第2章　感知机

2019.10.18

目录

content

感知机模型

感知机模型

定义：假设输入空间（特征空间）是$X \subseteq R^n$，输出空间是$Y=\{+1,-1\}$。输入$x \in X$表示实例的特征向量，对应于输入空间（特征空间）的点；输出$y \in Y$表示实例的类别。由输入空间到输出空间的如下函数：

$$f(x) = sign(w * x + b)$$

$w \in R^n$ 权值（权值向量）

$b \in R$ 偏置

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

感知机是二类分类的线性分类模型，属于判别模型。

感知机模型

$$w * x + b = 0$$

对应于特征空间R^n 中的一个超平面，其中w是超平面的法向量，b是超平面的截距。

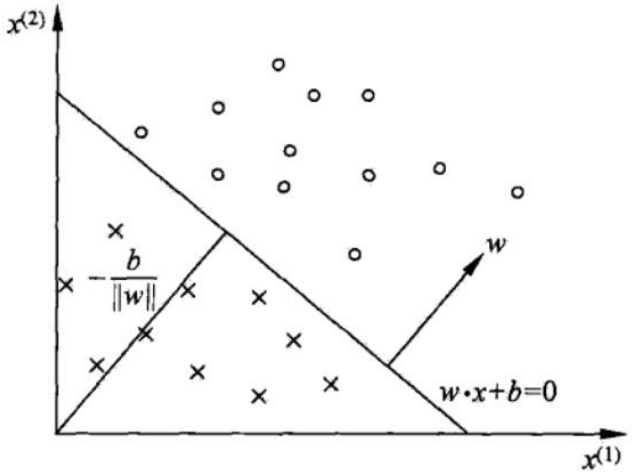


图 2.1　感知机模型

证明：在超平面上任意取两点x_1,x_2,假设$\vec{w}$是平面的法向量。

$$\begin{cases} \vec{w} * x_1 + b = 0 \quad (1) \\ \vec{w} * x_2 + b = 0 \quad (2) \end{cases}$$

由(1)-(2)式得　$\vec{w} * x_1 x_2 = 0$

感知机学习策略

数据集的线性可分性

线性可分数据集

若存在某个超平面S，能够将数据集的正实例点和负实例点完全正确地划分到超平面的两侧，即对所有$y_i = +1$的实例i，有$w * x + b > 0$，对所有$y_i = -1$的实例i，有$w * x + b < 0$,则称数据集T为线性可分数据集。

假设数据集是线性可分的，我们的目的是找到一个能够将训练集正实例和负实例完全正确分开的分离超平面。

感知机学习策略

确定一个学习策略，定义（经验）损失函数并将损失函数极小化。

① 误分类点的总数 （不易优化）
② 误分类点到超平面S的总距离 ★

对于标记为+1的点，若$w*x+b<0$
对于标记为-1的点，若$w*x+b>0$

$-y_i\ (w*x+b)>0$

输出空间R^n 中任一点x_0到超平面S的距离：

$$\frac{1}{||w||}|w*x_0+b|$$

||w||是w的L_2范数

范数定义：$\|x\|_2=\sqrt{\sum_i x_i^2}$

对于误分类的数据(χ_i,y_i) $-y_i\ (w*x+b)>0$

因此误分类点x_i到超平面的距离为 $$-\frac{1}{\|w\|}y_i(w*x_i+b)$$

假设误分类点集合为M，则它们到超平面的总距离为 $$-\frac{1}{\|w\|}\sum_{x_i\epsilon M}y_i(w*x_i+b)$$

感知机学习策略

输出空间R^n 中任一点x_0到超平面S的距离：

$$\frac{1}{||w||}|w * x_0 + b|$$

对于误分类的数据(χ_i, y_i)

$$-y_i\ (w * x + b) > 0$$

因此误分类点x_i到超平面的距离为

$$-\frac{1}{\|w\|} y_i (w * x_i + b)$$

假设误分类点集合为M，则它们到超平面的总距离为

$$-\frac{1}{\|w\|} \sum_{x_i \in M} y_i (w * x_i + b)$$

如果不考虑$\frac{1}{\|w\|}$，得到的损失函数

$$L(\omega, b) = -\sum_{x_i \in M} y_i (w * x_i + b)$$

感知机学习算法

- 原始形式
- 算法的收敛性
- 对偶形式

原始形式

给定一个训练数据集 $T = \{(x_1, y_1), (x_2 y_2), \ldots, (x_N, y_N)\}$ 其中 $x_i \in X = R^n$ $y_i \in y = \{-1, 1\}$

