

## 关于论文《Community Preserving Network Embedding》的学习报告

网络嵌入主要是用来把网络中的节点在低维空间中表示出来，并且要求这一过程能够保存下尽量多关于原本网络的信息，包括网络连接信息与网络的社区信息等。目前存在的大多数网络嵌入研究方法都是考虑的是网络的连接信息，也就是只考虑了网络中各个节点的连通性（包括直接连接和间接连接），但是却忽略了网络的社区信息。因此，在这篇论文中，作者们提出了一个M-NMF方法，处理网络嵌入问题的时候，在原来的网络连接的信息上同时考虑网络的社区架构，从而从更充分的角度对网络嵌入问题进行研究。以下是算法的具体内容：

1. 算法主要糅合了不同的两个模型，而这两个模型是从不同的角度表征了网络的某些特性。首先第一个模型是基于模块性这一概念，根据Newman的提出，这个概念的定义如下：

$$Q = \frac{1}{4e} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2e}) h_i h_j,$$

模块性衡量的的是一个社区的划分效果情况，Q值越大表示社区的划分效果越好。从公式上解释， $A_{ij}$ 是原来存储网络信息的邻接矩阵，而 $k_i$ 是i节点的度， $e$ 是边的总数，因此 $2e$ 就是整个网络图中的总度数，而 $\frac{k_i k_j}{2e}$ 表示的是当网络中的边是任意添加的时候节点ij之间期望出现的边数（这一点没想通）。因此，这个公式衡量的是当前图划分的实际情况与理想的划分情况之间的差别，当实际划分情况越接近理想的划分情况时，表明社区的划分效果越好。

将公式用矩阵的形式表示出来，就得到：

$$Q = \text{tr}(\mathbf{H}^T \mathbf{B} \mathbf{H}), \quad \text{s.t.} \quad \text{tr}(\mathbf{H}^T \mathbf{H}) = n, \quad (2)$$

其中 $\text{tr}$ 函数是计算给定矩阵的对角线上元素的和，公式中的H矩阵是声明某一节点在社区划分以后属于的第几个社区中，后面的约束条件是为了保证每一个节点只能划分到一个社区中。基于网络的模块性，我们得到了第一个目标函数，最大化公式（2），这是从网络的划分效果这个角度来考虑的，在考虑网络嵌入进行降维的时候保证网络的划分效果尽量好。

2. 第二个模型是网络的微观架构，也就是网络的架构包含的信息。在论文中，作者同时考虑了网络架构中较为重要的两种信息，分别是一阶连通与二阶连通，所谓一阶连通就是原本网络中节点之间的直接连通情况，当两个节点之间有边连接到一起的时候，表明这两个节点具有很大的相似度；而二阶连通是通过余弦相似度计算得到的，余弦相似度计算的是两个向量之间的正交性情况，数值越大表示二者越相似。

在考虑了网络微观架构的一阶与二阶连通以后，就得到了一个网络的相似度矩阵， $\mathbf{S} = \mathbf{S}^1 + \eta \mathbf{S}^2$ ，这个相似度矩阵有一阶相似度与二阶相似度组成。

得到了网络微观架构的相似度矩阵以后，论文采用了一种非负矩阵分解的方法把原来的矩阵分解成两个权重矩阵M与特征矩阵U（两个矩阵都

是非负的)，从而实现矩阵的降维效果，把原来的多维矩阵压缩成低维的矩阵存储，而分解后的特征矩阵U的每一行向量就是原来矩阵的每一行向量在新空间中的表示，也即是网络中每一个节点的表示。于是，我们就得到第二个目标函数，也就是从网络微观架构的角度考虑的损失函数：

$$\min \|S - MU^T\|_F^2 \quad s.t. \quad M \geq 0, \quad U \geq 0. \quad (3)$$

这个函数的意义是让分解后的两个矩阵尽量地去拟合原始矩阵，使相似度矩阵分解成的两个矩阵尽可能多地包含原来矩阵的信息，达到分解降维的效果的情况下又不损失太多信息的目的。

