机器学习算法系列（12）：SVM（3）—非线性支持向量机

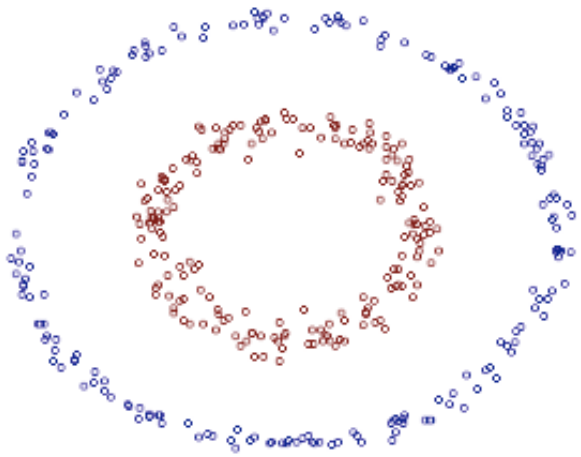
当输入空间为欧式空间或离散集合、特征空间为希尔伯特空间时，核函数表示将输入从输入空间映射到特征空间得到的特征向量之间的内积。通过使用核函数可以学习非线性支持向量机，等价于隐式地在高维的特征空间中学习线性支持向量机。此为核方法，是比支持向量机更为一般的机器学习方法。

三、非线性支持向量机与核函数

3.1 核技巧

前面我们介绍了线性情况下的支持向量机，他通过寻找一个现行的超平面来达到对数据线性分类的目的。不过，由于是线性方法，所以对非线性的数据就没有办法处理了。

例如图中的两类数据，分别分布为两个圆圈的形状，不论是任何高级的分类器，只要他是线性的，就没有办法处理，SVM也不行。因为这样的数据本身就是线性不可分的。



此数据集为两个半径不同的圆圈加上了少量的噪音得到，所以一个理想的分界应该是一个圆圈而不是一条直线。如果用X_1 和 X_2来表示这个二维平面的两个坐标的话，则此方程可以写作

$$a_1X_1 + a_2X_1^2 + a_3X_2 + a_4X_2^2 + a_5X_1X_2 + a_6 = 0$$

注意上面的形式，如果我们构造另外一个无谓的空间，其中五个坐标的值分别为
$Z_1 = X_1, Z_2 = X_1^2, Z_3 = X_2, Z_4 = X_2^2, Z_5 = X_1 \cdot X_2$

那么显然，上面的方程在新的坐标系下可以写作：

$$\sum_{i=1}^{5} a_i Z_i + a_6 = 0$$

如果我们做一个映射$\phi: R^2 \to R^5$，将X按照上面的规则映射为Z那么在新的空间中原来的数据将变成线性可分的，从而使用之前我们推倒的线性分类算法就可以进行处理了。这正是核方法处理非线性问题的基本思想。

总结一下，用线性分类方法求解非线性分类问题分为两步：首先使用一个变换将原空间的数据映射到新空间；然后在新空间里用线性分类学习方法从训练数据中学习分类模型。核技巧就属于这样的方法。

现在回到SVM的情形，假设原始的数据是非线性的，我们通过一个映射$\phi(\cdot)$将其映射到一个高维空间中，数据变得线性可分了，这个时候，我们就可以使用原来的推导来进行计算，只是所有的推导现在是在新的空间，而不是原始空间中进行。当然，推导过程也并不是可以简单地直接类比的，例如，原本我们要求超平面的法向量w，但是如果映射之后得到的新空间的维度是无穷维的（确实会出现这样的情况，比如后面会提到的高斯核函数），要表示一个无穷维的向量描述起来就比较麻烦。

我们似乎可以这样做，拿到非线性数据，就找一个映射$\phi（\cdot）$，然后一股脑把原来的数据映射到新空间，再做线性SVM即可。但是在之前对一个二维空间做映射，选择的新空间是原始空间的所有一阶和二阶的组合，得到了五个维度；但如果原始空间是三维，我们就会得到19维的新空间，这个数目是呈爆炸性增长的，这给映射的计算带来了很大困难，而且如果遇到无穷维的情况，就根本无从计算了，所以就需要核函数出马了。

