机器学习

第3章 回归

朱桂祥 (9120201070@nufe.edu.cn)
南京财经大学信息工程学院
江苏省电子商务重点实验室
电子商务信息处理国家级国际联合研究中心
电子商务交易技术国家地方联合工程实验室

1. 线性回归

回归的概念

监督学习分为回归和分类

- ✓ 回归（Regression、Prediction）
  - ✓ 如何预测南京仙林大学城的房价?
  - ✓ 未来的股票市场走向?
- ✓ 分类（Classification）
  - ✓ 身高1.85m，体重100kg的男人穿什么尺码的T恤?
  - ✓ 根据肿瘤的体积、患者的年龄来判断良性或恶性?

标签连续

标签离散

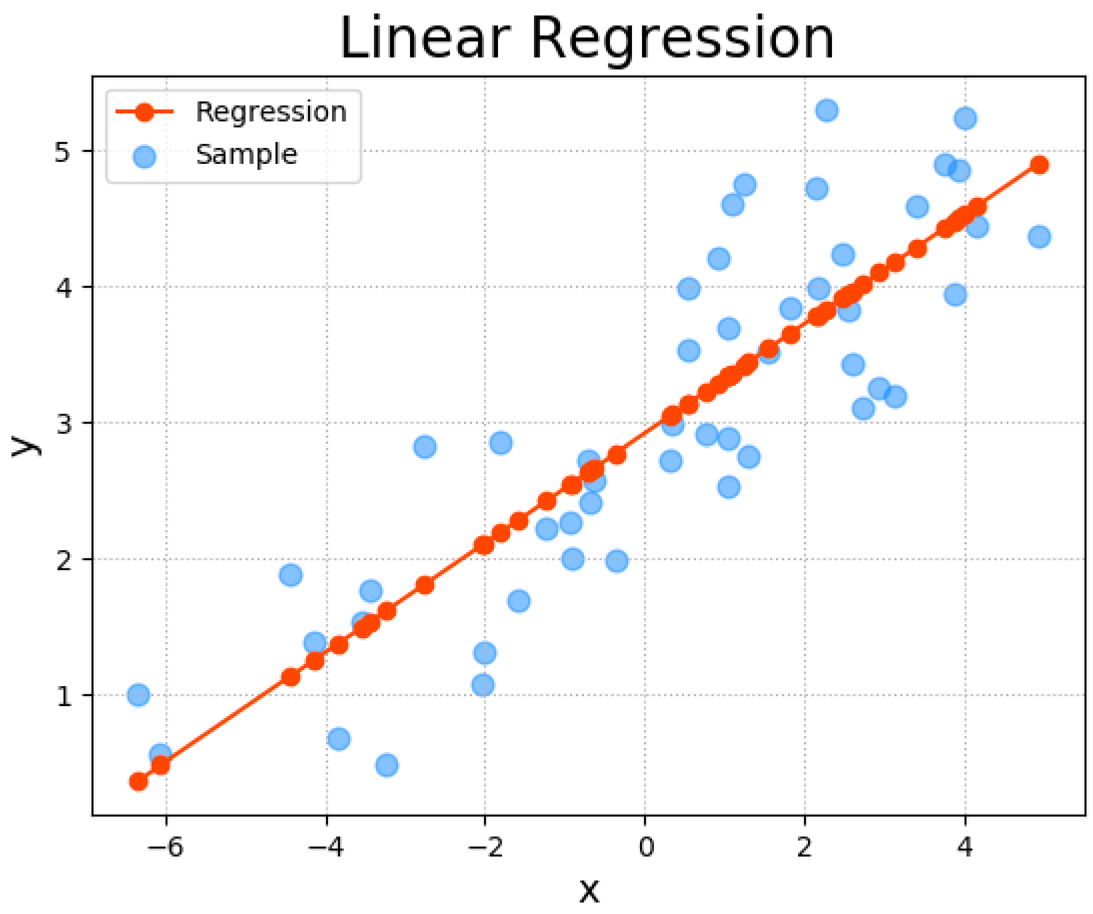
线性回归

- ✓ 回归问题：研究输入变量与输出变量之间的关系。
  - ✓ 因变量 $\xrightarrow{?}$ 自变量
  - ✓ 房价预测
- ✓ 回归模型：表示从输入变量到输出变量之间映射的函数。
  - ✓ 线性
  - ✓ 非线性

线性回归

线性回归（Linear Regression）

是一种通过属性的线性组合来进行预测的**线性模型**，其目的是找到一条直线或者一个平面或者更高维的超平面，**使得预测值与真实值之间的误差最小化。**



线性回归-符号约定

- ✓ 输入变量/特征：$\boldsymbol{x} \in \mathbb{R}^d$

  d：特征维度/特征数量

- ✓ 输出变量/标记：$y \in \mathbb{R}$

  m：样本数量

- ✓ 训练集：$D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$
- ✓ 模型/假设：$f: \mathbb{R}^d \rightarrow \mathbb{R}$

