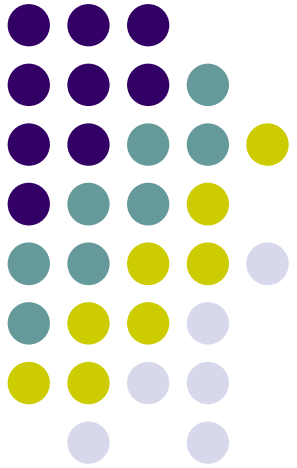
分类及回归问题

哈尔滨工业大学计算学部 刘远超



分类及回归问题

本章内容简介

- 分类与回归问题概述
- 分类问题的常见性能度量方法
- 典型的浅层机器学习分类方法
- 回归问题的常见性能度量方法
- 线性回归及基于梯度下降的线性回归模型参数求解方法
- 多项式回归
- 过拟合现象及损失函数的正则化处理
- 逻辑回归

分类及回归问题

本章内容简介

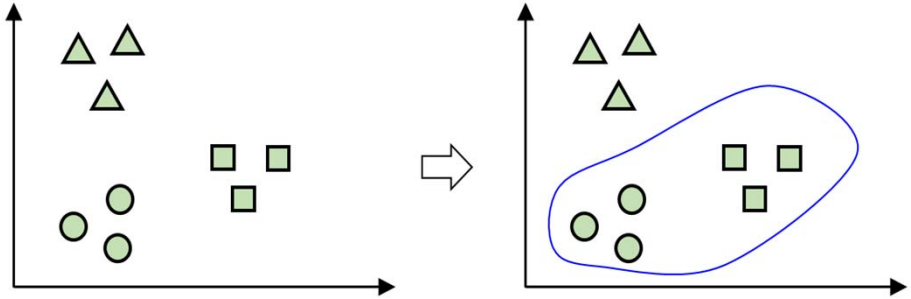
- **分类与回归问题概述**
- 分类问题的常见性能度量方法
- 典型的浅层机器学习分类方法
- 回归问题的常见性能度量方法
- 线性回归及基于梯度下降的线性回归模型参数求解方法
- 多项式回归
- 过拟合现象及损失函数的正则化处理
- 逻辑回归

分类与回归问题概述

分类问题

- **分类问题**是有监督学习的核心问题，其目的是预测输入样本所属的有限个类别，输出变量为离散值。例如，预测输入图像中动物的类别是猫还是狗，就是一个图像分类问题。

- **两类分类问题与多类分类问题**

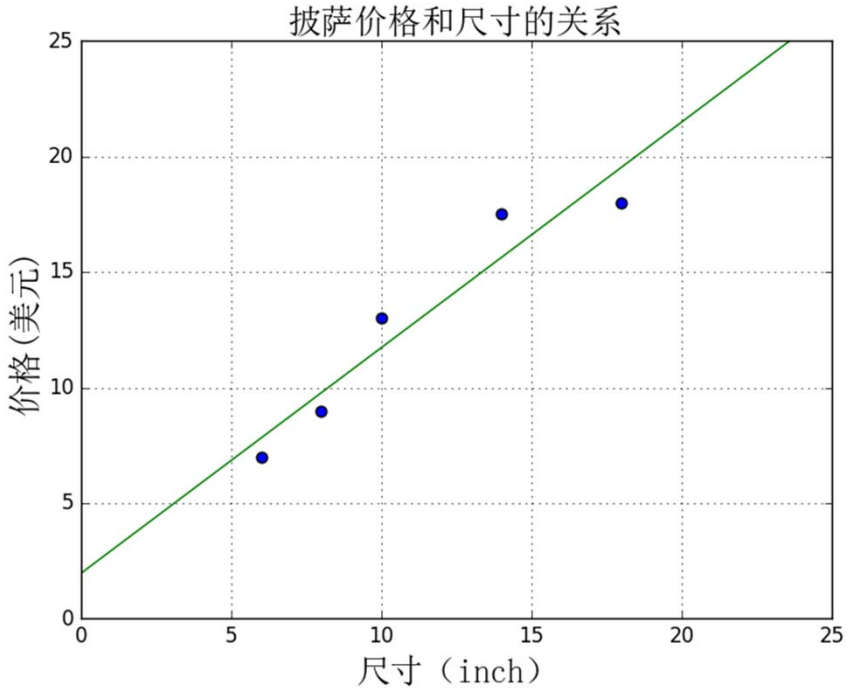


两类分类问题是核心问题，而多类问题可以转化为两类问题解决。一种常用的策略是采用一对其余（One-vs-Rest）：每次将其中一个类标记为正类，然后将剩余的其它类都标记成负类。

分类与回归问题概述

回归问题

什么是回归问题。回归也属于有监督学习范畴，其侧重在预测输入样本的连续的量化数值。例如根据披萨的尺寸预测其具体的价格就属于回归问题。



分类与回归问题概述

回归和分类的区别和联系

- 区别：
  - 分类：使用训练集推断输入 x 所对应的离散类别（如：**+1**，**-1**）。
  - 回归：使用训练集推断输入x所对应的输出值，为连续实数。
- 联系：
  - 利用回归模型进行分类：可将回归模型的输出离散化以进行分类，即$y = sign(f(x))$。
  - 利用分类模型进行回归：也可利用分类模型的特点，输出其连续化的数值。

分类及回归问题

本章内容简介

- 分类与回归问题概述
- 分类问题的常见性能度量方法
- 典型的浅层机器学习分类方法
- 回归问题的常见性能度量方法
- 线性回归及基于梯度下降的线性回归模型参数求解方法
- 多项式回归
- 过拟合现象及损失函数的正则化处理
- 逻辑回归

分类及回归问题

本章内容简介

- 分类与回归问题概述
- **分类问题的常见性能度量方法**
- 典型的浅层机器学习分类方法
- 回归问题的常见性能度量方法
- 线性回归及基于梯度下降的线性回归模型参数求解方法
- 多项式回归
- 过拟合现象及损失函数的正则化处理
- 逻辑回归

分类问题的常见性能度量方法

准确率

- 假设只有两类样本，即正例(positive)和负例(negative)。通常以关注的类为正类，其他类为负类。

实际类别	预测类别		
	正	负	总计
正	TP	FN	P(实际为正)
负	FP	TN	N(实际为负)

表中AB模式：第二个符号表示预测的类别，第一个表示预测结果对了(True)还是错了(False)

- 分类准确率（accuracy）：分类器正确分类的样本数与总样本数之比:

$$accuracy = \frac{TP+TN}{P+N}$$

思考： *假设共有100个短信，其实际情况为，其中有1个是垃圾短信，99个是非垃圾短信。某分类模型将这100个短信都分为非垃圾短信，则准确率（**accuracy**）为？*

分类问题的常见性能度量方法

精确率和召回率

实际类别	预测类别		
	正例	负例	总计
正例	TP	FN	P(实际为正例)
负例	FP	TN	N(实际为负例)

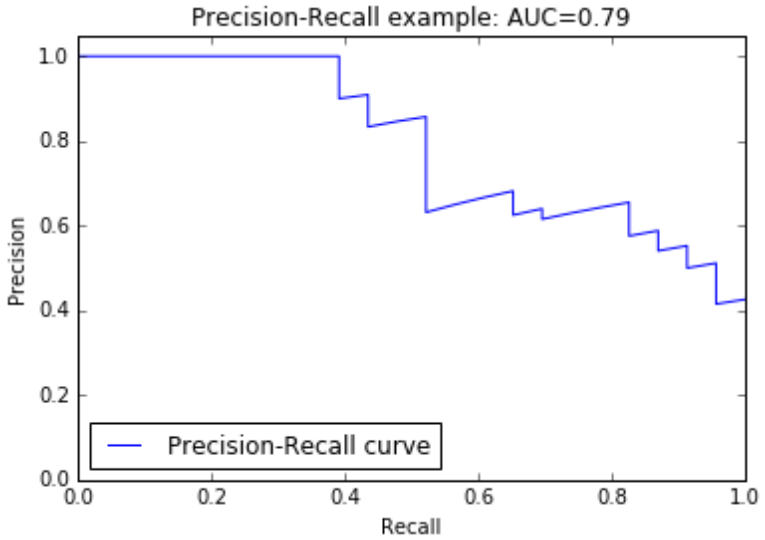
- **精确率(precision)和召回率(recall)：** 是二类分类问题常用的评价指标。

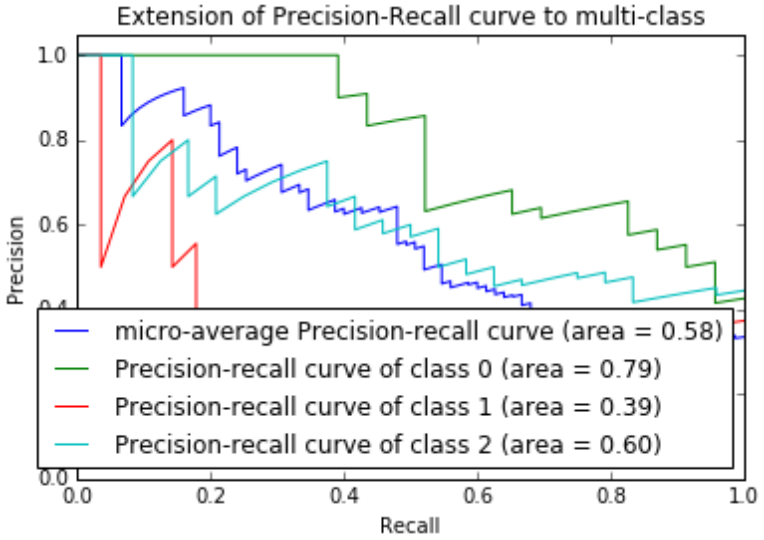
$$\text{precision} = \frac{TP}{TP+FP} \qquad \text{recall} = \frac{TP}{P}$$

- **精确率**反映了模型判定的正例中真正正例的比重。在垃圾短信分类器中，是指预测出的垃圾短信中真正垃圾短信的比例。

- **召回率**反映了总正例中被模型正确判定正例的比重。医学领域也叫做灵敏度（sensitivity）。在垃圾短信分类器中，指所有真的垃圾短信被分类器正确找出来的比例。

