Καθηγητής Π. Λουρίδας

Τμήμα Διοικητικής Επιστήμης και Τεχνολογίας

Οικονομικό Πανεπιστήμιο Αθηνών

Μεγιστοποίηση Επιρροής

Μια εφαρμογή των κοινωνικών δικτύων είναι η μέτρησης της *επιρροής* (influence) που μπορεί να ασκηθεί στο δίκτυο από κάποια μέλη του. Η ιδέα είναι ότι κάποιοι στο κοινωνικό δίκτυο μπορούν να επηρεάζουν κάποιους άλλους—εξού και ο όρος «επηρεαστής» (influencer). Αν λοιπόν θέλαμε να επηρεάσουμε όσο το δυνατόν περισσότερα μέλη ενός κοινωνικού δικτύου, και μπορούσαμε για το σκοπό αυτό να χρησιμοποιήσουμε έναν συγκεκριμένο αριθμό από επηρεαστές, πώς θα τους επιλέγαμε;

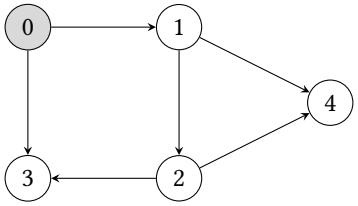
Ένα κοινωνικό δίκτυο το αναπαριστούμε με τη μορφή ενός γράφου $G = (V, E)$. Οι κορυφές του γράφου αντιστοιχούν στα μέλη του κοινωνικού δικτύου, και ένας κατευθυνόμενος σύνδεσμος (u, v) σημαίνει ότι ο κόμβος u κατά κάποιον τρόπο επηρεάζει τον v· για παράδειγμα, μπορεί ο v να ακολουθεί τον u στο κοινωνικό δίκτυο.

Αν έχουμε ένα τέτοιο κοινωνικό δίκτυο, μπορούμε να μοντελοποιήσουμε τη *διάχυση* (diffusion) της επιρροής ως εξής. Έχουμε τον γράφο $G = (V, E)$, και ένα αρχικό σύνολο κόμβων $S \in V$, τους οποίους ονομάζουμε *σπόρους* (seeds). Επιπλέον, ορίζουμε μια πιθανότητα p, η οποία δείχνει πόσο πιθανό είναι ένας κόμβος να επηρεάσει έναν άλλο κόμβο που συνδέεται με αυτόν.

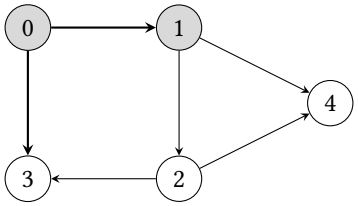
- Σε κάθε κόμβο του γράφου αντιστοιχούμε μια κατάσταση. Οι σπόροι γίνονται «ενεργοί» (active), οι υπόλοιποι «ανενεργοί» (inactive).
- Για κάθε σύνδεσμο του γράφου σημειώνουμε αν τον έχουμε επεξεργαστεί ή όχι. Αρχικά δεν έχουμε επεξεργαστεί κανένα σύνδεσμο.
- Για κάθε ενεργό κόμβο u και κάθε σύνδεσμο (u, v) που δεν έχουμε επεξεργαστεί:
  - Παίρνουμε έναν τυχαίο αριθμό $p_{u,v}$ στο διάστημα $[0, 1]$. Αν $p_{u,v} < p$, τότε κάνουμε τον κόμβο v ενεργό. Σημειώνουμε ότι έχουμε επεξεργαστεί τον σύνδεσμο (u, v).

Το μοντέλο αυτό ονομάζεται *μοντέλο ανεξάρτητης μετάπτωσης* (independent cascade model). Μπορούμε να παρατηρήσουμε ότι αν ένας κόμβος γίνει κάποια στιγμή ενεργός, θα παραμείνει ενεργός. Ένας κόμβος μπορεί να γίνει ενεργός με πιθανότητα p για κάθε σύνδεσμο που έχει από ενεργό κόμβο. Αν δηλαδή σε έναν κόμβο δείχνουν δύο ενεργοί κόμβοι, μπορεί να γίνει ενεργός είτε χάρη στον πρώτον από αυτούς με πιθανότητα p, είτε χάρη στον δεύτερο με πιθανότητα p, ή να μη γίνει καθόλου, με πιθανότητα $1 - p^2$.

