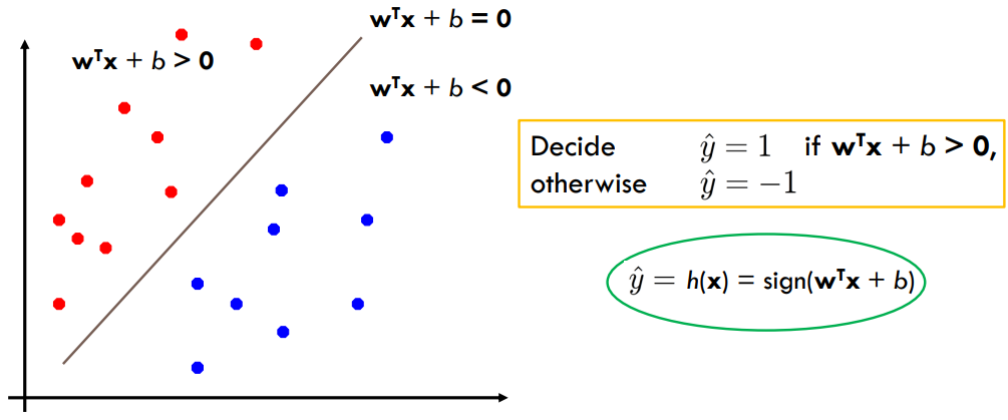
Linear predictions 线性预测

分类

从具有有限离散标签的数据中学习。机器学习中的主导问题

线性分类器

二元分类可以看作是在特征空间中分离类的任务(特征空间)



感知机

感知机是一种线性分类模型。它的决策边界是一个超平面，将特征空间分成了两个部分，分别对应于输出为 +1 和 -1 的两个类别。

感知机的线性分类模型可以表示为：

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

其中w和b为感知机模型参数，$\boldsymbol{w} \in \mathbf{R}^n$ 是权重或者权值向量，$\boldsymbol{x}$ 是输入的特征向量，$b \in \mathbf{R}$ 是偏置(bias)，$w \cdot x$ 表示w和x的内积，$\operatorname{sign}(\cdot)$ 是符号函数。

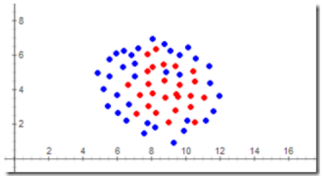
$$\operatorname{sign}(x)=\begin{cases}+1, & x \geq 0 \\ -1, & x<0\end{cases}$$

感知机的训练目标是找到一个能够将训练集中的正负样本完全分开的超平面，即找到一个合适的权重向量 $\boldsymbol{w}$ 和偏置项 b，使得对于任意训练样本 $(\boldsymbol{x}_i, y_i)$，都有
$y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) > 0$

> **这个式子表达了感知机模型的训练目标**，即让感知机能够对训练集中的正负样本进行完全分开，使得每个样本都被正确分类。
>
> **对于一个训练样本** $(\boldsymbol{x}_i, y_i)$，y_i 是其真实的标签，$\boldsymbol{x}_i$是其特征向量，$\boldsymbol{w}$ 是感知机的权重向量，b 是偏置项。根据感知机模型的定义，当 $\boldsymbol{w} \cdot \boldsymbol{x}_i + b > 0$ 时，模型的输出为 +1，否则输出为 -1。
>
> 因此，当 $y_i = +1$ 时，要求 $\boldsymbol{w} \cdot \boldsymbol{x}_i + b > 0$，即要求模型将正样本正确分类为 +1。同理，当 $y_i = -1$ 时，要求 $\boldsymbol{w} \cdot \boldsymbol{x}_i + b < 0$，即要求模型将负样本正确分类为 -1。
>
> **综合以上两种情况**，我们可以将训练目标表达为 $y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) > 0$，即要求每个训练样本都被正确分类，即对于任意训练样本 $(\boldsymbol{x}_i, y_i)$，都有 $y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) > 0$。如果存在任意一个训练样本不能被正确分类，那么就需要不断地更新权重向量和偏置项，直到训练集中的所有样本都能被正确分类。

