ARISTOTLE UNIVERSITY OF THESSALONIKI

Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Εργασία για το μάθημα Θεωρία Δικτύων

Γενάρης 2024

Μπέκου Βασιλική

ΑΕΜ: 10524

mpekvasi@ece.auth.gr

Ανάλυση εντοπισμού Κοινοτήτων χρησιμοποιώντας συνάρτηση ποιότητας απόστασης

Εισαγωγή

Αυτή η αναφορά παρουσιάζει την υλοποίηση και την ανάλυση ενός αλγορίθμου εντοπισμού κοινοτήτων που βασίζεται στη συνάρτηση ποιότητας απόστασης για την ανάλυση δικτύων (distance quality function). Ο αλγόριθμος υλοποιήθηκε σε Python με χρήση της βιβλιοθήκης NetworkX και συγκρίθηκε με την παραδοσιακή μέθοδο εντοπισμού κοινοτήτων σε δίκτυα (modularity). Η υλοποίηση δοκιμάστηκε στο ενδεδειγμένο από την εκφώνηση δίκτυο (**dataset email-Eu-core)**, το οποίο περιλαμβάνει 1005 κόμβους και 25.571 ακμές, ενώ τα αποτελέσματα οπτικοποιήθηκαν με χρήση του Gephi.

1. Θεωρητική βάση και εξέλιξη υλοποίησης

1.1 Αρχική συνάρτηση ποιότητας απόστασης

Σύμφωνα με τη θεωρία η αρχική συνάρτηση για την ποιότητα απόστασης (distance quality) ορίζεται ως εξής:

$$Pr[i,j,k] = dk(i) * dk(j) / (2mk(G) * 2mk(G))$$
$$\bar{D}_v(i,j) = \Sigma(k=1 \text{ to } diam(G))\ k * Pr[i,j,k]$$
$$Qd(G,C) = \Sigma(C \in C)\ (\bar{D}_v(C) - D_v(C))$$

Πρακτικά αποσκοπεί:

- Στη σύγκριση πραγματικών αποστάσεων εντός των κοινοτήτων με τις αναμενόμενες αποστάσεις.
- Στον εντοπισμό κοινοτήτων όπου οι κόμβοι είναι πιο κοντά μεταξύ τους απ' όσο θα αναμενόταν τυχαία.
- Στην παροχή ενός μέτρου ποιότητας που μπορεί να βελτιστοποιηθεί για την ανεύρεση καλών κοινοτικών δομών.

1.2 Προκλήσεις υλοποίησης

Η υλοποίηση αυτής της συνάρτησης για τον αλγόριθμο εύρεσης κοινοτήτων ειδικά για μεγάλα δίκτυα απαιτεί μεγάλη μνήμη και παρουσιάζει ιδιαίτερη υπολογιστική πολυπλοκότητα διότι υπολογίζει όλες τις συντομότερες διαδρομές: O(V × (V + E)) και την πιθανότητα διαδρομών για όλα τα ζεύγη κόμβων.

Στο κομμάτι της βελτιστοποίησης ιδιαίτερη πρόκληση ήταν η διατήρηση της ισορροπίας μεταξύ ποιότητας και χρόνου υπολογισμού αλλά και η διαχείριση ασύνδετων συνιστωσών

1.3 Εξέλιξη λύσης

Αρχική προσέγγιση (Άμεση υλοποίηση):

Η αρχική προσέγγιση ήταν η άμεση υλοποίηση της δοθείσας συνάρτησης όπως φαίνεται παρακάτω, αλλά πρόκειται για μια πολύ υπολογιστικά δαπανηρή διαδικασία ειδικά για μεγαλύτερα δίκτυα οπότε εγιναν μερικές τροποποιήσεις.

```
def calculate_path_probability(i, j, k):
    dk_i = len([v for v in G[i] if shortest_paths[i][v] == k])
    dk_j = len([v for v in G[j] if shortest_paths[j][v] == k])
    mk = total_paths_length_k(k)
    return (dk_i * dk_j) / (4 * mk * mk)
```

Τροποποιημένη συνάρτηση ποιότητας (distance quality)

Η υλοποίησή μου προσαρμόζει τον αρχικό τύπο, διατηρώντας τις βασικές αρχές της συνάρτησης.

