웹 그래프 상에서의 Node Embedding 알고리즘 성능 분석

정연성[1] 황지영[2‡]
[1]성균관대학교 전자전기컴퓨터공학과
[2]성균관대학교 컴퓨터공학과
{ys.jung, jjwhang}@skku.edu

Performance Analysis of Node Embedding Algorithms on Web Graphs

Yeonsung Jung[1] Joyce Jiyoung Whang[2‡]
[1]Department of Electronic, Electrical & Computer Engineering, Sungkyunkwan University
[2]Department of Computer Science and Engineering, Sungkyunkwan University

요 약

노드 임베딩(Node Embedding)은 그래프 형태의 데이터에서 지도 학습(supervised learning)을 수행할 때 매우 효과적인 표현(representation) 기법으로 대두되고 있다. 이 기법은 각 노드의 이웃 노드(neighborhood node)의 정의 및 선택하는 방법과 각 노드들의 연결성(connectivity) 및 구조적인 역할(structural role) 등을 표현하는 방법이 중요하다. 최근 연구된 알고리즘들은 각각의 우선순위에 따라 타겟으로 하는 특성이 존재하기 때문에, 데이터의 형태에 따라 성능이 많이 달라지는 현상을 보일 것으로 추론된다. 본 논문에서는 실제 검색 엔진의 웹 그래프 데이터 상에서 노드 임베딩 알고리즘들이 기존의 랭킹 알고리즘보다 노드 분류(node classification) 측면에서 성능적 우위를 점하는지 실험하고 비교·분석하였다.

1. 서 론

노드 임베딩이란 그래프의 노드들의 표현을 학습(representation learning)하는 것을 의미한다. 다시 말해 노드의 특성을 추출하여 d-차원의 실수 벡터(real number vector)로 대응(mapping)시켜 기계학습에 효율적으로 사용할 수 있도록 치환해 주는 것을 말한다. 이를 그래프 구조의 관점에서 보면 임의의 그래프 G의 n개의 노드들이 다른 임의의 그래프 G'의 n개의 노드들과 일대일 대응이 되도록 하고, G의 간선 (i, j)들이 G'에서 i와 j에 대응되는 노드 사이의 경로(path)로 대응 되도록 하는 것을 의미한다. 이때 그래프 G에서의 노드간의 연결성(connectivity), 구조적 동일성(structural equivalence) 등의 특성들을 고려하여 관계가 깊은 노드간의 경로가 짧아지도록 G'에 대응시킨다.

노드의 특성 표현(feature representation)을 만들기 위해 이전에는 전문가의 해당 분야의 배경지식(domain knowledge)에 의존하거나, 주성분 분석(principal component analysis)과 같이 고유값 분해(eigen decomposition)를 사용하였다. 먼저, 전문지식을 활용한 수작업(hand-engineering)은 특정한 데이터에 사용할 수는 있으나 여러 데이터에 일반화(generalization)할 수 없다는 단점이 있다. 그리고 주성분 분석과 같은 기법들은 고유값 분해를 사용하기 때문에 복잡도가 높아 거대 네트워크에서 사용하기 비효율적이고 예측(prediction)측면에서 성능이 좋지 않다는 단점이 있다. 이후 등장한 노드 임베딩 알고리즘들은 텍스트 마이닝(text mining) 알고리즘인 word2vec[1]의 이론에 기반하여 앞서 말한 알고리즘들의 단점을 보완하며 궁극적으로 확장성(scalable) 있는 자동 특성 추출을 목표로 한다.

원 그래프의 특성을 잘 반영하기 위해서는 연결성(connectivity)과 구조적 역할(structural role)을 고려해야 한다. 연결성은 유사한 노드들일수록 짧은 경로를 유지하며 가까이 존재하는 동종 선호(homophily) 특성을 의미하는 지표이다. 따라서 각 노드간의 관계에서 짧은 경로를 유지하고 있거나, 구조적인 역할이 비슷하면 두 노드의 유사도(similarity)가 높다고 할 수 있다. 이외에도 그래프에 존재하는 다양한 패턴을 노드간의 유사도를 판단하는 기준으로 삼을 수 있다.

