Μέθοδοι εξαγωγής ενσωματώσεων (embeddings) για κόμβους και γραφήματα

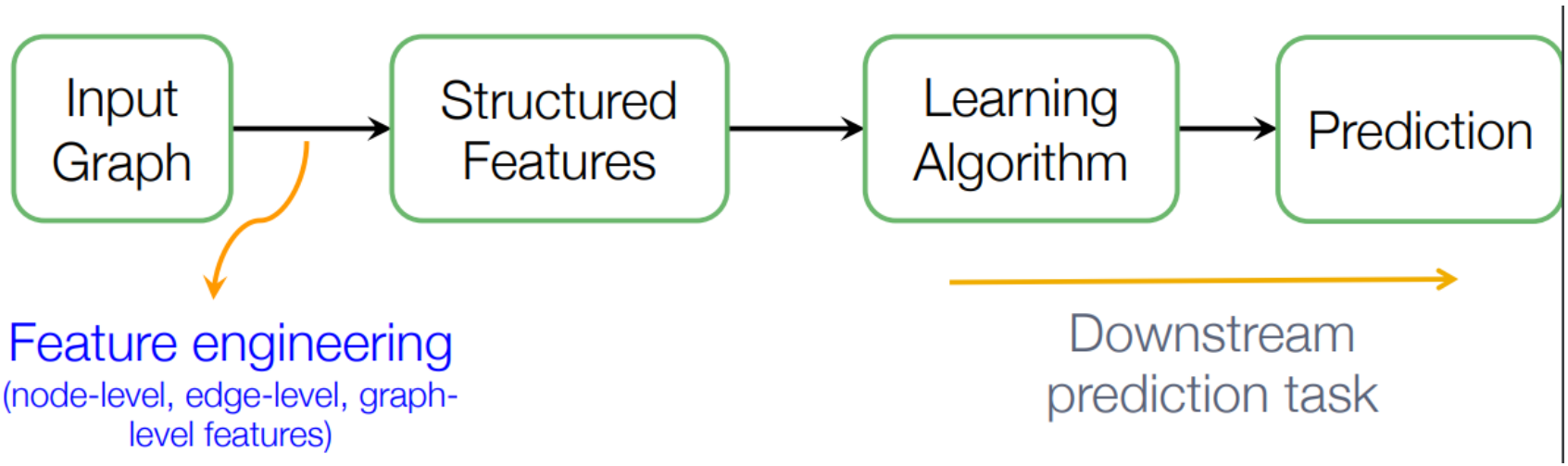
Δημήτριος Κοσμόπουλος

Πανεπιστήμιο Πατρών
Τμήμα Μηχανικών Η/Υ κ Πληροφορικής

7 Δεκεμβρίου 2023

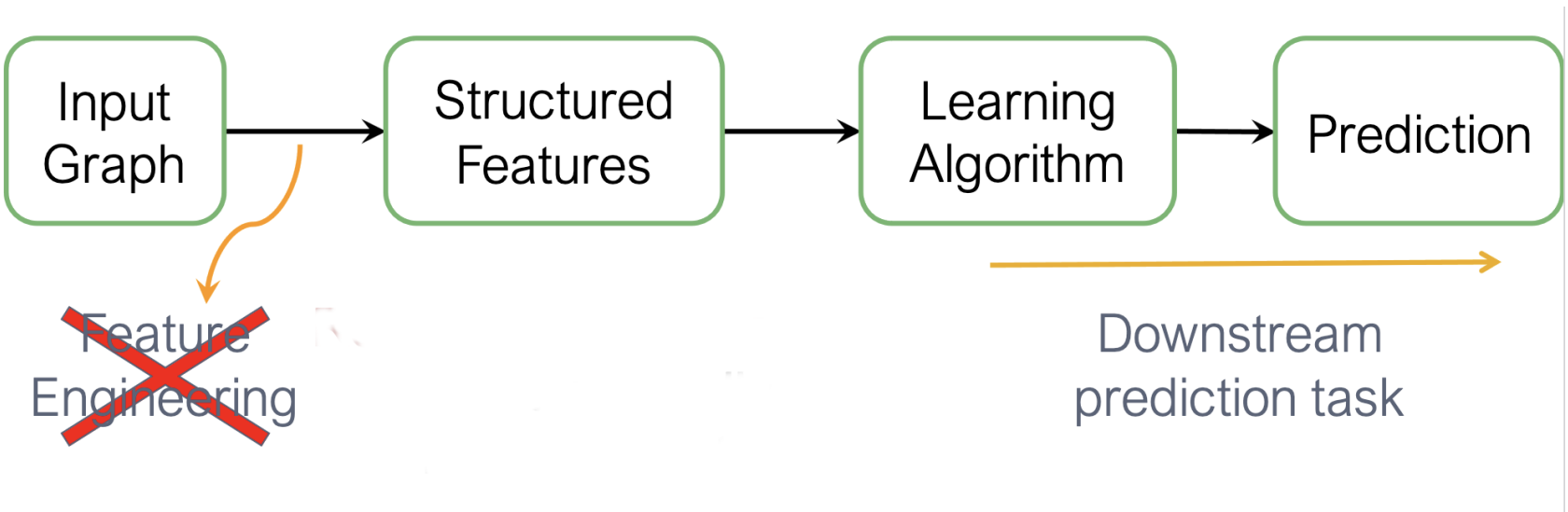
Παραδοσιακές μέθοδοι μηχανικής μάθησης σε γραφήματα

Δεδομένου ενός γραφήματος, εξάγετε χαρακτηριστικά κόμβων, συνδέσμων και επιπέδου γράφου, στη συνέχεια εκπαιδεύετε ένα μοντέλο (SVM, νευρωνικό δίκτυο, κλπ.) που αντιστοιχεί τα χαρακτηριστικά σε ετικέτες.



Σύγχρονες μέθοδοι μηχανικής μάθησης σε γραφήματα

Δεν ορίζουμε τα χαρακτηριστικά χειρωνακτικά, αλλα τα μαθαίνουμε αυτόματα.