感知机学习策略

1. 数据集线性可分性
2. 在二维平面中，可以用一条直线将+1类和-1类完美分开，那么这个样本空间就是线性可分的。下图中的样本就是线性不可分的，感知机就不能处理这种情况。因此，感知机都基于一个前提：问题空间线性可分



损失函数的选取

1. 损失函数的一个自然选择就是误分类点的总数，要找到决策边界，即分类超平面，使得分类误差最小，可以通过最小化期望的 0/1 损失函数来实现。对于一个样本 (x, y)，0/1 损失函数定义为：

$$L(h(\mathbf{x}), y)=\begin{cases}0, & h(\mathbf{x})=y \\ 1, & h(\mathbf{x}) \neq y\end{cases}$$

   其中，$h(\mathbf{x})$ 是模型预测的输出，y 是样本的真实标签。但是这样的点不是参数w，b的连续可导函数，不易优化

2. 损失函数的另一个选择就是误分类点到划分超平面$S(\mathbf{w} \cdot \mathbf{x} + b = 0)$的总距离

假设数据集 $T=\{(x_1,y_1),(x_2,y_2)...(x_n,y_n)\}$ 中所有的 $y_i=+1$ 的实例i，有 $w \bullet x+b>0$ ； 对 $y_i=-1$ 的实例有$w \bullet x+b<0$

这里 先给出输入空间 R^n 中任意一点x_0 到超平面S的距离：

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

这里 $\frac{1}{\|w\|}$ 是W的 l_2 范数。所以，对于误分类数据 (x_i, y_i) 有

$$-y_i(w \bullet x_i+b)>0$$

因为对x_i错分了，所以若y_i为-1，则计算的$(w.x_i+b)>0$，反之若y_i为+1，则计算的$(w.x_i+b)<0$

距离的计算

点 x_0 到超平面 $S: w^T x + b = 0$ 的距离d(注: x_0, w, x全为N维向量)可以通过以下步骤计算:

1. 设点 x_0 在平面 S 上的投影为 x_1，则有 $w^T x_1 + b = 0$
2. -

由于向量$\overrightarrow{x_0x_1}$与S平面的法向量w平行，所以（乘积的模=模的乘积）

$$|w \bullet \overrightarrow{x_0x_1}| = |w||\overrightarrow{x_0x_1}| = \sqrt{(w^1)^2 + \ldots + (w^N)^2}\, d = \|w\| d$$

L_2范数

$$\begin{aligned} \text{又} \quad w \bullet \overrightarrow{x_0x_1} &= w^1(x^1{}_0 - x_1^1) + w^2(x^2{}_0 - x_1^2) + \ldots + w^N(x^N{}_0 - x_1^N) \\ &= w^1x^1{}_0 + w^2x^2{}_0 + \ldots + w^Nx^N{}_0 - (w^1x^1{}_1 + w^2x^2{}_1 + \ldots + w^Nx^N{}_1) \\ &= w^1x^1{}_0 + w^2x^2{}_0 + \ldots + w^Nx^N{}_0 - (-b) \end{aligned}$$

x_1在平面S上，所以 $\mathbf{w}.x_1+b=0$

$$\text{所以} \quad \|w\| d = |w^1x^1{}_0 + w^2x^2{}_0 + \ldots + w^Nx^N{}_0 + b| = |w \bullet x_0 + b|$$

$$\text{即} \quad d = \frac{1}{\|w\|}|w \bullet x_0 + b|$$

因此误分类点(x_i, y_i)到超平面S的距离可以写作：

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

假设误分类点的集合为M，那么所有误分类点到超平面S的总距离为：

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

这里的||w||值对求极值没有影响，不必考虑，这样就得到了感知机学习的损失函数：在误分类时是参数w，b的线性函数。也就是说，为求得正确的参数w，b，我们的目标函数为

