

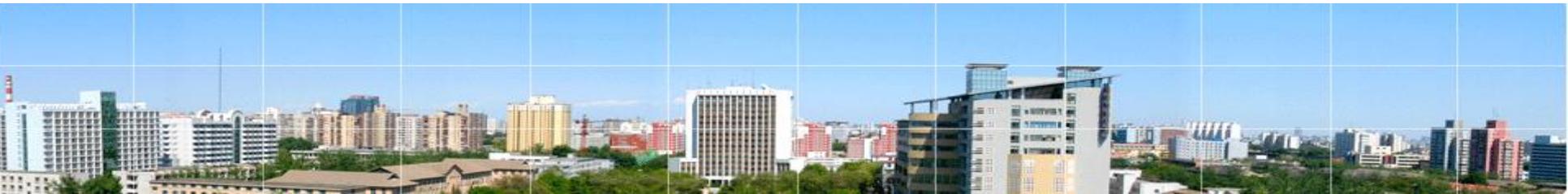


北京邮电大学
Beijing University of Posts and Telecommunications

第三章

图嵌入

石川 教授
数据科学与服务中心 计算机学院





- 课程介绍
- 讲课内容
 - 3.1 图嵌入的基本概念
 - 3.2 基于流形的图嵌入模型
 - 3.3 结构信息保持的图嵌入模型
 - 3.4 侧信息保持的图嵌入模型
- 重点难点
 - 理解图嵌入的基本概念和意义
 - 了解各种图嵌入模型算法及原理

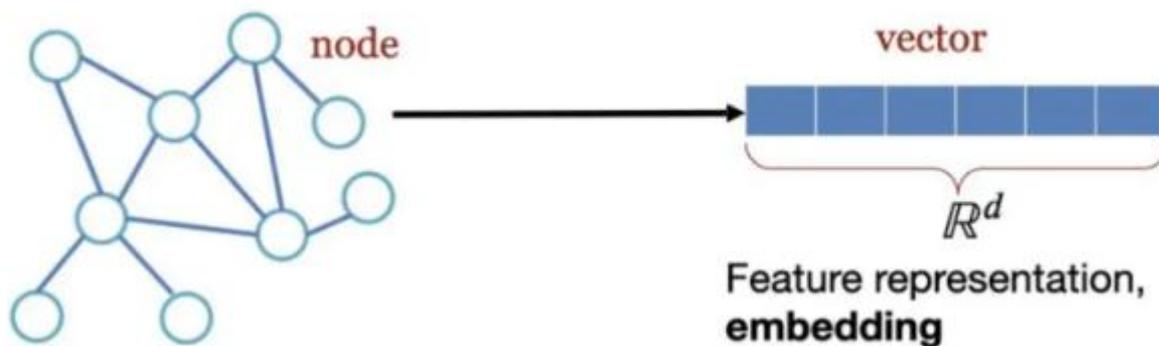


3.1 图嵌入基本概念

- 图结构数据广泛分布于多种场景中，如社交网络、生物分子结构和交通网络等。这类数据的结构复杂多样，具有很强的应用价值。
- 图数据通常具有高度稀疏的特点，导致使用传统的存储方法（如邻接矩阵）时，面临**存储效率低**的问题。
- 在大规模图结构中，传统图分析算法面临**计算和存储复杂度**高的挑战，限制了其在实际应用场景中的使用。
- 图数据的不规则性和稀疏性使得传统的**机器学习算法难以适用**，无法高效解决复杂的下游任务，进而限制了其在图数据处理中的应用效果。

3.1 图嵌入基本概念

- 为了解决上述挑战，研究人员探寻新的嵌入方法，为节点学习低维向量表示。
- 图嵌入的定义如下：给定一个图 $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ 。对每一个顶点 $v \in \mathcal{V}$ ，构建一个低维表示 $z_v \in R^d$ ，其中 d 是表示空间的维度，并且通常远小于 $|\mathcal{V}|$ 。

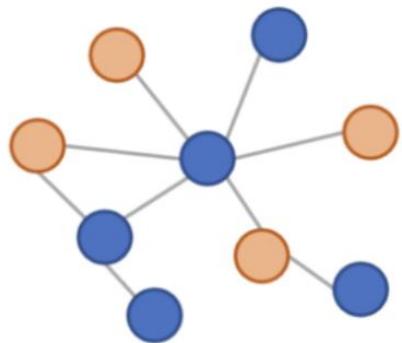


图嵌入示意图

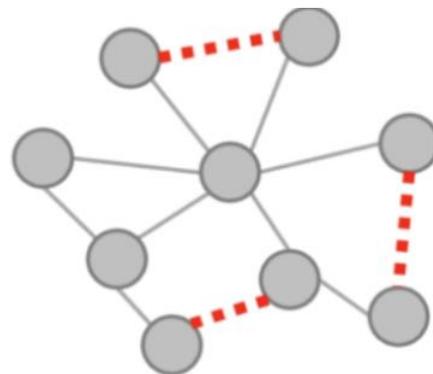
3.1 图嵌入基本概念



- 图嵌入的优势
 - 节点降维处理：将图中节点映射到低维嵌入空间中，节点的低维嵌入表示可以反映其在原图的结构特性。
 - 编码复杂信息：图嵌入可以将图结构特性、属性特征和标签类别等丰富信息统一编码到节点嵌入表示中。
 - 支持下游任务：得到的节点嵌入可以被机器学习方法应用，从而解决多种网络分析任务，如节点分类，链接预测，聚类等。



节点分类



链接预测

3.1 图嵌入基本概念

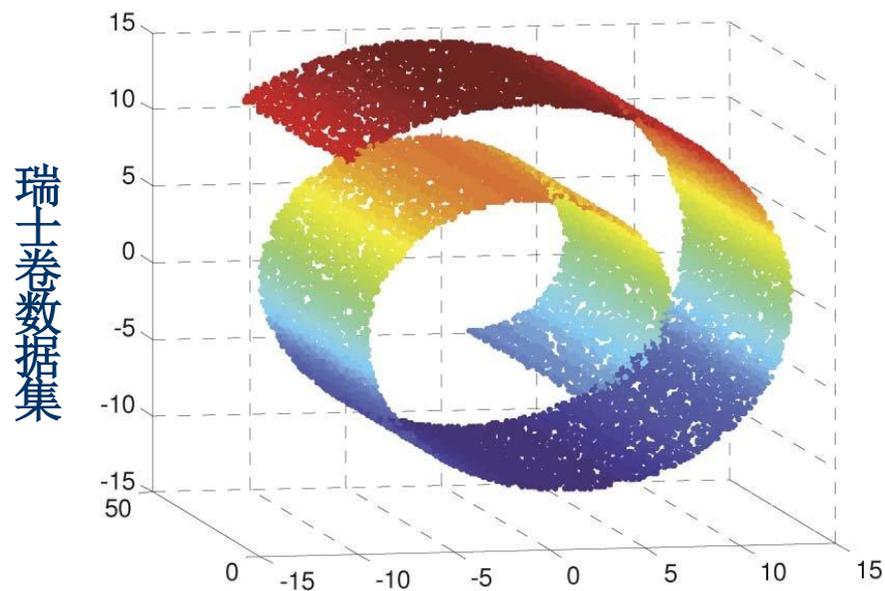


- 图嵌入模型可以被划分为：基于流形的图嵌入模型，结构信息保持的图嵌入模型，侧信息保持的图嵌入模型。
 - 基于流形的图嵌入模型
 - 包含基于低维流形上测地距离的图嵌入方法，如Isomap、LLE、LE。
 - 结构信息保持的图嵌入模型
 - 包含邻域信息保持的DeepWalk、结构角色保持的Struct2Vec、社区信息保持的M-NMF、全局信息保持的DP等。
 - 侧信息保持的图嵌入模型
 - 包含节点属性信息保持的TADW、标签信息保持的MMDW、多种信息保持的Planetoid等。



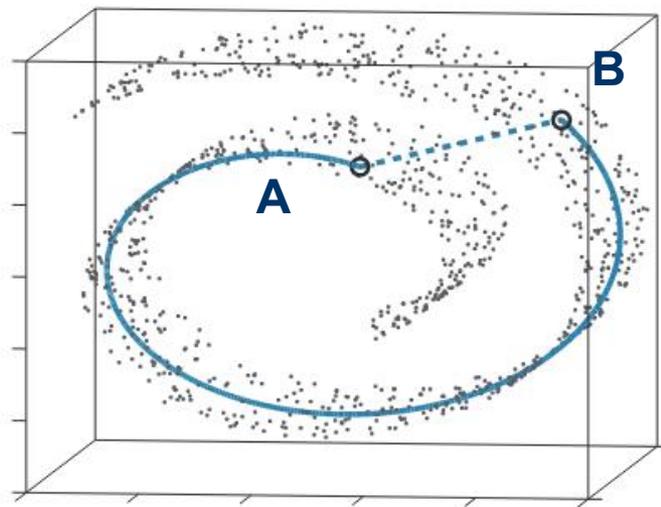
3.2 基于流形的图嵌入模型

- 传统线性降维方法，如主成分分析（PCA）和多维缩放（MDS），通过在高维输入空间中基于数据的线性子空间距离进行降维，使得相近的数据在低维嵌入中仍保持相对接近。然而，对于许多数据集（如瑞士卷数据集），数据点往往分布在高维空间的非线性低维流形上，这时传统方法难以有效降维。



3.2 基于流形的图嵌入模型

- 非线性带来的挑战如图所示。在瑞士卷数据集中，两个点 **A** 和 **B** 在高维空间中的线性距离非常接近（虚线表示）。然而，它们实际上位于瑞士卷型的低维流形上，在该流形上它们之间的真实距离（实线表示）却相距甚远。因此，为了有效降维，需要准确测量数据点在流形上的真实距离（即测地距离）。

