

深度学习推荐系统中各类流行的Embedding方法（下）

原创 Microstrong Microstrong 5月23日

收录于话题
#推荐系统原理解析与实践部署

6个

Embedding技术概览：



[深度学习推荐系统中各类流行的Embedding方法（上）](#)

1. Graph Embedding简介

Word2Vec和其衍生出的Item2Vec类模型是Embedding技术的基础性方法，二者都是建立在“序列”样本（比如句子、用户行为序列）的基础上的。在互联网场景下，数据对象之间更多呈现的是图结构，所以Item2Vec在处理大量的网络化数据时往往显得捉襟见肘，在这样的背景下，Graph Embedding成了新的研究方向，并逐渐在深度学习推荐系统领域流行起来。

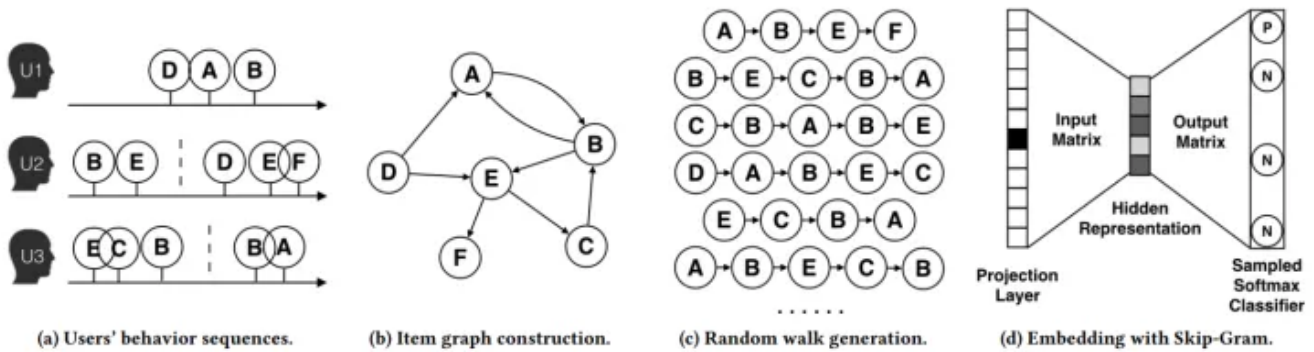
Graph Embedding也是一种特征表示学习方式，借鉴了Word2Vec的思路。在Graph中随机游走生成顶点序列，构成训练集，然后采用Skip-gram算法，训练出低维稠密向量来表示顶点。之后再学习出的向量解决下游问题，比如分类，或者连接预测问题等。可以看做是两阶段的学习任务，第一阶段先做无监督训练生成表示向量，第二阶段再做有监督学习，解决下游问题。

总之，Graph Embedding是一种对图结构中的节点进行Embedding编码的方法。最终生成的节点Embedding向量一般包含图的结构信息及附近节点的局部相似性信息。不同Graph

Embedding方法的原理不尽相同，对于图信息的保留方式也有所区别，下面就介绍几种主流的Graph Embedding方法和它们之间的区别与联系。

2. DeepWalk-Graph Embedding早期技术

早期，影响力较大的Graph Embedding方法是于2014年提出的DeepWalk，它的主要思想是在由物品组成的图结构上进行随机游走，产生大量物品序列，然后将这些物品序列作为训练样本输入Word2Vec进行训练，得到物品的Embedding。因此，DeepWalk可以被看作连接序列Embedding和Graph Embedding的过渡方法。



论文《Billion-scale Commodity Embedding for E-commerce Recommendation in Alibaba》用上图所示的方法展现了DeepWalk的算法流程。DeepWalk算法的具体步骤如下：

- 图（a）是原始的用户行为序列。
- 图（b）基于这些用户行为序列构建了物品关系图。可以看出，物品A和B之间的边产生的原因是用户U1先后购买了物品A和物品B。如果后续产生了多条相同的有向边，则有向边的权重被加强。在将所有用户行为序列都转换成物品关系图中的边之后，全局的物品关系图就建立起来了。
- 图（c）采用随机游走的方式随机选择起始点，重新产生物品序列。
- 将这些物品序列输入图（d）所示的Word2Vec模型中，生成最终的物品Embedding向量。

在上述DeepWalk的算法流程中，唯一需要形式化定义的是随机游走的跳转概率，也就是到达结点 v_i 后，下一步遍历 v_i 的邻接点 v_j 的概率。如果物品关系图是有向有权图，那么从节点 v_i 跳转到节点 v_j 的概率定义如下式所示。

$$P(v_j|v_i) = \begin{cases} \frac{M_{ij}}{\sum_{j \in N_+(v_i)} M_{ij}}, & v_j \in N_+(v_i) \\ 0, & v_j \notin N_+(v_i) \end{cases}$$

其中 ε 是物品关系图中所有边的集合， $N_+(v_i)$ 是节点 v_i 所有的出边集合， M_{ij} 是节点 v_i 到节点 v_j 边的权重，即DeepWalk的跳转概率就是跳转边的权重占所有相关出边权重之和的比例。

如果物品关系图是无向无权图，那么跳转概率将是上式的一个特例，即权重 M_{ij} 将为常数1，且 $N_+(v_i)$ 应是节点 v_i 所有“边”的集合，而不是所有“出边”的集合。

注意：在DeepWalk论文中，作者只提出DeepWalk用于无向无权图。DeepWalk用于有向有权图的内容是阿里巴巴论文《Billion-scale Commodity Embedding for E-commerce Recommendation in Alibaba》中提出的Base Graph Embedding（BGE）模型，其实该模型就是对DeepWalk模型的实践，本文后边部分会讲解该模型。

DeepWalk相关论文：

【1】Perozzi B, Alrfou R, Skiena S, et al. DeepWalk: online learning of social representations[C]. knowledge discovery and data mining, 2014: 701-710.

