Ατομική Διπλωματική Εργασία

ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΠΕΙΡΑΜΑΤΙΚΗ ΑΞΙΟΛΟΓΗΣΗ ΑΛΓΟΡΙΘΜΩΝ ΓΙΑ ΑΝΑΚΑΛΥΨΗ ΓΝΩΣΗΣ ΣΕ ΚΟΙΝΩΝΙΚΑ ΔΙΚΤΥΑ

Ζήνων Ζένιου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2011

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Υλοποίηση και πειραματική αξιολόγηση αλγορίθμων για ανακάλυψη γνώσης σε κοινωνικά δίκτυα

Ζήνων Ζένιου

Επιβλέπων Καθηγητής
Γιώργος Πάλλης

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2011

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου Δρ. Γιώργο Πάλλη, καθώς και τον Νικόλα Λουλλούδη, για την ανεκτίμητη βοήθεια και καθοδήγηση που μου έχουν δώσει για την εκπλήρωση αυτής της διπλωματικής εργασίας αυτής.
Επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου για την συμπαράστασή τους.

Περίληψη

Τα κοινωνικά δίκτυα αναπτύσσονται ραγδαία σε βαθμό που γίνονται πλέον αναπόσπαστο μέρος του Διαδικτύου. Ιστοσελίδες όπως το Facebook, το YouTube και το MySpace είναι χαρακτηριστικά παραδείγματα διάσημων κοινωνικών δικτύων που έχουν προσελκύσει εκατομμύρια χρήστες οι οποίοι αλληλεπιδρούν μεταξύ τους και δημοσιεύουν περιεχόμενα. Η μελέτη και ανάλυση τέτοιων δικτύων μπορούν να μας παράσχουν σημαντικές πληροφορίες σε θέματα δομής δικτύου και ιδιοτήτων, καθώς και διάδοσης πληροφορίας.

Η μελέτη αυτή παρουσιάζει την ανάλυση του κοινωνικού δικτύου της Νέας Ορλεάνης το οποίο αποτελείται από 63 χιλιάδες χρήστες και 817 χιλιάδες κοινωνικές σχέσεις μεταξύ τους και εξετάζει τα χαρακτηριστικά του καθώς και την ανίχνευση κοινοτήτων. Τα αποτελέσματα καταδεικνύουν την ύπαρξη ιδιοτήτων power-law και small-world μέσα στο δίκτυο. Παρατηρείται ότι οι χρήστες που έχουν λίγους φίλους τείνουν να είναι πιο συνδεδεμένοι. Επίσης παρατηρείται ότι οι κοινότητες μεγέθους μέχρι 100 κόμβων παρουσιάζουν πολύ καλή ποιότητα, ενώ κοινότητες πέρα των 100 κόμβων παρουσιάζουν χειρότερη ποιότητα. Επίσης, βάση ενός αλγόριθμου το δίκτυο χωρίστηκε σε κοινότητες και παρατηρήθηκε μεγάλος αριθμός μικρών κοινοτήτων και τέσσερις κοινότητες με τεράστιο αριθμό κόμβων. Τέλος, εξετάστηκε το centrality και παρατηρήθηκε ότι οι κόμβοι δεν επηρεάζουν τις αλληλεπιδράσεις μεταξύ άλλων κόμβων και η απόσταση όλων των κόμβων μεταξύ τους είναι περίπου η ίδια.

Περιεχόμενα

Κατάλογος Σχημάτων

Κεφάλαιο 1

Εισαγωγή

1.1 Εισαγωγή στα Online κοινωνικά δίκτυα

Πρόσφατα έχουν κάνει την εμφάνιση τους μια νέα κατηγορία δικτύων τα λεγόμενα online κοινωνικά δίκτυα (online social networks - OSN) τα οποία έχουν γίνει διάσημα και έχουν ξεπεράσει σε σύντομο χρονικό διάστημα το παραδοσιακό WWWeb σε χρήση [32]. Αυτό είχε ως αποτέλεσμα να προσελκύσουν ολοένα και περισσότερο την προσοχή των ακαδημαϊκών και ερευνητών οι οποίοι, γοητευμένοι από τις πολλές δυνατότητες που παρέχουν και την προσβασιμότητα τους, θέλουν να τα μελετήσουν.

Ακριβώς όπως και ένα δίκτυο υπολογιστών είναι ένα σύνολο από μηχανήματα που συνδέονται μεταξύ τους με καλώδια, έτσι και ένα κοινωνικό δίκτυο (social network) μπορεί να αναπαρασταθεί ως ένα σύνολο ανθρώπων (ή οργανώσεων ή άλλων κοινωνικών φορέων) που συνδέονται με ένα σύνολο από κοινωνικές σχέσεις (social relationships), όπως η φιλία, η συνεργασία ή η ανταλλαγή πληροφοριών και πόρων.

Τα online κοινωνικά δίκτυα εξελίσσονται και αναπτύσσονται ραγδαία σε τέτοιο βαθμό που γίνονται πλέον αναπόσπαστο μέρος του Internet. Οι άνθρωποι συνεχώς επιθυμούν να αλληλεπιδρούν μεταξύ τους τόσο σε προσωπικές επαφές όσο και στις επιχειρήσεις.

Η ικανότητα του Internet να προσφέρει αυτή τη δυνατότητα δικτύωσης γίνεται ολοένα και πιο ισχυρή και πλούσια, με αποτέλεσμα να έχουμε μια πληθώρα από online social network websites.

Ιστοσελίδες όπως το Facebook [1] (με πάνω από 750 εκατομμύρια χρήστες) [37], LinkedIn (με πάνω από 80 εκατομμύρια χρήστες), MySpace (με πάνω από 74 εκατομμύρια χρήστες) και Orkut (με πάνω από 74 εκατομμύρια χρήστες) είναι χαρακτηριστικά παραδείγματα διάσημων κοινωνικών δικτύων που έχουν προσελκύσει εκατομμύρια χρήστες, από τους οποίους πολλοί τα έχουν ενσωματώσει στην καθημερινότητα τους. Καθώς αυτές οι ιστοσελίδες υποστηρίζουν την διατήρηση των υπαρχόντων κοινωνικών δικτύων (φιλίες που αναπτύχθηκαν στην καθημερινή ζωή, σχολείο, οικογένεια, γείτονες κτλ), υποστηρίζουν επίσης την σύνδεση μεταξύ νέων χρηστών με βάση κοινά ενδιαφέροντα, δραστηριότητες, φίλους κτλ. Επίσης άλλες ιστοσελίδες προορίζονται για συγκεκριμένους χρήστες, όπως το DeviantArt (με πάνω από 14 εκατομμύρια χρήστες) που είναι για γραφίστες, ενώ άλλες ιστοσελίδες διαφοροποιούνται ως προς τον τρόπο ενσωμάτωσης και ανταλλαγής πληροφοριών, όπως το Youtube (με πάνω από 720 εκατομμύρια χρήστες) για βίντεο και μουσική, το Flickr για φωτογραφίες καθώς, και το LiveJournal (με πάνω από 24 εκατομμύρια χρήστες) για blogs.

Παρά την ποικιλία των online κοινωνικών δικτύων όλα βασίζονται σε ένα βασικό μηχανισμό. Οι χρήστες, που είναι το βασικό συστατικό κάθε δικτύου, γίνονται μέλη του κοινωνικού δικτύου, δημοσιεύουν το δικό τους περιεχόμενο, και δημιουργούν σχέσεις με άλλους χρήστες.

1.2 Κίνητρο για τη Μελέτη

Η εξαιρετικά υψηλή δημοτικότητα και ραγδαία ανάπτυξη των online κοινωνικών δικτύων, δίνει μια μοναδική ευκαιρία για μελέτη και κατανόηση των ιδιοτήτων αυτών των κοινωνικών δικτύων. Η συγκεκριμένη ανάλυση μπορεί να βοηθήσει σε πολλούς τομείς.

[1] Τα στατιστικά για τον αριθμό χρηστών στα online κοινωνικά δίκτυα πάρθηκαν από την ιστοσελίδα www.google.com/adplanner/planning/site_profile#siteDetails για όλα τα κοινωνικά δίκτυα. Τελευταία επίσκεψη της σελίδας έγινε τον Μάιο του 2011.

Μια εις βάθος κατανόηση της δομής ενός κοινωνικού δικτύου, καθώς και της εξέλιξης της δομής αυτής στην πάροδο του χρόνου, μπορεί να οδηγήσει σε χρήσιμες πληροφορίες και γνώση που μπορούν να χρησιμοποιηθούν όχι μόνο στην αξιολόγηση και βελτίωση των υφιστάμενων πλατφόρμων που υποστηρίζουν τα online κοινωνικά δίκτυα (όπως το Facebook), αλλά και στην δημιουργία καλύτερων μελλοντικών πλατφόρμων που θα προσφέρουν βελτιωμένες υπηρεσίες στους χρήστες.

Όσον αφορά τα πληροφοριακά συστήματα (Information Systems - IS), τα κοινωνικά δίκτυα μπορούν να δώσουν μια σημαντική γνώση για τις ιδιότητες τους, παρέχοντας έτσι τη δυνατότητα εκμετάλλευσης και αξιοποίησης τους έτσι ώστε να βελτιωθούν τα συστήματα. Τέτοιες βελτιώσεις θα μπορούσαν να εφαρμοστούν για τη διάδοση της πληροφορίας και τον έλεγχο της διάδοσης καθώς επίσης και σε νέους τρόπους αναζήτησης και ανάκτησης πληροφορίας.

Πέραν από την χρησιμότητα τους στην πληροφορική, η μελέτη των online κοινωνικών δικτύων είναι εξίσου σημαντική και για τους ειδικευμένους στην κοινωνιολογία, στην πολιτική και στο μάρκετινγκ. Προσφέρουν μια τεράστια ποσότητα δεδομένων που προηγουμένως ήταν αδύνατο να συλλεχθεί. Δίνεται η δυνατότητα στους κοινωνιολόγους να εξετάσουν τα δεδομένα έτσι ώστε να μπορέσουν να επαληθεύσουν τις θεωρίες επικοινωνίας, καθώς και να αναζητήσουν νέους τρόπους και μορφές επικοινωνίας.

Για τους εξειδικευμένους στην πολιτική και στο μάρκετινγκ, η μελέτη και η παρατήρηση της ροής της πληροφορίας μέσα στο κοινωνικό δίκτυο, μπορεί να συμβάλει στην βελτίωση των τεχνικών διαφήμισης και διάδοσης της πληροφορίας.

Στόχος λοιπόν της εργασίας είναι να εξετάσουμε και να κατανοήσουμε τις ιδιότητες ενός online κοινωνικού δίκτυου, και να καταλήξουμε σε χρήσιμα συμπεράσματα.

1.3 Συνεισφορά

Σε αυτή τη διπλωματική έχουν εφαρμοστεί γραφο-θεωρητικές έννοιες πάνω στο online κοινωνικό δίκτυο της Νέας Ορλεάνης. Έχει παρατηρηθεί ότι η κατανομή του degree

παρουσιάζει ιδιότητες του power law. Επίσης έχει παρατηρηθεί ψηλό clustering coefficient και χαμηλή διάμετρος. Για περαιτέρω ανάλυση, έχει κατασκευαστεί ένας τυχαίος γράφος ο οποίος συγκρίνεται με το clustering coefficient. Έχει λοιπόν παρατηρηθεί ότι ο συντελεστής του κοινωνικού δικτύου είναι 24 φορές πιο μεγάλος από τον τυχαίο γράφο. Με συνδυασμό ψηλού clustering coefficient και μικρής διαμέτρου, συμπεραίνεται ότι το δίκτυο ανήκει στην κατηγορία των small-world δικτύων. Στη συνέχεια, έχει εξεταστεί η ύπαρξη κοινοτήτων μέσα στο κοινωνικό δίκτυο. Έχει παρατηρηθεί πως η ποιότητα της καλύτερης δυνατής κοινότητας ως συνάρτηση του μεγέθους είναι στους 100 κόμβους και όσο μεγαλώνει ο αριθμός κόμβων τόσο χειροτερεύει η ποιότητα. Έχει εφαρμοστεί ο αλγόριθμος modularity για να χωριστεί το δίκτυο σε κοινότητες. Επίσης έχει παρατηρηθεί ψηλή τιμή modularity με μεγάλο αριθμό μικρών κοινοτήτων, με εξαίρεση 4 κοινοτήτων που έχουν μεγάλο αριθμό κόμβων. Τέλος, έχει εξεταστεί το centrality των κόμβων και έχει παρατηρηθεί ότι η συντριπτική πλειοψηφία κόμβων παρουσιάζουν χαμηλό betweenness και τιμές closeness πολύ κοντά στο μέσο όρο. Τα συμπεράσματα αυτά καταδεικνύουν ότι οι κόμβοι είναι αποκεντρωμένοι από το δίκτυο και δεν επηρεάζουν τις αλληλεπιδράσεις μεταξύ άλλων κόμβων. Επίσης έχουν σχετικά την ίδια απόσταση μεταξύ τους

Το υπόλοιπο μέρος της διπλωματικής είναι οργανωμένο ως εξής. Το Κεφάλαιο 2 παρέχει τη σχετική βιβλιογραφία με την οποία σχετίζεται το θέμα. Το Κεφάλαιο 3 περιγράφει τις μετρικές που έχουν χρησιμοποιηθεί στα πειράματα. Στο Κεφάλαιο 4 περιγράφεται η μεθοδολογία που ακολουθήθηκε για την διεξαγωγή της έρευνας. Στο Κεφάλαιο 5 παρουσιάζονται τα πειράματα και τα αποτελέσματα τους. Τέλος, στο κεφάλαιο 6 παρουσιάζονται τα τελικά συμπεράσματα, εφαρμογές και περιοχές για μελλοντική έρευνα.

Κεφάλαιο 2

Σχετική Βιβλιογραφία

Η μελέτη των κοινωνικών δικτύων και η κατανόηση των χαρακτηριστικών τους καθώς και συσχέτιση τους με την θεωρία του γράφου είναι μια από τις πολλές και πρώτες έρευνες που έγιναν από ερευνητές διαφόρων κλάδων επιστήμης όπως Πληροφορική, Κοινωνιολογία και Οικονομικά. Οι Jon Kleinberg (καθηγητής Πληροφορικής) και David Easley (καθηγητής Κοινωνικών Επιστημών) συνεργάστηκαν και έχουν εκδώσει ένα βιβλίο που συνδυάζει διαφορετικές επιστημονικές απόψεις στην προσέγγιση της κατανόησης δικτύων και της συμπεριφοράς τους. [10].

Ερευνητές έδειξαν ότι το Web [26] και τα κοινωνικά δίκτυα [2] παρουσιάζουν power law distribution καθώς και πρότειναν αλγόριθμους για αποτελεσματική αναζήτηση σε τέτοια δίκτυα [3] . Power law δίκτυα είναι τα δίκτυα όπου η πιθανότητα ενός κόμβου να έχει degree k είναι ανάλογη του $k^{-\alpha}$ για μεγάλο k και $a > 1$. Οπότε το degree των κόμβων παρουσιάζει heavy-tailed κατανομή.

Μια ενδιαφέρουσα έρευνα είναι τα πειράματα του Milgram [24]. Σκοπός των πειραμάτων ήταν η μελέτη των μηκών των μονοπατιών σε ένα δίκτυο, δίνοντας στους συμμετέχοντες ένα γράμμα να διαβιβάσουν σε γνωστό τους, με την ελπίδα ότι θα φτάσει σε ένα συγκεκριμένο παραλήπτη. Τα πειράματα έδειξαν ότι παρόλο που τα περισσότερα γράμματα χάθηκαν, το ένα τρίτο των γραμμάτων έφτασαν στον τελικό παραλήπτη περνώντας κατά μέσο όρο από 6 άτομα. Τα πειράματα αυτά ήταν η προέλευση της δημοφιλούς έννοιας “six degrees of separation” ή αλλιώς “small world phenomenon”. Περαιτέρω έρευνες έδειξαν ότι αρκετά δίκτυα, including Web, παρουσιάζουν small-world χαρακτηριστικά [4] [16] [2] όπως μικρή διάμετρο και ψηλό high clustering coefficient [35].

Ο Rapoport [31] και οι Erdos και Renyi [11] ήταν από τους πρώτους που μελέτησαν τους τυχαίους γράφους, το πιο απλό και χρήσιμο μοντέλο δικτύων και πρότειναν το απλό κλασσικό Poisson μοντέλο γράφου. Σε αυτό το μοντέλο, οι (undirected) ακμές τοποθετούνται τυχαία μεταξύ n αριθμό κόμβων για να δημιουργηθεί ένα δίκτυο το οποίο κάθε από τις $\frac{n\ n-1}{2}$ πιθανές ακμές παρουσιάζονται με πιθανότητα π, και ο αριθμός των ακμών που ενώνονται σε κάθε κόμβο (degree) κατανέμεται ανάλογα της διωνυμικής κατανομής ή την κατανομή Poisson στο όριο ενός μεγάλου n.

Η ανακάλυψη του power law degree distribution και των small world δικτύων, οδήγησε στην ανάπτυξη τυχαίων γράφων που παρουσιάζουν αυτά τα χαρακτηριστικά. Τέτοια μοντέλα είναι το web graph που προτείνουν οι Cooper και Frieze [8] και το Growing random network των Krapivsky και Redner [18] τα οποία έχουν σταθερή διάμετρο και λογαριθμικά αυξανόμενο degree.

Έρευνα για την εξέλιξη του δικτύου όσο περνάει ο χρόνος [20] έδειξε ότι τα δίκτυα παρουσιάζουν densification power law, δηλαδή το out-degree των κόμβων αυξάνεται με τον χρόνο ακολουθώντας ένα φυσικό μοτίβο (pattern). Επίσης, έδειξε ότι η διάμετρος μικραίνει όσο μεγαλώνει (σε κόμβους) το δίκτυο και πρότεινε το μοντέλο τυχαίου γράφου Forest Fire model το οποίο φέρει τα πιο πάνω χαρακτηριστικά. Λίγο καιρό μετά, οι ίδιοι ερευνητές παρουσίασαν τους πρωτοποριακούς γράφους Kronecker οι οποίοι όχι μόνο παρουσιάζουν τα χαρακτηριστικά των δικτύων αλλά και τα διαχρονικά χαρακτηριστικά (densification power-law, shrinking diameter). Επίσης με 4 μόνο παραμέτρους μπορούν να μοντελοποιήσουν αποτελεσματικά υπάρχοντα δίκτυα.

Κάποιοι μαθητές θέλοντας να δουν πως δημιουργούνται οι σχέσεις μέσα στα κοινωνικά δίκτυα [17], χρησιμοποίησαν ένα κοινωνικό δίκτυο και εξέτασαν τις αλληλεπιδράσεις μεταξύ χρηστών βάση των e-mail header και έδειξαν ότι νέες σχέσεις μέσα στο δίκτυο είναι πιο πιθανόν να δημιουργηθούν μεταξύ κόμβων που βρίσκονται κοντά ο ένας στον άλλο.

Ερευνητές κατέδειξαν την ύπαρξη κοινοτήτων μέσα στα κοινωνικά δίκτυα [25] τα οποία είναι υποσύνολα του δικτύου και είναι πιο συνδεδεμένο εσωτερικά απ' ότι το δίκτυο συνολικά. Αυτό πυροδότησε την μελέτη αλγορίθμων για ανίχνευση κοινοτήτων μέσα στα δίκτυα. Οι πρώτες προσεγγίσεις χώριζαν τους κόμβους σε κοινότητες κρατώντας μικρό τον αριθμό των ακμών μεταξύ κοινοτήτων. Με αυτή την κλασσική προσέγγιση, προτάθηκαν αλγόριθμοι [30][15] οι οποίοι χωρίζουν το γράφο σε δυο κοινότητες, και στη συνέχεια τα χωρίζουν ξανά και ξανά ώσπου να φτάσουν στον αριθμό των κοινοτήτων που επέλεξε ο χρήστης. Το μειονέκτημα αυτών των αλγορίθμων είναι η προεπιλογή αριθμού τελικών κοινοτήτων.

Οι Girvan και Newman [25] πρότειναν αλγόριθμο που δεν χρειαζόταν να υπάρχει προϋπάρχουσα γνώση για τις κοινότητες μέσα σε δίκτυο. Ο αλγόριθμος υπολογίζει την πιο σημαντική ακμή μέσα στο δίκτυο και την αφαιρεί. Αυτό επαναλαμβάνεται μέχρι ο γράφος να χωριστεί σε κομμάτια, και αυτά τα κομμάτια θεωρούνται κοινότητες. Ο υπολογισμός της πιο σημαντικής ακμής γίνεται με την μετρική betweenness η οποία τείνει να δίνει ψηλή τιμή σε ακμές που είναι γέφυρες μεταξύ κοινοτήτων. Επειδή σε δίκτυα μεγάλης κλίμακας ο αλγόριθμος είναι αργός, προτάθηκε ένας εναλλακτικός αλλά πιο γρήγορος αλγόριθμος με βάση το modularity [27]. Ο αλγόριθμος αυτός αρχίζει με τον κάθε κόμβο να ανήκει σε μια ξεχωριστή κοινότητα και στη συνέχεια ενώνει ζεύγη κοινοτήτων, διαλέγοντας κάθε φορά το ζεύγος που θα δώσει την μεγαλύτερη αύξηση (ή μείωση) στο modularity. Οι Clauset *et al* [7] πρότειναν μια παραλλαγή του αλγόριθμου που είναι πιο βελτιστοποιημένος και αποδοτικός και επιτρέπει την χρησιμοποίηση του σε τεράστιους γράφους με χιλιάδες ακμές.

