关于论文《Multi-view Clustering with Graph Embedding for Connectome Analysis》的学习报告

基于多视图的聚类方法能够集成多个视图的信息，从而可以利用多个视图都共有的信息、或通过不同视图的信息的互补作用，因此能够更全面地考虑来自多方面的信息对数据进行聚类，达到更好的效果。当前存在的大多数多视图聚类方法利用的都是基于向量的数据表示作为数据的特征，然而，现实世界中存在一些用图作为实例的数据类型，因此曾经基于向量表示的多视图聚类方法不能直接应用到基于图表示的多视图聚类问题中。为了解决这个问题，作者在论文中提出了一个基于图嵌入的关于图实例的多视图聚类框架。在这一框架中，作者把多视图的图数据表示成张量，接着利用张量分解得到多视图的图嵌入结果，从而捕获图的局部架构。然而，仅有图的局部架构的图嵌入表示损失了原始数据中的全局信息，因此，作者们在图嵌入的基础上增加图的多视图聚类，从而能够考虑图的全局架构。接着，二者的结果相互为对方提供下一次更新的信息，从而使得图的局部信息与全局信息不断地增加到算法的学习过程中，直至算法收敛。接下来是对算法细节的讨论：

一、　MCGE 框架中遇到的一些 Notations：

a) 张量：一般一维数组，我们称之为向量（vector）,二维数组，我们称之为矩阵（matrix）;三维数组以及多维数组，我们称之为张量（tensor）。简而言之，在多视图的图表示这个问题上，每一个图可以表示成一个矩阵的形式，假如要表示一个视图的信息（该视图包含多个图实例），可以在矩阵的基础上拓展一个维度，该维度就是表示同一视图中的多个图的维度。而这个拓展的表示就是张量。

b) outer product：向量集x_k（$k = 1，2，3\dots\dots n$）的 outer product 得到的结果是取每个向量的任意一个数出来相乘得到的 n 维空间表示。

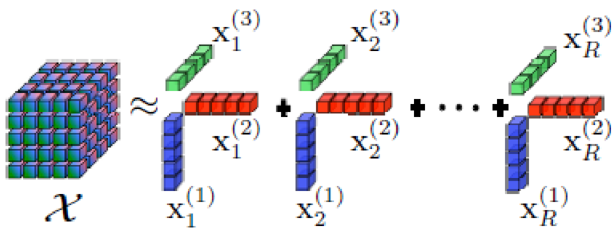
c) mode-k 分解：对于一个张量$X \in R^{I_1*I_2*I_3***I_n}$，它的 mode-k 分解就是把这个多维表示的张量表示成一个矩阵，把第 k 维作为矩阵的其中一维，其他维在同一个维度进行拼接得到矩阵的另一维。

d) CP 分解：CP 分解在张量中的作用相当于矩阵分解在矩阵中的作用。类似的，在对多视图进行图嵌入的时候可以把图表示成张量的形式，再对张量进行 CP 分解，分解得到的结果就是图的嵌入表示。

CP 分解的公式为：

$$\mathcal{X} = [\![\mathrm{X}^{(1)}, \cdots, \mathrm{X}^{(n)}]\!] \equiv \sum_{r=1}^{R} \mathbf{x}_r^{(1)} \circ \cdots \circ \mathbf{x}_r^{(n)},$$

CP 分解的图例形式是：



二、 MCGE 框架

a) 多视图的图嵌入：要对不同视图中的同一个图进行图嵌入，为了糅合该图实例在不同视图中的信息，MCGE 框架首先把该图在不同视图中的实例表示成张量的形式，也就是把图矩阵扩展一维得到视图维度。然后对张量进行 CP 分解得到该图的图嵌入表示，因此得到以下的目标函数：

$$\min_{\mathrm{F_i, H_i}} \left\| \mathcal{T}_i - [\![\mathbf{F}_i, \mathbf{F}_i, \mathbf{H}_i]\!] \right\|_F^2$$

$$\text{s.t. } \mathbf{F}_i{}^{\mathrm{T}} \mathbf{F}_i = \mathbf{I}_c$$

其中F_i表示糅合了 v 个视图的信息后图 i 的嵌入表示的结果，这里有个不理解的问题是约束条件$F_i{}^T F_i = I_c$的含义是什么？

由于对图的嵌入表示的结果只提炼了某个图的局部信息而没有考虑该图在整个网络体系中的全局信息。为了全面考虑这些信息，在图嵌入时候考虑了结果对聚类的影响。考虑这样一个事实，对同一个簇中的两个图，这两个图在簇中的位置越接近，它们就具有更相似的局部结构。因此目标函数修正为：