3. 在提出了上面的两个单独模型以后，这篇论文的主要思想就是综合上述的两个模型来进行网络嵌入问题的研究，从模型2中，我们得到了原始网络的降维表示，存储到特征矩阵U中，因此我们可以通过对特征矩阵U进行社区划分，从而将降维后的特征矩阵应用到社区划分问题中，这里引入了一个辅助矩阵C，而矩阵U\*C得到的结果就是每个节点属于每个社区的可能性，那么这个可能性应该用什么作为衡量正确性的指标的，这就要用到第一个模型模块性的概念了。由于算法需要最大化网络的模块性，而网络的模块性存储在矩阵H中，因此我们可以通过U\*C与H的差值来表征网络嵌入的结果的社区划分效果，因而得到最终的目标函数：

$$\min_{M,U,H,C} \|S - MU^T\|_F^2 + \alpha \|H - UC^T\|_F^2 - \beta \text{tr}(H^T B H)$$

$$s.t., M \geq 0, U \geq 0, H \geq 0, C \geq 0, \text{tr}(H^T H) = n,$$

4. 感想：这篇论文的思想比较简单直接，考虑的问题是网络嵌入问题，网络嵌入问题是基于一个图，将节点或者边投影到低维向量空间中，再用于后续的机器学习或者数据挖掘任务，也就是一个降维方法，因此算法考虑的一个最重要的指标是降维后的结果不能丢失太多的信息；而这篇论文做到的很优秀的一点是用模块性这个指标辅助衡量降维的效果。因为降维只是一种数据处理手段，主要还是要应用到不同的具体问题中，而对于不同的问题，不仅仅要考虑尽量保证降维后的结果能保留尽量多的信息，更重要的是从问题出发提出适用于这个问题的解决方法。正是出于这个目的，论文的作者在社区检测领域提出的这个网络嵌入方法，考虑的一个重要指标是社区划分的效果，用社区划分效果与降维信息丢失情况共同作为惩罚项来联合优化这两部分。

## 关于论文《Node Attribute-enhanced Community Detection in Complex Networks》的学习报告

在一个网络中,可能存在某些节点在属性的取值方面有很大的相似性,但是这些节点之间并没有直接连接,尤其是对新生成的网络而言这种情况就更加普遍,从而导致了网络的稀疏性问题。而在现实世界中,假如两个节点的属性的取值很相似,那么这两个节点就很有可能是属于同一类节点,也就是属于聚类结果的同一个簇中。基于这一思想,作者们在论文中提出了一种kNN-enhance算法,利用节点之间的属性相似度,在原来网络的基础上人为地给这些基于属性相似的节点添加边,缓解了原来存在的稀疏网络存在的问题,同时强化了网络的结构,并把网络中的节点属性转换为边属性。此时再利用聚类算法对强化的网络结构进行聚类的时候既能考虑网络的原始结构,又能考虑网络中节点的属性,使聚类结果能糅合更多的现实信息,得到考虑更周到的结果。以下是算法的细节:

1. kNN-enhance: 对于一个给定的属性网络,我们可以通过kNN-enhance算法为这个原始网络中的节点之间增加一些边,而这些边的增加原则是: 对于一个特定的节点,计算出这个节点与同一网络中不同节点之间的相似度,选取相似度最接近的k个节点,然后对原始的网络进行操作,假如这个特定的节点与这k个节点之间本来存在连接,那么就无操作;假如原来不存在连接,则给这两个节点之间增加一个连接。相似度的计算方法是对于标称型属性数据,采用余弦相似度;对于数值型属性数据,采用欧拉距离;最后结合这两种类型的属性数据得到两个节点的相似度。因此,一个新的网络图(kNN图)就这样构建出来了,而后续的聚类算法就能应用到这个KNN图上从而达到比原始网络图更好的聚类效果。
2. cluster-dp算法: 该算法基于两个参数表征一个数据点作为聚类中心的可能性: 第一个是局部密度,表征了该数据点附近点的密集程度,可是局部密度不能唯一表征一个聚类中心,因为可能存在两个具有高局部密度的点距离很近,这样也只是属于同一个类。于是,算法的作者们很巧妙地提出了第二个参数——高密度最近零距离,表征了局部密度比某个数据点大的点中离它最近的那一个点。考虑这样一种情况,如果一个点拥有较大的高密度最近零距离,表示密度比它大的点都离它较远,如果此时该点的局部密度也较大(排除掉离散点的情况),那么这个点作为聚类中心的可能性就很大。这两个参数的计算公式如下:

$$\rho_i = \sum_j \chi(d_{ij} - d_c), \delta_i = \min_{j: \rho_j > \rho_i} (d_{ij})$$