分类性能度量—P-R曲线





- Area (Area Under Curve, 或者简称AUC)
  - Area的定义（p-r曲线下的面积）如下:

    $$Area = \int_0^1 p(r)dr$$

  - Area有助于弥补P、R的单点值局限性，可以反映全局性能。

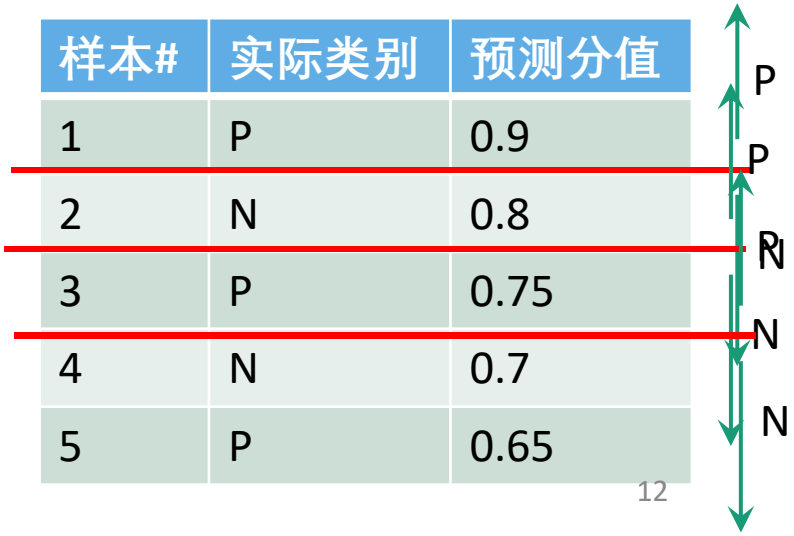
如何绘制P-R曲线

- 要得到P-R曲线，需要一系列Precision和Recall的值。这些系列值是通过阈值来形成的。对于每个测试样本，分类器一般都会给了“Score”值，表示该样本多大概率上属于正例。
- 步骤：
  1. 从高到低将“Score”值排序并依此作为阈值threshold；
  2. 对于每个阈值，“Score”值大于或等于这个threshold的测试样本被认为正例，其它为负例。从而形成一组预测数据。

实际类别	预测类别		
	正例	负例	总计
正例	TP	FN	P(实际为正例)
负例	FP	TN	N(实际为负例)

$$(\text{precision} = \frac{TP}{TP+FP}, \quad \text{recall} = \frac{TP}{P})$$

样本#	实际类别	预测分值
1	P	0.9
2	N	0.8
3	P	0.75
4	N	0.7
5	P	0.65



分类性能度量--F值

- **F值($F_\beta-score$)是精确率和召回率的调和平均:**

$$F_\beta-score=\frac{(1+\beta^2)*precision*recall}{(\beta^2*precision+recall)}$$

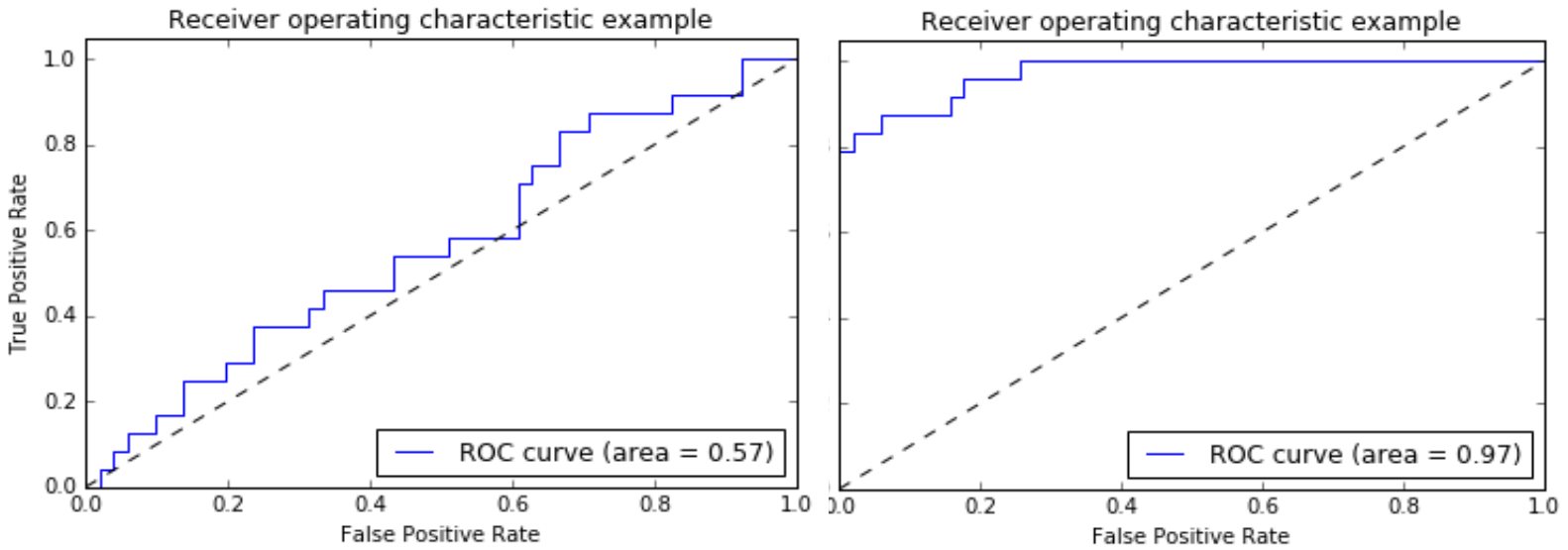
- β一般大于0。当β=1时，退化为F1:

$$F_1-score=\frac{2*precision*recall}{(precision+recall)}$$

- 比较常用的是F_1，即表示二者同等重要

分类问题的常见性能度量方法

ROC值



- **横轴**：假正例率 $fp\ rate = \frac{FP}{N}$
- **纵轴**：真正例率 $tp\ rate = \frac{TP}{P}$
- ROC (受试者工作特征曲线, receiver operating characteristic curve)描绘了分类器在$tp\ rate$(真正正例占总正例的比率，反映**命中概率**，纵轴)和$fp\ rate$ (错误的正例占反例的比率，反映误诊率、假阳性率、**虚惊概率**，横轴) 间的trade-off。

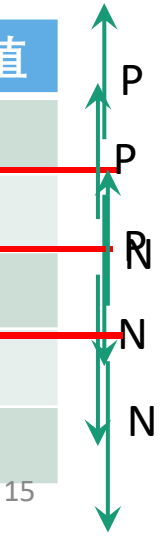
分类性能度量—ROC曲线绘制

- 要得到一个曲线，需要一系列$fp\ rate$和$tp\ rate$的值。这些系列值是通过阈值来形成的。对于每个测试样本，分类器一般都会给了“Score”值，表示该样本多大概率上属于正例（或负例）。
- 步骤：
  1. 从高到低将“Score”值排序并依此作为阈值threshold；
  2. 对于每个阈值，“Score”值大于或等于这个threshold的测试样本被认为正例，其它为负例。从而形成一组预测数据。

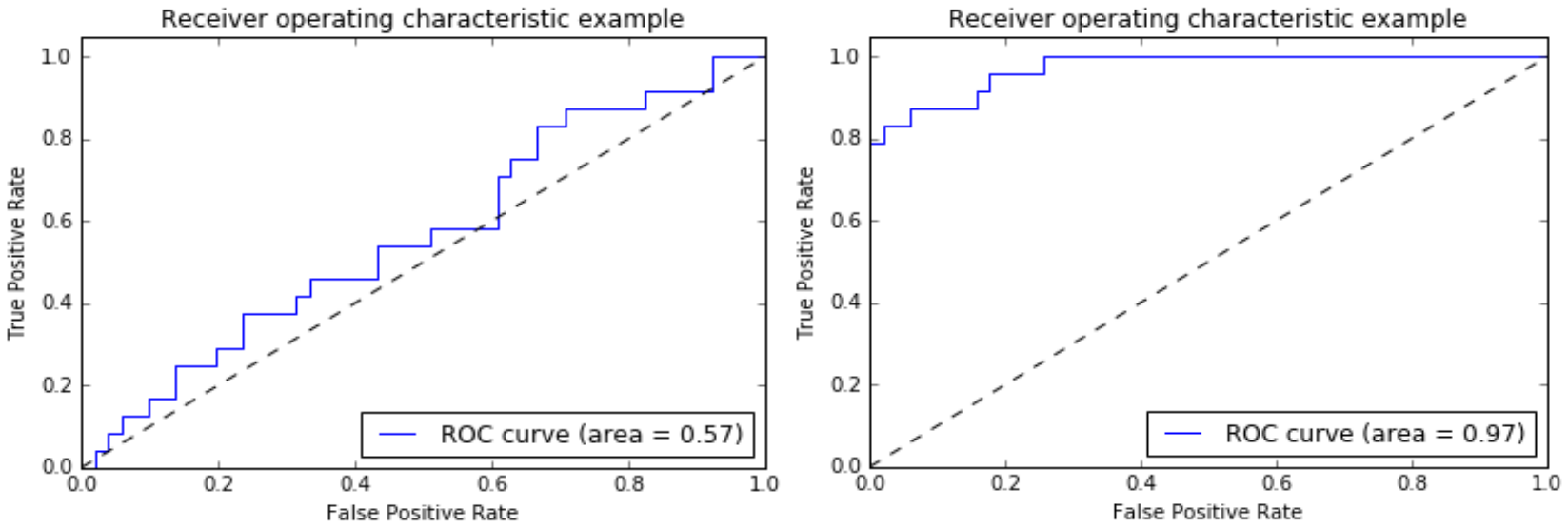
实际类别	预测类别		
	正例	负例	总计
正例	TP	FN	P(实际为正例)
负例	FP	TN	N(实际为负例)

$$(fp\ rate = \frac{FP}{N},\quad tp\ rate = \frac{TP}{P})$$

样本#	实际类别	预测分值
1	P	0.9
2	N	0.8
3	P	0.75
4	N	0.7
5	P	0.65



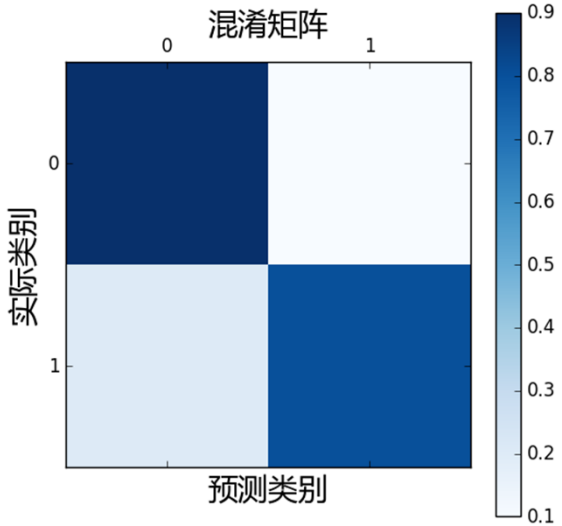
分类性能度量—ROC-AUC计算



- **ROC- AUC（Area Under Curve）定义**为ROC曲线下的面积
- AUC值提供了分类器的一个整体数值。通常AUC越大，分类器更好
- 取值范围为[0,1]

分类性能可视化

实际类别	预测类别		
	正例	负例	总计
正例	TP	FN	P(实际为正例)
负例	FP	TN	N(实际为负例)



- **混淆矩阵（Confusion matrix）的可视化**
  - 如用热图（heatmap）直观地展现类别的混淆情况（每个类有多少样本被错误地预测成另一个类）

分类报告

- **分类报告(Classification report)**显示每个类的分类性能。包括每个类标签的精确率、召回率、F1值等。

	precision	recall	f1-score	support
class 0	0.67	1.00	0.80	2
class 1	0.00	0.00	0.00	1
class 2	1.00	1.00	1.00	2
avg / total	0.67	0.80	0.72	5

