支持向量机

@author Jimmy

2016年3月4日

摘要

支持向量机（Support Vector Machine，SVM）是一种二分类模型（也可以扩展到多分类），它尝试在输入空间中找到一个超平面（hyperplane）来划分两类数据，并使该超平面到支持向量的间隔最大。最终，分类问题被转化为凸二次规划问题来求解。

1 线性分类

在逻辑回归中，我们对输入x被预测为正类的概率$p(y=1 \mid x;\theta)$建立模型：$h_\theta(x)=g(\theta^T x)$。当$h_\theta(x)\geq 0.5$时，我们预测输入$x$为正例；当$h_\theta(x)<0.5$时，我们预测输入$x$为负例。而且，当$\theta^T x \gg 0$时，输入$x$被预测为正例的可信度极大；当$\theta^T x \ll 0$时，输入$x$被预测为负例的可信度极大。接下来，让我们对逻辑回归做一些改造：

1. 令目标变量$y\in\{-1,1\}$，-1表示负例、1表示正例。
2. 令$\theta^T x=\omega^T x+b$，这只是形式上的变化，没有本质区别。
3. 简化映射层函数$g(z)$，令$g(z)=1 \text{ if } z\geq 0, \text{and } g(z)<0 \text{ otherwise}$。

至此我们可以得到新的分类器$h_{\omega,b}(x)=g(\omega^T x+b)$。

1.1 函数间隔及几何间隔

给定训练集$S=\left\{\left(x^{(i)},y^{(i)}\right);i=1,\cdots,m\right\}$，我们定义关于$(\omega,b)$的函数间隔（functional margin）为

$$\hat{\gamma}^{(i)}=y^{(i)}(\omega^T x^{(i)}+b)$$

显然，当$y^{(i)}=1$且$\omega^T x+b$取正数或者$y^{(i)}=-1$且$\omega^T x+b$取负数时，函数间隔是一个正数。类似于逻辑回归，当$|\omega^T x+b|$越大模型预测结果的可信度越高，而在分类错误时会函数间隔是一个负数，因此间隔函数取一个大正数代表模型比较好。

定义超平面$\omega^T x+b=0$，它将输入空间（在引入核函数之后，这里应该是特征空间）一分为二，是正负例的分界面。当参数ω,b扩大成$2\omega,2b$时，我们的模型并没有改变（超平面方程没有改变），但这时的函数间隔扩大成原来的2倍，因此函数间隔并不能定量的表达模型的分类能力。为解决这个问题，我们需要引入几何间隔（geometric margin）。

对于样本$\left(x^{(i)},y^{(i)}\right)$，令输入$x^{(i)}$到超平面的距离为$\gamma^{(i)}$，由超平面的法向量为$\omega$可知点$x^{(i)}-\gamma^{(i)}\omega y^{(i)}$落在超平面上，那么有

$$\omega^T\left(x^{(i)}-\gamma^{(i)}\omega y^{(i)}\right)+b=0$$
$$\Rightarrow\gamma^{(i)}=y^{(i)}\left(\left(\frac{\omega}{\|\omega\|}\right)^T x^{(i)}+\frac{b}{\|\omega\|}\right)$$

这里$\gamma^{(i)}$就是几何间隔，它衡量的是输入$x^{(i)}$到超平面的距离，而且它的取值不受参数ω, b同步缩放的影响，也就是说，一旦超平面确定，输入$x^{(i)}$的几何间隔也就确定。

1.2 最优间隔分类器

定义在训练集S上最小的几何间隔

$$\gamma=\min_{i=1,2,\cdots,m}\gamma^{(i)}$$

这个间隔代表训练集S中所有输入到超平面的最近距离。根据之前讨论的内容，我们希望这个间隔越大越好，于是我们可以得到一个目标函数

$$\begin{aligned}\max_{\gamma,\omega,b}\quad & \gamma\\ \text{s.t.}\quad & y^{(i)}\left(\omega^T x^{(i)}+b\right)\geq\gamma, i=1,\cdots,m\\ & \|\omega\|=1.\end{aligned}$$

