Lecture 12　Nonlinear Transformation

整理者：LobbyBoy* 2020年2月26日

1. Quadratic Hypotheses

目前为止，我们使用的hypothesis都是线性的(linear)。线性hypothesis在空间中对应的是line-like的boundary，即hyperplane。线性假说有优点也有缺点：优点是简单，简单意味着复杂度小，复杂度小就会使得泛化上界更紧，VC bound的保证作用更强，即有很大概率$E_{in} = E_{out}$；缺点也是简单，简单意味着很可能无法找到一个E_{in}很低的hypothesis。例如，对于线性不可分的训练数据集，线性hypothesis无论如何也无法达到$E_{in} = 0$的程度。对于下面的训练数据集：

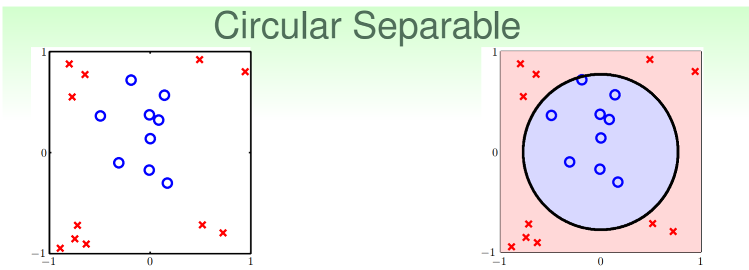

图 1: Circular separable

显然不是线性可分的，但可以被一个圆完美分开，我们则暂时称其为Circular separable的训练数据集。假设上图中将其完美分开的圆的方程为：

$$x_1^2 + x_2^2 = r^2$$

并将输入空间称为$\mathcal{X}$，则该$E_{in} = 0$的hypothesis为：

$$h(\mathbf{x}) = sign\Big(-x_1^2 - x_2^2 + 0.6\Big), \quad (x_1, x_2) \in \mathcal{X}$$

令$\mathbf{z} = (z_0, z_1, z_2) = (1, x_1^2, x_2^2)$，$\tilde{\mathbf{w}} = (-1, -1, +0.6)$，则上面的hypothesis可以写成：

*本笔记根据台湾大学林轩田教授于线上教育平台Coursera开设的“机器学习基石”课程整理而成(课程内容见：https://www.coursera.org/learn/ntumlone-mathematicalfoundations/home/welcome)。笔记内的大多数图片来自于林老师的课程slides。感谢林老师能够将如此精彩的课程通过线上平台同所有人分享，thanks!

$$\tilde{h}(\mathbf{z}) = sign\left(\tilde{\mathbf{w}}^T\mathbf{z}\right), \quad (z_1, z_2) \in \mathcal{Z}$$

其中，$\mathcal{Z}$即为坐标系第一象限，因为$z_1 = x_1^2 \geq 0$，$z_2 = x_2^2 \geq 0$。观察$\tilde{h}(\mathbf{z})$的形式，可以看出它就是$\mathcal{Z}$空间中的直线。因此，我们通过坐标变换的方式，将原空间映射到了一个新空间——在原空间中训练数据非线性可分，但在新空间中训练数据线性可分。这是由于，原空间中的圆，透过这层坐标变换，投到新空间中，变成了一条直线：

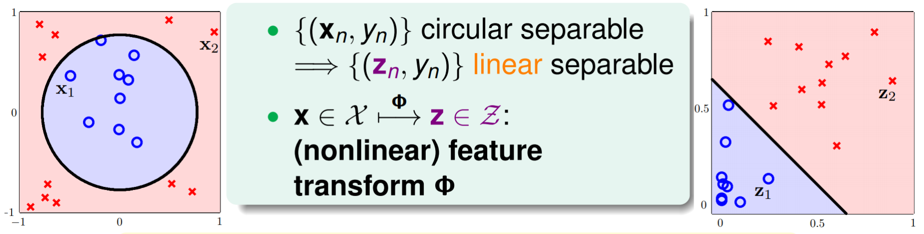

图 2: Feature transform

我们将上述的坐标变换，称为“特征转换”(feature transform)，记作Φ。在上面的问题中，有：$\mathbf{z} = \Phi(\mathbf{x})$。并且我们知道了，$\mathcal{X}$空间中的圆(圆心为原点)，对应为$\mathcal{Z}$空间中的线。那么$\mathcal{Z}$空间中的线，是否对应$\mathcal{X}$中的圆？不完全是：

