图表示学习

Zhou Shen

School of Computer Science, Wuhan University

October 20, 2021

Outline

Word2Vec

One-Hot Encoding

1-of-N Encoding

apple = [1 0 0 0 0]

bag = [0 1 0 0 0]

cat = [0 0 1 0 0]

dog = [0 0 0 1 0]

elephant = [0 0 0 0 1]

Word2Vec

One-Word Context

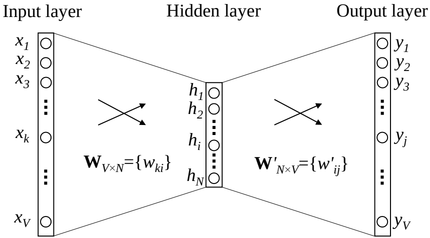


Figure 1: A simple CBOW model with only one word in the context

$$\mathbf{h} = \mathbf{W}^T\mathbf{x} = \mathbf{W}^T_{(k,\cdot)} := \mathbf{v}^T_{w_I}, u_j = \mathbf{v}'_{w_j} h \tag{1}$$

$$p\left(w_j \mid w_I\right) = y_j = \frac{\exp\left(u_j\right)}{\sum_{j'=1}^{V} \exp\left(u_{j'}\right)} \tag{2}$$

Word2Vec

Objective Function

$$\begin{aligned} \max p\left(w_O \mid w_I\right) &= \max y_{j^*} \\ &= \max \log y_{j^*} \\ &= u_{j^*} - \log \sum_{j'=1}^{V} \exp\left(u_{j'}\right) := -E \end{aligned} \tag{3}$$

常用变形

$\min E = -u_{j^*} + \log \sum_{j'=1}^{V} \exp\left(u_{j'}\right)$

Word2Vec

CBOW

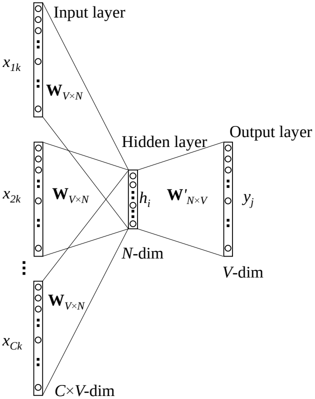


Figure 2: Continuous bag-of-word model

$$\mathbf{h} = \frac{1}{C}(\mathbf{v}_{w_1} + \mathbf{v}_{w_2} + \cdots + \mathbf{v}_{w_C})^T$$

$$E = -\log p\left(w_O \mid w_{I,1}, \cdots, w_{I,C}\right)$$

$$= -u_{j^*} + \log \sum_{j'=1}^{V} \exp\left(u_{j'}\right)$$

Word2Vec

Skip-Gram

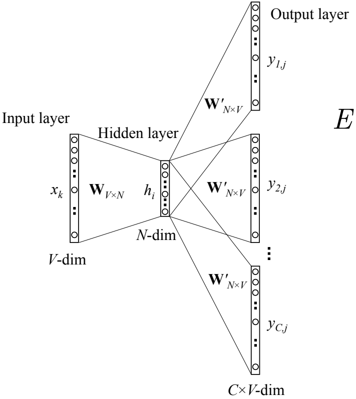
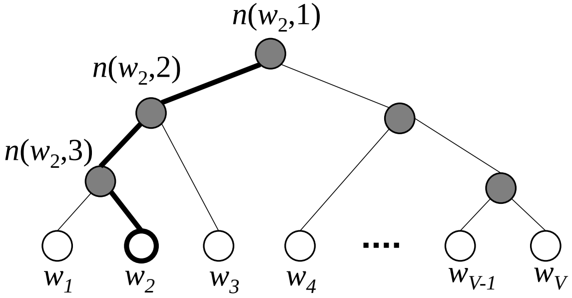


Figure 3: The skip-gram model.

$$
\begin{aligned}
E &= -\log p\left(w_{O,1}, w_{O,2}, \cdots, w_{O,C} \mid w_I\right) \\
&= -\log \prod_{c=1}^{C} \frac{\exp\left(u_{c,j_c^*}\right)}{\sum_{j'=1}^{V} \exp\left(u_{j'}\right)} \\
&= -\sum_{c=1}^{C} u_{j_c^*} + C \cdot \log \sum_{j'=1}^{V} \exp\left(u_{j'}\right)
\end{aligned}
$$

