Data Processing Inequality

http://freemind.pluskid.org/machine-learning/data-processing-inequality

posted on Free Mind on June 12, 2013
generated with pandoc on December 3, 2015
category: Machine Learning

tags: Information Theory, Statistics

上个学期有一门课的 project 尝试做了一个 feature extraction 的算法，在特征 A 的基础上做提取得到特征 B，然后进行分类实验。因为数据点是 million 级别的，所以跑 SVM 之类的都只用了线性版本，结果是 B 特征的效果比原始的 A 要好挺多，但是当我们尝试使用 Deep Neural Network (DNN) 这个利器的时候（一方面也是因为 A+DNN 算是目前该问题的 state-of-the-art），B 却怎么都比不过原始的 A。于是我突然想到了另一门课上学过的一个叫做 Data Processing Inequality (DPI) 的东西。

DPI 顾名思义就是描述对数据进行处理的一个不等式，简单来说，就是任何数据处理工序都只会造成信息丢失。所以说，我们在 A 的基础上进行处理得到 B，实际上这是一个信息丢失的过程，于是最后结果变差了好像也可以理解了？这好像很矛盾，因为这不是把各种数据预处理、降噪、降维、特征提取等等全都否定掉吗？其实并没有矛盾的地方，之所以矛盾是我们把抽象的数学上的不等式用具体的概念非常不严格地进行了带入解释，这其实是许多看似荒谬的悖论的来源。简而言之，为了能理解 DPI 到底代表了什么，我们还得从具体的数学定义出发。

首先，所谓 “信息丢失” 中的 “信息”，当然就是指信息论中的熵了，由于连续型随机变量的情况下熵的定义变得复杂了，我们这里只讨论离散的情况，一个随机变量 $\mathsf{x} \in \mathcal{X}$ 的熵定义为：

$$H(\mathsf{x}) = -\sum_{x' \in \mathcal{X}} P_{\mathsf{x}}(x') \log P_{\mathsf{x}}(x')$$

关于熵的定义的意义有许多不同的解释，比如描述所需的 bit 数之类的，不过我们这里用一个和 Inference 相关的解释。这需要再引入这里的 Inference 问题：给定一个随机变量 x，求一个分布 q 用于描述 x。结果的好坏用一个 loss 函数 $C(\mathsf{x}, q)$ 来衡量。比如一个典型的 loss function 就是 log loss:

$$C(\mathsf{x}, q) = -\log q(\mathsf{x})$$

而求解分布 q 的过程则变成如下的优化问题：

$$\min_q \mathbb{E}_{\mathsf{x}}\left[C(\mathsf{x}, q)\right]$$

当然 loss function 可以随问题的不同而自己定义，但是一般情况下我们会偏向具有一些良好性质的 loss function。比如一个 loss function

被称为是 proper 的，如果该 cost 对应的最优解就是 x 的真实分布本身[1]；另外，一个 loss function 被称为是 local 的，如果它只 care x 在 q 下的取值而对 q 的其他可能具有的结构等特征不感兴趣，换句话说：$C(\mathsf{x}, q) = C(\mathsf{x}, q(\mathsf{x}))$。

然后 log loss 的特殊性在于（当 $|\mathcal{X}| \geq 3$ 的情况下）它是唯一的 smooth, proper and local loss。于是使用 log loss 的正当性就得到了充分地 justification。然后我们可以再回到熵，关于我们刚才所说的 Inference 问题，由于最优解就是 $q = P_{\mathsf{x}}$，所以直接将 P_{x} 带入到 log loss 里，再加上外围的 expectation 的话，立刻就可以看到：其实熵就是在最优解的情况下该问题的 loss。换句话说，也就是在使用 log loss 的时候该问题所能达到的最优 loss，可以看成是代表了该问题的 intrinsic complexity，或者是 uncertainty。

