστες (Riquelme & Gonzalez-Cantergiani, 2016). Σημειώνουμε ότι με τον όρο συμμετοχή εννοούμε πράξεις που είναι μετρήσιμες (tweet, retweet, reply, mention κλπ). Για παράδειγμα, κάποιος χρήστης που δαπανά πολύ χρόνο στο δίκτυο διαβάζοντας μόνο tweet δεν αφήνει ίχνη και άρα δε θεωρείται δραστήριος. Η συμμετοχή μπορεί να οριστεί και ως η πιθανότητα ο χρήστης να δει ένα συγκεκριμένο tweet (Yin Zhang, 2012).

Το πιο απλό μέτρο δραστηριότητας στο Twitter είναι ίσως το *TweetRank* μία μετρική που μετρά το πλήθος των tweet κάθε χρήστη, δηλαδή εάν i κόμβος/χρήστης τότε (Nagmoti, Teredesai, & De Cock, 2010):

$$TweetRank(i) = OT1(i) \tag{5.1}$$

Εναλλακτικά χρησιμοποιείται το Tweet count score που μετρά το πλήθος των προτώτυπων tweet και των retweet (Noro, Ru, Xiao, & Tokuda, 2012):

$$TweetCountScore(i) = OT1(i) + RT1(i) \tag{5.2}$$

$$= TweetRank(i) + RT1(i) \tag{5.3}$$

Ορίζουμε επίσης το μέτρο της συνολικής δραστηριότητας του χρήστη i ως:

$$GeneralActivity(i) = OT1(i) + RT1(i) + RP1(i) + FT1(i) \tag{5.4}$$

$$= TweetCountScore(i) + RP1(i) + FT1(i) \tag{5.5}$$

όπου έχει ληφθεί υπ'όψιν και το πλήθος απαντήσεων από το χρήστη ($RP1$) αλλά και το πλήθος tweet που έχουν γίνει like από το χρήστη ($FT1$). Εάν $N(i)$ το συνολικό πλήθος των tweet του χρήστη i μπορούμε να γράψουμε την κανονικοποιημένη μορφή της *γενικής δραστηριότητας* (Riquelme & Gonzalez-Cantergiani, 2016):

$$GeneralActivity(i) = \frac{OT1(i) + RT1(i) + RP1(i) + FT1(i)}{N(i)} \tag{5.6}$$

Καθώς η έννοια του θέματος (topic) είναι κεντρική στο Twitter είναι χρήσιμο να θεωρήσουμε και μετρικές σε συγκεκριμένα θέματα. Μία τέτοια μετρική είναι το *θεματικό σήμα* (Pal & Counts, 2011):

$$TS(i) = \frac{(OT1(i) + RP1(i) + RT1(i))|_{\text{specific topic}}}{N(i)} \tag{5.7}$$

$$= \frac{(GeneralActivity(i) - FT1(i))|_{\text{specific topic}}}{N(i)} \tag{5.8}$$

και η *ένταση σήματος* (signal strength) που υποδεικνύει πόσο ισχυρό είναι το τοπικό σήμα του χρήστη (Pal & Counts, 2011):

$$SS(i) = \frac{OT1}{OT1(i) + RT1(i)} \tag{5.9}$$

$$= \frac{TweetRank(i)}{TweetCountScore(i)} \tag{5.10}$$

μετρά την αυθεντικότητα των tweet του χρήστη.

Υπάρχουν και άλλα ελαφρώς πιο πολύπλοκα μέτρα δραστηριότητας (Riquelme & Gonzalez-Cantergiani, 2016). Το *ActivityScore* (Yuan, Li & Huang, 2013)λαμβάνει υπ' όψιν του το χρόνο. Μετρά το πλήθος των ακόλουθουν, ακολουθούμενων και tweet σε έναν γράφο της μορφής G_3 για κάθε χρήστη σε μία συγκεκριμένη χρονική περιόδο. Το συνολικό πλήθος effective readers όλων των tweet λαμβάνει επίσης υπόψιν του τον χρόνο (Lee, Kwak, Park, & Moon, 2010). Ως effective reader ενός tweet θεωρούμε έναν ακόλουθο που δεν έχει κάνει tweet για κανένα trending topic τη στιγμή που ο χρήστης γράφει ένα προτότυπο tweet. Υπό μία έννοια μετρά την ταχύτητα με την οποία ένας χρήστης tweet για ένα νέο θέμα. Το *DiscussRank* εφαρμόζεται σε multigraphs με κόμβους χρήστες και ακμές τις πιθανές σχέσεις χρήστη-προς-χρήστη (Πίνακας 5.1) και μετρά πόσο δραστήριος είναι ο χρήστης με την της εκκίνησης συζητήσεων γύρω από ένα συγκεκριμένο θέμα (Ben Jabeur, Tamine, & Boughanem, 2012). Το Competency ταξινομεί τους χρήστες σύμφωνα με την ικανότητά τους να δημοσιεύουν tweets σχετικά με επίκαιρα θέματα (Aleahmad, Karisani, Rahgozar, & Oroumchian, 2016). Το IP influence μετρά τόσο την επιδραστικότητα κάθε χρήστη όσο και το passivity του (Romero, Galuba, Asur, & Huberman, 2011). Το passivity του χρήστη ορίζεται ως η δυσκολία ο χρήστης να επηρρεαστεί από έναν άλλο σε μια ορισμενη χρονική περίοδο. Οι περισσότεροι χρήστες σε ένα κοινωνικό δίκτυο είναι άλλωστε παθητικοί υπο την έννοια ότι δεν αλληλεπιδρούν με το διαμοιρασμό περιεχομένου στο δίκτυο. Οι περισσότεροι χρήστες με υψηλό passivity είναι είτε spammers είτε bots. Υπάρχουν ωστόσο ενστάσεις ως προς την συσχέτιση του συγκερκιμένου δείκτη με την επιδραστικότητα (Riquelme & Gonzalez-Cantergiani, 2016).

5.4.2 Μέτρα δημοτικότητας

Δημοφιλής είναι ο χρήστης που αναγνωρίζεται από πολλούς άλλους χρήστες στο δίκτυο. Για παράδειγμα ένας διάσημος ηθοποιός που μπορεί να μην έχει έναν δραστήριο και επιδραστικό λογαριασμό μπορεί παρ' όλα αυτά

να είναι δημοφιλής.

Τα πιο απλά μέτρα δημοτικότητας χρησιμοποιούν μόνο σχέσεις ακολουθίας μεταξύ των χρηστών. Το μέτρο *FollowerRank* είναι η κανονικοποιημένη μορφή του inlink βαθμού κάθε κόμβου $\deg^{in}$ (Riquelme & Gonzalez-Cantergiani, 2016):

$$FollowerRank(i) = \frac{F1(i)}{F1(i) + F3(i)} \tag{5.11}$$

$$= \frac{\deg^{in}(i)}{\deg^{in}(i) + \deg^{out}(i)} \tag{5.12}$$

$$= \frac{\deg^{in}(i)}{\deg(i)} \tag{5.13}$$

Παραλλαγή του συγκεκριμένου μέτρου είναι το *Twitter Follower-Followee ration (TFF)* (Bigonha, Cardoso, Moro, Gonçalves, & Almeida, 2011):

$$TFF(i) = \frac{F1(i)}{F3(i)} \tag{5.14}$$

$$= \frac{\deg^{in}(i)}{\deg^{out}(i)} \tag{5.15}$$

Οι ποσότητες $F1$ και $F3$ μπορεί να διαφέρουν σημαντικά και ο αριθμός ακόλουθων συγκεκριμένων χρηστών να είναι σημαντικά μεγαλύτερος σε σχέση με τους υπόλοιπους χρήστες. Τέτοιου τύπου διαφορές εξομαλύνονται χρησιμοποιώντας το μέτρο δημοτικότητας *Popularity* , το οποίο ορίζεται ως (Aleahmad, Karisani, Rahgozar, & Oroumchian, 2016):

$$Popularity(i) = 1 - e^{-\lambda \cdot F1} \tag{5.16}$$

$$= 1 - e^{-\lambda \cdot \deg^{in}(i)} \tag{5.17}$$

όπου λ μία σταθερά με default τιμή μονάδα.

Το *πηλίκο ακόλουθων/ακολουθούμενων με αντιφατική εκπτωτική αμοιβαιότητα* (Followers to followee ratio with paradoxical discounted reciprocity) ορίζεται ως (Gayo-Avello, 2013):

$$Paradoxicaldiscounted(i) = \begin{cases} F1/F3 & , \text{ αν } F1 > F3 \\ \frac{F1-reciprocal(i)}{F3-reciprocal(i)} & , \text{ διαφορετικά} \end{cases} \tag{5.18}$$

$$= \begin{cases} \deg^{in}(i)/\deg^{out}(i) & , \text{ αν } F1 > F3 \\ \frac{\deg^{in}(i)-reciprocal(i)}{\deg^{out}(i)-reciprocal(i)} & , \text{ διαφορετικά} \end{cases} \tag{5.19}$$

όπου έχει εισαχθεί μία νέα μετρική αυτή του πλήθους των ακόλουθων που είναι και ακολουθούμενοι (reciprocal actors of a user). Χρησιμοποιώντας το συγκεκριμένο μέτρο τιμωρούνται οι spammers (χρήστες με πολλούς

ακολουθούμενους και λίγους ακόλουθους). Ωστόσο ο υπολογισμός του $reciprocal(i)$ αυξάνει σημαντικά το υπολογιστικό κόστος.

Το *Network score* είναι μία θεματική μετρική δημοτικότητας που βασίζεται στους ενεργούς non-reciprocal ακόλουθους του χρήστη (Pal & Counts, 2011):

$$NS(i) = \log(F2(i) + 1) - \log(F4(i) + 1) \tag{5.20}$$

$$= \log\left(\frac{F2(i) + 1}{F4(i) + 1}\right) \tag{5.21}$$

Μπορούμε να ορίσουμε μετρικές δημοτικότητας που δε βασίζονται μόνο σε σχέσεις ακολουθίας. Μία χαρακτηριστική μετρική είναι το *Acquaintance score* που μετρά πόσο γνωστός είναι ο κάθε χρήστης (Srinivasan, Srinivasa, & Thulasidasan, 2013):

$$A(i) = \frac{F1(i) + M4(i) + RP3(i) + RT3(i)}{N} \tag{5.22}$$

όπου N το πλήθος των κόμβων/χρηστών.

Το *Acquaintance-Affinity Score* μετρά πόσο αγαπητός είναι ένας χρήστης λαμβάνοντας υπ' όψιν πόσο γνωστοί είναι οι χρήστες που τον θέλουν (Srinivasan, Srinivasa, & Thulasidasan, 2013):

$$AA(i) = \sum_{j \in E_{RP}(i)} A(j) \cdot \frac{RP(ji)}{RP_{tot}(j)} + \sum_{j \in E_M(i)} A(j) \cdot \frac{M(ji)}{M_{tot}(j)} + \sum_{j \in E_{RT}(i)} A(j) \cdot \frac{RT(ji)}{RT_{tot}(j)} \tag{5.23}$$

όπου:

- $E_{RP}(i)$ το σύνολο των χρηστών που κάνουν reply σε tweet του i
- $RP(ji)$ το πλήθος των reply από τον j στον i
- $RP_{tot}(j)$ το πλήθος των reply του j
- $E_M(i)$ το σύνολο των χρηστών που κάνουν mention tweet του i
- $M(ji)$ το πλήθος mention από το χρήστη j στον i
- $M_{tot}(j)$ το πλήθος των mention του j
- $E_{RT}(i)$ το σύνολο των χρηστών που κάνουν retweet σε tweet του i
- $RT(ji)$ το πλήθος των retweet από τον j στον i
- $RT_{tot}(j)$ το πλήθος των retweet του j

Το *Acquaintance-Affinity-Identification Score* μετρά πόσο αναγνωρίσιμος είναι ο χρήστης i κοιτώντας πόσο κοντά του είναι αυτοί που τον αναγνωρίζουν (Srinivasan, Srinivasa, & Thulasidasan, 2013):

$$AAI(i) = \sum_{j \in Fr(i)} \frac{AA(i)}{Fe_{tot}(i)} \tag{5.24}$$

όπου: $Fr(i)$ το σύνολο των ακόλουθων του i και $Fe_{tot}(i)$ το πλήθος των ακολουθούμενων από τον i. Σημειώνουμε ότι το AAI παρουσιάζει ισχυρή συσχέτιση με τη μετρική $F1$ (Riquelme & Gonzalez-Cantergiani, 2016).

Το Action-Reaction χρησιμοποιείται για τον εντοπισμό διασήμων στο δίκτυο του Twitter (Srinivasan, Srinivasa, & Thulasidasan, 2014). Συνδυάζει δύο μέτρα το Action score που μετρά το πόσο πιστοί είναι οι θαυμαστές του χρήστη και το Reaction score που μετρά την προσοχή που συγκεντρώνει ο διάσημος και οι πράξεις του. Το συγκεκριμένο μέτρο χρησιμοποιεί μεταβλητές δεσμευμένης πιθανότητας που βασίζοντα στα reply, mention και retweet.

Το Starrank είναι ένα μέτρο δημοτικότητας που βασίζεται στον αλγόριθμο PageRank για να κάνει μία ημερήσια δυναμική ανάλυση ενός γράφου G_2 από αναφορές (mention) μεταξύ χρηστών (Khrabrov & Cybenko, 2010). Λαμβάνει υπόψιν ποσότητες όπως η επιτάχυνση των mention συναρτήσει του χρόνου. Παρουσιάζει ενδιαφέρον καθώς δε χρησιμοποιεί άμεσα σχέσεις ακολουθίας αλλά τις φορές που ένας χρήστης γίνεται mention από άλλους λογαριασμούς.

5.4.3 Μέτρα Επιρροής

Όπως έχει τονιστεί επανειλημμένως η επιδραστικότητα ενός χρήστη δέχεται έναν πολύ χαλαρό ορισμό. Επιδραστικός θεωρείται ο χρήστης οι πράξεις του οποίου μπορούν να επηρεάσουν πράξεις πολλών άλλων χρηστών. Στην περίπτωση των κοινωνικών δικτύων, ένας χρήστης με μεγάλη επιρροή είναι και ικανός να μεταδίδει πληροφορίες εντός του δικτύου.

Υπό το φως της ανάλυσης περί δραστήριων και δημοφιλών χρηστών στις προηγούμενες ενότητες αξίζει να αναφέρουμε ότι ένας επιδραστικός χρήστης είναι και ενεργός χρήστης. Η συνθήκη αυτή είναι ικανή ωστόσο δεν είναι αναγκαία. Λίγοι ενεργοί χρήστες είναι και πραγματικά επιδραστικοί. Μία εξίσου σημαντική παρατήρηση είναι ότι ένας επιδραστικός χρήστης στο δίκτυο δεν είναι έχει απαραίτητα και μεγάλη επιρροή στην πραγματική ζωή. Δεν πρέπει να ξεχνάμε πως η ανάλυση της επιδραστικότητας χρηστών κοινωνικών δικτύων περιορίζεται φυσικά στο στενό πλαίσιο του δικτύου. Οποιαδήποτε επέκταση εκτός του θεωρείται αυθαίρετη και καταχρηστική.

Στην ανάλυση επιρροής επικρατούν δύο κύριες οπτικές (Riquelme & Gonzalez-Cantergiani, 2016):

- Η επιρροή μονοπωλείται από λίγους καλά συνδεδεμένους χρήστες ή χρήστες με ισχυρή πειθώ.
- Αρκετοί χρήστες είναι επιδραστικοί κατά λάθος κάτι που εξαρτάται από πολλούς απρόβλεπτους παράγο-

ντες.

Η δεύτερη οπτική αποτελεί και μία ιδιαίτερα ενδιαφέρουσα ιδιαιτερότητα στη μέτρηση επιδραστικότητας κοινωνικών δικτύων. Ωστόσο, παρά τις δυσκολίες που παρουσιάζει αυτός ο απρόβλεπτος παράγοντας μπορούν να οριστούν αρκετοί τρόποι ώστε η επιδραστικότητα να καταστεί μετρήσιμη.

Παραδοσιακές μετρικές κεντρικότητας

Πολλές παραδοσιακές μετρικές επιδραστικότητας/κεντρικότητας όπως αναλύθηκαν σε προηγούμενο κεφάλαιο μπορούν και έχουν χρησιμοποιηθεί για μετρήσεις σε κοινωνικά δίκτυα (Riquelme & Gonzalez-Cantergiani, 2016). Χαρακτηριστική είναι η χρήση της κεντρικότητας closeness που μετρά την ορατότητα/προσβασιμότητα κάθε κόμβου σε σχέση με όλο το υπόλοιπο δίκτυο (εξίσωση (4.9)). Επίσης, έχει γίνει χρήση της κεντρικότητας betweeness που μετρά την ικανότητα κάθε κόμβου να μεσολαβεί στην επικοινωνία μεταξύ των υπόλοιπων κόμβων (εξίσωση (4.12)).

Πέρα από τα παραδοσιακά μέτρα κεντρικότητας μπορεί να γίνει επίσης χρήση του Hirsch index ή H-index (Hirsch, 2010). Ο H-index χρησιμοποιείται για τη μέτρηση της παραγωγικότητας των μελών της επιστημονικής κοινότητας λαμβάνοντας υπ' όψιν του τις ετεροαναφορές δημοσιεύσεων. Συγκεκριμένα στο Twitter ορίζεται ως η μέγιστη τιμή h τ.ω. h τω πλήθος tweet του χρήστη να έχουν γίνει reply/retweet/like τουλάχιστον h φορές. Μια απλοποιημένη εκδοχή του κατασκευάζεται εάν θεωρήσουμε μόνο τα retweet από tweet που περιέχουν URL.

Για τη μέτρηση της κεντρικότητας χρηστών κοινωνικων δικτύων «κεντρικό» ρόλο παίζει ο αλγόριθμος PageRank ιδιαίτερα σε γράφους της μορφής G_2. Αξίζει να σημειωθεί ότι μία σχετικά απλή τροποποίησή του, ο αλγόριθμος NodeRanking,έχει χρησιμοποιηθεί επίσης για δίκτυα με βάρη (Pujol, Sanguesa, & Delgado, 2002). Ωστόσο, καθώς οι συγκεκριμένοι αλγόριθμοι είναι αναδρομικοί (recursive) δεν επιτρέπουν ανάλυση σε πραγματικό χρόνο (το Twitter είναι κατά βάση ένα δυναμικό δίκτυο). Ωστόσο, μπορούν να χρησιμοποιηθούν σε offline ανάλυση. Αξίζει επίσης να σημειωθεί ότι ο παρόμοιος αλγόριθμος HITS που διαχωρίζει τους κόμβους σε hubs και authorities δεν είναι κατάλληλος για χρήση σε κοινωνικά δίκτυα καθώς δίνει ιδιαίτερη δύναμη σε spammers (Riquelme & Gonzalez-Cantergiani, 2016).

Σε αυτό το σημείο και πριν προχωρήσουμε σε εκτενή ανάλυση μετρικών που βασίζονται στον αλγόριθμο PageRankέχει ενδιαφέρον να δούμε μετρικές λιγότερο προφανείς. Για παράδειγμα, έχει χρησιμοποιηθεί μία μετρική εμπνευσμένη από ένα φυσικό σύστημα μεταβλητής μάζας. Το σκορ κάθε χρήστη αντιστοιχίζεται με την ταχύτητα ενός κινητού σε ένα χρονικό διάστημα (Gayo-Avello et al., 2011):

$$v_t = v_{t-1} + \frac{F}{m} - c \tag{5.25}$$

όπου:

- t ένα χρονικό διάστημα (π.χ. 1 ώρα)

- F ο αριθμός mention στο χρονικό διάστημα t
- m ο αριθμός ακόλουθων
- c μία σταθερά με $c \in \mathbb{R}$

Παρά την αυθαίρετη φύση του συγκεκριμένου μέτρου έχει βρεθεί θετική συσχέτιση με τον αριθμό των κλικ σε URL που μάλιστα είναι ανάλογη της συσχέτισης με μέτρα όπως τα IP influence, PageRank, TunkRank.

Εναλλακτικά έχει χρησιμοποιηθεί ο αλγόριθμος k-shell decomposition που χρησιμοποιείται στην αναγνώριση επιδραστικών κόμβων στη δυναμική επιδημιών (Kitsak et al., 2010). Ωστόσο, η φύση της μετάδοσης πληροφοριών και της μετάδοσης ασθενειών είναι διαφορετική. Ένας καλά συνδεδεμένος κόμβος μεταδίδει εύκολα ασθένειες ωστόσο, μπορεί να επιλέγει να μη μεταδώσει μία πληροφορία παρά το γεγονός ότι την γνωρίζει εκ των προτέρων.

Επίσης, το μέτρο F-measure που χρησιμοποιείται στη στατιστική ανάλυση της δυαδικής ταξινόμησης έχει χρησιμοποιηθεί για τη μελέτη της δύναμης των retweet συγκρίνοντας τις σχέσεις ακολουθίας με μετρικές κεντρικότητας (Riquelme & Gonzalez-Cantergiani, 2016).

Μέτρα βασιζόμενα σε μετρήσιμες ποσότητες Twitter και τον αλγόριθμο PageRank

Ένα απλό μέτρο επιδραστικότητας που χρησιμοποιεί τις μετρήσιμες ποσότητες του Πίνακα 5.2 είναι το *Retweet Impact* που υπολογίζει την επίδραση (impact) των tweet ενός χρήστη με βάση τα retweet του (Pal & Counts, 2011):

$$RI(i) = RT2(i) \cdot \log(RT3(i)) \tag{5.26}$$

ο λογαριθμικός όρος $\log(RT3(i))$ μετριάζει τον ενθουσιασμό ορισμένων χρηστών που μπορεί να κάνουν retweet το ίδιο περιεχόμενο πολλές φορές. Μία γενίκευση θα μπορούσε να χρησιμοποιεί και τις μετρικές $FT2$, $FT3$.

Ένα εξίσου απλό μέτρο είναι το *Mention Impact* που δίνει μία εκτίμηση της επιρροής των tweet του χρήστη λαμβάνοντας υπόψιν τα mention που κάνουν άλλοι χρήστες:

$$MI(i) = M3(i) \cdot \log(M4(i)) - M1(i) \cdot \log(M2(i)) \tag{5.27}$$

Ο δεύτερος όρος διασφαλίζει ότι τα mention έχουν γίνει αυθόρμητα και όχι επειδή ο χρήστης έκανε mention κάποιον άλλο χρήστη πρώτος.

Ένα τρίτο απλό μέτρο είναι το *Social Networking Potential* (Anger & Kittl, 2011):

$$SNP(i) = \frac{Ir(i) + RMr(i)}{2} \tag{5.28}$$

όπου:

- $Ir(i)$: το Interactor Ratio

$$Ir(i) = \frac{RT3(i) + M4(i)}{F1(i)} \tag{5.29}$$

το οποίο μετρά πόσοι διαφορετικοί χρήστες αλληλεπιδρούν με τον χρήστη.

- $RMe(i)$: το Retweet and Mention ratio

$$RMr(i) = \frac{\#\text{tweet του } i \text{ που έγιναν retweet } + \#\text{tweet του } i \text{ που έγιναν reply}}{\#\text{tweet του } i} \tag{5.30}$$

το οποίο μετρά πόσα tweet του χρήστη υποδεικνύουν κάποια αλληλεπίδραση με το κοινό του.

Σημειώνουμε ότι το μέτρο $SNP(i)$ λαμβάνει υπόψιν πολλές πράξεις στο Twitter ωστόσο αγνοεί τα like. Το ίδιο συμβαίνει και με το κριτήριο περιεχομένου που μετρά τον αριθμό των δημοσιευμένων tweet και τις σχέσεις ακολουθίας δίνοντας τους βάρος $1/4$ ενώ το κριτήριο συζήτησης μετρά τον αριθμό replykαι τον αριθμό των ακόλουθων που συνδέονται με mention και retweet με τον χρήστη δίνοντας τους βάρος $3/4$.

