Competitive analysis with graph embedding on patent networks

Wang, Yunli; Richard, Rene; Mcdonald, Daniel

Publisher's version / Version de l'éditeur:
https://doi.org/10.1109/CBI49978.2020.00009
2020 IEEE 22nd Conference on Business Informatics (CBI), pp. 10-19, 2020-07-15

Competitive Analysis with Graph Embedding on Patent Networks

Yunli Wang[1], René Richard[2] and Daniel McDonald[2]

***Abstract*— Advanced competitive analysis is increasingly becoming important in business analytics. A key component of strategic research is to collect and review information from multiple unstructured sources to identify major competitors and their technology development trends. Topic modeling techniques such as Latent Dirichlet Allocation (LDA) have been applied to competitive analysis, which mainly use the semantic similarities between documents to infer competitive relationships. In this study, we propose using graph embedding methods to learn the implicit competitive relationships between firms. Using patent networks with patents and organizations as nodes, we learn the embeddings of nodes, and then cluster organizations into groups. Organizations within the same groups are considered competitors. We applied three graph embedding methods: node2vec, metapath2vec, and GraphSAGE to learn node embeddings. Two of these methods use the structural information in patent networks: node2vec for homogeneous networks and metapath2vec for heterogeneous networks. While, GraphSAGE uses both the structure and content information in the patent network. The results are compared with a baseline author-topic modeling method. The graph embedding methods outperform the author-topic modeling approach in learning the competitive relationships. A case study, examining the evolution of competitors over multiple years, shows the graph embedding method learns meaningful node embeddings.**

I. Introduction

In the modern world, business competition is becoming increasingly fierce for limited resources and market share in all industries. Identifying competitors is critical for companies to make decisions and strategic plans. Competitive analysis involves identifying competitors and technology trends of competitors by collecting their business data and information.

Patents can be used as an important source for competitive analysis since they protect a firm's inventions from inception to monetisation and are often a constituent part of a firm's current and future industrial directions. Patent data has been used for mining competitive relationships using topic modeling methods [17], [19]. A patent is represented as a multinomial distribution of words in a topic model, and an organization is considered as a mixture of topic models from all patents owned by this organization. The competitive relationships between organizations are learned from the topic models of organizations. Therefore, topic models mainly utilize the semantic similarities between documents to infer the competitive relationships. This approach does not make use of graph-structured data in patent networks.

*This work was supported by National Research Council Canada

[1] Digital Technologies Research Centre, National Research Council Canada, 1200 Montreal Rd, Ottawa, ON, K1A 0R6, Canada Yunli.Wang@nrc-cnrc.gc.ca

[2] National Research Council Canada, Fredericton NB, E3B 9W4, Canada Rene.Richard@nrc-cnrc.gc.ca, Daniel.McDonald@nrc-cnrc.gc.ca

In this study, we propose to use graph embedding methods on competitive analysis. Graph embedding, also called representation learning on graph, learns a mapping that embeds nodes, or entire sub-graphs, as points in a low-dimensional vector space. Graph embedding methods are usually used for node classification[10], node clustering [3], link prediction etc, and has been applied in many applications such as recommender systems [20]. To our knowledge, no other studies have used graph embedding for competitive analysis using patent data. We formulate the competitive analysis as a node clustering problem in patent networks. Patent networks are heterogeneous networks composed of different types of nodes and links. We applied three graph embedding methods: node2vec [5] for homogeneous networks, metapath2vec [3] for heterogeneous networks, and GraphSAGE [6] for incorporating the node features in our study. node2vec and metapath2vec use only the graph structure, while GraphSAGE uses both the graph structure and node features in learning the node embeddings. Using multi-year data, our approach is able to capture the dynamic nature of shifting competitor relationships over time. This results in the unique capability of investigating changing competitor relationships as they evolve.

II. Related work

The challenge in competitive analysis is to extract useful information from a large amount of unstructured documents in an unsupervised manner. Document classification, document summarization, and clustering can be used to identify structures in larger text collections. Topic modeling is a popular document clustering approach. In previous work on competitive analysis, topic models and their extensions have been used to discover competitor relationships. Two basic statistical topic models are Probabilistic Latent Semantic Analysis (PLSA) [8] and Latent Dirichlet Allocation (LDA) [1]. As an extension of LDA, Tang et al. proposed topic-driven patent analysis and mining methods [17], in which both global competitors and topic-level competitors were learned. Yang et al. proposed a Topical Factor Graph Model (TFGM) for mining competitive relationships in heterogeneous networks, combining the PLSA and factor graph [19]. Wang et al. used an LDA model to extract competitors from patent data and validated the effectiveness of the model on next generation telecommunication technology [18].

In recent years, graph embedding (network embedding) has been widely used in node classification [10], link prediction [15], event detection [12], recommender systems [20], and on-line learning [11], but few studies have used it for competitive analysis. Many representation learning methods

have been proposed such as DeepWalk [13], LINE [16], and node2vec [5]. The goal of graph embedding methods is to optimize the mapping from nodes in graphs to low dimensional vector space, so that the similarities in the embedding space approximates the similarities in the orginal graph. Some studies have used graph embedding methods for patent classification. Fu et al. proposed HIN2vec, a method for generating vector representations of nodes in heterogeneous networks, and applied the method on patent classification and link prediction [4].