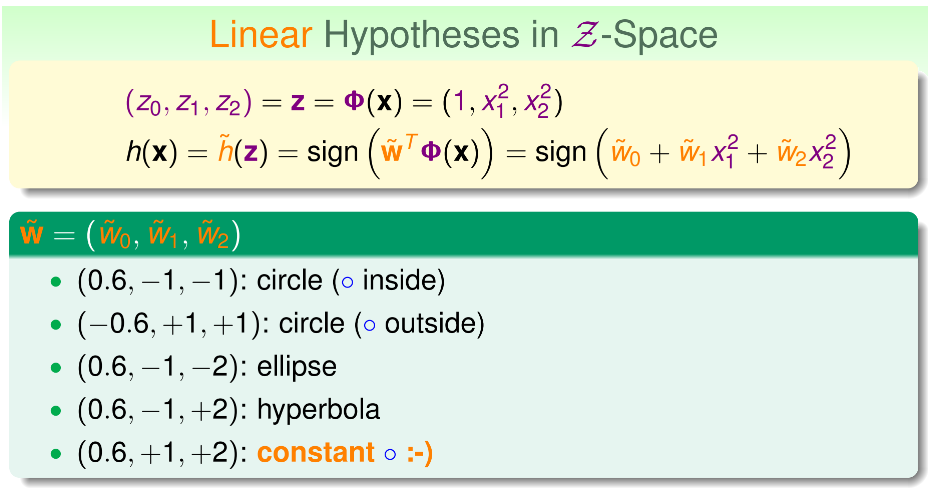

图 3: Feature transform

上面的简单数学变形说明，$\mathcal{Z}$空间中的线可以对应$\mathcal{X}$空间中的圆、椭圆、双曲线等。但需要注意到，$\mathcal{Z}$空间中的线无法映射到$\mathcal{X}$空间中圆心不在原点的二次曲线——因为没有交叉型二次项。因此，我们需要构造一个更大的、包含交叉二次项的$\mathcal{Z}$空间：

$$\mathbf{z} = \Phi_2(\mathbf{x}) = (1, x_1, x_2, x_1^2, x_1x_2, x_2^2)$$

在这样的$\mathcal{Z}$空间中，线等价于$\mathcal{X}$空间中的二次曲线。因此，我们可以通过feature transform将我们原来在$\mathcal{X}$空间中的线性假说集合拓展为$\mathcal{X}$空间中的二次曲线假说集合(当然该假说集合也包含所有的线性假说——当二次项的系数全部为0时)：

$$\mathcal{H}_{\Phi_2} = \left\{ h(\mathbf{x}) : h(\mathbf{x}) = \tilde{h}(\Phi_2(\mathbf{x}))\ for\ some\ linear\ \tilde{h}\ in\ \mathcal{Z} \right\}$$

2. Nonlinear Transform

在上一讲中，我们通过feature transform，将原特征映射到另一个空间中，并在新的空间中寻找E_{in}较低的超平面。而新空间中的超平面往往对应的是原空间中的非线性边界，如二次曲线等等。因此，通过feature transform+linear model，我们可以拟合出原空间中除超平面外各种各样复杂的非线性边界。

使用这种nonlinear feature transform进行机器学习的步骤如下：①选择合适的Φ，将原数据进行非线性转化：从$\{(\mathbf{x}_n, y_n)\}$到$\{(\mathbf{z}_n = \Phi(\mathbf{x}_n), y_n)\}$；②在新的$\mathcal{Z}$空间中使用某个合适的线性分类算法(linear classification algorithm)训练出一个E_{in}很低的perceptron，$\tilde{\mathbf{w}}$；③回传结果：$g(\mathbf{x}) = sign(\tilde{\mathbf{w}}^T\Phi(\mathbf{x}))$。如下图：