Επεξεργασία εισόδου:

- Δέχεται μια λίστα από κοινότητες (κάθε κοινότητα είναι ένα σύνολο από IDs κόμβων).
- Δέχεται ένα dictionary με προϋπολογισμένες συντομότερες διαδρομές
- Επιστρέφει έναν float δείκτη που υποδεικνύει την ποιότητα της δομής των κοινοτήτων.

```
def distance_quality(self, communities: List[Set[int]], sample_paths: Dict) -> float
```

Για κάθε κοινότητα, υπολογίζει διάφορους παράγοντες:

a. **Εσωτερική πυκνότητα (Internal density)**:
   Υπολογίζει τις πραγματικές ακμές μέσα στην κοινότητα και διαιρεί με τις δυνατές ακμές για να βρει την πυκνότητα.

```
internal_edges = sum(1 for i, j in combinations(cluster, 2)
                     if j in self.DG[i] or i in self.DG[j])
possible_edges = len(cluster) * (len(cluster) - 1) / 2
density = internal_edges / possible_edges if possible_edges > 0 else 0
```

b. **Διαγωγιμότητα (Conductance)**:
Μετρά πόσο «απομονωμένη» είναι η κοινότητα. Χαμηλή διαγωγιμότητα σημαίνει λιγότερες εξωτερικές συνδέσεις.

```
conductance = self.calculate_community_conductance(cluster)
```

c. **Ποινή μεγέθους (Size penalty)**:
Επιβάλλει ποινή σε κοινότητες που είναι πολύ διαφορετικές από το μέσο μέγεθος

```
size_ratio = len(cluster) / avg_community_size
size_penalty = math.log(size_ratio + 1) + 1
```

Για κάθε ζεύγος κόμβων στην κοινότητα:

- **path_quality:** Υψηλότερη βαθμολογία για μικρότερες αποστάσεις.
- **density_factor:** Υψηλότερη βαθμολογία για πυκνές, καλά απομονωμένες κοινότητες. Επιβραβεύει καλά συνδεδεμένες κοινότητες.
- **size_factor:** Ισορροπεί τα μεγέθη των κοινοτήτων.

```
path_quality = (self.diameter - actual_dist) / self.diameter
density_factor = (density + 0.1) * (1.2 - conductance)
size_factor = 1 / size_penalty
```

Τελικό σκορ ποιότητας:

Υπολογίζει τον μέσο όρο της ποιότητας για όλα τα ζεύγη και προσθέτει bonus για μικρότερο συνολικό αριθμό κοινοτήτων

```
community_factor = math.exp(-len(communities) / 100)
return (quality / max(1, total_pairs)) * (1 + community_factor)
```

Σύνοψη στόχου:

Η συνάρτηση είναι υλοποιημένη έτσι ώστε να προτιμά κοινότητες όπου:

- Οι κόμβοι βρίσκονται κοντά μεταξύ τους (μικρές αποστάσεις).
- Υπάρχουν πολλές εσωτερικές συνδέσεις (υψηλή πυκνότητα).
- Υπάρχουν λίγες εξωτερικές συνδέσεις (χαμηλή διαγωγιμότητα).
- Τα μεγέθη είναι ισορροπημένα.
- Ο συνολικός αριθμός κοινοτήτων είναι λογικός.

Διαφορές με την αρχική μαθηματική έκφραση:

1. Χρησιμοποιεί πραγματικά μήκη διαδρομών αντί για αναμενόμενες αποστάσεις βάσει πιθανότητας.
2. Προσθέτει ρητούς παράγοντες για την πυκνότητα και τη διαγωγιμότητα.
3. Περιλαμβάνει εξισορρόπηση μεγέθους και αριθμού κοινοτήτων.