【2】Wang J, Huang P, Zhao H, et al. Billion-scale Commodity Embedding for E-commerce Recommendation in Alibaba[C]. knowledge discovery and data mining, 2018: 839-848.

3. LINE-DeepWalk的改进

DeepWalk使用DFS（Deep First Search，深度优先搜索）随机游走在图中进行节点采样，使用Word2Vec在采样的序列上学习图中节点的向量表示。LINE（Large-scale Information Network Embedding）也是一种基于邻域相似假设的方法，只不过与DeepWalk使用DFS构造邻域不同的是，LINE可以看作是一种使用BFS（Breath First Search，广度优先搜索）构造邻域的算法。

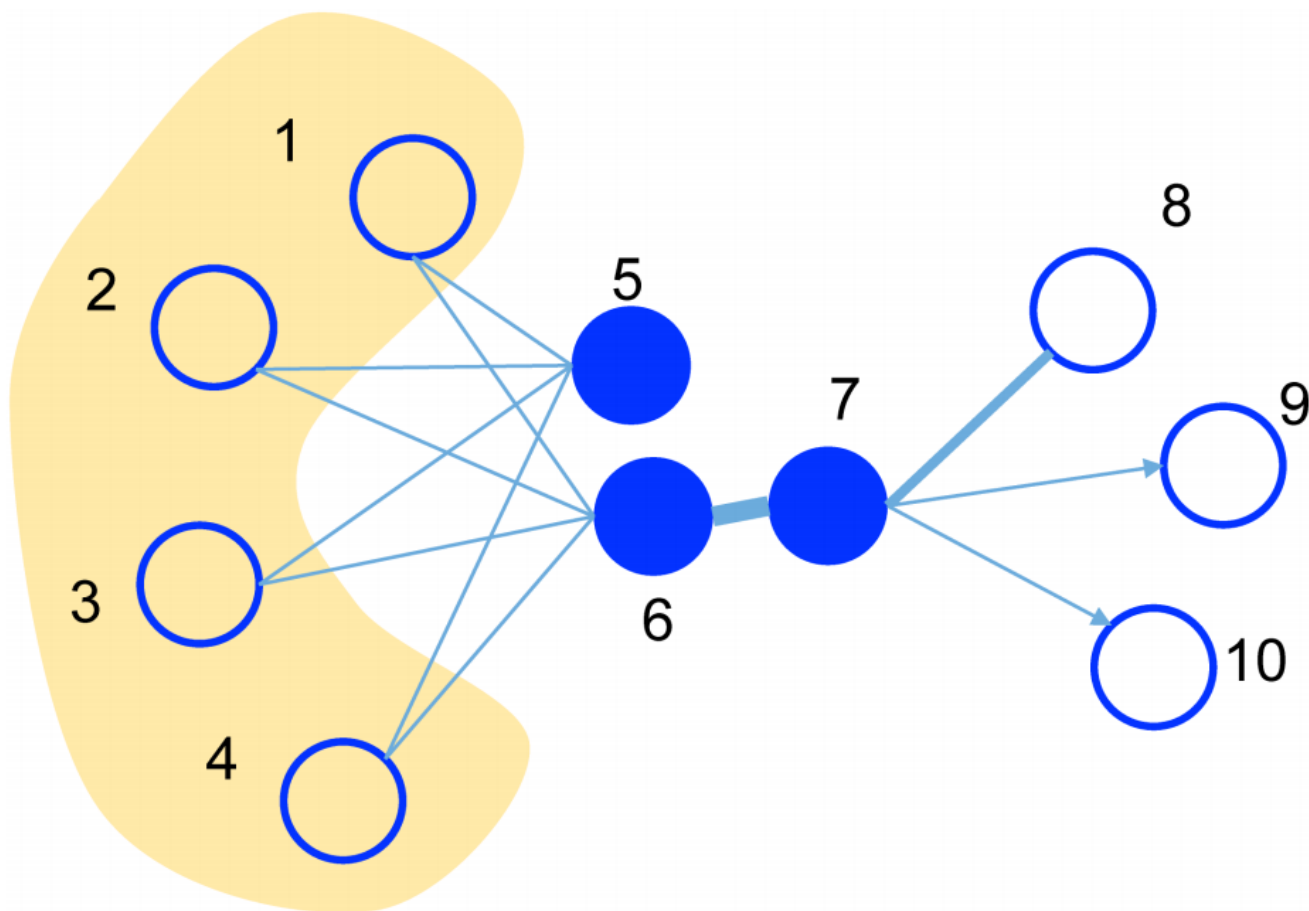
在Graph Embedding各个方法中，一个主要区别是对图中顶点之间的相似度的定义不同，所以先看一下LINE对于相似度的定义。

3.1 LINE定义图中节点之间的相似度

现实世界的网络中，相连接的节点之间存在一定的联系，通常表现为比较相似或者在向量空间中距离接近。对于带权网络来说，节点之间的权越大，相似度会越高或者距离越接近，这种关系称为一阶近邻。一阶近邻关系用于描述图中相邻顶点之间的局部相似度，形式化描述为若顶点 u 、 v 之间存在直连边，则边权 w_{uv} 即为两个顶点的相似度，若不存在直连边，则一阶相似度为0。如下图所示，6和7之间存在直连边，且边权较大（表现为图中顶点之间连线较粗），则认为两者相似且一阶相似度较高，而5和6之间不存在直连边，则两者间一阶相似度为0。

