

Introduction and Graph Structure

我们从怎么利用图形或网络表示数据这一动机开始。网络成为了用于描述复杂系统中交互实体的通用语言。从图片上讲，与其认为我们的数据集由一组孤立的数据点组成，不如考虑这些点之间的相互作用和关系

在不同种类的网络之间进行哲学上的区分是有启发性的。对网络的一种解释是作为现实生活中出现的现象的例子。我们称这些网络为*自然图*。像如下例子：

- 人类社交网络(7+十亿个人的集合)
- 互联网通信系统(电子设备的集合)

网络的另一种解释是作为一种数据结构，可用于解决特定的预测问题。在这种情况下，我们对实体之间的关系更感兴趣，因此我们可以有效地执行学习任务。我们称这些网络为*信息图*，比如：

- 场景图(场景中的对象如何相互关联)
- 相似性网络(其中连接了数据集中的相似点)。

在本课程中，我们将主要学习一些涉及这些系统的组织方式和设计特性的问题。也可以将具有丰富关系结构的数据集表示为用于许多预测任务的图形：在这种情况下，我们希望显式建模关系以获得更好的预测性能。此类预测性任务的一些示例包括

1. *节点分类*，我们在其中预测给定节点的类型/颜色
2. *链接预测*，我们在其中预测两个节点是否链接
3. *社区检测*，我们在其中识别密集链接的节点簇
4. *相似度计算*，我们在其中测量两个节点或网络的相似度

总而言之，网络是一种用于描述复杂数据的通用语言，并且可以应用于各种不同的领域。随着数据可用性的提高和各种计算挑战，学习了解网络可以使人们有能力做出各种各样的贡献。

A Review of Graphs

基本概念

网络/图形（从技术上讲，网络通常是指真实的系统(网络，社交网络等)，而图形通常是指网络的数学表示形式(网络图，社会图等)。在这些笔记中，我们将互换使用这些术语。）被定义为对象的集合，其中一些对象对通过链接连接。我们将对象(节点)的集合定义为 N , 对象之间的交互(边/链接)定义为 E , 将图形定义为 $G(N, E)$ 。

无向图具有对称/双向链接(例如，Facebook上的朋友关系)。我们定义节点度 k_i 为无向图中与节点 i 相邻的边数。那么平均程度是

$$\bar{k} = \langle k \rangle = \frac{1}{|N|} \sum_{i=1}^{|N|} k_i = \frac{2|E|}{N}$$

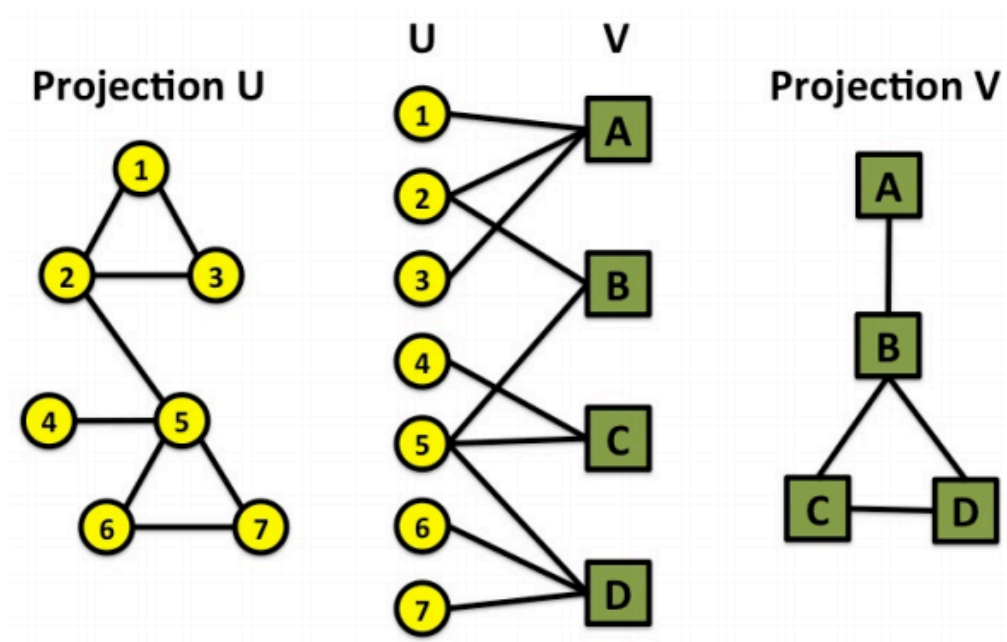
有向图具有定向链接(例如，在Twitter上关注)。我们定义入度 k_i^{in} 为进入节点 i 的边数。同样，我们定义出度 k_i^{out} 为离开节点 i 的边数。