3. 虽然这个算法能很巧妙表征出一个聚类中心,但是这个算法存在一些很严重的缺点: 首先是当网络架构不明确的时候无法在决策图上明确聚类中心与其他点的界限,也就无法判断出哪些是聚类中心。其次是参数 $d_c$ 的影响,我上学期发表的通信原理论文,做的研究就是基于这个算法的,其中对参数 $d_c$ 印象非常深刻,当时为了让聚类中心能与其他数据点有一个明确的分界线,我们一直在调密度的计算方式以及参数 $d_c$ ,二者对决

策图上的结果是起到非常重要的影响的，这个算法的核心就是决策图，假如无法跑出一个具有明显区分性的决策图，这个算法的效果是非常差的。

4. 为了克服上述算法存在的缺陷，论文中作者们提出了与cluster-dp类似的K-rand-D算法。在K-rand-D算法中，聚类中心的表征变成了两个参数，分别是中心性程度与分散性程度。其中所有节点的中心性程度都可以通过PageRank centrality算法的公式计算得到：

$$v^{t+1} = \left( (1 - \beta)P + e \frac{\beta}{n} \right) v^t, \quad P_{ij} = \frac{A_{ij}}{\sum_j A_{ij}}, \quad i, j \in \{1, 2, \dots, n\}$$

迭代多次后向量v的每一项都表示该节点的中心性程度，这个对应于cluster-dp算法的局部密度表征聚类中心的思想，而这个公式有效地消除了敏感参数 $d_c$ 的影响。

其次是分散性的计算，分散性就对应于cluster-dp的高密度最近邻距离，用于排除由于离聚类中心较近导致中心性程度大的非聚类中心点。而分散性程度用的距离计算方式如下：

$$S = (A + I)^T, \quad \bar{s}_{ij} = S_{ij} / \sqrt{\sum_j S_{ij}^2}, \quad d_{ij} = \|\bar{s}_i - \bar{s}_j\|_2.$$

得到了网络中每个节点的中心性程度和散列性程度以后，K-rand-D算法没有采用决策图的方法，而是采用了两个参数的联合公式作为指标，用以选择聚类中心：

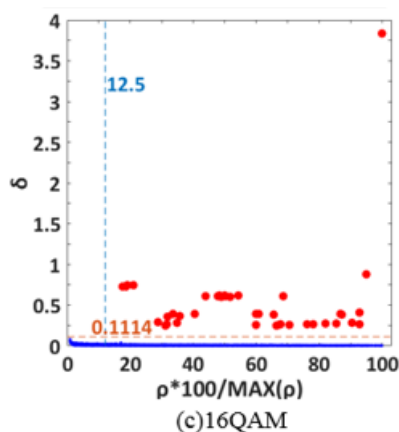
$$CV(i) = v_i \cdot \bar{\delta}_i / (\max_{j=1}^n (v_j) \cdot \max_{j=1}^n (\bar{\delta}_j)).$$

5. 在选择了k个初始聚类中心以后，就可以应用kNN或K-Means算法把其余的点划分到对应的聚类中心，也就是不同的簇中，得到最终的聚类结果。

感想：首先我对作者们kNN-enhance算法思想的提出感到非常佩服，这种利用网络中节点的相似度人为地增加两个节点的思想很巧妙地在考虑节点连接情况的同时也用到了节点的原来属性，使聚类过程考虑的准则更全面。但是我想提出的一点是这个算法可能并不非常适合K-rand-D，因为正如cluster-dp算法一样，这两个算法都是基于密度进行聚类的，中心性与散布性其实也是以密度作为本质的吧，而kNN-enhance算法在原来的网络中增加有权边的思想则更侧重于节点之间的连接性。作者在论文中虽然没提到，但是我认为作者使用kNN-enhance算法只是计算了少数对的节点之间的距离，把这些点设为连接的点，而其他不连接的点对的距离设为0，但是这种基于密度的聚类方法这样做是不太妥的吧。因此我觉得kNN-enhance算法的思想是很优秀的，假如它再结合了图划分算法而非基于密度聚类的算法可能会达到更好的效果。除此以外，对于K-rand-D最后采用的两个参数的联合公式来选取最终的k个初始质心的思想我也存在怀疑。我觉得这样做可能会丢失一些决策图中的信息，我的想法是与cluster-dp一样把这两个参数映射到一个二维决策图中。针对无法正确划分聚类中心与其他数据点的边界这个问题，其实很多研究者已经对cluster-dp这个算法进行解决过，正如我上学期读到过的一篇



