第四章、支撑向量机

赵涵

2022 年 1 月 23 日

支撑向量机（Support Vector Machine）算法在分类问题中有着重要地位，其主要思想是最大化两类之间的间隔。按照数据集的特点：

1. 线性可分问题，如之前的感知机算法处理的问题。
2. 线性可分，只有一点点错误点，如感知机算法发展出来的Pocket算法处理的问题。
3. 非线性问题，完全不可分，如在感知机问题发展出来的多层感知机和深度学习。

这三种情况对于SVM分别有下面三种处理手段：

1. 硬间隔hard-margin SVM
2. 软间隔soft-margin SVM
3. 核方法kernel Method

SVM的求解中，大量用到了Lagrange乘子法，首先对这种方法进行介绍。

1 约束优化问题

一般地，约束优化问题（原问题）可以写成：

$$\min_{x\in\mathbb{R}^p} f(x) \tag{1}$$

$$s.t.\ m_i(x)\le 0, i=1,2,\dots,M \tag{2}$$

$$n_j(x)=0, j=1,2,\dots,N \tag{3}$$

定义Lagrange函数：

$$L(x,\lambda,\eta)=f(x)+\sum_{i=1}^{M}\lambda_i m_i(x)+\sum_{i=1}^{N}\eta_i n_i(x) \tag{4}$$

那么原问题可以等价于无约束形式：

$$\min_{x\in\mathbb{R}^p}\max_{\lambda,\eta} L(x,\lambda,\eta)\ s.t.\ \lambda_i\ge 0 \tag{5}$$

这是由于，当满足原问题的不等式约束的时候，$\lambda_i=0$才能取得最大值，直接等价于原问题，如果不满足原问题的不等式约束，那么最大值就为$+\infty$，由于需要取最小值，于是不会取到这个情况。

这个问题的对偶形式：

$$\max_{\lambda,\eta}\min_{x\in\mathbb{R}^p} L(x,\lambda,\eta)\ s.t.\ \lambda_i\ge 0 \tag{6}$$

对偶问题是关于λ,η的最大化问题。由于：

$$\max_{\lambda,\eta}\min_{x\in\mathbb{R}^p} L(x,\lambda,\eta)\le\min_{x\in\mathbb{R}^p}\max_{\lambda,\eta} L(x,\lambda,\eta)^1 \tag{7}$$

[1]证明：显然有$\min_x L\le L\le\max_{\lambda,\eta} L$，于是显然有$\max_{\lambda,\eta} L\le L$，且$\min_x\max_{\lambda,\eta} L\ge L$。

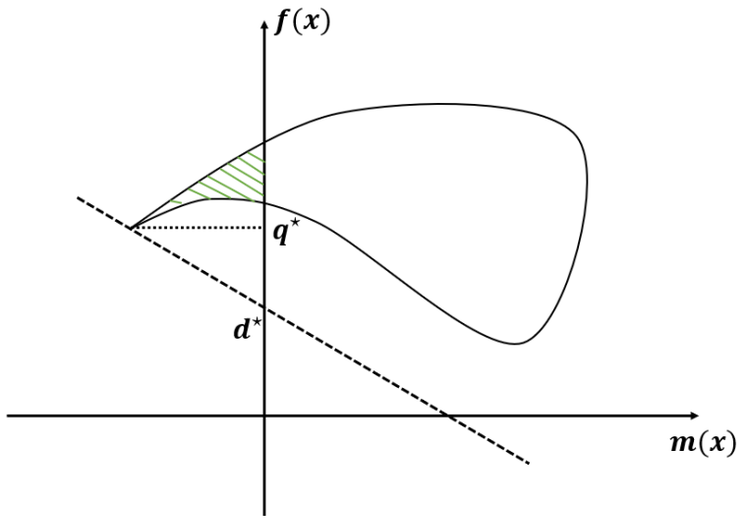

Figure 1: 对偶问题之间的关系。

对偶问题地解小于原问题，有两种情况（等号，小于号）：

1. 强对偶：可以取等于号
2. 弱对偶：只能取小于号

我们通过以下的图（1），来进行说明： 首先我们需要确定一个定义域，我们这里画了一个一般的定义域，即是非凸的。原问题是在限定的可行域内，有：

$$\min f(x) \quad = \quad q^\star \tag{8}$$
$$s.t.\ m(x) \quad \leq \quad 0 \tag{9}$$

