# 支持向量机

1. 支持向量机(`support vector machines`： `SVM`）是一种二分类模型。它是定义在特征空间上的、间隔最大的线性分类器。
   - 间隔最大使得支持向量机有别于感知机。

     如果数据集是线性可分的，那么感知机获得的模型可能有很多个，而支持向量机选择的是间隔最大的那一个。
   - 支持向量机还支持核技巧，从而使它成为实质上的非线性分类器。
2. 支持向量机支持处理线性可分数据集、非线性可分数据集。
   - 当训练数据线性可分时，通过硬间隔最大化，学习一个线性分类器，即线性可分支持向量机（也称作硬间隔支持向量机）。
   - 当训练数据近似线性可分时，通过软间隔最大化，学习一个线性分类器，即线性支持向量机（也称为软间隔支持向量机）。
   - 当训练数据不可分时，通过使用核技巧以及软间隔最大化，学习一个非线性分类器，即非线性支持向量机。
3. 当输入空间为欧氏空间或离散集合、特征空间为希尔伯特空间时，将输入向量从输入空间映射到特征空间，得到特征向量。

   支持向量机的学习是在特征空间进行的。
   - 线性可分支持向量机、线性支持向量机假设这两个空间的元素一一对应，并将输入空间中的输入映射为特征空间中的特征向量。
   - 非线性支持向量机利用一个从输入空间到特征空间的非线性映射将输入映射为特征向量。
     - 特征向量之间的内积就是核函数，使用核函数可以学习非线性支持向量机。
     - 非线性支持向量机等价于隐式的在高维的特征空间中学习线性支持向量机，这种方法称作核技巧。
4. 欧氏空间是有限维度的，希尔伯特空间为无穷维度的。
   - 欧式空间 $\subseteq$ 希尔伯特空间 $\subseteq$ 内积空间 $\subseteq$ 赋范空间。
     - 欧式空间，具有很多美好的性质。
     - 若不局限于有限维度，就来到了希尔伯特空间。

       从有限到无限是一个质变，很多美好的性质消失了，一些非常有悖常识的现象会出现。
     - 如果再进一步去掉完备性，就来到了内积空间。
     - 如果再进一步去掉"角度"的概念，就来到了赋范空间。此时还有“长度”和“距离”的概念。
   - 越抽象的空间具有的性质越少，在这样的空间中能得到的结论就越少
   - 如果发现了赋范空间中的某些性质，那么前面那些空间也都具有这个性质。

# 一、 线性可分支持向量机

1. 给定一个特征空间上的训练数据集 $\mathbb{D}=\{(\vec{\mathbf{x}}_1,\tilde{y}_1),(\vec{\mathbf{x}}_2,\tilde{y}_2),\cdots,(\vec{\mathbf{x}}_N,\tilde{y}_N)\}$，其中 $\vec{\mathbf{x}}_i\in\mathcal{X}=\mathbb{R}^n,\tilde{y}_i\in\mathcal{Y}=\{+1,-1\},i=1,2,\cdots,N$。

   $\vec{\mathbf{x}}_i$ 为第 $i$ 个特征向量，也称作实例； $\tilde{y}_i$ 为 $\vec{\mathbf{x}}_i$ 的类标记； $(\vec{\mathbf{x}}_i,\tilde{y}_i)$ 称作样本点。
   - 当 $\tilde{y}_i=+1$ 时，称 $\vec{\mathbf{x}}_i$ 为正例。

- 当 $\tilde{y}_i=-1$ 时，称 $\vec{\mathbf{x}}_i$ 为负例。

假设训练数据集是线性可分的，则学习的目标是在特征空间中找到一个分离超平面，能将实例分到不同的类。

分离超平面对应于方程 $\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}+b=0$， 它由法向量 $\vec{\mathbf{w}}$ 和截距 $b$ 决定，可以用 $(\vec{\mathbf{w}},b)$ 来表示。

2. 给定线性可分训练数据集，通过间隔最大化学习得到的分离超平面为：$\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}}+b^*=0$ ，

   相应的分类决策函数：$f(\vec{\mathbf{x}})=\text{sign}(\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}}+b^*)$ ，称之为线性可分支持向量机。

3. 当训练数据集线性可分时，存在无穷个分离超平面可以将两类数据正确分开。
   - 感知机利用误分类最小的策略，求出分离超平面。但是此时的解有无穷多个。
   - 线性可分支持向量机利用间隔最大化求得最优分离超平面，这样的解只有唯一的一个。

## 1.1 函数间隔

1. 可以将一个点距离分离超平面的远近来表示分类预测的可靠程度：
   - 一个点距离分离超平面越远，则该点的分类越可靠。
   - 一个点距离分离超平面越近，则该点的分类则不那么确信。
2. 在超平面 $\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}+b=0$ 确定的情况下:
   - $|\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b|$ 能够相对地表示点 $\vec{\mathbf{x}}_i$ 距离超平面的远近。
   - $\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b$ 的符号与类标记 $\tilde{y}_i$ 的符号是否一致能表示分类是否正确
     - $\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b>0$ 时，即 $\vec{\mathbf{x}}_i$ 位于超平面上方，将 $\vec{\mathbf{x}}_i$ 预测为正类。

       此时若 $\tilde{y}_i=+1$ 则分类正确；否则分类错误。
     - $\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b<0$ 时，即 $\vec{\mathbf{x}}_i$ 位于超平面下方，将 $\vec{\mathbf{x}}_i$ 预测为负类。

       此时若 $\tilde{y}_i=-1$ 则分类正确；否则分类错误。
3. 可以用 $y(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}+b)$ 来表示分类的正确性以及确信度，就是函数间隔的概念。
   - 符号决定了正确性。
   - 范数决定了确信度。
4. 对于给定的训练数据集 $\mathbb{D}$ 和超平面 $(\vec{\mathbf{w}},b)$
   - 定义超平面 $(\vec{\mathbf{w}},b)$ 关于样本点 $(\vec{\mathbf{x}}_i,\tilde{y}_i)$ 的函数间隔为： $\hat{\gamma}_i=\tilde{y}_i(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b)$
   - 定义超平面 $(\vec{\mathbf{w}},b)$ 关于训练集 $\mathbb{D}$ 的函数间隔为：超平面 $(\vec{\mathbf{w}},b)$ 关于 $\mathbb{D}$ 中所有样本点 $(\vec{\mathbf{x}}_i,\tilde{y}_i)$ 的函数间隔之最小值： $\hat{\gamma}=\min_{\mathbb{D}}\hat{\gamma}_i$ 。

## 1.2 几何间隔

1. 如果成比例的改变 $\vec{\mathbf{w}}$ 和 $b$ ，比如将它们改变为 $100\vec{\mathbf{w}}$和 $100b$，超平面 $100\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}+100b=0$ 还是原来的超平面，但是函数间隔却成为原来的100倍。

   因此需要对分离超平面施加某些约束，如归一化，令 $||\vec{\mathbf{w}}||_2=1$，使得函数间隔是确定的。此时的函数间隔成为几何间隔。

2. 对于给定的训练数据集 $\mathbb{D}$ 和超平面 $(\vec{\mathbf{w}},b)$
   - 定义超平面 $(\vec{\mathbf{w}},b)$ 关于样本点 $(\vec{\mathbf{x}}_i,\tilde{y}_i)$ 的几何间隔为：

$$\gamma_i=\tilde{y}_i(\frac{\vec{\mathbf{w}}}{||\vec{\mathbf{w}}||_2}\cdot\vec{\mathbf{x}}_i+\frac{b}{||\vec{\mathbf{w}}||_2})$$

- 定义超平面 $(\vec{\mathbf{w}},b)$ 关于训练集 $\mathbb{D}$ 的几何间隔为：超平面 $(\vec{\mathbf{w}},b)$ 关于 $\mathbb{D}$ 中所有样本点 $(\vec{\mathbf{x}}_i,\tilde{y}_i)$ 的几何间隔之最小值：$\gamma=\min_{\mathbb{D}}\gamma_i$ 。

3. 由定义可知函数间隔和几何间隔有下列的关系：

$$\gamma_i=\frac{\hat{\gamma}_i}{||\vec{\mathbf{w}}||_2},\quad \gamma=\frac{\hat{\gamma}}{||\vec{\mathbf{w}}||_2}$$

   - 当 $||\vec{\mathbf{w}}||_2=1$ 时，函数间隔和几何间隔相等。
   - 当超平面参数 $\vec{\mathbf{w}},b$ 等比例改变时：
     - 超平面并没有变化。
     - 函数间隔也按比例改变。
     - 几何间隔保持不变。

## 1.3 硬间隔最大化

1. 支持向量机学习基本思想：求解能够正确划分训练数据集并且几何间隔最大的分离超平面。几何间隔最大化又称作硬间隔最大化。

   对于线性可分的训练数据集而言，线性可分分离超平面有无穷多个（等价于感知机），但几何间隔最大的分离超平面是唯一的。

2. 几何间隔最大化的物理意义：不仅将正负实例点分开，而且对于最难分辨的实例点（距离超平面最近的那些点），也有足够大的确信度来将它们分开。

3. 求解几何间隔最大的分离超平面可以表示为约束的最优化问题：

$$\max_{\vec{\mathbf{w}},b}\gamma$$
$$s.t.\quad \tilde{y}_i(\frac{\vec{\mathbf{w}}}{||\vec{\mathbf{w}}||_2}\cdot\vec{\mathbf{x}}_i+\frac{b}{||\vec{\mathbf{w}}||_2})\geq\gamma,i=1,2,\cdots,N$$

   考虑几何间隔和函数间隔的关系，改写问题为：

$$\max_{\vec{\mathbf{w}},b}\frac{\hat{\gamma}}{||\vec{\mathbf{w}}||_2}$$
$$s.t.\quad \tilde{y}_i(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b)\geq\hat{\gamma},i=1,2,\cdots,N$$

4. 函数间隔 $\hat{\gamma}$ 的大小并不影响最优化问题的解。

   假设将 $\vec{\mathbf{w}},b$ 按比例的改变为 $\lambda\vec{\mathbf{w}},\lambda b$，此时函数间隔变成 $\lambda\hat{\gamma}$ （这是由于函数间隔的定义）：

   - 这一变化对求解最优化问题的不等式约束没有任何影响。
   - 这一变化对最优化目标函数也没有影响。

   因此取 $\hat{\gamma}=1$，则最优化问题改写为：

$$\max_{\vec{\mathbf{w}},b}\frac{1}{||\vec{\mathbf{w}}||_2}$$
$$s.t.\quad \tilde{y}_i(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b)\geq1,i=1,2,\cdots,N$$

5. 注意到 $\max\frac{1}{||\vec{\mathbf{w}}||_2}$和$\min\frac{1}{2}||\vec{\mathbf{w}}||_2^2$ 是等价的，于是最优化问题改写为：

$$\min_{\vec{\mathbf{w}},b}\frac{1}{2}||\vec{\mathbf{w}}||_2^2$$
$$s.t.\quad \tilde{y}_i(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b)-1\geq0,i=1,2,\cdots,N$$

