支持向量机 SVM（下）

JerryLead
csxulijie@gmail.com
2011 年 3 月 17 日星期四

7 核函数（Kernels）

考虑我们最初在“线性回归”中提出的问题，特征是房子的面积 x，这里的 x 是实数，结果 y 是房子的价格。假设我们从样本点的分布中看到 x 和 y 符合 3 次曲线，那么我们希望使用 x 的三次多项式来逼近这些样本点。那么首先需要将特征 x 扩展到三维(x, x^2, x^3)，然后寻找特征和结果之间的模型。我们将这种特征变换称作特征映射（feature mapping）。映射函数称作ϕ，在这个例子中

$$\phi(x) = \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix}$$

我们希望将得到的特征映射后的特征应用于 SVM 分类，而不是最初的特征。这样，我们需要将前面$w^T x + b$公式中的内积从$< x^{(i)}, x >$，映射到$< \phi(x^{(i)}), \phi(x) >$。

至于为什么需要映射后的特征而不是最初的特征来参与计算，上面提到的（为了更好地拟合）是其中一个原因，另外的一个重要原因是样例可能存在线性不可分的情况，而将特征映射到高维空间后，往往就可分了。（在《数据挖掘导论》Pang-Ning Tan 等人著的《支持向量机》那一章有个很好的例子说明）

将核函数形式化定义，如果原始特征内积是$< x, z >$，映射后为$< \phi(x), \phi(z) >$，那么定义核函数（Kernel）为

$K(x, z) = \phi(x)^T \phi(z)$

到这里，我们可以得出结论，如果要实现该节开头的效果，只需先计算$\phi(x)$，然后计算$\phi(x)^T \phi(z)$即可，然而这种计算方式是非常低效的。比如最初的特征是 n 维的，我们将其映射到n^2维，然后再计算，这样需要$O(n^2)$的时间。那么我们能不能想办法减少计算时间呢？

先看一个例子，假设 x 和 z 都是 n 维的，

$$K(x, z) = (x^T z)^2$$

展开后，得

$$K(x, z) = (x^T z)^2 = \left(\sum_{i=1}^{n} x_i z_i \right) \left(\sum_{j=1}^{n} x_j z_j \right) = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j z_i z_j$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} (x_i x_j)(z_i z_j) = \phi(x)^T \phi(z)$$

这个时候发现我们可以只计算原始特征 x 和 z 内积的平方（时间复杂度是 O(n)），就等价与计算映射后特征的内积。也就是说我们不需要花$O(n^2)$时间了。

现在看一下映射函数（n=3 时），根据上面的公式，得到

$$\phi(x) = \begin{bmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_1 \\ x_2x_2 \\ x_2x_3 \\ x_3x_1 \\ x_3x_2 \\ x_3x_3 \end{bmatrix}.$$

也就是说核函数$K(x,z) = (x^Tz)^2$只能在选择这样的ϕ作为映射函数时才能够等价于映射后特征的内积。

再看一个核函数

$$\begin{aligned} K(x,z) &= (x^Tz + c)^2 \\ &= \sum_{i,j=1}^{n} (x_ix_j)(z_iz_j) + \sum_{i=1}^{n} (\sqrt{2c}x_i)(\sqrt{2c}z_i) + c^2. \end{aligned}$$

对应的映射函数（n=3 时）是

$$\phi(x) = \begin{bmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_1 \\ x_2x_2 \\ x_2x_3 \\ x_3x_1 \\ x_3x_2 \\ x_3x_3 \\ \sqrt{2c}x_1 \\ \sqrt{2c}x_2 \\ \sqrt{2c}x_3 \\ c \end{bmatrix},$$

更一般地，核函数$K(x,z) = (x^Tz + c)^d$对应的映射后特征维度为$\binom{\mathrm{n}+\mathrm{d}}{d}$。（这个我一直没有理解）。

由于计算的是内积，我们可以想到 IR 中的余弦相似度，如果 x 和 z 向量夹角越小，那么核函数值越大，反之，越小。因此，核函数值是$\phi(x)$和$\phi(z)$的相似度。