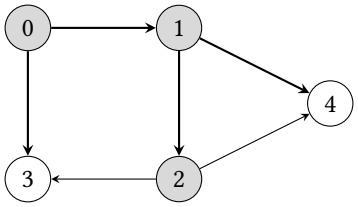
Μπορούμε να δούμε πώς εξελίσσεται η διαδικασία σε ένα παράδειγμα. Έστω ότι έχουμε τον παρακάτω γράφο, και ξεκινάμε με έναν σπόρο, τον κόμβο 0.



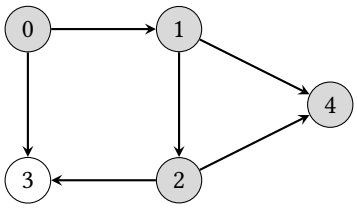
Επεξεργαζόμαστε τους συνδέσμους $(0, 1)$ και $(0, 3)$. Έστω ότι $p_{0,1} \leq p$ και $p_{0,3} > p$. Τότε κάνουμε τον κόμβο 1 ενεργό.



Έχουμε έναν ενεργό κόμβο, τον κόμβο 1, με σύνδεσμο (για την ακρίβεια σύνδεσμους) που δεν έχουμε επεξεργαστεί, τους $(1, 2)$ και $(1, 4)$. Έστω ότι $p_{1,4} > p$ και $p_{1,2} \leq p$. Τότε κάνουμε τον κόμβο 2 ενεργό.



Έχουμε έναν ενεργό κόμβο, τον κόμβο 2, με δύο συνδέσμους που δεν έχουμε επεξεργαστεί, το σύνδεσμο $(2, 4)$ και το σύνδεσμο $(2, 3)$. Έστω ότι $p_{2,4} \leq p$ και $p_{2,3} > p$. Τότε ο κόμβος (4) γίνεται ενεργός, ενώ ο κόμβος 3 θα παραμείνει ανενεργός. Τελικά επηρεάστηκαν οι τέσσερεις από τους πέντε κόμβους.



Στην πραγματικότητα οι γράφοι των κοινωνικών δικτύων είναι πολύ μεγαλύτεροι, και δεν ξέρουμε ποιους κόμβους πρέπει να επιλέξουμε ως σπόρους. Αυτό μας οδηγεί στον παρακάτω γενικό αλγόριθμο μεγιστοποίησης επιρροής `MaximizeInfluence`.

Η καρδιά του αλγορίθμου είναι η επιλογή του κάθε σπόρου, μέσω της συνάρτησης `SelectSeed`. Υπάρχουν διάφορες προσεγγίσεις να το κάνουμε αυτό. Ιδού δύο από αυτές:

`MaximizeInfluence`$(G, p, k) \to S$

Input: $G = (V, E)$, ένας γράφος με V κόμβους και E συνδέσμους.
p, ένας πραγματικός αριθμός $p \in [0, 1]$ που αντιστοιχεί στην πιθανότητα με την οποία ένας κόμβος μπορεί να επηρεάσει έναν άλλο.
k, ο αριθμός των σπόρων που θέλουμε να επιλέξουμε.

Output: S, ένα σύνολο κόμβων μεγέθους k.

```
1  S = ∅
2  for i = 0 to k do
3      s ← SelectSeed(G, S, p, k)
4      S ← S ∪ {s}
5  return S
```

1. Κάθε φορά επιλέγουμε τον κόμβο με τους περισσότερους εξερχόμενους συνδέσμους, δηλαδή τον μεγαλύτερο βαθμό εξόδου, (out-degree). Σε περίπτωση που δύο κόμβοι έχουν τον ίδιο βαθμό εξόδου, τους επιλέγουμε τον πρώτο σε αύξουσα σειρά του ονόματός τους (δηλαδή με το μικρότερο όνομα).
2. Κάθε φορά επιλέγουμε τον κόμβο που επιτυγχάνει τη μεγαλύτερη αύξηση στην επιρροή. Αυτό είναι μία περίπτωση *λαίμαργου* (greedy) αλγόριθμου: κάθε φορά κάνουμε την επιλογή με τη μεγαλύτερη δυνατή απολαβή. Σε περίπτωση ισοπαλίας, επιλέγουμε τον πρώτο σε αύξουσα σειρά του ονόματος (δηλαδή πάλι με το μικρότερο όνομα).

