线性模型

1. 给定样本 $\vec{\mathbf{x}}$ ，其中 $\vec{\mathbf{x}}=(x_1,x_2,\cdots,x_n)^T$， x_i 为样本 $\vec{\mathbf{x}}$ 的第 i 个特征，特征有 n 种。

   线性模型(`linear model`)的形式为：$f(\vec{\mathbf{x}})=\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}+b$。

   其中 $\vec{\mathbf{w}}=(w_1,w_2,\cdots,w_n)^T$ 为每个特征对应的权重生成的权重向量。

2. 线性模型的优点是：
   - 模型简单。
   - 可解释性强，权重向量 $\vec{\mathbf{w}}$ 直观地表达了各个特征在预测中的重要性。

3. 很多功能强大的非线性模型(`nolinear model`)可以在线性模型的基础上通过引入层级结构或者非线性映射得到。

一、线性回归

1.1 问题

1. 给定数据集 $\mathbb{D}=\{(\vec{\mathbf{x}}_1,\tilde{y}_1),(\vec{\mathbf{x}}_2,\tilde{y}_2),\cdots,(\vec{\mathbf{x}}_N,\tilde{y}_N)\}$，其中 $\vec{\mathbf{x}}_i=(x_{i,1},x_{i,2},\cdots,x_{i,n})^T\in\mathcal{X}\subseteq\mathbb{R}^n,\ \tilde{y}_i\in\mathcal{Y}\subseteq\mathbb{R}$。

   线性回归问题试图学习模型： $f(\vec{\mathbf{x}})=\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}+b$

   > 该问题也被称作多元线性回归(`multivariate linear regression`)

2. 对于每个 $\vec{\mathbf{x}}_i$，其预测值为 $\hat{y}_i=f(\vec{\mathbf{x}}_i)=\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b$。采用平方损失函数，则在训练集 $\mathbb{D}$ 上，模型的损失函数为：

$$L(f)=\sum_{i=1}^{N}(\hat{y}_i-\tilde{y}_i)^2=\sum_{i=1}^{N}(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b-\tilde{y}_i)^2$$

   优化目标是损失函数最小化，即：$(\vec{\mathbf{w}}^*,b^*)=\arg\min_{\vec{\mathbf{w}},b}\sum_{i=1}^{N}(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b-\tilde{y}_i)^2$ 。

1.2 求解

1. 可以用梯度下降法来求解上述最优化问题的数值解，但是实际上该最优化问题可以通过最小二乘法获得解析解。

2. 令：

$$\vec{\tilde{\mathbf{w}}}=(w_1,w_2,\cdots,w_n,b)^T=(\vec{\mathbf{w}}^T,b)^T$$
$$\vec{\tilde{\mathbf{x}}}=(x_1,x_2,\cdots,x_n,1)^T=(\vec{\mathbf{x}}^T,1)^T$$
$$\vec{\mathbf{y}}=(\tilde{y}_1,\tilde{y}_2,\cdots,\tilde{y}_N)^T$$

   则有：

$$\sum_{i=1}^{N}(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b-\tilde{y}_i)^2=\left(\vec{\mathbf{y}}-(\vec{\tilde{\mathbf{x}}}_1,\vec{\tilde{\mathbf{x}}}_2,\cdots,\vec{\tilde{\mathbf{x}}}_N)^T\vec{\tilde{\mathbf{w}}}\right)^T\left(\vec{\mathbf{y}}-(\vec{\tilde{\mathbf{x}}}_1,\vec{\tilde{\mathbf{x}}}_2,\cdots,\vec{\tilde{\mathbf{x}}}_N)^T\vec{\tilde{\mathbf{w}}}\right)$$

   令：

$$\mathbf{X}=(\vec{\mathbf{x}}_1,\vec{\mathbf{x}}_2,\cdots,\vec{\mathbf{x}}_N)^T=\begin{bmatrix}\vec{\mathbf{x}}_1^T\\ \vec{\mathbf{x}}_2^T\\ \vdots\\ \vec{\mathbf{x}}_N^T\end{bmatrix}=\begin{bmatrix}x_{1,1}&x_{2,1}&\cdots&x_{n,1}&1\\ x_{1,2}&x_{2,2}&\cdots&x_{n,2}&1\\ \vdots&\vdots&\ddots&\vdots&1\\ x_{1,N}&x_{2,N}&\cdots&x_{n,N}&1\end{bmatrix}$$

则：

$$\vec{\mathbf{w}}^*=\arg\min_{\vec{\mathbf{w}}}(\vec{\mathbf{y}}-\mathbf{X}\vec{\mathbf{w}})^T(\vec{\mathbf{y}}-\mathbf{X}\vec{\mathbf{w}})$$

3. 令 $E_{\vec{\mathbf{w}}}=(\vec{\mathbf{y}}-\mathbf{X}\vec{\mathbf{w}})^T(\vec{\mathbf{y}}-\mathbf{X}\vec{\mathbf{w}})$ 。为求得它的极小值，可以通过对 $\vec{\mathbf{w}}$ 求导，并令导数为零，从而得到解析解：

$$\frac{\partial E_{\vec{\mathbf{w}}}}{\partial\vec{\mathbf{w}}}=2\mathbf{X}^T(\mathbf{X}\vec{\mathbf{w}}-\vec{\mathbf{y}})=\vec{\mathbf{0}}\Longrightarrow\mathbf{X}^T\mathbf{X}\vec{\mathbf{w}}=\mathbf{X}^T\vec{\mathbf{y}}$$

   - 当 $\mathbf{X}^T\mathbf{X}$ 为满秩矩阵时，可得：$\vec{\mathbf{w}}^*=(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\vec{\mathbf{y}}$ 。

     其中 $(\mathbf{X}^T\mathbf{X})^{-1}$ 为 $\mathbf{X}^T\mathbf{X}$ 的逆矩阵。

     最终学得的多元线性回归模型为：$f(\vec{\mathbf{x}}_i)=\vec{\mathbf{w}}^{*T}\vec{\mathbf{x}}_i=\vec{\mathbf{w}}^{*T}\vec{\mathbf{x}}_i+b^*$ 。
   - 当 $\mathbf{X}^T\mathbf{X}$ 不是满秩矩阵。此时存在多个解析解，他们都能使得均方误差最小化。究竟选择哪个解作为输出，由算法的偏好决定。

     > 比如 $N<n$ （样本数量小于特征种类的数量），根据 $\mathbf{X}$ 的秩小于等于 N,n 中的最小值，即小于等于 N（矩阵的秩一定小于等于矩阵的行数和列数）； 而矩阵 $\mathbf{X}^T\mathbf{X}$ 是 $n\times n$ 大小的，它的秩一定小于等于 N，因此不是满秩矩阵。

     常见的做法是引入正则化项：

     - L_1 正则化：此时称作`Lasso Regression`：

       $$\vec{\mathbf{w}}^*=\arg\min_{\vec{\mathbf{w}}}\left[(\vec{\mathbf{y}}-\mathbf{X}\vec{\mathbf{w}})^T(\vec{\mathbf{y}}-\mathbf{X}\vec{\mathbf{w}})+\lambda||\vec{\mathbf{w}}||_1\right]$$