Άλλες προσεγγίσεις εξέτασαν την ανίχνευση πολλαπλών, overlapping κοινοτήτων και έρχονται σε αντίθεση με τις προσεγγίσεις που παρουσιάζονται πιο πάνω, οι οποίες χωρίζουν τους κόμβους σε μοναδικά non-overlapping κοινότητες. Ο Palla [29] πρότεινε την χρήση των k-cliques (Clique Percolation Method) για την ανίχνευση κοινοτήτων σε διαφορετικά μεγέθη. Οι Baumes *et al* [5] πρότειναν παρόμοια προσέγγιση η οποία κοιτάζει πρώτα για στενά συνδεδεμένους κόμβους.

Ένας άλλος αλγόριθμος προτάθηκε από τον Leskovec [23] ο οποίος μετρά την ποιότητα της καλύτερης δυνατής κοινότητας με βάση το μέγεθος, χρησιμοποιώντας τη

μετρική conductance. Στην έρευνα του, ο Leskovec παρατήρησε ότι σχεδόν όλα τα δίκτυα που μελετήθηκαν, παρουσίασαν κοινότητες σε πολύ μικρά μεγέθη και όσο αυξανόταν το μέγεθος τόσο χειροτέρευε η τιμή conductance.

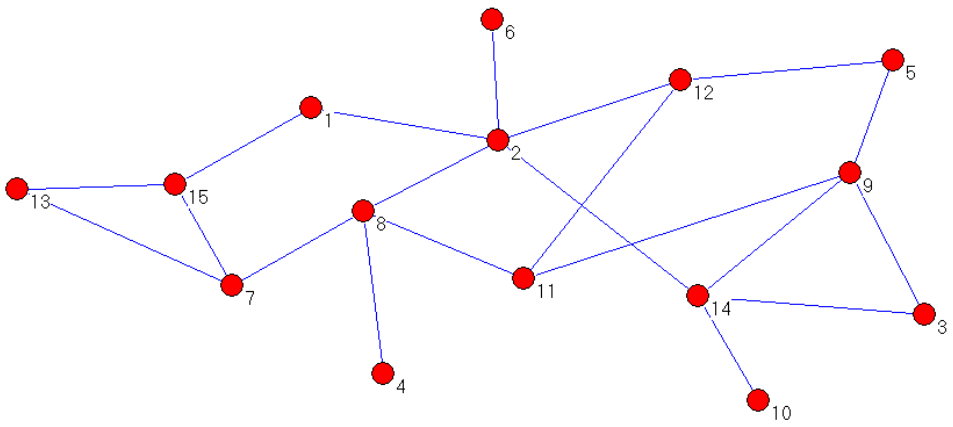
Πρόσφατες έρευνες σχετικά με την αλληλεπίδραση χρηστών σε online κοινωνικά δίκτυα έδειξαν ότι υπάρχουν δυο ειδών δραστηριοτήτων: οι ορατές (messages, share content) και οι σιωπηλές (browse user's profile) [6]. Μια έρευνα [33] παρατήρησε ότι μια μειονότητα των χρηστών δημιουργούσαν την πλειοψηφία των ορατών δραστηριοτήτων. Παρατήρησε επίσης παρατήρησε μια γενική παρακμή στον αριθμό των δραστηριοτήτων μεταξύ ζευγών χρηστών το οποίο υποδηλώνει ότι η δραστηριότητα αλλάζει συνεχώς στην πάροδο του χρόνου, ενώ τα χαρακτηριστικά του κοινωνικού δικτύου παραμένουν σταθερά. Μια άλλη έρευνα [14] έδειξε ότι οι σιωπηλές δραστηριότητες αποτελούν την συντριπτική πλειοψηφία των δραστηριοτήτων μεταξύ χρηστών.

Κεφάλαιο 3

Ανάλυση

3.1 Αναπαράσταση κοινωνικών δικτύων με γράφο

Ένας γράφος είναι ένας απλός τρόπος για να προσδιοριστούν οι σχέσεις ανάμεσα σε μια συλλογή από αντικείμενα. Τα αντικείμενα στον γράφο αντιπροσωπεύονται ως *κόμβοι* και οι σχέσεις ανάμεσα τους ως ακμές. Ένα online social network, όπως το Facebook, μπορεί να αναπαρασταθεί με ένα undirected γράφο $G = (V, E)$, όπου V είναι το σύνολο των κόμβων και E το σύνολο των ακμών. Στην περίπτωση του Facebook, ένας κόμβος αντιπροσωπεύει έναν χρήστη του Facebook και μια ακμή μεταξύ δυο κόμβων αντιπροσωπεύει την φιλία μεταξύ των δυο χρηστών.



Σχήμα 3.1 Παράδειγμα ενός undirected γράφου G=(15,19)

3.2 Μετρικές

Σε αυτό το υποκεφάλαιο θα εξεταστούν οι μετρικές που χρησιμοποιούνται στα πειράματα. Τα παραδείγματα και ο αριθμός κόμβων που αναφέρονται παρακάτω, αφορούν το γράφο στο Σχήμα 3.1.

3.2.1 Density

Το density είναι μια από τις πιο ευρέως χρησιμοποιούμενες μετρικές για την ανάλυση της δομής ενός κοινωνικού δικτύου..

Είναι ο αριθμός των σχέσεων (ακμών) που υπάρχουν σε ένα κοινωνικό δίκτυο σε σχέση με το μέγιστο αριθμό των πιθανών σχέσεων (ακμών). Τυπικά, το density ενός γράφου G με V κόμβους και E ακμές ορίζεται ως:

$$D\ G\ = \frac{2\ E}{V\ (\ V\ -1)}$$

Μεγάλο density σημαίνει καλή συνοχή στο κοινωνικό δίκτυο (μεγάλος αριθμός σχέσεων-ακμών). Π.χ. $D\ G\ = 0.19$

3.2.2 Degree

Το degree ενός κόμβου ορίζεται ως ο αριθμός των σχέσεων του με άλλους κόμβους, με άλλα λόγια, ο αριθμός των ακμών του.

Τυπικά, έστω ότι Nv_i είναι το σύνολο των γειτόνων του κόμβου v_i τότε το degree του v_i ορίζεται ως:

$$Deg\ v_i\ =\ Nv_i$$

Μεγάλο degree κόμβων αποδεικνύει tight δίκτυο διότι οι κόμβοι έχουν μεγάλο αριθμό σχέσεων. Το average degree μπορεί να δώσει μια εικόνα της συνεκτικότητας του κοινωνικού δικτύου. Π.χ. $Deg\ v_{11} = 3$

3.2.3 Degree Distribution

Επειδή μέσα σε ένα κοινωνικό δίκτυο δεν έχουν όλοι οι κόμβοι το ίδιο degree, για να υπολογιστεί η διασπορά (εύρος) του degree των κόμβων, χρησιμοποιείται το degree distribution.

Οπότε το degree distribution $P(k)$ ενός κοινωνικού δικτύου ορίζεται ως το ποσοστό των κόμβων του δικτύου με degree k. Π.χ. $P\ k = 3 = 0.2$

3.2.4 Diameter and Effective Diameter

Η διάμετρος ενός κοινωνικού δικτύου είναι η μεγαλύτερη απόσταση μεταξύ κάθε ζεύγους κόμβων σε ένα δίκτυο, όπου η απόσταση ορίζεται ως το μήκος της συντομότερης διαδρομής μεταξύ των κόμβων.

Η ανάλυση της διαμέτρου σε ένα κοινωνικό δίκτυο μπορεί να δώσει αρκετές σημαντικές πληροφορίες όπως πόσο καλά (ισχυρά) συνδεδεμένο είναι το δίκτυο και πόσο γρήγορα μπορεί να φτάσει μια πληροφορία από τον ένα κόμβο στον άλλο. Αν η διάμετρος είναι χαμηλή, αυτό σημαίνει ότι οι κόμβοι μπορούν να προσεγγίσουν άλλους κόμβους πιο εύκολα.
Π.χ. Diameter του G είναι 5, με το μεγαλύτερο κοντινό μονοπάτι να είναι από τον κόμβο 3 στο κόμβο 13.

Λόγο του ότι υπάρχει πιθανότητα ένας μικρός αριθμός κόμβων να σχηματίσουν μια "αλυσίδα" με αποτέλεσμα να παρεκκλίνει σημαντικά η διάμετρος, χρησιμοποιείται μια "smooth" μορφή αυτής της μετρικής, το effective diameter.

Οπότε, effective diameter ορίζεται ως η ελάχιστη απόσταση κατά την οποία το 90% (90th-percentile) όλων των συνδεδεμένων ζευγών κόμβων μπορούν να φτάσουν ο ένας τον άλλο.

3.2.5 Συντελεστής Clustering (Clustering coefficient)

Ο Συντελεστής coefficient ενός κόμβου είναι η πιθανότητα δυο τυχαίων γειτόνων του κόμβου να είναι φίλοι (συνδεδεμένοι) μεταξύ τους. Με άλλα λόγια είναι το ποσοστό των ζευγών των γειτονικών κόμβων του κόμβου που είναι συνδεδεμένοι μεταξύ τους. [10]

Το clustering coefficient του v_i είναι ο αριθμός των ακμών μεταξύ γειτόνων του κόμβου v_i δια τον αριθμό όλων των πιθανών ακμών μεταξύ τους και ορίζεται ως εξής:

$$cf\ v_i = \frac{2n}{k_i(k_{i-1})}$$

Όπου n είναι ο αριθμός των ακμών μεταξύ φίλων του v_i και k_i είναι ο αριθμός των γειτόνων του v_i ($= \deg(v_i)$).

Π.χ. $cf\ v_7 = 0.33$ και $cf\ v_3 = 1$

Το clustering coefficient του γράφου ορίζεται ως:

$$CC = \frac{3N_\Delta}{N_2}$$

Όπου N_2 είναι το σύνολο των triplet κόμβων που είναι συνδεδεμένοι και N_Δ το σύνολο των triangles κόμβων που είναι εντελώς συνδεδεμένοι μεταξύ τους. Ο αριθμός 3 προκύπτει από το γεγονός ότι το κάθε triangle κόμβων μπορεί να αναπαρασταθεί με τρία διαφορετικά triplets. Π.χ. CC= 0.13

3.2.6 Betweenness Centrality

Όταν δύο κόμβοι δεν είναι άμεσα συνδεδεμένοι, τότε οι αλληλεπιδράσεις μεταξύ τους μπορεί να εξαρτώνται από ένα άλλο κόμβο, ειδικά αν ο συγκεκριμένος κόμβος βρίσκεται στο σύντομο μονοπάτι τους. Οπότε λέγεται ότι όσο πιο συχνά ένας κόμβος βρίσκεται σε αυτή τη θέση τόσο πιο κεντρικά είναι μέσα στο κοινωνικό δίκτυο. Έχει δηλαδή την δύναμη του ελέγχου πληροφορίας και αυτό συλλαμβάνει το betweenness centrality [12].

Betweenness centrality ορίζεται ως ο κανονικοποιημένος (normalized) αριθμός σύντομων μονοπατιών που περνούν από τον κόμβο μέσα στο κοινωνικό δίκτυο. Το betweenness centrality ενός κόμβου v_i είναι:

$$C_B(v_i) = \frac{\frac{\sum_{j<k} g_{jk}(v_i)}{g_{jk}}}{\left[\frac{(V-1)(V-2)}{2}\right]}, i \neq j \neq k$$

Όπου g_{jk} είναι ο αριθμός των μονοπατιών μεταξύ κόμβων v_j και v_k και $g_{jk}(v_i)$ είναι ο αριθμός των μονοπατιών μεταξύ v_j και v_k που περιέχουν τον v_i. (V-1)(V-2) είναι ο μέγιστος αριθμός ζευγών κόμβων που δεν περιέχουν το v_i [34].
Π.χ. $C_B\ v_2\ = 0.47$

Οπότε, το betweenness centrality ενός κόμβου δίνει τον βαθμό ελέγχου που ασκεί ο κόμβος στις αλληλεπιδράσεις άλλων κόμβων μέσα στο κοινωνικό δίκτυο. Διαισθητικά σε ένα κοινωνικό δίκτυο, η μετρική αυτή ευνοεί ιδιαίτερα τους κόμβους που ενώνουν δυο η περισσότερες κοινότητες μέσα στο δίκτυο έναντι κόμβων που βρίσκονται μέσα στις κοινότητες.

3.2.7 Closeness Centrality

Η ιδέα του closeness μέσα σε ένα κοινωνικό δίκτυο είναι η εξής: εάν ένα άτομο είναι κοντά στα άλλα άτομα μέσα στο δίκτυο, τότε αυτό το άτομο μπορεί να αλληλεπιδράσει

πιο γρήγορα με τα υπόλοιπα άτομα μέσα στο δίκτυο. Αυτό είναι το λεγόμενο "independent communication".

Το closeness centrality προσεγγίζει την απόσταση ενός κόμβου σε όλους τους υπόλοιπους κόμβους μέσα στο κοινωνικό δίκτυο, λαμβάνοντας υπόψη και την απόσταση του κάθε άλλου κόμβου με όλους τους υπόλοιπους.

Το closeness centrality ενός κόμβου v_i είναι:

$$C_C(v_i) = \frac{V - 1}{\Sigma_{j=1}^{V} \Sigma d(v_i, v_j)}, i \neq j$$

$d(v_i, v_j)$ είναι η απόσταση μεταξύ v_i και v_j [34].
Π.χ. $C_2\ v_7\ = 0.58$

Οπότε, το closeness centrality ορίζεται ως ο κανονικοποιημένος αριθμός "βημάτων" που χρειάζονται για ένα οποιοδήποτε κόμβο για να έχει πρόσβαση σε όλους τους υπόλοιπους κόμβους μέσα στο κοινωνικό δίκτυο.

Σε γενικές γραμμές, ένας κόμβος που είναι ενωμένος σε άλλους κόμβους με πολλές μικρές αποστάσεις μπορεί να χαρακτηριστεί ως αυτόνομος σε σχέση με όλους τους υπόλοιπους κόμβους που είναι λιγότερο ενωμένοι (με μικρές αποστάσεις).

3.2.8 Network Community Profile Plot

Το network community profile plot μετρά την ποιότητα της καλύτερης δυνατής κοινότητας ως συνάρτηση του μεγέθους της κοινότητας σε ένα κοινωνικό δίκτυο [23]. Για να υπολογιστεί η ποιότητα μιας κοινότητας χρησιμοποιείται η μετρική conductance που μπορεί να θεωρηθεί ως ο αριθμός ακμών που υπάρχουν (δείχνουν) έξω από την κοινότητα δια τον αριθμό των ακμών στο εσωτερικό της κοινότητας. Όσο πιο μικρή είναι η τιμή conductance, τόσο καλύτερη θεωρείται η κοινότητα.

Βάση της μετρικής αυτής, μια καλή κοινότητα είναι αυτή που περιέχει πολλές ακμές μεταξύ των κόμβων της και συνδέεται με το υπόλοιπο δίκτυο μέσω ελάχιστων ακμών.

$$\boldsymbol{\Phi\ k\ = min}_{\boldsymbol{S=k}}\boldsymbol{\varphi\ S}$$

$$\text{όπου } \boldsymbol{\varphi\ S\ } = \frac{\boldsymbol{\#\ ακμές\ έξω\ απο\ την\ κοινότητα}}{\boldsymbol{\#\ αμκές\ μέσα\ στην\ κοινότητα}}$$

3.2.9 Modularity

Το modularity χρησιμοποιείται για την εύρεση κοινοτήτων μέσα σε ένα κοινωνικό δίκτυο. Το modularity Q ενός κοινωνικού δικτύου ορίζεται ως ο αριθμός των ακμών μέσα σε τμήματα του δικτύου *μείον* τον αναμενόμενο αριθμό ακμών μέσα σε τμήματα ενός τυχαίου γράφου [25]. Η τιμές του modularity είναι από το -1 ως το 1. Η τιμή είναι θετική εάν ο αριθμός των ακμών μέσα στις κοινότητες ξεπερνούν τον αναμενόμενο αριθμό. Δηλαδή, ποσοτικοποιεί την ποιότητα ενός δικτύου σε κοινότητες. Ένα καλό δίκτυο που έχει ψηλή τιμή modularity, είναι αυτό που έχει πολλές ακμές μεταξύ κόμβων μέσα στις κοινότητες και ελάχιστες ακμές μεταξύ κοινοτήτων [28].

$$Q = (\#\ ακμών\ μέσα\ σε\ τμήμα\ του\ δικτύου) - (\#\ αναμενόμενων\ ακμών\ μέσα\ σε\ τμήμα\ τυχαίου\ γράφου)$$

Κεφάλαιο 4

Μεθοδολογία

4.1 Σκοπός Κεφαλαίου

Σκοπός του παρόντος κεφαλαίου είναι η περιγραφή της μεθοδολογίας που ακολουθήθηκε για την εκπόνηση της παρούσας έρευνας. Στο πλαίσιο αυτό, θα γίνει μια σύντομη ανασκόπηση των μεθόδων και προσεγγίσεων που είναι κατάλληλες για τις έρευνες και μελέτες αξιολόγησης και ανάλυσης online κοινωνικών δικτύων.

Στη συνέχεια, παρουσιάζονται οι βασικές τεχνικές που είναι στη διάθεση κάθε μεθοδολογίας για τη διενέργεια της ανάλυσης online κοινωνικών δικτύων, με ιδιαίτερη έμφαση στις αυτοματοποιημένες ποσοτικές τεχνικές που προσφέρουν τα εργαλεία αλληλεπίδρασης, ή άλλα αναλυτικά εργαλεία ευρείας χρήσης.

4.2 Στρατηγική της Ερευνητικής Μεθοδολογίας

Για την διενέργεια μιας μελέτης ή ερευνητικής δραστηριότητας είναι απαραίτητη η κατάστρωση μιας στρατηγικής μεθοδολογίας η οποία θα διέπει τη μελέτη. Υπάρχουν

διάφορες μεθοδολογικές προσεγγίσεις ως προς τον τρόπο διενέργειας των μελετών και τη φύση των δεδομένων που είναι διαθέσιμα [39].

4.2.1 Δευτερογενής Έρευνα

Η παρούσα έρευνα είναι *δευτερογενής έρευνα* καθότι τα δεδομένα που χρησιμοποιούνται στην ανάλυση είναι υφιστάμενα δεδομένα τα οποία έχουν συλλεχθεί από τρίτους μέσω διαφόρων μεθόδων και τεχνικών.

Επίσης η μελέτη της σχετικής βιβλιογραφίας και ερευνών θεωρείται μέρος της δευτερογενής έρευνας και πιο γενικά της μεθοδολογίας αφού μέσα από την μελέτη αυτή παρέχονται κατευθυντήριες γραμμές για την διεξαγωγή παρόμοιων πειραμάτων και αναλύσεων για το υπό μελέτη κοινωνικό δίκτυο

Σημειώνεται ότι δεν ήταν δυνατό να εξασφαλιστούν πρωτογενή δεδομένα λόγω της φύσης των δεδομένων που είναι αποθηκευμένα στους servers των πλατφόρμων των online κοινωνικών δικτύων. Πολλά δίκτυα δεν προσφέρουν αυτά τα δεδομένα οπότε πρέπει να εφαρμοστούν μέθοδοι συλλογής δεδομένων όπως web crawling ή click stream τα οποία είναι και αυτά χρονοβόρα και πολύπλοκα. Υπάρχει επίσης ο περιορισμός του χρόνου και του κόστους για να εξασφαλιστούν πρωτογενή δεδομένα, τόσο ποσοτικά όσο και ποιοτικά.

Η απευθείας επαφή με τα μέλη των κοινωνικών δικτύων για την εξασφάλιση ποσοτικών ή ποιοτικών πληροφοριών με τη διεξαγωγή ερωτηματολογίων ή/και συνεντεύξεων δεν ήταν δυνατή λόγω, μεταξύ άλλων, της εμπιστευτικότητας των προσωπικών δεδομένων καθώς και την μεγάλη μάζα των δεδομένων που θα έπρεπε να συλλεχθεί. Επίσης, δεν ήταν εφικτό να γίνει μεγάλης κλίμακας συλλογή δεδομένων μέσο μεθόδων όπως web crawling διότι αυτό θα χρειαζόταν όχι μόνο δυνατούς υπολογιστές και servers αλλά χρόνο και ψηλό οικονομικό κόστος.