对偶问题为：

$$\max_{\lambda} \min f(x) + \lambda m(x) \quad = \quad d^\star \tag{10}$$
$$s.t.\ \lambda \quad \geq \quad 0 \tag{11}$$

通过上图，就很明显能看到，$d^\star \leq q^\star$。去等号的条件是，可行域必须是凸集。对于一个凸优化问题，有如下的定理：
如果凸优化问题满足某些条件如Slater 条件，那么它和其他对偶问题满足强对偶关系。记问题的定义域为：$\mathcal{D} = domf(x) \cap domm_i(x) \cap n_j(x)$。于是Slater条件为：

$$\exists \hat{x} \in Relint\mathcal{D}\ s.t. \forall\ i = 1, 2, \ldots, M\ m_i(x) < 0 \tag{12}$$

其中Relint表示相对内部（不包含边界的部分）。[2]那么在两个情况下Slater 条件成立：

1. 对于大多数凸优化问题，Slater条件成立。
2. 松弛Slater条件，如果M个不等式约束中，有K 个函数为仿射函数（可简单理解为一阶多项式函数），那么只要其余的函数满足Slater 条件即可。

[2]我们这里只是陈述性说明强对偶成立的条件，即凸优化+Slater条件可以推出强对偶，但是强对偶不一定推出Slater条件，换句话说，可能是其他条件。

上面介绍了原问题和对偶问题的关系，但是实际还需要对参数进行求解，求解方法使用KKT条件进行，KKT 条件和强对偶关系是等价关系。KKT条件对最优解的条件为：

1. 可行域：

$$\begin{aligned} m_i(x^\star) &\leq 0 && (13)\\ n_j(x^\star) &= 0 && (14)\\ \lambda_i^\star &\geq 0 && (15)\end{aligned}$$

2. 互补松弛$\forall m_i, \lambda_i^\star m_i(x^\star) = 0$，对偶问题的最佳值为$d^\star$，原问题为$p^\star$：

$$\begin{aligned} d^\star &= \max_{\lambda,\eta} g(\lambda,\eta) = g(\lambda^\star,\eta^\star) && (16)\\ &= \min_x L(x,\lambda^\star,\eta^\star) && (17)\\ &\leq L(x^\star,\lambda^\star,\eta^\star) && (18)\\ &= f(x^\star) + \sum_{i=1}^{M} \lambda^\star m_i(x^\star) && (19)\\ &\leq f(x^\star) = p^\star && (20)\end{aligned}$$

为了满足相等，两个不等式必须成立，于是，对于第一个不等于号，需要有梯度为0 条件，对于第二个不等于号需要满足互补松弛条件。

3. 梯度为0：$\frac{\partial L(x,\lambda^\star,\eta^\star)}{\partial x}|_{x=x^\star} = 0$

2 硬分类的SVM

支撑向量机也是一种硬分类模型，在之前的感知机模型中，我们在线性模型的基础上叠加了符号函数，在几何直观上，可以看到，如果两类分的很开的话，那么其实会存在无穷多条线可以将两类分开。在SVM 中，我们引入最大化间隔这个概念，间隔指的是数据和直线的距离的最小值，因此最大化这个值反映了我们的模型倾向。

分割的超平面可以写为：

$$0 = w^T x + b \tag{21}$$

这里$w, x, b \in \mathbb{R}^n$，n是空间维度。那么最大化间隔（约束为分类任务的要求）：

$$\arg\max_{w,b}[\min_i \frac{|w^T x_i + b|}{||w||}] s.t.\ y^{(i)}(w^T x^{(i)} + b) > 0 \tag{22}$$

$$\Rightarrow \arg\max_{w,b}[\min_i \frac{y^{(i)}(w^T x^{(i)} + b)}{|||w|}] s.t.\ y^{(i)}(w^T x^{(i)} + b) > 0 \tag{23}$$

对于这个约束$y^i(w^T x^{(i)} + b) > 0$，不妨固定$\min y^{(i)}(w^T x^{(i)} + b) = 1$[3]，这是由于分开两类的超平面的系数经过比例放缩不会改变这个平面，这也相当于给超平面的系数作出了约束。化简后的式子可以表示为：

$$\arg\min_{w,b} \frac{1}{2} w^T w\ s.t. \min_i y^{(i)}(w^T x^{(i)} + b) = 1 \tag{24}$$

$$\Rightarrow \arg\min_{w,b} \frac{1}{2} w^T w\ \ s.t.\ y^{(i)}(w^T x^{(i)} + b) \geq 1, i = 1, 2, ..., N \tag{25}$$