Σύνοψη μεθόδων ενσωμάτωσης

Για κόμβους:

- Lookup table
- Random walk
  - Deep walk
  - node2vec

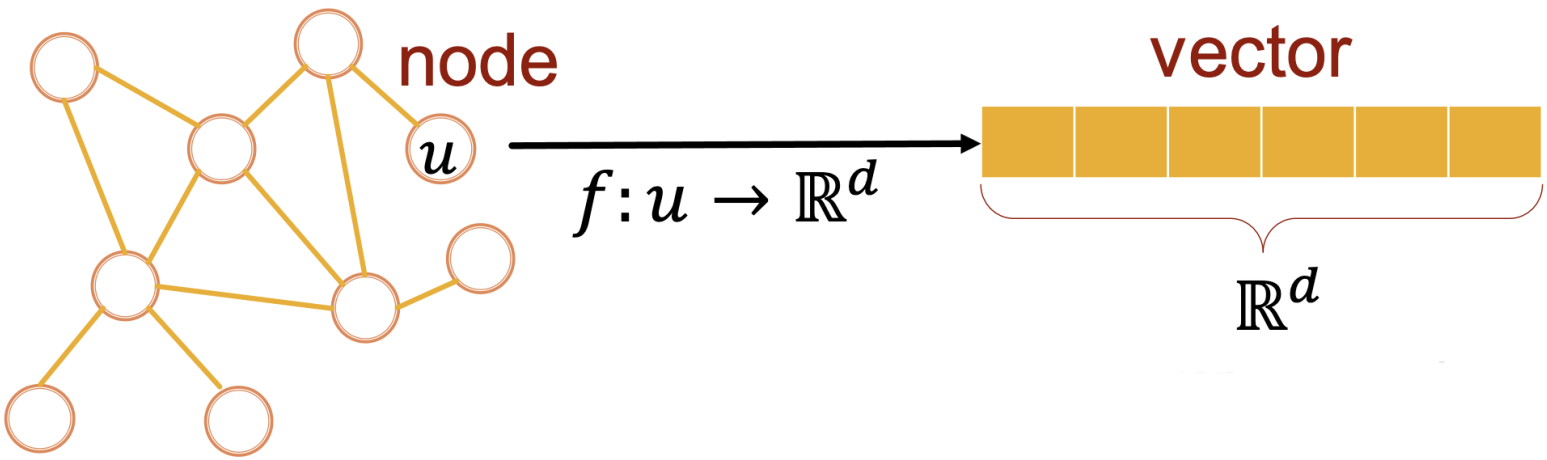
Για γραφήματα:

- Μέση τιμή
- Εικονικός κόμβος

Στόχος ενσωματώσεων (embeddings)

Τρόπος αναπαράστασης που είναι:

- υπολογιστικά αποδοτικός
- εξαρτάται μόνο από την τοπολογία του γραφήματος και όχι από το είδος της εργασίας μάθησης



Γιατί ενσωματώσεις ;

- Τα διανύσματα είναι εύχρηστα και ξέρουμε πώς να τα χρησιμοποιήσουμε για κλασσικά προβλήματα μάθησης
- Κωδικοποιούν πληροφορία του δικτύου
- Ομοιότητα στο χώρο ενσωμάτωσης συνεπάγεται ομοιότητα στο γράφημα (κόμβους)

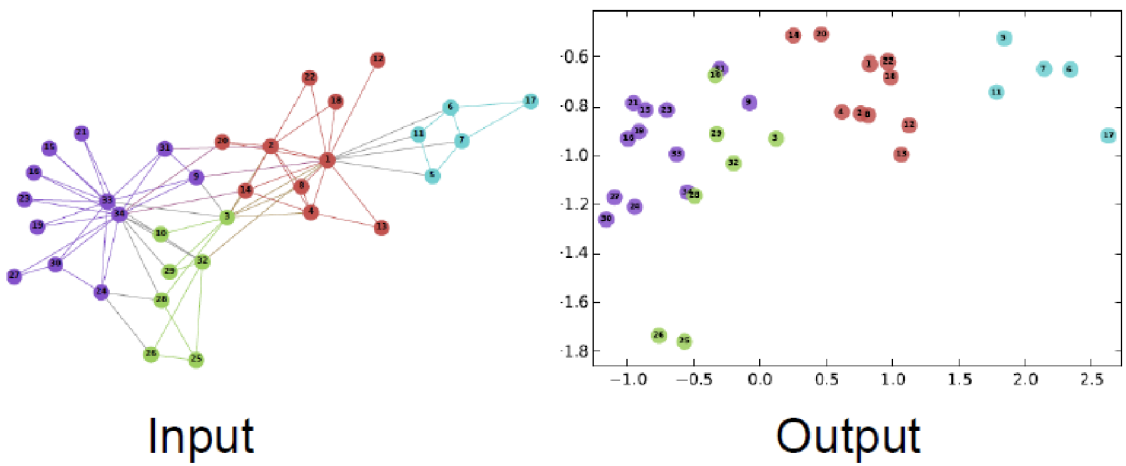
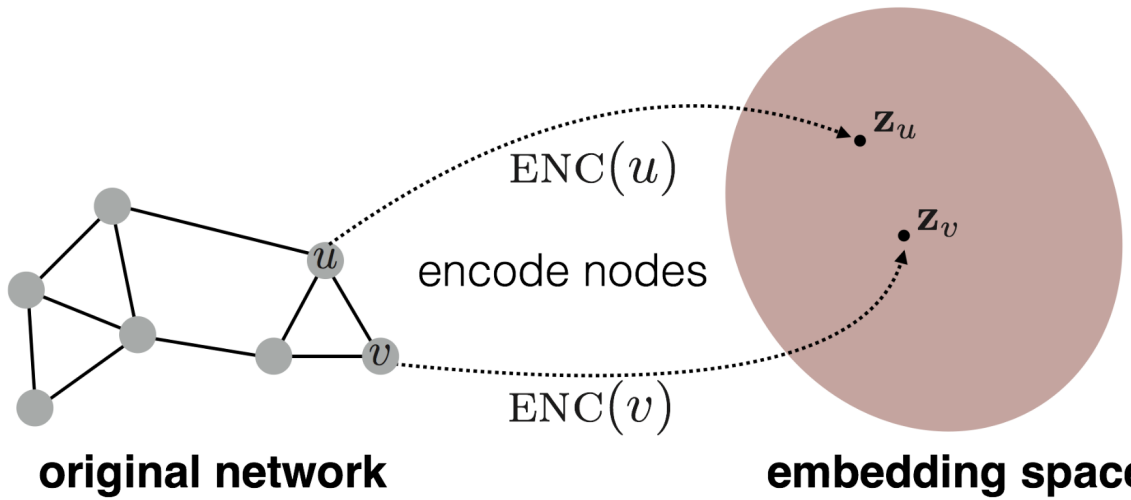


Image from: Perozzi et al. DeepWalk: Online Learning of Social Representations. *KDD 2014.*

Ενσωμάτωση κόμβων

- Ένα γράφημα $\mathcal{G} = (V, \mathcal{E})$ καθορίζεται από ένα σύνολο κόμβων $\mathcal{V}$ και ένα σύνολο ακμών $\mathcal{E}$ μεταξύ αυτών των κόμβων.
- Για απλότητα θα χρησιμοποιήσουμε μόνο την πληροφορία από τον πίνακα γειτνίασης A.
- Στόχοι:
  - ορισμός ομοιότητας στο χώρο των κόμβων (γραφήματος)
  - εκμάθηση κωδικοποίησης ώστε η ομοιότητα στο χώρο ενσωμάτωσης, π.χ. εσωτερικό γινόμενο, να προσεγγίζει την ομοιότητα στο χώρο των κόμβων (γραφήματος).



