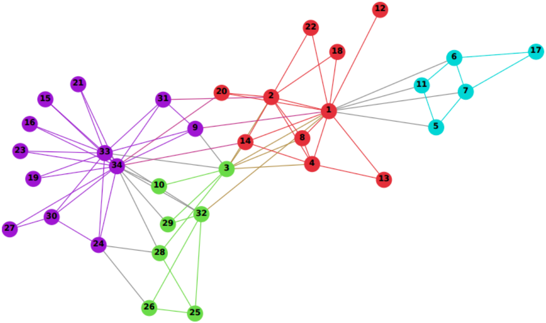
DeepWalk: Online Learning of Social Representations [1]

Presented by Carlos Oliver for COMP 766

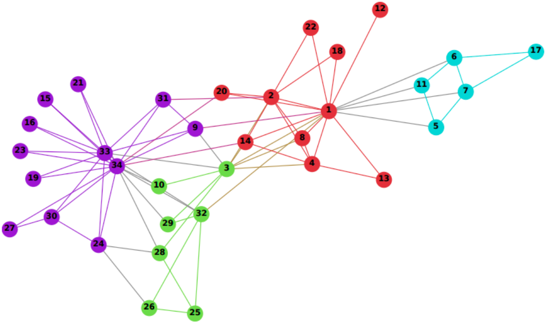
January 20, 2020

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

Motivation

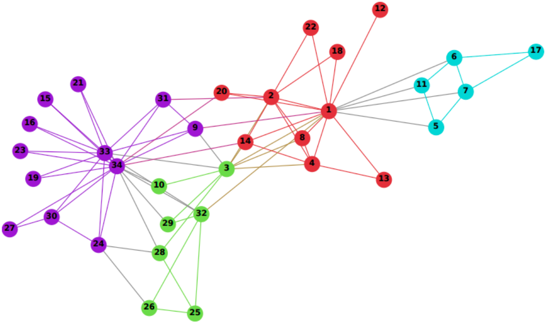

12
22
18
17
6
11
7
21
31
20
2
1
15
5
16
9
14
8
23
33
34
3
4
13
19
10
32
29
30
27
24
28
26
25

Motivation



- Can we predict the **label** of a node given the graph?

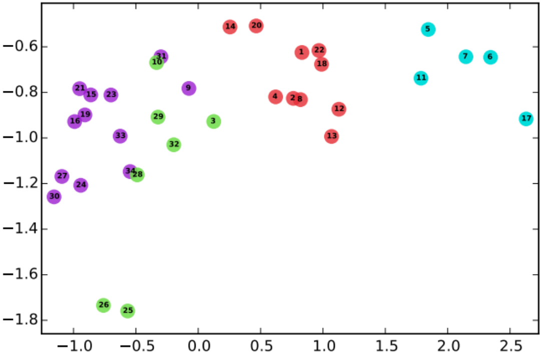
Motivation



- Can we predict the **label** of a node given the graph?
- **Problem:** labels not i.i.d so traditional methods can't be used. (MRF, Graph Kernels and other structured learning models needed)

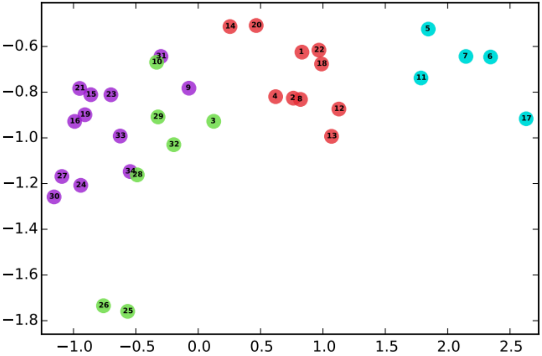
Idea

- Separate labels from underlying structure.



Idea

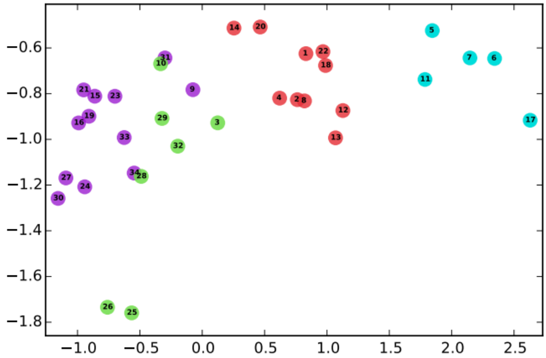
- Separate labels from underlying structure.



