A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications

1. Introduction

1.1 Graph Representation

1. 对每个node进行编码：基于流学习 4.1 节(早期方法)
2. 对graph的子部分进行编码 (对应下图bd)
    - 基于深度学习 4.2
    - 基 于对计算「edge重建概率目标函数」的设计 4.3
3. 对graph整体进行编码：4.4(对应下图e)

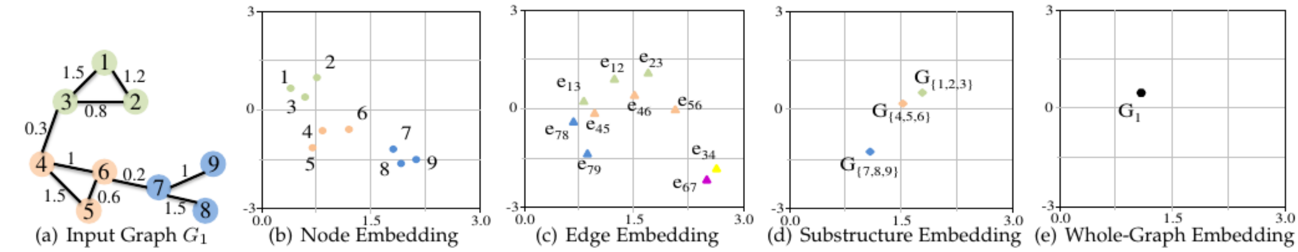


Fig. 1. A toy example of embedding a graph into 2D space with different granularities. $G_{\{1,2,3\}}$ denotes the substructure containing node v_1, v_2, v_3.

1.2 Graph Embedding

1.2.1 Graph Embedding != Graph Representation

相关的两个问题

- Graph Representation：表示Graph.
- Graph Analytics：从Graph中抽取信息

Graph Embedding可以解决上面两个问题. 即 **在表示的过程中包含了graph的信息**

这个**包含信息**的方法是, 将Graph表示为 低维向量(s), 与此相对, **无法包含信息**的Representation方法也有, 比如one-hot向量, 它就只是表示, 而无法**包含信息**

1.2.2 Graph Embedding的种类

按照input输入:

- Homogeneous graph : 同构网络, 网络中的节点没有进行类型标注.
- Heterogeneous graph : 异质网络, 网络中的节点有不同的类型属性.
- 外部信息+graph : 比如关系抽取.
- 非结构性数据(中构建Graph)

按照output分类:

- node embedding
- edge embedding
- 混合embedding
- 整个Graph的embedding.

1.3 分类方法

按照两种分类规则进行介绍, 并非上面的两种, 而是

- 根据问题设置的文献分类 : 这里在安装1.2.2 节进行分类. (**本文独到**)
- 按照使用技术的分类 : 这里按照使用的手法的insight进行分类.(**本文独到是解释每个方法的本质**)

见下图:

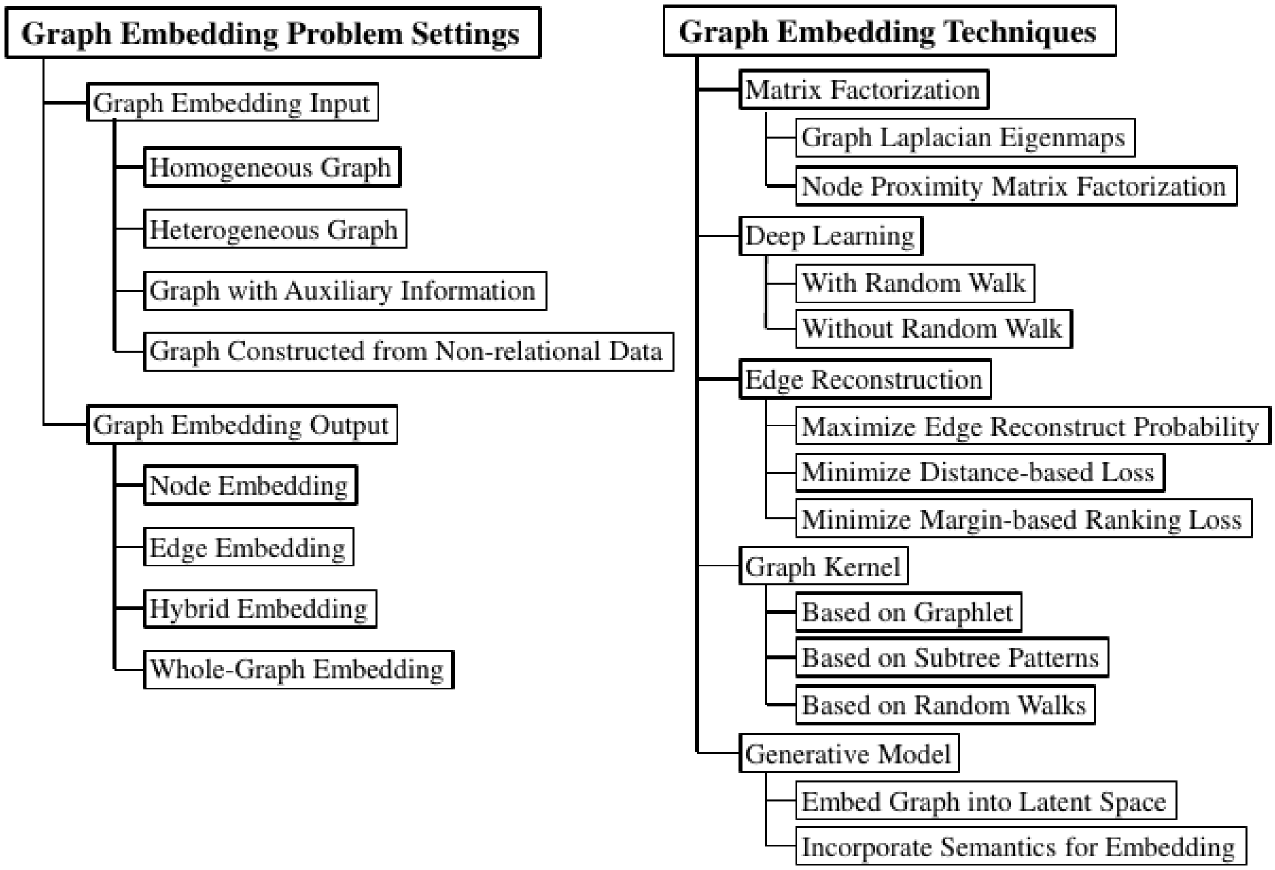


Fig. 2. Graph embedding taxonomies by problems and techniques.

1.4 本文贡献

- 从问题设定和方法两个方向进行survey
- 对使用的方法进行了详细的本质上的分析
- 对使用Graph Embedding的研究进行了介绍
- 从四个方向提出了未来的研究方向：计算效率的提升, 问题设定, 解决方案, applications

1.5 本文架构

2. 介绍定义
3. 在问题设定上的介绍
4. 在解决方案上的介绍.
5. 应用介绍
6. 未来方向

2. Problem Formalization

定义如下:

Definition 1. A **graph** is $\mathcal{G} = (V, E)$, where $v \in V$ is a node and $e \in E$ is an edge. $\mathcal{G}$ is associated with a node type mapping function $f_v : V \rightarrow \mathcal{T}^v$ and an edge type mapping function $f_e : E \rightarrow \mathcal{T}^e$.

$\mathcal{T}^v$ and $\mathcal{T}^e$ denote the set of node types and edge types, respectively. Each node $v_i \in V$ belongs to one particular type, i.e., $f_v(v_i) \in \mathcal{T}^v$. Similarly, for $e_{ij} \in E$, $f_e(e_{ij}) \in \mathcal{T}^e$.

TABLE 1
Notations used in this paper.

Notations	**Descriptions**
$\lvert \cdot \rvert$	The cardinality of a set
$\mathcal{G} = (V, E)$	Graph $\mathcal{G}$ with nodes set V and edges set E
$\hat{\mathcal{G}} = (\hat{V}, \hat{E})$	A substructure of graph $\mathcal{G}$, where $\hat{V} \subseteq V$, $\hat{E} \subseteq E$
v_i, e_{ij}	A node $v_i \in V$ and an edge $e_{ij} \in E$ connecting v_i and v_j
A	The adjacent matrix of $\mathcal{G}$
A_i	The i-th row vector of matrix A
$A_{i,j}$	The i-th row and j-th column in matrix A
$f_v(v_i)$, $f_e(e_{ij})$	Type of node v_i and type of edge e_{ij}
$\mathcal{T}^v$, $\mathcal{T}^e$	The node type set and edge type set
$\mathcal{N}_k(v_i)$	The k nearest neighbours of node v_i
$X \in \mathbb{R}^{\lvert V \rvert \times N}$	A feature matrix, each row X_i is a N-dimensional vector for v_i
y_i, y_{ij}, $y_{\hat{\mathcal{G}}}$	The embedding of node v_i, edge e_{ij}, and structure $\hat{\mathcal{G}}$
d	The dimensionality of the embedding
$< h, r, t >$	A knowledge graph triplet, with head entity h, tail entity t and the relation between them r
$s_{ij}^{(1)}$, $s_{ij}^{(2)}$	First- and second-order proximity between node v_i and v_j
c	An information cascade
$\mathcal{G}^c = (V^c, E^c)$	A cascade graph which adopts the cascade c