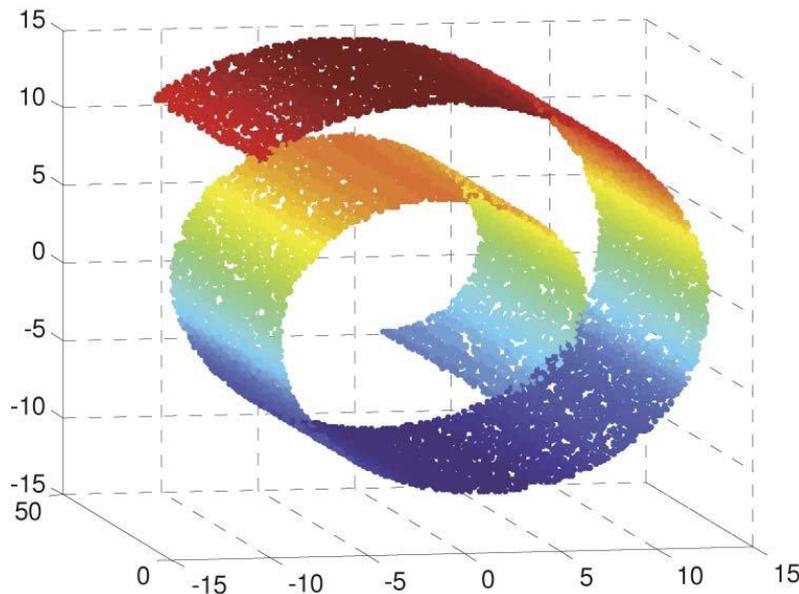


非线性数据集上两个点

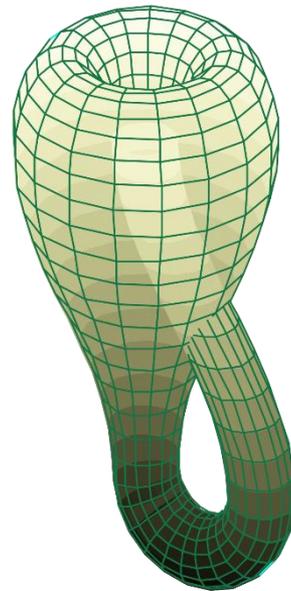
3.2 基于流形的图嵌入模型

- **流形 (manifold)** 是一个拓扑空间，局部上可近似为欧几里得空间，但整体上可能具有复杂的非线性结构。例如，瑞士卷数据集中的数据虽然分布在三维空间，但实际上位于一个二维的瑞士卷型平面上，这个平面就是一个流形。类似地，克莱因瓶的表面也可视为三维空间中的一个低维流形。

瑞士卷数据集



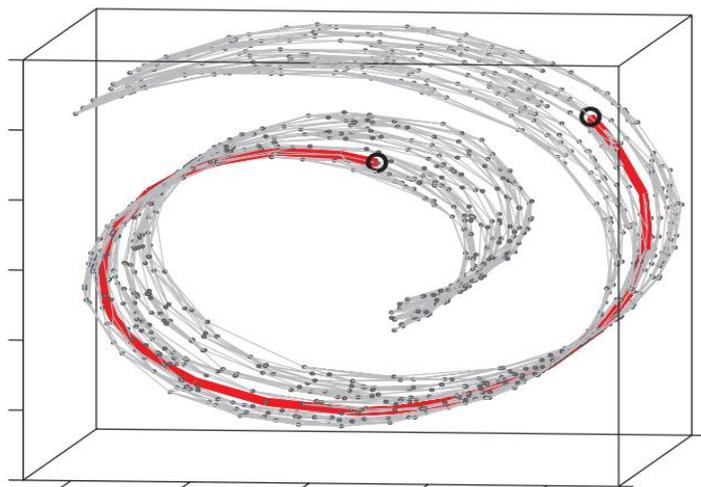
克莱因瓶



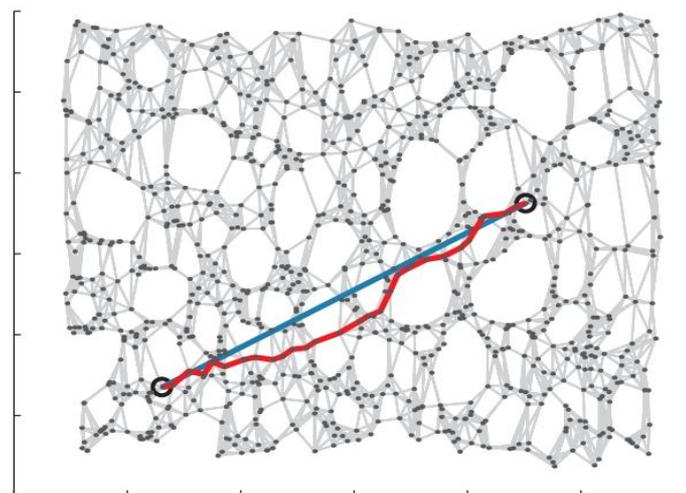
3.2 基于流形的图嵌入模型



- **测地距离 (geodesic distance)** 是指在流形上两点之间的最短距离。它类似于平面几何中两点之间的直线距离，但在流形中，路径通常是曲线的。测地距离可以通过寻找最短路径（称为测地线）来近似计算。



通过最短路径近似
计算测地距离



测地距离 (蓝)
近似测地距离 (红)



3.2 基于流形的图嵌入模型

- 利用测地距离，可以将数据构建为图从而保持流形结构。
- 图嵌入模型最早利用流形学习中的测地距离来应对非线性带来的挑战，从而有效地对非线性数据集进行降维。
- 经典的基于流形的图嵌入算法包括：
 - 等距特征映射（Isomap）
 - 局部线性嵌入（LLE）
 - 拉普拉斯特征映射（LE）

3.2 基于流形的图嵌入模型



- **Isomap**（即等距特征映射）的核心思想是保持数据点在低维流形上的测地距离，可以视为对多维尺度分析（**MDS**）的一个扩展。它的算法流程分为三步：
 - 根据预先定义的距离函数（欧式距离或邻域特定的距离函数），通过**K**近邻等算法决定每个数据点的直接邻居。
 - 通过最短路径距离（通常采用 **Dijkstra** 算法或者 **Floyd** 算法）估计任意两点之间的测地距离，从而得到测地距离矩阵**A**，其中 a_{ij} 表示点 v_i 和 v_j 之间的测地距离。
 - 通过传统**MDS**方法将测地距离映射到低维空间。



3.2 基于流形的图嵌入模型

- **MDS**是一种用于数据降维的技术，其目标是将高维空间中的数据点映射到低维空间中，同时保留每对样本之间的距离不变。
- 在得到点间距离矩阵 **A** 后，使用**MDS**的目的是得到降维到**d**维后的表示矩阵 **Z**。具体过程如下：
 - 双中心化距离矩阵 **A**，将 **A** 转化矩阵 **B**：

$$B = -\frac{1}{2}HA^2H$$

其中 $H = I - \frac{1}{n}J_n$ 是一个中心化矩阵， J_n 是一个全 1 矩阵。



3.2 基于流形的图嵌入模型

- 对内积矩阵 \mathbf{B} 进行特征值分解，得到特征值和特征向量：

$$\mathbf{B} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$$

其中 \mathbf{V} 是特征向量矩阵， $\mathbf{\Lambda}$ 是对角特征值矩阵。

- 选择前 d 个最大的特征值和对应的特征向量，令 $\mathbf{\Lambda}_d$ 为包含前 d 个特征值的对角矩阵， \mathbf{V}_d 为相应的特征向量矩阵，低维矩阵 \mathbf{Z} 可以通过以下公式得到：

$$\mathbf{Z} = \mathbf{V}_d\mathbf{\Lambda}_d^{1/2}$$

- Isomap 相对于 MDS 的改进就是在于使用测地距离构造点之间的距离矩阵。



3.2 基于流形的图嵌入模型

- **LLE**（局部线性嵌入）起源于一种基本假设：数据点的局部信息往往更为关键。在某些任务中，聚焦局部信息并适当忽略远距离的数据点通常能取得更好的效果。
- **LLE**假设在较小的局部内，数据表现为线性可重构的。即一个节点可以通过其邻近点的线性组合来近似表示。给定顶点 v_i 及其 k 个邻居点 $\{v_1, v_2, \dots, v_k\}$ ，顶点的特征矩阵为 \mathbf{X} ，顶点 v_i 的特征可以由其邻居节点的特征近似重构：

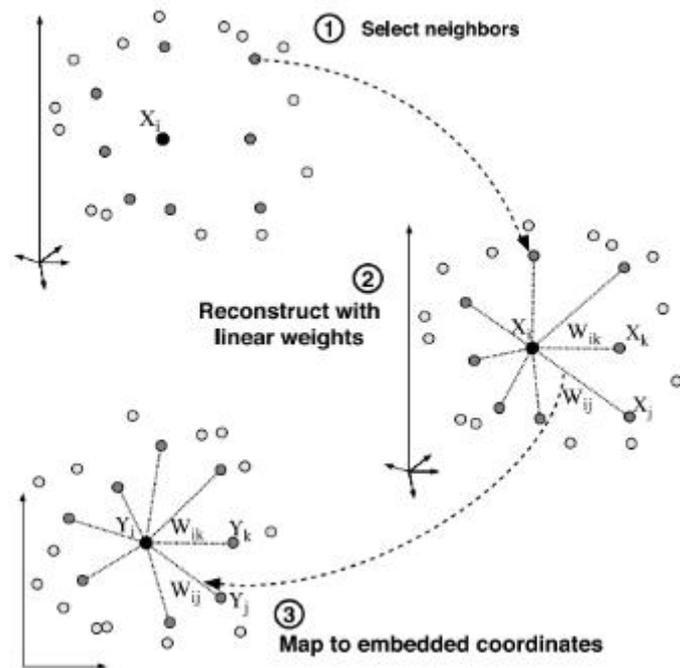
$$X_i \approx \sum_{j=1}^k W_{ij} X_j$$

3.2 基于流形的图嵌入模型

LLE的算法流程如下：

- 邻居选择
 - 基于流形学习，为每个顶点 v_i 分配邻居。
- 权重计算
 - 计算权重 W_{ij} ，优化目标是使得 v_i 与邻居 v_j 之间的距离平方和最小，即：

$$\min_w \sum_{i=1}^N \|X_i - \sum_{j=1}^k W_{ij} X_j\|^2$$



LLE算法示意图

- 低维嵌入
 - 降维后，权重 W_{ij} 在低维中保持不变。
 - 优化函数为： $\min_Z \sum_{i=1}^N \|Z_i - \sum_{j=1}^k W_{ij} Z_j\|^2$ 。