$$\min_{\mathrm{F_i, H_i}} \left\| \mathcal{T}_i - [\![\mathbf{F}_i, \mathbf{F}_i, \mathbf{H}_i]\!] \right\|_F^2 + \beta \left\| \mathbf{F}_i - \sum_j w_{ij} \mathbf{F}_j \right\|_F^2 \quad (8)$$

$$\text{s.t. } \mathbf{F}_i{}^{\mathrm{T}} \mathbf{F}_i = \mathbf{I}_c$$

其中w_{ij}表示图 j 对图 i 的权重。我对目标函数的第二项的个人理解为：同一个簇中的多个图存在相互影响，每一个图的构成都来自同一个簇中其他图的相似度的影响程度的叠加。这一点我觉得有点像三角形的三个点与重心的关系。

b) 利用图嵌入的多视图聚类：聚类要用到图与图之间的相似度信息，为了融合不同视图的整体信息，论文首先对于基于 graph kernel 方法计算每个视图中所有图之间的相似性，在糅合所有视图的相似度信息得到张量 X，对 X 进行 CP 分解得到图的整体相似度嵌入式表示 B，这一结果可以用来进行聚类。

为了利用图嵌入的结果 F，论文也利用 F 计算不同图之间的相似性 S。

$$s_{ij} = 1 - \left\| \mathbf{F}_i - \mathbf{F}_j \right\|_F^2$$

然后利用谱分析方法结合这两个方面的图相似度，利用以上两种相似度的谱分析同时考虑了图的局部与全局信息，得到以下的目标函数：

$$\min_{\mathrm{B}} = \sum_{i,j=1}^{n} s_{ij} \left\| \frac{\mathbf{b}_i}{\sqrt{d_{ii}}} - \frac{\mathbf{b}_j}{\sqrt{d_{jj}}} \right\|_2^2 = \mathrm{Tr}\left(\mathbf{B}^{\mathrm{T}} \mathbf{L} \mathbf{B}\right)$$

$$\text{s.t. } \mathbf{B}^{\mathrm{T}} \mathbf{B} = \mathbf{I}_k$$

因此在聚类步骤得到最终的目标函数为：

$$\min_{\mathrm{B, A}} \left\| \mathcal{X} - [\![\mathbf{B}, \mathbf{B}, \mathbf{A}]\!] \right\|_F^2 + \alpha \mathrm{Tr}\left(\mathbf{B}^{\mathrm{T}} \mathbf{L} \mathbf{B}\right)$$

$$\text{s.t. } \mathbf{B}^{\mathrm{T}} \mathbf{B} = \mathbf{I}_k$$

增加了图的嵌入表达的相似度作为惩罚项，得到的最终 B 不仅考虑了图的全局信息，也能包含图嵌入过程中得到的局部信息。对 B 矩阵利用 k-Means 算法得到最终的聚类划分结果。这一聚类结果能用来更新权重矩阵 W。更新的步骤产生了以下的目标函数：

$$\min_{\mathrm{W}} \sum_i \left\| \mathrm{X}_i - \sum_j w_{ij} \mathrm{X}_j \right\|_F^2$$

$$\text{s.t.} \sum_j w_{ij} = 1$$

此处 X 是一个包含了图的亲近性与多视图的图的嵌入表示结果的矩阵。

c) MCGE 的整体框架：首先对于每个图实例G_i，考虑该图在多个视图中的信息得到图 i 的张量T_i，接着利用T_i的 CP 分解得到多视图的图嵌入，这一结果部分取决于相同簇中其他图的嵌入结果的影响。同时，通过 graph kernel 计算每个视图中不同图之间的相似性，并整合所有视图得到张量 X，得到整合了所有视图的图的相似度信息。对 X 进行 CP 分解，得到相似度信息的嵌入表示 B。为了结合图嵌入的结果包含的图的局部信息，得到了基于图嵌入的相似度 S。对 S 和 B 进行谱分析可以得到综合了两方面相似度的信息。增加谱分析作为惩罚项的目标函数得到的 B 可以用来进行聚类，继而计算同一个簇中不同图之间的相互权重 W。W 的计算可以用在下一步迭代中计算图的嵌入的更新。两个步骤交互影响，直至算法收敛。