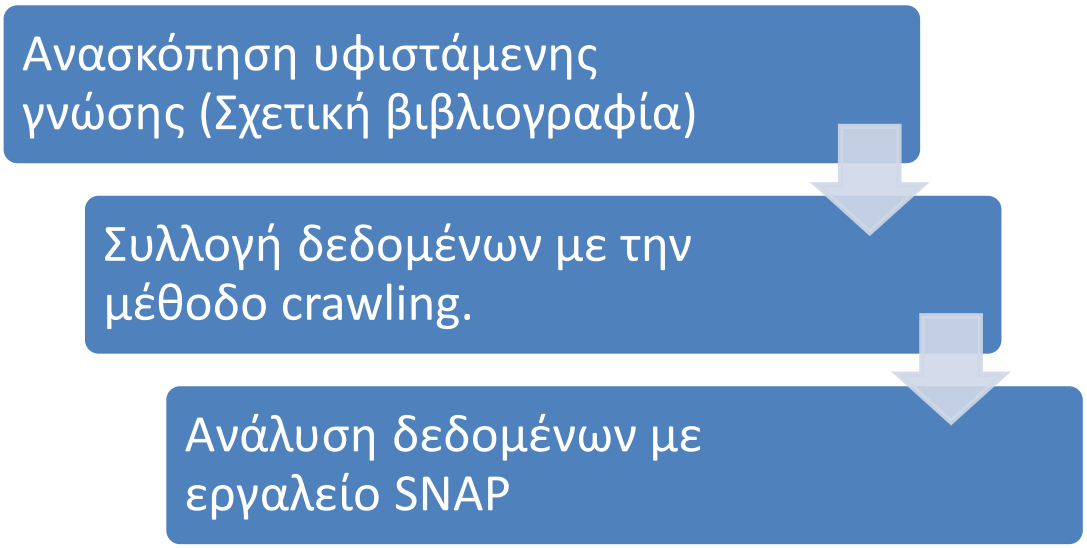
Παρόλο που τα δεδομένα που χρησιμοποιούνται στην έρευνα, είναι δευτερογενής φύσης, είναι αξιόπιστα και έγκυρα καθώς τρίτα άτομα έχουν κάνει μεγάλης κλίμακας

συλλογή δεδομένων και δημοσίευσαν τα δεδομένα αυτά προς το ευρύ κοινό. Τα δεδομένα αυτά περιγράφονται αναλυτικά στο υποκεφάλαιο 4.3.1

4.2.2 Ποσοτικές Μέθοδοι Ανάλυσης

Η παρούσα έρευνα χρησιμοποιεί *ποσοτικές μεθόδους*. Οι ποσοτικές μέθοδοι διερευνούν τις συσχετίσεις μεταξύ διαφόρων μεταβλητών και, αν είναι δυνατόν, επιχειρούν να ανακαλύψουν σχέσεις αιτίου - αιτιατού μεταξύ τους. Ως *μεταβλητή* ορίζεται κάποιο παρατηρήσιμο χαρακτηριστικό μιας μονάδας, οι τιμές του οποίου μπορεί να ποικίλλουν ανάμεσα σε διαφορετικές μονάδες ενός συνόλου. Οι ποσοτικές μεταβλητές παίρνουν αριθμητικές τιμές (συνεχείς ή διακριτές).

Οι ποσοτικές τεχνικές που παρουσιάζονται στην παρούσα μελέτη θεωρούνται κατάλληλες για την ανάλυση της δομής των online κοινωνικών δικτύων και την αλληλεπίδραση των χρηστών. Επεξεργάζονται πληροφορίες που αφορούν ενέργειες και ιδιότητες ξεχωριστών ατόμων, έστω και αν σε πολλές περιπτώσεις τα αποτελέσματα της ανάλυσης προκύπτουν από επεξεργασία συνδυασμού στοιχείων που αναφέρονται σε διαφορετικά άτομα. Περαιτέρω, οι τεχνικές αυτές καλύπτουν τον ορισμό συγκεκριμένων κριτηρίων για τη μέτρηση των σχέσεων μεταξύ των διαφορετικών ατόμων που συνεργάζονται σε ένα online κοινωνικό δίκτυο.



Σχήμα 4. 1: Διάγραμμα ροής μεθοδολογίας

4.3 Τεχνικές Συλλογής και Ανάλυσης Δεδομένων

Για την υποστήριξη της διεξαγωγής της έρευνας που ακολουθεί την ποσοτική μεθοδολογία υπάρχουν διάφορες τεχνικές[2] και στατιστικά εργαλεία για τον υπολογισμό των συσχετίσεων μεταξύ μεταβλητών. Δηλαδή, μετά τον καθορισμό της μεθοδολογικής προσέγγισης σε μια έρευνα είναι απαραίτητο να χρησιμοποιηθούν διάφορες τεχνικές για την άντληση της γνώσης από το υπό θεώρηση περιβάλλον.

4.3.1 Συλλογή δεδομένων

Τα δεδομένα που έχουν χρησιμοποιηθεί στην εργασία έχουν συλλεχθεί από τους Viswanath et al.(2009) [33] για την έρευνα τους την οποία παρουσίασαν στο 2^{ο} ACM SIGCOMM Workshop για τα κοινωνικά δίκτυα [36].

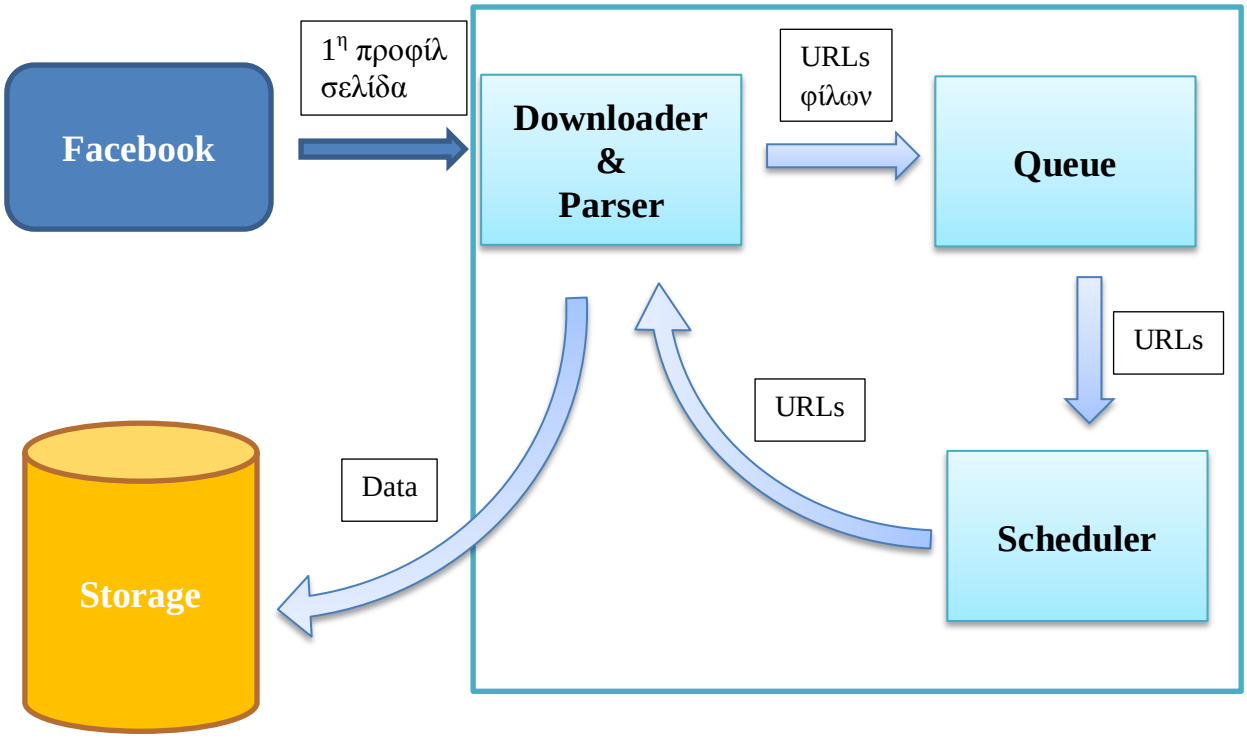
Η συλλογή τους έγινε με την μέθοδο web crawling του τοπικού δικτύου της Νέας Ορλεάνης του Facebook. Δημιουργήθηκε ένας αριθμός λογαριασμών του Facebook οι οποίοι εντάχθηκαν στο τοπικό δίκτυο και έκαναν crawl τις προφίλ σελίδες των χρηστών που ανήκουν στο τοπικό δίκτυο.

Θα πρέπει να σημειωθεί ότι υπάρχουν τα ονομαζόμενα λογισμικά web crawlers τα οποία επισκέπτονται συγκεκριμένες διευθύνσεις URL και τις αποθηκεύουν στον υπολογιστή για περαιτέρω επεξεργασία. Παράλληλα καθώς επισκέπτονται αυτές τις διευθύνσεις, αναγνωρίζουν της υπερσυνδέσεις στη σελίδα και τους προσθέτουν σε ένα κατάλογο διευθύνσεων URL και στη συνέχεια τις επισκέπτονται και αυτές αναδρομικά με βάση ένα σύνολο κανόνων που ορίζονται από τον χρήστη. Στο σχήμα παρουσιάζεται συνοπτικά πως λειτουργεί ένα λογισμικό web crawler (πιο παρακάτω περιγράφεται πιο αναλυτικά το σχήμα).

Οι σελίδες προφίλ περιέχουν τα στοιχεία του χρήστη, την λίστα φίλων καθώς και το "wall" το οποίο είναι μια μορφή αλληλεπίδρασης χρηστών. Οι φίλοι του χρήστη

[2] Οι τεχνικές διαφέρουν από τις μεθόδους. Οι τεχνικές χρησιμοποιούνται για την καταγραφή και ανάλυση των δεδομένων που είναι χρήσιμα σε μια μελέτη, ενώ οι μέθοδοι χρησιμεύουν στο γενικό σχεδιασμό μιας μελέτης, αν και στη βιβλιογραφία πολλές φορές οι δύο όροι χρησιμοποιούνται εναλλακτικά.

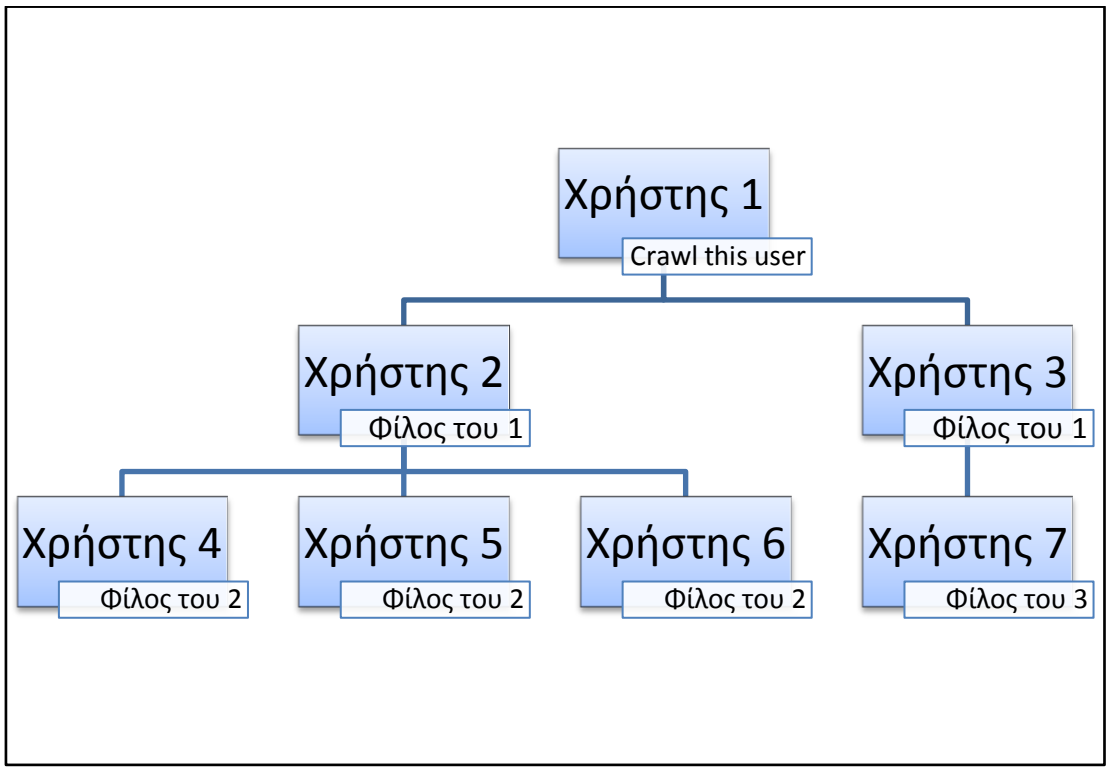
μπορούν να δημοσιεύσουν σχόλια πάνω στο wall του χρήστη. Τα σχόλια αυτά μπορούν να δουν όλοι όσοι επισκέπτονται τη *σελίδα προφίλ* του χρήστη



Σχήμα 4. 2: Παράδειγμα λειτουργίας τυπικού λογισμικού web crawler

Η συλλογή των δεδομένων έγινε σε δύο χρονικές περιόδους. Η πρώτη έγινε μεταξύ Δεκεμβρίου 2009 και Ιανουαρίου 2009, όπου συλλέχθηκαν πληροφορίες σχετικά με τις σχέσεις φιλίας μεταξύ χρηστών. Το λογισμικό crawler ξεκινά από ένα χρήστη του τοπικού δικτύου της Νέας Ορλεάνης, και επισκέπτεται όλους τους φίλους του χρήστη με την μέθοδο breadth-first-search (BFS). Στο σχήμα φαίνεται καθαρά η διαδικασία συλλογής με την μέθοδο crawling. Κατεβάζεται η πρώτη *προφίλ σελίδα* ενός χρήστη και τίθεται υπό επεξεργασία. Οποιαδήποτε χρήσιμα δεδομένα φυλάγονται στον αποθηκευτικό χώρο. Ο crawler αναλύει την *σελίδα προφίλ* και βρίσκει τις διευθύνσεις URL των φίλων του χρήστη και τα βάζει σε μια ουρά. Ο χρονοπρογραμματιστής (scheduler) σε συγκεκριμένα χρονικά διαστήματα (τα οποία ορίζονται από τον χρήστη) παίρνει το πρώτο URL από την ουρά, κατεβάζει την σελίδα και την επεξεργάζεται. Αυτό συνεχίζεται αναδρομικά ώσπου να αδειάσει η ουρά ή να τερματιστεί το πρόγραμμα από τον χρονοπρογραμματιστή.

Στο σχήμα φαίνεται η μέθοδος breadth-first-search και δείχνει την σειρά με την οποία εξετάζονται οι κόμβοι (πρώτα ο χρήστης 1, μετά ο 2, ο 3 και ούτω καθεξής). Η μέθοδος ξεκινά από την ρίζα και εξετάζει όλους τους γειτονικούς κόμβους (τους φίλους) που συνδέονται με την ρίζα. Στη συνέχεια εξετάζει τους γειτονικούς κόμβους που συνδέονται με τον προηγούμενο γειτονικό κόμβο. Η διαδικασία αυτή συνεχίζεται μέχρι να εξεταστούν όλοι οι κόμβοι.



Σχήμα 4. 3 Παράδειγμα μεθόδου breadth-first-search

Η δεύτερη συλλογή έγινε μεταξύ 20 Ιανουαρίου και 22 Ιανουαρίου 2009, όπου συλλέχθηκαν πληροφορίες σχετικά με το "wall" των χρηστών που ανακαλύφθηκαν στην πρώτη περίοδο συλλογής. Δηλαδή κάθε καταχώρηση αντιστοιχεί με μια δημοσίευση σε wall και περιέχει πληροφορίες σχετικά με το ποιος έκανε τη δημοσίευση, σε ποιόν την έκανε, την ημερομηνία που έγινε η δημοσίευση καθώς και το περιεχόμενό της.

Αποτέλεσμα της συλλογής δεδομένων με την μέθοδο του crawling είναι ένα dataset που περιέχει πληροφορίες 90,269 χρηστών και 1,823,331 σχέσεων φιλίας μεταξύ των

χρηστών. Με βάση τις στατιστικές του Facebook για το τοπικό δίκτυο, το dataset καλύπτει το 52% των χρηστών που ανήκουν στο τοπικό δίκτυο. Σημειώνεται όμως ότι μόνο το 66.7% των χρηστών (63,731 χρήστες) είχαν το προφίλ τους προσβάσιμο προς το κοινό. Οπότε, για την ανάλυση των δεδομένων χρησιμοποιείται το υποσύνολο των 63,731 χρηστών και των 817,090 σχέσεων φιλίας μεταξύ τους.

Η συλλογή δεδομένων παρουσιάζει κάποιους περιορισμούς. Με την μέθοδο του crawling ήταν δυνατή η συλλογή δεδομένων των χρηστών που είχαν δημοσιοποιήσει την *προφίλ σελίδα* τους στα άτομα που ανήκουν στο ίδιο τοπικό δίκτυο. (Τα στοιχεία όμως είναι αντιπροσωπευτικά επειδή τα δεδομένα καλύπτουν την πλειοψηφία του δικτύου, 66,7%). Επίσης η συλλογή δεδομένων ήταν δυνατή μόνο στο giant connected component του δικτύου (δηλαδή κάθε στοιχείο του δικτύου μπορεί να φτάσει σε κάθε άλλο στοιχείο μέσω ενός μονοπατιού) αλλά βάση προηγούμενης έρευνας [38] ένα giant connected component τείνει να περιέχει αρκετά μεγάλη πλειοψηφία των χρηστών.

Αξίζει να σημειωθεί ότι τα δεδομένα που συλλέχθηκαν επεξεργάστηκαν ούτως ώστε να διατηρηθεί η ανωνυμία για την προστασία της ιδιωτικής ζωής των ίδιων των χρηστών (π.χ. Οι χρήστες αναπαριστούνται ως αριθμοί). Επίσης έχουν συλλεχθεί στοιχεία για όσους χρήστες έχουν το προφίλ τους δημόσιο προς το κοινό. Με άλλα λόγια, δεν ήταν δυνατή η συλλογή στοιχείων για όσους χρήστες έχουν κάνει το προφίλ να εμφανίζεται μόνο σε φίλους (ή μερίδα ατόμων ή σε κανέναν).

4.3.2 Ανάλυση δεδομένων με το εργαλείο SNAP

Επόμενο βήμα είναι η ανάλυση των δεδομένων με το εργαλείο SNAP και η πειραματική αξιολόγησή τους. Το SNAP δίνει την δυνατότητα ανάγνωσης των δεδομένων και δημιουργία του γράφου του κοινωνικού δικτύου, όπου κάθε κόμβος αναπαριστά ένα χρήστη και κάθε ακμή την σχέση φιλίας μεταξύ δυο χρηστών. Στη συνέχεια εφαρμόζονται διάφορες γράφο-θεωρητικές έννοιες (όπως clustering coefficient, density κτλ) και αλγόριθμοι (όπως modularity) πάνω στο γράφο και εξάγονται πληροφορίες για ανάλυση.

Το εργαλείο SNAP (Stanford Network Analysis Platform) είναι μια γενικής χρήσης βιβλιοθήκη για ανάλυση δικτύων γραμμένη στην C++, το οποίο αναπτύχθηκε από τον Jure Leskovec το 2004, για σκοπούς ανάλυσης τεράστιων κοινωνικών και πληροφοριακών δικτύων. Είναι βασισμένο σε μεθόδους αναδρομής κόμβων και ακμών οι οποίοι επιτρέπουν την γρήγορη διάσχιση κόμβων ή ακμών καθώς και την αποτελεσματική υλοποίηση αλγορίθμων που λειτουργούν στα δίκτυα, ανεξάρτητα του τύπου τους (directed, undirected κτλ).

Το εργαλείο SNAP αναπτύσσεται συνεχώς και σε κάθε καινούργια έκδοση διορθώνονται τυχόν λάθη στο κώδικα καθώς και προστίθενται νέοι αλγόριθμοι και μέθοδοι στην βιβλιοθήκη. Στην τελευταία έκδοση (17 Απριλίου, 2011) το SNAP υποστηρίζει undirected, directed και multi-directed (πολλές ακμές μεταξύ δυο κόμβων) γράφους και περιέχει νέες μεθόδους για εύρεση overlapped κοινοτήτων. Παρόλο που το SNAP αποθηκεύει πληροφορίες σχετικά με τη διασυνδεσιμότητα του δικτύου, επιτρέπει επίσης την αποθήκευση τιμών στις ακμές και στους κόμβους, κάτι το οποίο μπορεί να επιτρέψει την υλοποίηση αλγορίθμων για weighted γράφους (γράφοι οι οποίοι οι ακμές έχουν μια τιμή αξίας/σημαντικότητας).

Επίσης έχει την δυνατότητα όχι μόνο να διαβάζει διάφορους τύπους αρχείων για φόρτωση δικτύων από datasets αλλά και να αποθηκεύει γράφους σε αρχεία. Επίσης προσφέρει απλές μεθόδους για τον αποτελεσματικό υπολογισμό χαρακτηριστικών των δικτύων, όπως το degree distribution, clustering coefficient, diameter και τα λοιπά. Εκτός από τον υπολογισμό χαρακτηριστικών των δικτύων, προσφέρει και έτοιμους αλγόριθμους που έχουν μελετηθεί σε σχετικές έρευνες, όπως ο αλγόριθμος του Girvan-Newman.

Το εργαλείο συνεργάζεται άψογα με τρίτα προγράμματα όπως το gnuplot (το οποίο δημιουργεί γραφικές παραστάσεις) και προσφέρει μεθόδους οι οποίες δημιουργούν γραφικές παραστάσεις για διάφορες μετρικές. Επίσης υπάρχουν προγράμματα όπως το NodeXL της Excell τα οποία ενσωματώνουν την βιβλιοθήκη του SNAP, και εκμεταλλεύονται έτσι τις δυνατότητες του.