3.2 基于流形的图嵌入模型

- **LE**（拉普拉斯特征映射）与**LLE**类似，也是一种保持数据局部结构信息的算法。**LE**通过图的拉普拉斯矩阵保持图的局部邻接性质。
- **LE**的算法流程如下：
 - 构建图：基于流形学习分配邻居（**KNN**等）。
 - 计算权值矩阵： $w_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$ 。
 - 计算拉普拉斯矩阵： $L = D - W$ 。
 - 图嵌入： $\min_Z \sum_{i,j} w_{ij} \|z_i - z_j\|^2$ ，该目标反映了越相邻的节点其嵌入也越相似。可以通过拉普拉斯矩阵计算其全局最优解。

3.3 结构信息保持的图嵌入模型



- 随着大数据时代的发展，收集到的网络数据规模不断扩大，其结构信息也变得愈加复杂和丰富。
 - 如何从大规模网络数据中高效挖掘有效信息并学习嵌入表示？
 - 如何将网络中复杂丰富的结构信息保持到节点的嵌入表示中？
- 图的结构信息可以分成四类：
 - 邻域信息：即一阶、二阶和高阶邻近性。
 - 结构角色：节点在图中角色或功能的相似性。
 - 社区信息：同一社区内部节点表现出相似性。
 - 全局信息：图的一些全局特性，如度的长尾分布等。

3.3.1 邻域信息保持的图嵌入模型



- 邻域信息主要指节点间的一阶邻近、一阶邻近和高阶邻近
 - 一阶邻近 (First-order proximity)描述了两个连通顶点之间的局部成对临近度。
 - 在无权图中, 如果两个顶点 v_i 和 v_j 之间存在边, 则一阶临近度 w_{ij} 定义为

:

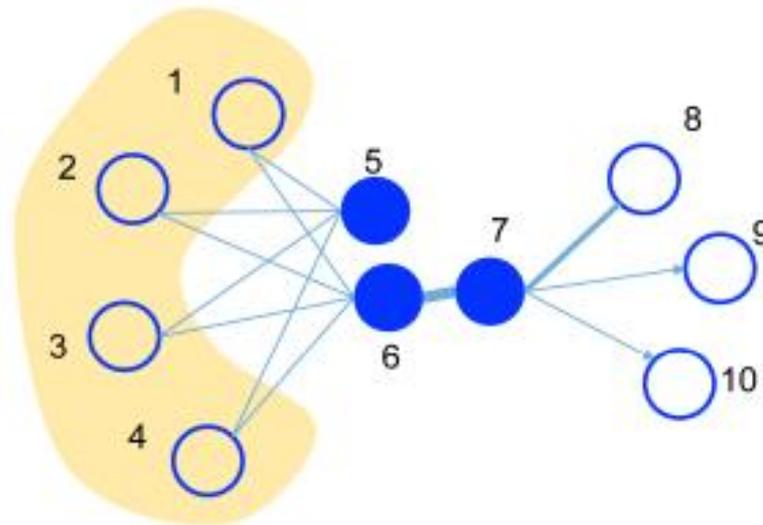
$$w_{ij} = \begin{cases} 1 & \text{如果 } (v_i, v_j) \in \mathcal{E}, \\ 0 & \text{如果 } (v_i, v_j) \notin \mathcal{E}. \end{cases}$$

- 在加权图中, 一阶临近度可以直接用边的权重值表示, 即:

$$w_{ij} = \text{边 } (v_i, v_j) \text{ 的权重}$$

3.3.1 邻域信息保持的图嵌入模型

- 二阶邻近（Second-order proximity）不仅考虑节点之间是否直接相连，还考虑两个节点共享邻居的数量和质量。
- 高阶邻近（High-order proximity）进一步扩展二阶邻近的概念，捕获更多的全局结构，探索顶点之间的多步关系。



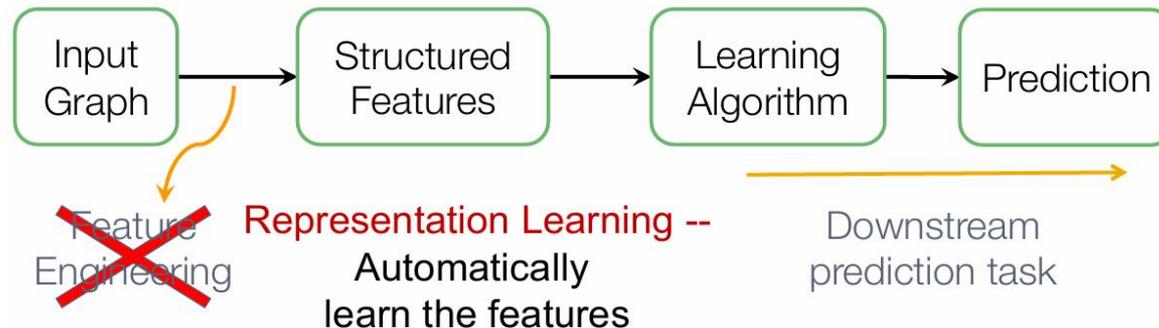
节点5与节点6二阶邻近



3.3.1 邻域信息保持的图嵌入模型

● DeepWalk

- 网络结构的稀疏性使得图数据难以泛化到统计学习中，无法被机器学习算法有效应用。
- DeepWalk 算法通过无监督学习方式为图中每个节点学习一个嵌入向量，把深度学习引入到了图的领域中。
- 以往要对图做下游任务之前，需要人为地对图做一系列特征工程，而 DeepWalk 这种图嵌入算法可以自动的学习节点信息并嵌入成向量形式。

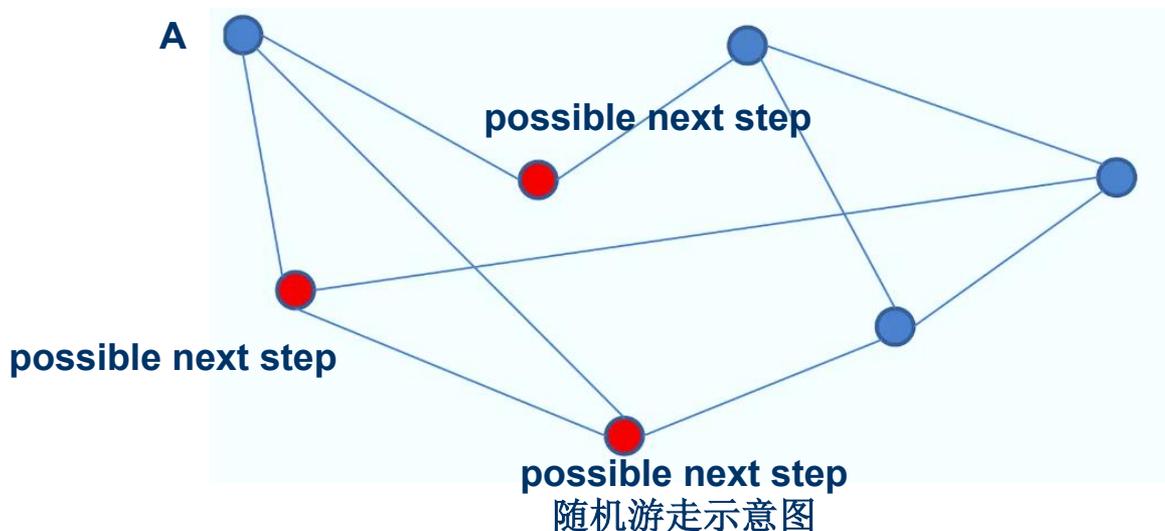


Bryan Perozzi, et al. "DeepWalk: Online Learning of Social Representations." KDD, 2014.

