Graph Embeddings και Εφαρμογές τους στην Εξόρυξη Πληροφορίας

Καλογερόπουλος Νικήτας - Ρήγας

Τμήμα Μηχανικών Η/Υ & Πληροφορικής
Πανεπιστήμιο Πατρών

6 Απριλίου 2023

Γραφήματα (1).

Ορισμός

Γράφημα είναι ένα μαθηματικό ή συνδυαστικό αντικείμενο που μπορεί εύκολα να αναπαρασταθεί με εικόνες, ή ισοδύναμα, επιδέχεται απλή και εύκολη εικονογραφημένη αναπαράσταση.

1. $G = (V, E)$ ένα ζεύγος συνόλων V, E
2. V: σύνολο κορυφών, μη κενό.
3. E: σύνολο ακμών, που αποτελείται από ζεύγη κορυφών.
4. Βεβαρημένα - αβαρή γραφήματα.
5. Κατευθυνόμενα - μη κατευθυνόμενα γραφήματα.

Γραφήματα (2).

Μητρώο γειτνίασης, A, χρησιμοποιείται για να αναπαραστήσει τις κορυφές ενός γραφήματος και να εκφράσει τη μεταξύ τους σύνδεση.
$A = (a_{ij})_{(nxn)}$ ορίζεται:

$$a_{ij} = \begin{cases} w & \text{Αν } v_i v_j \in E \\ 0 & \text{αλλού} \end{cases} \quad (1)$$

Η τιμή του w σε μη βεβαρημένη περίπτωση θα είναι 1, ενώ σε βεβαρημένα γραφήματα θα ισούται με το βάρος της εκάστοτε ακμής.

Γραφήματα (3)

- *Degree ή βαθμός ενός κόμβου: Ο βαθμός ενός κόμβου είναι ο αριθμός των ακμών που συνδέονται με αυτόν.*
- *Συντομότερο Μονοπάτι ή Shortest Path: το μονοπάτι μεταξύ δύο ακμών στο οποίο το άθροισμα του βάρους των ακμών ελαχιστοποιείται.*
- Random Walk ή Τυχαίος Περίπατος: Διατρέχονται κατά μήκος οι ακμές του γραφήματος και η επιλογή για μετάβαση από κάποιο κόμβο σε άλλο είναι ισοπίθανα τυχαία.
- 2^{nd} Order Random Walk ή Τυχαίος Περίπατος $2^{ης}$ Τάξης: Διατρέχονται κατά μήκος οι ακμές του γραφήματος για 2 ακριβώς ανεξάρτητες μεταβάσεις και η μετάβαση εξαρτάται από γνώση της προηγούμενης κατάστασης.

Εισαγωγή - Graph Embeddings

Embeddings

Πραγματικές τιμές σε ένα διάνυσμα προκαθορισμένων διαστάσεων από μία συλλογή γραφημάτων που βασίζονται στις ιδιότητες τους.

Τέτοιες ιδιότητες μπορεί να βασίζονται:

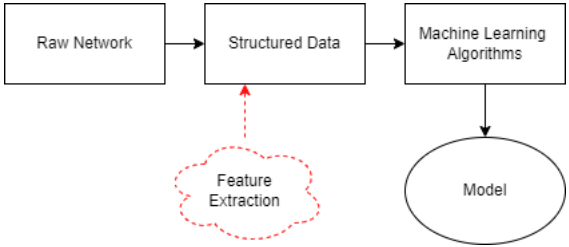
- στην τοπολογία.
- στη σχέση μεταξύ ακμών.
- στην ύπαρξη σε συγκεκριμένα υπογραφήματα.

Αλγόριθμοι:

1. DeepWalk [5]
2. Node2Vec [1]

Μηχανική Μάθηση σε Γραφήματα

- Κατηγοριοποίηση κόμβων ή Γραφημάτων.
- Προβλέψεις Σύνδεσης.
- Εντοπισμός Κοινοτήτων.



Πώς όμως θα γίνει η επιλογή χαρακτηριστικών;

Ιδιαιτερότητες Γραφημάτων

1. Τα πραγματικά γραφήματα είναι πολύπλοκα και τοπολογικά.
2. Δεν έχουν χωρική τοπικότητα.
3. Δεν υπάρχει κάποια ταξινόμηση στους κόμβους.
4. Είναι δυναμικά και με πληροφορία συνήθως στις ακμές ή ακόμα και στους κόμβους.