Ενσωμάτωση κόμβων

- Ορισμός κωδικοποιητή από το χώρο των κόμβων στο χώρο ενσωμάτωσης
- Ορισμός συνάρτησης ομοιότητας των κόμβων στο χώρο του γραφήματος
- Ορισμός αποκωδικοποιητή ώστε να ικανοποιεί τον περιορισμό της ομοιότητας
- Εκμάθηση παραμέτρων κωδικοποιητή ώστε: $similarity(u, v) \approx z_u{}^T \cdot z_v$

Ορισμός ομοιότητας κόμβων

- Η αντικειμενική συνάρτηση για την μάθηση του κωδικοποιητή μεγιστοποιεί το εσωτερικό γινόμενο κόμβων που είναι **όμοιοι** στο χώρο του γραφήματος.
- τι σημαίνει **όμοιοι**;
  - συνδεδεμένοι;
  - κοινοί γείτονες;
  - κοινό ρόλο στο γράφημα;

Παρατηρήσεις για την ακολουθούμενη προσέγγιση

Παρατηρήσεις:

- Δεν χρησιμοποιούμε ετικέτες για τους κόμβους
- Δεν χρησιμοποιούμε χαρακτηριστικά των κόμβων
- Χρησιμοποιούμε μόνο τη δομή του δικτύου

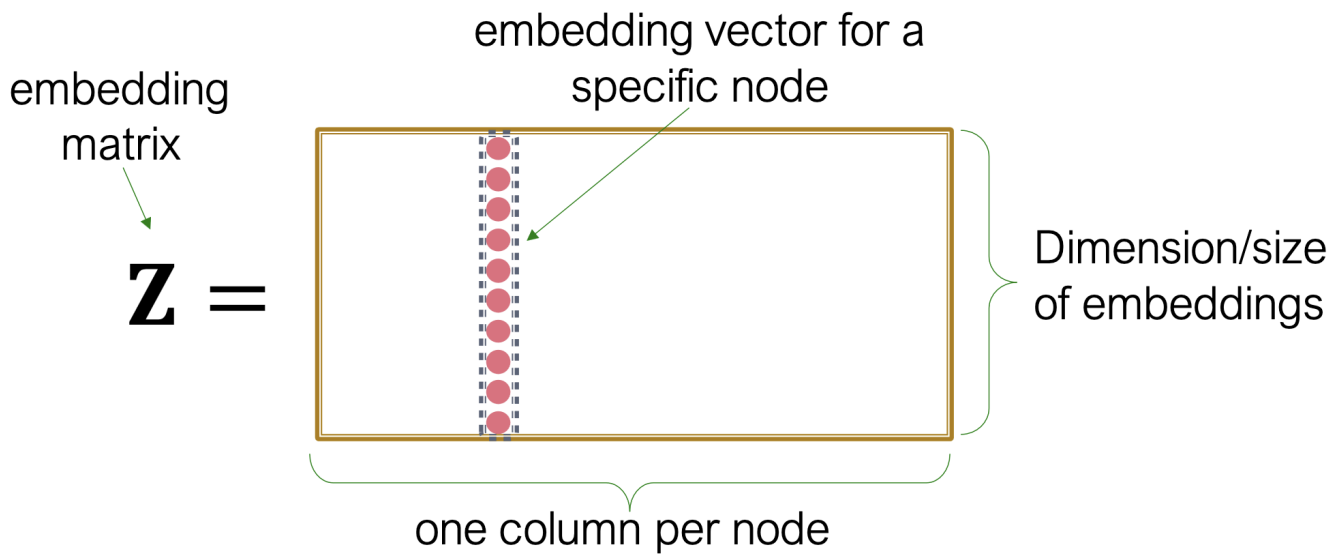
Ως συνέπεια των παραπάνω:

- Μάθηση ενσωματώσεων δεν ανήκει στο κλασσικό παράδειγμα επιβλεπόμενης μάθησης.
- Η υπόθεση ανεξαρτησίας των δειγμάτων (iid) δεν ισχύει διότι ο κόμβος επηρεάζεται από τους γείτονες.
- Οι ενσωματώσεις δεν σχετίζονται με τον τύπο μάθησης.

Ορισμός Κωδικοποιητή με Lookup table

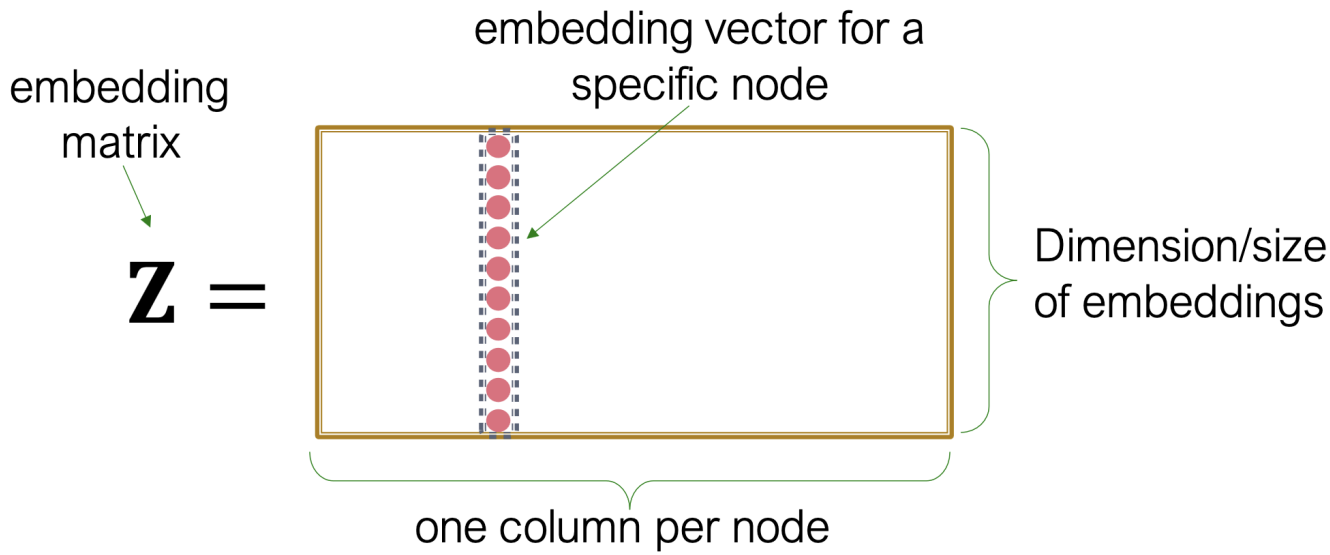
Απλούστερη προσέγγιση:

- Lookup table: $ENC(u) = z_u = Z \cdot u$
- $Z \in \mathbb{R}^{d \times |V|}$ πίνακας με στήλες τις ενσωματώσεις (αυτό μαθαίνουμε)
- $u \in \mathbb{I}^{|V|}$ διάνυσμα τύπου indicator με μηδενικά παντού εκτός από τη θέση που αντιστοιχεί στον κόμβο