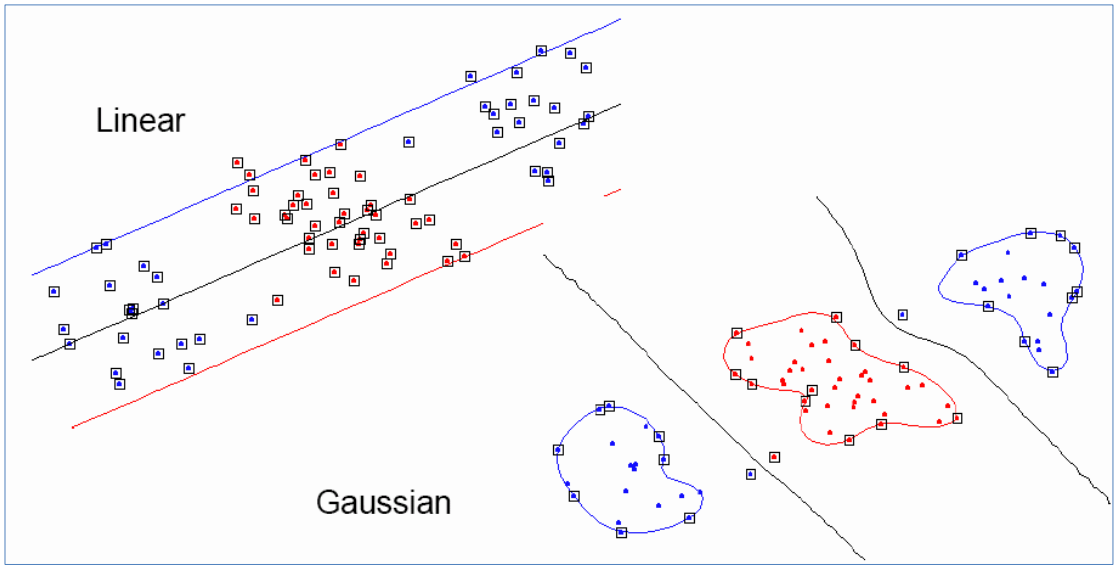
再看另外一个核函数

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

这时，如果 x 和 z 很相近（$||x-z|| \approx 0$），那么核函数值为 1，如果 x 和 z 相差很大（$||x-z|| \gg 0$），那么核函数值约等于 0。由于这个函数类似于高斯分布，因此称为高斯核函数，也叫做径向基函数(Radial Basis Function 简称 RBF)。它能够把原始特征映射到无穷维。

既然高斯核函数能够比较 x 和 z 的相似度，并映射到 0 到 1，回想 logistic 回归，sigmoid 函数可以，因此还有 sigmoid 核函数等等。

下面有张图说明在低维线性不可分时，映射到高维后就可分了，使用高斯核函数。



来自 Eric Xing 的 slides

注意，使用核函数后，怎么分类新来的样本呢？线性的时候我们使用 SVM 学习出 w 和 b，新来样本 x 的话，我们使用$w^T x + b$来判断，如果值大于等于 1，那么是正类，小于等于是负类。在两者之间，认为无法确定。如果使用了核函数后，$w^T x + b$就变成了$w^T \phi(x) + b$，是否先要找到$\phi(x)$，然后再预测？答案肯定不是了，找$\phi(x)$很麻烦，回想我们之前说过的

$$\begin{aligned} w^T x + b &= \left(\sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} \right)^T x + b \\ &= \sum_{i=1}^{m} \alpha_i y^{(i)} \langle x^{(i)}, x \rangle + b. \end{aligned}$$

只需将$< x^{(i)}, x >$替换成$K(x^{(i)}, x)$，然后值的判断同上。

8 核函数有效性判定

问题：给定一个函数 K，我们能否使用 K 来替代计算 $\phi(x)^T \phi(z)$，也就说，是否能够找出一个ϕ，使得对于所有的 x 和 z，都有$K(x,z) = \phi(x)^T \phi(z)$?

比如给出了$K(x,z) = (x^T z)^2$，是否能够认为 K 是一个有效的核函数。

下面来解决这个问题，给定 m 个训练样本$\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$，每一个$x^{(i)}$对应一个特征向量。那么，我们可以将任意两个$x^{(i)}$和$x^{(j)}$带入 K 中，计算得到$K_{ij} = K(x^{(i)}, x^{(j)})$。I 可以从 1 到 m，j 可以从 1 到 m，这样可以计算出 m*m 的核函数矩阵（Kernel Matrix）。为了方便，我们将核函数矩阵和$K(x,z)$都使用 K 来表示。

如果假设 K 是有效的核函数，那么根据核函数定义

$$K_{ij} = K\left(x^{(i)}, x^{(j)}\right) = \phi\left(x^{(i)}\right)^T \phi\left(x^{(j)}\right) = \phi\left(x^{(j)}\right)^T \phi\left(x^{(i)}\right) = K\left(x^{(j)}, x^{(i)}\right) = K_{ji}$$

可见，矩阵 K 应该是个对称阵。让我们得出一个更强的结论，首先使用符号$\phi_k(x)$来表示映射函数$\phi(x)$的第 k 维属性值。那么对于任意向量 z，得

$$\begin{aligned} z^T K z &= \sum_i \sum_j z_i K_{ij} z_j \\ &= \sum_i \sum_j z_i \phi(x^{(i)})^T \phi(x^{(j)}) z_j \\ &= \sum_i \sum_j z_i \sum_k \phi_k(x^{(i)}) \phi_k(x^{(j)}) z_j \\ &= \sum_k \sum_i \sum_j z_i \phi_k(x^{(i)}) \phi_k(x^{(j)}) z_j \\ &= \sum_k \left(\sum_i z_i \phi_k(x^{(i)}) \right)^2 \\ &\geq 0. \end{aligned}$$

最后一步和前面计算$K(x,z) = (x^T z)^2$时类似。从这个公式我们可以看出，如果 K 是个有效的核函数（即$K(x,z)$和$\phi(x)^T\phi(z)$等价），那么，在训练集上得到的核函数矩阵 K 应该是半正定的（$K \geq 0$）

