Paper one

Network Representation Learning: A Survey

Purpose

This paper tries to give us a thorough review of current research on Network Representation Learning (NRL). It not only proposes a new categorization of NRL techniques, but also provides a detailed study of these methods including their computational models and cons-and-pros. Furthermore, the applications of NRL are shown in the paper.

NRL's brief Introduction

Its main idea is to learn a mapping function from an information network G=(V,E,X,Y): $\mathrm{f}: v \to r_v \in \mathrm{R}^d$, where r_v is the learned representation of vertex v, and d is the dimension of the learned representation. Also, NRL must meet the following three requirements: low-dimensional, informative and continuous.

There are many kinds of measure of distance: first-order proximity, second-order proximity, high-order proximity and intra-community proximity.

NRL's Challenges

- Structure-preserving
- Content-preserving (mainly about content on attributes of vertices and edges)
- Data sparsity (real-world information networks)
- Scalability

NRL's Categorization

- Unsupervised NRL
  - Unsupervised Structure Preserving NRL
  - Unsupervised Content Augmented NRL
- Semi-supervised NRL
  - Semi-supervised Structure Preserving NRL
  - Semi-supervised Content Augmented NRL

This paper firstly categorizes NRL techniques into two groups, unsupervised NRL and semi-supervised NRL, on the basis of whether vertex labels are used in learning. Secondly, it further categorizes those techniques into two subgroups, depending on whether content of vertex attributes are used in learning.

NRL's Application

- Vertex Classification
- Link Prediction
- Clustering
- Visualization

- Recommendation

NRL's Algorithms:

Unsupervised Structure Preserving NRL

- Learning Latent Social Dimensions

This problem is closely related to Community Detection, whose aim is to discover a community with concentrated inside than outside. The algorithms which belong to this kind of NRL are followed: Modularity Maximization, Spectral Clustering, Edge Clustering and so on.

- Large-scale Information Network Embedding (LINE)

LINE algorithm learns the Network Representation by explicitly modeling the first-order and second-order proximity. The following formulas are one example of explicit models on first-order and second-order proximity:

$$p_1(v_i, v_j) = \frac{1}{1 + \exp(-\vec{u}_i^T \cdot \vec{u}_j)} \qquad p_2(v_j|v_i) = \frac{\exp(\vec{u}_j'^T \cdot \vec{u}_i)}{\sum_{k=1}^{|V|} \exp(\vec{u}_k'^T \cdot \vec{u}_i)}$$

And the goal of this algorithm is to preserve the first-order or second-order proximity, thus minimizing there two objectives: $O_1 = d(\hat{p}_1(\cdot,\cdot), p_1(\cdot,\cdot)), \qquad O_2 = \sum_{v_i \in V} \lambda_i d(\hat{p}_2(\cdot|v_i), p_2(\cdot|v_i)),$

For example, in order to keep the first-order proximity, O1 can be expressed in this way: $O_1 = -\sum_{(i,j)\in E} w_{ij} \log p_1(v_i, v_j)$ which correspnds the above notation p1.

- DeepWalk

Randomly pick the nodes of the graph, and make a vertex sentence with a fixed size of length. Then the goal is to minimize the formula in skip-gram model: $\min_f -\log \Pr(\{v_{i-t}, \cdots, v_{i+t}\} \setminus v_i | f(v_i)),$, where $\{v_{i-t}, \cdots, v_{i+t}\} \setminus v_i$ represents the context vertices within t window size. If we make conditional independence assumption, the above formula can be transformed to $\Pr(\{v_{i-t}, \cdots, v_{i+t}\} \setminus v_i | f(v_i)) = \prod_{i=i-t, i\neq i}^{i+t} \Pr(v_j | f(v_i)).$.

The steps of DeepWalk are as follows:

Graph → Random walk → Representation mapping → Using skip-gram Model → Network representation

- GraRep

This algorithm inherits the idea of RandomWalk, using the skip-gram Model to capture the high-order proximity. But specially, it defines each vertex's k-step neighbors as context vertices to learn k-step vertex representations. What's more, GraRep employs the matrix factorization.