3.3.1 邻域信息保持的图嵌入模型



- DeepWalk 基于随机游走捕获图结构信息
 - 给定图 $G = \{V, E\}$, 随机游走可以表示为一个随机变量序列 v_0, v_1, v_2, \dots , 其中 v_0 是初始节点, 而每个 v_{i+1} 是从 v_i 的邻居节点中随机选取的。
 - 通过随机游走探索到的节点集合反映了该节点在图中的局部邻域结构。



3.3.1 邻域信息保持的图嵌入模型



- DeepWalk: 随机游走+skip-gram
 - 首先采用随机游走算法为每个节点采样一条随机游走序列，该序列能够有效刻画节点的邻域信息。
 - 将该序列作为 skip-gram 模型的输入，并通过预测窗口内的其他节点来进行节点嵌入向量的学习，概率预测的实现使用 SoftMax 函数：

$$\Pr(v_j|z_i) = \frac{\exp(z_i \cdot z_j)}{\sum_{k \in \mathcal{V}} \exp(z_i \cdot z_k)}$$

- 令w代表窗口大小，其目标函数如下：

$$\min_{z_i} - \log \Pr(\{v_{i-w}, \dots, v_{i-1}, v_{i+1}, \dots, v_{i+w}\} | z_i)$$

3.3.1 邻域信息保持的图嵌入模型

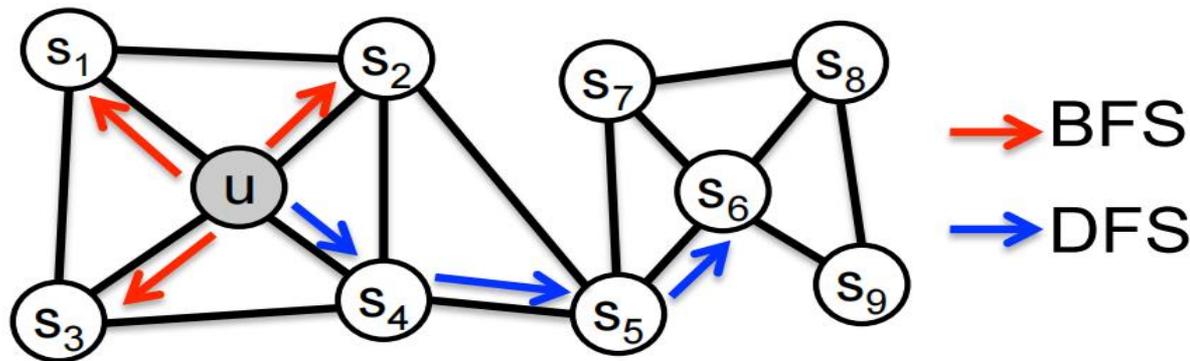


- DeepWalk算法流程
 - 输入：图 $G = \{V, E\}$ ，窗口大小 w ，嵌入维度 d ，每个节点的游走次数 γ ，游走序列长度 t 。
 - 先把节点集合 V 打乱成节点集合 O ，对于集合 O 中的每一个节点 v ，生成长度为 t 的随机游走序列。
 - 依次把生成的随机游走序列输入skip-gram模型。
 - 在skip-gram模型中，在要求的窗口大小 w 下做预测，并计算损失函数，然后使用梯度下降算法更新对应节点的嵌入向量。
 - 不断重复上面三步，直到每个节点游走完指定次数 γ 。

3.3.1 邻域信息保持的图嵌入模型



- 基于随机游走的图嵌入模型利用不同的随机游走策略来捕获图的各种结构特性，如同配性和结构等价性。
 - 同配性（**Homophily**）：高度互连且属于相似网络社区的节点应嵌入在相近位置（如图中的 s_1 与 s_2 ），可通过深度优先游走策略（**DFS**）来捕获。**DeepWalk** 采用随机游走，在一定程度上可以捕获同配性。
 - 结构等价性（**Structural equivalence**）：在网络中具有相似结构角色的节点应嵌入在相近位置（如图中的 u 与 s_6 ），可通过





3.3.1 邻域信息保持的图嵌入模型

● Node2vec

- 通过定义一种有偏的二阶随机游走方法，实现了BFS和DFS的灵活组合，可以同时捕获节点的同配性和结构等价性。
- 二阶随机游走
 - 在进行随机游走时，序列中的下一节点由当前节点和上一节点共同决定。
 - 形式上，给定一个网络 $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ 和一个随机游走序列。该序列的第 i 个节点为 u_i 。从源节点 u_0 开始，节点 u_i 由下列分布生成：

$$P(u_i = x | u_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{Z} & \text{if } (v, x) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$

π_{vx} 是节点 v 和 x 之间的未归一化转移概率， Z 是归一化常数。



3.3.1 邻域信息保持的图嵌入模型

- Node2Vec 随机游走策略

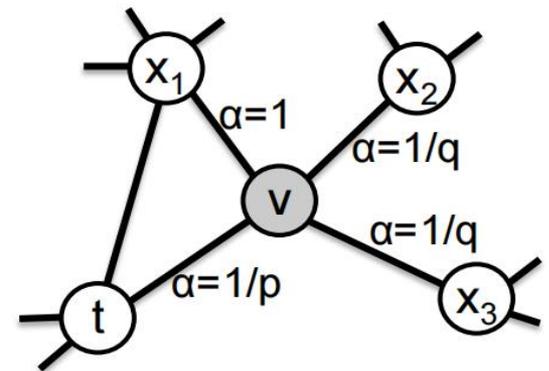
- Node2vec中 π_{vx} 由上一节点 t 和下一节点 x 的最短距离 r_{tx} 来决定。

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

- ω_{vx} 为边 (v, x) 的权重，这里默认为1。

- 如下图所示，node2vec方法中将进行有偏的随机游走， α 为转移概率， p 为返回参数、 q 为向外参数， p 越小返回的概率就大， q 越小向外的概率就越大。

- 这种结合 BFS 和 DFS 策略的随机游走方式，使得 Node2Vec 能够灵活调整对同配性和结构等价性的捕获倾向。

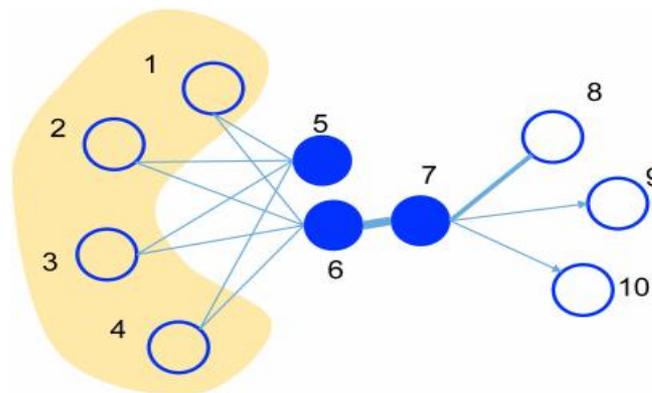


3.3.1 邻域信息保持的图嵌入模型



- **LINE**

- DeepWalk 和 Node2Vec 算法通过随机游走来捕捉节点的邻域信息，从而学习网络结构。而 **LINE** 算法则通过显式建模一阶和二阶邻近性来生成节点嵌入。
- 它能够适用于各种类型的网络，包括无向图、有向图、无权图和有权图。
- 网络中的一阶邻近性是指两个节点有一对边相连，二阶邻近性则是两个节点的邻域结构有许多相似的地方。





3.3.1 邻域信息保持的图嵌入模型

- 一阶邻近性建模

- 对于每个无向边 (v_i, v_j) ，定义节点 v_i 和 v_j 之间的联合概率

$$p_1(v_i, v_j) = \frac{1}{1 + \exp(-z_i^T \cdot z_j)}, \text{ 其经验概率可以定义为 } \widehat{p}_1(i, j) = \frac{w_{ij}}{\sum_{(i,j) \in \mathcal{E}} w_{ij}}, \text{ 其中 } \mathcal{E} \text{ 为网络的边的集合, } w_{ij} \text{ 为边权重。}$$

- 为了保持一阶邻近性，我们采用以下的优化目标：

$$O_1 = d(\widehat{p}_1(\cdot, \cdot), p_1(\cdot, \cdot))$$

这里 $d(\cdot, \cdot)$ 表示两个分布之间的距离。

- 采用 KL 散度作为距离函数，并去掉一些常数可以得到简化的目标函数：

$$O_1 = - \sum_{(i,j) \in \mathcal{E}} w_{ij} \log p_1(v_i, v_j)$$

3.3.1 邻域信息保持的图嵌入模型



- 二阶邻近性建模
 - 在网络中每个节点都有二重身份：节点本身和其他节点的邻居，这里定义两个向量 z_i 和 z'_i ：
 - z_i 是将 v_i 当成节点本身时的嵌入向量。
 - z'_i 是将 v_i 当成其他节点邻居时的嵌入表示。
 - 对于每条有向边 (v_i, v_j) ，定义节点 v_i 生成邻居 v_j 的概率如下：

$$p_2(v_j|v_i) = \frac{\exp(z_i^T \cdot z'_j)}{\sum_{k=1}^{|V|} \exp(z_i^T \cdot z'_k)}$$

其中 $|V|$ 是网络中的节点数量。



3.3.1 邻域信息保持的图嵌入模型

- 二阶邻近度的经验分布 $\hat{p}_2(v_j|v_i) = \frac{w_{ij}}{d_i}$ ，其中 $d_i = \sum_{k \in N(i)} w_{ik}$ 。
- 二阶邻近度建模，要优化下面的目标函数：

$$O_2 = \sum_{i \in \mathcal{V}} \lambda_i d(\hat{p}_2(\cdot|v_i), p_2(\cdot|v_i))$$

- 令 $\lambda_i = d_i$ ，并采用KL散度作为距离函数，得到如下目标函数：

$$O_2 = - \sum_{(i,j) \in \mathcal{E}} w_{ij} \log p_2(v_j|v_i)$$

- 分别训练这两个模型，然后把每个节点训练得到的两种嵌入向量拼接起来就可以同时保持一阶邻近性和二阶邻近性。

3.3.1 邻域信息保持的图嵌入模型

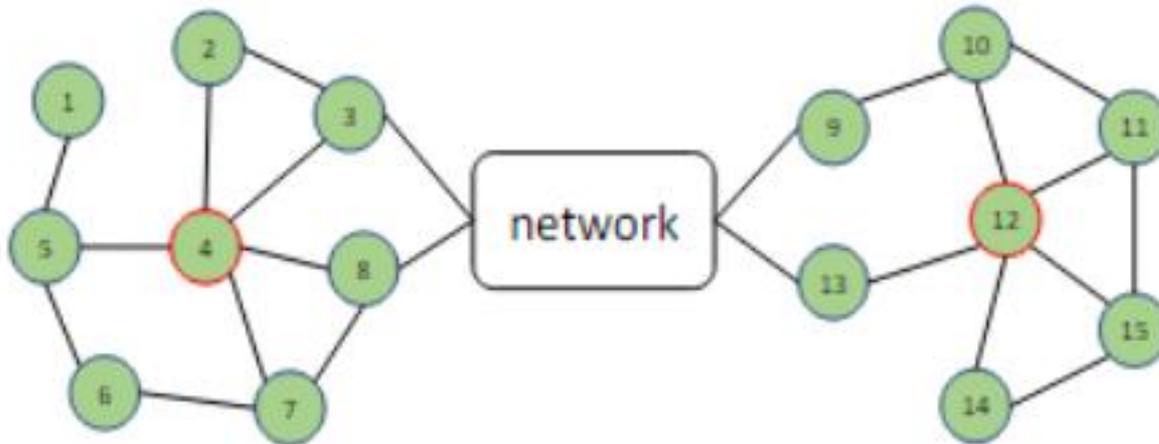


- DeepWalk, Node2vec 和 LINE 的比较

算法	一阶邻近性	二阶邻近性	高阶邻近性	所属类别
Deepwalk		√	√	随机游走
Node2vec		√	√	随机游走
LINE	√	√		显式建模

3.3.2 结构角色保持的图嵌入模型

- **结构角色相似性 (Structural Role Proximity)**关注于图中节点的角色或功能的相似性，而不仅仅是节点之间的直接相连。节点的角色通常由其结构位置决定。例如，一个节点可以是网络中的中心节点，桥接节点或叶子节点。