       $\lambda>0$ 为正则化系数，调整正则化项与训练误差的比例。
     - L_2 正则化：此时称作`Ridge Regression`：

       $$\vec{\mathbf{w}}^*=\arg\min_{\vec{\mathbf{w}}}\left[(\vec{\mathbf{y}}-\mathbf{X}\vec{\mathbf{w}})^T(\vec{\mathbf{y}}-\mathbf{X}\vec{\mathbf{w}})+\lambda||\vec{\mathbf{w}}||_2^2\right]$$

       $\lambda>0$ 为正则化系数，调整正则化项与训练误差的比例。
     - 同时包含 L_1,L_2 正则化：此时称作`Elastic Net`：

       $$\vec{\mathbf{w}}^*=\arg\min_{\vec{\mathbf{w}}}\left[(\vec{\mathbf{y}}-\mathbf{X}\vec{\mathbf{w}})^T(\vec{\mathbf{y}}-\mathbf{X}\vec{\mathbf{w}})+\lambda\rho||\vec{\mathbf{w}}||_1+\frac{\lambda(1-\rho)}{2}||\vec{\mathbf{w}}||_2^2\right]$$

       其中：
       - $\lambda>0$ 为正则化系数，调整正则化项与训练误差的比例。
       - $1\geq\rho\geq0$ 为比例系数，调整 L_1 正则化与 L_2 正则化的比例。

1.3 算法

1. 多元线性回归算法：

- 输入：
  - 数据集 $\mathbb{D}=\{(\vec{\mathbf{x}}_1,\tilde{y}_1),(\vec{\mathbf{x}}_2,\tilde{y}_2),\cdots,(\vec{\mathbf{x}}_N,\tilde{y}_N)\},\vec{\mathbf{x}}_i\in\mathcal{X}\subseteq\mathbb{R}^n,y_i\in\mathcal{Y}\subseteq\mathbb{R}$
  - L_2 正则化项系数 $\lambda>0$
- 输出模型：$f(\vec{\mathbf{x}})=\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}}+b^*$
- 算法步骤：

  令：

$$\vec{\tilde{\mathbf{w}}}=(w_1,w_2,\cdots,w_n,b)^T=(\vec{\mathbf{w}}^T,b)^T$$
$$\vec{\tilde{\mathbf{x}}}=(x_1,x_2,\cdots,x_n,1)^T=(\vec{\mathbf{x}}^T,1)^T$$
$$\vec{\mathbf{y}}=(\tilde{y}_1,\tilde{y}_2,\cdots,\tilde{y}_N)^T$$
$$\mathbf{X}=(\vec{\tilde{\mathbf{x}}}_1,\vec{\tilde{\mathbf{x}}}_2,\cdots,\vec{\tilde{\mathbf{x}}}_N)^T=\begin{bmatrix}\vec{\tilde{\mathbf{x}}}_1^T\\ \vec{\tilde{\mathbf{x}}}_2^T\\ \vdots\\ \vec{\tilde{\mathbf{x}}}_N^T\end{bmatrix}=\begin{bmatrix}x_{1,1}&x_{2,1}&\cdots&x_{n,1}&1\\ x_{1,2}&x_{2,2}&\cdots&x_{n,2}&1\\ \vdots&\vdots&\ddots&\vdots&1\\ x_{1,N}&x_{2,N}&\cdots&x_{n,N}&1\end{bmatrix}$$

  求解：

$$\vec{\tilde{\mathbf{w}}}^*=\arg\min_{\vec{\tilde{\mathbf{w}}}}\left[(\vec{\mathbf{y}}-\mathbf{X}\vec{\tilde{\mathbf{w}}})^T(\vec{\mathbf{y}}-\mathbf{X}\vec{\tilde{\mathbf{w}}})+\lambda||\vec{\tilde{\mathbf{w}}}||_2\right]$$

  最终学得模型：$f(\vec{\mathbf{x}}_i)=\vec{\tilde{\mathbf{w}}}^{*T}\vec{\tilde{\mathbf{x}}}_i=\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}}+b^*$

二、广义线性模型

2.1 广义线性模型的函数定义

1. 考虑单调可微函数 $g(\cdot)$，令 $g(y)=\vec{\mathbf{w}}^T\vec{\mathbf{x}}+b$，这样得到的模型称作广义线性模型（`generalized linear model`）。

   其中函数 $g(\cdot)$ 称作联系函数（`link function`）。

2. 对数线性回归是广义线性模型在 $g(\cdot)=\ln(\cdot)$ 时的特例。即：$\ln y=\vec{\mathbf{w}}^T\vec{\mathbf{x}}+b$。
   - 它实际上是试图让 $\exp(\vec{\mathbf{w}}^T\vec{\mathbf{x}}+b)$ 逼近 y。
   - 它在形式上仍是线性回归，但是实质上是非线性的。

2.2 广义线性模型的概率定义

1. 如果给定 $\vec{\mathbf{x}}$ 和 $\vec{\mathbf{w}}$之后，y 的条件概率分布 $p(y\mid\vec{\mathbf{x}};\vec{\mathbf{w}})$ 服从指数分布族，则该模型称作广义线性模型。

   指数分布族的形式为：$p(y;\eta)=b(y)*\exp(\eta T(y)-a(\eta))$。
   - η 是 $\vec{\mathbf{x}}$ 的线性函数：$\eta=\vec{\mathbf{w}}^T\vec{\mathbf{x}}$
   - $b(y),T(y)$ 为 y 的函数
   - $a(\eta)$ 为 η 的函数

2.3 常见分布的广义线性模型

2.3.1 高斯分布

1. 高斯分布：

$$p(y)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{y^2}{2\sigma^2}\right)\exp\left(\frac{\mu}{\sigma^2}\times y-\frac{\mu^2}{2\sigma^2}\right)$$

令：

$$b(y)=\frac{1}{\sqrt{2\pi}\sigma}\times\exp\left(-\frac{y^2}{2\sigma^2}\right)$$
$$T(y)=y$$
$$\eta=\frac{\mu}{\sigma^2}$$
$$a(\eta)=\frac{\mu^2}{2\sigma^2}$$

则满足广义线性模型。

2.3.2 伯努利分布

1. 伯努利分布（二项分布，y 为 0 或者 1，取 1的概率为 ϕ ）：

$$p(y;\phi)=\phi^y(1-\phi)^{1-y}=\exp\left(y\ln\frac{\phi}{1-\phi}+ln(1-\phi)\right)$$