这是一个凸二次规划问题。

6. 凸优化问题 ， 指约束最优化问题：

$$\min_{\vec{\mathbf{w}}} f(\vec{\mathbf{w}})$$
$$s.t. \quad g_j(\vec{\mathbf{w}}) \leq 0, j = 1, 2, \cdots, J$$
$$h_k(\vec{\mathbf{w}}) = 0, k = 1, 2, \cdots, K$$

其中：

- 目标函数 $f(\vec{\mathbf{w}})$ 和约束函数 $g_j(\vec{\mathbf{w}})$ 都是 $\mathbb{R}^n$ 上的连续可微的凸函数。
- 约束函数 $h_k(\vec{\mathbf{w}})$ 是 $\mathbb{R}^n$ 上的仿射函数。

> $h(\vec{\mathbf{x}})$ 称为仿射函数，如果它满足 $h(\vec{\mathbf{x}}) = \vec{\mathbf{a}} \cdot \vec{\mathbf{x}} + b, \quad \vec{\mathbf{a}} \in \mathbb{R}^n, b \in \mathbb{R}, x \in \mathbb{R}^n$

当目标函数 $f(\vec{\mathbf{w}})$ 是二次函数且约束函数 $g_j(\vec{\mathbf{w}})$ 是仿射函数时，上述凸最优化问题成为凸二次规划问题。

7. 线性可分支持向量机原始算法：
   - 输入：线性可分训练数据集 $\mathbb{D} = \{(\vec{\mathbf{x}}_1, \tilde{y}_1), (\vec{\mathbf{x}}_2, \tilde{y}_2), \cdots, (\vec{\mathbf{x}}_N, \tilde{y}_N)\}$ ， 其中 $\vec{\mathbf{x}}_i \in \mathcal{X} = \mathbb{R}^n, \tilde{y}_i \in \mathcal{Y} = \{+1, -1\}, i = 1, 2, \cdots, N$
   - 输出：
     - 最大几何间隔的分离超平面
     - 分类决策函数
   - 算法步骤：
     - 构造并且求解约束最优化问题：

$$\min_{\vec{\mathbf{w}}, b} \frac{1}{2}||\vec{\mathbf{w}}||_2^2$$
$$s.t. \quad \tilde{y}_i(\vec{\mathbf{w}} \cdot \vec{\mathbf{x}}_i + b) - 1 \geq 0, i = 1, 2, \cdots, N$$

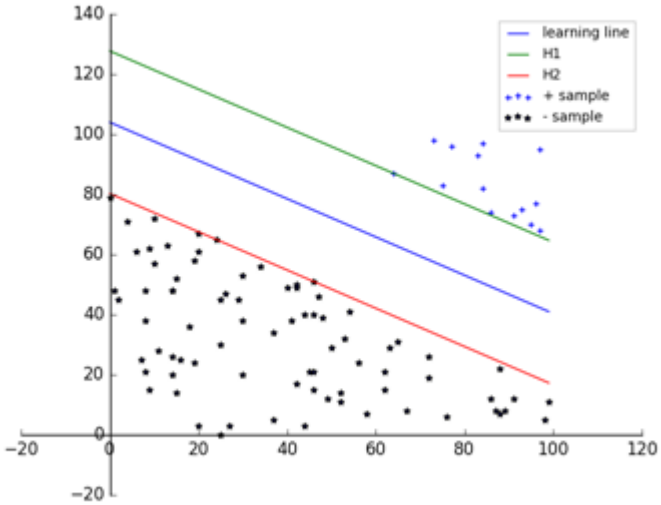
       求得最优解 $\vec{\mathbf{w}}^*, b^*$
     - 由此得到分离超平面：$\vec{\mathbf{w}}^* \cdot \vec{\mathbf{x}} + b^* = 0$ ，以及分类决策函数： $f(\vec{\mathbf{x}}) = \text{sign}(\vec{\mathbf{w}}^* \cdot \vec{\mathbf{x}} + b^*)$ 。
8. 可以证明：若训练数据集 $\mathbb{D}$ 线性可分，则可将训练数据集中的样本点完全正确分开的最大间隔分离超平面存在且唯一。

## 1.4 支持向量

1. 在训练数据集线性可分的情况下，训练数据集的样本点中与分离超平面距离最近的样本点的实例称为支持向量。

   支持向量是使得约束条件等号成立的点，即 $\tilde{y}_i(\vec{\mathbf{w}} \cdot \vec{\mathbf{x}}_i + b) - 1 = 0$ ：
   - 对于正实例点，支持向量位于超平面 $H_1$ ：$\vec{\mathbf{w}} \cdot \vec{\mathbf{x}} + b = 1$
   - 对于负实例点，支持向量位于超平面 $H_2$ ：$\vec{\mathbf{w}} \cdot \vec{\mathbf{x}} + b = -1$
2. 超平面 $H_1$ 、 $H_2$ 称为间隔边界， 它们和分离超平面 $\vec{\mathbf{w}} \cdot \vec{\mathbf{x}} + b = 0$ 平行，且没有任何实例点落在$H_1$ 、 $H_2$ 之间。

   在 $H_1$ 、 $H_2$ 之间形成一条长带，分离超平面位于长带的中央。长带的宽度称为 $H_1$ 、 $H_2$ 之间的距离，也即间隔，间隔大小为 $\frac{2}{||\vec{\mathbf{w}}||_2}$ 。




3. 在决定分离超平面时，只有支持向量起作用，其他的实例点并不起作用。
    - 如果移动支持向量，将改变所求的解。
    - 如果在间隔边界以外移动其他实例点，甚至去掉这些点，则解是不变的。
4. 由于支持向量在确定分离超平面中起着决定性作用，所以将这种分离模型称为支持向量机。
5. 支持向量的个数一般很少，所以支持向量机由很少的、重要的训练样本确定。

## 1.5 对偶算法

1. 将线性可分支持向量机的最优化问题作为原始最优化问题，应用拉格朗日对偶性，通过求解对偶问题得到原始问题的最优解。这就是线性可分支持向量机的对偶算法。
2. 对偶算法的优点：
    - 对偶问题往往更容易求解。
    - 引入了核函数，进而推广到非线性分类问题。
3. 原始问题：

$$\min_{\vec{\mathbf{w}},b}\frac{1}{2}||\vec{\mathbf{w}}||_2^2$$
$$s.t.\quad \tilde{y}_i(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b)-1\geq 0, i=1,2,\cdots,N$$

    定义拉格朗日函数：

$$L(\vec{\mathbf{w}},b,\vec{\alpha})=\frac{1}{2}||\vec{\mathbf{w}}||_2^2-\sum_{i=1}^{N}\alpha_i\tilde{y}_i(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b)+\sum_{i=1}^{N}\alpha_i$$

    其中 $\vec{\alpha}=(\alpha_1,\alpha_2,\cdots,\alpha_N)^T$ 为拉格朗日乘子向量。
    - 根据拉格朗日对偶性，原始问题的对偶问题是极大极小问题：

$$\max_{\vec{\alpha}}\min_{\vec{\mathbf{w}},b}L(\vec{\mathbf{w}},b,\vec{\alpha})$$

    - 先求 $\min_{\vec{\mathbf{w}},b}L(\vec{\mathbf{w}},b,\vec{\alpha})$。拉格朗日函数分别为 $\vec{\mathbf{w}},b$ 求偏导数，并令其等于 0

$$\nabla_{\vec{\mathbf{w}}}L(\vec{\mathbf{w}},b,\vec{\alpha})=\vec{\mathbf{w}}-\sum_{i=1}^{N}\alpha_i\tilde{y}_i\vec{\mathbf{x}}_i=0$$

$$\nabla_{b}L(\vec{\mathbf{w}},b,\vec{\alpha})=\sum_{i=1}^{N}\alpha_i\tilde{y}_i=0$$

$$\Rightarrow \vec{\mathbf{w}}=\sum_{i=1}^{N}\alpha_i\tilde{y}_i\vec{\mathbf{x}}_i,\quad \sum_{i=1}^{N}\alpha_i\tilde{y}_i=0$$

- 代入拉格朗日函数：

$$L(\vec{\mathbf{w}},b,\vec{\alpha})=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\tilde{y}_i\tilde{y}_j(\vec{\mathbf{x}}_i\cdot\vec{\mathbf{x}}_j)-\sum_{i=1}^{N}\alpha_i\tilde{y}_i\left[\left(\sum_{j=1}^{N}\alpha_j\tilde{y}_j\vec{\mathbf{x}}_j\right)\cdot\vec{\mathbf{x}}_i+b\right]+\sum_{i=1}^{N}\alpha_i$$

$$=-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\tilde{y}_i\tilde{y}_j(\vec{\mathbf{x}}_i\cdot\vec{\mathbf{x}}_j)+\sum_{i=1}^{N}\alpha_i$$

- 对偶问题极大值为：

$$\max_{\vec{\alpha}}-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\tilde{y}_i\tilde{y}_j(\vec{\mathbf{x}}_i\cdot\vec{\mathbf{x}}_j)+\sum_{i=1}^{N}\alpha_i$$

$$s.t.\quad \sum_{i=1}^{N}\alpha_i\tilde{y}_i=0$$

$$\alpha_i\geq 0,i=1,2,\cdots,N$$

4. 设对偶最优化问题的 $\vec{\alpha}$ 的解为 $\vec{\alpha}^*=(\alpha_1^*,\alpha_2^*,\cdots,\alpha_N^*)$，则根据 KKT 条件有：

$$\nabla_{\vec{\mathbf{w}}}L(\vec{\mathbf{w}}^*,b^*,\vec{\alpha}^*)=\vec{\mathbf{w}}^*-\sum_{i=1}^{N}\alpha_i^*\tilde{y}_i\vec{\mathbf{x}}_i=0$$

$$\nabla_{b}L(\vec{\mathbf{w}}^*,b^*,\vec{\alpha}^*)=\sum_{i=1}^{N}\alpha_i^*\tilde{y}_i=0$$

$$\alpha_i^*[\tilde{y}_i(\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}}_i+b^*)-1]=0,i=1,2,\cdots,N$$

$$\tilde{y}_i(\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}}_i+b^*)-1\geq 0,i=1,2,\cdots,N$$

$$\alpha_i^*\geq 0,i=1,2,\cdots,N$$

- 根据第一个式子，有：$\vec{\mathbf{w}}^*=\sum_{i=1}^{N}\alpha_i^*\tilde{y}_i\vec{\mathbf{x}}_i$ 。
- 由于 $\vec{\alpha}^*$ 不是零向量（若它为零向量，则 $\vec{\mathbf{w}}^*$ 也为零向量，矛盾），则必然存在某个 $j$ 使得 $\alpha_j^*>0$。

  根据第三个式子，此时必有 $\tilde{y}_j(\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}}_j+b^*)-1=0$。同时考虑到 $\tilde{y}_j^2=1$，得到：

$$b^*=\tilde{y}_j-\sum_{i=1}^{N}\alpha_i^*\tilde{y}_i(\vec{\mathbf{x}}_i\cdot\vec{\mathbf{x}}_j)$$

- 于是分离超平面写作：$\sum_{i=1}^{N}\vec{\alpha}_i^*\tilde{y}_i(\vec{\mathbf{x}}\cdot\vec{\mathbf{x}}_i)+b^*=0$ 。