特征 → ? → 标签

色泽	根蒂	敲声	好瓜
青绿	蜷缩	浊响	是
乌黑	蜷缩	沉闷	是
青绿	硬挺	清脆	否
乌黑	稍蜷	沉闷	否

线性回归-线性模型

✓ 基本形式：通过属性的线性组合进行预测的函数

$$f(\boldsymbol{x}) = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + b$$

其中$\boldsymbol{x} = (x_1; x_2; \cdots; x_d)$

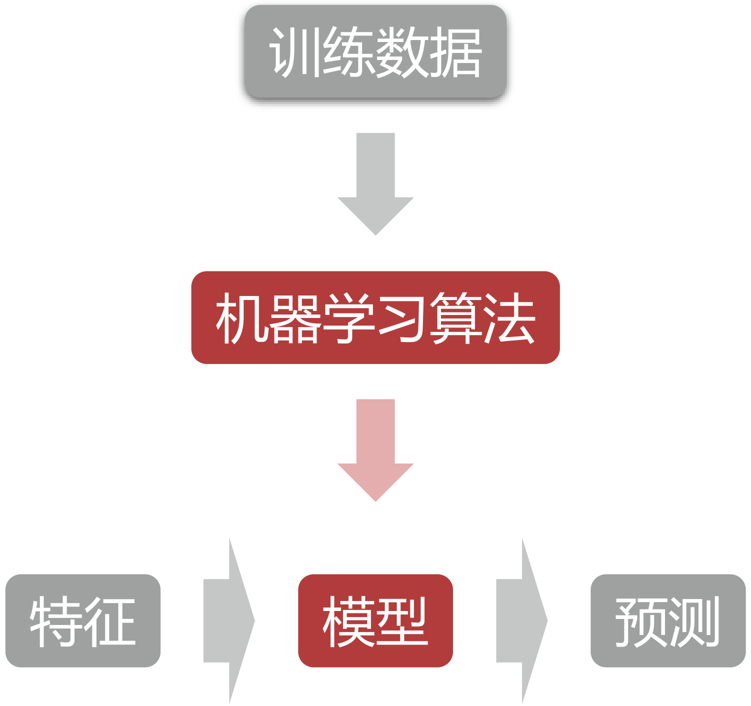
✓ 向量形式

偏置

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b$$

未知参数

其中$\boldsymbol{w} = (w_1; w_2; \cdots; w_d)$



线性回归-线性模型

- ✓ 优点:
  - ✓ 形式简单、易于建模
  - ✓ 可解释性
- ✓ 一个例子
  - ✓ 综合考虑色泽、根蒂和敲声来判断西瓜好不好
  - ✓ 其中根蒂的系数最大，表明根蒂最要紧；而敲声的系数比色泽大，说明敲声比色泽更重要

$$f_{\text{好瓜}}(\boldsymbol{x}) = 0.2 \cdot x_{\text{色泽}} + 0.5 \cdot x_{\text{根蒂}} + 0.3 \cdot x_{\text{敲声}} + 1$$

线性回归-目标

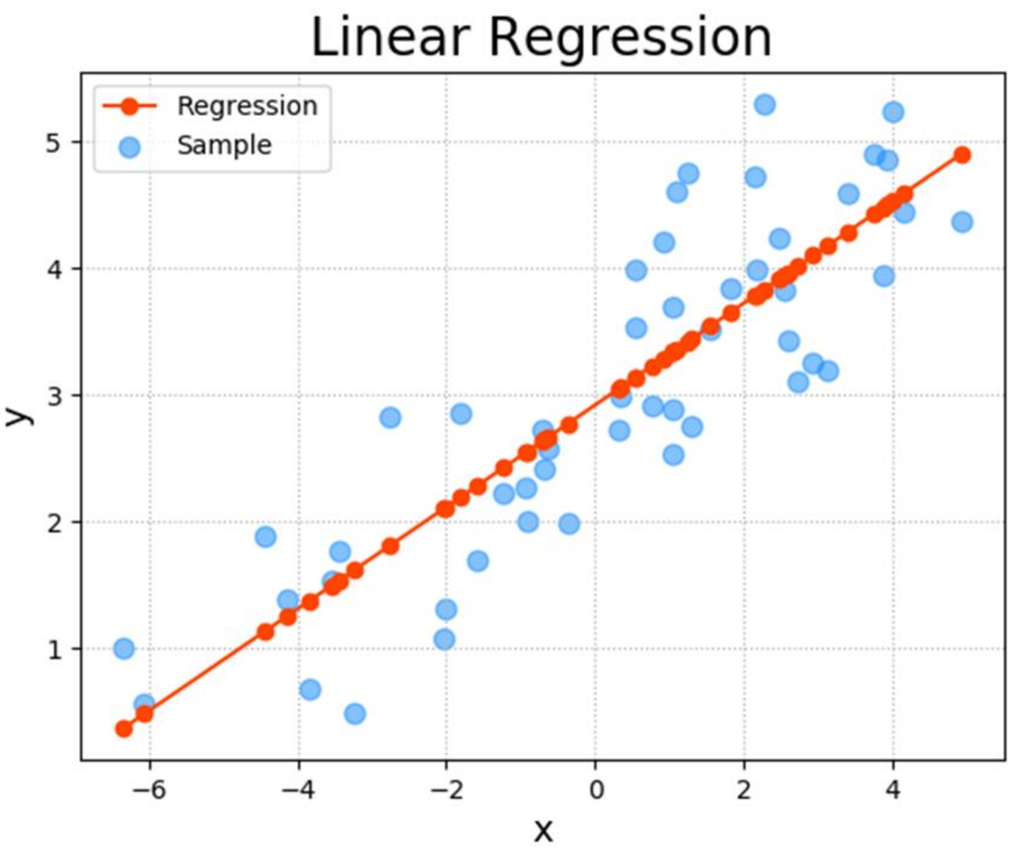
- ✓ 尽可能准确预测 $f(\boldsymbol{x}_i) \cong y_i$
  - ✓ 在训练集上预测值与真实值之间的**误差最小化**
  - ✓ 均方误差（mean square error, MSE）最小化

$$\underbrace{\min_{\boldsymbol{w},b} J(\boldsymbol{w},b) = \frac{1}{m}\sum_{i=1}^{m}\underbrace{\left(f(\boldsymbol{x}_i) - y_i\right)^2}_{\text{损失函数}}}_{\text{代价函数}}$$

损失函数：度量单个样本预测的误差。

代价函数：度量全部样本预测的平均误差。

线性回归-损失函数



损失函数(Loss Function)度量单样本预测的错误程度，损失函数值越小，模型就越好。常用的损失函数包括：0-1损失函数、平方损失函数、绝对损失函数、对数损失函数等。