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

而它是连续可导的，这就使得我们比较容易求得其最小值

$$\begin{aligned} L_2(h(\mathbf{x}), y) &= (y - \mathbf{w}^\top \mathbf{x} - b)^2 = (1 - y(\mathbf{w}^\top \mathbf{x} + b))^2 \\ L_1(h(\mathbf{x}), y) &= |y - \mathbf{w}^\top \mathbf{x} - b| = |1 - y(\mathbf{w}^\top \mathbf{x} + b)| \\ L_{hinge}(h(\mathbf{x}), y) &= \big(1 - y(\mathbf{w}^\top \mathbf{x} + b)\big)_+ \end{aligned}$$

> **Lhinge**（也称为 hinge 损失函数）是一种常用的分类器损失函数，通常用于支持向量机（SVM）和感知机等模型 中。它的形式如下：
>
> $L_{hinge}(w, b; x, y) = \max\{0, 1 - y(w^T x + b)\}$
>
> **其中，w 和 b 是模型的参数**，x 是输入样本，y 是样本的标签，取值为 $+1$ 或 -1，表示正类或负类。$h(x) = \text{sign}(w^T x + b)$ 表示模型对样本 x 的预测输出。
> **当模型对样本 x 的预测结果正确时**（即 $y(h(x)) > 0$），损失函数为 0；当模型对样本 x 的预测结果错误时（即 $y(h(x)) \leq 0$），损失函数为 $1 - y(h(x))(w^T x + b)$。
> **这个损失函数的含义是**，对于误分类的样本，我们希望让其距离超平面至少为 1，即 $y(w^T x + b) \geq 1$，如果距离小于 1，则损失函数为 $1 - y(w^T x + b)$，表示分类错误的程度；如果距离大于等于 1，则损失函数为 0，表示分类正确。
> **Lhinge 损失函数是一个凸函数，但不是连续可导的**，因为在 $y(w^T x + b) = 1$ 时，存在一个“拐点”，导致其导数不连续。因此，在使用梯度下降等优化算法时，需要使用次梯度

> （subgradient）来代替导数，使得算法可以收敛。同时，Lhinge
> 损失函数也可以通过一些优化算法（如 SMO 算法）来求解支持向量机模型的最优解。

最小二乘法分类

最小二乘损失函数 $L_2(h(\mathbf{x}), y) = (y - \mathbf{w}^\top \mathbf{x} - b)^2$

目标：学习分类器$h(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x} + b)$以最小化最小二乘损失

$$\begin{aligned} Loss &= \min_{\mathbf{W},b} \sum_i L_2(h(\mathbf{x}_i), y_i) \\ &= \min_{\mathbf{W},b} \sum_i (y_i - \mathbf{w}^\top \mathbf{x}_i - b)^2 \end{aligned}$$

求解最小二乘分类

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1d} \\ \vdots & & \vdots & \\ 1 & x_{N1} & \cdots & x_{Nd} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} b \\ \vdots \\ w_d \end{bmatrix}$$

$$\begin{aligned} Loss &= \min_{\mathbf{w}} \sum_i (\mathbf{y} - X\mathbf{w})_i^2 \\ &= \min_{\mathbf{w}} (X\mathbf{w} - \mathbf{y})^\top (X\mathbf{w} - \mathbf{y}) \end{aligned}$$

$$\begin{aligned} \frac{\partial Loss}{\partial \mathbf{w}} = 2(X\mathbf{w} - \mathbf{y})^\top X &= 0 \\ X^\top X\mathbf{w} - X^\top \mathbf{y} &= 0 \\ \mathbf{w}^* &= (X^\top X)^{-1} X^\top \mathbf{y} \end{aligned}$$