但是，网络中的边往往比较稀疏，仅仅依靠一阶近邻关系，难以描述整个网络的结构。论文中定义了另外一种关系叫做二阶近邻。例如下图中的网络，节点5和节点1,2,3,4相连，节点6也和节点1,2,3,4相连，虽然节点5和6之间没有直接联系，但是节点5和6之间很可能存在某种相似性。举个例子，在社交网络中，如果两个人的朋友圈重叠度很高，或许两个人之间具有相同的兴趣并有可能成为朋友；在NLP中，如果不同的词经常出现在同一个语境中，那么两个词很可能意思相近。

LINE通过捕捉网络中的一阶近邻关系和二阶近邻关系，更加完整地描述网络。并且LINE适用于有向图、无向图、有权图、无权图。



3.2 LINE算法模型

（1）一阶近邻关系模型

一阶近邻关系模型中定义了两个概率，一个是联合概率，如下公式所示：

$$p_1(v_i, v_j) = \frac{1}{1 + \exp(-\vec{u}_i^T \cdot \vec{u}_j)}$$

其中， \vec{u}_i 是图中节点 v_i 的向量表示，上式表示节点 v_i 和 v_j 之间的相似程度，这是一个sigmoid函数。

另外一个经验概率，如下公式所示：

$$\hat{p}_1(i, j) = \frac{w_{ij}}{W}, W = \sum_{(i,j) \in E} w_{ij}$$

其中， w_{ij} 是节点 v_i 和 v_j 之间的权重。优化目标为最小化下式：

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

其中， $d(\cdot, \cdot)$ 是两个分布的距离，目标是期望两个概率分布接近，利用KL散度来计算相似性，丢掉常数项之后，得到下面公式：

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

一阶近邻关系模型的优化目标就是最小化 O_1 。可以看到，上面这些公式无法表达方向概念，因此一阶近邻关系模型只能描述无向图。

（2）二阶近邻关系模型

二阶近邻关系描述的是节点与邻域的关系，每个节点有两个向量，一个是该顶点本身的表示向量，一个是该顶点作为其他顶点的邻居时的表示向量，因此论文中对每个节点定义了两个向量， \vec{u}_i 表示节点*i*本身， \vec{u}'_j 是节点*j*作为邻居的向量表示。针对每一个从节点*i*到*j*的有向边 (i, j) ，定义一个条件概率，如下式：

$$p_2(v_j|v_i) = \frac{\exp(\vec{u}'_j \cdot \vec{u}_i)}{\sum_{k=1}^{|V|} \exp(\vec{u}'_k \cdot \vec{u}_i)}$$

其中， $|V|$ 是图中所有的节点数量，这其实是一个 $softmax$ 函数。同样，还有一个经验概率，如下式：

$$\hat{p}_2(v_j|v_i) = \frac{w_{ij}}{d_i}, d_i = \sum_{k \in N(i)} w_{ik}$$

其中， w_{ij} 是边 (i, j) 的边权， d_i 是从顶点 v_i 出发指向邻居节点的所有边权之和， $N(i)$ 是从节点*i* 出发指向邻居的所有边集合。同样需要最小化条件概率和经验概率之间的距离，优化目标为：

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

其中， λ_i 为控制节点重要性的因子，可以通过顶点的度数或者PageRank等方法估计得到。假设度比较高的节点权重较高，令 $\lambda_i = d_i$ ，采用KL散度来计算距离，略去常数项后，得到公式：

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

直接优化上式计算复杂度很高，每次迭代需要对所有的节点向量做优化，论文中使用Word2Vec中的负采样方法，得到二阶近邻的优化目标，如下公式所示。从计算的过程可以看到，二阶相似度模型可以描述有向图。

$$O_2 = - \sum_{(i,j) \in E} w_{ij} \cdot (\log \sigma(\vec{u}'_j \cdot \vec{u}_i) + \sum_{i=1}^K E_{v_n \sim P_n(v)} [\log \sigma(-\vec{u}'_n \cdot \vec{u}_i)])$$

对比一阶近邻模型和二阶近邻模型的优化目标，差别就在于，二阶近邻模型对每个节点多引入了一个向量表示。实际使用的时候，对一阶近邻模型和二阶近邻模型分别训练，然后将两个向量拼接起来作为节点的向量表示。

此外有一点需要说明，在Graph Embedding方法中，例如DeepWalk、Node2Vec、EGES，都是采用随机游走的方式来生成序列再做训练，而LINE直接用边来构造样本，这也是他们的一点区别。

LINE论文：

【1】Tang J, Qu M, Wang M, et al. Line: Large-scale information network embedding[C]//Proceedings of the 24th international conference on world wide web. 2015: 1067-1077.