The majority of graph embedding methods use a "shallow" encoder-decoder architecture. Recently, graph neural networks (GNN) adopted a deep encoder for generating node embeddings from local neighbors. GNNs use neural networks such as convolutional networks for neighborhood aggregation functions. They differ in how they aggregate information from a local neighborhood. Graph convolutional networks (GCN) share the parameters between one node and its neighbors and also normalize across all neighbors in aggregation function [10]. GCNs have been applied to semi-supervised node classification and link prediction. GraphSAGE was designed for graph embeddings for large scale networks and can be trained for node classification using a supervised or unsupervised approach.

In this study, we build a heterogeneous network consisting of different types of nodes and links. Patents and organizations are represented by vertices. The patent-patent and patent-organization relationships form the links. Organizations and patents have a hierarchical relationship in this heterogeneous network. We use graph embedding methods to learn the implicit competitive relationships between organizations. Our approach is different from [4], which used different types of links in patent networks to learn patent embeddings for patent classification. Our study perform node clustering by learning organization embeddings. Additionally, we compare the graph embedding methods with topic models, which learn the implicit topics and organization topic distributions.

III. Method

We adopted three graph embedding methods: node2vec, metapath2vec and GraphSAGE to learn the implicit competitive relationships on patent networks. The patent network includes nodes representing patents, organizations, and patent groups expressed as patent classification codes. The network nodes are linked by patent-organization and patent-patent group associations. In addition, we used an author-topic model as a baseline in our study since it captures the semantic similarities of patents in patent networks. As far as we know, the author-topic model has not been used for competitive analysis. This section initially introduces the author-topic model, then describe the three graph embedding approaches used in our competitive analysis approach.

A. Author-topic model

The author-topic model is an extension of the Latent Dirichlet Allocation (LDA) model [1]. In the LDA model, a distribution over topics is sampled for each document from a Dirichlet distribution. For each word in the document, a single topic is chosen according to this distribution. Then, each topic is sampled from a multinomial distribution over words specific to the sampled topic. The author-topic model uses a topic-based representation to model both the content of documents and the interests of authors [14]. It extends LDA to author modeling by allowing the mixture weights for different topics to be determined by the authors of the document. For each word in the document, an author is chosen uniformly at random. Then, as in the topic model, a topic is chosen from a distribution over topics specific to that author, and the word is generated from the chosen topic.

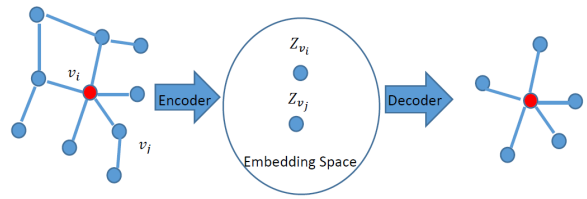


Fig. 1: The architecture of graph embedding methods

We trained a author-topic model on patent data by treating patent abstracts as documents and patent-owner organizations as authors of the documents. The method uses patent abstracts in addition to structural information that exists between organizations and patents to represent the collection of documents. Although the author-topic model ignores some structural components such as links between patents and patent groups, it does utilize other structural information as well as semantic information. In our study, the author-topic model simultaneously models the content of patents and the business interests of organizations. It is a statistical model for unsupervised learning of the organization topic distributions. From the organization topic distributions, we calculate the Hellinger distances between organizations and use *k-medoids* to cluster organizations. The Hellinger distance between two distributions p and q is represented as: $H(p,q) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{k}(\sqrt{p_i}-\sqrt{q_i})^2}$, and the similarity between p and q is calculated as: $S(p,q) = \frac{1}{1+H(p,q)}$. Organizations within the same clusters are considered to be competitors.

B. Graph Embedding Approaches

In a network, the goal of graph embedding is to optimize the mapping of nodes to low-dimensional vectors where geometric relationships in the learned embeddings space reflect the structure of the original groups. The input of a graph embedding algorithm is an undirected graph $G = (V, E, X)$, where V are sets of nodes, E are sets of edges, and node attributes are represented as $X \in R^{m\times|V|}$. Node embedding approaches map nodes v_i to low dimensional vectors $Z_{v_i} \in R^d$. Most these methods use an encoder-decoder framework, in which the encoder maps nodes to vector embeddings, and the decoder decodes a graph proximity measure from node embeddings (Figure 1).

1) node2vec: The node2vec method treats the input network as a homogeneous network and uses a second-order proximity. The first-order proximity is based on direct neighbors, while the second-order keeps the neighborhood structure of the nodes in a graph. node2vec samples neighbors using a random walk generator, which estimates the similarities between nodes v_i and v_j in embedding space based on the probability that v_i and v_j co-occur on a random walk over the network. The probability of visiting node v_j on a random walk starting from node v_i using some random walk strategy is expressed as :

$$p(v_j|v_i) = \frac{exp(Z_{v_i}^T Z_{v_j})}{\sum_{n_i \in V} exp(Z_{v_i}^T Z_{n_i})} \tag{1}$$

Where Z_{v_i} and Z_{v_j} are embedding vectors of v_i and v_j, and Z_{n_i} represents the embeddings of neighbors n_i.