Παραγωγή Χαρακτηριστικών από Γραφήματα

Στόχος

Ανεξάρτητα του μοντέλου που θα εφαρμοστεί να υπάρχει ένας τρόπος για την παραγωγή χαρακτηριστικών αυτόματα και με επιτυχία διατηρώντας τις αρχικές ιδιότητες.

Χαρακτηριστικά του νέου χώρου:

- Να υπάρχει κάποια μετρική ομοιότητας - απόστασης.
- Η ομοιότητα του νέου χώρου να ακολουθεί του παλιού.
- Να μπορεί να κωδικοποιηθεί πληροφορία δικτύου και κόμβων στον νέο χώρο.

Παραγωγή Αναπαραστάσεων

Έστω γράφημα $G = \{V, E\}$ με V, το σύνολο των κόμβων, E, το σύνολο των ακμών και A το μητρώο γειτνίασης.

Πρέπει να οριστούν:

- Κωδικοποιητής - Encoder $En(u) = Z_u$, όπου μεταφέρει τον κόμβο στον νέο χώρο.
- Η ομοιότητα των κόμβων.
- Παραμετροποίηση της ομοιότητας ώστε: $sim(u, v) \approx Z_u^T \cdot Z_v$

Παραγωγή Αναπαραστάσεων

Έστω γράφημα $G = \{V, E\}$ με V, το σύνολο των κόμβων, E, το σύνολο των ακμών και A το μητρώο γειτνίασης.

Πώς θα ορίσουμε την ομοιότητα κόμβων;

1. Αν 2 κόμβοι συνδέονται.
2. Αν 2 κόμβοι έχουν κοινούς γείτονες.
3. Αν 2 κόμβοι έχουν κοινούς δομικούς ρόλους.
4. Την ύπαρξη σε τυχαίους περιπάτους στο δίκτυο, που αποτελούν τη γειτονία N_R του κόμβου. ⟵

Βελτιστοποίηση της Αναπαράστασης

- Στόχος είναι με βάση κάποιον κόμβο u να προβλέπω του γείτονές του.
- Η αναπαράσταση Z_u είναι αυτή που ελαχιστοποιεί την ποσότητα ℓ.
- Πρόβλημα βελτιστοποίησης $\rightarrow$ Stochastic gradient descent.

$$\ell = \sum_{u \in V} \sum_{v \in N_R(u)} -\log\left(\frac{\exp\left(Z_u^T Z_v\right)}{\sum_{n \in V} \exp\left(Z_u^T Z_v\right)}\right)$$

- Negative Sampling[3].

DeepWalk αλγόριθμος

- Δειγματοπληπτεί με τυχαίους περιπάτους το γράφημα.
- Τα δείγματα ισοδυναμούν με προτάσεις.
- Πρόβλημα βελτιστοποίησης πρόβλεψης γειτόνων $\longrightarrow$ Word2vec skipgramm[3].
- Node embeddings: οι τιμές της εξόδου του κρυφού επιπέδου.

