

Node2Vec

Authors: Felipe Perez, Ella Chen, and Serena McDonnell

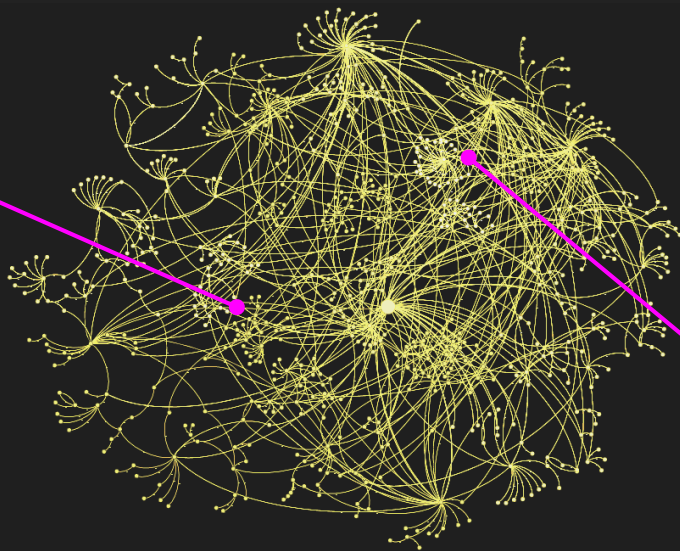
Nov 2019

This Slide Deck is part of the workshop "Recommender Systems" run by Aggregate Intellect Inc. (<https://ai.science>), and is released under 'Creative Commons Attribution-NonCommercial-ShareAlike CC BY-NC-SA' license. This material can be altered and distributed for non-commercial use with reference to Aggregate Intellect Inc. as the original owner, and any material generated from it must be released under similar terms (<https://creativecommons.org/licenses/by-nc-sa/4.0/>).

Why study graphs for recommender?

- Graphs encode underlying structure of the data
 - Modeling interactions between interactions
- Data can contain more than one underlying graph structure.

Getting info from graphs



Getting info from graphs

- Usual feature generation:

Number of neighbors, sum of weights, cluster size, etc.

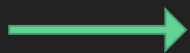
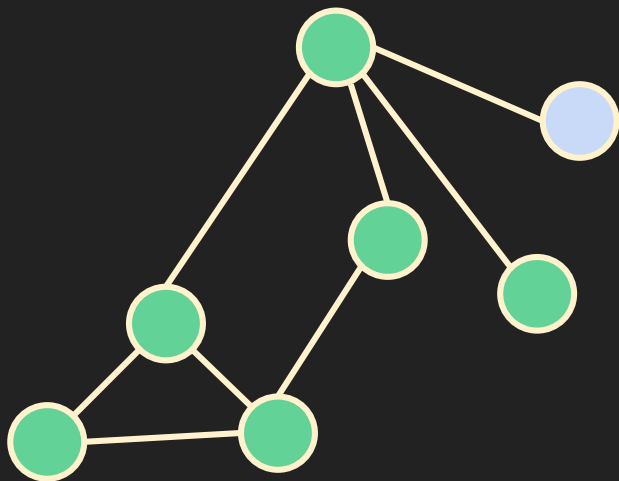
- Algorithmic approaches, (e.g. PageRank):

Unsupervised way to get a feature (that we can describe)

- Automatic feature generation:

Node2vec

Node2vec



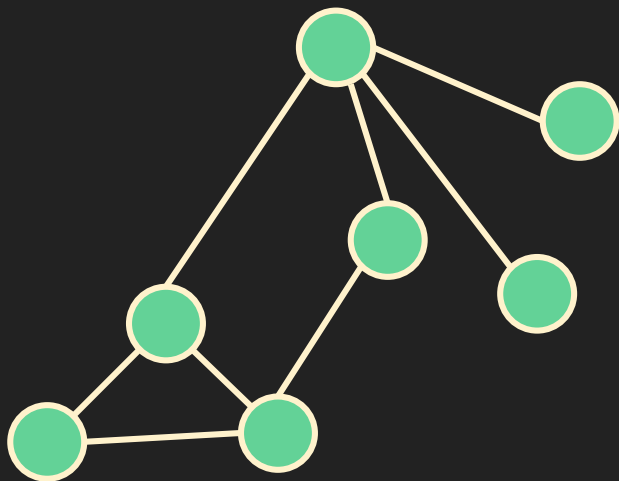
1	2	1	4	1	2	9	6
---	---	---	---	---	---	---	---

