Structural Information and Dynamical Complexity of Networks

概论

- [ ] 真实世界的网络结构熵都是低的。
- [ ] 由于新型数据的涌现，原始形式的香农熵已经过时——于是定义网络的结构熵来量度网络节点中的通信、网络的演进等等。
- [ ] 各类结构熵不能很好反映图的各项性质——尤其是真实世界中的复杂网络，且最后结构熵结果是一个实数不利于分析。
- [ ] 本文针对下列场景提出$K-dimensional$结构熵解决相应问题：

  场景

- 图G是嘈杂的，但其本身具有结构 （或说是有结构的噪声图）
- 图G是演化而来的（G是随时间变化的，时序之间有过渡）
- 图G的演化有其自身的规律可寻，同时有$random\ variations$（类比基因突变？）

  问题

- 如何区分真实存在的规律和$random\ variations$对G的作用
- 进一步抓取这种规律
- 同时度量$random\ variations$在图G演进中的作用

图的生成过程

- [ ] 再将以上场景扩展到$dynamical\ complexity\ of\ networks$——即我们要对真实世界的网络进行分析，而这些网络无不快速地迭代，内部节点更是频繁在通信。
- [ ] 由真实网络的问题复杂性，本文假定了$network$或说$graph$的生成过程如下：

- 某些$laws$ →生成$knowledge\ trees$ → 生成$graph$

  而抽取图的规律(想学到的是具体网络背后抽象的$laws$等等)是生成的逆过程：

- 已有的$graph$ ‣ ‣ $knowledge\ trees$ ‣ ‣ $laws$ (有些类似经历$auto-encoder$后再$decode$)

Structural Information by Partition

- [ ] $One-Dimensional$只是简单的香农熵，用到的是度的分布（我理解之所以称为单维是因未用到下述的$hierarchy\ struct$）

- [ ] $Structural\ Information\ by\ Partition$ 属于$Two-Dimensional$ 的信息度量。因其将G的点全集V分割成L 个互不包含点集，每个子集分别度量其结构信息，再进行汇合。公式为：

$$\mathrm{H}^P(G)=-\sum_{j=1}^{L}\frac{V^j}{2m}\sum_{i=1}^{n_j}\frac{d_i^j}{V_j}log_2\frac{d_i^j}{V_j}-\sum_{i=1}^{L}\frac{g_j}{2m}log_2\frac{V_j}{2m}$$

第一项由熵的性质得显然非负，第二项当且仅当某点能 $random\ walk$ 到（有边连接到）不包含该点的点子集时才为非0。我理解：公式的物理意义是第一项刻画了同一点集内的通信量，第二项刻画了不同点集间的通信量。

- [ ] 正式定义$Two-Dimensional\ Structural\ Information\ of\ Disconnected\ Graphs$ ：

(文章常常给出连通图的公式，再类推到非连通图等情况，无非要做加权平均，思想是一致的。)

$$H^2(G)=min_P(H^P(G))$$

- [ ] 将给定点集分割成子集是一个**组合爆炸**问题。如何找到最优的$Partition$ ？遍历显然不可行。此外我疑惑为什么要找min_P ，猜想是出于文章$introduction$ 中的信念——真实世界的网络结构熵都是低的，或说自然界是熵减的。

- [ ] 注：文章中$Partition$是对点集V进行的操作，输出的结果也是一些元素为点的子集。度量这些子集的结构信息时，无疑要用到子集中各点间的边。所以不妨将$Partition$理解为针对给定图G 输出各子图——尤其当我们要用到某点集对应的边的信息时。

向更高维演绎

- [ ] 上节阐述了对图结构信息的二维度量。当图的层次更高维时只对点集做单次$Partition$ 是不够的，于是引入$Partitioning\ Tree$树的高度就是图结构的层数。根据二维情况一贯的思路，则有以下$K-dimensional$ 的结构信息度量：

$$H^K(G)=min_T\{\sum_{\alpha\in T,\alpha\neq\lambda}-\frac{g_a}{2m}log_2\frac{V_\alpha}{V_{\alpha^-}}\}$$

- [ ] 上式刻画的是不同层级间通信的信息量。现实中复杂网络的分层是显而易见的。症结还是在难解出T 。
- [ ] 简记作者列举的$K-dimensional$ 度量优点如下：

- $Network\ dependency$ 较以往的度量更能反映图自身的性质（猜想是引入了$Partition$）。
- $Additivity$ 可加性（这里并未理解）。
- $Locality$ 每个点作为点集和点集全集作为$Partition$结果，这两者的结构信息是相同的。
- $Dynamics$ 图的动态性质反映在其内部的通信行为上。
- $Robustness$ 即图的变化微小时，$K-dimensional$ 度量的变化也是小的（文章只是定性提及）。

- $Incremental\ computability$ 这条性质很有用处。大意是$Partition$时避免将没有边直连的点集，譬如X与Y 划分在同一个$Partition$子元素里。

  (不划分为同一元素，这种现象叫**分隔**，示例：点集X -----------存在某非空分隔点集Z ------------- 点集Y)

解法

- [ ] 上面已经提到，寻找最优的$Partition$ 方案是组合优化问题，**遍历**不可行的情况下，采用启发式算法。
- [ ] 由$Incremental\ computability$ 这条性质带来启示，我们要寻找的就是**没有直连边的点集**尽量**分隔**的$Partition$方案。若两点集X_i和Y_j之间没有直连边的话，则$\Delta_{i,j}^{P}(G) < 0$ ，这也是算法中要用到的条件。

- [ ] 下面简述算法的思想（一阶情况）：

1. 初始化$Partition$ 。
2. 在当前$Partition$中计算$\Delta_{i,j}^{P}(G)$ ，目的是确认点集的分隔情况。如果各点集都是分隔的（即任意的$\Delta_{i,j}^{P}(G)$都非正），则输出当前$Partition$。否则下一步。
3. 找到$max_{i0,\ j0} = \Delta_{i,j}^{P}(G)$，将 $i0$、$j0$ 两索引对应的点集合并加入$Partition$，$Partition$的其余部分不变。转到上一步。
4. 算法终止。

通俗地讲，若$Partition$中各点集已经分隔则是优解，否则合并不分隔的点集（以$\Delta_{i,j}^{P}(G)$作量度，值越大，说明不分隔的程度越"剧烈"），直到各点集两两分隔为止。其实还是$Incremental\ computability$ 的应用。

- [ ] 推演到$K-dimensional$ 情况下找$Partitioning\ Tree$ 。
- [ ] 定义两个操作：

- $M(T;\alpha,\beta)$ 向同一颗树上添加节点
- $C(T;\alpha,\beta)$ 连接两颗不同的树

  > 待填。

杂项

- [ ] 略过了很多对算法界的证明。
- [ ] 作者力主将$Structural\ Information$用于下一代搜索引擎，认为更能抓到事物本质。为此提出了$Natural\ Rank$ 。
- [ ] 以上内容多为抽象的公式，文章作者给了一个应用场景——分析淋巴癌的基因图谱。

- 弥漫大B细胞淋巴瘤的特点是基因十分复杂。传统的$Modularity\ Maximisation$ 分析其基因图谱只能发掘四种类型，而本文的方法可以发掘十几种。每种类型对应的癌细胞都有相似的存活周期、存活率，所以能针对地进行治疗。