求参数w, b, 使其成为以下损失函数极小化问题的解

$$min_{w,b} L(\omega, b) = -\sum_{x_i \in M} y_i (w * x_i + b)$$

对w进行求导

$$\nabla_w L_{(w,b)} = -\sum_{x_i \in M} y_i x_i$$

对b进行求导

$$\nabla_b L_{(w,b)} = -\sum_{x_i \in M} y_i$$

随机选取一个误分类点(x_i, y_i), 对w, b进行更新

$$w \leftarrow w + \eta y_i x_i$$

$$b \leftarrow b + \eta y_i$$

η: 学习率 $(0 < \eta \ll 1)$

通过迭代可以期待损失函数$L(\omega, b)$不断减小，直到为0.

原始形式

给定一个训练数据集 $T = \{(x_1, y_1), (x_2 y_2), \ldots, (x_N, y_N)\}$ 其中 $x_i \in X = R^n$ $y_i \in y = \{-1, 1\}$

输出:w, b;感知机模型 $f(x) = \text{sign}(w \cdot x + b)$

01 选取初值w_0, b_0

02 在训练集中选取数据（x_i, y_i）

03 如果$y_i (w * x_i + b) \leq 0$

$$w \leftarrow w + \eta y_i x_i$$
$$b \leftarrow b + \eta y_i$$

04 转至（2），直到训练集中没有误分类点

算法收敛性证明

把偏置b并入向量w，记作$\hat{w} = (\omega^T, b)^T$，输入向量并入1，记作$\hat{x} = (x^T, 1)^T$

设训练数据集$T = \{(x_1, y_1), (x_2 y_2), \dots, (x_N, y_N)\}$是线性可分的，其中$x_i \in X = R^n$ $y_i \in y = \{-1,1\}$

（1）存在满足条件$\|\hat{w}_{opt}\| = 1$的超平面$\hat{w}_{opt} \cdot \hat{x} = w_{0pt} \cdot x + b_{opt}$将训练数据集完全正确分开；且存在，对于所有i=1, 2, …N

$$y_i(\hat{w}_{opt} \cdot \hat{x}_i) = y_i \cdot (w_{opt} \cdot x_i + b_{opt}) \geq \gamma$$

（2）令$R = \max_{1 \ll i \ll N} \|\hat{x}_i\|$，则感知及算法在训练数据集上的误分类次数k满足不等式

$$k \leq \left(\frac{R}{\gamma}\right)^2$$

算法收敛性证明

（1）存在满足条件$\|\hat{w}_{opt}\| = 1$的超平面$\hat{w}_{opt} \cdot \hat{x} = w_{0pt} \cdot x + b_{opt}$=0将训练数据集完全正确分开；且存在，对于所有i=1, 2, …N

$$y_i(\hat{w}_{opt} \cdot \hat{x}_i) = y_i \cdot (w_{opt} \cdot x_i + b_{opt}) \geq \gamma$$

存在超平面可以将数据集完全正确分离，取该超平面$\hat{w}_{opt} \cdot \hat{x} = w_{0pt} \cdot x + b_{opt}$=0 使$\|\hat{w}_{opt}\|$=1

$$y_i(\hat{w}_{opt} \cdot \hat{x}_i) = y_i \cdot (w_{opt} \cdot x_i + b_{opt}) > 0$$

所以存在 $\gamma = \min_i \{y_i \cdot (w_{opt} \cdot x_i + b_{opt})\}$

感知机算法从$\hat{w}_0$=0 开始，如果实例被误分类，则更新权重。令$\hat{w}_{k-1}$是第k个误分类实例之前的扩充权重向量 $\hat{w}_{k-1} = ({w_{k-1}}^T, b_{k-1})^T$

则第k个误分类的条件是 $y_i(\hat{w}_{k-1} \cdot \hat{x}_i) = y_i \cdot (w_{k-1} \cdot x_i + b_{k-1}) \leq 0$

若(x_i, y_i)是被$\hat{w}_{k-1} = ({w_{k-1}}^T, b_{k-1})^T$误分类的点，则w和b的更新是

$$w_k \leftarrow w_{k-1} + \eta y_i x_i$$

$$b_k \leftarrow b_{k-1} + \eta y_i$$

$$\Longrightarrow \quad \hat{\omega}_k = \hat{w}_{k-1} + \eta y_i x_i$$

下面开始推导公式

算法收敛性证明

(1) 存在满足条件$\|\hat{w}_{opt}\| = 1$的超平面$\hat{w}_{opt} \cdot \hat{x} = w_{0pt} \cdot x + b_{opt}$=0将训练数据集完全正确分开；且存在，对于所有i=1, 2, …N　　$y_i(\hat{w}_{opt} \cdot \hat{x}_i) = y_i \cdot (w_{opt} \cdot x_i + b_{opt}) \geq \gamma$