  分类决策函数写作：$f(\vec{\mathbf{x}})=\text{sign}\left(\sum_{i=1}^{N}\alpha_i^*\tilde{y}_i(\vec{\mathbf{x}}\cdot\vec{\mathbf{x}}_i)+b^*\right)$ 。

  上式称作线性可分支持向量机的对偶形式。

可以看到：分类决策函数只依赖于输入 $\vec{\mathbf{x}}$ 和训练样本的内积。

5. 线性可分支持向量机对偶算法：
   - 输入：线性可分训练数据集 $\mathbb{D}=\{(\vec{\mathbf{x}}_1,\tilde{y}_1),(\vec{\mathbf{x}}_2,\tilde{y}_2),\cdots,(\vec{\mathbf{x}}_N,\tilde{y}_N)\}$ ，其中 $\vec{\mathbf{x}}_i\in\mathcal{X}=\mathbb{R}^n,\tilde{y}_i\in\mathcal{Y}=\{+1,-1\},i=1,2,\cdots,N$
   - 输出：
     - 最大几何间隔的分离超平面
     - 分类决策函数
   - 算法步骤：
     - 构造并且求解约束最优化问题：

$$\min_{\vec{\alpha}}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\tilde{y}_i\tilde{y}_j(\vec{\mathbf{x}}_i\cdot\vec{\mathbf{x}}_j)-\sum_{i=1}^{N}\alpha_i$$

$$s.t.\quad\sum_{i=1}^{N}\alpha_i\tilde{y}_i=0$$

$$\alpha_i\geq 0,i=1,2,\cdots,N$$

       求得最优解 $\vec{\alpha}^*=(\alpha_1^*,\alpha_2^*,\cdots,\alpha_N^*)^T$ 。
     - 计算 $\vec{\mathbf{w}}^*=\sum_{i=1}^{N}\alpha_i^*\tilde{y}_i\vec{\mathbf{x}}_i$ 。
     - 选择 $\vec{\alpha}^*$ 的一个正的分量 $\alpha_j^*>0$，计算 $b^*=\tilde{y}_j-\sum_{i=1}^{N}\alpha_i^*\tilde{y}_i(\vec{\mathbf{x}}_i\cdot\vec{\mathbf{x}}_j)$ 。
     - 由此得到分离超平面：$\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}}+b^*=0$ ，以及分类决策函数： $f(\vec{\mathbf{x}})=\text{sign}(\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}}+b^*)$ 。
6. $\vec{\mathbf{w}}^*,b^*$ 只依赖于 $\alpha_i^*>0$ 对应的样本点 $\vec{\mathbf{x}}_i,\tilde{y}_i$，而其他的样本点对于 $\vec{\mathbf{w}}^*,b^*$ 没有影响。
   - 将训练数据集里面对应于 $\alpha_i^*>0$ 的样本点对应的实例 $\vec{\mathbf{x}}_i$ 称为支持向量。
   - 对于$\alpha_i^*>0$ 的样本点，根据 $\alpha_i^*[\tilde{y}_i(\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}}_i+b^*)-1]=0$ ，有：$\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}}_i+b^*=\pm 1$ 。

     即 $\vec{\mathbf{x}}_i$ 一定在间隔边界上。这与原始问题给出的支持向量的定义一致。

# 二、线性支持向量机

1. 对于线性不可分训练数据，线性支持向量机不再适用，但可以想办法将它扩展到线性不可分问题。

## 2.1 原始问题

1. 设训练集为 $\mathbb{D}=\{(\vec{\mathbf{x}}_1,\tilde{y}_1),(\vec{\mathbf{x}}_2,\tilde{y}_2),\cdots,(\vec{\mathbf{x}}_N,\tilde{y}_N)\}$ ，其中 $\vec{\mathbf{x}}_i\in\mathcal{X}=\mathbb{R}^n,\tilde{y}_i\in\mathcal{Y}=\{+1,-1\},i=1,2,\cdots,N$。

   假设训练数据集不是线性可分的，这意味着某些样本点 $(\vec{\mathbf{x}}_i,\tilde{y}_i)$ 不满足函数间隔大于等于 1 的约束条件。
   - 对每个样本点 $(\vec{\mathbf{x}}_i,\tilde{y}_i)$ 引进一个松弛变量 $\xi_i\geq 0$，使得函数间隔加上松弛变量大于等于 1。

     即约束条件变成了： $\tilde{y}_i(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i)\geq 1-\xi_i$ 。
   - 对每个松弛变量 $\xi_i$ ，支付一个代价 $\xi_i$。目标函数由原来的 $\frac{1}{2}||\vec{\mathbf{w}}||_2^2$ 变成：

$$\min\frac{1}{2}||\vec{\mathbf{w}}||_2^2+C\sum_{i=1}^{N}\xi_i$$

     这里 $C>0$ 称作惩罚参数，一般由应用问题决定。
     - $C$ 值大时，对误分类的惩罚增大，此时误分类点凸显的更重要

- $C$ 值较大时，对误分类的惩罚增加，此时误分类点比较重要。
- $C$ 值较小时，对误分类的惩罚减小，此时误分类点相对不重要。

2. 相对于硬间隔最大化， $\frac{1}{2}||\vec{\mathbf{w}}||_2^2+C\sum_{i=1}^N\xi_i$ 称为软间隔最大化。

   于是线性不可分的线性支持向量机的学习问题变成了凸二次规划问题：

$$\begin{aligned}&\min_{\vec{\mathbf{w}},b,\vec{\xi}}\frac{1}{2}||\vec{\mathbf{w}}||_2^2+C\sum_{i=1}^N\xi_i\\ s.t.\quad&\tilde{y}_i(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b)\geq 1-\xi_i,\quad i=1,2,\cdots,N\\&\xi_i\geq 0,\quad i=1,2,\cdots,N\end{aligned}$$

   - 这称为线性支持向量机的原始问题。
   - 因为这是个凸二次规划问题，因此解存在。

     可以证明 $\vec{\mathbf{w}}$ 的解是唯一的； $b$ 的解不是唯一的，$b$ 的解存在于一个区间。

3. 对于给定的线性不可分的训练集数据，通过求解软间隔最大化问题得到的分离超平面为：$\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}}+b^*=0$ ，以及相应的分类决策函数：$f(\vec{\mathbf{x}})=\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}}+b^*$ ，称之为线性支持向量机。

- 线性支持向量机包含线性可分支持向量机。
- 现实应用中训练数据集往往是线性不可分的，线性支持向量机具有更广泛的适用性。

## 2.2 对偶问题

1. 定义拉格朗日函数为：

$$\begin{aligned}L(\vec{\mathbf{w}},b,\vec{\xi},\vec{\alpha},\vec{\mu})=\frac{1}{2}||\vec{\mathbf{w}}||_2^2+C\sum_{i=1}^N\xi_i-\sum_{i|1}^N\alpha_i[\tilde{y}_i(\vec{\mathbf{w}}_i\cdot\vec{\mathbf{x}}_i+b)-1+\xi_i]-\sum_{i=1}^N\mu_i\xi_i\\ \alpha_i\geq 0,\mu_i\geq 0\end{aligned}$$

   原始问题是拉格朗日函数的极小极大问题；对偶问题是拉格朗日函数的极大极小问题。

   - 先求 $L(\vec{\mathbf{w}},b,\vec{\xi},\vec{\alpha},\vec{\mu})$ 对 $\vec{\mathbf{w}},b,\vec{\xi}$ 的极小。根据偏导数为0：

$$\begin{aligned}\nabla_{\vec{\mathbf{w}}}L(\vec{\mathbf{w}},b,\vec{\xi},\vec{\alpha},\vec{\mu})&=\vec{\mathbf{w}}-\sum_{i=1}^N\alpha_i\tilde{y}_i\vec{\mathbf{x}}_i=\vec{\mathbf{0}}\\ \nabla_b L(\vec{\mathbf{w}},b,\vec{\xi},\vec{\alpha},\vec{\mu})&=-\sum_{i=1}^N\alpha_i\tilde{y}_i=0\\ \nabla_{\xi_i}L(\vec{\mathbf{w}},b,\vec{\xi},\vec{\alpha},\vec{\mu})&=C-\alpha_i-\mu_i=0\end{aligned}$$

     得到：

$$\begin{aligned}\vec{\mathbf{w}}&=\sum_{i=1}^N\alpha_i\tilde{y}_i\vec{\mathbf{x}}_i\\ \sum_{i=1}^N\alpha_i\tilde{y}_i&=0\\ C-\alpha_i-\mu_i&=0\end{aligned}$$

   - 再求极大问题：将上面三个等式代入拉格朗日函数：

$$\max_{\vec{\alpha},\vec{\mu}}\min_{\vec{\mathbf{w}},b,\vec{\xi}}L(\vec{\mathbf{w}},b,\vec{\xi},\vec{\alpha},\vec{\mu})=\max_{\vec{\alpha},\vec{\mu}}\left[-\frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N\alpha_i\alpha_j\tilde{y}_i\tilde{y}_j(\vec{\mathbf{x}}_i\cdot\vec{\mathbf{x}}_j)+\sum_{i=1}^N\alpha_i\right]$$

于是得到对偶问题：

$$\min_{\vec{\alpha}} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\tilde{y}_i\tilde{y}_j(\vec{\mathbf{x}}_i\cdot\vec{\mathbf{x}}_j)-\sum_{i=1}^{N}\alpha_i$$

$$s.t.\quad \sum_{i=1}^{N}\alpha_i\tilde{y}_i=0$$

$$0\leq\alpha_i\leq C, i=1,2,\cdots,N$$

2. 根据 KKT 条件：

$$\nabla_{\vec{\mathbf{w}}}L(\vec{\mathbf{w}}^*,b^*,\vec{\xi}^*,\vec{\alpha}^*,\vec{\mu}^*)=\vec{\mathbf{w}}^*-\sum_{i=1}^{N}\alpha_i^*\tilde{y}_i\vec{\mathbf{x}}_i=\vec{\mathbf{0}}$$

$$\nabla_b L(\vec{\mathbf{w}}^*,b^*,\vec{\xi}^*,\vec{\alpha}^*,\vec{\mu}^*)=-\sum_{i=1}^{N}\alpha_i^*\tilde{y}_i=0$$

$$\nabla_{\xi_i} L(\vec{\mathbf{w}}^*,b^*,\vec{\xi}^*,\vec{\alpha}^*,\vec{\mu}^*)=C-\alpha_i^*-\mu_i^*=0$$

$$\alpha_i^*[\tilde{y}_i(\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}}_i+b^*)-1+\xi_i^*]=0$$

$$\mu_i^*\xi_i^*=0$$

$$\tilde{y}_i(\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}}_i+b^*)-1+\xi_i^*\geq 0$$

$$\xi_i^*\geq 0$$

$$C\geq\alpha_i^*\geq 0$$

$$\mu_i^*\geq 0$$

$$i=1,2,\cdots,N$$

则有：$\vec{\mathbf{w}}^*=\sum_{i=1}^{N}\vec{\alpha}^*\tilde{y}_i\vec{\mathbf{x}}_i$ 。