Note: $d(\mathbf{Ax+b})^T\mathbf{C}(\mathbf{Dx+e}) = ((\mathbf{Ax+b})^T\mathbf{CD} + (\mathbf{Dx+e})^T\mathbf{C}^T\mathbf{A})\, d\mathbf{x}$

$d(\mathbf{Ax+b})^T(\mathbf{Ax+b}) = (2(\mathbf{Ax+b})^T\mathbf{A})\, d\mathbf{x}$

- $X^+ = (X^\top X)^{-1} X^\top$ is called the Moore-Penrose pseudoinverse（伪逆） of X

矩阵解法

线性回归定义为：

$h_\theta(x_1, x_2, \ldots x_{n-1}) = \theta_0 + \theta_1 x_1 + \ldots + \theta_{n-1} x_{n-1}$

其中 θ 为参数。假设现在有 m 个样本，每个样本有 $n-1$ 维特征，将所有样本点代入模型中得：

$$\begin{aligned} h_1 &= \theta_0 + \theta_1 x_{1,1} + \theta_2 x_{1,2} + \ldots + \theta_{n-1} x_{1,n-1} \\ h_2 &= \theta_0 + \theta_1 x_{2,1} + \theta_2 x_{2,2} + \ldots + \theta_{n-1} x_{2,n-1} \\ &\vdots \\ h_m &= \theta_0 + \theta_1 x_{m,1} + \theta_2 x_{m,2} + \ldots + \theta_{n-1} x_{m,n-1} \end{aligned}$$

为方便用矩阵表示，我们令 $x_0 = 1$，于是上述方程可以用矩阵表示为：$\mathbf{h} = \mathbf{X}\theta$

其中，$\mathbf{h}$ 为 $m \times 1$ 的向量，代表模型的理论值，θ 为 $n \times 1$ 的向量，$\mathbf{X}$ 为 $m \times n$ 维的矩阵，m 代表样本的个数，n 代表样本的特征数。于是目标损失函数用矩阵表示为：

$J(\theta) = \|\mathbf{h} - \mathbf{Y}\|^2 = \|\mathbf{X}\theta - \mathbf{Y}\|^2 = (\mathbf{X}\theta - \mathbf{Y})^T(\mathbf{X}\theta - \mathbf{Y})$

其中 $\mathbf{Y}$ 是样本的输出向量，维度为 $m \times 1$。

根据高数知识我们知道函数取得极值就是导数为0的地方，所以我们只需要对损失函数求导令其等于0就可以解出 θ。矩阵求导属于矩阵微积分的内容，我们先介绍两个用到的公式：

$\frac{\partial x^T a}{\partial x} = \frac{\partial a^T x}{\partial x} = a$

$\frac{\partial x^T Ax}{\partial x} = Ax + A^T x$

如果矩阵 A 是对称的：

$Ax + A^T x = 2Ax$

对目标函数化简：

$J(\theta) = \theta^T X^T X\theta - \theta^T X^T Y - Y^T X\theta + Y^T Y$

求导令其等于0：

$\frac{\partial}{\partial\theta} J(\theta) = 2X^T X\theta - 2X^T Y = 0$

解得：

$\theta = \left(X^T X\right)^{-1} X^T Y$

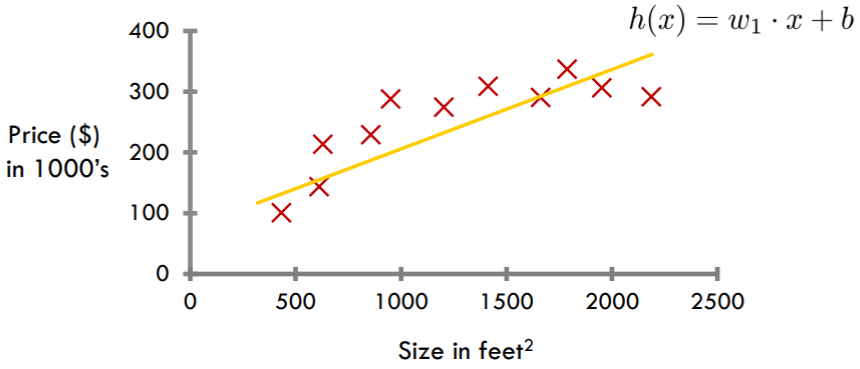
经过推导我们得到了 θ 的解析解，现在只要给了数据，我们就可以带入解析解中直接算出 θ。