Definition 2. A **homogeneous graph** $\mathcal{G}_{homo} = (V, E)$ is a graph in which $|\mathcal{T}^v| = |\mathcal{T}^e| = 1$. All nodes in $\mathcal{G}$ belong to a single type and all edges belong to one single type.

Definition 3. A **heterogeneous graph** $\mathcal{G}_{hete} = (V, E)$ is a graph in which $|\mathcal{T}^v| > 1$ and/or $|\mathcal{T}^e| > 1$.

Definition 4. A **knowledge graph** $\mathcal{G}_{know} = (V, E)$ is a directed graph whose nodes are *entities* and edges are *subject-property-object* triple facts. Each edge of the form (*head entity, relation, tail entity*) (denoted as $< h, r, t >$) indicates a relationship of r from entity h to entity t.

Definition 5. The **first-order proximity** between node v_i and node v_j is the weight of the edge e_{ij}, i.e., $A_{i,j}$.

Definition 6. The **second-order proximity** $s_{ij}^{(2)}$ between node v_i and v_j is a similarity between v_i's neighbourhood $s_i^{(1)}$ and v_j's neighborhood $s_j^{(1)}$.

这里着重说一下, 一阶近似和二阶近似.

- 首先这里的近似是指, 两个点之间的关联度.
- n阶近似的计算方法 : s_{ij}^n

  $s_{ij}^n = f(s_i^{n-1}, s_j^{n-1})$

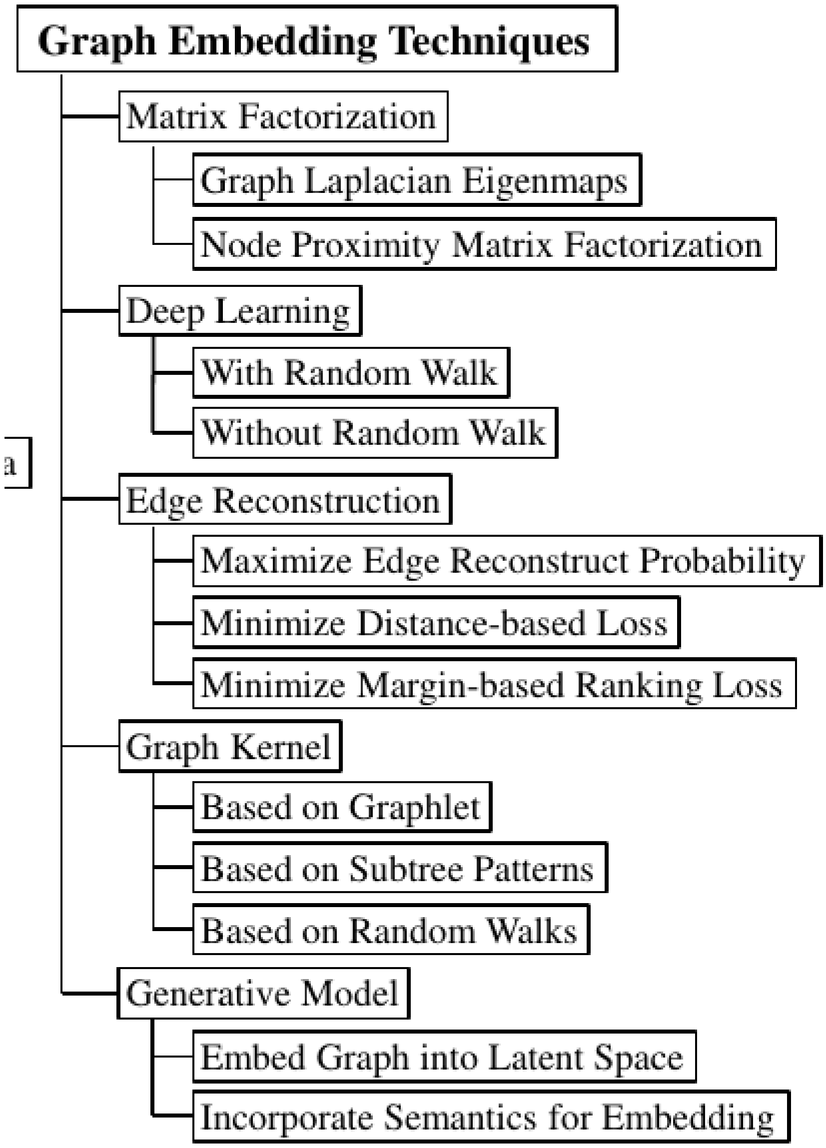
  其中, $s_i^n = [s_{i1}^n, s_{i2}^n, \ldots, s_{ik}^n]$, 这个维度就是node数减一. f 是相似函数.
- 一阶中, s_{ij}^1 就是两个点之间的weight.