节点4与节点12结构相似



- **Struct2vec**

- 与 Node2Vec 类似，Struct2Vec 也是基于有偏随机游走的方法，不过 Struct2Vec 会基于结构相似度重新构图并在重构的图上进行有偏随机游走。
 - 重构的图为一个层次化的带权图。
 - 在当前的层内以及层与层之间会有不同的随机游走转移概率。

- 度量结构角色相似度

- Struct2Vec 方法中，如果两个节点的节点度越接近那么这两个节点就越相似。进一步的，如果两个节点的邻居节点的度分布也接近，那么这两个节点就更相似。

3.3.2 结构角色保持的图嵌入模型



- 构图

- 符号定义:

- $R_k(x)$ 表示与节点 x 距离为 k 的节点集合
- $s(S)$ 表示节点集合 S 的有序度序列
- $f_k(u, v)$ 表示考虑 k 跳邻居时, 节点 u 和 v 结构角色的差异度
- $g(D_1, D_2)$ 表示两个有序度序列 D_1 和 D_2 的距离

- 通过如下公式衡量两个节点的结构角色差异度:

$$f_{-1} = 0$$

$$f_k(u, v) = f_{k-1}(u, v) + g\left(s(R_k(u)), s(R_k(v))\right)$$

$$k \geq 0 \text{ and } |R_k(u)|, |R_k(v)| > 0$$

3.3.2 结构角色保持的图嵌入模型



- 令重构后的图为 M ，图 M 的第 k 层是在考虑 k 跳邻居时，根据角色差异度 f_k 构建的无向带权完全图。

- 定义在第 k 层中两个顶点 u 和 v 的边权重 d_k 如下：

$$d_k(u, v) = e^{-f_k(u, v)}, k = 0, \dots, k^*$$

- 层与层之间每个顶点都会和其对应的上下层顶点相连，若当前节点为 v ，定义层与层之间的边权重如下：

$$d(v_k, v_{k+1}) = \log(\Gamma_k(v) + e), k = 0, \dots, k^* - 1$$

$$d(v_k, v_{k-1}) = 1$$

$\Gamma_k(v)$ 是第 k 层与 v 相连的边中，边权重大于该层平均边权重的边的数量。



3.3.2 结构角色保持的图嵌入模型

- 有偏随机游走
 - 如果当前游走到了第 k 层，在同一层中，假设当前节点为 u ，那么下一个节点为 v 的概率为：

$$p_k(u, v) = \frac{e^{-f_k(u, v)}}{\sum_{v \in V, v \neq u} e^{-f_k(u, v)}}$$

- 如果当前节点要切换到第 $k - 1$ 或第 $k + 1$ 层，其概率如下：

$$p_k(u_k, u_{k+1}) = \frac{d(u_k, u_{k+1})}{d(u_k, u_{k+1}) + d(u_k, u_{k-1})}$$

$$p_k(u_k, u_{k-1}) = 1 - p_k(u_k, u_{k+1})$$

- 训练：把随机游走序列输入skip-gram模型。

3.3.2 结构角色保持的图嵌入模型



- **GraphWave**

- GraphWave 是一种基于矩阵分解的无监督的节点嵌入学习方法。它通过以节点为中心的谱图小波的扩散，来学习无向图上对应节点的嵌入向量。

- 拉普拉斯矩阵

- 假定有个无向图 $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ ，令 $|\mathcal{V}| = N$ ，并定义一个度矩阵 \mathbf{D} ，其元素 $D_{ii} = \sum_j A_{ij}$ ，其余元素为0。
- 则有拉普拉斯矩阵 $\mathbf{L} = \mathbf{D} - \mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ ，其中 $\mathbf{\Lambda} = \text{Diag}(\lambda_1, \dots, \lambda_N)$ ， $\lambda_1 < \lambda_2 \leq \dots \leq \lambda_N$ 为 \mathbf{L} 特征值。

3.3.2 结构角色保持的图嵌入模型

- 热核函数与谱图小波
 - 热核函数形式如下（其中 s 为放缩参数）：

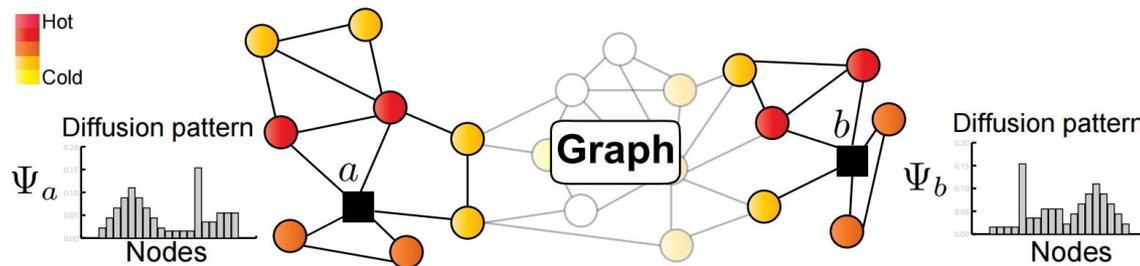
$$g_s(\lambda) = e^{-\lambda s}$$

- 根据热核函数定义谱图小波如下：

$$\Psi = U \text{Diag}(g_s(\lambda_1), \dots, g_s(\lambda_N))U^T, \Psi_a = \Psi \delta_a$$

δ_a 是一个 **one-hot** 向量，用提取相应节点的谱图小波。 Ψ 是存储了网络中所有节点能量信息的矩阵。

- 结构角色相似的节点拥有相似的谱图小波分布。



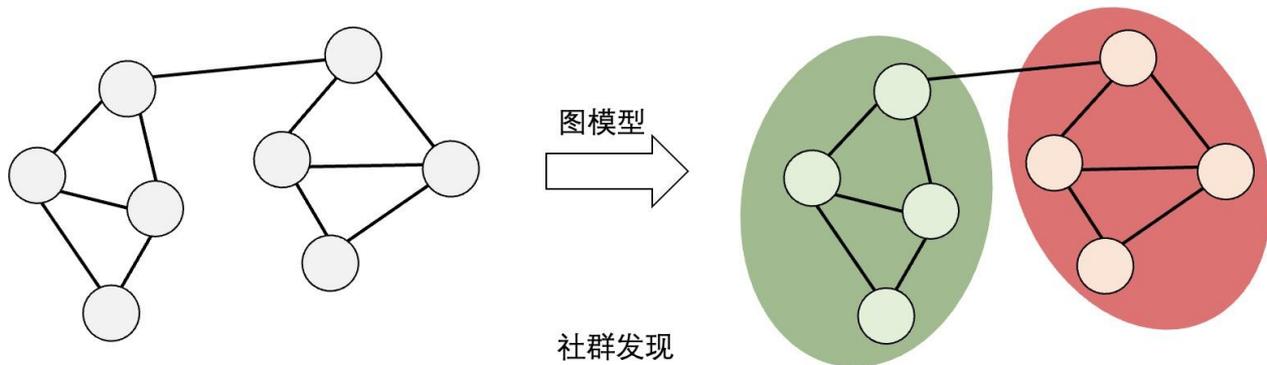


3.3.2 结构角色保持的图嵌入模型

- Graphwave算法
 - 计算每个节点的谱图小波 Ψ_a 。
 - 定义概率分布 X 的特征函数为 $\Phi_X(t) = E[e^{itX}]$, $t \in \mathbb{R}$, 则节点 a 的谱图小波 Ψ_a 的特征函数定义为 $\Phi_a(t) = \frac{1}{N} \sum_{m=1}^N e^{it\Psi_m a}$ 。
 - 采样 d 个均匀间隔点 $\{t_1, t_2, \dots, t_d\}$, 把这 d 个点输入到每个节点的特征函数中。因为特征函数值为复数, 将其实部和虚部分别作为嵌入向量的元素, 最终节点的嵌入向量的长度是 $2d$ 。

3.3.3 社区信息保持的图嵌入模型

- 社区通常被定义为图中一组相互紧密连接的子图，这些子图反映了现实社会中具有共同特征或兴趣的人或对象的集合。社区内部的节点之间表现为高内聚性，而与外部社区的节点之间表现为低耦合性。
- 社区发现的目的是从图中识别出一组社区 $\mathbf{C} = \{C_1, C_2, \dots, C_k\}$ 每个社区中的内部度数显著多于其外部度数。通过社区发现，可以揭示隐藏在复杂网络连接模式背后的结构。



