

# Machine Learning - P35

逻辑回归

对于训练集  $T = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_N, y_N)\}$

其中输入空间为  $D$  维实数列向量  $\vec{x} \in \mathbb{R}^D$ , 轮出空间为二分类标记  $y_i \in \{0, 1\}$

有权重向量为  $D$  维实数列向量  $\vec{w} = (w_1, w_2, \dots, w_D)^T$

有负对数似然函数  $NLL(\vec{w}) = -\frac{1}{N} \sum_{i=1}^N [y_i \ln G(\vec{w}^T \vec{x}_i) + (1-y_i) \ln (1-G(\vec{w}^T \vec{x}_i))]$

梯度函数  $g(\vec{w}) = \frac{1}{N} \sum_{i=1}^N (G(\vec{w}^T \vec{x}_i) - y_i) \vec{x}_i = \frac{1}{N} \vec{x}^T (\vec{\mu} - \vec{y})$

其中  $N \times D$  矩阵  $\vec{x} = (\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N)^T$ ,  $N$  维列向量  $\vec{y} = (y_1, y_2, \dots, y_N)^T$

$N$  维列向量  $\vec{\mu} = (\mu_1, \mu_2, \dots, \mu_N)^T = (G(\vec{w}^T \vec{x}_1), G(\vec{w}^T \vec{x}_2), \dots, G(\vec{w}^T \vec{x}_N))^T$

再对  $g(\vec{w})^T$  求  $\vec{w}$  的导数可以得到 Hessian 矩阵

$$\begin{aligned} \text{有 } \bullet \bullet \bullet \text{ } DXD \text{ 矩阵 } H &= \frac{d}{d\vec{w}} g(\vec{w})^T = \frac{d}{d\vec{w}} \frac{1}{N} \sum_{i=1}^N (G(\vec{w}^T \vec{x}_i) - y_i) \vec{x}_i^T \\ &= \frac{d}{d\vec{w}} \frac{1}{N} \sum_{i=1}^N G(\vec{w}^T \vec{x}_i) \vec{x}_i^T \\ &= \frac{1}{N} \sum_{i=1}^N G(\vec{w}^T \vec{x}_i) (1 - G(\vec{w}^T \vec{x}_i)) \vec{x}_i \vec{x}_i^T \\ &= \vec{x}^T S \vec{x} \end{aligned}$$

其中  $S$  为  $N \times N$  对角矩阵

$$S = \text{diag}\left(\frac{1}{N} G(\vec{w}^T \vec{x}_i) (1 - G(\vec{w}^T \vec{x}_i))\right) = \begin{bmatrix} \frac{1}{N} \mu_1 (1 - \mu_1) & & \\ & \ddots & \\ & & \frac{1}{N} \mu_N (1 - \mu_N) \end{bmatrix}$$

又对于任意  $i = 1, 2, \dots, N$ ,  $0 < \frac{1}{N} \mu_i (1 - \mu_i) < 1$

则对于任意非零向量  $\vec{v} = (v_1, v_2, \dots, v_D)^T \in \mathbb{R}^D$

$$\text{有 } \vec{v}^T H \vec{v} = \vec{v}^T \vec{x}^T S \vec{x} v = (\vec{x} \vec{v})^T S (\vec{x} \vec{v})$$

$$= \sum_{i=1}^N \frac{1}{N} \mu_i (1 - \mu_i) \|\vec{x}_i \vec{v}\|_2^2 > 0$$

于是可知 Hessian 矩阵  $H$  是正定的 (positive definite)

与线性回归中的岭回归类似地在逻辑回归中加入  $L_2$  regularization

用于防止在拟合的过程中出现  $\|\vec{w}\|$  过大和过小的情况

在负对数似然函数中加入惩罚项  $\frac{\lambda}{2N} \|\vec{w}\|_2^2$

其中  $\lambda$  为正则化参数 (regularization parameter)

$$\text{则有 } NLL'(\vec{w}) = -\frac{1}{N} \sum_{i=1}^N [y_i \ln G(\vec{w}^T \vec{x}_i) + (1-y_i) \ln (1-G(\vec{w}^T \vec{x}_i))] + \frac{\lambda}{2N} \|\vec{w}\|_2^2$$

$$= -\frac{1}{N} \sum_{i=1}^N [y_i \ln G(\vec{w}^T \vec{x}_i) + (1-y_i) \ln (1-G(\vec{w}^T \vec{x}_i))] + \frac{\lambda}{2N} \vec{w}^T \vec{w}$$

于是有梯度函数  $g'(\vec{w}) = \frac{d}{d\vec{w}} NLL'(\vec{w})$

$$= \frac{1}{N} \sum_{i=1}^N (G(\vec{w}^T \vec{x}_i) - y_i) \vec{x}_i + \frac{\lambda}{N} \vec{w}$$

以及 Hessian 矩阵  $H = \frac{d}{d\vec{w}} g'(\vec{w}) = \vec{x}^T S \vec{x} + \frac{\lambda}{N} I_D$

其中  $I_D$  为  $D \times D$  的单位矩阵

# Machine Learning - P36

逻辑回归

对于训练集  $T = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_N, y_N)\}$

其中输入空间为  $D$  维实数列向量  $\vec{x} \in \mathbb{R}^D$

输出空间为分类标记集合  $y_i \in \{1, 2, \dots, C\}$

对于分类  $Y=C$ , 有权重向量  $\vec{w}_c$

为  $D$  维实数列向量  $\vec{w}_c = (w_{c1}, w_{c2}, \dots, w_{cD})^T$

于是有  $D \times C$  的权重矩阵  $W = (\vec{w}_1, \vec{w}_2, \dots, \vec{w}_C)$

又有分类偏好 (class biases) 为  $C$  维实数列向量  $\vec{w}_0 = (w_{01}, w_{02}, \dots, w_{0C})^T$

则有  $C$  维实数列向量  $\vec{\eta} = (\eta_1, \eta_2, \dots, \eta_C)^T$

$$= (w_{01} + \vec{w}_1^T \vec{x}, w_{02} + \vec{w}_2^T \vec{x}, \dots, w_{0C} + \vec{w}_C^T \vec{x})^T \\ = \vec{w}_0 + W^T \vec{x}$$

