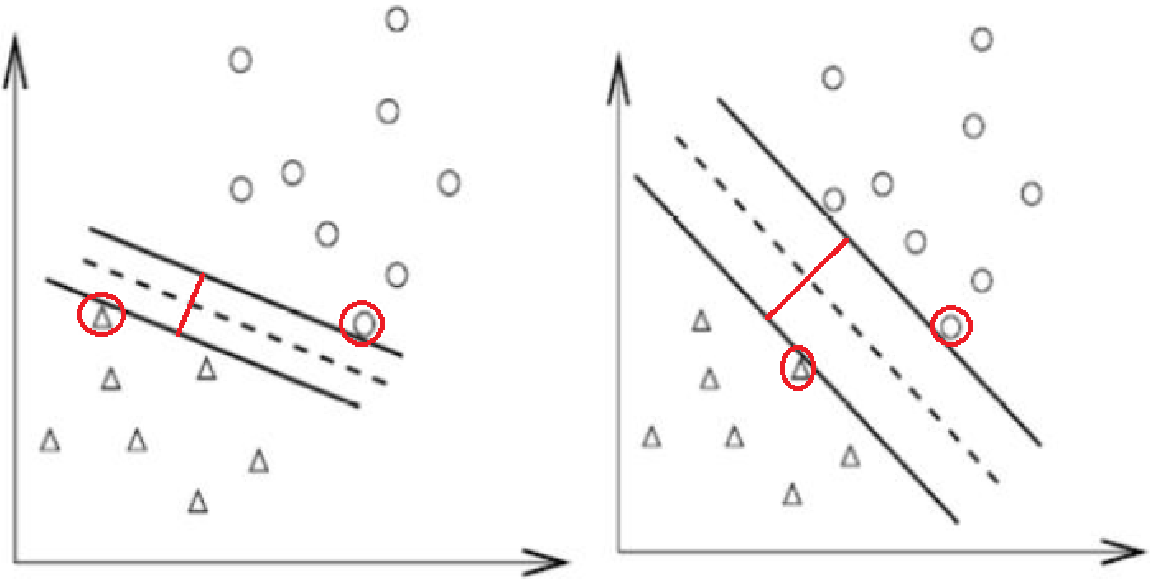
SVM心路历程

1. SVM原理推导

1.1. 定义任务

任务：对线性可分的二分类问题，寻找一条直线，对于距离该直线最近的正负样本点，使得它们到直线的距离相等，且它们之间的距离最大。**即寻找最宽的路，将正负样本分开。**

如下图所示，右侧的决策边界要优于左侧决策边界。



设$\overrightarrow{w}$为决策边界的法向量，C为原点到决策边界的距离，则决策函数为：

$$\begin{cases} \overrightarrow{w} \cdot \overrightarrow{x} > C, then+ \\ \overrightarrow{w} \cdot \overrightarrow{x} < C, then- \end{cases} \tag{27}$$

稍作化简，可得：

$$\begin{cases} \overrightarrow{w} \cdot \overrightarrow{x} + b > 0, then+ \\ \overrightarrow{w} \cdot \overrightarrow{x} + b < 0, then- \end{cases} \tag{28}$$

对y做如下定义：

$$y = \begin{cases} 1, + \\ -1, - \end{cases} \tag{29}$$

则决策函数应该满足：

$$y(\overrightarrow{w} \cdot \overrightarrow{x} + b) > 0 \tag{30}$$

1.2. 决策函数标准化

对决策函数加入距离限制，比如要求最近的点到决策边界的函数间隔至少为1，则决策函数：

$$\begin{gathered} y(\overrightarrow{w} \cdot \overrightarrow{x} + b) \geqslant 1 \\ x \in \text{训练集} \end{gathered} \tag{31}$$

含义：正负样本点距离决策边界的最近距离为1。对于等于1的点，我们称之为支持向量。

1.3. 计算街宽

$$\begin{aligned} WIDTH &= (\overrightarrow{x_+} - \overrightarrow{x_-}) \cdot \frac{\overrightarrow{w}}{|\overrightarrow{w}|} \quad \overrightarrow{x_+}\text{和}\overrightarrow{x_-}\text{均为支持向量} \\ &= \frac{2}{|\overrightarrow{w}|} \end{aligned} \tag{32}$$