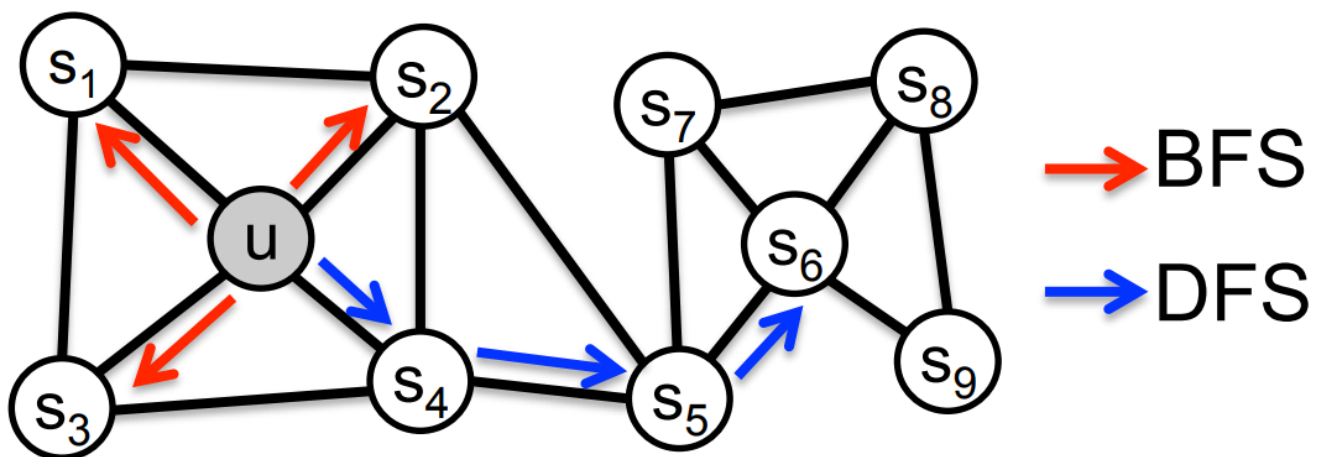
4. node2vec - DeepWalk的改进

2016年，斯坦福大学的研究人员在DeepWalk的基础上更进一步，提出了node2vec模型，它通过调整随机游走权重的方法使Graph Embedding的结果更倾向于体现网络的同质性（homophily）或结构性（structural equivalence）。

4.1 node2vec的同质性和结构性

具体的讲，网络的“同质性”指的是距离相近节点的Embedding应尽量近似，如下图所示，节点 u 与其相连的节点 s_1, s_2, s_3, s_4 的Embedding表达应该是接近的，这就是网络的“同质性”的体现。

“结构性”指的是结构上相似的节点Embedding应尽量近似，下图中节点 u 和节点 s_6 都是各自局域网络的中心节点，结构上相似，其Embedding的表达也应该近似，这是“结构性”的体现。



为了使Graph Embedding的结果能够表达网络的“结构性”，在随机游走过程中，需要让游走的过程更倾向于BFS，因为BFS会更多地当前节点的邻域中游走遍历，相当于对当前节点周边的网络结构进行一次“微观扫描”。当前节点是“局部中心节点”，还是“边缘节点”，或是“连接性节点”，其生成的序列包含的节点数量和顺序必然是不同的，从而让最终的Embedding抓取到更多结构性信息。

另外，为了表达“同质性”，需要让随机游走的过程更倾向于DFS，因为DFS更有可能通过多次跳转，游走到远方的节点上，但无论如何，DFS的游走更大概率会在一个大的集团内部进行，

这就使得一个集团或者社区内部的节点的Embedding更为相似，从而更多地表达网络的“同质性”。

但是在不同的任务中需要关注的重点不同，可能有些任务需要关注网络的homophily，而有些任务比较关注网络的structural equivalence，可能还有些任务两者兼而有之。在DeepWalk中，使用DFS随机游走在图中进行节点采样，使用Word2Vec在采样的序列学习图中节点的向量表示，无法灵活地捕捉这两种关系。

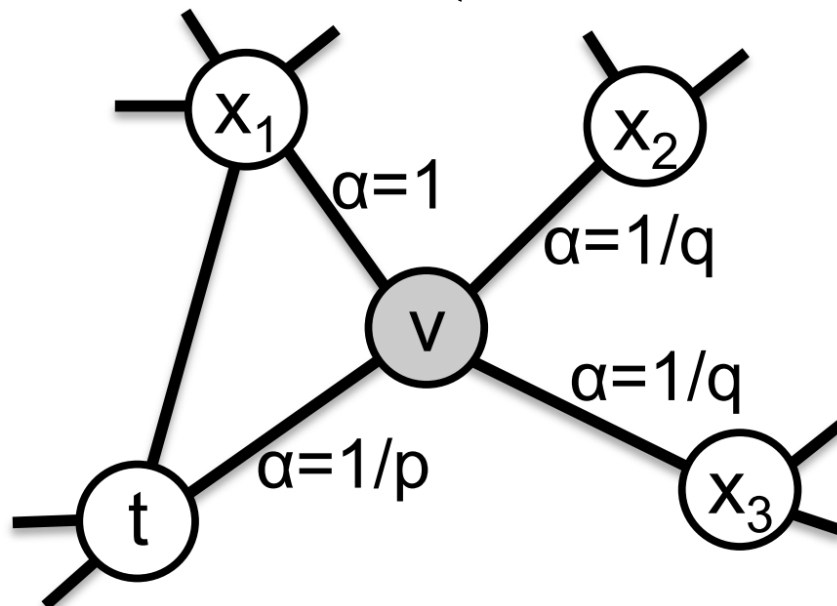
实际上，对于这两种关系的偏好，可以通过不同的序列采样方式来实现。有两种极端的方式，一种是BFS，如上图红色箭头所示，从u出发做随机游走，但是每次都只采样顶点u的直接邻域，这样生成的序列通过无监督训练之后，特征向量表现出来的是structural equivalence特性。另外一种DFS，如上图蓝色箭头所示，从u出发越走越远，学习得到的特征向量反应的是图中的homophily关系。