3.3.3 社区信息保持的图嵌入模型



- 社区检测（聚类）技术
 - 模块度最大化
 - 目的是寻找网络中具有高内聚性和低耦合性的社区结构
 - 通过最大化网络的模块度，来识别网络中的社区结构
 - 谱聚类
 - 对网络的拉普拉斯矩阵进行特征分解，将节点投影到低维空间进行聚类
 - 边聚类
 - 重点关注网络中边的特征
 - 将具有相似连接模式或特征的边分组到同一类



3.3.3 社区信息保持的图嵌入模型

- M-NMF旨在让学得的低维节点嵌入保持基于邻域信息的微观相似性和基于社区信息的宏观相似性。
- 社区建模
 - 模块度是用来衡量网络中社区结构质量的指标，反映了网络中实际边的连接情况与随机网络中期望连接情况的差异。
 - 当只有两个社区时，模块度公式如下

$$Q = \frac{1}{2e} \sum_{ij} \left[A_{ij} - \frac{d_i d_j}{2e} \right] h_i h_j$$

d_i 为节点*i*的度，若节点*i*属于第一个社区，则 $h_i = 1$ 否则 $h_i = -1$ ， $\frac{d_i d_j}{2e}$ 是当边被随机设置时，节点*i*和节点*j*之间的期望边数。

3.3.3 社区信息保持的图嵌入模型



- 模块度矩阵 \mathbf{B} :

$$\mathbf{B} \in \mathbb{R}^{n \times n}$$

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2e}$$

- 模块指示矩阵 \mathbf{H} :

$$\mathbf{H} \in \mathbb{R}^{n \times k}$$

$$\text{tr}(\mathbf{H}^T \mathbf{H}) = n$$

如果只有两个社区， \mathbf{H} 可以用模块指示向量 \mathbf{h} 替代。

- 模块度公式的矩阵形式:

$$Q = \text{tr}(\mathbf{H}^T \mathbf{B} \mathbf{H}), \text{ s.t. } \text{tr}(\mathbf{H}^T \mathbf{H}) = n$$

- 可以用这个公式对社区信息进行建模，即最大化模块度 Q 。

3.3.3 社区信息保持的图嵌入模型



- 一阶及二阶邻近性建模

- 一阶邻近性建模

- 如果两个节点之间有边相连那么就认为这两个节点是一阶邻近的，所以简单起见把图的邻接矩阵 \mathbf{A} 作为其一阶邻近性矩阵 $\mathbf{S}^{(1)}$ 。

- 二阶邻近性建模

- 对于一个节点 v_i ，定义它的邻居向量为 $T_i = (S_{i,1}^{(1)}, \dots, S_{i,n}^{(1)})$ 。
- 对于一对节点 v_i 和 v_j 的二阶邻近性矩阵由下列公式定义：

$$S_{i,j}^{(2)} = \frac{T_i \cdot T_j}{\|T_i\| \|T_j\|}$$

3.3.3 社区信息保持的图嵌入模型



- 总体优化目标
 - 对一阶、二阶邻近性做融合
 - 邻近性矩阵 \mathbf{S} :

$$\mathbf{S} = \mathbf{S}^{(1)} + \eta \mathbf{S}^{(2)}, \eta > 0$$

这里 η 是二阶邻近性矩阵的权重，默认设置为 5。

- 定义非负基矩阵: $\mathbf{M} \in \mathbb{R}^{n \times d}$ 。
- 目标函数如下:

$$\min \|\mathbf{S} - \mathbf{M}\mathbf{Z}^T\|_F^2, \text{ s.t. } \mathbf{M} \geq 0, \mathbf{Z} \geq 0$$

- 融合社区信息
 - 辅助社区表示矩阵 $\mathbf{C} \in \mathbb{R}^{k \times d}$ ，这个矩阵也是非负的，其功能和非负基矩阵类似。

3.3.3 社区信息保持的图嵌入模型



- 总体优化目标:

$$\min_{M,Z,H,C} \|S - MZ^T\|_F^2 + \alpha \|H - ZC^T\|_F^2 - \beta \text{tr}(H^T B H)$$
$$s. t. M \geq 0, Z \geq 0, H \geq 0, C \geq 0, \text{tr}(H^T H) = n$$

这里面 α 和 β 是用于调整相应项贡献的正参数。在这个公式里 M 和 C 两个矩阵把节点嵌入矩阵的维度变形到了合适的维度。该目标实现了社区信息和微观结构的同时保持。

3.3.4 全局信息保持的图嵌入模型



- DP (度惩罚原则) 思想
 - 度惩罚原则（DP）考虑到现实网络结构中普遍存在长尾分布现象，即大多数节点连接稀疏，只有少数节点具有密集连接。
 - DP方法的思想：惩罚高度的节点，使其相较于低度数节点更难表现出邻近性。
 - DP方法可以用于随机游走模型和谱嵌入模型。
 - 例如在随机游走中加入一定的偏向（如设置特定的转移概率函数），让随机游走不倾向于高度节点。

3.3.4 全局信息保持的图嵌入模型

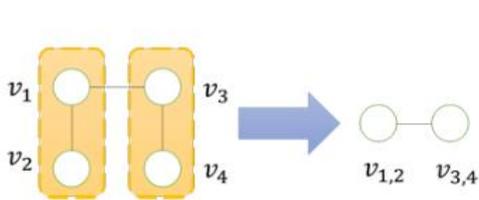
● HARP

- 相较于DP只关注网络中的长尾分布的现象，HARP则更具与一般性。为了捕捉网络中的全局信息，HARP方法首先基于图粗化方法，得到一系列具有层次的粗化图，从而可以根据在粗化图中学到的嵌入表示来保留网络的全局结构信息。

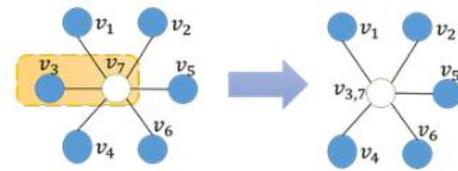
- 粗化方法：边坍塌和星型坍塌

- ▣ 边坍塌：把一条边两端的两个节点合并成一个超节点。

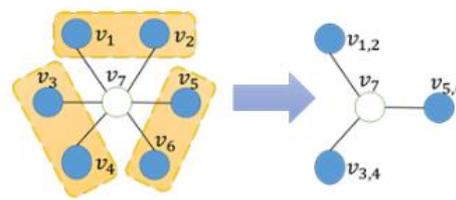
- ▣ 星型坍塌：针对图中的星形结构，将除中心节点外的节点合并成超节点。



(a) Edge Collapsing.



(b) Edge Collapsing fails to collapse stars.



(c) Star Collapsing.

3.3.4 全局信息保持的图嵌入模型



- 算法流程
 - 对图进行递归的粗化，这个过程会得到一系列规模逐渐减小的粗化图。
 - 然后在规模最小的粗化图上用图嵌入方法（DeepWalk, Line等）学习到一组超节点的嵌入。
 - 迭代地从小到大地用当前粗化图的嵌入去求下一个更大一点的粗化图的嵌入，具体来说就是把当前粗化图学习到的嵌入向量作为下一个粗化图学习嵌入的初始嵌入向量。
 - 迭代地求解到整个图，学习全图的嵌入向量。

3.4 侧信息保持的图嵌入模型



- 在现实世界中，节点不仅包含网络中的结构信息，还包含丰富的特征和标签等额外信息，这些信息统称为侧信息。在图嵌入的背景下，侧信息可分为两种类型：

- 属性信息

举例来说：维基百科文章相互连接并形成图，每篇文章作为一个节点，有大量的文本信息，这些信息对节点表示十分重要。

- 监督信息

举例来说：**Cora**和**Citeseer**中的论文标注了领域标签，便于检索。这些标签通常包含了节点的重要特征或摘要，引入标签数据可以提升节点区分能力。

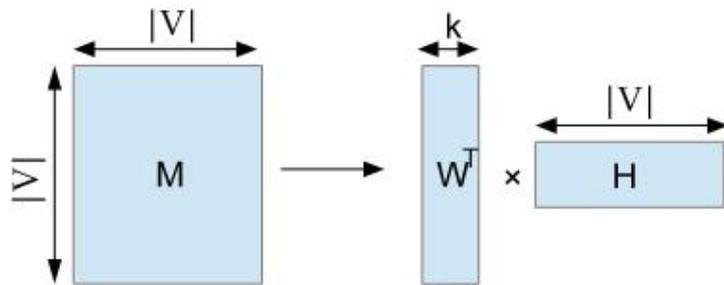


3.4.1 属性信息保持的图嵌入模型

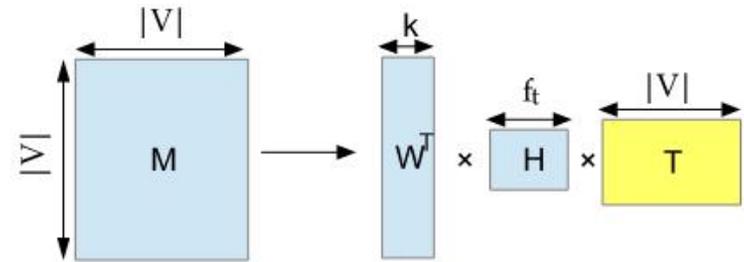
- 属性信息主要是指节点内容，下面主要介绍两种文本关联的图嵌入模型，即**TADW**（Text-Associated DeepWalk）和**HSCA**（Homophily, Structure, and Content Augmented Network Representation Learning）。
- 首先，给出含有文本信息的图嵌入问题定义：给定一个图 $G = \{\mathcal{V}, \mathcal{E}\}$ ，它的文本特征矩阵为 $T \in \mathbb{R}^{f_t \times |\mathcal{V}|}$ ，文本属性保持的图嵌入的目的是从结构和文本特征两个部分中学习每个节点 $v \in \mathcal{V}$ 的表示。