Αιτιολόγηση των τροποποιήσεων

- **Υπολογιστική πολυπλοκότητα**: Η αρχική συνάρτηση απαιτεί τον υπολογισμό όλων των πιθανοτήτων διαδρομής, για ενα δίκτυο με 1005 κόμβους και 25571 ακμές, αυτό θα ήταν υπολογιστικά πολύ απαιτητικό. Με τις τροποποιήσεις μειώνεται η πολυπλοκότητα ενώ παράλληλα διατηρείται η ουσία της συνάρτησης.
- **Οφέλη απόδοσης**: Μειωμένος χρόνος εκτέλεσης από μέρες σε περίπου 1-2 ώρες και χαμηλότερη κατανάλωση μνήμης.
- **Πρακτικότητα:** Η προσθήκη μέτρων πυκνότητας και αγωγιμότητας διασφαλίζει ουσιαστικές κοινότητες. Ενσωμάτωση ποινών μεγέθους για αποφυγή πολύ μικρών ή μεγάλων κοινοτήτων και χρήση δειγματοληψίας για διαχείριση μεγάλων δικτύων.

2. Επισκόπηση αλγορίθμου

Ο υλοποιημένος αλγόριθμος ανιχνεύει κοινότητες σε κατευθυνόμενα δίκτυα χρησιμοποιώντας μια συνάρτηση ποιότητας απόστασης, συγκρίνοντας τα αποτελέσματα με τη μέθοδο αρθρωτότητας (modularity). Ο αλγόριθμος επικεντρώνεται στη βελτιστοποίηση της δομής των κοινοτήτων με βάση τις αποστάσεις μεταξύ των κόμβων.

2.1 Ροή αλγορίθμου

1. Αρχικοποίηση

Φόρτωση δικτύου(email):

- Ανάγνωση του δικτύου από αρχείο edge list
- Δημιουργία κατευθυνόμενου γράφου (DG) με το NetworkX
- Μετατροπή σε μη κατευθυνόμενο γράφο (G) για υπολογισμούς απόστασης
- Φόρτωση ετικετών τμημάτων (departments) για τους κόμβους

Υπολογισμός της διαμέτρου του δικτύου:

- Εάν το δίκτυο είναι συνεκτικό: χρήση της συνάρτησης diameter του NetworkX
- Εάν είναι μη συνεκτικό: χρήση της διαμέτρου του μεγαλύτερου συνδετικού υπογράφου.

Υπολογισμός δειγματοληπτικών συντομότερων διαδρομών

- Δειγματοληψία έως 1005 κόμβους(εδώ όλο το δικτυο)
- Υπολογισμός συντομότερων διαδρομών από τους δειγματοληπτικούς κόμβους προς όλους τους άλλους

2. Διαδικασία ανίχνευσης κοινοτήτων (find communities)

Αρχικά ξεκινάμε με κοινότητες ενός κόμβου, υπολογίζουμε τον αρχικό δεικτη ποιοτητας και τον αποθηκεύουμε για συγκριση.

Κύριος βρόχος(while loop):

Εύρεση ζευγών κοινοτήτων που μπορούν να συγχωνευτούν:

- Για κάθε ζεύγος κοινοτήτων (i,j):
    - Καταμέτρηση άμεσων συνδέσεων.
    - Υπολογισμός πυκνότητας σύνδεσης.
    - Υπολογισμός ισορροπίας μεγέθους.
    - Υπολογισμός ομοιότητας πυκνότητας.
    - Βαθμολόγηση της πιθανής συγχώνευσης.
- Ταξινόμηση ζευγών κατά σειρά βαθμολογίας.

Δοκιμή συγχωνεύσεων:

- Για κάθε υποψήφιο ζεύγος:
    - Δημιουργία συγχωνευμένης κοινότητας.
    - Έλεγχος περιορισμών μεγέθους.
    - Υπολογισμός νέας ποιότητας.
    - Αποδοχή εάν η ποιότητα βελτιώνεται ή πληροί το όριο.

Ενημέρωση Κατάστασης:

- Εάν γίνει αποδεκτή συγχώνευση:
    - Ενημέρωση λίστας κοινοτήτων.
    - Ενημέρωση δείκτη ποιότητας.
    - Επαναφορά μετρητή βελτίωσης.
- Εάν δεν γίνει αποδεκτή συγχώνευση:
    - Αύξηση μετρητή μη βελτίωσης.
    - Δοκιμή αναγκαστικών συγχωνεύσεων αν η διαδικασία κολλήσει

3. Υπολογισμός distance quality

Για κάθε κοινότητα υπολογίζει τις βασικές μετρικές (εσωτερική πυκνοτητα, conductance, σχετικό μεγεθος).