这样我们得到一个核函数的必要条件：

K 是有效的核函数 ==> 核函数矩阵 K 是对称半正定的。

可幸的是，这个条件也是充分的，由 Mercer 定理来表达。

Mercer 定理：

如果函数 K 是$\mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$上的映射（也就是从两个 n 维向量映射到实数域）。那么如果 K 是一个有效核函数（也称为 Mercer 核函数），那么当且仅当对于训练样例$\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$，其相应的核函数矩阵是对称半正定的。

Mercer 定理表明为了证明 K 是有效的核函数，那么我们不用去寻找ϕ，而只需要在训练集上求出各个K_{ij}，然后判断矩阵 K 是否是半正定（使用左上角主子式大于等于零等方法）即可。

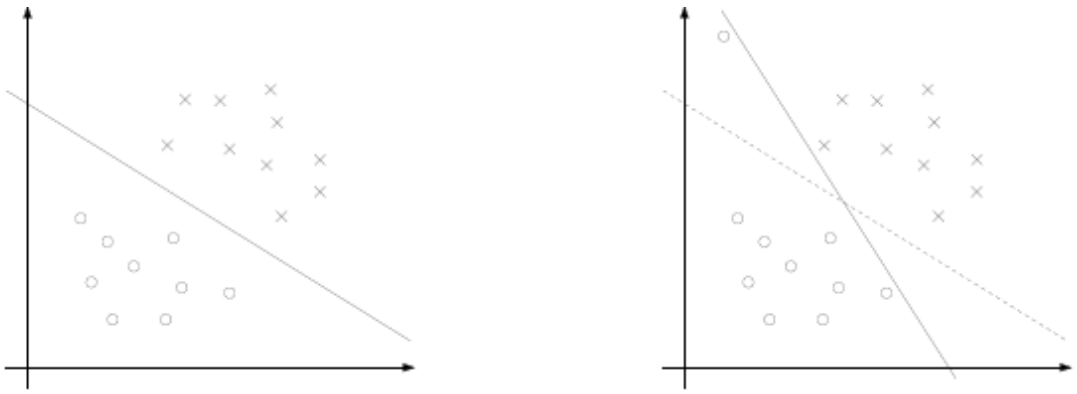
许多其他的教科书在 Mercer 定理证明过程中使用了L^2范数和再生希尔伯特空间等概念，但在特征是 n 维的情况下，这里给出的证明是等价的。

核函数不仅仅用在 SVM 上，但凡在一个模型后算法中出现了$< x, z >$，我们都可以常使用$K(x,z)$去替换，这可能能够很好地改善我们的算法。

9 规则化和不可分情况处理（Regularization and the non-separable case）

我们之前讨论的情况都是建立在样例线性可分的假设上，当样例线性不可分时，我们可以尝试使用核函数来将特征映射到高维，这样很可能就可分了。然而，映射后我们也不能100%保证可分。那怎么办呢，我们需要将模型进行调整，以保证在不可分的情况下，也能够尽可能地找出分隔超平面。

看下面两张图：



可以看到一个离群点（可能是噪声）可以造成超平面的移动，间隔缩小，可见以前的模型对噪声非常敏感。再有甚者，如果离群点在另外一个类中，那么这时候就是线性不可分了。

这时候我们应该允许一些点游离并在在模型中违背限制条件（函数间隔大于 1）。我们设计得到新的模型如下（也称软间隔）：

$$\begin{aligned} \min_{\gamma,w,b} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, \quad i = 1, \ldots, m \\ & \xi_i \geq 0, \quad i = 1, \ldots, m. \end{aligned}$$

引入非负参数ξ_i后（称为松弛变量），就允许某些样本点的函数间隔小于 1，即在最大间隔区间里面，或者函数间隔是负数，即样本点在对方的区域中。而放松限制条件后，我们需要重新调整目标函数，以对离群点进行处罚，目标函数后面加上的$C\sum_{i=1}^{m}\xi_i$就表示离群点越多，目标函数值越大，而我们要求的是尽可能小的目标函数值。这里的 C 是离群点的权重，C 越大表明离群点对目标函数影响越大，也就是越不希望看到离群点。我们看到，目标函数控制了离群点的数目和程度，使大部分样本点仍然遵守限制条件。

模型修改后，拉格朗日公式也要修改如下：

$$\mathcal{L}(w, b, \xi, \alpha, r) = \frac{1}{2}w^T w + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i\left[y^{(i)}(x^T w + b) - 1 + \xi_i\right] - \sum_{i=1}^{m} r_i\xi_i.$$

这里的α_i和γ_i都是拉格朗日乘子，回想我们在拉格朗日对偶中提到的求法，先写出拉格朗日公式（如上），然后将其看作是变量 w 和 b 的函数，分别对其求偏导，得到 w 和 b 的表达式。然后代入公式中，求带入后公式的极大值。整个推导过程类似以前的模型，这里只写出最后结果如下：

$$\begin{aligned} \max_{\alpha} \quad & W(\alpha) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)}y^{(j)}\alpha_i\alpha_j\langle x^{(i)}, x^{(j)}\rangle \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, m \\ & \sum_{i=1}^{m}\alpha_i y^{(i)} = 0, \end{aligned}$$