- Existing Approaches:
  - Graph statistics (neighbourhood overlap...)
  - Spectral clustering

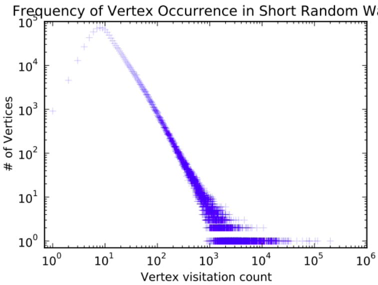
Idea

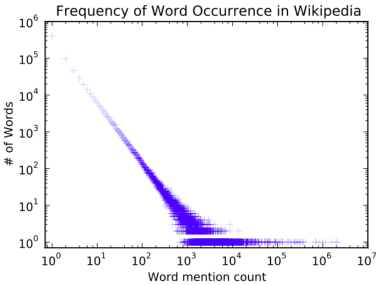
- Separate labels from underlying structure.



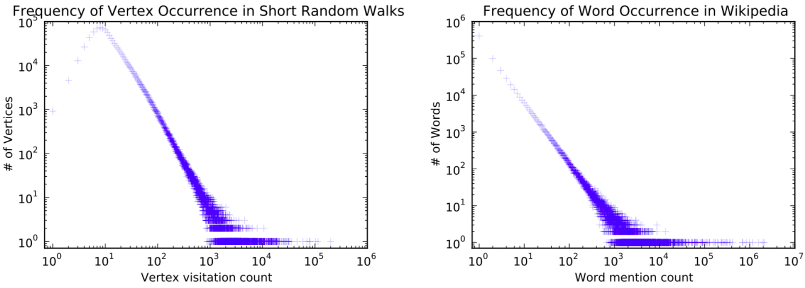
- Existing Approaches:
  - Graph statistics (neighbourhood overlap...)
  - Spectral clustering
- **Limitations:** often require full graph to compute or domain-specific knowledge.

Relationship between social graphs and natural language



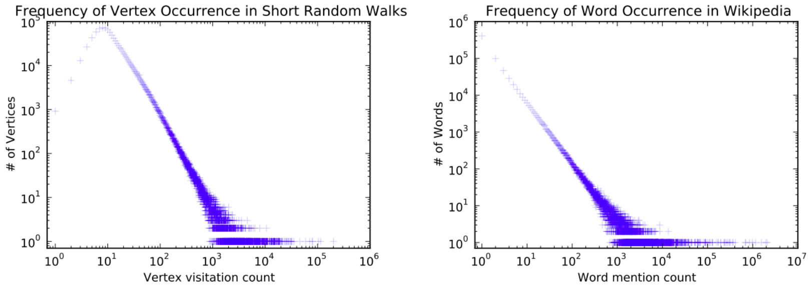


Relationship between social graphs and natural language



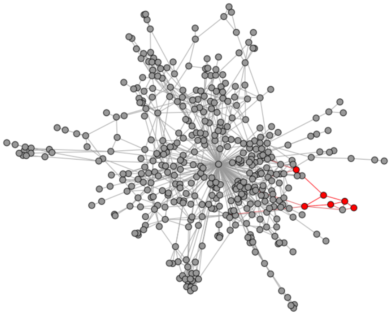
- This tells us that modeling the co-occurence of vertices gives us similar information to measuring co-occurence of words.

Relationship between social graphs and natural language



- This tells us that modeling the co-occurence of vertices gives us similar information to measuring co-occurence of words.
- Seeing words co-occur gives us information about the structure of the language (or graph).

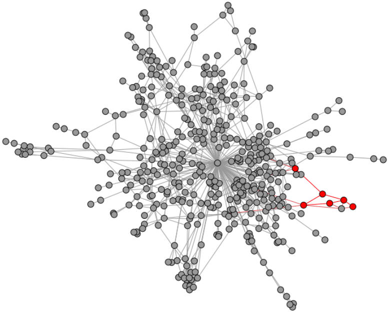
Random walks ∼ sentences



(a) Random walk generation.

