解开玻尔兹曼机的封印会发生什么?

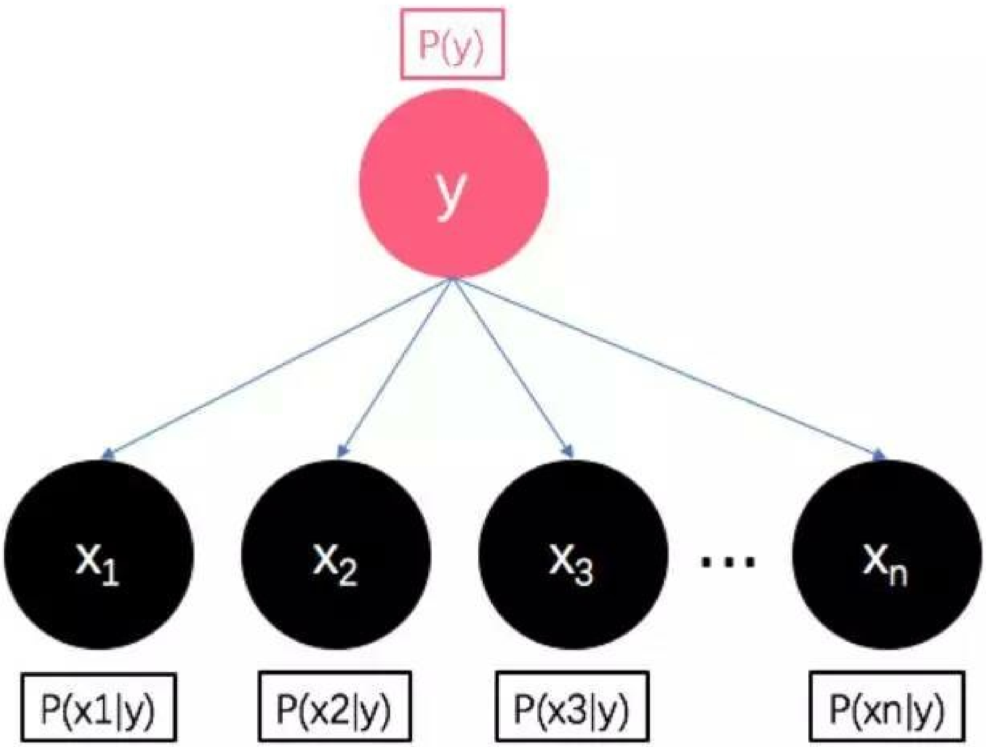
原创 夕小瑶 夕小瑶的卖萌屋 2017-06-17

在上一篇文章中，小夕讲述了逻辑回归为了抗衡贝叶斯网，也开始了自己的进化。然而令我们没有想到的是，逻辑回归最终竟然进化成了一个生成式模型——受限玻尔兹曼机（RBM），也就是变成了敌方（生成式模型）的武器。

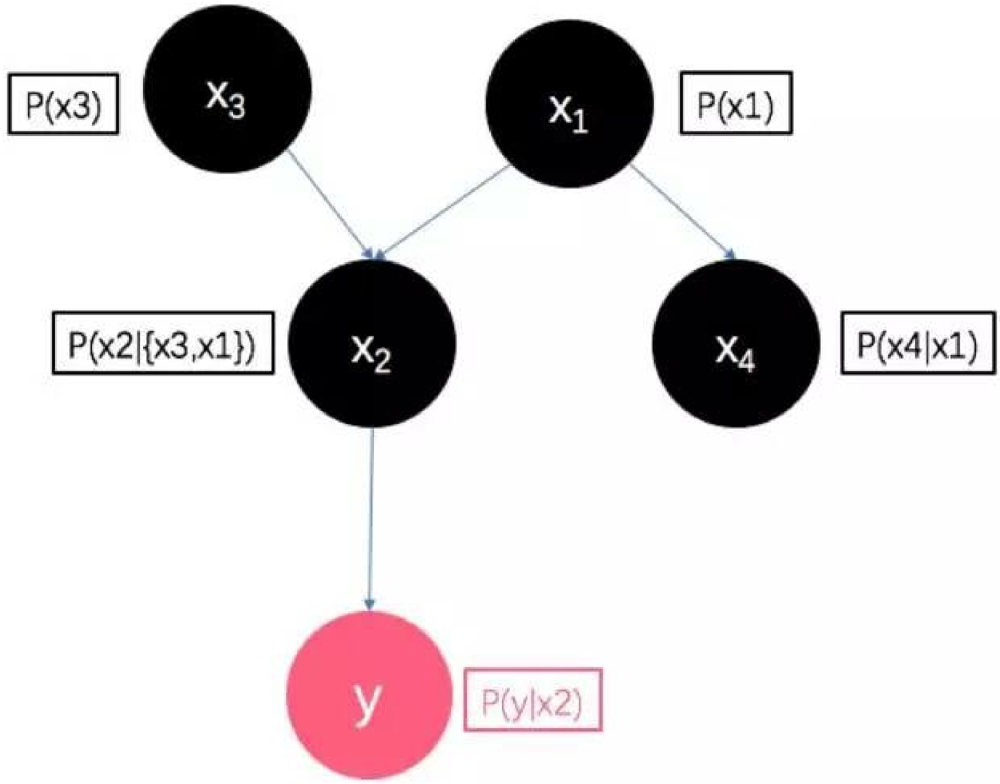
意外得到RBM的朴素贝叶斯万分惊喜，并且燃起了将它自己做的贝叶斯网与敌方送的RBM融合的冲动！