核技巧的想法是，再学习与预测中只定义核函数$K(x, z)$，而不显式地定义映射函数$\phi（\cdot）$。不像之前是映射到高维空间中，然后再根据内积公式进行计算，现在我们直接在原来的低维空间中进行计算，而不需要显式的写出映射后的结果。通常，直接计算$K(x, z)$比较容易，而通过$\phi(x)$和$\phi(z)$计算$K(x, z)$并不容易。

最理想的情况下，我们希望知道数据的具体形状和分布，从而得到一个刚好可以将数据映射成线性可分的$\phi（\cdot）$，然后通过这个$\phi（\cdot）$得到对应的$K(\cdot, \cdot)$进行内积计算。然而，第二步通常是非常困难甚至完全没法做的。不过，由于第一步也是几乎无法做到的，因为对于任意的数据分析其形状找到合适的映射本身就不是什么容易的事情，所以，人们通常是“胡乱”选择一个核函数即可——我们直到她对应了某个映射，虽然我们不知道这个映射具体是什么，由于我们的计算只需要核函数即可，所以我们也并不关心也没有必要求出所对应的映射的具体形式。

我们注意到在线性支持向量机的对偶问题中，无论是目标函数还是决策函数（分离超平面）都只涉及输入实例与实例之间的内积。在对偶问题的目标函数中的内积$x_i \cdot x_j$可以用核函数$K(x_i \cdot x_j) = \phi\left(x_i\right) \cdot \phi\left(x_j\right)$来代替，此时对偶问题的目标函数成为：

$$\max_{a} \quad \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j y_i y_j K\langle x_i, x_j \rangle$$

同样，分类决策函数中的内积也可以用核函数代替，而分类决策函数式成为

$$sign\left(\sum_{i=1}^{N} a_i^* y_i K\langle x_i, x \rangle + b \right)$$

在核函数$K(x, z)$给定的条件下，可以利用解线性分类问题的方法求解非线性分类问题的支持向量机。学习是隐性地在特征空间进行的，不需要显式地定义特征空间和映射函数。这样的技巧称为核技巧，它是巧妙地利用线性分类学习方法与核函数解决非线性问题的技术。在实际应用中，往往依赖领域知识直接选择核函数，核函数选择的有效性需要通过实验验证。