分类及回归问题

本章内容简介

- 分类与回归问题概述
- 分类问题的常见性能度量方法
- 典型的浅层机器学习分类方法
- 回归问题的常见性能度量方法
- 线性回归及基于梯度下降的线性回归模型参数求解方法
- 多项式回归
- 过拟合现象及损失函数的正则化处理

分类及回归问题

本章内容简介

- 分类与回归问题概述
- 分类问题的常见性能度量方法
- **典型的浅层机器学习分类方法**
  - 支持向量机
  - 朴素贝叶斯分类器
- 回归问题的常见性能度量方法
- 线性回归及基于梯度下降的线性回归模型参数求解方法
- 多项式回归
- 过拟合现象及损失函数的正则化处理
- 逻辑回归

分类及回归问题

本章内容简介

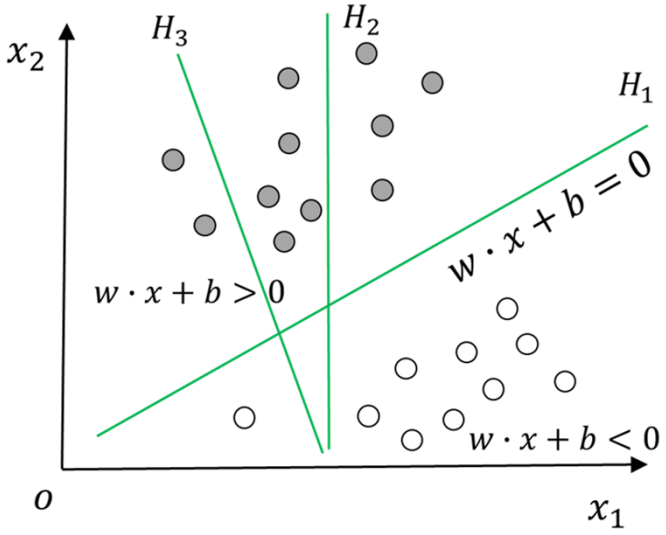
- 分类与回归问题概述
- 分类问题的常见性能度量方法
- **典型的浅层机器学习分类方法**
  - **支持向量机**
  - 朴素贝叶斯分类器
- 回归问题的常见性能度量方法
- 线性回归及基于梯度下降的线性回归模型参数求解方法
- 多项式回归
- 过拟合现象及损失函数的正则化处理
- 逻辑回归

支持向量机

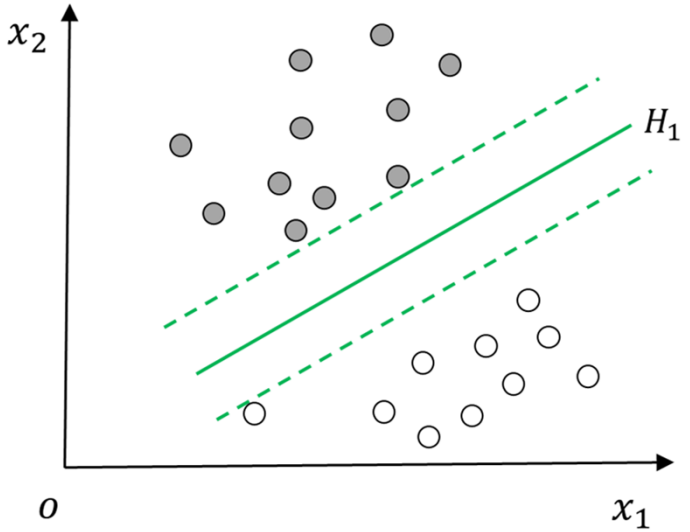
什么是支持向量机?

先看一个简单的二维平面上数据点分类的例子。

问题：如何对图4.5(a)所示的二维特征空间中的两类数据点（分别由实心点和空心点代表）进行分类?



（a）不同分类超平面的比较

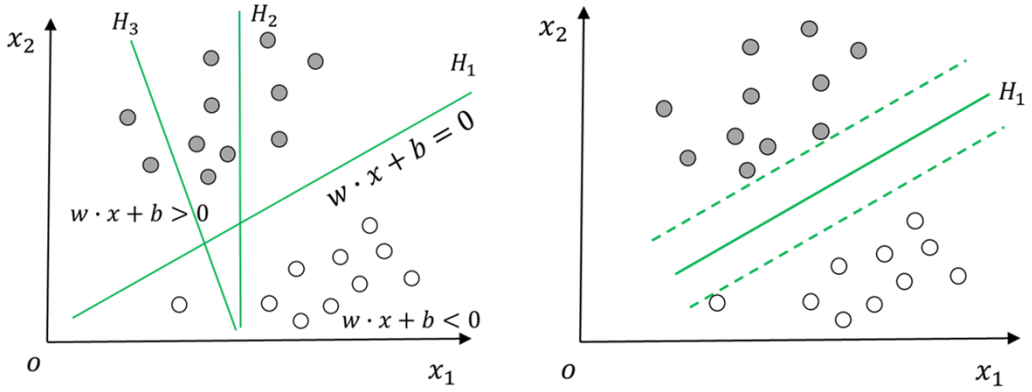


（b）最大分类间隔

对于图 **4.5** 所示的数据点，线性可分支持向量机是指能将两类数据正确划分并且间隔最大的直线。

支持向量机

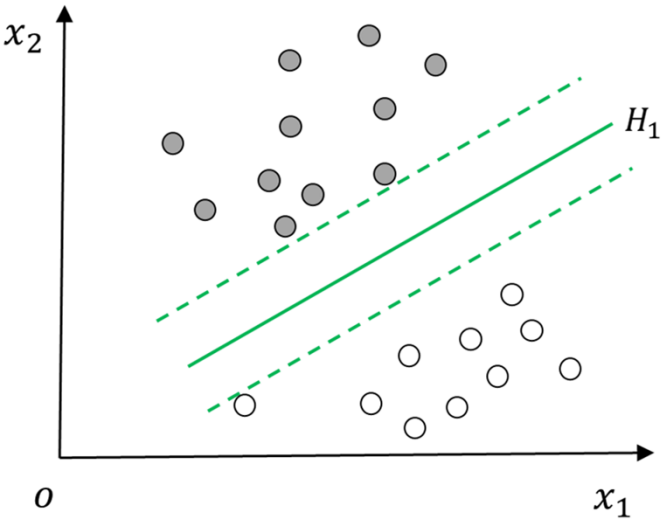
相关概念



（a）不同分类超平面的比较　　（b）最大分类间隔

- **什么是超平面？**在几何形状中，n维空间V的超平面是维度为n-1的子空间。
- **数据集数学描述：**假设有训练数据集 $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$，$x_i$ 为第 i 个样本的特征向量，$x_i \in R^n$；y_i 为第 i 个样本的类别标签，$y_i \in \{+1, -1\}$，$i = 1,2, \dots, N$。**定义当 $y_i = +1$ 时，样本为正类；当 $y_i = -1$ 时，样本为负类。**
- **数据集的线性可分性。**如果存在某个分离超平面 $H: w \cdot x + b = 0$ 能将数据集T中的两类样本点完全正确地分到该超平面的两侧，则称数据集T为线性可分数据集；

支持向量机



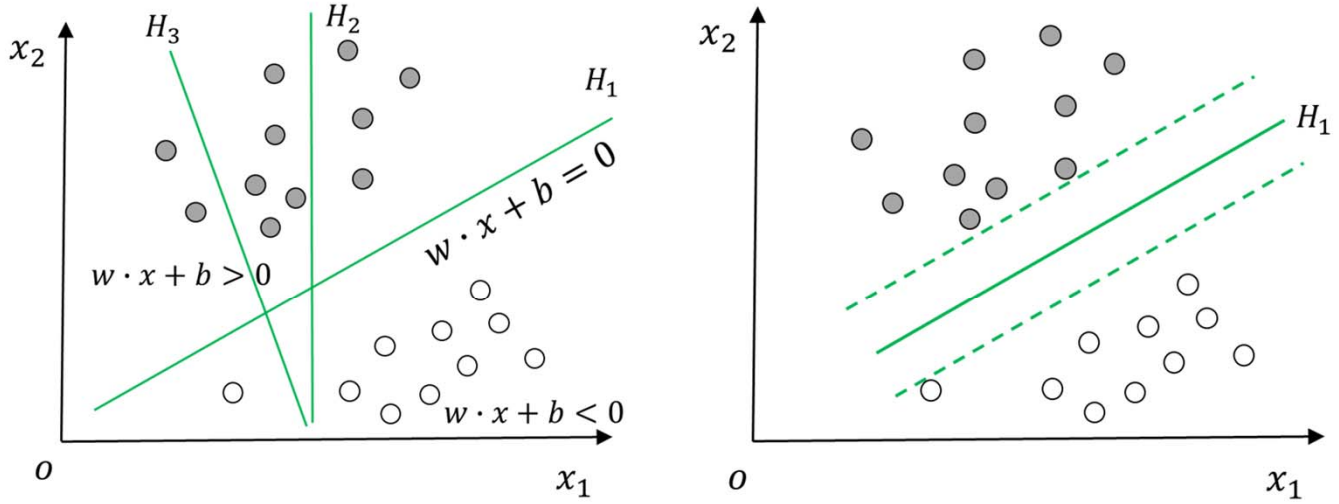
- **支持向量机的学习目标**：当训练数据集线性可分时，**支持向量机的学习目标**就是找到将两类数据正确划分并且分类间隔最大的超平面。

- **超平面关于样本点的几何间隔：**
  - 设分离超平面为S: $w \cdot x + b = 0$，则样本点x_i到该超平面的距离为$\frac{1}{||w||}|w \cdot x_i + b|$（具体推导过程略），也称之为**几何间隔$\gamma_i$**。

- **超平面关于样本点的函数间隔**：
  - 由于$\frac{1}{||w||}$只与权重有关，取值相对固定，因此可以用$\hat{\gamma}_i = |w \cdot x_i + b|$相对地表示样本点$x_i$到分离超平面的远近程度。
  - 由于根据之前的定义，对于线性可分数据集的正确分离超平面，类标签y_i的符号与$w \cdot x_i + b$的符号始终一致，因此可以用$\hat{\gamma}_i = y_i(w \cdot x_i + b)$表示超平面关于样本点$(x_i, y_i)$的距离，也称为**函数间隔**。
  - 函数间隔的取值大于等于0，因而可以用于表示距离。

支持向量机

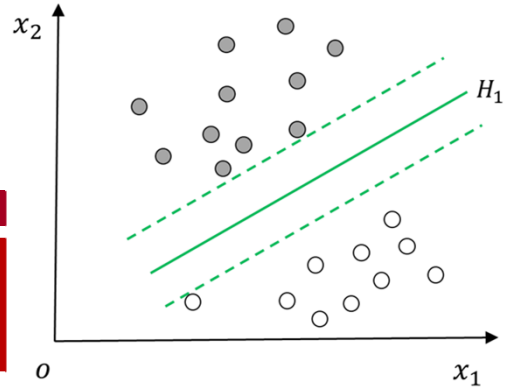
分离超平面关于训练集T的函数间隔



(a) 不同分类超平面的比较　　(b) 最大分类间隔

- 据此，**可以给出分离超平面关于训练集T的函数间隔**。其定义为该超平面关于训练集T中所有样本点的函数间隔的**最小值**，即$\hat{\gamma} = \min_{i=1,\dots,N} \hat{\gamma}_i$。
- 同理可以定义超平面关于训练数据集T的几何间隔$\gamma = \min_{i=1,\dots,N} \gamma_i$。