《Automatic Clustering via Outward Statistical Testing on Density Metrics》论文就用概率论与统计学的方法寻找outliers解决了。而从我之前投的通信原理论文中对这个算法的研究的经验，这些聚类中心点与其他数据点之间虽然没有存在一个明确的边界，但是有一个很明显的特征就是在决策图上这些点都是比较突出的，这些点附近的点比较稀疏，而其他的非聚类中心点基本上聚集在x轴附近，分布情况大致如下图：



做完那篇通信原理论文以后我有一个新想法，结合K-rand-D算法说明如下：第一次的K-rand-D我们是从原始的网络图中找中心性程度和散列性程度都大的点，此时我们得到了第一次的决策图。我们需要从这个决策图中找到聚类中心，而这些聚类中心在决策图中都是稀疏分布的outliers，因此我们可以把第一次的决策图再应用一次K-rand-D算法，而这次我们并不是要找那些中心性程度和散列性程度大的点，而是决策图中的分布稀疏的点，因此应该取中心性程度和散列性程度小的点。

# 关于论文《BL-MNE: Emerging Heterogeneous Social Network Embedding through Broad Learning with Aligned Autoencoder》的学习报告

这篇论文也是关于网络嵌入问题，旨在把网络中的数据投影到一个低维的特征空间中，数据在低维特征空间中的表示应该保留尽量多原来网络数据的信息。而这篇论文是从广泛学习的角度进行网络嵌入问题的研究，用以解决新生成网络的稀疏性问题，算法的大致思想是从其他成熟的对齐异构网络中学习网络中节点的知识（节点的属性）从而能传递到新生成的网络（通过anchor link）中，使这个新生成的网络的某些节点在一开始就建立特定的关系，从而缓解初始网络的稀疏性问题。以下是这个算法的细节：

1. 关于属性异构社交网络的概念：一个属性异构社交网络由三部分组成，包括网络中的节点 $V$ 、网络中节点之间的连接关系 $E$ 以及与节点相关的属性 $T$ 。关于多对齐社交网络的概念：多对齐社交网络是由多个属性异构社交网络组成，而这些属性异构社交网络之间存在anchor link的联系，当不同属性异构社交网络中的节点代表同一个实体时，这些节点就存在一个anchor link的关系，而两个属性异构社交网络中的anchor link集合就表示这两个社交网络的相互关系。由多对齐社交网络以及这些网络之间的相互关系组成了多对齐社交网络。BL-MNE问题就是对于给定的多对齐社交网络，把这些社交网络中的属性异构社交网络的用户信息映射到低维的特征空间中存储，同时让这些属性异构社交网络之间能够传递信息，能够从成熟的网络向新形成的网络传递信息，从而克服新生成网络的稀疏性问题。
2. 基于友谊的meta proximity：在同一个社交网络中，假如两个用户是朋友关系，那么在该网络中这两个用户的基于友谊的meta proximity为1，否则为0。
3. 属性增强网络架构：由于新生成的网络中的用户存在很少的朋友关系，导致了该网络中的meta proximity为一个稀疏矩阵，为了解决这个问题，作者们在论文中提出了属性增强网络架构：对于网络中的一个用户节点，对于这个用户拥有的属性，把这些属性也当成一个节点，为用户与这些属性之间添加一个“have”连接，从而把新生成的网络中没有建立朋友关系的用户通过他们之间的属性建立起连接，缓解了新生成网络的稀疏性问题。属性增强元路径：在属性增强网络架构中，由于存在用户节点和属性节点，因此就存在多种节点序列组合（图中的路径）。在论文中定义了序列组合的起始和结束都为用户节点的序列为属性增强元路径，通过属性增强元路径，我们能够基于属性为两个非朋友用户建立特定连接，挖掘出更多的信息。在论文中，针对于论文的数据集，作者们提出了期终属性增强元路径。在这里，前面提到的基于友谊的meta proximity也被认为是属性增强元路径的一种。
4. 同一个社交网络中基于不同的属性增强元路径，两个用户之间的meta