- Since co-occurence tells us about the structural context. Sample random walks from each node.

Random walks ∼ sentences



(a) Random walk generation.

- Since co-occurence tells us about the structural context. Sample random walks from each node.
- **Bonus:** natural way to split up graph (i.e. don't need whole graph to get a node's embedding)

Learning objective

- Given a random walk, $(v_1, .., v_i)$, we update representation $\Phi(v_i) \in \mathbb{R}^d$ to maximizes the likelihood of the walk.

$$\arg\min_{\Phi} -\log \mathbb{P}(v_1, .., v_{i-1} | \Phi(v_i))$$

Learning objective

- Given a random walk, $(v_1, .., v_i)$, we update representation $\Phi(v_i) \in \mathbb{R}^d$ to maximizes the likelihood of the walk.

$$\arg\min_{\Phi} -\log \mathbb{P}(v_1, .., v_{i-1} | \Phi(v_i))$$

- Probabilities are given by a multi-label classifier which maps $\Phi(v_i) \to V^k$ where k is the size of the walks.

Learning objective

- Given a random walk, $(v_1, .., v_i)$, we update representation $\Phi(v_i) \in \mathbb{R}^d$ to maximizes the likelihood of the walk.

$$\arg\min_{\Phi} -\log \mathbb{P}(v_1, .., v_{i-1} | \Phi(v_i))$$

- Probabilities are given by a multi-label classifier which maps $\Phi(v_i) \to V^k$ where k is the size of the walks.
- Nodes with similar Φ will have similar local graph 'structure'.

Speedups: SkipGram

- **Skip-gram**: lets us split the walk into sliding windows size w and update nodes in each window using SGD.

Speedups: SkipGram

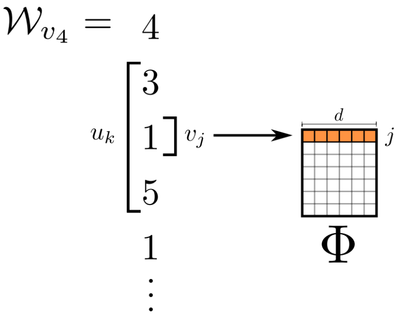
- **Skip-gram**: lets us split the walk into sliding windows size w and update nodes in each window using SGD.
- $\arg\min_{\Phi} - \log \mathbb{P}(v_1, .., v_{i-1} | \Phi(v_i))$ becomes

Speedups: SkipGram

- **Skip-gram**: lets us split the walk into sliding windows size w and update nodes in each window using SGD.
- $\arg\min_{\Phi} -\log \mathbb{P}(v_1, .., v_{i-1} | \Phi(v_i))$ becomes
- $\arg\min_{\Phi} -\log \mathbb{P}(v_{i-w}, v_{i-1}, v_{i+1} .., v_{i+w} | \Phi(v_i))$

Speedups: SkipGram

- **Skip-gram**: lets us split the walk into sliding windows size w and update nodes in each window using SGD.
- $\arg\min_\Phi - \log \mathbb{P}(v_1, .., v_{i-1} | \Phi(v_i))$ becomes
- $\arg\min_\Phi - \log \mathbb{P}(v_{i-w}, v_{i-1}, v_{i+1}.., v_{i+w} | \Phi(v_i))$



(b) Representation mapping.

Speedups: Hierarchical Softmax

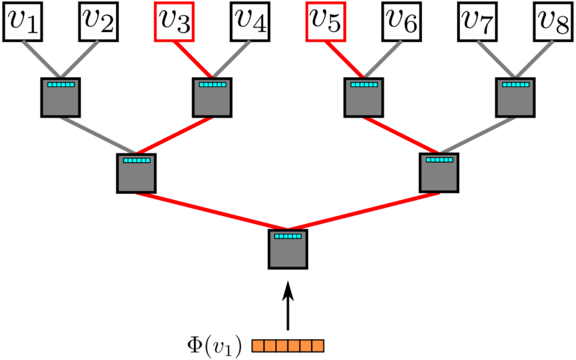
- **Hierarchical Softmax**: reduces softmax normalization from $\mathcal{O}(|V|)$ to $\mathcal{O}(\log |V|)$ by building tree of binary classifiers.