Για τη δεύτερη προσέγγιση, αν έχουμε το σύνολο S, θέλουμε να επιλέξουμε να προσθέσουμε στο S έναν από τους υπόλοιπους κόμβους, $V - S$. Εξετάζουμε κάθε έναν $u \in V - S$. Υπολογίζουμε την επιρροή των κόμβων του συνόλου S, $f_{\text{inf}}(S)$, την επιρροή των κόμβων του συνόλου $S \cup \{u\}$, $f_{\text{inf}}(S \cup \{u\})$, και επιλέγουμε τον κόμβο με τη μέγιστη αύξηση $f_{\text{inf}}(S \cup \{u\}) - f_{\text{inf}}(S)$.

Εδώ όμως συναντάμε ένα άλλο πρόβλημα: Πώς θα υπολογίσουμε την επιρροή, αφού η διαδικασία είναι στοχαστική, και κόμβοι ενεργοποιούνται με βάση κάποια πιθανότητα, και όχι ντετερμινιστικά; Για να το αντιμετωπίσουμε αυτό, χρησιμοποιούμε τη μέθοδο Monte Carlo. Προσομοιώνουμε τη διαδικασία του μοντέλου ανεξάρτητης μετάπτωσης πολλές φορές, τουλάχιστον 1000, για κάθε σύνολο του οποίου την επιρροή θέλουμε να μετρήσουμε. Σε κάθε προσομοίωση σημειώνουμε την τελική επιρροή, δηλαδή τον αριθμό των ενεργών κόμβων στο τέλος. Ο μέσος όρος των επιρροών αυτών θα μας δώσει μια εκτίμηση για την επιρροή του εν λόγω συνόλου.

Ο υπολογισμός της επιρροής μας είναι χρήσιμος και στην πρώτη προσέγγιση. Παρότι δεν χρειάζεται στην επιλογή του κόμβου, θέλουμε να ξέρουμε την επιρροή κάθε φορά που επιλέγουμε έναν κόμβο, και την τελική επιρροή.

Όσον αφορά τώρα το μοντέλο ανεξάρτητης μετάπτωσης, στην πραγματικότητα η εκτέλεσή του δεν είναι παρά η διάσχιση του γράφου ξεκινώντας από κάθε ενεργό κόμβο u και ακολουθώντας τους συνδέσμους με πιθανότητα p. Για το σκοπό αυτό

μπορούμε να χρησιμοποιήσουμε, ελαφρώς τροποποιημένη, την κατά πλάτος αναζήτηση.

Σκοπός της εργασίας είναι η συγγραφή προγράμματος στη γλώσσα Python με το οποίο θα επιλέγετε τους σπόρους για τη μεγιστοποίηση της επιρροής, σύμφωνα με τη προσέγγιση που θα επιλέγει ο χρήστης.

Απαιτήσεις Προγράμματος

Κάθε φοιτητής θα εργαστεί σε αποθετήριο στο GitHub. Για να αξιολογηθεί μια εργασία θα πρέπει να πληροί τις παρακάτω προϋποθέσεις:

- Για την υποβολή της εργασίας θα χρησιμοποιηθεί το ιδιωτικό αποθετήριο του φοιτητή που δημιουργήθηκε για τις ανάγκες του μαθήματος και του έχει αποδοθεί. Το αποθετήριο αυτό έχει όνομα του τύπου `username-algo-assignments`, όπου `username` είναι το όνομα του φοιτητή στο GitHub. Για παράδειγμα, το σχετικό αποθετήριο του διδάσκοντα θα ονομαζόταν `louridas-algo-assignments` και θα ήταν προσβάσιμο στο https://github.com/dmst-algorithms-course/louridas-algo-assignments. *Τυχόν άλλα αποθετήρια απλώς θα αγνοηθούν.*
- Μέσα στο αποθετήριο αυτό θα πρέπει να δημιουργηθεί ένας κατάλογος `assignment-2024-3`.
- Μέσα στον παραπάνω κατάλογο το πρόγραμμα θα πρέπει να αποθηκευτεί με το όνομα `influence_maximization.py`.
- Δεν επιτρέπεται η χρήση έτοιμων βιβλιοθηκών γράφων ή τυχόν έτοιμων υλοποιήσεων των αλγορίθμων, ή τμημάτων αυτών, εκτός αν αναφέρεται ρητά ότι επιτρέπεται.
- Επιτρέπεται η χρήση δομών δεδομένων της Python όπως στοίβες, λεξικά, σύνολα, κ.λπ.
- Επιτρέπεται η χρήση των παρακάτω βιβλιοθηκών ή τμημάτων τους όπως ορίζεται:
  - `sys.argv`
  - `argparse`
  - `random`. Η συνάρτηση `random.random()` παράγει τυχαίους αριθμός στο διάστημα $[0, 1)$, και όχι στο διάστημα $[0, 1]$ που αναφέραμε παραπάνω. Αυτό δεν έχει κάποια πρακτική σημασία.
  - `collections.deque`
- Το πρόγραμμα θα πρέπει να είναι γραμμένο σε Python 3.
- Η εργασία είναι αποκλειστικά ατομική. Δεν επιτρέπεται συνεργασία μεταξύ φοιτητών στην εκπόνησή της, με ποινή το μηδενισμό. Επιπλέον η εργασία δεν μπορεί να είναι αποτέλεσμα συστημάτων Τεχνητής Νοημοσύνης (όπως ChatGPT). Ειδικότερα όσον αφορά το τελευταίο σημείο προσέξτε ότι τα συστήματα αυτά χωλαίνουν στην αλγοριθμική σχέση, άρα τυχόν προτάσεις που

