线性判别函数和决策边界

两类分类

线性判别函数 f(**x**, **w**)=$\mathbf{w}^T\mathbf{x}$+b。特征空间 R_d 中所有满足 f(**x**, **w**)=0 的点组成用一个分割超平面（hyperplane），称为决策边界（decision boundary）或决策平面（decision surface）。

所谓“线性分类模型”就是指其决策边界是线性超平面。在特征空间中，决策平面与权重向量 w 正交。特征空间中每个样本点到决策平面的有向距离（signed distance）为

$$\gamma = \frac{f(\mathbf{x}, \mathbf{w})}{\|\mathbf{w}\|}.$$

定义 3.1 – 两类线性可分： 对于训练集 $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$，如果存在权重向量 $\mathbf{w}^*$，对所有样本都满足 $yf(\mathbf{x}, \mathbf{w}^*) > 0$，那么训练集 $\mathcal{D}$ 是线性可分的。

多分类

1. “一对其余”方式：把多类分类问题转换为 C 个“一对其余”的两类分类问题。这种方式共需要 C 个判别函数，其中第 c 个判别函数 f_c 是将类 c 的样本和不属于类 c 的样本分开。
2. “一对一”方式：把多类分类问题转换为 $C(C-1)/2$ 个“一对一”的两类分类问题。这种方式共需要 $C(C-1)/2$ 个判别函数，其中第 (i, j) 个判别函数是把类 i 和类 j 的样本分开。
3. “*argmax*”方式：这是一种改进的“一对其余”方式，共需要 C 个判别函数

$$f_c(\mathbf{x}, \mathbf{w}_c) = \mathbf{w}_c^{\mathrm{T}}\mathbf{x} + b_c, \qquad c = [1, \cdots, C] \tag{3.10}$$

如果存在类别 c，对于所有的其他类别 $\tilde{c}(\tilde{c} \neq c)$ 都满足 $f_c(\mathbf{x}, \mathbf{w}_c) > f_{\tilde{c}}(\mathbf{x}, \mathbf{w}_{\tilde{c}})$，那么 $\mathbf{x}$ 属于类别 c。即

$$y = \arg\max_{c=1}^{C} f_c(\mathbf{x}, \mathbf{w}_c). \tag{3.11}$$

定义 3.2 – 多类线性可分： 对于训练集 $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$，如果存在 C 个权重向量 $\mathbf{w}_c^*, 1 \le c \le C$，对所有第 c 类的样本都满足 $f_c(\mathbf{x}, \mathbf{w}_c) > f_{\tilde{c}}(\mathbf{x}, \mathbf{w}_{\tilde{c}}), \forall \tilde{c} \neq c$，那么训练集 $\mathcal{D}$ 是线性可分的。

如果数据集可以多类线性可分的，那么一定存在一个“argmax”方式的线性分类器可以将它们正确分开。

Logistic 回归(处理两类分类问题的线性模型)

为了解决连续的线性函数不适合进行分类的问题，我们引入非线性函数 g:R^{d}→(0,1)来预测类别标签的后验概率 p(y=1|**x**)。

$$p(y=1|\mathbf{x})=g(f(\mathbf{x},\mathbf{w})), \tag{3.12}$$

其中 $g(\cdot)$ 通常称为*激活函数*（activation function），其作用是把线性函数的值域从实数区间“挤压”到了 $(0,1)$ 之间，可以用来表示概率。在统计文献中，$g(\cdot)$ 的逆函数 $g^{-1}(\cdot)$ 也称为*联系函数*（link function）。

在 logistic 回归中，我们使用 logistic 函数来作为激活函数。标签 $y=1$ 的后验概率为

$$\begin{aligned}p(y=1|\mathbf{x})&=\sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x}) &(3.13)\\ &\triangleq\frac{1}{1+\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x})}, &(3.14)\end{aligned}$$

标签 $y=0$ 的后验概率为

$$\begin{aligned}p(y=0|\mathbf{x})&=1-p(y=1|\mathbf{x}) &(3.15)\\ &=\frac{\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x})}{1+\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x})}. &(3.16)\end{aligned}$$

将公式 (3.14) 进行变换后得到

$$\begin{aligned}\mathbf{w}^{\mathrm{T}}\mathbf{x}&=\log\frac{p(y=1|\mathbf{x})}{1-p(y=1|\mathbf{x})} &(3.17)\\ &=\log\frac{p(y=1|\mathbf{x})}{p(y=0|\mathbf{x})}, &(3.18)\end{aligned}$$