$$f : \text{Nodes} \rightarrow \mathbb{R}^n$$

Node2vec - Difficulties

- What is the right dimension to choose?
- Do we want the embedding to preserve/encode a particular property of the graph (local density, long paths, sparsity, etc.)?
- Scalability.

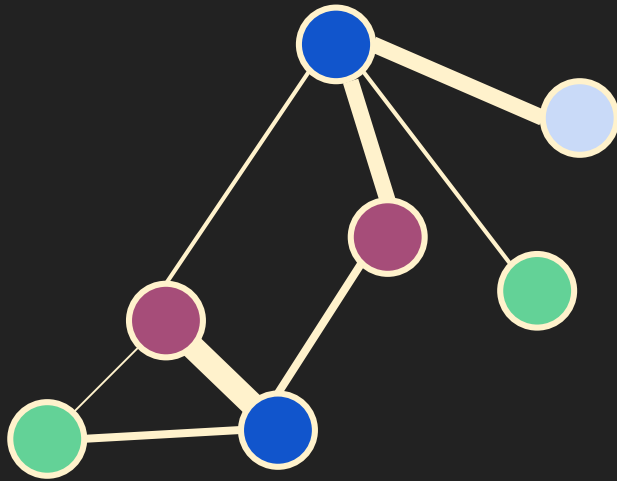
Node2vec



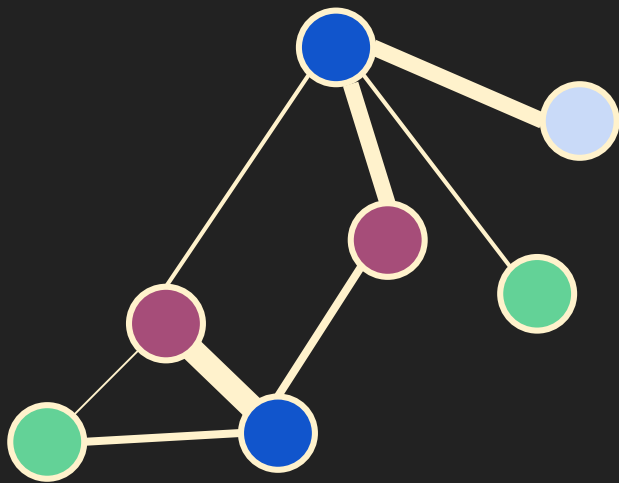
Graphs have properties:

- Nodes represent different identities.
- Edges can represent strength of the relation.

Graph Properties to use



A node is known by its neighbors

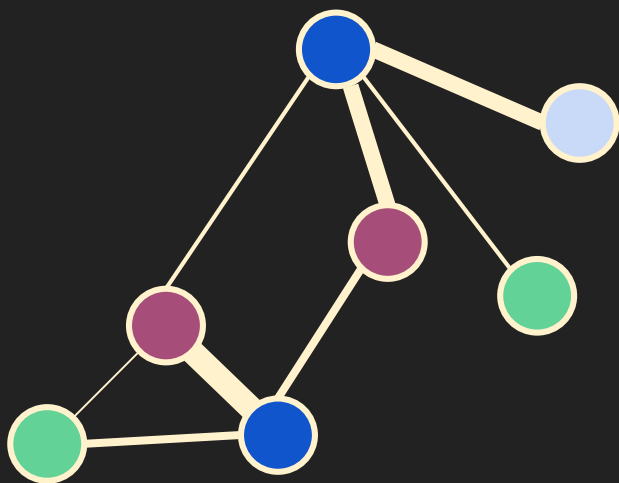


$P(\text{Node } a \text{ is neighbour} |_{\text{staying at } b})$

$$= \frac{e^{a_c \cdot b_n}}{\sum_{b'} e^{a_c \cdot b'_n}}$$

This is similar to Word2Vec!