node2vec uses a second-order biased random walk generator, which samples nodes based on both the current node and previously visited nodes. node2vec finds the node context with a hybrid strategy of breath-first sampling (BFS) and depth-first sampling (DFS). BFS random walks are effective for capturing structural roles whereas DFS walks are capable of capturing community structures [7]. node2vec optimizes random walk embeddings based on a loss function (Equation 2), in which σ is the sigmoid function, k is the size of negative sampling, and n_i are sampled from P_V, which is a random distribution over all nodes. The first part of the loss function minimizes the predicted probability of v_i and v_j co-occurring on random walks and the second part reduces the number of pairs that need to be normalized in the loss function using the skip-gram model with negative sampling.

$$\mathcal{L} = \sum_{v \in V} -log(\sigma(Z_{v_i}^T Z_{v_j})) + \sum_{i=1}^{k} log(\sigma(Z_{v_i}^T Z_{n_i})), n_i \sim P_V \tag{2}$$

We formulate the competitive analysis problem as an unsupervised clustering problem. From patent networks, we generate organization embeddings using node2vec and then cluster organizations into groups based on organization embeddings. Organizations within the same group are considered to be competitors. node2vec only considers the network structure and does not consider different node and link types. Patent networks are typically heterogeneous and composed of different types of nodes or links. In addition, organizations in these networks have a hierarchical relationship with patents.

2) metapath2vec: In many real-world situations, networks are heterogeneous with multiple types of nodes or edges. metapath2vec was designed to learn embeddings in such networks [3]. Given a heterogeneous network, $G = (V, E)$, each node v and each link e is associated with $V \longrightarrow T_V$ and $E \longrightarrow T_E$. Where T_V and T_E denote the sets of objects and relation types, with $|T_V| + |T_E| > 2$.

In metapath2vec, meta-path-based random walks and heterogeneous negative sampling were used to learn the embedding of different types of nodes. A meta-path scheme $\mathcal{P}$ is represented as $V_1 \longrightarrow V_2 \longrightarrow ... \longrightarrow V_t ... \longrightarrow V_l$. For example, meta-path "APA" represents co-authorship on a paper (P) between two authors(A). The transition probability at step i is defined as:

$$p(v^{i+1}|v_t^i, \mathcal{P}) = \begin{array}{cc} \frac{1}{|N_{t+1}(v_t^i)|} & (v^{i+1}, v_t^i) \in E, v^{i+1} \in V_{t+1} \\ 0 & (v^{i+1}, v_t^i) \in E, v^{i+1} \notin V_{t+1} \\ 0 & (v^{i+1}, v_t^i) \notin E \end{array} \tag{3}$$

Where $N_{t+1}(v_t^i)$ denote V_{t+1} type of neighbors of node v_t^i. $v_t^i \in V_t$ and $v^{i+1} \in V_{t+1}$ mean walks need to follow the pre-defined meta-path $\mathcal{P}$. metapath2vec learns the node embeddings by maximizing the probability of having a heterogeneous context. After generating meta-path-based random walks, metapath2vec uses the heterogeneous negative sampling in which the softmax function is normalized with the node type of context c_t.

$$p(c_t|v_i) = \frac{exp(Z_{v_i}^T Z_{c_t})}{\sum_{n_t \in V_t} exp(Z_{v_i}^T Z_{n_t})} \tag{4}$$

Where V_t is the node set of type t in the network, Z_{v_i} is the embedding vector of v_i, and Z_{n_t} is the embedding vectors of node type t. metapath2vec generates sets of distributions P_t corresponding to type of neighbors. The loss function of metapath2vec is defined as:

$$\mathcal{L} = \sum_{v \in V} -log(\sigma(Z_{v_i}^T Z_{c_t})) + \sum_{i=1}^{k} log(\sigma(Z_{v_i}^T Z_{n_t})), n_t^k \sim P_t(n_t) \tag{5}$$

Where different type of nodes n_t are sampled from distributions $P_t(n_t)$.

Adopting metapath2vec on patent networks leverages the node types in generating meta-path-based random walks and learning node embeddings. The frequent meta-paths on patent networks are "PGP", which represents two patents (P) in the same patent group (G), and "POP", which indicates two patents belonging to the same organization (O). The meta-path for inferring competitive relationships is "OPGPO", which represents two organizations having patents in the same patent group.

3) GraphSAGE: The above two methods, node2vec and metapath2vec, map nodes to low-dimensional embeddings, but they do not incorporate the node features. They incorporate the structural information about a node's local neighborhood directly into the encoder. We assume that using the patent abstract of each patent along with the graph-structure in patent networks can improve learning the similarity of nodes in embedding space. Therefore, GraphSAGE was chosen for the graph neural network model to incorporate node features since it can be trained in an unsupervised way.

In general, GraphSAGE uses three steps for generating node embeddings: sample neighborhood, aggregate feature information from neighbors, and predict graph context and label using aggregated information [6]. To adapt to large scale networks, GraphSAGE uniformly samples a fixed-size