3.4.1 属性信息保持的图嵌入模型

- TADW的灵感来源于DeepWalk的矩阵分解形式。
- DeepWalk 被证明等价于将随机游走矩阵 \mathbf{M} 分解为两个低维矩阵 $\mathbf{W} \in \mathbb{R}^{k \times |\mathcal{V}|}$ 和 $\mathbf{H} \in \mathbb{R}^{k \times |\mathcal{V}|}$ ，其中 $k \ll |\mathcal{V}|$ 。
- TADW则将矩阵 \mathbf{M} 分解为三个矩阵乘积： $\mathbf{W} \in \mathbb{R}^{k \times |\mathcal{V}|}$ 、 $\mathbf{H} \in \mathbb{R}^{k \times f_t}$ 和文本特征矩阵 $\mathbf{T} \in \mathbb{R}^{f_t \times |\mathcal{V}|}$ 。



(1) DeepWalk



(2) TADW



3.4.1 属性信息保持的图嵌入模型

- TADW将矩阵 \mathbf{M} 简化定义为: $\mathbf{M} = (\mathbf{A} + \mathbf{A}^2)/2$ 。
- 通过优化平方损失函数求解矩阵 $\mathbf{W} \in \mathbb{R}^{k \times |V|}$ 和 $\mathbf{H} \in \mathbb{R}^{k \times f_t}$ ，并通过添加一个惩罚项来进一步约束，优化函数为：

$$\min_{\mathbf{W}, \mathbf{H}} \|\mathbf{M} - \mathbf{W}^T \mathbf{H} \mathbf{T}\|_F^2 + \frac{\lambda}{2} (\|\mathbf{W}\|_F^2 + \|\mathbf{H}\|_F^2)$$

无论对 \mathbf{W} 还是 \mathbf{H} 来说，这个损失函数都是凸函数，因此，TADW 通过交替优化 \mathbf{W} 和 \mathbf{H} 来最小化损失函数。

- 最终嵌入表示为: $\begin{bmatrix} \mathbf{W} \\ (\mathbf{H} \mathbf{T}) \end{bmatrix}^T$



3.4.1 属性信息保持的图嵌入模型

- TADW只考虑了网络顶点的结构上下文，如二阶和高阶邻近性，却忽略了一阶邻近性。**HSCA**方法弥补了这一点缺陷。
- 为了强化一阶邻近性，**HSCA**引入了一个正则化项来增强直接连接的节点在嵌入空间中的同配性，正则化项为：

$$R(\mathbf{W}, \mathbf{H}) = \frac{1}{4} \sum_{i,j=1}^{|V|} S_{ij} \left\| \begin{bmatrix} \mathbf{W}_i \\ \mathbf{H}\mathbf{T}_i \end{bmatrix} - \begin{bmatrix} \mathbf{W}_j \\ \mathbf{H}\mathbf{T}_j \end{bmatrix} \right\|_2^2$$

- 则HSCA的优化函数为：

$$\min_{\mathbf{W}, \mathbf{H}} \|\mathbf{M} - \mathbf{W}^T \mathbf{H} \mathbf{T}\|_F^2 + \frac{\lambda}{2} (\|\mathbf{W}\|_F^2 + \|\mathbf{H}\|_F^2) + \mu R(\mathbf{W}, \mathbf{H})$$



3.4.2 监督信息保持的图嵌入模型

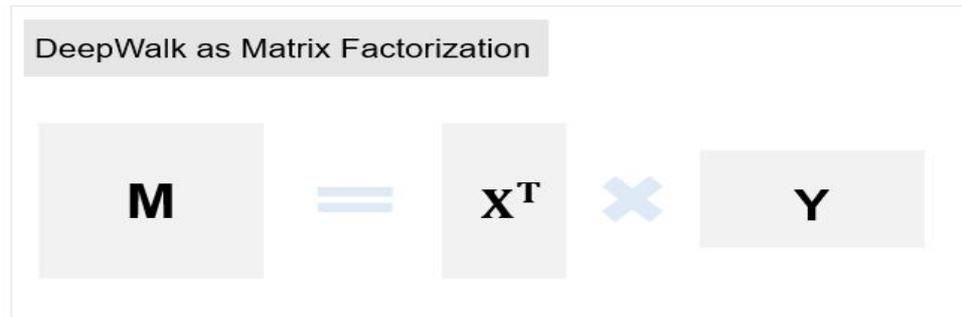
- 监督信息保持的图嵌入是指利用额外的监督信号，如节点标签来进一步提升图嵌入的质量。
- 接下来将介绍两种利用标签信息的半监督嵌入算法，即最大间隔深度游走(MMDW)和判别性深度随机游走(DDRW)。
- 首先，给出含有标签信息的图嵌入问题定义：给定一个图 $G = \{\mathcal{V}, \mathcal{E}\}$ ，每个节点有其标签 $l \in \{1, \dots, m\}$ ，对每一个顶点 $v \in \mathcal{V}$ ，从结构和标签中学习一个低维表示 $\mathbf{z}_v \in \mathbb{R}^k$ 。



3.4.2 监督信息保持的图嵌入模型

- 受最大边际原则的启发，并将节点的标签信息加入到节点表示学习中的半监督模型 MMDW（Max-Margin DeepWalk）被提了出来。
- MMDW由两部分组成：
 - 基于矩阵分解的DeepWalk，此处遵循TADW的设置，将 M 定义为 $M = (A + A^2)/2$ ，并使用如下目标进行优化：

$$\min_{X,Y} \mathcal{L}_{DW} = \min_{X,Y} \| M - (X^T Y) \|_2^2 + \frac{\lambda}{2} (\| X \|_2^2 + \| Y \|_2^2)$$



3.4.2 监督信息保持的图嵌入模型



- 最大边界DeepWalk

学习到的 \mathbf{X} 被用作特征，训练一个用于节点分类的SVM。假设训练集为 $T = \{(\mathbf{X}_1, l_1), \dots, (\mathbf{X}_T, l_T)\}$, $\mathbf{W} = [w_1, \dots, w_m]^T$ 是SVM的权重矩阵, $\xi = [\xi_1, \dots, \xi_T]$ 是容忍训练集中错误的松弛变量。

优化目标是:

$$\begin{aligned} \min_{\mathbf{W}, \xi} \mathcal{L}_{\text{SVM}} &= \min_{\mathbf{W}, \xi} \left(\frac{1}{2} \|\mathbf{W}\|_2^2 + C \sum_{i=1}^T \xi_i \right) \\ \text{s. t. } \mathbf{w}_{l_i}^T \mathbf{X}_i - \mathbf{w}_j^T \mathbf{X}_i &\geq e_i^j - \xi_i, \quad \forall i, j \\ e_i^j &= \begin{cases} 1, & \text{if } l_i \neq j \\ 0, & \text{if } l_i = j \end{cases} \end{aligned}$$



3.4.2 监督信息保持的图嵌入模型

- MMDW旨在同时优化SVM的最大边界分类器和基于矩阵分解的DeepWalk。因此它的联合优化目标为：

$$\min_{\mathbf{X}, \mathbf{Y}, \mathbf{W}, \xi} \mathcal{L} = \min_{\mathbf{X}, \mathbf{Y}, \mathbf{W}, \xi} \left(\mathcal{L}_{DW} + \frac{1}{2} \|\mathbf{W}\|_2^2 + C \sum_{i=1}^T \xi_i \right)$$

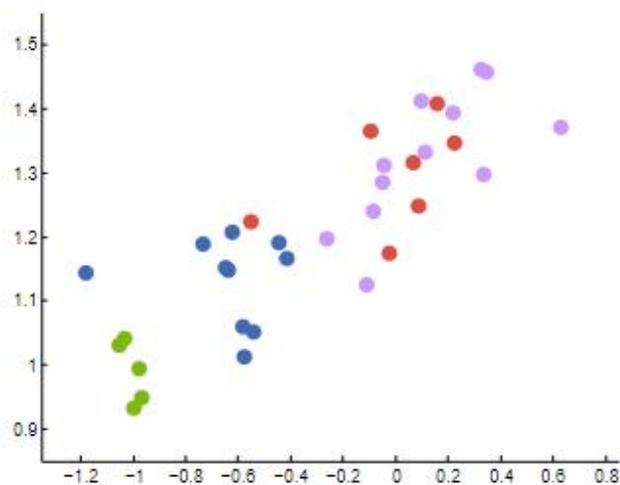
$$\text{s. t. } \mathbf{W}_{l_i}^T \mathbf{X}_i - \mathbf{W}_j^T \mathbf{X}_i \geq e_i^j - \xi_i, \quad \forall i, j$$

- 针对该多参数优化问题，MMDW将参数分为两部分并分别优化求解：
 - 固定 \mathbf{X} 和 \mathbf{Y} ，优化 \mathbf{W} 和 ξ
 - 固定 \mathbf{W} 和 ξ ，优化 \mathbf{X} 和 \mathbf{Y}

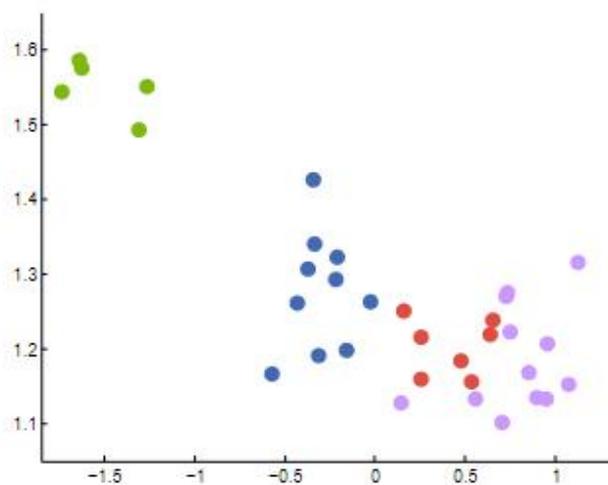
3.4.2 监督信息保持的图嵌入模型



- DDRW采用了类似 DeepWalk 的随机游走来提取网络结构，并采用 Word2Vec 方法将节点映射到潜在空间中，利用这些潜在表示进行分类任务，实现更好的嵌入表示学习。