Για κάθε ζεύγος κόμβων υπολογίζεται η πραγματική αποσταση, το σκορ διαδρομής = (διάμετρος - actual_dist) / διάμετρος και γίνεται εφαρμογή πυκνότητας και μεγέθους.

4. Συγχωνεύσεις

Όταν η διαδικασία κολλάει:

- Εντοπισμός μικρότερων κοινοτήτων.
- Αναγκαστική συγχώνευση εάν είναι πάνω από το ελάχιστο μέγεθος.

3. Πειραματικά αποτελέσματα

Μετά την εκτέλεση του αλγορίθμου τα αποτελέσματα φαίνονται παρακάτω:

Modularity

- Κοινότητες: 28
- Δείκτης Modularity: 0.4315
- Γρήγορη εκτέλεσγ

```
Detecting communities using Modularity...
Modularity Results:
Found 28 communities
Modularity score: 0.4315

Exporting to modularity_communities.gexf
```

Distance quality

- Κοινότητες: 50
- Δείκτης ποιότητας: 0.1170
- Χρόνος: 1h 58m 48s

```
Distance Quality Results:
Found 50 communities
Quality score: 0.1170

Exporting to distance_communities.gexf

Analysis complete!
Total execution time: 1h 58m 48s
Results exported to:
- modularity_communities.gexf
- distance_communities.gexf
```

Ανάλυση αποτελεσμάτων:

Η μέθοδος modularity βρήκε λιγότερες κοινότητες (28) σε σύγκριση με τη distance quality (50), αυτό υποδηλώνει ότι η μέθοδος distance quality τείνει να εντοπίζει πιο λεπτομερείς δομές κοινοτήτων.

Αναφορικά με τους δεικτες δεν μπορουμε να τους συγκρίνουμε αμεσα καθως χρησιμοποιουν διαφορετικές μετρικές παρολα αυτά ο δείκτης modularity (0.4315) είναι αρκετά καλός (κυμαίνεται από -1 έως 1) και ο θετικός δείκτης distance quality δείχνει την ύπαρξη ουσιαστικών κοινοτήτων.

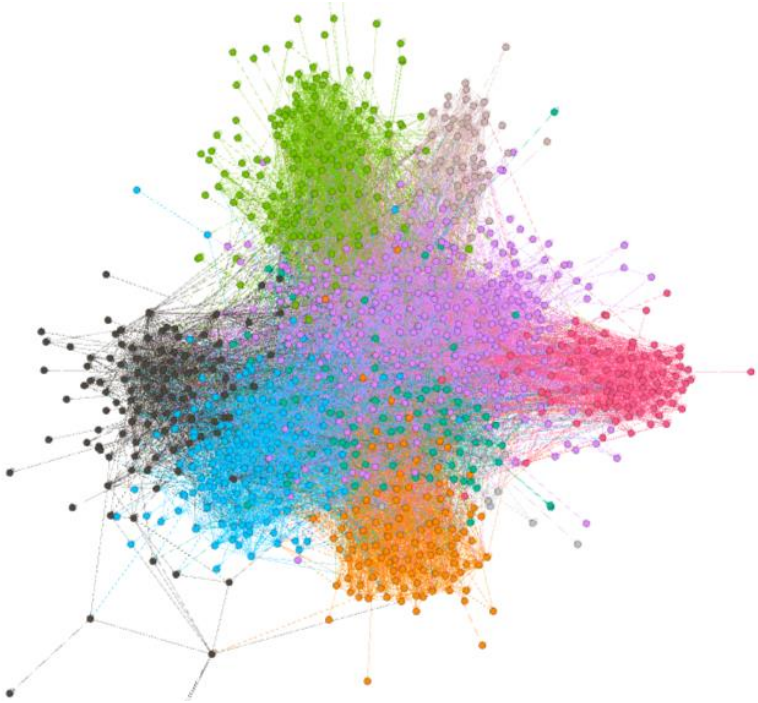
Ο αλγόριθμός είναι αποδοτικός καθως παρουσιάζει λογικό χρόνο εκτέλεσης(1-2ωρες) και καλή συμπεριφορά σύγκλισης.