3. PROBLEM SETTINGS OF GRAPH EMBEDDING

本节不介绍, 就是讲了几种不同的输入和输出, 空想都想得到, 这写了三页.

4. GRAPH EMBEDDING TECHNIQUES

4.1 方法概览



4.2 Matrix Factorization

矩阵分解方法, 这里的通常处理的是由特性进行总结出来的图结构数据, 也就是说图中的点都可以分解为多个属性值的组合. 比如说在另外的笔记中提到的演员的例子.

4.2.1 Graph Laplacian Eigenmaps

- **Insight**

  如果两个相似的点的Embedding相距较远的话, 会给损失函数大的惩罚.

这里用到的是谱聚类的思想, 具体见博客 以及我基于这个的自己的总结笔记[谱聚类]

简单来说,谱聚类是以图切分为目的, 最后求出了一般的Node Embedding, 然后又通过K-means进行聚类的过程, 但是我们其实可以只去前一部分即可.

- **具体过程**

  Input

  1. 一个点集 E, 每个点有一定的特征. 这里的特征可以是各式各样的, 例如:

     1) 对于高维空间上的聚类问题：特征可以是空间位置

     2) 对于图的切分问题：特征可以是与其他点的连接的权重.

     3) 对于知识图谱：每个点的特征与其他点的连接：$\{(e, r_i, e_j)\}_{i\in R, j\in E}$, 也可以是从文本中提取的特征

  2. 一个计算点之间距离的函数 $\rightarrow W$.

     这个可以采用多种方式, 既可以直接利用input中已有的边数据(如果给出的话). 也可以利用KNN, 高斯核等等的方法. 利用这个函数和上面的点集可以计算出邻接矩阵 W

  3. 目标函数(模型)

     1) 基本函数:

     $$y^* = \arg\min_{y^TDy=1} y^TLy = \arg\min \frac{y^TLy}{y^TDy} = \arg\max \frac{y^TWy}{y^TDy}$$

     2) 改进：上面的这个模型只能处理出现过的点, 针对没有出现的这里采用了特征向量的思想, 即, $y = X^Ta$. 其中, a 是关于这个点的特征(这个必须给出) . 那么对于一个新的点, 因为知道的它的特征, 再乘以训练得出来的矩阵 X 就好. 即将y替换为 X^Ta. 目标函数为:

     $$a^* = \arg\min \sum_{i\neq j} \|a^TX_i - a^TX_j\|^2 W_{ij} = \arg\min a^TXLX^Ta$$

     3) 其他等等等等

  Output

node向量.

- **各种各样的模型**

  就是在计算距离的函数和目标函数的不同上. 如下图:

TABLE 4
Graph Laplacian eigenmaps based graph embedding.