代价函数(Cost Function)度量全部样本集的平均误差。常用的代价函数包括均方误差、均方根误差、平均绝对误差等。

目标函数(Object Function)代价函数和正则化函数，最终要优化的函数。

目录

最小二乘法

- ✓ 一元线性回归

$$\min_{w,b} J(w,b) = \frac{1}{m}\sum_{i=1}^{m}(y_i - wx_i - b)^2$$

- ✓ 参数/模型求解
  - ✓ 最小二乘估计
  - ✓ 若函数f在w_0处可导，且w_0是函数的极值点，则导数$f'(w_0) = 0$

最小二乘法

- ✓ 分别对w和b求导，可得

$$\frac{\partial J(w,b)}{\partial w}=\frac{2}{m}\left(w\sum_{i=1}^{m}x_i^2-\sum_{i=1}^{m}(y_i-b)\,x_i\right)$$

$$\frac{\partial J(w,b)}{\partial b}=\frac{2}{m}\left(mb-\sum_{i=1}^{m}(y_i-wx_i)\right)$$

求导法则

$$(x^{\mu})'=\mu x^{\mu-1}$$

链式法则

$$\frac{dy}{dx}=\frac{dy}{du}\cdot\frac{du}{dx}$$

最小二乘法

- ✓ 令导数为0，得到闭式解（closed-form solution）

$$w = \frac{\sum_{i=1}^{m} y_i (x_i - \bar{x})}{\sum_{i=1}^{m} x_i^2 - \frac{1}{m}\left(\sum_{i=1}^{m} x_i\right)^2}$$

$$b = \frac{1}{m}\sum_{i=1}^{m}(y_i - wx_i)$$

其中

$$\bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i$$

最小二乘法

- ✓ 多元线性回归

$$\min_{\boldsymbol{w},b} J(\boldsymbol{w},b)=\sum_{i=1}^{m}(y_i-\boldsymbol{w}^T\boldsymbol{x}_i-b)^2$$

- ✓ 为了方便优化，我们将目标重写为

$$\min_{\widehat{\boldsymbol{w}}} J(\widehat{\boldsymbol{w}})=(\boldsymbol{y}-\boldsymbol{X}\widehat{\boldsymbol{w}})^T(\boldsymbol{y}-\boldsymbol{X}\widehat{\boldsymbol{w}})$$

最小二乘法

- ✓ 参数向量

$$\hat{\boldsymbol{w}} = (\boldsymbol{w}; b)$$

- ✓ 特征矩阵

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1^T & 1 \\ \vdots & \vdots \\ \boldsymbol{x}_m^T & 1 \end{pmatrix}$$

- ✓ 标签向量

$$\boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}$$

最小二乘法

- ✓ 对$\hat{\boldsymbol{w}}$求导，可得

$$\frac{\partial J(\hat{\boldsymbol{w}})}{\partial \hat{\boldsymbol{w}}} = 2\boldsymbol{X}^T(\boldsymbol{X}\hat{\boldsymbol{w}} - \boldsymbol{y})$$

- ✓ 令导数为0，若矩阵$\boldsymbol{X}^T\boldsymbol{X}$**可逆**，得到闭式解

$$\hat{\boldsymbol{w}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

矩阵求导法则

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

$$\frac{\partial \boldsymbol{x}^T A \boldsymbol{x}}{\partial \boldsymbol{x}} = (A + A^T)\boldsymbol{x}$$

线性回归-预测

✓ 对于新样本$\mathbf{x}_{m+1}$，记$\hat{\mathbf{x}}_{m+1} = (\boldsymbol{x}_{m+1}^T, 1)$

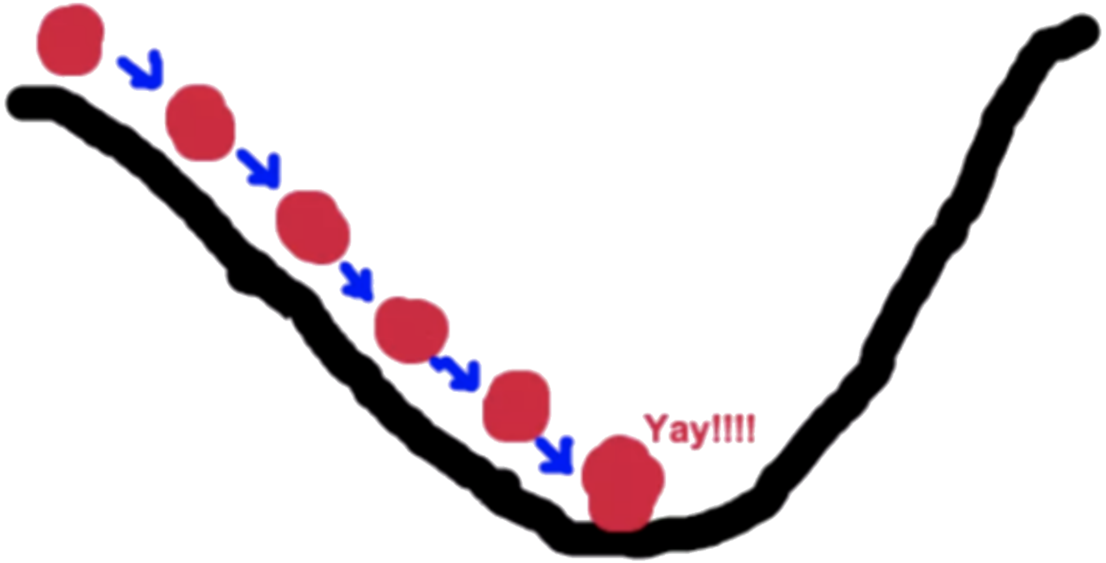
$$f(\hat{\mathbf{x}}_{m+1}) = \hat{\mathbf{x}}_{m+1}\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