- 若存在 $\vec{\alpha}^*$ 的某个分量 $\alpha_j^*, 0<\alpha_j^*<C$，则有：$\mu_j^*=C-\alpha_j^*>0$。

  > 若 $\vec{\alpha}^*$ 的所有分量都等于 0 ，则得出 $\vec{\mathbf{w}}^*$ 为零，没有任何意义。
  >
  > 若 $\vec{\alpha}^*$ 的所有分量都等于 $C$，根据 $\sum\alpha_i^*\tilde{y}_i=0$，则要求 $\sum\tilde{y}_i=0$。这属于强加的约束，

  - 根据 $\mu_j^*\xi_j^*=0$ ，有 $\xi_j^*=0$。
  - 考虑 $\alpha_j^*[\tilde{y}_j(\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}}_j+b^*)-1+\xi_j^*]=0$ ，则有：$b^*=\tilde{y}_j-\sum_{i=1}^{N}\vec{\alpha}^*\tilde{y}_i(\vec{\mathbf{x}}_i\cdot\vec{\mathbf{x}}_j)$
- 分离超平面为：$\sum_{i=1}^{N}\vec{\alpha}^*\tilde{y}_i(\vec{\mathbf{x}}_i\cdot\vec{\mathbf{x}})+b^*=0$ 。
- 分类决策函数为：$f(\vec{\mathbf{x}})=\text{sign}\left[\sum_{i=1}^{N}\vec{\alpha}^*\tilde{y}_i(\vec{\mathbf{x}}_i\cdot\vec{\mathbf{x}})+b^*\right]$ 。

3. 线性支持向量机对偶算法：

- 输入：训练数据集 $\mathbb{D}=\{(\vec{\mathbf{x}}_1,\tilde{y}_1),(\vec{\mathbf{x}}_2,\tilde{y}_2),\cdots,(\vec{\mathbf{x}}_N,\tilde{y}_N)\}$ ，其中 $\vec{\mathbf{x}}_i\in\mathcal{X}=\mathbb{R}^n,\tilde{y}_i\in\mathcal{Y}=\{+1,-1\},i=1,2,\cdots,N$
- 输出：
  - 分离超平面
  - 分类决策函数
- 算法步骤：
  - 选择惩罚参数 $C>0$ ，构造并且求解约束最优化问题：

$$\min_{\vec{\alpha}} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\tilde{y}_i\tilde{y}_j(\vec{\mathbf{x}}_i\cdot\vec{\mathbf{x}}_j)-\sum_{i=1}^{N}\alpha_i$$

$$s.t.\quad \sum_{i=1}^{N}\alpha_i\tilde{y}_i=0$$

$$C\geq\alpha_i\geq 0, i=1,2,\cdots,N$$

求得最优解 $\vec{\alpha}^*=(\alpha_1^*,\alpha_2^*,\cdots,\alpha_N^*)^T$ 。

- 计算： $\vec{\mathbf{w}}^*=\sum_{i=1}^{N}\alpha_i^*\tilde{y}_i\vec{\mathbf{x}}_i$ 。
- 选择 $\vec{\alpha}^*$ 的一个合适的分量 $C>\alpha_j^*>0$，计算： $b^*=\tilde{y}_j-\sum_{i=1}^{N}\alpha_i^*\tilde{y}_i(\vec{\mathbf{x}}_i\cdot\vec{\mathbf{x}}_j)$ 。

  > 可能存在多个符合条件的 $\alpha_j^*$。这是由于原始问题中，对 $b$ 的解不唯一。所以实际计算时可以取在所有符合条件的样本点上的平均值。

- 由此得到分离超平面： $\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}}+b^*=0$，以及分类决策函数： $f(\vec{\mathbf{x}})=\text{sign}(\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}}+b^*)$ 。

## 2.3 支持向量

1. 在线性不可分的情况下，对偶问题的解 $\vec{\alpha}^*=(\alpha_1^*,\alpha_2^*,\cdots,\alpha_N^*)^T$ 中，对应于 $\alpha_i^*>0$ 的样本点 $(\vec{\mathbf{x}}_i,\tilde{y}_i)$ 的实例点 $\vec{\mathbf{x}}_i$ 称作支持向量，它是软间隔的支持向量。
2. 线性不可分的支持向量比线性可分时的情况复杂一些：

   根据 $\nabla_{\xi_i}L(\vec{\mathbf{w}},b,\vec{\xi},\vec{\alpha},\vec{\mu})=C-\alpha_i-\mu_i=0$，以及 $\mu_j^*\xi_j^*=0$，则：

   - 若 $\alpha_i^*<C$ ，则 $\mu_i>0$， 则松弛量 $\xi_i=0$。此时：支持向量恰好落在了间隔边界上。
   - 若 $\alpha_i^*=C$， 则 $\mu_i=0$，于是 $\xi_i$ 可能为任何正数：
     - 若 $0<\xi_i<1$，则支持向量落在间隔边界与分离超平面之间，分类正确。
     - 若 $\xi_i=1$，则支持向量落在分离超平面上。
     - 若 $\xi_i>1$，则支持向量落在分离超平面误分类一侧，分类错误。

## 2.4 合页损失函数

1. 定义取正函数为：

$$\text{plus}(z)=\begin{cases}z, & z>0\\ 0, & z\leq 0\end{cases}$$

   定义合页损失函数为： $L(\tilde{y},\hat{y})=\text{plus}(1-\tilde{y}\hat{y})$ ，其中 $\tilde{y}$ 为样本的标签值，$\hat{y}$ 为样本的模型预测值。

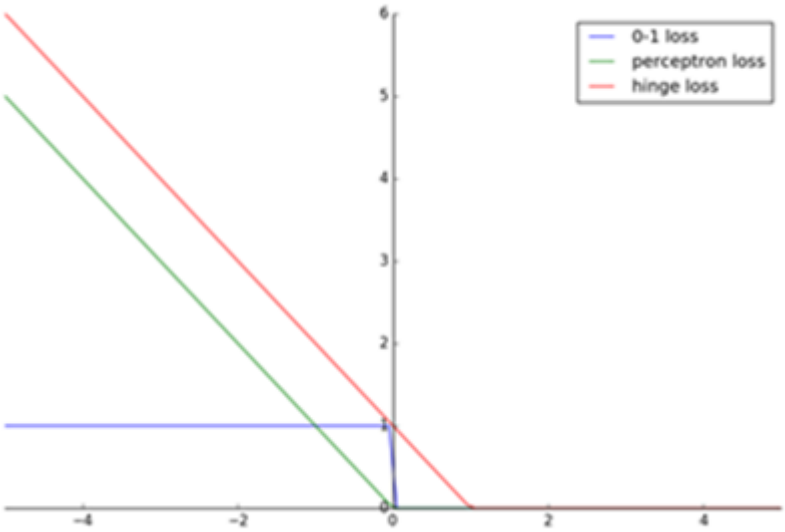
   则线性支持向量机就是最小化目标函数：

$$\sum_{i=1}^{N}\text{plus}(1-\tilde{y}_i(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b))+\lambda||\vec{\mathbf{w}}||_2^2,\quad \lambda>0$$

2. 合页损失函数的物理意义：
   - 当样本点 $(\vec{\mathbf{x}}_i,\tilde{y}_i)$ 被正确分类且函数间隔（确信度） $\tilde{y}_i(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b)$ 大于 1 时，损失为0
   - 当样本点 $(\vec{\mathbf{x}}_i,\tilde{y}_i)$ 被正确分类且函数间隔（确信度） $\tilde{y}_i(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b)$ 小于等于 1 时损失为 $1-\tilde{y}_i(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b)$
   - 当样本点 $(\vec{\mathbf{x}}_i,\tilde{y}_i)$ 未被正确分类时损失为 $1-\tilde{y}_i(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b)$
3. 可以证明：线性支持向量机原始最优化问题等价于最优化问题：

$$\min_{\vec{\mathbf{w}},b}\sum_{i=1}^{N}\text{plus}(1-\tilde{y}_i(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b))+\lambda||\vec{\mathbf{w}}||_2^2,\quad \lambda>0$$

4. 合页损失函数图形如下：
   - 感知机的损失函数为 $\text{plus}(-\tilde{y}(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}+b))$，相比之下合页损失函数不仅要分类正确，而且要确信度足够高（确信度为1）时，损失才是0。即合页损失函数对学习有更高的要求。
   - 0-1损失函数通常是二分类问题的真正的损失函数，合页损失函数是0-1损失函数的上界。
     - 因为0-1损失函数不是连续可导的，因此直接应用于优化问题中比较困难。
     - 通常都是用0-1损失函数的上界函数构成目标函数，这时的上界损失函数又称为代理损失函数。




5. 理论上 `SVM` 的目标函数可以使用梯度下降法来训练。但存在三个问题：
   - 合页损失函数部分不可导。这可以通过 `sub-gradient descent` 来解决。
   - 收敛速度非常慢。
   - 无法得出支持向量和非支持向量的区别。

# 三、非线性支持向量机

1. 非线性分类问题是指利用非线性模型才能很好的进行分类的问题。
2. 对于给定的训练集 $\mathbb{D}=\{(\vec{\mathbf{x}}_1,\tilde{y}_1),(\vec{\mathbf{x}}_2,\tilde{y}_2),\cdots,(\vec{\mathbf{x}}_N,\tilde{y}_N)\}$ ，其中 $\vec{\mathbf{x}}_i\in\mathcal{X}=\mathbb{R}^n,\tilde{y}_i\in\mathcal{Y}=\{+1,-1\},i=1,2,\cdots,N$，如果能用 $\mathbb{R}^n$ 中的一个超曲面将正负实例正确分开，则称这个问题为非线性可分问题。
3. 设原空间为 $\mathcal{X}\subset\mathbb{R}^2,\vec{\mathbf{x}}=(x_1,x_2)^T\in\mathcal{X}$，新的空间为 $\mathcal{Z}\subset\mathbb{R}^2,\vec{\mathbf{z}}=(z_1,z_2)^T\in\mathcal{Z}$ 。定义

   从原空间到新空间的变换（映射）为：$\vec{\mathbf{z}}=\phi(\vec{\mathbf{x}})=(x_1^2,x_2^2)^T$ 。

   则经过变换 $\vec{\mathbf{z}}=\phi(\vec{\mathbf{x}})$：
   - 原空间 $\mathcal{X}\subset\mathbb{R}^2$ 变换为新空间 $\mathcal{Z}\subset\mathbb{R}^2$ ， 原空间中的点相应地变换为新空间中的点。
   - 原空间中的椭圆 $w_1x_1^2+w_2x_2^2+b=0$ 变换为新空间中的直线 $w_1z_1+w_2z_2+b=0$ 。
   - 若在变换后的新空间，直线 $w_1z_1+w_2z_2+b=0$ 可以将变换后的正负实例点正确分开，则原空间的非线性可分问题就变成了新空间的线性可分问题。

3. 用线性分类方法求解非线性分类问题分两步：

- 首先用一个变换将原空间的数据映射到新空间。
- 再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。