当$\|\omega\|=1$时函数间隔等于几何间隔，但这时的优化问题是一个非凸（non-convex）优化问题，我们无法使用现成的商业二次规划（quadratic programming, QP）软件来解决它。为此我们改造我们的目标函数为

$$\begin{aligned}\max_{\gamma,\omega,b}\quad & \frac{\hat{\gamma}}{\|\omega\|}\\ \text{s.t.}\quad & y^{(i)}\left(\omega^T x^{(i)}+b\right)\geq\hat{\gamma}, i=1,\cdots,m\end{aligned}$$

其中$\hat{\gamma}$表示在训练集S上最小的函数间隔，很遗憾，这仍然是一个非凸优化问题。我们在前面提到同步缩放ω, b不会影响模型，于是我们可以引入一个很有用的限制条件，即令最小函数间隔$\hat{\gamma}=1$（同步缩放ω, b总能够做到这一点）。此时的目标是最大化$1/\|\omega\|$，等价于最小化$\|\omega\|^2$。我们继续改造我们的目标函数为

$$\begin{aligned}\min_{\gamma,\omega,b}\quad & \frac{1}{2}\|\omega\|^2\\ \text{s.t.}\quad & y^{(i)}\left(\omega^T x^{(i)}+b\right)\geq 1, i=1,\cdots,m\end{aligned}$$

这样就得到了我们的最优间隔分类器（optimal margin classifier）。

1.3 拉格朗日对偶性

在本节里我们将讨论拉格朗日对偶性（lagrange duality）。由于最优间隔分类器结构特殊，我们可以利用拉格朗日对偶性得到其对偶形式，然后再解出分类器（比二次规划软件解出的结果一般来得更好）。这样做的好处在于：一者对偶问题更容易求解；二者可以自然的引入核函数，进而推广到非线性分类问题。

1.3.1 拉格朗日因子法

首先让我们考虑一个条件受限优化问题，它的一般形式为

$$\begin{aligned} \min_{\omega} \quad & f(\omega) \\ \text{s.t.} \quad & h_i(\omega) = 0, i = 1, \cdots, l. \omega \end{aligned}$$

解决这类问题我们一般采用拉格朗日因子法，构造拉格朗日函数（lagrangian）得

$$\mathcal{L}(\omega, \beta) = f(\omega) + \sum_{i=1}^{l} \beta_i h_i(\omega)$$

其中β_i称为拉格朗日乘数。令$\mathcal{L}$对所有参数的偏导数等于0，得方程组

$$\frac{\partial L}{\partial \omega_i} = 0; \frac{\partial L}{\partial \beta_i} = 0$$

可解出ω, β。（注：以上方程组成立处即$\mathcal{L}$的极值点，而$\mathcal{L}$是一个凸函数，这意味着其极值点也是最值点。）

1.3.2 原始问题和对偶问题

下面让我们来讨论原始优化问题（primal optimization problem），它的一般形式为

$$\begin{aligned} \min_{\omega} \quad & f(\omega) \\ \text{s.t.} \quad & g_i(\omega) \leq 0, i = 1, \cdots, k \\ & h_i(\omega) = 0, i = 1, \cdots, l \end{aligned}$$

同样的，我们构造拉格朗日函数得

$$\mathcal{L}(\omega, \alpha, \beta) = f(w) + \sum_{i=1}^{k} \alpha_i g_i(\omega) + \sum_{i=1}^{l} \beta_i h_i(\omega)$$

记$\theta_{\mathcal{P}}(\omega) = \max_{\alpha, \beta; \alpha_i \geq 0} \mathcal{L}(\omega, \alpha, \beta)$，那么显然有

$$\theta_{\mathcal{P}}(\omega) = \begin{cases} f(w) & \text{if } \omega \text{ satisfies primal constraints} \\ \infty & \text{otherwise} \end{cases}$$

也就是说，只有当所有的限制条件都满足时，$\theta_{\mathcal{P}}(\omega)$等于我们的优化目标$f(\omega)$。于是我们的优化问题变成了

$$\min_{\omega} \theta_{\mathcal{P}}(\omega) = \min_{w} \max_{\alpha, \beta; \alpha_i \geq 0} \mathcal{L}(\omega, \alpha, \beta)$$