1.4. 定义优化目标

$$\begin{gathered} max\frac{2}{|\overrightarrow{w}|} \\ \Rightarrow min|\overrightarrow{w}| \\ \Rightarrow \begin{cases} & min\frac{1}{2}\left\|\overrightarrow{w}\right\|_2^2 \\ s.t. & y_i(\overrightarrow{w}\cdot\overrightarrow{x_i}+b)\geqslant 1, i=1\ldots N \end{cases} \end{gathered} \tag{33}$$

1.5. 定义拉格朗日函数

$$\begin{gathered} L(\alpha,w,b)=\frac{1}{2}\left\|\overrightarrow{w}\right\|_2^2+\sum_{i=1}^{N}\alpha_i[1-y_i(\overrightarrow{w}\cdot\overrightarrow{x_i}+b)] \\ \overrightarrow{\alpha}=(\alpha_1,\alpha_2,\cdots,\alpha_N)\geq 0 \end{gathered} \tag{34}$$

优化目标是使L最小，使用数值方法，L分别对$\overrightarrow{w}$和b求导：

$$\begin{gathered} \frac{\partial L}{\partial\overrightarrow{w}}=\overrightarrow{w}-\sum_{i=1}^{N}\alpha_i y_i x_i=0 \\ \frac{\partial L}{\partial b}=-\sum_{i=1}^{N}\alpha_i y_i=0 \end{gathered} \tag{35}$$

得$\overrightarrow{w}$的最优解：

$$\begin{gathered} \overrightarrow{w^*}=\sum_{i=1}^{N}\alpha_i y_i x_i \\ \sum_{i=1}^{N}\alpha_i y_i=0 \end{gathered} \tag{36}$$

对b的最优解，可以任意找一个支持向量，根据$y_i(\overrightarrow{w}\cdot\overrightarrow{x_i}+b)=1$进行求解。

至此，求得决策边界：

$$\begin{cases} \sum_{i=1}^{N}\alpha_i y_i(\overrightarrow{x_i}\cdot\overrightarrow{x_{new}})+b>0, then+ \\ \sum_{i=1}^{N}\alpha_i y_i(\overrightarrow{x_i}\cdot\overrightarrow{x_{new}})+b<0, then- \end{cases} \tag{37}$$

1.6. 核方法

$$\begin{gathered} L=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j), x\in \text{训练集} \\ \text{注：由此可见，}L\text{取决于训练集}x_i\cdot x_j \end{gathered} \tag{38}$$

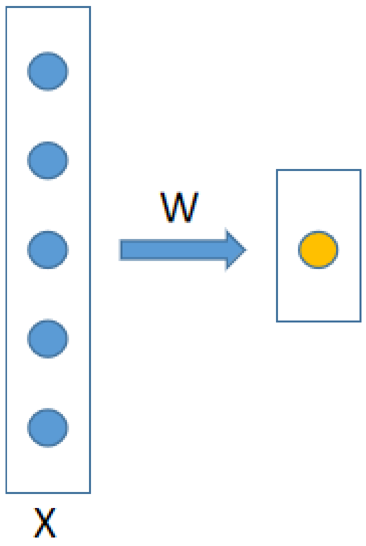
定义：

$$K(x_i,x_j)=\Phi(x_i)\cdot\Phi(x_j) \tag{39}$$

则$K(x_i,x_j)$为核函数，Φ为变换函数。左侧为点乘后的变换，右侧为变换后的点乘，核方法本质是将**变换后的点乘**改为**点乘后的变换**。我们不需要知道如何变换，只需要知道变换后的结果。

1.7. 神经网络

从神经网络的角度理解，SVM的本质是通过全连接变换，采用hinge-loss作为损失函数的分类问题。



hinge-loss（经验风险）为：

$$
\begin{gathered}
L(y\cdot(w\cdot x+b))=[1-y\cdot(w\cdot x+b)]_+ \\
\text{其中，}[z]_+\begin{cases} z, z>0 \\ 0, z<=0 \end{cases}
\end{gathered}
\tag{40}
$$

为了防止过拟合，实际优化目标还会加入正则化项（结构风险）：

$$
\begin{gathered}
min\{[1-y\cdot(w\cdot x+b)]_+ + \lambda||w||^2\}_{w,b} \\
\text{其中，}[z]_+\begin{cases} z, z>0 \\ 0, z<=0 \end{cases}
\end{gathered}
\tag{41}
$$