3.2 常用核函数

通常人们会从一些常用的核函数中选择，根据问题和数据的不同，选择不同的参数，实际上就是得到了不同的核函数。

1. 多项式核函数 `polynomial kernel function`

   $$K(x, z) = (x \cdot z + 1)^p$$

   对应的支持向量机是一个p次多项式分类器。在此情形下，分类决策函数成为

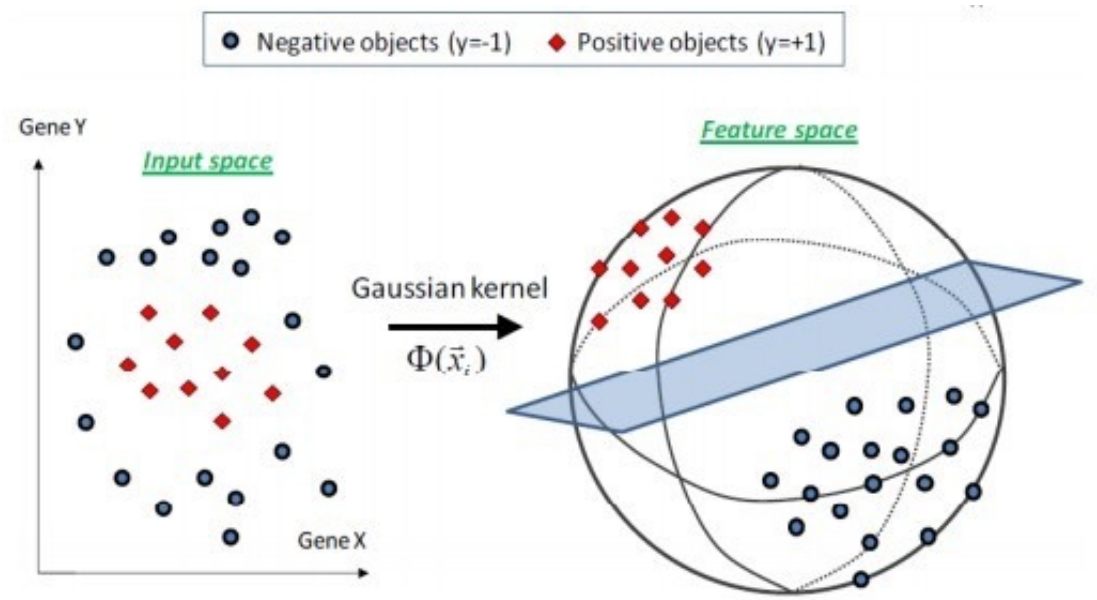
   $$f(x) = sign\left(\sum_{i=1}^{N} a_i y_i \left(x_i \cdot x + 1 \right)^p + b \right)$$

2. 高斯核函数 `gaussian kernel function`

   $$K(x, z) = \exp\left(- \frac{||x - z||^2}{2\sigma^2} \right)$$

   这个核就是会将原始空间映射为无穷维空间的那个家伙。不过如果σ选得很大的话，高次特征上的权重实际上衰减的非常快，所以实际上相当于一个低维的子空间；反过来，如果σ选得很小，则可以将任意的数据映射为线性可分，当然，这并不一定是好事，因为随之而来的可能是非常严重的过拟合问题。不过，总的来说，通过调控参数σ，高斯核实际上具有相当高的灵活性，也是使用最为广泛的核函数之一。它对应的支持向量机是高斯径向基函数分类器，在此情形下，分类决策函数称为

$$f(x) = sign\left(\sum_{i=1}^{N} a_i y_i \exp\left(-\frac{||x-z||^2}{2\sigma^2}\right) + b\right)$$



3. 字符串核函数：核函数不仅可以定义在欧式空间上，还可以定义在离散数据的集合上，比如，字符串核实定义在字符串集合上的核函数，字符串核函数在文本分类、信息检索、生物信息学等方面都有应用。

4. 线性核 $\kappa(x_1, x_2) = \langle x_1, x_2 \rangle$ ，这实际上就是原始空间中的内积。这个核存在的主要目的是使得“映射后空间中的问题”和“映射前空间中的问题”两者在形式上统一起来了。

最后，总结一下：对于非线性的情况，SVM 的处理方法是选择一个核函数 $\kappa(\cdot,\cdot)$ ，通过将数据映射到高维空间，来解决在原始空间中线性不可分的问题。由于核函数的优良品质，这样的非线性扩展在计算量上并没有比原来复杂多少，这一点是非常难得的。当然，这要归功于核方法——除了 SVM 之外，任何将计算表示为数据点的内积的方法，都可以使用核方法进行非线性扩展。

非线性支持向量机学习算法步骤如下：

1. 选取适当的核函数$K(x, z)$和适当的参数C，构造并求解最优化问题

$$\max_{a} \quad \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j y_i y_j K\left(x_i, x_j\right)$$