通过归一化指数函数将向量  $\vec{\eta}$  转化为概率

$$\text{则有 } P(Y=c | \vec{x}, \vec{w}) = \exp(\vec{w}_{0c} + \vec{w}_c^T \vec{x}) / \sum_{c=1}^C \exp(\vec{w}_{0c} + \vec{w}_c^T \vec{x})$$

令  $\eta_i = \vec{w}_0 + W^T \vec{x}_i$ , 对应于 softmax 函数  $S(\vec{\eta}_i)$

$$\mu_{ic} = P(Y=c | \vec{x}_i, \vec{w}) = S(\vec{\eta}_i)^{(c)} \text{ 对应 } C \text{ 维实数列向量 } \vec{\mu}_i = (\mu_{i1}, \dots, \mu_{iC})^T$$

$$y_{ic} = I(y_i = c) \text{ 其中 } c=1, 2, \dots, C, \text{ 对应 } C \text{ 维实数列向量 } \vec{y}_i = (y_{i1}, \dots, y_{iC})^T$$

注意  $\vec{y}_i$  是 one-hot 向量, 即仅有对应于  $y_i$  的元素为 1, 其余元素为 0

于是有负对数似然函数, 其中  $P(y_i | \vec{x}_i, \vec{w}) = \prod_{c=1}^C \mu_{ic}^{y_{ic}}$

$$NLL(W) = -\ln \prod_{i=1}^N P(y_i | \vec{x}_i, \vec{w}) = -\ln \prod_{i=1}^N \prod_{c=1}^C \mu_{ic}^{y_{ic}}$$

$$= -\sum_{i=1}^N \sum_{c=1}^C y_{ic} \ln \mu_{ic}$$

$$= -\sum_{i=1}^N \sum_{c=1}^C y_{ic} \ln [\exp(\eta_i^{(c)}) / \sum_{c'=1}^C \exp(\eta_i^{(c')})]$$

$$= -\sum_{i=1}^N \left[ \left( \sum_{c=1}^C y_{ic} (\vec{w}_{0c} + \vec{w}_c^T \vec{x}_i) \right) - \ln \left[ \sum_{c=1}^C \exp(\vec{w}_{0c} + \vec{w}_c^T \vec{x}_i) \right] \right]$$

再对  $NLL(W)$  分别求权重向量  $\vec{w}_c$  的导数

$$\text{则有 } \frac{\partial}{\partial \vec{w}_c} NLL(W) = -\sum_{i=1}^N [y_{ic} \cdot \vec{x}_i - \frac{1}{\sum_{c'=1}^C \exp(\vec{w}_{0c'} + \vec{w}_{c'}^T \vec{x}_i)} \cdot \exp(\vec{w}_{0c} + \vec{w}_c^T \vec{x}_i) \vec{x}_i]$$

$$= -\sum_{i=1}^N [y_{ic} \cdot \vec{x}_i - \mu_{ic} \cdot \vec{x}_i]$$

$$= \sum_{i=1}^N (\mu_{ic} - y_{ic}) \vec{x}_i$$

取克罗内克积 (Kronecker product), 对于  $m \times n$  矩阵  $A$  和  $p \times q$  矩阵  $B$

$$\text{则有 } m \times p \times n \times q \text{ 矩阵 } A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix}$$

$$\text{于是有梯度函数 } g(W) = \sum_{i=1}^N (\vec{\mu}_i - \vec{y}_i)^T \otimes \vec{x}_i$$

$$\text{注意这个矩阵 } = \left[ \sum_{i=1}^N (\mu_{i1} - y_{i1}) \vec{x}_i, \sum_{i=1}^N (\mu_{i2} - y_{i2}) \vec{x}_i, \dots, \sum_{i=1}^N (\mu_{iC} - y_{iC}) \vec{x}_i \right]$$

$$\text{也是 } D \times C \text{ 矩阵} = \left[ \sum_{i=1}^N (\mu_{i1} - y_{i1}) \vec{x}_i^{(1)}, \sum_{i=1}^N (\mu_{i2} - y_{i2}) \vec{x}_i^{(1)}, \dots, \sum_{i=1}^N (\mu_{iC} - y_{iC}) \vec{x}_i^{(1)} \right]$$

于是有梯度下降法  $W_{k+1} = W_k - \alpha g(W)$

$$\left[ \sum_{i=1}^N (\mu_{i1} - y_{i1}) \vec{x}_i^{(0)}, \sum_{i=1}^N (\mu_{i2} - y_{i2}) \vec{x}_i^{(0)}, \dots, \sum_{i=1}^N (\mu_{iC} - y_{iC}) \vec{x}_i^{(0)} \right]$$

# Machine

## Learning - P37

PCA

(principal component analysis, 主成分分析), 非监督学习算法

目的是在尽可能好地代表原特征的情况下,

对原特征进行线性变换, 映射至低维度空间中

对于训练集  $T = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N\}$

其中 特征空间为  $D$  维实数列向量  $\vec{x} \in \mathbb{R}^D$

考虑将样本特征映射到  $M$  维子空间, 其中  $M < D$

则可以使用  $D \times M$  的矩阵  $U$

于是  $U^T \vec{x}$  成为一个  $M$  维实数列向量

特别地当  $M=1$  时, 有  $D$  维实数列向量  $\vec{u}$

有  $\vec{u}^T \vec{x}$  为一个标量值

假设希望通过一个点,  $\vec{x}_0$  来代表所有样本, 则希望这个点到所有样本的距离之和最小

$$\begin{aligned} \text{再取 } \vec{\mu} = \frac{1}{N} \sum_{i=1}^N \vec{x}_i, \text{ 则 } f(\vec{x}_0) &= \sum_{i=1}^N \|\vec{x}_0 - \vec{x}_i\|_2^2 = \sum_{i=1}^N \|(\vec{x}_0 - \vec{\mu}) - (\vec{x}_i - \vec{\mu})\|_2^2 \\ &= \sum_{i=1}^N \|\vec{x}_0 - \vec{\mu}\|_2^2 - 2 \sum_{i=1}^N (\vec{x}_0 - \vec{\mu})^T (\vec{x}_i - \vec{\mu}) + \sum_{i=1}^N \|\vec{x}_i - \vec{\mu}\|_2^2 \\ &= \sum_{i=1}^N \|\vec{x}_0 - \vec{\mu}\|_2^2 - 2(\vec{x}_0 - \vec{\mu})^T \sum_{i=1}^N (\vec{x}_i - \vec{\mu}) + \sum_{i=1}^N \|\vec{x}_i - \vec{\mu}\|_2^2 \end{aligned}$$

$$\text{又 } \sum_{i=1}^N (\vec{x}_i - \vec{\mu}) = \sum_{i=1}^N \vec{x}_i - N \cdot \vec{\mu} = \vec{0} \quad = N \|\vec{x}_0 - \vec{\mu}\|_2^2 + \sum_{i=1}^N \|\vec{x}_i - \vec{\mu}\|_2^2$$

