No.

Date. . .

chap-3 线性模型

	激活函数	损失函数	优化方法
Linear Regression	-	$(y-\vec{w}^T\vec{x})^2$	最小二乘法，梯度下降法
Logistic R	$\sigma(\vec{w}^T\vec{x})$	$\vec{y}\log\sigma(\vec{w}^T\vec{x})$	梯度下降法
softmax R	$softmax(W^T\vec{x})$	$\vec{y}\log softmax(W^T\vec{x})$	梯度下降
	$sgn(\vec{w}^T\vec{x})$	$\max(0,-y\vec{w}^T\vec{x})$	SGD
SVM	$sgn(\vec{w}^T\vec{x})$	$\max(0,1-y\vec{w}^T\vec{x})$	二次规化，SMO

输出离散值寻找分类超平面

线性分类模型一般是一个线性判别函数 $f(\vec{x},\vec{w})=\vec{w}^T\vec{x}$ 加上一个激活函数 $g(\cdot)$

Logistic R 和 Softmax R 中，y 为 one-hot 向量表示，

和 SVM 中，y 为 $\{+1,-1\}$

No.

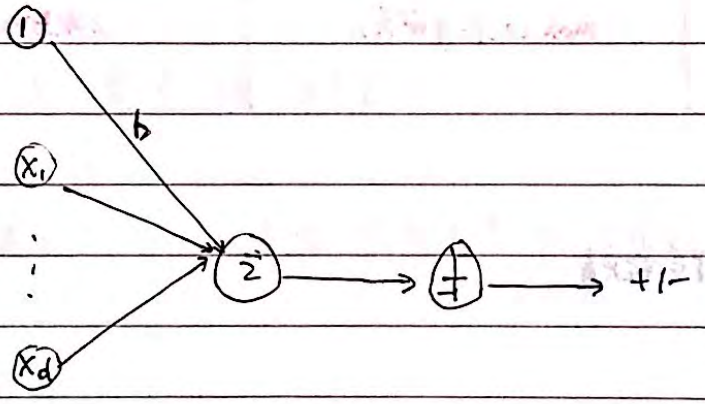
Date. . .

$f(\vec{x},\vec{w}) = \vec{w}^T\vec{x} + b$ 判别函数

$y = g(f(\vec{x},\vec{w}))$ 决策函数

符号函数 $g(f(\vec{x},\vec{w})) = sgn(f(\vec{x},\vec{w})) \triangleq \begin{cases} +1 & \text{if } f(\vec{x},\vec{w}) > 0 \\ -1 & \text{if } f(\vec{x},\vec{w}) < 0 \end{cases}$

两类分类的线性模型



两类分类的线性模型

Logisitic R., softmax R., Perceptron, SVM 的区别主要在于使用了不同的损失函数!

No.

Date. . .

3.1 线性判别函数和决策边界

LCM 线性分类模型

⇒ 3.1.1 两类分类 {+1, -1}

LCM的决策边界是线性超平面.

$$r = \frac{f(\vec{x}, \vec{w})}{\|\vec{w}\|}$$

r是 特征空间中每个样本点到决策平面的有向距离

点 $\vec{x}$ 在 $\vec{w}$ 方向上的投影

得到参数 $\vec{w}^*$, 使得 $y^{(n)}$ 和 $f(x^{(n)}, w^*)$ 尽量同符号

$$y^{(n)} f(x^{(n)}, \vec{w}^*) > 0$$

⇒ 3.1.2 多类分类

1. 一对其余.

2. 一对一

3. arg max

No.

Date.

3.2 logistic 回归.

$R^d \to (0,1)$

$$p(y=1|\vec{x}) = \sigma(\vec{w}^T\vec{x})$$

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

$$p(y=0|\vec{x}) = 1 - p(y=1|\vec{x}) = \frac{\exp(\vec{w}^T\vec{x})}{1+\exp(-\vec{w}^T\vec{x})}$$

$$\vec{w}^T\vec{x} = \log \frac{p(y=1|\vec{x})}{p(y=0|\vec{x})}$$ 对数几率

⇒ 3.2.1 参数学习

$\hat{y}^{(n)} = \sigma(\vec{w}^T\vec{x}^{(n)})$ 模型输出概率

样本 $(\vec{x}^{(n)}, y^{(n)})$ 的真实条件概率，$y^{(n)} \in \{0,1\}$

$$p_r(y^{(n)}=1|\vec{x}^{(n)}) = y^{(n)}$$

$$p_r(y^{(n)}=0|\vec{x}^{(n)}) = 1-y^{(n)}$$

emp Risky 为

$$R(\vec{w}) = -\frac{1}{N}\sum\left(y^{(n)}\log\hat{y}^{(n)} + (1-y^{(n)})\log(1-\hat{y}^{(n)})\right)$$

No.