那么朴素贝叶斯的疯狂想法能不能实现呢？

还是按照惯例，先叙述背景姿势。在《朴素贝叶斯到贝叶斯网》中，小夕为朴素贝叶斯画了一幅肖像：



朴素贝叶斯看到自己的肖像后，深感自己的朴（弱）素（鸡），于是进行了进化——抛弃自己的条件独立性假设，建模特征向量X内部各个维度的条件依赖关系。于是，朴素贝叶斯进化出了下图的贝叶斯网：

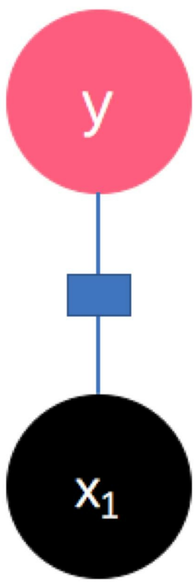


在上一篇文章《逻辑回归到受限玻尔兹曼机》中，RBM相比逻辑回归有了非常大的进步，可以更加合理的计算每个样本与每个类别的“亲密度”，也就是与之关联的概率图模型中能量函数E(v1,v2)的大小。

RBM看起来这么复杂，那么灵魂画师夕小瑶能不能像对贝叶斯一样，给RBM画一个肖像呢？

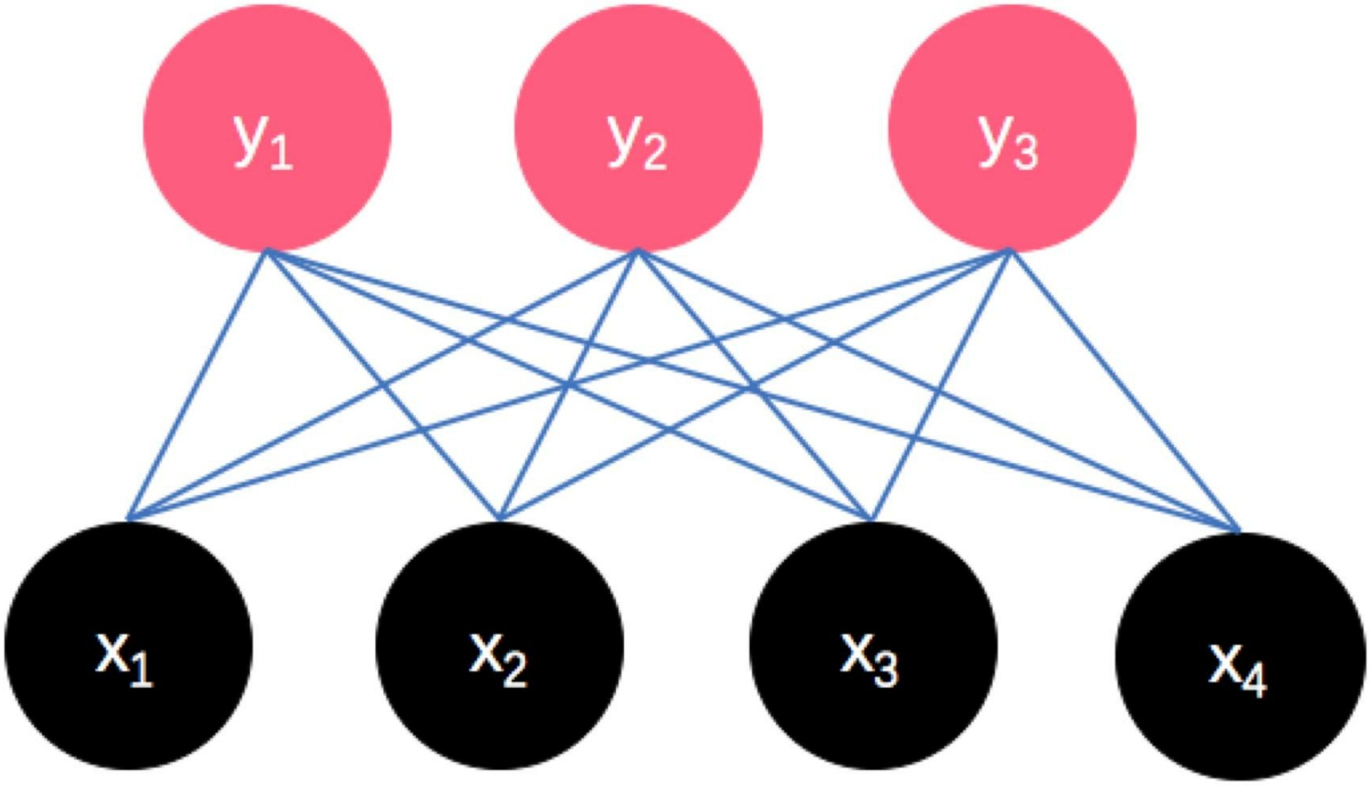
RBM的肖像?