$$s.t. \quad \sum_{i=1}^{N} a_i y_i = 0$$

$$0 \le a_i \le C \ , \quad i = 1, 2, \cdots, N$$

求得最优解

$$a^* = \left(a_1^*, a_2^*, \cdots, a_N^*\right)^T$$

2. 选择a^*的一个正分量a_i^*适合约束条件$0 < a_i < C$,计算

$$b^* = y_j - \sum_{i=1}^{N} y_i a_i^* K\left(x_i \cdot x_j\right)$$

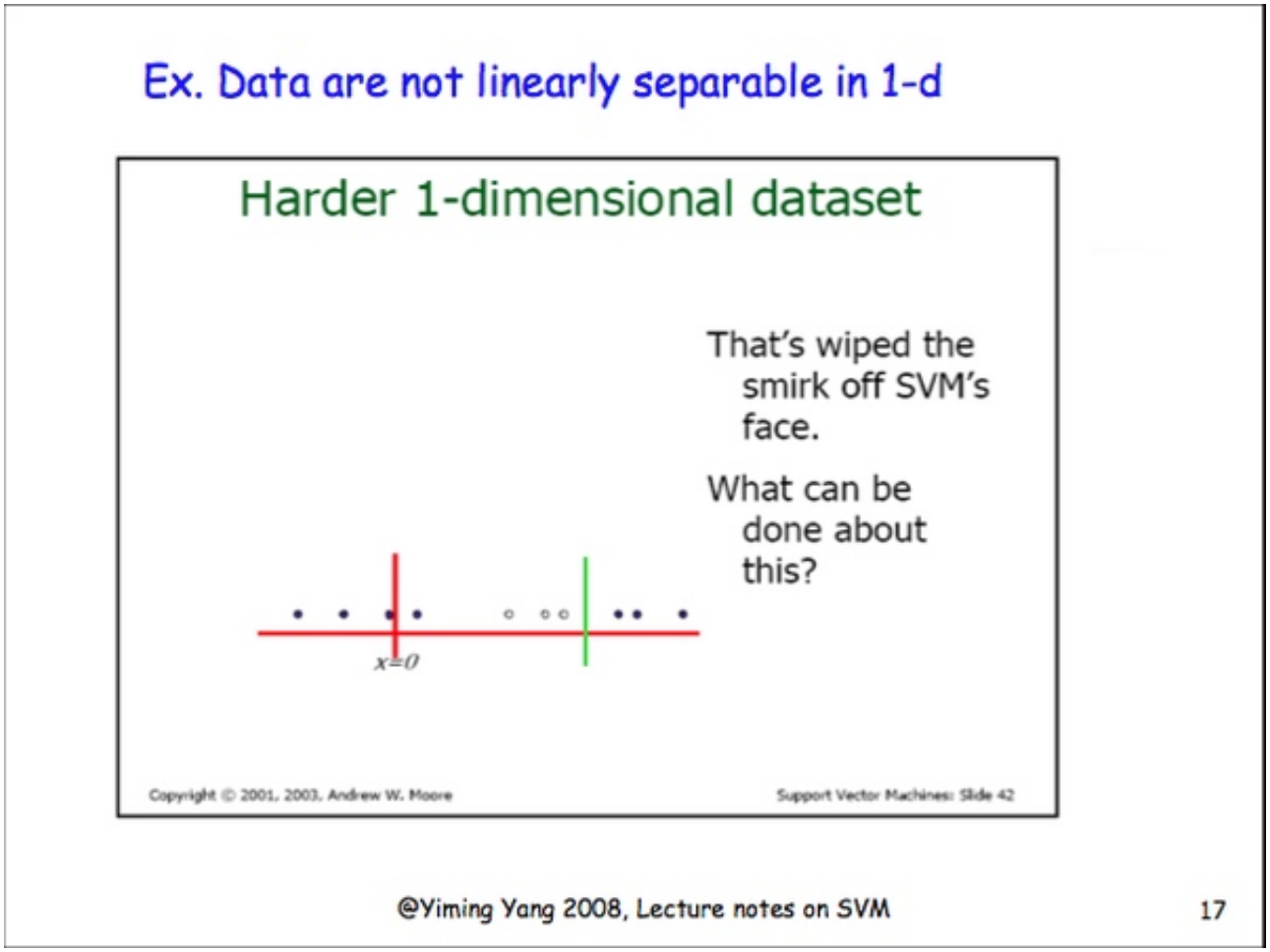
3. 构造决策函数：

$$f(X) = sign\left(\sum_{i=1}^{N} a_i^* y_i K\left(x_i, x\right) + b\right)$$

当$K(x, z)$是正定核函数时，该问题为凸二次规划问题，解是存在的。

3.3 感性认识

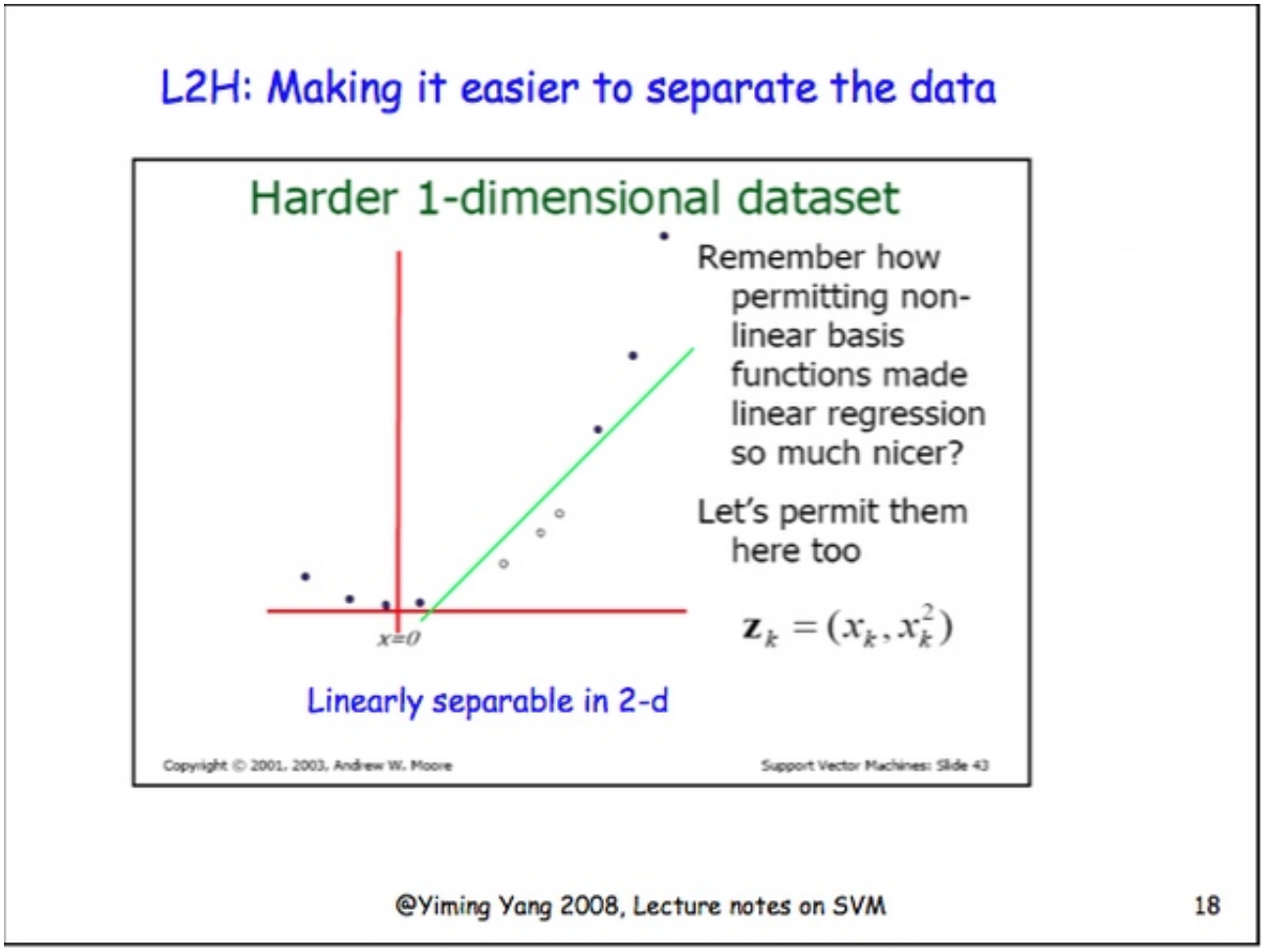
比如我们有一个一维的数据分布是如下图的样子，你想把它用一个直线来分开，你发现是不可能的，因为他们是间隔的。所以不论你画在哪，比如绿色竖线，都不可能把两个类分开。