Παρόλο που οι δυνατότητες του SNAP είναι απεριόριστες και προσφέρει πληθώρα μεθόδων για υπολογισμό μετρικών και αλγορίθμων, για αυτή την έρευνα έχει χρησιμοποιηθεί ένας συγκεκριμένος αριθμός μεθόδων οι οποίες θεωρήθηκαν ως οι πιο χρήσιμες και καταλληλότερες. Πιο κάτω γίνεται συνοπτική αναφορά των μεθόδων που έχουν χρησιμοποιηθεί στα πλαίσια της έρευνας:

Χαρακτηριστικά δικτύου

- Density
- Degree distribution
- Clustering Coefficient
- Diameter

Centrality κόμβων

- Degree
- Betweenness
- Closeness

Αλγόριθμοι για εύρεση κοινοτήτων

- Network community profile plot
- Modularity

Όπως έχει ήδη αναφερθεί το SNAP παρέχει πληθώρα λειτουργιών και μεθόδων γι αυτό και στο Παράρτημα Α δίνεται το εγχειρίδιο του εργαλείου SNAP το οποίο περιέχει σύντομη περιγραφή όλων των λειτουργιών.

Κεφάλαιο 5

Πειραματική αξιολόγηση

5.1 Σύνολο δεδομένων

Τα σύνολο δεδομένων (dataset) είναι μια συλλογή από δεδομένα που αφορούν τις σχέσεις μέσα σε ένα κοινωνικό δίκτυο. Συνήθως είναι σε μορφή πίνακα όπου υπάρχουν δυο στήλες κόμβων και κάθε γραμμή αντιπροσωπεύει μια σχέση μεταξύ δυο κόμβων. Επίσης σε αρκετά dataset υπάρχει και μια επιπλέον στήλη που αντιπροσωπεύει την ημερομηνία ίδρυσης της κάθε σχέσης.

Τα σύνολα δεδομένων εισάγονται σε ένα εργαλείο το οποίο δημιουργεί την αναπαράσταση του κοινωνικού δικτύου σε γράφο και στη συνέχεια εκτελούνται πειράματα.

Για τα πειράματα, χρησιμοποιείται το WOSN 2009 dataset το οποίο περιγράφεται στο παρακάτω υπο-κεφάλαιο.

5.1.1 WOSN2009 Dataset

Το WOSN 2009 έχει χρησιμοποιηθεί για το ACM SIGCOMM Workshop που έγινε το 2009 [36] [1] και πιο συγκεκριμένα στην έρευνα των Viswanath *et al* [33]. Το σύνολο δεδομένων αντιπροσωπεύει το τοπικό δίκτυο της Νέας Ορλεάνης στο Facebook. Η συλλογή των δεδομένων έγινε με την μέθοδο crawling των προφίλ των χρηστών που ανήκουν σ' αυτό το τοπικό δίκτυο, συλλέγοντας πληροφορίες σχετικά με τις φιλίες τους με άλλους χρήστες. Αυτό είχε ως αποτέλεσμα να συλλεχθούν πληροφορίες για 63,731 χρήστες και για 817,090 διασυνδέσεις μεταξύ τους . Λεπτομέρειες σχετικά με την συλλογή τους αναφέρονται στο Κεφάλαιο 4.

5.1.2 Περιορισμοί

Λόγω της προστασίας απορρήτου, δεν ήταν δυνατή η συλλογή πληροφοριών για όλους τους χρήστες του δικτύου. Κάποιοι χρήστες είχαν αυστηρές ρυθμίσεις απορρήτου, με αποτέλεσμα να μην είναι δυνατόν η παρουσίαση των προφίλ τους ή των φίλων τους. Περισσότερα για τους περιορισμούς αναφέρονται στο Κεφάλαιο 4.

5.2 Αποτελέσματα

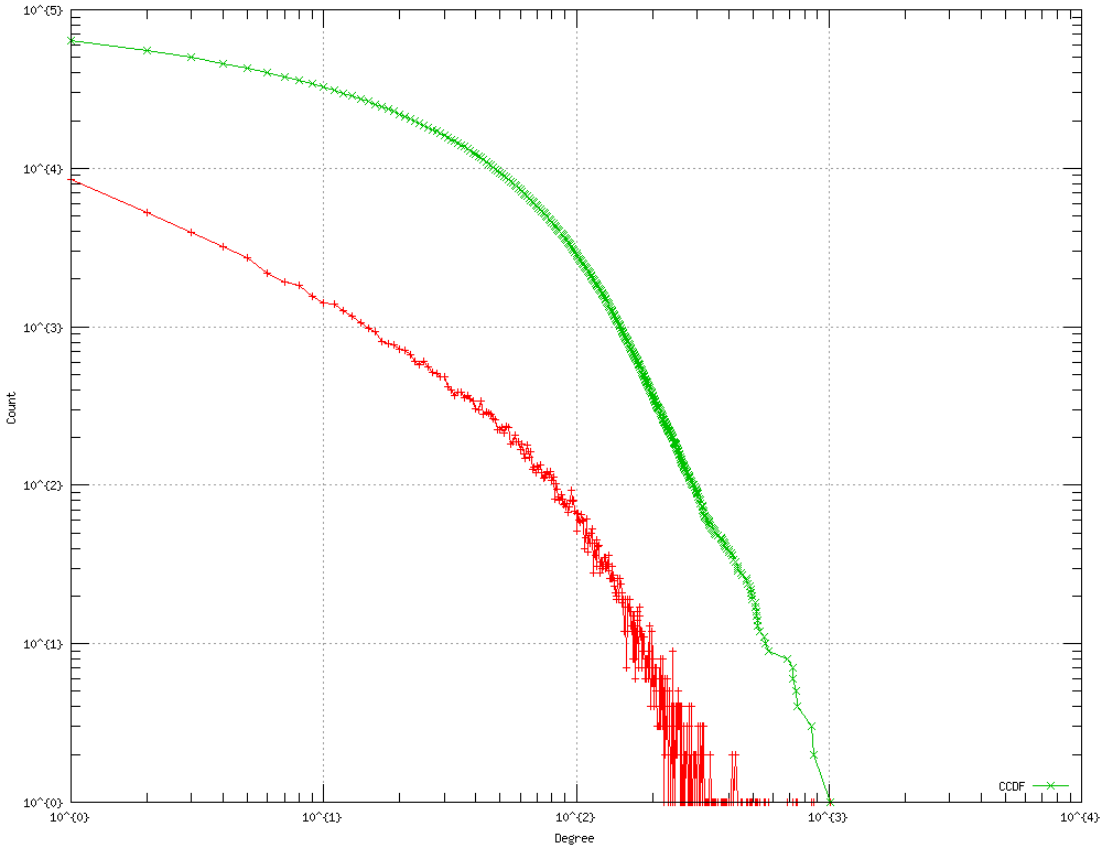
5.2.1 Density και Degree Distribution

Το density του δικτύου είναι 0.0007, παρατηρείται ότι υπάρχει πολύ μικρός αριθμός ακμών σε σχέση με τον μέγιστο αριθμό των ακμών που μπορούν να υπάρχουν μέσα στο δίκτυο. Το χαμηλό density είναι κάτι το κοινό και παρουσιάζεται σε μεγάλα κοινωνικά δίκτυα. Το density είναι αντιστρόφως ανάλογο του μεγέθους του κοινωνικού δικτύου. Όσο πιο μεγάλο είναι το δίκτυο, τόσο πιο χαμηλό είναι το density. Ο λόγος είναι γιατί όταν προστίθενται καινούργιοι κόμβοι, ο μέγιστος αριθμός των σχέσεων αυξάνεται κατά πολύ, και ένα άτομο μπορεί να διατηρήσει περιορισμένο αριθμό σχέσεων. Αυτός είναι ένας σημαντικός περιορισμός και μπορεί να εξηγηθεί εύκολα. Για το πιο μεγάλο online κοινωνικό δίκτυο (Facebook, με 750 εκ. χρήστες), ο μέσος χρήστης έχει 130 φίλους[3], καθώς υπάρχει και ένα όριο από το ίδιο online κοινωνικό δίκτυο στο πόσους φίλους μπορεί να έχει ένας χρήστης (στο Facebook, ο μέγιστος αριθμός φίλων που μπορεί να έχει ένας χρήστης είναι 5000)

[3] Τα στατιστικά για το Facebook πάρθηκαν από την σελίδα www.facebook.com/press/info.php?statistics

Για τον λόγο αυτό η μετρική αυτή δεν μπορεί να χρησιμοποιηθεί αποτελεσματικά για σύγκριση μεταξύ δυο κοινωνικών δικτύων με διαφορετικό μέγεθος. Για τέτοιου είδους συγκρίσεις μεταξύ δυο η περισσότερων κοινωνικών δικτύων που διαφέρουν σε μέγεθος, εξετάζεται το degree.

Όπως έχει αναφερθεί στις ενότητες 3.3.3 και 3.3.4, degree είναι ο αριθμός των ακμών ενός κόμβου και degree distribution P(k) ο αριθμός τον κόμβων με degree k. Το σχήμα 5.1 δείχνει το degree distribution του online κοινωνικού δικτύου. Το μέσο degree είναι 48.5. Ποσοστό 15% των κόμβων έχουν μεγαλύτερο degree από το μέσο (με μόνο το 4% να έχει περισσότερο από το διπλάσιο του average). Έχει παρατηρηθεί λοιπόν, ότι το μεγαλύτερο ποσοστό των κόμβων (80%) έχει χαμηλό degree και λίγοι κόμβοι έχουν αρκετά υψηλό degree. Επίσης το κοινωνικό δίκτυο φαίνεται να παρουσιάζει power law distribution αφού το degree distribution ακολουθεί μια εκθετική φθίνουσα ουρά.

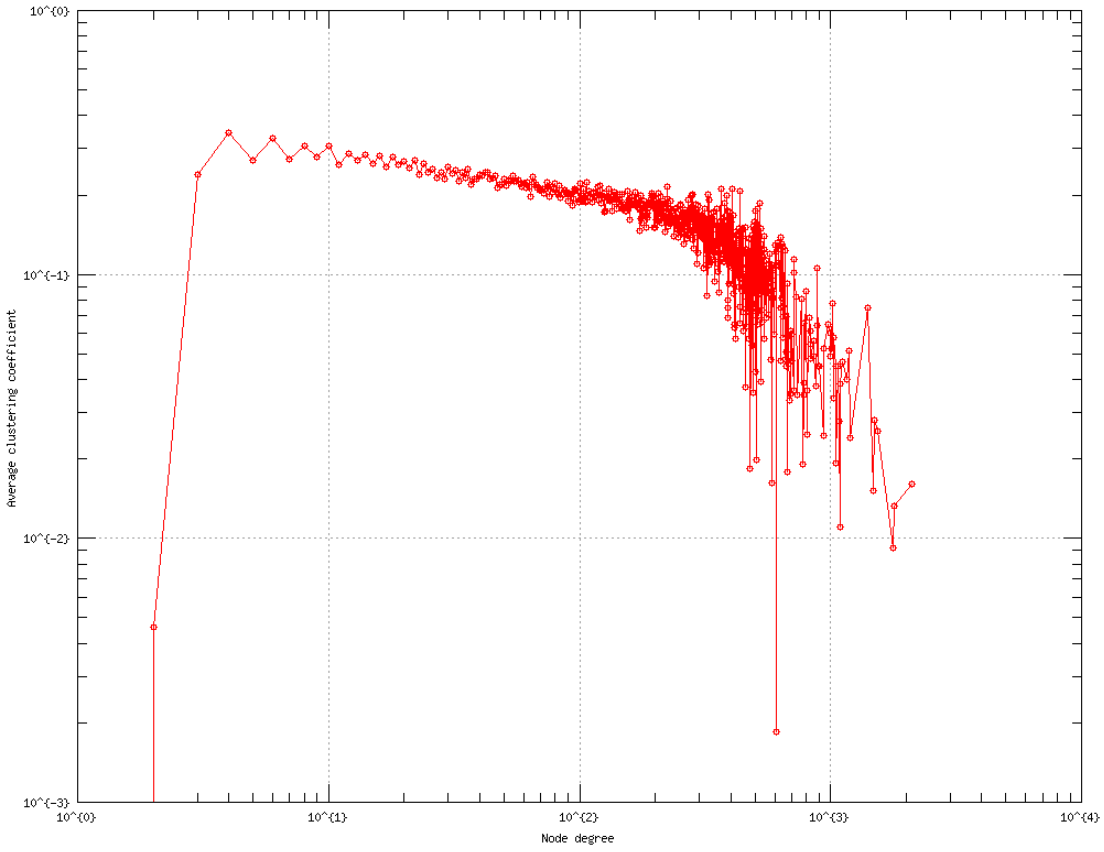


Σχήμα 5.1 - Degree Distribution του δικτύου

Με βάση την σχετική βιβλιογραφία, αρκετά είδη δικτύων παρουσιάζουν power law distribution οπότε εξηγείται το γεγονός ότι και το κοινωνικό δίκτυο της Νέας Ορλεάνης ακολουθεί power law distribution.

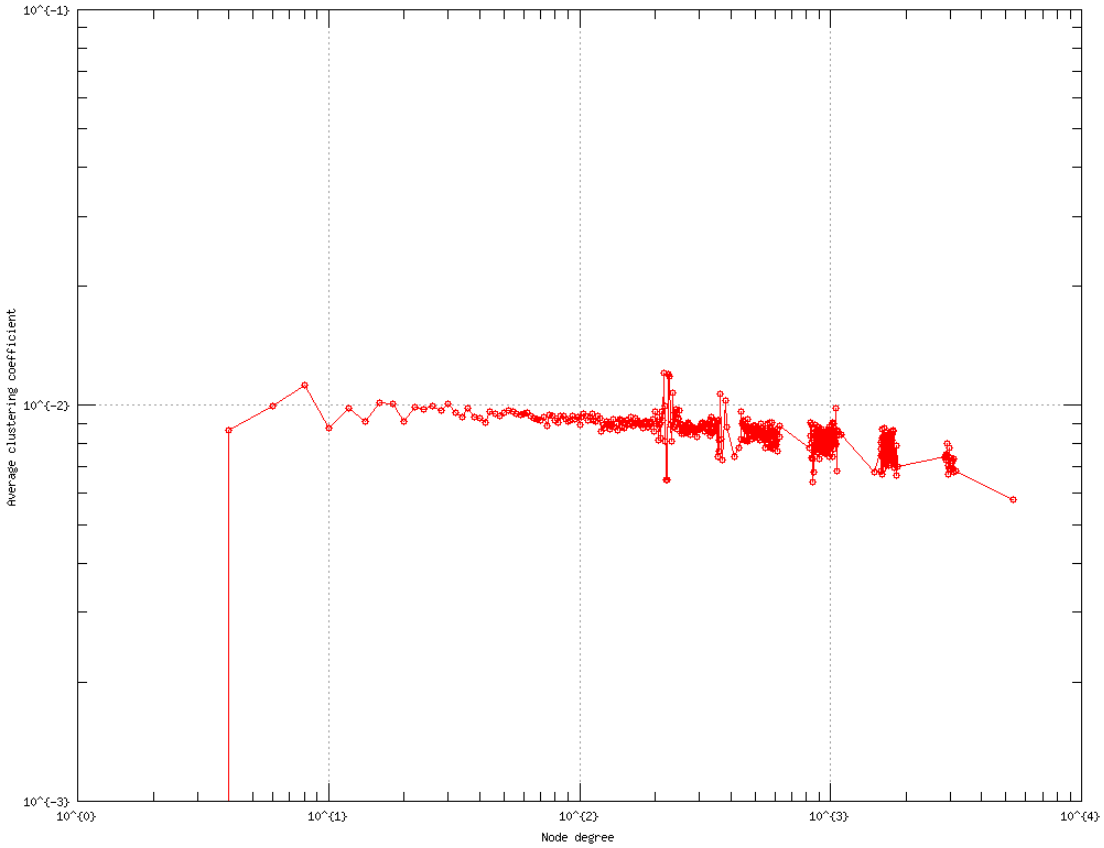
5.2.2 Clustering Coefficient

Το Clustering coefficient μετρά το "cliquiness" του δικτύου. Το κοινωνικό δίκτυο της Νέας Ορλεάνης παρουσιάζει average clustering coefficient 0.221. Στο σχήμα 5.2 παρατηρείται ότι οι κόμβοι με χαμηλό degree παρουσιάζουν μεγαλύτερο clustering coefficient από τους κόμβους με υψηλό degree. Αυτό σημαίνει ότι υπάρχει υψηλό "cliquiness" (clustering) μεταξύ κόμβων με χαμηλό degree (Αυτό σημαίνει ότι οι κόμβοι με χαμηλό degree είναι στενά



Σχήμα 5.2 - Clustering Coefficient του δικτύου

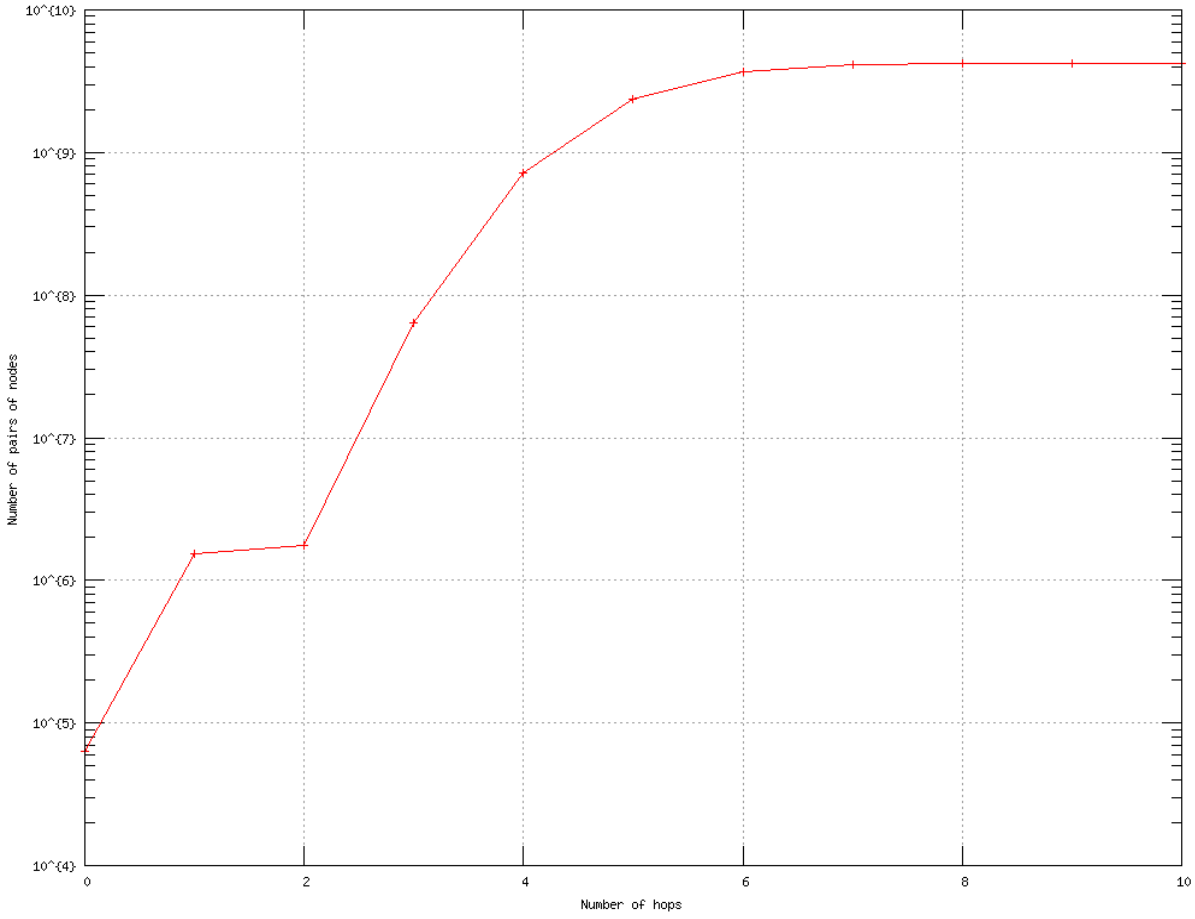
συνδεδεμένοι μεταξύ τους). Για να γίνουν περαιτέρω παρατηρήσεις, θα γίνει σύγκριση του κοινωνικού δικτύου με το αντίστοιχο knonecker τυχαίο γράφο του, που περιέχει σχεδόν τον ίδιο αριθμό κόμβων και ακμών. Παρατηρείται λοιπόν ότι το κοινωνικό δίκτυο της Νέας Ορλεάνης έχει clustering coefficient 24 φορές πιο μεγάλο από τον τυχαίο γράφο και αυτό σημαίνει ότι υπάρχει υψηλό clustering στο network. Αυτό συμβαίνει διότι οι άνθρωποι έχουν την τάση να γνωρίζουν άλλους ανθρώπους μέσω κοινών φίλων (το οποίο αυξάνει την πιθανότητα ότι δύο φίλοι ενός χρήστη να είναι επίσης φίλοι).