一般线性分类

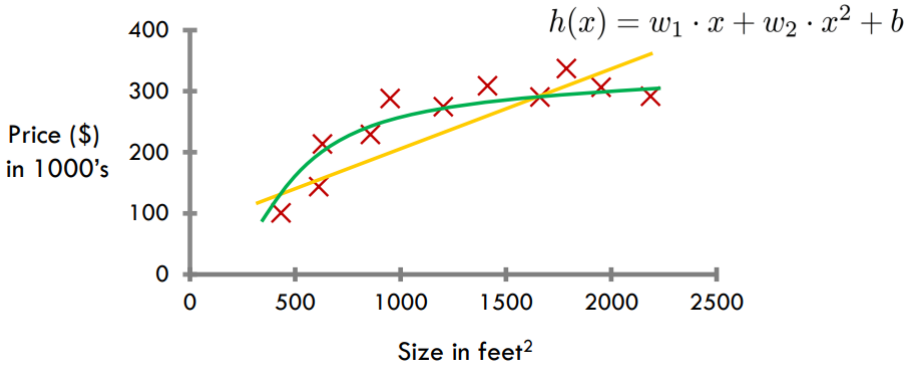
基函数（非线性）

$$f(\mathbf{x}, \mathbf{w}) = b + w_1\phi_1(\mathbf{x}) + w_2\phi_2(\mathbf{x}) + \cdots + w_m\phi_m(\mathbf{x})$$

其中ϕ为基函数，用于拟合非线性

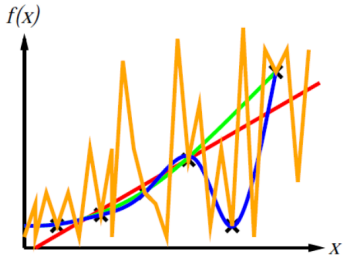


$$\begin{aligned} Loss &= \min_{\mathbf{w},b} \sum_i L_2(h(x_i), y_i) \quad \text{Least Squares} \\ &= \min_{\mathbf{w},b} \sum_i (y_i - w_1 \cdot x_i - b)^2 \end{aligned}$$



$$\begin{aligned} Loss &= \min_{\mathbf{w},b} \sum_i L_2(h(x_i), y_i) \quad \text{Least Squares} \\ &= \min_{\mathbf{w},b} \sum_i (y_i - w_1 \cdot x_i - w_2 \cdot x_i^2 - b)^2 \end{aligned}$$

模型复杂性和过度拟合



Ockham's razor（奥卡姆剃刀原则）: maximize a combination of consistency and simplicity
优先选择与数据一致的最简单的假设

训练误差

训练误差（Training errors），也称为表观误差（Apparent errors），是指在训练数据集上用模型进行训练时所产生的误差。训练误差是模型在学习过程中不可避免的产物，通常我们希望训练误差越小越好，因为模型对训练数据的拟合程度越高，就越可能对未知数据产生良好的预测能力。但是，如果模型过于拟合训练数据，训练误差可能会变得非常小，但是这种模型可能会过度拟合，对于未知数据的预测性能将变得很差。