(2) 令R= $\max_{1 \ll i \ll N} \|\hat{x}_i\|$, 则感知机算法在训练数据集上的误分类次数k满足不等式　　$k \leq \left(\frac{R}{\gamma}\right)^2$

$\hat{\omega}_k = \hat{w}_{k-1} + \eta y_i x_i$　　$y_i(\hat{w}_{opt} \cdot \hat{x}_i) = y_i \cdot (w_{opt} \cdot x_i + b_{opt}) \geq \gamma$

$$\begin{aligned} \hat{w}_k \cdot \hat{w}_{opt} &= \hat{w}_{k-1} \cdot \hat{w}_{opt} + \eta y_i \hat{w}_{opt} \cdot x_i \\ &\geq \hat{w}_{k-1} \cdot \hat{w}_{opt} + \eta\gamma \\ &\geq \cdots \\ &\geq k\eta\gamma \end{aligned}$$

由递推得到不等式　　$\hat{w}_k \bullet \hat{w}_{opt} \geq \hat{w}_{k-1} \bullet \hat{w}_{opt} + \eta\gamma \geq \hat{w}_{k-2} \bullet \hat{w}_{opt} + 2\eta\gamma \geq \cdots \geq k\eta\gamma$

$$\begin{aligned} ||\hat{w}_k||^2 &= ||\hat{w}_{k-1} + \eta y_i x_i||^2 \\ &= ||\hat{w}_{k-1}||^2 + \eta^2 ||x_i||^2 + 2\eta \hat{w}_{k-1} y_i x_i \\ &\leq ||\hat{w}_{k-1}||^2 + \eta^2 R^2 \\ &\leq \cdots \\ &\leq k\eta^2 R^2 \end{aligned}$$

柯西-施瓦兹不等式$|(x, y)| \leq \|x\| \cdot \|y\|$

$\|\hat{w}_{opt}\| = 1$

$$k\eta\gamma \leq \hat{w}_k \cdot \hat{w}_{opt} \leq ||\hat{w}_k|| ||\hat{w}_{opt}|| \leq \sqrt{k}\eta R$$

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

误分类的次数是有上界的

对偶形式

由原始形式得，随机选取一个误分类点(x_i, y_i)，对w，b进行更新

$$w \leftarrow w + \eta y_i x_i$$

$$b \leftarrow b + \eta y_i$$

η：学习率 $(0 < \eta \ll 1)$

修改n次，w，b的增量表示为

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i$$

$$\mathrm{b} = \sum_{i=1}^{N} \alpha_i y_i$$

$\alpha_i = n_i \eta$，η=1时，α_i表示为第i个误分类点进行更新的次数

感知机模型

$$f(x) = \mathrm{sign}(w \cdot x + b)$$

$$f(x) = sign\left(\sum_{j=1}^{N} \alpha_j y_j x_j \cdot x + b\right)$$

对偶形式

给定一个训练数据集 $T=\{(x_1,y_1),(x_2y_2),\dots,(x_N,y_N)\}$ 其中 $x_i \in X=R^n$ $y_i \in y=\{-1,1\}$

输出:w, b;感知机模型 $f(x)=sign\left(\sum_{j=1}^{N}\alpha_j y_j x_j \cdot x+b\right)$,其中$\alpha=(\alpha_1,\alpha_2,\dots,\alpha_N)^T$

01 $\alpha \leftarrow \mathbf{0}, b \leftarrow \mathbf{0}$

02 在训练集中选取数据 (x_i, y_i)

03 如果$y_i\left(\sum_{j=1}^{N}\alpha_j y_j x_j * x_i+b\right) \ll 0$

$\alpha_i \leftarrow \alpha_i+\eta$

$b \leftarrow b+\eta y_i$

04 转至（2），直到训练集中没有误分类点

内积 $G=[x_i \cdot x_j]_{N*N}$

$G=x \cdot x^T$

$$\begin{bmatrix}3 & 3\\4 & 3\\1 & 1\end{bmatrix}\cdot\begin{bmatrix}3 & 4 & 1\\3 & 3 & 1\end{bmatrix}=\begin{bmatrix}18 & 21 & 6\\21 & 25 & 7\\6 & 7 & 2\end{bmatrix}$$