Date. . .

$$\frac{\partial R(\vec{w})}{\partial \vec{w}} = -\frac{1}{N}\sum \left(y^{(n)} \frac{\hat{y}^{(n)}(1-\hat{y}^{(n)})}{\hat{y}^{(n)}} \vec{x}^{(n)} + (1-y^{(n)}) \frac{\hat{y}^{(n)}(1-\hat{y}^{(n)})}{1-\hat{y}^{(n)}} \vec{x}^{(n)} \right)$$

$$= -\frac{1}{N}\sum \left(y^{(n)}(1-\hat{y}^{(n)})\vec{x}^{(n)} - (1-y^{(n)})\hat{y}^{(n)}\vec{x}^{(n)} \right)$$

$$= -\frac{1}{N}\sum \left(y^{(n)} - y^{(n)}\hat{y}^{(n)} - \hat{y}^{(n)} + y^{(n)}\hat{y}^{(n)} \right)\vec{x}^{(n)}$$

$$= -\frac{1}{N}\sum \left[(y^{(n)} - \hat{y}^{(n)})\vec{x}^{(n)} \right]$$

$$= -\frac{1}{N}\left[(y^{(n)} - \hat{y}^{(n)})\vec{x}^{(n)} \right]$$

$\vec{w}_0 \leftarrow 0$, SGD

$$w_{t+1} \leftarrow w_t + \alpha \cdot \frac{1}{N}\sum_{n=1}^{N} \vec{x}^{(n)}(y^{(n)} - \hat{y}^{(n)}_{w_t})$$

No.

Date.

3.3 softmax 回归

条件概率

$$p(y=c|\vec{x}) = \text{softmax}(\vec{w}_c^T \vec{x}) \quad (3.28)$$

$$= \frac{\exp(\vec{w}_c^T \vec{x})}{\sum_{c=1}^{C} \exp(\vec{w}_c^T \vec{x})} \quad (3.29)$$

$\vec{w}_c$ 第 c 类的权重向量

决策函数

$$\hat{y} = \arg\max_c \; p(y=c|\vec{x}) \quad (3.30)$$

$$= \arg\max_c \; \vec{w}_c^T \vec{x} \quad (3.31)$$

向量表示

$$\hat{\vec{y}} = \text{softmax}(W^T \vec{x})$$

$$= \frac{\exp(W^T \vec{x})}{1^T \exp(W^T \vec{x})}$$

W 是由 $[w_1, \cdots, w_C]$ C 个类的权重向量组成的矩阵

$$W^T = \begin{bmatrix} w_1 \\ \vdots \\ w_C \end{bmatrix} \qquad \vec{x} = [x_1, \cdots, x_d]$$

$\hat{y}$ 为所有类别的预测条件概率组成的向量

$$\begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_C \end{bmatrix}$$

No.

Date. . .

⇒3.3.1 参数学习

$$R(W) = -\frac{1}{N}\sum_{n\in E}\sum_{c} \tilde{y}_c^{(n)} \log \hat{y}_c^{(n)}$$

$$= -\frac{1}{N}\sum_{n} (y^{(n)})^T \log \hat{y}^{(n)}$$

$\hat{y}^{(n)}$ 为样本 $x^{(n)}$ 在每个类别的后验概率

$$\frac{\partial R(W)}{\partial W} = -\frac{1}{N}\sum_{n} x^{(n)} (y^{(n)} - \hat{y}^{(n)})^T$$

No.

Date.

3.4 感知器

$$\hat{y} = sgn(\vec{w}^T \vec{x})$$

⇒ 3.4.1 参数学习.

$$y^{(n)} \vec{w}^T x^{(n)} > 0$$

$$\vec{w} \leftarrow \vec{w} + y\vec{x}$$

$$L(\vec{w}; x, y) = \max(0, -y\vec{w}^T\vec{x}) \qquad (3.57)$$

$$\frac{\partial L(\vec{w}; x, y)}{\partial \vec{w}} = \begin{cases} 0 & \text{if } \vec{w}^T\vec{x} > 0 \\ -y\vec{x} & \text{if } \vec{w}^T\vec{x} < 0 \end{cases}$$

⇒ 3.4.2 感知器的收敛性.