Speedups: Hierarchical Softmax

- **Hierarchical Softmax**: reduces softmax normalization from $\mathcal{O}(|V|)$ to $\mathcal{O}(\log |V|)$ by building tree of binary classifiers.
-
$$\mathbb{P}(b_k|\Phi(v_j)) = \Pi_{l=1}^{\log |V|}\mathbb{P}(b_l|\Phi(v_j))$$

Speedups: Hierarchical Softmax

- **Hierarchical Softmax**: reduces softmax normalization from $\mathcal{O}(|V|)$ to $\mathcal{O}(\log |V|)$ by building tree of binary classifiers.
- $$\mathbb{P}(b_k|\Phi(v_j)) = \Pi_{l=1}^{\log |V|}\mathbb{P}(b_l|\Phi(v_j))$$



(c) Hierarchical Softmax.

Training loop

Algorithm 1 DeepWalk(G, w, d, γ, t)

Input: graph $G(V, E)$
window size w
embedding size d
walks per vertex γ
walk length t
Output: matrix of vertex representations $\Phi \in \mathbb{R}^{|V| \times d}$

Initialization: Sample Φ from $\mathcal{U}^{|V| \times d}$
Build a binary Tree T from V
for $i = 0$ to γ **do**
    $\mathcal{O} = \text{Shuffle}(V)$
    for each $v_i \in \mathcal{O}$ **do**
        $\mathcal{W}_{v_i} = RandomWalk(G, v_i, \text{t})$
        SkipGram(Φ, $\mathcal{W}_{v_i}$, w)
    end for
end for

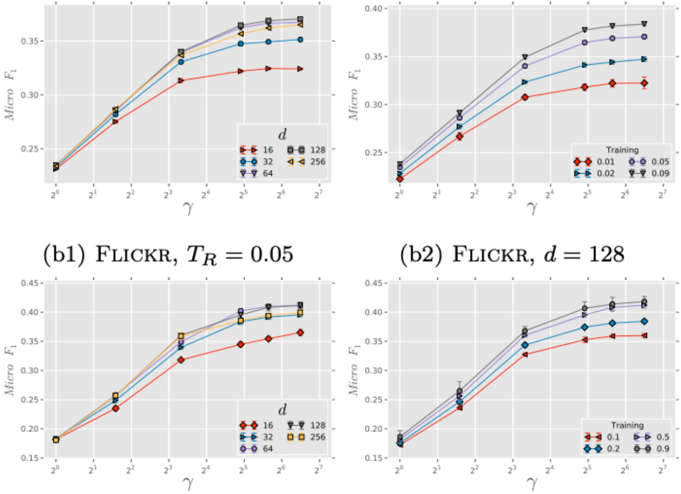
Evaluation

- **Task:** Multi-label node classification on large social graphs.
- **Evaluation:** micro,macro F1 score
  - F1 score is the harmonic mean of precision and recall
  - Macro is the arithmetic mean of F1 over all classes
  - Micro is total proportion of correct labels over all samples.

	% Labeled Nodes	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%
Micro-F1(%)	DeepWalk	**37.95**	**39.28**	**40.08**	**40.78**	**41.32**	**41.72**	**42.12**	**42.48**	**42.78**	**43.05**
	SpectralClustering	—	—	—	—	—	—	—	—	—	—
	EdgeCluster	23.90	31.68	35.53	36.76	37.81	38.63	38.94	39.46	39.92	40.07
	Modularity	—	—	—	—	—	—	—	—	—	—
	wvRN	26.79	29.18	33.1	32.88	35.76	37.38	38.21	37.75	38.68	39.42
	Majority	24.90	24.84	25.25	25.23	25.22	25.33	25.31	25.34	25.38	25.38
Macro-F1(%)	DeepWalk	**29.22**	**31.83**	**33.06**	**33.90**	**34.35**	**34.66**	**34.96**	**35.22**	**35.42**	**35.67**
	SpectralClustering	—	—	—	—	—	—	—	—	—	—
	EdgeCluster	19.48	25.01	28.15	29.17	29.82	30.65	30.75	31.23	31.45	31.54
	Modularity	—	—	—	—	—	—	—	—	—	—
	wvRN	13.15	15.78	19.66	20.9	23.31	25.43	27.08	26.48	28.33	28.89
	Majority	6.12	5.86	6.21	6.1	6.07	6.19	6.17	6.16	6.18	6.19