支持向量机



（一）完全线性可分情况下的支持向量机

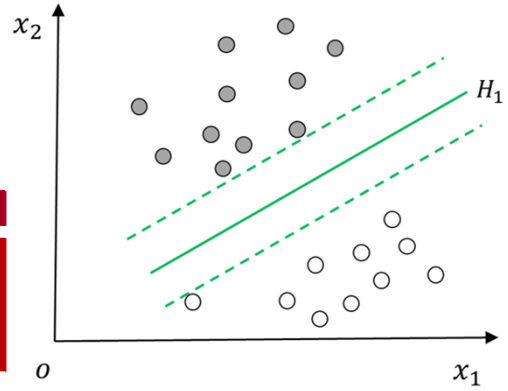
最优分离超平面的标准为：几何间隔（或函数间隔）最大化。

- 基于上述关于间隔的定义，可以得到训练数据集T的最优分离超平面的参数w和b的确定方法，即其应该使分离超平面关于训练集T的几何间隔最大化：

$$\max_{w,b} \gamma, \quad \textbf{s.t. } y_i\left(\frac{w}{||w||} \cdot x_i + \frac{b}{||w||}\right) \geq \gamma, i = 1, \dots, N \qquad \textbf{(4.6)}$$

公式含义：在分离超平面关于训练集中每个样本点的几何间隔大于或者等于γ的约束条件下，求使几何间隔γ最大的w和b取值，以得到最优分离超平面。

支持向量机



（一）完全线性可分情况下的支持向量机

补充：根据几何间隔γ和函数间隔$\hat{\gamma}$的关系，上述基于几何间隔的优化可利用函数间隔$\hat{\gamma}$表示：

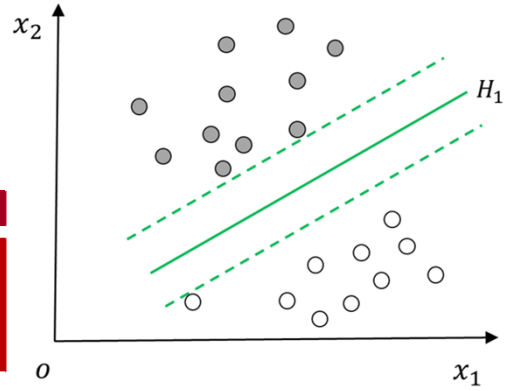
$$\max_{w,b} \frac{\hat{\gamma}}{||w||}, \quad \textbf{s.t.} \ y_i(w \cdot x_i + b) \geq \hat{\gamma}, \ i = 1, \dots, N \qquad (4.7)$$

- 函数间隔$\hat{\gamma}$的取值不影响此最优化问题的解，例如将w和b按照比例缩放，则对不等式约束以及目标函数的优化没有影响，因此可以取$\hat{\gamma} = 1$。
- 另外由于最大化$\frac{1}{||w||}$与最小化$\frac{1}{2}||w||^2$是等价的，因此得到如下的线性可分支持向量机的优化问题:

$$\min_{w,b} \frac{1}{2}||w||^2, \quad \textbf{s.t.} \ y_i(w \cdot x_i + b) - 1 \geq 0, \ i = 1, \dots, N \qquad (4.8)$$

- 求解该优化问题，得到优化的参数w^*和b^*，即可得到优化的分离超平面$w^* \cdot x + b^* = 0$。相应地，分类决策函数为$f(x) = sign(w^* \cdot x + b^*)$，将测试集中的样本特征$x$代入该分类决策函数，可以通过返回的符号判断样本的分类结果。

支持向量机



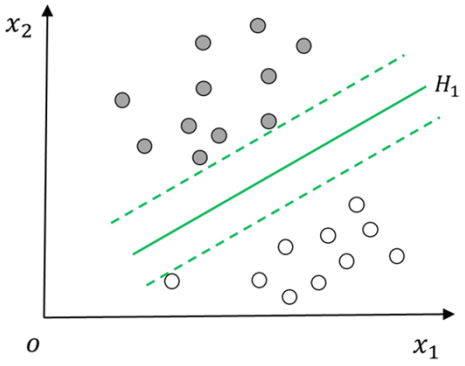
（一）完全线性可分情况下的支持向量机

- **支持向量：**
  - 在线性可分情况下，训练数据集中与分离超平面距离最近的样本点称为支持向量(support vector)。因而支持向量是使约束条件中等号成立的样本点，即 $y_i(w \cdot x_i + b) - 1 = 0$。
- **将优化问题转换为对偶问题：**
  - 为了求解上述约束最优化问题，支持向量机利用拉格朗日对偶性(Lagrange duality)将其转换为对偶问题进行求解。其对每个不等式约束引进拉格朗日乘子$\alpha_i \geq 0$，$i = 1, \ldots, N$，从而得到拉格朗日函数[Friedman, 2001]：

$$L(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{N} \alpha_i y_i (w \cdot x_i + b) + \sum_{i=1}^{N} \alpha_i \qquad \textbf{(4.9)}$$

其中$\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N)^T$为拉格朗日乘子向量。

支持向量机



（一）完全线性可分情况下的支持向量机

- **则原始问题的优化转为：** $\max_{\alpha} \min_{w,b} L(w,b,\alpha)$**。即先求**$L$**对**$w,b$**的极小，再求对**$\alpha$**的极大：**

（1）求 $\min_{w,b} L(w,b,\alpha)$**。**将拉格朗日函数$L(w,b,\alpha)$分别对w,b求偏导数并令其为0，可以得到$w=\sum_{i=1}^{N}\alpha_i y_i x_i$，$\sum_{i=1}^{N}\alpha_i y_i=0$，从而

$$\min_{w,b} L(w,b,\alpha)= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i\cdot x_j)+\sum_{i=1}^{N}\alpha_i$$

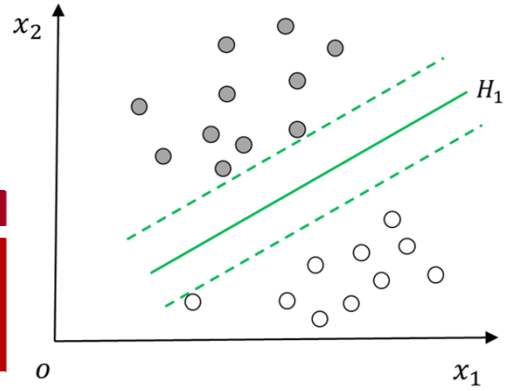
（2）求 $\min_{w,b} L(w,b,\alpha)$ **对**α**的极大，即求：**

$$\max_{\alpha} -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i\cdot x_j)+\sum_{i=1}^{N}\alpha_i \text{，} \quad \textbf{s.t.} \quad \sum_{i=1}^{N}\alpha_i y_i=0 \text{，} \quad \alpha_i\geq 0, i=1,\dots,N\text{。}$$

将此目标函数由求极大转换成求极小，可得如下的对偶最优化问题：

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i\cdot x_j)-\sum_{i=1}^{N}\alpha_i\text{，} \ \textbf{s.t.} \ \sum_{i=1}^{N}\alpha_i y_i=0\text{，} \ \alpha_i\geq 0, i=1,\dots,N$$

支持向量机



（一）完全线性可分情况下的支持向量机

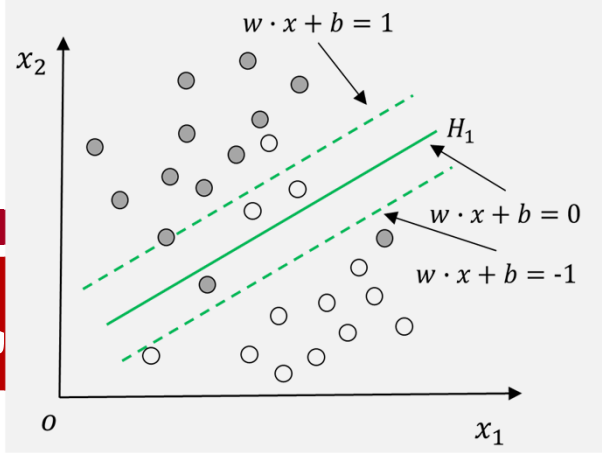
- 假设在线性可分训练集上，该对偶最优化问题的解为$\alpha^* = (\alpha_1^*, \alpha_2^*, \ldots, .\alpha_N^*)^T$，则可以证明（具体过程略）存在$\alpha^*$的一个正分量$\alpha_j^* > 0$，得到原始最优化问题的解为：

$$w^* = \sum_{i=1}^{N} \alpha_i^* y_i x_i \qquad (4.11)$$

$$b^* = y_j - \sum_{i=1}^{N} \alpha_i^* y_i (x_i \cdot x_j) \qquad (4.12)$$

从而可以得到分离超平面$w^* \cdot x + b^* = 0$，以及分类决策函数$f(x) = sign(w^* \cdot x + b^*)$。

支持向量机



（二）不完全线性可分情况下的支持向量机

- 应对策略：从硬间隔变为软间隔

- 在**软间隔最大化**方法中，为每个样本点额外引入一个松弛变量ξ_i $(\xi_i \geq 0)$，使得函数间隔加上松弛变量后大于等于1[Bennett, 1999]。因此**约束条件修正为$y_i(w \cdot x_i + b) + \xi_i \geq 1$。**

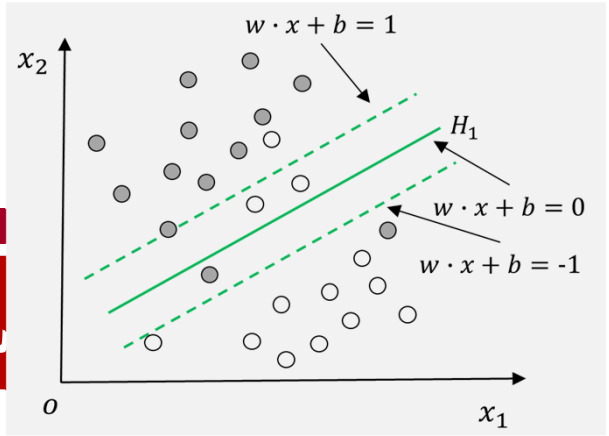
- 则线性不可分数据集上支持向量机的参数优化变为如形式（原始问题）：

$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\xi_i,$$

$$\text{s.t. } y_i(w \cdot x_i + b) \geq 1 - \xi_i,\ i = 1, 2, \ldots, N,\ \ \xi_i \geq 0,\ i = 1, 2, \ldots, N \qquad (4.13)$$