当然可以啦~从上一篇文章对能量函数E(v1,v2)的定义来看，这个函数计算向量v1和v2的“亲密度”时是没有方向的，也就是说E(v1,v2)一定等于E(v2,v1)的。所以对于图中的任意两个点来说，他们之间的边是没有方向的（注意区分朴素贝叶斯和贝叶斯网中的有向边哦）。所以对于RBM中有连接的两个点：



边是无向的，并且用一个蓝色小方块表示用能量函数连接。

按照上一篇文章的表述，RBM中的一个参数——矩阵W连接了特征向量X中的每个维度与类别向量Y的每个维度，因此RBM应该是下面这个样子的：

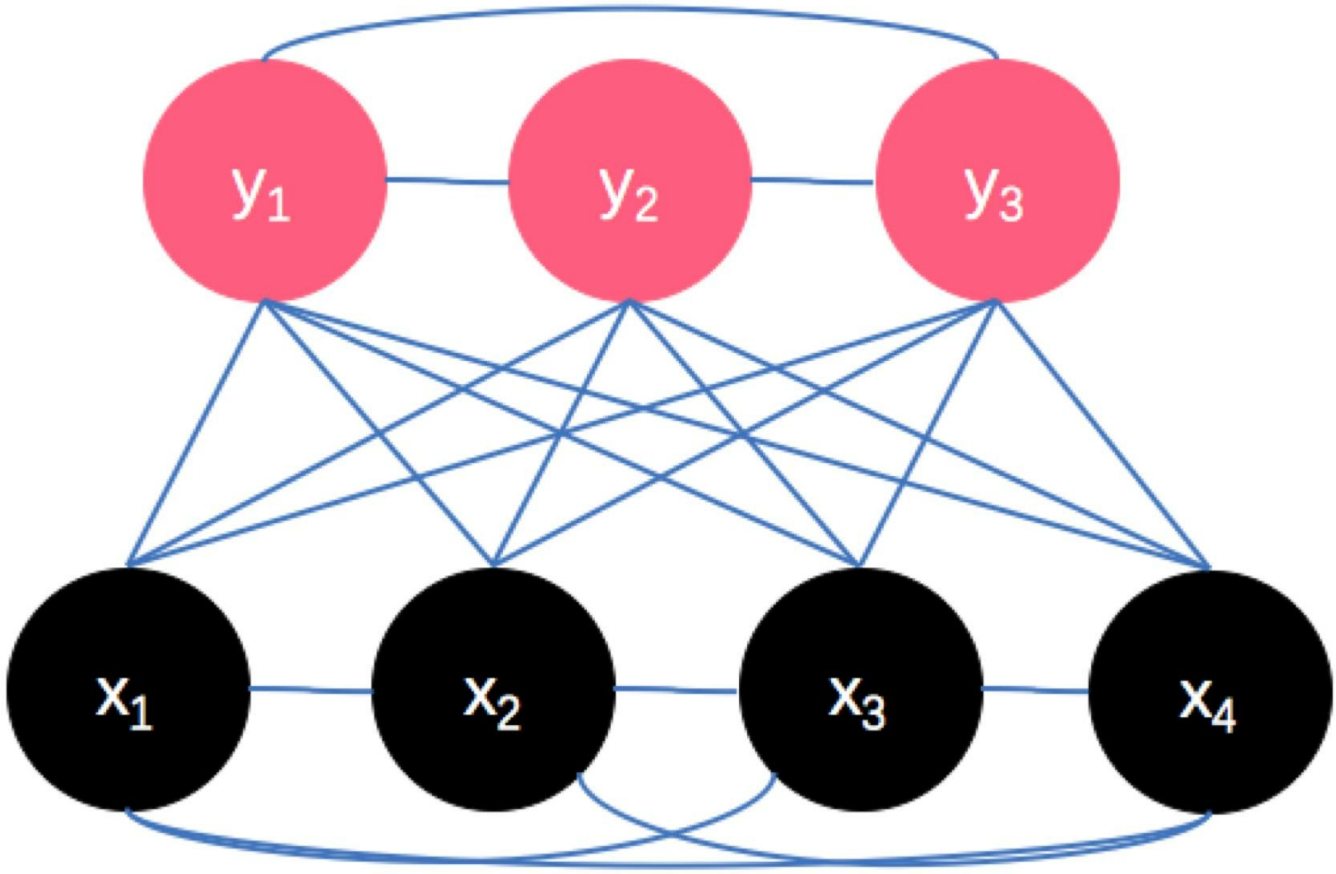


（小方块太多啦，省略掉了~但是每条边依然代表着这两个节点用能量函数(中的参数)连接）

诶？有没有觉得。。。还不够乱！（好丧心病狂的想法

回想一下，从朴素贝叶斯到贝叶斯网就是经历了将X内部的各个维度（各个随机变量）之间的关系也建模了！而RBM则是跟朴素贝叶斯一样的，对X内部（更一般化的说还包括Y内部）的随机变量是有独立性假设的。而在《朴素贝叶斯到贝叶斯网》中已经详细叙述了这个独立性假设在很多情况下是非常致命的！

所以，才不要什么人为的假设呢~解开玻尔兹曼机身上的枷锁吧~让X内部以及Y内部的随机变量也可以随意的交流（即，使模型具备描述X和Y内部各随机变量之间条件依赖性关系的能力，只不过这里直接描述了双向的条件依赖性关系）。所以从图中看是这样子的：



玻尔兹曼机：喵喵喵~最喜欢自由的感觉啦~

那么在数学上怎么描述呢?当然可以直接照搬RBM中的做法啦。

RBM的假设函数中的能量函数：

$$E(v1,v2) = -(b^T v1 + c^T v2 + v1^T \cdot W \cdot v2)$$

用一个矩阵W连接了X所有维度与Y的所有维度。因此要连接X内部所有维度和Y内部所有维度的话，只需要：

$$E(v1,v2) = -(v1^T \cdot R \cdot v1 + v2^T \cdot S \cdot v2 + b^T v1 + c^T v2 + v1^T \cdot W \cdot v2)$$

相信聪明的你很轻松就看懂啦，这里用两个与W同理的**矩阵R、S**来分别连接v1内部的各维度以及v2内部的各维度。