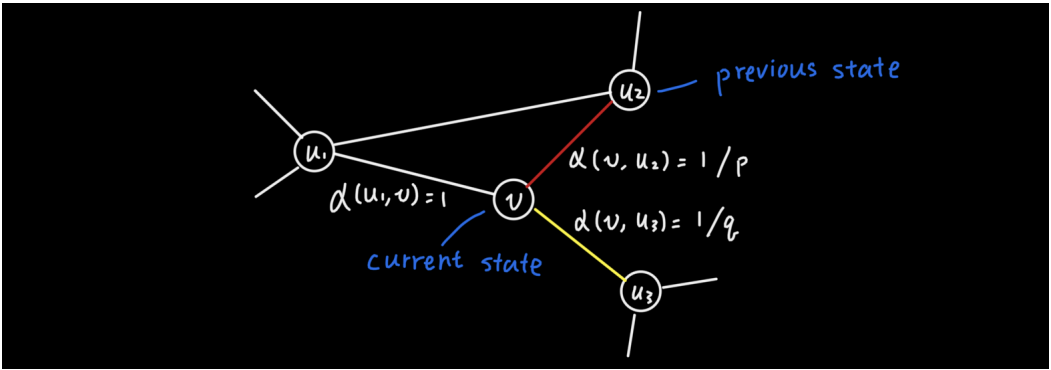
Phases of DeepWalk approach

Node2Vec Αλγόριθμος

- Επέκταση του DeepWalk.
- 2ης τάξης περίπατοι, με γνώση παρούσας και προηγουμένης κατάστασης.
- Παράμετροι: Επιστροφής (p), εξόδου (q), καθορίζουν την μετάβαση σε κάποιον κόμβο $\longrightarrow$ προκατειλημμένοι περίπατοι.

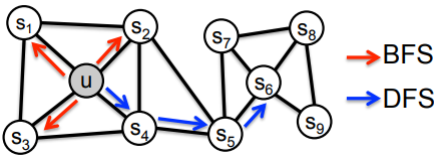
Η συνάρτηση α καθορίζει την μετάβαση ή όχι στον επόμενο κόμβο, με είσοδο την προηγούμενη κατάσταση και την πιθανή επόμενη. $\alpha_{pq}(t,x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases}$

Node2Vec - Παράδειγμα



Χαρακτηριστικά του Node2Vec

- Αναζήτηση κατά πλάτος → Τοπική οπτική (Local micro - view).
- Αναζήτηση κατά βάθος → Συνολική οπτική (Global macro - view).



Πώς επηρεάζουν οι τιμές p, q την οπτική;

Συνδυασμός Embeddings Κόμβων

Operator	Symbol	Definition
Average	$\boxplus$	$[f(u) \boxplus f(v)]_i = \frac{f_i(u)+f_i(v)}{2}$
Hadamard	$\boxdot$	$[f(u) \boxdot f(v)]_i = f_i(u) * f_i(v)$
Weighted-L1	$\|\cdot\|_{\bar{1}}$	$\|f(u) \cdot f(v)\|_{\bar{1}i} = \|f_i(u) - f_i(v)\|$
Weighted-L2	$\|\cdot\|_{\bar{2}}$	$\|f(u) \cdot f(v)\|_{\bar{2}i} = \|f_i(u) - f_i(v)\|^2$

Από Αναπαραστάσεις Κόμβων σε Αναπαραστάσεις Γραφημάτων

- Παραγωγή και ένωση (πρόσθεση, μέση τιμή κ.α.) διανυσμάτων κόμβων.
- Δημιουργία ενός υπερ - κόμβου και παραγωγή αναπαράστασης για αυτόν, με διαφορους αλγορίθμους.
- Με ανώνυμους περιπάτους.
- Με υπογραφήματα - Graph Kernels [4].

Παράδειγμα παραγωγής χαρακτηριστικών και ομαδοποίησης με το σύνολο δεδομένων του Τιτανικού

Νέα χαρακτηριστικά

```
corr_df = train_df.drop(['PassengerId'], axis = 1)
corr_df['Family'] = corr_df.Parch + corr_df.SibSp
corr_df['Is_Alone'] = corr_df.Family == 0
corr_df['Fare_Category'] = pd.cut(corr_df['Fare'], bins=[0,7.90,14.45,31.28,120], labels=['Low','Mid',
                                'High_Mid','High'])
corr_df['Age_group'] = pd.cut(corr_df['Age'], bins=[0,10,20,40,60,80,100],
                              labels=['0-10','10-20', '20-40', '40-60','60-80','80-100'])
corr_df['Title'] = train_df.Name.apply(lambda name: name.split(',')[1].split('.')[0].strip())



corr_df.drop([ 'Fare'], axis=1, inplace=True)
corr_df.drop([ 'SibSp', ], axis=1, inplace=True)
corr_df.drop([ 'Parch'], axis=1, inplace=True)
corr_df.drop([ 'Name'], axis=1, inplace=True)
corr_df.drop([ 'Ticket'], axis=1, inplace=True)
```