4.2 node2vec算法

那么在node2vec算法中，是怎么控制BFS和DFS的倾向性呢？主要是通过节点间的跳转概率。下图所示为node2vec算法从节点t跳转到节点v，再从节点v跳转到周围各点的跳转概率。假设从某顶点出发开始随机游走，第 $i-1$ 步走到当前顶点v，要探索第 i 步的顶点x，如下图所示。下面的公式表示从顶点v到x的跳转概率， E 是图中边的集合， (v, x) 表示顶点v和x之间的边， π_{vx} 表示从节点v跳转到下一个节点x的概率， Z 是归一化常数。

带偏随机游走的最简单方法是基于下一个节点边权重 w_{vx} 进行采样，即 $\pi_{vx} = w_{vx}$ ， Z 是权重之和。对于无权重的网络， $w_{vx} = 1$ 。最简单的方式，就是按照这个转移概率进行随机游走，但是无法控制BFS和DFS的倾向性。

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

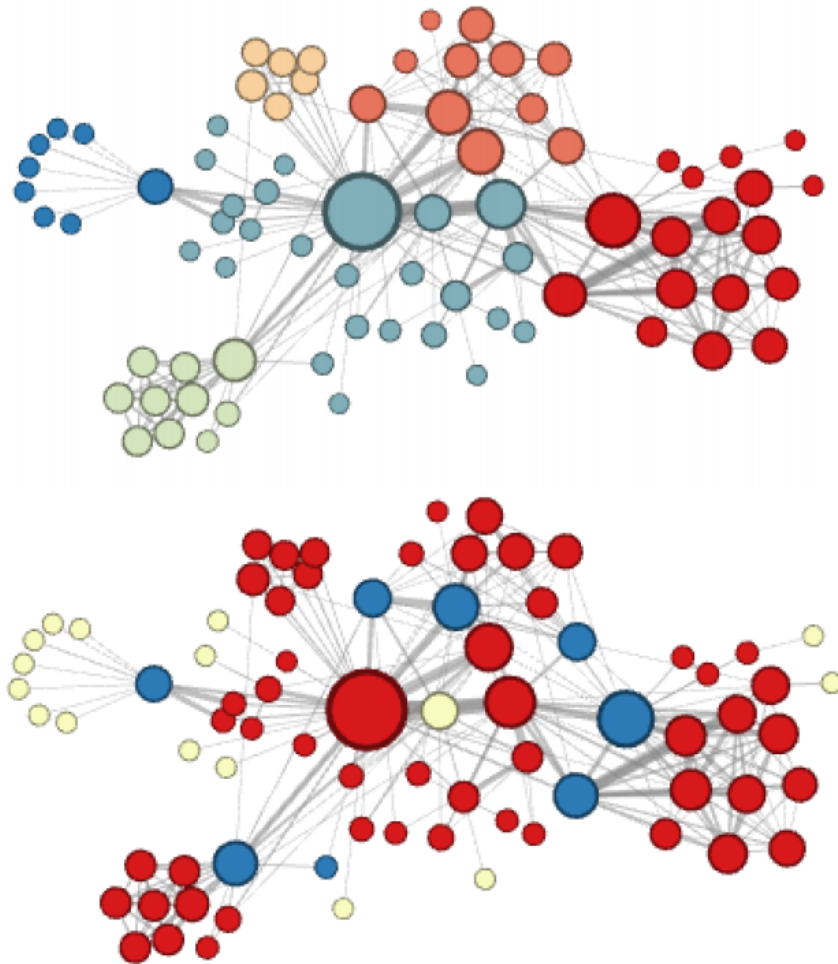


node2vec用两个参数 p 和 q 定义了一个二阶随机游走，以控制随机游走的策略。假设当前随机游走经过边 (t, v) 到达顶点 v ，现在要决定从节点 v 跳转到下一个节点 x ，需要依据边 (v, x) 上的跳转概率 π_{vx} 。设 $\pi_{vx} = \alpha_{pq}(t, x) \cdot w_{vx}$ ， w_{vx} 是顶点 v 和 x 之间的边权； $\alpha_{pq}(t, x)$ 是修正系数，定义如下：

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

上式中 d_{tx} 表示下一步顶点 x 和顶点 t 之间的最短距离，只有3种情况，如果又回到顶点 t ，那么 $d_{tx} = 0$ ；如果 x 和 t 直接相邻，那么 $d_{tx} = 1$ ；其他情况 $d_{tx} = 2$ 。参数 p 和 q 共同控制着随机游走的倾向性。参数 p 被称为返回参数（return parameter），控制着重新返回顶点 t 的概率。如果 $p > \max(q, 1)$ ，那么下一步较小概率重新返回顶点 t ；如果 $p < \max(q, 1)$ ，那么下一步会更倾向于回到顶点 t ，**node2vec**就更注重表达网络的结构性。参数 q 被称为进出参数（in-out parameter），如果 $q > 1$ ，那么下一步倾向于回到 t 或者 t 的临近顶点，这接近于BFS的探索方式；如果 $q < 1$ ，那么下一步倾向于走到离 t 更远的顶点，接近于DFS寻路方式，**node2vec**就更加注重表达网络的同质性。因此，可以通过设置 p 和 q 来控制游走网络的方式。所谓的二阶随机游走，意思是说下一步去哪，不仅跟当前顶点的转移概率有关，还跟上一顶点相关。在论文中试验部分，作者对 p 和 q 的设置一般是2的指数，比如 $\{\frac{1}{4}, \frac{1}{2}, 1, 2, 4\}$ 。

node2vec这种灵活表达同质性和结构性的特点也得到了实验的证实，通过调整参数 p 和 q 产生了不同的Embedding结果。下图中的上半部分图片就是**node2vec**更注重同质性的体现，可以看到距离相近的节点颜色更为接近，下图中下半部分图片则更注重体现结构性，其中结构特点相近的节点的颜色更为接近。