Πρόβλημα για γραφήματα με πολλούς κόμβους

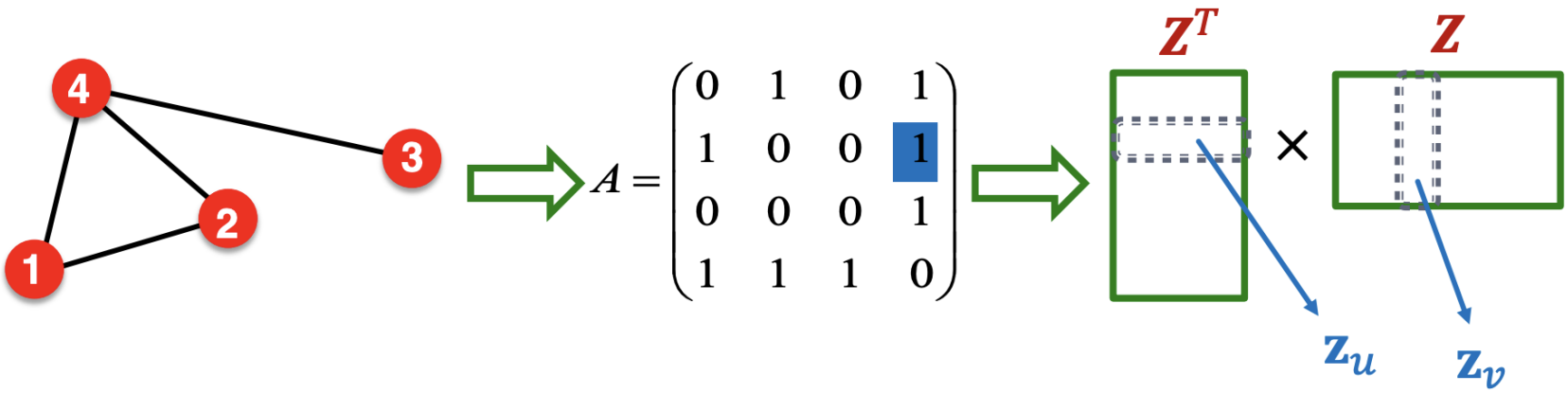
Ενσωμάτωση ως Παραγοντοποίηση πινάκων



Μεγιστοποίησε το γινόμενο $z_u \cdot z_v^t$ για u, v που είναι όμοια.

Ενσωμάτωση ως Παραγοντοποίηση πινάκων

- Απλούστερη ομοιότητα: οι κόμβοι συνδέονται με ακμή
- Αυτό σημαίνει ότι $z_u^t \cdot z_v = A_{uv}$
- Και συνολικά επομένως $Z^t \cdot Z = A$



Ενσωμάτωση ως Παραγοντοποίηση πινάκων

- Η διάσταση της ενσωμάτωσης d (αριθμός γραμμών του Z) είναι πολύ μικρότερη από τον αριθμό κόμβων n
- Ακριβής παραγοντοποίση $A = Z^T Z$ είναι γενικά μη εφικτή
- Μπορεί να υπολογιστεί προσεγγιστικά: $\min_Z \left\| A - Z^T Z \right\|_2$
- Βρίσκουμε το Z που ελαχιστοποιεί την L_2 νόρμα
- Ομοίως μπορούμε να χρησιμοποιήσουμε τη softmax όπως είδαμε

Συμπέρασμα: Αποκωδικοποιητής εσωτερικού γινομένου με ομοιότητα τη σύνδεση μέσω ακμής ισοδυναμεί με παραγοντοποίηση

Random walks: Σημειογραφία

- Διανυσμα ενσωμάτωσης z_u του κόμβου u (αυτό που θέλουμε να υπολογίσουμε)
- Πιθανότητα επίσκεψης του κόμβου v δεδομένου του z_u : $p(v|z_u)$ (βάσει του εκπαιδευμένου μοντέλου μας)
- Συνάρτηση softmax: $\sigma(z)_i = \frac{e^{z[i]}}{\sum_{j=1}^{K} e^{z[j]}}$

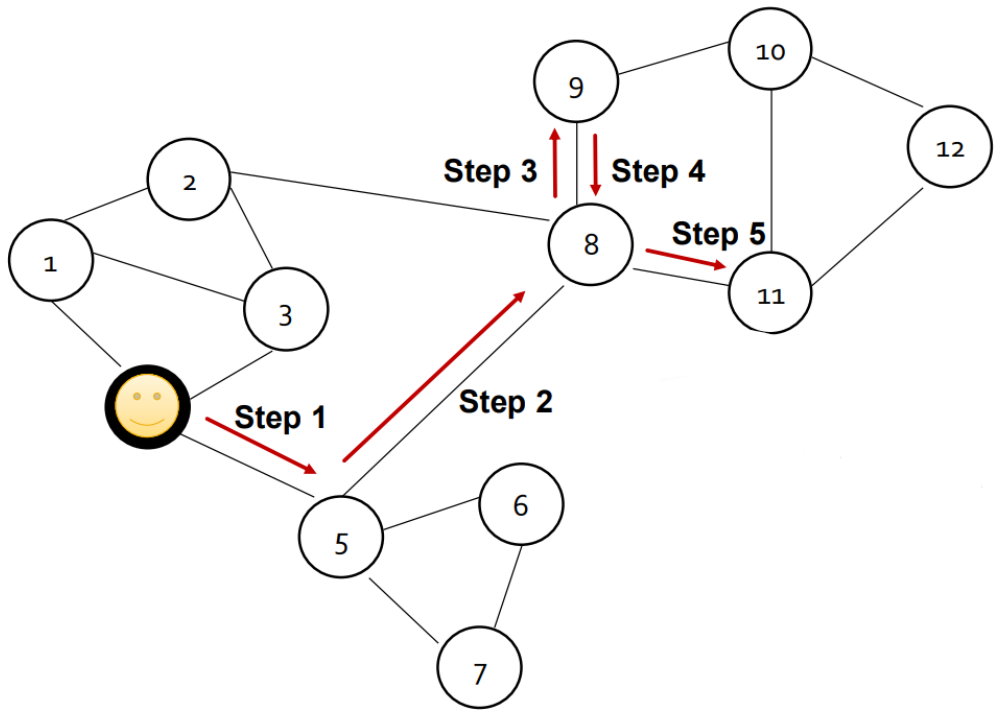
  Μετατρέπει διάνυσμα z διάστασης K σε πιθανότητες.
- Σιγμοειδής συνάρτηση:

$$s(x) = \frac{1}{1 + e^{-x}}$$

  Απεικονίζει τις πραγματικές τιμές στο διάστημα (0,1).