(b) DeepWalk Embedding



(c) DDRW Embedding

DDRW使不同类别的顶点更加分离



3.4.2 监督信息保持的图嵌入模型

- DDRW有两个优化目标：

- 嵌入目标： $\min \mathcal{L}_r(\boldsymbol{\theta}, \alpha)$

旨在最小化随机游走得到的序列 α 与通过模型得到的节点向量表示 $\boldsymbol{\theta}$ 之间的差异，即为： $\min \mathcal{L}_r(\boldsymbol{\theta}, \alpha)$ 。

- 分类目标： $\min_{\boldsymbol{\theta}, \boldsymbol{\beta}} \mathcal{L}_c(\boldsymbol{\theta}, \mathbf{y}; \boldsymbol{\beta})$

旨在利用嵌入目标得到的向量表示 $\boldsymbol{\theta}$ 和节点标签 \mathbf{y} 训练一个分类器，其参数为 $\boldsymbol{\beta}$ 。

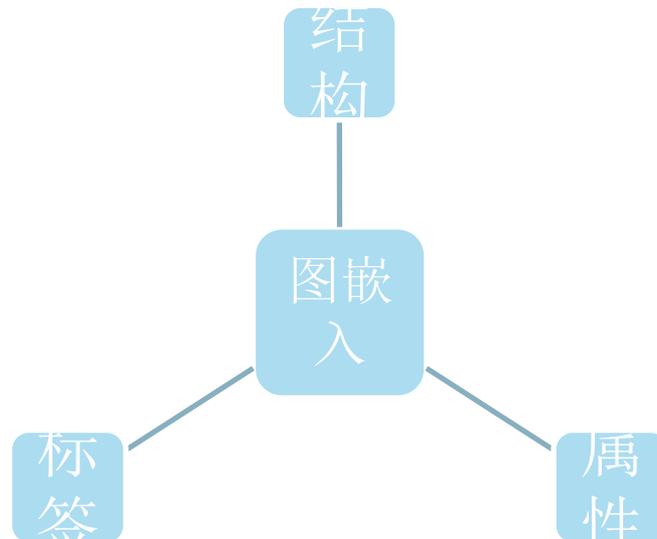
- DDRW 通过同时优化嵌入和分类两个目标来学习节点表示，其联合优化目标为：

$$\min \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\beta}, \alpha, \mathbf{y}) = \eta \mathcal{L}_r(\boldsymbol{\theta}, \alpha) + \mathcal{L}_c(\boldsymbol{\theta}, \boldsymbol{\beta}, \mathbf{y})$$



3.4.3 多种侧信息保持的图嵌入模型

- 侧信息保持的图嵌入不仅局限于单一类别的侧信息。还有许多方法致力于整合多种信息源来提升图嵌入的效果和实用性。这类方法通过融合网络结构、节点属性及节点标签等多维度数据，使得生成的嵌入表示更加丰富和具有辨别力。
- Planetoid 和 LANE 是两种结合多种信息源的图嵌入模型。





3.4.3 多种侧信息保持的图嵌入模型

- Planetoid算法通过整合网络结构、节点属性和标签信息来执行半监督学习，旨在学习图中每个顶点的表示。
- 首先，给定图 A ，Planetoid根据预先设定的策略采样 (i, c, γ) ， i 和 c 表示实例和上下文， $\gamma \in \{+1, -1\}$ 表示该上下文对是正例还是负例。
- Planetoid使用参数 $r_1 \in (0,1)$ 来控制正负样本的比例，并使用 $r_2 \in (0,1)$ 来控制两种类型的上下文（结构上下文和标签上下文）的比率。

3.4.3 多种侧信息保持的图嵌入模型



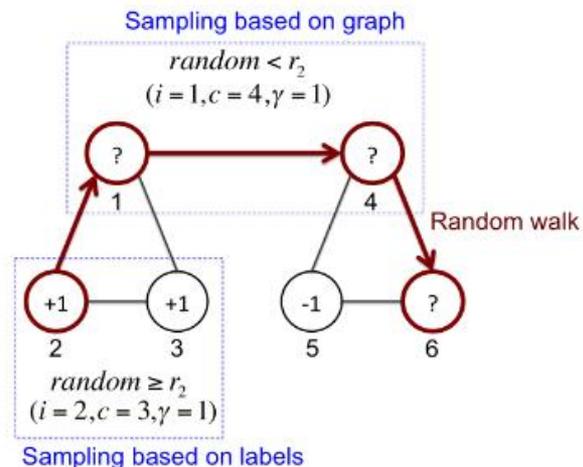
- Planetoid的采样过程如下：

- 通过随机数与阈值 r_1 的比较确定标签 γ ：

- 若随机数小于 r_1 ，则 $\gamma = +1$ ，表示正样本；
- 否则 $\gamma = -1$ ，表示负样本。

- 通过与 r_2 的比较确定采样方式：

- 若随机数小于 r_2 ，则基于图结构进行随机游走采样：随机游走生成长度为 q 的节点序列 S ，从中选择距离 $|j - k| < d$ 的节点对 S_j 和 S_k 分别作为实例节点 i 和上下文节点 c 。对负样本重新随机选择上下文节点。
- 若随机数不小于 r_2 ，则进行基于标签的采样：正样本选择同标签节点作为上下文，负样本则选择不同标签节点。



Planetoid 的随机游走采样



3.4.3 多种侧信息保持的图嵌入模型

- 采样后，定义用于结构上下文预测的损失函数 \mathcal{L}_u ：

$$\mathcal{L}_u = -\mathbb{E}_{(i,c,\gamma)} \log(\sigma(\gamma w_c^T z_i))$$

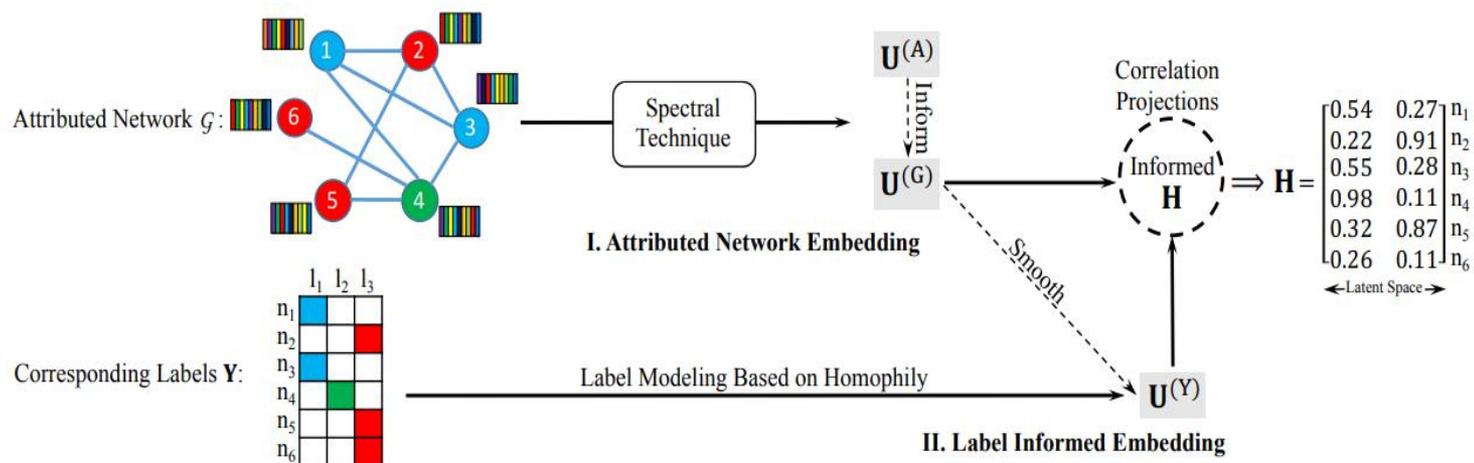
- 再将顶点嵌入 z_i 和节点的属性 x_i 通过神经网络映射到隐藏空间，并将这两个隐藏层表示拼接起来。最终分类目标为：

$$\mathcal{L}_s = -\frac{1}{l} \sum_{i=1}^l \log p(y_i | x_i, z_i)$$

- 通过联合最小化 L_u 和 L_s ，Planetoid 不仅通过网络结构和属性学习节点嵌入，还整合了标签信息，增强模型对未见数据的推广能力，从而在图数据的分类任务中获得更优性能。

3.4.3 多种侧信息保持的图嵌入模型

- LANE(Label Informed Attribute Network Embedding) 算法是一种将网络结构、节点属性和标签信息有机结合的图嵌入方法。
- LANE 首先将节点属性相似度、结构相似度和标签相似度分别映射到 $U^{(A)}$, $U^{(G)}$, $U^{(Y)}$, 并通过相关性投影将这三个表示映射到统一的联合嵌入表示 H 中。





- 在本章，主要介绍了常见类型的图嵌入方法。
 - 在第一节，介绍了图嵌入的背景，概念和其定义，并对其进行分类。
 - 第二节重点介绍了基于流形的图嵌入模型，如 **Isomap**、局部线性嵌入 (**LLE**) 和拉普拉斯特征映射 (**LE**)，这些有效解决了早期线性降维方法无法应用于非线性数据集的弊端。
 - 第三节介绍了更加先进的结构信息保持的图嵌入模型，如 **DeepWalk** 和 **LINE**，它们通过引入随机游走等高级机制，捕捉图中的邻域信息、结构角色信息和全局信息。
 - 第四节进一步探讨了 **TADW**、**MMDW** 和 **DDRW** 等模型，展示了如何将节点的文本属性和标签等侧信息整合到图嵌入中，使嵌入表示不仅保留了图的结构特征，还增强了节点区分度。