于是当  $\vec{x}_0 = \vec{\mu} = \frac{1}{N} \sum_{i=1}^N \vec{x}_i$  时,  $f(\vec{x}_0)$  取得最小值  $\sum_{i=1}^N \|\vec{x}_i - \vec{\mu}\|_2^2$

则当希望压缩至一维时, 即有一条直线, 取所有样本到此直线的投影

由于特殊的 0 维(点)的情形应包含于这条直线,

则这条直线可以表示为  $\vec{y} = \vec{\mu} + \alpha \vec{u}$

其中  $\vec{u}$  为该直线方向的单位向量, 即  $\|\vec{u}\|_2^2 = \vec{u}^T \vec{u} = 1$

$\alpha$  为样本  $\vec{x}_i$  在直线上的投影与  $\vec{\mu}$  的距离, 即  $(\vec{x}_i - \vec{\mu})$  在  $\vec{u}$  上的投影

$$\text{于是有 } \alpha = \|\vec{x}_i - \vec{\mu}\|_2 \cos \theta = \|\vec{x}_i - \vec{\mu}\|_2 \|\vec{u}\|_2 \cos(\vec{x}_i - \vec{\mu}, \vec{u})$$

$$= \vec{u}^T (\vec{x}_i - \vec{\mu})$$

可知对于任意单位向量  $D$  维实数列向量  $\vec{u} = (u_1, u_2, \dots, u_D)^T$

对每一个样本特征  $\vec{x}_i$ , 都有一个与  $\vec{u}$  相关的  $\alpha_i$  表示  $\vec{x}_i$

我们希望有一条直线方向为  $\vec{u}$ , 使得样本在其上的投影尽可能分散

$$\text{即 } \vec{u} \text{ 取得 } \operatorname{argmax}_{\vec{u}} \sum_{i=1}^N \alpha_i^2 = \operatorname{argmax}_{\vec{u}} \sum_{i=1}^N [\vec{u}^T (\vec{x}_i - \vec{\mu})]^2$$

由于不同的特征具有不同的方差, 则可以先对训练集的数据进行归一化

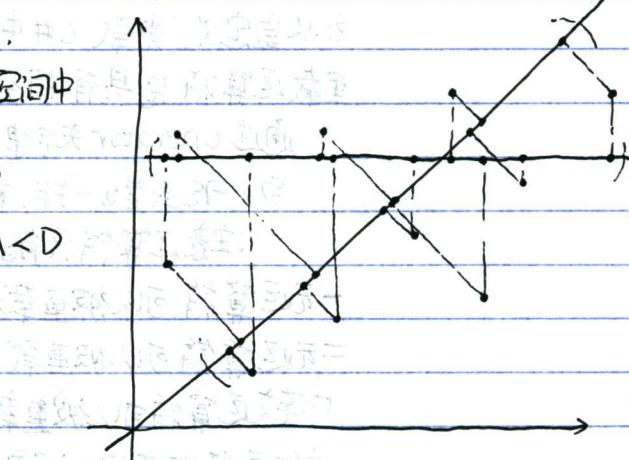
$$1. \vec{\mu}_j := \frac{1}{N} \sum_{i=1}^N \vec{x}_i \quad (\vec{\mu}_j = \frac{1}{N} \sum_{i=1}^N \vec{x}_i^{(j)})$$

$$2. \vec{x}_i := \vec{x}_i - \vec{\mu} \quad (\vec{x}_i^{(j)} := \vec{x}_i^{(j)} - \vec{\mu}^{(j)})$$

$$3. \vec{s}^{(j)} := \frac{1}{N} \sum_{i=1}^N (\vec{x}_i^{(j)})^2, \text{ 此时 } \vec{x}_i \text{ 已替换为 } \vec{x}_i - \vec{\mu}$$

$$4. \vec{x}_i^{(j)} := \vec{x}_i^{(j)} / \vec{s}^{(j)}$$

完成后, 样本在样本空间中的均值移动到原点, 即  $\vec{\mu} = \vec{0}$ , 以方便计算



# Machine

## Learning - P38

PCA

最大方差理论

对训练集  $T = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n\}$ , 其中特征空间为  $D$  维实数列向量  $\vec{x} \in \mathbb{R}^D$

希望将特征映射到  $k < D$  维子空间

并最大限度地代表原有特征

希望将样本投影到某一维的直线上

取一条过原点的直线, 有方向向量  $\vec{u}$

其中方向向量  $\vec{u}$  为  $D$  维列向量

希望在投影之后, 样本在直线上的方差尽可能大

已知  $\vec{u}^\top \vec{x}_i$  表示样本在直线上投影与原点的距离

且这个距离与  $\vec{u}$  方向相同时为正, 相反时为负

$$\bar{u} = \frac{1}{N} \sum_{i=1}^N \vec{x}_i, \text{ 且 } \vec{u}^\top \bar{u} = \frac{1}{N} \sum_{i=1}^N \vec{u}^\top \vec{x}_i$$