参数学习

采用交叉熵作为损失函数，并使用梯度下降法或者牛顿法来对参数进行优化。

给定 N 个训练样本 $\{(\mathbf{x}^{(n)},y^{(n)})\}_{n=1}^{N}$，用 logistic 回归模型对每个样本 $\mathbf{x}^{(n)}$ 进行预测，并用输出 $\mathbf{x}^{(n)}$ 的标签为1的后验概率，记为 $\hat{y}^{(n)}$，

$$\hat{y}^{(n)}=\sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)}),\qquad 1\leq n\leq N. \tag{3.19}$$

由于 $y^{(n)}\in\{0,1\}$，样本 $(\mathbf{x}^{(n)},y^{(n)})$ 的真实条件概率可以表示为

$$\begin{aligned}p_r(y^{(n)}=1|\mathbf{x}^{(n)})&=y^{(n)}, &(3.20)\\ p_r(y^{(n)}=0|\mathbf{x}^{(n)})&=1-y^{(n)}. &(3.21)\end{aligned}$$

使用交叉熵损失函数，其风险函数为：

$$\begin{aligned}\mathcal{R}(\mathbf{w})&=-\frac{1}{N}\sum_{n=1}^{N}\left(p_r(y^{(n)}=1|\mathbf{x}^{(n)})\log\hat{y}^{(n)}+p_r(y^{(n)}=0|\mathbf{x}^{(n)})\log(1-\hat{y}^{(n)})\right) &(3.22)\\ &=-\frac{1}{N}\sum_{n=1}^{N}\left(y^{(n)}\log\hat{y}^{(n)}+(1-y^{(n)})\log(1-\hat{y}^{(n)})\right). &(3.23)\end{aligned}$$

Softmax 回归（logistic 回归在多类分类问题上的推广）

对于多类问题，类别标签 $y \in \{1, 2, \cdots, C\}$ 可以有 C 个取值。给定一个样本 $\mathbf{x}$，softmax 回归预测的属于类别 c 的条件概率为：

$$p(y = c|\mathbf{x}) = \mathrm{softmax}(\mathbf{w}_c^{\mathrm{T}}\mathbf{x}) \tag{3.28}$$

$$= \frac{\exp(\mathbf{w}_c^{\top}\mathbf{x})}{\sum_{c=1}^{C}\exp(\mathbf{w}_c^{\top}\mathbf{x})}, \tag{3.29}$$

其中 $\mathbf{w}_c$ 是第 c 类的权重向量。

Softmax 回归的决策函数可以表示为：

$$\hat{y} = \arg\max_{c=1}^{C} p(y = c|\mathbf{x}) \tag{3.30}$$

$$= \arg\max_{c=1}^{C} \mathbf{w}_c^{\mathrm{T}}\mathbf{x}. \tag{3.31}$$

采用交叉熵损失函数，softmax 回归模型的风险函数为：

$$\mathcal{R}(W) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C}\mathbf{y}_c^{(n)}\log\hat{\mathbf{y}}_c^{(n)} \tag{3.38}$$

$$= -\frac{1}{N}\sum_{n=1}^{N}(\mathbf{y}^{(n)})^{\mathrm{T}}\log\hat{\mathbf{y}}^{(n)}, \tag{3.39}$$

其中 $\hat{\mathbf{y}}^{(n)} = \mathrm{softmax}(W^{\mathrm{T}}\mathbf{x}^{(n)})$ 为样本 $\mathbf{x}^{(n)}$ 在每个类别的后验概率。

风险函数 $\mathcal{R}(W)$ 关于 W 的梯度为

$$\frac{\partial\mathcal{R}(W)}{\partial W} = -\frac{1}{N}\sum_{n=1}^{N}\mathbf{x}^{(n)}\left(\mathbf{y}^{(n)} - \hat{\mathbf{y}}^{(n)}\right)^{\mathrm{T}}. \tag{3.40}$$

softmax 回归中使用的 C 个权重向量是冗余的，即对所有的权重向量都减去一个同样的向量 $\mathbf{v}$，不改变其输出结果。因此，softmax 往往需要使用正则化来约束其参数。此外，我们可以利用这个特性来避免计算 softmax 函数时在数值计算上溢出问题。