此时，我们发现没有了参数ξ_i，与之前模型唯一不同在于α_i又多了$\alpha_i \leq C$的限制条件。需要提醒的是，b 的求值公式也发生了改变，改变结果在 SMO 算法里面介绍。先看看 KKT

条件的变化：

$$\alpha_i = 0 \quad \Rightarrow \quad y^{(i)}(w^T x^{(i)} + b) \geq 1 \tag{14}$$

$$\alpha_i = C \quad \Rightarrow \quad y^{(i)}(w^T x^{(i)} + b) \leq 1 \tag{15}$$

$$0 < \alpha_i < C \quad \Rightarrow \quad y^{(i)}(w^T x^{(i)} + b) = 1. \tag{16}$$

第一个式子表明在两条间隔线外的样本点前面的系数为 0，离群样本点前面的系数为 C，而支持向量（也就是在超平面两边的最大间隔线上）的样本点前面系数在(0,C)上。通过 KKT 条件可知，某些在最大间隔线上的样本点也不是支持向量，相反也可能是离群点。

10 坐标上升法（Coordinate ascent）

在最后讨论W(α)的求解之前，我们先看看坐标上升法的基本原理。假设要求解下面的优化问题：

$$\max_{\alpha} W(\alpha_1, \alpha_2, \ldots, \alpha_m).$$

这里 W 是α向量的函数。之前我们在回归中提到过两种求最优解的方法，一种是梯度下降法，另外一种是牛顿法。现在我们再讲一种方法称为坐标上升法（求解最小值问题时，称作坐标下降法，原理一样）。

方法过程：

Loop until convergence: {

  For $i = 1, \ldots, m$, {

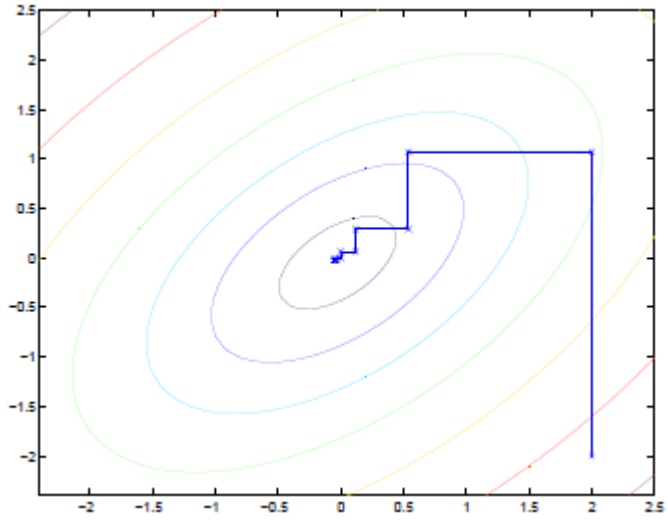
    $\alpha_i := \arg\max_{\hat{\alpha}_i} W(\alpha_1, \ldots, \alpha_{i-1}, \hat{\alpha}_i, \alpha_{i+1}, \ldots, \alpha_m)$.

  }

}

最里面语句的意思是固定除α_i之外的所有$\alpha_j (j \neq i)$，这时 W 可看作只是关于α_i的函数，那么直接对α_i求导优化即可。这里我们进行最大化求导的顺序 i 是从 1 到 m，可以通过更改优化顺序来使 W 能够更快地增加并收敛。如果 W 在内循环中能够很快地达到最优，那么坐标上升法会是一个很高效的求极值方法。

下面通过一张图来展示：



椭圆代表了二次函数的各个等高线，变量数为 2，起始坐标是(2,-2)。图中的直线式迭代优化的路径，可以看到每一步都会向最优值前进一步，而且前进路线是平行于坐标轴的，因为每一步只优化一个变量。

11 SMO 优化算法（Sequential minimal optimization）

SMO 算法由 Microsoft Research 的 John C. Platt 在 1998 年提出，并成为最快的二次规划优化算法，特别针对线性 SVM 和数据稀疏时性能更优。关于 SMO 最好的资料就是他本人写的《Sequential Minimal Optimization A Fast Algorithm for Training Support Vector Machines》了。

我拜读了一下，下面先说讲义上对此方法的总结。

首先回到我们前面一直悬而未解的问题，对偶函数最后的优化问题：

$$\begin{aligned}\max_\alpha \quad & W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle \\ \text{s.t.} \quad & 0 \le \alpha_i \le C, \quad i = 1, \ldots, m \\ & \sum_{i=1}^{m} \alpha_i y^{(i)} = 0,\end{aligned}$$

要解决的是在参数$\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$上求最大值 W 的问题，至于$x^{(i)}$和$y^{(i)}$都是已知数。C 由我们预先设定，也是已知数。

按照坐标上升的思路，我们首先固定除α_1以外的所有参数，然后在α_1上求极值。等一下，这个思路有问题，因为如果固定α_1以外的所有参数，那么α_1将不再是变量（可以由其他值推出），因为问题中规定了

$$\alpha_1 y^{(1)} = -\sum_{i=2}^{m} \alpha_i y^{(i)}.$$

因此，我们需要一次选取两个参数做优化，比如α_1和α_2，此时α_2可以由α_1和其他参数表示出来。这样回带到 W 中，W 就只是关于α_1的函数了，可解。

这样，SMO 的主要步骤如下：

Repeat till convergence {

1. Select some pair α_i and α_j to update next (using a heuristic that tries to pick the two that will allow us to make the biggest progress towards the global maximum).
2. Reoptimize $W(\alpha)$ with respect to α_i and α_j, while holding all the other α_k's $(k \neq i, j)$ fixed.