于是有样本投影方差为  $\frac{1}{N} \sum_{i=1}^N [\vec{u}^\top \vec{x}_i - \bar{u}^\top \bar{u}]^2$

$$\begin{aligned} \text{于是有目标函数 } J(\vec{u}) &= \frac{1}{N} \sum_{i=1}^N [\vec{u}^\top (\vec{x}_i - \bar{u})]^2 \\ &= \vec{u}^\top \left[ \frac{1}{N} \sum_{i=1}^N (\vec{x}_i - \bar{u})(\vec{x}_i - \bar{u})^\top \right] \vec{u} \end{aligned}$$

$$\text{又有样本的协方差矩阵 } \Sigma = \frac{1}{N} \sum_{i=1}^N (\vec{x}_i - \bar{u})(\vec{x}_i - \bar{u})^\top$$

$$\text{于是有 } J(\vec{u}) = \vec{u}^\top \Sigma \vec{u}$$

如果限制  $\vec{u}$  为单位向量, 即  $\vec{u}^\top \vec{u} = 1$

则对拉格朗日函数  $L(\vec{u}) = \vec{u}^\top \Sigma \vec{u} - \lambda(\vec{u}^\top \vec{u} - 1)$ , 其中  $\lambda$  为实数

$$\text{则令导数 } L'(\vec{u}) = 2 \Sigma \vec{u} - 2\lambda \vec{u} = 0$$

$$\text{即有 } \Sigma \vec{u} = \lambda \vec{u}$$

于是有  $\lambda$  为协方差矩阵的特征值

而  $\vec{u}$  为协方差矩阵的特征向量

而由于  $J(\vec{u})$  是标量, 且有  $\vec{u}^\top \vec{u} = 1$

$$\text{则有 } J(\vec{u}) \cdot \vec{u} = \vec{u} \cdot J(\vec{u}) = \vec{u}^\top \Sigma \vec{u} = \Sigma \vec{u}$$

$$\text{即有 } \Sigma \vec{u} = J(\vec{u}) \vec{u}$$

于是可知  $J(\vec{u})$  的最大值即协方差矩阵特征值最大的, 并对应于特征向量  $\vec{u}$ ,

而令  $J(\vec{u})$  取得第二大值即对应协方差矩阵中值第二大的特征向量  $\vec{u}_2$

于是可以取对应前  $k$  大的协方差矩阵特征值的特征向量  $\vec{u}_1, \vec{u}_2, \dots, \vec{u}_k$

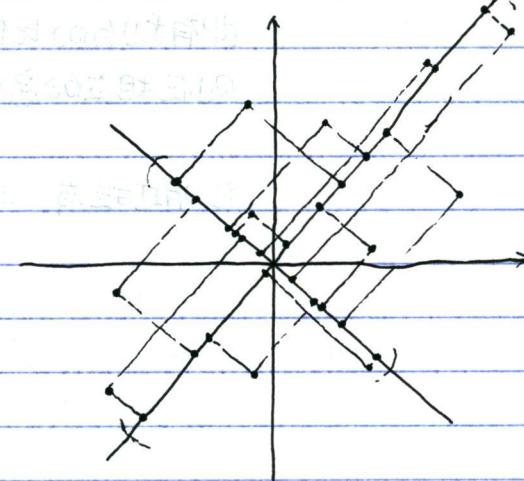
且  $\vec{u}_1, \vec{u}_2, \dots, \vec{u}_k$  是正交的

构造  $D \times k$  矩阵  $U = (\vec{u}_1, \vec{u}_2, \dots, \vec{u}_k)$

则映射到  $k$  维子空间的样本为  $\vec{y}_i = U^\top \vec{x}_i = (\vec{u}_1^\top \vec{x}_i, \dots, \vec{u}_k^\top \vec{x}_i)^\top$  为  $k$  维列向量

选取的特征值占所有特征值和的比率为贡献率

$$\text{即 } \sum_{i=1}^k \lambda_i / \sum_{i=1}^D \lambda_i, \text{ 也有使用 } \sum_{i=1}^k \lambda_i / \sum_{i=1}^D \lambda_i$$



# Machine

## Learning - P39

PCA

对于训练集  $T = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n\}$ , 其中特征空间为  $D$  维实数列向量  $\vec{X} \in \mathbb{R}^D$

如果选取一点  $\vec{x}_0$  “代表”所有样本

即使得  $J_0(\vec{x}_0) = \sum_{i=1}^N \|\vec{x}_0 - \vec{x}_i\|_2^2$  最小化

其中  $J_0(\vec{x}_0)$  称为平方误差评价函数 (squared-error criterion function)

可知当  $\vec{x}_0 = \frac{1}{N} \sum_{i=1}^N \vec{x}_i = \bar{\mu}$  时,  $J_0(\vec{x}_0)$  取得最小值

假定过  $\bar{\mu}$  取一直线, 并在其上任取一点  $\vec{x}'_i$  来“代表”  $\vec{x}_i$

令  $\alpha_i$  表示直线上  $\vec{x}'_i$  沿方向向量到  $\bar{\mu}$  的距离, 其中方向向量为  $\vec{u}$

则有  $\vec{x}'_i = \bar{\mu} + \alpha_i \vec{u}$

于是有目标函数  $J(\alpha_1, \dots, \alpha_N, \vec{u}) = \sum_{i=1}^N \|\bar{\mu} + \alpha_i \vec{u} - \vec{x}_i\|_2^2$

$$= \sum_{i=1}^N \|\alpha_i \vec{u} - (\vec{x}_i - \bar{\mu})\|_2^2$$

$$= \sum_{i=1}^N \alpha_i^2 \|\vec{u}\|_2^2 - 2 \sum_{i=1}^N \alpha_i \vec{u}^T (\vec{x}_i - \bar{\mu}) + \sum_{i=1}^N \|\vec{x}_i - \bar{\mu}\|_2^2$$

令  $\vec{u}$  为单位向量, 即  $\|\vec{u}\|_2^2 = \vec{u}^T \vec{u} = 1$ , 并分别对  $\alpha_i$  求偏导数,  $i=1, 2, \dots, N$

$$\frac{\partial}{\partial \alpha_i} J(\alpha_1, \dots, \alpha_N, \vec{u}) = 2\alpha_i - 2\vec{u}^T (\vec{x}_i - \bar{\mu}) = 0$$

则有  $\alpha_i = \vec{u}^T (\vec{x}_i - \bar{\mu})$ , 代入  $J(\alpha_1, \dots, \alpha_N, \vec{u})$

$$\text{则有 } J(\vec{u}) = \sum_{i=1}^N \alpha_i^2 - 2 \sum_{i=1}^N \alpha_i + \sum_{i=1}^N \|\vec{x}_i - \bar{\mu}\|_2^2$$

$$= - \sum_{i=1}^N \vec{u}^T (\vec{x}_i - \bar{\mu}) (\vec{x}_i - \bar{\mu})^T \vec{u} + \sum_{i=1}^N \|\vec{x}_i - \bar{\mu}\|_2^2$$

其中散列矩阵  $S = \sum_{i=1}^N (\vec{x}_i - \bar{\mu})(\vec{x}_i - \bar{\mu})^T$

取  $N \times D$  矩阵  $X = (\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N)^T$