4.3 node2vec在推荐系统中的思考

node2vec所体现的网络的同质性和结构性在推荐系统中可以被很直观的解释。同质性相同的物品很可能是同品类、同属性，或者经常被一同购买的商品，而结构性相同的物品则是各品类的爆款、各品类的最佳凑单商品等拥有类似趋势或者结构性属性的商品。毫无疑问，二者在推荐系统中都是非常重要的特征表达。由于node2vec的这种灵活性，以及发掘不同图特征的能力，甚至可以把不同node2vec生成的偏向“结构性”的Embedding结果和偏向“同质性”的Embedding结果共同输入后续的深度学习网络，以保留物品的不同图特征信息。

node2vec论文：

【1】Grover A, Leskovec J. node2vec: Scalable feature learning for networks[C]//Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. 2016: 855-864.

5. EGES - Graph Embedding最佳实践

2018 年，阿里巴巴公布了其在淘宝应用的Embedding方法 EGES（Enhanced Graph Embedding with Side Information）算法，其基本思想是Embedding过程中引入带权重的补充

信息（Side Information），从而解决冷启动的问题。

淘宝平台推荐的三个问题：

- **可扩展性(scalability)**：已有的推荐算法(CF、Base-Content、DL)可以在小数据集上有不错效果，但是对于10亿用户和20亿商品这样海量的数据集上效果差。
- **稀疏性(sparsity)**：用户仅与小部分商品交互，难以训练准确的推荐模型。
- **冷启动(cold start)**：物品上新频繁，然而这些商品并没有用户行为，预测用户对这些商品的偏好是十分具有挑战性的。

现在业界针对海量数据的推荐问题通用框架是分成两个阶段，即 **matching** 和 **ranking**。在 **matching** 阶段，我们会生成一个候选集，它的items会与用户接触过的每个item具有相似性；接着在 **ranking** 阶段，我们会训练一个深度神经网络模型，它会为每个用户根据他的偏好对候选items进行排序。论文关注的问题在推荐系统的 **matching** 阶段，也就是从商品池中召回候选商品的阶段，核心的任务是计算所有item之间的相似度。

为了达到这个目的，论文提出根据用户历史行为构建一个item graph，然后使用DeepWalk学习每个item的embedding，即**Base Graph Embedding (BGE)**。BGE优于CF，因为基于CF的方法只考虑了为用户行为历史中的items的共现率，但是对于少量或者没有交互行为的item，仍然难以得到准确的embedding。为了减轻该问题，论文提出使用side information来增强embedding过程，提出了**Graph Embedding with Side information (GES)**。例如，属于相似类别或品牌的item的embedding应该相近。在这种方式下，即使item只有少量交互或没有交互，也可以得到准确的item embedding。在淘宝场景下，side information包括：category、brand、price等。不同的side information对于最终表示的贡献应该不同，于是论文进一步提出一种加权机制用于学习Embedding with Side Information，称为**Enhanced Graph Embedding with Side information (EGES)**。

5.1 基于图的Embedding (BGE)

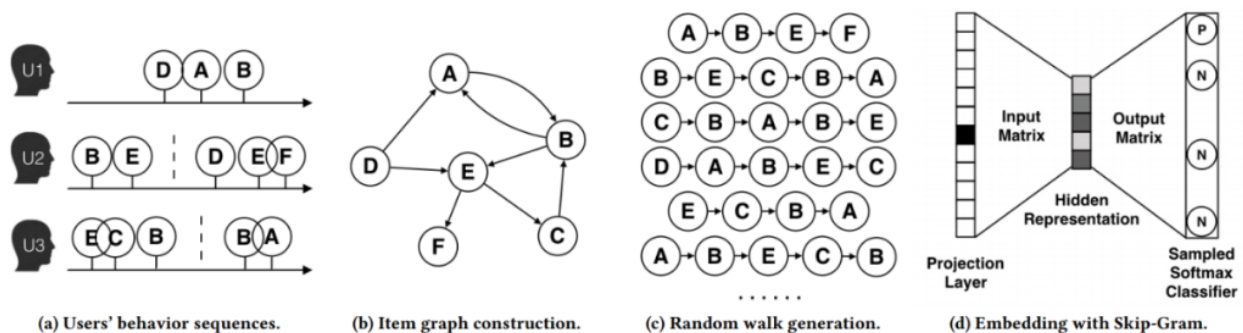


Figure 2: Overview of graph embedding in Taobao: (a) Users' behavior sequences: One session for user u1, two sessions for user u2 and u3; these sequences are used to construct the item graph; (b) The weighted directed item graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$; (c) The sequences generated by random walk in the item graph; (d) Embedding with Skip-Gram.