感知器（一种错误驱动的在线学习算法）

根据感知器的学习策略，可以反推出感知器的损失函数为：

$$\mathcal{L}(\mathbf{w}; \mathbf{x}, y) = \max(0, -y\mathbf{w}^{\mathrm{T}}\mathbf{x}). \tag{3.57}$$

采用随机梯度下降，其每次更新的梯度为

$$\frac{\partial\mathcal{L}(\mathbf{w}; \mathbf{x}, y)}{\partial\mathbf{w}} = \begin{cases} 0 & \text{if} \quad y\mathbf{w}^{\mathrm{T}}\mathbf{x} > 0, \\ -y\mathbf{x} & \text{if} \quad y\mathbf{w}^{\mathrm{T}}\mathbf{x} < 0. \end{cases} \tag{3.58}$$

算法 3.1: 两类感知器算法

输入: 训练集 $\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$，迭代次数 T

初始化：$\mathbf{w}_0 \leftarrow 0,\ k \leftarrow 0$;

for $t = 1 \cdots T$ **do**

  随机对训练样本进行随机排序;

  for $n = 1 \cdots N$ **do**

    选取一个样本 $(\mathbf{x}^{(n)}, y^{(n)})$;

    if $\mathbf{w}_k^{\mathrm{T}}(y^{(n)}\mathbf{x}^{(n)}) \leq 0$ **then**

      $\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k + y^{(n)}\mathbf{x}^{(n)}$;

      $k \leftarrow k + 1$;

    end

  end

end

输出: $\mathbf{w}_k$

收敛性

如果训练集是线性可分的，那么感知器算法可以在有限次迭代后收敛。然而，如果训练集不是线性分隔的，那么这个算法则不能确保会收敛。

当数据集是两类线性可分时，对于训练集 $\mathcal{D} = \left\{(\mathbf{x}^{(n)}, y^{(n)})\right\}_{n=1}^{N}$，其中 $\mathbf{x}^{(n)}$ 为样本的增广特征向量，$y^{(n)} \in \{-1, 1\}$，那么存在一个正的常数 $\gamma(\gamma > 0)$ 和权重向量 $\mathbf{w}^*$，并且 $\|\mathbf{w}^*\| = 1$，对所有 n 都满足 $(\mathbf{w}^*)^{\mathrm{T}}(y^{(n)}\mathbf{x}^{(n)}) \geq \gamma$。我们可以证

定理 3.1 – 感知器收敛性： 给定一个训练集 $\mathcal{D} = \left\{(\mathbf{x}^{(n)}, y^{(n)})\right\}_{n=1}^{N}$，假设 R 是训练集中最大的特征向量的模，

$$R = \max_n \|x^{(n)}\|.$$

如果训练集 $\mathcal{D}$ 线性可分，感知器学习算法3.1的权重更新次数不超过 $\frac{R^2}{\gamma^2}$。

对于 $w_k = w_{k-1} + y^{(k)}x^{(k)} = \sum_{k=0}^{K} y^{(k)}x^{(k)}$（初始化为 0，K 次分类错误后更新的权重向量大小）：

$y_k \mathrm{w}_{K-1}^{\mathrm{T}} \mathrm{x}^{(K)} \leq 0.$

$$\|\mathrm{w}_K\|^2 = \|\mathrm{w}_{K-1} + y^{(K)}\mathrm{x}^{(K)}\|^2 \tag{3.61}$$

$$= \|\mathrm{w}_{K-1}\|^2 + \|y^{(K)}\mathrm{x}^{(K)}\|^2 + 2y^{(K)}\mathrm{w}_{K-1}^{\mathrm{T}}\mathrm{x}^{(K)} \tag{3.62}$$

$$\leq \|\mathrm{w}_{K-1}\|^2 + R^2 \tag{3.63}$$

$$\leq \|\mathrm{w}_{K-2}\|^2 + 2R^2 \tag{3.64}$$

$$\leq KR^2 \tag{3.65}$$

$$\|\mathbf{w}_K\|^2 = \boxed{\|\mathbf{w}^*\|^2} \cdot \|\mathbf{w}_K\|^2 \tag{3.66}$$