而对于矩阵  $X$  进行中心化 (centralize)

则有  $N \times D$  矩阵  $X_c = (\vec{x}_1 - \bar{\mu}, \vec{x}_2 - \bar{\mu}, \dots, \vec{x}_N - \bar{\mu})^T$

于是有  $S = \sum_{i=1}^N (\vec{x}_i - \bar{\mu})(\vec{x}_i - \bar{\mu})^T = X_c^T X_c$

取拉格朗日函数  $L(\vec{u}) = -\vec{u}^T S \vec{u} + \lambda (\vec{u}^T \vec{u} - 1)$

令  $L'(\vec{u}) = -2S\vec{u} + 2\lambda\vec{u} = 0$ , 即有  $S\vec{u} = \lambda\vec{u}$

于是可知  $\vec{u}$  对应于散列矩阵  $S = X_c^T X_c$  的特征向量

当对于  $X_c$  进行奇异值分解则有  $X_c = U \Sigma V^T$

其中  $D \times D$  矩阵  $V$  对角线上的元素为  $X_c$  特征值的平方根

$D \times D$  矩阵  $V$  为  $X_c^T X_c$  的单位化特征向量 ( $\vec{u}_1, \vec{u}_2, \dots, \vec{u}_D$ )

且可以认为所对应特征值按降序排列

取  $D \times D$  矩阵的前  $k < D$  列可以组成一组规范正交基

即有  $D \times k$  矩阵  $V_k = (\vec{u}_1, \vec{u}_2, \dots, \vec{u}_k)$

则有样本到  $k$  维子空间的投影  $N \times k$  矩阵  $X_{ck} = X_c V_k$

也可以进行重构  $N \times D$  矩阵  $X'_c = X_{ck} V_k^T$

# Machine

## Learning - P40

核函数

(kernel function), 对于抽象空间 (abstract space)  $X$

核函数  $K(\bar{x}, \bar{x}')$  为定义在  $X \times X$  的实数函数, 即  $K: X \times X \rightarrow \mathbb{R}$

对于参数  $\bar{x}, \bar{x}' \in X$ ,  $K(\bar{x}, \bar{x}')$  用于衡量  $\bar{x}, \bar{x}'$  之间的相似性

some measure of similarity between object  $x, x' \in X$

通常核函数满足非负性 (non-negative)

即  $\forall x, x' \in X \quad K(x, x') \geq 0$

以及对称性 (symmetric)

即  $\forall x, x' \in X \quad K(x, x') = K(x', x)$

对于抽象空间为  $D$  维欧几里得空间, 即对象  $\bar{x} \in \mathbb{R}^D$

最简单的核函数为线性核 (linear kernel)

即有  $K(\bar{x}, \bar{x}') = \bar{x}^T \bar{x}'$

定义 对于输入序列  $\{\bar{x}_i\}_{i=1}^N \subseteq X$ , 有  $N \times N$  的 Gram matrix

$$\text{即 } K = \begin{bmatrix} K(x_1, x_1) & K(x_1, x_2) & \cdots & K(x_1, x_N) \\ K(x_2, x_1) & K(x_2, x_2) & \cdots & K(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ K(x_N, x_1) & K(x_N, x_2) & \cdots & K(x_N, x_N) \end{bmatrix}$$

如果对于任意输入  $\{\bar{x}_i\}_{i=1}^N \subseteq X$ , 都有矩阵  $K$  是正定的 (positive definite)

则称核函数  $K(x, x')$  为 Mercer kernel, 或 positive definite kernel

再根据 Mercer's theorem, 当核函数  $K(x, x')$  为 Mercer kernel 时

可以对矩阵  $K$  进行特征值分解 (eigenvalue decomposition)

则有  $K = U^T \Delta U$ , 其中对角矩阵  $\Delta$  中特征值  $\lambda_i > 0$

令  $k_{ij}$  表示矩阵  $K$  中第  $i$  行第  $j$  列的元素

则  $K = U^T \Delta U = U^T (\Delta^{\frac{1}{2}} \Delta^{\frac{1}{2}}) U = (\Delta^{\frac{1}{2}} U)^T (\Delta^{\frac{1}{2}} U)$

令函数  $\phi(x_i)$  表示矩阵  $(\Delta^{\frac{1}{2}} U)$  的第  $i$  列, 即  $\Delta^{\frac{1}{2}} U[:, i]$

则有  $k_{ij} = (\Delta^{\frac{1}{2}} U[:, i])^T (\Delta^{\frac{1}{2}} U[:, j])$

$= \phi(x_i)^T \cdot \phi(x_j) = K(x_i, x_j)$

即对于任意核函数  $K(x_i, x_j)$ , 如果  $K(x_i, x_j)$  是 Mercer Kernel

则存在从抽象空间  $X$  到  $D$  维欧几里得空间  $\mathbb{R}^D$  的函数  $\phi(x)$ ,  $x \in X$

有  $\phi: X \rightarrow \mathbb{R}^D$ , 且  $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$

其中  $\phi$  depend on eigenfunction of  $K(x_i, x_j)$

implicitly defined by eigenvectors  $U$  of Gram matrix  $K$

$D$  potentially infinite dimensional space

# Machine

## Learning - P41

### 核函数

注意 Mercer condition 也定义为对于核函数  $K(x, x')$ ,  $x, x' \in X$

Gram matrix  $K = [K(x_1, x_1) \dots K(x_1, x_N) \dots K(x_N, x_1) \dots K(x_N, x_N)]$  是半正定的 (positive semidefinite)

即对矩阵  $K$  进行特征值分解, 有  $K = U \Lambda U^T$

其中对角矩阵  $\Lambda$  中有矩阵  $K$  的特征值  $\lambda_i \geq 0$ ,  $i = 1, 2, \dots, N$

则令  $k_{ij}$  为矩阵  $K$  中第  $i$  行第  $j$  列的元素, 即  $k_{ij} = K(x_i, x_j)$

令  $(\Delta^{\frac{1}{2}} U)[:, i]$  表示矩阵  $\Delta^{\frac{1}{2}} U$  的第  $i$  行

则有  $k_{ij} = (\Delta^{\frac{1}{2}} U[:, i])^T (\Delta^{\frac{1}{2}} U[:, j])$

再令  $\phi(x_i) = \Delta^{\frac{1}{2}} U[:, i]$ ,  $\phi: X \rightarrow \mathbb{R}^d$