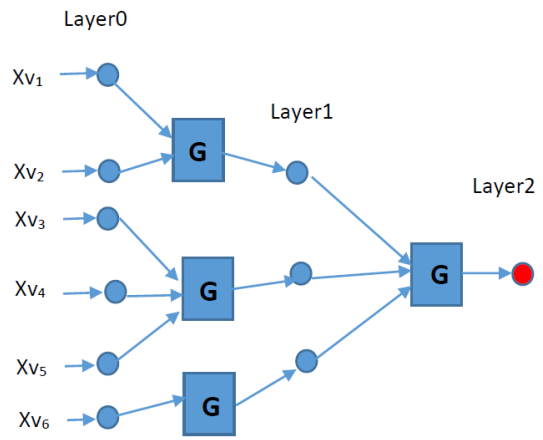


Fig. 2: The architecture of aggregation information from neighbors in GraphSAGE

set of neighbors for each node. To learn node embeddings, GraphSAGE uses multiple layer neural networks to aggregate information from neighbors (Figure 2). In the training process, the initial layer (Layer 0) of the neural network is assigned as the node features X. The node embeddings Z_{v_i} are aggregated from the direct neighbors in Layer1 and the second order neighbors in Layer0 using aggregation function G. GraphSAGE concatenates the node embedding of a node with neighbor embeddings, and uses mean, max-pooling and LSTM as aggregators in G [6]. Based on the same architecture of encode-decoder, GraphSAGE has several variations: GraphSAGE-mean, GraphSAGE-pool, GraphSAGE-LSTM, and GraphSAGE-GCN based on the aggergator used.

Similar to node2vec and metapath2vec, GraphSAGE is trained in an unsupervised manner, which enforces that nearby nodes have similar embeddings in embedding space. Applying GraphSAGE on patent networks, patent and organization embeddings are learned based on node features and the network structure. The node features of patents are represented as 128 dimension doc2vec vectors [2] generated from patent abstracts. The node features of organizations and patent groups are not available, so they are aggregated from the node features of associated patents.

After node embeddings of organizations are learned by node2vec, metapath2vec, or GraphSAGE, we use *k-means* for clustering organization embeddings. The organizations within the same cluster are considered to have competitive relationships.

IV. Experiments

In our analysis, topic modeling and graph embedding methods were used to obtain organization groupings in an unsupervised manner. These groupings can be thought of as clusters of competitors. Three graph embedding methods (node2vec, metapath2vec, and GraphSAGE) were used to perform the competitive analysis. node2vec, metapath2vec, and GraphSAGE all generated 128 dimension embeddings for organizations using the number of walks and walk lengths of 100. The similarities between embeddings was utilized to quantify similarities between organizations. The graph embedding method results were then compared to the baseline author-topic modelling results, which used five topics.

Year	#Patent	#Org
2006	15733	1802
2007	11717	1673
2008	14328	1897
2009	11178	1799
2010	17261	2182
2011	14364	2126
2012	19984	2389
2013	20232	2523
2014	26761	2687
2015	25866	2695
2016	28948	2729
2017	23059	2618
2018	29429	2657

TABLE I: The number of patents and organizations 2006-2018

A. Datasets

We used USPTO patent data from 2006 to 2018 for the competitive analysis [1]. The patent data includes patent, inventor, and assignee (organization) information. Each patent is assigned to multiple sections, subsections, and groups. Patent groups are finer-grained classification codes than the sections and subsections. We used the Cooperative Patent Classification (CPC) code as patent groups. Large firms usually have patents in a variety of patent groups while small firms have patents in few patent groups. One patent can have several inventors and each inventor is linked to an assignee. To simplify the patent network, we included patent-organization relationships and assigned each patent to only one patent group. The patent networks include three types of nodes: patents, organizations, and patent groups; and two types of links : patent-organization and patent-patent group. One patent network is generated from one-year patent data. While other patent networks use patent citation links, we ignore those links since the number of patent citations within the same year is very small. To reduce the number of organizations in the analysis, we only included organizations with at least 10 patents from 2006 to 2018. Also, we removed patents with no organization information. In total, the analysis included 280,827 patents in 63 patent groups from 2006 to 2018.

We split the patent data by year. The number of patents increased every year while the number of organizations remained relatively stable (Table I).

B. Evaluation

To establish a gold-standard for company similarities, we collected Forbes Global 2000 lists from 2006 to 2018, and

[1] https://www.patentsview.org/download/

then matched the companies on the Forbes lists with organizations in the patent data. Many companies in the patent dataset are large public companies posted on the Forbes Global 2000 list. We used the industry of these companies on the Forbes lists as the gold standard for organization groups. Since companies on the Forbes list and the patent dataset change every year, we had 300$\sim$400 organizations in the patent data matched to companies on the Forbes list, and resulted in 30$\sim$60 industry clusters each year. We only evaluated the clustering results on matched organizations.

We used Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) to evaluate the clustering results. Mutual information measures the dependence of two discrete random variables. It indicates the amount of information obtained by observing another random variable. We used normalized mutual information to measure how well the clustering results of the three models align with the Forbes list. Rand index measures the similarity between two data clusters. We used corrected-for-chance version of the Rand index: ARI [9].

V. Results

Patent networks were generated each year to learn the evolution of competitors. Since the business direction of a firm typically remains stable over the years, we included the patents in current years with patents in previous years in the same patent network. As the years progress, this forms incrementally larger patent networks used to learn the node embedding of organizations and patent groups. This section outlines the node embeddings learned by node2vec, metapath2vec and GraphSAGE. The clustering performance results of the author-topic, node2vec, metapath2vec and GraphSAGE models are then evaluated each year (Table II). Next, the performance of metapath2vec and node2vec is compared using multiple-year patent networks and single-year patent networks (Figure 5). Finally, the evolution of competitors is demonstrated in one case study.