Εξέχουσα θέση έχουν και πολυπλοκότερα μέτρα που βασίζονται στον αλγόριθμο PageRank. Χαρακτηριστικότερο είναι το μέτρο *TunkRank* (Tunkelang, 2009):

$$TunkRank(i) = \sum_{j \in Followers(i)} \frac{1 + p \cdot TunkRank(j)}{\#foloweesofj} \tag{5.31}$$

όπου σε αντιστοιχία με τον αλγόριθμο PageRank $p \in [0, 1]$ η πιθανότητα ένα tweet να γίνει retweet. Η πιθανότητα θεωρείται ίση για όλους τους χρήστες και παίρνει default τιμή $p = 0.5$.

Μία παραλλαγή του TunkRank είναι το *UserRank* που μετρά την επιρροή λαμβάνοντας υπόψιν τη σχετικότητα των tweet του χρήστη (Majer, & Simko, 2012):

$$UserRank(i) = \sum_{j \in Followers(i)} \frac{1 + \frac{\#followersofi}{tweetsofi} \cdot UserRank(j)}{\#followersofj} \tag{5.32}$$

Σημειώνουμε ότι τα συγκεκριμένα μέτρα αυτά έχουν την τάση να μειώνουν τη σημαντικότητα των spammer.

Το μέτρο *TrueTop* είναι ένα μέτρο επιδραστικότητας ανθεκτικό σε επιθέσεις Sybil (Zhang, Zhang, Sun, Zhang, & Zhang, 2016), δηλαδή σε χρήστες που ανοίγουν πολλούς ψεύτικους λογαριασμούς για να αυξήσουν τεχνητά την επιρροή τους. Εφαρμόζει κεντρικότητα ιδιοδιανύσματος με βάρος σε ένα δίκτυο G_2 με retweet, reply και mention όπου τα βάρη των ακμών αναπαριστούν το πλήθος από κάθε τέτοια αλληλεπίδραση μεταξύ χρηστών. Ο αλγόριθμος χωρίζει το δίκτυο σε μία περιοχή με χρήστες sybil και σε μία χωρίς, με την τελευταία να λαμβάνει υψηλότερες τιμές σκορ.

Το *Diversity-dependent Influence Score (DIS)* είναι μια παραλλαγή του PageRank σε έναν G_2 γράφο από retweet και ακόλουθους (Huang, Liu, Lin, & Cheng, 2013). Δίνει μεγαλύτερο βάρος σε χρήστες που είναι ικανοί να επηρρεάσουν άλλους (σύμφωνα με σχέσεις ακολουθίας) δηλαδή τους λιγότερο πυκνούς χρήστες (χρήστες με

μεγάλο diversity).

Το *Influence Rank (IR)* συνδυάζει σχέσεις ακολουθίας, mention, likes, retweetsγια να εντοπίσει τους ηγέτες κοινής γνώμης (opinion leaders) δηλαδή αυτούς που μπορούν να επηρεάσουν άλλους επιδραστικούς χρήστες (Hajian & White, 2011). Εφαρμόζεται πολυωνυμική προσέγγιση του αλγορίθμου PageRank

Μία άλλη ιδέα για την επιδραστικότητα (Li, Cheng, Chen & Jiang, 2013) είναι ότι ένας χρήστης i επηρεάζει περισσότερο έναν χρήστη j εάν ο i γράφει tweet σχετικά με αυτά που γράφει ο j. Έτσι, μπορεί να κατασκευαστεί μία παραλλαγή του PageRank που βασίζεται στην ομοιότητα των tweet σε έναν γράφο G_2. Ορισμένα προβλήματα παρουσιάζονται που σχετίζονται με το viral περιεχόμενο, ένας χρήστης μπορεί να είναι επιδραστικός χωρίς να διαβάζει συνεχώς τι γράφουν οι άλλοι. Έτσι, ένας επιπλέον παράγοντας είναι αυτός του γενικότερου στυλ των tweet αλλά και του προφίλ του χρήστη.

Το *InfRank* είναι παραλλαγή του PageRank που μετρά την επιδραστικότητα σε σχέση με την ικανότητα του χρήστη να μεταδίδει πληροφορίες που θα κάνουν retweet άλλοι επιδραστικοί χρήστες (Jabeur, Tamine & Boughanem, 2012). Ο αλγόριθμος είναι αρκετά πιο περίπλοκος καθώς χρησιμοποιεί ένα multigraph με κόμβους χρήστες και ακμές τις σχέσεις retweet μεταξύ τους.

Το *LeadRank* μετρά την ηγετική ικανότητα κάθε χρήστη, δηλαδή την ικανότητά του να κάνει κινητοποιεί άλλους χρήστες να κάνουν retweet και mention ειδικά όταν αυτοί οι χρήστες είναι και αυτοί ηγετικές φυσιογνωμίες (Jabeur, Tamine & Boughanem, 2012) . Ο αλγόριθμος αυτός λειτουργεί σε ένα παρόμοιο multigraph με αυτό του InfRank ωστόσο οι ακμές του αναπαριστούν retweets and mentions.

Όλα τα μέτρα που είδαμε παραπάνω βασίζονται κυρίως σε μετρήσιμες ποσότητες που δίνει το Twitter API η διεπαφή προγραμματισμού εφαρμογών που παρέχει το ίδιο το Twitter σε προγραμματιστές. Όμως, υπάρχουν και μέτρα που δεν περιορίζονται από τις ποσότητες που παρέχει το ίδιο το Twitter μέσω του API του. Το *SpreadRank* μετρά το spreadability των χρηστών σε έναν G_2 γράφο από retweet του οποίου οι ακμές έχουν βάρος ανάλογο του ποσοστού των retweet σε σχέση με τον συνολικό αριθμό tweet του χρήστη που έγινε retweet (Ding et al., 2013). Η μετάδοση της επιρροής είναι μεγαλύτερη όσο πιο γρήγορα τα tweet γίνονται retweet. Το μέτρο λαμβάνει υπόψιν πτώσεις πληροφορίας (information cascades) που περιγράφονται ως δέντρο που μεγαλώνει συναρτήσει του χρόνου. Οι κόμβοι είναι τα retweet της αρχικής δημοσίευσης. Όσο πιο κοντά είναι ένας χρήστης στον κόμβο-ρίζα (νωρίτερα στο χρόνο) τόσο περισσότερο μεταδίδεται η επιρροή του. Το συγκεκριμένο μέτρο λαμβάνει υπόψιν τόσο το χρονικό διάστημα μεταξύ των retweet όσο και την τοποθεσία των χρηστών στην πτώση πληροφορίας.

Το *ProfileRank* (Silva, Guimaraes, Meira, & Zaki, 2013) θεωρεί ότι οι χρήστες που επηρεάζουν άλλους (influencer) είναι χρήστες που παράγουν περιεχόμενο σχετικό με άλλους (που θέλουν να δουν άλλοι). Βασίζεται στον αλγόριθμο PageRankκαι υπολογίζεται εκτελώντας τυχαίους περιπάτους σε ένα bipartite γράφημα G_3, του οποίου οι ακμές αναπαριστούν τη δημιουργία και κατανάλωση περιεχομένου από χρήστες συναρτήσει του χρόνου.

Το *MultiRank* είναι ένα πολύπλοκο μέτρο τύπου PageRank το οποίο υπολογίζεται εκτελώντας τυχαίους

περιπάτους σε γραφήματα G_2 από retweet, reply και δύο επιπλέον σχέσεις την reintroduce και read (Ding et al., 2013). Το reintroduce αναφέρεται στη δημοσίευση tweet που είναι παρόμοια με άλλα που έχουν δημοσιευθεί από άλλους χρήστες, χωρίς να κάνουν αναφορά στους πρώτους. Το read αναφέρεται στην πιθανότητα tweet που έχουν γραφτεί από άλλους να διαβαστούν σύμφωνα με τη σειρά εμφάνισής τους στο timeline του χρήστη. Επίσης, γίνεται μία κατηγοριοποίηση των spammer ώστε να λαμβάνουν χαμηλότερες τιμές.

Το *TURank* (Yamaguchi, Takahashi, Amagasa, & Kitagawa, 2010) βασίζεται στο *ObjectRank* το οποίο είναι επέκταση του PageRank. Χρησιμοποιεί έναν G_3 γράφο από ακόλουθους, tweet και retweet. Από ένα άλλο σχετικό γράφημα που περιέχει επίσης reply προστίθεται ένας συντελεστής time-effectiveness attenuation coefficient (TAC) που επιστρέφει τη χρονική στιγμή που το tweet δημοσιεύθηκε ώστε τα tweet που δημοσιεύθηκαν πρώτα να χάνουν σταδιακά τη σημαντικότητά τους (Liu, Wu, & Han, 2013). Σημειώνουμε ωστόσο ότι τέτοιου τύπου μέτρα περιορίζονται από τις μετρήσιμες ποσότητες που παρέχει το ίδιο το Twitter μέσω του API του.

Κεφάλαιο 6

Μελέτες επιδραστικότητας - δίκτυο Higgs Twitter

Στο παρόν κεφάλαιο μελετάμε την επιδραστικότητα χρηστών Twitter σε ένα πραγματικό σύνολο δεδομένων. Όπως αναφέρθηκε σε προηγούμενα κεφάλαια η ταξινόμηση χρηστών ως προς την επιρροή τους είναι ένα ιδιαίτερα σημαντικό πρόβλημα που βρίσκει εφαρμογές στο χώρο του viral μάρκετινγκ αλλά και γενικότερα την προώθηση προτάσεων σε χρήστες μέσων κοινωνικής δικτύωσης που μπορούν να αφορούν αγορές, γεγονότα, δραστηριότητες ακόμα και ανθρώπους.

Υπολογίζουμε το πλήθος των μετρήσιμων ποσοτήτων του δικτύου που παρέχονται από τα δεδομένα σύμφωνα με τον Πίνακα 5.2 καθώς και κάποια από τα μέτρα κεντρικότητας/επιδραστικότητας που παρουσιάζονται σε προηγούμενα κεφάλαια. Επιχειρούμε να αξιολογήσουμε ποιοι είναι οι χρήστες με τη μεγαλύτερη επιρροή στο δίκτυο σύμφωνα με πληθώρα μέτρων επιδραστικότητας, δραστηριότητας και δημοτικότητας καθώς και να αναζητήσουμε πιθανές συσχετίσεις μεταξύ αυτών.

6.1 Παρουσίαση δεδομένων

Το δίκτυο Higgs Twitter κατασκευάστηκε παρακολουθώντας τη διάδοση πληροφοριών στο Twitter πριν, κατά τη διάρκεια και μετά την ανακοίνωση της ανακάλυψης του σωματιδίου Higgs τον Ιούλιο του 2012 στον Ευρωπαϊκό Οργανισμό Πυρηνικής Έρευνας CERN (De Domenico, Lima, Mougel, & Musolesi, 2013). Στο σύνολο των δεδομένων περιέχονται μηνύματα που δημοσιεύθηκαν μεταξύ της 1ης και 7ης Ιουλίου 2012. Τα δεδομένα είναι ελεύθερα διαθέσιμα από το Standford Network Analysis Project γνωστό και ως SNAP ("SNAP: Stanford Network Analysis Project", 2009).

Παρέχονται 4 κατευθυνόμενα γραφήματα που μπορούν να περιγράψουν τη δραστηριότητα των χρηστών:

Δίκτυο	Πλήθος κόμβων $\|V\|$	Πλήθος ακμών E
Σχέσεις ακολουθίας	456626	14855842
Σχέσεις retweet	256491	328132
Σχέσεις reply	38918	32523
Σχέσεις mention	116408	150818

Πίνακας 6.1: Πλήθος κόμβων και ακμών για το σύνολο δεδομένων Higgs Twitter.

- δίκτυο των retweet των tweet των χρηστών
- δίκτυο των reply σε tweet χρηστών
- δίκτυο των mention σε άλλους χρήστες
- δίκτυο των σχέσεων ακολουθίας των χρηστών

Τα user ID, δηλαδή οι ταυτότητες των χρηστών, έχουν μετατραπεί σε αριθμούς ώστε κάθε χρήστης να αντιστοιχείται μοναδικά σε έναν αριθμό. Η μετατροπή αυτή εξασφαλίζει την ανωνυμία των χρηστών. Τα τέσσερα δίκτυα που αναφέρονται παραπάνω, χρησιμοποιούν τα ίδια user Id δηλαδή κάθε πράξη στα επί μέρους δίκτυα αφορά το ίδιο σύνολο χρηστών. Έτσι, ένα δίκτυο αφορά την κοινωνική δομή (σχέσεις ακολουθίας) και τρία επιπλέον επίπεδα κωδικοποιούν τις διαφορετικές αλληλεπιδράσεις.

Ο βασικός κορμός του κώδικα που χρησιμοποιούμε γράφεται σε μορφή `Jupyter notebook` εκμεταλλευόμενοι την διαδραστική φύση των notebook ενώ το σύνολο των συναρτήσεων γράφεται σε επιμέρους αρχεία Python `.py`. Το σύνολο του κώδικα διατίθεται με ελεύθερη πρόσβαση στο Github [1].

Επιλέγουμε να εισάγουμε τα δεδομένα μας σε μορφή pandas DataFrame για να εκμεταλλευτούμε πολλές χρήσιμες λειτουργίες που παρέχει το συγκεκριμένο module. Έτσι έχουμε:

```
import pandas as pd
df_social = pd.read_table('./data/higgs-social_network.edgelist', sep = ' ', names = ['A', 'B'])
df_mention = pd.read_table('./data/higgs-mention_network.edgelist', sep = ' ', names = ['A', 'B',
    'w'])
df_retweet = pd.read_table('./data/higgs-retweet_network.edgelist', sep = ' ', names = ['A', 'B',
    'w'])
df_reply = pd.read_table('./data/higgs-reply_network.edgelist', sep = ' ', names = ['A', 'B', 'w'
    ])
```

Τα δεδομένα για το γράφημα των σχέσεων ακολουθίας `df_social` έχουν την εξής μορφή:

```
df_social
```

	A	B
0	1	2

[1] https://github.com/lbitsiko/twitter-influence-master-thesis

```
1               1       3
2               1       4
3               1       5
4               1       6
...           ...     ...
14855837   456624       1
14855838   456625     220
14855839   421799   81585
14855840   421799  100470
14855841   456626       1
```

Κάθε γραμμή του dataframe υπονοεί ότι ο χρήστης που αναγράφεται στη στήλη A ακολουθεί τον χρήστη που αναφέρεται στη στήλη B, δηλαδή οι κόμβοι A και B συνδέονται με μία κατευθυνόμενη ακμή από τον A στον B.

Τα δεδομένα για τα γραφήματα των σχέσεων mention, reply και retweet έχουν την εξής μορφή (ενδεικτικά παρουσιάζουμε το mention):

```
df_mention
```

```
                A       B  w
0        316609    5011  1
1        439696   12389  1
2         60059    6929  1
3        161345    8614  1
4        137487     759  1
...         ...     ... ..
150813    82342    1343  1
150814   341806   29259  1
150815   142102    2164  1
150816    25907     677  2
150817   170251     677  1
```

Κάθε γραμμή υπονοεί ότι ο χρήστης που αναγράφεται στη στήλη A κάνει mention τον χρήστη της στήλης B τόσες φορές όσες αναγράφονται στη στήλη w (αντίστοιχα για τα retweet και reply).

Στη συνέχεια μετατρέπουμε το dataframe με τις σχέσεις ακολουθίας σε γράφο χρησιμοποιώντας το module `networkx`.

```
import time
start = time.time()
```

```
3
4 G_social = nx.from_pandas_edgelist(df_social, source = 'A', target = 'B', create_using=nx.DiGraph
      )
5
6 end = time.time()
7 print(f'{end - start} seconds')
8 print(f'{(end - start)/60.} min')
```

```
479.2698624134064 seconds
7.98783104022344 min
<networkx.classes.digraph.DiGraph at 0x1a95b7e6dc8>
```

βλέπουμε ότι και μόνο η μετατροπή των δεδομένων σε κατευθυνόμενο γράφημα είναι αρκετά χρονοβόρα καθώς απαιτεί περίπου 8 λεπτά. Το μεγάλο υπολογιστικό κόστος οφείλεται φυσικά στον μεγάλο όγκο δεδομένων (Πίνακας 6.1):

```
1 len(G_social.nodes())
```

```
456626
```

```
1 len(G_social.edges())
```

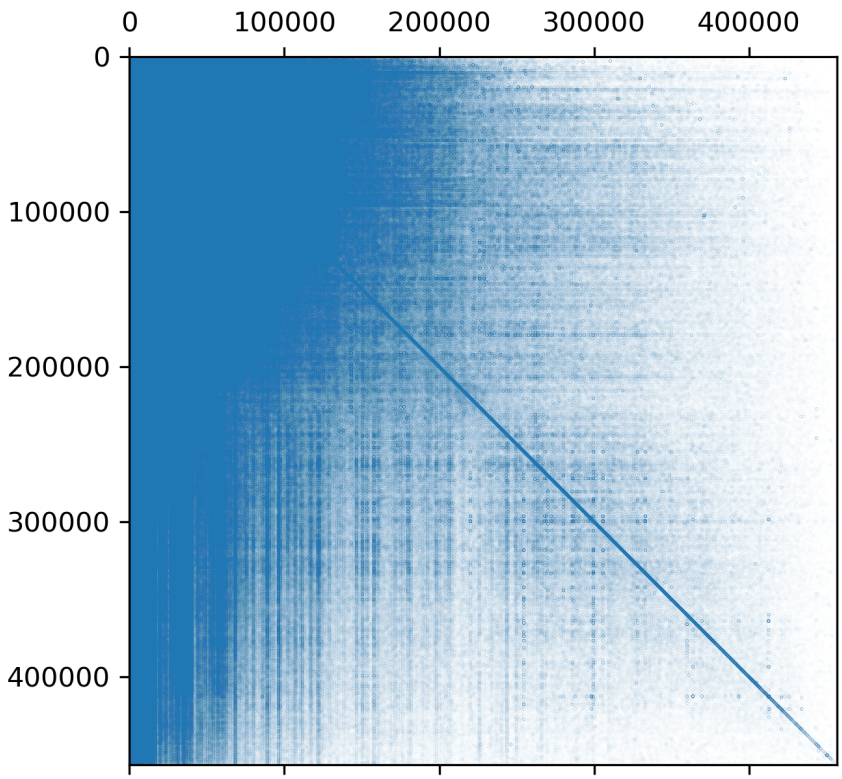
```
14855842
```

Ενδεικτικά σχεδιάζουμε και τον πίνακα γειτνίασης σε μορφή διαγράμματος (Σχήμα 6.1):

```
1 import scipy.sparse as sparse
2
3 start = time.time()
4
5 fig, ax = plt.subplots(1,1, figsize=(6.4, 4.8))
6 ax.spy(nx.adjacency_matrix(G_social), markersize =0.01, marker = '.')
7
8 fig.savefig('./plots/adj_matr.png', dpi = 300)
9 end = time.time()
10 print(f'{end - start} seconds')
11 print(f'{(end - start)/60.} min')
```

```
368.95525789260864 seconds
6.149254298210144 min
```

Γι' αυτό το λόγο κατά τη διάρκεια της υλοποίησης είναι ιδιαίτερα χρήσιμο να αποθηκεύουμε ορισμένες υπολογιστικά κοστοβόρες μεταβλητές στο δίσκο, χρησιμοποιώντας το module `pickle`



Σχήμα 6.1: Ο πίνακας γειτνίασης του γραφήματος. Παρατηρούμε ότι το γράφημα έχει μεγάλο αριθμό κόμβων ενώ είναι σημαντικά πυκνό.

```
import pickle
with open('./pickles/G_social.pickle', 'wb') as file:
    pickle.dump(G_social, file)
```

έτσι ο ακριβός υπολογισμός γίνεται μία φορά και στη συνέχεια εάν χρειαστεί μπορεί να διαβαστεί από το δίσκο σε σημαντικά λιγότερο χρόνο:

```
start = time.time()

with open('./pickles/G_social_DiGraph.pickle', 'rb') as file:
    G_social = pickle.load(file)

end = time.time()
print(f'{end - start} seconds')
print(f'{(end - start)/60.} min')
```

```
39.184739112854004 seconds
0.6530789852142334 min
```

έτσι σε περίπτωση που αναγκαστούμε να διακόψουμε την υλοποίηση, μπορούμε πολύ γρηγορότερα να συνεχίσουμε από το σημείο που είχαμε σταματήσει, μειώνοντας σημαντικά τον χρόνο υλοποίησης της λύσης μας.

6.2 Υπολογισμός μετρήσιμων ποσοτήτων Twitter

Οι μετρήσιμες ποσότητες που αναφέρονται στον Πίνακα 5.2 μπορούν να υπολογιστούν άμεσα από τα τέσσερα dataframe καθώς και το γράφημα `G social` .

Σημειώνουμε ωστόσο ότι η φύση των δεδομένων δε μας επιτρέπει τον υπολογισμό όλων των ποσοτήτων που αναφέρονται στον Πίνακα 5.2. Οι ποσότητες που θα υπολογίσουμε και στη συνέχεια θα χρησιμοποιήσουμε για τον εντοπισμό των πιο επιδραστικών κόμβων είναι οι εξής:

- $F1$: πλήθος follower
- $F3$: πλήθος ακολουθούμενων
- $M1$: πλήθος mention προς άλλους χρήστες από τον χρήστη
- $M2$: πλήθος χρηστών που έχουν γίνει mentionαπό τον χρήστη
- $M3$: πλήθος mention από άλλους χρήστες προς τον χρήστη
- $M4$: πλήθος χρηστών που έχουν κάνει mention τον χρήστη
- $RT1$: πλήθος retweet του χρήστη
- $RT2$: πλήθος πρωτότυπων tweet δημοσιευμένα από το χρήστη και retweeted από άλλους
- $RT3$: πλήθος χρηστών που έχουν κάνει retweet σε tweet του χρήστη
- $RP1$: πλήθος απαντήσεων (reply) δημοσιευμένα από το χρήστη
- $RP3$: πλήθος χρηστών που έχουν απαντήσει στα tweet του χρήστη

Για τον υπολογισμό των υπολοίπων μετρήσιμων ποσοτήτων $OT1, OT2, OT3, RP2, FT1, FT2, FT3, F2, F4, F4, F6$ χρειαζόμαστε περισσότερες πληροφορίες σχετικά με τα θέματα των tweet καθώς και τη δραστηριότητα των χρηστών.

Παρά τις όποιες ελλείψεις του συνόλου των δεδομένων, όπως θα δούμε στη συνέχεια, ο υπολογισμός των πιο κεντρικών κόμβων παραμένει υπολογιστικά απαιτητικός.

6.2.1 Ποσότητες F1, F3

Οι μετρικές $F1$ και $F3$ αντιστοιχούν στους inlink και outlink βαθμούς ($\deg^{in}$ και $\deg^{out}$ του γραφήματος) και επομένως ο υπολογισμός τους είναι σχετικά απλός. Επιλέγουμε να υλοποιήσουμε δύο συναρτήσεις που μπορούν να χρησιμοποιηθούν για οποιοδήποτε κατάλληλο γράφημα:

```
def f1_metric(graph):
    """
        F1: Number of followers.
    """
    return dict(graph.in_degree)


def f3_metric(graph):
    """
        F3: Number of followees.
    """
    return dict(graph.out_degree)
```

Οι μετρικές δίνονται στη μορφή ενός python dictionary [2] με κλειδιά τους κόμβους του γραφήματος και τιμές τις αντίστοιχες τιμές για κάθε κόμβο. Η επιλογή του dictionary κρίνεται η πιο κατάλληλη για όλες τις μετρικές καθώς ο αριθμός των κόμβων είναι ιδιαίτερα μεγάλος ($\sim 5 \times 10^5$).

Ενδεικτικά δίνεται η μορφή του python dictionary στον οποίο αποθηκεύονται οι τιμές της παρατηρήσιμης ποσότητας $F1$:

```
f1_metric(G_social)
```

```
{1: 16280,
 2: 4707,
 3: 137,
 4: 8643,
 5: 2194,
 6: 27088,
 7: 2146,
 8: 32106,
 9: 567,
 10: 10204,
 ...
```

[2] Τα python dictionary είναι μία δομή δεδομένων που ονομάζεται associative array και αποτελείται από ζεύγη (κλειδιά, τιμές).