Τα αποτελέσματα και των δύο μέθοδών είναι συγκρίσιμα

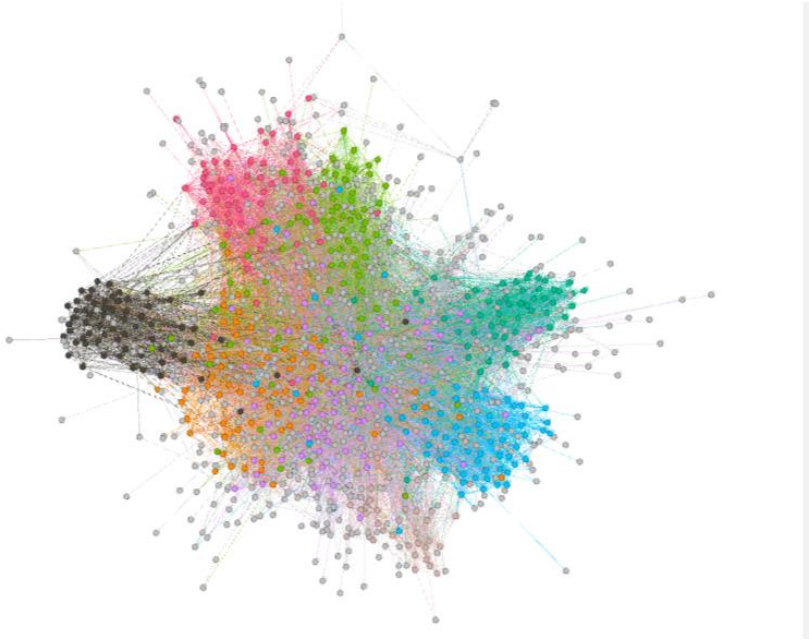
4. Οπτικοποιηση αποτελεσμάτων

Για την οπτικοποίηση των κοινοτήτων που παράγει ο αλγόριθμος μου έγινε χρήση του Gephi.

Modularity function:



Distance quality function:



Παρατηρήσεις

Η οπτικοποίηση δείχνει σαφείς, διακριτές κοινότητες με διαφορετικά χρώματα. Οι κοινότητες φαίνονται καλά διαχωρισμένες.
Ο μεγαλύτερος αριθμός κοινοτήτων που προκύπτει από τη μέθοδο distance quality μπορεί να καταγράφει υπο-δομές που η μέθοδος Modularity να συγχώνευσε.

5. Συμπεράσματα

Ο αλγόριθμος κατάφερε με επιτυχία να προσαρμόσει τη θεωρητική συνάρτηση ποιότητας απόστασης για πρακτική χρήση, να επιτύχει καλή ισορροπία μεταξύ ακρίβειας και απόδοσης παράγοντας ουσιαστική δομή κοινοτήτων.
Τέλος εξάγει επιτυχώς τα αποτελέσματα σε μορφή gexf για οπτικοποιηση στο Gephi.
Η συγκριση των δύο μεθόδων δείχνει ότι και οι δύο μέθοδοι εντόπισαν ουσιαστικές δομές κοινοτήτων, αλλά δίνουν έμφαση σε διαφορετικές πτυχές:

- **Modularity**: Εστιάζει στην πυκνότητα των συνδέσεων σε σχέση με αυτό που θα περιμέναμε κατά τύχη
- **Distance quality**: Εστιάζει στα μήκη των διαδρομών και την εσωτερική συνδεσιμότητα

Τα αποτελέσματα δείχνουν ότι η προσέγγιση ποιότητας απόστασης είναι βιώσιμη για την ανίχνευση κοινοτήτων σε πραγματικά δίκτυα, προσφέροντας μια εναλλακτική λύση στις μεθόδους βάσει αρθρωτότητας(modularity), ενώ διατηρεί λογικές απαιτήσεις υπολογισμού.