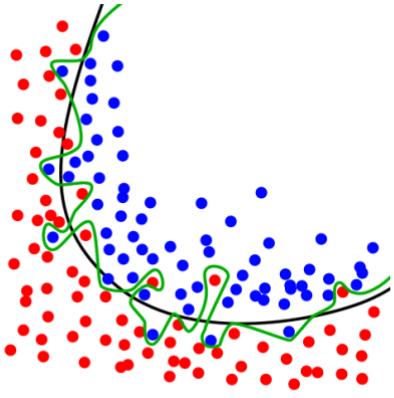
测试误差

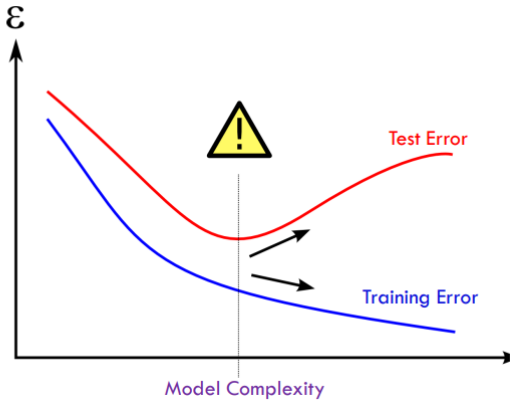
测试误差（Test errors）是指在测试数据集上用模型进行测试时所产生的误差。测试误差用于评估模型的泛化能力，即模型是否具有对未知数据的预测能力。测试误差与训练误差之间的差异通常被称为泛化误差。

泛化误差

泛化误差（Generalization errors）是指模型在未知数据上的期望误差，即模型对于新数据的预测能力。泛化误差是机器学习中最为重要的概念之一，因为机器学习的目标是构建泛化能力强的模型，而不是仅仅在训练数据上表现良好的模型。泛化误差通常用测试误差来近似估计，但是测试误差只能提供模型的一种局部估计，因此为了更准确地估计模型的泛化误差，需要使用交叉验证等方法。

复杂度与过拟合





欠拟合：当模型过于简单时，训练和测试误差都很大
过拟合：当建模过于复杂时，训练误差很小，但测试误差很大

给定两个具有相似泛化误差的模型，人们应该更喜欢更简单的模型而不是更复杂的模型
复杂的模型更有可能因数据错误而意外拟合
因此，在评估模型时应包括模型复杂性

规范化

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \; Loss + \lambda \cdot penalty(\mathbf{w})$$

L2 regularization $$\mathbf{w}^* = \arg\min_{\mathbf{w}} \; Loss + \lambda \|\mathbf{w}\|^2$$

L1 regularization $$\mathbf{w}^* = \arg\min_{\mathbf{w}} \; Loss + \lambda |\mathbf{w}|$$

Regularization parameter

- Solving L2-regularized LS

$$\min_{\mathbf{w}} (X\mathbf{w} - \mathbf{y})^2 + \lambda \|\mathbf{w}\|^2$$

Solution?

λ是惩罚权重

取w的L2范数作为惩罚项，让系数的取值变得平均。对于关联特征，能够获得更相近的对应系数

L1范数，系数w的L1范数作为惩罚项加到损失函数上，由于非零，迫使那些弱的特征所对应的系数变成0，达到特征选择

当λ 足够大，相当于：

$\min_{\mathbf{w}} \; Loss$ subject to $\sum_i |w_i|^q \leq \eta$

- L1范数正则化会使得解更加稀疏。在损失函数中加入L1范数惩罚项会使得模型尽可能地将参数稀疏化，即让一些参数变为0，从而剔除那些对模型影响较小的特征，提高模型的泛化性能。因此，L1范数正则化通常用于特征选择和稀疏性建模。
- L2范数正则化会使得解更加平滑。在损失函数中加入L2范数惩罚项会使得模型的参数尽可能小，从而降低模型的复杂度，防止过拟合。L2范数正则化通常用于提高模型的泛化性能和防止过拟合。

需要注意的是，当L1范数惩罚项足够大时，模型的参数会变为0，从而得到一个稀疏解。而L2范数惩罚项则不会将参数完全变为0，而是将参数约束在一个较小的范围内，从而获得一个平滑的解。

不同q值的正则化项的等高线