矩阵$\boldsymbol{X}^T\boldsymbol{X}$可逆

逆矩阵计算复杂度$O(d^3)$

目录

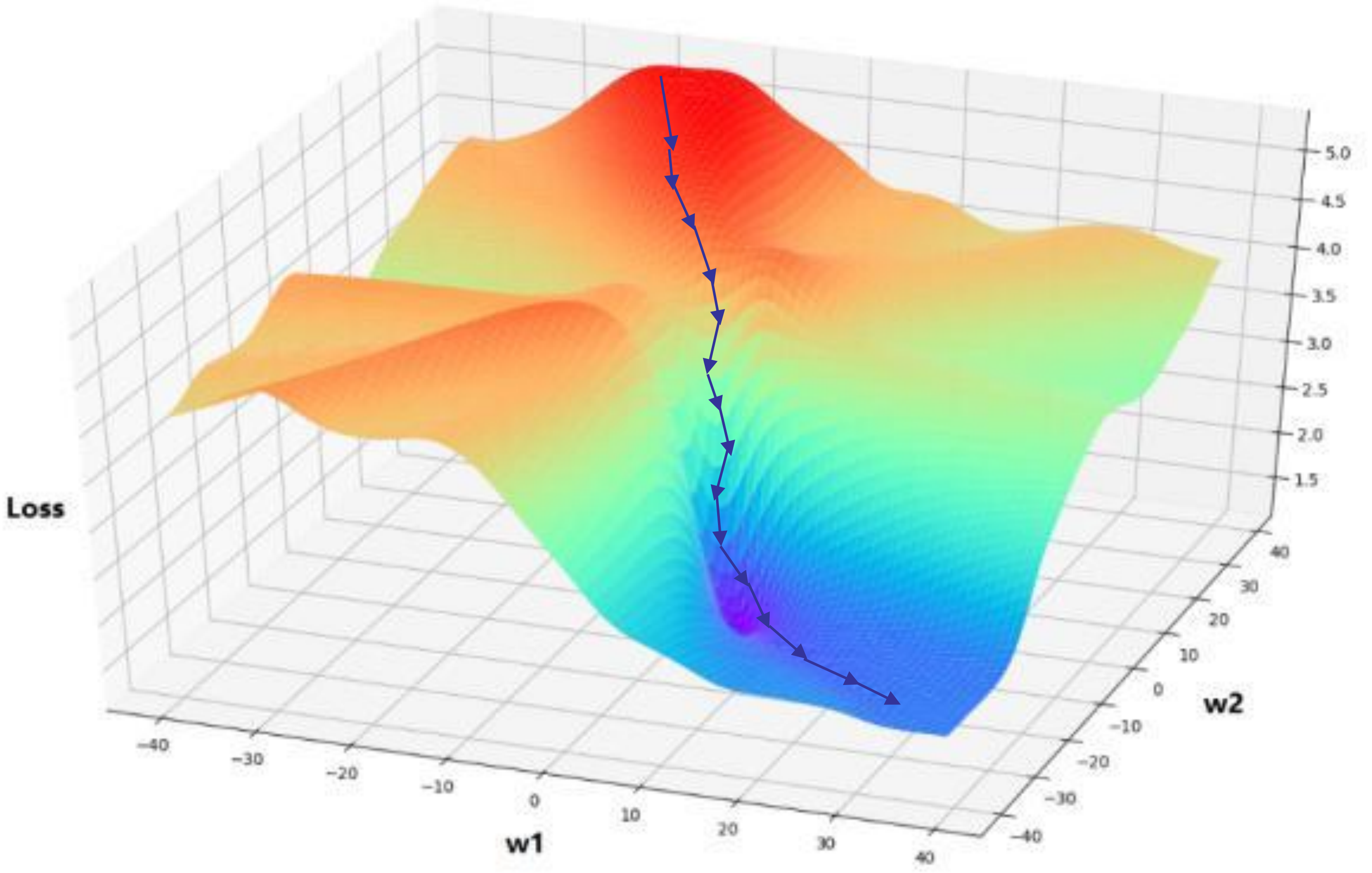
梯度下降法



沿着当前所在位置附近**最陡峭**的那条路就能最快到达山谷

- ✓ 梯度下降法：沿**梯度反方向更新参数**不断地**逼近极小值**的方法。
  - ✓ **梯度**是一个向量（矢量），表示某一函数在该点处的方向导数沿着该方向取得最大值，即函数在该点处沿着梯度方向变化最快，变化率最大。

梯度下降法



- ✓ 方向
- ✓ 距离/步长
- ✓ 终止条件

梯度下降法

- ✓ 多元线性回归

$$\min_{\hat{\boldsymbol{w}}} J(\hat{\boldsymbol{w}}) = (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{w}})^T(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{w}})$$

  - ✓ 求出梯度

$$\frac{\partial J(\hat{\boldsymbol{w}})}{\partial \hat{\boldsymbol{w}}} = -2\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{w}}) = 2\boldsymbol{X}^T(\boldsymbol{X}\hat{\boldsymbol{w}} - \boldsymbol{y})$$

  - ✓ 更新参数

$$\boxed{\hat{\boldsymbol{w}}_{t+1} := \hat{\boldsymbol{w}}_t - \eta\boldsymbol{X}^T(\boldsymbol{X}\hat{\boldsymbol{w}}_t - \boldsymbol{y})}$$

梯度下降的三种形式

批量梯度下降（Batch Gradient Descent,BGD）

梯度下降的每一步中，都用到了所有的训练样本

随机梯度下降（Stochastic Gradient Descent,SGD）

梯度下降的每一步中，用到一个样本，在每一次计算之后便更新参数 ，而不需要首先将所有的训练集求和

小批量梯度下降（Mini-Batch Gradient Descent,MBGD）

梯度下降的每一步中，用到了一定批量的训练样本

批量梯度下降

批量梯度下降

- ✓ 多元线性回归

$$\min_{\widehat{\boldsymbol{w}}} J(\widehat{\boldsymbol{w}}) = (\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{w}})^T (\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{w}})$$

- ✓ 求出梯度

$$\frac{\partial J(\widehat{\boldsymbol{w}})}{\partial \widehat{\boldsymbol{w}}} = -2\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{w}}) = 2\boldsymbol{X}^T(\boldsymbol{X}\widehat{\boldsymbol{w}} - \boldsymbol{y})$$

- ✓ 更新参数

$$\widehat{\boldsymbol{w}}_{t+1} := \widehat{\boldsymbol{w}}_t - \eta \boldsymbol{X}^T(\boldsymbol{X}\widehat{\boldsymbol{w}}_t - \boldsymbol{y})$$