6.2.2 Μετρήσιμες ποσότητες $M1$, $M2$, $M3$, $M4$

Ο αλγόριθμος υπολογισμού των ποσοτήτων που αναφέρονται στα mention είναι ελαφρώς πιο περίπλοκος, ωστόσο τα τέσσερα μέτρα μοιράζονται μία κοινή αλγοριθμική δομή.

Θα περιγράψουμε εκτενώς την περίπτωση $M1$ και $M2$ ενώ οι $M3$ και $M4$ είναι αρκετά παρόμοιες. Για αρχή, κατασκευάζουμε ένα κενό dictionary `m1_dict` με κλειδιά τους κόμβους του γράφου και μηδενικές τιμές (αρχικοποίηση):

```
m1_dict = {}
for i in graph_social.nodes:
    m1_dict[i] = 0.0
```

Στη συνέχεια ανατρέχουμε στο σύνολο των γραμμών του `df_social`η κάθε τιμή στη στήλη A είναι το κλειδί για το `m1_dict`και αθροίζουμε τον αριθμό αναφορών στη στήλη `w`

```
for _, row in df_mention.iterrows():
    m1_dict[row.A] += row.w
```

έτσι σε μορφή συνάρτησης έχουμε:

```
def m1_metric(graph_social, df_mention):
    """
        M1: Number of mentions to other users by the author.
    """
    m1_dict = {}
    for i in graph_social.nodes:
        m1_dict[i] = 0.0
    for _, row in df_mention.iterrows():
        # if row.A in m1_dict.keys():
        #     m1_dict[row.A] += row.w
        try:
            m1_dict[row.A] += row.w
        except KeyError:
            pass
    return m1_dict
```

όπου επίσης έχει ληφθεί υπόψιν η περίπτωση να χρησιμοποιούμε κάποιο υποσύνολο των δεδομένων.

Ακριβώς την ίδια αλγοριθμική δομή έχει και η συνάρτηση υπολογισμού της $M2$, ωστόσο αντί να αθροίζουμε τη στήλη `w`απλώς αθροίζουμε μονάδες καθώς ενδιαφερόμαστε για τον αριθμό των χρηστών που γίνονται mention από τον κάθε χρήστη, έτσι:

```
def m2_metric(graph_social, df_mention):
    """
        M2: Number of users mentioned by the author.
    """
    m2_dict = {}
```

```
    for i in graph_social.nodes:
        m2_dict[i] = 0.0
    for _, row in df_mention.iterrows():
        # if row.A in m2_dict.keys():
        #     m2_dict[row.A] += 1.0
        try:
            m2_dict[row.A] += 1.0
        except KeyError:
            pass
    return m2_dict
```

Όμοια υπολογίζονται και οι ποσότητες $M3$ και $M4$ χρησιμοποιώντας ως κλειδί τη στήλη B:

```
def m3_metric(graph_social, df_mention):
    """
        M3: Number of mentions to the author by other users.
    """
    m3_dict = {}
    for i in graph_social.nodes:
        m3_dict[i] = 0.0
    for _, row in df_mention.iterrows():
        try:
            m3_dict[row.B] += row.w
        except KeyError:
            pass
        # if row.B in m3_dict.keys():
        #     m3_dict[row.B] += 1.0
    return m3_dict

def m4_metric(graph_social, df_mention):
    """
        M4: Number of users mentioning the author
    """
    m4_dict = {}
    for i in graph_social.nodes:
        m4_dict[i] = 0.0
    for _, row in df_mention.iterrows():
        try:
            m4_dict[row.B] += 1.0
        except KeyError:
            pass
    return m4_dict
```

6.2.3 Μετρήσιμες ποσότητες $RT1$, $RT2$, $RT3$

Οι μετρήσιμες ποσότητες για τα retweet δε διαφέρουν καθόλου από τις αντίστοιχες για τα mention. Η υλοποίησή τους είναι αρκετά παρόμοια με αυτή των $M1$, $M2$, $M3$, $M4$ χρησιμοποιώντας ωστόσο το αντίστοιχο dataframe `df_retweet`

```
def rt1_metric(graph_social, df_retweet):
    """
        RT1: Number of retweets accomplished by the author.
    """
    rt1_dict = {}
    for i in graph_social.nodes:
        rt1_dict[i] = 0.0
    for _, row in df_retweet.iterrows():
        try:
            rt1_dict[row.A] += row.w
        except KeyError:
            pass
    return rt1_dict


def rt2_metric(graph_social, df_retweet):
    """
        RT2: Number of OTs posted by the author and retweeted by other users.
    """
    rt2_dict = {}
    for i in graph_social.nodes:
        rt2_dict[i] = 0.0
    for _, row in df_retweet.iterrows():
        try:
            rt2_dict[row.B] += row.w
        except KeyError:
            pass
    return rt2_dict


def rt3_metric(graph_social, df_retweet):
    """
        RT3: Number of users who have retweeted at[U+FFFD]s tweets.
    """
    rt3_dict = {}
    for i in graph_social.nodes:
        rt3_dict[i] = 0.0
    for _, row in df_retweet.iterrows():
```

```
        try:
            rt3_dict[row.B] += 1.0
        except KeyError:
            pass
    return rt3_dict
```

6.2.4 Μετρήσιμες ποσότητες $RP1$, $RP3$

Οι μετρήσιμες ποσότητες για τα reply υπολογίζονται με αντίστοιχο τρόπο χρησιμοποιώντας φυσικά το `df_reply`:

```
def rp1_metric(graph_social, df_reply):
    """
        RP1: Number of replies posted by the author.
    """
    rp1_dict = {}
    for i in graph_social.nodes:
        rp1_dict[i] = 0.0
    for _, row in df_reply.iterrows():
        try:
            rp1_dict[row.A] += row.w
        except KeyError:
            pass
    return rp1_dict


def rp3_metric(graph_social, df_reply):
    """
        RP3: Number of users who have replied to[U+FFFD]s tweets.
    """
    rp3_dict = {}
    for i in graph_social.nodes:
        rp3_dict[i] = 0.0
    for _, row in df_reply.iterrows():
        try:
            rp3_dict[row.B] += 1.0
        except KeyError:
            pass
    return rp3_dict
```

6.2.5 Συνάρτηση υπολογισμού μετρήσιμων ποσοτήτων

Σε αυτό το σημείο είναι χρήσιμο να συλλέξουμε το σύνολο των υπολογισμών σε έναν βρόγχο εντός μίας συνάρτησης η οποία θα δέχεται το γράφημα, τα dataframe με τις σχέσεις ακολουθίας, reply, mention και retweet και θα επιστρέφει ένα dictionary με κλειδιά την κάθε ποσότητα και τιμές τα επιμέρους dictionary των ποσοτήτων:

```
def all_metrics(graph, df_mention, df_reply, df_retweet):
    return_dict = {
        'f1': f1_metric(graph),
        'f3': f3_metric(graph),
        'm1': {},
        'm2': {},
        'm3': {},
        'm4': {},
        'rt1': {},
        'rt2': {},
        'rt3': {},
        'rp1': {},
        'rp3': {}
    }
    for i in graph.nodes():
        return_dict['m1'][i] = 0.0
        return_dict['m2'][i] = 0.0
        return_dict['m3'][i] = 0.0
        return_dict['m4'][i] = 0.0
        return_dict['rt1'][i] = 0.0
        return_dict['rt2'][i] = 0.0
        return_dict['rt3'][i] = 0.0
        return_dict['rp1'][i] = 0.0
        return_dict['rp3'][i] = 0.0

    for _, row in df_mention.iterrows():
        if row.A in return_dict['m1'].keys():
            return_dict['m1'][row.A] += row.w
        if row.A in return_dict['m2'].keys():
            return_dict['m2'][row.A] += 1.0
        if row.B in return_dict['m3'].keys():
            return_dict['m3'][row.B] += row.w
        if row.B in return_dict['m4'].keys():
            return_dict['m4'][row.B] += 1.0

    for _, row in df_retweet.iterrows():
        if row.A in return_dict['rt1'].keys():
            return_dict['rt1'][row.A] += row.w
```

```
        if row.B in return_dict['rt2'].keys():
            return_dict['rt2'][row.B] += row.w
        if row.B in return_dict['rt3'].keys():
            return_dict['rt3'][row.B] += 1.0

    for _, row in df_reply.iterrows():
        if row.A in return_dict['rp1'].keys():
            return_dict['rp1'][row.A] += row.w
        if row.B in return_dict['rp3'].keys():
            return_dict['rp3'][row.B] += 1.0
    return return_dict
```

Έτσι, έχουμε συλλέξει σε μία μεταβλητή το σύνολο των μετρήσιμων ποσοτήτων του δικτύου.

6.2.6 Υπολογισμός ποσοτήτων και γράφημα

Οι συναρτήσεις των προηγούμενων ενοτήτων αποθηκεύονται σε ένα αρχείο `metrics.py` το οποίο και εισάγουμε στο notebook. Υπολογίζουμε το σύνολο των μετρήσιμων ποσοτήτων ως εξής:

```
# Compute all metrics from dataframes

start = time.time()

import metrics
all_metrics = metrics.all_metrics(G_social, df_mention, df_reply, df_retweet)

end = time.time()
print(f'{end - start} seconds')
print(f'{(end - start)/60.} min')
```

```
69.0646619796753 seconds
1.1510776996612548 min
```

Τα κλειδιά του dictionary:

```
all_metrics.keys()
```

```
dict_keys(['f1', 'f3', 'm1', 'm2', 'm3', 'm4', 'rt1', 'rt2', 'rt3', 'rp1', 'rp3'])
```

Οι ποσότητες έχουν την μορφή:

```
all_metrics['f1']
```

```
{1: 16280,
2: 4707,
3: 137,
4: 8643,
5: 2194,
6: 27088,
7: 2146,
8: 32106,
9: 567,
10: 10204,
...
```

Μετατρέπουμε το σύνολο των μετρήσιμων ποσοτήτων σε dataframe για καλύτερη εποπτεία:

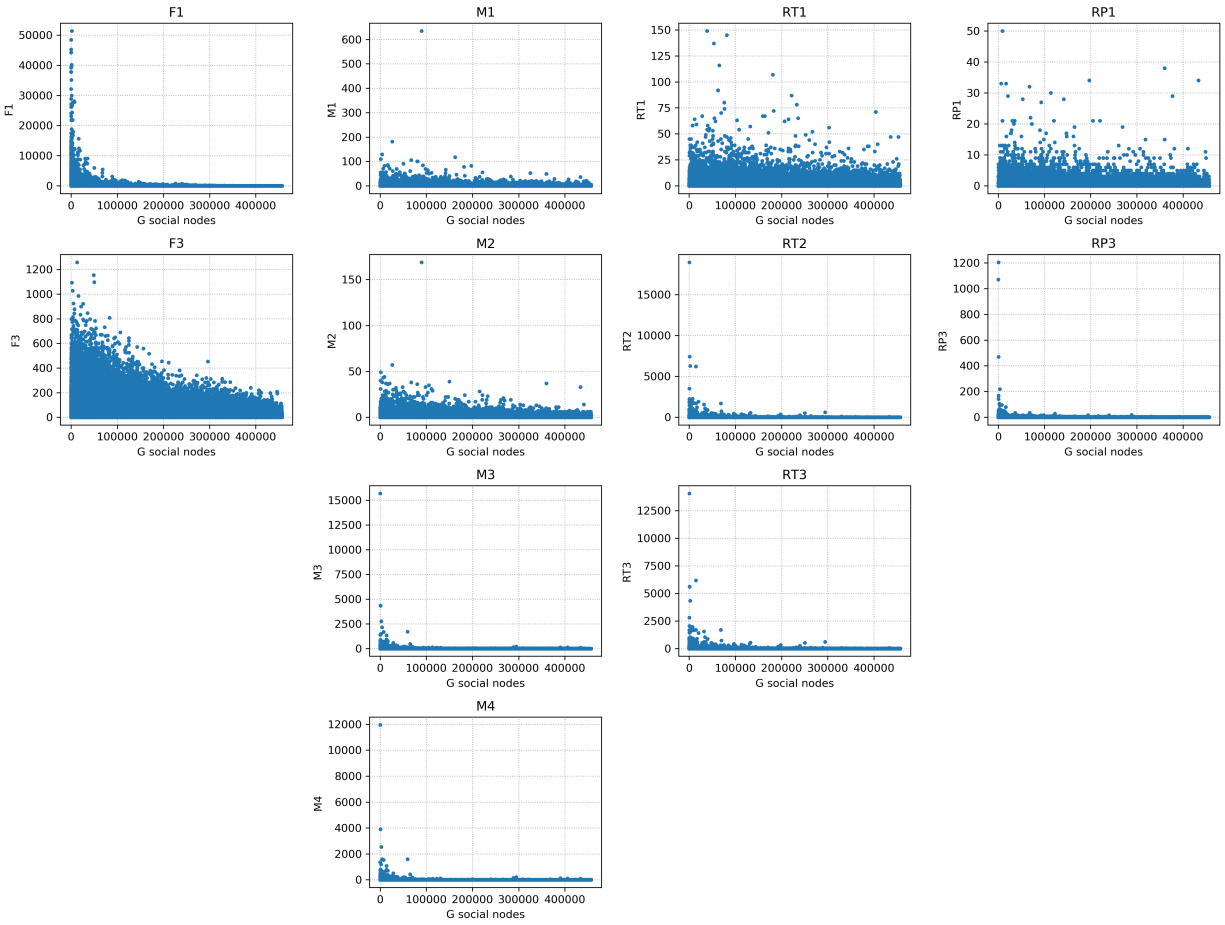
```
# Metrics stored in dataframe
df_metrics = pd.DataFrame()
df_metrics['nodeId'] = G_social.nodes()

from utilities import list_of_values

for key in all_metrics.keys():
    df_metrics[key] = list_of_values(all_metrics[key])
df_metrics
```

	nodeId	f1	f3	m1	m2	m3	m4	rp1	rp3	rt1	rt2	rt3
1	1	16280	22	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	2	4707	77	0.0	0.0	2.0	2.0	0.0	1.0	1.0	0.0	0.0
3	3	137	25	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
4	4	8643	402	7.0	6.0	106.0	104.0	1.0	3.0	7.0	86.0	77.0
5	5	2194	58	0.0	0.0	4.0	4.0	0.0	1.0	0.0	24.0	24.0
...	...	...	...	...	...	...	...	...	...	...	...	...
456622	456622	0	2	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
456623	456623	0	12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
456624	456624	0	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
456625	456625	0	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
456626	456626	0	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Η κατανομή των μετρικών για κάθε κόμβο φαίνεται στο Σχήμα 6.2 και σχεδιάζεται χρησιμοποιώντας τις εξής εντολές:

```
plt.close('all')
fig, axs = plt.subplots(4,4, figsize=(2.5* 6.4, 2.5*4.8),tight_layout = True)#,dpi = 300)

axs[0][0].set_title('F1')
axs[1][0].set_title('F3')
for i in [0,1,2,3]:
    axs[i][1].set_title(f'M{i+1}')
axs[0][2].set_title('RT1')
axs[1][2].set_title('RT2')
axs[2][2].set_title('RT3')
axs[0][3].set_title('RP1')
axs[1][3].set_title('RP3')

for i in axs:
    for j in i:
        j.set_xlabel('G social nodes')
        j.grid(ls = ':')

node_size_factor = 5

axs[0][0].plot(df_metrics.nodeId, df_metrics.f1, '.', markersize = node_size_factor)
axs[0][0].set_ylabel('F1')

axs[1][0].plot(df_metrics.nodeId, df_metrics.f3, '.', markersize = node_size_factor)
axs[1][0].set_ylabel('F3')

axs[2][0].axis('off')
axs[3][0].axis('off')

axs[0][1].plot(df_metrics.nodeId, df_metrics.m1, '.', markersize = node_size_factor)
axs[0][1].set_ylabel('M1')

axs[1][1].plot(df_metrics.nodeId, df_metrics.m2, '.', markersize = node_size_factor)
axs[1][1].set_ylabel('M2')

axs[2][1].plot(df_metrics.nodeId, df_metrics.m3, '.', markersize = node_size_factor)
axs[2][1].set_ylabel('M3')

axs[3][1].plot(df_metrics.nodeId, df_metrics.m4, '.', markersize = node_size_factor)
axs[3][1].set_ylabel('M4')

axs[0][2].plot(df_metrics.nodeId, df_metrics.rt1, '.', markersize = node_size_factor)
axs[0][2].set_ylabel('RT1')

```

```
45 axs[1][2].plot(df_metrics.nodeId, df_metrics.rt2, '.', markersize = node_size_factor)
46 axs[1][2].set_ylabel('RT2')
47
48 axs[2][2].plot(df_metrics.nodeId, df_metrics.rt3, '.', markersize = node_size_factor)
49 axs[2][2].set_ylabel('RT3')
50
51 axs[3][2].axis('off')
52
53 axs[0][3].plot(df_metrics.nodeId, df_metrics.rp1, '.', markersize = node_size_factor)
54 axs[0][3].set_ylabel('RP1')
55
56 axs[1][3].plot(df_metrics.nodeId, df_metrics.rp3, '.', markersize = node_size_factor)
57 axs[1][3].set_ylabel('RP3')
58
59 axs[2][3].axis('off')
60 axs[3][3].axis('off')
61 fig.savefig('./plots/metrics.png', dpi = 300)
```



Σχήμα 6.2: Τιμές των μετρικών $F1$, $F3$, $M1$, $M2$, $M3$, $M4$, $RT1$, $RT2$, $RT3$, $RP1$, $RP3$ για τους κόμβους του γράφου των χρηστών.

Όπως είναι φανερό η γνώση μόνο των τιμών των μετρικών δεν μας δίνει εικόνα για την επιδραστικότητα των χρηστών. Για κάτι τέτοιο πρέπει να καταφύγουμε σε μέτρα επιδραστικότητας.

6.3 Μέτρα επιδραστικότητας

Ο κώδικας των μέτρων επιδραστικότητας βρίσκεται στο αρχείο `measures.py` με τη μορφή συναρτήσεων το οποίο και εισάγουμε αργότερα στο notebook.

6.3.1 FollowerRank

Για το μέτρο επιδραστικότητας FollowerRank χρησιμοποιούμε την εξίσωση (5.11). Η συνάρτηση δέχεται ως ορίσματα το γράφημα των σχέσεων ακολουθίας και τις τιμές των $F1$ και $F2$ ενώ επιστρέφει τιμές για κάθε κόμβο υπό μορφή dictionary:

```
def follower_rank(graph, f1, f3):
    """
        followerRank(i) = F1 / (F1 + F3)
        Nagmoti et al. (2010)
    """
    # degree_dict = dict(graph.degree)
    fr_dict = {}
    for i in graph.nodes:
        if float(f1[i] + f3[i]) != 0.0:
            fr_dict[i] = float(f1[i]) / float(f1[i] + f3[i])
        else:
            fr_dict[i] = 1.0
        # fr_dict[i] = float(f1[i]) / float(f1[i] + f3[i])
    return fr_dict
```

6.3.2 TFF

Ο κώδικας για το TFF βασίζεται στην εξίσωση 5.14 και είναι ο εξής:

```
def tff(graph, f1, f3):
    """
        Twitter Follower-Followee ratio
        TFF(i) = F1 / F3
        Bigonha, Cardoso, Moro, G[U+FFFD]ves and Almeida (2012)
    """
    tff_dict = {}
    for i in graph.nodes:
        try:
            tff_dict[i] = float(f1[i]) / float(f3[i])
        except ZeroDivisionError:
            tff_dict[i] = 1.0  # 'infty'
    return tff_dict
```

6.3.3 Popularity

Το μέτρο Popularity βασίζεται στην εξίσωση (5.16) και υλοποιείται ως εξής:

```
def popularity(graph, f1):
    """
        Popularity(i) = 1 - exp(-F1)

        Aleahmad et al. (2015)
    """
    pop_dict = {}
```

```
    for i in graph.nodes:
        pop_dict[i] = 1.0 - np.exp(0.0 - f1[i])
    return pop_dict
```

6.3.4 A-score

Το μέτρο Acquaintance Score βασίζεται στην εξίσωση (5.22) με αντίστοιχη υλοποίηση:

```
def a_score(graph, f1, m4, rp3, rt3):
    """
        Acquaintance Score

        A(i) = (F1 + M4 + RP3 + RT3) / N

        Srinivasan et al. (2013)
    """
    a_score_dict = {}
    for i in graph.nodes():
        val = f1[i] + m4[i] + rp3[i] + rt3[i] / len(graph.nodes())
        a_score_dict[i] = val
    return a_score_dict
```

6.3.5 Retweet Impact

Το μέτρο Retweet Impact βασίζεται στην εξίσωση (5.26) και υλοποιείται:

```
def retweet_impact(graph_social, rt2, rt3):
    """
        Retweet Impact

        RI(i) = RT2 * log(RT3)

        Pal and Counts (2011)
    """
    ri_dict = {}
    for i in graph_social.nodes():
        if rt3[i] != 0.0:
            ri_dict[i] = rt2[i] * np.log(rt3[i])
        else:
            ri_dict[i] = 1.0  # '-infty'
    return ri_dict
```

6.3.6 Mention Impact

Για την υλοποίηση του mention impact βασιζόμαστε στην εξίσωση (5.27):

```
def mention_impact(graph_social, m1, m2, m3, m4):
    """
        Mention Impact

        Mi(i) = M3 * log(M4) - M1 * log(M2)

        Pal and Counts (2011)
    """
    mi_dict = {}
    for i in graph_social.nodes():
        if m4[i] != 0.0 and m2[i] != 0.0:
            mi_dict[i] = m3[i] * np.log(float(m4[i])) - m1[i] * np.log(float(m2[i]))
        else:
            mi_dict[i] = 1.0  # '-infty'
    return mi_dict
```

6.3.7 Παραδοσιακά μέτρα κεντρικότητας και αλγόριθμοι

Για παραδοσιακά μέτρα κεντρικότητητας και αλγορίθμους καταφεύγουμε στο module `networkx`. Συγκεκριμένα χρησιμοποιούμε:

- `nx.eigenvector_centrality`
- `nx.pagerank`
- `nx.betweenness_centrality`
- `nx.degree_centrality`
- `nx.in_degree_centrality`
- `nx.closeness_centrality`

Ωστόσο, ειδικά για το closeness χρησιμοποιούμε έναν αλγόριθμο προσέγγισης των τιμών closeness (pj, 2018)

```
import scipy.sparse
import scipy.sparse.csgraph
def closeness_centrality_approx(graph):
    """
        Closeness centrality approximation
        based on:
```

```
        https://medium.com/@pasdan/closeness-centrality-via-networkx-is-taking-too-long-1a58e648f5ce
    """
    adj_matr = nx.adjacency_matrix(graph).tolil()
    dcap = scipy.sparse.csgraph.floyd_warshall(adj_matr, directed=True, unweighted=False)

    n = dcap.shape[0]
    closeness_centrality = {}
    for r in range(0, n):
        cc = 0.0
        possible_paths = list(enumerate(dcap[r, :]))
        shortest_paths = dict(filter(lambda x: not x[1] == np.inf, possible_paths))

        total = sum(shortest_paths.values())
        n_shortest_paths = len(shortest_paths) - 1.0

        if total > 0.0 and n > 1:
            s = n_shortest_paths / (n - 1)
            cc = (n_shortest_paths / total) * s
        closeness_centrality[r] = cc

    return closeness_centrality
```

καθώς ο builtin αλγόριθμος είναι σημαντικά πιο αργός.

6.3.8 Υπολογισμοί μέτρων κεντρικότητας

Οι υπολογισμοί των μέτρων κεντρικότητας και οι αντίστοιχοι χρόνοι εκτέλεσης παρατίθενται στη συνέχεια.