则有核函数  $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$

如对于多项式核 (polynomial kernel),  $K: X \times X \rightarrow \mathbb{R}$ , 其中  $X \subseteq \mathbb{R}^d$

$K(\vec{x}, \vec{x}') = (\gamma \vec{x}^T \vec{x}' + \gamma)^M$ , 其中  $\gamma, \gamma, M$  为实数

如取  $X \subseteq \mathbb{R}^2$ ,  $\gamma = \gamma = 1$ ,  $M = 2$

$$K(\vec{x}, \vec{x}') = (\vec{x}^T \vec{x}' + 1)^2 = (1 + \vec{x}_1 \vec{x}'_1 + \vec{x}_2 \vec{x}'_2)^2$$

$$= 1 + 2\vec{x}_1 \vec{x}'_1 + 2\vec{x}_2 \vec{x}'_2 + (\vec{x}_1 \vec{x}'_1)^2 + (\vec{x}_2 \vec{x}'_2)^2 + \vec{x}_1 \vec{x}'_1 \vec{x}_2 \vec{x}'_2$$

于是有  $\phi(\vec{x}) : \mathbb{R}^2 \rightarrow \mathbb{R}^6$

$$\phi(\vec{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, x_1 x_2)^T$$

$$\text{则有 } K(\vec{x}, \vec{x}') = \phi(\vec{x})^T \phi(\vec{x}')$$

Sigmoid kernel, S型的核函数, 用于“激活函数”

$$K(\vec{x}, \vec{x}') = \tanh(\alpha \vec{x}^T \vec{x}' + c), \text{ 其中 } \vec{x}, \vec{x}' \in \mathbb{R}^d, \alpha, c \text{ 为实数}$$

注意 Sigmoid kernel 的 Gram matrix 不全是正定的

如取  $\alpha = c = 1$ , 则有  $K(\vec{x}, \vec{x}') = \tanh(\vec{x}^T \vec{x}')$

取  $\vec{x} = (1, 1)^T$ ,  $\vec{x}' = (-1, -1)^T$

$$\text{则有 } K = \begin{bmatrix} \tanh(\vec{x}^T \vec{x}) & \tanh(\vec{x}^T \vec{x}') \\ \tanh(\vec{x}'^T \vec{x}) & \tanh(\vec{x}'^T \vec{x}') \end{bmatrix} = \begin{bmatrix} \tanh(2) & \tanh(-2) \\ \tanh(-2) & \tanh(2) \end{bmatrix}$$

双曲正切函数  $\tanh(x) = (e^x - e^{-x}) / (e^x + e^{-x})$  为奇函数

$$\text{于是 } K = \begin{bmatrix} \tanh(2) & -\tanh(2) \\ -\tanh(2) & \tanh(2) \end{bmatrix}$$

$$\text{进行特征值分解则有 } \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 2\tanh(2) & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

其中特征值  $\lambda_1 = 2\tanh(2)$ ,  $\lambda_2 = 0$

# Machine

## Learning - P42

核函数

高斯核 (Gaussian kernel), 也称平方指数核 (squared exponential kernel, SE kernel)

$$K(\vec{x}, \vec{x}') = \exp\left[-\frac{1}{2}(\vec{x}-\vec{x}')^\top \Sigma^{-1}(\vec{x}-\vec{x}')\right]$$

其中  $\vec{x}, \vec{x}' \in \mathbb{R}^d$ ,  $d \times d$  对称矩阵  $\Sigma$  可视为多元正态分布的协方差矩阵

如果矩阵  $\Sigma$  为对角矩阵

$$\text{则 } \Sigma^{-1} = \begin{bmatrix} \frac{1}{6_1^2} & & \\ & \ddots & \\ & & \frac{1}{6_d^2} \end{bmatrix}$$

$$\text{则核函数 } K(\vec{x}, \vec{x}') = \exp\left[-\frac{1}{2} \sum_{j=1}^d \frac{1}{6_j^2} (\vec{x}^{(j)} - \vec{x}'^{(j)})^2\right]$$

其中参数  $6_j$  为特征向量第  $j$  维的 characteristic length scale (特征长度尺度)

特别地当  $6_j \rightarrow \infty$  时, 对应的维即被忽略了

也称为 ARD kernel

如果矩阵  $\Sigma$  是 spherical 的, 即为对角矩阵且对角线上元素均相等

$$\text{则核函数 } K(\vec{x}, \vec{x}') = \exp\left[-\frac{\|\vec{x}-\vec{x}'\|_2^2}{26^2}\right] = \exp\left[-\frac{1}{26^2}(\vec{x}-\vec{x}')^\top(\vec{x}-\vec{x}')\right]$$

其中参数  $6^2$  称为 bandwidth

核函数也称为径向基函数核 (radial basis function, RBF kernel)

对于数据集中的噪音有较好的抗干扰能力

参数决定了函数作用范围, 超过这个范围则数据集的作用“基本消失”

广泛使用, 但核函数的性能对参数十分敏感

指数核 (exponential kernel), 对于  $\vec{x}, \vec{x}' \in \mathbb{R}^d$ , 实数参数  $6^2$

$$K(\vec{x}, \vec{x}') = \exp\left[-\frac{1}{6^2} \|\vec{x}-\vec{x}'\|_2^2\right]$$

相比于高斯核, 对参数的依赖性降低, 但适用范围相对狭窄

拉普拉斯核 (Laplacian kernel), 对于  $\vec{x}, \vec{x}' \in \mathbb{R}^d$ , 实数参数  $6$

$$K(\vec{x}, \vec{x}') = \exp\left[-\frac{1}{6} \|\vec{x}-\vec{x}'\|_1\right]$$

等价于指数核, 但对参数的敏感性提高

ANOVA kernel 适用于多维回归问题, 对于  $\vec{x}, \vec{x}' \in \mathbb{R}^d$ , 参数实数  $6, m, M$

$$K(\vec{x}, \vec{x}') = \prod_{j=1}^d \exp\left[-6(\vec{x}^{(j)} - \vec{x}'^{(j)})^2\right]^m$$

二次有理函数核 (rational quadratic kernel, RQ kernel), 对于  $\vec{x}, \vec{x}' \in \mathbb{R}^d$ , 起参数  $\alpha, l > 0$

$$K(\vec{x}, \vec{x}') = \left(1 + \frac{\|\vec{x}-\vec{x}'\|_2^2}{2\alpha l}\right)^{-\alpha}$$

