Feature and Representation Learning for Graphs

node2vec and graph2vec

Agam Goyal

REU 2022
University of Wisconsin - Madison

August 3, 2022

Table of Contents

Table of Contents

Skip-gram

In many NLP tasks, better results can be achieved if algorithms can learn to group similar words. **Skip-gram** (Mikolov et al. [2013]) is a model that is used to find word representations useful for predicting the surrounding words in a sentence or document.

Skip-gram

In many NLP tasks, better results can be achieved if algorithms can learn to group similar words. **Skip-gram** (Mikolov et al. [2013]) is a model that is used to find word representations useful for predicting the surrounding words in a sentence or document.

Given a sequence of training words: $w_1,\ w_2,\ w_3, \cdots,\ w_T$, we want to maximize the average log-probability

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} log\ Pr(w_{t+j} \mid w_t)$$

where c is the size of training context for the center word w_t. This problem is solved using a softmax function formulation.

Skip-gram

In many NLP tasks, better results can be achieved if algorithms can learn to group similar words. **Skip-gram** (Mikolov et al. [2013]) is a model that is used to find word representations useful for predicting the surrounding words in a sentence or document.

Given a sequence of training words: $w_1,\ w_2,\ w_3, \cdots,\ w_T$, we want to maximize the average log-probability

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} log\ Pr(w_{t+j} \mid w_t)$$

where c is the size of training context for the center word w_t. This problem is solved using a softmax function formulation.

There exist efficient estimation strategies for softmax function like the heirarchical softmax and the negative sampling methods.

Table of Contents

Introduction to node2vec

node2vec (Grover and Leskovec [2016]) is a *semi-supervised*[1] framework for learning continuous feature representations for nodes in networks, having the following features:

[1] uses small amount of labeled data combined with large amount of unlabeled data

Introduction to node2vec

node2vec (Grover and Leskovec [2016]) is a *semi-supervised*[1] framework for learning continuous feature representations for nodes in networks, having the following features:

- **Objective:** Learn a mapping of nodes to a low dimensional feature space that maximizes the likelihood of preserving network neighborhoods of nodes.

[1] uses small amount of labeled data combined with large amount of unlabeled data

Introduction to node2vec

node2vec (Grover and Leskovec [2016]) is a *semi-supervised*[1] framework for learning continuous feature representations for nodes in networks, having the following features:

- **Objective:** Learn a mapping of nodes to a low dimensional feature space that maximizes the likelihood of preserving network neighborhoods of nodes.
- **Flexibility:** The notion of a network neighborhood is flexible by using a biased random walk procedure using different sampling techniques like BFS and DFS

[1] uses small amount of labeled data combined with large amount of unlabeled data

Introduction to node2vec

node2vec (Grover and Leskovec [2016]) is a *semi-supervised*[1] framework for learning continuous feature representations for nodes in networks, having the following features:

- **Objective:** Learn a mapping of nodes to a low dimensional feature space that maximizes the likelihood of preserving network neighborhoods of nodes.
- **Flexibility:** The notion of a network neighborhood is flexible by using a biased random walk procedure using different sampling techniques like BFS and DFS
- **Use cases:** node2vec performs better than the state-of-the-art algorithms on the following tasks:
  - Multi-label Classification of nodes in a network
  - Link prediction between nodes of a network

[1] uses small amount of labeled data combined with large amount of unlabeled data

Framework for node2vec

Let $G = (V, E)$ be any (un)directed and (un)weighted network and let $f : V \to \mathbb{R}^d$ be the mapping function from nodes to the feature representation of dimension d.

[2] using conditional probability and negative sampling

Framework for node2vec

Let $G = (V, E)$ be any (un)directed and (un)weighted network and let $f : V \to \mathbb{R}^d$ be the mapping function from nodes to the feature representation of dimension d.

For every node $u \in V$, we define a *network neighborhood* $N_S(u) \subset V$ that is generated through a sampling strategy S - biased random walk.

[2] using conditional probability and negative sampling

Framework for node2vec

Let $G = (V, E)$ be any (un)directed and (un)weighted network and let $f : V \to \mathbb{R}^d$ be the mapping function from nodes to the feature representation of dimension d.

For every node $u \in V$, we define a *network neighborhood* $N_S(u) \subset V$ that is generated through a sampling strategy S - biased random walk.