Για το betweeness έχουμε:

```
# betweeness
start = time.time()
betweeness_dict = nx.betweenness_centrality(G_social, k = 10)
end = time.time()
print(f'{end - start} seconds')
print(f'{(end - start)/60.} min')
```

```
590.0320670604706 seconds
9.833867784341177 min
```

Για την κεντρικότητα ιδιοδιανύσματος έχουμε:

```
# # Compute eigenvector
start = time.time()

```

```
eigenvector_dict = nx.eigenvector_centrality(G_social, max_iter=50, tol=1e-04)

end = time.time()
print(f'{end - start} seconds')
print(f'{(end - start)/60.} min')
```

```
21.40028715133667 seconds
0.35667145252227783 min
```

Ο αλγόριθμος PageRank υπολογίζεται ως εξής

```
# # Compute PageRank
start = time.time()

pagerank_dict = nx.pagerank(G_social ,tol=1e-02)

end = time.time()
print(f'{end - start} seconds')
print(f'{(end - start)/60.} min')
```

```
447.7723832130432 seconds
7.46287305355072 min
```

Σημειώνουμε ότι έχουμε μείωση την τιμή tolerance σε 10^{-2} ώστε ο αλγόριθμος να συγκλίνει (default τιμή 10^{-6}).

Η κεντρικότητα βαθμού δίνεται από τις εντολές:

```
# degree centrality
start = time.time()
degc_dict = nx.degree_centrality(G_social)
end = time.time()
print(f'{end - start} seconds')
print(f'{(end - start)/60.} min')
```

```
 2.2150933742523193 seconds
0.03691822290420532 min
```

Το μέτρο FollowerRank υπολογίζεται χρησιμοποιώντας την υλοποίηση που αναφέραμε παραπάνω:

```
#  followerRank
start = time.time()
followerRank_dict = measures.follower_rank(G_social, f1_dict, f3_dict)
end = time.time()
print(f'{end - start} seconds')
print(f'{(end - start)/60.} min')
```

```
0.16405415534973145 seconds
0.002734235922495524 min
```

Το μέτρο TFF υπολογίζεται χάρις στην αντίστοιχη συνάρτηση:

```
# TFF
start = time.time()
tff_dict = measures.tff(G_social,f1_dict, f3_dict)
end = time.time()
print(f'{end - start} seconds')
print(f'{(end - start)/60.} min')
```

```
 0.21016287803649902 seconds
0.0035027146339416506 min
```

Το μέτρο Popularity υπολογίζεται από τις εντολές:

```
# popularity
start = time.time()
pop_dict = measures.popularity(G_social, f1_dict)
end = time.time()
print(f'{end - start} seconds')
print(f'{(end - start)/60.} min')
```

```
 0.7197210788726807 seconds
0.011995351314544678 min
```

Το A-score υπολογίζεται ως εξής:

```
# A-score
start = time.time()
a_score_dict = measures.a_score(G_social, f1_dict, m4_dict, rp3_dict, rt3_dict)
end = time.time()
print(f'{end - start} seconds')
print(f'{(end - start)/60.} min')
```

```
0.8100881576538086 seconds
0.013501469294230144 min
```

Το Retweet Impact είναι:

```
# RI
start = time.time()

retweet_impact_dict = measures.retweet_impact(G_social, rt2_dict, rt3_dict)
```

```
end = time.time()
print(f'{end - start} seconds')
print(f'{(end - start)/60.} min')
```

```
 0.16015386581420898 seconds
0.0026692310969034833 min
```

To Mention Impact:

```
# MI
start = time.time()

mention_impact_dict = measures.mention_impact(G_social, m1_dict, m2_dict, m3_dict, m4_dict)

end = time.time()
print(f'{end - start} seconds')
print(f'{(end - start)/60.} min')
```

```
0.1699974536895752 seconds
0.002833290894826253 min
```

Στη συνέχεια μετατρέπουμε τις τιμές για τα μέτρα κεντρικότητας από python dictionaries σε pandas dataframe

```
df_social_centralities = pd.DataFrame()

cols = ['degc',
        'indegc',
        'betwc',
        'eigenvectorc',
        'pagerankc',
        'follower_rank',
        'tff',
        'popularity',
        'a_score',
        'retweet_imp',
        'mention_imp']

dicts = [degc_dict,
         in_degc_dict,
         betweeness_dict,
         eigenvector_dict,
         pagerank_dict,
```

```
        followerRank_dict,
        tff_dict,
        pop_dict,
        a_score_dict,
        retweet_impact_dict,
        mention_impact_dict]

df_social_centralities['nodeId'] = G_social.nodes()
for col, dicti in zip(cols, dicts):
    df_social_centralities[col] = list_of_values(dicti)
df_social_centralities
```

	nodeId	degc	indegc	betwc	eigenvectorc	pagerankc	\
1	1	0.035701	0.035653	0.079047	7.556036e-03	3.023522e-02	
2	2	0.010477	0.010308	0.001071	2.893191e-02	2.317112e-04	
3	3	0.000355	0.000300	0.000108	2.099042e-03	5.675661e-06	
4	4	0.019808	0.018928	0.013509	5.788731e-02	3.825706e-04	
5	5	0.004932	0.004805	0.000749	2.487288e-02	7.903862e-05	
...	...	...	...	...	...	...	
456622	456622	0.000004	0.000000	0.000000	4.560751e-10	3.284964e-07	
456623	456623	0.000026	0.000000	0.000000	4.560751e-10	3.284964e-07	
456624	456624	0.000002	0.000000	0.000000	4.560751e-10	3.284964e-07	
456625	456625	0.000002	0.000000	0.000000	4.560751e-10	3.284964e-07	
456626	456626	0.000002	0.000000	0.000000	4.560751e-10	3.284964e-07	

	follower_rank	tff	popularity	a_score	retweet_imp	\
1	0.998650	740.000000	1.0	16280.000000	0.000000	
2	0.983905	61.129870	1.0	4710.000000	0.000000	
3	0.845679	5.480000	1.0	137.000000	0.000000	
4	0.955556	21.500000	1.0	8750.000169	373.567266	
5	0.974245	37.827586	1.0	2199.000053	76.273292	
...	...	...	...	...	...	
456622	0.000000	0.000000	0.0	0.000000	0.000000	
456623	0.000000	0.000000	0.0	0.000000	0.000000	
456624	0.000000	0.000000	0.0	0.000000	0.000000	
456625	0.000000	0.000000	0.0	0.000000	0.000000	
456626	0.000000	0.000000	0.0	0.000000	0.000000	

```
        mention_imp
1          0.000000
2          0.000000
3          0.000000
4        479.763119
5          0.000000
...              ...
456622     0.000000
456623     0.000000
456624     0.000000
456625     0.000000
456626     0.000000

[456626 rows x 12 columns]
```

Σχεδιάζουμε τις κατανομές των μέτρων επιδραστικότητας (Σχήμα 6.3):

```
plt.close('all')
fig, axs = plt.subplots(3,3, figsize=(2* 6.4, 2.5*4.8), tight_layout = True)#,dpi = 300)
for i in axs:
    for j in i:
        j.grid(ls= ':')
axs[2][0].set_xlabel('node(i)')
axs[2][1].set_xlabel('node(i)')
axs[2][2].set_xlabel('node(i)')

ylabels = ['degree_centrality(i)',
           'betweeness(i)',
           'eigenvector_centrality(i)',
           'pageRank(i)',
           'RI(i)',
           'in_degree_centrality(i)',
           'TFF(i)',
           'MI(i)',
           'A_score(i)']
titles = ['degree centrality',
          'betweeness' ,
          'eigenvector centrality',
          'pageRank',
```

```
          'retweet impact',
          'in degree centrality',
          'tff',
          'mention impact',
          'a-score']

dfcolumns = ['degc',
             'betwc',
             'eigenvectorc',
             'pagerankc',
             'retweet_imp',
             'indegc',
             'tff',
             'mention_imp',
             'a_score']
point_size = 1.0

for ax, ylabel, title, dfcolumn in zip(axs.flatten(), ylabels, titles, dfcolumns):
    ax.set_ylabel(ylabel)
    ax.set_title(title)
    ax.plot(df_social_centralities.nodeId, df_social_centralities[dfcolumn], '-' , markersize =
    point_size)
fig.savefig('./plots/measures_all_nodes_lines.png', dpi = 300)
```

Ωστόσο, παίρνουμε πολύ λίγες πληροφορίες ως προς την επιδραστικότητα των κόμβων στη συγκεκριμένη μορφή.

6.4 Σχεδιασμός γραφήματος συναρτήσει επιδραστικότητας

Θα θέλαμε να μπορούμε να σχεδιάσουμε το σύνολο των κόμβων του γραφήματος οπτικοποιώντας την επιρροή κάθε χρήστη. Ωστόσο, κάτι τέτοιο είναι αδύνατο δεδομένου του μεγέθους του γραφήματος.

Έτσι, επιλέγουμε τους 1000 επιδραστικότερους κόμβους για κάθε μέτρο, σε μια προσπάθεια να μειώσουμε το μέγεθος του γραφήματος. Άλλωστε, το ενδιαφέρον μας επικεντρώνεται σε αυτό το υποσύνολο των πιο σημαντικών κόμβων. Λαμβάνουμε τους πιο επιδραστικούς κόμβους για κάθε μέτρο, ταξινομώντας το dataframe ως προς το εκάστοτε μέτρο και κρατώντας το σύνολο των κόμβων που έχουν την υψηλότερη τιμή:

```
# top 1000 nodes for each measure
top_follower_rank = set(df_social_centralities.sort_values(by = 'follower_rank', ascending =
    False).head(1000).nodeId)
top_tff = set(df_social_centralities.sort_values(by = 'tff', ascending = False).head(1000).nodeId
    )
top_popularity = set(df_social_centralities.sort_values(by = 'popularity', ascending = False).
```

```
    head(1000).nodeId)
top_a_score = set(df_social_centralities.sort_values(by = 'a_score', ascending = False).head
    (1000).nodeId)
top_retweet_imp = set(df_social_centralities.sort_values(by = 'retweet_imp', ascending = False).
    head(1000).nodeId)
top_mention_imp = set(df_social_centralities.sort_values(by = 'mention_imp', ascending = False).
    head(1000).nodeId)
top_mention_imp = set(df_social_centralities.sort_values(by = 'mention_imp', ascending = False).
    head(1000).nodeId)
top_pagerank = set(df_social_centralities.sort_values(by = 'pagerankc', ascending = False).head
    (1000).nodeId)
top_degc = set(df_social_centralities.sort_values(by = 'degc', ascending = False).head(1000).
    nodeId)
top_betw = set(df_social_centralities.sort_values(by = 'betwc', ascending = False).head(1000).
    nodeId)
top_eigc = set(df_social_centralities.sort_values(by = 'eigenvectorc', ascending = False).head
    (1000).nodeId)
top_indegc = set(df_social_centralities.sort_values(by = 'indegc', ascending = False).head(1000).
    nodeId)
```

Επιλέγουμε να μελετήσουμε τα εξής μέτρα:

- Κεντρικότητα βαθμού
- Κεντρικότητα betweeness
- Κεντρικότητα ιδιοδιανύσματος
- PageRank
- Retweet Impact
- in degree centrality
- TFF
- Mention Impact
- A-score

Έτσι, το σύνολο των κόμβων με τις υψηλότερες τιμές κεντρικότητας στο σύνολο των κόμβων δίνεται από την τομή των συνόλων:

```
top_1000_nodes_for_each_measure = top_degc.union(top_betw) \
                                          .union(top_eigc) \
                                          .union(top_pagerank) \
                                          .union(top_retweet_imp) \
```

```
                                        .union(top_indegc) \
                                        .union(top_tff) \
                                        .union(top_mention_imp) \
                                        .union(top_a_score)
```

Συνολικά λαμβάνουμε 3581 κόμβους:

```
len(top_1000_nodes_for_each_measure)
```

```
3581
```

Έτσι από τον γράφο `G_social`παίρνουμε τον υπογράφο που περιέχει τους 3581 κόμβους με τις υψηλότερες τιμές επιδραστικότητας:

```
top_1000_nodes = list(top_1000_nodes_for_each_measure)
G_top_1000_nodes = G_social.subgraph(top_1000_nodes)
```

Υπολογίζουμε τις θέσεις τον κόμβων στο επίπεδο για την αναπαράσταση του γραφήματος:

```
start = time.time()

nd_positions = nx.spring_layout(G_top_1000_nodes)

end = time.time()
print(f'{end - start} seconds')
print(f'{(end - start)/60.} min')
```

```
66.11467003822327 seconds
1.1019111673037212 min
```

Σκοπεύουμε να σχεδιάσουμε τους κόβμους συναρτήσει των τιμών επιδραστικότητας. Όσο πιο επιδραστικός ο κομβος τόσο μεγαλύτερο το μέγεθος του και τόσο πιο έντονο το χρώμα του στη γραφική αναπαράσταση. Έτσι, κανονικοποιούμε τις τιμές των μέτρων επιδραστικότητας για το σύνολο των πιο επιδραστικών κόμβων χρησιμοποιώντας δύο συναρτήσεις που περιλαμβάνονται στο `utilities.py`:

```
def normalize_vector(v):
    """
        Normalize vector

        normalized_v = v / ||v||

        where ||v|| : 2-norm of vector v
    """
    return v/np.linalg.norm(v)

```

```
def factor_vector(v, factor):
    """
        Multiply scalar with vector:    a * v
    """
    return factor * np.array(v)
```

Έτσι κανονικοποιούμε τις τιμές των μέτρων:

```
scaling_factor = 1000.0

top_1000_tff_normalized = [tff_dict[node] for node in list(top_1000_nodes)]
top_1000_tff_normalized= normalize_vector(top_1000_tff_normalized)
top_1000_tff_normalized = factor_vector(top_1000_tff_normalized, scaling_factor)

top_1000_a_score_normalized = [a_score_dict[node] for node in list(top_1000_nodes)]
top_1000_a_score_normalized= normalize_vector(top_1000_a_score_normalized)
top_1000_a_score_normalized = factor_vector(top_1000_a_score_normalized, scaling_factor)

top_1000_retweet_impact_normalized = [retweet_impact_dict[node] for node in list(top_1000_nodes)]
top_1000_retweet_impact_normalized= normalize_vector(top_1000_retweet_impact_normalized)
top_1000_retweet_impact_normalized = factor_vector(top_1000_retweet_impact_normalized,
    scaling_factor)

top_1000_mention_impact_normalized = [mention_impact_dict[node] for node in list(top_1000_nodes)]
top_1000_mention_impact_normalized= normalize_vector(top_1000_mention_impact_normalized)
top_1000_mention_impact_normalized = factor_vector(top_1000_mention_impact_normalized,
    scaling_factor)

top_1000_pagerank_normalized = [pagerank_dict[node] for node in list(top_1000_nodes)]
top_1000_pagerank_normalized= normalize_vector(top_1000_pagerank_normalized)
top_1000_pagerank_normalized = factor_vector(top_1000_pagerank_normalized, scaling_factor)

top_1000_degc_normalized = [degc_dict[node] for node in list(top_1000_nodes)]
top_1000_degc_normalized= normalize_vector(top_1000_degc_normalized)
top_1000_degc_normalized = factor_vector(top_1000_degc_normalized, scaling_factor)

top_1000_betwc_normalized = [betweeness_dict[node] for node in list(top_1000_nodes)]
top_1000_betwc_normalized= normalize_vector(top_1000_betwc_normalized)
top_1000_betwc_normalized = factor_vector(top_1000_betwc_normalized, scaling_factor)

top_1000_eigc_normalized = [eigenvector_dict[node] for node in list(top_1000_nodes)]
top_1000_eigc_normalized= normalize_vector(top_1000_eigc_normalized)
top_1000_eigc_normalized = factor_vector(top_1000_eigc_normalized, scaling_factor)

top_1000_indegc_normalized = [in_degc_dict[node] for node in list(top_1000_nodes)]
```

```
top_1000_indegc_normalized= normalize_vector(top_1000_indegc_normalized)
top_1000_indegc_normalized = factor_vector(top_1000_indegc_normalized, scaling_factor)
```

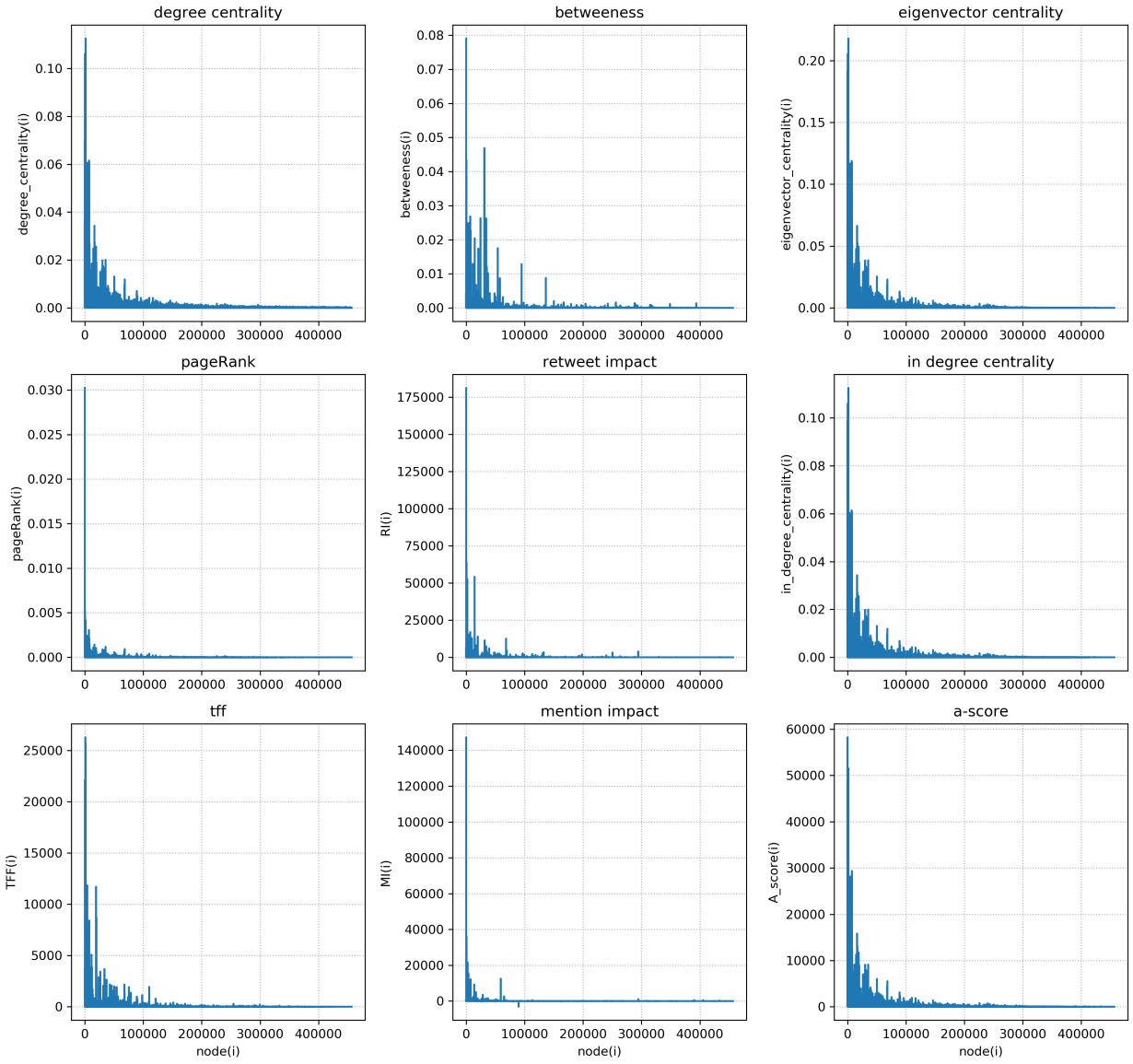
και σχεδιάζουμε το γράφημα των 3581 πιο επιδραστικών κόμβων όπως φαίνεται στο Σχήμα 6.4 σύμφωνα με τις εντολές:

```
plt.close('all')
fig, axs = plt.subplots(3,3, figsize=(2* 6.4, 2.5*4.8))#,dpi = 300)

fig.patch.set_facecolor('silver')

titles = ['degree centrality',
          'betweeness',
          'eigenvector centrality',
          'pageRank',
          'retweet impact',
          'in degree centrality',
          'tff',
          'mention impact',
          'a-score']

data = [top_1000_degc_normalized,
        top_1000_betwc_normalized,
        top_1000_eigc_normalized,
        top_1000_pagerank_normalized,
        top_1000_retweet_impact_normalized,
        top_1000_indegc_normalized,
        top_1000_tff_normalized,
        top_1000_mention_impact_normalized,
        top_1000_a_score_normalized]

pos = nd_positions
alpha_factor  = 1.0
cmap_all = plt.cm.Reds #BuGn#rainbow#

for ax, titlei, datai in zip(axs.flatten(), titles, data):
    nx.draw_networkx_nodes(
        G_social,
        pos,
        nodelist = list(top_1000_nodes),
        node_size = datai,
        node_color = datai,
        cmap= cmap_all,
        alpha = alpha_factor,
        ax = ax
```

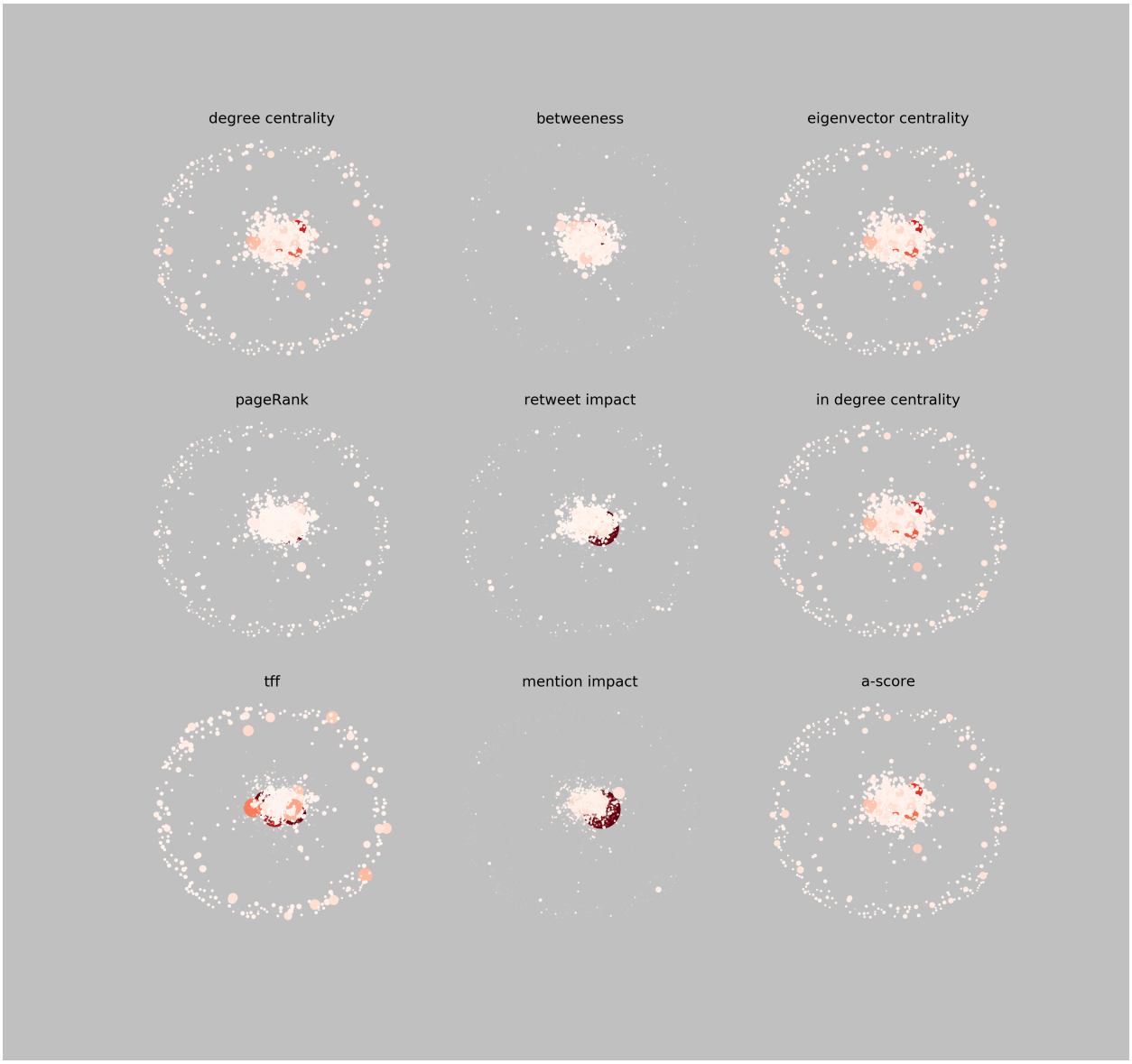
```
    )
    ax.axis('off')
    ax.set_title(titlei)
fig.savefig('./plots/measures_graph.png', dpi = 300,facecolor=fig.get_facecolor())
```

Παρατηρούμε ότι τα μέτρα TFF, $a-score$ αλλά και in degree centrality (και άρα followerRank) παρουσιάζουν παρόμοια συμπεριφορά. Αυτό μπορεί να επιβεβαιωθεί διαισθητικά από τον ορισμό κάθε μέτρου αφού περιλαμβάνει τις μετρικές $F1$, $F3$ δηλαδή τους βαθμούς κάθε κόμβου.