- Node2Vec

This algorithm upgrades DeepWalk, by considering Breadth-first or Depth-first

strategy. This kind of biased random walk can better preserve both the local and global structure which corresponds to second-order proximity and high-order proximity respectively.

Unsupervised Content Augmented Network Representation Learning

- Text-Associated DeepWalk (TADW)

Without content textual features, the original goal is to minimize the formula, $\min_{W,H} \|M - W^{\mathrm{T}}H\|_F^2 + \frac{\lambda}{2}(\|W\|_F^2 + \|H\|_F^2),$, the matrix factorization from DeepWlak, from where W and H are learned latent embeddings and M is the node-context matrix.
After adding textual features, the original formula change into this way:
$\min_{W,H} \|M - W^{\mathrm{T}}HT\|_F^2 + \frac{\lambda}{2}(\|W\|_F^2 + \|H\|_F^2),$ where T is vertex textual feature matrix.

- Homophily, Structure, and Content Augmented Network Representation Learning (HSCA)

According to the paper, TADW pays more attention to structural context of network vertex, but less attention to the homophily property. Therefore, HSCA is going to enhance the homophily property of Network Representation Learning, namely the first-order proximity. The method is to penalize the distance between vertices by introducing a regularization term.

Semi-Supervised Structure Preserving NRL

- Discriminative Deep Random Walk (DDRW)

DDRW proposes to optimize the objective of DeepWalk together with the following L2-loss Support Vector Classification objective: $\mathcal{L}_c = C\sum_{i=1}^{|V|}(\sigma(1 - Y_{ik}\beta^{\mathrm{T}}r_{v_i}))^2 + \frac{1}{2}\beta^{\mathrm{T}}\beta,$,

where $\sigma(\mathrm{x}) = \begin{cases} x, x > 0 \\ 0, otherwise \end{cases}$. After adding the term, the joint objective is going to be $\mathcal{L} = \eta\mathcal{L}_{DW} + \mathcal{L}_c,$, where L_{DW} is the original objective of DeepWalk.

- Transductive LINE (TLINE)

TLINE combines LINE algorithm with SVM classifier. TLINE trains an multi-class SVM classifier on the set of labeled by optimizing the objective O_{SVM} . Also, according to LINE's attention to first-order and second-order proximity, TLINE optimizes two objective functions: $\mathcal{O}_{TLINE(1st)} = \mathcal{O}_{line1} + \beta\mathcal{O}_{svm},$ and $\mathcal{O}_{TLINE(2nd)} = \mathcal{O}_{line2} + \beta\mathcal{O}_{svm},$.
Comparing with LINE, TLINE works with lower time and memory cost.

Semi-Supervised Content Augmented NRL

- Tri-Party Deep Network Representation (TriDNR)

This technique considers these three aspects: network structure, vertex label and vertex content. It aims to maximize the following objective:

$$\mathcal{L}_{PV} = \sum_{i \in L} \log \Pr(w_{-b} : w_b | c_i) + \sum_{i=1}^{|V|} \log \Pr(w_{-b} : w_b | v_i),$$

, where $\{w_{-b} : w_b \mid c_i\}$ means a string of words in a 2b-length-long window and c_i represents the class label of vertex v_i.

- **Linked Document Embedding (LDE)**

LDE is designed to learn representations for linked documents. Similar to TriDNR, LDE also models three kinds of relations. This technique considers word-word-document relations, document-document relations and document-label relations. Its goal is to minimize the objective as followed:

$$\begin{aligned}\min_{\boldsymbol{W},\boldsymbol{D},\boldsymbol{Y}} & -\frac{1}{|\mathcal{P}|} \sum_{(w_i,w_j,d_k)\in\mathcal{P}} \log \Pr(w_j|w_i,d_k) \\ & -\frac{1}{|E|} \sum_{i} \sum_{j:(v_i,v_j)\in E} \log \Pr(d_j|d_i) \\ & -\frac{1}{|\mathcal{Y}|} \sum_{i:y_i\in\mathcal{Y}} \log \Pr(y_i|d_i) \\ & +\gamma(\|\boldsymbol{W}\|_F^2 + \|\boldsymbol{D}\|_F^2 + \|\boldsymbol{Y}\|_F^2).\end{aligned}$$

,

where the first term is associated with word-word-document relations, and the second and third term represent document-document relations and document-label relations respectively. The last regularization term, from which W, D, Y is the embedding matrix for words, documents and labels respectively, is introduced to avoid overfitting.