记原始问题的值$p^* = min_{\omega} \theta_{\mathcal{P}}(\omega)$。在对偶优化问题（dual optimization problem）中，我们要考虑的是

$$\max_{\alpha,\beta;\alpha_i\geq 0}\ \theta_D(\alpha,\beta)=\max_{\alpha,\beta;\alpha_i\geq 0}\ \min_{\omega}\mathcal{L}(\omega,\alpha,\beta)$$

只是将原始问题中maximize和minimize的位置交换了一下。记对偶问题的值$d^*=\max_{\alpha,\beta;\alpha_i\geq 0}\theta_D(\alpha,\beta)$，已证明有$d^*\leq p^*$，然而当优化问题满足KKT条件（这里不展开叙述）时等号成立。

1.4 解最优间隔分类器

在1.2节中我们得到了最优间隔分类器，不妨改变一下限制条件的形式为

$$g_i(\omega)=-y^{(i)}\left(\omega^T x^{(i)}+b\right)+1\leq 0$$

这样我们就得到了一个满足KKT条件的原始问题，构造拉格朗日函数得

$$\mathcal{L}(\omega,b,\alpha)=\frac{1}{2}\|\omega\|^2-\sum_{i=1}^{m}\alpha_i\left[y^{(i)}\left(\omega^T x^{(i)}+b\right)-1\right]$$

在原始问题中我们的目标是$\min_{\omega,b}\max_{\alpha_i\geq 0}\mathcal{L}(\omega,b,\alpha)$，转换成对偶问题后我们的新目标为$\max_{\alpha_i\geq 0}\min_{\omega,b}\mathcal{L}(\omega,b,\alpha)$。解这个优化问题，我们首先要求$\mathcal{L}$关于$\omega,b$的极小值，再求关于$\alpha$的极大值。第一步求极小，令$\mathcal{L}$对$\omega,b$的偏导等于0得

$$\omega=\sum_{i=1}^{m}\alpha_i y^{(i)}x^{(i)}\qquad(1)$$

$$\sum_{i=1}^{m}\alpha_i y^{(i)}=0$$

将这两个式子带入$\mathcal{L}$中得

$$\mathcal{L}(\omega,b,\alpha)=\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{m}y^{(i)}y^{(j)}\alpha_i\alpha_j\left(x^{(i)}\right)^T x^{(j)}$$

于是我们的目标又变成了

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

下面我们要用序列最小优化（sequential minimal optimization）算法求解α，这个算法留待3.2节再讨论。求出α后带入式(1)可得ω。令ω的取值为ω^*，那么有

$$b=-\frac{\max_{i;y^{(i)}=-1}\omega^{*T}x^{(i)}+\min_{i;y^{(i)}=1}\omega^{*T}x^{(i)}}{2}\qquad(2)$$

对此式的直观理解是，找到正例中$w^{*T}x$最小的一个值和负例中$w^{*T}x$最大的一个值，取它们平均值的负数。（注：在限制条件$g_i(\omega) \leq 0$下，正例$w^{*T}x$最小意味着该输入的函数间隔等于1，负例$w^{*T}x$最大也会得到相同的结论，这些点被称为支持向量。我们还可以验算得到，在式(2)成立时，我们将支持向量带入$y\left(w^{*T}x + b^*\right)$得到相等的值，也就是说$b$的取值依据是找到一个到支持向量距离相等的超平面。）此时，我们的分类函数为

$$\begin{aligned}\omega^T x + b &= \left(\sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}\right)^T x + b \\ &= \sum_{i=1}^{m} \alpha_i y^{(i)} \left\langle x^{(i)}, x \right\rangle + b \qquad (3)\end{aligned}$$

当$\omega^T x + b \geq 0$时预测新输入为正例。（注：事实上，只有支持向量对应的参数α不为0，因此我们对新输入作出预测只需要用到支持向量。）

2 非线性分类

在之前的讨论中，我们已经构造出了一个可以对输入空间进行线性分类的模型。然而当输入空间线性不可分时，我们就需要把输入空间映射到维度更高的空间来寻找线性划分的可能。

2.1 核（kernels）

定义特征映射（feature mapping）$\phi : \mathcal{X} \mapsto \mathcal{F}$表示输入空间到某特征空间的映射，升级我们的分类函数为

$$\omega^T x + b = \sum_{i=1}^{m} \alpha_i y^{(i)} \left\langle \phi\left(x^{(i)}\right), \phi(x) \right\rangle + b$$