令：

$$b(y)=1$$
$$\eta=\ln\frac{\phi}{1-\phi}$$
$$T(y)=y$$
$$a(\eta)=-\ln(1-\phi)$$

则满足广义线性模型。

2. 根据 $\eta=\vec{\mathbf{w}}^T\vec{\mathbf{x}}$，有 $\eta=\vec{\mathbf{w}}^T\vec{\mathbf{x}}=\ln\frac{\phi}{1-\phi}$ 。 则得到：

$$\phi=\frac{1}{1+\exp(-\vec{\mathbf{w}}^T\vec{\mathbf{x}})}$$

因此 `logistic` 回归属于伯努利分布的广义形式。

2.3.3 多元伯努利分布

1. 假设有 K 个分类，样本标记 $\tilde{y}\in\{1,2,\cdots,K\}$ 。每种分类对应的概率为 $\phi_1,\phi_2,\cdots,\phi_K$。则根据全概率公式，有

$$\sum_{i=1}^{K}\phi_i=1$$
$$\phi_K=1-\sum_{i=1}^{K-1}\phi_i$$

   - 定义 $T(y)$ 为一个 $K-1$ 维的列向量：

$$T(1)=\begin{bmatrix}1\\0\\0\\\vdots\\0\end{bmatrix},T(2)=\begin{bmatrix}0\\1\\0\\\vdots\\0\end{bmatrix},\cdots,T(K-1)=\begin{bmatrix}0\\0\\0\\\vdots\\1\end{bmatrix},T(K)=\begin{bmatrix}0\\0\\0\\\vdots\\0\end{bmatrix}$$

- 定义示性函数 :$I(y=i)$ 表示属于 i 分类； $I(y\neq i)$ 表示不属于 i 分类。则有： $T(y)_i=I(y=i)$
- 构建概率密度函数为：

$$\begin{aligned}p(y;\phi)&=\phi_1^{I(y=1)}\times\phi_2^{I(y=2)}\times\cdots\times\phi_K^{I(y=K)}\\&=\phi_1^{I(y=1)}\times\phi_2^{I(y=2)}\times\cdots\times\phi_K^{1-\sum_{i=1}^{K-1}I(y=i)}\\&=\phi_1^{T(y)_1}\times\phi_2^{T(y)_2}\times\cdots\times\phi_K^{1-\sum_{i=1}^{K-1}T(y)_i}\\&=\exp\left(T(y)_1\times\ln\phi_1+T(y)_2\times\ln\phi_2+\cdots+(1-\sum_{i=1}^{K-1}T(y)_i)\times\ln\phi_K\right)\\&=\exp\left(T(y)_1\times\ln\frac{\phi_1}{\phi_K}+T(y)_2\times\ln\frac{\phi_2}{\phi_K}+\cdots+T(y)_{K-1}\times\ln\frac{\phi_{K-1}}{\phi_K}+\ln\phi_K\right)\end{aligned}$$

- 令

$$\eta=(\ln\frac{\phi_1}{\phi_K},\ln\frac{\phi_2}{\phi_K},\cdots,\ln\frac{\phi_{K-1}}{\phi_K})^T$$

则有：

$$p(y;\phi)=\exp(\eta\cdot T(y)+\ln\phi_K)$$

令 $b(y)=1,a(\eta)=-\ln\phi_K$，则满足广义线性模型。

2. 根据：

$$\eta_i=\ln\frac{\phi_i}{\phi_K}\rightarrow\phi_i=\phi_Ke^{\eta_i}$$

则根据：

$$1=\sum_{i=1}^{K}\phi_i=\phi_K\left(1+\sum_{i=1}^{K-1}e^{\eta_i}\right)\rightarrow\phi_K=\frac{1}{1+\sum_{i=1}^{K-1}e^{\eta_i}}$$

于是有：

$$\phi_i=\begin{cases}\frac{e^{\eta_i}}{1+\sum_{j=1}^{K-1}e^{\eta_j}},&i=1,2,\cdots,K-1\\\frac{1}{1+\sum_{j=1}^{K-1}e^{\eta_j}},&i=K\end{cases}$$

.

三、对数几率回归

1. 线性回归不仅可以用于回归任务，还可以用于分类任务。

3.1 二分类模型

1. 考虑二分类问题。

   给定数据集 $\mathbb{D}=\{(\vec{\mathbf{x}}_1,\tilde{y}_1),(\vec{\mathbf{x}}_2,\tilde{y}_2),\cdots,(\vec{\mathbf{x}}_N,\tilde{y}_N)\},\vec{\mathbf{x}}_i\in\mathcal{X}\subseteq\mathbb{R}^n,y_i\in\mathcal{Y}=\{0,1\},i=1,2,\cdots,N$ 。

   - 考虑到 $\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}+b$ 取值是连续的，因此它不能拟合离散变量。

     可以考虑用它来拟合条件概率 $p(y=1\mid\vec{\mathbf{x}})$，因为概率的取值也是连续的。

   - 但是对于 $\vec{\mathbf{w}}\neq\vec{\mathbf{0}}$（若等于零向量则没有什么求解的价值），$\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}+b$ 取值是从 $-\infty\sim+\infty$，不符合概率取值为，$0\sim1$因此考虑采用广义线性模型。

     最理想的是单位阶跃函数：

     $$p(y=1\mid\vec{\mathbf{x}})=\begin{cases}0, & z<0\\0.5, & z=0\\1, & z>0\end{cases},z=\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}+b$$

   - 但是阶跃函数不满足单调可微的性质，不能直接用作 $g(\cdot)$ 。

     对数几率函数(`logistic function`)就是这样的一个替代函数：

     $$p(y=1\mid\vec{\mathbf{x}})=\frac{1}{1+e^{-z}},z=\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}+b$$

     这样的模型称作对数几率回归(`logistic regression`或`logit regression`）模型。

2. 由于 $p(y=0\mid\vec{\mathbf{x}})=1-p(y=1\mid\vec{\mathbf{x}})$，则有：

   $$\ln\frac{P(y=1\mid\vec{\mathbf{x}})}{P(y=0\mid\vec{\mathbf{x}})}=z=\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}+b$$

   - 比值 $\frac{P(y=1|\vec{\mathbf{x}})}{P(y=0|\vec{\mathbf{x}})}$ 表示样本为正例的可能性比上反例的可能性，称作几率(`odds`）。几率反映了样本作为正例的相对可能性。

     几率的对数称作对数几率(`log odds`，也称作`logit`）。

   - 对数几率回归就是用线性回归模型的预测结果去逼近真实标记的对数几率。

3. 虽然对数几率回归名字带有回归，但是它是一种分类的学习方法。其优点：
   - 直接对分类的可能性进行建模，无需事先假设数据分布，这就避免了因为假设分布不准确带来的问题。
   - 不仅预测出来类别，还得到了近似概率的预测，这对许多需要利用概率辅助决策的任务有用。
   - 对数函数是任意阶可导的凸函数，有很好的数学性质，很多数值优化算法都能直接用于求取最优解。

3.2 参数估计

1. 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 $\vec{\mathbf{x}}_i\in\mathbb{R}^n,y_i\in\{0,1\}$。可以用极大似然估计法估计模型参数，从而得出模型。