$\|\mathbf{w}^*\| = 1.$

$$\geq \|\mathbf{w}^{*\mathrm{T}}\mathbf{w}_K\|^2 \tag{3.67}$$

两个向量内积的平方一定小于等于这两个向量的模的乘积

$$= \|\mathbf{w}^{*\mathrm{T}}\sum_{k=1}^{K}(y^{(k)}\mathbf{x}^{(k)})\|^2 \tag{3.68}$$

$$= \|\sum_{k=1}^{K}\mathbf{w}^{*\mathrm{T}}(y^{(k)}\mathbf{x}^{(k)})\|^2 \tag{3.69}$$

$\mathbf{w}^{*\mathrm{T}}(y^{(n)}\mathbf{x}^{(n)}) \geq \gamma, \forall n.$

$$\geq K^2\gamma^2. \tag{3.70}$$

缺陷：1）在数据集线性可分时，感知器虽然可以找到一个超平面把两类数据分开，但并不能保证能其泛化能力（判别函数是最优的）；2）感知器对样本顺序比较敏感。每次迭代的顺序不一致时，找到的分割超平面也往往不一致（在迭代次序上排在后面的错误样本，比前面的错误样本对最终的权重向量影响更大）；3）如果训练集不是线性可分的，就永远不会收敛。

平均感知器

1）投票感知器（Voted Perceptron）：记录第 k 次更新后得到的权重 w_k 在之后的训练过程中正确分类样本的次数 c_k。虽然提高了感知器的泛化能力，但是需要保存 K 个权重向量。在实际操作中会带来额外的开销。

$$\hat{y} = \mathrm{sgn}\left(\sum_{k=1}^{K} c_k\,\mathrm{sgn}(\mathbf{w}_k^{\mathrm{T}}\mathbf{x})\right)$$

2）平均感知器（Averaged Perceptron）：

$$\hat{y} = \mathrm{sgn}\left(\sum_{k=1}^{K} c_k(\mathbf{w}_k^{\mathrm{T}}\mathbf{x})\right) \tag{3.74}$$

$$= \mathrm{sgn}\left((\sum_{k=1}^{K} c_k\mathbf{w}_k)^{\mathrm{T}}\mathbf{x}\right) \tag{3.75}$$

$$= \mathrm{sgn}(\bar{\mathbf{w}}^{\mathrm{T}}\mathbf{x}), \tag{3.76}$$

其中 $\bar{\mathbf{w}}$ 为平均的权重向量。

假设 $\mathbf{w}_{t,n}$ 是在第 t 轮更新到第 n 个样本时权重向量的值，平均的权重向量 $\bar{\mathbf{w}}$ 也可以写为

$$\bar{\mathbf{w}} = \frac{\sum_{t=1}^{\mathrm{T}}\sum_{n=1}^{n}\mathbf{w}_{t,n}}{nT} \tag{3.77}$$

这个方法非常简单，只需要在算法3.1中增加一个 $\bar{\mathbf{w}}$，并且在处理每一个样本后，更新

$$\bar{\mathbf{w}} \leftarrow \bar{\mathbf{w}} + \mathbf{w}_{t,n}. \tag{3.78}$$

在处理每一个样本时都要更新$\bar{\mathrm{w}}$。因为$\bar{\mathrm{w}}$和$\mathrm{w}_{t,n}$都是稠密向量，因此更新操作比较费时。为了提高迭代速度，让这个更新只需要在错误预测发生时才进行更新。

算法 3.2: 平均感知器算法

输入: 训练集 $\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$，最大迭代次数 T

初始化：$\mathbf{w} \leftarrow 0$, $\mathbf{u} \leftarrow 0$, $c \leftarrow 0$;

for $t = 1 \cdots T$ **do**

  随机对训练样本进行随机排序;

  for $n = 1 \cdots N$ **do**

    选取一个样本 $(\mathbf{x}^{(n)}, y^{(n)})$;

    计算预测类别 $\hat{y}_t$;

    if $\hat{y}_t \neq y_t$ **then**

      $\mathbf{w} \leftarrow \mathbf{w} + y^{(n)}\mathbf{x}^{(n)}$;

      $\mathbf{u} \leftarrow \mathbf{u} + cy^{(n)}\mathbf{x}^{(n)}$;

    end

    $c \leftarrow c + 1$;

  end

end

$\bar{\mathbf{w}} = \mathbf{w}_T - \frac{1}{c}\mathbf{u}$;