GE Algorithm	W	Objective Function
MDS [74]	W_{ij} = Euclidean distance (X_i, X_j)	Eq. 2
Isomap [78]	KNN, W_{ij} is the sum of edge weights along the shortest path between v_i and v_j	Eq. 2
LE [96]	KNN, $W_{ij} = \exp(\frac{\lVert X_i - X_j \rVert^2}{2t^2})$	Eq. 2
LPP [97]	KNN, $W_{ij} = \exp(\frac{\lVert X_i - X_j \rVert^2}{t})$	Eq. 4
AgLPP [79]	anchor graph, $W = Z\Lambda^{-1}Z^T, \Lambda_{kk} = \sum Z_{ik}, Z_{ik} = \frac{K_\sigma(X_i, U_k)}{\sum_j K_\sigma(X_i, U_j)}$	$a^* = \arg\min \frac{a^T ULU^T a}{a^T UDU^T a}$
LGRM [98]	KNN, $W_{ij} = \exp(\frac{\lVert X_i - X_j \rVert^2}{2t^2})$	$y^* = \arg\min \frac{y^T(L_{le} + \mu L_g)y}{y^T y}$
ARE [88]	KNN, $W_{ij} = \exp(\frac{-\rho^2(X_i X_j)}{t}), W_{ij}^{ARE} = \begin{cases} -\gamma, X_i \in F^+ \& X_j \in F^+ \\ 1, \mathcal{L}(X_i) \neq \mathcal{L}(X_j) \\ 0, otherwise \end{cases}$ F^+ denotes the images relevant to a query, γ controls the unbalanced feedback	$a^* = \arg\max \frac{a^T XL^{ARE}X^T a}{a^T XLX^T a}$
SR [99]	KNN, $W_{ij} = \begin{cases} 1, \mathcal{L}(X_i) = \mathcal{L}(X_j) \\ 0, \mathcal{L}(X_i) \neq \mathcal{L}(X_j) \\ W_{ij}, otherwise \end{cases} \quad W_{ij}^{SR} = \begin{cases} 1/l_r, \mathcal{L}(X_i) = \mathcal{L}(X_j) = C_r \\ 0, otherwise \end{cases}$ C_r is the r-th class, $l_r = \vert X_i \in X : \mathcal{L}(X_i) = C_r \vert$	$a^* = \arg\max \frac{a^T XW^{SR}X^T a}{a^T X(D^{SR} + L)X^T a}$
HSL [87]	$S = I - L$, where L is normalized hypergraph Laplacian	$a^* = \arg\max tr(a^T XSX^T a)$, s.t. $a^T XX^T a = I_k$
MVU [100]	KNN, $W^* = \arg\max tr(W)$, s.t. $W \geq 0, \sum_{ij} W_{ij} = 0$ and $\forall i, j$, where $W_{ii} - 2W_{ij} + W_{jj} = \lVert X_i - X_j \rVert^2$	Eq. 2
SLE [86]	KNN, $W_{ij} = \begin{cases} 1/l_r, \mathcal{L}(X_i) = \mathcal{L}(X_j) = C_r \\ -1, \mathcal{L}(X_i) \neq \mathcal{L}(X_j) \end{cases}$ C_r is the r-th class, $l_r = \vert X_i \in X : \mathcal{L}(X_i) = C_r \vert$	Eq. 4
NSHLRR [76]	normal graph: KNN, $W_{ij} = 1$	Eq. 2
	hypergraph: $W(\mathbf{e})$ is the weight of a hyperedge $\mathbf{e}$ $h(v, \mathbf{e}) = \begin{cases} 1, v \in \mathbf{e} \\ 0, otherwise \end{cases}, d(\mathbf{e}) = \sum_{v \in \mathcal{V}} h(v, e)$	$y^* = \arg\min \sum_{\mathbf{e}} \lVert y_i - y_j \rVert^2 \frac{W(\mathbf{e})}{d(\mathbf{e})}$
[77]	$W_{ij} = \begin{cases} \frac{\lVert X_i - X_{m+1} \rVert_2^2 - \lVert X_i - X_j \rVert_2^2}{k \lVert X_i - X_{k+1} \rVert_2^2 - \sum_{m=1}^{k} \lVert X_i - X_k \rVert_2^2}, j \leq k \\ 0, j > k \end{cases}$	$y^* = \arg\min_{y^T y = 1} \sum_{i \neq j} W_{ij} \min(\lVert y_i - y_j \rVert_2^p, \theta)$
PUFS [75]	KNN, $W_{ij} = \exp(-\frac{\lVert X_i - X_j \rVert^2}{2t^2})$	Eq. 4 +(must-link and cannot link constraints)
RF-Semi-NMF-PCA [101]	KNN , $W_{ij} = 1$	Eq. 2 +$\mathcal{O}$(PCA) + $\mathcal{O}$(kmeans)