   为了便于讨论，将参数 b 吸收进 $\vec{\mathbf{w}}$ 中。

   令：

   $$\vec{\mathbf{w}}=(w_1,w_2,\cdots,w_n,b)^T\in\mathbb{R}^{n+1}$$
   $$\vec{\mathbf{x}}=(x_1,x_2\cdots,x_n,1)^T\in\mathbb{R}^{n+1}$$

   令

$$p(y=1\mid\vec{\mathbf{x}})=\pi(\vec{\mathbf{x}})=\frac{\exp(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}})}{1+\exp(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}})}$$
$$p(y=0\mid\vec{\mathbf{x}})=1-\pi(\vec{\mathbf{x}})$$

则似然函数为：$\prod_{i=1}^{N}[\pi(\vec{\mathbf{x}}_i)]^{\tilde{y}_i}[1-\pi(\vec{\mathbf{x}}_i)]^{1-\tilde{y}_i}$ 。

对数似然函数为：

$$\begin{aligned}L(\vec{\mathbf{w}})&=\sum_{i=1}^{N}[\tilde{y}_i\log\pi(\vec{\mathbf{x}}_i)+(1-\tilde{y}_i)\log(1-\pi(\vec{\mathbf{x}}_i))]\\&=\sum_{i=1}^{N}[\tilde{y}_i\log\frac{\pi(\vec{\mathbf{x}}_i)}{1-\pi(\vec{\mathbf{x}}_i)}+\log(1-\pi(\vec{\mathbf{x}}_i))]\end{aligned}$$

2. 由于 $\pi(\vec{\mathbf{x}})=\frac{\exp(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}})}{1+\exp(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}})}$，因此：

$$L(\vec{\mathbf{w}})=\sum_{i=1}^{N}[\tilde{y}_i(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i)-\log(1+\exp(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i))]$$

则需要求解最优化问题：

$$\vec{\mathbf{w}}^*=\arg\max_{\vec{\mathbf{w}}}L(\vec{\mathbf{w}})=\arg\max_{\vec{\mathbf{w}}}\sum_{i=1}^{N}[\tilde{y}_i(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i)-\log(1+\exp(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i))]$$

最终 `logistic` 回归模型为：

$$p(y=1\mid\vec{\mathbf{x}})=\frac{\exp(\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}})}{1+\exp(\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}})}$$
$$p(y=0\mid\vec{\mathbf{x}})=\frac{1}{1+\exp(\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}})}$$

3. `logistic` 回归的最优化问题，通常用梯度下降法或者拟牛顿法来求解。

3.3 多分类模型

1. 可以推广二分类的 `logistic` 回归模型到多分类问题。
2. 设离散型随机变量 y 的取值集合为：$\{1,2,\cdots,K\}$，则多元 `logistic` 回归模型为：

$$p(y=k\mid\vec{\mathbf{x}})=\frac{\exp(\vec{\mathbf{w}}_k\cdot\vec{\mathbf{x}})}{1+\sum_{j=1}^{K-1}\exp(\vec{\mathbf{w}}_j\cdot\vec{\mathbf{x}})},\quad k=1,2,\cdots,K-1$$
$$p(y=K\mid\vec{\mathbf{x}})=\frac{1}{1+\sum_{j=1}^{K-1}\exp(\vec{\mathbf{w}}_j\cdot\vec{\mathbf{x}})}$$

其中 $\vec{\mathbf{x}}\in\mathbb{R}^{n+1},\vec{\mathbf{w}}_k\in\mathbb{R}^{n+1}$ 。

其参数估计方法类似二项 logistic 回归模型。

四、线性判别分析

1. 线性判别分析 `Linear Discriminant Analysis:LDA` 基本思想：

- 训练时：给定训练样本集，设法将样例投影到某一条直线上，使得同类样例的投影点尽可能接近、异类样例的投影点尽可能远离。要学习的就是这样的一条直线。
- 预测时：对新样本进行分类时，将其投影到学到的直线上，在根据投影点的位置来确定新样本的类别。

4.1 二分类模型

1. 考虑二类分类问题。设数据集为：
$\mathbb{D}=\{(\vec{\mathbf{x}}_1,\tilde{y}_1),(\vec{\mathbf{x}}_2,\tilde{y}_2),\cdots,(\vec{\mathbf{x}}_N,\tilde{y}_N)\},\vec{\mathbf{x}}_i\in\mathcal{X}\subseteq\mathbb{R}^n,y_i\in\mathcal{Y}=\{0,1\}$ 。

4.1.1 投影

1. 设 $\mathbb{D}_0$ 表示类别为 `0` 的样例的集合，这些样例的均值向量为 $\vec{\mu}_0=(\mu_{0,1},\mu_{0,2},\cdots,\mu_{0,n})^T$，这些样例的特征之间协方差矩阵为 Σ_0（协方差矩阵大小为 $n\times n$）。

   设 $\mathbb{D}_1$ 表示类别为 `1` 的样例的集合，这些样例的均值向量为 $\vec{\mu}_1=(\mu_{1,1},\mu_{1,2},\cdots,\mu_{1,n})^T$，这些样例的特征之间协方差矩阵为 Σ_1（协方差矩阵大小为 $n\times n$）

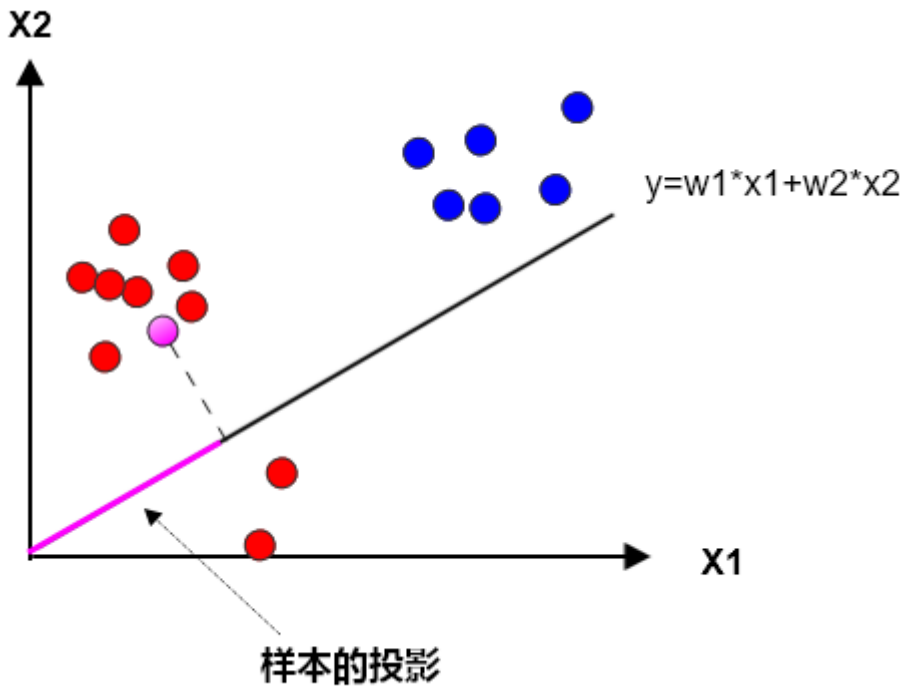
2. 假定直线为： $y=\vec{\mathbf{w}}^T\vec{\mathbf{x}}$，其中 $\vec{\mathbf{w}}=(w_1,w_2,\cdots,w_n)^T,\vec{\mathbf{x}}=(x_1,x_2,\cdots,x_n)^T$。