Paper two

A Survey of Heterogeneous Information Network Analysis

Purpose

This paper mainly summaries the research analysis on heterogeneous information network(HIN), by introducing basic concepts of HIN as well as discussing advanced topics on the basis of HIN.

Definition

Heterogeneous information network (HIN)

Types of objects $|A| > 1$ or types of relations $|R| > 1$

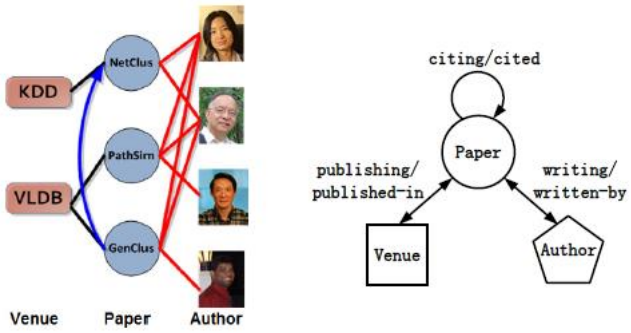
Network schema

Just like the relationship between database schema and database instance from the theory of database, the relationship between network schema and network instance

are as follows: For a graph $G=(V,E)$, its schema is $T_G=(A,R)$ and $\varphi: V \to A$ $\phi: E \to R$ where φ is object type mapping and ϕ is link type mapping.

Other notation $S \xrightarrow{R} T$ means a link type R connecting object type S to object type T. The inverse relation R^{-1} holds $T \xrightarrow{R^{-1}} R$ naturally.

The following picture illustrates clearly about the relations between Network Instance and Network Schema



(a) Network instance (b) Network schema

Most common HIN:

a) Multi-relational network with single-typed object, like Facebook
b) Bipartite network, like user-item and document-word.
c) Star-schema network, most popular because one kind of object and its attributes naturally construct an HIN in database table.
d) Multiple-hub network, on the basis of star-schema, this kind of network contains some hub objects.

HIN's advantages

1. New development of data mining:

As a semi-structured with more different types of objects and relations, heterogeneous information network is capable to handle more complex big data.

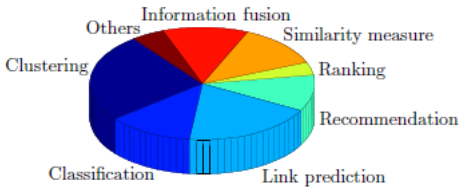
2. Fuse more information:

To construct a heterogeneous network by combining information from different platforms

3. Contain rich semantics:

In heterogeneous information network, different-typed objects and links carry different semantic meanings.

HIN's research developments



Similarity Measure

Its goal is to find a similarity evaluation of objects. Popular examples are: PageRank, SimRank, Jaccard coefficient, cosine coefficient and so on. On the basis of HIN, similarity measure can take the meta path connecting two objects into account.

Clustering

It aims to partition objects into several clusters, such that objects in a cluster are similar to each other while dissimilar to objects in other clusters. Conventional techniques are: k-means, spectral clustering, greedy method and so on. Recently some ranking-based clustering methods on HIN have emerged, which is capable to improve mutual performances.

Classification

It is proposed to build a model or classifier to predict class labels. Traditional methods pay more attention to the objects' features, making independent identically distribution(IID) assumption. But in the graph, objects does not satisfy independent identically distribution, due to the existing links. In HIN, the label of objects is decided by the joint effect of different-typed objects along with different-typed links.

Link Prediction

Its task is to decide whether the link exists or not for any two potentially linked objects. Link prediction in HIN need at least two steps: Firstly, extract meta path on the basis of feature vectors. Secondly, train a regression or classification model to predict the existence probability of a link.