Σχήμα: Παραγωγή Χαρακτηριστικών

Νέα χαρακτηριστικά

```
corr_df['Is_Alone']=corr_df['Is_Alone'].map({False:0,True:1})
corr_df['Title'] = LabelEncoder().fit_transform(corr_df['Title'])
corr_df['Fare_Category'] = LabelEncoder().fit_transform(corr_df['Fare_Category'])
labels=['0-10','10-20', '20-40', '40-60','60-80','80-100']
mapping ={}
for label in labels:
    temp = corr_df[corr_df['Age_group']==label]
    mea = temp.Age.mean()
    if isnan(mea):
        mea = 0
    mapping[label] = mea
#print(mapping)
corr_df['Age_group'] = corr_df['Age_group'].map(mapping)
```

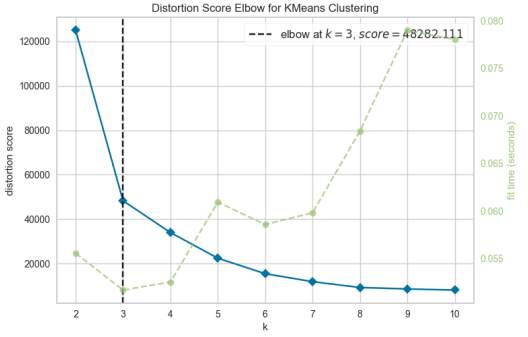
Σχήμα: Κωδικοποίηση χαρακτηριστικών

Clustering

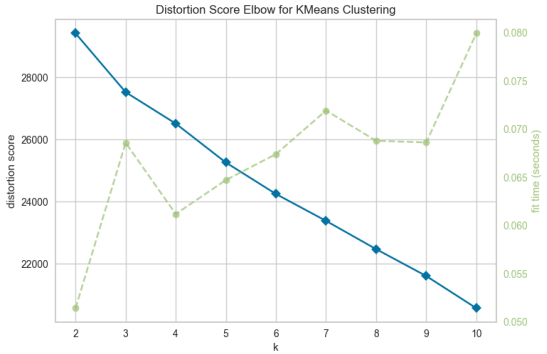
```
kmeans = KMeans(n_init=10)
sil_vis = KElbowVisualizer(kmeans,numeric_only=None)
sil_vis.fit(X)
sil_vis.show()

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
kmeans = KMeans(n_init=10)
sil_vis = KElbowVisualizer(kmeans,numeric_only=None)
sil_vis.fit(X_scaled)
sil_vis.show()
```