该方案是 DeepWalk 算法的实践，具体流程如下：

- 首先，我们拥有上亿用户的行为数据，不同的用户，在每个 Session 中，访问了一系列商品，例如用户 u2 两次访问淘宝，第一次查看了两个商品 B-E，第二次产看了三个商品 D-E-F。
- 然后，通过用户的行为数据，我们可以建立一个商品图（Item Graph），可以看出，物品 A，B之间的边产生的原因就是用户U1先后购买了物品A和物品B，所以产生了一条由A到B的有向边。如果后续产生了多条相同的有向边，则有向边的权重被加强。在将所有用户行为序列都转换成物品相关图中的边之后，全局的物品相关图就建立起来了。
- 接着，通过 Random Walk 对图进行采样，重新获得商品序列。
- 最后，使用 Skip-gram 模型进行 Embedding。

Base Graph Embedding 与 **DeepWalk** 不同的是：通过 **user** 的行为序列构建网络结构，并将网络定义为有向有权图。其中：根据行为的时间间隔，将一个 **user** 的行为序列分割为多个session。session分割可以参考Airbnb这篇论文《Real-time Personalization using Embeddings for Search Ranking at Airbnb》。

5.2 使用Side Information的GE（GES）

通过使用BGE，我们能够将items映射到高维向量空间，并考虑了CF没有考虑的用户序列关系。但是我们依然没有解决冷启动的问题。为了解决冷启动问题，我们使用边信息（category, shop, price, etc）赋值给不同的item。因为边信息相同的两个item，理论而言会更接近。通过DeepWalk方案得到item的游走序列，同时得到对应的边信息（category, brand, price）序列。然后将所有序列放到Word2Vec模型中进行训练。针对每个 item，将得到：item_embedding, category_embedding, brand_embedding, price_embedding 等 embedding 信息。为了与之前的item embedding区分开，在加入Side information之后，我们称得到的embedding为商品的aggregated embeddings。商品v的aggregated embeddings为：

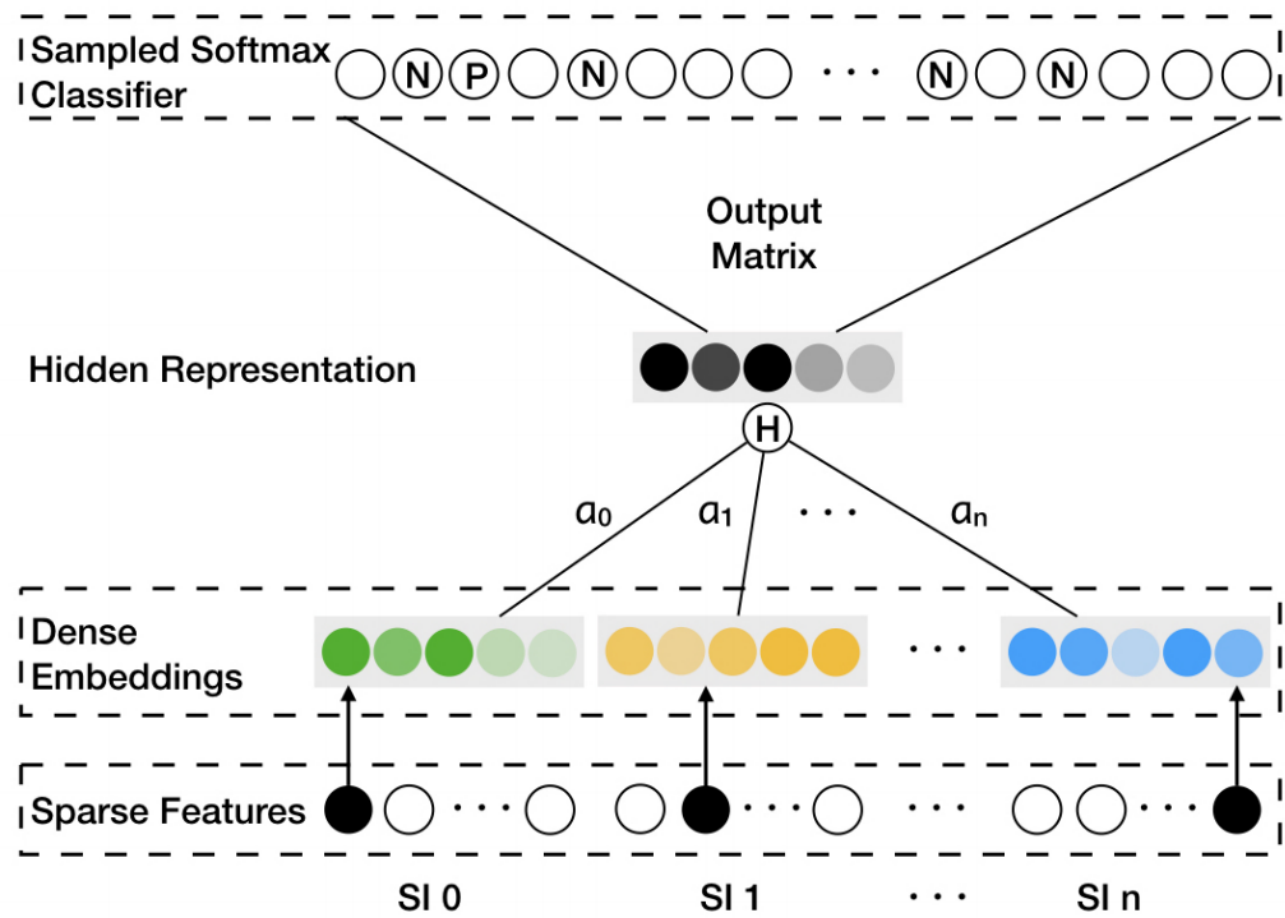
$$H_v = \frac{1}{n+1} \sum_{s=0}^n W_v^s$$