这一策略称作核技巧。

# 3.1 核函数

## 3.1.1 核函数定义

1. 设 $\mathcal{X}$ 是输入空间（欧氏空间 $\mathbb{R}^n$ 的子集或者离散集合）， $\mathcal{H}$ 为特征空间（希尔伯特空间）。若果存在一个从 $\mathcal{X}$ 到 $\mathcal{H}$ 的映射 $\phi(\vec{\mathbf{x}}):\mathcal{X}\to\mathcal{H}$ ，使得所有的 $\vec{\mathbf{x}},\vec{\mathbf{z}}\in\mathcal{X}$， 函数 $K(\vec{\mathbf{x}},\vec{\mathbf{z}})=\phi(\vec{\mathbf{x}})\cdot\phi(\vec{\mathbf{z}})$，则称 $K(\vec{\mathbf{x}},\vec{\mathbf{z}})$ 为核函数。

   即：核函数将原空间中的任意两个向量 $\vec{\mathbf{x}},\vec{\mathbf{z}}$ ，映射为特征空间中对应的向量之间的内积。

2. 实际任务中，通常直接给定核函数 $K(\vec{\mathbf{x}},\vec{\mathbf{z}})$ ，然后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。
   - 学习是隐式地在特征空间进行的，不需要显式的定义特征空间和映射函数。
   - 通常直接计算 $K(\vec{\mathbf{x}},\vec{\mathbf{z}})$ 比较容易，反而是通过 $\phi(\vec{\mathbf{x}})$ 和 $\phi(\vec{\mathbf{z}})$ 来计算 $K(\vec{\mathbf{x}},\vec{\mathbf{z}})$ 比较困难。
     - 首先特征空间 $\mathcal{H}$ 一般是高维的，甚至是无穷维的，映射 $\phi(\vec{\mathbf{x}})$ 不容易定义。
     - 其次核函数关心的是希尔伯特空间两个向量的内积，而不关心这两个向量的具体形式。因此对于给定的核函数，特征空间 $\mathcal{H}$ 和 映射函数 $\phi(\vec{\mathbf{x}})$ 取法并不唯一。
       - 可以取不同的特征空间 $\mathcal{H}$ 。
       - 即使是在同一个特征空间 $\mathcal{H}$ 里，映射函数 $\phi(\vec{\mathbf{x}})$ 也可以不同。
3. 在线性支持向量机的对偶形式中，无论是目标函数还是决策函数都只涉及输入实例之间的内积。
   - 在对偶问题的目标函数中的内积 $\vec{\mathbf{x}}_i\cdot\vec{\mathbf{x}}_j$ 可以用核函数 $K(\vec{\mathbf{x}}_i,\vec{\mathbf{x}}_j)=\phi(\vec{\mathbf{x}}_i)\cdot\phi(\vec{\mathbf{x}}_j)$ 来代替。

     此时对偶问题的目标函数成为：

     $$L(\vec{\alpha})=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\tilde{y}_i\tilde{y}_jK(\vec{\mathbf{x}}_i,\vec{\mathbf{x}}_j)-\sum_{i=1}^{N}\alpha_i$$

   - 分类决策函数中的内积也可以用核函数代替：$f(\vec{\mathbf{x}})=\text{sign}\left(\sum_{i=1}^{N}\alpha_i^*\tilde{y}_iK(\vec{\mathbf{x}}_i,\vec{\mathbf{x}})+b^*\right)$ 。
4. 核函数替代法，等价于：
   - 首先经过映射函数 $\phi$ 将原来的输入空间变换到一个新的特征空间。
   - 然后将输入空间中的内积 $\vec{\mathbf{x}}_i\cdot\vec{\mathbf{x}}_j$ 变换为特征空间中的内积 $\phi(\vec{\mathbf{x}}_i)\cdot\phi(\vec{\mathbf{x}}_j)$ 。
   - 最后在新的特征空间里从训练样本中学习线性支持向量机。
5. 若映射函数 $\phi$ 为非线性函数，则学习到的含有核函数的支持向量机是非线性分类模型。

   若映射函数 $\phi$ 为线性函数，则学习到的含有核函数的支持向量机依旧是线性分类模型。

## 3.1.2 核函数选择

1. 在实际应用中，核函数的选取往往依赖领域知识，最后通过实验验证来验证核函数的有效性。
2. 若已知映射函数 $\phi$，那么可以通过 $\phi(\vec{\mathbf{x}})$ 和 $\phi(\vec{\mathbf{z}})$ 的内积求得核函数 $K(\vec{\mathbf{x}},\vec{\mathbf{z}})$。现在问题是：不用构造映射 $\phi$ ， 那么给定一个函数 $K(\vec{\mathbf{x}},\vec{\mathbf{z}})$ 判断它是否是一个核函数？

   即： $K(\vec{\mathbf{x}},\vec{\mathbf{z}})$ 满足什么条件才能成为一个核函数？

可以证明：设 $K:\mathcal{X}\times\mathcal{X}\to\mathbb{R}$ 是对称函数，则 $K(\vec{\mathbf{x}},\vec{\mathbf{z}})$ 为正定核函数的充要条件是：对任意 $\vec{\mathbf{x}}_i\in\mathcal{X},i=1,2,\cdots,N$，$K(\vec{\mathbf{x}},\vec{\mathbf{z}})$ 对应的 `Gram` 矩阵：$K=[K(\vec{\mathbf{x}}_i,\vec{\mathbf{x}}_j)]_{N\times N}$ 是半正定矩阵。

3. 对于一个具体函数 $K(\vec{\mathbf{x}},\vec{\mathbf{z}})$ 来说，检验它为正定核函数并不容易。因为要求对任意有限输入集 $\{\vec{\mathbf{x}}_1,\vec{\mathbf{x}}_2,\cdots,\vec{\mathbf{x}}_N\}$ 来验证 $K(\cdot,\cdot)$ 对应的 `Gram` 矩阵是否为半正定的。

   因此，实际问题中往往应用已有的核函数。

8. 常用核函数：
   - 多项式核函数：$K(\vec{\mathbf{x}},\vec{\mathbf{z}})=(\vec{\mathbf{x}}\cdot\vec{\mathbf{z}}+1)^p$ 。

     对应的支持向量机是一个 $p$ 次多项式分类器。
   - 高斯核函数：

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

     - 它是最常用的核函数，对应于无穷维空间中的点积。
     - 它也被称作径向基函数`radial basis function:RBF`，因为其值从 $\vec{\mathbf{x}}$ 沿着 $\vec{\mathbf{z}}$ 向外辐射的方向减小。
     - 对应的支持向量机是高斯径向基函数分类器(`radial basis function`)。
   - `sigmod`核函数：$K(\vec{\mathbf{x}},\vec{\mathbf{z}})=\tanh(\gamma(\vec{\mathbf{x}}\cdot\vec{\mathbf{z}})+r)$ 。

     对应的支持向量机实现的就是一种神经网络。

## 3.2 学习算法

1. 非线性支持向量机学习算法：
   - 输入：训练数据集 $\mathbb{D}=\{(\vec{\mathbf{x}}_1,\tilde{y}_1),(\vec{\mathbf{x}}_2,\tilde{y}_2),\cdots,(\vec{\mathbf{x}}_N,\tilde{y}_N)\}$，其中 $\vec{\mathbf{x}}_i\in\mathcal{X}=\mathbb{R}^n,\tilde{y}_i\in\mathcal{Y}=\{+1,-1\},i=1,2,\cdots,N$ 。
   - 输出：分类决策函数
   - 算法步骤：
     - 选择适当的核函数 $K(\vec{\mathbf{x}},\vec{\mathbf{z}})$ 和惩罚参数 $C>0$，构造并且求解约束最优化问题：

       $$\min_{\vec{\alpha}}\frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N\alpha_i\alpha_j\tilde{y}_i\tilde{y}_jK(\vec{\mathbf{x}}_i,\vec{\mathbf{x}}_j)-\sum_{i=1}^N\alpha_i$$

       $$s.t.\quad\sum_{i=1}^N\alpha_i\tilde{y}_i=0$$

       $$C\geq\alpha_i\geq0,i=1,2,\cdots,N$$

       求得最优解 $\vec{\alpha}^*=(\alpha_1^*,\alpha_2^*,\cdots,\alpha_N^*)^T$

       > 当 $K(\vec{\mathbf{x}},\vec{\mathbf{z}})$ 是正定核函数时，该问题为凸二次规划问题，解是存在的。

     - 计算：$\vec{\mathbf{w}}^*=\sum_{i=1}^N\alpha_i^*\tilde{y}_i\vec{\mathbf{x}}_i$ 。
     - 选择 $\vec{\alpha}^*$ 的一个合适的分量 $C>\alpha_j^*>0$，计算：$b^*=\tilde{y}_j-\sum_{i=1}^N\alpha_i^*\tilde{y}_iK(\vec{\mathbf{x}}_i,\vec{\mathbf{x}}_j)$ 。
     - 构造分类决策函数：$f(\vec{\mathbf{x}})=\text{sign}\left(\sum_{i=1}^N\alpha_i^*\tilde{y}_iK(\vec{\mathbf{x}}_i,\vec{\mathbf{x}})+b^*\right)$ 。

# 四、支持向量回归

1. 支持向量机不仅可以用于分类问题，也可以用于回归问题。
2. 给定训练数据集 $\mathbb{D}=\{(\vec{\mathbf{x}}_1,\tilde{y}_1),(\vec{\mathbf{x}}_2,\tilde{y}_2),\cdots,(\vec{\mathbf{x}}_N,\tilde{y}_N)\}$，其中 $\vec{\mathbf{x}}_i\in\mathcal{X}=\mathbb{R}^n,\tilde{y}_i\in\mathcal{Y}=\mathbb{R}$。
   - 对于样本 $(\vec{\mathbf{x}}_i,\tilde{y}_i)$，传统的回归模型通常基于模型输出 $f(\vec{\mathbf{x}}_i)$ 与真实输出 $\tilde{y}_i$ 之间的差别来计算损失。当且仅当 $f(\vec{\mathbf{x}}_i)$ 与 $\tilde{y}_i$ 完全相同时，损失才为零。
   - 支持向量回归(`Support Vector Regression:SVR`)不同：它假设能容忍 $f(\vec{\mathbf{x}}_i)$ 与 $\tilde{y}_i$ 之间最多有 $\epsilon$ 的偏差。仅当 $|f(\vec{\mathbf{x}}_i)-\tilde{y}_i|>\epsilon$ 时，才计算损失。

     支持向量回归相当于以 $f(\vec{\mathbf{x}}_i)$ 为中心，构建了一个宽度为 $2\epsilon$ 的间隔带。若训练样本落在此间隔带内则被认为是预测正确的。

## 4.1 原始问题

1. `SVR` 问题形式化为：

$$f(\vec{\mathbf{x}})=\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}+b$$