输出: $\bar{\mathbf{w}}$

扩展到多类分类

引入一个构建输入输出联合空间上的特征函数 $\phi(x, y)$，将样本 (x, y) 对映射到一个特征向量空间。

$$\hat{\mathbf{y}} = \underset{\mathbf{y} \in \text{Gen}(\mathbf{x})}{\arg\max}\, \mathbf{w}^{\mathrm{T}} \phi(\mathbf{x}, \mathbf{y}),$$

算法 3.3: 广义感知器参数学习算法

输入: 训练集:$\{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^{N}$，最大迭代次数 T

初始化：$\mathbf{w}_0 \leftarrow 0$, $k \leftarrow 0$;

for $t = 1 \cdots T$ **do**

  随机对训练样本进行随机排序;

  for $n = 1 \cdots N$ **do**

    选取一个样本 $(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})$;

    用公式 (3.79) 计算预测类别 $\hat{\mathbf{y}}^{(n)}$;

    if $\hat{\mathbf{y}}^{(n)} \neq \mathbf{y}^{(n)}$ **then**

      $\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k + (\phi(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}) - \phi(\mathbf{x}^{(n)}, \hat{\mathbf{y}}^{(n)}))$;

      $k = k + 1$;

    end

  end

end

输出: $\mathbf{w}_k$

定义 3.3–广义线性可分: 对于训练集 $\mathcal{D} = \left\{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\right\}_{n=1}^{N}$，如果存在一个正的常数 $\gamma(\gamma > 0)$ 和权重向量 $\mathbf{w}^*$，并且 $\|\mathbf{w}^*\| = 1$，对所有 n 都满足 $\langle \mathbf{w}^*, \phi(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}) \rangle - \langle \mathbf{w}^*, \phi(\mathbf{x}^{(n)}, \mathbf{y}) \rangle \geq \gamma, \mathbf{y} \neq \mathbf{y}^{(n)}$

（$\phi(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}) \in \mathbb{R}^d$ 为样本 $\mathbf{x}^{(n)}, \mathbf{y}^{(n)}$ 的联合特征向量），那么训练集 $\mathcal{D}$ 在联合特征向量空间中是线性可分的。

定理 3.2 – 广义感知器收敛性： 对于满足广义线性可分的训练集 $\mathcal{D} = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^{N}$，假设 R 是所有样本中**真实标签和错误标签**在特征空间 $\phi(\mathbf{x}, \mathbf{y})$ 最远的距离。

$$R = \max_{n} \max_{\mathbf{z} \neq \mathbf{y}^{(n)}} \|\phi(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}) - \phi(\mathbf{x}^{(n)}, \mathbf{z})\|.$$

那么广义感知器参数学习算法3.3的权重更新次数不超过 $\frac{R^2}{\gamma^2}$。

给定一个两类分类器数据集 $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$，其中 $y_n \in \{+1, -1\}$，如果两类样本是线性可分的，即存在一个超平面

$$\mathbf{w}^{\mathrm{T}}\mathbf{x} + b = 0 \tag{3.82}$$

将两类样本分开，那么对于每个样本都有 $y^{(n)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + b) > 0$。

数据集 $\mathcal{D}$ 中每个样本 $\mathbf{x}^{(n)}$ 到分割超平面的距离为：

$$\gamma^{(n)} = \frac{\|\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + b\|}{\|\mathbf{w}\|} = \frac{y^{(n)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + b)}{\|\mathbf{w}\|}. \tag{3.83}$$

我们定义整个数据集 D 中所有样本到分割超平面的最短距离为间隔（Margin γ

$$\gamma = \min_{n} \gamma^{(n)}. \tag{3.84}$$

如果间隔 γ 越大，其分割超平面对两个数据集的划分越稳定，不容易受噪声等因素影响。支持向量机的目标是寻找一个超平面 $(\mathbf{w}^*, b^*)$ 使得 γ 最大，即

$$\begin{aligned} \max_{\mathbf{w}.b} \quad & \gamma \qquad (3.85) \\ \text{s.t.} \quad & \frac{y^{(n)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + b)}{\|\mathbf{w}\|} \geq \gamma, \forall n \end{aligned}$$