}

意思是，第一步选取一对α_i和α_j，选取方法使用启发式方法（后面讲）。第二步，固定除α_i和α_j之外的其他参数，确定 W 极值条件下的α_i，α_j由α_i表示。

SMO 之所以高效就是因为在固定其他参数后，对一个参数优化过程很高效。

下面讨论具体方法：

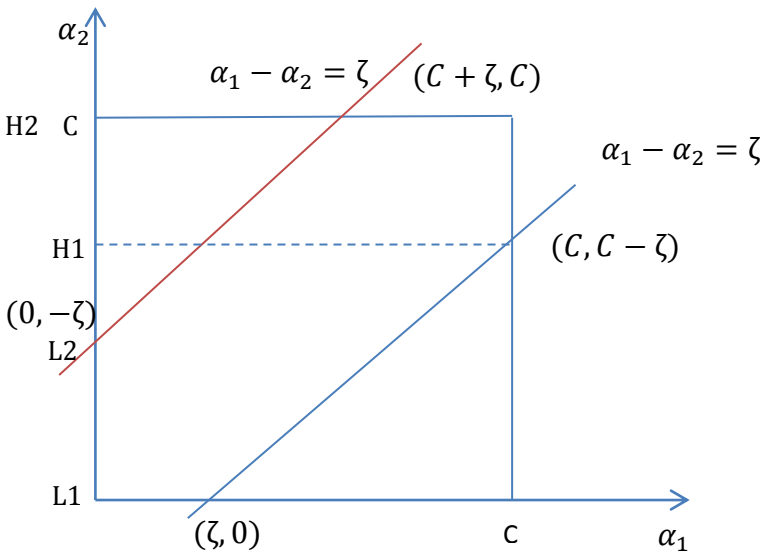
假设我们选取了初始值$\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$满足了问题中的约束条件。接下来，我们固定$\{\alpha_3, \alpha_4, \ldots, \alpha_n\}$，这样 W 就是$\alpha_1$和$\alpha_2$的函数。并且$\alpha_1$和$\alpha_2$满足条件：

$$\alpha_1 y^{(1)} + \alpha_2 y^{(2)} = -\sum_{i=3}^{m} \alpha_i y^{(i)}.$$

由于$\{\alpha_3, \alpha_4, \ldots, \alpha_n\}$都是已知固定值，因此为了方面，可将等式右边标记成实数值ζ。

$$\alpha_1 y^{(1)} + \alpha_2 y^{(2)} = \zeta.$$

当$y^{(1)}$和$y^{(2)}$异号时，也就是一个为 1，一个为-1 时，他们可以表示成一条直线，斜率为 1。如下图：



横轴是α_1，纵轴是α_2，α_1和α_2既要在矩形方框内，也要在直线上，因此

$$L = max(0, \alpha_2 - \alpha_1),\ H = min(C, C + \alpha_2 - \alpha_1)$$

同理，当$y^{(1)}$和$y^{(2)}$同号时，

$$L = max(0, \alpha_2 + \alpha_1 - C),\ H = min(C, \alpha_2 + \alpha_1)$$

然后我们打算将α_1用α_2表示：

$$\alpha_1 = (\zeta - \alpha_2 y^{(2)}) y^{(1)}.$$

然后反代入 W 中，得

$$W(\alpha_1, \alpha_2, \ldots, \alpha_m) = W((\zeta - \alpha_2 y^{(2)}) y^{(1)}, \alpha_2, \ldots, \alpha_m).$$

展开后 W 可以表示成a${\alpha_2}^2 + b\alpha_2 + c$。其中 a,b,c 是固定值。这样，通过对 W 进行求导可以得到α_2，然而要保证α_2满足$L \le \alpha_2 \le H$，我们使用${\alpha_2}^{new,unclipped}$表示求导求出来的$\alpha_2$，然而最后的$\alpha_2$，要根据下面情况得到：

$$\alpha_2^{new} = \begin{cases} H & \text{if } \alpha_2^{new,unclipped} > H \\ \alpha_2^{new,unclipped} & \text{if } L \le \alpha_2^{new,unclipped} \le H \\ L & \text{if } \alpha_2^{new,unclipped} < L \end{cases}$$

这样得到${\alpha_2}^{new}$后，我们可以得到α_1的新值${\alpha_1}^{new}$。

下面进入 Platt 的文章，来找到启发式搜索的方法和求 b 值的公式。

这篇文章使用的符号表示有点不太一样，不过实质是一样的，先来熟悉一下文章中符号的表示。

文章中定义特征到结果的输出函数为

$$u = \vec{w} \cdot \vec{x} - b, \qquad (1)$$

与我们之前的$w^T x^{(i)} + b$实质是一致的。

原始的优化问题为：

$$\min_{w,b} \tfrac{1}{2} \|\vec{w}\|^2 \text{ subject to } y_i(\vec{w} \cdot \vec{x}_i - b) \ge 1, \forall i, \qquad (3)$$

