**引言**

GJB2基因突变是新生儿听力障碍的主要原因之一。每个人都携带两个不同的基因，因此每个人都有可能拥有听力障碍版本GJB2的0、1或2个基因。然而，除非一个人接受基因测试，否则要知道一个人有多少个变异的GJB2副本并不容易。这是某种“隐藏状态”：具有我们可以观察到（听力障碍）但不一定直接知道的影响的信息。毕竟，有些人可能有1或2个变异的GJB2基因，但没有表现出听力障碍，而其他人可能没有变异的GJB2的基因，但仍然表现出听力障碍。

每个孩子都从每个父母那里继承一份GJB2基因。如果父母有两个突变基因，那么他们将把突变基因传递给孩子；如果父母没有突变基因，那么他们不会将突变基因传递给孩子；如果父母有一个突变基因，那么该基因将以0.5的概率传递给孩子。然而，在基因传承后，它有一定的可能性发生额外的突变：从导致听力障碍的基因版本改为没有，反之亦然。

我们可以尝试通过形成所有相关变量的贝叶斯网络来建模所有这些关系。

**正文**

在这个网络中，每个人都有一个基因随机变量，表示一个人拥有特定基因的数量，可以是基因的0、1或2个遗传因子。每个人都有一个特征随机变量，该变量可以是或否，这取决于该人是否表达了由基因引起的特征（听力障碍）。

构建的模型如下，在我们使用的模型中，每个人的基因随机变量都有一个基于其父母基因变量的条件概率。如果这个人是家谱的根（没有父信息），那么基因随机变量被赋予拥有0、1或2个基因的无条件概率。

每个人的特征随机变量根据他们拥有的基因数量有条件概率-拥有0个基因的人拥有该特征的概率非常低，而按枚举推断给定一个有多个成员的家谱，其中一些是父母，以及某些家庭成员是否具有遗传特征的知识，如何计算每个家庭成员拥有0、1或2个遗传因子的条件概率？

假设的参数如下：X是查询变量

e是证据（我们持有的关于家庭的知识）

P(X | e)是给定知识e的变量X的概率分布

α是一个归一化因子（查询变量X的概率之和必须为1）

y范围在所有隐藏变量的所有值上

从本质上讲，要找到每个人拥有的基因数量的概率分布，以及它们是否具有该特征（如果尚未知道）：那就必须：

为家庭生成各种可能的基因和性状组合。

丢弃任何与已知家庭证据不一致的组合。

计算每个组合的联合概率。

将每个组合的联合概率添加到每个人拥有0、1或2个基因并拥有或不具有该性状的概率中。

规范每个人的基因概率分布和性状概率分布。

需要输入四个变量-人、one\_gene、two\_genes和 have\_trait，输出的是联合概率密度

人是一本人字典。键代表名称，值是包含母键和父键的字典。您可以假设父母都是空白的（数据集中没有父母信息），或者父母都会在人物词典中引用其他人。

one\_gene是所有人的集合，我们希望为他们计算他们拥有一个基因副本的概率。

two\_genes是所有人的集合，我们希望为他们计算他们拥有两个基因副本的概率。

have\_trait是所有我们想计算他们的人有特征概率的集合。

对于任何不处于one\_gene或two\_genes的人，我们想计算他们没有遗传因子的概率；对于任何不处于 have\_trait的人，我们想计算他们没有特征的概率。

**学到的东西**

因为之前做过的大创项目需要用到python，所以熟悉一些常用的术语，比如字典、元组、循环结构等。其对我的最初的印象就是，相较于我学习过的c语言编程，它更加的简洁。所有的变量都不需要像c语言编程那样需要提前去定义，这样给了编程者很大的自由空间与方便。如x=2，即可同时完成变量的定义与赋值。对于简化程序的代码，起到了许多的作用。而且，在字符串上的处理，python相对于c语言也是极大的便利。在c语言中，只能用字符类的数组对字符串进行相应的操作，步骤也是相对于比较繁琐的，而在python中，当我们需要创建一个字符串的时候，只需要在创建字符串的时候用“s=”就可以了。而python不仅如此，它的库也很多，正因为它强大的库，让编程变得不再艰难。我们只需要调用库中的函数，而对于函数的具体实现，也没有特殊的需求。而通过思考完这次的信网应用层大作业后，我学会了如何将一个具体的问题抽象为数学问题，并通过贝叶斯模型对其预测，虽然只完成了其中的一小部分，但收获也是极大的，我学会了条件概率在代码上具体的体现，也学会了通过多个变量控制输出的联合概率，使我受益匪浅。

github链接：

<https://github.com/12444444444/jiyin/blob/80fcdbd2dc5374712536371c7461c2d479e99262/%E9%81%97%E4%BC%A0%E8%B4%9D%E5%8F%B6%E6%96%AF%E7%AE%97%E6%B3%95>

视频因为要审核来不及了所以就压缩了