Ωστόσο, για μια πιο ολοκληρωμένη εικόνα απαιτείται ανάλυση των συσχετίσεων μεταξύ των μέτρων.



Σχήμα 6.3: Διάγραμμα τιμών μέτρων επιδραστικότητας για κάθε κόμβο.



Σχήμα 6.4: Γράφημα των 3581 πιο επιδραστικών κόμβων του δικτύου. Οι μεγαλύτεροι και πιο έντονα χρωματισμένοι κόμβοι αντιστοιχούν σε μεγαλύτερες τιμές επιδραστικότητας σύμφωνα με το εκάστοτε μέτρο.

6.5 Μελέτες συσχέτισης μεταξύ των μέτρων επιδραστικότητας

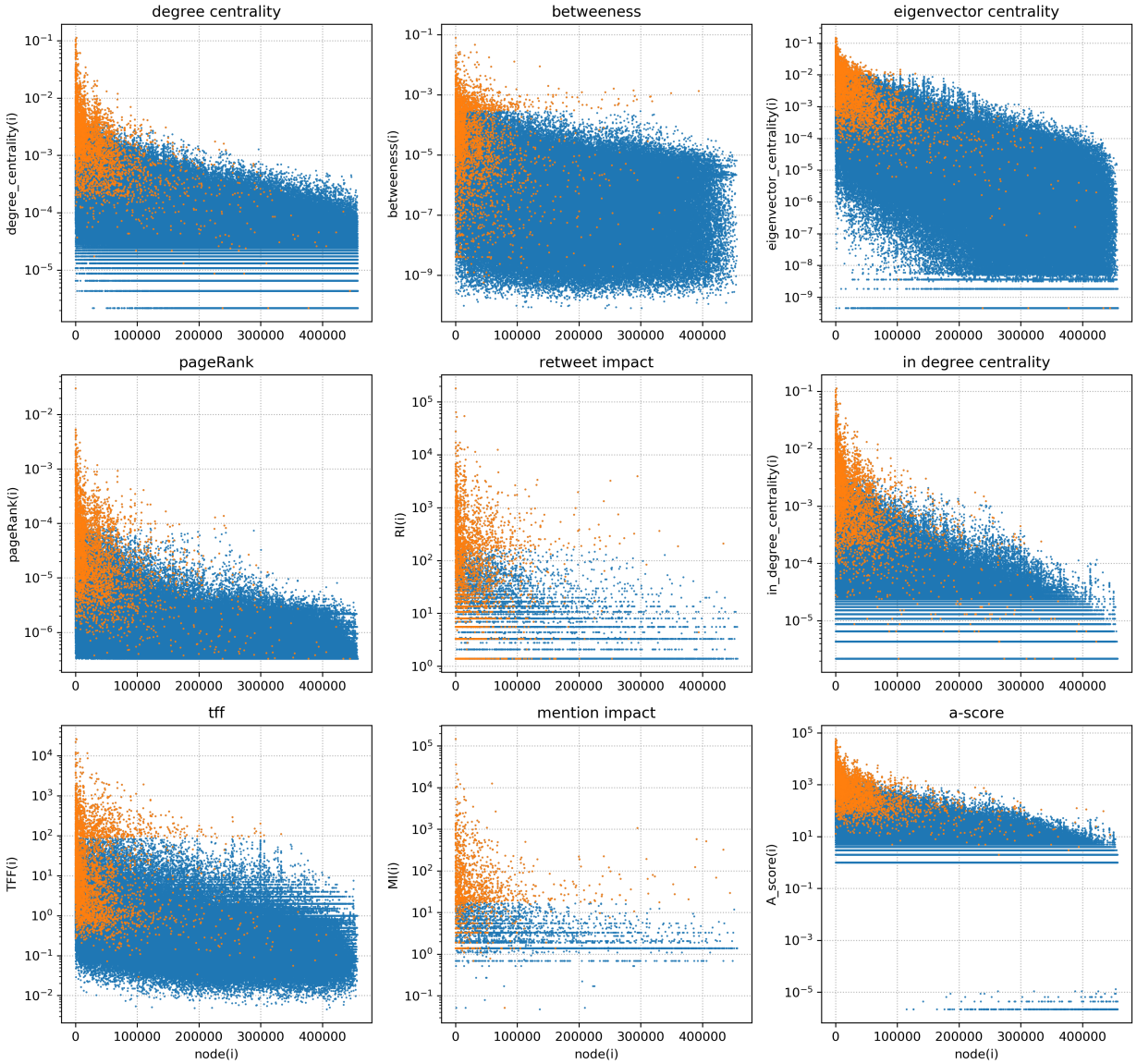
Όμοια με παραπάνω σχεδιάζουμε τις τιμές της επιδραστικότητας για κάθε κόμβο (Σχήμα 6.5) εκτελώντας τις εντολές:

```
plt.close('all')
fig, axs = plt.subplots(3,3, figsize=(2* 6.4, 2.5*4.8), tight_layout = True)#,dpi = 300)
for i in axs:
    for j in i:
        j.grid(ls= ':')
axs[2][0].set_xlabel('node(i)')
axs[2][1].set_xlabel('node(i)')
axs[2][2].set_xlabel('node(i)')

ylabels = ['degree_centrality(i)',
           'betweeness(i)',
           'eigenvector_centrality(i)',
           'pageRank(i)',
           'RI(i)',
           'in_degree_centrality(i)',
           'TFF(i)',
           'MI(i)',
           'A_score(i)']
titles = ['degree centrality',
          'betweeness' ,
          'eigenvector centrality',
          'pageRank',
          'retweet impact',
          'in degree centrality',
          'tff',
          'mention impact',
          'a-score']

dfcolumns = ['degc',
             'betwc',
             'eigenvectorc',
             'pagerankc',
             'retweet_imp',
             'indegc',
             'tff',
             'mention_imp',
             'a_score']
point_size = 1.0

```

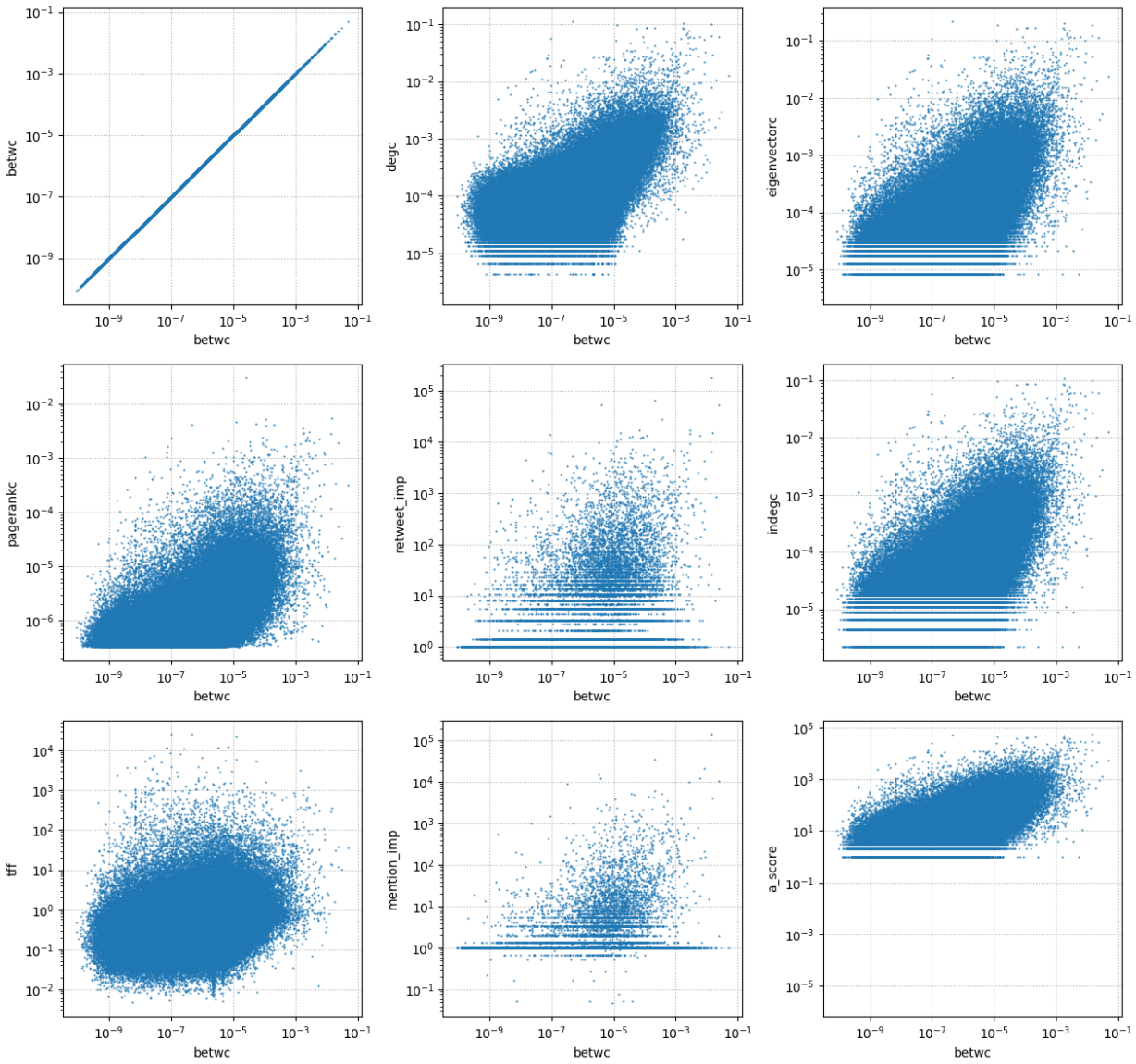
```
for ax, ylabel, title, dfcolumn in zip(axs.flatten(), ylabels, titles, dfcolumns):
    ax.set_ylabel(ylabel)
    ax.set_title(title)
    ax.semilogy(df_social_centralities.nodeId, df_social_centralities[dfcolumn], '.' , markersize = point_size)
axs[0][0].semilogy(top_1000_nodes, [degc_dict[node] for node in list(top_1000_nodes)], '.' , markersize = point_size)
axs[0][1].semilogy(top_1000_nodes, [betweeness_dict[node] for node in list(top_1000_nodes)], '.', markersize = point_size)
axs[0][2].semilogy(top_1000_nodes, [eigenvector_dict[node] for node in list(top_1000_nodes)], '.' , markersize = point_size)
axs[1][0].semilogy(top_1000_nodes, [pagerank_dict[node] for node in list(top_1000_nodes)], '.', markersize = point_size)
axs[1][1].semilogy(top_1000_nodes, [retweet_impact_dict[node] for node in list(top_1000_nodes)], '.', markersize = point_size)
axs[1][2].semilogy(top_1000_nodes, [in_degc_dict[node] for node in list(top_1000_nodes)], '.', markersize = point_size)
axs[2][0].semilogy(top_1000_nodes, [tff_dict[node] for node in list(top_1000_nodes)], '.', markersize = point_size)
axs[2][1].semilogy(top_1000_nodes, [mention_impact_dict[node] for node in list(top_1000_nodes)], '.', markersize = point_size)
axs[2][2].semilogy(top_1000_nodes, [a_score_dict[node] for node in list(top_1000_nodes)], '.', markersize = point_size)

fig.savefig('./plots/measures_all_nodes_semilogy_and_top_nodes.png', dpi = 300)
```

Ωστόσο με ένα τέτοιο γράφημα δύσκολα μπορούμε να αποφανθούμε για τυχόν συσχετίσεις μεταξύ των διαφορετικών μέτρων.

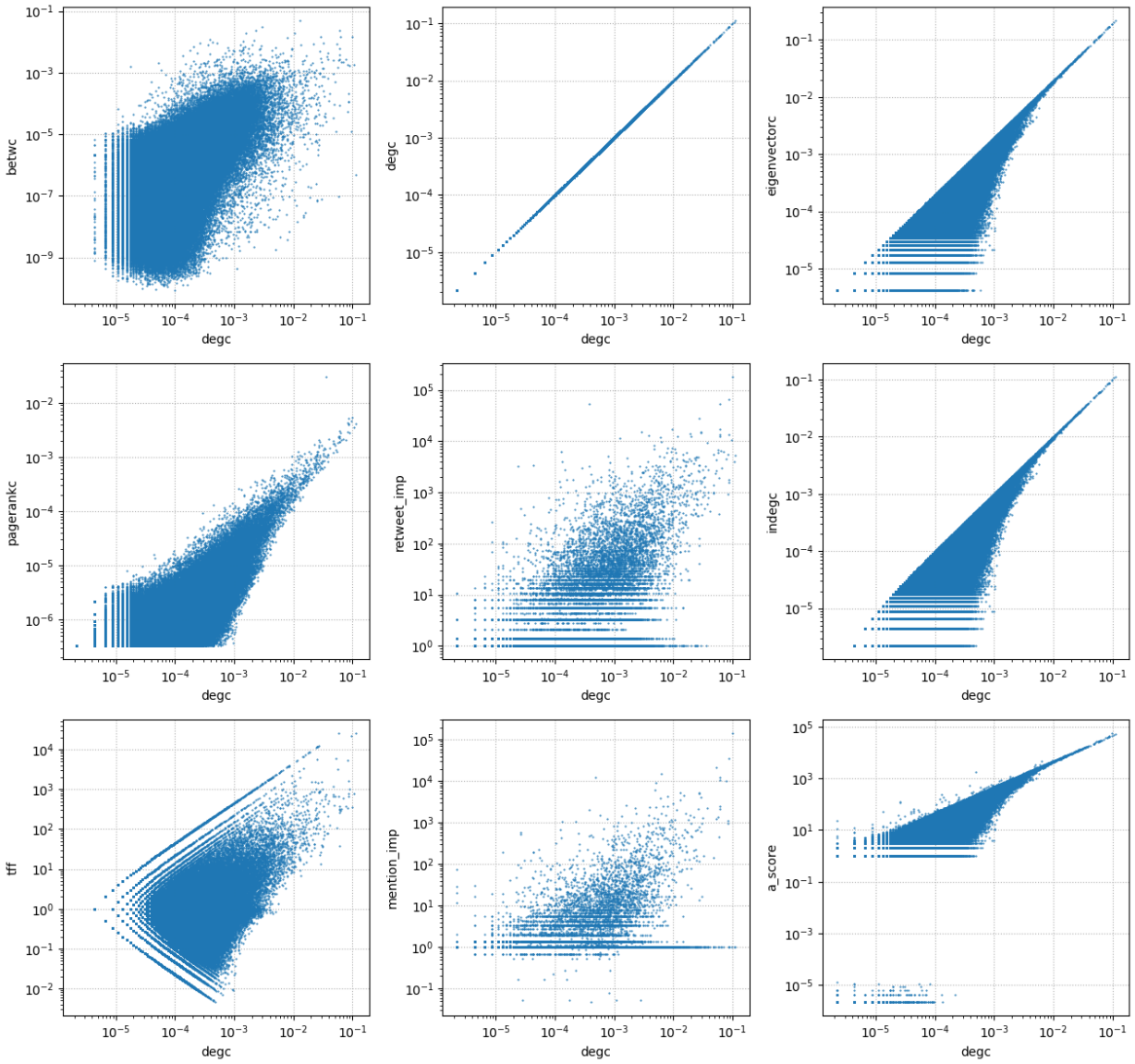
Για μια πρώτη ποιοτική ανάλυση σχεδιάζουμε κάθε μέτρο συναρτήσει των υπολοίπων όπως φαίνεται στα παρακάτω Σχήματα 6.6, 6.7, 6.8, 6.9, 6.10, 6.11, 6.12, 6.13.



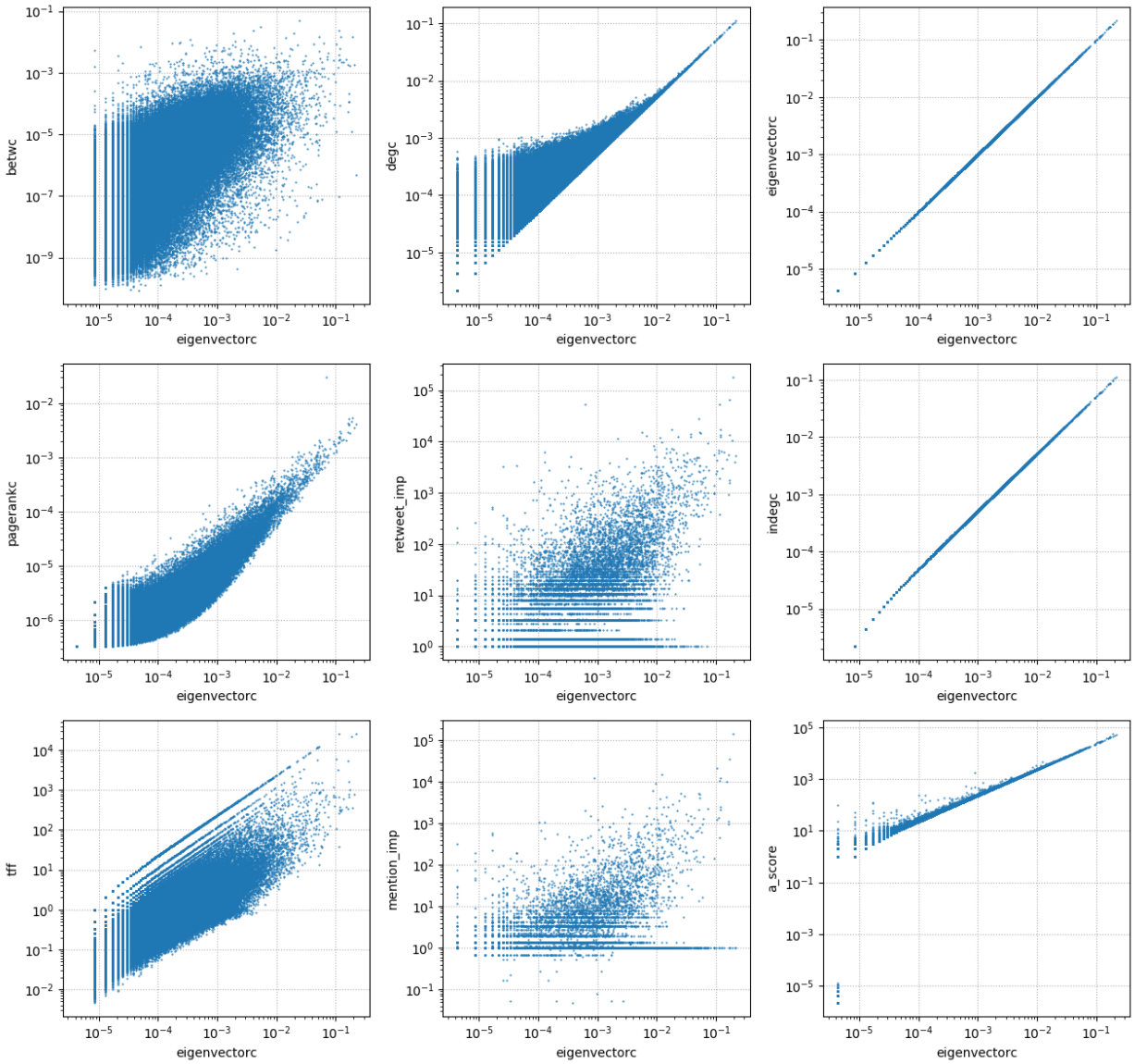
Σχήμα 6.5: Μέτρα επιδραστικότητας συναρτήσει του κόμβου σε ημιλογαριθμική κλίμακα. Με πορτοκαλί χρώμα παρουσιάζεται το σύνολο των 3581 σημαντικότερων κόμβων και με μπλε όλοι οι υπόλοιποι κόμβοι.



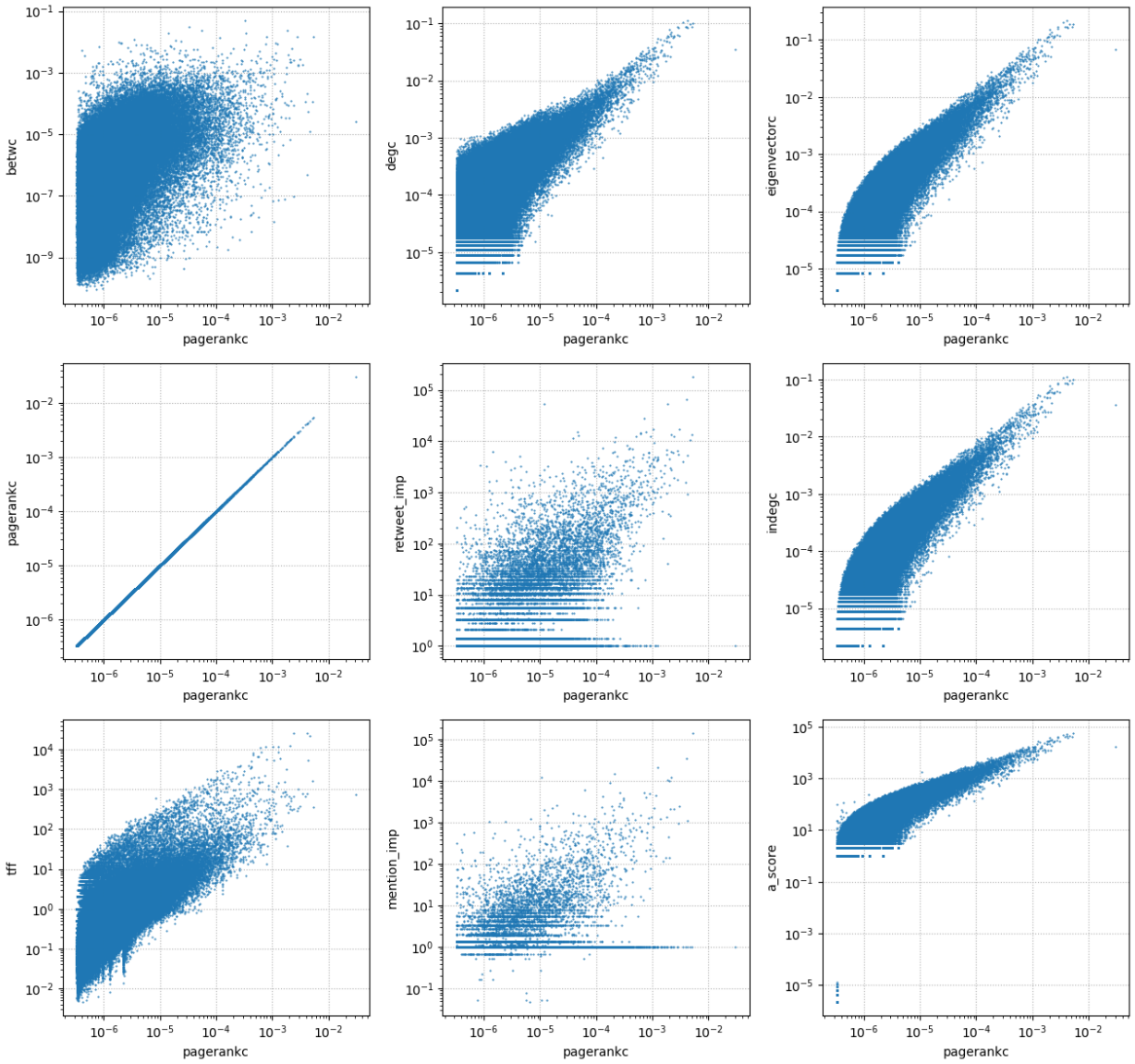
Σχήμα 6.6: Κεντρικότητα betweeness συναρτήσει υπολοίπων μέτρων επιδραστικότητας για τους 3581 πιο επιδραστικούς κόμβους του δικτύου σε λογαριθμική κλίμακα.