在特征空间里计算内积$\left\langle \phi\left(x^{(i)}\right), \phi(x) \right\rangle$是一项成本极高的运算（因为输入空间映射到特征空间，维数的增长是爆炸性的），如果可以在输入空间中直接计算出内积$\left\langle \phi\left(x^{(i)}\right), \phi(x) \right\rangle$就好了。怀揣着这样一个美好的目的，我们提出了核函数K，它满足$K(x,z) = \langle \phi(x), \phi(z) \rangle$。举一个关于核函数的例子：假设$x, z \in \mathcal{R}^n$，并且

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

可以发现$K(x,z) = \phi(x)^T \phi(z)$，其中特征映射$\phi$（假设n=3）为

$$\phi(x) = [x_1 x_1, x_1 x_2, x_1 x_3, x_2 x_1, x_2 x_2, x_2 x_3, x_3 x_1, x_3 x_2, x_3 x_3]^T$$

这时，计算$\phi(x)$的时间复杂度为$O(n^2)$，而计算核函数$K(x,z)$的时间复杂度为$O(n)$，通过这

个例子可以看出引入核函数的必要性。凑巧的是，在我们的SVM里需要计算的地方输入总是以内积的形式出现（对偶问题的优化目标和分类函数），这样一来虽然我们将SVM扩展到了非线性分类，但我们完全避免了在高维空间里进行运算，而结果却是等价的！（注：由于这里的例子比较简单，所以构造核函数比较容易，如果对应任意一个映射，想要构造对应核函数就很困难了。）

2.2 几个核函数

这里简要的介绍几种核函数

- 多项式核$K(x_1, x_2) = (\langle x_1, x_2 \rangle + R)^d$，刚刚举的例子显然是多项式核的一个特例（$R = 0, d = 2$）。该空间的维度是$C_{m+d}^d$，其中$m$是原始（输入）空间的维度。
- 高斯核$K(x_1, x_2) = \exp(-\|x_1 - x_2\|^2/2\sigma^2)$，它会将原始空间映射到无穷维空间。当$\sigma$很大时，高次特征上的权重衰减很快，所以相当于一个低维的子空间；当σ很小时，则可以将任意的空间线性划分——这并不是一件好事，因为随之而来的是严重的过拟合问题。通过调控参数σ，高斯核具有相当高的灵活性。
- 线性核$K(x_1, x_2) = \langle x_1, x_2 \rangle$,这个核存在的目的是为了使在原始空间上的线性分类和非线性分类问题形式上统一。

2.3 Mercer定理

Mercer定理的内容是

给定函数$K : \mathcal{R}^n \times \mathcal{R}^n \mapsto \mathcal{R}$，那么它是一个有效核函数的充分必要条件是：对应于输入$\left\{x^{(i)}, \cdots, x^{(m)}\right\}, (m < \infty)$，的核矩阵（kernel matrix）是对称半正定的。其中，对核矩阵K有

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

（注：后面的K表示核函数。）关于这个定理就不深入了。

3 扩展

3.1 规范软间隔（norm soft margin）SVM

在之前介绍的SVM中，当数据不是线性可分时，我们通过将原始空间映射到高维空间来寻找线性划分的可能，然而这种办法并非万能，因为造成线性不可分的原因可能仅仅是存在噪音。对那些偏离正常位置很远的点（噪音），我们称为outlier。事实上，我们的超平面是由极少量的支持向量决定，当支持向量中存在outlier时，它对超平面的影响很大。为了使SVM对outlier具有很好的容错性，我们需要更新优化函数为

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

在这里我们允许样本函数间隔小于1，但是当大量样本如此时，模型就没有意义了，因此我们

加上了惩罚项$C\epsilon_i$来惩罚函数间隔小于1的情况。其中，C称为松弛因子，用来调节$\|w\|^2$与惩罚项之间的关系，并保证多数样本的函数间隔不小于1。构造拉格朗日函数得