令 $\|\mathbf{w}\| \cdot \gamma = 1$，则公式 (3.85) 等价于

$$\begin{aligned} \max_{\mathbf{w},b} \quad & \frac{1}{\|\mathbf{w}\|^2} \qquad (3.86) \\ \text{s.t.} \quad & y^{(n)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + b) \geq 1, \forall n \end{aligned}$$

支持向量机

数据集中所有满足$y^{(n)}\left(w^T x^{(n)} + b\right) = 1$的样本点，都称为支持向量

（Support Vector）。对于一个线性可分的数据集，其分割超平面有很多个，但为了找到最大间隔分割超平面，将公式 (3.86) 的目标函数写为凸优化问题

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} \quad & 1 - y^{(n)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + b) \le 0, \qquad \forall n \end{aligned} \tag{3.87}$$

使用拉格朗日乘数法，公式 (3.87) 的拉格朗日函数为

$$\Lambda(\mathbf{w}, b, \lambda) = \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{n=1}^{N} \lambda_n \Big(1 - y^{(n)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + b)\Big), \tag{3.88}$$

其中 $\lambda_1 \ge 0, \cdots, \lambda_N \ge 0$ 为拉格朗日乘数。计算 $\Lambda(\mathbf{w}, b, \lambda)$ 关于 $\mathbf{w}$ 和 b 的导数，并令其等于 0 得到

$$\mathbf{w} = \sum_{n=1}^{N} \lambda_n y^{(n)} \mathbf{x}^{(n)}, \tag{3.89}$$

$$0 = \sum_{n=1}^{N} \lambda_n y^{(n)}. \tag{3.90}$$

将公式 (3.89) 代入公式 (3.88)，并利用公式 (3.90)，得到拉格朗日对偶函数

$$\Gamma(\lambda) = -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \lambda_m \lambda_n y^{(m)} y^{(n)} (\mathbf{x}^{(m)})^{\mathrm{T}} \mathbf{x}^{(n)} + \sum_{n=1}^{N} \lambda_n. \tag{3.91}$$

根据KKT条件中的互补松弛条件,最优解满足 $\lambda_n^*\big(1-y^{(n)}(\mathbf{w}^{*\mathrm{T}}\mathbf{x}^{(n)}+b^*)\big) = 0$。如果样本 $\mathbf{x}^{(n)}$ 不在约束边界上，$\lambda_n^* = 0$，其约束失效；如果样本 $\mathbf{x}^{(n)}$ 在约束边界上，$\lambda_n^* \ge 0$。这些在约束边界上样本点称为**支持向量**（support vector），即离决策平面距离最近的点。

再计算出 λ^* 后，根据公式 (3.89) 计算出最优权重 $\mathbf{w}^*$，最优偏置 b^* 可以通过任选一个支持向量 $(\tilde{\mathbf{x}}, \tilde{y})$ 计算得到。

$$b^* = \tilde{y} - \mathbf{w}^{*\mathrm{T}}\tilde{\mathbf{x}}. \tag{3.92}$$

最优参数的支持向量机的决策函数为

$$\begin{aligned} f(\mathbf{x}) &= \operatorname{sgn}(\mathbf{w}^{*\mathrm{T}}\mathbf{x} + b^*) & (3.93) \\ &= \operatorname{sgn}\left(\sum_{n=1}^{N} \lambda_n^* y^{(n)} (\mathbf{x}^{(n)})^{\mathrm{T}}\mathbf{x} + b^*\right). & (3.94) \end{aligned}$$

支持向量机的决策函数只依赖 $\lambda_n^* > 0$ 的样本点，即支持向量。
是间隔最大的超平面是唯一的。

支持向量机的目标函数可以通过 SMO 等优化方法得到全局最优解，因此比其它分类器的学习效率更高。此外，支持向量机的决策函数只依赖与支持向量，与训练样本总数无关，分类速度比较快。

可以使用核函数（kernel function）隐式地将样本从原始特征空间映射到更高维的空间，并解决原始特征空间中的线性不可分问题。

软间隔

能够容忍部分不满足约束的样本，我们可以引入松弛变量 ξ，变为

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n \\ \text{s.t.} \quad & 1 - y^{(n)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + b) - \xi_n \le 0, \qquad \forall n \\ & \xi_n \ge 0, \qquad \forall n \end{aligned} \tag{3.98}$$