求导得到：

$$\vec{w} = \sum_{i=1}^{N} y_i \alpha_i \vec{x}_i, \quad b = \vec{w} \cdot \vec{x}_k - y_k \text{ for some } \alpha_k > 0. \qquad (7)$$

经过对偶后为：

$$\min_{\vec{\alpha}} \Psi(\vec{\alpha}) = \min_{\vec{\alpha}} \tfrac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j (\vec{x}_i \cdot \vec{x}_j) \alpha_i \alpha_j - \sum_{i=1}^{N} \alpha_i,$$

s.t. $\alpha_i \ge 0, \forall i,$

$$\sum_{i=1}^{N} y_i \alpha_i = 0.$$

这里与 W 函数是一样的，只是符号求反后，变成求最小值了。y_i和$y^{(i)}$是一样的，都表示第 i 个样本的输出结果（1 或-1）。

经过加入松弛变量ξ_i后，模型修改为：

$$\min_{\vec{w},b,\vec{\xi}} \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{N}\xi_i \quad \text{subject to } y_i(\vec{w}\cdot\vec{x}_i - b) \geq 1-\xi_i, \forall i, \tag{8}$$

$$0 \leq \alpha_i \leq C, \forall i. \tag{9}$$

由公式（7）代入（1）中可知，

$$u = \sum_{j=1}^{N} y_j \alpha_j K(\vec{x}_j, \vec{x}) - b, \tag{10}$$

这个过程和之前对偶过程一样。

重新整理我们要求的问题为：

$$\begin{aligned} \min_{\vec{\alpha}} \Psi(\vec{\alpha}) = \min_{\vec{\alpha}} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j K(\vec{x}_i, \vec{x}_j)\alpha_i\alpha_j - \sum_{i=1}^{N}\alpha_i, \\ 0 \leq \alpha_i \leq C, \forall i, \\ \sum_{i=1}^{N} y_i \alpha_i = 0. \end{aligned} \tag{11}$$

与之对应的 KKT 条件为：

$$\begin{aligned} \alpha_i = 0 &\Leftrightarrow y_i u_i \geq 1, \\ 0 < \alpha_i < C &\Leftrightarrow y_i u_i = 1, \\ \alpha_i = C &\Leftrightarrow y_i u_i \leq 1. \end{aligned} \tag{12}$$

这个 KKT 条件说明，在两条间隔线外面的点，对应前面的系数α_i为 0，在两条间隔线里面的对应α_i为 C，在两条间隔线上的对应的系数α_i在 0 和 C 之间。

将我们之前得到 L 和 H 重新拿过来：

$$L = \max(0, \alpha_2 - \alpha_1), \qquad H = \min(C, C + \alpha_2 - \alpha_1). \tag{13}$$

$$L = \max(0, \alpha_2 + \alpha_1 - C), \qquad H = \min(C, \alpha_2 + \alpha_1). \tag{14}$$

之前我们将问题进行到这里，然后说将α_1用α_2表示后代入 W 中，这里将代入Ψ中，得

$$\Psi = \frac{1}{2}K_{11}\alpha_1^2 + \frac{1}{2}K_{22}\alpha_2^2 + sK_{12}\alpha_1\alpha_2 + y_1\alpha_1 v_1 + y_2\alpha_2 v_2 - \alpha_1 - \alpha_2 + \Psi_{\text{constant}}, \tag{24}$$

其中

$$\begin{aligned} K_{ij} &= K(\vec{x}_i, \vec{x}_j), \\ v_i &= \sum_{j=3}^{N} y_j \alpha_j^* K_{ij} = u_i + b^* - y_1\alpha_1^* K_{1i} - y_2\alpha_2^* K_{2i}, \end{aligned} \tag{25}$$

这里的α_1^*和α_2^*代表某次迭代前的原始值，因此是常数，而α_1和α_2是变量，待求。公式（24）中的最后一项是常数。

由于α_1和α_2满足以下公式

$$y_1\alpha_1{}^* + y_2\alpha_2{}^* = -\sum_{i=3}^{n} y_i\alpha_i{}^* = y_1\alpha_1 + y_2\alpha_2$$

因为$\alpha_i{}^*(i > 2)$的值是固定值，在迭代前后不会变。

那么用 s 表示y_1y_2，上式两边乘以y_1时，变为：

$$\alpha_1 + s\alpha_2 = \alpha_1^* + s\alpha_2^* = w. \tag{26}$$

其中

$$w = -y_1\sum_{i=3}^{n} y_i\alpha_i{}^*$$

代入（24）中，得

$$\begin{aligned}\Psi = {} & \tfrac{1}{2}K_{11}(w - s\alpha_2)^2 + \tfrac{1}{2}K_{22}\alpha_2^2 + sK_{12}(w - s\alpha_2)\alpha_2 \\ & + y_1(w - s\alpha_2)v_1 - w + s\alpha_2 + y_2\alpha_2 v_2 - \alpha_2 + \Psi_{\text{constant}}.\end{aligned} \tag{27}$$

