word2vec

Task: Learn good representations of words

why : Unlike images you can't feed-in words to ML models.
One-hot → sparse

How : Learn embeddings as a bi-product of some task.
what task? Data?

Language Modelling

Predict w_{i+1} given $w_0\ w_1 \dots w_i$

why is this good? Suddenly you have lots & lots of labelled data.

Skip Gram	CBOW.
given word predict context	given context predict word.

word2vec

→ words occuring together should be close.

→ words not occuring together should be far.

$$p(c|w) = \frac{e^{v_c \cdot v_w}}{\sum_{c' \in C} e^{v_{c'} \cdot v_w}}$$

Why is this intractable?

Negative Sampling.

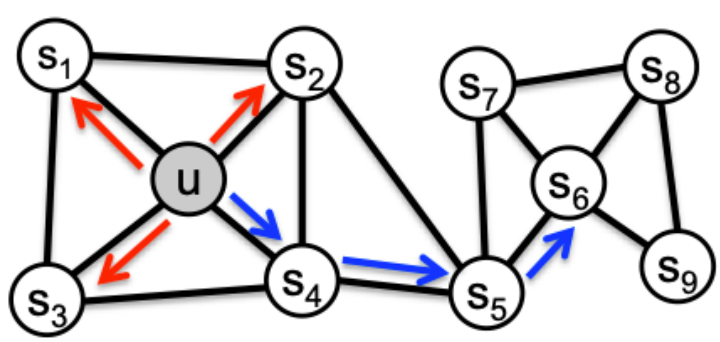
$$\arg\max_{\theta} \prod_{(w,c) \in D} p(D=1 | c, w, \theta) \prod_{(w,c) \in D'} p(D=0 | c, w, \theta)$$

$$= \arg\max_{\theta} \underbrace{\sum_{(w,c) \in D} \log\left[\frac{1}{1+e^{-v_c \cdot v_w}}\right]}_{\text{bring } (w,c) \text{ closer if in data}} + \underbrace{\sum_{(w,c) \in D'} \log\left[\frac{1}{1+e^{v_c \cdot v_w}}\right]}_{\text{get them farther away if not in data.}}$$

You have seen that this works. ← Reading.

Question: Is language related to graphs?

node 2 vec.



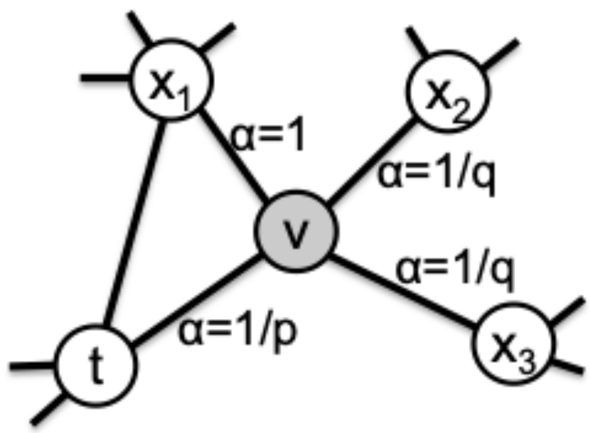
Option 1 : Walks.

BFS DFS.

structural homophily.

similar structure highly interconnected.

Option 2: Random walks.



p ↑ ⟶ sample new node

return param.

q ↑ ⟶ baised towards nodes closer to t.
(BFS)

in-out param

Can we apply node2vec to our graph?

→ we can. BUT . . .

consider the api - api edges!.

↳ I will dominate all other types.

useless. as all will be connected.

Benefits of this approach?

→ interpretability. ← plot the apps. and api's.

→ gives you a representation of app's and api's which can be used in any model of your choice.