随机梯度下降法

✓ 多元线性回归

随机梯度下降

$$\min_{\hat{\boldsymbol{w}}} J(\hat{\boldsymbol{w}}) = \left(y_{i_t} - \boldsymbol{X}[i_t,:]\hat{\boldsymbol{w}}\right)^T \left(y_{i_t} - \boldsymbol{X}[i_t,:]\hat{\boldsymbol{w}}\right)$$

✓ 求出梯度

$$\frac{\partial J(\hat{\boldsymbol{w}})}{\partial \hat{\boldsymbol{w}}} = -2\boldsymbol{X}[i_t,:]^T\left(y_{i_t} - \boldsymbol{X}[i_t,:]\hat{\boldsymbol{w}}\right) = 2\boldsymbol{X}[i_t,:]^T\left(\boldsymbol{X}[i_t,:]\hat{\boldsymbol{w}} - y_{i_t}\right)$$

✓ 更新参数

$$\hat{\boldsymbol{w}} := \hat{\boldsymbol{w}} - \eta \boldsymbol{X}[i_t,:]^T\left(\boldsymbol{X}[i_t,:]\hat{\boldsymbol{w}} - y_{i_t}\right)$$

小批量梯度下降法

小批量梯度下降

- ✓ 多元线性回归

$$\min_{\widehat{\boldsymbol{w}}} J(\widehat{\boldsymbol{w}}) = (\boldsymbol{y}[\Omega_t] - \boldsymbol{X}[\Omega_t, :]\widehat{\boldsymbol{w}})^T (\boldsymbol{y}[\Omega_t] - \boldsymbol{X}[\Omega_t, :]\widehat{\boldsymbol{w}})$$

  - ✓ 求出梯度

$$\frac{\partial J(\widehat{\boldsymbol{w}})}{\partial \widehat{\boldsymbol{w}}} = -2\boldsymbol{X}[\Omega_t, :]^T (\boldsymbol{y}[\Omega_t] - \boldsymbol{X}[\Omega_t, :]\widehat{\boldsymbol{w}})$$
$$= 2\boldsymbol{X}[\Omega_t, :]^T (\boldsymbol{X}[\Omega_t, :]\widehat{\boldsymbol{w}} - \boldsymbol{y}[\Omega_t])$$

  - ✓ 更新参数

$$\widehat{\boldsymbol{w}} := \widehat{\boldsymbol{w}} - \eta \boldsymbol{X}[\Omega_t, :]^T (\boldsymbol{X}[\Omega_t, :]\widehat{\boldsymbol{w}} - \boldsymbol{y}[\Omega_t])$$

批量梯度下降

$$\Omega_t = \{1, \cdots, m\}$$

随机梯度下降

$$\Omega_t = \{i_t\}$$

梯度下降与最小二乘法比较

- ✓ **最小二乘法**：
  - ✓ 不需要选择学习率η
  - ✓ 只需一次计算得出解析解
  - ✓ 需要矩阵求逆运算$(X^TX)^{-1}$，计算时间复杂度为$O(d^3)$。当特征数量d较大运算代价大，通常来说$d < 10000$可以接受。
  - ✓ 只适用于线性模型，不适合逻辑回归模型等其他模型。
- ✓ **梯度下降**：
  - ✓ 需要选择学习率η
  - ✓ 需要多次迭代逼近最优解
  - ✓ 当特征数量d较大时也能适用
  - ✓ 适用于各种类型的模型