[3]这里涉及到函数间隔和几何间隔的关系，我们再此不在讨论，简单起见，直接给出结论，但是你要知道，这种设定，是合理的，并且可以简化运算。

但是，如果样本数量或维度非常高，直接求解困难甚至不可解，于是需要对这个问题进一步处理。引入Lagrange函数：

$$L(w,b,\lambda)=\frac{1}{2}w^Tw+\sum_{i=1}^{N}\lambda_i(1-y^{(i)}(w^Tx^{(i)}+b)) \tag{26}$$

原问题就等价于：

$$\arg_{w,b}\min\max_{\lambda}L(w,b,\lambda_i)\ s.t.\ \lambda_i\geq 0 \tag{27}$$

交换最小和最大值的符号得到对偶问题：

$$\max_{\lambda_i}\min_{w,b}L(w,b,\lambda_i)\ s.t.\ \lambda_i\geq 0 \tag{28}$$

由于不等式约束是仿射函数，对偶问题和原问题等价：

- $b:\frac{\partial}{\partial b}L=0\Rightarrow\sum_{i=1}^{N}\lambda_iy^{(i)}=0$
- w :将b的表达式代入：

$$L(w,b,\lambda_i)=\frac{1}{2}w^Tw+\sum_{i=1}^{N}(1-y^{(i)}w^Tx^{(i)}-y^{(i)}b)=\frac{1}{2}w^Tw+\sum_{i=1}^{N}\lambda_i-\sum_{i=1}^{N}\lambda_iy^{(i)}w^Tx^{(i)} \tag{29}$$

  继续对拉格朗日函数对w求偏导并令其为0：

$$\frac{\partial}{\partial w}L(w,b,\lambda_i)=0\Rightarrow w=\sum_{i=1}^{N}\lambda_iy^{(i)}x^{(i)} \tag{30}$$

- 将上面两个参数代入：

$$L(w,b,\lambda_i)=-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_jy^{(i)}y^{(j)}x^{(i)^T}x^{(j)}+\sum_{i=1}^{N}\lambda_i \tag{31}$$

因此，对偶问题就是：

$$\max_{\lambda}-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\lambda_i\lambda_jy^{(i)}y^{(j)}x^{(i)^T}x^{(j)}+\sum_{i=1}^{N}\lambda_i,\ s.t.\ \lambda_i\geq 0 \tag{32}$$

总结来说，原问题和对偶问题满足强对偶关系的充要条件为其满足KKT条件：

$$\frac{\partial L}{\partial w}=0,\frac{\partial L}{\partial b}=0 \tag{33}$$

$$\lambda_k(1-y^{(k)}(w^Tx^{(k)}+b))=0\ \text{(slackness complementary)} \tag{34}$$

$$\lambda_i\geq 0 \tag{35}$$

$$1-y^{(i)}(w^Tx^{(i)}+b)\leq 0 \tag{36}$$

从KKT条件得到超平面的参数：

$$\hat{w}=\sum_{i=1}^{N}\lambda_iy^{(i)}x^{(i)} \tag{37}$$

$$\hat{b}=y^{(k)}-w^Tx^{(k)}=y^{(k)}-\sum_{i=1}^{N}\lambda_iy^{(i)}x^{(i)^T}x^{(k)},\exists k,1-y^{(k)}(w^Tx^{(k)}+b)=0 \tag{38}$$

于是这个超平面的参数w就是数据点的线性组合，最终的参数值就是部分满足$y^{(i)}(w^Tx^{(i)}+b)=1$ 向量的线性组合（互补松弛条件给出），这些向量也叫支撑向量。

3 软分类的SVM

Hard-margin 的SVM只对可分数据可解，如果不可分的情况，我们的基本想法是在损失函数中加入错误分类的可能性。错误分类的个数可以写成：

$$error = \sum_{i=1}^{N} \mathbb{I}\{y^{(i)}(w^T x^{(i)} + b) < 1\} \tag{39}$$

这个函数不连续，如果添加到目标函数中去，是不方便对目标函数求导的。退而求其次，使用分错的样本，离超平面的距离来代替错误样本的个数。可以将其改写为：

$$error = \sum_{i=1}^{N} \max\{0, 1 - y^{(i)}(w^T x^{(i)} + b)\} \tag{40}$$

求和符号中的式子右叫做Hinge Function。将这个错误加入Hard-margin 的SVM中，于是得到如下的目标函数：

$$\arg\min_{w,b} \frac{1}{2} w^T w + C \sum_{i=1}^{N} \max\{0, 1 - y^{(i)}(w^T x^{(i)} + b)\} s.t.\ y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi^i, i = 1, 2, ..., N \tag{41}$$