这时候只有α_2是变量了，求导

$$\frac{d\Psi}{d\alpha_2} = -sK_{11}(w - s\alpha_2) + K_{22}\alpha_2 - K_{12}\alpha_2 + sK_{12}(w - s\alpha_2) - y_2v_1 + s + y_2v_2 - 1 = 0. \tag{28}$$

如果Ψ的二阶导数大于 0（凹函数），那么一阶导数为 0 时，就是极小值了。

假设其二阶导数为 0（一般成立），那么上式化简为：

$$\alpha_2(K_{11} + K_{22} - 2K_{12}) = s(K_{11} - K_{12})w + y_2(v_1 - v_2) + 1 - s. \tag{29}$$

将 w 和 v 代入后，继续化简推导，得（推导了六七行推出来了）

$$\alpha_2(K_{11} + K_{22} - 2K_{12}) = \alpha_2^*(K_{11} + K_{22} - 2K_{12}) + y_2(u_1 - u_2 + y_2 - y_1). \tag{30}$$

我们使用η来表示：

$$\eta = K(\vec{x}_1, \vec{x}_1) + K(\vec{x}_2, \vec{x}_2) - 2K(\vec{x}_1, \vec{x}_2). \tag{15}$$

通常情况下目标函数是正定的，也就是说，能够在直线约束方向上求得最小值，并且$\eta > 0$。

那么我们在（30）两边都除以η可以得到

$$\alpha_2^{\text{new}} = \alpha_2 + \frac{y_2(E_1 - E_2)}{\eta}, \tag{16}$$

这里我们使用$\alpha_2{}^{new}$表示优化后的值，α_2是迭代前的值，$E_i = u_i - y_i$。

与之前提到的一样$\alpha_2{}^{new}$不是最终迭代后的值，需要进行约束：

$$\alpha_2^{\text{new,clipped}} = \begin{cases} H & \text{if} \quad \alpha_2^{\text{new}} \geq H; \\ \alpha_2^{\text{new}} & \text{if} \quad L < \alpha_2^{\text{new}} < H; \\ \text{L} & \text{if} \quad \alpha_2^{\text{new}} \leq L. \end{cases} \tag{17}$$

那么

$$\alpha_1^{\text{new}} = \alpha_1 + s(\alpha_2 - \alpha_2^{\text{new,clipped}}). \tag{18}$$

在特殊情况下，η可能不为正，如果核函数 K 不满足 Mercer 定理，那么目标函数可能变得非正定，η可能出现负值。即使 K 是有效的核函数，如果训练样本中出现相同的特征 x，那么η仍有可能为 0。SMO 算法在η不为正值的情况下仍有效。为保证有效性，我们可以推导出η就是Ψ的二阶导数，$\eta < 0$，Ψ没有极小值，最小值在边缘处取到（类比$y = -x^2$），$\eta = 0$时更是单调函数了，最小值也在边缘处取得，而α_2的边缘就是 L 和 H。这样将$\alpha_2 = L$和$\alpha_2 = H$分别代入Ψ中即可求得Ψ的最小值，相应的$\alpha_2 =$ L还是$\alpha_2 = H$也可以知道了。具体计算公式如下：

$$\begin{aligned}
f_1 &= y_1(E_1 + b) - \alpha_1 K(\vec{x}_1, \vec{x}_1) - s\alpha_2 K(\vec{x}_1, \vec{x}_2), \\
f_2 &= y_2(E_2 + b) - s\alpha_1 K(\vec{x}_1, \vec{x}_2) - \alpha_2 K(\vec{x}_2, \vec{x}_2), \\
L_1 &= \alpha_1 + s(\alpha_2 - L), \\
H_1 &= \alpha_1 + s(\alpha_2 - H), \\
\Psi_L &= L_1 f_1 + L f_2 + \tfrac{1}{2} L_1^2 K(\vec{x}_1, \vec{x}_1) + \tfrac{1}{2} L^2 K(\vec{x}_2, \vec{x}_2) + sLL_1 K(\vec{x}_1, \vec{x}_2), \\
\Psi_H &= H_1 f_1 + H f_2 + \tfrac{1}{2} H_1^2 K(\vec{x}_1, \vec{x}_1) + \tfrac{1}{2} H^2 K(\vec{x}_2, \vec{x}_2) + sHH_1 K(\vec{x}_1, \vec{x}_2).
\end{aligned} \tag{19}$$

至此，迭代关系式除了 b 的推导式以外，都已经推出。

b 每一步都要更新，因为前面的 KKT 条件指出了α_i和$y_i u_i$的关系，而u_i和 b 有关，在每一步计算出α_i后，根据 KKT 条件来调整 b。

b 的更新有几种情况：

b的更新：选择b使得关于乘子α_1或α_1的KKT条件成立

$$b_1 = E_1 + y_1(\alpha_1^{new} - \alpha_1)k(x_1, x_1) + y_2(\alpha_2^{new,clipped} - \alpha_2)k(x_1, x_2) + b \quad (7)$$

$$b_2 = E_2 + y_1(\alpha_1^{new} - \alpha_1)k(x_1, x_2) + y_2(\alpha_2^{new,clipped} - \alpha_2)k(x_2, x_2) + b \quad (8)$$

如果α_1^{new}在界内，则$b^{new} = b_1$；如果$\alpha_2^{new,clipped}$在界内，$b^{new} = b_2$；

如果α_1^{new}和$\alpha_2^{new,clipped}$都在界内，那么$b_1 = b_2$，则$b^{new} = b_1 = b_2$；

如果α_1^{new}和$\alpha_2^{new,clipped}$都在界上，那么b_1和b_2之间的任何数都满足KKT条件，都可作为b的更新值，一般取$b^{new} = (b_1 + b_2)/2$.