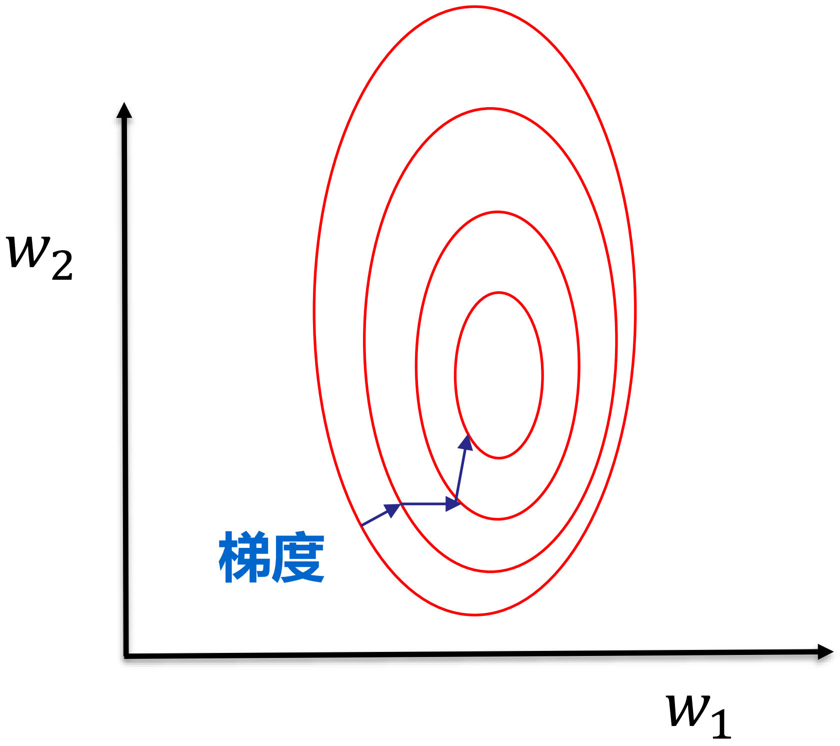
目录

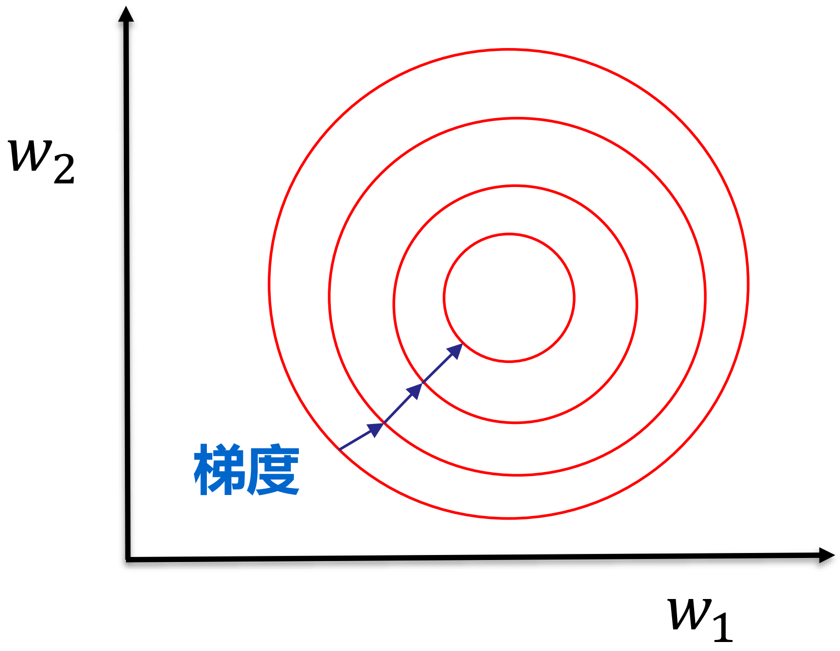
数据归一化/标准化

为什么要标准化/归一化?

提升模型精度：不同维度之间的特征在数值上有一定比较性，可以大大提高分类器的准确性。

加速模型收敛：最优解的寻优过程明显会变得平缓，更容易正确的收敛到最优解。





数据归一化/标准化

归一化（最大 - 最小规范化）

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

将数据映射到[0,1]区间

Z-Score标准化

$$x^* = \frac{x - \mu}{\sigma}$$

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu)^2$$

$$\mu = \frac{1}{m}\sum_{i=1}^{m}x^{(i)}$$

处理后的数据均值为0，方差为1

数据归一化的目的是使得各特征对目标变量的影响一致，会将特征数据进行伸缩变化。

数据标准化为了不同特征之间具备可比性，当数据特征取值范围或单位差异较大时，最好是做一下标准化处理。

数据归一化/标准化

需要做数据归一化/标准化

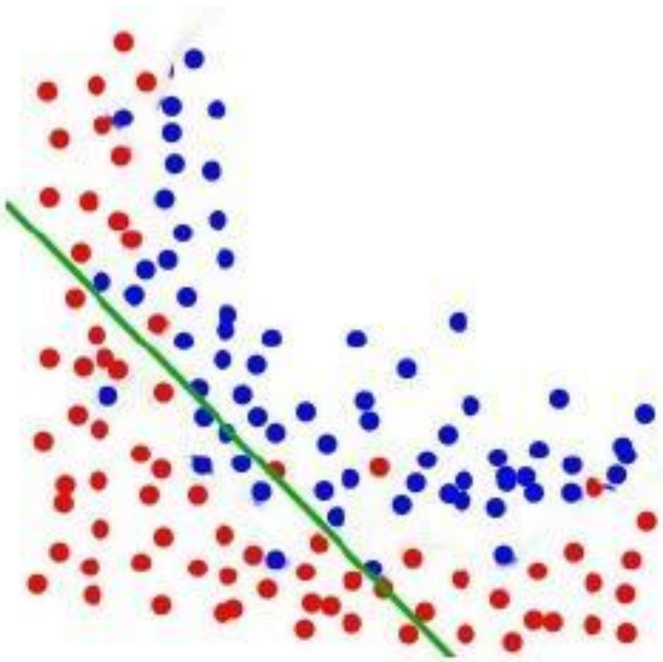
线性模型，如基于距离度量的模型包括KNN(K近邻)、K-means聚类、感知机和SVM。另外，线性回归类的几个模型一般情况下也是需要做数据归一化/标准化处理的。

不需要做数据归一化/标准化

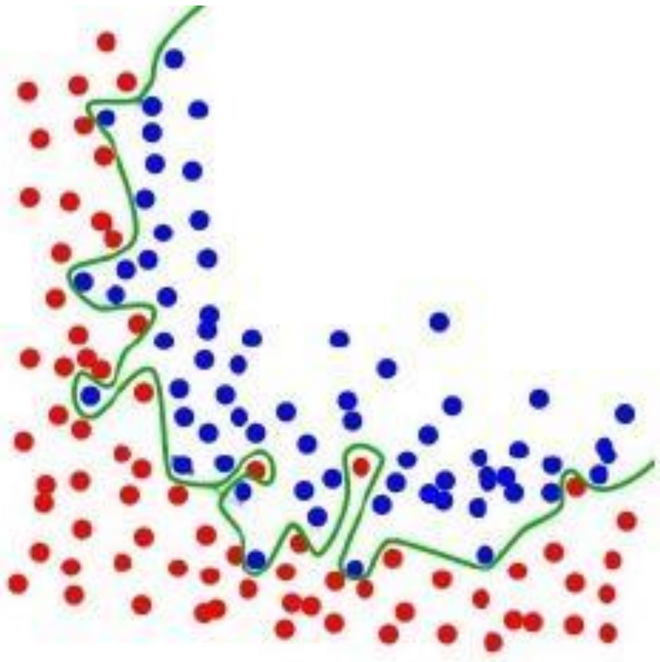
决策树、基于决策树的Boosting和Bagging等集成学习模型对于特征取值大小并不敏感，如随机森林、XGBoost、LightGBM等树模型，以及朴素贝叶斯，以上这些模型一般不需要做数据归一化/标准化处理。