$$\bar{k} = \langle k \rangle = \frac{|E|}{N}$$

完全图，具有最大数量的边的无向图称为完全图(这样所有节点对都被连接)。完全图有 $|E| = \binom{N}{2} = \frac{N(N-1)}{2}$ 条边，并且平均度为 $|N| - 1$ 。

二分图，二分图是其节点可以分为两个不相交的集和 U 和 V ，使得每个边都连接一个集合 U 的节点和一个集合 V 的节点（也就是说，集合 U 内的节点没有边，集合 V 内的节点没有边，我们称 U 和 V 为独立的集合）。如果独立集合 U 和 V 共享至少一个共同的邻居，我们可以通过在独立集合中创建边来“折叠”二分图。

在这里，如果集合 U 中的节点至少共享一个在集合 V 中的邻居节点，则集合 U 中的节点将相连形成投影 U ，采用相同的过程来获得投影 V 。



Representing Graphs

我们使用邻接矩阵 A 来表示图 G , 其中 $A_{ij} = 1$ 表示节点 i 和 j 相连(如果 $A_{ij} = 0$ 则表示没有相邻边)。对于有向图则 A 是不对称的。例如, 一个4节点的有向图的邻接矩阵可以表示为

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

对于无向图,

$$k_i = \sum_{j=1}^{|N|} A_{ij} \quad \text{and} \quad k_j = \sum_{i=1}^{|N|} A_{ij}$$

同样, 对于有向图,

$$k_i^{out} = \sum_{j=1}^{|N|} A_{ij} \quad \text{and} \quad k_j^{in} = \sum_{i=1}^{|N|} A_{ij}$$

但是, 大多数现实世界的网络都很稀疏($|E| \ll E_{max}$, or $\bar{k} \ll |N| - 1$)。结果导致邻接矩阵被大量的零填充(不期望的属性)。

为了缓解此问题, 我们可以将图形表示为一组边(边链表)。这虽然使边缘查找更加困难, 但是节省了内存。

Graph Connectivity

如果无向图 G 图中任意一对节点之间存在路径, 则我们称 G 为强连通图。非强连通图由两个或多个连接的组件组成。如果移除某一条边将强连通图变为非强连通图, 我们称这条边为桥边; 关键节点是指移除改节点后导致强连通图变为非强连通图的节点。具有多个组成部分的网络邻接矩阵可以按块对角线的形式编写(这样, 非零元素将被限制为正方形, 而所有其他元素均为0)。

我们可以将这些概念进一步扩展到有向图, 将强连接的有向图定义为一个有从某个节点到任何其他节点的路径的图, 反之亦然, (即同时具有 $A \rightarrow B$ 和 $B \rightarrow A$ 的路径)。如果忽略边缘方向, 则将形成一个弱连接的有向图。我们进一步将强连接组件(SCCs)定义为 G 的强连接子图。可以到达SCC的节点是其内部组件的一部分, 可以从SCC到达的节点是其外部组件的一部分。

下图虽然已连接但不是强连接图，包含了一个SCC(图 $G' = G[A, B, C]$)。