   这里省略了常量 b，因为考察的是样本点在直线上的投影，总可以平行移动直线到原点而保持投影不变，此时 $b=0$。

   将数据投影到直线上，则：

   - 两类样本的中心在直线上的投影分别为 $\vec{\mathbf{w}}^T\vec{\mu}_0$ 和 $\vec{\mathbf{w}}^T\vec{\mu}_1$
   - 两类样本投影的方差分别为 $\vec{\mathbf{w}}^T\Sigma_0\vec{\mathbf{w}}$ 和 $\vec{\mathbf{w}}^T\Sigma_1\vec{\mathbf{w}}$

   > 由于直线是一维空间，因此上面四个值均为实数



3. 根据线性判别分析的思想：
   - 要使得同类样例的投影点尽可能接近，则可以使同类样例投影点的方差尽可能小，即 $\vec{\mathbf{w}}^T\Sigma_0\vec{\mathbf{w}}+\vec{\mathbf{w}}^T\Sigma_1\vec{\mathbf{w}}$ 尽可能小

- 要使异类样例的投影点尽可能远，则可以使异类样例的中心的投影点尽可能远，即 $||\vec{\mathbf{w}}^T\vec{\mu}_0-\vec{\mathbf{w}}^T\vec{\mu}_1||_2^2$ 尽可能大
- 同时考虑两者，则得到最大化的目标：

$$J=\frac{||\vec{\mathbf{w}}^T\vec{\mu}_0-\vec{\mathbf{w}}^T\vec{\mu}_1||_2^2}{\vec{\mathbf{w}}^T\Sigma_0\vec{\mathbf{w}}+\vec{\mathbf{w}}^T\Sigma_1\vec{\mathbf{w}}}=\frac{\vec{\mathbf{w}}^T(\vec{\mu}_0-\vec{\mu}_1)(\vec{\mu}_0-\vec{\mu}_1)^T\vec{\mathbf{w}}}{\vec{\mathbf{w}}^T(\Sigma_0+\Sigma_1)\vec{\mathbf{w}}}$$

.

4.1.2 求解

1. 定义类内散度矩阵和类间散度矩阵：
   - 类内散度矩阵 `within-class scatter matrix`：

$$\mathbf{S}_w=\Sigma_0+\Sigma_1=\sum_{\vec{\mathbf{x}}\in\mathbb{D}_0}(\vec{\mathbf{x}}-\vec{\mu}_0)(\vec{\mathbf{x}}-\vec{\mu}_0)^T+\sum_{\vec{\mathbf{x}}\in\mathbb{D}_1}(\vec{\mathbf{x}}-\vec{\mu}_1)(\vec{\mathbf{x}}-\vec{\mu}_1)^T$$

     它是每个类的散度矩阵之和。
   - 类间散度矩阵 `between-class scatter matrix`：$\mathbf{S}_b=(\vec{\mu}_0-\vec{\mu}_1)(\vec{\mu}_0-\vec{\mu}_1)^T$ 。

     它是向量 $(\vec{\mu}_0-\vec{\mu}_1)$ 与它自身的外积。
2. 利用类内散度矩阵和类间散度矩阵，线性判别分析的最优化目标为：

$$J=\frac{\vec{\mathbf{w}}^T\mathbf{S}_b\vec{\mathbf{w}}}{\vec{\mathbf{w}}^T\mathbf{S}_w\vec{\mathbf{w}}}$$

   J 也称作 $\mathbf{S}_b$ 与 $\mathbf{S}_w$ 的广义瑞利商 。
3. 现在求解最优化问题：

$$\vec{\mathbf{w}}^*=\arg\max_{\vec{\mathbf{w}}}\frac{\vec{\mathbf{w}}^T\mathbf{S}_b\vec{\mathbf{w}}}{\vec{\mathbf{w}}^T\mathbf{S}_w\vec{\mathbf{w}}}$$

   - 考虑到分子与分母都是关于 $\vec{\mathbf{w}}$ 的二次项，因此上式的解与 $\vec{\mathbf{w}}$ 的长度无关，只与 $\vec{\mathbf{w}}$ 的方向有关。令 $\vec{\mathbf{w}}^T\mathbf{S}_w\vec{\mathbf{w}}=1$，则最优化问题改写为：

$$\vec{\mathbf{w}}^*=\arg\min_{\vec{\mathbf{w}}}-\vec{\mathbf{w}}^T\mathbf{S}_b\vec{\mathbf{w}}$$
$$s.t.\ \vec{\mathbf{w}}^T\mathbf{S}_w\vec{\mathbf{w}}=1$$

   - 应用拉格朗日乘子法，上式等价于 $\mathbf{S}_b\vec{\mathbf{w}}=\lambda\mathbf{S}_w\vec{\mathbf{w}}$
     - 令 $(\vec{\mu}_0-\vec{\mu}_1)^T\vec{\mathbf{w}}=\lambda_{\vec{\mathbf{w}}}$，其中 $\lambda_{\vec{\mathbf{w}}}$ 为实数。则 $\mathbf{S}_b\vec{\mathbf{w}}=(\vec{\mu}_0-\vec{\mu}_1)(\vec{\mu}_0-\vec{\mu}_1)^T\vec{\mathbf{w}}=\lambda_{\vec{\mathbf{w}}}(\vec{\mu}_0-\vec{\mu}_1)$ 。代入上式有：

$$\mathbf{S}_b\vec{\mathbf{w}}=\lambda_{\vec{\mathbf{w}}}(\vec{\mu}_0-\vec{\mu}_1)=\lambda\mathbf{S}_w\vec{\mathbf{w}}$$

     - 由于与 $\vec{\mathbf{w}}$ 的长度无关，可以令 $\lambda_{\vec{\mathbf{w}}}=\lambda$ 则有：

$$(\vec{\mu}_0-\vec{\mu}_1)=\mathbf{S}_w\vec{\mathbf{w}}\Longrightarrow\vec{\mathbf{w}}=\mathbf{S}_w^{-1}(\vec{\mu}_0-\vec{\mu}_1)$$

     - 考虑数值解的稳定性，在实践中通常是对 $\mathbf{S}_w$ 进行奇异值分解：$\mathbf{S}_w=\mathbf{U}\Sigma\mathbf{V}^T$，其中 Σ 为实对角矩阵，对角线上的元素为 $\mathbf{S}_w$ 的奇异值；$\mathbf{U},\mathbf{V}$ 均为酉矩阵，它们的列向量分别构成了标准正交基。