其中参数 $C > 0$ 用来控制间隔和松弛变量惩罚的平衡。引入松弛变量的间隔称为软间隔（soft margin）。公式 (3.98) 也可以表示为经验风险 + 正则化项的形式。

$$\min_{\mathbf{w},b} \quad \sum_{n=1}^{N}\max\left(0, 1 - y^{(n)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + b)\right) + \frac{1}{C}\cdot\frac{1}{2}\|\mathbf{w}\|^2, \tag{3.99}$$

其中 $\max\left(0, 1 - y^{(n)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + b)\right)$ 称为 *hinge* 损失函数，$\frac{1}{C}$ 可以看作是正则化系数。

软间隔支持向量机的参数学习和原始支持向量机类似，其最终决策函数也只和支持向量有关，即满足 $1 - y^{(n)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + b) - \xi_n = 0$ 的样本。

损失函数对比

对于两类分类来说，当$yf(x,w) > 0$时，分类器预测正确，并且$yf(x,w)$越大，模型的预测越正确；当$yf(x,w) < 0$时，分类器预测错误，并且$yf(x,w)$越小，模型的预测越错误。因此，一个好的损失函数应该随着$yf(x,w)$的增大而减少。除了平方损失，其它损失函数都比较适合于两类分类问题。

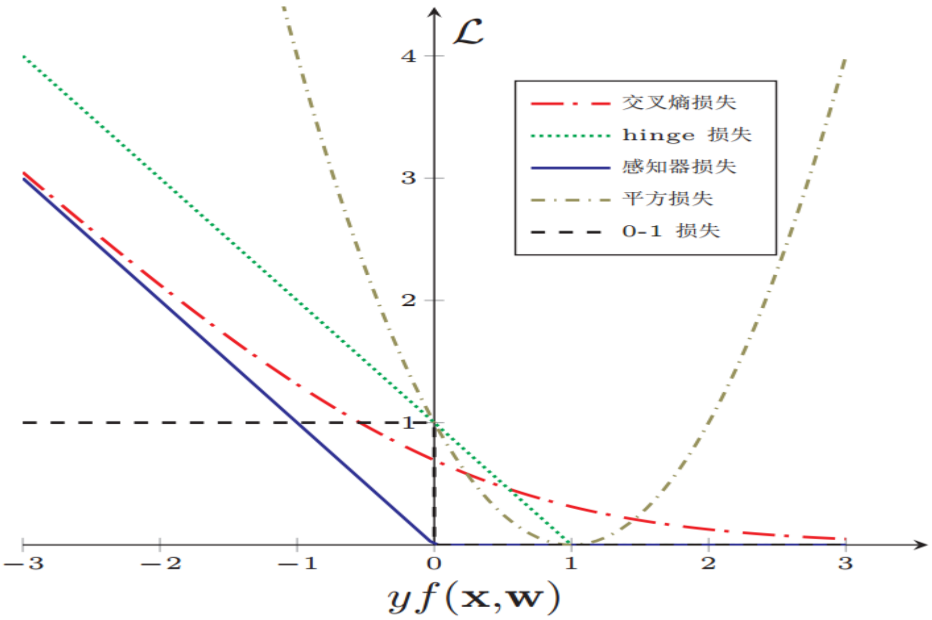

图 3.7 不同损失函数的对比

	激活函数	损失函数	优化方法
线性回归	-	$(y - \mathbf{w}^{\mathrm{T}}\mathbf{x})^2$	最小二乘、梯度下降
Logistic 回归	$\sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x})$	$\mathbf{y} \log \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x})$	梯度下降
Softmax 回归	$\mathrm{softmax}(W^{\mathrm{T}}\mathbf{x})$	$\mathbf{y} \log \mathrm{softmax}(W^{\mathrm{T}}\mathbf{x})$	梯度下降
感知器	$\mathrm{sgn}(\mathbf{w}^{\mathrm{T}}\mathbf{x})$	$\max(0, -y\mathbf{w}^{\mathrm{T}}\mathbf{x})$	随机梯度下降
支持向量机	$\mathrm{sgn}(\mathbf{w}^{\mathrm{T}}\mathbf{x})$	$\max(0, 1 - y\mathbf{w}^{\mathrm{T}}\mathbf{x})$	二次规划、SMO 等

表 3.1 几种不同的线性模型对比