κάνουν σε σχετικά θέματα μπορεί να είναι λανθασμένες. Επιπλέον, αν θέλετε να χρησιμοποιήσετε τέτοια συστήματα, τότε αν και κάποιος άλλος το κάνει αυτό, μπορεί οι προτάσεις που θα λάβετε να είναι παρόμοιες, οπότε οι εργασίες θα παρουσιάσουν ομοιότητες και άρα θα μηδενιστούν.

- Η έξοδος του προγράμματος θα πρέπει να περιλαμβάνει μόνο ό,τι φαίνεται στα παραδείγματα που παρατίθενται. *Η φλυαρία δεν επιβραβεύεται.*

Χρήση του GitHub

Όπως αναφέρθηκε το πρόγραμμά σας για να αξιολογηθεί θα πρέπει να αποθηκευθεί στο GitHub. Επιπλέον, θα πρέπει το GitHub να χρησιμοποιηθεί καθόλη τη διάρκεια της ανάπτυξης του.

Αυτό σημαίνει ότι *δεν θα ανεβάσετε στο GitHub απλώς την τελική λύση του προβλήματος* μέσω της λειτουργίας "Upload files". Στο GitHub θα πρέπει να φαίνεται *το ιστορικό της συγγραφής του προγράμματος.* Αυτό συνάδει και με τη φιλοσοφία του εργαλείου: λέμε "commit early, commit often". Εργαζόμαστε σε ένα πρόγραμμα και κάθε μέρα, ή όποια στιγμή έχουμε κάνει κάποιο σημαντικό βήμα, αποθηκεύουμε την αλλαγή στο GitHub. Αυτό έχει σειρά ευεργετικών αποτελεσμάτων:

- Έχουμε πάντα ένα αξιόπιστο εφεδρικό μέσο στο οποίο μπορούμε να ανατρέξουμε αν κάτι πάει στραβά στον υπολογιστή μας (μας γλιτώνει από πανικούς του τύπου: ένα βράδυ πριν από την παράδοση ο υπολογιστής μας ή ο δίσκος του πνέει τα λοίσθια, και εμείς τι θα υποβάλουμε στον Λουρίδα;).
- Καθώς έχουμε πλήρες ιστορικό των σημαντικών αλλαγών και εκδόσεων του προγράμματός μας, μπορούμε να επιστρέψουμε σε μία προηγούμενη αν συνειδητοποιήσουμε κάποια στιγμή ότι πήραμε λάθος δρόμο (μας γλιτώνει από πανικούς του τύπου: μα το πρωί δούλευε σωστά, τι στο καλό έκανα και τώρα δεν παίζει τίποτε;).
- Καθώς φαίνεται η πρόοδος μας στο GitHub, επιβεβαιώνουμε ότι το πρόγραμμα δεν είναι αποτέλεσμα της όποιας επιφοίτησης (μας γλιτώνει από πανικούς του τύπου: καλά, πώς το έλυσες το πρόβλημα αυτό με τη μία, μόνος σου;).
- Αν δουλεύετε σε μία ομάδα *που βεβαίως δεν είναι καθόλου η περίπτωση μας εδώ*, μπορούν όλοι να εργάζονται στα ίδια αρχεία στο GitHub εξασφαλίζοντας ότι ο ένας δεν γράφει πάνω στις αλλαγές του άλλου. Παρά το ότι η εργασία αυτή είναι ατομική, καλό είναι να αποκτάτε τριβή με το εργαλείο git και την υπηρεσία GitHub μιας και χρησιμοποιούνται ευρέως και όχι μόνο στη συγγραφή κώδικα.