Ranking

It aims to evaluate the importance of objects according to specific ranking methods. But when do this work in an HIN, problems emerge like mixing different types of objects together when treating all objects equally, and getting different results according to different meta paths. Therefore, many works in this field consider various types of relations but only ranks one type of objects.

Recommendation

Recommendation system is designed to introduce products that are likely to be of interest to people, in order to save consumers' time on searching. On the basis of similarities among items and customer preferences, traditional way utilizes the user-item rating feedback for recommendation, such as collaborative filtering. What's more, making recommendations based on an HIN can utilize more

information and semantics hidden in HIN.

Information Fusion
It is a process of merging different types of information -- conceptual, contextual, typographical representations. To make it works in multiple HINs, it is essential to align the HINs via the shared common information entities. By finishing fusing multiple HINs successfully, the heterogeneous information available in each network can be transferred to other aligned networks, with a great number of applications carried out.

My Another Work
The two papers in week two are all survey papers, which means they are great summaries in each field with a lot of theories or algorithms. But the amount of theories and algorithms is huge, we students may not dig them too deep within a week. Therefore, I not only read and try to understand the two papers as much as possible, but I am going to study a specific technique which is mentioned in both papers – spectral clustering.

Spectral Clustering

Main Algorithm
==========
INPUT: dataset (x1,x2,…xn), dimension k1 after dimension-reduction, dimension k2 after clustering

OUTPUT: clusters C(c1,c2,…cn)

1. Build the Similarity Matrix S
2. Construct Laplacian Matrix L
3. Compute the standardized Laplacian Matrix $\mathrm{D}^{-1/2}LD^{1/2}$
4. Compute the smallest k_1 eigenvalues an eigenvectors f
5. Use eigenvectors f to make up a matrix and normalize the matrix following the rows to get a feature matrix F ($R^{n\times k_1}$)
6. For each row of the matrix F, cluster the data samples into k_2 clusters.

==========
Specific Theory
1) Similarity Matrix S
   Generally, there are three ways to get similarity matrix. The most common one is full-link method. The method usually utilize Gauss kernel function RBF to define the similarity thus constructing similarity matrix, namely adjacency matrix:

$$W_{ij} = S_{ij} = \exp(-\frac{\left\|x_i - x_j\right\|_2^2}{2\sigma^2})$$

2) Laplacian Matrix L

Its definition is simple and clear: $L = D - W$, where D denotes the degree matrix D and W denotes the adjacency matrix W.

From 1), we have already known how to get W. And D is also easy to compute since its definition is:

$$\mathbf{D} = \begin{pmatrix} d_1 & \dots & \dots \\ \dots & d_2 & \dots \\ \vdots & \vdots & \ddots \\ \dots & \dots & d_n \end{pmatrix}$$

, where di represents the degree of node vi.

3) Compute eigenvalues and eigenvectors

Eigenvalue λ and eigenvector v satisfy the following equation:

$$A\vec{v} - \lambda\vec{v} = 0 \rightarrow \vec{v}(A - \lambda I) = 0,$$

When v is not an empty vector, from the above equation we can know that the determinant of the matrix should be zero.

$$Det(A - \lambda I) = 0.$$

Thus, we can calculate the eigenvalue λ from the above equation. Then we can also compute the eigenvector v by assigning the eigenvalue λ to the original equation which is a definition.

4) Clustering method

As we get the feature matrix F, we need to divide the data of each row into clusters with size k2. The commonly used method of clustering here is K-means, which is familiar to me.

Advantages and Disadvantages

advantages:

a) Spectral Clustering utilizes similarity matrix. Therefore, it is efficient to deal with sparse data, which means it is better than many conventional clustering methods like K-means in this aspect.

b) Spectral Clustering works better when handling high-dimension data because it contains the technique of dimension-reduction.

disadvantages:

a) Spectral Clustering highly depends on the similarity matrix. Thus, different kinds of similarity matrix we get will leads to different results.