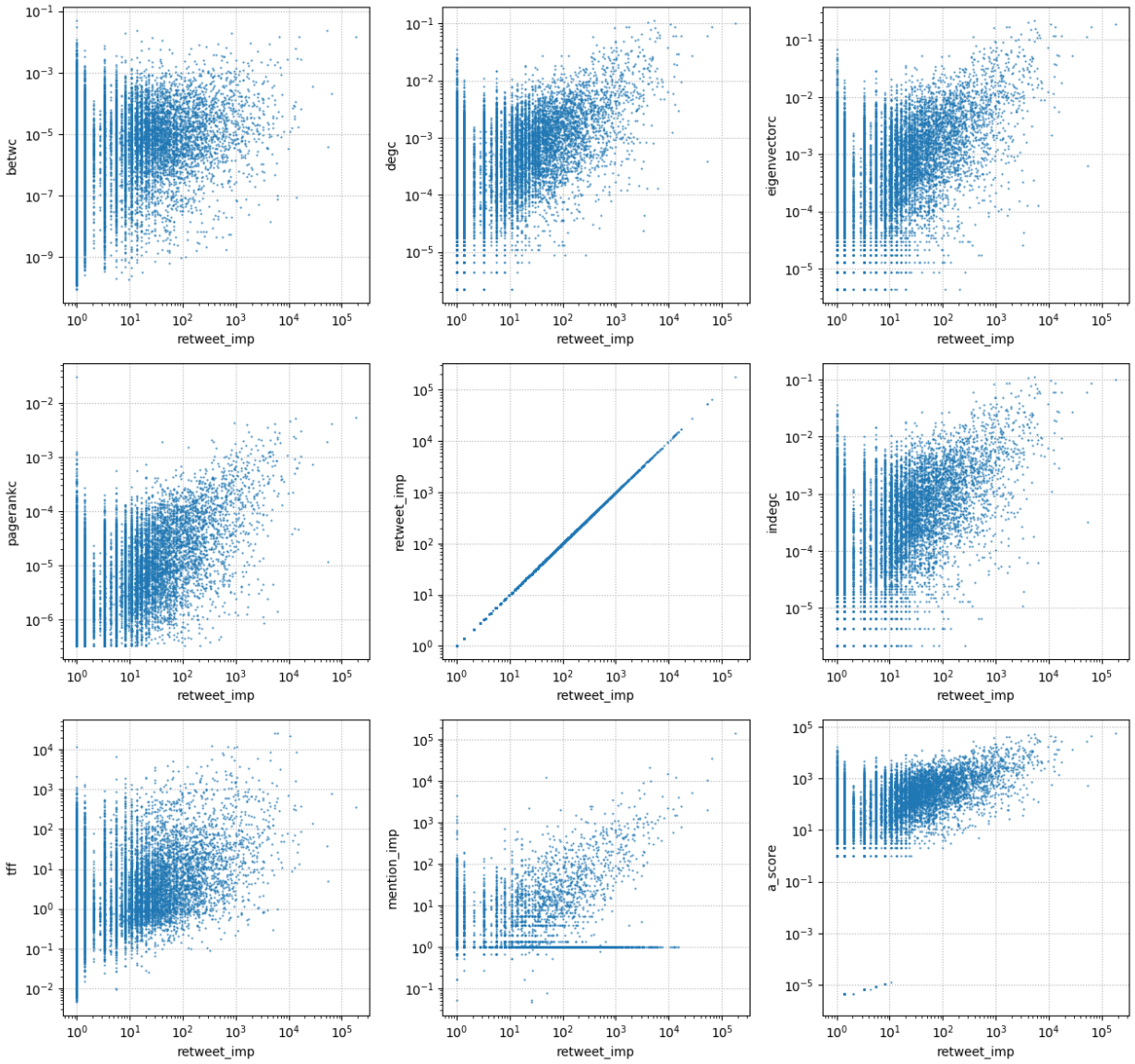
Σχήμα 6.7: Κεντρικότητα βαθμού συναρτήσει υπολοίπων μέτρων επιδραστικότητας για τους 3581 πιο επιδραστικούς κόμβους του δικτύου σε λογαριθμική κλίμακα.



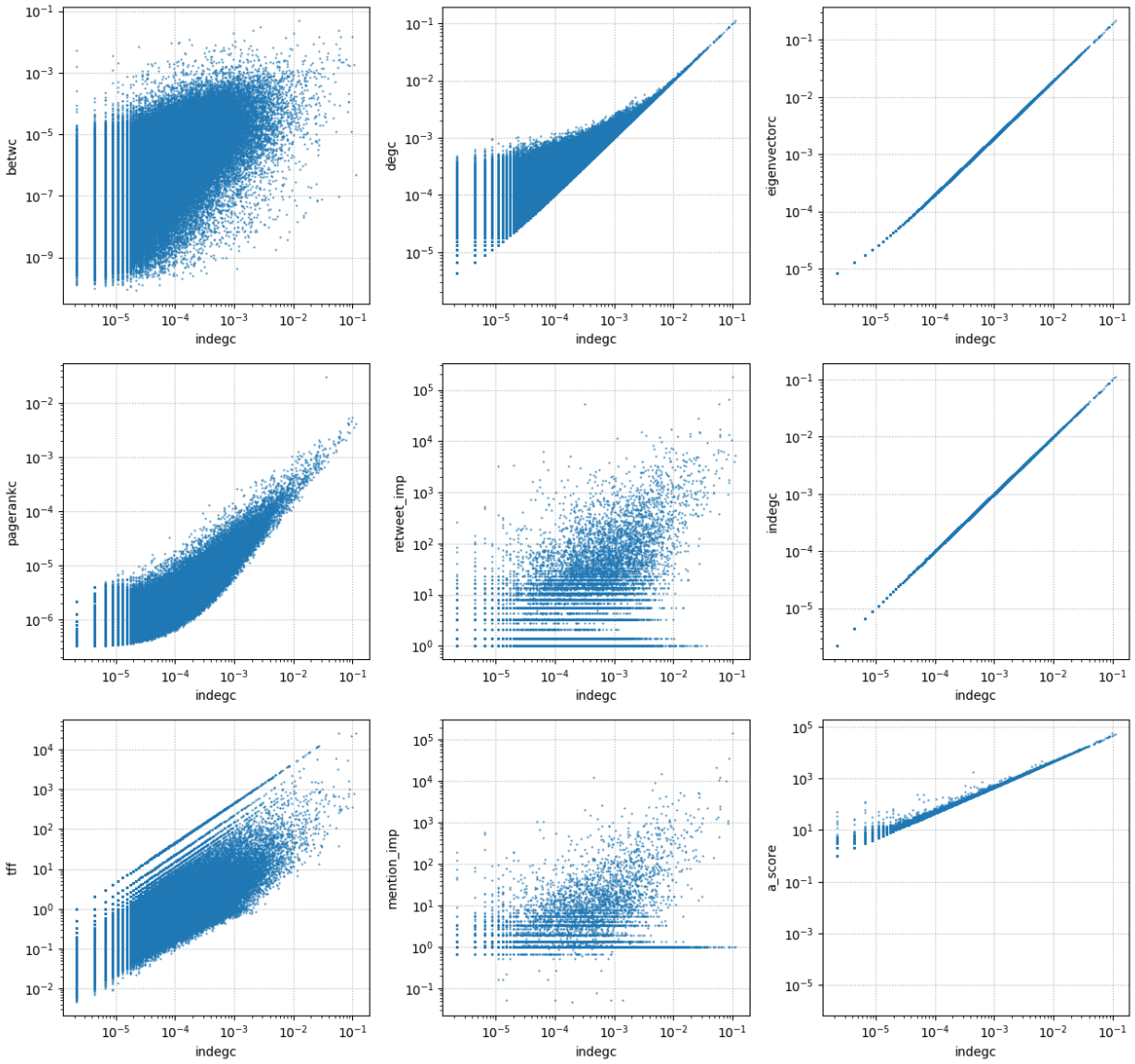
Σχήμα 6.8: Κεντρικότητα ιδιοδιανύσματος συναρτήσει υπολοίπων μέτρων επιδραστικότητας για τους 3581 πιο επιδραστικούς κόμβους του δικτύου σε λογαριθμική κλίμακα.



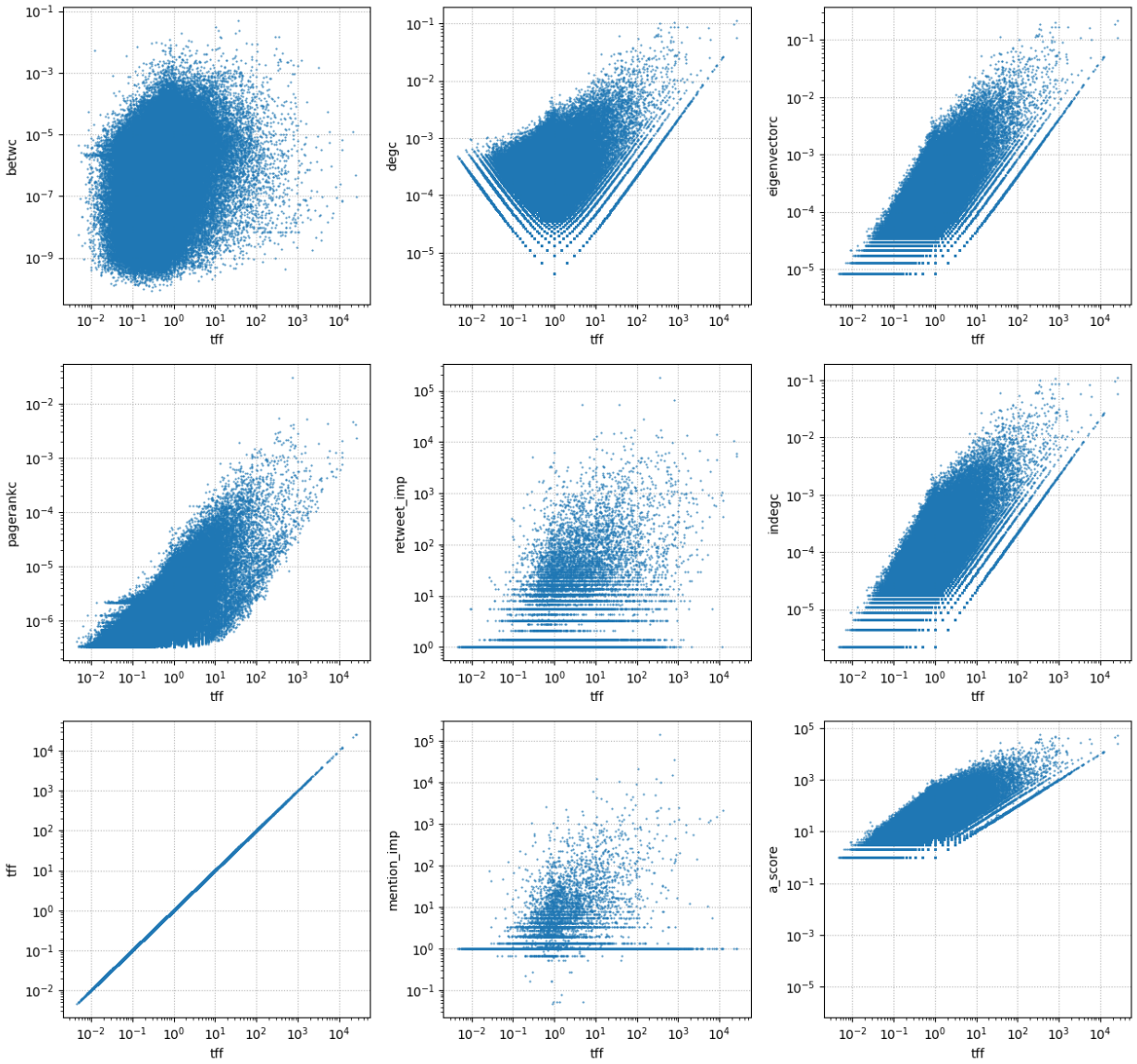
Σχήμα 6.9: Κεντρικότητα PageRank συναρτήσει υπολοίπων μέτρων επιδραστικότητας για τους 3581 πιο επιδραστικούς κόμβους του δικτύου σε λογαριθμική κλίμακα.



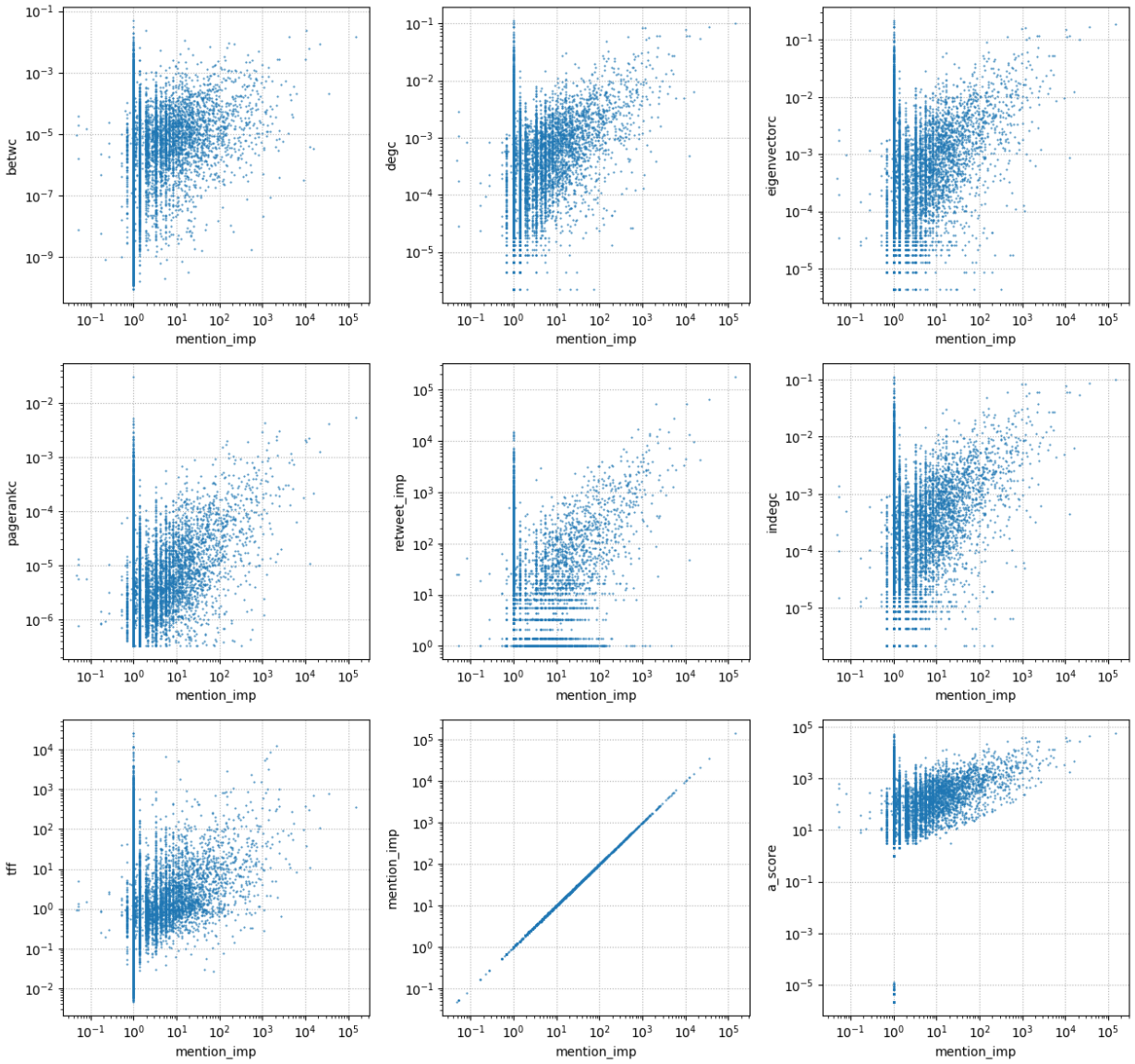
Σχήμα 6.10: Κεντρικότητα Retweet Impact συναρτήσει υπολοίπων μέτρων επιδραστικότητας για τους 3581 πιο επιδραστικούς κόμβους του δικτύου σε λογαριθμική κλίμακα.



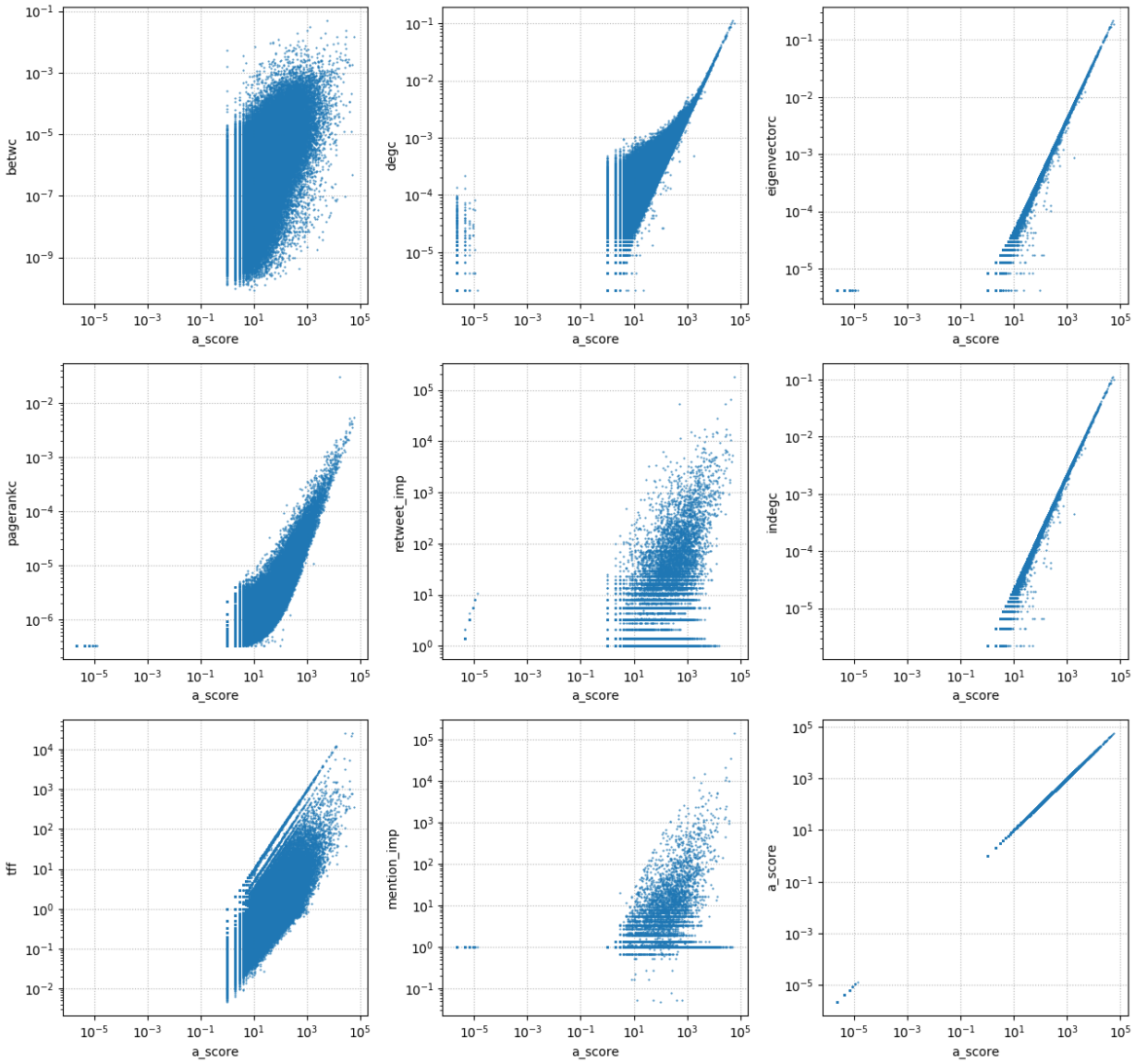
Σχήμα 6.11: Κεντρικότητα in degree συναρτήσει υπολοίπων μέτρων επιδραστικότητας για τους 3581 πιο επιδραστικούς κόμβους του δικτύου σε λογαριθμική κλίμακα.



Σχήμα 6.12: Μέτρο επιδραστικότητας TFF συναρτήσει υπολοίπων μέτρων επιδραστικότητας για τους 3581 πιο επιδραστικούς κόμβους του δικτύου σε λογαριθμική κλίμακα.



Σχήμα 6.13: Μέτρο επιδραστικότητας Mention Impact συναρτήσει υπολοίπων μέτρων επιδραστικότητας για τους 3581 πιο επιδραστικούς κόμβους του δικτύου σε λογαριθμική κλίμακα.



Σχήμα 6.14: Μέτρο επιδραστικότητας A-score συναρτήσει υπολοίπων μέτρων επιδραστικότητας για τους 3581 πιο επιδραστικούς κόμβους του δικτύου σε λογαριθμική κλίμακα.

Αρχικά παρατηρούμε ότι τα Retweet Impact και Mention Impact δε φαίνεται να συσχετίζονται ισχυρά με κάποιο από τα υπόλοιπα μέτρα.

Επίσης, ήδη από αυτή την ποιοτική ανάλυση είναι προφανής η ισχυρή συσχέτιση μεταξύ αρκετών από τα υπόλοιπα μέτρα. Το **betweeness** φαίνεται να παρουσιάζει υψηλή συσχέτιση με όλα τα υπόλοιπα μέτρα. Επιπλέον η **κεντρικότητα βαθμού** φαίνεται να παρουσιάζει ισχυρή συσχέτιση με την πλειοψηφία των μέτρων, ειδκά το A-score και το in degree. Επίσης κάποια συσχέτιση φαίνεται έχει και με τα μέτρα pagerank και ιδιοδιανύσματος. Επίσης **κεντρικότητα ιδιοδιανύσματος** φαίνεται να παρουσιάζει κάποια εξάρτηση από τα pagerank, A-score και κεντρικότητα βαθμού. Ακόμα το **pagerank** φαίνεται να συσχετίζεται με τα in degree, TFFκαι A-score. Επιπλέον το **in degree** φαίνεται να εξαρτάται από το TFF ενώ φαίνεται να παρουσιάζει υψηλή συσχέτιση με το A-score. Το TFF φαίνεται να παρουσιάζει υψηλή συσχέτιση με το A-score.

Για μια πιο λεπτομερή ανάλυση καταφεύγουμε στον υπολογισμό συντελεστών συσχέτισης για τα μέτρα επιδραστικότητας.