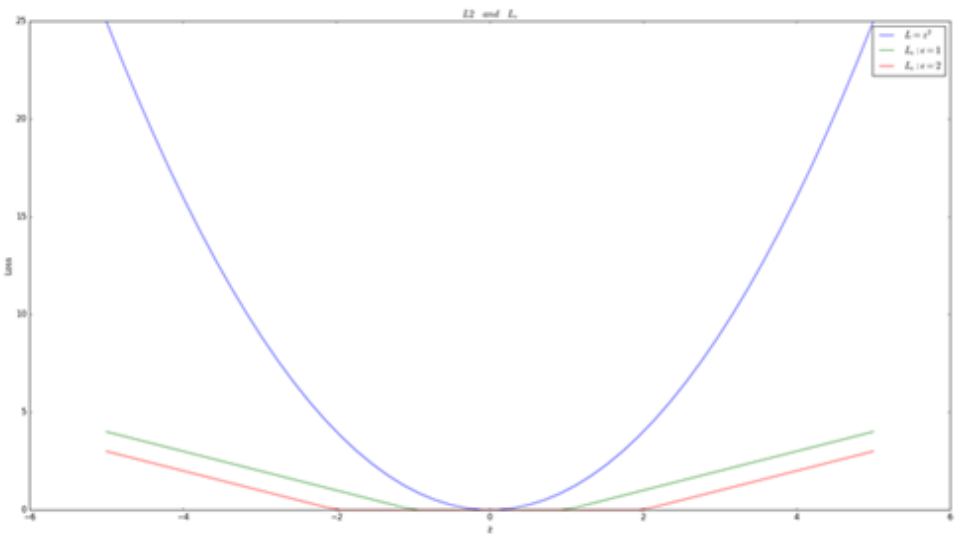
$$\min_{\vec{\mathbf{w}},b}\frac{1}{2}||\vec{\mathbf{w}}||_2^2+C\sum_{i=1}^{N}L_\epsilon\left(f(\vec{\mathbf{x}}_i)-\tilde{y}_i\right)$$

其中：

- $C$ 为罚项常数。
  - 若 $C$ 较大，则倾向于 $f(\vec{\mathbf{x}}_i)$ 与 $\tilde{y}_i$ 之间较小的偏差
  - 若 $C$ 较小，则能容忍 $f(\vec{\mathbf{x}}_i)$ 与 $\tilde{y}_i$ 之间较大的偏差
- $L_\epsilon$ 为损失函数。其定义为：

$$L_\epsilon(z)=\begin{cases}0 & ,\text{if}|z|\leq\epsilon\\ |z|-\epsilon & ,\text{else}\end{cases}$$

> 线性回归中，损失函数为 $L(z)=z^2$



2. 引入松弛变量 $\xi_i,\hat{\xi}_i$，将上式写做：

$$\min_{\vec{\mathbf{w}},b,\xi_i,\hat{\xi}_i} \frac{1}{2}||\vec{\mathbf{w}}||_2^2 + C\sum_{i=1}^{N}(\xi_i + \hat{\xi}_i)$$
$$s.t.\ f(\vec{\mathbf{x}}_i) - \tilde{y}_i \le \epsilon + \xi_i,$$
$$\tilde{y}_i - f(\vec{\mathbf{x}}_i) \le \epsilon + \hat{\xi}_i,$$
$$\xi_i \ge 0, \hat{\xi}_i \ge 0, i = 1, 2, \cdots, N$$

这就是 SVR 原始问题。

## 4.2 对偶问题

1. 引入拉格朗日乘子，$\mu_i \ge 0, \hat{\mu}_i \ge 0, \alpha_i \ge 0, \hat{\alpha}_i \ge 0$，定义拉格朗日函数：

$$L(\vec{\mathbf{w}}, b, \vec{\alpha}, \hat{\vec{\alpha}}, \vec{\xi}, \hat{\vec{\xi}}, \vec{\mu}, \hat{\vec{\mu}}) = \frac{1}{2}||\vec{\mathbf{w}}||_2^2 + C\sum_{i=1}^{N}(\xi_i + \hat{\xi}_i) - \sum_{i=1}^{N}\mu_i\xi_i - \sum_{i-1}^{N}\hat{\mu}_i\hat{\xi}_i$$
$$+ \sum_{i=1}^{N}\alpha_i\left(f(\vec{\mathbf{x}}_i) - \tilde{y}_i - \epsilon - \xi_i\right) + \sum_{i-1}^{N}\hat{\alpha}_i\left(\tilde{y}_i - f(\vec{\mathbf{x}}_i) - \epsilon - \hat{\xi}_i\right)$$

根据拉格朗日对偶性，原始问题的对偶问题是极大极小问题：

$$\max_{\vec{\alpha},\hat{\vec{\alpha}}} \min_{\vec{\mathbf{w}},b,\vec{\xi},\hat{\vec{\xi}}} L(\vec{\mathbf{w}}, b, \vec{\alpha}, \hat{\vec{\alpha}}, \vec{\xi}, \hat{\vec{\xi}}, \vec{\mu}, \hat{\vec{\mu}})$$

2. 先求极小问题：根据 $L(\vec{\mathbf{w}}, b, \vec{\alpha}, \hat{\vec{\alpha}}, \vec{\xi}, \hat{\vec{\xi}}, \vec{\mu}, \hat{\vec{\mu}})$ 对 $\vec{\mathbf{w}}, b, \vec{\xi}, \hat{\vec{\xi}}$ 偏导数为零可得：

$$\vec{\mathbf{w}} = \sum_{i=1}^{N}(\hat{\alpha}_i - \alpha_i)\vec{\mathbf{x}}_i$$
$$0 = \sum_{i=1}^{N}(\hat{\alpha}_i - \alpha_i)$$
$$C = \alpha_i + \mu_i$$
$$C = \hat{\alpha}_i + \hat{\mu}_i$$

3. 再求极大问题（取负号变极小问题）：

$$\min_{\vec{\alpha},\hat{\vec{\alpha}}} \sum_{i=1}^{N}[\tilde{y}_i(\hat{\alpha}_i - \alpha_i) - \epsilon(\hat{\alpha}_i + \alpha_i)] - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j)\vec{\mathbf{x}}_i^T\vec{\mathbf{x}}_j$$
$$s.t. \sum_{i=1}^{N}(\hat{\alpha}_i - \alpha_i) = 0$$
$$0 \le \alpha_i, \hat{\alpha}_i \le C$$

4. 上述过程需要满足 KKT 条件，即：

$$\begin{cases} \alpha_i\left(f(\vec{\mathbf{x}}_i) - \tilde{y}_i - \epsilon - \xi_i\right) = 0 \\ \hat{\alpha}_i\left(\tilde{y}_i - f(\vec{\mathbf{x}}_i) - \epsilon - \hat{\xi}_i\right) = 0 \\ \alpha_i\hat{\alpha}_i = 0 \\ \xi_i\hat{\xi}_i = 0 \\ (C - \alpha_i)\xi_i = 0 \\ (C - \hat{\alpha}_i)\hat{\xi}_i = 0 \end{cases}$$

5. 可以看出：
   - 当样本 $(\vec{\mathbf{x}}_i,\tilde{y}_i)$ 不落入$\epsilon$ 间隔带中时，对应的 $\alpha_i,\hat{\alpha}_i$ 才能取非零值：
     - 当且仅当 $f(\vec{\mathbf{x}}_i)-\tilde{y}_i-\epsilon-\xi_i=0$ 时， $\alpha_i$ 能取非零值
     - 当且仅当 $\tilde{y}_i-f(\vec{\mathbf{x}}_i)-\epsilon-\hat{\xi}_i=0$ 时， $\hat{\alpha}_i$ 能取非零值
   - 此外约束 $f(\vec{\mathbf{x}}_i)-\tilde{y}_i-\epsilon-\xi_i=0$ 与 $\tilde{y}_i-f(\vec{\mathbf{x}}_i)-\epsilon-\hat{\xi}_i=0$ 不能同时成立，因此 $\alpha_i,\hat{\alpha}_i$ 中至少一个为零。
6. 设最终解 $\vec{\alpha}$ 中，存在 $C>\alpha_j>0$，则有：

$$b=\tilde{y}_j+\epsilon-\sum_{i=1}^{N}(\hat{\alpha}_i-\alpha_j)\vec{\mathbf{x}}_i^T\vec{\mathbf{x}}_j$$

$$f(\vec{\mathbf{x}})=\sum_{i=1}^{N}(\hat{\alpha}_i-\alpha_i)\vec{\mathbf{x}}_i^T\vec{\mathbf{x}}+b$$

7. 最后若考虑使用核技巧，则 `SVR` 可以表示为：$f(\vec{\mathbf{x}})=\sum_{i=1}^{N}(\hat{\alpha}_i-\alpha_i)K(\vec{\mathbf{x}}_i,\vec{\mathbf{x}})+b$ 。

# 五、SVDD

## 5.1 one class 分类

1. 通常分类问题是两类或者多类，但有一种分类为一类 `one class` 的分类问题：它只有一个类，预测结果为是否属于这个类。
2. 一类分类的策略是：训练出一个最小的超球面把正类数据包起来。识别一个新的数据点时，如果这个数据点落在超球面内，则属于正类；否则不是。
3. 示例：给定一些用户的购物行为日志，其中包括两类用户：
   - 购买了某个商品的用户。可以肯定该类用户对于该商品是感兴趣的（标记为正例）。
   - 未购买某个商品的用户。此时无法断定该用户是对该商品感兴趣，还是不感兴趣（无法标记为反例）。

   现在给定一群新的用户，预测这些用户中，哪些可能对该商品有兴趣。

   如果简单的使用二类分类问题，则有两个问题：
   - 未购买商品的用户，不一定是对该商品不感兴趣，可能是由于某些原因未能购买。
   - 通常未购买商品的用户数量远大于购买用户的数量。如果使用二类分类，则容易造成正负样本不均匀。

## 5.2 SVDD 算法

1. `support vector domain description:SVDD` 可以用于一类分类问题。
2. 给定训练集 $\mathbb{D}=\{\vec{\mathbf{x}}_1,\vec{\mathbf{x}}_2,\cdots,\vec{\mathbf{x}}_N\}$，这些样本都是属于同一类。 `SVDD` 的的优化目标是：求一个中心为 $\vec{\mathbf{o}}$，半径为 $R$ 的最小球面，使得 $\mathbb{D}$ 中的样本都在该球面中。
3. 类似 `SVR` ， `SVDD` 允许一定程度上的放松，引入松弛变量。对松弛变量 $\xi_i$，其代价为 $C\xi_i$。

$$L(R,\vec{\mathbf{o}},\vec{\xi})=R^2+C\sum_{i=1}^{N}\xi_i$$

$$s.t.\quad ||\vec{\mathbf{x}}_i-\vec{\mathbf{o}}||_2^2\le R^2+\xi_i$$

$$\xi_i\ge 0$$

$$i=1,2,\cdots,N$$

   其中 $C>0$ 为惩罚系数：

- 若 $C$ 较大，则不能容忍那些球面之外的点，因此球面会较大。
- 若 $C$ 较小，则给予球面之外的点较大的弹性，因此球面会较小。

4. `SVDD` 的求解也是采用拉格朗日乘子法：

$$L(R,\vec{\mathbf{o}},\vec{\alpha},\vec{\xi},\vec{\gamma})=R^2+C\sum_{i=1}^{N}\xi_i-\sum_{i=1}^{N}\alpha_i\left(R^2+\xi_i-||\vec{\mathbf{x}}_i-\vec{\mathbf{o}}||_2^2\right)-\sum_{i=1}^{N}\gamma_i\xi_i$$
$$s.t.\ \alpha_i\geq 0,\gamma_i\geq 0,\xi_i\geq 0$$