其中$C > 0$为惩罚参数，其取值越大表示对误分类点的惩罚越大。

支持向量机



（二）不完全线性可分情况下的支持向量机

- 可以证明（具体过程略），该原始问题的对偶问题是：

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{N}\alpha_i$$

$$\text{s.t.} \sum_{i=1}^{N}\alpha_i y_i = 0, 0 \le \alpha_i \le C, i = 1, \dots, N \qquad (4.14)$$

- 假设 $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, .\alpha_N^*)^T$ 是该对偶问题的一个解，选择 α^* 的一个分量 α_j^*，满足条件 $0 < \alpha_j^* < C$，则原始问题的解 w^* 和 b^* 可按如下公式求得：

$$w^* = \sum_{i=1}^{N}\alpha_i^* y_i x_i \qquad (4.15)$$

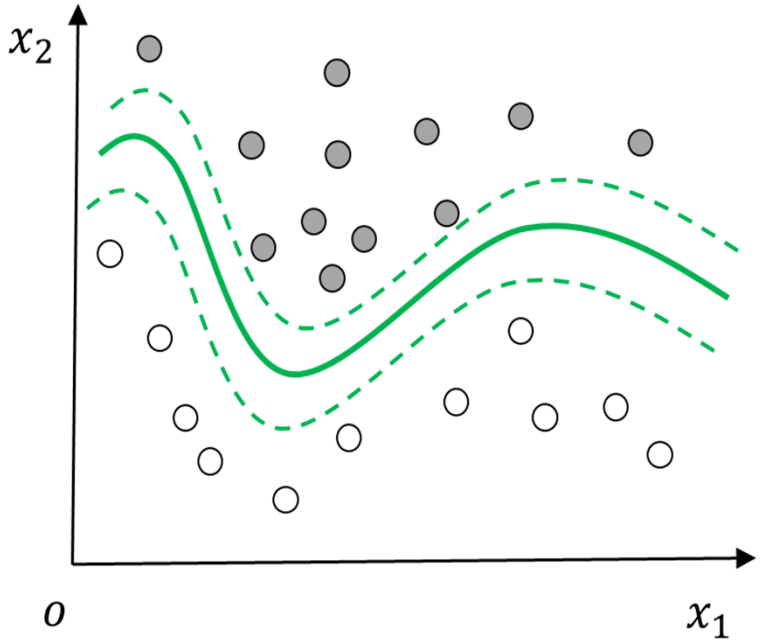
$$b^* = y_j - \sum_{i=1}^{N} y_i \alpha_i^* (x_i \cdot x_j) \qquad (4.16)$$

- 因此可得到分离超平面 $w^* \cdot x + b^* = 0$，分类决策函数 $f(x) = sign(w^* \cdot x + b^*)$。由于b的解不唯一，实际计算时可以取平均值处理。

- 因此，相对于上一节介绍的硬间隔模型，软间隔模型通过引入松弛变量，使分类器可以有一些误分类的点。

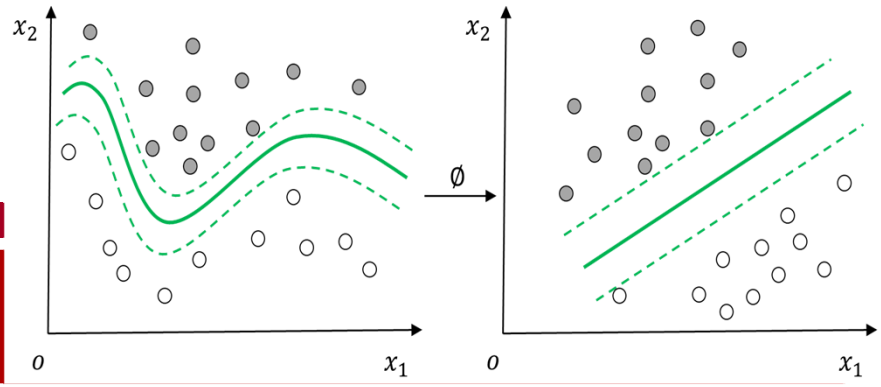
支持向量机

（三）非线性可分情况下的支持向量机



- 非线性可分问题是指利用非线性模型才能很好地分类的问题。如图4.7左侧的数据点无法用直线（线性模型）将正负类样本正确分开，但可以用一条曲线，即非线性模型将它们正确分开。
- 推而广之到高维空间，则非线性可分问题使用的是超曲面，而不是超平面进行划分。

支持向量机



（三）非线性可分情况下的支持向量机

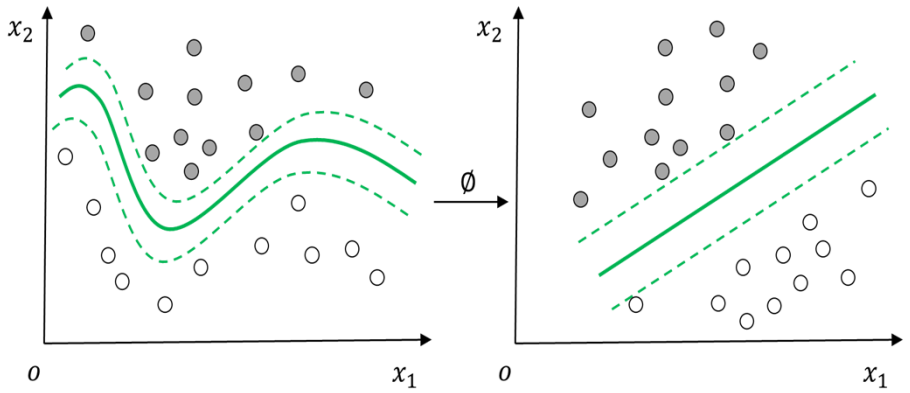
- 对于非线性可分的样本集，可以先使用一个变换$z = \phi(x)$**将非线性特征空间x映射到新的线性特征空间z**，进而在新的z特征空间里使用前文所述的支持向量机方法学习分类模型。
- 相应地，支持向量机的目标函数变为：

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(\boldsymbol{\phi}(\boldsymbol{x_i})\cdot\boldsymbol{\phi}(\boldsymbol{x_j})) - \sum_{i=1}^{N}\alpha_i$$

$$\text{s.t.}\ \sum_{i=1}^{N}\alpha_i y_i = 0,\ \ \alpha_i \geq 0, i = 1,\ldots,N \qquad (4.17)$$

- **核技巧[Hsieh, 2009]** [李航, 2012]：
  - 上述新的目标函数中的$\phi(x_i)\cdot\phi(x_j)$的计算存在的问题是，**难以显式定义映射函数$\phi(x)$和目标特征空间z（一般是较高维度）**。
  - **因此通常使用核技巧来简化这个运算过程。**即使用一个常用的关于x的核函数（如后面介绍的多项式核、高斯核等）$\boldsymbol{K(x_i, x_j)=\phi(x_i)\cdot\phi(x_j)}$来隐式地实现从低维到高维的映射，使问题得以简化。

支持向量机



（三）非线性可分情况下的支持向量机

- 基于核技巧，支持向量机对偶问题的目标函数变为：

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \boldsymbol{K}(\boldsymbol{x_i}, \boldsymbol{x_j}) - \sum_{i=1}^{N}\alpha_i \qquad \textbf{(4.18)}$$

- 可以参照之前介绍的方法求解并获得原始问题的优化参数取值w^*和b^*。
- 相应地，分类决策函数中的内积也可以用核函数来代替：

$$f(x) = sign\left(\sum_{i=1}^{N_s}\alpha_i^* y_i \phi(x_i)\cdot\phi(x) + b^*\right) = sign\left(\sum_{i=1}^{N_s}\alpha_i^* y_i K(x_i, x) + b^*\right) \quad (4.19)$$

- 常用核函数。核函数一般和应用场景相关，在不同领域所应用的核函数可能也不相同[Gönen, 2011]。常用的核函数有：
  - 多项式核函数：$K(x,z) = (x\cdot z + 1)^p$。
  - 高斯核函数：$K(x,z) = \exp\left(-\frac{||x-z||^2}{2\sigma^2}\right)$。

分类及回归问题

本章内容简介

- 分类与回归问题概述
- 分类问题的常见性能度量方法
- 典型的浅层机器学习分类方法
- 回归问题的常见性能度量方法
- 线性回归及基于梯度下降的线性回归模型参数求解方法
- 多项式回归
- 过拟合现象及损失函数的正则化处理

分类及回归问题

本章内容简介

- 分类与回归问题概述
- 分类问题的常见性能度量方法
- **典型的浅层机器学习分类方法**
  - 支持向量机
  - 朴素贝叶斯分类器
- 回归问题的常见性能度量方法
- 线性回归及基于梯度下降的线性回归模型参数求解方法
- 多项式回归
- 过拟合现象及损失函数的正则化处理
- 逻辑回归

分类及回归问题

本章内容简介

- 分类与回归问题概述
- 分类问题的常见性能度量方法
- **典型的浅层机器学习分类方法**
  - 支持向量机
  - **朴素贝叶斯分类器**
- 回归问题的常见性能度量方法
- 线性回归及基于梯度下降的线性回归模型参数求解方法
- 多项式回归
- 过拟合现象及损失函数的正则化处理
- 逻辑回归

朴素贝叶斯分类器

朴素贝叶斯分类器概述

- **根据贝叶斯（Naive Bayes）定理**：
  - 给定某样本 x，由特征向量$(x_1, \ldots, x_n)$描述，则其类别标签y的概率可以计算为：$P(y|x_1, \ldots, x_n) = \frac{P(y)P(x_1,\ldots,x_n|y)}{P(x_1,\ldots,x_n)}$。
  - 假设特征之间彼此独立，则上式中 $P(x_1, \ldots, x_n|y) = \prod_{i=1}^{n} P(x_i|y)$，从而有

$$P(y|x_1, \ldots, x_n) = \frac{P(y)\prod_{i=1}^{n} \boldsymbol{P(x_i|y)}}{P(x_1, \ldots, x_n)}$$

  - 对于给定输入样本 x，由于 $P(x_1, \ldots, x_n)$ 是不变的，所以有 $P(y|x_1, \ldots, x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y)$，因此求得

$$\hat{y} = arg\max_{y} P(y)\prod_{i=1}^{n} P(x_i|y) \qquad \textbf{(4.20)}$$

可以根据该公式获得样本的分类。

朴素贝叶斯分类器

朴素贝叶斯分类器概述

有如下三类常用NB模型：

1）**多项式（**Multinomial NB**）**：适合样本特征为多元离散值场景；

2）**伯努利**（Bernoulli NB）：适合样本特征为二元离散值或者很稀疏的多元离散值场景；

3）**高斯（**Gaussian NB**）**：适合样本特征分布大部分是连续值场景。

下面将依次介绍这三种贝叶斯分类器。

朴素贝叶斯分类器

多项式贝叶斯分类器

- 多项式贝叶斯分类器假设样本特征的条件概率分布满足简单的多项式分布[Manning, 2010]，其比较适合特征是离散的场景。
- 以文本分类为例，多项式朴素贝叶斯模型假设每个文档的特征向量$x = (x_1, \dots\dots, x_n)$中的每个$x_i$表示事件$i$在某个文档样本实例中被观察到的出现次数，即事件代表一个词在一个文档中的出现情况（基于BOW假设）。
- 因此，多项式贝叶斯分类器是在上述模型基础上加上贝叶斯假设。当特征为离散时，多项式模型会做平滑处理，则条件概率$P(x_i|y)$计算为：

$$\boldsymbol{P(x_i|y) = \frac{N_{yi}+\alpha}{N_y+\alpha n}} \tag{4.21}$$

  - 其中，N_{yi}是特征i在类y中出现的次数，N_y是类y中的特征总次数，n 是特征的数目（在文本分类中词汇表的大小）。
  - α为平滑因子，这是因为测试集中的特征可能没有在训练集样本中出现过，因此导致$P(x_i|y) = 0$，并影响分类结果。**当$\alpha = 1$时被称为拉普拉斯平滑（Laplace smoothing）**，$\alpha < 1$ 为利德斯通平滑（Lidstone smoothing）。

朴素贝叶斯分类器

多项式贝叶斯分类器

■ **利用多项式贝叶斯分类器进行文本分类举例：**

假设有如下的文档数据集，试利用该数据集训练贝叶斯分类器，并判断测试集的数据是否属于fruit类?