R 是训练集中最大的特征向量的模

$$R = \max_n \|x^{(n)}\| \qquad \gamma \text{ 是间隔}$$

如果训练集D线性可分，感知机权重更新次数不超过 $\frac{R^2}{\gamma^2}$

⇒ 3.4.3 参数平均感知器

No.

Date.

3.5 支持向量机

在D上，$y_n \in \{+1, -1\}$，存在一个超平面 $\vec{w}^T\vec{x}+b=0$ 将两类样本分开，使得对每个样本都有 $y^{(n)}(\vec{w}^T\vec{x}^{(n)}+b)>0$

样本 $\vec{x}^{(n)}$ 到分割超平面的距离为

$$r^{(n)} = \frac{|\vec{w}^T\vec{x}^{(n)}+b|}{\|\vec{w}\|} = \frac{y^{(n)}(\vec{w}^T\vec{x}^{(n)}+b)}{\|\vec{w}\|}$$

间隔：整个数据集D中所有样本到分割超平面的最短距离

$$r = \min_n r^{(n)}$$

SVM的目标是寻找一个超平面 $(\vec{w}^*, b^*)$ 使得 r 最大，即

$$\max_{\vec{w},b} \quad r$$

$$\text{s.t.} \quad \frac{y^{(n)}(\vec{w}^T\vec{x}^{(n)}+b)}{\|\vec{w}\|} \geq r$$

令 $\|\vec{w}\| \cdot r = 1$，则

$$\max_{\vec{w},b} \quad \frac{1}{\|\vec{w}\|^2}$$

$$\text{s.t.} \quad y^{(n)}(\vec{w}^T\vec{x}^{(n)}+b) \geq 1$$

支持向量

D中所有满足 $y^{(n)}(\vec{w}^T\vec{x}^{(n)}+b)=1$ 的样本点。

No.

Date.

⇒ 3.5.1 参数学习

凸优化问题

$$\min_{\vec{w},b} \ \frac{1}{2}\|w\|^2$$

$$\text{s.t.} \quad 1 - y^{(n)}(\vec{w}^T\vec{x} + b) \le 0$$

拉格朗日函数为

$$\Lambda(\vec{w}, b, \eta) = \frac{1}{2}\|w\|^2 + \sum_n \eta_n (1 - y^{(n)}(\vec{w}^T\vec{x}^{(n)} + b)) \qquad (3.88)$$

计算 $\frac{\partial \Lambda}{\partial w}$ 和 $\frac{\partial \Lambda}{\partial b}$ 得到

$$\vec{w} = \sum_n \eta_n y^{(n)} \vec{x}^{(n)} \qquad (3.89)$$

$$0 = \sum_n \eta_n y^{(n)} \qquad (3.90)$$

将式(3.89)代入式(3.88)，并利用式(3.90)得到

拉格朗日对偶函数

$$\Gamma(\eta) = -\frac{1}{2}\sum_n \sum_m \eta_m \eta_n y^{(m)} y^{(n)} (\vec{x}^{(m)})^T \vec{x}^{(n)} + \sum_n \eta_n$$

最大化对偶函数 $\max_{\eta \ge 0} \Gamma(\eta)$

自主学习补充

No.

Date.

根据KKT条件，最优解需满足 $\lambda_n^*(1-y^{(n)}(\vec{w}^*\vec{x}^{(n)}+b^*))=0$.

若样本 $\vec{x}^{(n)}$ 不在间隔边界上，$\lambda_n^*=0$，该样本无效.

若样本 $\vec{x}^{(n)}$ 在间隔边界上，$\lambda_n^*>0$，这些点被称为支持向量.

计算出 λ^* 后，根据公式(3.8)计算出最优权重 $\vec{w}^*$，b^* 可以通过任选一个支持向量 $(\vec{x},y)$ 得到.

$$b^*=y-\vec{w}^{*T}\vec{x} \tag{3.92}$$

SVM的决策函数为

$$f(\vec{x})=\mathrm{sgn}(\vec{w}^{*T}\vec{x}+b^*)$$

$$=\mathrm{sgn}\left(\sum \lambda_n^* y^{(n)}(\vec{x}^{(n)})^T\vec{x}+b^*\right)$$

SVM只依赖 $\lambda_n^*>0$ 的样本，即支持向量.