然后 $\mathbf{S}_w^{-1}=\mathbf{V}\Sigma^{-1}\mathbf{U}^T$。

4.2 多分类模型

1. 可以将线性判别分析推广到多分类任务中。
2. 假定存在 M 个类，属于第 i 个类的样本的集合为 $\mathbb{D}_i$，$\mathbb{D}_i$ 中的样例数为 m_i 。其中：$\sum_{i=1}^M m_i=N$，N 为样本总数。
   - 定义类别 i 的均值向量为：所有该类别样本的均值：
     $$\vec{\mu}_i=(\mu_{i,1},\mu_{i,2},\cdots,\mu_{i,n})^T=\frac{1}{m_i}\sum_{\vec{\mathbf{x}}_i\in\mathbb{D}_i}\vec{\mathbf{x}}_i$$
     类别 i 的样例的特征之间协方差矩阵为 Σ_i （协方差矩阵大小为 $n\times n$）。
   - 定义 $\vec{\mu}=(\mu_1,\mu_2,\cdots,\mu_n)^T=\frac{1}{N}\sum_{i=1}^N\vec{\mathbf{x}}_i$ 是所有样例的均值向量。
3. 定义各类别的类内散度矩阵、总的类内散度矩阵、总的类间散度矩阵：
   - 定义类别 i 的类内散度矩阵为：
     $$\mathbf{S}_{wi}=\sum_{\vec{\mathbf{x}}\in\mathbb{D}_i}(\vec{\mathbf{x}}-\vec{\mu}_i)(\vec{\mathbf{x}}-\vec{\mu}_i)^T$$
     它实际上就等于样本集 $\mathbb{D}_i$ 的协方差矩阵 Σ_i， 刻画了同类样例投影点的方差。
   - 定义总的类内散度矩阵为：$\mathbf{S}_w=\sum_{i=1}^M\mathbf{S}_{wi}$。
     它 刻画了所有类别的同类样例投影点的方差。
   - 定义总的类间散度矩阵为：$\mathbf{S}_b=\sum_{i=1}^M m_i(\vec{\mu}_i-\vec{\mu})(\vec{\mu}_i-\vec{\mu})^T$ 。
     它刻画了异类样例的中心的投影点的相互距离。
     注意：$(\vec{\mu}_i-\vec{\mu})(\vec{\mu}_i-\vec{\mu})^T$ 也是一个协方差矩阵，它刻画的是第 i 类与总体之间的关系。
     - 由于这里有不止两个中心点，因此不能简单的套用二分类线性判别分析的做法。
       这里用每一类样本集的中心点与总的中心点的距离作为度量。
     - 考虑到每一类样本集的大小可能不同（密度分布不均），对这个距离施加权重，权重为每类样本集的大小。
4. 根据线性判别分析的思想，设$\mathbf{W}\in\mathbb{R}^{n\times(M-1)}$ 是投影矩阵。经过推导可以得到最大化的目标：
   $$J=\frac{tr(\mathbf{W}^T\mathbf{S}_b\mathbf{W})}{tr(\mathbf{W}^T\mathbf{S}_w\mathbf{W})}$$
   其中 $tr(.)$ 表示矩阵的迹。
   > - 一个矩阵的迹是矩阵对角线的元素之和，它是一个矩阵不变量，也等于所有特征值之和。
   > - 还有一个常用的矩阵不变量就是矩阵的行列式，它等于矩阵的所有特征值之积。
5. 与二分类线性判别分析不同，在多分类线性判别分析中投影方向是多维的，因此使用投影矩阵 $\mathbf{W}$ 。
   二分类线性判别分析的投影方向是一维的（只有一条直线），所以使用投影向量 $\vec{\mathbf{w}}$。
6. 上述最优化问题可以通过广义特征值问题求解：$\mathbf{S}_b\mathbf{W}=\lambda\mathbf{S}_w\mathbf{W}$
   - $\mathbf{W}$ 的解析解为 $\mathbf{S}_w^{-1}\mathbf{S}_b$ 的 $M-1$ 个最大广义特征值所对应的特征向量组成的矩阵。
   - 多分类线性判别分析将样本投影到 $M-1$ 维空间。
   - 通常 $M-1$ 远小于数据原有的特征数，`LDA` 因此也被视作一种经典的监督降维技术。

五、感知机

5.1 定义

1. 感知机是二分类的线性分类模型，属于判别模型。
   - 模型的输入为实例的特征向量，模型的输出为实例的类别：正类取值 `+1`，负类取值 `-1` 。
   - 感知机的物理意义：将输入空间（特征空间）划分为正、负两类的分离超平面。
2. 设输入空间（特征空间）为 $\mathcal{X} \subseteq \mathbb{R}^n$；输出空间为 $\mathcal{Y} = \{+1, -1\}$；输入 $\vec{\mathbf{x}} \in \mathcal{X}$ 为特征空间的点；输出 $y \in \mathcal{Y}$ 为实例的类别。

   定义函数 $f(\vec{\mathbf{x}}) = sign(\vec{\mathbf{w}} \cdot \vec{\mathbf{x}} + b)$ 为感知机。其中：
   - $\vec{\mathbf{w}} \in \mathbb{R}^n$ 为权值向量，$b \in \mathbb{R}$ 为偏置。它们为感知机的参数。
   - `sign` 为符号函数：

$$sign(x) = \begin{cases} +1, & x \geq 0 \\ -1, & x < 0 \end{cases}$$

3. 感知机的几何解释： $\vec{\mathbf{w}} \cdot \vec{\mathbf{x}} + b = 0$ 对应特征空间 $\mathbb{R}^n$ 上的一个超平面 S ，称作分离超平面。
   - $\vec{\mathbf{w}}$ 是超平面 S 的法向量， b 是超平面的截距。
   - 超平面 S 将特征空间划分为两个部分：
     - 超平面 S 上方的点判别为正类。
     - 超平面 S 下方的点判别为负类。

5.2 损失函数

1. 给定数据集 $\mathbb{D} = \{(\vec{\mathbf{x}}_1, \tilde{y}_1), (\vec{\mathbf{x}}_2, \tilde{y}_2), \cdots, (\vec{\mathbf{x}}_N, \tilde{y}_N)\}$，其中 $\vec{\mathbf{x}}_i \in \mathcal{X} \subseteq \mathbb{R}^n, \tilde{y}_i \in \mathcal{Y} = \{+1, -1\}$。

   若存在某个超平面 S： $\vec{\mathbf{w}} \cdot \vec{\mathbf{x}} + b = 0$， 使得将数据集中的正实例点与负实例点完全正确地划分到超平面的两侧，则称数据集 $\mathbb{D}$ 为线性可分数据集；否则称数据集 $\mathbb{D}$ 线性不可分。