Table 4: Multi-label classification results in YouTube

Parameter Sensitivity



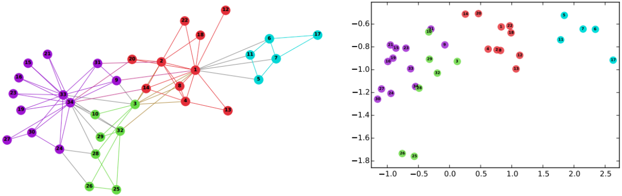
(b1) FLICKR, $T_R = 0.05$ (b2) FLICKR, $d = 128$

(b3) BLOGCATALOG, $T_R = 0.5$ (b4) BLOGCATALOG, $d = 128$

(a) Stability over number of walks, γ

Summary

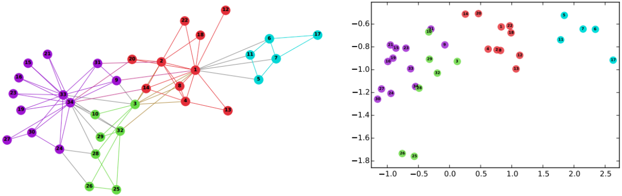
- Modelling random walks as sentences in a language gives us:



(a) Input: Karate Graph (b) Output: Representation

Summary

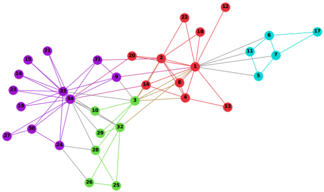
- Modelling random walks as sentences in a language gives us:
  - **Continuous & Low Dimensional representations** $\rightarrow$ robustness



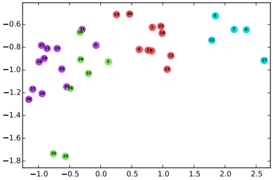
(a) Input: Karate Graph (b) Output: Representation

Summary

- Modelling random walks as sentences in a language gives us:
  - **Continuous & Low Dimensional representations** $\rightarrow$ robustness
  - **Online training** $\rightarrow$ walks naturally partition the graph



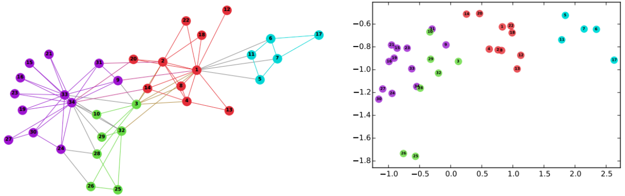
(a) Input: Karate Graph



(b) Output: Representation

Summary

- Modelling random walks as sentences in a language gives us:
  - **Continuous & Low Dimensional representations** → robustness
  - **Online training** → walks naturally partition the graph
  - **Scalable implementation** → SkipGram & Hierarchical Softmax



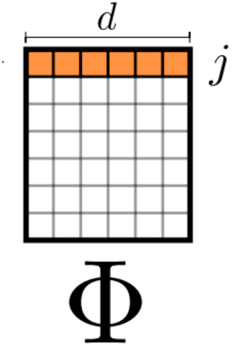
(a) Input: Karate Graph (b) Output: Representation

Limitations

- Representations themselves left unexplored.

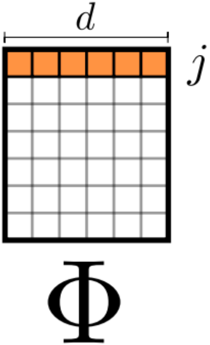
Limitations

- Representations themselves left unexplored.
- Number of parameters $\propto$ number of nodes in the graph.



Limitations

- Representations themselves left unexplored.
- Number of parameters $\propto$ number of nodes in the graph.



- Similarity is only defined on a single graph.