Τυχαίος περίπατος (Random Walk - RW)

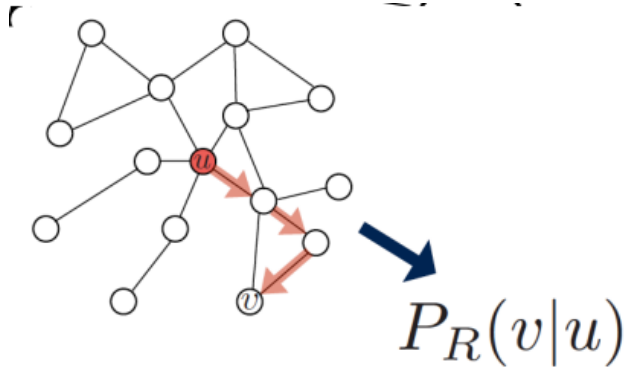
- Ξεκίνα από ένα κόμβο και προχώρα στον επόμενο τυχαία. Στη συνέχεια επανάλαβε ν φορές.
- Η ακολουθία κόμβων που προκύπτει είναι ο τυχαίος περίπατος.



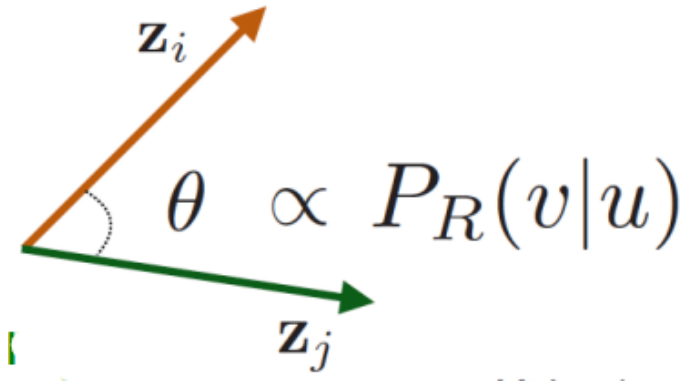
$z_u^T z_v$: η πιθανότητα u, v να συμπεριλαμβάνονται στο ίδιο RW.

Ενσωματώσεις βασισμένες σε Random Walk - RW

- Υπολόγισε την πιθανότητα να φτάσεις στον κόμβο v ξεκινώντας από τον u με βάση μια στρατηγική RW.



- Βελτιστοποίησε τις ενσωματώσεις ώστε να κωδικοποιούν αυτές τις πιθανότητες (π.χ. μέσω εσωτερικού γινομένου).



Γιατί Random Walk?

- Εκφραστικότητα: αναπαριστά τόσο τη γειτονιά όσο και την ευρύτερη δομή γύρω από τον κόμβο
- Υπολογιστικά αποδοτικό: δεν χρειάζεται να επεξεργαστούμε όλα τα ζεύγη κόμβων, μόνο αυτό που συνυπάρχουν στο RW

Η ιδέα:

- Βρες ενσωματώσεις στο d-διάστατο χώρο που διατηρούν την ομοιότητα.
- Κοντινοί κόμβοι στο χώρο ενσωμάτωσης θα πρέπει να είναι κοντινοί στο γράφημα
- Πώς ορίζεται η εγγύτητα στο γράφημα; με μια στρατηγική RW

Γιατί Random Walk?

- Μάθε συνάρτηση

$$f : u \rightarrow \Re^d$$
$$f(u) = z_u$$

- $$\max_f \sum_{u \in V} \log \mathrm{P}(N_{\mathrm{R}}(u) | \ z_u)$$

  όπου $N_{\mathrm{R}}(u)$ η γειτονιά με βάση στρατηγική R, δηλαδή οι κόμβοι που επισκεπτόμαστε κατά το RW
- Δεδομένου κόμβου u θέλουμε αναπαράσταση f τέτοια ώστε να μπορεί να χρησιμοποιηθεί για να προβλέπει τους κόμβους στη γειτονιά $N_{\mathrm{R}}(u)$

Βελτιστοποίηση

- Κάνε *RW* με περιορισμένο μήκος χρησιμοποιώντας μια στρατηγική R ξεκινώντας από κάθε κόμβο u
- Για κάθε u όρισε το σύνολο κόμβων που επισκέπτεσαι κατά το *RW* ξεκινώντας από το u δεχομενος ότι ο ίδιος κόμβος μπορεί να εμφανίζεται περισσότερες φορές
- Βελτιστοποίησε τις ενσωματώσεις προσπαθώντας να προβλέψεις τη γειτονιά $N_R(u)$ δεδομένου του u

$$\max_f \sum_{u \in V} \log \mathrm{P}(N_{\mathrm{R}}(u) |\ \mathrm{z}_u)$$

Βελτιστοποίηση

- Ισοδύναμα ελαχιστοποίησε:

$$\mathcal{L} = \sum_{u \in V} \sum_{v \in N_R(u)} -\log \mathrm{P}(v|\ \mathrm{z}_u)$$

- Διαίσθηση: βελτιστοποίησε τα z_u για να μεγιστοποιήσεις την πιθανότητα να επισκευτούμε τα v από τα u κατά το *RW*
- Παραμετροποιούμε την $\mathrm{P}(v|\ \mathrm{z}_u)$ με χρήση της *softmax*
  $P(v \mid \mathrm{z}_u) = \frac{\exp\left(\mathrm{z}_u^{\mathrm{T}} \mathrm{z}_v\right)}{\sum_{n \in V} \exp(\mathrm{z}_u^{\mathrm{T}} \mathrm{z}_n)}$
  διότι θέλουμε η αναπαράσταση του z_u να ταιριάζει με την αναπαράσταση του z_v
- Λόγω του διπλού αθροίσματος και του παρονομαστή πολυπλοκότητα $O(V^2)$

Βελτιστοποίηση

Θέλουμε να ελαχιστοποιήσουμε:

$$\mathcal{L} = \sum_{u \in V} \sum_{v \in N_R(u)} -\log \frac{\exp\left(\mathrm{z}_u^{\mathrm{T}} \mathrm{z}_v\right)}{\sum_{n \in V} \exp\left(\mathrm{z}_u^{\mathrm{T}} \mathrm{z}_n\right)}$$

Κάνουμε προσέγγιση για να μειώσουμε το κόστος υπολογισμού του παρονομαστή:

$$\log \frac{\exp\left(\mathrm{z}_u^{\mathrm{T}} \mathrm{z}_v\right)}{\sum_{n \in V} \exp\left(\mathrm{z}_u^{\mathrm{T}} \mathrm{z}_n\right)} \approx$$

$$\log\left(\sigma\left(\mathrm{z}_u^{\mathrm{T}} \mathrm{z}_v\right)\right) - \sum_{i=1}^{k} \log\left(\sigma\left(\mathrm{z}_u^{\mathrm{T}} \mathrm{z}_{n_i}\right)\right), n_i \sim P_V$$

όπου σ η σιγμοειδής συνάρτηση και P_V μια κατανομή πιθανότητας.

Κανονικοποιούμε για k δείγματα κόμβων αντί για όλους τους κόμβους (αρνητικά δείγματα).