Έτσι έχουμε τις εξής εντολές:

```
# correlation matrix
corr_matr = df_social_centralities.drop(columns = 'nodeId').corr('pearson')
corr_matr
```

```
degc            1.000000  0.990724  0.257425      0.990724   0.561150
indegc          0.990724  1.000000  0.246287      1.000000   0.573312
betwc           0.257425  0.246287  1.000000      0.246287   0.114189
eigenvectorc    0.990724  1.000000  0.246287      1.000000   0.573312
pagerankc       0.561150  0.573312  0.114189      0.573312   1.000000
follower_rank   0.206692  0.193198  0.057511      0.193198   0.081174
tff             0.421347  0.435057  0.009230      0.435057   0.264453
popularity      0.105945  0.062873  0.027160      0.062873   0.022987
a_score         0.988801  0.998348  0.252330      0.998348   0.573846
retweet_imp     0.393224  0.401973  0.222105      0.401973   0.260098
mention_imp     0.283240  0.290364  0.184959      0.290364   0.196409

                follower_rank       tff  popularity   a_score  retweet_imp  \
degc                 0.206692  0.421347    0.105945  0.988801     0.393224
indegc               0.193198  0.435057    0.062873  0.998348     0.401973
betwc                0.057511  0.009230    0.027160  0.252330     0.222105
eigenvectorc         0.193198  0.435057    0.062873  0.998348     0.401973
pagerankc            0.081174  0.264453    0.022987  0.573846     0.260098
follower_rank        1.000000  0.067651    0.611903  0.190354     0.032787
```

```
tff                0.067651  1.000000    0.015529  0.426940     0.088661
popularity         0.611903  0.015529    1.000000  0.061853     0.007472
a_score            0.190354  0.426940    0.061853  1.000000     0.447582
retweet_imp        0.032787  0.088661    0.007472  0.447582     1.000000
mention_imp        0.014357  0.019867    0.002941  0.343474     0.889266
```

```
               mention_imp
degc              0.283240
indegc            0.290364
betwc             0.184959
eigenvectorc      0.290364
pagerankc         0.196409
follower_rank     0.014357
tff               0.019867
popularity        0.002941
a_score           0.343474
retweet_imp       0.889266
mention_imp       1.000000
```

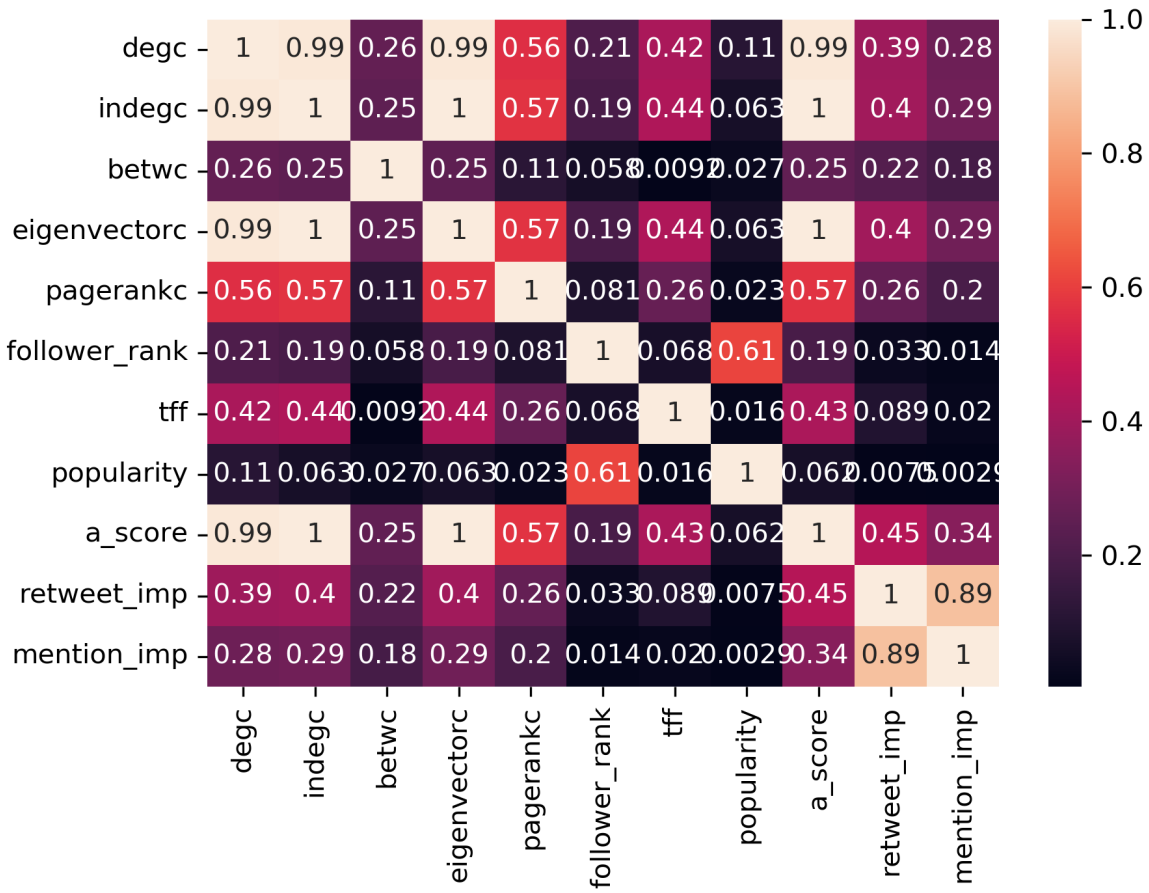
Σχεδιάζουμε έναν πίνακα συσχέτισης που οπτικοποιεί καλύτερα τα αποτελέσματα (Σχήμα 6.15)

```
import seaborn as sns
sns.heatmap(corr_matr, annot = True)
```

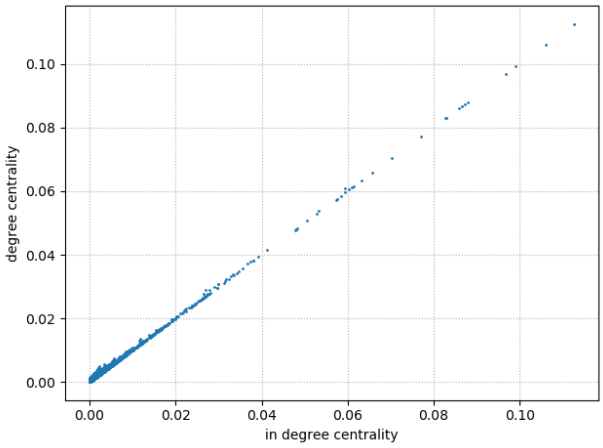
Όπως βλέπουμε στο Σχήμα 6.15 υψηλή συχέτιση παρουσιάζουν τα εξής ζεύγη μέτρων:

- Κεντρικότητα βαθμού - in degree centrality
- Κεντρικότητα βαθμού - κεντρικότητα ιδιοδιανύσματος
- Κεντρικότητα βαθμού - $Ascore$
- in degree centrality- κεντρικότητα ιδιοδιανύσματος
- in degree centrality- $Ascore$
- Κεντρικότητα ιδιοδιανύσματος - $Ascore$
- Retweet Impact - Mention Impact

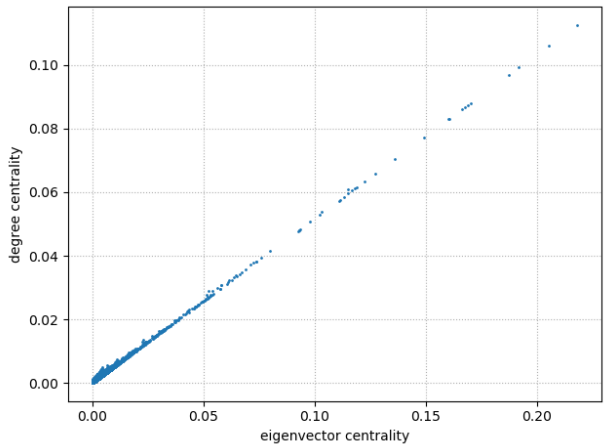
Σχετικά διαγράμματα μπορούν να βρεθούν στα Σχήματα 6.6, 6.7, 6.8, 6.9, 6.10, 6.11, 6.12, 6.13. Ωστόσο επανασχεδιάζουμε τα διαγράμματα εδώ για λόγους απλότητας και καλύτερης εποπτείας (Σχήματα 6.16, 6.17, 6.18, 6.19, 6.20, 6.21, 6.22).



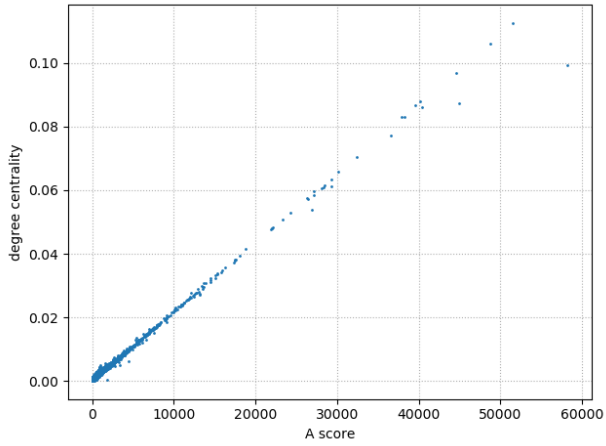
Σχήμα 6.15: Συντελεστές συσχέτισης Pearson για πολλά μέτρα επιδραστικότητας. Αναδεικνύονται κάποιες ισχυρές συσχετίσεις οι οποίες ωστόσο δεν είναι γραμμικές.



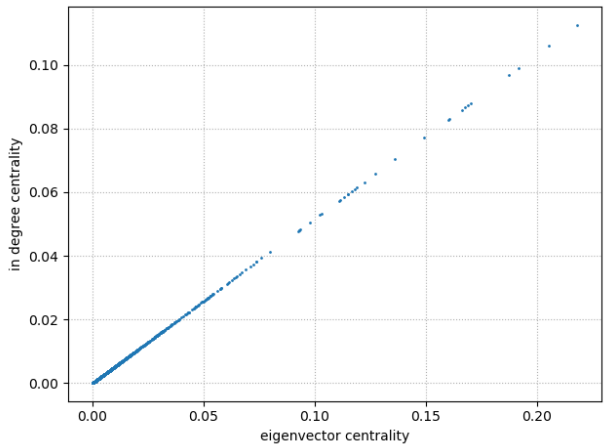
Σχήμα 6.16: Διάγραμμα κεντρικότητας βαθμού συναρτήσει in degree centrality. Παρατηρούμε μία ισχυρή γραμμική συσχέτιση.



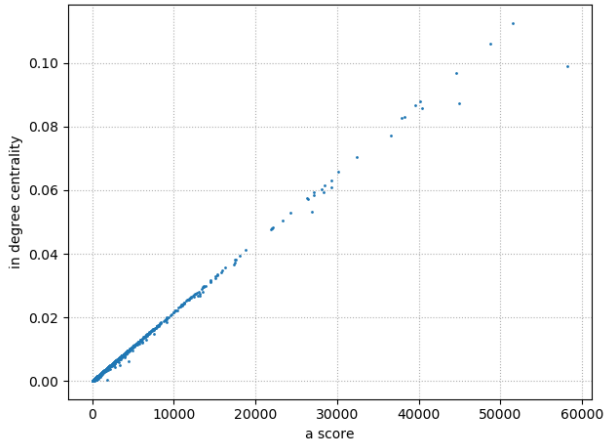
Σχήμα 6.17: Διάγραμμα κεντρικότητας βαθμού συναρτήσει κεντρικότητας ιδιοδιανύσματος. Παρατηρούμε μία ισχυρή γραμμική συσχέτιση.



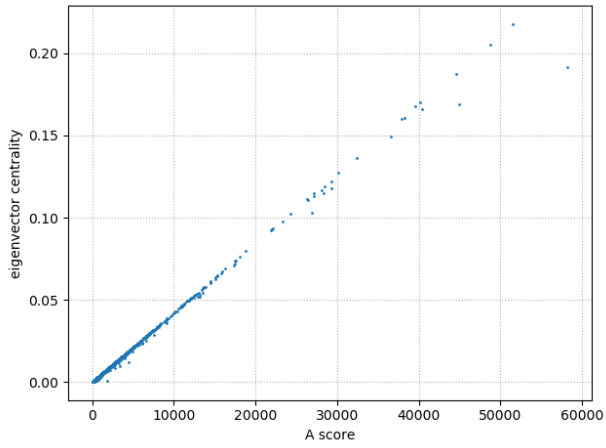
Σχήμα 6.18: Διάγραμμα κεντρικότητας βαθμού συναρτήσει $Ascore$. Παρατηρούμε μία ισχυρή γραμμική συσχέτιση.



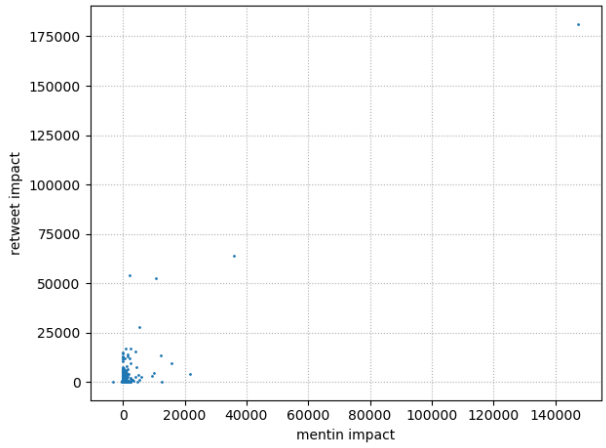
Σχήμα 6.19: Διάγραμμα in degree centrality συναρτήσει κεντρικότητας ιδιοδιανύσματος. Παρατηρούμε μία ισχυρή γραμμική συσχέτιση.



Σχήμα 6.20: Διάγραμμα in degree centrality συναρτήσει $Ascore$. Παρατηρούμε μία ισχυρή γραμμική συσχέτιση.



Σχήμα 6.21: Διάγραμμα κεντρικότητας ιδιοδιανύσματος συναρτήσει $Ascore$. Παρατηρούμε γραμμική συσχέτιση.



Σχήμα 6.22: Διάγραμμα retweet impact συναρτήσει mention impat. Παρατηρούμε γραμμική συσχέτιση.

Αξίζει να σημειωθεί ότι ο συντελεστής Pearson μας πληροφορεί για γραμμική συσχέτιση των μεταβλητών. Ωστόσο, πολλά μέτρα εξαρτώνται από άλλα μέτρα με μη γραμμικό τρόπο. Έτσι, εξετάζουμε τη συσχέτιση των μέτρων επιδραστικότητας με τον συντελεστή Spearman, ο οποίος υποδεικνύει μη γραμμικές συσχετίσεις. Έτσι, έχουμε:

```
corr_matr = df_social_centralities.drop(columns = 'nodeId').corr('spearman')
```

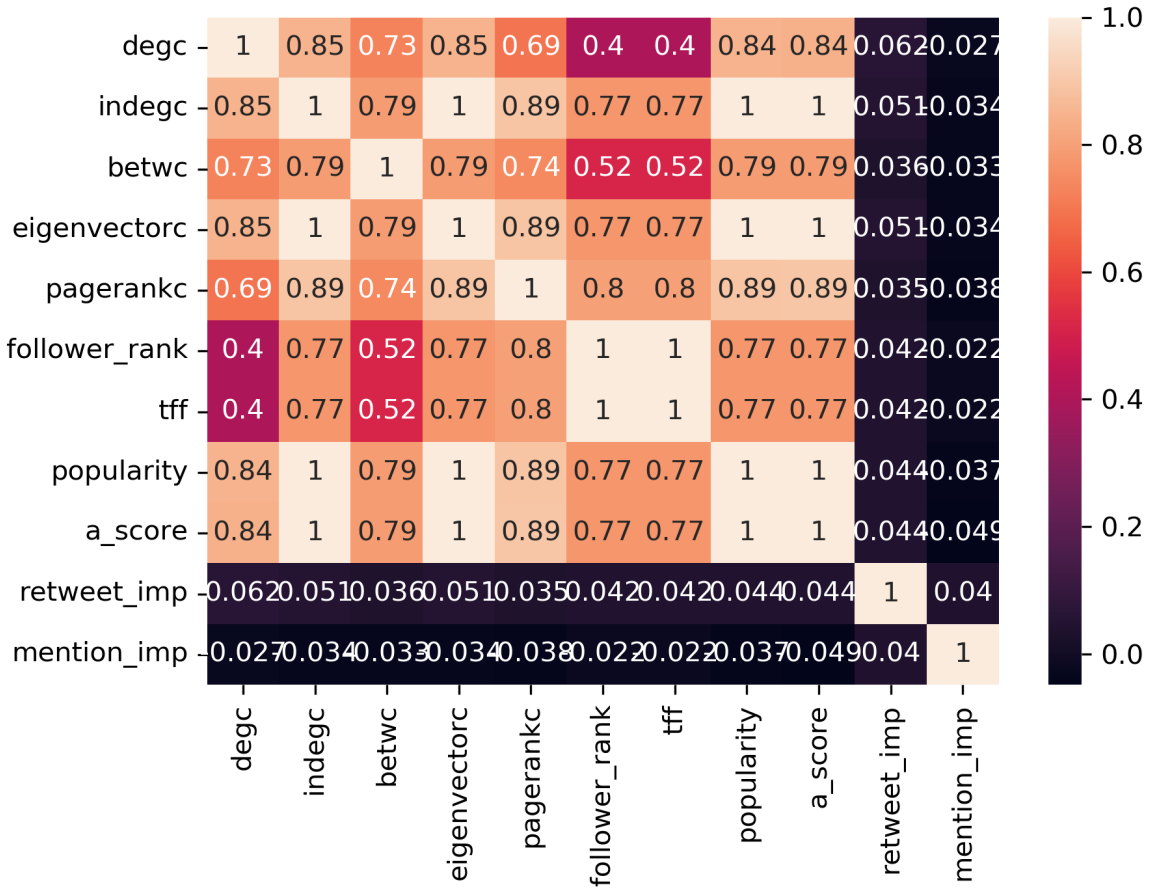
	degc	indegc	betwc	eigenvectorc	pagerankc	\
degc	1.000000	0.846516	0.727599	0.846516	0.694877	
indegc	0.846516	1.000000	0.789526	1.000000	0.892805	
betwc	0.727599	0.789526	1.000000	0.789526	0.743894	
eigenvectorc	0.846516	1.000000	0.789526	1.000000	0.892805	
pagerankc	0.694877	0.892805	0.743894	0.892805	1.000000	
follower_rank	0.399867	0.773629	0.515533	0.773629	0.796472	
tff	0.399867	0.773629	0.515533	0.773629	0.796472	
popularity	0.844918	0.999035	0.788127	0.999035	0.891032	
a_score	0.843286	0.997738	0.788136	0.997738	0.894504	
retweet_imp	0.062391	0.051111	0.035974	0.051111	0.035227	
mention_imp	-0.026938	-0.034334	-0.033455	-0.034334	-0.038070	

	follower_rank	tff	popularity	a_score	retweet_imp	\
degc	0.399867	0.399867	0.844918	0.843286	0.062391	
indegc	0.773629	0.773629	0.999035	0.997738	0.051111	
betwc	0.515533	0.515533	0.788127	0.788136	0.035974	
eigenvectorc	0.773629	0.773629	0.999035	0.997738	0.051111	
pagerankc	0.796472	0.796472	0.891032	0.894504	0.035227	
follower_rank	1.000000	1.000000	0.770951	0.773894	0.041702	
tff	1.000000	1.000000	0.770951	0.773894	0.041702	
popularity	0.770951	0.770951	1.000000	0.996770	0.044389	
a_score	0.773894	0.773894	0.996770	1.000000	0.043519	
retweet_imp	0.041702	0.041702	0.044389	0.043519	1.000000	
mention_imp	-0.021584	-0.021584	-0.036710	-0.048693	0.039804	

	mention_imp
degc	-0.026938
indegc	-0.034334

```
betwc            -0.033455
eigenvectorc     -0.034334
pagerankc        -0.038070
follower_rank    -0.021584
tff              -0.021584
popularity       -0.036710
a_score          -0.048693
retweet_imp       0.039804
mention_imp       1.000000
```

Για καλύτερη εποπτεία και λόγους απλότητας σχεδιάζουμε το διάγραμμα των συντελεστών συσχέτισης Spearman(Σχήμα 6.23).



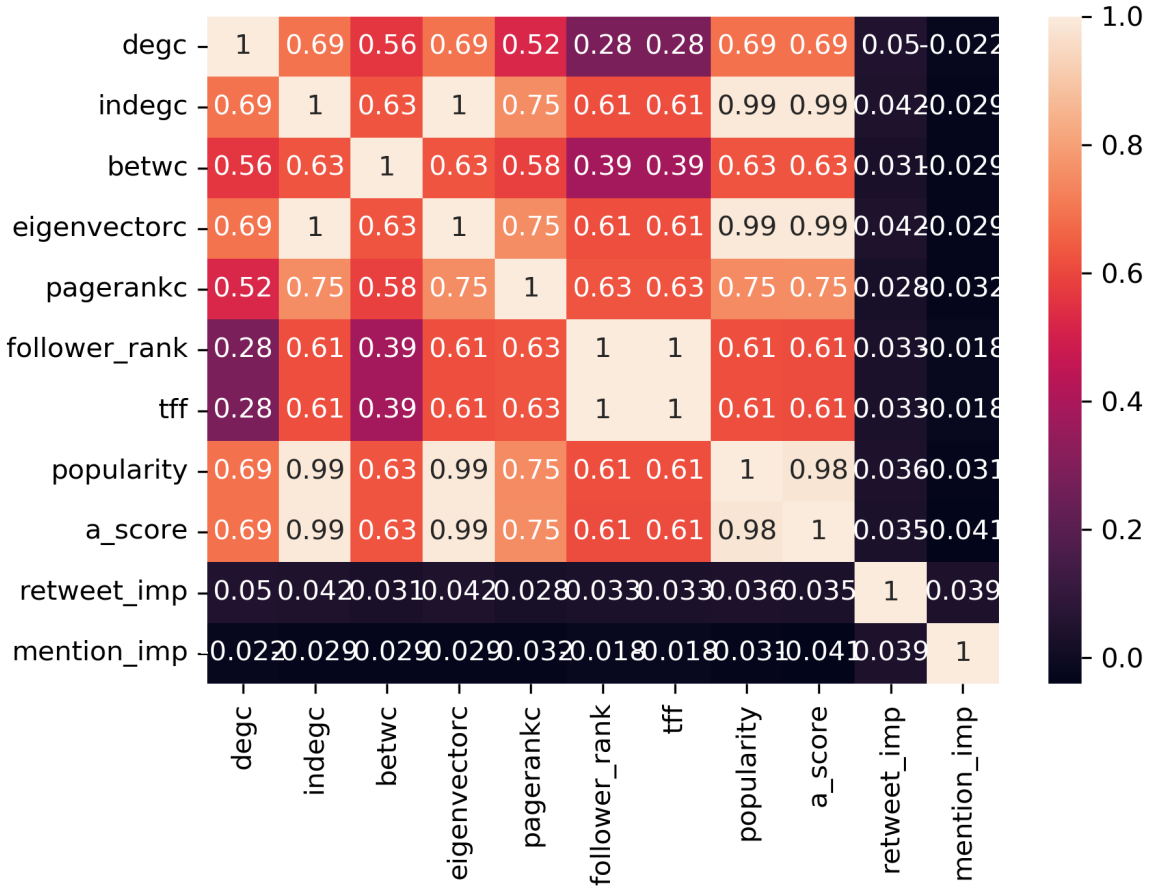
Σχήμα 6.23: Διάγραμμα συντελεστών συσχέτισης Spearman για μέτρα επιδραστικότητας. Αναδεικνύονται ισχυρές μη γραμμικές συσχετίσεις μεταξύ της πλειοψηφίας των μέτρων.

Παρατηρούμε ότι η πλειοψηφία των μέτρων επιδραστικότητας (εκτός των RI και MI) παρουσιάζουν υψηλή

συσχέτιση. Κάτι που επιβεβαιώσαμε άλλωστε ήδη γραφικά αλλά και αναμέναμε από τις συναρτήσεις των μετρικών.

Επίσης, σημειώνουμε ότι και ο συντελεστής Kendall δίνει παρόμοια αποτελέσματα (Σχήμα 6.24):

```
plt.close('all')
fig, ax = plt.subplots(1,1, figsize=(6.4, 4.8), tight_layout = True)#,dpi = 300)

# correlation matrix
corr_matr = df_social_centralities.drop(columns = 'nodeId').corr('kendall')
sns.heatmap(corr_matr, annot = True)
```



Σχήμα 6.24: Διάγραμμα συντελεστών συσχέτισης Kendall για μέτρα επιδραστικότητας. Αναδεικνύονται ισχυρές μη γραμμικές συσχετίσεις μεταξύ της πλειοψηφίας των μέτρων σε συμφωνία με το συντελεστή Spearman.