Σχήμα: Ομαδοποίηση με και χωρίς Scaling

Επιλογή Clustering

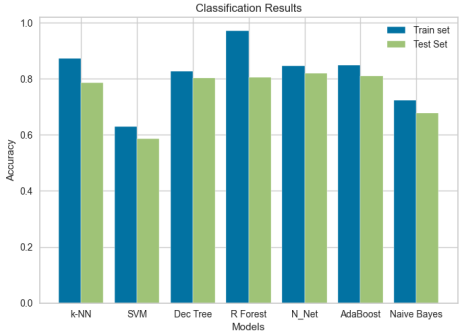


Σχήμα: Χωρίς Επεξεργασία

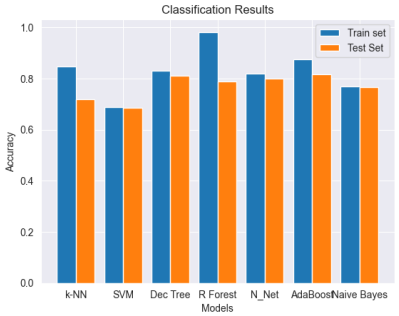


Σχήμα: Με Scaling

Αποτελέσματα χωρίς scaling

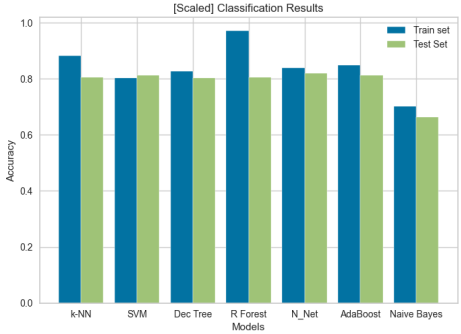


Σχήμα: Νέα Χαρακτηριστικά

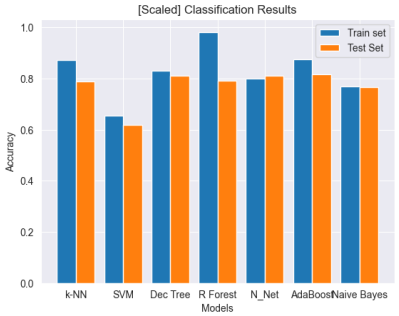


Σχήμα: Παλιά Χαρακτηριστικά

Αποτελέσματα με scaling



Σχήμα: Νέα Χαρακτηριστικά

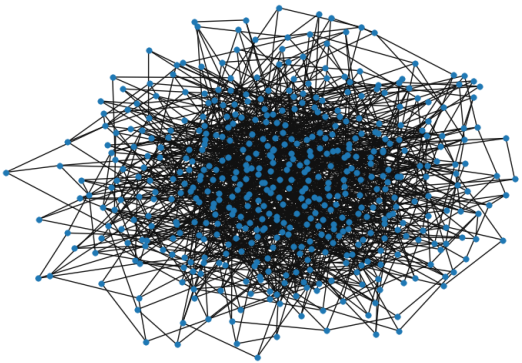


Σχήμα: Παλιά Χαρακτηριστικά

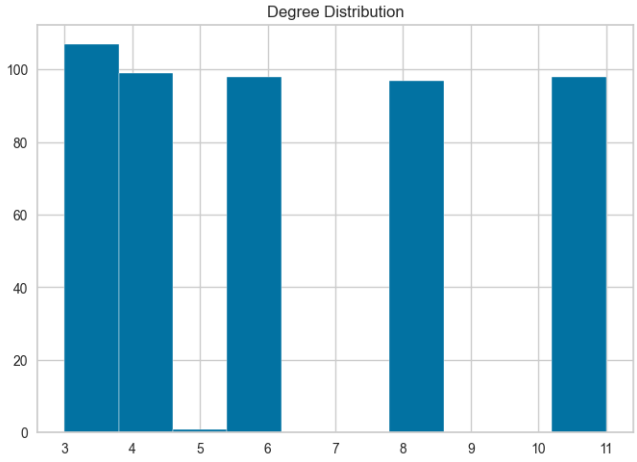
Παράδειγμα ομαδοποίησης με τυχαία κατασκευασμένο γράφημα, παραγωγή αναπαραστάσεων και οπτικοποίηση

Κατασκευή του Γραφήματος[2]

```
G = generate_graph_deg_dist(
        deg_dist = {
            6:0.2,
            3:0.14,
            8:0.35,
            4:0.3,
            11:0.01
        },
        n = 500
)

print(nx.info(G))
```



Κατανομή του Γραφήματος



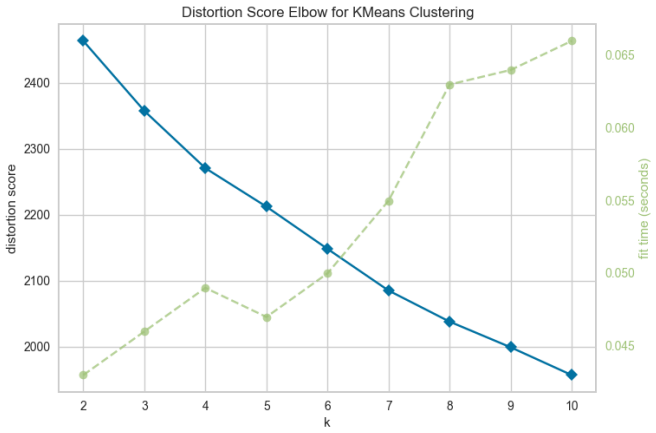
Εφαρμογή Node2Vec

```
%%time

from node2vec import Node2Vec
g_emb = Node2Vec(G,dimensions=16,walk_length=100,num_walks=20,q=1,p=2)

mdl = g_emb.fit(window=5, min_count=3)

emb_df = (
    pd.DataFrame(
        [mdl.wv.get_vector(str(n)) for n in G.nodes()],
        index = G.nodes
    )
)
```