这个式子中，常数C可以看作允许的错误水平，同时上式为了进一步消除max符号，对数据集中的每一个观测，我们可以认为其大部分满足约束，但是其中部分违反约束，因此这部分约束变成$y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi^{(i)}$，其中$\xi^{(i)} = 1 - y^{(i)}(w^T x^{(i)} + b)$，进一步的化简：

$$\arg\min_{w,b} \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi^{(i)} s.t.\ y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi^i, \xi^{(i)} \geq 0, i = 1, 2, ..., N \tag{42}$$

至此我们把**软间隔**（soft-margin）的SVM 模型与目标函数都介绍清楚了。求解方法，一般情况下，Matlab 有优化二次型的程序包，后面我们介绍一种优化方法，能非常好的求解SVM的优化问题。

4 核方法

核方法可以应用在很多问题上，在分类问题中，对于严格不可分问题，我们引入一个特征转换函数将原来的不可分的数据集变为可分的数据集，然后再来应用已有的模型。往往将低维空间的数据集变为高维空间的数据集后，数据会变得可分（数据变得更为稀疏）。应用在SVM 中时，观察上面的SVM对偶问题：

$$\max_{\lambda} -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j y^{(i)} y^{(j)} x^{(i)^T} x^{(j)} + \sum_{i=1}^{N} \lambda_i,\ s.t. \lambda_i \geq 0 \tag{43}$$

在求解的时候需要求得内积，于是不可分数据在通过特征变换后，需要求得变换后的内积。我们常常很难求得变换函数的内积。于是直接引入内积的变换函数：

$$\forall x, x' \in \mathcal{X}, \exists \phi \in \mathcal{H} : x \to z \ \ s.t.\ k(x, x') = \phi^T(x)\phi(x') = z^T z' \tag{44}$$

称$k(x, x')$为一个正定核函数，其中$\mathcal{H}$ 是Hilbert空间（完备的线性内积空间），如果去掉内积这个条件我们简单地称为核函数。我们下面举一个例子，高斯分布的指数部分，是一个核函数。即：

$$k(x, x') = \exp(-\frac{(x - x')^2}{2\sigma^2}) \tag{45}$$

证明：

$$\begin{aligned}
\exp(-\frac{(x-x')^2}{2\sigma^2}) &= \exp(-\frac{x^2}{2\sigma^2})\exp(\frac{xx'}{\sigma^2})\exp(-\frac{x'^2}{2\sigma^2}) && (46)\\
&= \exp(-\frac{x^2}{2\sigma^2})\sum_{n=0}^{+\infty}\frac{x^n x'^n}{\sigma^{2n}n!}\exp(-\frac{x'^2}{2\sigma^2}) && (47)\\
&= \sum_{n=0}^{+\infty}\frac{1}{n!\sigma^{2n}}\exp(-\frac{x^2}{2\sigma^2})x^n\exp(-\frac{x'^2}{2\sigma^2})x'^n && (48)\\
&= \sum_{n=0}^{+\infty}\Big[(\sqrt{\frac{1}{n!\sigma^{2n}}}\exp(-\frac{x^2}{2\sigma^2})x^n)(\sqrt{\frac{1}{n!\sigma^{2n}}}\exp(-\frac{x'^2}{2\sigma^2})x'^n)\Big] && (49)
\end{aligned}$$

第二个等号，使用了e指数函数的泰勒展开：

$$e^x = \sum_{n=0}^{+\infty}\frac{x^n}{n!} \tag{50}$$

我们定义映射函数为：

$$\phi(x) = \sqrt{\frac{1}{n!\sigma^{2n}}}\exp(-\frac{x^2}{2\sigma^2})x^n \tag{51}$$