A. Node embedding

node2vec and metapath2vec use different random walk strategies to learn shallow node embeddings. The node embeddings of organizations and patent groups of these two methods in a single year 2009 and merged multiple-year 2018 patent networks are shown in Figure 3. node2vec learns node embeddings of patents, organizations, and patent groups simultaneously using a biased random walk. The nodes of organizations and patent groups are mixed together even if we used all patent data in 2018 on node2vec (a and b in Figure 3). metapath2vec only learns the node embedding of organizations and patent groups (c and d in Figure 3). metapath2vec uses meta-path-based random walks and heterogeneous negative sampling, so adjacent nodes on meta-paths need to be different node types. The negative sampling is normalized with the node type as context for nodes. In 2009, the node embeddings of patent groups in metapath2vec are separated from organizations. This trend is more obvious using all patent data from 2016 sim 2018 (d in Figure 3).

The node embeddings of organizations generated from GraphSAGE mixed with patent groups (Figure 4), which is similar to node2vec. The node embeddings of patent groups are more centralized than those of node2vec, which indicates the close distance between them. The similarities between patent groups may be partially caused by aggregated node features from patent abstracts. Since the computation time for GraphSAGE is long, only single-year patent data for year 2018 was used to generate b in Figure 4.

B. Single-year patent networks

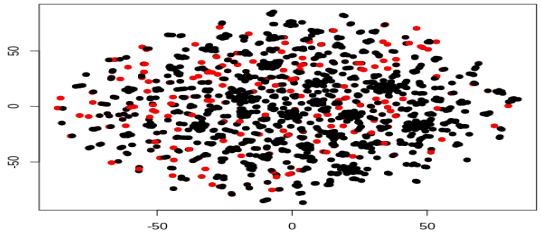
The clustering results of competitive relationships identified using four models improve over the years (Table II). This may be caused by the number of patents increasing in recent years, which would reflect the reality of normal business activity for companies. On average, the author-topic model was the worst performer. It used patent abstracts to infer the similarities between companies. All three graph embedding methods outperformed the author-topic model although metapath2vec and node2vec only used the structural information of networks. This indicates that it may be insufficient to rely simply on textual information to generate accurate company profiles. Surprisingly, node2vec performed better than metapath2vec by a large margin. metapath2vec was designed for heterogeneous networks, and the distance between same the type of nodes is smaller than different types of nodes in embedding space (Figure 3). Another unexpected observation is node2vec performed better than GraphSAGE, which incorporates the node features from patent abstracts to generate node embeddings.

C. Multiple-year patent networks

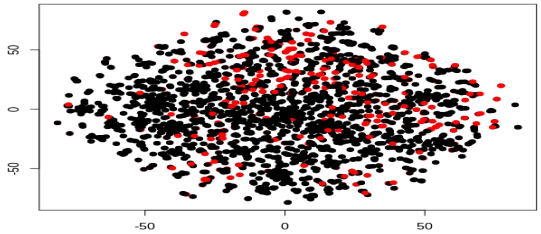
The performance of the metapath2vec and node2vec methods improves as the models incorporate patent data from previous years (Figure 5). Every year, node2vec and metapath2vec achieved better performance on the cumulative, multi-year patent network than the equivalent single-year patent network. node2vec reaches the best performance in 2018 (NMI = 0.657), which includes all patent data from 2006 to 2018. This implies the models learn community structures better by using patent data from multiple years simultaneously.

D. Evolution of competitors

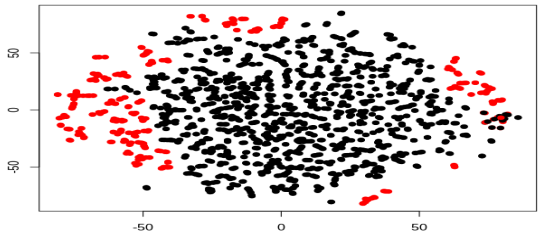
We used "Intel Corporation" as a case study to examine whether the competitors learned using node2vec were meaningful. The evolution of competitors were observed from single-year and multi-year patent networks. *Intel* is a multinational corporation and the second largest semiconductor chip manufacturer in the world. Competitors were calculated using the cosine similarity of normalized node embeddings between *Intel* and all other companies each year. The most similar companies were considered competitors in that year. The top competitors from 2015 to 2018 are showed in Table III. Some well known semiconductor manufactures such as



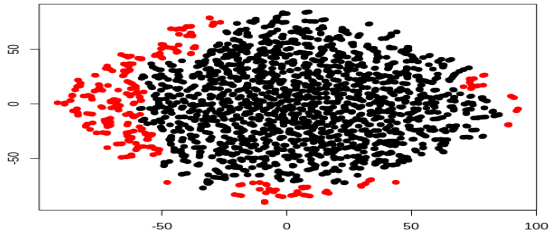
(a) node2vec in 2009



(b) node2vec in M2018

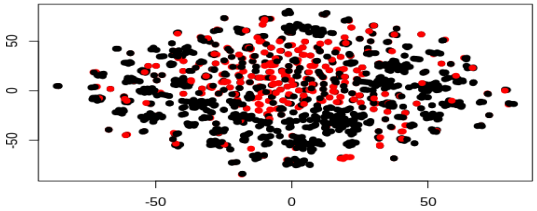


(c) metapath2vec in 2009

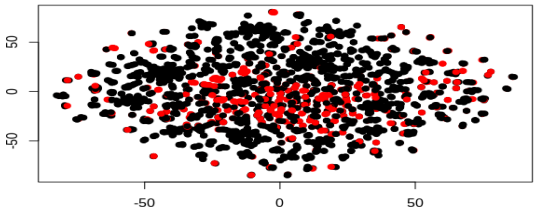


(d) metapath2vec in M2018

Fig. 3: Node embedding of organizations (black nodes) and patent groups (red nodes) learned by node2vec and metapath2vec on 2009 and merged 2018 patent networks