Σχήμα 5.3 - Clustering Coefficient του τυχαίου γράφου

5.2.3 Diameter

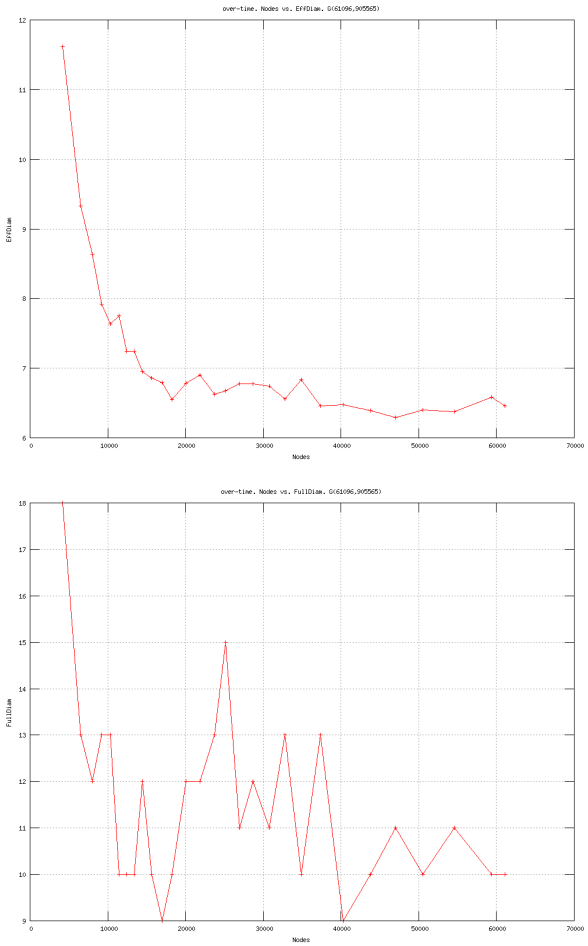
Σε αυτή τη ενότητα παρουσιάζονται και αναλύονται οι μετρήσεις του diameter, το οποίο είναι το μεγαλύτερο κοντινό μονοπάτι μεταξύ όλων των ζευγών κόμβων στο κοινωνικό δίκτυο, με βάση τον αριθμό των ζευγών κόμβων καθώς και την ενναλαγή του στην πάροδο του χρόνου. Στο Σχήμα 5.4 παρατηρείται ότι, ένας μέγιστος αριθμός 8 βημάτων είναι αναγκαίος για να προσεγγιστεί ένας οποιοσδήποτε κόμβος σε κάθε άλλο κόμβο και για το 50% των ζευγών χρειάζεται ένας αριθμός μεταξύ 2 και 3 βημάτων. Το effective diameter είναι 6.2 και παρατηρείται ότι είναι σχετικά μικρό σε σύγκριση με το effective diameter του Web που είναι 8.1 (diameter: 25) [10].



Σχήμα 5.4 – Διάμετρος του δικτύου

Στη συνέχεια εξετάζεται η αλλαγή της διαμέτρου στην πάροδο του χρόνου. Βάση της ημερομηνίας δημιουργίας μιας κοινωνικής σχέσης, για κάθε 30 μέρες δημιουργείται

ένα στιγμιότυπο του dataset, με αποτέλεσμα να έχουμε σύνολο 29 στιγμιότυπων. Στο Σχήμα 5.5 παρατηρείται ένα αρχικά μεγάλο diameter αλλά όσο περνά ο χρόνος και προστίθενται στο κοινωνικό δίκτυο καινούργιοι κόμβοι, παρατηρείται μείωση και σταθεροποίηση του diameter. Το χαμηλό diameter και το υψηλό clustering coefficient χαρακτηρίζει το υπό εξέταση κοινωνικό δίκτυο (Facebook της Νέας Ορλεάνης) ως small-world δίκτυο.

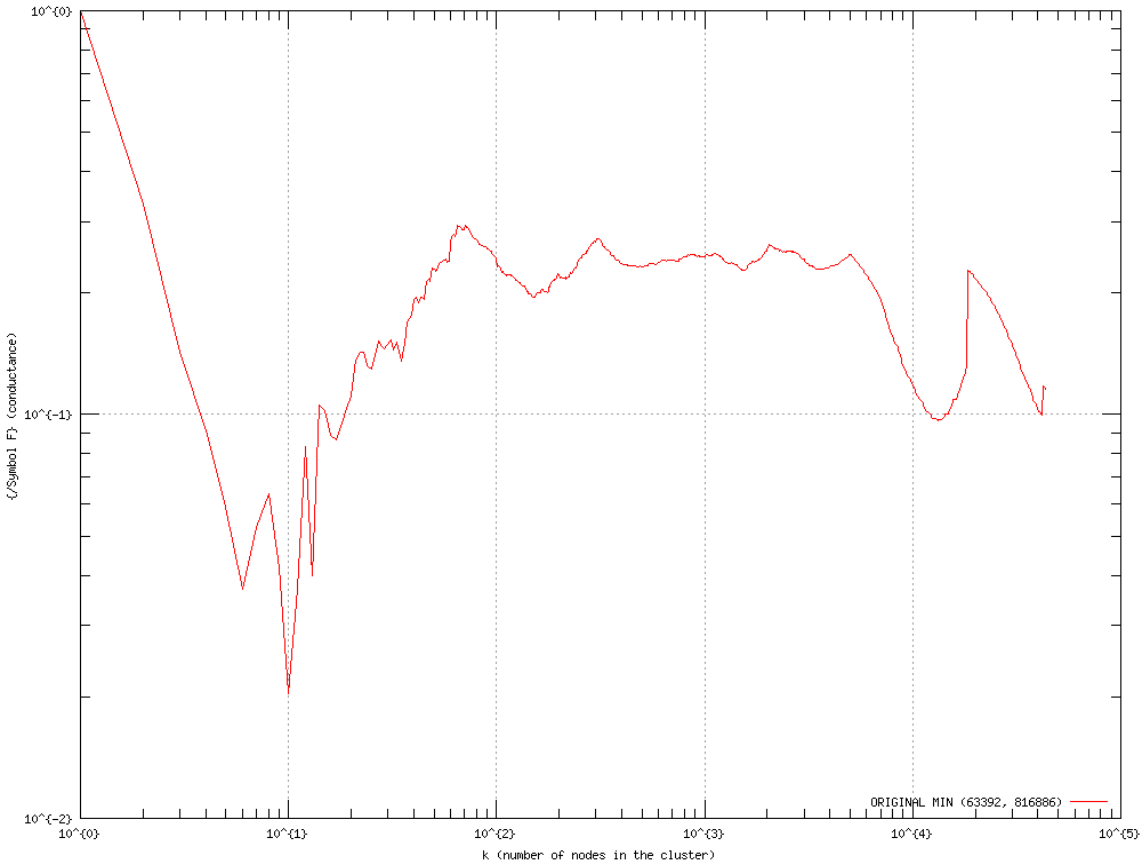


Σχήμα 5.5 Διάμετρος του κοινωνικού δικτύου στην πάροδο του χρόνου

5.2.4 Network Community Profile Plot

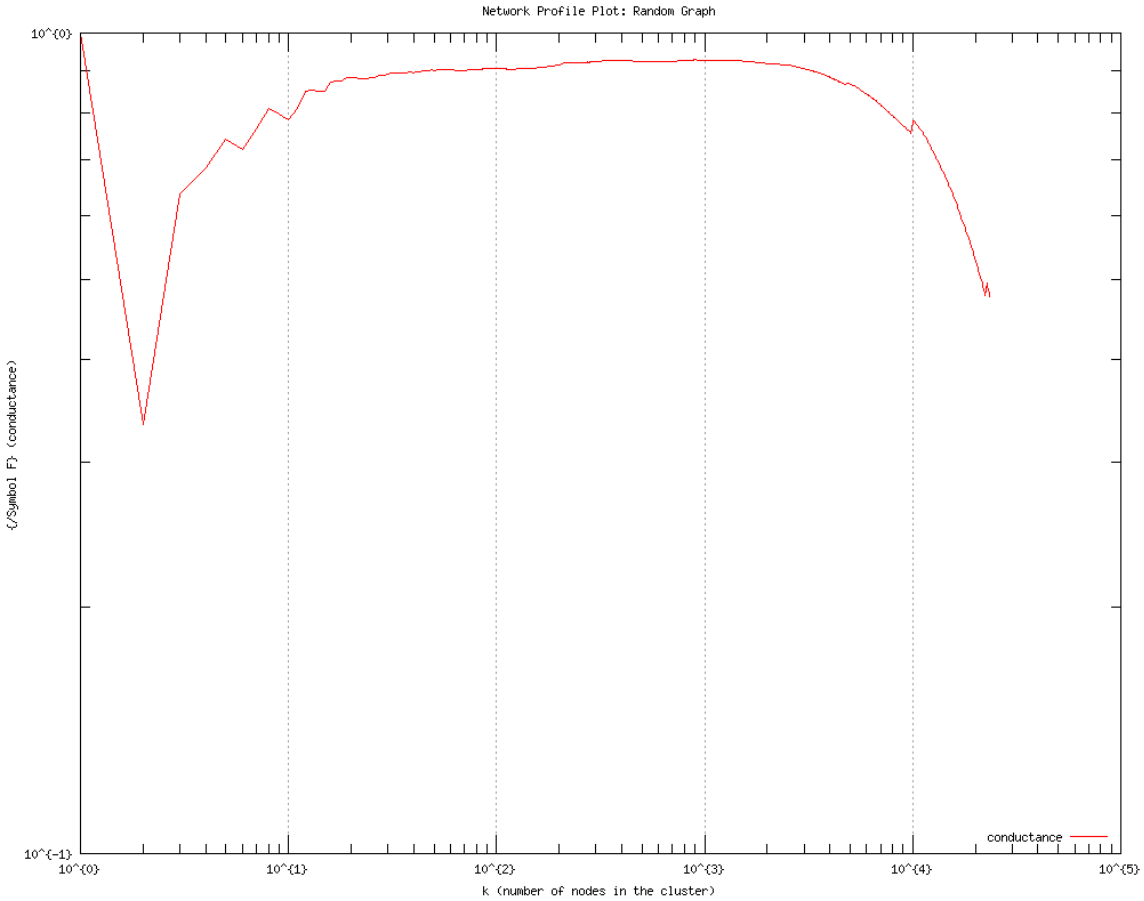
Θα πρέπει να σημειωθεί ότι το γράφημα (plot) του network community profile (NCP) υπολογίζει την ποιότητα της καλύτερης δυνατής κοινότητας ως συνάρτηση του μεγέθους της κοινότητας. Στο Σχήμα 5.6 φαίνεται ότι μέχρι τους 100 κόμβους παρατηρείται μια φθίνουσα κλίση που υποδηλώνει καλή ποιότητα κοινοτήτων στους

100 κόμβους. Από τους 100 κόμβους και μετά αυξάνεται το NCP που σημαίνει ότι οι κοινότητες ενώνονται περισσότερο με το υπόλοιπο δίκτυο, δηλαδή υπάρχουν περισσότερες ακμές που συνδέουν την κοινότητα με το δίκτυο.



Σχήμα 5.6 Network Community Profile Plot του δικτύου

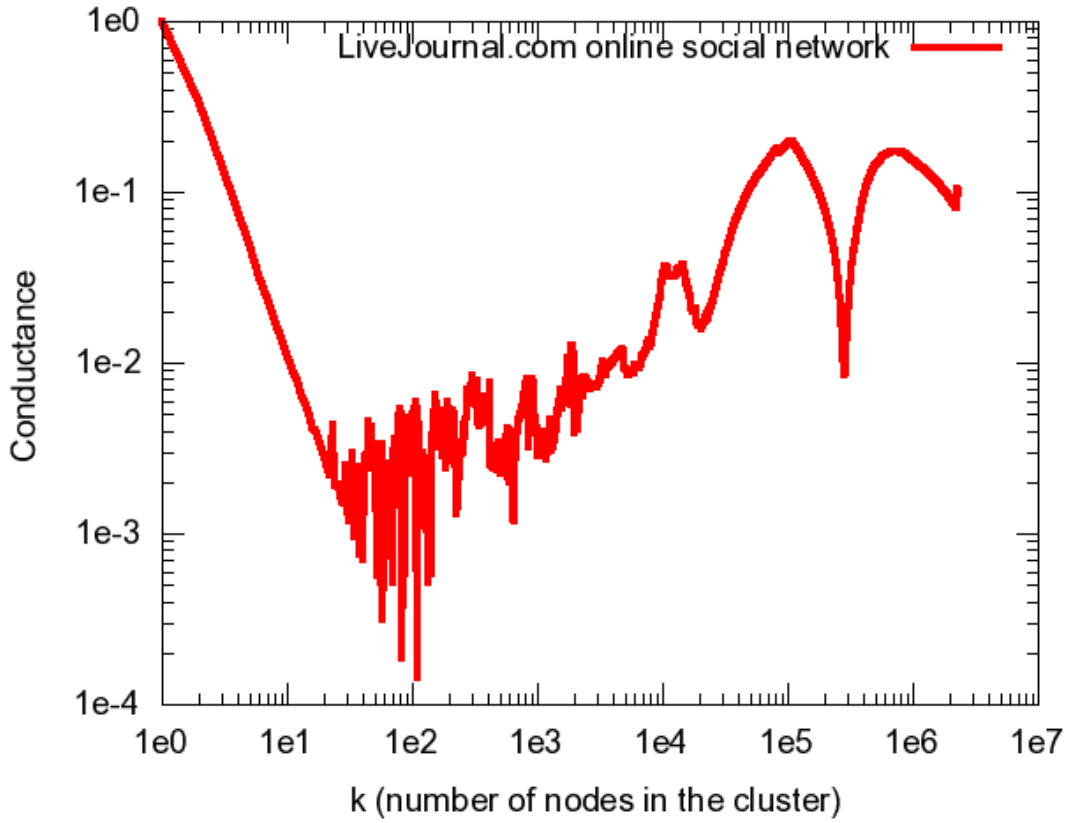
Συμπεραίνεται λοιπόν ότι το μέγεθος της καλύτερης δυνατής κοινότητας είναι στους 100 κόμβους. Επίσης στο Σχήμα 5.7 παρουσιάζεται το network profile plot του τυχαίου γράφου και στο Σχήμα 5.8 το network profile plot του LiveJournal [22]. Παρατηρείται ότι στους 30 κόμβους παρουσιάζεται η καλύτερη δυνατή ποιότητα για τον τυχαίο γράφο, αλλά η ποιότητα (conductance) δεν είναι τόσο καλή όσο της Νέας Ορλεάνης.



Σχήμα 5.7 Network Community Profile Plot τυχαίου γράφου

Στο Σχήμα 5.8 παρατηρείται ότι το δίκτυο LiveJournal παρουσιάζει αρκετές ομοιότητες με εκείνο της Νέας Ορλεάνης, σε ότι αφορά στη διαμόρφωση του network community profile plot όσο αυξάνεται ο αριθμός κόμβων στο cluster. Η καλύτερη δυνατή ποιότητα για το LiveJournal παρατηρείται στους 200 κόμβους.

Συμπεραίνεται λοιπόν ότι το κοινωνικό δίκτυο της Νέας Ορλεάνης έχει καλύτερη ποιότητα από το αντίστοιχο τυχαίο γράφο και επίσης πως το network community profile plot έχει αρκετές ομοιότητες στα μεγάλα κοινωνικά δίκτυα με τις καλύτερες κοινότητες να έχουν αρκετά μικρό μέγεθος κόμβων.

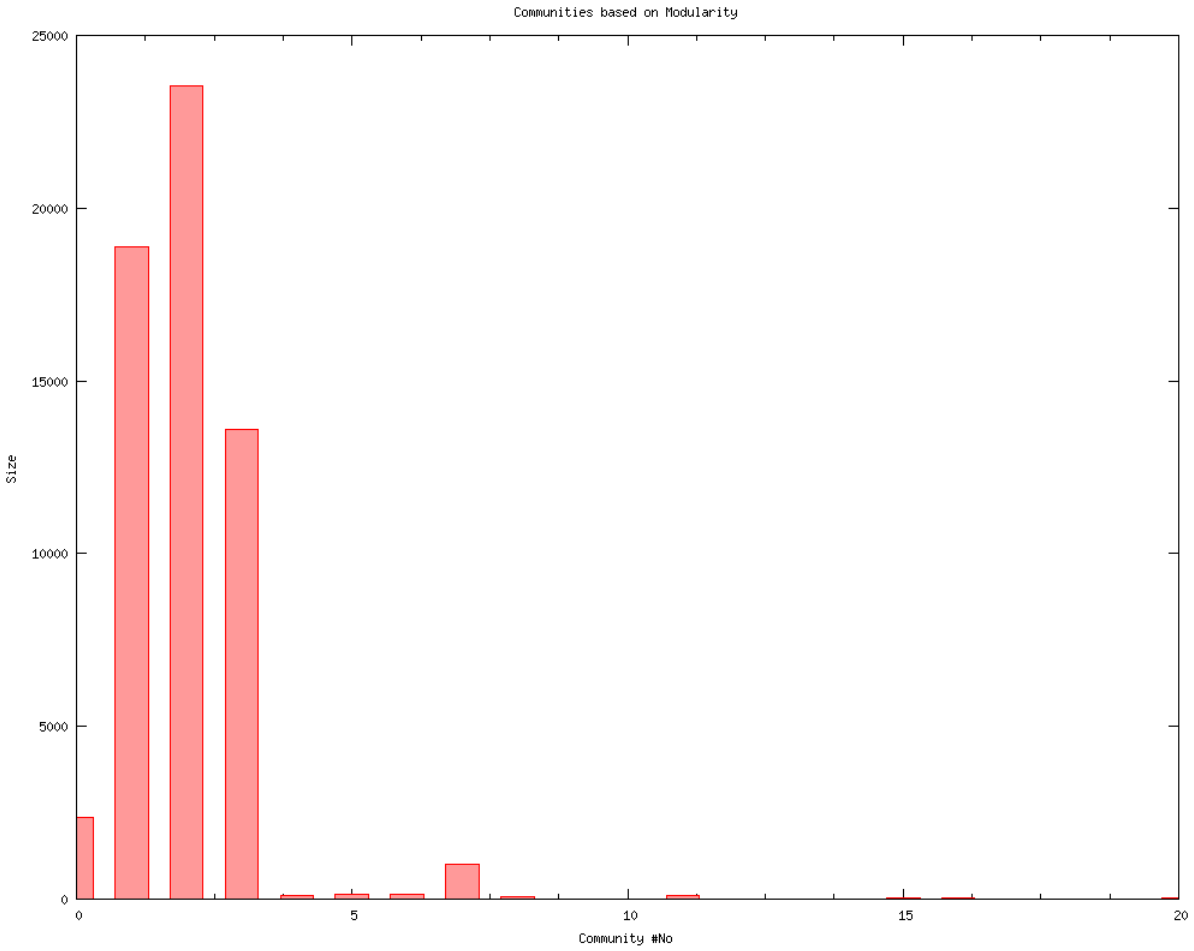


Σχήμα 5.8 Network Community Profile Plot του LifeJournal

5.2.5 Modularity

Τα αποτελέσματα της έρευνας παρουσιάζουν αρκετά καλό modularity Q για το κοινωνικό δίκτυο, με τιμή 0.52. Με βάση τον αλγόριθμο του Newman, βρέθηκαν 769 κοινότητες, όπως φαίνεται στο Σχήμα 5.9. Παρατηρούνται επίσης τρείς (3) τεράστιες κοινότητες και πολλές μικρές κάτω των 200 κόμβων. Για παράδειγμα, όπως στην περίπτωση του πληθυσμού μιας χώρας, όπου συνηθίζεται να παρουσιάζονται ελάχιστες μεγάλες πόλεις και πολλά μικρά χωριά, έτσι και εδώ εντοπίζεται μια μορφή power law αφού τα αποτελέσματα παρουσιάζουν 3-4 μεγάλες κοινότητες και πάρα πολλές μικρές κοινότητες. Το πρόβλημα που παρουσιάζεται, όμως, είναι ότι το αποτέλεσμα δεν λαμβάνει υπόψη την περίπτωση που ένας χρήστης ανήκει σε περισσότερες από μια κοινότητα. Για παράδειγμα, ένας φοιτητής που παρακολουθεί μαθήματα πληροφορικής και επίσης ανήκει σε ένα αθλητικό σύλλογο καλαθόσφαιρας, είναι εμφανές ότι ανήκει σε δυο κοινότητες, εκείνη των συμφοιτητών που παρακολουθούν το ίδιο μάθημα και

εκείνη των συμπαιχτών της καλαθόσφαιρας. Αυτό είναι το λεγόμενο overlapping κοινωτήτων και δυστυχώς δεν καθίσταται δυνατό να παρουσιαστεί με τον αλγόριθμο του Newman που χωρίζει το δίκτυο σε κομμάτια χωρίς να λαμβάνει υπόψη το overlapping.



Σχήμα 5.9 Κοινότητες βασισμένες στο modularity

5.2.6 Betweenness Centrality

Το betweenness centrality είναι βασισμένο στα σύντομα μονοπάτια. Με βάση τα αποτελέσματα από τα πειράματα, έχει βρεθεί μέσος όρος 0.0001, με την μεγαλύτερη τιμή betweenness να είναι 0.0331. Στο Σχήμα 5.10 παρουσιάζονται κάποια στατιστικά σχετικά με τα αποτελέσματα των πειραμάτων, ενώ στο Σχήμα 5.11 η γραφική παράσταση της κατανομής συχνότητας του betweenness centrality (κόκκινη γραμμή:

συχνότητα, πράσινη γραμμή: αθροιστική συχνότητα). Παρατηρείται ότι υπάρχει εξαιρετικά μικρός αριθμός κόμβων (μικρότερο από 1%) με ψηλό betweenness (>0.005), με το μεγαλύτερο ποσοστό κόμβων (80%) να είναι μεταξύ 0.0 και 002 και το 19% των κόμβων να έχει 0 betweenness. Εξάγεται λοιπόν το συμπέρασμα ότι η πλειοψηφία των κόμβων δεν ασκεί δύναμη στις αλληλεπιδράσεις μεταξύ των υπόλοιπων κόμβων και είναι αποκεντρωμένα από το δίκτυο.