3. 正则化

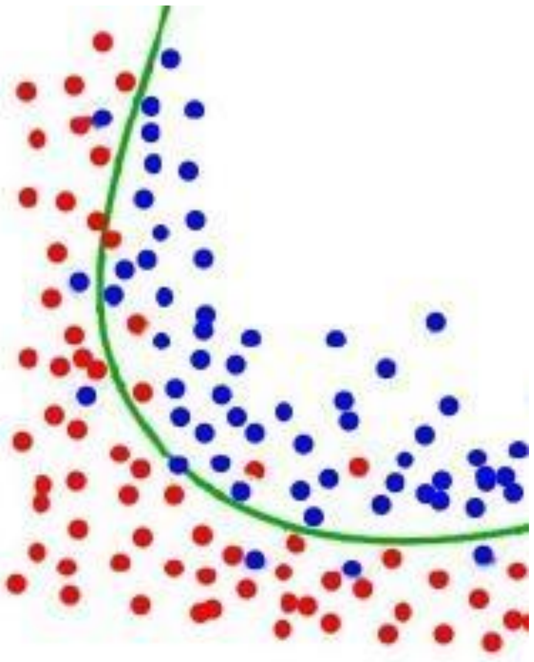
过拟合和欠拟合



欠拟合



过拟合



正合适

过拟合的处理

1.获得更多的训练数据

使用更多的训练数据是解决过拟合问题最有效的手段，因为更多的样本能够让模型学习到更多更有效的特征，减小噪声的影响。

2.**降维**

即丢弃一些不能帮助我们正确预测的特征。可以是手工选择保留哪些特征，或者使用一些模型选择的算法来帮忙（例如PCA）。

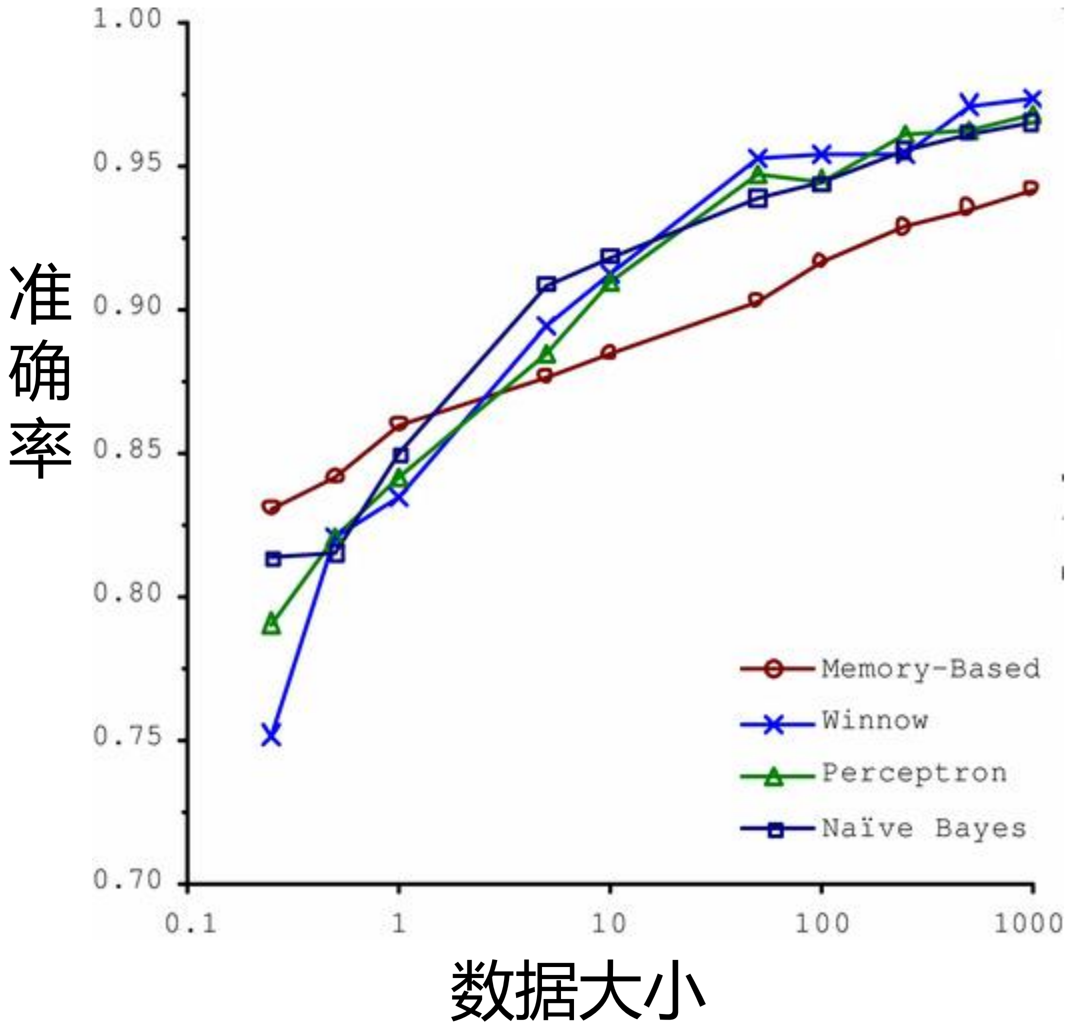
3.**正则化**

正则化(regularization)的技术，保留所有的特征，但是减少参数的大小（magnitude），它可以改善或者减少过拟合问题。

4.**集成学习方法**

集成学习是把多个模型集成在一起，来降低单一模型的过拟合风险。

数据决定一切



通过这张图可以看出，各种不同算法在输入的数据量达到一定级数后，都有相近的高准确度。于是诞生了机器学习界的名言：

成功的机器学习应用不是拥有最好的算法，而是拥有最多的数据！

欠拟合的处理

1.**添加新特征**

当特征不足或者现有特征与样本标签的相关性不强时，模型容易出现欠拟合。通过挖掘组合特征等新的特征，往往能够取得更好的效果。

2.**增加模型复杂度**

简单模型的学习能力较差，通过增加模型的复杂度可以使模型拥有更强的拟合能力。例如，在线性模型中添加高次项，在神经网络模型中增加网络层数或神经元个数等。

3.**减小正则化系数**

正则化是用来防止过拟合的，但当模型出现欠拟合现象时，则需要有针对性地减小正则化系数。

正则化

L_1**正则化**：$J(w)=\frac{1}{2}\sum_{i=1}^{m}(h(x^{(i)})-y^{(i)})^2+\lambda\sum_{j=1}^{n}|w_j|$，Lasso Regression（Lasso回归）