(a) GraphSAGE in 2009



(b) GraphSAGE in 2018

Fig. 4: Node embedding of organizations (black nodes) and patent groups (red nodes) learned by GraphSAGE on 2009 and 2018 patent networks

Year	author-topic NMI	metapath2vec NMI	GraphSAGE NMI	node2vec NMI
2006	0.393	0.461	0.473	**0.549**
2007	0.409	0.420	0.458	**0.544**
2008	0.385	0.444	0.447	**0.496**
2009	0.309	0.366	0.388	**0.467**
2010	0.337	0.402	0.444	**0.523**
2011	0.380	0.447	0.467	**0.529**
2012	0.402	0.468	0.496	**0.568**
2013	0.420	0.478	0.509	**0.582**
2014	0.456	0.514	0.525	**0.603**
2015	0.459	0.501	0.520	**0.599**
2016	0.483	0.527	0.537	**0.604**
2017	0.499	0.539	0.545	**0.604**
2018	0.488	0.551	0.535	**0.616**
Average	0.417	0.470	0.488	**0.560**
	ARI	ARI	ARI	ARI
2006	0.023	0.112	0.097	**0.192**
2007	0.048	0.065	0.091	**0.222**
2008	0.047	0.114	0.101	**0.163**
2009	0.040	0.080	0.102	**0.174**
2010	0.044	0.102	0.141	**0.202**
2011	0.019	0.093	0.096	**0.146**
2012	0.023	0.079	0.113	**0.180**
2013	0.019	0.086	0.104	**0.208**
2014	0.046	0.096	0.105	**0.177**
2015	0.031	0.077	0.097	**0.183**
2016	0.026	0.072	0.067	**0.134**
2017	0.034	0.067	0.076	**0.135**
2018	0.017	0.099	0.068	**0.156**
Average	0.032	0.088	0.097	**0.175**

TABLE II: Clustering performance of four models

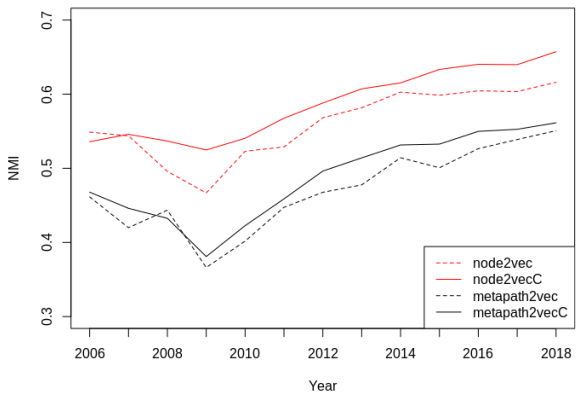


Fig. 5: The performance of node2vec and metapath2vec on single-year patent networks and node2vecC and metapth2vecC on multi-year patent networks

QUALCOMM, *Broadcom* and *SAMSUNG* were identified as *Intel* competitors. Also, new players in the field such as *Huawei* ad *Tencent* were also discovered from single year networks.

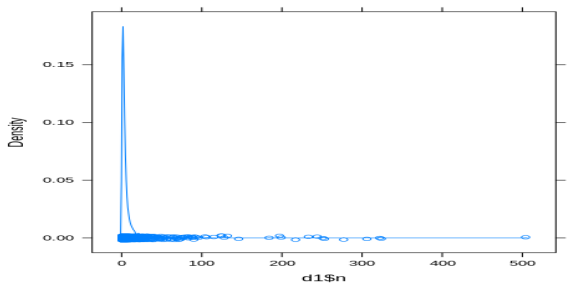
The top competitors from the multiple-year patent network modeling results are also described in Table III. The competitors identified from the merged patent networks share many common companies such as *IBM* and *QUALCOMM*. However, competitors from the merged patent networks reflect historical competitive relationships. For instance, *IBM* was identified as the #1 competitor from the merged patent networks since *IBM* has the largest number of patents in a broad range of areas over many years. Although semiconductor is not the main business area of *IBM*, the historical influence is strong on generating node embeddings. Also, the competitor lists from the merged patent networks are more stable. That may explain why the performance of clustering was better on multiple year patent networks. The competitor lists discovered from both the single and multi-year patent networks are useful: new competitors can be identified from single-year data and long-standing competitive relationship can be explored from multi-year data.