	文档ID	文档中的词	属于fruit类
训练集	1	apple banana	1
	2	apple apple	1
	3	apple orange	1
	4	cucumber cabbage Apple	-1
测试集	5	apple apple apple cucumber cabbage	?

朴素贝叶斯分类器

多项式贝叶斯分类器

求解过程：根据公式 $\hat{y} = arg\max_y P(y)\prod_{i=1}^{n} P(x_i|y)$，要分别求$P(y)$和$P(x_i|y)$。 **其中$P(y)$被称作先验概率，$P(x_i|y)$是条件概率。**

（一）先求$P(y)$：类yes下总共有3个文档，类no下有1个文档，训练样本文档总数为4，因此$P(y = fuit) = \frac{3}{4}$， $P(y = \widetilde{fuit}) = \frac{1}{4}$。

（二）再求条件概率$P(x_i|y)$： $P(x_i|y) = \frac{N_{yi}+\alpha}{N_y+\alpha n}$，即各类中每个词出现的概率。因此需要统计y和$x_i$的各种可能的组合。利用拉普拉斯平滑，分别计算如下：

- $P(apple|fuit) = (4+1)/(6+5) = 5/11$
- $P(cucumber|fuit) = P(cabbage|fruit) = (0+1)/(6+5) = 1/11$
- $P(apple|\widetilde{fuit}) = (1+1)/(3+5) = 2/8$
- $P(cucumber|\widetilde{fuit}) = P(cabbage|\widetilde{fuit}) = (1+1)/(3+5) = 2/8$

以上过程为训练阶段，得到的是相关的参数。

朴素贝叶斯分类器

多项式贝叶斯分类器

（三）测试阶段: 计算测试集中样本，即文档5属于每个类别的概率

- 由于测试集中的cucumber、cabbage在训练集的fruit类中没有出现，其条件概率就为0，会影响到测试集类别的估计，因此，在计算时采用拉普拉斯平滑，有
  - $P(fuit|Document\ 5) \propto P(fuit) \cdot P(apple|fuit)^3 \cdot P(cucumber|fuit) \cdot P(cabbage|fuit) = \frac{3}{4} \cdot \left(\frac{5}{11}\right)^3 \cdot \frac{1}{11} \cdot \frac{1}{11} = 0.000582$
  - $P(\widetilde{fuit}|Document\ 5) \propto P(\widetilde{fuit}) \cdot P(Apple|\widetilde{fuit})^3 \cdot P(cucumber|\widetilde{fuit}) \cdot P(cabbage|\widetilde{fuit}) = \frac{1}{4} \cdot \left(\frac{2}{8}\right)^3 \cdot \frac{2}{8} \cdot \frac{2}{8} = 0.000244$
- 前者更大，因此分类器将测试集中的文档5数据分为fruit类。

朴素贝叶斯分类器

伯努利贝叶斯分类器

- 伯努利分布又名两点分布或0-1分布，即每次试验只有两种可能结果。因此在伯努利贝叶斯分类器模型中每个特征取值为布尔型。其和多项式贝叶斯分类器类似，也适合处理离散数据。
- 文本分类中，主要利用特征是否在文档中出现的信息。
- **文本分类的训练阶段：**
  - **先验概率**$\mathbf{P(y)}=$ 类y中的文档总数/整个训练集中的文档总数
  - **条件概率**$\mathbf{P(x_i=1|y)}=$(类y下有单词x_i的文档数$+1$)/(类y下文档总数$+2$)
- **文本分类的测试阶段**：仍然根据公式 $\hat{y}=arg\max_y P(y)\prod_{i=1}^{n}P(x_i|y)$计算后验概率。但对于其中涉及的训练集中每个特征$x_i$：
  - 如果x_i在测试文档中出现，则$P(xi|y)=P(xi=1|y)$；
  - 如果x_i没有在测试文档中出现,那么$P(xi|y)=1-P(xi=1|y)$，即“没有某个特征”也是一个特征。

朴素贝叶斯分类器

伯努利贝叶斯分类器

- **先验概率**$\mathbf{P(y)}$ = 类y中的文档总数/整个训练集中的文档总数
- **条件概率**$\mathbf{P(x_i = 1|y)}$ = (类y下有单词x_i的文档数 + 1)/(类y下文档总数 + 2)

- **仍然以表4.2中所示的数据为例，使用伯努利贝叶斯分类器的训练及预测过程为：**
- **求解过程：根据公式** $\hat{y} = arg\max_y P(y)\prod_{i=1}^{n} P(x_i|y)$，**要分别求**$P(y)$**和**$P(x_i|y)$。
  其中$P(y)$ **被称作先验概率，**$P(x_i|y)$**是条件概率。**

  (一) 先求$\boldsymbol{P(y)}$：$P(y = fuit) = \frac{3}{4}$，$P(y = \widetilde{fuit}) = \frac{1}{4}$。

  (二) 再求条件概率$\boldsymbol{P(x_i|y)}$：
  - $\boldsymbol{P(Apple \mid yes) = (3+1)/(3+2) = 4/5}$
  - $\boldsymbol{P(Banana \mid yes) = P(Orange \mid yes) = (1+1)/(3+2) = 2/5}$
  - $\boldsymbol{P(Cucumber \mid yes) = P(Cabbage|yes) = (0+1)/(3+2) = 1/5}$
  - $P(Apple \mid no) = (1+1)/(1+2) = 2/3$
  - $P(Banana \mid no) = P(Orange \mid no) = (0+1)/(1+2) = 1/3$
  - $P(Cucumber \mid no) = P(Cabbage|no) = (1+1)/(1+2) = 2/3$
- **以上过程为训练阶段，得到的是相关的参数。**

朴素贝叶斯分类器

伯努利贝叶斯分类器

（三）测试阶段: 计算测试集中样本，即文档5属于每个类别的概率

- $P(yes \mid Document\ 5) = P(yes) \times P(apple|yes) \times P(cucumber|yes) \times P(cabbage|yes) \times (1 - P(banana|yes)) \times (1 - P(orange|yes)$

$$= \frac{3}{4} \times \frac{4}{5} \times \frac{1}{5} \times \frac{1}{5} \times \left(1 - \frac{2}{5}\right) \times \left(1 - \frac{2}{5}\right) = \mathbf{0.00864}$$

- $P(no \mid Document\ 5) = P(no) \times P(apple|no) \times P(cucumber|no) \times P(cabbage|no) \times (1 - P(banana|no)) \times (1 - P(orange|no))$

$$= \frac{1}{4} \times \frac{2}{3} \times \frac{2}{3} \times \frac{2}{3} \times (1 - \frac{1}{3}) \times (1 - \frac{1}{3}) = \mathbf{0.032922}$$

后者较大，因此，这个文档不属于类别fruit。

- 可见，虽然文档5中出现了3次apple一词，但**在伯努利贝叶斯分类器中，只关注是否出现，不关注频率。**

朴素贝叶斯分类器

高斯贝叶斯分类器

- 当特征取值为连续变量时，使用多项式贝叶斯模型会导致条件概率出现很多$P(x_i|y)=0$。此时即使进行平滑处理，得到的条件概率也难以描述真实情况。
- 所以当特征取值为连续变量时，应该采用高斯贝叶斯模型，此时假设条件概率服从高斯分布：$P(x_i|y)\sim \mathrm{N}(\mu_{y,i},\sigma_{y,i}^2)$，即

$$P(x_i|y)=\frac{1}{\sqrt{2\pi\sigma_{y,i}^2}}\exp\left(-\frac{(x_i-\mu_{y,i})^2}{2\sigma_{y,i}^2}\right) \tag{4.22}$$

其中，$\mu_{y,i}$和$\sigma_{y,i}^2$分别是第y类样本在第i个属性上取值的均值和方差，可以使用最大似然估计得到。

高斯贝叶斯分类器

高斯贝叶斯举例。对于鸢尾花数据集，假设某测试样本的特征向量为(5.9，3.0，5.1，1.8)。则如何使用高斯贝叶斯模型预测其类别？假设使用全部样本训练。

求解过程：根据公式 $\hat{y} = arg\max_y P(y)\prod_{i=1}^{n} P(x_i|y)$，要分别求$P(y)$和$P(x_i|y)$。其中$P(y)$ 被称作先验概率，$P(x_i|y)$是条件概率。

（一）先求$P(y)$：由于鸢尾花数据集共计3个类，每类50个样本，因此有$P(y=1)=\frac{50}{150}=\frac{1}{3}$；$P(y=2)=\frac{50}{150}=\frac{1}{3}$；$P(y=3)=\frac{50}{150}=\frac{1}{3}$

（二）再求条件概率$P(x_i|y)$：对于测试样本(5.9，3.0，5.1，1.8)，根据高斯贝叶斯的条件概率公式$P(x_i|y)=\frac{1}{\sqrt{2\pi\sigma_{y,i}^2}}\exp(-\frac{(x_i-\mu_{y,i})^2}{2\sigma_{y,i}^2})$，依次计算$x_i$与y的各种组合的条件概率，共计有4*3=12个。

以$P(x_1|y=1)$为例，有

$$P(x_1=5.9|y=1)=\frac{1}{\sqrt{2\pi\sigma_{y=1,x=x_1}^2}}\exp\left(-\frac{(x_1-\mu_{y=1,x=x_1})^2}{2\sigma_{y=1,x=x_1}^2}\right)$$

$$=\frac{1}{\sqrt{2\pi * 0.121764}}\exp\left(-\frac{(5.9-5.0006)^2}{2 * 0.121764}\right)$$

其中，$\mu_{y=1,x=x_1}$=5.0006，而$\sigma_{y=1,x=x_1}^2=0.121764$为根据鸢尾花数据集当类别标签y=1时的训练数据（如表4.3所示）统计得到。