当  $\alpha \rightarrow \infty$  时, 等价于特征尺度为  $l$  的 RBF 核

$$\text{也定义为 } K(\vec{x}, \vec{x}') = 1 - \frac{\|\vec{x}-\vec{x}'\|_2^2}{(\|\vec{x}-\vec{x}'\|_2^2 + c)}, \text{ 其中 } c \text{ 为实数参数}$$

# Machine

## Learning - P43

核函数 (kernel) 多元二次核 (multiquadric kernel), 非正定核函数, 用于替代二次有理核  
 $K(\vec{x}, \vec{x}') = (\|\vec{x} - \vec{x}'\|_2^2 + c^2)^{-1/2}$ , 其中  $c$  为实数参数

逆多元二次核 (inverse multiquadric kernel), 核相关矩阵似乎不会遇到奇异的情况  
 $K(\vec{x}, \vec{x}') = (\|\vec{x} - \vec{x}'\|_2^2 + c^2)^{-1/2}$ , 其中  $c$  为实数参数

马顿核 (Matern kernel), commonly used in Gaussian process regression

$$K(\vec{x}, \vec{x}') = \frac{2^{1-v}}{\Gamma(v)} \left( \frac{\sqrt{v} \|\vec{x} - \vec{x}'\|_2}{l} \right)^v K_v \left( \frac{\sqrt{v} \|\vec{x} - \vec{x}'\|_2}{l} \right), \text{ 其中超参数 } l > 0, v > 0$$

$K_v$  为修正贝塞尔函数 (modified Bessel function)

由  $K_v$  定义可知, 马顿核是指数函数与多项式函数的乘积  
可导性 / 平滑程度与参数  $v$  相关

当  $v \rightarrow \infty$  时, 马顿核等价于以为特征尺度的 RBF 核

当  $v = \frac{1}{2}$  时, 马顿核简化为  $K(\vec{x}, \vec{x}') = \exp(-\frac{\|\vec{x} - \vec{x}'\|_2}{c})$

Circular kernel: 圆形核

$$K(\vec{x}, \vec{x}') = \frac{2}{\pi} \arccos\left(-\frac{\|\vec{x} - \vec{x}'\|_2}{6}\right) - \frac{2}{\pi} \cdot \frac{\|\vec{x} - \vec{x}'\|_2}{6} \cdot \left(1 - \frac{\|\vec{x} - \vec{x}'\|_2^2}{36}\right)^{1/2}$$

Spherical kernel: Circular kernel 的简化版

$$K(\vec{x}, \vec{x}') = 1 - \frac{3}{2} \cdot \frac{\|\vec{x} - \vec{x}'\|_2}{6} + \frac{1}{2} \cdot \left(\frac{\|\vec{x} - \vec{x}'\|_2}{6}\right)^3$$

Wave kernel: 适用于语音处理场景

$$K(\vec{x}, \vec{x}') = \frac{\theta}{\|\vec{x} - \vec{x}'\|_2} \sin\left(\frac{\|\vec{x} - \vec{x}'\|_2}{\theta}\right)$$

三角函数核 (triangular kernel)

$$K(\vec{x}, \vec{x}') = -\|\vec{x} - \vec{x}'\|_2^d$$

对数核 (log kernel): 经常用于图像分割

$$K(\vec{x}, \vec{x}') = -\log(1 + \|\vec{x} - \vec{x}'\|_2^d)$$

柯西核 (Cauchy kernel), 基于柯西分布

$$K(\vec{x}, \vec{x}') = \frac{1}{\|\vec{x} - \vec{x}'\|_2^2 / 6 + 1}$$

定义域广泛, 可应用于原始维度很高的数据

# Machine

## Learning - P44

支持向量机 (Support vector machine, SVM) 二分类模型

目的是寻找一个超平面将样本进行划分

划分的原则是间隔最大化

并最终转化为一个凸二次规划问题求解

对于训练集  $T = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_N, y_N)\}$

其中输入空间为  $D$  维列向量  $\vec{x}_i \in \mathbb{R}^D$

输出空间为二分类标记  $y_i \in \{-1, 1\}$

有 large-margin separating hyperplane

即找到超平面  $\vec{w}^\top \vec{x} + b = 0$ , 其中  $b$  为实数

$\vec{w}$  为  $D$  维实数列向量  $\vec{w} \in \mathbb{R}^D$

使得这个超平面可以准确地划分训练集中的样本

$$\begin{cases} (\vec{w}^\top \vec{x}_i + b) / \|\vec{w}\| \geq d, & \forall y_i = 1 \\ (\vec{w}^\top \vec{x}_i + b) / \|\vec{w}\| \leq -d, & \forall y_i = -1 \end{cases}$$

并最大化决策面间的距离  $d = \min \text{distance}(\vec{x}_i, \vec{w}, b)$

由于  $y_i \in \{-1, 1\}$ , 于是有  $\forall y_i: y_i(\vec{w}^\top \vec{x}_i + b) / \|\vec{w}\| \geq d$

由于  $\vec{w}$  与  $b$  可以任意缩放相同倍数以表示同一直线

则令  $d = \min \text{distance}(\vec{x}_i, \vec{w}, b)$

$$= \min \frac{1}{\|\vec{w}\|} |y_i(\vec{w}^\top \vec{x}_i + b)| = \frac{1}{\|\vec{w}\|} \min |y_i(\vec{w}^\top \vec{x}_i + b)|$$

再令  $\min |y_i(\vec{w}^\top \vec{x}_i + b)| = 1$

则  $\max d(\vec{w}, b) = \max \frac{1}{\|\vec{w}\|}$ , s.t.  $\min |y_i(\vec{w}^\top \vec{x}_i + b)| = 1$

相对地放松约束条件, 即得到必要条件

于是有  $\max \frac{1}{\|\vec{w}\|}$ , s.t.  $\forall i: y_i(\vec{w}^\top \vec{x}_i + b) \geq 1$

最大化  $\frac{1}{\|\vec{w}\|}$  等价于最小化  $\frac{1}{2} \|\vec{w}\|^2 = \frac{1}{2} \vec{w}^\top \vec{w}$

于是有  $\min \frac{1}{2} \vec{w}^\top \vec{w}$ , s.t.  $\forall i: y_i(\vec{w}^\top \vec{x}_i + b) \geq 1$