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

与之前的过程一样，我们求出$\mathcal{L}$关于ω, b偏导等于0的方程组，并将结果代回后得对偶形式优化问题为

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

这与引入松弛因子之前的对偶形式几乎一致，区别只在于α_i的限制条件变成了$0 \le \alpha_i \le C$。此外，根据KKT条件还能够得出拉格朗日乘数α取值的意义，即

$$\begin{aligned}\alpha_i = 0 &\Rightarrow y^{(i)}(\omega^T x^{(i)} + b) \ge 1 \\ \alpha_i = C &\Rightarrow y^{(i)}(\omega^T x^{(i)} + b) \le 1 \\ 0 < \alpha_i < C &\Rightarrow y^{(i)}(\omega^T x^{(i)} + b) = 1\end{aligned}$$

第一种情况表明α_i是正常分类，在边界内部，第二种情况表明α_i是支持变量，在边界上，第三种情况表明α_i在两条边界之间。

3.2 SMO算法

下面让我们继续我们未完成的事业：求原始问题转换成对偶形式后的最优问题。

3.2.1 坐标上升算法

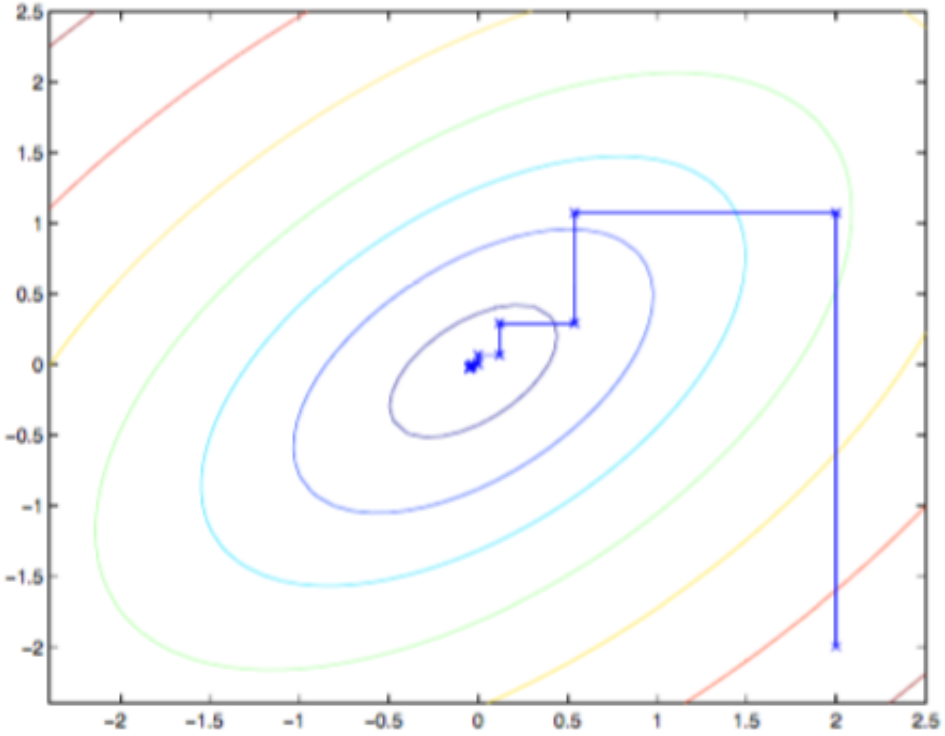
对偶形式的最优问题可以表述为

$$\max_{\alpha} W(\alpha_1, \cdots, \alpha_m)$$

这是一个关于参数α的最大值求解。我们在notes1和notes2中介绍了两种解这类问题的办法——梯度上升和牛顿法，在这里我们来看一种称为坐标上升（coordinate ascent）的新算法，它可以表述为

```
Loop until convergence {
    For i=1,...,m {
        固定除alpha(i)外的其他参数，令alpha(i)的取值使得W(alpha)最大
    }
}
```

（注：以上算法中alpha(i)代表α_i，下同。）这个算法和梯度上升有些许类似，不过梯度上升是向梯度方向优化，而坐标上升是轮流向每个坐标方向优化，下图给出坐标上升算法中参数更新的路径。