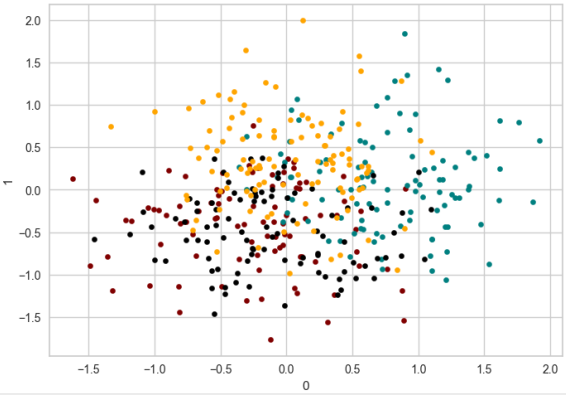
Executed in 27s, 4 Apr at 07:08:29

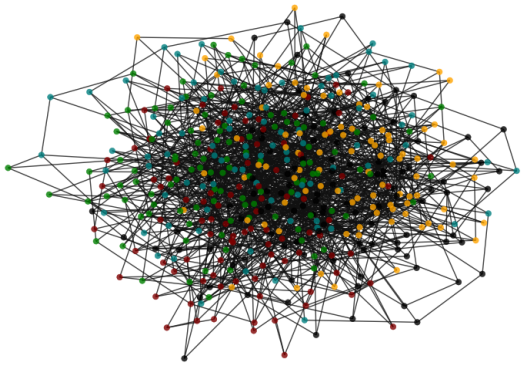
```
Computing transition probabilities: 100% ████████████████████ 500/500 [00:00<00:00, 1620.69it/s]

Generating walks (CPU: 1): 100%|██████████| 20/20 [00:07<00:00,  2.76it/s]

CPU times: total: 26.8 s
Wall time: 27.3 s
```

Clustering

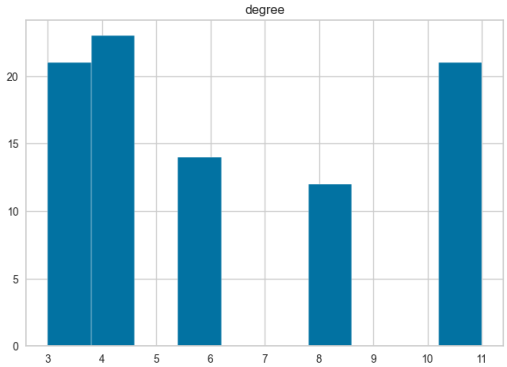

Distortion Score Elbow for KMeans Clustering
distortion score
fit time (seconds)
k
2400
2300
2200
2100
2000
0.065
0.060
0.055
0.050
0.045
2
3
4
5
6
7
8
9
10

Clustering

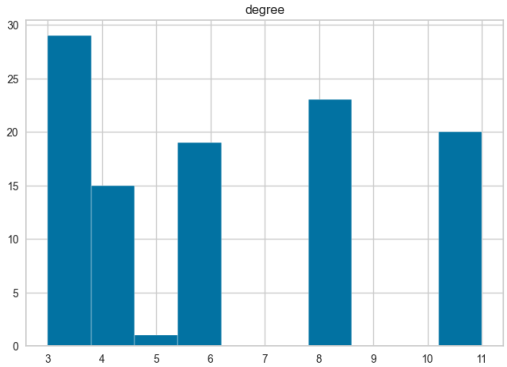




Κατανομή Ενδεικτικών Ομάδων



Σχήμα: Cluster 1



Σχήμα: Cluster 2

References I

[1] Aditya Grover and Jure Leskovec. "Node2vec: Scalable Feature Learning for Networks". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 855–864. ISBN: 9781450342322. DOI: 10.1145/2939672.2939754. URL: https://doi.org/10.1145/2939672.2939754.

[2] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. "Exploring Network Structure, Dynamics, and Function using NetworkX". In: *Proceedings of the 7th Python in Science Conference*. Ed. by Gaël Varoquaux, Travis Vaught, and Jarrod Millman. Pasadena, CA USA, 2008, pp. 11–15.

References II

[3] Tomás Mikolov et al. "Efficient Estimation of Word Representations in Vector Space". In: *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2013.

[4] Annamalai Narayanan et al. "graph2vec: Learning distributed representations of graphs". In: *arXiv preprint arXiv:1707.05005* (2017).

[5] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014, pp. 701–710.