高斯贝叶斯分类器

鸢尾花数据集类别标签y=1时的训练数据

样本ID	x_1	x_2	x_3	x_4	类别标签y
1	5.1	…	…	…	1
2	49	…	…	…	1
3	4.7	…	…	…	1
4	4.6	…	…	…	1
…	…	…	…	…	1
48	4.6	…	…	…	1
49	5.3	…	…	…	1
50	5	…	…	…	1

同理，可以利用表4.3计算

$$P(x_2 = 3.0|y = 1),\ P(x_3 = 5.1|y = 1),\ P(x_4 = 1.8|y = 1)$$

- **根据类别标签y=2时的训练数据计算**

$$P(x_1 = 5.9|y = 2),\ P(x_2 = 3.0|y = 2),\ P(x_3| = 5.1y = 2),\ P(x_4 = 1.8|y = 2)$$

- **根据类别标签y=3时的训练数据计算**

$$P(x_1 = 5.9|y = 3),\ P(x_2 = 3.0|y = 3),\ P(x_3| = 5.1y = 3),\ P(x_4 = 1.8|y = 3)$$

（三）得到以上条件概率后，可以根据公式 $\hat{y} = arg\max_y P(y)\prod_{i=1}^{n} P(x_i|y)$计算该样本属于每个类的概率，并预测其类别（具体过程略）

分类及回归问题

本章内容简介

- 分类与回归问题概述
- 分类问题的常见性能度量方法
- 典型的浅层机器学习分类方法
- 回归问题的常见性能度量方法
- 线性回归及基于梯度下降的线性回归模型参数求解方法
- 多项式回归
- 过拟合现象及损失函数的正则化处理
- 逻辑回归

分类及回归问题

本章内容简介

- **分类与回归问题概述**
- 分类问题的常见性能度量方法
- 典型的浅层机器学习分类方法
- 回归问题的常见性能度量方法
- 线性回归及基于梯度下降的线性回归模型参数求解方法
- 多项式回归
- 过拟合现象及损失函数的正则化处理
- 逻辑回归

回归问题性能度量方法

常用的评价回归问题的方法

- 平均绝对误差MAE(mean_absolute_error)
- 均方误差MSE (mean_squared_error)及均方根差RMSE
- Log loss，或称交叉熵loss(cross-entropy loss)
- R方值，确定系数(r2_score) (后文介绍)

平均绝对误差MAE

- **MAE (Mean absolute error)**是绝对误差损失 (absolute error loss）的期望值。
- 如果 $\hat{y}_i$ 是第i个样本的预测值，y_i 是相应的真实值，那么在$n_{samples}$个测试样本上的平均绝对误差（MAE）的定义如下：

$$MAE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} |y_i - \hat{y}_i|$$

均方差MSE

- **MSE(Mean squared error),** 该指标对应于平方误差损失 (squared error loss) 的期望值。
- 如果$\hat{y}_i$是第i个样本的预测值，y_i 是相应的真实值，那么在$n_{samples}$上的均方差的定义如下：

$$MSE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} |y_i - \hat{y}_i|^2$$

- **均方根差RMSE:** Root Mean Squared Error, RMSE,是MSE的平方根

均方差MSE的应用举例

- 假设模型预测某日下雨的概率为P:

预测概率	Ground truth（1表示下雨，0表示没下雨）	MSE	模型性能
1.0	1	$(1.0-1.0)^2=0$	完美
1.0	0	$(1.0-0)^2=1$	糟糕
0.7	1	$(0.7-1.0)^2=0.09$	误差较小
0.3	1	$(0.3-1.0)^2=0.49$	误差较大

讨论: 回归评价的ground truth如何获得？

- MAE, RMSE(MSE）常用于评分预测评价
  - 很多提供推荐服务的网站都有一个让用户给物品打分的功能。预测用户对物品评分的行为称为评分预测。

回归问题性能度量方法

R方

R方（R-square）可以用于评估回归模型对现实数据拟合的程度。R方取值越大，说明模型效果越好。对于样本i，如果$\hat{y}_i$是其预测值，y_i是真实值，$\bar{y}$ 是真实值的均值，则在n个测试样本上的R方计算步骤为：

1）计算残差平方和：$SS_{res} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ ；

2）计算总离差平方和：$SS_{tss} = \sum_{i=1}^{n}(y_i - \bar{y})^2$；

3）得到R方：$R^2 = 1 - \frac{SS_{res}}{SS_{tss}}$。

表4.4 若干披萨样本的真实价格和预测价格（测试集）

样本序号	真实值y_i	预测值$\hat{y}_i$
1	11	9.775
2	8.5	10.75
3	15	12.70
4	18	17.58
5	11	13.68

R方的计算举例。对于如表4.4所示的若干披萨样本的真实价格和预测价格测试集。则R方的计算过程如下：

1). 计算残差平方和： $SS_{res} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

$= (11 - 9.775)^2 + (8.5 - 10.75)^2 + \ldots + (11 - 13.68)^2 = 19.19$

2). 计算样本总离差平方和： $SS_{tss} = \sum_{i=1}^{n}(y_i - \bar{y})^2$

$= (11 - 12.7)^2 + (8.5 - 12.7)^2 + \ldots + (11 - 12.7)^2 = 56.8$

3). 最后得到R方： $R^2 = 1 - \frac{SS_{res}}{SS_{tss}} = 1 - \frac{19.19}{56.8} = 0.66$

R方是0.66说明测试集中过半数的价格都可以通过模型解释。

线性回归

什么是线性回归

- 狭义线性（linear）模型：
  - 通常指自变量与因变量之间按比例、成直线的关系，在数学上可理解为一阶导数为常数的函数，如$y = \theta^T x$；
  - 线性通常表现为一次曲线。
- 广义线性（generalized linear model， GLM ）：
  - 是线性模型的扩展，主要通过联结函数$g()$(link function)，使预测值落在响应变量的变幅内。例如逻辑回归

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$$ （括号内为线性函数）

非线性模型

- 非线性non-linear模型：
  - 非线性一般指不按比例、不成直线的关系，一阶导数不为常数
- 常见的非线性模型
  - 2次以上的多项式 $y = a_n x^n + a_{n-1} x^{n-1} + \ldots, + a_1 x + a_0$
  - 幂函数模型 $y = \beta_0 {x_1}^{\beta_1} {x_2}^{\beta_2} \cdots {x_k}^{\beta_k}$
  - 指数函数模型 $y = \beta_0 e^{\beta_1 x}$
  - 对数函数模型： $y = \beta_0 + \beta_1 ln x$
  - 等等

线性回归

- 线性回归模型中，假设自变量和因变量满足如下形式：

  $$y = h_\theta(x) = \theta^T x$$

- 问题：已知一些数据，如何求里面的未知参数，给出一个最优解。
- 因此通常将参数求解问题转化为求最小误差问题。 一般采用模型预测结果与真实结果的差的平方和作为损失函数：

  $$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta\left(x^{(i)}\right) - y^{(i)})^2.$$

  即求$\theta = \min_\theta(J(\theta))$

概率解释

- 设预测结果$\theta^T x^{(i)}$与真实结果$y^{(i)}$之间误差为$\epsilon^{(i)}$，即$y^{(i)} = \theta^T\ x^{(i)}+\epsilon^{(i)}$
- 通常误差满足平均值为0的高斯分布，即正态分布。那么在一个样本i上x和y的概率密度公式为$p\left(y^{(i)}\middle|x^{(i)};\theta\right) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(y^{(i)}-\theta^T x^{(i)})^2}{2\sigma^2})$
- 模型在全部样本上预测的最大似然估计为

$$L(\theta) = \prod_{i=1}^{m}\frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(y^{(i)}-\theta^T\ x^{(i)})^2}{2\sigma^2})$$

$$l(\theta) = lnL(\theta) = -mln\left(\sqrt{2\pi}\sigma\right) - \sum_{i=1}^{m}\frac{(y^{(i)}-\theta^T\ x^{(i)})^2}{2\sigma^2}$$

从而，需要$\frac{1}{2}\sum_{i=1}^{m}(y^{(i)}-\theta^T\ x^{(i)})^2$最小

求解参数

- 接下来，就是求解使得$\frac{1}{2}\sum_{i=1}^{m}(y^{(i)}-\theta^T x^{(i)})^2$最小的参数$\theta$。
- 解法有：
  - 矩阵解法。scikit-learn中的LinearRegression类使用的是矩阵解法（有时也称为最小二乘法）。可以解出线性回归系数θ。
  - 梯度下降法。 梯度下降（Gradient descent ）是利用一阶的梯度信息找到函数局部最优解的一种方法。

参数的矩阵解法

例如，设$Y_i = \widehat{\beta_0} + \widehat{\beta_1} X_i + \varepsilon_i$，*即为线性关系*$\Rightarrow \varepsilon_i = Y_i - \widehat{\beta_0} - \widehat{\beta_1} X_i$

$$\boldsymbol{Q = \sum_{i=1}^{m} \varepsilon_i^2 = \sum_{i=1}^{m} (Y_i - \widehat{Y_i})^2 = \sum_{i=1}^{m} (Y_i - \widehat{\beta_0} - \widehat{\beta_1} X_i)^2}$$

通过使Q 最小，即可确定$\widehat{\beta_0}$，$\widehat{\beta_1}$。

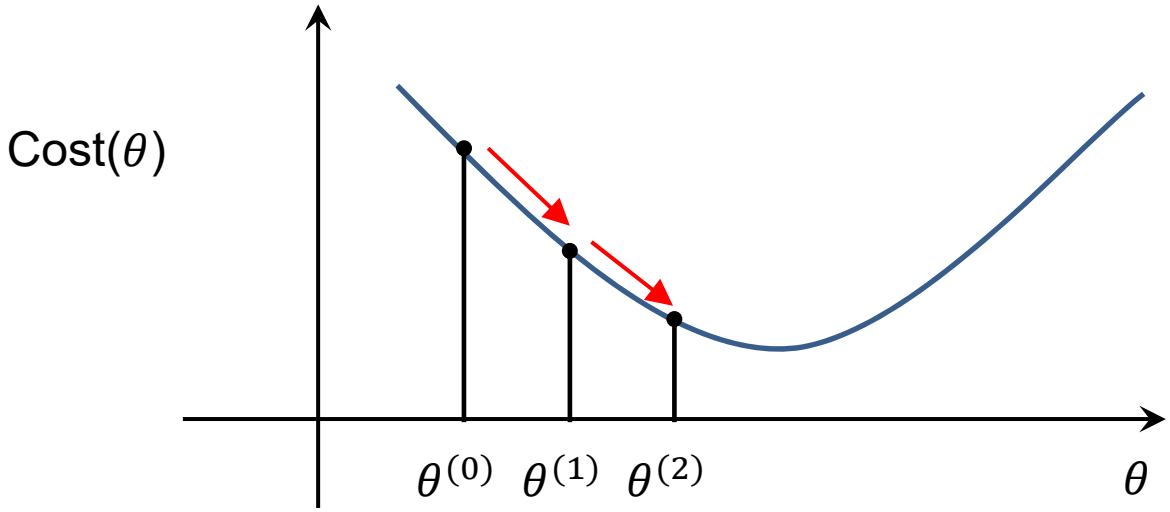
根据数学知识我们知道，函数的极值点为偏导为0的点，即

$$\begin{cases} \frac{\partial Q}{\partial \widehat{\beta_0}} = 2\sum_{i=1}^{m}\left(Y_i - \widehat{\beta_0} - \widehat{\beta_1} X_i\right)(-1) = 0 \\ \frac{\partial Q}{\partial \widehat{\beta_1}} = 2\sum_{i=1}^{m}\left(Y_i - \widehat{\beta_0} - \widehat{\beta_1} X_i\right)(-X_i) = 0 \end{cases}$$

$$\Rightarrow \beta_0 = \frac{n\sum X_i Y_i - \sum X_i \sum Y_i}{n\sum {X_i}^2 - (\sum X_i)^2}$$

$$\beta_1 = \frac{\sum {X_i}^2 \sum Y_i - \sum X_i \sum X_i Y_i}{n\sum {X_i}^2 - (\sum X_i)^2}$$