VI. Discussion

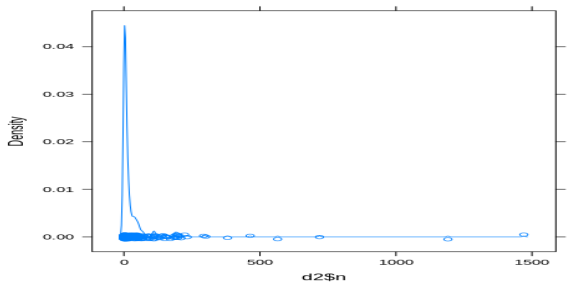
The overall performance of either graph embedding models or author-topic model was low for a few reasons: difficulty of the task; noise in the data; unbalanced data distribution; and small datasets. At first, clustering a small number of companies into a relatively large number of clusters is a challenging task. Typically, only 300∼400 companies were matched with the Forbes list and grouped into 30∼60 industries each year. Second, our data, which included many domains, was more noisy than the single-domain data used in other graph embedding studies [3], [4]. Third, the distribution of the number of patents owned by each organization and patent group are skewed (Figure 6). The unbalanced distributions make it harder to learn

2015	2016	2017	2018
QUALCOMM	QUALCOMM	QUALCOMM	QUALCOMM
Broadcom	IBM	IBM	SAMSUNG
IBM	BlackBerry	MEDIATEK	IBM
FUJITSU	FUJITSU	HUAWEI	NEC
Atmel	Microsoft	FUJITSU	Google
SEVEN Networks	Broadcom	AT&T Mobility	Microsoft
Microsoft	HUAWEI	Microsoft	Apple
NOKIA	NEC	Renesas Electronics	Facebook
Lenovo	Apple	ERICSSON	MEDIATEK
FireEye	Lenovo	TENCENT	Alcatel Lucent
M2015	M2016	M2017	M2018
IBM	IBM	IBM	IBM
Synopsys	LSI Logic	QUALCOMM	QUALCOMM
Oracle	QUALCOMM	VIA	FUJITSU
Broadcom	Agere Systems	Nutanix	Broadcom
Open Text S.A.	Oracle	Synopsys	Cadence Design Systems
Altera	Altera	Broadcom	Oracle
FUJITSU	Broadcom	Cadence Design Systems	HUAWEI
Dell	AMDOCS	YANDEX	NEC
QUALCOMM	VIA TECHNOLOGIES	XILINX	NVIDIA
NVIDIA	Cadence Design Systems	NVIDIA	Altera

TABLE III: Intel competitors from 2015 to 2018 on single and multi-year patent networks



(a) Patents by organization



(b) Patents by patent groups

Fig. 6: Density distribution of patents by organizations and patent groups

the implicit relationships between high-degree nodes of organizations and patent groups. Most importantly, only a relatively small number of patents each year were used to learn the node embeddings in our study. It shows that the performance of graph embedding methods to discover competitive relationships is better on larger merged patent networks.

A. *node2vec vs. metapath2vec*

The main focus of this study was to investigate graph embedding methods and infer implicit similarities between organizations from explicit relationships between organizations and patents in patent networks. node2vec performed better than metapath2vec at this task. metapath2vec utilizes meta-path-based random walks and heterogeneous negative sampling to learn node embeddings such that the distance

between nodes of the same type was closer than ones with differing types (Equation 5). The meta-path random walks in metapath2vec may be too strict and prevent it from learning implicit organization relationships. node2vec uses biased random walk strategies with BFS and DFS. Although node2vec ignores node types on biased random walks, the node similarities between organizations are captured.

B. node2vec vs. GraphSAGE

The purpose of using GraphSAGE is to aggregate the patent abstracts and structural information in graph embedding methods to learn the similarities of nodes. However, GraphSAGE performs worse than node2vec, which does not use node features. It may due to two reasons: the random walk strategy and the aggregation function of node features for organizations and patent groups. node2vec uses a hybrid strategy of breath-first sampling (BFS) and depth-first sampling (DFS) biased random walk. Therefore, it is able to explore the direct neighbors and also distant neighbors. To reduce the computational time on large networks, GraphSAGE generates node embeddings of each node based on a uniformly sampling a fixed size neighbors. This strategy may not be an optimal solution for our task of learning implicit relationships between patent groups and organizations, which are not direct neighbors.

Another limitation of applying GraphSAGE in learning organization embeddings is we simply average node features of patents owned by an organization to represent the node features of the organization. The same dimension of node features of organizations and patent groups as patents was used, but using 128 dimension vectors to represent organizations and patent groups may not be sufficient to represent semantic features of organizations and patent groups. The node features are used for the initial layer of node embedding in GraphSAGE. The limitations of initial node features is more obvious because GraphSAGE was trained in an unsupervised way, so there is not enough guidance on learning node embeddings. Unlike other supervised tasks using graph embedding such as node classification, the node features of these aggregated nodes (organizations and patent groups) were adjusted in the training process to be an appropriate representation.

C. Parameter Analysis

We performed parameter analysis on all models used in this study. First, in the baseline model, the number of topics is the main parameter in the author-topic model. The average NMI dropped from 0.416 to 0.401 when the number of topics was changed from 5 to 10. It did not have a large impact on the overall clustering results with the varied number of topics.

For graph embedding models, the number of walks and walk length are major parameters for random walk strategies, and the dimension of node embeddings and the number of epochs are the primary training parameters. The number of walks and walk lengths did have some influence on the clustering result in node2vec and metapath2vec. The performance of node2vec improves with the increase in the number of walks and walk length, while the clustering results from metapath2vec does not vary much (Table IV). The variant of GraphSAGE is mainly on aggregator used for aggregate messages from neighbors and the neural network applied. We compared three GraphSAGE variants: mean, pooling, and GCN, and found GCN, which was used in our study, works better than the other two (Table V).