proximity计算方法如下:

$$p_{\Phi_k}^{(1)}(u_i^{(1)}, u_j^{(1)}) = \frac{2|\mathcal{P}_{\Phi_k}^{(1)}(u_i^{(1)}, u_j^{(1)})|}{|\mathcal{P}_{\Phi_k}^{(1)}(u_i^{(1)}, \cdot)| + |\mathcal{P}_{\Phi_k}^{(1)}(\cdot, u_j^{(1)})|}.$$

5. 深度自动编码模型: 深度自动编码模型其实就是一个神经网络架构, 其中的编码步骤是把输入的多维向量 $\mathbf{x}$ 通过多个隐藏层 $\mathbf{y}$ 得到低维向量 $\mathbf{z}$ , 通过如下公式表示这一过程:

$$\begin{cases} \mathbf{y}_i^1 &= \sigma(\mathbf{W}^1 \mathbf{x}_i + \mathbf{b}^1), \\ \mathbf{y}_i^k &= \sigma(\mathbf{W}^k \mathbf{y}_i^{k-1} + \mathbf{b}^k), \forall k \in \{2, 3, \dots, o\}, \\ \mathbf{z}_i &= \sigma(\mathbf{W}^{o+1} \mathbf{y}_i^o + \mathbf{b}^{o+1}). \end{cases}$$

那么如何判断最终得到的低维向量 $\mathbf{z}$ 包含的信息是否与原来的多维向量 $\mathbf{x}$ 尽量相同呢? 这里就要触发一个解码操作, 把低维向量 $\mathbf{z}$ 反向通过神经网络得到原来的多维向量的预测 (类似于BPNN神经网络), 以下公式表示了这一过程:

$$\begin{cases} \hat{\mathbf{y}}_i^o &= \sigma(\hat{\mathbf{W}}^{o+1} \mathbf{z}_i + \hat{\mathbf{b}}^{o+1}), \\ \hat{\mathbf{y}}_i^{k-1} &= \sigma(\hat{\mathbf{W}}^k \hat{\mathbf{y}}_i^k + \hat{\mathbf{b}}^k), \forall k \in \{2, 3, \dots, o\}, \\ \hat{\mathbf{x}}_i &= \sigma(\hat{\mathbf{W}}^1 \hat{\mathbf{y}}_i^1 + \hat{\mathbf{b}}^1). \end{cases}$$

模型的目的是在编码步骤得到的低维向量 $\mathbf{z}$ 解码以后得到的原始向量的估计就可能接近原始向量, 因此得到以下目标函数并使其最小化:

$$\mathcal{L} = \sum_i \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2.$$

6. Deep DIME-SH 模型: 在深度自动编码模型的基础上, 考虑多个属性增强元路径, 对每一条属性增强元路径都单独计算到最后一个隐藏层, 最后把每条路径的最后一个隐藏层的输出结合得到最终的低维空间表示, 因此编码和解码步骤过程可以用如下公式体现:

$$\begin{cases} \text{\# extra encoder steps} \\ \mathbf{y}_i^{(1), o+1} = \sigma(\sum_{\Phi_k \in \{\Phi_0, \dots, \Phi_7\}} \mathbf{W}_{\Phi_k}^{(1), o+1} \mathbf{y}_{i, \Phi_k}^{(1), o} + \mathbf{b}_{\Phi_k}^{(1), o+1}), \\ \mathbf{z}_i^{(1)} = \sigma(\mathbf{W}^{(1), o+2} \mathbf{y}_i^{(1), o+1} + \mathbf{b}^{(1), o+2}). \\ \text{\# extra decoder steps} \\ \hat{\mathbf{y}}_i^{(1), o+1} = \sigma(\hat{\mathbf{W}}^{(1), o+2} \mathbf{z}_i^{(1)} + \hat{\mathbf{b}}^{(1), o+2}), \\ \hat{\mathbf{y}}_{i, \Phi_k}^{(1), o} = \sigma(\hat{\mathbf{W}}_{\Phi_k}^{(1), o+1} \hat{\mathbf{y}}_i^{(1), o+1} + \hat{\mathbf{b}}_{\Phi_k}^{(1), o+1}). \end{cases}$$