所以玻尔兹曼机（BM）的假设函数跟RBM的形式一样，都是

$$f = P(x=i, y=j) = \frac{1}{Z} exp(-E(x=i, y=j))$$

其中配分函数Z：

$$Z = \sum_{m=1}^{M} \sum_{k=1}^{K} e^{-E(x=m,y=k)}$$

只是将里面的能量函数换成上面的更丧心病狂的形式了而已~

而根据《一般化机器学习》，我们已经理解了玻尔兹曼机（BM）的假设函数，那么还需要探索如何训练这个自由而强大的模型。而如何训练，即寻找或设计一个合适的损失函数，然后选择合适的最优化算法来最小化损失函数从而得到model。

然而，自由而随意所带来的代价就是：
非！常！难！以！训！练！

设想一下，假如损失函数我们就用最通用的似然函数（诶诶？是不是还缺一篇写损失函数的文章？没事没事，还好在《EM算法》中有讲似然函数），那么任何主流的最优化算法都会计算量爆炸（想象一下对玻尔兹曼机的假设函数求导，**尤其是对配分函数这个恐怖大分母**！），哪怕是最简单的梯度下降法（当然这里**最大化**似然函数就是梯度上升啦），都会使得工程上训练BM很不现实。

那么怎么办呢？

一个主流的解决办法是使用一种改良的梯度上升法——**MCMC算法**来最大化似然函数。

这个算法的由来就是一个纯数学过程了，需要一篇很长很崩溃的堆砌公式的文章才能讲清楚，所以。。。详情可以参考《Deep Learning》(中文版链接https://github.com/exacity/deeplearningbook-chinese)的第18章啦(￣▽￣)，数学不太好的慎入哦（小夕诚实的讲，小夕也没有很清晰的理解透...所以现在讲的肯定不如书上好~）

诶诶？爱思考的小夕又有疑问惹：

首先，这就是玻尔兹曼机的全部潜力了吗？怎么看着能量函数有点眼熟呢。。。有点像，神经张量网络(NTN)？诶？会不会跟神经网络有关系呢？会不会。。。会不会跟深度学习碰撞出火花？叫做**深度玻尔兹曼机**？

而且，既然贝叶斯网络是用有方向的边去描述的，玻尔兹曼机是用无方向的边去描述的，而且这两者看起来都很自由~那这两个。。。哪个更好呢？

就让小夕站在概率图的高度来描述一场新的战争吧~


蟹蟹你o(≧v≦)o
小红包
微信支付
Transfer to 夕小瑶