최근 연구된 노드 임베딩 알고리즘들은[2, 3] 각각의 우선순위에 따라 유사도를 판단하는 기준을 선택하고 있다. 그렇기 때문에 기준에 부합하는 형태의 데이터에만 좋은 성능을 보일 것을 추론된다.

본 논문에서는 검색 엔진의 실제 웹 그래프 데이터에서 최근 노드 임베딩 알고리즘들의 성능과 기존의 랭킹 알고리즘의 성능을 실험하여 비교·분석하고, 이를 종합하여 웹 그래프 데이터에서 고려해야 할 점을 제안한다.

2. 관련 연구

노드 임베딩 연구는 텍스트 마이닝(text mining) 연구인 word2vec에서 비롯되었다. word2vec 모델은 각 단어가 함께 corpus

‡ 교신저자 (Corresponding author)
이 논문은 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2016R1D1A1B03934766, NRF-2010-0020210). 또한, 과학기술정보통신부 및 정보통신기술진흥센터의 SW중심대학지원사업의 연구결과로 수행되었음(2015-0-00914). 본 연구에서 사용된 모든 데이터는 네이버(주)에서 제공받았음

에 존재할 확률을 목적 함수(objective function)로 두고 stochastic gradient descent method를 통해서 최적화(optimization)한다. 각 단어들을 d-차원의 벡터로 표현하여 각 단어의 연관성에 대해서 수리적으로 계산할 수 있다. 이후 한 단어가 주어졌을 때, 학습 단계에서 가까이 존재 했던 단어들을 기반으로 해당 단어 주변에 등장할 단어들을 유추하는 방식으로 작동한다.

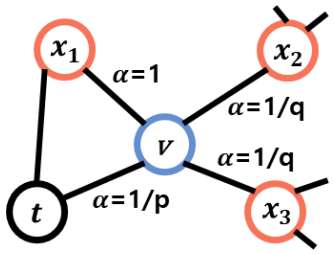
word2vec의 개념을 그래프형 데이터에 적용한 노드 임베딩 알고리즘인 node2vec, struc2vec이 있다. node2vec은 파라미터를 통해, 노드간의 연결성과 구조적 역할에 맞게 이웃노드를 선택하고, struc2vec은 구조적 역할에 초점을 둔 알고리즘으로 노드간의 구조적 거리를 다차원적으로 계산하여 이웃노드를 선택한다.

3. 노드 임베딩 알고리즘

3.1 node2vec

node2vec은 기존의 1-hop 거리의 노드들을 의미하는 이웃 노드(Neighborhood node) 개념을 2nd order 랜덤 워크로 접근 가능한 노드들로 확장하여 정의하였다. 그리고 2nd order 랜덤워크를 수행할 때, 파라미터 p, q를 정의하여 이웃노드를 유연하게 선택할 수 있게 하였다. 랜덤워크가 간선 (t,v)를 지나 노드 v에 도착한 상황일 때, 다음 노드(x_1, x_2, x_3)를 선택하기 위해 파라미터 p, q를 활용하여 전이확률(transition probability) $\alpha_{pq}(t,x)$를 계산하는데 다음과 같다.