接下来要引入数据，如果我们观察到了 $\mathsf{y} = y$，原来的问题的最优 loss 则变成了：

$$\begin{aligned}\min_q \mathbb{E}[C(\mathsf{x}, q)|\mathsf{y} = y] &= \mathbb{E}[C(\mathsf{x}, P_{\mathsf{x}|\mathsf{y}})|\mathsf{y} = y] \\ &= -\sum_x P_{\mathsf{x}|\mathsf{y}}(x|y) \log P_{\mathsf{x}|\mathsf{y}}(x|y) \\ &\triangleq H(\mathsf{x}|\mathsf{y} = y)\end{aligned}$$

为了考虑所有可能出现的数据，我们在求期望的过程中实际上也是要对随机变量 y 求期望的，所以得到如下的量：

$$\mathbb{E}[H(\mathsf{x}|\mathsf{y} = y)] \triangleq H(\mathsf{x}|\mathsf{y})$$

叫做条件熵，同熵类似，这是在观察到了数据 y 的情况下对 x 进行描述所能得到的最优的 loss，于是可以理解为在观察到了 y 的情况下 x 所蕴含的不确定性。

一个很自然的问题就是：通过观察到数据 y，我们将 x 的不确定性降低了多少？也就是如下的量：

$$H(\mathsf{x}) - H(\mathsf{x}|\mathsf{y}) \triangleq I(\mathsf{x}; \mathsf{y})$$

被称作 Mutual Information，直观地来讲就是 y 里所包含的有多少关于 x 的信息。不过互信息实际上是对称的，所以 x 和 y 的角色反过来也是对的。

到这里，我们已经做了许多准备工作了，接下来可以直接陈述 DPI 的一个数学描述：对于任意函数 g，我们有

$$I(\mathsf{x}; \mathsf{y}) \geq I(\mathsf{x}; g(\mathsf{y}))$$

[1] 这里看起来有些奇怪的是：如果我们都知道了 x 的分布，那么干嘛还要去费力求一个 q？特别是最优的 q 就是 x 的分布本身的情况下。理由在于一般情况下我们会对 q 可以取的情况加以限制，从而得到在某些限制条件下（比如 q 被限制为高斯分布）关于 x 的最好的描述。

这也就是信息减少的具体数学表达了。简单证明如下：

$$\begin{aligned} I(\mathsf{x}; g(\mathsf{y})) &= I(\mathsf{x}; \mathsf{y}, g(\mathsf{y})) - I(\mathsf{x}; \mathsf{y} | g(\mathsf{y})) \\ &= I(\mathsf{x}; \mathsf{y}) + I(\mathsf{x}; g(\mathsf{y}) | \mathsf{y}) - I(\mathsf{x}; \mathsf{y} | g(\mathsf{y})) \\ &= I(\mathsf{x}; \mathsf{y}) - I(\mathsf{x}; \mathsf{y} | g(\mathsf{y})) \\ &\leq I(\mathsf{x}; \mathsf{y}) \end{aligned}$$

另外，如果不等号的等式成立的话，这个函数 $g(\cdot)$ 通常被称为 sufficient statistics。简而言之就是说，所以要利用 y 来对 x 做的 Inference，我都只需要知道 $g(\mathsf{y})$，而不是完整的 y 就能做到一样好了。

当然，所谓 “信息量” 的说法完全是从信息论的角度来说的，在实际中最有信息量的东西并不一定是最有用的，因为把有用信息找出来可能反而是问题的瓶颈。比如，仅从信息量的角度来说的话，我可以在原始数据中加入无比多的不相关的干扰数据，虽然所包含的有用信息没有变化，但是为后续的处理过程却增加了很多麻烦。

同样的道理，各种降维降噪数据变换之类的数据预处理步骤，虽然从信息论的角度上来说并不会增加 “信息量”，甚至还会丢失有用信息，但是它们可能会让后续的信息提取步骤变得更加容易。而 DPI 本身则是站在 “后续的信息提取工具无比强大” 的立场来说的，这样一来，如果承认 DNN 擅长于抽丝剥茧即使是高度非线性的关系也能从数据中很好地提取出来的话，为什么线性的方法会结果比较好而 DNN 却结果比不过也可以解释得过去了。