即高斯核函数将输入空间映射到了无穷多维空间。正定核函数有下面的等价定义：

如果核函数满足：

1. 对称性。
2. 正定性。

那么这个核函数是正定核函数。接下来我们对上面的结论进行一个证明，我们分成两部走：

1. 对称性$\Leftrightarrow k(x,z)=k(z,x)$，显然满足内积的定义。
2. 正定性$\Leftrightarrow \forall\ N, x^{(1)}, x^{(2)}, ..., x^{(N)} \in \mathcal{X}$，对应的Gram Matrix $K=[k(x^{(i)},x^{(j)})]$ 是半正定的。

首先我们证明充分性，即当核函数为$k(x,z)=\phi^T(x)\phi(z)$能推出K 是半正定+ 对称性。

内积的定义就是对称的，所以直接来看正定性：

$$K = \begin{bmatrix} k(x^{(1)},x^{(1)}) & \cdots & k(x^{(1)},x^{(N)}) \\ \vdots & \ddots & \vdots \\ k(x^{(N)},x^{(1)}) & \cdots & k(x^{(N)},x^{(N)}) \end{bmatrix} \tag{52}$$

任意取$\alpha \in \mathbb{R}^N$ ，即需要证明$\alpha^T K\alpha \geq 0$：

$$\alpha^T K\alpha = \sum_{ij}\alpha_i\alpha_j K_{ij} = \sum_{ij}\alpha_i\phi^T(x^{(i)})\phi(x^{(j)})\alpha_j = \sum_i \alpha_i\phi^T(x^{(i)})\sum_j\alpha_j\phi(x^{(j)}) \tag{53}$$

我们可以看到，等式最后一项是两项相同的乘积，即平方，所以总是大于等于0，于是正定性得证。

其次我们来看必要性：对于K进行分解，对于对称矩阵$K=V\Lambda V^T$，那么令$\phi(x^{(i)})=\sqrt{\lambda_i}V^{(i)}$，其中$V^{(i)}$是特征向量，于是就构造了$k(x,z)=\sqrt{\lambda_i\lambda_j}V^{(i)^T}V^{(j)}$。

5 核方法与梯度下降

这一节我们看核方法的具体应用。我们已经在上一节介绍了核方法，本质是把一个低维线性不可分数据，通过核函数映射到高维线性可分的空间中去。上面我们已经举了一个例子，核函数是高斯核，

还有一些常见的核函数，我们在这通过另一个核函数，多项式核函数。在引入了核函数之后，我们区分对一个数据特征的特征，在没有经过核函数映射前，把原始数据具有的分量，称为**属性**（attributes），通过核函数映射后的向量分量，称为**特征**（feature），我们把映射后的向量称为**特征图谱**（feature map）。

5.1 特征图谱下的LMS

这里我们推导使用特征图谱的梯度下降算法。首先我们复习以下梯度下降的更新公式：

$$\theta := \theta + \alpha \sum_{i=1}^{n} (y^{(i)} - h_\theta(x^{(i)}))x^{(i)} \tag{54}$$

$$:= \theta + \alpha \sum_{i=1}^{n} (y^{(i)} - \theta^T x^{(i)})x^{(i)} \tag{55}$$

我们定义一个映射$\phi : \mathbb{R}^d \to \mathbb{R}^p$是一个特征图谱。现在我们的目标是拟合函数$\theta^T\phi(x)$，这里需要注意，参数向量$\theta$的维度从$d$变成了$p$。我们能替代原先的更新公式里面的$x^{(i)}$，得到新的更新规则：

$$\theta := \theta + \alpha \sum_{i=1}^{n} (y^{(i)} - \theta^T\phi(x^{(i)}))\phi(x^{(i)}) \tag{56}$$

以上可以看出，是批量梯度下降对应的更新公式，把求和号去掉，就是对应的随机梯度下降：

$$\theta := \theta + \alpha (y^{(i)} - \theta^T\phi(x^{(i)}))\phi(x^{(i)}) \tag{57}$$