Word2Vec

Hierarchical Softmax



$$p\left(w = w_O\right) = \prod_{j=1}^{L(w)-1} \sigma\left(\left[n(w, j+1) = \mathrm{ch}(n(w, j))\right] \cdot \mathbf{v}'_{n(w,j)}{}^T \mathbf{h}\right)$$

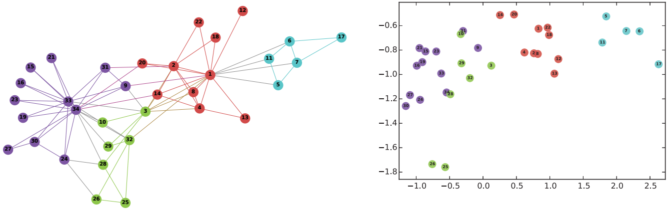
1. 词频计算

$$weight(w_i) = \frac{count(w_i)^{0.75}}{\sum_j^V count(w_j)^{0.75}}$$

2. 目标函数

$$E = -\log \sigma \left(\mathbf{v}'_{w_O}{}^T \mathbf{h}\right) - \sum_{w_j \in \mathcal{W}_{\text{neg}}} \log \sigma \left(-\mathbf{v}'_{w_j}{}^T \mathbf{h}\right)$$

DeepWalk

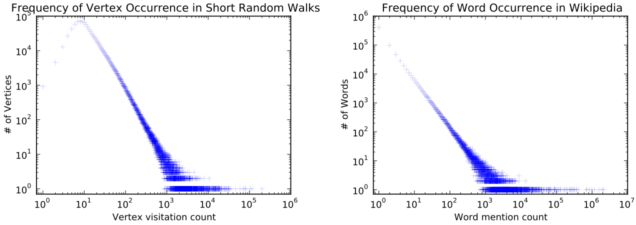


(a) Input: Karate Graph (b) Output: Representation

Definition

Problem Definition Let $G = (V, E)$, where V are the members of the network, and E be its edgs. Our goal is to learn $X_E \in \mathcal{R}^{|V| \times d}$, where d is number of latent dimensions.

DeepWalk



(a) YouTube Social Graph (b) Wikipedia Article Text

Connection

The power-law distribution of vertices appearing in the short random walks follows a power-law, much like the distribution of words in natural language.

DeepWalk

1. give a sequence of random walk from vertice i

$$\mathcal{W}_i^n = \{v_1, v_2, v_3, \cdots, v_n\} \tag{4}$$

2. maximize the likehood

$$\Pr\left(v_i \mid (\Phi(v_1), \Phi(v_2), \cdots, \Phi(v_{i-1}))\right) \tag{5}$$

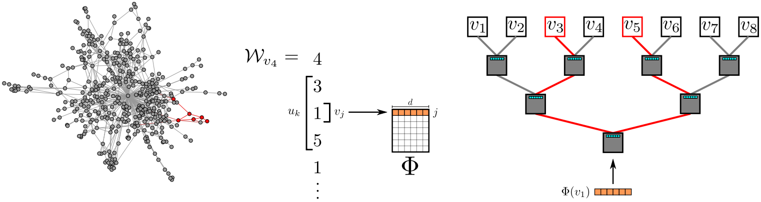
Mapping Function Φ

$\Phi : v \in V \longmapsto \mathcal{R}^{|V| \times d}$ where d is the number of dimensions

DeepWalk

Disadvantages and Optimizations

- As the walk length grows, computing this objective function becomes unfeasible.
- Restricts the order in which vertices appear



(a) Random walk generation. (b) Representation mapping. (c) Hierarchical Softmax.

DeepWalk
Algorithm

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 Φ ∈ ℝ^{|V|×d}
Initialization: Sample Φ from U^{|V|×d}
Build a binary Tree T from V
for i = 0 to γ do
    O = Shuffle(V)
    for each v_i ∈ O do
        W_{v_i} = RandomWalk(G, v_i, t)
        SkipGram(Φ, W_{v_i}, w)
    end for
end for
```

Corollary

$$\underset{\Phi}{\text{minimize}} - \log \Pr\left(\{v_{i-w}, \cdots, v_{i-1}, v_{i+1}, \cdots, v_{i+w}\} \mid \Phi(v_i)\right)$$

Node2Vec

Struc2Vec

LINE

SDNE