   - 根据拉格朗日对偶性，原始问题的对偶问题是极大极小问题：$\max_{\vec{\alpha},\vec{\gamma}}\min_{R,\vec{\mathbf{a}},\vec{\xi}}L(R,\vec{\mathbf{o}},\vec{\alpha},\vec{\xi},\vec{\gamma})$ 。
   - 先求极小问题：根据 $L(R,\vec{\mathbf{o}},\vec{\alpha},\vec{\xi},\vec{\gamma})$ 对 $R,\vec{\mathbf{o}},\vec{\xi}$ 偏导数为零可得：

$$\sum_{i=1}^{N}\alpha_i=1$$
$$\vec{\mathbf{o}}=\frac{\sum_{i=1}^{N}\alpha_i\vec{\mathbf{x}}_i}{\sum_{i=1}^{N}\alpha_i}=\sum_{i=1}^{N}\alpha_i\vec{\mathbf{x}}_i$$
$$C-\alpha_i-\gamma_i=0,i=1,2,\cdots,N$$

   - 代入拉格朗日函数有：

$$L=\sum_{i=1}^{N}\alpha_i(\vec{\mathbf{x}}_i\cdot\vec{\mathbf{x}}_i)-\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j(\vec{\mathbf{x}}_i\cdot\vec{\mathbf{x}}_j)$$
$$s.t.\ 0\leq\alpha_i\leq C$$
$$\sum_{i=1}^{N}\alpha_i=1$$

   - 引入核函数：

$$L=\sum_{i=1}^{N}\alpha_iK(\vec{\mathbf{x}}_i,\vec{\mathbf{x}}_i)-\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_jK(\vec{\mathbf{x}}_i,\vec{\mathbf{x}}_j)$$
$$s.t.\ 0\leq\alpha_i\leq C$$
$$\sum_{i=1}^{N}\alpha_i=1$$

     其解法类似支持向量机的解法。

5. 判断一个新的数据点 $\vec{\mathbf{z}}$ 是否属于这个类，主要看它是否在训练出来的超球面内：若 $||\vec{\mathbf{z}}-\vec{\mathbf{o}}||_2^2\leq R^2$，则判定为属于该类。
   - 如果使用支持向量，则判定准则为：$(\vec{\mathbf{z}}\cdot\vec{\mathbf{z}})-2\sum_{i=1}^{N}\alpha_i(\vec{\mathbf{z}}\cdot\vec{\mathbf{x}}_i)+\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j(\vec{\mathbf{x}}_i\cdot\vec{\mathbf{x}}_j)\leq R^2$ 。
   - 如果是用核函数，则判定准则为：$K(\vec{\mathbf{z}}\cdot\vec{\mathbf{z}})-2\sum_{i=1}^{N}\alpha_iK(\vec{\mathbf{z}}\cdot\vec{\mathbf{x}}_i)+\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_jK(\vec{\mathbf{x}}_i\cdot\vec{\mathbf{x}}_j)\leq R^2$ 。

# 六、序列最小最优化方法

1. 支持向量机的学习问题可以形式化为求解凸二次规划问题。这样的凸二次规划问题具有全局最优解，并且有多种算法可以用于这一问题的求解。

当训练样本容量非常大时，这些算法往往非常低效。而序列最小最优化(`sequential minimal optimization:SMO`)算法可以高效求解。

2. `SMO`算法的思路：
   - 若所有变量都满足条件，则最优化问题的解就得到了。
   - 否则，选择两个变量的同时固定其他所有变量，针对这两个变量构建一个二次规划子问题。
     - 这个二次规划子问题关于这两个变量的解应该更接近原始二次规划问题的解，因为这会使得原始二次规划问题的目标函数值变得更小。
     - 更重要的是，这个二次规划子问题可以通过解析的方法求解。
     - 此时子问题有两个变量，至少存在一个变量不满足约束条件（否则就是所有变量满足条件了）。

       假设其中一个是违反约束最严重的那个，另一个由约束等式自动确定：$\sum_{i=1}^N \alpha_i\tilde{y}_i=0$。

3. `SMO`算法将原始问题不断地分解为子问题并且对子问题求解，进而达到求解原问题的目的。

   整个`SMO`算法包括两部分：
   - 求解两个变量二次规划的解析方法。
   - 选择变量的启发式方法。

## 6.1 子问题的求解

1. 假设选择的两个变量是 $\alpha_1,\alpha_2$， 其他变量 $\alpha_i,i=3,4,\cdots,N$ 是固定的。

   于是`SMO`的最优化问题的子问题为：

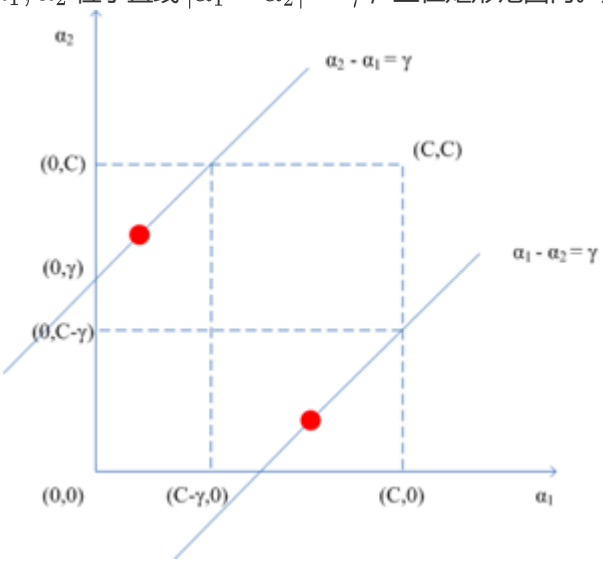
$$\begin{aligned}\min_{\alpha_1,\alpha_2}L(\alpha_1,\alpha_2)=&\frac{1}{2}K_{11}\alpha_1^2+\frac{1}{2}K_{22}\alpha_2^2+\tilde{y}_1\tilde{y}_2K_{12}\alpha_1\alpha_2-(\alpha_1+\alpha_2)\\&+\tilde{y}_1\alpha_1\sum_{i=3}^N\tilde{y}_i\alpha_iK_{i1}+\tilde{y}_2\alpha_2\sum_{i=3}^N\tilde{y}_i\alpha_iK_{i2}\\s.t.\quad&\alpha_1\tilde{y}_1+\alpha_2\tilde{y}_2=-\sum_{i=3}^N\tilde{y}_i\alpha_i=\gamma\\&0\leq\alpha_i\leq C,i=1,2\end{aligned}$$

   其中 $K_{ij}=K(\vec{\mathbf{x}}_i,\vec{\mathbf{x}}_j),i,j=1,2,\cdots,N,\quad\gamma$ 为常数， 且目标函数式中省略了不含 $\alpha_1,\alpha_2$ 的常数项。

### 6.1.1 取值范围约束

1. $\alpha_1,\alpha_2$ 的约束条件为：

- 当 $\tilde{y}_1, \tilde{y}_2$ 异号时，$\alpha_1, \alpha_2$ 位于直线 $|\alpha_1 - \alpha_2| = \gamma$，且在矩形范围内。矩形的长为 $C$，宽为 $C$， 起始点坐标为 `(0,0)` ：




此时 $\alpha_2$ 和 $\alpha_1$ 的取值范围为：

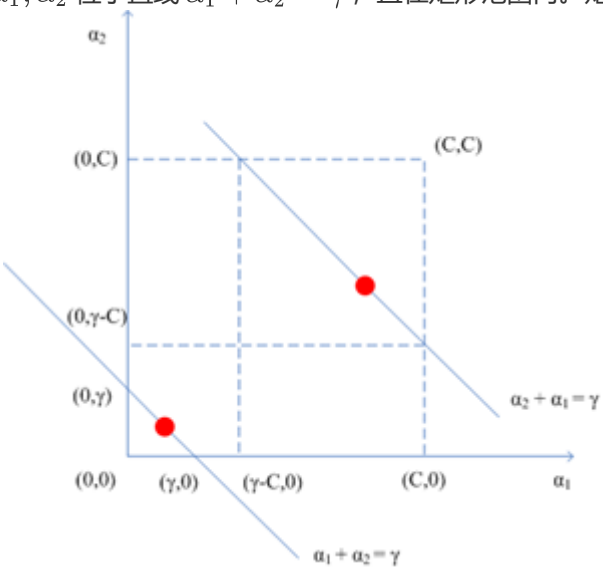
  - 当 $\alpha_2 \geq \alpha_1$ 时（上面那条线）：

$$\gamma \leq \alpha_2 \leq C$$
$$0 \leq \alpha_1 \leq C - \gamma$$

  - 当 $\alpha_1 > \alpha_2$ 时（下面那条线）：

$$0 \leq \alpha_2 \leq C - \gamma$$
$$\gamma \leq \alpha_1 \leq C$$

- 当 $\tilde{y}_1, \tilde{y}_2$ 同号时，$\alpha_1, \alpha_2$ 位于直线 $\alpha_1 + \alpha_2 = \gamma$，且在矩形范围内。矩形的长为 $C$，宽为 $C$， 起始点坐标为 `(0,0)` ：




此时 $\alpha_1, \alpha_2$ 取值范围为：

  - 当 $\gamma \geq C$ 时：（上面那条线）

$$\gamma - C \leq \alpha_2 \leq C$$
$$\gamma - C \leq \alpha_1 \leq C$$

  - 当 $\gamma < C$ 时：（下面那条线）

$$0 \leq \alpha_2 \leq \gamma$$
$$0 \leq \alpha_1 \leq \gamma$$

2. 假设 $\alpha_2$ 的最优解为 $\alpha_2^{new}$，其初始可行解为 $\alpha_2^{old}$；$\alpha_1$ 的初始可行解为 $\alpha_1^{old}$ 。
   - 既然是初始可行解，则需要满足约束条件。因此有
   $$|\alpha_1^{old} - \alpha_2^{old}| = \gamma, \quad \text{if } \tilde{y}_1 \neq \tilde{y}_2$$
   $$\alpha_1^{old} + \alpha_2^{old} = \gamma, \quad \text{if } \tilde{y}_1 = \tilde{y}_2$$
   - 假设 $\alpha_2^{new}$ 的取值范围为 $[L, H]$，则有：
     - 当 $\tilde{y}_1 \neq \tilde{y}_2$时：

       若 $\alpha_2^{new} \geq \alpha_1^{new}$，则 $\gamma \leq \alpha_2^{new} \leq C$；若 $\alpha_1^{new} > \alpha_2^{new}$，则 $0 \leq \alpha_2^{new} \leq C - \gamma$。

       根据：
       $$\gamma = |\alpha_1^{old} - \alpha_2^{old}| = \begin{cases} \alpha_2^{old} - \alpha_1^{old}, & \text{if } \alpha_2^{old} \geq \alpha_1^{old} \\ \alpha_1^{old} - \alpha_2^{old}, & \text{else} \end{cases}$$