代码实现

截图

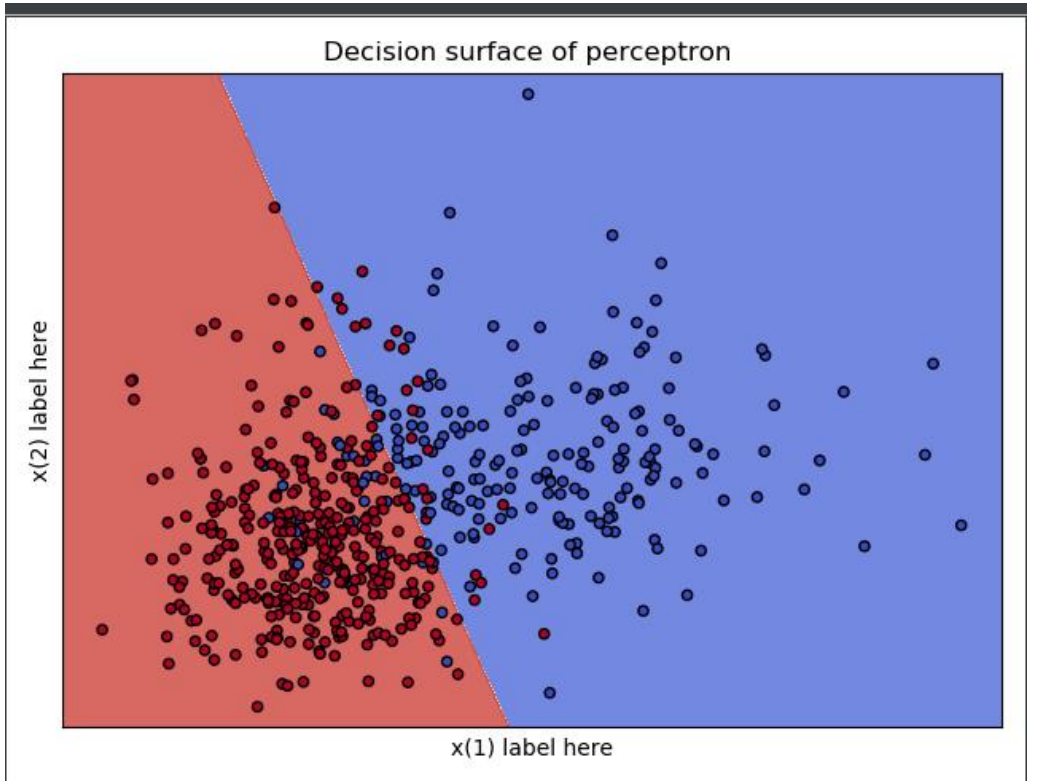
```
X, y = load_breast_cancer(return_X_y=True)
X = X[:, 0:2]


def make_meshgrid(x, y, h=.02):
    x_min, x_max = x.min() - 1, x.max() + 1
    y_min, y_max = y.min() - 1, y.max() + 1
    xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))
    return xx, yy


def plot_contours(ax, clf, xx, yy, **params):
    Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
    Z = Z.reshape(xx.shape)
    out = ax.contourf(xx, yy, Z, **params)
    return out


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=44, stratify=y)
clf = Perceptron(tol=None, random_state=44, verbose=1)
clf.fit(X_train, y_train)  # fit表示训练的意思
print("train accuracy: ", clf.score(X_train, y_train))  # train accuracy, bias
y_pred = clf.predict(X_test)  # test examples
print("test accuracy: ", accuracy_score(y_test, y_pred))
fig, ax = plt.subplots()
# title for the plots
title = ('Decision surface of perceptron ')
# Set-up grid for plotting.
X0, X1 = X[:, 0], X[:, 1]
xx, yy = make_meshgrid(X0, X1)

plot_contours(ax, clf, xx, yy, cmap=plt.cm.coolwarm, alpha=0.8)
ax.scatter(X0, X1, c=y, cmap=plt.cm.coolwarm, s=20, edgecolors='k')
ax.set_ylabel('x(2) label here')
ax.set_xlabel('x(1) label here')
ax.set_xticks(())
ax.set_yticks(())
ax.set_title(title)
plt.show()
```




```
Total training time: 0.06 seconds.
train accuracy:  0.8793969849246231
test accuracy:  0.935672514619883
```

感 谢 您 的 观 看

THE PROFESSIONL POWERPOINT TEMPLATE