因此最终的损失函数变成:

$$\mathcal{L}^{(1)} = \sum_{\Phi_k \in \{\Phi_0, \dots, \Phi_7\}} \sum_{u_i \in \mathcal{V}} \left\| \left( \mathbf{x}_{i, \Phi_k}^{(1)} - \hat{\mathbf{x}}_{i, \Phi_k}^{(1)} \right) \odot \mathbf{b}_{i, \Phi_k}^{(1)} \right\|_2^2,$$

这里的 $\mathbf{b}$ 是一个权重向量, 为了解决初始网络的稀疏性问题不会导致在损失函数中网络信息的大部分丢失。

7. 深度DIME框架: 深度DMIE框架是糅合了多个网络的信息并允许这些网络之间进行传递信息, 从而使得成熟的网络包含的部分信息能够传递到

新生成的对齐网络中，解决网络的稀疏性问题。深度DIME框架的算法思想如下：给定两个对齐异构网络G1和G2，应用网络嵌入方法，可以把这两个网络的信息通过两个低维特征向量Z1和Z2表示出来，这里的Z向量跟前面提到的自动编码模型中的z是对应的。两个网络之间的传递信息是通过anchor link来实现的，而anchor link可以通过一个矩阵T表示，当网络G1中的i用户和网络G2中的j用户表示同一个用户时，T(i,j)=1否则为0。为了解决两个网络中特征向量维度不一样的问题，论文中引入了一个W矩阵进行投影使得两个网络的特征向量能够进行比较，因此得到以下的目标函数：

$$\mathcal{L}^{(1,2)} = \left\| (\mathbf{T}^{(1,2)})^\top \mathbf{Z}^{(1)} \mathbf{W}^{(1,2)} - \mathbf{Z}^{(2)} \right\|_F^2$$

这个公式的解释意义是：给定存储了同一批用户信息的两个异构网络，对于用户来说，这些信息应该是一样的，而向量Z是网络中用户信息的特征向量表示，每一行表示的是特定用户的特征空间的表示，虽然对于不同的网络，但是由于是同一个用户，因此用户信息应该是一样的，也就是在特征空间中的表示应该就可能相似。于是就可以通过最小化目标函数实现两个不同异构网络的互相拟合。



## 关于论文《Broad Learning based Multi-Source Collaborative Recommendation》的学习报告

这篇论文研究的问题是基于anchor link的跨网络推荐问题，使用的思想与上一篇论文的部分思想很类似，都是运用广泛学习的知识在不同的网络中通过anchor link传递信息，继而进行两个网络把自身学习到的知识互相传递，使得两个网络能掌握更多的知识。论文中，作者们针对推荐问题基于广泛学习的思想提出了CCMF推荐框架，集成了多个网络的信息并能缓解网络的稀疏性问题，以下是算法的具体思想：

### 1. Low-Rank Matrix Factorization (LRMF) :

$$\min_{U,V} L = \frac{1}{2} \sum_{i,j} W_{i,j} (R_{i,j} - U_i V_j^T)^2 + \frac{\lambda}{2} (\|U\|^2 + \|V\|^2)$$

关于用户-商品评价的矩阵R可以分解成U和T，其中矩阵U和V分别表示用户和商品的潜在分布情况。这个分解的目标函数是使分解后得到的U和V矩阵尽可能多地包含原始矩阵的信息。而目标函数中后面一项是正则化，添加正则化项是为了避免过拟合，这种方法在人工智能实验课上已经学习过。

2. CCMF框架：基于LRMF算法，当研究的对象涉及到两个网络中的用户-商品评价信息时，作者们在论文中提出了CCMF框架，这个框架考虑了多方面的内容，首先是各个网络内部的信息，与LRMF的分解思想一样，我们首要目标也是令分解后的U和V矩阵尽量拟合原始矩阵。除此以外，论文综合有一点考虑得很周到的是保证分解以后V矩阵中两个商品的相似度并没有太大的改变（由于这是针对商品进行推荐，因此可以忽略用户矩阵U的相似情况）。因而对于各个网络，得到了以下的子目标函数：

$$Reg^k = \frac{1}{2} \sum_{i=0}^a \sum_{j=0}^b W_{i,j}^{(k)} (R_{i,j}^{(k)} - U_i^{(k)} V_j^{(k)T})^2 + \frac{\gamma}{2} \sum_{i=0}^b \sum_{j=0}^b S_{i,j}^{(k)} \|V_i^{(k)} - V_j^{(k)}\|^2$$