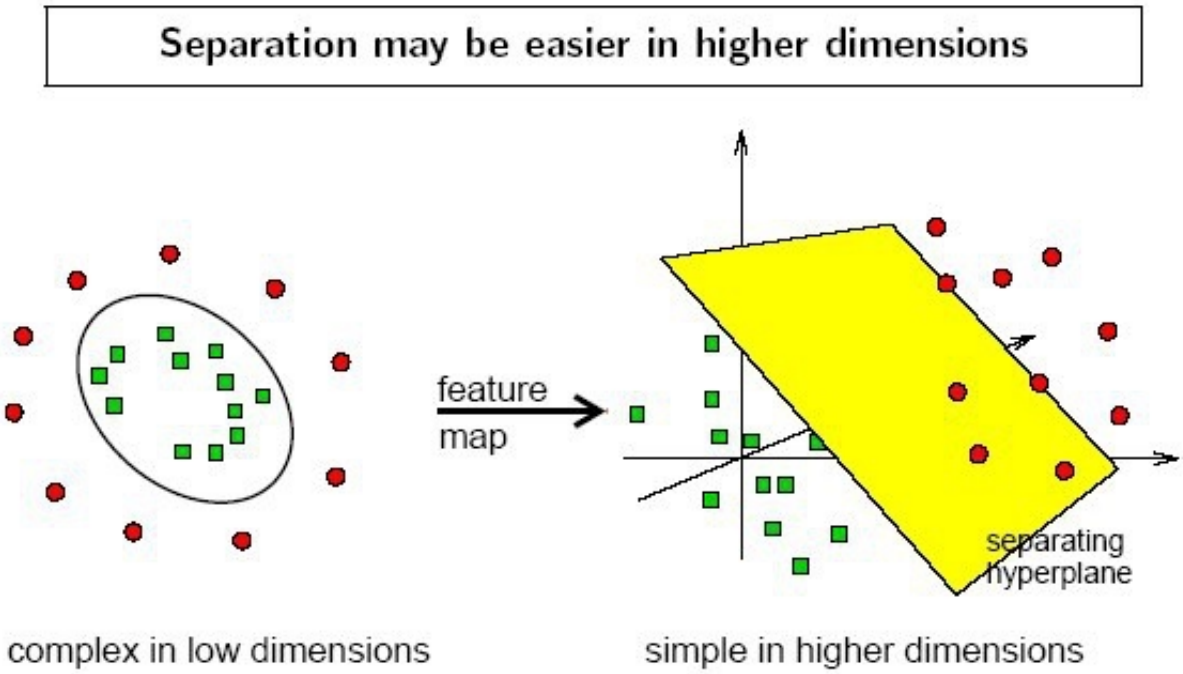
但是我们使用一个简单的升维的方法，把原来一维的空间投射到二维中，$x->(x,x^2)$。比如:

0->(0,0)
1->(1,1)
2->(2,4)

这时候就线性可分了



再举个例子，在一个二维平面里面，这样的情况是不可能只用一个平面来分类的，但是只要把它投射到三维的球体上，就可能很轻易地分类。



理论上，由于train set是有限的，当你把data投射到无限维度的空间上是一定可以在train set上完美分类的，至于在test set上就不一定了。

在实用中，很多使用者都是盲目地试验各种核函数，并扫描其中的参数，选择效果最好的，

来“避免过拟合”。至于什么样的核函数适用于什么样的问题，大多数人都不懂。核函数要满足的条件称为Mercer's condition。使用SVM的很多人甚至都不知道这个条件，也不关心它；有些不满足该条件的函数也被拿来当核函数用。

Mercer's Condition

机器学习有很多关于核函数的说法，核函数的定义和作用是什么?