       则有：$L = \max(0, \alpha_2^{old} - \alpha_1^{old}),\ H = \min(C, C + \alpha_2^{old} - \alpha_1^{old})$。
     - 当 $\tilde{y}_1 = \tilde{y}_2$ 时：

       若 $\gamma \geq C$，则 $\gamma - C \leq \alpha_2^{new} \leq C$；若$\gamma < C$，则 $0 \leq \alpha_2^{new} \leq \gamma$。

       根据 $\gamma = \alpha_1^{old} + \alpha_2^{old}$，则有：$L = \max(0, \alpha_2^{old} + \alpha_1^{old} - C),\ H = \min(C, \alpha_2^{old} + \alpha_1^{old})$

### 6.1.2 解析解

1. 令 $g(\vec{\mathbf{x}}) = \sum_{i=1}^{N} \alpha_i \tilde{y}_i K(\vec{\mathbf{x}_i}, \vec{\mathbf{x}}) + b$。它表示解得 $\vec{\alpha}, b$ 参数之后，对 $\vec{\mathbf{x}}$ 的预测值。预测值的正负代表了分类的结果。

   令 $E_i = g(\vec{\mathbf{x}}_i) - \tilde{y}_i = \left(\sum_{j=1}^{N} \alpha_j \tilde{y}_j K(\mathbf{x}_j, \vec{\mathbf{x}}_i) + b\right) - \tilde{y}_i, \quad i = 1, 2$。
   - $E_1$ 表示 $g(\vec{\mathbf{x}}_1)$ 的预测值与真实输出 $\tilde{y}_1$ 之差。
   - $E_2$ 表示 $g(\vec{\mathbf{x}}_2)$ 的预测值与真实输出 $\tilde{y}_2$ 之差。
2. 根据 $\alpha_1 \tilde{y}_1 + \alpha_2 \tilde{y}_2 = \gamma$，将 $\alpha_1 = \frac{\gamma - \tilde{y}_2 \alpha_2}{\tilde{y}_1}$ 代入 $L(\alpha_1, \alpha_2)$ 中。

   求解 $d\frac{L(\alpha_2)}{d\alpha_2} = 0$，即可得到 $\alpha_2$ 的最优解（不考虑约束条件）：
   $$\alpha_2^{new,unc} = \alpha_2^{old} + \frac{\tilde{y}_2(E_1 - E_2)}{\eta}$$

   其中：
   - $\eta = K_{11} + K_{22} - 2K_{12}$
   - $E_1, E_2$ 中的 $\alpha_1, \alpha_2$ 分别为 $\alpha_1^{old}, \alpha_2^{old}$ （它们表示初始的可行解，用于消掉 $\gamma$）。
3. 将 $\alpha_2^{new,unc}$ 截断，则得到 $\alpha_2^{new}$ 的解析解为：
   $$\alpha_2^{new} = \begin{cases} H, & \alpha_2^{new,unc} > H \\ \alpha_2^{new,unc}, & L \leq \alpha_2^{new,unc} \leq H \\ L, & \alpha_2^{new,unc} < L \end{cases}$$

   其中 $\alpha_1^{old}, \alpha_2^{old}$ 为初始可行解， $\alpha_2^{new}$ 为最终解。
4. 根据 $\tilde{y}_1 \alpha_1^{new} + \tilde{y}_2 \alpha_2^{new} = \gamma = \tilde{y}_1 \alpha_1^{old} + \tilde{y}_2 \alpha_2^{old}$，以及 $\tilde{y}_1^2 = \tilde{y}_2^2 = 1$，得到 $\alpha_1^{new}$：

$\alpha_1^{new}=\alpha_1^{old}+\tilde{y}_1\tilde{y}_2(\alpha_2^{old}-\alpha_2^{new})$ 。

其中 $\alpha_1^{old},\alpha_2^{old}$ 为初始可行解， $\alpha_1^{new}$ 为最终解。

## 6.2 变量选择

1. SMO 算法在每个子问题中选择两个变量进行优化，其中至少一个变量是违反约束条件的。如果都不违反约束条件，则说明已经求解了。

### 6.2.1 外层循环

1. 第一个变量的选择过程称为外层循环。
2. 外层循环在训练样本中选择违反约束条件最严重的样本点，并将对应的变量作为第一个变量。

   具体来讲，就是检验训练样本点 $(\vec{\mathbf{x}}_i,\tilde{y}_i)$ 是否满足约束条件( KKT 条件)：

   $$\alpha_i=0\Leftrightarrow\tilde{y}_ig(\vec{\mathbf{x}}_i)\geq 1$$
   $$0<\alpha_i<C\Leftrightarrow\tilde{y}_ig(\vec{\mathbf{x}}_i)=1$$
   $$\alpha_i=C\Leftrightarrow\tilde{y}_ig(\vec{\mathbf{x}}_i)\leq 1$$

   其中， $g(\vec{\mathbf{x}}_i)=\sum_{j=1}^N\alpha_j\tilde{y}_jK(\vec{\mathbf{x}}_i,\vec{\mathbf{x}}_j)+b$ 。
3. 检验时：
   - 外层循环首先遍历所有满足条件 $0<\alpha_i<C$ 的样本点，即间隔边界上的支持向量点。
   - 如果这些样本点都满足条件，再遍历整个训练集，检验是否满足条件。

### 6.2.2 内存循环

1. 第二个变量的选择过程称为内层循环。
2. 假设已经在外层循环中找到第一个变量 $\alpha_1$， 现在要在内层循环中找到第二个变量 $\alpha_2$。第二个变量选择标准是希望能够使得 $\alpha_2$ 有足够大的变化。
3. 由前面式子可知， $\alpha_2^{new}$ 依赖于 $E_1-E_2$ 。一种简单的做法是选择 $\alpha_2$ ，使得对应的 $|E_1-E_2|$ 最大。因为 $\alpha_1$ 已经确定， $E_1$ 也已经确定。
   - 如果 $E_1$ 为正数，则选择最小的 $E_i$ 作为 $E_2$。
   - 如果 $E_1$ 为负数，则选择最大的 $E_i$ 作为 $E_2$。

   > 为了节省计算时间，可以将所有 $E_i$ 值保存在一个列表中。
4. 特殊情况下，若内层循环找到的 $\alpha_2$ 不能使得目标函数有足够的下降，则采用以下的启发式规则继续选择 $\alpha_2$ ：
   - 遍历在间隔边界上的支持向量点，依次将其对应的变量作为 $\alpha_2$ 试用，直到目标函数有足够的下降。
   - 若还是找不到合适的 $\alpha_2$ ，则遍历训练数据集寻找。
   - 若还是找不到合适的 $\alpha_2$ ，则放弃找到的 $\alpha_1$，再通过外层循环寻求另外的 $\alpha_1$ 。

### 6.2.3 参数更新

1. 每次完成两个变量的优化后，都要重新计算 $b$。根据约束条件有：
   - 当 $0<\alpha_1^{new}<C$ 时： $\sum_{i=1}^N\alpha_i\tilde{y}_iK_{i1}+b=\tilde{y}_1$ 。

     代入 $E_1$ 的定义式有： $b_1^{new}=-E_1-\tilde{y}_1K_{11}(\alpha_1^{new}-\alpha_1^{old})-\tilde{y}_2K_{21}(\alpha_2^{new}-\alpha_2^{old})+b^{old}$ 。

- 同样，当 $0<\alpha_2^{new}<C$ 时有：$b_2^{new}=-E_2-\tilde{y}_1K_{12}(\alpha_1^{new}-\alpha_1^{old})-\tilde{y}_2K_{22}(\alpha_2^{new}-\alpha_2^{old})+b^{old}$ 。
- 如果 $\alpha_1^{new}$，$\alpha_2^{new}$ 同时满足 $0<\alpha_i^{new}<C$，则有：$b_1^{new}=b_2^{new}$
- 如果 $\alpha_1^{new},\alpha_2^{new}$ 或者为 0，或者为 $C$，则 $[b_1^{new},b_2^{new}]$ 区间内的数都可以作为 $b^{new}$。此时一个选择是：

$$b^{new}=\frac{b_1^{new}+b_2^{new}}{2}$$

2. 每次完成两个变量的优化后，需要更新对应的 $E_i$ 值。 $E_i$ 的更新要用到 $b^{new}$ 以及所有支持向量对应的 $\alpha_j$ 。

## 6.3 SMO算法

1. `SMO` 算法：
   - 输入：
     - 训练数据集 $\mathbb{D}=\{(\vec{\mathbf{x}}_1,\tilde{y}_1),(\vec{\mathbf{x}}_2,\tilde{y}_2),\cdots,(\vec{\mathbf{x}}_N,\tilde{y}_N)\}$，其中 $\vec{\mathbf{x}}_i\in\mathcal{X}=\mathbb{R}^n,\tilde{y}_i\in\mathcal{Y}=\{+1,-1\}$
     - 精度 $\varepsilon$
   - 输出：近似解 $\hat{\vec{\alpha}}$
   - 算法步骤：
     - 取初值 $\vec{\alpha}^{(0)}=0$，$k=0$
     - 选取优化变量 $\alpha_1^{(k)},\alpha_2^{(k)}$，解析求解两个变量的最优化问题，求得最优解 $\alpha_1^{(k+1)},\alpha_2^{(k+1)}$，更新 $\vec{\alpha}$ 为 $\vec{\alpha}^{(k+1)}$ 。
     - 若在精度 $\varepsilon$ 范围内满足停机条件：

$$\sum_{i=1}^{N}\alpha_i\tilde{y}_i=0$$
$$0\leq\alpha_i\leq C,i=1,2,\cdots,N$$
$$\tilde{y}_ig(\vec{\mathbf{x}}_i)=\begin{cases}\geq 1, & \alpha_i=0\\ =1, & 0<\alpha_i<C\\ \leq 1, & \alpha_i=C\end{cases}$$

     则退出迭代并令 $\hat{\vec{\alpha}}=\vec{\alpha}^{(k+1)}$；否则令 $k=k+1$，继续迭代。

     其中 $g(\vec{\mathbf{x}}_i)=\sum_{j=1}^{N}\alpha_j\tilde{y}_jK(\vec{\mathbf{x}}_i,\vec{\mathbf{x}}_j)+b$ 。

# 七、其它讨论

1. 支持向量机的优点：
   - 有严格的数学理论支持，可解释性强。
   - 能找出对任务至关重要的关键样本（即：支持向量）。
   - 采用核技巧之后，可以处理非线性分类/回归任务。
2. 支持向量机的缺点：
   - 训练时间长。当采用 `SMO` 算法时，由于每次都需要挑选一对参数，因此时间复杂度为 $O(N^2)$，其中 $N$ 为 $\vec{\alpha}$ 的长度，也就是训练样本的数量。
   - 当采用核技巧时，如果需要存储核矩阵，则空间复杂度为 $O(N^2)$ 。
   - 模型预测时，预测时间与支持向量的个数成正比。当支持向量的数量较大时，预测计算复杂度较高。

因此支持向量机目前只适合小批量样本的任务，无法适应百万甚至上亿样本的任务。