来自罗林开的 ppt

这里的界内指$0 < \alpha_i < C$，界上就是等于 0 或者 C 了。

前面两个的公式推导可以根据

$$y_1\alpha_1^* + y_2\alpha_2^* = -\sum_{i=3}^{n} y_i\alpha_i^* = y_1\alpha_1 + y_2\alpha_2$$

和对于$0 < \alpha_i < C$有$y_i u_i = 1$的 KKT 条件推出。

这样全部参数的更新公式都已经介绍完毕，附加一点，如果使用的是线性核函数，我们就可以继续使用 w 了，这样不用扫描整个样本库来作内积了。

w 值的更新方法为：

$$\vec{w}^{\text{new}} = \vec{w} + y_1(\alpha_1^{\text{new}} - \alpha_1)\vec{x}_1 + y_2(\alpha_2^{\text{new,clipped}} - \alpha_2)\vec{x}_2. \quad (22)$$

根据前面的

$$\vec{w} = \sum_{i=1}^{N} y_i \alpha_i \vec{x}_i, \quad b = \vec{w} \cdot \vec{x}_k - y_k \text{ for some } \alpha_k > 0. \quad (7)$$

公式推导出。

12 SMO 中拉格朗日乘子的启发式选择方法

终于到了最后一个问题了，所谓的启发式选择方法主要思想是每次选择拉格朗日乘子的时候，优先选择样本前面系数$0 < \alpha_i < C$的α_i作优化（论文中称为无界样例），因为在界上（α_i为 0 或 C）的样例对应的系数α_i一般不会更改。

这条启发式搜索方法是选择第一个拉格朗日乘子用的，比如前面的α_2。那么这样选择的话，是否最后会收敛？可幸的是 Osuna 定理告诉我们只要选择出来的两个α_i中有一个违背了 KKT 条件，那么目标函数在一步迭代后值会减小。违背 KKT 条件不代表$0 < \alpha_i < C$，在界上也有可能会违背。是的，因此在给定初始值α_i=0 后，先对所有样例进行循环，循环中碰到违背 KKT 条件的（不管界上还是界内）都进行迭代更新。等这轮过后，如果没有收敛，第二轮就只针对$0 < \alpha_i < C$的样例进行迭代更新。

在第一个乘子选择后，第二个乘子也使用启发式方法选择，第二个乘子的迭代步长大致正比于$|\mathrm{E}_1 - \mathrm{E}_2|$，选择第二个乘子能够最大化$|\mathrm{E}_1 - \mathrm{E}_2|$。即当$\mathrm{E}_1$为正时选择负的绝对值最大的$\mathrm{E}_2$，反之，选择正值最大的$\mathrm{E}_2$。

最后的收敛条件是在界内（$0 < \alpha_i < C$）的样例都能够遵循 KKT 条件，且其对应的α_i只在极小的范围内变动。

至于如何写具体的程序，请参考 John C. Platt 在论文中给出的伪代码。

13 总结

这份 SVM 的讲义重点概括了 SVM 的基本概念和基本推导，中规中矩却又让人醍醐灌顶。起初让我最头疼的是拉格朗日对偶和 SMO，后来逐渐明白拉格朗日对偶的重要作用是将 w 的计算提前并消除 w，使得优化函数变为拉格朗日乘子的单一参数优化问题。而 SMO 里面迭代公式的推导也着实让我花费了不少时间。

对比这么复杂的推导过程，SVM 的思想确实那么简单。它不再像 logistic 回归一样企图去拟合样本点（中间加了一层 sigmoid 函数变换），而是就在样本中去找分隔线，为了评判哪条分界线更好，引入了几何间隔最大化的目标。

之后所有的推导都是去解决目标函数的最优化上了。在解决最优化的过程中，发现了 w 可以由特征向量内积来表示，进而发现了核函数，仅需要调整核函数就可以将特征进行低维到高维的变换，在低维上进行计算，实质结果表现在高维上。由于并不是所有的样本都可分，为了保证 SVM 的通用性，进行了软间隔的处理，导致的结果就是将优化问题变得更加复杂，然而惊奇的是松弛变量没有出现在最后的目标函数中。最后的优化求解问题，也被拉格朗日对偶和 SMO 算法化解，使 SVM 趋向于完美。

另外，其他很多议题如 SVM 背后的学习理论、参数选择问题、二值分类到多值分类等等还没有涉及到，以后有时间再学吧。其实朴素贝叶斯在分类二值分类问题时，如果使用对数比，那么也算作线性分类器。