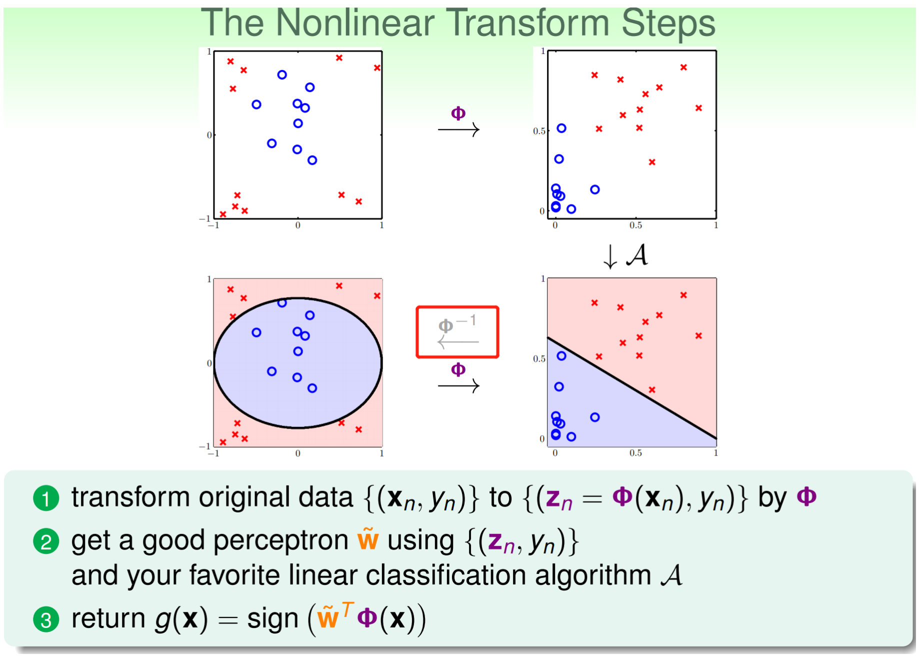

图 4: The nonlinear transform steps

上图中值得注意的地方是，当我们在$\mathcal{Z}$空间中训练出了一个perceptron后，我们并不需要将其映射回原来的$\mathcal{X}$空间，实际上这种映射也可能并不存在(一对多，没有办法保证一定是一对一)。因此，我们可以就保留$\mathcal{Z}$空间中的那个perceptron，然后在测试时将测试数据的输入先进行feature transform，转换到$\mathcal{Z}$空间中，再拿其中的那个perceptron进行预测，得到结果。

3. Price of Nonlinear Transform

Nonlinear Transform虽然十分强大，但有两大缺点。

第一，computation/storage price，即需要大量的计算与储存成本。例如，对于Q阶多项式的转换，使得维度数从原来的$d+1$，增长到约$O(Q^d)$。这使得训练中的各种计算都很艰难，且需要很多空间储存这么多维度的训练结果。

第二，complexity price，即模型的复杂度大大提高，VC dimension增加：$d_{VC}(\mathcal{H}_{\Phi_Q}) \approx \tilde{d}+1 = O(Q^d)$。所以，Q很大的时候，VC dimension也很大，不利于泛化。

因此，在进行多项式转换的时，要慎重选择阶数Q，因为对Q的选择实际上就是对“拟合”和“泛化”的一个trade-off：Q大，我们可以把E_{in}做的很小，但我们不能保证E_{in}很接近E_{out}；Q小，我们可以保证E_{in}很接近E_{out}，但不能把E_{in}做的如上面那样小。

所以到底如何选择Q？一种思路是，画出训练数据的图像，观察，看哪个Q适合。这种做法是错误的。首先，高维度数据无法可视化，可视化最高用在三维数据上。其次，只要你看过数据，那么你的大脑就会潜意识地进行human learning，此时你选择的transform方式就已经受到了主观意识的影响，包含了brain’s model complexity。因此，即使你认为你选择的transform的复杂度很小，但这只是一部分复杂度而已，你并没有把你帮机器进行选择时产生的复杂度算进去：