Using the idea of the *skip-gram* architecture, we solve a **modified version**[2] of the following optimization problem using Stochastic Gradient Descent, which maximizes the log-likelihood of observing a network neighborhood $N_S(u)$ for node u conditioned on its feature representation:

$$\max_f \sum_{u \in V} log\ Pr(N_S(u) \mid f(u))$$

[2] using conditional probability and negative sampling

Table of Contents

Introduction to graph2vec

graph2vec (Narayanan et al. [2017]) is an *unsupervised* framework for learning distributed representations of arbitrary sized graphs, having the following features:

[3] evaluate the similarity (using kernel function) between a pair of graphs G and G' by recursively decomposing them into substructures

Introduction to graph2vec

graph2vec (Narayanan et al. [2017]) is an *unsupervised* framework for learning distributed representations of arbitrary sized graphs, having the following features:

- **Objective:** Learn low dimensional graph embeddings/distributed representations without the need of class labels primarily for graph classification and clustering

[3] evaluate the similarity (using kernel function) between a pair of graphs G and G' by recursively decomposing them into substructures

Introduction to graph2vec

graph2vec (Narayanan et al. [2017]) is an *unsupervised* framework for learning distributed representations of arbitrary sized graphs, having the following features:

- **Objective:** Learn low dimensional graph embeddings/distributed representations without the need of class labels primarily for graph classification and clustering
- **Novelty:** Two features of graph2vec makes it stand out as opposed to existing frameworks like *Graph Kernels*[3]:
  - Allows the use of ML algorithms because of a generic representation
  - Promotes data-driven embeddings as opposed to handcrafted decompositions (like Hamiltonian path in our case)

[3] evaluate the similarity (using kernel function) between a pair of graphs G and G' by recursively decomposing them into substructures

Introduction to graph2vec

graph2vec (Narayanan et al. [2017]) is an *unsupervised* framework for learning distributed representations of arbitrary sized graphs, having the following features:

- **Objective:** Learn low dimensional graph embeddings/distributed representations without the need of class labels primarily for graph classification and clustering
- **Novelty:** Two features of graph2vec makes it stand out as opposed to existing frameworks like *Graph Kernels*[3]:
  - Allows the use of ML algorithms because of a generic representation
  - Promotes data-driven embeddings as opposed to handcrafted decompositions (like Hamiltonian path in our case)
- **Structural equivalence:** graph2vec samples non-linear substructures in form of rooted subgraphs, which helps in yielding similar embeddings for structurally similar graphs.

[3] evaluate the similarity (using kernel function) between a pair of graphs G and G' by recursively decomposing them into substructures

Framework for graph2vec

Let $\mathbb{G} = \{G_1, G_2, \cdots\}$ be a set of graphs and δ a positive integer, we want to learn $\delta-$dimensional distributed representations for every graph $G_i \in \mathbb{G}$. The matrix representation of embeddings is denoted as $\Phi \in \mathbb{R}^{|G| \times \delta}$.

Framework for graph2vec

Let $\mathbb{G} = \{G_1, G_2, \cdots\}$ be a set of graphs and δ a positive integer, we want to learn $\delta-$dimensional distributed representations for every graph $G_i \in \mathbb{G}$. The matrix representation of embeddings is denoted as $\Phi \in \mathbb{R}^{|G| \times \delta}$.

For each node $n \in N_i$ in graph $G_i \in \mathbb{G}$, get a rooted subgraph using the Weisfeiler-Lehman (WL) kernel. Rooted subgraphs are used because compared to lower order substructures like nodes, they capture graph features better, and are non-linear leading to structural equivalence.

Framework for graph2vec

Let $\mathbb{G} = \{G_1, G_2, \cdots\}$ be a set of graphs and δ a positive integer, we want to learn $\delta-$dimensional distributed representations for every graph $G_i \in \mathbb{G}$. The matrix representation of embeddings is denoted as $\Phi \in \mathbb{R}^{|G| \times \delta}$.

For each node $n \in N_i$ in graph $G_i \in \mathbb{G}$, get a rooted subgraph using the Weisfeiler-Lehman (WL) kernel. Rooted subgraphs are used because compared to lower order substructures like nodes, they capture graph features better, and are non-linear leading to structural equivalence.

Again using *skip-gram* architecture, we solve the following optimization problem, again using Stochastic Gradient Descent, which maximizes the log-likelihood of observing a rooted subgraph neighborhood $sg_n^{(d)}$) for node n with degree d, conditioned on its embedded representation:

$$\max_{\Phi} \sum_{n \in N_i} log\ Pr(sg_n^{(d)} \mid \Phi(\mathbb{G}))$$

References

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. graph2vec: Learning distributed representations of graphs. *arXiv preprint arXiv:1707.05005*, 2017.