Άρα επενδύστε λίγο χρόνο στην εκμάθηση των git / GitHub, των οποίων ο σωστός τρόπος χρήσης είναι μέσω γραμμής εντολών (command line), ή ειδικών προγραμμάτων (clients) ή μέσω ενοποίησης στο περιβάλλον ανάπτυξης (editor, IDE).

Τελικό Πρόγραμμα

Το πρόγραμμα θα καλείται ως εξής (όπου `python` η κατάλληλη εντολή στο εκάστοτε σύστημα):

```
python influence_maximization.py [-r RANDOM_SEED] graph k {greedy,max_degree}
  probability mc
```

Η σημασία των παραμέτρων του προγράμματος είναι:

- `-r RANDOM_SEED` αν δίνεται, η τιμή του σπόρου (seed) της γεννήτριας ψευδοτυχαίων αριθμών της Python. Με την τιμή αυτή θα καλείτε τη συνάρτηση `random.seed()` πριν ξεκινήσετε την επιλογή των κόμβων.
- `graph` Το αρχείο με τους συνδέσμους του γράφου.
- `k` Ο αριθμός των κόμβων που θέλουμε να επιλέξουμε ως σπόρους.
- `{greedy,max_degree}` αν ο χρήστης δίνει `greedy`, το πρόγραμμα θα εκτελεί τον λαίμαργο αλγόριθμο· αν ο χρήστης δίνει `max_degree`, το πρόγραμμα θα επιλέγει τους σπόρους με βάση το βαθμό εξόδου.
- `probability` η πιθανότητα με την οποία επηρεάζεται ένας κόμβος από έναν γείτονά του.
- `mc` ο αριθμός επαναλήψεων στη μέθοδο Monte Carlo.

Σκοπός της εργασίας είναι η συγγραφή προγράμματος που θα εντοπίζει τους κόμβους τους οποίους θα πρέπει να επιλέξουμε για να μεγιστοποιήσουμε την επιρροή με τον αλγόριθμο που περιγράψαμε. *Υλοποιήσεις άλλων αλγορίθμων δεν είναι αποδεκτές.*

Παραδείγματα

Παράδειγμα 1

Αν ο χρήστης του προγράμματος δώσει:

```
python influence_maximization.py erdos_renyi.txt 10 max_degree 0.1 1000 -r 42
```

το πρόγραμμά σας θα διαβάσει το αρχείο `erdos_renyi.txt`, θα εντοπίσει δέκα κόμβους ως σπόρους, θα χρησιμοποιήσει ως πιθανότητα επηρεασμού $p = 0.1$, θα εκτελέσει 1000 επαναλήψεις σε κάθε χρήση της μεθόδου Monte Carlo, και θα χρησιμοποιήσει ως σπόρο της γεννήτριας ψευδοτυχαίων αριθμό τον αριθμό 42. Η έξοδος του προγράμματος θα είναι:

```
Seeds [40, 0, 2, 11, 20, 28, 36, 42, 45, 22]
Influences [2.578, 5.05, 6.554, 8.256, 9.694, 11.155, 12.697, 13.831, 14.591, 15.761]
```

Η πρώτη γραμμή δείχνει τους κόμβους που επιλέγονται, με τη σειρά που επιλέγονται. Η δεύτερη γραμμή δίνει τη συνολική επιρροή των επιλεγμένων κάθε φορά κόμβων.

Παράδειγμα 2

Αν ο χρήστης του προγράμματος δώσει:

```
python influence_maximization.py erdos_renyi.txt 10 greedy 0.1 1000 -r 42
```

το πρόγραμμά σας θα διαβάσει το αρχείο erdos_renyi.txt, και στην έξοδο θα εμφανίσει:

```
Seeds [40, 0, 29, 6, 36, 35, 39, 32, 14, 38]
Influences [2.613, 4.977, 6.993, 8.722, 10.33, 11.976, 13.649, 15.051, 16.512, 17.877]
```

Παράδειγμα 3

Αν ο χρήστης του προγράμματος δώσει:

```
python influence_maximization.py barabasi_albert.txt 10 max_degree 0.1 1000 -r 42
```