5.2 核函数下的LMS

无论是批量梯度下降还是随机梯度下降，特征图谱$\phi(x)$都是一个高维向量。这对于计算机来说，无疑是一个灾难。但是因为核函数的引入，我们接下来会看到，会大大简化对数据的存储与运算。我们一般把使用核函数的方法，称为**核技巧**（kernel trick）。无论什么算法更新参数，我们都有：

$$\theta = \sum_{i=1}^{n} \beta_i \phi(x^{(i)}) \tag{58}$$

参数$\beta_1, ..., \beta_n \in \mathbb{R}$。我们还应该知道，下一次更新后的参数，仍然是所有样本特征图谱对应的线性组合，因为：

$$\theta = \theta + \alpha \sum_{i=1}^{n} (y^{(i)} - \theta^T\phi(x^{(i)}))\phi(x^{(i)}) \tag{59}$$

$$= \sum_{i=1}^{n} \beta_i \phi(x^{(i)}) + \alpha \sum_{i=1}^{n} (y^{(i)} - \theta^T\phi(x^{(i)}))\phi(x^{(i)}) \tag{60}$$

$$= \sum_{i=1}^{n} (\beta_i + \alpha(y^{(i)} - \theta^T\phi(x^{(i)})))\phi(x^{(i)}) \tag{61}$$

最后一个等式可以看出，新的β_i就应该是括号里面的公式。换句话说，我们可以直接更新β，跳过更新θ的方法，去更新模型。β_i的更新规则为：

$$\beta_i := \beta_i + \alpha(y^{(i)} - \theta^T\phi(x^{(i)})) \tag{62}$$

上面的更新过程，依旧包含θ，我们可以进一步进行替换，利用$\theta = \sum_{j=1}^{n} \beta_j \phi(x^{(j)})$，我们得到如下的更新规则：

$$\forall i \in \{1, ..., n\}, \beta_i := \beta_i + \alpha\Big(y^{(i)} - \sum_{j=1}^{n} \beta_j \phi^T(x^{(j)})\phi(x^{(i)})\Big) \tag{63}$$

对于两个向量的内积$\phi^T(x^{(j)})\phi(x^{(i)})$，我们一般写成$\langle\phi(x^{(j)}),\phi(x^{(i)})\rangle$。这样，我们就成功的把对参数$\theta$的更新，转化为对参数$\beta$的更新。此时，虽然我们对梯度下降已经进行了化简，但是在计算内积的地方，还是不可避免。对内积运算进行观察，我们看出：

1. 我们每次更新都必须要重复计算内积。
2. 内积的计算，是可以进行化简的。

所谓化简，我们通过一个三次方的多项式特征图谱，做一个示范：

$$\begin{aligned}\langle\phi(x),\phi(z)\rangle &= 1+\sum_{i=1}^{d}x_iz_i+\sum_{i,j\in\{1,...,d\}}x_i,x_jz_iz_j+\sum_{i,j,k\in\{1,...,d\}}x_ix_jx_kz_iz_jz_k &(64)\\ &= 1+\sum_{i=1}^{d}x_iz_i+\Big(\sum_{i=1}^{d}x_iz_i\Big)^2+\Big(\sum_{i=1}^{d}x_iz_i\Big)^3 &(65)\\ &= 1+\langle x,z\rangle+\langle x,z\rangle^2+\langle x,z\rangle^3 &(66)\end{aligned}$$

因此，计算$\langle\phi(x),\phi(z)\rangle$，我们可以先计算$\langle x,z\rangle$，然后对计算结果，直接计算多次方就行了。

通过以上的例子，我们可以看出核函数的定义就是特征图谱的内积，而特征图谱与属性之间的映射，可以说是千变万化，总之，我们说核函数定义为：

$$K(x,z)\equiv\langle\phi(x),\phi(z)\rangle \tag{67}$$

最后我们总结以下核技巧在梯度下降当中的应用：

1. 计算所有的核函数对应的元素：$K(x^{(i)},x^{(j)})\equiv\langle\phi(x^{(i)}),\phi(x^{(j)})\rangle$，这里不同的核函数，对应着不同的映射。设置$\beta:=0$
2. 循环：
$$\forall i\in\{1,...,n\},\beta_i:=\beta_i+\alpha\Big(y^{(i)}-\sum_{j=1}^{n}\beta_jK(x^{(i)},x^{(j)})\Big) \tag{68}$$
如果写成向量的形式，具有如下的表达式：
$$\beta:=\beta+\alpha(\overrightarrow{y}-K\beta) \tag{69}$$

上面的算法，对应的时间复杂度为$O(n^2)$。最终我们需要去对新的数据进行预测，我们同样可以使用核函数：

$$\beta^T\phi(x)=\sum_{i=1}^{n}\beta_i\phi^T(x^{(i)})\phi(x)=\sum_{i=1}^{n}\beta_iK(x^{(i)},x) \tag{70}$$