   划分到超平面两侧，用数学语言描述为： $(\vec{\mathbf{w}} \cdot \vec{\mathbf{x}}_i + b)\tilde{y}_i > 0$
2. 根据感知机的定义：
   - 对正确分类的点 $(\vec{\mathbf{x}}_i, \tilde{y}_i)$，有 $(\vec{\mathbf{w}} \cdot \vec{\mathbf{x}} + b)\tilde{y}_i > 0$
   - 对误分类的点 $(\vec{\mathbf{x}}_i, \tilde{y}_i)$，有 $(\vec{\mathbf{w}} \cdot \vec{\mathbf{x}} + b)\tilde{y}_i < 0$
3. 如果将感知机的损失函数定义成误分类点的中总数，则它不是 $\vec{\mathbf{w}}, b$ 的连续可导函数，不容易优化。

   因此，定义感知机的损失函数为误分类点到超平面 S 的总距离。

   对误分类的点$(\vec{\mathbf{x}}_i, \tilde{y}_i)$，则 $\vec{\mathbf{x}}_i$ 距离超平面的距离为：

$$\frac{1}{||\vec{\mathbf{w}}||_2}|\vec{\mathbf{w}} \cdot \vec{\mathbf{x}}_i + b|$$

   由于 $|\tilde{y}_i| = 1$，以及 $(\vec{\mathbf{w}} \cdot \vec{\mathbf{x}}_i + b)\tilde{y}_i < 0$ ，因此上式等于

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

   不考虑 $\frac{1}{||\vec{\mathbf{w}}||_2}$，则得到感知机学习的损失函数：

$$L(\vec{\mathbf{w}}, b) = -\sum_{\vec{\mathbf{x}}_i \in \mathbb{M}} \tilde{y}_i(\vec{\mathbf{w}} \cdot \vec{\mathbf{x}}_i + b)$$

其中：

- $\mathbb{M}$ 为误分类点的集合。它隐式的与 $\vec{\mathbf{w}},b$ 相关，因为 $\vec{\mathbf{w}},b$ 优化导致误分类点减少从而使得 $\mathbb{M}$ 收缩。
- 之所以不考虑 $\frac{1}{||\vec{\mathbf{w}}||_2}$，因为感知机要求训练集线性可分，最终误分类点数量为零，此时损失函数为零。即使考虑分母，也是零。若训练集线性不可分，则感知机算法无法收敛。
- 误分类点越少或者误分类点距离超平面 S 越近， 则损失函数 L 越小。

4. 对于特定的样本点，其损失为：
   - 若正确分类，则损失为 0 。
   - 若误分类，则损失为 $\vec{\mathbf{w}},b$ 的线性函数。

因此给定训练集 $\mathbb{D}$ ， 损失函数 $L(\vec{\mathbf{w}},b)$ 是 $\vec{\mathbf{w}},b$ 的连续可导函数。

5.3 学习算法

1. 给定训练集 $\mathbb{D}=\{(\vec{\mathbf{x}}_1,\tilde{y}_1),(\vec{\mathbf{x}}_2,\tilde{y}_2),\cdots,(\vec{\mathbf{x}}_N,\tilde{y}_N)\},\vec{\mathbf{x}}_i\in\mathcal{X}\subseteq\mathbb{R}^n,\tilde{y}_i\in\mathcal{Y}=\{+1,-1\}$，求参数 $\vec{\mathbf{w}},b$ ：

$\vec{\mathbf{w}}^*,b^*=\min_{\vec{\mathbf{w}},b}L(\vec{\mathbf{w}},b)=\min_{\vec{\mathbf{w}},b}\left[-\sum_{\vec{\mathbf{x}}_i\in\mathbb{M}}\tilde{y}_i(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b)\right]$ 。

5.3.1 原始形式

1. 假设误分类点集合 $\mathbb{M}$ 是固定的，则损失函数 $L(\vec{\mathbf{w}},b)$ 的梯度为：

$$\nabla_{\vec{\mathbf{w}}}L(\vec{\mathbf{w}},b)=-\sum_{\vec{\mathbf{x}}_i\in\mathbb{M}}\tilde{y}_i\vec{\mathbf{x}}_i$$
$$\nabla_b L(\vec{\mathbf{w}},b)=-\sum_{\vec{\mathbf{x}}_i\in\mathbb{M}}\tilde{y}_i$$

2. 通过梯度下降法，随机选取一个误分类点 $(\vec{\mathbf{x}}_i,\tilde{y}_i)$，对 $\vec{\mathbf{w}},b$ 进行更新：

$$\vec{\mathbf{w}}\leftarrow\vec{\mathbf{w}}+\eta\tilde{y}_i\vec{\mathbf{x}}_i$$
$$b\leftarrow b+\eta\tilde{y}_i$$

   其中 $\eta\in(0,1]$ 是步长，即学习率。

   通过迭代可以使得损失函数 $L(\vec{\mathbf{w}},b)$ 不断减小直到 0 。

3. 感知机学习算法的原始形式：
   - 输入：
     - 线性可分训练集 $\mathbb{D}=\{(\vec{\mathbf{x}}_1,\tilde{y}_1),(\vec{\mathbf{x}}_2,,\tilde{y}_2),\cdots,(\vec{\mathbf{x}}_N,,\tilde{y}_N)\},\vec{\mathbf{x}}_i\in\mathcal{X}\subseteq\mathbb{R}^n,,\tilde{y}_i\in\mathcal{Y}=\{+1,-1\}$
     - 学习率 $\eta\in(0,1]$
   - 输出：
     - $\vec{\mathbf{w}}^*,b^*$
     - 感知机模型： $f(\vec{\mathbf{x}})=sign(\vec{\mathbf{w}}^*\cdot\vec{\mathbf{x}}+b^*)$
   - 步骤：
     - 选取初始值 $\vec{\mathbf{w}}_0,b_0$ 。
     - 在训练集中选取数据 $(\vec{\mathbf{x}}_i,\tilde{y}_i)$ 。若 $\tilde{y}_i(\vec{\mathbf{w}}\cdot\vec{\mathbf{x}}_i+b)\leq 0$ 则：

$$\vec{\mathbf{w}}\leftarrow\vec{\mathbf{w}}+\eta\tilde{y}_i\vec{\mathbf{x}}_i$$
$$b\leftarrow b+\eta\tilde{y}_i$$

- 在训练集中重复选取数据来更新 $\vec{\mathbf{w}}, b$ 直到训练集中没有误分类点。

5.3.2 性质

1. 对于某个误分类点 $(\vec{\mathbf{x}}_i, \tilde{y}_i)$ ，假设它被选中用于更新参数。
   - 假设迭代之前，分类超平面为 S ，该误分类点距超平面的距离为 d 。
   - 假设迭代之后，分类超平面为 S' ，该误分类点距超平面的距离为 d' 。