το πρόγραμμά σας θα διαβάσει το αρχείο barabasi_albert.txt, και στην έξοδο θα εμφανίσει:

```
Seeds [0, 1, 8, 4, 5, 6, 7, 10, 12, 13]
Influences [4.801, 7.87, 10.297, 11.726, 12.922, 14.019, 14.852, 15.939, 17.121, 18.172]
```

Παράδειγμα 4

Αν ο χρήστης του προγράμματος δώσει:

```
python influence_maximization.py barabasi_albert.txt 10 greedy 0.1 1000 -r 42
```

το πρόγραμμά σας θα διαβάσει το αρχείο barabasi_albert.txt, και στην έξοδο θα εμφανίσει:

```
Seeds [1, 8, 0, 10, 20, 25, 38, 4, 31, 27]
Influences [4.804, 7.653, 10.199, 12.068, 13.452, 14.721, 16.171, 17.38, 18.458, 19.666]
```

Παράδειγμα 5

Αν ο χρήστης του προγράμματος δώσει:

```
python influence_maximization.py watts_strogatz.txt 10 max_degree 0.1 1000 -r 42
```

το πρόγραμμά σας θα διαβάσει το αρχείο watts_strogatz.txt και στην έξοδο θα εμφανίσει:

```
Seeds [3, 5, 6, 10, 14, 17, 20, 34, 36, 40]
Influences [1.675, 3.059, 4.307, 5.932, 7.607, 8.759, 10.339, 11.961, 13.017, 14.605]
```

Παράδειγμα 6

Αν ο χρήστης του προγράμματος δώσει:

```
python influence_maximization.py watts_strogatz.txt 10 greedy 0.1 1000 -r 42
```

το πρόγραμμά σας θα διαβάσει πάλι το αρχείο `watts_strogatz.txt` και στην έξοδο θα εμφανίσει:

```
Seeds [10, 40, 3, 34, 20, 44, 23, 14, 29, 48]
Influences [1.736, 3.407, 5.133, 6.804, 8.519, 9.95, 11.359, 13.011, 14.312, 15.637]
```

Περισσότερες Πληροφορίες

Η προσέγγιση για την μεγιστοποίηση της επιρροής που αποτελεί το αντικείμενο της εργασίας παρουσιάστηκε από τους Kempe, Kleinberg και Tardos το 2003 [2]. Ο αλγόριθμος μπορεί να βελτιωθεί ώστε να εκτελείται πιο γρήγορα, αποφεύγοντας τον υπολογισμό περιττών προσομοιώσεων. Αυτό εκμεταλλεύεται ο αλγόριθμος CELF [3], ο οποίος βελτιστοποιήθηκε επιπλέον με τον αλγόριθμο CELF++ [1]. Το πρόβλημα έχει προσελκύσει το ενδιαφέρον πολλών ερευνητών, βλ. και σχετικές επισκοπήσεις [4, 5].

Βιβλιογραφία

[1] Amit Goyal, Wei Lu, and Laks V.S. Lakshmanan. "CELF++: optimizing the greedy algorithm for influence maximization in social networks". In: *Proceedings of the 20th International Conference Companion on World Wide Web*. WWW '11. Hyderabad, India: Association for Computing Machinery, 2011, pp. 47–48. ISBN: 9781450306379. DOI: 10.1145/1963192.1963217.

[2] David Kempe, Jon Kleinberg, and Éva Tardos. "Maximizing the spread of influence through a social network". In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '03. Washington, D.C.: Association for Computing Machinery, 2003, pp. 137–146. ISBN: 1581137370. DOI: 10.1145/956750.956769.

[3] Jure Leskovec et al. "Cost-effective outbreak detection in networks". In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '07. San Jose, California, USA: Association for Computing Machinery, 2007, pp. 420–429. ISBN: 9781595936097. DOI: 10.1145/1281192.1281239.

[4] Yandi Li et al. "A Survey on Influence Maximization: From an ML-Based Combinatorial Optimization". In: *ACM Transactions on Knowledge Discovery from Data* 17.9 (July 2023). ISSN: 1556-4681. DOI: 10.1145/3604559.

[5] Yuchen Li et al. "Influence Maximization on Social Graphs: A Survey". In: *IEEE Transactions on Knowledge and Data Engineering* 30.10 (2018), pp. 1852–1872. DOI: 10.1109/TKDE.2018.2807843.

Καλή Επιτυχία!