Βελτιστοποίηση

Αρνητική δειγματοληψία: Δειγματίζουμε με πιθανότητα ανάλογη με το βαθμό του κάθε κόμβου.

Μεγάλο k:

- Πιο εύρωστες εκτιμήσεις (θετικό)
- Ο 2ος όρος αποκτά μεγαλύτερο βάρος (αρνητικό)
- Αριθμός εξαρτάται από τα χαρακτηριστικά του γραφήματος, αλλά συνήθως επιλέγουμε 5-20 αρνητικά δείγματα
- Πιο σωστό να μη συμπεριλαμβάνουμε κόμβους που ανήκουν στο RW

$[https://arxiv.org/pdf/1402.3722.pdf]$

Βελτιστοποίηση

Stochastic Gradient Descent

- Αρχικοποίησε τα z_u σε τυχαίες τιμές
- Επανάλαβε μέχρι τη σύγκλιση:
  - Δειγμάτισε κόμβο u και θεώρησε την $\mathcal{L}^{(u)}$ το *loss* μόνο για τον κόμβο u
  - Για όλα τα v υπολόγισε την παράγωγο $\frac{\partial \mathcal{L}^{(u)}}{\partial z_v}$
  - $z_v \leftarrow z_v - \eta \frac{\partial \mathcal{L}^{(u)}}{\partial z_v}$

Για ισοπίθανη επιλογή ακμών ειναι η μέθοδος *Deepwalk* : Perozzi et al. 2014. DeepWalk: Online Learning of Social Representations. KDD.

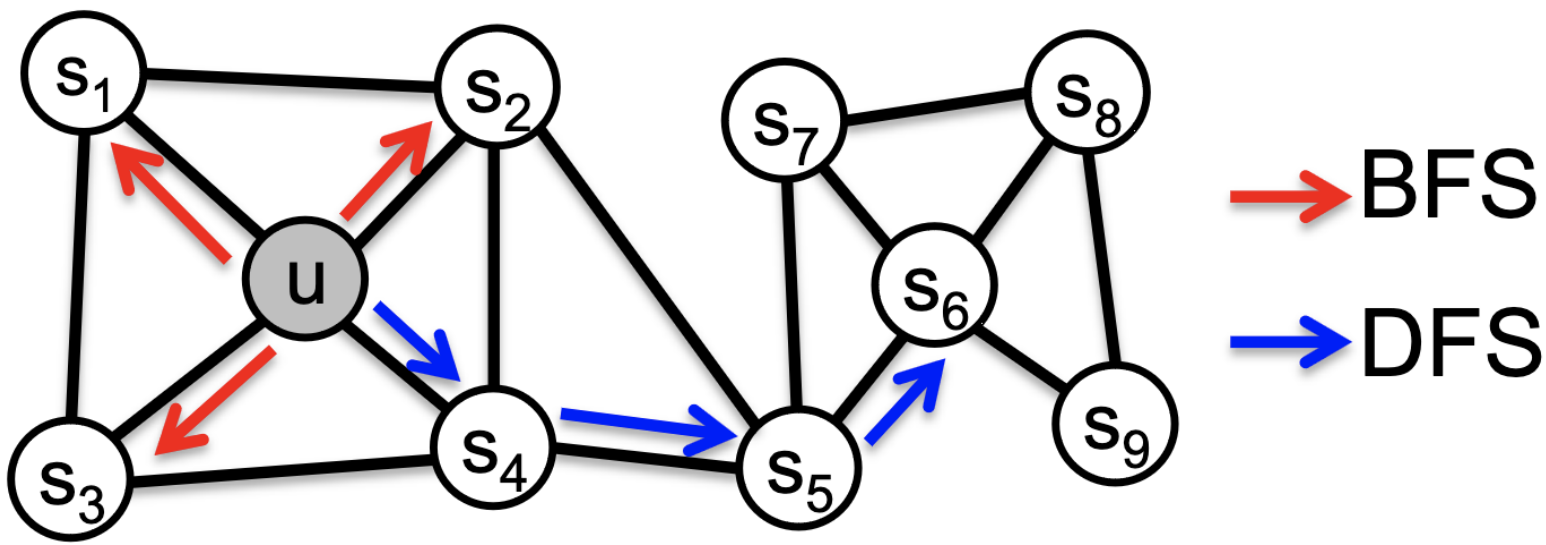
Αλγόριθμος node2vec

1. Στόχος: κόμβοι με παρόμοιους γείτονες αναπαρίστανται κοντά στο χώρο ενσωμάτωσης
2. Τυποποιούμε το πρόβλημα ως βελτιστοποίηση μέγιστης πιθανοφάνειας και ανεξάρτητα από το τι είδους πρόβλημα έχουμε
3. Παρατήρηση: η ευέλικτη έννοια της γειτονιάς $N_R(u)$ του κόμβου u δίνει δυνατότητα κωδικοποίησης της πληροφορίας
4. Δημιουργούμε δεύτερης τάξης *RW*
5. Διαφορά με *deepwalk* : ο τρόπος με τον οποίο ορίζεται το *RW* και επομένως η γειτονιά.

Αλγόριθμος node2vec

Ιδέα: χρησιμοποίησε ευέλικτα *RW* που να συνδυάζουν τοπική και συνολική πληροφορία του γραφήματος.

1. *BFS*: δίνει τοπική πληροφορία $N_{BFS}(u) = \{s_1, s_2, s_3\}$
2. *DFS*: δίνει υπερτοπική πληροφορία $N_{DFS}(u) = \{s_4, s_5, s_6\}$



Αλγόριθμος node2vec

Πώς ρυθμίζουμε το συνδυασμό *BFS* – *DFS* κατά το *RW*;

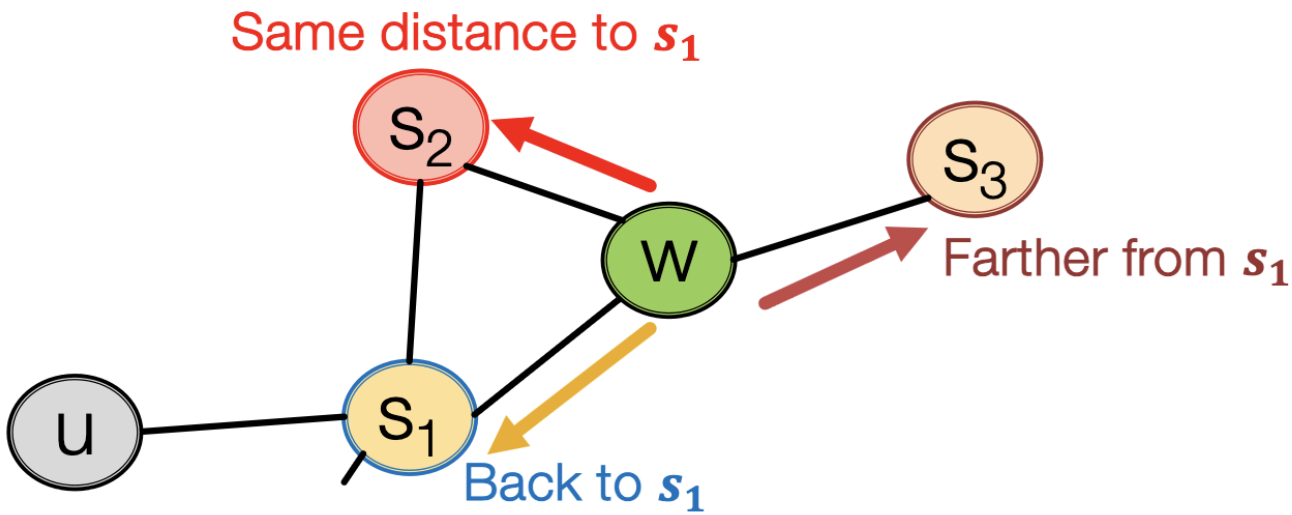
Δύο παράμετροι:

1. p καθορίζει την επιστροφή στον προηγούμενο κόμβο
2. q ο λόγος *BFS* (κατευθύνσου προς τα μέσα) προς *DFS* (κατευθύνσου προς τα έξω)

Αλγόριθμος node2vec

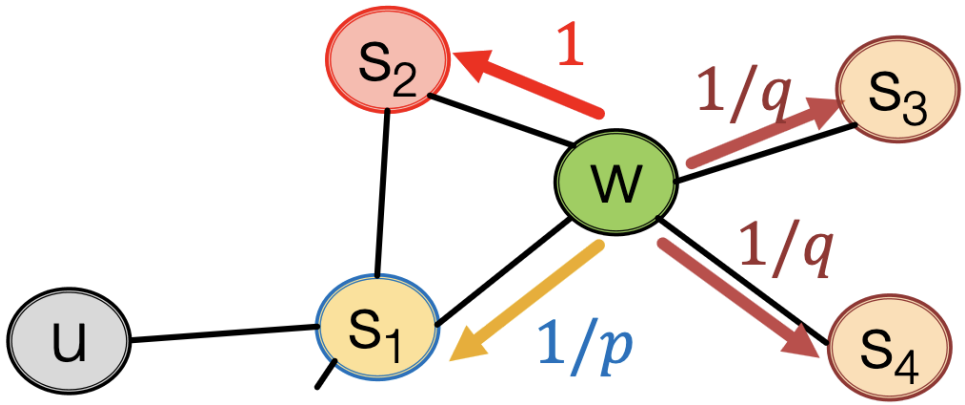
RW 2ης τάξης (θυμάται από που προήλθε)

RW μόλις πήγε από το s_1 στο w. Οι επιλογές για τη συνέχεια είναι:



Αλγόριθμος node2vec

RW μόλις πήγε από το s_1 στο w. Οι επιλογές για τη συνέχεια είναι:



1. παράμεινε στην ίδια απόσταση (s_2) με πιθανότητα ανάλογη μιας σταθεράς
2. γύρνα πίσω (s_1) με πιθανότητα $1/p$
3. προχώρα πιο μακριά με πιθανότητα $1/q$ (s_3, s_4)

Οι πιθανότητες είναι μη κανονικοποιημένες.

- μικρό $p \rightarrow$ *BFS*
- μικρό $q \rightarrow$ *DFS*

Αλγόριθμος node2vec

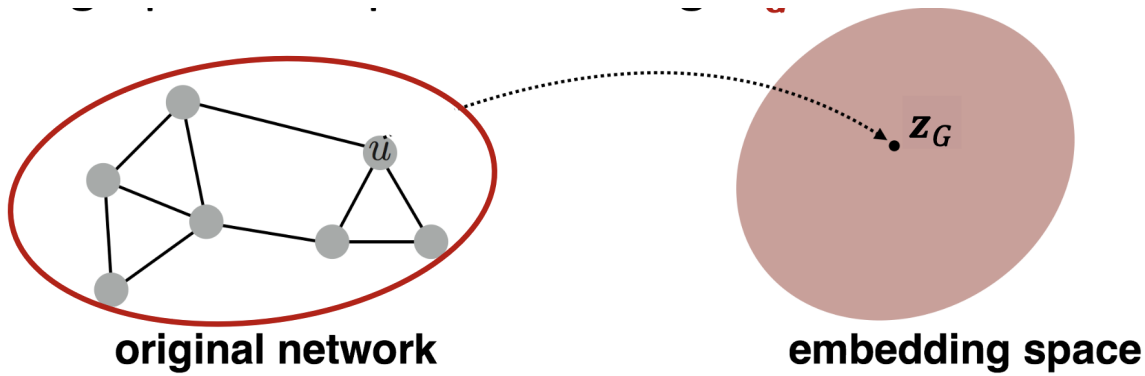
1. Υπολόγισε τις πιθανότητες μετάβασης
2. Υλοποίησε r *RWs* μήκους l ξεκινώντας από κάθε κόμβο u
3. Βελτιστοποίησε την αντικειμενική συνάρτηση χρησιμοποιώντας *SGD* (όμοια με *DeepWalk*)

(+) Γραμμικό κόστος - όλα τα τρία βήματα είναι παραλληλοποιήσιμα
(-) Για κάθε κόμβο πρέπει να μάθουμε ένα διαφορετικό διάνυσμα ενσωμάτωσης

Ενσωμάτωση ολόκληρων γραφημάτων

Στόχος η ενσωμάτωση ολόκληρου γραφήματος ή υποσυνόλου του.

π.χ. για ταξινόμηση σε τοξική ή μη τοξική κοινότητα σε ΚΔ.



Ενσωμάτωση ολόκληρων γραφημάτων: Προσέγγιση Α

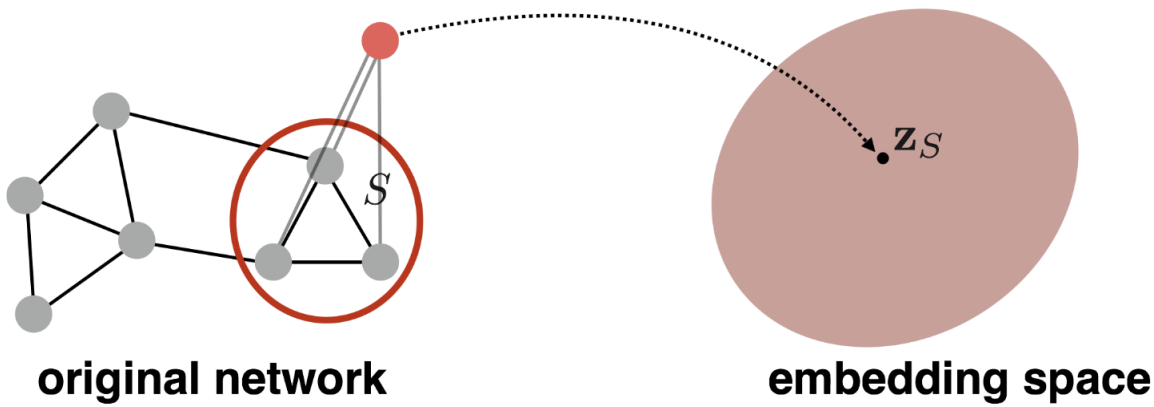
Απλή αλλά πολλές φορές αποτελεσματική:

1. Τρέξε κάποιον αλγόριθμο ενσωμάτωσης για κόμβους
2. Βγάλε τη μέση τιμή των ενσωματώσεων z_u

π.χ. Duvenaud et al., 2016 για ταξινόμηση μορίων

Ενσωμάτωση ολόκληρων γραφημάτων Προσέγγιση Β

1. Όρισε ένα κόμβο πλήρως συνδεδεμένο με το (υπο)γράφημα που μας ενδιαφέρει
2. Τρέξε για το συγκεκριμένο κόμβο κάποιον αλγόριθμο ενσωμάτωσης για κόμβους



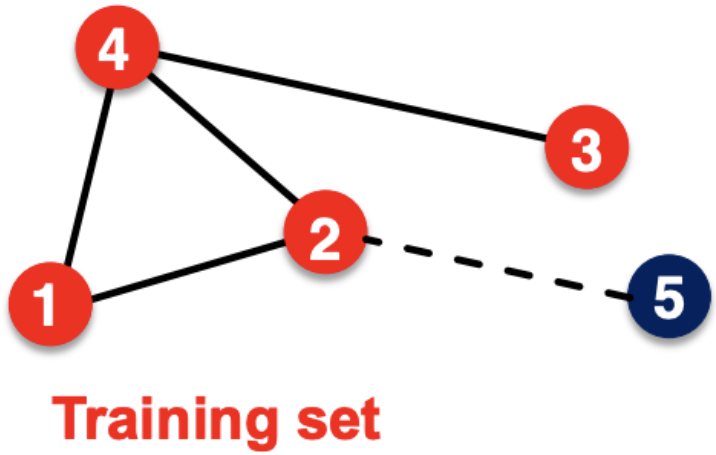
π.χ. Li et al., 2016 για γενική χρήση

Χρήση διανυσμάτων ενσωμάτωσης

- Ομοδοποίηση
- Ταξινόμηση: πρόβλεψε την ετικέτα του u βάσει του z_u
- Πρόβλεψη ακμής (u, v) βάσει του (z_u, z_v)
- Ταξινόμηση γραφήματος: πάρε aggregate (π.χ. άθροισμα, μέση τιμή) των κόμβων, ή εικονικός κόμβος

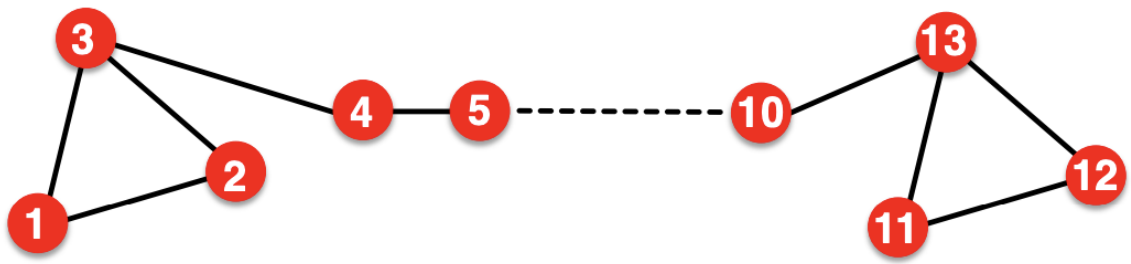
Περιορισμοί *RW* και παραγοντοποίησης

- Δεν μπορούν να υπολογίσουν ενσωμάτωση κόμβων που δεν ανήκουν στο σύνολο εκπαίδευσης
- Δεν μπορούν να εφαρμοστούν απευθείας σε νέα/εξελισσόμενα γραφήματα.
- Πρέπει να γίνει υπολογισμός από την αρχή σε όλους τους κόμβους



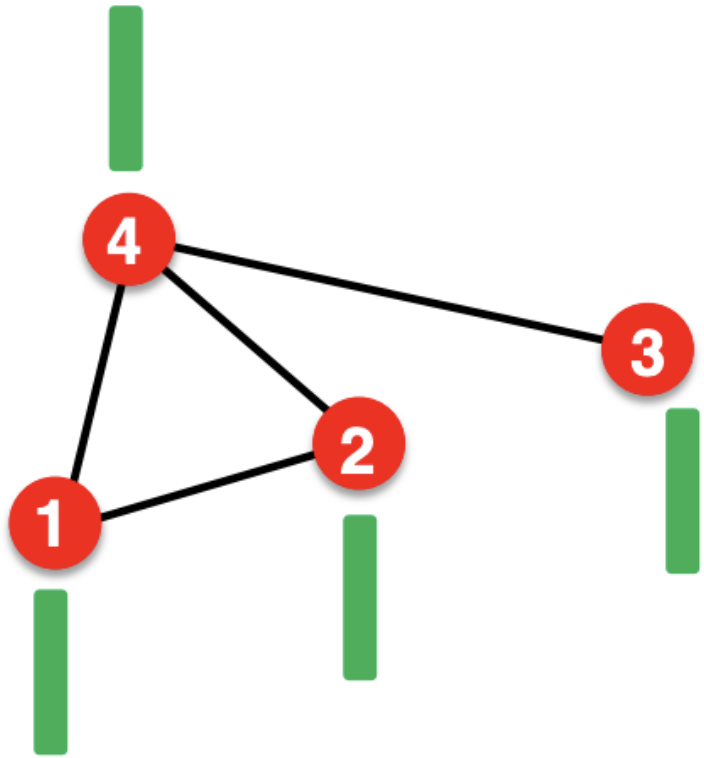
Περιορισμοί *RW* και παραγοντοποίησης

- Δεν μπορούν να υπολογίσουν γενικά την δομική ομοιότητα
- Οι κόμβοι 1, 11 είναι δομικά παρόμοιοι (μέλη τριγώνων, βαθμός 2), αλλά το *RW* είναι μάλλον απίθανο να φτάσει από τον ένα στον άλλο αν απέχουν πολύ



Περιορισμοί *RW* και παραγοντοποίησης

- Δεν μπορούν να χρησιμοποιήσουν χαρακτηριστικά/ιδιότητες που υπάρχουν ήδη σε κόμβους/ακμές/γραφήματα π.χ. ιδιότητες ατόμων σε ένα μόριο
- Λύση: βαθιά αναπαράσταση



Βιβλιογραφία

1. W.L. Hamilton, Graph Representation Learning, McGill University, 2020
2. J. Leskovec, Machine Learning with Graphs, Stanford University, Fall 2023