$$\alpha_{pq}(t,x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases}$$



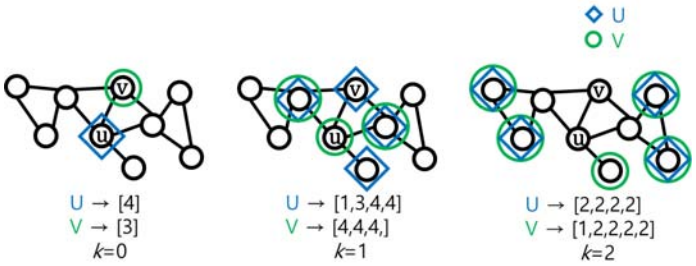
<그림 1 전이확률>

d_{tx}는 노드 t와 x간의 최단 거리를 의미한다. 파라미터 p는 기존의 노드로 돌아올 확률을 조정하는 파라미터 이고, q는 랜덤워크가 뻗어나가는 정도를 조정하는 파라미터다. 이 두 파라미터를 조정하여, 각 노드의 이웃을 연결성에 집중해서 선택할 것인지, 구조적 역할에 집중해서 선택할 것인지를 데이터 형태에 맞게 조정할 수 있다. 하지만 매번 각 데이터에 맞게 파라미터를 조절하는 과정이 필요하고, p와 q는 상충관계가 존재하기 때문에 동시에 연결성과 구조적 역할을 충분히 고려하지 못한다는 단점이 존재한다.

3.2 struc2vec

struc2vec은 구조적 역할에 집중한 알고리즘이다.

먼저 노드에서 k-hop 떨어진 노드들의 차수들을 정렬하여 <그림 2>와 같이 각각 벡터로 표현한다. 해당 벡터들을 이용하여 전체 노드 쌍의 거리를 구하고, k-hop 이하의 거리를 모두 합하여 구조적 거리(structural distance)를 정의한다. 각



<그림 2 k-hop 이웃노드의 차수>

hop별로 하나의 계층(layer) 그래프를 생성하고, 노드간의 간선 가중치(edge weight)는 두 노드간의 k-hop 구조적 거리로 한다. 인접 계층에 존재하는 같은 노드를 연결하는 간선이 존재하기 때문에 연결된 다층 그래프(multi-layer graph)가 된다. 이후 랜덤 워크를 통하여 계층을 넘나들며 이웃 노드를 선택한 뒤, skip-gram 모델을 채택하여 학습한다. 구조적 역할에만 집중한 알고리즘이기 때문에 노드간의 연결성, 즉 동종 선호 특성을 충분히 반영하지 못할 것으로 추론된다.

4. 실험 결과

4.1 실험 데이터

(주)네이버에서 웹 그래프 데이터를 제공받아 실험하였다. 기본적인 데이터 요약 정보는 <표 1>과 같고, 노드들의 정상/스팸 분류 정보는 <표 2>와 같다.

<표 1 데이터 요약>

노드 수	간선 수	연결 요소 수	GCC 크기
856,404	3,955,939	16,129	669,619

* GCC (giant connected component)

<표 2 노드 분류>

Page		
Normal	Spam	Undefined
797,718 (93.15%)	47,301 (5.52%)	11,385 (1.33%)
(93.15%)	(5.52%)	(1.33%)

4.2 노드 분류 (node classification)

4.2.1 ATR 알고리즘

Anti-TrustRank(ATR) 알고리즘[4, 5]은 널리 알려진 링크 기반 스팸 탐지 알고리즘으로 스팸 시드 노드(spam seed)들로부터 Anti-TrustRank 점수를 전파해 나가는 알고리즘이다. Personalized PageRank[6] 알고리즘의 원리와 유사하

며, 높은 Anti-TrustRank 점수를 가진 노드는 스팸일 확률이 높다는 것을 의미한다.

4.2.2 Parameter 설정

node2vec의 경우, 연결성과 구조적 역할을 고려할 수 있게 'p=0.5, q=1' 과 'p=2, q=1' 두 가지로 설정하여 실험하였고, struc2vec의 경우 기본 값인 'p=1, q=1' 로 설정하였다. 또한 본 실험에서 사용한 데이터의 크기가 해당 논문에서의 실험에 비해 85~850배 큰 것을 고려하여, 임베딩 결과 값의 차원 파라미터(dimension)를 효율성(efficiency) 측면에서 128에서 50으로 하향 설정하였다.

4.2.3. 랜덤 포레스트 (Random Forest)

대표적인 분류 모델인 랜덤 포레스트를 사용하여 node2vec, struc2vec과 Anti-TrustRank의 성능을 비교하였다. 서로 다른 방식으로 작동하는 알고리즘들을 공평하게 비교하고자 다음과 같은 성능평가 방법을 사용하였다.