A node is known by its neighbors



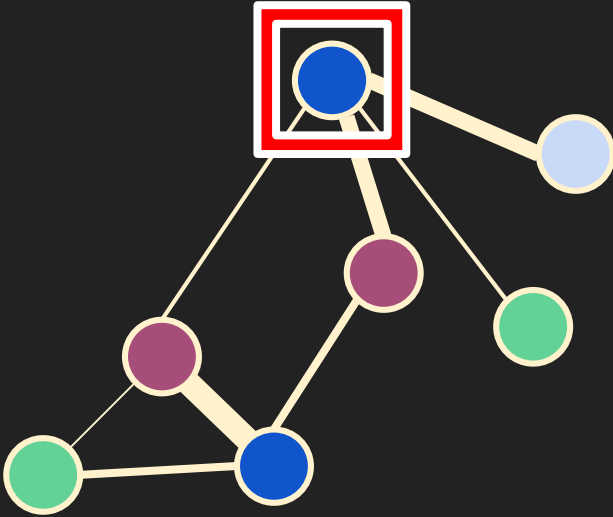
$$P(\text{Node } a \text{ is neighbour} | \text{staying at } b)$$

$$= \frac{e^{a_c \cdot b_n}}{\sum_{b'} e^{a_c \cdot b'_n}}$$

This is similar to Word2Vec!

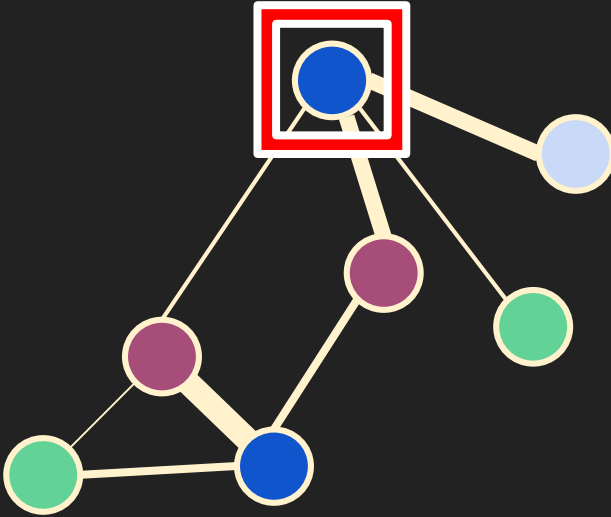
**WHAT DOES IT MEAN TO BE A
NEIGHBOUR?**

A node is known by its neighbors



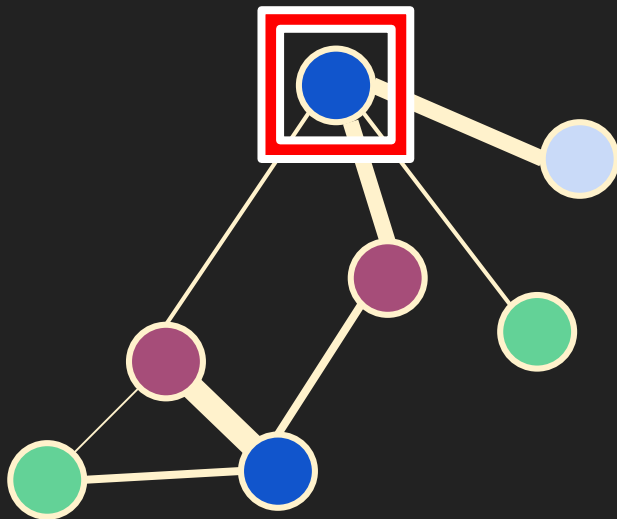
- What are the neighbours of this node?