实现代码为：

```
class LinearSVM(nn.Module):
    def __init__(self):
        super(LinearSVM, self).__init__()
        self.linear = nn.Linear(in_features=2, out_features=1)
    def forward(self, x):
        y = self.linear(x)
        return y

svm = LinearSVM()
optimizer = optim.SGD(svm.parameters(), lr=0.1)

batch_size = 1
epoch_num = 30
N = 500

for epoch in range(1,epoch_num+1):
    for i in range(0, N, batch_size):
        x = torch.Tensor(X[i: i+batch_size])
        y = torch.Tensor(Y[i: i+batch_size])

```

```
        y_pred = svm(x)

        loss = torch.mean(torch.clamp(1 - y_pred * y, min=0))       # hinge loss
        loss += 0.01 * torch.mean(svm.linear.weight ** 2) / 2       # l2 pently

        loss.backward()
        optimizer.step()
        optimizer.zero_grad()

```

2. 对偶问题转化

1.原始问题：

$$
\begin{gathered}
L(\alpha, w, b)=\frac{1}{2}\|\vec{w}\|_2^2+\sum_{i=1}^N \alpha_i[1-y_i(\vec{w}\cdot\vec{x_i}+b)] \\
\vec{\alpha}=(\alpha_1,\alpha_2,\cdots,\alpha_N)\geq 0
\end{gathered} \tag{42}
$$

2.对偶函数：

$$
g(\alpha)=\min_{w,b} L(\alpha, w, b) \tag{43}
$$

3.对偶问题：

$$
\max_{\alpha,\alpha\geq 0} g(\alpha)=\max_{\alpha,\alpha\geq 0}\min_{w,b} L(\alpha, w, b) \tag{44}
$$

4.求解对偶函数：

$$
\begin{gathered}
\nabla_w L(w,b,\alpha)=w-\sum_{i=1}^N \alpha_i y_i x_i=0 \\
\nabla_b L(w,b,\alpha)=-\sum_{i=1}^N \alpha_i y_i=0
\end{gathered} \tag{45}
$$

解得：

$$
\begin{gathered}
w=\sum_{i=1}^N \alpha_i y_i x_i \\
\sum_{i=1}^N \alpha_i y_i=0
\end{gathered} \tag{46}
$$

对$L(\alpha, w, b)$化简得：

$$
\begin{aligned}
L(w,b,\alpha) &= \frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N \alpha_i\alpha_j y_i y_j\left(x_i\cdot x_j\right)-\sum_{i=1}^N \alpha_i y_i\left(\left(\left(\sum_{j=1}^N \alpha_j y_j x_j\right)\cdot x_i+b\right)\right)+\sum_{i=1}^N \alpha_i \\
&= -\frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N \alpha_i\alpha_j y_i y_j\left(x_i\cdot x_j\right)+\sum_{i=1}^N \alpha_i
\end{aligned} \tag{47}
$$

所以，对偶函数为:

$$g(\alpha)=\min_{w,b} L(\alpha, w, b)=-\frac{1}{2} \sum_{i=1} \sum_{j=1} \alpha_i \alpha_j y_i y_j\left(x_i \cdot x_j\right)+\sum_{i=1} \alpha_i \tag{48}$$

5.对偶函数极大化，即将原问题转化为对偶问题：

$$\begin{array}{ll}\max\limits_{\alpha} & -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j\left(x_i \cdot x_j\right)+\sum_{i=1}^N \alpha_i \\ \text{s.t.} & \sum_{i=1}^N \alpha_i y_i=0 \\ & \alpha_i \geqslant 0, \quad i=1,2, \cdots, N\end{array} \tag{49}$$

将问题改为等价的 min 问题：

$$\begin{array}{ll}\min\limits_{\alpha} & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j\left(x_i \cdot x_j\right)-\sum_{i=1}^N \alpha_i \\ \text{s.t.} & \sum_{i=1}^N \alpha_i y_i=0 \\ & \alpha_i \geqslant 0, \quad i=1,2, \cdots, N\end{array} \tag{50}$$

3. SMO算法

待补充。

4. 附录

4.1. 二范数求导

$$\left\|\overrightarrow{w}\right\|_2^2=\overrightarrow{w}^T \cdot \overrightarrow{w}=w_1^2+w_2^2+\ldots+w_n^2 \tag{51}$$

$$\frac{\partial \overrightarrow{w}^T \cdot \overrightarrow{w}}{\partial \overrightarrow{w}}=2 \overrightarrow{w} \tag{52}$$

解释：$w_1^2+w_2^2+\ldots+w_n^2$分别对$\overrightarrow{w}$中的每个变量求偏导。