对上式做一个简单的解释：针对每个 item，将得到：item_embedding, category_embedding, brand_embedding, price_embedding 等 embedding 信息。将这些 embedding 信息求均值来表示该 item的Embedding。

需要注意的一点是，**item** 和 **side information**（例如category, brand, price等）的 **Embedding** 是通过 **Word2Vec** 算法一起训练得到的。如果分开训练，得到的item_embedding和category_embedding、brand_embedding、price_embedding不在一个向量空间中，做运算无意义。即：通过 DeepWalk 方案得到 item 的游走序列，同时得到对应的{category, brand, price}序列。然后将所有序列数据放到Word2Vec模型中进行训练。

5.3 增强型GES（EGES）

GES中存在一个问题是，针对每个item，它把所有的side information embedding求和后做了平均，没有考虑不同的side information 之间的权重，EGES就是让不同类型的side information具有不同的权重，提出来一个加权平均的方法来聚集这些边界embedding。



因为每个item对其不同边信息的权重不一样，所以需要额外矩阵A来表示每个item边信息的权重，其大小为 $v * (n + 1)$ ， v 是item的个数， n 是边信息的个数，加1是还要考虑item自身Embedding的权重。为了简单起见，我们用 α_v^j 表示第 v 个item、第 j 个类型的side information的权重。 α_v^0 表示第 v 个item自身Embedding的权重。这样就可以获得加权平均的方法：

$$H_v = \frac{\sum_{j=0}^n e^{\alpha_v^j} W_v^j}{\sum_{j=0}^n e^{\alpha_v^j}}$$

这里对权重项 α_v^j 做了指数变换，目的是为了保证每个边信息的贡献都能大于0。权重矩阵A通过模型训练得到。

EGES算法应用改进的Word2Vec算法（Weighted Skip-Gram）确定模型的参数。对上图中EGES算法简单说明如下：

- 上图的Sparse Features代表 item 和 side information 的ID信息；
- Dense Embeddings 表示 item 和 side information 的 embedding 信息；
- $\alpha_0, \alpha_1, \dots, \alpha_n$ 分别代表 item 和 side information 的 embedding 权重；
- Sampled Softmax Classifier中的 N 代表采样的负样本（见论文中的Algorithm 2 Weighted Skip-Gram描述的第8行）， P 代表正样本（某个item周边上下n个item均为正样本，在模型

中表示时不区分远近）；

EGES并没有过于复杂的理论创新，但给出了一个工程上的融合多种Embedding的方法，降低了某类信息缺失造成的冷启动问题，是实用性极强的Embedding方法。

EGES论文：

【1】Wang J, Huang P, Zhao H, et al. Billion-scale Commodity Embedding for E-commerce Recommendation in Alibaba[C]. knowledge discovery and data mining, 2018: 839-848.

6. 总结

时至今日，Graph Embedding仍然是工业界和学术界研究和实践的热点，除了本文介绍的DeepWalk、LINE、node2vec、EGES等主流方法，SDNE、struct2vec等方法也是重要的Graph Embedding模型，感兴趣的读者可以自己查找相关文献进一步学习。

7. Reference

- 【1】《深度学习推荐系统》王喆编著。
- 【2】【Graph Embedding】DeepWalk：算法原理，实现和应用 - 浅梦的文章 - 知乎
<https://zhuanlan.zhihu.com/p/56380812>
- 【3】【论文笔记】DeepWalk - 陌上疏影凉的文章 - 知乎
<https://zhuanlan.zhihu.com/p/45167021>
- 【4】【Graph Embedding】LINE：算法原理，实现和应用 - 浅梦的文章 - 知乎
<https://zhuanlan.zhihu.com/p/56478167>
- 【5】Graph Embedding：从DeepWalk到SDNE - 羽刻的文章 - 知乎
<https://zhuanlan.zhihu.com/p/33732033>
- 【6】Graph Embedding之探索LINE - 张备的文章 - 知乎
<https://zhuanlan.zhihu.com/p/74746503>
- 【7】【Graph Embedding】node2vec：算法原理，实现和应用 - 浅梦的文章 - 知乎
<https://zhuanlan.zhihu.com/p/56542707>
- 【8】node2vec在工业界的应用-《当机器学习遇上复杂网络：解析微信朋友圈 Lookalike 算法》，地址：<https://mp.weixin.qq.com/s/EV-25t2lWT2JJMLhXsz4zQ>
- 【9】graph embedding之node2vec - 张备的文章 - 知乎
<https://zhuanlan.zhihu.com/p/63631102>
- 【10】Graph Embedding在淘宝推荐系统中的应用 - 张备的文章 - 知乎
<https://zhuanlan.zhihu.com/p/70198918>
- 【11】Graph Embedding - 阿里EGES算法 - 王多鱼的文章 - 知乎
<https://zhuanlan.zhihu.com/p/69069878>

【12】Graph Embedding: 深度学习推荐系统的"基本操作" - 顾鹏的文章 - 知乎

<https://zhuanlan.zhihu.com/p/68247149>

【13】论文阅读:Billion-scale Commodity Embedding for E-commerce Recommendation in Alibaba, 地址: https://blog.csdn.net/Super_Json/article/details/85537938



长按二维码扫描关注

Microstrong

ID:MicrostrongAI

Microstrong(小强)同学主要研究兴趣是机器学习、深度学习、推荐系统、自然语言处理等相关内容, 分享在学习过程中的读书笔记! 期待您的关注, 欢迎一起学习交流进步!