NMI	W10 L80	W 100 L20	W100 L100
node2vec	0.545	0.546	0.560
metapath2vec	0.488	0.493	0.488

TABLE IV: The influence of number of walks and walk length on the average performance of node2vec and metapath2vec

	GraphSAGE		
Avg	mean	pooling	GCN
NMI	0.468	0.467	**0.488**
ARI	0.083	0.087	**0.097**

TABLE V: The average performance of GraphSAGE using different aggregators

D. Evaluation

One challenge for evaluating competitive analysis is the difficulty of compiling a gold standard of competitor relationships. In our work we only evaluated patent organizations matched with companies found inside the Forbes lists. This complication has also been contended with in other related work. For example, Yang et al. used Yahoo's company financial profiles, which provided a static view of competitor relationships with only a portion being linked to patent organizations [19]. While the Forbes lists are not a perfect gold standard, we decided to use them as they are a source of dynamic groupings of competitors over time, which was essential for evaluating the evolution of competitors.

VII. Conclusion

In this study, we used the node2vec, metapath2vec, and GraphSAGE graph embedding methods to perform competitive analysis. These models were trained in an unsupervised manner to learn the node embeddings in patent networks. node2vec and metapath2vec only use the structural information and GraphSAGE uses the graph structure and patent abstracts to learn node embeddings. We then compared results obtained using these methods with the results of the author-topic method, which infers the similarities of companies using semantic information inside patent abstracts. The results show the graph embedding methods outperformed the topic modeling method. Unexpectedly, node2vec outperformed metapath2vec and GraphSAGE. metapath2vec was designed for heterogeneous networks, and it learns node embeddings in latent space such that similar types of nodes are close to each other. GraphSAGE learns the node embeddings in a multi-layer neural network by aggregating information from neighbors. These three graph embedding

methods use different random walk strategies for sampling neighbors. node2vec uses a biased random walk strategy which explores direct and undirect neighbors; metapath2vec uses meta-path neighbors in different types of nodes; GraphSAGE adopts a uniform sample of fix-sized neighbors. The superior performance of node2vec may be due to its random walk strategy. Using multi-year patent networks, the results of competitors identified from node2vec and metapath2vec are improved. We also show the evolution of competitors in a case study of *Intel* from the node embedding learned by node2vec.

This work applied several well-known graph embedding methods to uncover implicit competitive relationships between companies in patent networks. The experimental results confirm the superior performance of graph embedding methods over topic models and also identified potential problems in such applications. For example, how can competitors in a specific aspect of a technology be identified, which is useful for analyzing the strength of competitors. Also, designing an effective graph embedding method on large scale heterogeneous networks is still challenging. In addition, graph embedding methods have shown strong performance on supervised learning tasks such as node classification and link prediction. However, more studies are still needed to determine how to learn node embeddings more effectively in an unsupervised graph embedding model.

REFERENCES

[1] Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research **3**(Jan), 993–1022 (2003)

[2] Dai, A.M., Olah, C., Le, Q.V.: Document embedding with paragraph vectors. arXiv preprint arXiv:1507.07998 (2015)

[3] Dong, Y., Chawla, N.V., Swami, A.: metapath2vec: Scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 135–144. ACM (2017)

[4] Fu, T.y., Lee, W.C., Lei, Z.: Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 1797–1806. ACM (2017)

[5] Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864. ACM (2016)

[6] Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in neural information processing systems. pp. 1024–1034 (2017)

[7] Hamilton, W.L., Ying, R., Leskovec, J.: Representation learning on graphs: Methods and applications. arXiv preprint arXiv:1709.05584 (2017)

[8] Hofmann, T.: Probabilistic latent semantic analysis. In: Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence. pp. 289–296. Morgan Kaufmann Publishers Inc. (1999)

[9] Hubert, L., Arabie, P.: Comparing partitions. Journal of classification **2**(1), 193–218 (1985)

[10] Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)

[11] Li, C., Song, Z., Tang, J.: User tagging in moocs through network embedding. In: 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC). pp. 235–241. IEEE (2018)

[12] Peng, H., Li, J., Gong, Q., Song, Y., Ning, Y., Lai, K., Yu, P.S.: Fine-grained event categorization with heterogeneous graph convolutional networks. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19. pp. 3238–3245. International Joint Conferences on Artificial Intelligence Organization (7 2019)

[13] Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 701–710. ACM (2014)

[14] Rosen-Zvi, M., Griffiths, T., Steyvers, M., Smyth, P.: The author-topic model for authors and documents. In: Proceedings of the 20th conference on Uncertainty in artificial intelligence. pp. 487–494. AUAI Press (2004)

[15] Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: European Semantic Web Conference. pp. 593–607. Springer (2018)

[16] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: Proceedings of the 24th international conference on world wide web. pp. 1067–1077. International World Wide Web Conferences Steering Committee (2015)

[17] Tang, J., Wang, B., Yang, Y., Hu, P., Zhao, Y., Yan, X., Gao, B., Huang, M., Xu, P., Li, W., et al.: Patentminer: topic-driven patent analysis and mining. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1366–1374. ACM (2012)

[18] Wang, B., Liu, S., Ding, K., Liu, Z., Xu, J.: Identifying technological topics and institution-topic distribution probability for patent competitive intelligence analysis: a case study in lte technology. Scientometrics **101**(1), 685–704 (2014)

[19] Yang, Y., Tang, J., Keomany, J., Zhao, Y., Li, J., Ding, Y., Li, T., Wang, L.: Mining competitive relationships by learning across heterogeneous networks. In: Proceedings of the 21st ACM international conference on Information and knowledge management. pp. 1432–1441. ACM (2012)

[20] Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 974–983. ACM (2018)