Σημειώνουμε ότι η συσχέτιση των μέτρων επιδραστικότητας είναι αναμενόμενη και τα αποτελέσματά μας βρίσκονται σε απόλυτη συμφωνία με μελέτες κεντρικότητας σε δίκτυα. Τα μέτρα κεντρικότητας παρουσιάζουν εν γένει θετική συσχέτιση η οποία διαφέρει από δίκτυο σε δίκτυο (Oldham et al., 2019).

6.6 Αξιολόγηση αποτελεσμάτων και μελλοντικές μελέτες

Στις προηγούμενες ενότητες είδαμε πως ο υπολογισμός της επιδραστικότητας των χρηστών κοινωνικών δικτύων είναι ένα ιδιαίτερα περίπλοκο και υπολογιστικά κοστοβόρο πρόβλημα. Υπολογίσαμε την επιδραστικότητα κάθε κόμβου/χρήστη και αναδείξαμε σημαντικές διαφορές μεταξύ αρκετών μέτρων κεντρικότητας/επιδραστικότητας. Επίσης, η μελέτη συσχέτισης των μέτρων αυτών υποδεικνύει κάποιες ισχυρές συσχετίσεις μεταξύ τους. Ωστόσο, τονίζουμε ότι κάθε μετρική αξιολογεί διαφορετικά την επιδραστικότητα σύμφωνα με τον εκάστοτε ορισμό της.

Η προσέγγισή που ακολουθήθηκε με την κατασκευή του δικτύου των σχέσεων ακολουθίας και τα παράλληλα δίκτυα σχέσεων reply, retweet και mention μεταξύ του ίδιου συνόλου χρηστών, οφείλεται καθαρά στη φύση των δεδομένων μας. Αρχικά, οι μελέτες μας πρέπει να επεκταθούν σε δεδομένα που επιτρέπουν τον υπολογισμό περισσότερων μετρήσιμων ποσοτήτων του Πίνακα 5.2 και κατ' επέκταση σε περισσότερα μέτρα επιδραστικότητας. Συγκεκριμένα, απαιτείται η συλλογή δεδομένων γύρω από θέματα (topics) και ο υπολογισμός των αντίστοιχων μετρήσιμων ποσοτήτων. Ακόμα, κρίνεται απαραίτητη η μέτρηση της συνολικής δραστηριότητας του χρήστη εντός του δικτύου. Τα δεδομένα αυτά είναι αναγκαία για τον ακριβέστερο εντοπισμό των πιο επιδραστικών κόμβων αλλά και βέλτιστη μετάδοση πληροφορίας μέσω αυτών για λόγους διαδικτυακού μάρκετινγκ. Η συλλογή ταξινόμηση και εκτίμηση τέτοιων δεδομένων είναι, ωστόσο, ιδιαίτερα χρονοβόρα και αφήνεται για μελλοντική μελέτη.

Με περισσότερα σύνολα δεδομένων στη διάθεσή μας, θα μπορούσαμε επίσης να συγκρίνουμε τη συμπεριφορά των μέτρων επιδραστικότητας και να επιβεβαιώσουμε ότι οι συσχετίσεις που αναγνωρίσαμε στη μελέτη μας γενικεύονται μεν, αποτελούν δε εγγενές χαρακτηριστικό του δικτύου μας όπως υποδεικνύει και η σχετική βιβλιογραφία.

Τέλος, αξίζει να πραγματοποιηθούν μελέτες ταξινόμησης χρηστών με χρήση αλγορίθμων και τεχνικών μηχανικής μάθησης και σύγκριση με παραδοσιακά μέτρα κεντρικότητας αλλά και μέτρα επιδραστικότητας ειδικά σχεδιασμένα για κοινωνικά δίκτυα. Οι μελλοντικές αυτές μελέτες επιτρέπουν τον ακριβέστερο υπολογισμό των επιδραστικών χρηστών, καλύτερη εποπτεία του δικτύου αλλά και στρατηγικότερη λήψη αποφάσεων σχετικά με προώθηση προϊόντων και διαφημιστικές προσεγγίσεις εντός του δικτύου.

Κεφάλαιο 7

Συμπεράσματα

Τα ψηφιακά κοινωνικά δίκτυα είναι ο χώρος στον οποίο οι άνθρωποι αλληλεπιδρούν σε μεγάλο βαθμό σήμερα. Η μοντελοποίησή τους αποτελεί ένα αντικείμενο μελέτης στο οποίο συμπράττουν οι κοινωνικές, ανθρωπιστικές και θετικές επιστήμες. Η ανάλυση κοινωνικών δικτύων δανείζεται τεχνικές, μεθόδους και αλγορίθμους από πολλές περιοχές των Μαθηματικών, ενώ παρουσιάζει μεγάλο ενδιαφέρον από την πλευρά της Επιχειρησιακής Έρευνας. Ολοένα και αυξανόμενος αριθμός επιχειρήσεων και οργανισμών δραστηριοποιείται σήμερα στα μέσα κοινωνικής δικτύωσης. Μεγάλο μέρος της προώθησης προϊόντων και της διαφήμισης βρίσκεται πλέον στα συγκεκριμένα μέσα. Γι' αυτό το λόγο κρίνεται απαραίτητη η μοντελοποίηση, μελέτη και ανάλυση των ιδιοτήτων τους.

Σημαντική ιδιότητα των χρηστών μέσων κοινωνικής δικτύωσης είναι η επιδραστικότητά τους, δηλαδή η δυνατότητα κάθε χρήστη να επηρεάζει τη δραστηριότητα άλλων χρηστών. Η μελέτη της επιδραστικότητας δεν παρουσιάζει μόνο μαθηματικό ενδιαφέρον αλλά αποτελεί επίσης σημαντική υπολογιστική πρόκληση εξαιτίας του μεγάλου όγκου δεδομένων. Η έννοια της επιδραστικότητας δεν είναι σαφώς ορισμένη γι αυτό και καταφεύγουμε σε πληθώρα μέτρων για την προσέγγιση και υπολογισμό της. Κεντρικό ρόλο στην ανάλυση επιδραστικότητας παίζει ο αλγόριθμος PageRank, ένας αλγόριθμος ταξινόμησης ιστοσελίδων σύμφωνα με τη σημαντικότητά τους. Ο συγκεκριμένος αλγόριθμος δεν εφαρμόζεται μόνο αυτούσιος σε κοινωνικά δίκτυα αλλά τροποποιείται και προσαρμόζεται ειδικά σε αυτά.

Συγκεκριμένα, μελετήσαμε την επιδραστικότητα χρηστών στο Twitter, ένα πολύπλοκο κοινωνικό δίκτυο που μπορεί να μοντελοποιηθεί χρησιμοποιώντας δύο τύπους κόμβων (χρήστες και δημοσιεύσεις - tweet) αλλά και πληθώρα πιθανών αλληλεπιδράσεων μεταξύ τους. Εναλλακτικά, το δίκτυο μπορεί να μοντελοποιηθεί ως σύνολο χρηστών με σχέσεις ακολουθίας μεταξύ τους ανάγοντας τις υπόλοιπες σχέσεις σε πρόσθετες αλληλεπιδράσεις μεταξύ των χρηστών. Έτσι, κατασκευάσαμε ένα βασικό δίκτυο αλληλεπιδράσεων και επιμέρους παράλληλα δίκτυα σε αυτό για κάθε μορφή αλληλεπίδρασης. Με τη συγκεκριμένη προσέγγιση ορίσαμε μετρήσιμες ποσότητες για κάθε κόμβο. Οι συγκεκριμένες ποσότητες χρησιμοποιήθηκαν για τον ορισμό μέτρων

επιδραστικότητας εξειδικευμένων στο Twitter σύμφωνα με διαφορετικούς ορισμούς της. Πολλά από τα μέτρα αυτά αποτελούν παραλλαγές του αλγορίθμου PageRank εξειδικευμένα στο συγκεκριμένο κοινωνικό μέσο.

Ένα μεγάλο σύνολο δεδομένων Twitter με περίπου $\sim 5 \times 10^5$ κόμβους και $\sim 10^6$ ακμές χρησιμοποιήθηκε για τη μελέτη της επιδραστικότητας των χρηστών του. Τα διαθέσιμα δεδομένα επέβαλλαν τη δημιουργία ενός δικτύου με κόμβους τους χρήστες και ακμές τις σχέσεις ακολουθίας μεταξύ τους αλλά και τριών επιπλέον δικτύων με διαφορετικές αλληλεπιδράσεις μεταξύ των χρηστών (retweet, reply και mention). Ο απαραίτητος κώδικας για την ανάλυση γράφτηκε σε γλώσσα Python εισάγοντας τα δεδομένα σε μορφή κειμένου και μοντελοποιώντας το δίκτυο με χρήση συγκεκριμένων βιβλιοθηκών κώδικα. Επίσης, κατασκευάστηκαν συναρτήσεις που λαμβάνουν ως είσοδο τα δεδομένα και υπολογίζουν τις μετρήσιμες ποσότητες κάθε κόμβου αλλά και πληθώρα μέτρων επιδραστικότητας για το δίκτυο σχέσεων ακολουθίας. Έτσι, με χρήση των συγκεκριμένων συναρτήσεων υπολοίστηκαν τόσο οι ποσότητες όσο και οι μετρικές επιδραστικότητας για το συγκεκριμένο σύνολο δεδομένων, ενώ επιπλέον εφαρμόστηκαν παραδοσιακά μέτρα κεντρικότητας αλλά και ο αλγόριθμος PageRank.

Τα αποτελέσματα της ανάλυσης υποδεικνύουν σημαντικές διαφορές μεταξύ των μέτρων που χρησιμοποιήθηκαν. Συγκεκριμένα, η τομή των συνόλων των 1000 πιο επιδραστικών κόμβων για κάθε μέτρο δίνει ένα σύνολο 3581 κόμβων. Η οπτικοποίηση των αποτελεσμάτων, με σχεδιασμό του δικτύου λαμβάνοντας υπόψιν τις τιμές επιδραστικότητας κάθε μέτρου, αναδεικνύει σημαντικές ποιοτικές διαφορές αλλά και ομοιότητες. Η ποσοτική σύγκριση των μέτρων επιδραστικότητας υποδεικνύει ισχυρές συσχετίσεις μεταξύ τους που βρίσκεται σε συμφωνία τόσο με τους ορισμούς τους αλλά και αντίστοιχες μελέτες συσχέτισης που παρουσιάζονται στη σχετική βιβλιογραφία. Τέλος, η συσχέτιση των ποικίλων μετρικών χρήζει περαιτέρω διερεύνησης με μεγαλύτερα σύνολα δεδομένων, περισσότερες μετρήσιμες ποσότητες αλλά και μέτρα επιδραστικότητας για ορθότερο και ακριβέστερο εντοπισμό των σημαντικότερων χρηστών εντός του δικτύου.

Βιβλιογραφία

[1] Κολέτσος, Ι., Στογιαννης, Δ. (2012). *Εισαγωγή στην Επιχειρησιακή Έρευνα*. Εκδόσεις Συμεών.

[2] Aggarwal, C. C. (2011). *Social network data analytics*. New York: Springer.

[3] Ahuja, R. K., Magnanti, T. L., Orlin, J. B., Reddy, M. R. (1995). Chapter 1 Applications of network optimization. *Handbooks in Operations Research and Management Science*, 7, 1{83. https://doi.org/10.1016/s0927-0507(05)80118-5

[4] Aleahmad, A., Karisani, P., Rahgozar, M., & Oroumchian, F. (2016). OLFinder: Finding opinion leaders in online social networks. *Journal of Information Science*, 42(5), 659{674. https://doi.org/10.1177/0165551515605217

[5] Anger, I., Kittl, C. (2011). Measuring influence on Twitter. *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies - I-KNOW '11*. https://doi.org/10.1145/2024288.2024326

[6] Ben Jabeur, L., Tamine, L., & Boughanem, M. (2012). Active Microbloggers: Identifying Influencers, Leaders and Discussers in Microblogging Networks. *String Processing and Information Retrieval*, 111{117. https://doi.org/10.1007/978-3-642-34109-0_12

[7] Bigonha, C., Cardoso, T. N. C., Moro, M. M., Gonçalves, M. A., & Almeida, V. A. F. (2011). Sentiment-based influence detection on Twitter. *Journal of the Brazilian Computer Society*, 18(3), 169{183. https://doi.org/10.1007/s13173-011-0051-5

[8] Brin S., Page L. (1998). The anatomy of a large-scale hypertectual Web search engine.*Computer Networks and ISDN Systems*, 30(1-7), 107-17. https://doi.org/10.1016/s0169-7552(98)00110-x

[9] Bona, M. (2017). *A Walk Through Combinatorics: An Introduction to Enumeration and Graph Theory* (4th e.d.). World Scientific.

[10] Bonacich, P. (1987) Power and Centrality: A Family of Measures. *American Journal of Sociology*, 92(5), 1170-1182. http://dx.doi.org/10.1086/228631

[11] Bondy, J. A., Murty, U. S. (1982). *Graph Theory with Applications* (5th e.d.). North-Holland.

[12] De Domenico, M., Lima, A., Mougel, P., Musolesi, M. (2013). The Anatomy of a Scientific Rumor. *Scientific Reports*, 3(1), 2980. https://doi.org/10.1038/srep02980

[13] Diestel, R. (200). *Graph Theory* (2nd e.d.). Springer.

[14] Ding, Z., Jia, Y., Zhou, B., Han, Y., He, L., Zhang, J. (2013). Measuring the spreadability of users in microblogs. *Journal of Zhejiang University SCIENCE C*, 14(9), 701{710. https://doi.org/10.1631/jzus.ciip1302

[15] Dodge, M., Kitchin, R. (2001). *Atlas of Cyberspace*. Addison-Wesley.

[16] Freeman, L. C. (2004). *The Development of Social Network Analysis: A study in the Sociology of Science*. Empirical Press.

[17] Freeman, L. C. (1977). A set of measures of centrality based upon bernreenness, *Sociometry* 40(1). https://doi.org/10.2307/3033543

[18] Gayo-Avello, D. (2013). Nepotistic relationships in Twitter and their impact on rank prestige algorithms. *Information Processing & Management*, 49(6), 1250{1280. https://doi.org/10.1016/j.ipm.2013.06.003

[19] Gayo-Avello, D. , Brenes, D. J., Fernández-Fernández, D., Fernández-Menéndez, M. E., & García-Suárez, R. (2011). De retibus socialibus et legibus momenti. *EPL (Europhysics Letters)*, 94 (3), 38001.

[20] Hajian, B., & White, T. (2011, October 1). Modelling Influence in a Social Network: Metrics and Evaluation. https://doi.org/10.1109/PASSAT/SocialCom.2011.118

[21] Hillier, F. S., Lieberman, G. J. (2001). *Introduction to Operations Research* (7th e.d.). McGraw-Hill.

[22] Hirsch, J. E. (2010). An index to quantify an individual's scientific research output that takes into account the effect of multiple coauthorship. *Scientometrics*, 85(3), 741{754. https://doi.org/10.1007/s11192-010-0193-9

[23] Huang, P.-Y., Liu, H.-Y., Lin, C.-T., & Cheng, P.-J. (2013). A Diversity-Dependent Measure for Discovering Influencers in Social Networks. *Information Retrieval Technology*, 368{379. https://doi.org/10.1007/978-3-642-45068-6_32

[24] Jackson, M. O. (2011). *Social and economic networks*. Princeton, N.J. ; Woodstock: Princeton University Press.

[25] Jabeur, L. B., Tamine, L., & Boughanem, M. (2012). Active microbloggers: Identifying influencers, leaders and discussers in microblogging networks. In L. Calderón-Benavides, C. N. González-Caro, E. Chávez, & N. Ziviani (Eds.), String processing and information retrieval - 19th international symposium, SPIRE 2012, cartagena de indias, colombia, october 21-25, 2012. proceedings. In *Lecture Notes in Computer Science*. 7608:111-117. Springer.

[26] Katz, L.(1953). A new status index derived from sociometric analysis, *Psychometrika*, 18(1), 39-43. https://doi.org/10.1007/BF02289026

[27] Khrabrov, A. , Cybenko, G. (2010). *Discovering influence in communication networks using dynamic graph analysis*. In A. K. Elmagarmid, & D. Agrawal (Eds.), *Proceedings of the 2010 IEEE second international conference on social computing, socialcom / IEEE international conference on privacy, security, risk and trust, PASSAT 2010, Minneapolis, Minnesota, USA, August 20-22, 2010* (288{294). IEEE Computer Society.

[28] Kitsak, M., Gallos, L. K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H. E., & Makse, H. A. (2010). Identification of influential spreaders in complex networks. *Nature Physics*, 6 , 888{893.

[29] Langville, A. N., Meyer, C. D. (2006). *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton University Press.

[30] Lee, C., Kwak, H., Park, H., & Moon, S. (2010). Finding influentials based on the temporal order of information adoption in twitter. *Proceedings of the 19th International Conference on World Wide Web - WWW '10*. https://doi.org/10.1145/1772690.1772842

[31] Li, X., Cheng, S., Chen, W., & Jiang, F. (2013). Novel user influence measurement based on user interaction in microblog. *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. https://doi.org/10.1145/2492517.2492635

[32] Liu, D., Wu, Q., Han, W. (2013). Measuring Micro-blogging User Influence Based on User-Tweet Interaction Model. *Lecture Notes in Computer Science*, 146{153. https://doi.org/10.1007/978-3-642-38715-9_18

[33] Majer, T., & Simko, M. (2012). Leveraging microblogs for resource ranking. In M. Bieliková, G. Friedrich, G. Gottlob, S. Katzenbeisser, & G. Turan (Eds.), *SOFSEM 2012: Theory and practice of computer science - 38th conference on current trends in theory and practice of computer science, Špindleruv Czech Republic, January 21-27, 2012 Proceedings* . In *Lecture Notes in Computer Science*: 7147 (518{529). Springer.

[34] Nagmoti, R., Teredesai, A., De Cock, M. (2010, August 1). Ranking Approaches for Microblog Search. https://doi.org/10.1109/WI-IAT.2010.170

[35] Newman, M. (2010). *Networks: An Introduction*. Oxford University Press.

[36] Oldham, S., Fulcher, B., Parkes, L., Arnatkeviciute, Suo, C., Fornito, A. (2019). Consistency and differences between centrality measures across distinct classes of networks. *PLOS ONE*, 14(7), e0220061. https://doi.org/10.1371/journal.pone.0220061

[37] Omodei, E., De Domenico, M., & Arenas, A. (2015). Characterizing interactions in online social networks during exceptional events. *Frontiers in Physics*, 3. https://doi.org/10.3389/fphy.2015.00059

[38] Pal, A., Counts, S. (2011). Identifying topical authorities in microblogs. *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining - WSDM 2011*. https://doi.org/10.1145/1935826.1935843

[39] pj. (2018, October 29). Closeness Centrality via NetworkX is taking too long! Retrieved February 25, 2022, from Medium website: https://medium.com/@pasdan/closeness-centrality-via-networkx-is-taking-too-long-1a58e648f5ce

[40] Pujol, J. M., Sanguesa, R., & Delgado, J. (2002). Extracting reputation in multi agent systems by means of social network topology. *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems Part 1 - AAMAS '02*. https://doi.org/10.1145/544741.544853

[41] Raj P.M., K., Mohan, A., Srinivasa, K. G. (2018). *Practical Social Network Analysis with Python*. In *Computer Communications and Networks*. https://doi.org/10.1007/978-3-319-96746-2

[42] Rossi, R. A., Ahmed, N. K. (2015). The Network Data Repository with Interactive Graph Analytics and Visualization. http://networkrepository.com

[43] Riquelme, F., Gonzalez-Cantergiani, P. (2016). Measuring user influence on Twitter: A survey. *Information Processing and Management*, 52:949-975.

[44] Romero, D. M., Galuba, W., Asur, S., Huberman, B. A. (2011). Influence and Passivity in Social Media. *Machine Learning and Knowledge Discovery in Databases*, 6913, 18{33. https://doi.org/10.1007/978-3-642-23808-6_2

[45] Silva, A., Guimarães, S., Meira, W., Zaki, M. (2013). ProfileRank. *Proceedings of the 7th Workshop on Social Network Mining and Analysis - SNAKDD '13*. https://doi.org/10.1145/2501025.2501033

[46] Simmie, D. S., Vigliotti, M. G., & Hankin, C. (2014). Ranking twitter influence by combining network centrality and influence observables in an evolutionary model, *Journal of Complex Networks*, 2(4), 495{517. https://doi.org/10.1093/comnet/cnu024

[47] SNAP: Stanford Network Analysis Project. (2009). Retrieved February 23, 2022, from snap.stanford.edu website: https://snap.stanford.edu/index.html

[48] SNAP: Network datasets: Higgs Twitter Dataset. (2013). Retrieved February 23, 2022, from snap.stanford.edu website: https://snap.stanford.edu/data/higgs-twitter.html

[49] Srinivasan, M. S., Srinivasa, S., & Thulasidasan, S. (2013). Exploring celebrity dynamics on Twitter. *Proceedings of the 5th IBM Collaborative Academia Research Exchange Workshop on - I-CARE '13*. https://doi.org/10.1145/2528228.2528242

[50] Srinivasan, M. S. , Srinivasa, S. , & Thulasidasan, S. (2014). *A comparative study of two models for celebrity identification on twitter*. In S. Bedathur, D. Sri- vastava, S. R. Valluri (Eds.), *20th international conference on management of data, COMAD 2014, Hyderabad, India, December 17-19, 2014* (57{65). Computer Society of India.

[51] Stewart, W. J. (1994). *Introduction to the numerical solution of Markov chains*. Princeton, N.J.: Princeton University Press.

[52] Taha, H.A. (2007). *Operations Research: An Introduction* (8th e.d.). Pearson Prentice Hall.

[53] Tunkelang, D. (2009, January 13). A Twitter Analog to PageRank. Retrieved February 27, 2022, from The Noisy Channel website: https://thenoisychannel.com/2009/01/13/a-twitter-analog-to-pagerank/

[54] Noro, T. , Ru, F. , Xiao, F. , Tokuda, T. (2012). Twitter user rank using keyword search. In P. Vojtas, Y. Kiyoki, H. Jaakkola, T. Tokuda, & N. Yoshida (Eds.), *Information modelling and knowledge bases XXIV, 22nd european-japanese conference on information modelling and knowledge bases (EJC 2012), Prague, Czech Republic, June, 4-9, 2012*. In *Frontiers in Artificial Intelligence and Applications*: 251 (31{48). IOS Press.

[55] Wilson, R. J. (1996). *Introduction to graph theory*. Harlow: Longman.

[56] Winston, W. L. (2004). *Operations Research: Applications and Algorithms* (4th e.d.). Thomson Learning.

[57] Yamaguchi, Y., Takahashi, T., Amagasa, T., Kitagawa, H. (2010). TURank: Twitter User Ranking Based on User-Tweet Graph Analysis. *Web Information Systems Engineering { WISE 2010*, 240{253. https://doi.org/10.1007/978-3-642-17616-6_22

[58] Ye, S., Wu, S. F. (2013). Measuring message propagation and social influence on Twitter.com. *Int. J. Communication Networks and Distributed Systems*, 11(1), 59-76. http://dx.doi.org/10.1007/978-3-642-16567-2_16

[59] Yin, Z., Zhang, Y. (2012). Measuring pair-wise social influence in microblog. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*. 502{507. IEEE. https://doi.org/10.1109/SocialCom-PASSAT.2012.10

[60] Yuan, J. , Li, L. , Huang, L. L. M. (2013). Topology-based algorithm for users' influence on specific topics in micro-blog. *Journal of Information and Computational Science*, 10 (8), 2247{2259. https://doi.org/10.12733/jics20102229

[61] Zhang, J., Zhang, R., Sun, J., Zhang, Y., Zhang, C. (2016). TrueTop: A Sybil-Resilient System for User Influence Measurement on Twitter. *IEEE/ACM Transactions on Networking*, 24(5), 2834{2846. https://doi.org/10.1109/tnet.2015.2494059