Χαροκόπειο Πανεπιστήμιο
Τμήμα Πληροφορικής και Τηλεματικής

Τεχνολογίες Γραφημάτων
Εργασία

Διδάσκων: Δημήτρης Μιχαήλ

2023-2024

Σκοπός της εργασίας αυτής είναι η εξοικείωση σας με αλγορίθμους υπολογισμού node embeddings σε γραφήματα. Ως παράδειγμα θα χρησιμοποιήσουμε ένα πολύ απλό αλγόριθμο που περιγράφεται από τους Ahmed et al. [1].

Ο αλγόριθμος υπολογίζει για κάθε κόμβο ένα διάνυσμα Z_i μέσω των οποίων μπορεί κανείς να αποφασίσει την ύπαρξη μίας ακμής (i, j). Στην πιο απλή μορφή του μπορεί κανείς να χρησιμοποιήσει ένα απλό μοντέλο εσωτερικού γινομένου όπου η πληροφορία της ύπαρξης μίας ακμής (i, j) μπορεί να περιγραφεί αποτελεσματικά με το εσωτερικό γινόμενο $\langle Z_i, Z_j \rangle$.

Στην συγκεκριμένη εργασία θα χρησιμοποιήσετε Python και πιο συγκεκριμένα την βιβλιοθήκη NetworkX. Μπορείτε επίσης να χρησιμοποιήσετε την βιβλιοθήκη numpy.

1 Υπολογισμός πίνακα Z

Στο πρώτο μέρος της εργασίας καλείστε να αναπτύξετε κώδικα που να υπολογίζει τα διανύσματα Z_i για κάθε κόμβο ενός μη-κατευθυνόμενου γραφήματος. Διαβάστε προσεκτικά την Ενότητα 2 του [1] και υλοποιήστε την σειριακή έκδοση του αλγορίθμου οπως φαίνεται παρακάτω:

Algorithm 1 Sequential stochastic gradient descent

Require: Matrix $Y \in \mathbb{R}^{n \times n}$, rank r, accuracy ϵ
Ensure: Find a local minimum of (1)
1: Initialize $Z' \in \mathbb{R}^{n \times r} \in$ at random
2: $t \leftarrow 1$
3: **repeat**
4: $\quad Z' \leftarrow Z$
5: $\quad$ **for all** edges $(i, j) \in E$ **do**
6: $\quad\quad \eta \leftarrow \frac{1}{\sqrt{t}}$
7: $\quad\quad t \leftarrow t + 1$
8: $\quad\quad Z_i \leftarrow Z_i + \eta[\big(Y_{ij} - \langle Z_i, Z_j \rangle\big) Z_j + \lambda Z_i]$
9: $\quad$ **end for**
10: **until** $\|Z - Z'\|^2_{\text{Frob}} \leq \epsilon$
11: **return** Z

Υλοποιήστε τον αλγόριθμο αυτό με την βοήθεια της NetworkX και δοκιμάστε τον σε πραγματικά γραφήματα που θα βρείτε σε κάποια από τις παρακάτω ιστοσελίδες στο διαδίκτυο.

- `https://snap.stanford.edu/data/index.html`
- `http://konect.cc/networks`

- `https://networks.skewed.de`

Δείξτε την δουλειά σας με παραδείγματα. Εκτός από πραγματικά δίκτυα, φτιάξτε και μικρά παραδείγματα γραφημάτων για να ελέγξετε τον αλγόριθμο σας αλλά και για να δείξετε στην αναφορά σας.

2 Εφαρμογές

Βρείτε παραδείγματα εφαρμογών που μπορείτε να υλοποιήσετε με την προσέγγιση αυτή. Σε ποια προβλήματα είναι χρήσιμη η τεχνική αυτή; Μπορείτε να πάρετε ιδέες από την Ενότητα 8 του [1]. Διαλέξτε μία από τις πιθανές εφαρμογές και κάντε υλοποιήση. Παρουσιάστε την δουλειά σας με λεπτομέρεια στην αναφορά. Τρέξτε την υλοποίηση σας και δείξτε τα αποτελέσματα.

3 Παραδοτέα

Η άσκηση έχει ένα παραδοτέο που αποτελείται από 3 μέρη:

1. Ο πηγαίος κώδικας ο οποίος θα πρέπει να είναι γραμμένος σε python και να εκτελείται πολύ εύκολα.
2. Μαζί με τον πηγαίο κώδικα θα πρέπει να υπάρχει και ένα αρχείο README.md το οποίο να περιγράφει αναλυτικά πως τρέχει.
3. Τέλος θα πρέπει να υπάρχει και ένα αρχείο report.pdf το οποίο να περιγράφει αναλυτικά την δουλειά σας, να εξηγεί τον κώδικα σας, και να περιέχει παραδείγματα εκτέλεσης του κώδικα σας.

Προσοχή η βαθμολόγηση δεν γίνεται μόνο με βάση την λειτουργικότητα αλλά και με βάση την ποιότητα του κώδικα. Επιπρόσθετα σημαντικό ρόλο παίζει η αναλυτική αναφορά.

References

[1] Amr Ahmed, Nino Shervashidze, Shravan Narayanamurthy, Vanja Josifovski, and Alexander J Smola. Distributed large-scale natural graph factorization. In *Proceedings of the 22nd international conference on World Wide Web*, pages 37–48, 2013.