

组会

1. PC一族算法的进展

1.1 算法选择

目前来看，之前讨论的PC和FCI效果没有明显区别。

FCI从算法流程上也是采取PC类似的，从完全图---独立性检验---定向化的过程。

1.2 代码

- Tetrad源码：和殷博交流，实现Tetrad源码的分离以及服务化，有了较大进展
- Python脚本：框架基本可用，调整空间在于独立性检验细节
- R语言：
 - 还未尝试
 - 文献《Causal Inference Using Graphical Models with the R Package pcalg》
 - 主要是配置包环境，指令式代码，难度不大

以上三种都可以为论文提供选择

2. 贝叶斯构建算法

2.1 起因

- 文献：《Learning Belief Networks from Data: An Information Theory Based Approach》
- 利用信息论，在数据集上构建贝叶斯网络

独立性检验利用信息论的互信息（mutual information）

$$I(X_i, X_j | C) = \sum_{x_i, x_j, c} P(x_i, x_j, c) \log \frac{P(x_i, x_j | c)}{P(x_i | c) P(x_j | c)}, \quad (2)$$

互信息(Mutual Information)是[信息论](#)里一种有用的信息度量，它可以看成是一个[随机](#)变量中包含的关于另一个随机变量的信息量，或者说是一个随机变量由于已知另一个随机变量而减少的不肯定性

- 算法
 - drafting：起草
 1. 起始：图G，节点V，边E为 **空集**
 2. 对每对节点计算互信息
 3. 对于具有大于某个小值ε的互信息的所有节点对，按其互信息对它们进行排序，并将这

些节点对从大到小放入列表L.

4. 获取列表L的前两对节点并从中删除它们。将相应的边添加到E

5. 重复

- thickening: 加边

- 1. 重扫描L, 调用try_to_separate_A 来判断是否添加更多边

- thinning: 减边

- 1. 对于E中的每个边, 如果两个节点之间除此边之外还有其他路径, 则暂时从E中删除此边并调用过程try_to_separate_A。如果两个节点是相关的, 则将此边缘添加回E;否则永久删除边缘。

- 2. 将try_to_separate_B替换A, 重复1步

- 3. 调用orient_edges, 定向化

try_to_separate_A, try_to_separate_B通过条件互信息检验独立性, 决定是否删除边

2.2 同类

有一系列用于从数据中获取贝叶斯网络结构的算法

- 分类:

- 基于评分的: K2算法

- 基于条件独立性测试的: TPDA (本文), SGS (PC的源头)

- 混合: MCMC

2.3 作用

- 为论文提供新的对比角度

- 完全替代PC系列