⇒ 3.5.2 核函数

核函数 低维→高维，不可分→线性可分.

$$f(\vec{x})=\mathrm{sgn}(\vec{w}^{*T}\phi(\vec{x})+b^*)$$

$$=\mathrm{sgn}\left(\sum \lambda_n^* y^{(n)}K(\vec{x}^{(n)},\vec{x})+b^*\right)$$

No.

Date. . .

$k(\vec{x},\vec{z}) = \phi(\vec{x})^T\phi(\vec{z})$ 为核函数，不需要显式给出 $\phi(\vec{x})$ 的具体形式

$k(\vec{x},\vec{z}) = (1+\vec{x}^T\vec{z})^2 = \phi(\vec{x})\phi(\vec{z})$

⇒ 3.3.3 软间隔

引入松弛变量.

$$\min_{\vec{w},b} \ \frac{1}{2}\|w\|^2 + C\cdot\sum \xi_n$$

$$s.t. \quad 1 - y^{(n)}(\vec{w}^T\vec{x}^{(n)}+b) - \xi_n \le 0 \qquad (3.98)$$

$$\xi_n \ge 0$$

C 控制间隔和松弛变量惩罚的平衡了

软间隔 引入 ξ_n 的间隔

公式 (3.98) 可以表示为经验风险+正则化项的形式

$$\min_{\vec{w},b} \sum \max(0, 1-y^{(n)}(\vec{w}^T\vec{x}^{(n)}+b)) + \frac{1}{C}\cdot\frac{1}{2}\|w\|^2$$

$\max(0, 1-y^{(n)}(\vec{w}^T\vec{x}^{(n)}+b))$ hinge 损失函数

$\frac{1}{C}$ 正则化系数

最终决策函数只和支持向量有关，即当 $1-y^{(n)}(\vec{w}^T\vec{x}^{(n)}+b)-\xi_n = 0$ 的样本.

No.

Date. . .

3.6 损失函数对比

LR, Perceptron, SVM

统一定义标签 $y \in \{+1, -1\}$

LR

$$L_{LR} = -\log p(y|\vec{x})$$

$$= -\mathbb{I}(y=1)\log\sigma(f(\vec{x},\vec{w})) - \mathbb{I}(y=-1)\log\sigma(-f(\vec{x},\vec{w})) \qquad 1-\sigma(x) = \sigma(-x)$$

$$= -\mathbb{I}(y=1)\log\sigma(x) - \mathbb{I}(y=-1)\log(1-\sigma(x))$$

$$= -\mathbb{I}(y=1)\log\sigma(x) - (1-\mathbb{I}(y=1))\log(1-\sigma(x))$$

$$= -\mathbb{I}(y=1)\log\sigma(x) - \log(1-\sigma(x)) + \mathbb{I}(y=1)\log(1-\sigma(x))$$

$$= \mathbb{I}(y=1)\log\frac{1-\sigma(x)}{\sigma(x)} - \log(1-\sigma(x))$$

$$= \log\left(\frac{1}{\sigma(x)}\right)^{y} - \log(1-\sigma(x))^{(1-y)}$$

$$= \log\sigma(x)^{-y} - \log(1-\sigma(x))^{(1-y)}$$

$$= \log\frac{\sigma(x)^{-y}}{\sigma(x)(1-\sigma(x))^{-y}} = \log\frac{1}{\sigma(x)}\cdot\left(\frac{1-\sigma(x)}{\sigma(x)}\right)^{y}$$

No.

Date.

$L_{LR} = \log(1+\exp(-yf(\vec{x},\vec{w})))$

$L_{perceptron} = \max(0, -yf(\vec{x},\vec{w}))$

软间隔的支持向量机

$L_{hinge} = \max(0, 1-yf(\vec{x},\vec{w}))$

平方损失可以写为

$L_{squared} = (y-f(\vec{x},\vec{w}))^2$

$= 1 - 2yf(\vec{x},\vec{w}) + (yf(\vec{x},\vec{w}))^2 \qquad y^2=1$

$= (1-yf(\vec{x},\vec{w}))^2$

一个好的损失函数在误随着 $yf(\vec{x},\vec{w})$ 增大而减小