   则：

$$\begin{aligned}\Delta d = d' - d &= \frac{1}{||\vec{\mathbf{w}}'||_2}|\vec{\mathbf{w}}' \cdot \vec{\mathbf{x}}_i + b'| - \frac{1}{||\vec{\mathbf{w}}||_2}|\vec{\mathbf{w}} \cdot \vec{\mathbf{x}}_i + b| \\ &= -\frac{1}{||\vec{\mathbf{w}}'||_2}\tilde{y}_i(\vec{\mathbf{w}}' \cdot \vec{\mathbf{x}}_i + b') + \frac{1}{||\vec{\mathbf{w}}||_2}\tilde{y}_i(\vec{\mathbf{w}} \cdot \vec{\mathbf{x}}_i + b) \\ &\simeq -\frac{\tilde{y}_i}{||\vec{\mathbf{w}}||_2}[(\vec{\mathbf{w}}' - \vec{\mathbf{w}}) \cdot \vec{\mathbf{x}}_i + (b' - b)] \\ &= -\frac{\tilde{y}_i}{||\vec{\mathbf{w}}||}[\eta\tilde{y}_i\vec{\mathbf{x}}_i \cdot \vec{\mathbf{x}}_i + \eta\tilde{y}_i] \\ &= -\frac{\tilde{y}_i^2}{||\vec{\mathbf{w}}||_2}(\eta\vec{\mathbf{x}}_i \cdot \vec{\mathbf{x}}_i + 1) < 0\end{aligned}$$

   因此有 $d' < d$ 。

   这里要求 $\vec{\mathbf{w}}' \simeq \vec{\mathbf{w}}$，因此步长 η 要相当小。

2. 几何解释： 当一个实例点被误分类时，调整 $\vec{\mathbf{w}}, b$ 使得分离平面向该误分类点的一侧移动，以减少该误分类点与超平面间的距离，直至超平面越过所有的误分类点以正确分类。

3. 感知机学习算法由于采用不同的初值或者误分类点选取顺序的不同，最终解可以不同。

5.3.3 收敛性

1. 感知机收敛性定理：设线性可分训练集
   $\mathbb{D} = \{(\vec{\mathbf{x}}_1, \tilde{y}_1), (\vec{\mathbf{x}}_2, \tilde{y}_2), \cdots, (\vec{\mathbf{x}}_N, \tilde{y}_N)\}, \vec{\mathbf{x}}_i \in \mathcal{X} \subseteq \mathbb{R}^n, \tilde{y}_i \in \mathcal{Y} = \{+1, -1\}$ 。
   - 存在满足 $||\hat{\vec{\mathbf{w}}}_{opt}|| = 1$ 的超平面：$\hat{\vec{\mathbf{w}}}_{opt} \cdot \hat{\vec{\mathbf{x}}} = \vec{\mathbf{w}}_{opt} \cdot \vec{\mathbf{x}} + b_{opt} = 0$ ，该超平面将 $\mathbb{D}$ 完全正确分开。

     且存在 $r > 0$ ，对所有的 $i = 1, 2, \cdots, N$ 有：$\tilde{y}_i(\hat{\vec{\mathbf{w}}}_{opt} \cdot \hat{\vec{\mathbf{x}}}) = \tilde{y}_i(\vec{\mathbf{w}}_{opt} \cdot \vec{\mathbf{x}} + b_{opt}) \geq r$ 。

     其中 $\hat{\vec{\mathbf{w}}} = (\vec{\mathbf{w}}, b), \hat{\vec{\mathbf{x}}} = (\vec{\mathbf{x}}, 1), \hat{\vec{\mathbf{w}}} \in \mathbb{R}^{n+1}, \hat{\vec{\mathbf{x}}} \in \mathbb{R}^{n+1}, \vec{\mathbf{w}} \in \mathbb{R}^n, \vec{\mathbf{x}} \in \mathbb{R}^n$ 。
   - 令 $R = \max_{1 \leq i \leq N} ||\hat{\vec{\mathbf{x}}}_i||$，则感知机学习算法原始形式在 $\mathbb{D}$ 上的误分类次数 k 满足：

$$k \leq (\frac{R}{r})^2$$

2. 感知机收敛性定理说明了：
   - 当训练集线性可分时，感知机学习算法原始形式迭代是收敛的。
     - 此时算法存在许多解，既依赖于初值，又依赖于误分类点的选择顺序。
     - 为了得出唯一超平面，需要对分离超平面增加约束条件。
   - 当训练集线性不可分时，感知机学习算法不收敛，迭代结果会发生震荡。

5.3.4 对偶形式

1. 根据原始迭代形式：

$$\vec{\mathbf{w}} \leftarrow \vec{\mathbf{w}} + \eta \tilde{y}_i \vec{\mathbf{x}}_i$$
$$b \leftarrow b + \eta \tilde{y}_i$$

取初始值 $\vec{\mathbf{w}}_0, b_0$ 均为 0。则 $\vec{\mathbf{w}}, b$ 可以改写为：

$$\vec{\mathbf{w}} = \sum_{i=1}^{N} \alpha_i \tilde{y}_i \vec{\mathbf{x}}_i$$
$$b = \sum_{i=1}^{N} \alpha_i \tilde{y}_i$$

这就是感知机学习算法的对偶形式。

2. 感知机学习算法的对偶形式：
   - 输入：
     - 线性可分训练集 $\mathbb{D} = \{(\vec{\mathbf{x}}_1, \tilde{y}_1), (\vec{\mathbf{x}}_2, , \tilde{y}_2), \cdots, (\vec{\mathbf{x}}_N, , \tilde{y}_N)\}, \vec{\mathbf{x}}_i \in \mathcal{X} \subseteq \mathbb{R}^n, , \tilde{y}_i \in \mathcal{Y} = \{+1, -1\}$
     - 学习率 $\eta \in (0, 1]$
   - 输出：
     - $\vec{\alpha}, b$ ， 其中 $\vec{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_N)^T$ 。
     - 感知机模型 $f(\vec{\mathbf{x}}) = sign(\sum_{j=1}^{N} \alpha_j \tilde{y}_j \vec{\mathbf{x}}_j \cdot \vec{\mathbf{x}} + b)$。
   - 步骤：
     - 初始化：$\vec{\alpha} \leftarrow \vec{\mathbf{0}}, b \leftarrow 0$ 。
     - 在训练集中随机选取数据 $(\vec{\mathbf{x}}_i, \tilde{y}_i)$ ， 若 $\tilde{y}_i(\sum_{j=1}^{N} \alpha_j \tilde{y}_j \vec{\mathbf{x}}_j \cdot \vec{\mathbf{x}} + b) \leq 0$ 则更新：

$$\alpha_i \leftarrow \alpha_i + \eta$$
$$b \leftarrow b + \eta \tilde{y}_i$$

     - 在训练集中重复选取数据来更新 $\vec{\alpha}, b$ 直到训练集中没有误分类点。
3. 在对偶形式中， 训练集 $\mathbb{D}$ 仅仅以内积的形式出现，因为算法只需要内积信息。

   可以预先将 $\mathbb{D}$ 中的实例间的内积计算出来，并以矩阵形式存储。即 `Gram` 矩阵： $\mathbf{G} = [\vec{\mathbf{x}}_i \cdot \vec{\mathbf{x}}_j]_{N \times N}$
4. 与原始形式一样，感知机学习算法的对偶形式也是收敛的，且存在多个解。