먼저, 페이지들을 Anti-TrustRank의 경우 ATR score가 높은 순서로, node2vec과 struc2vec의 경우 classification 결과 Spam class에 속할 확률이 높은 순서로 내림차순 정렬한다. 그리고 false positive 비율의 upper bound를 5%로 설정하고, 해당 상한선을 초과하지 않는 상위 n개의 페이지를 선택한다. 해당 상위 n개의 페이지에 대하여 스팸 페이지의 수와 정확도(accuracy)를 기준으로 각 방법론의 성능을 평가한다. 아래 <표 3>, <표 4>, <표 5>는 ATR의 Seed Spam의 사이즈와 Classification의 training set의 비율을 달리하여 실험한 결과다.

<표 3. 20% Training >

		training = 20%		
	ATR	node2vec p=0.5 q=1	node2vec p=2 q=1	struc2vec p=1 q=1
No. of Seed Spam	6,586	9,460	9,460	9,460
No. of Spams	19,758	15,039	12,131	1,157
Accuracy	97.51%	97.66%	97.5%	95.07%

<표 4. 30% Training>

		training = 30%		
	ATR	node2vec p=0.5 q=1	node2vec p=2 q=1	struc2vec p=1 q=1
No. of Seed Spam	17,805	14,190	14,190	14,190
No. of Spams	33,088	22,616	22,618	1,747
Accuracy	97.67%	95.63%	95.81%	96.73%

<표 5. 50% Training>

		training = 50%		
	ATR	node2vec p=0.5 q=1	node2vec p=2 q=1	struc2vec p=1 q=1
No. of Seed Spam	24,964	23,650	23,650	23,650
No. of Spams	38,282	28,203	28,004	15,064
Accuracy	97.52%	97.25%	97.53%	95.5%

Anti-TrustRank 알고리즘은 수행과정에서 Normal Seed를 필요로 하지 않는다. 또한 실험결과에서 볼 수 있듯이, Anti-TrustRank 알고리즘은 임베딩 알고리즘과 비슷하거나 더 적은 Seed Spam을 사용함에도 불구하고 더 많은 스팸 페이지를 정확도 있게 탐지한다. 결론적으로 스팸 탐지의 관점에서 node2vec과 struc2vec은 효율적이지 않다고 볼 수 있다.

5. 결론 및 향후 연구

본 논문에서는 실제 검색 엔진의 웹 그래프 데이터에서 노드 임베딩 알고리즘인 node2vec, struc2vec과 Anti-TrustRank 알고리즘의 노드분류 성능을 실험하고 비교·분석하였다. Anti-TrustRank의 경우 학습을 위한 정상 페이지를 사용하지 않고, 더 적은 스팸 시드로 더 많은 스팸 페이지를 정확도 있게 탐지한다는 점에서 node2vec과 struc2vec에 비해 더 좋은 성능을 보인다고 할 수 있다. 향후 실제 웹 그래프 데이터에 적용 할 수 있는 노드 임베딩 알고리즘을 개발하고, 다양한 노드분류 기법들을 적용하여 보다 정밀한 성능평가를 실행할 계획이다.

참 고 문 헌

[1] T. Mikolov et al. "Distributed Representations of Words and Phrases and Their Compositionality," *NIPS*, 2013.

[2] Aditya et al., "node2vec : Scalable Feature Learning for Networks," *KDD*, 2016

[3] Ribeiro et al., "struc2vec : Learning Node Representations from Structural Identity," *KDD*, 2017

[4] V. Krishnan., "Web Spam Detection with Anti-Trust Rank," *ACM SIGIR Workshop AIRweb*, 2006.

[5] J.J. Whang et al., "Fast Asynchronous Anti-TrustRank for Web Spam Detection," *WSDM Worskshop MIS2*, 2018

[6] G. Jeh et al., "Scaling Personalized Web Search.," *WWW, 2002.*