其中，相似度的计算公式为：

$$S = \left\{ S_{i,j} \mid S_{i,j} = \frac{M_i M_j^T}{\|M_i\| \|M_j\|} \right\}$$

对于商品的相似度的计算问题，由于两个商品的相似度情况并不应该取决于不同的网络，而是取决于商品的本身特性，因此对于同一对商品，在不同的网络中它们的相似度应该是一样的。在论文中，作者们的处理方法是基于信息量这个指标：

$$Q_{i,j}^{(1)} = \sum_k W_{k,i}^{(1)} W_{k,j}^{(1)}, Q_{i,j}^{(2)} = \sum_k W_{k,i}^{(2)} W_{k,j}^{(2)}$$

对于两个不同的网络，选择商品具有更大信息量的网络中的信息来进行计算，从而统筹了两个不同的网络针对同样的实体进行统一，最终得到以下的相似度计算公式：

$$S_{i,j}^{(1)} = \begin{cases} \frac{M_i^{(1)} M_j^{(1)T}}{\|M_i^{(1)}\| \|M_j^{(1)}\|}, & \text{if } Q_{i,j}^{(1)} > \tilde{Q}_{i,j}^{(2)} \\ \frac{\tilde{M}_i^{(2)} \tilde{M}_j^{(2)T}}{\|\tilde{M}_i^{(2)}\| \|\tilde{M}_j^{(2)}\|}, & \text{if } Q_{i,j}^{(1)} < \tilde{Q}_{i,j}^{(2)} \\ \frac{M_i^{(1)} M_j^{(1)T} + \tilde{M}_i^{(2)} \tilde{M}_j^{(2)T}}{(\|M_i^{(1)}\| + \|\tilde{M}_i^{(2)}\|)(\|M_j^{(1)}\| + \|\tilde{M}_j^{(2)}\|)}, & \text{otherwise} \end{cases}$$

与上一篇论文类似，虽然这是考虑两个不同网络中的商品实体进行推荐，但是这些实体应该具有统一性而不应该由于网络的不同导致商品的表示矩阵V不同，通过anchor link的跨网络传递以及维度统一进行比较以后，我们得到以下的目标函数，要优化该目标函数从而使得两个网络中的相同实体的表示形式(V矩阵)得到统一，从而达到广泛学习的目的。

$$Reg^3 = \frac{1}{2} \|T^T V^{(1)} H - T^T T V^{(2)}\|^2$$

最后就是通过增加正则化项避免算法出现严重的过拟合现象。

# 关于论文《BL-ECD: Broad Learning based Enterprise Community Detection via Hierarchical Structure Fusion》的学习报告

这篇论文同样是关于广泛学习这方面的，研究的问题同时综合考虑公司中与员工相关的线上和线下信息，整合来个多方面的不同源的信息来进行社区发现研究。为了解决论文中提出的基于广泛学习的企业社区探测问题，论文提出了异构多源聚类算法（HUMOR），算法分为两部分，分别是各个信息源的内部社区结构的信息提取得到局部信息，以及考虑线上与线下信息源的交互得到全局信息。以下是算法的具体内容：

1. 线上企业社交网络的社交亲密度：这一概念是衡量企业内部线上网络中两个用户的亲密程度，这一点可以基于社交网络中以后之间的连接，通过比较这两个用户的共同好友的数量来衡量，假如两个用户有较多的共同好友，那么这两个用户的社交亲密度就较高。在论文中，作者们是使用信息论中互信息这个指标再结合共同好友来计算两个用户的社交亲密度的，计算公式如下：

$$EI_s^g(u, v) = \frac{|\Gamma_s^g(u) \cap \Gamma_s^g(v)|}{|\mathcal{U}|} \log \frac{\frac{|\Gamma_s^g(u) \cap \Gamma_s^g(v)|}{|\mathcal{U}|}}{\frac{|\Gamma_s^g(u)|}{|\mathcal{U}|} \cdot \frac{|\Gamma_s^g(v)|}{|\mathcal{U}|}}.$$