L_2**正则化**：$J(w)=\frac{1}{2}\sum_{i=1}^{m}(h(x^{(i)})-y^{(i)})^2+\lambda\sum_{j=1}^{n}w_j^2$，Ridge Regression （岭回归）

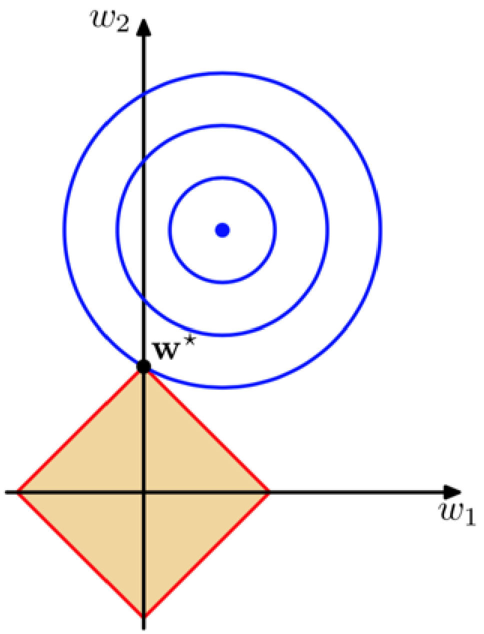
Elastic Net：$J(w)=\frac{1}{2}\sum_{i=1}^{m}(h(x^{(i)})-y^{(i)})^2+\lambda(\rho\cdot\sum_{j=1}^{n}|w_j|+(1-\rho)\cdot\sum_{j=1}^{n}w_j^2)$

(弹性网络)

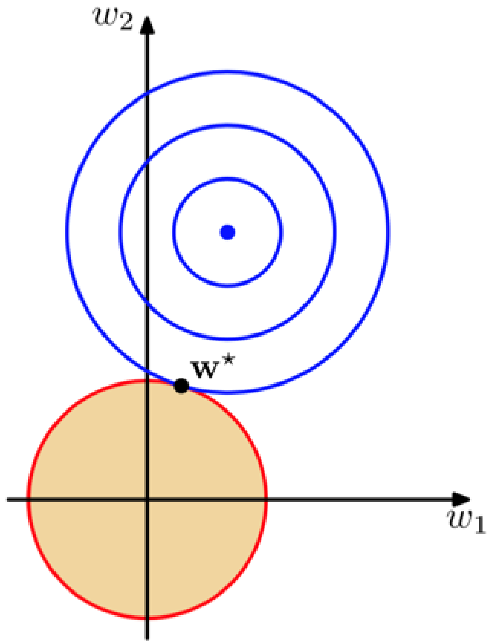
其中：

- λ为正则化系数，调整正则化项与训练误差的比例，λ>0。
- 1≥ρ≥0为比例系数，调整L_1正则化与L_2正则化的比例。

正则化



L_1正则化是指在损失函数中加入权值向量w的绝对值之和，L_1的功能是使权重稀疏



在损失函数中加入权值向量w的平方和，L_2的功能是使权重平滑。

L_1正则化可以产生稀疏模型

L_2正则化可以防止过拟合

图上面中的蓝色轮廓线是没有正则化损失函数的等高线，中心的蓝色点为最优解，左图、右图分别为L_1、L_2正则化给出的限制。

可以看到在正则化的限制之下, L_2正则化给出的最优解w*是使解更加靠近原点,也就是说L_2正则化能降低参数范数的总和。

L_1正则化给出的最优解w*是使解更加靠近某些轴,而其它的轴则为0,所以L_1正则化能使得到的参数稀疏化。

4. 回归的评价指标

回归的评价指标

均方误差（Mean Square Error,MSE）

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - \hat{y}^{(i)})^2$$

均方根误差 RMSE(Root Mean Square Error,RMSE)

$$RMSE(y,\hat{y}) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - \hat{y}^{(i)})^2}$$

平均绝对误差（Mean Absolute Error,MAE）

$$MAE(y,\hat{y}) = \frac{1}{m}\sum_{i=1}^{n}\left|y^{(i)} - \hat{y}^{(i)}\right|$$

其中，$y^{(i)}$ 和$\hat{y}^{(i)}$分别表示第i个样本的真实值和预测值，m 为样本个数。

回归的评价指标

R方 [$RSquared$(r2score)]

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{i=0}^{m}(y^{(i)} - \hat{y}^{(i)})^2}{\sum_{i=0}^{m}(y^{(i)} - \bar{y})^2} = \frac{\mathrm{SSR}}{\mathrm{SST}}$$

$$= 1 - \frac{\mathrm{SSE}}{\mathrm{SST}}$$

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{i=0}^{m}(y^{(i)} - \hat{y}^{(i)})^2/m}{\sum_{i=0}^{m}(y^{(i)} - \bar{y})^2/m}$$

$$= 1 - \frac{\mathrm{MSE}}{\mathrm{Var}}$$

越接近于1,说明模型拟合得越好

$$SSR = \sum_{i=0}^{m}(\hat{y}^{(i)} - \bar{y})^2$$

$$SSE = \sum_{i=0}^{m}(y^{(i)} - \hat{y}^{(i)})^2$$

$$SST = \sum_{i=0}^{m}(y^{(i)} - \bar{y})^2$$

其中，$y^{(i)}$ 和$\hat{y}^{(i)}$分别表示第i个样本的真实值和预测值，m 为样本个数。

参考文献

1. Prof. Andrew Ng. Machine Learning. Stanford University
2. 《统计学习方法》，清华大学出版社，李航著，2019年出版
3. 《机器学习》，清华大学出版社，周志华著，2016年出版
4. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag, 2006
5. Stephen Boyd, Lieven Vandenberghe, Convex Optimization, Cambridge University Press, 2004

谢谢观赏 下节课见