3.2.2 SMO

梯度上升方法并不能直接的用于求解对偶形式最优问题，这是因为在限制条件$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0$下，由于每一轮更新$\alpha$时要固定m-1个位置，这样的话最后一个位置的取值就已经确定，α也不会更新。例如，固定第2到m个位置，那么第一个位置的取值为

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

为了使算法在限制条件下适用，我们提出SMO算法，它可以表述为

```
Repeat util convergence {
    1. 选择一对参数alpha(i)和alpha(j)（选取方法采用启发式方法）
    2. 固定除alpha(i)和alpha(j)外的其他参数，确定W(alpha)极值条件下的alpha(i),
alpha(j)由alpha(i)表示
}
```

与坐标上升算法相比，SMO的改进之处在于每轮更新了α的两个位置。现假设更新α_1, α_2，那么根据限制条件有

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

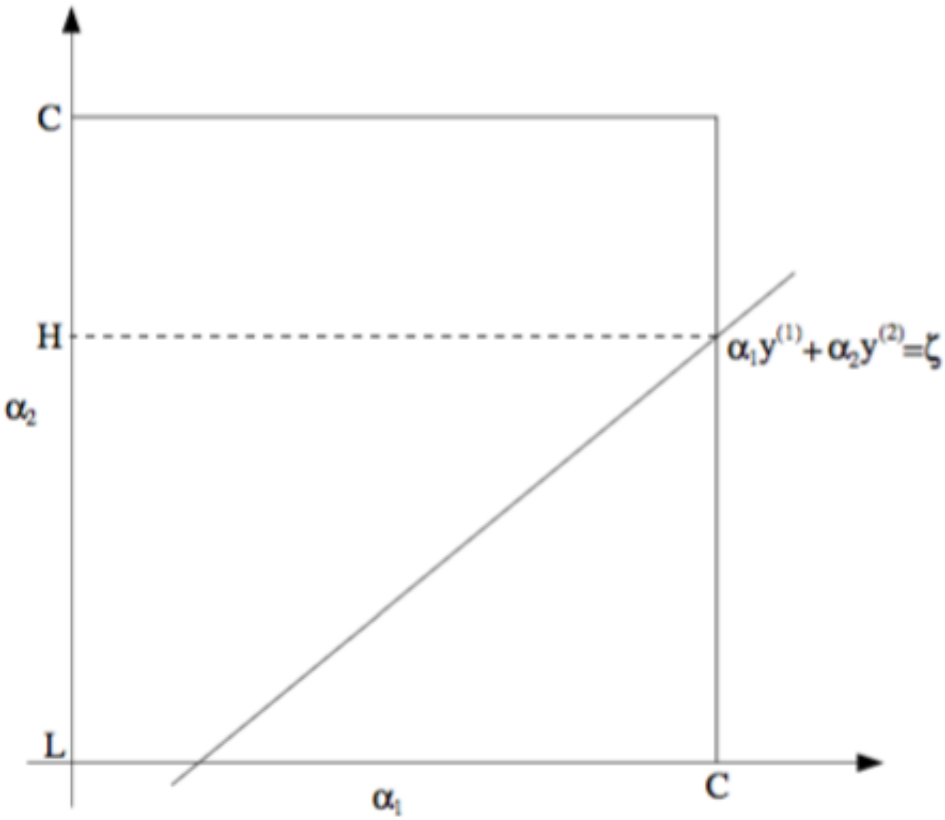
等式右边是一个常数，记为ζ，那么

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

进一步，$W(\alpha)$可以写成

$$W(\alpha) = W\left(\left(\zeta - \alpha_2 y^{(2)}\right) y^{(1)}, \alpha_2, \cdots, \alpha_m\right)$$

当$\alpha_3, \cdots, \alpha_m$是常数时，$W$是一个关于$\alpha_2$的二元一次方程。又根据限制条件，我们可以得到$\alpha_2$的取值范围落在如下图所示的区间$[L, H]$中。



剩下的就是在给定取值区间下求$W(\alpha_2)$最大值的问题了。