Dimension: 63731
The lowest value: 0.0000
The highest value: 0.0331

Highest values:

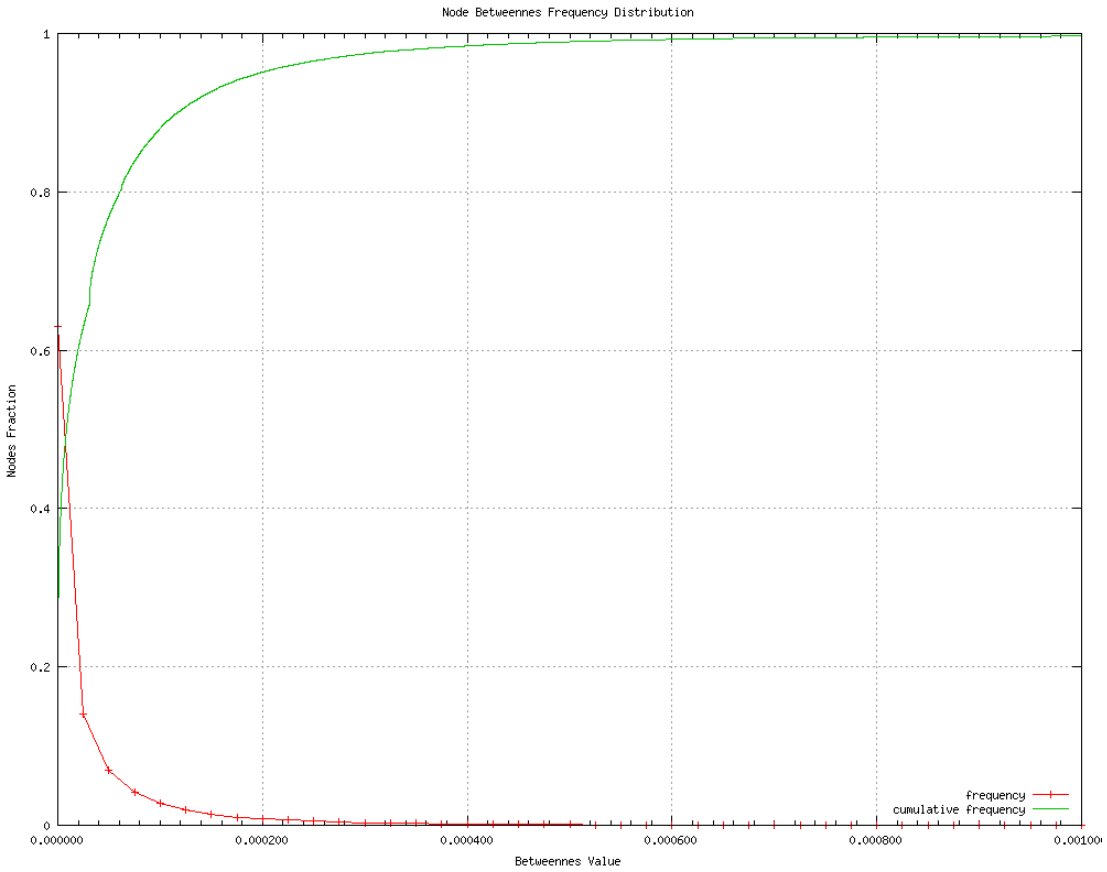
Rank	Vertex	Value	Id
1	554	0.0331	
2	471	0.0289	
3	2332	0.0289	
4	23	0.0172	
5	451	0.0158	
6	280	0.0141	
7	1463	0.0125	
8	207	0.0121	
9	84	0.0115	
10	1996	0.0097	

Sum (all values): 3.2867

Arithmetic mean: 0.0001
Median: 0.0000
Standard deviation: 0.0003
2.5% Quantile: 0.0000
5.0% Quantile: 0.0000
95.0% Quantile: 0.0002
97.5% Quantile: 0.0003

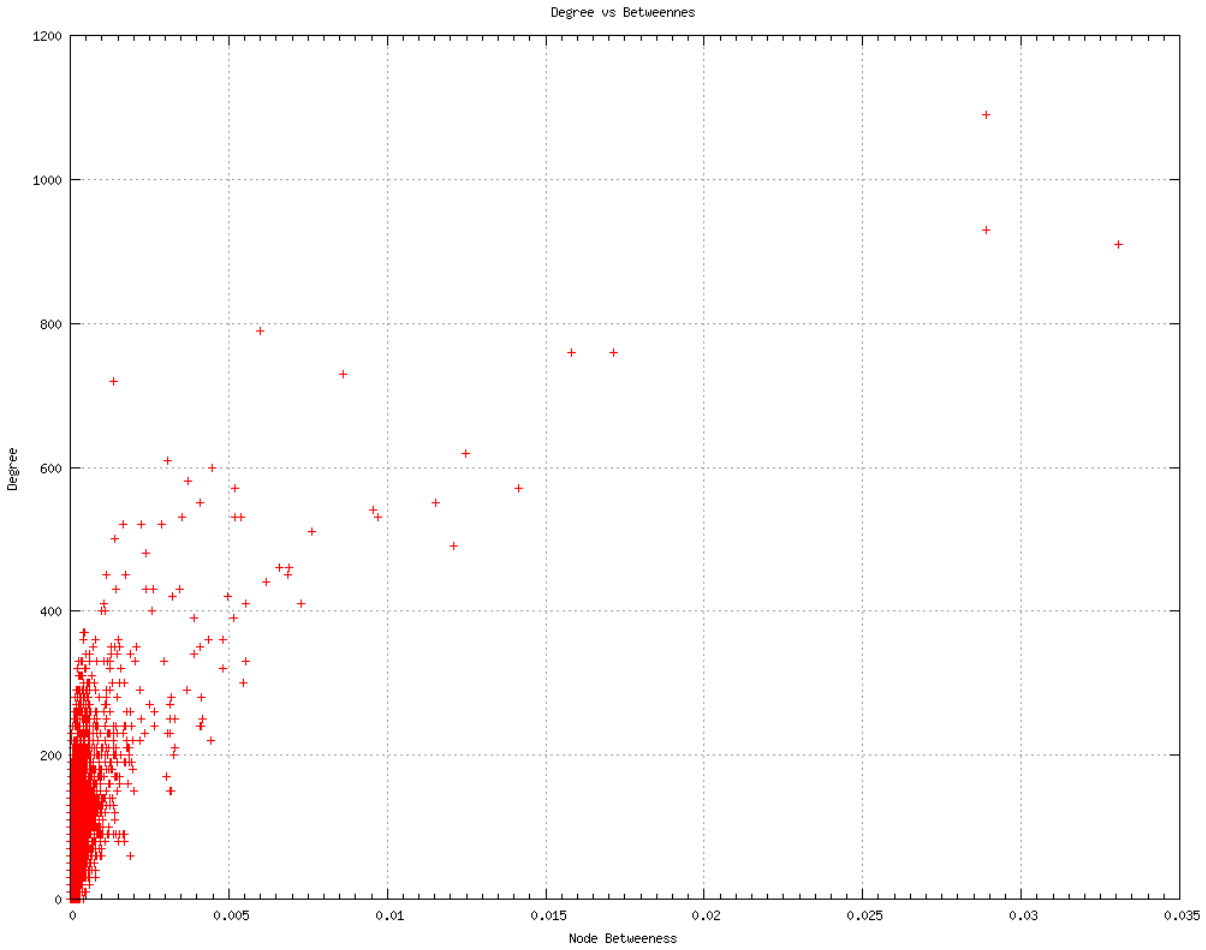
Vector Values			Frequency	Freq%	CumFreq	CumFreq%
(	...	0.000]	12355	19.3862	12355	19.3862
(0.000	...	0.002]	51284	80.4695	63639	99.8556
(0.002	...	0.003]	49	0.0769	63688	99.9325
(0.003	...	0.005]	20	0.0314	63708	99.9639
(0.005	...	0.007]	9	0.0141	63717	99.9780
(0.007	...	0.009]	3	0.0047	63720	99.9827
(0.009	...	0.010]	2	0.0031	63722	99.9859
(0.010	...	0.012]	2	0.0031	63724	99.9890
(0.012	...	0.014]	1	0.0016	63725	99.9906
(0.014	...	0.016]	1	0.0016	63726	99.9922
(0.016	...	0.017]	2	0.0031	63728	99.9953
(0.017	...	0.019]	0	0.0000	63728	99.9953
(0.019	...	0.021]	0	0.0000	63728	99.9953
(0.021	...	0.023]	0	0.0000	63728	99.9953
(0.023	...	0.024]	0	0.0000	63728	99.9953
(0.024	...	0.026]	0	0.0000	63728	99.9953
(0.026	...	0.028]	0	0.0000	63728	99.9953
(0.028	...	0.030]	2	0.0031	63730	99.9984
(0.030	...	0.031]	0	0.0000	63730	99.9984
(0.031	...	0.033]	1	0.0016	63731	100.0000
Total			63731	100.0000		

Σχήμα 5.10 Στατιστικά για betweenness centrality



Σχήμα 5.11 Betweenness Frequency Distribution κόμβων του δικτύου

Η ανάλυση εξετάζει επίσης αν το betweenness centrality συσχετίζεται με το degree. Στο Σχήμα 5.12, κάθε σημείο αντιπροσωπεύει το betweenness centrality και το degree ενός κόμβου. Είναι εμφανές ότι το betweenness centrality έχει συσχέτιση με το degree του κόμβου. Όσο πιο μεγάλο είναι το degree ενός κόμβου τόσο μεγάλο είναι και το betweenness centrality του κόμβου.



Σχήμα 5.12 Συσχέτιση Betweenness και Degree

5.2.7 Closeness centrality

Το Closeness centrality μετρά την απόσταση του κόμβου από όλους τους υπόλοιπους. Καταγράφεται ένας μέσος όρος 0.233 με την πιο μεγάλη τιμή να είναι 0.335. Στο Σχήμα 5.13 παρουσιάζονται κάποια στατιστικά σχετικά με τα αποτελέσματα των πειραμάτων ενώ στο Σχήμα 5.14 παρουσιάζεται η γραφική παράσταση (plot) της κατανομής συχνότητας. Με βάση τα αποτελέσματα, παρατηρείται ότι το 96% των κόμβων έχουν closeness μεταξύ 0.15 και 0.3 με την πλειοψηφία (~75%) να είναι μεταξύ 0.2 και 0.25, που είναι πολύ κοντά στον μέσο όρο 0.233. Συμπεραίνεται λοιπόν ότι αρκετά μεγάλο ποσοστό κόμβων έχουν την ίδια απόσταση από άλλους κόμβους.

Dimension: 63731
The lowest value: 0.0000
The highest value: 0.3550

Highest values:

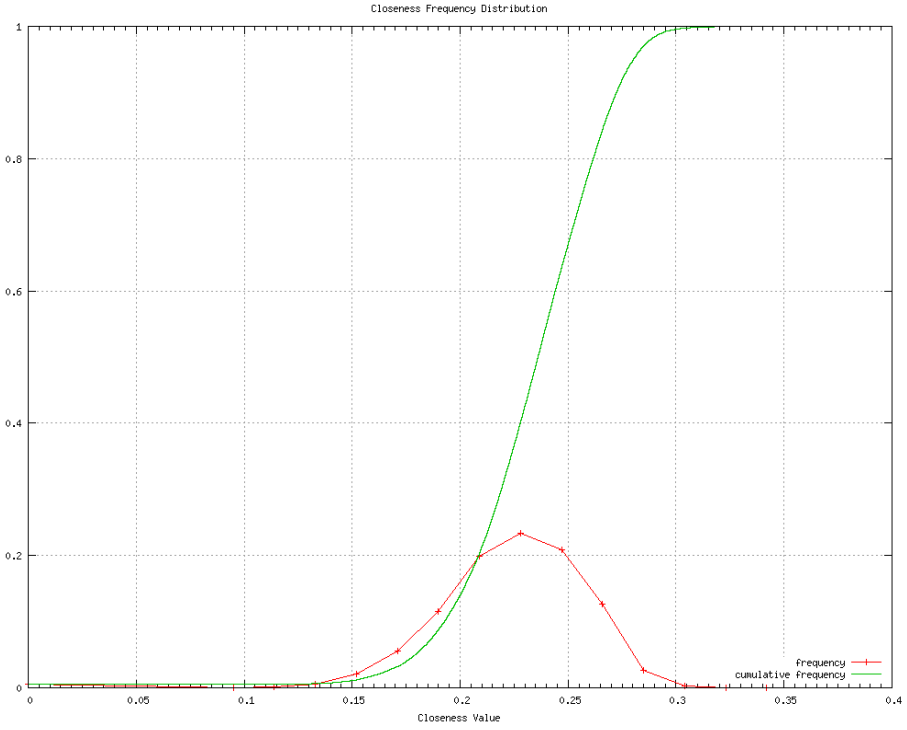
Rank	Vertex	Value	Id
1	2332	0.3550	
2	471	0.3533	
3	23	0.3442	
4	554	0.3433	
5	1463	0.3411	
6	207	0.3409	
7	280	0.3408	
8	451	0.3406	
9	1996	0.3374	
10	2805	0.3357	

Sum (all values): 14863.9847

Arithmetic mean: 0.2332
Median: 0.2360
Standard deviation: 0.0349
2.5% Quantile: 0.1660
5.0% Quantile: 0.1792
95.0% Quantile: 0.2803
97.5% Quantile: 0.2864

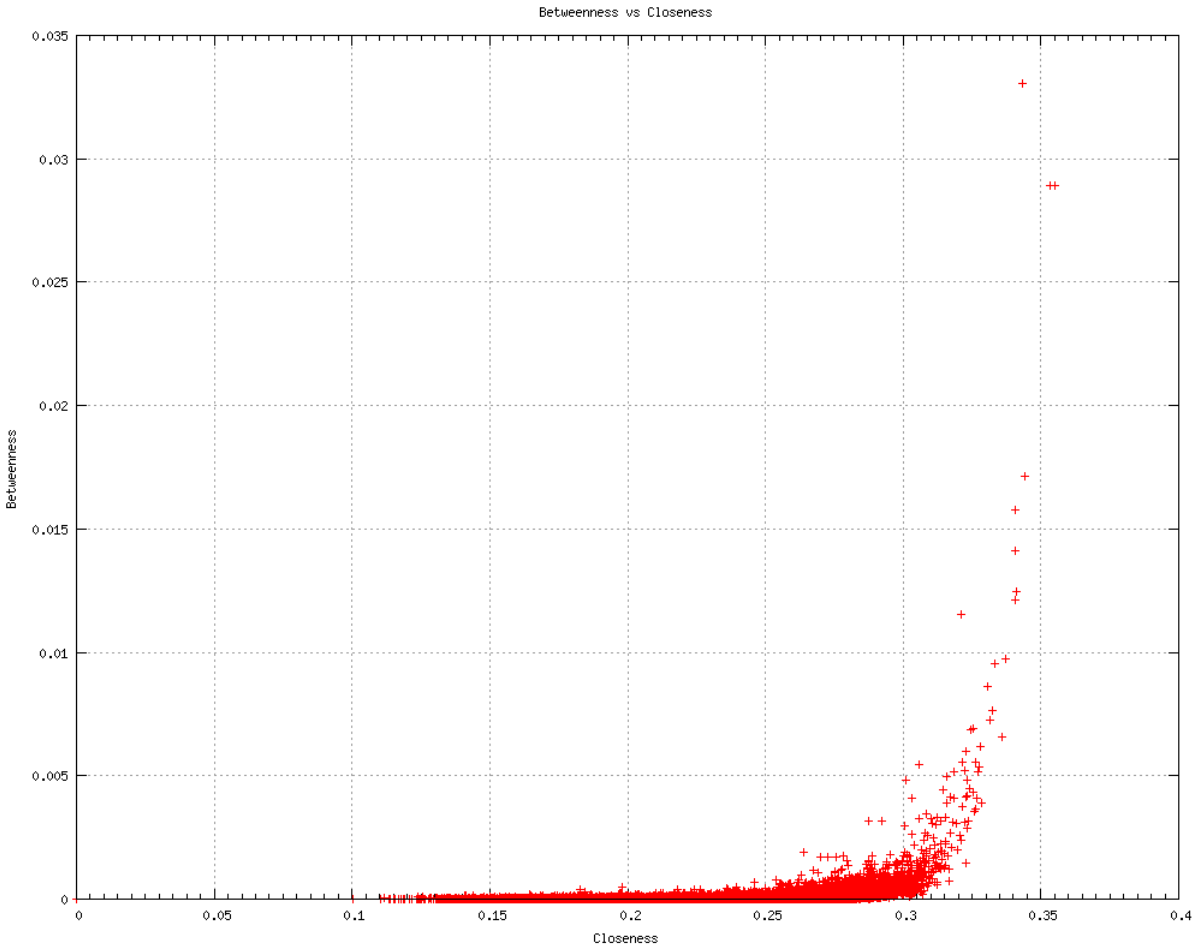
Vector Values		Frequency	Freq%	CumFreq	CumFreq%
(...	0.000]	0	0.0000	0	0.0000
(0.000 ...	0.019]	339	0.5319	339	0.5319
(0.019 ...	0.037]	0	0.0000	339	0.5319
(0.037 ...	0.056]	0	0.0000	339	0.5319
(0.056 ...	0.075]	0	0.0000	339	0.5319
(0.075 ...	0.093]	0	0.0000	339	0.5319
(0.093 ...	0.112]	6	0.0094	345	0.5413
(0.112 ...	0.131]	49	0.0769	394	0.6182
(0.131 ...	0.149]	282	0.4425	676	1.0607
(0.149 ...	0.168]	1124	1.7637	1800	2.8244
(0.168 ...	0.187]	2973	4.6649	4773	7.4893
(0.187 ...	0.206]	6496	10.1928	11269	17.6821
(0.206 ...	0.224]	11479	18.0116	22748	35.6938
(0.224 ...	0.243]	14563	22.8507	37311	58.5445
(0.243 ...	0.262]	13574	21.2989	50885	79.8434
(0.262 ...	0.280]	9615	15.0868	60500	94.9303
(0.280 ...	0.299]	2897	4.5457	63397	99.4759
(0.299 ...	0.318]	289	0.4535	63686	99.9294
(0.318 ...	0.336]	36	0.0565	63722	99.9859
(0.336 ...	0.355]	9	0.0141	63731	100.0000
Total		63731	100.0000		

Σχήμα 5.13 Στατιστικά για Closeness Centrality



Σχήμα 5.14 Closeness Frequency distribution κόμβων του δικτύου

Στο Σχήμα 5.15 εξετάζεται η συσχέτιση μεταξύ closeness και betweenness. Κάθε σημείο αντιπροσωπεύει το betweenness και το closeness ενός κόμβου. Παρατηρείται λοιπόν ότι από μια τιμή του closeness περίπου 0.25 όσο μεγαλώνει το closeness ενός κόμβου, μεγαλώνει σταδιακά και το betweenness του όπου από το 0.3 μεγαλώνει πολύ απότομα. Η συσχέτιση δηλαδή μεταξύ closeness και betweenness είναι θετική.



Σχήμα 5.15 Συσχέτιση Closeness με Betweenness

Κεφάλαιο 6

Συμπεράσματα

6.1 Γενικά Συμπεράσματα

Στη εργασία έχει γίνει μελέτη και ανάλυση του online κοινωνικού δικτύου της Νέας Ορλεάνης. Πρώτα έχουν υπολογιστεί τα χαρακτηριστικά του δικτύου και έχει παρατηρηθεί ότι το degree distribution ακολουθεί power law μορφή και επίσης ότι οι κόμβοι με χαμηλό degree τείνουν να είναι στενά συνδεδεμένοι μεταξύ τους. Διαπιστώθηκε επίσης υψηλό clustering coefficient και χαμηλή διάμετρος τα οποία είναι χαρακτηριστικά ενός small-world δικτύου. Παρατηρήθηκε επίσης ότι με την πάροδο του χρόνου η διάμετρος μικραίνει.

Για εύρεση κοινοτήτων και αξιολόγηση της ποιότητα τους, τρέξαμε τους αλγόριθμους modularity και network community profile plot. Διαπιστώθηκε υψηλό modularity και ποιότητα καλύτερης δυνατής κοινότητας στους 100 κόμβους. Επίσης παρατηρήθηκε μεγάλος αριθμός κοινοτήτων με μικρό μέγεθος, εκτός από 4 κοινότητες που παρουσίασαν τεράστιο αριθμό κόμβων.

Υπολογίστηκε το betweenness και closeness centrality των κόμβων. Παρατηρήθηκε ότι η πλειοψηφία των κόμβων έχουν χαμηλό betweenness και closeness πολύ κοντά στο μέσο όρο καθώς και θετική συσχέτιση μεταξύ betweenness και closeness. Συμπεραίνεται ότι οι κόμβοι δεν είναι κεντρικοί στο δίκτυο, δεν επηρεάζουν στις αλληλεπιδράσεις μεταξύ άλλων κόμβων και έχουν σχεδόν την ίδια απόσταση μεταξύ τους

6.2 Περιοχές για έρευνα

Σαν μελλοντικού στόχους για αυτή την εργασία, γίνεται εισήγηση για μελέτη και σύγκριση διαφόρων ειδών δικτύων για να εντοπιστούν και καταγραφούν ομοιότητες και διαφορές. Επίσης γίνεται εισήγηση για μελέτη ανίχνευσης overlapping κοινοτήτων με βάση τις μεθόδους Clique Percolation Method και k-cores. Ο αλγόριθμος modularity υποθέτει ότι ο κάθε κόμβος ανήκει σε μια μόνο κοινότητα, αυτό όμως δεν είναι πάντα σωστό. Είναι δυνατόν ένας κόμβος να ανήκει σε περισσότερες από μια κοινότητα.