最后，我们依旧可以看到，我们所有的根基，都来自于核函数$K(\cdot,\cdot)$。

6 坐标下降法

这里考虑一个多参数优化问题：

$$\max_{\alpha}W(\alpha_1,\alpha_2,...,\alpha_n) \tag{71}$$

SVM模型的优化问题，可以认为是上述优化问题的一个具体实现。我们在之前，介绍了梯度下降和牛顿方法。这里我们给出一个新的算法，称为**坐标上升**（coordinate ascent）：

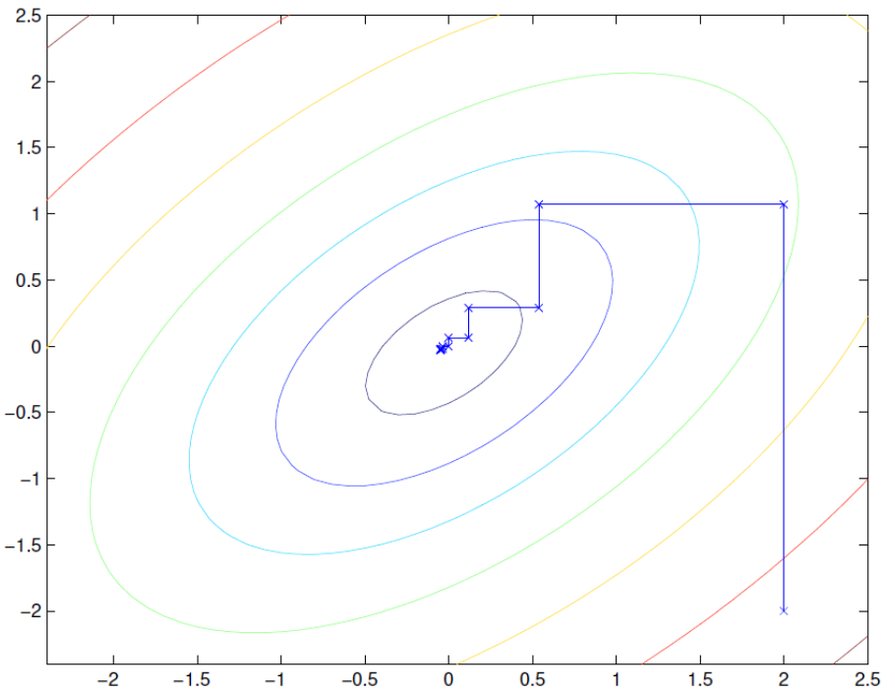

Figure 2: 在凸优化问题上，使用坐标上升方法，逐步达到目标函数的极值。

1. 迭代直到收敛：

$$\{ \tag{72}$$

$$\text{For i=1,2,...,n,}\{\alpha_i := \arg\max_{\hat{\alpha_i}} W(\alpha_1, ..., \alpha_{i-1}, \hat{\alpha_i}, \alpha_{i+1}, ..., \alpha_n)\} \tag{73}$$

$$\} \tag{74}$$

最内层的循环，我们固定除了α_i以外的参数，只对α_i这个参数进行更新，更新规则是取最大值。内部的循环，一般按照顺序更新，即：$\alpha_1, \alpha_2, ..., \alpha_n$。我们通过如下的图片展示坐标上升的形式： 上图我们把坐标初始化在（2，-2）点上，然后两个坐标交替更新，从图上的轨迹能看出来，和梯度下降方法是不同的。随着迭代次数增加，逐步找到最优解。

最后，SVM的更新算法，称为SMO算法，这里我们就不再进行介绍，作为课后阅读内容，可以查看吴恩达老师的讲义notes3。

7 小结

分类问题在很长一段时间都依赖SVM，对于严格可分的数据集，Hard-margin SVM选定一个超平面，保证所有数据到这个超平面的距离最大，对这个平面施加约束，固定$y^{(i)}(w^T x^{(i)} + b) = 1$，得到了一个凸优化问题并且所有的约束条件都是仿射函数，于是满足Slater条件，将这个问题变换成为对偶的问题，可以得到等价的解，并求出约束参数。

当允许一点错误的时候，可以在Hard-margin SVM中加入错误项。用Hinge Function表示错误项的大小。

对于完全不可分的问题，我们采用特征转换的方式，在SVM中，我们引入正定核函数来直接对内积进行变换，只要这个变换满足对称性和正定性，那么就可以用做核函数。

References