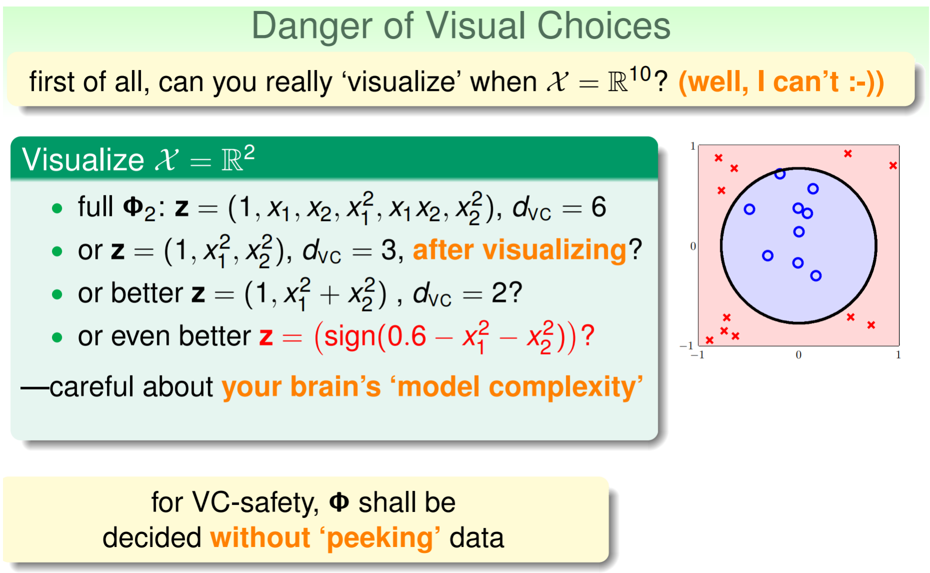

图 5: The nonlinear transform steps

如何科学地选Q？Validation。

4. Structured Hypothesis Sets

多项式转换(Polynomial Transform)的Hypothesis Set呈现一种结构性关系：高次项转换的$\mathcal{H}$包含低次项转换的$\mathcal{H}$(新增项系数置0即可得到低次的所有hypotheses)：

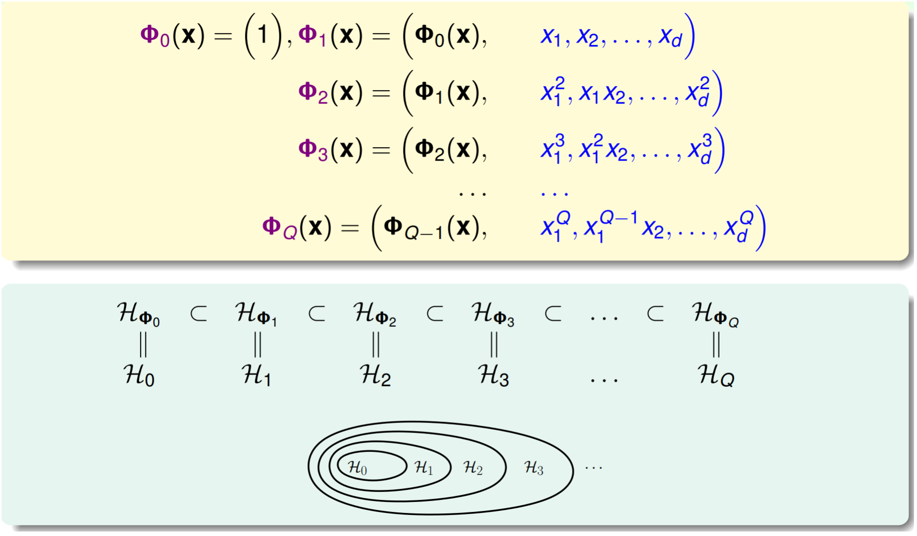

图 6: Nested

我们将使用假说集合$\mathcal{H}_i$训练出的hypothesis记作g_i，即：

$$g_i = \underset{h \in \mathcal{H}_i}{argmax}\ E_{in}(h)$$

我们知道，随着假说集合越来越复杂(i变大)，$E_{in}(g_i)$会越来越小，但是假说集合的VC dimension会越来越大，即$d_{VC}(\mathcal{H}_i)$会逐渐变大。因此，$E_{out}(g_i)$会先降低，后增大，如下图所示：

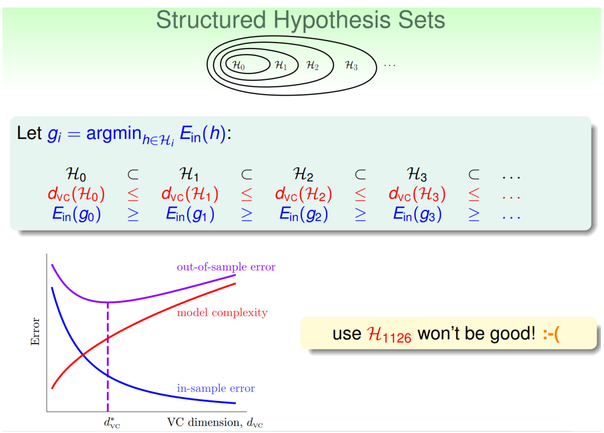

图 7: Nested

因此，一开始直接使用复杂度高的假说集合并不是一个明智的选择。合适的做法是，

先用最简单的假说集合，即线性模型。如果线性模式的结果已经很好了，那么就停止；如果E_{in}较大，则增加一点复杂度重新训练。