2. 基于群组的社交亲密度：假如企业中两个用户同时属于同一个群组中，那么这两个用户的社交亲密度应该较高；并且随着同属的群组的数量越多，社交亲密度越高。基于群组的社交亲密度的计算公式如下：

$$EI_g^g(u, v) = \sum_{g \in \Gamma_g^g(u) \cap \Gamma_g^g(v)} IMF(g) = \sum_{g \in \Gamma_g^g(u) \cap \Gamma_g^g(v)} \log \frac{|\mathcal{U}|}{|\Gamma_g^g(g)|}.$$

3. 基于文本内容的社交亲密度：假如两个用户在社交网络上发布的内容相似，或者内容有联系，那么我们可以认为这两个用户存在一定的亲密关系。基于文本内容的社交亲密度的计算公式如下：

$$EI_p^g(u, v) = JC(\Gamma_p^g(u), \Gamma_p^g(v)) = \frac{|\Gamma_p^g(u) \cap \Gamma_p^g(v)|}{|\Gamma_p^g(u) \cup \Gamma_p^g(v)|}.$$

4. 根据上述三点线上企业社交网络的社交亲密度的计算标准，我们能得到三个亲密度矩阵，因此此时可以提取出线上网络的内部信息。论文中作者的做法是应用NMF算法把三个亲密度矩阵分解成三个社区隐含因子矩阵 $U_x$ 。最终的目标函数就是找到一个线上社交网络的U去尽量同时拟合基于三个指标得出的 $U_x$ ：

$$J_R(A_s^g, A_g^g, A_p^g) = \|U_s - U\|_F^2 + \|U_g - U\|_F^2 + \|U_p - U\|_F^2$$

看到论文最后的组合模型的优化那里我对这里的做法有一些疑问，对三

个矩阵分解得到三个 $U_x$ ，目标函数对这三个 $U_x$ 都要进行更新，既然这三个 $U$ 的本质表示的都是社交亲密度，那么为什么不在这里把基于不同指标的三个亲密度矩阵变成一个总的线上社交亲密度矩阵，再对这个总的矩阵进行NMF分解，这样一来就不会导致后面出现11个变量需要更新的情况。

5. 提取企业的线下内部信息：企业的线下内部信息是从三方面得到，第一部分是企业的组织架构图，也就是企业内部员工的职位等级，等级越接近的员工就存在更亲密的关系；第二部分是工作内容所属的类，当两个员工所处理的工作内容越接近时，那么就存在很大的可能性在这两个员工存在较亲密的关系；第三部分是员工的工作地点，相同工作地点的员工的亲密度一定比不同工作地点的员工的亲密度大。跟线上信息类似，基于这三部分线下信息，我们得到三个线下亲密度矩阵，也是采用NMF算法把亲密度矩阵分解成线性亲密度矩阵隐含因子矩阵 $V_x$ ，用以探测社区的内部信息。由于这里的三种线下信息的本质不一样，因此我并不确定对于上面提到的整合三种线上信息的方法用到这里是否有一定的合理性。
6. 分别得到企业社区的线上与线下信息，我们就可以通过广泛学习的知识通过两个使信息源的互相传递信息进而得到一个更全面的模型。与之前的两篇与广泛学习相关的论文类似，广泛学习的目标函数就是要使两个信息源得到的网络representation尽可能保持一致，因此产生了以下的目标函数：

$$J_D(U, V) = \left\| (T^T U)(T^T U)^T - VV^T \right\|_F^2$$

对于广泛学习的感想：通过学习上述三篇与广泛学习相关的论文，发现这个课题的方法都是一个套路，这一点从与广泛学习相关的目标函数就可以很容易发现，所谓广泛学习就是在考虑不同信息源或来自不同网络中的实体的时候，虽然这些实体以不同的形式存在于不同的地方，但是在现实世界中这些实体不应该随着他们的呈现空间的不同而不同，而应该取决于实体本身的特性。基于这一个大前提，我们可以通过实体在这些不同空间中的表现形式的互相拟合（其实就是一个实体在不同形式间互相学习的过程），最终得到实体的统一表现形式。