A node is known by its neighbors



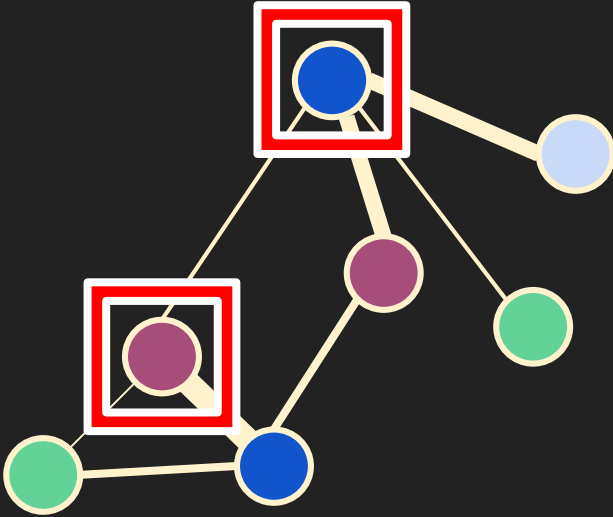
- What are the neighbours of this node?
- Are there some nodes that are more neighbours than others?

A node is known by its neighbors



- What are the neighbours of this node?
- Are there some nodes that are more neighbours than others?
- How to encode the node properties into the neighbours?

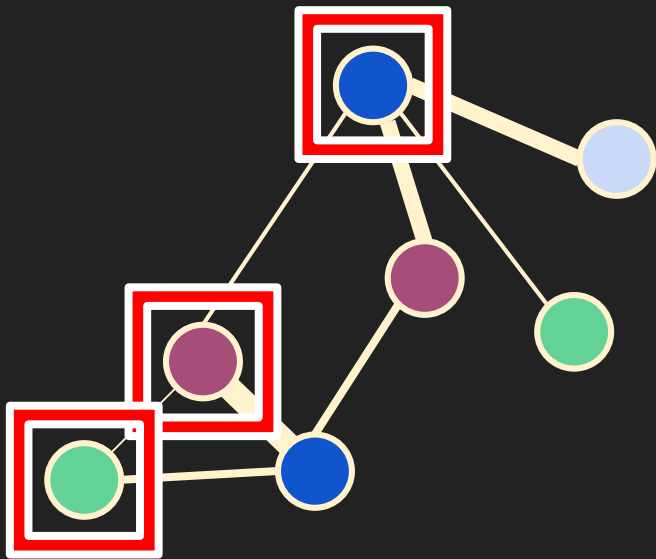
Knowing the neighbors by walking around



Random walk:

- Start at node and walk around by using local info as probabilities.

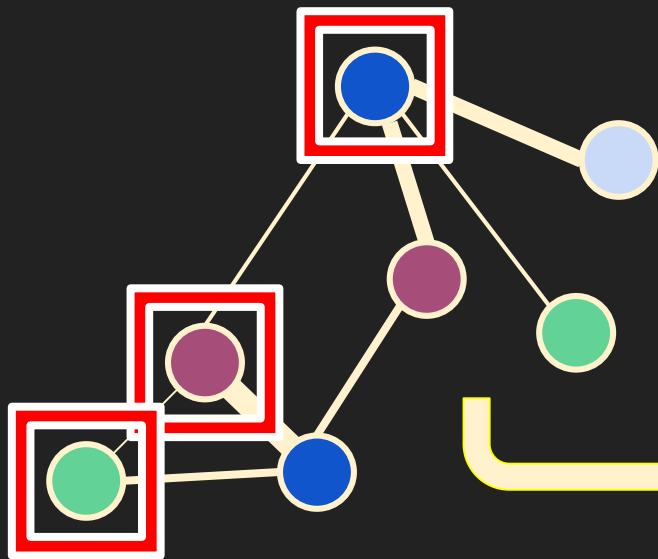
Knowing the neighbors by walking around



Random walk:

- Start at node and walk around by using local info as probabilities.

Knowing the neighbors by walking around



Walk = Sentences:

- Make the walks into sentences and apply word2vec