Επίσης γίνεται εισήγηση για μελέτη αλληλεπίδρασης κόμβων βάση μηνυμάτων και δημοσιεύσεων και να δημιουργηθεί ο αντίστοιχος γράφος αλληλεπίδρασης για να γίνουν συγκρίσεις με τον γράφο της δομής του δικτύου. Θεωρείται σημαντικό να μελετηθεί πόσο συχνά αλληλεπιδρούν ή όχι δύο κόμβοι μεταξύ τους και κατα πόσο αξιοποιούνται οι μεταξύ τους σχέσεις. Σημαντική μελέτη θεωρείται και η σημαντικότητα των κόμβων στο γράφο αλληλεπίδρασης (centrality, hubs & authorities κτλ).

6.3 Εφαρμογές

Εφαρμογές της ανάλυσης του κοινωνικού δικτύου μπορούν να υπάρχουν σε πληροφοριακά συστήματα ή και ακόμη στα ίδια τα κοινωνικά δίκτυα. Επίσης εφαρμογές υπάρχουν όχι μόνο στον κλάδο της πληροφορικής αλλά και πολλούς άλλους όπως στο μάρκετινγκ και στην πολιτική.

Στις πλατφόρμες των κοινωνικών δικτύων, μια τέτοια εφαρμογή είναι να δημιουργηθεί μια λειτουργία η οποία να προτείνει στους χρήστες καινούργιους φίλους βάση με τις κοινωνικές σχέσεις του χρήστη. Για παράδειγμα εάν δυο χρήστες δεν είναι φίλοι μεταξύ τους αλλά έχουν αρκετούς κοινούς φίλους (οι οποίοι είναι και μεταξύ τους φίλοι) τότε υπάρχει μεγάλη πιθανότητα να γνωρίζονται οι δυο χρήστες και έτσι η πλατφόρμα του κοινωνικού δικτύου θα μπορούσε να προτείνει την φιλία μεταξύ τους. Με αυτό τον τρόπο, παλιοί συμμαθητές, συμφοιτητές ή παιδικοί φίλοι που έχουν χάσει επαφή στον πραγματικό κόσμο, μπορούν να την αποκτήσουν μέσω του κοινωνικού δικτύου.

Επίσης για κάποιες πλατφόρμες κοινωνικών δικτύων θα μπορούσαν να προτείνουν συστάσεις για νέες φιλίες όχι μόνο βάση με κοινούς φίλους αλλά και βάση κοινών χαρακτηριστικών, και ενδιαφερόντων. Χρήστες που ανήκουν στην ίδια κοινότητα τείνουν να έχουν κοινά ενδιαφέροντα.

Στα πληροφοριακά συστήματα, μια σχέση μεταξύ δυο χρηστών υποδηλώνει μια μορφή εμπιστοσύνης, οπότε θα μπορούσε να υλοποιηθεί ένα σύστημα επικοινωνίας (όπως email) μεταξύ ατόμων με βάση το κοινωνικό δίκτυο. Μηνύματα που εισέρχονται από φίλους, μπορούν να θεωρηθούν αξιόπιστα καθώς και ακίνδυνα για τον ίδιο τον χρήστη (spam, links that forward to harmful websites κτλ). Μηνύματα που προέρχονται από ξένους δεν θα θεωρούνται τόσο αξιόπιστα όσο θεωρούνται τα μηνύματα από φίλους, οπότε το σύστημα θα μπορεί να επεξεργαστεί με ιδιαίτερη προσοχή τα μηνύματα αυτά για τυχόν spam ή βλαβερό περιεχόμενο.

Μια εξίσου καλή εφαρμογή είναι στις μηχανές αναζήτησης για καλύτερο φιλτράρισμα αποτελεσμάτων. Όπως είχαμε αναφέρει, οι χρήστες που ανήκουν σε κοινότητες τείνουν να έχουν κοινά ενδιαφέροντα, ενασχολήσεις κτλ. Αναγνωρίζοντας λοιπόν, τα κοινά χαρακτηριστικά που συνδέουν τα άτομα μιας κοινότητας, εύκολα θα μπορούσε να εφαρμοστεί στις μηχανές αναζήτησης έτσι ώστε να γίνει αναζήτηση πρώτα στις κοινότητες (εφόσον αυτό για το οποίο γίνεται αναζήτηση έχει σχέση με την κοινότητα).

Εκτός από τις πλατφόρμες των κοινωνικών δικτύων και τα πληροφοριακά συστήματα, υπάρχουν εφαρμογές στο μάρκετινγκ και στην πολιτική. Πιο συγκεκριμένα, αναγνωρίζοντας ποιός χρήστης είναι πιο κεντρικός μέσα στο κοινωνικό δίκτυο, η διάδοση πληροφορίας (όπως μια διαφήμιση) θα είναι ταχύτερη και θα έχει ως γενικό αποδέκτη περισσότερα άτομα από το να διαδοθεί η πληροφορία από έναν αποκεντρωμένο χρήστη.

Στην πολιτική, θα μπορούσε μέσω κοινοτήτων να αναγνωριστούν πιθανοί ψηφοφόροι έτσι ώστε να προσεγγιστούν τα συγκεκριμένα άτομα για την απόκτηση της ψήφου τους. Η αναγνώριση των κοινών χαρακτηριστικών της ομάδας ατόμων μπορεί να επηρεάσει σε μεγάλο βαθμό τον τρόπο που θα προσεγγιστούν αυτά τα άτομα.

Βιβλιογραφία

[1] ACM SITCOMM website [Available at: http://www.sigcomm.org/]

[2] Adamic, L.A., Buyukkokten, O., and E. Adar(2003) 'A social network caught in the Web' *First Monday*, 8(6)

[3] Adamic, L.A., Lukose. R M, Puniyani, A.R. and B.A. Huberman (2001) 'Search in power-law networks' *Physical Review E*, 64, 046135

[4] Albert, R., Jeong, H. and A.L. Barabas (1999) 'Diameter of the World-Wide Web', *Nature*, 401 (Sept.), 130-131

[5] Baumes, J., Goldberg, M., Krishnamoorhy, M., Magdon-Ismail, M and N. Preston (2005) 'Finding Communities by Dustering a Graph into Overlapping Subgraphs, *Proceedings of International Conference on Applied Computing (IADIS 05)* Algrave, Portugal, 27-36

[6] Benevenuto F, Rodrigues, T, Cha M. and V. Almeida (2009) 'Characterizing User Behavior in Online Social Networks' *Information Management Conference IMC'09*, 4(6)

[7] Clauset, A., Newman, M.E.J and C. Moore (2004) 'Finding Community Structure in Very Large Networks, *Physical Review, E.* 70 (6)

[8] Cooper, C. and A. Freize (2003) 'A General Model of Web Graphs' *Random Structures and Algorithms*, 22, 311-335

[9] Doubleclick ad banner by Google (2011) 'Site profile for Facebook.com' April, 2011 [Available at https://www.google.com/adplanner/planning/site_profile#siteDetails?identifier=facebook.com&geo=001&trait_type=1&lp=true]

[10] Easley, D. and J. Kleinberg (2010) *Networks, Crowds and Markets: Reasoning About a Highly Connected World*, Cambridge University Press

[11] Erdos, P. and A. Renyi. (1959) 'On Random Graphs I', *Publicationes Mathematicae*, 6, 290-297

[12] Freeman, L. C. (1979) 'Centrality in Social Networks: I. *Conceptual Clarification. Social Networks*, 1, 215–239

[13] Girvan, M., and M.E.J. Newman (2002) 'Community structure in social and biological network' *Physical Science, Applied Mathematics*, 99 (12), 7821-7826

[14] Jiang J, Wilson C, Wang X. Huang, P. Sha, W., Dai Y., and B.Y. Zhao (2010) 'Understanding Latent Interactions in Online Social Networks' *Information Management Conference IMC'10*. Nov 1-3

[15] Kernighan B.W. and S. Li (1970) 'An Efficient Hueristic procedure for Partitioning Graphs' *Bell System Technical Journal*, 49, 291-307

[16] Kleinberg, J. M. (2001) 'Small-world Phenomena and the Dynamics of Information' *Advances in Neural Information Processing Systems* (NIPS), 14

[17] Kossinets G. and D. J. Watts (2006) 'Empirical Analysis of Evolving Social Network' *Science 6*, 311(5757), 88-90

[18] Krapivsky P. L. and S. Redner (2001) 'Organization of Growing Random Networks' *Physical Review E*, 63, 066123

[19] Kumar R., Raghavan, P., Rajagopalan, S. and A. Tomkins (1999) 'Trawling the Web for Emerging Cyber-Communities', *Computer Networks*, 1481-1293

[20] Leskovec J, Kleinberg J. and C. Faloutsos (2007) 'Graph Evolution: Densification and Shrinking Diameters' *ACM Transactions on Knowledge Discovery from Data*, 1(1), Art. 2

[21] Leskovec, J., Chakrabarti, D., Kleinberg, J., Faloutsos, C., and Z. Ghahramani (2010) 'Knonecker Graphs: An Approach to Modelling Networks' *Journal of Machine Learning Research,* 11, 985-1042

[22] Leskovec, J., Lang, K.J., Dasgupta, A., and M.W. Mahoney (2008) 'Community Structure in Large Networks: Natural Cluster Sizes and the Absense of Large Well-Defined Clusters' *Computing Research Repository,* CORP

[23] Leskovec, J., Lay, K.J. Dasgupta, A. and M.W. Mahoney (2008) 'Statistical Properties of Community Structure in Large Social and Information Networks' *Proceedings of the 17th International WWW,* 695-704

[24] Milgram, S. (1967) 'The small world problem', *Psychology Today*, 1(1), 61-67

[25] Newman M. E. J. and M. Girvan (2004) 'Trading and Evaluating Community Structure in Networks', *Physical Review E*, 69, 026113

[26] Newman, M. E. J. (2003) 'The structure and function of complex networks' *SIAM Review,* 45, 167-256

[27] Newman, M. E. J. (2004) 'Fast Algorithm for Detecting Community Structure in Networks' *Physical Review, E*, 69 (6)

[28] Newman, M.E.J. (2006) 'Modularity and Community Structure in Networks' *Physical Science - Applied Mathematics,* 103 (23), 8577-8582

[29] Palla, G., Imre, D., Farkas, I and T. Vicsek (2005) 'Uncovering the Overlapping Community Structure of Complex Networks in Nature and Society' *Nature*, 435 (7043), 814-818

[30] Pothen, A. Horst, D.S. and K. P. Liou (1990) 'Partitioning Sparse Matrices with Eigenvectors of Graphs' Society for Industrial and Applied Mathematics (SIAM) *Journal of Matrix Analysis and Application*, 11(3), 430-453

[31] Rapoport. A. (1957) 'Contribution to the Theory of Random and Biased Net' *Bulletin of Mathematical Biology*, 19, 257-277

[32] Tancer, B. (2006) Analyst Weblog: 'MySpace moves to no.1 position for all internet sites' *Experian Hitwise Intelligenc*e, July, 11, 2006 [Available at http://weblogs.hitwise.com/bill-tancer/2006/07/myspace_moves_into_1_position.html]

[33] Viswanath B, Mislove A, Cha M, and K.P. Gummadi (2009) 'On the Evolution of User Interaction in Facebook' *Workshop on Online Social Network*s, WOSN '09.

[34] Wasserman, S., and K. Faust (1994). *Social Network Analysis: Methods and Applications* Cambridge University Press

[35] Watts, D.J., and S.H. Strogatz (1998) 'Collective dynamics of 'small-world' networks' *Nature*, 393, 440-442.

[36] WOSN (2009) An ACM SIGCOMM 2009 *Workshop*, August, 17th, Barcelona, Spain, [Available at: http://conferences.sigcomm.org/sigcomm/2009/workshops/wosn/]

[37] Zuckerberg M. (2010) '500 Millio Stories' *The Facebook Blog*, July 21, 2010, [Available at https://blog.facebook.com/blog.php?post=409753352130]

[38] Mislove, A., Marcon, M., Gummadi, K.P., Druschel, P. and B. Bhattacharjee (2007) Measurement and Analysis of Online Social Networks, in Proc. Of IMC 2007

[39] Καχριμάνης Γ., Κόμης, Β., και Αβούρης, Ν. (2008) 'Μεθοδολογίες ανάλυσης της συνεργασίας. Στο Ν. Αβούρης, Χ. Καραγιαννίδης, Β. Κόμης' (Επιμέλεια Έκδοσης) Συνεργατική τεχνολογία, συστήματα, και μοντέλα συνεργασίας για εργασία, μάθηση, κοινότητες πρακτικής και δημιουργία γνώσης (σελ. 179-212). Εκδόσεις Κλειδάριθμος, Αθήνα, 2008

Παράρτημα Α

Εγχειρίδιο εργαλείου SNAP

SNAP MANUAL CONTENTS

SNAP MANUAL

1. About Snap

Stanford Network Analysis Platform (SNAP) is a general purpose network analysis and graph mining library. It is written in C++ and easily scales to massive networks with hundreds of millions of nodes, and billions of edges. It efficiently manipulates large graphs, calculates structural properties, generates regular and random graphs, and supports attributes on nodes and edges.

SNAP is based on node and edge iterators which allows for efficient implementation of algorithms that work on networks regardless of their type (directed, undirected, graphs, networks) and specific implementation. SNAP was developed by Jure Leskovec and was built on top of a general purpose STL (Standard Template Library)-like library **GLib** that was developed at Jozef Stefan Institute.

2. Features offered by this release

As of release **version 2011-04-17**, SNAP offers the following features:

- Support of undirected, directed and directed multi-graph
- Iterators for fast node traversal
- Graph input and output in various text file formats
- Graph manipulation
- Generate random graphs
- Structural Properties measures
- Evolution of structural properties measures
- Community detection algorithms
- Node Centrality measures

3. Installation and Requirements

You can get the latest release at following website http://snap.stanford.edu/snap/download.html.

The whole installation consists of unpacking the file in a directory which will produce 3 folders: snap, glib and examples folder. The files should be unpacked in the same directory as your source code. In examples folder, a sample dataset *AS20GRAPH.txt* is provided.

For plotting structural properties of networks (e.g., degree distribution) the SNAP expects to find **GnuPlot** in the system path. Similarly for drawing and visualizing small graphs SNAP utilizes **GraphViz** which has to be installed on the system. Set system PATH variable appropriately or put the executables in the working directory.

GnuPlot is a free, command-driven, interactive, function and data plotting program that is used in conjunction with SNAP to plot the structural properties. The latest version of GnuPlot program can be obtained at page http://www.gnuplot.info/download.html

Graphviz is open source graph visualization software. Graph visualization is a way of representing structural information as diagrams of abstract graphs and networks. The latest version of Graphviz program can be obtained at http://www.graphviz.org/Download.php

4. How to Compile Code

To compile SNAP under Windows, you can use Visual Studio (preferably the latest since old versions of Visual studio might cause problems with missing symbols) or Cygwin with GCC.
Under Linux and other UNIX clones, you can make use of GCC.
Makefiles are provided but depending on your platform you may need to slightly modify the Makefile.

A Makefile example is the following:

```
#
# Makefile for non-Microsoft compilers
#

## Linux  (uncomment the 2 lines below for compilation on Linux)
#CXXFLAGS += -std=c++98 -Wall
#LDFLAGS += -lrt

## CygWin (uncomment the 2 lines below for compilation on CygWin)
#CXXFLAGS += -Wall
#LDFLAGS +=

## Main application file
MAIN = ncpplot

all: $(MAIN)

opt: CXXFLAGS += -O4
opt: LDFLAGS += -O4
opt: $(MAIN)

# COMPILE
$(MAIN): $(MAIN).cpp Snap.o
      g++ $(LDFLAGS) -o $(MAIN) $(MAIN).cpp ../../snap/ncp.cpp Snap.o
-I../../glib -I../../snap

Snap.o:
      g++ -c $(CXXFLAGS) ../../snap/Snap.cpp -I../../glib -
I../../snap

clean:
      rm -f *.o  $(MAIN)  $(MAIN).exe
      rm -rf Debug Release
```

5. Directories included in release

snap

the SNAP graph library

glib

contains the library that provides basic data structures (vectors, strings, hash tables), infrastructure (serialization, xml and html parsing), interface to GnuPlot and linear algebra (SVD).

examples

sample applications and examples of use. a sample dataset *AS20GRAPH.txt* is provided

cascades	Simulate a SI (susceptible-infected) model on a network and compute structural properties of cascades
centrality	Node centrality measures (closeness, eigen, degree, betweenness, page rank, hubs and authorities)
cliques	Clique Percolation Method for detecting overlapping communities
community	Network Community detection (Girvan-Newman and Clauset-Newman-Moore)
concomp	Manipulateds weakly/strongly/bi-connected components of a graph
forestfire	Forest Fire graph generator
kcores	Computes the k-core decomposition of the network
krongen	Kronecker graph generator
kronfit	Estimates Kronecker graph parameter matrix
motifs	Counts the number of occurence of every possible subgraph on K nodes in the network
ncpplot	Computers Network community profile plot of a network
netevol	computes properties of an evolving network, like evolution of diameter, degree distribution etc
netstat	computes statistical properties of a static network, like degree distribution, hop plot, clustering coefficient etc

6. Example Applications

Application: Cascades

Description

The application implements a simple SI (susceptible--infected) model and measures structural properties of cascades (propagation trees). The program measure how the cascade properties (like, number of nodes, edge and depth) change as a function of amount of missing data (number of random nodes removed from the cascade).

For more details and motivation what this code is trying to achieve see *"Correcting for Missing Data in Information Cascades" by E. Sadikov, M. Medina, J. Leskovec, H. Garcia-Molina. WSDM, 2011.*

Depending on the platform (Windows or Linux) you need to edit the Makefile.
Use 'make opt' to compile the optimized (fast) version of the code.

Parameters

```
-i:Input graph (tab separated list of edges) (default:'demo')
-o:Output file name (default:'demo')
-b:Infection (i.e., cascade propagation) probability
```

Usage

```
./cascades -i:demo -o:demo -b:0.1
```

Application: Centrality

Description

Loads a directed graph and computes the following node centrality measures. Measures defined on an undirected graph (we drop edge directions):

a. degree centrality
b. closeness centrality
c. betweenness centrality
d. eigenvector centrality

Measures defined on (the original) directed graph

a. page rank
b. hubs and authorities

For more details on definitions of these measures see http://en.wikipedia.org/wiki/Centrality

Depending on the platform (Windows or Linux) you need to edit the Makefile. Use 'make opt' to compile the optimized (fast) version of the code.

Parameters

```
-i:Input graph (tab separated list of edges)
(default:'graph.txt')
```

Usage

```
./centrality -i:as20graph.txt -
```

Application: Cliques

Description

The example find overlapping dense groups of nodes in networks, based on the Clique Percolation Method (CPM). For example, see http://cfinder.org/wiki/?n=Main.ImageWords

The clique percolation method builds up the communities from *k*-cliques, which correspond to complete (fully connected) sub-graphs of *k* nodes. Two *k*-cliques are considered adjacent if they share $k-1$ nodes. A community is defined as the maximal union of *k*-cliques that can be reached from each other through a series of adjacent *k*-cliques. Such communities can be best interpreted with the help of a *k*-clique template (an object isomorphic to a complete graph of *k* nodes). Such a template can be placed onto any *k*-clique in the graph, and rolled to an adjacent *k*-clique by relocating one of its nodes and keeping its other $k-1$ nodes fixed. Thus, the *k*-clique communities of a network are all those sub-graphs that can be fully explored by rolling a *k*-clique template in them, but cannot be left by this template. Since even small networks can contain a vast number of *k*-cliques, the implementation of this approach is based on locating the maximal cliques rather than the individual *k*-cliques. Thus, the complexity of this approach in practice is equivalent to that of the NP-complete maximal clique finding, (in spite that finding *k*-cliques is polynomial). This means that although networks with few million nodes have already been analyzed successfully with this approach, no prior estimate can be given for the runtime of the algorithm based simply on the system size.

The Clique Percolation Method is described in details in *G. Palla, I. Derenyi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435, 814-818 (2005).*

The maximal clique enumeration procedure implements the method by *E. Tomita, A. Tanaka, H. Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. Theoretical Computer Science, Volume 363, Issue 1, 2006.*

Depending on the platform (Windows or Linux) you need to edit the Makefile. Use 'make opt' to compile the optimized (fast) version of the code.

Parameters

```
-i:Input undirected graph file (single directed edge per
line) (default:'../as20graph.txt')
-k:Minimal clique overlap size (default:3)
-o:Output file prefix (default:'')
```

```
./cliques -i:../as20graph.txt -k:2 -
```

Application: community

Description

Implements two network community detection algorithms:

a. **Girvan-Newman algorithm** (*Girvan M. and Newman M. E. J., Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99, 7821–7826 (2002)*)

b. **Fast modularity maximization algorithm** *by 'Finding Large community in networks', A. Clauset, M.E.J. Newman, C. Moore, 2004*

Depending on the platform (Windows or Linux) you need to edit the Makefile. Use 'make opt' to compile the optimized (fast) version of the code.