这个问题称为标准问题 (standard problem)

而最优化得到的超平面直线  $\vec{w}^\top \vec{x} + b = 0$  称为支持向量机

由于最小化目标为一个凸 (convex) 二次函数, 约束条件为  $\vec{w}$  和  $b$  的一次式

则称为二次规划 (quadratic programming)

标准形式为 optimal  $\vec{u} \leftarrow QP(Q, \vec{p}, A, \vec{c})$

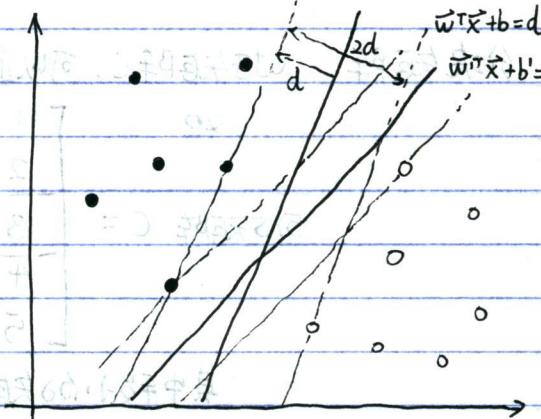
$$\min_{\vec{u}} \frac{1}{2} \vec{u}^\top Q \vec{u} + \vec{p}^\top \vec{u}$$

subject to  $\vec{a}_m^\top \vec{u} \geq c_m$  for  $m = 1, 2, \dots, M$

目标是最小化向量  $\vec{u}$  的二次函数, 矩阵  $Q$  为二次项系数, 向量  $\vec{p}$  为一次项系数

有  $M$  个向量  $\vec{a}_m$  的线性约束条件, 矩阵  $A = (\vec{a}_1, \vec{a}_2, \dots, \vec{a}_M)$ , 向量  $\vec{c} = (c_1, c_2, \dots, c_M)^\top$

其中  $a_{m,i}$  为第  $m$  个条件的一次项系数,  $c_m$  为第  $m$  个条件的常数项



# Machine

## Learning - P45

支持向量机

对于训练集  $T = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_N, y_N)\}$

其中输入空间为  $D$  维欧几里得空间，即  $\vec{x}_i \in \mathbb{R}^D$

输出空间为二分类标记集合，即  $y_i \in \{-1, 1\}$

目标是找到超平面  $\vec{w}^\top \vec{x} + b = 0$

其中  $b$  为实数参数， $\vec{w}$  为  $D$  维实数列向量  $\vec{w} \in \mathbb{R}^D$

即  $\min_{\vec{w}, b} \frac{1}{2} \vec{w}^\top \vec{w}$ , s.t.  $\forall i: y_i(\vec{w}^\top \vec{x}_i + b) \geq 1$

如果 SVM 需要进行非线性变换，则需要在转换后的  $\mathcal{X}$  空间进行线性 SVM 的求解

则  $\mathcal{X}$  空间中的线性分类对应于原来的  $\mathcal{X}$  空间中的可能的非线性分类

但是  $\mathcal{X}$  空间的 QP 问题有  $d+1$  个变量和  $N$  个约束条件

则当  $\mathcal{X}$  空间的维度  $d$  非常大甚至趋于无穷时，则无法求解这个 QP 问题

于是希望 SVM 模型的求解不依赖于转换后的  $\mathcal{X}$  空间维度  $d$

以便可以使用非常多的、非常复杂的特征转换

将 original 问题转换为 dual (对偶) 问题

(convex) QP of

$d+1$  variables

(convex) QP of

$N$  variables

$N$  constraints

$N+1$  constraints

定义  $N$  维实数列向量  $\vec{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_N)^\top$ ，且有  $\alpha_i \geq 0, i=1, 2, \dots, N$

拉格朗日函数  $L(\vec{w}, b, \vec{\alpha}) = \frac{1}{2} \vec{w}^\top \vec{w} + \sum_{i=1}^N \alpha_i [1 - y_i(\vec{w}^\top \vec{x}_i + b)]$

则原始问题是等价于  $\min_{\vec{w}, b} (\max_{\forall i: \alpha_i \geq 0} L(\vec{w}, b, \vec{\alpha}))$

当任意条件  $y_i(\vec{w}^\top \vec{x}_i + b) \geq 1$  不符合时， $1 - y_i(\vec{w}^\top \vec{x}_i + b) > 0$

则在  $\max_{\forall i: \alpha_i \geq 0} L(\vec{w}, b, \vec{\alpha})$  过程中将对应  $\alpha_i$  推向  $\infty$  以最大化

但是对应结果在  $\min_{\vec{w}, b}$  过程中被舍弃

当所有条件  $y_i(\vec{w}^\top \vec{x}_i + b) \geq 1$  均符合时， $1 - y_i(\vec{w}^\top \vec{x}_i + b) \leq 0$

则在  $\max_{\forall i: \alpha_i \geq 0} L(\vec{w}, b, \vec{\alpha})$  过程中将所有  $\alpha_i$  推向 0

则在  $\min_{\vec{w}, b}$  过程中等价于  $\min_{\vec{w}, b} \frac{1}{2} \vec{w}^\top \vec{w}$

于是有目标函数  $J(\vec{w}) = \begin{cases} \frac{1}{2} \vec{w}^\top \vec{w}, & \vec{x}_i \in \text{可行区域} \\ +\infty, & \vec{x}_i \in \text{不可行区域} \end{cases}$

由于对于给定的向量  $\vec{\alpha}'$ ，且有  $\alpha'_i \geq 0, i=1, 2, \dots, N$

都有  $\min_{\vec{w}, b} (\max_{\forall i: \alpha'_i \geq 0} L(\vec{w}, b, \vec{\alpha}')) \geq \min_{\vec{w}, b} L(\vec{w}, b, \vec{\alpha}')$

由于内部的  $\max$  始终大于等于给定的  $L(\vec{w}, b, \vec{\alpha}')$

则有  $\min_{\vec{w}, b} (\max_{\forall i: \alpha'_i \geq 0} L(\vec{w}, b, \vec{\alpha}')) \geq \max_{\forall i: \alpha'_i \geq 0} (\min_{\vec{w}, b} L(\vec{w}, b, \vec{\alpha}'))$

称为拉格朗日对偶问题 (Lagrange dual problem)