Parameters

```
-i:Input graph (tab separated list of edges)
(default:'graph.txt')
-o:Output file name (default:'communities.txt')
-a:Algorithm: 1:Girvan-Newman, 2:Clauset-Newman-Moore
```

Usage

Compute communities in the AS graph

```
./community -i:../as20graph.txt -a:2
```

Application: concomp

Description

The sample finds the connected components of a given network. A connected component of an undirected graph is a subgraph in which any two vertices are connected to each other by paths, and which is connected to no additional vertices

The example loads a directed (or undirected) graph and computes:

a. **weakly connected components:** for any pair of nodes there is an undirected path between nodes
b. **strongly connected components:** *(directed graph)* for any pair of nodes in the component there is a directed path between them
c. **biconnected components:** *(undirected graph)* any pair of nodes is connected by 2 disjoint paths (removal of any single edge does not disconnect the component
d. **articulation points:** *(undirected graph)* vertices that if removed disconnect the graph
e. **bridge edges:** *(undirected graph)* edges that if removed disconnect the graph

Depending on the platform (Windows or Linux) you need to edit the Makefile. Use 'make opt' to compile the optimized (fast) version of the code.

Parameters

```
-i:Input graphs (each file is a graph snapshot, or use "DEMO")
(default:'graph*.txt')
-o:Output file name prefix (default:'over-time')
```

Usage

```
./concomp -i:../as20graph.txt
```

Application: forestfire

Description

Forest Fire graph generation model, is based on having new nodes attach to the network by ``burning'' through existing edges in epidemic fashion. For a range of parameter values the model exhibits realistic behavior in densification, shrinking diameter, and degree distributions.

For more information about the model see: *Graph Evolution: Densification and Shrinking Diameters Jure Leskovec, Jon Kleinberg, Christos Faloutsos. ACM Transactions on Knowledge Discovery from Data (ACM TKDD), 1(1), 2007.* http://arxiv.org/abs/physics/0603229

Depending on the platform (Windows or Linux) you need to edit the Makefile. Use 'make opt' to compile the optimized (fast) version of the code.

Parameters

```
-o:Output graph file name (default:'graph.txt')
-n:Number of nodes (size of the generated graph) (default:10000)
-f:Forward burning probability (default:0.35)
-b:Backward burning probability (default:0.32)
-s:Start graph with S isolated nodes (default:1)
-a:Probability of a new node choosing 2 ambassadors (default:0)
-op:Probability of a new node being an orphan (node with zero out-
degree) (default:0)
```

Usage

Generate a Forest Fire graph on 1000 nodes with forward burning probability f=0.3 and backward burning probability b=0.25

```
./forestfire -o:graph.txt -n:1000 -f:0.3 -
```

Application: kcores

Description

This is a K-core network decomposition example. It Plot the number of nodes in a k-core of a graph as a function of k.

A subgraph H = (C,E|C) induced by the set C subset of V is a k-core or a core of order k if and only if the degree of every node v in C induced in H is greater or equal than k, and H is the maximum subgraph with this property.

A k-core of G can be obtained by recursively removing all the vertices of degree less than k, until all vertices in the remaining graph have degree at least k.

Depending on the platform (Windows or Linux) you need to edit the Makefile. Use 'make opt' to compile the optimized (fast) version of the code.

Parameters

```
-i:Input undirected graph file (single directed edge per
line) (default:'../as20graph.txt')
-k:Minimal clique overlap size (default:3)
-o:Output file prefix (default:'')
```

Usage

```
./cliques -i:../as20graph.txt -k:2 -o:as20
```

Application: kronfit

Description

KronFit is an estimate Kronecker graphs initiator matrix. It is a fast and scalable algorithm for fitting the Kronecker graph generation model to large real networks. A naive approach to fitting would take super-exponential time. In contrast, KronFit takes linear time. KronFit finds accurate parameters that very well mimic the properties of target networks. In fact, using just four parameters we can accurately model several aspects of global network structure.

For more information about the procedure see: *Kronecker Graphs: an approach to modeling networks Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, Zoubin Ghahramani.* *http://arxiv.org/abs/0812.4905*

Parameters

```
-i:Input graph file (single directed edge per
line)(default:'../as20graph.txt')
-o:Output file prefix (default:'')
-n0:Innitiator matrix size (default:2)
-m:Init Gradient Descent Matrix (R=random) (default:'0.9 0.7; 0.5 0.2')
-p:Initial node permutation: d:Degree, r:Random, o:Order (default:'d')
-gi:Gradient descent iterations (default:50)
-l:Learning rate (default:1e-05)
-mns:Minimum gradient step (default:0.005)
-mxs:Maximum gradient step (default:0.05)
-w:Samples to warm up (default:10000)
-s:Samples per gradient estimation (default:100000)
-sim:Scale the initiator to match the number of edges (default:'T')
-nsp:Probability of using NodeSwap (vs. EdgeSwap) MCMC proposal
distribution (default:1)
```

Usage

Estimate the 2-by-2 Kronecker initiator matrix for the Autonomous Systems network using 100 gradient descent iterations. We initialize the fitting with the [0.9 0.6; 0.6 0.1] initiator matrix.

```
./kronfit -i:../as20graph.txt -n0:2 -m:"0.9 0.6; 0.6 0.1" -
```

Application: krongen

Description

This example is a Kronecker graphs graph generator. Kronecker graphs is a generative network model which obeys all the main static network patterns that have appeared in the literature. The model also obeys recently discovered temporal evolution patterns like shrinking diameter and densification power law. Kronecker graphs also lead to tractable analysis and rigorous proofs. The model is based on a matrix operation, the Kronecker product, and produces networks with heavy-tailed distributions for in-degree, out-degree, eigenvalues, and eigenvectors.

Given an initiator matrix M the application generates a corresponding Kronecker graph. If you want to estimate M for a given graph G use the 'kronfit' application.

For more information about the procedure see: *Kronecker Graphs: an approach to modeling networks Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, Zoubin Ghahramani.* *http://arxiv.org/abs/0812.4905*

Depending on the platform (Windows or Linux) you need to edit the Makefile. Use 'make opt' to compile the optimized (fast) version of the code.

Parameters

```
-o:Output graph file name (default:'graph.txt')
-m:Matrix (in Maltab notation) (default:'0.9 0.5; 0.5 0.1')
-i:Iterations of Kronecker product (default:5)
-s:Random seed (0 - time seed) (default:0)
```

Usage

Generate a Stochastic Kronecker graph on 1024 (2^10) nodes with the initiator matrix [0.9 0.6; 0.6 0.1]

```
./krongen -o:kronecker_graph.txt -m:"0.9 0.6; 0.6 0.1" -i:10
```

Application: motifs

Description

Motifs is a fast and scalable algorithm for counting frequency of connected induced subgraphs in a network. The program counts the number of occurences of a every possible connected subgraph on K nodes in a given graph. Frequency of such network motifs can be used to compare and characterize networks.

The algorithm is described in *Efficient Detection of Network Motifs by Sebastian Wernicke. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2006.*
For information about network motifs refer to *Network motifs: Simple building blocks of complex networks by R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii and U. Alon. Science, October 2002.*

Depending on the platform (Windows or Linux) you need to edit the Makefile. Use 'make opt' to compile the optimized (fast) version of the code.

Parameters

```
-i:Input directed graph file (single directed edge per
line) (default:'../as20graph.txt')
-m:Motif size (has to be 3 or 4) (default:3)
-d:Draw motif shapes (requires GraphViz) (default:'T')
-o:Output file prefix (default:'')
```

Nodes of the graph have to be numbered 0...N-1

Usage

```
./motifs -i:../as20graph.txt -m:3 -d:T -o:as-3motifs
```

Application: ncplot

Description

This example creates the network community profile plot. Network Community Profile plot measures the score of ``best" community as a function of community size in a network. Formally, we define it as the conductance value of the minimum conductance set of cardinality k in the network, as a function of k. As defined, the NCP plot will be NP-hard to compute exactly, so operationally we will use several natural approximation algorithms for solving the Minimum Conductance Cut Problem in order to compute different approximations to it.

The shape of the plot offers insights into the large scale community structure of networks. Networks with nice and/or hierarchical community structure have a downward sloping NPC plot. Random graphs have flat NCP plot. Large real networks tend to have V shaped (down and up) NCP plot which illustrates not only tight communities at very small scales, but also that at larger and larger size scales the best possible communities gradually ``blendin" more and more with the rest of the network and thus gradually become less and less community-like.

For more information about the procedure see: *Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters Jure Leskovec, Kevin Lang, Anirban Dasgupta, Michael Mahoney.* *http://arxiv.org/abs/0810.1355*

Parameters

```
-i:Input undirected graph (one edge per line)
(default:'as20graph.txt')
-o:Output file name (default:'')
-d:Description (default:'')
-d:Draw largest D whiskers (default:-1)
-k:Take core (strip away whiskers) (default:'F')
-w:Do bag of whiskers (default:'F')
-r:Do rewired network (default:'F')
-s:Save info file (for each size store conductance,
modulariy, ...) (default:'F')
```

Usage

```
./ncpplot -i:as20graph.txt -s:T
```

Application: netevol

Description

This example computes how various statistical properties of a network change over time.

The program loads a sequence of snapshots and produces plots of densification power law, shrining diameter, fraction of nodes in largest connected component over time and similar.

Depending on the platform (Windows or Linux) you need to edit the Makefile. Use 'make opt' to compile the optimized (fast) version of the code.

Parameters

```
-i:Input graphs (each file is a graph snapshot, or use
"DEMO") (default:'graph*.txt')
-o:Output file name prefix (default:'over-time')
-t:Description (default:'')
-s:How much statistics to calculate?
1:basic, 2:degrees, 3:no-diameter, 4:no-distributions,
5:no-svd, 6:all-statistics (default:6)
```

Generally -s:1 is the fastest (computes the least statistics), while -s:6 takes longest to run but calculates all the statistics.

Usage

```
./netstat -i:DEMO
```

Application: netstat

Description

This example computes the structural properties of a network:

a. cumulative degree distribution
b. degree distribution
c. hop plot (diameter)
d. distribution of weakly connected components
e. distribution of strongly connected components
f. clustering coefficient
g. singular values
h. left and right singular vector

Depending on the platform (Windows or Linux) you need to edit the Makefile.
Use 'make opt' to compile the optimized (fast) version of the code.

Parameters

```
-i:Input graph (one edge per line, tab/space separated)
(default:'graph.txt')
-d:Directed graph (default:'T')
-o:Output file prefix (default:'graph')
-t:Title (description) (default:'')
-p:What statistics to plot string:
        c: cummulative degree distribution
        d: degree distribution
        h: hop plot (diameter)
        w: distribution of weakly connected components
        s: distribution of strongly connected components
        C: clustering coefficient
        v: singular values
        V: left and right singular vector
```

Usage

```
./netstat -i:../as20graph.txt -p:dC
```

7. Using Snap

Reference

When you write your code, it's essential to make references first to snap library. For an example:

```
#include "../snap/Snap.h"
```

This is the main include file of the library.

Graph Constructor

You can create a graph by calling this constructor:

```
PNGraph Graph = TNGraph::New();
```

Which creates an empty directed graph G.

SNAP supports three graph types:

TUNGraph: undirected graph (single edge between an unordered pair of nodes)
TNGraph: directed graph (single directed edge between an ordered pair of nodes)
TNEGraph: directed multi-graph (any number of directed edges between a pair of nodes)

```
  //// what type of graph do you want to
use?
  PUNGraph PGraph;  // undirected graph
  PNGraph PGraph;   // directed graph
  PNEGraph PGraph;  // directed multigraph
```

Random Graph Constructor

If you want to create a random undirected graph with 200 nodes and 500 edges you can use the following code:

```
PUNGraph Graph = TSnap::GenRndGnm<PUNGraph>(200, 500);
```

Adding Nodes and Edges

Adding nodes and edges to graph is really easy and can be accomplished with the following code:

```
// create a graph
PNGraph Graph = TNGraph::New();

for (int n = 0; n < 10; n++)
  {
    G->AddNode(); // if no parameter is given, edge ids are 0,1,...,9
  }


Graph->AddNode(15); //add a note with id 15
Graph->AddNode(32);

Graph->AddEdge(0,1);
Graph->AddEdge(15,32);


const int NId1 = G->GetRndNId(); //get random node id
const int NId2 = G->GetRndNId();
if (G->AddEdge(NId1, NId2) != -2) {
  printf("  Edge %d -- %d added\n", NId1,  NId2); }
else {
   printf("  Edge %d -- %d already exists\n", NId1, NId2); }
}
```

Nodes have explicit (and arbitrary) node ids. There is no restriction for node ids to be contiguous integers starting at 0. In a multi-graph TNEGraph edges have explicit integer ids. In TUNGraph and TNGraph edges have no explicit ids -- edges are identified by a pair node ids.

SNAP uses smart-pointers (TPt) so there is no need to explicitly free (delete) graph objects. They self-destruct when they are not needed anymore. Prefix P in the class name stands for a pointer, while T means a type.

You can also access a node using **constructor** `TNodeI`

```
PUNGraph::TObj::TNodeI NI = G->GetNI(0) //get node with id 0
int e = NI.GetDeg() //get the degree of node NI
```

Deleting Nodes and Edges

Deleting nodes and edges is easy and can be accomplished with the following code:

```
Graph::TObj::TNodeI NI = G->GetNI(0);
printf("Del edge %d -- %d\n", NI.GetId(), NI.GetOutNId(0)); //delete edge
G->DelEdge(NI.GetId(), NI.GetOutNId(0));

const int RndNId = G->GetRndNId();
printf("Delete node %d\n", RndNId); //delete node
G->DelNode(RndNId);
```

Iterators

SNAP heavily relies on the user of *iterators* that allow for fast traversals over the nodes or edges. For example:

```
// create a directed random graph on 100 nodes and 1k edges
  PNGraph Graph = TSnap::GenRndGnm<PNGraph>(100, 1000);

// traverse the nodes
  for (TNGraph::TNodeI NI = Graph->BegNI(); NI < Graph->EndNI(); NI++) {
    printf("node id %d with out-degree %d and in-degree %d\n",
      NI.GetId(), NI.GetOutDeg(), NI.GetInDeg());
  }

// traverse the edges
  for (TNGraph::TEdgeI EI = Graph->BegEI(); EI < Graph->EndEI(); EI++) {
    printf("edge (%d, %d)\n", EI.GetSrcNId(); EI.GetDstNId());
  }

// we can traverse the edges also like this
  for (TNGraph::TNodeI NI = Graph->BegNI(); NI < Graph->EndNI(); NI++) {
    for (int e = 0; e < NI.GetOutDeg(); e++) {
      printf("edge (%d %d)\n", NI.GetId(), NI.GetOutNId(e));
    }
  }
```

All graph and network datatypes define node and edge iterators. In general graph/network data types use the following functions to return various iterators:

```
  BegNI(): iterator to first node
  EndNI(): iterator to one past last node
  GetNI(u): iterator to node with id u
  BegEI(): iterator to first edge
  EndEI(): iterator to one past last edge
  GetEI(u, v): iterator to edge (u, v)
  GetEI(e): iterator to edge with id e (only for multigraphs)
```

In general node iterators provide the following functionality:

```
 GetDat(): return data type TNodeData associated with the node
 GetOutNDat(e): return data associated with node at endpoint of
e-th out-edge
 GetInNDat(e): return data associated with node at endpoint of
e-th in-edge
 GetOutEDat(e): return data associated with e-th out-edge
 GetInEDat(e): return data associated with e-th in-edge
```

In addition, iterators over networks also allow for easy access to the data:

```
GetId(): return node id
GetOutDeg(): return out-degree of a node
GetInDeg(): return in-degree of a node
GetOutNId(e): return node id of the endpoint of e-th out-edge
GetInNId(e): return node id of the endpoint of e-th in-edge
IsOutNId(int NId): do we point to node id n
IsInNId(n): does node id n point to us
IsNbhNId(n): is node n our neighbor
```

More functions can be found in files `graph.h` and `network.h`

Loading and Saving Graphs

With SNAP it is easy to save and load networks in various formats. Internally SNAP saves networks in compact binary format but functions for loading and saving networks in various other text and XML formats are also available (see gio.h)

```
// generate a network using Forest Fire model
PNGraph G = TSnap::GenForestFire(1000, 0.35, 0.35);

// save and load binary
G->Save(TFOut("test.graph"));
PNGraph G2 = TNGraph::Load(TFIn("test.graph"));

// save and load from a text file
TSnap::SaveEdgeList(G2, "test.txt", "Save as tab-separated list of edges");
PNGraph G3 = TSnap::LoadEdgeList("test.txt", 0, 1);
```

Converting Graphs

Functions are implemented as a part of namespace TSnap. All functions support any graph/network type.

```
// generate a network using Forest Fire model
PNGraph G = TSnap::GenForestFire(1000, 0.35, 0.35);

// convert to undirected graph TUNGraph
PUNGraph UG = TSnap::ConvertGraph<PUNGraph>(G);

// get largest weakly connected component of G
PNGraph WccG = TSnap::GetMxWcc(G);

// get a subgraph induced on nodes {0,1,2,3,4,5}
PNGraph SubG = TSnap::GetSubGraph(G, TIntV::GetV(0,1,2,3,4));

// get 3-core of G
PNGraph Core3 = TSnap::GetKCore(G, 3);

// delete nodes of degree 10
TSnap::DelDegKNodes(G, 10);
```

Computing structural properties

SNAP provides rich functionality to efficiently compute structural properties of networks. Functions are implemented as a part of namespace TSnap. All functions support any graph/network type.

```
  // get distribution of connected components (component size, count)
  TVec<TPair<TInt, TInt> > CntV; // vector of pairs of integers (size,
count)
  TSnap::GetWccSzCnt(G, CntV);
  // get degree distribution pairs (degree, count)
  TSnap::GetOutDegCnt(G, CntV);
  // get first eigenvector of graph adjacency matrix
  TFltV EigV; // vector of floats
  TSnap::GetEigVec(G, EigV);
  // get diameter of G
  TSnap::GetBfsFullDiam(G);
  // count the number of triads in G, get the clustering coefficient of G
  TSnap::GetTriads(G);
  TSnap::GetClustCf(G);
```

8.The Snap Library

The Snap library can be found in the directory SnapLib. The functionalities of SNAP are implemented in various files, based on the type or category of the functionality. Below we list the files and a description of the functionalities. Further down we take a closer look at all functionalities at each file.

File	Description
`alg.h`	Simple algorithms like counting node degrees, simple graph manipulation (adding/deleting self edges, deleting isolated nodes) and testing whether graph is a tree or a star
`anf.h`	Approximate Neighborhood Function: linear time algorithm to approximately calculate the diameter of massive graphs
`arxiv.h`	Functions for parsing Arxiv data and standardizing author names
`bfsdfs.h`	Algorithms based on Breath First Search (BFS) and Depth First Search (DFS): shortest paths, spanning trees, graph diameter, and similar
`bignet.h`	Memory efficient implementation of a network with data on nodes. Use when working with very large networks.
`cmty.h`	Algorithsm for network community detection: Modularity, Girvan-Newman, Clauset-Newman-Moore
`centr.h`	Node centrality measures: closeness, betweenness, PageRank, ...
`cliques.h`	Maximal clique detection and Clique Percolation method.
`cncom.h`	Connected components: weakly, strongly and bi connected components, articular nodes and bridge edges
`dblp.h`	Parser for XML dump of DBLP data
`ff.h`	Forest Fire model for generating networks that densify and have shrinking diameters
`gbase.h`	Defines flags that are used to identify functionality of graphs
`ggen.h`	Various graph generators: random graphs, copying model, preferential attachment, RMAT, configuration model, Small world model

ghash.h	Hash table with directed graphs (TNGraph) as keys. Uses efficient adaptive approximate graph isomorphism testing to scale to large graphs. Useful when one wants to count frequencies of various small subgraphs or cascades.
gio.h	Graph input output. Methods for loading and saving various textual and XML based graph formats: Pajek, ORA, DynNet, GraphML (GML), Matlab
graph.h	Implements graph types TUNGraph, TNGraph and TNEGraph
gstat.h	Computes many structural properties of static and evolving networks.
gsvd.h	Eigen and singular value decomposition of graph adjacency matrix
gviz.h	Interface to GraphViz.
imdbnet.h	Actors-to-movies bipartite network of IMDB.
kcore.h	K-core decomposition of networks
kronecker.h	Kronecker Graph generator and KronFit algorithm for estimating parameters of Kronecker graphs
ncp.h	Network community profile plot. Implements local spectral graph partitioning method to efficiently find communities in networks
network.h	Implements network types TNodeNet, TNodeEDatNet and TNodeEdgeNet
signnet.h	Networks with signed (+1, -1) edges that can denote trust/distrust between the nodes of the network
subgraphenum.h	Sub-graph enumeration and network motif computations
sir.h	SIR epidemic model and SIR parameter estimation
Snap.h	Main include file of the library
spinn3r.h	Past parser for loading blog post data from Spinn3r
statplot.h	Plots of various structural network properties: clustering, degrees, diameter, spectrum, connected components
subgraph.h	Extracting subgraphs and converting between different graph/network types
timenet.h	Temporally evolving networks
trawling.h	Algorithm of extracting bipartite cliques from the network
triad.h	Functions for counting triads (triples of connected nodes in the network) and computing clustering coefficient
util.h	Utilities to manipulate PDFs, CDFs and CCDFs. Quick and dirty string manipulation, URL and domain manipulation routines
wgtnet.h	Weighted networks
wikinet.h	Networks based on Wikipedia