参数的梯度下降求解法



- 梯度下降（Gradient descent ）是利用一阶的梯度信息找到函数局部最优解的一种方法，也是机器学习里面常用的一种优化方法。
- 其基本思想是，要找代价函数最小值，只需要每一步都往下走，也就是每一步都可以让误差损失函数小一点。
- 对于线性回归，参数的更新方法一般为：

$$\theta_j' = \theta_j - L\frac{\partial J(\theta)}{\partial \theta_j} = \theta_j - \frac{1}{m}L\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

- 如何求梯度？→

线性回归的梯度下降

（接上文）$\theta_j' = \theta_j - L\frac{\partial J(\theta)}{\partial \theta_j} = \theta_j - \frac{1}{m}L\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$，是如何得出的？

- 对于某个实例 i：

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{\partial}{\partial \theta_j}\cdot\frac{1}{2}\cdot(h_\theta(x^{(i)}) - y^{(i)})^2$$

$$= 2\cdot\frac{1}{2}\cdot\left(h_\theta(x^{(i)}) - y^{(i)}\right)\cdot\frac{\partial}{\partial \theta_j}\left(h_\theta(x^{(i)}) - y^{(i)}\right)$$

$$= \left(h_\theta(x^{(i)}) - y^{(i)}\right)\cdot\frac{\partial}{\partial \theta_j}\left(\sum_{k=1}^{n}\theta_k x_k^{(i)} - y^{(i)}\right)$$

$$= \left(h_\theta(x^{(i)}) - y^{(i)}\right)x_j^{(i)}$$

- 先初始化一组θ, 在这个θ值之上，用梯度下降法去求出下一组θ的值。当迭代到一定程度，$J(\theta)$的值趋于稳定，此时的θ即为要求得的值。

分类及回归问题

本章内容简介

- 分类与回归问题概述
- 分类问题的常见性能度量方法
- 典型的浅层机器学习分类方法
- 回归问题的常见性能度量方法
- 线性回归及基于梯度下降的线性回归模型参数求解方法
- 多项式回归
- 过拟合现象及损失函数的正则化处理
- 逻辑回归

分类及回归问题

本章内容简介

- 分类与回归问题概述
- 分类问题的常见性能度量方法
- 典型的浅层机器学习分类方法
- 回归问题的常见性能度量方法
- 线性回归及基于梯度下降的线性回归模型参数求解方法
- **多项式回归**
- 过拟合现象及损失函数的正则化处理
- 逻辑回归

多项式回归

什么是线性回归

- 在上一节介绍的线性回归模型中，我们假设因变量与自变量之间的关系是按比例的线性关系，但真实情况也许未必如此。如果因变量与自变量之间的关系未知，也可以用适当幂次的多项式来近似反映。
- 多项式是由常数与自变量 x经过有限次乘法与加法运算得到。例如 $p_n(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots, + a_1 x + a_0$ 就是多项式函数。
- 多项式回归通常的做法为首先生成多项式特征以体现特征之间的相互影响，然后在新的特征空间上将其视为线性回归问题处理。因此其基本步骤为：

  1. 生成多项式特征：例如假设样本x有 2 个特征x_1，x_2，则二阶多项式的特征集为$[1, x_1, x_2, x_1^2, x_2^2, x_1 x_2]$。
  2. 拟合得到模型：利用得到的多项式特征，使用线性器回归等模型进行拟合和预测处理。

多项式回归

（一）二次回归（Quadratic Regression）

表4.5 披萨的直径和价格之间关系的数据集（训练集和测试集）

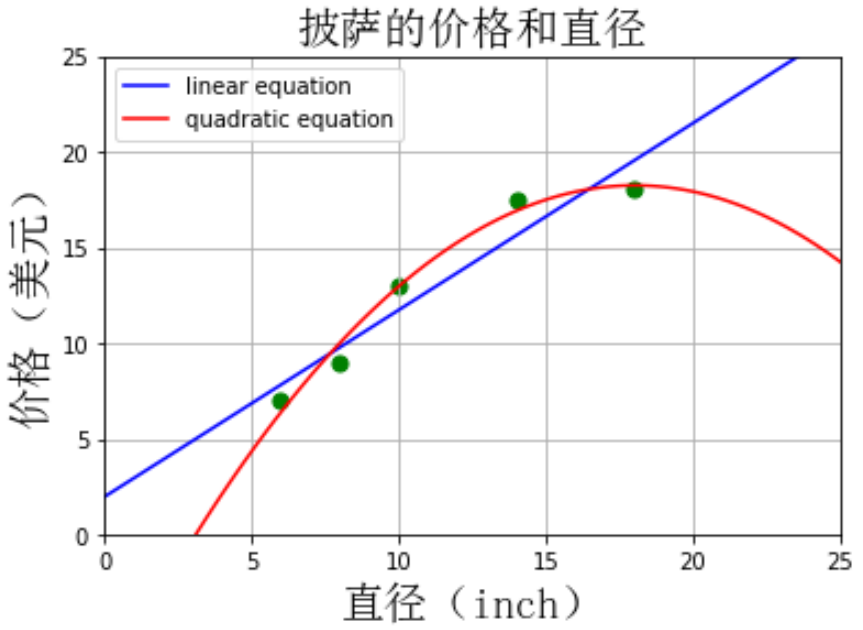
	样本序号	x	y
训练集	1	6	7
	2	8	9
	3	10	13
	4	14	17.5
	5	18	18
测试集	6	6	8
	7	8	12
	8	11	15
	9	16	18

- 所谓二次回归，即最高次为2次的多项式回归。以披萨的直径和价格之间的关系的回归为例。此时样本只有一个特征，即披萨的直径。此时的多项式回归模型为二次曲线$y=\beta_0+\beta_1 x+\beta_2 x^2$。
- 假设训练集和测试集如表4.5所示。
- 则使用表4.5所示的训练集采用线性回归模型拟合出的为直线，采用二次回归模型拟合出的为一条曲线，如图4.9所示。

多项式回归

（一）二次回归（Quadratic Regression）

下面分别从两个角度关注线性回归和二项式回归两种模型的性能对比：

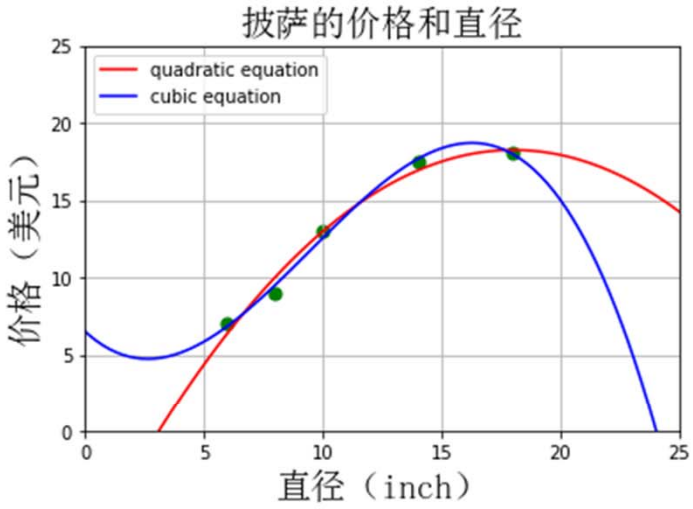


线性回归和二项式回归曲线（图中曲线为使用表4.5所示训练集获得）

- 首先是两种模型对训练集样本的拟合效果对比。从图4.9可见，训练集中的5个样本点与二项式回归曲线的接近程度比线性回归直线要好，即采用二项式回归对训练集样本的总体拟合效果比线性回归的总体拟合效果好。
- 其次是两种模型在测试集上的性能对比。这个是所有机器学习模型的最终目的，即获得一个泛化的模型，设法使其在未知标签的测试集上的性能最好。基于表4.5中所示的数据集，经编程或者计算可以得到测试集上的评价结果为：线性回归的R方为0.81，二次回归R方为0.87。由于R方取值越大说明其拟合效果更佳。因此，在表4.5所示的数据集例子中，采用二次多项式回归比单纯的线性回归效果更好。

多项式回归

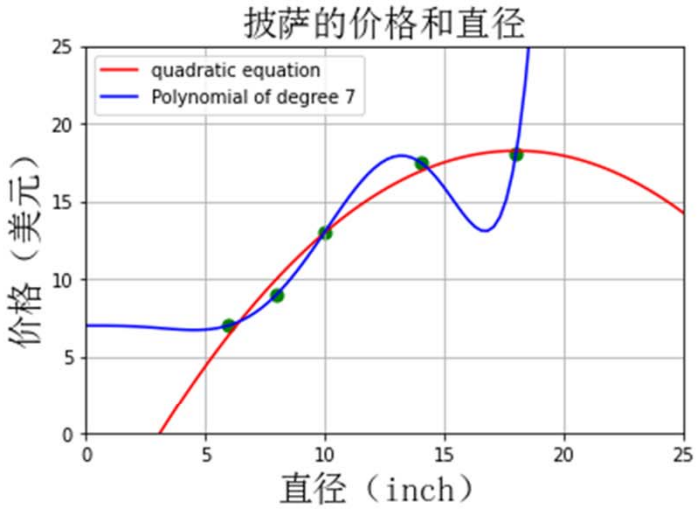
（二）更高次的多项式回归



注：测试集上的性能对比：

- 二次回归的 R 方为 0.87
- **三次回归的 R 方为 0.84**

（a）二次回归与三次回归的比较



注：测试集上的性能对比：

- 二次回归的 R 方为 0.87
- **七次回归的 R 方为 0.49**

（b）二次回归与七次回归的比较

分类及回归问题

本章内容简介

- 分类与回归问题概述
- 分类问题的常见性能度量方法
- 典型的浅层机器学习分类方法
- 回归问题的常见性能度量方法
- 线性回归及基于梯度下降的线性回归模型参数求解方法
- 多项式回归
- 过拟合现象及损失函数的正则化处理
- 逻辑回归

分类及回归问题

本章内容简介

- 分类与回归问题概述
- 分类问题的常见性能度量方法
- 典型的浅层机器学习分类方法
- 回归问题的常见性能度量方法
- 线性回归及基于梯度下降的线性回归模型参数求解方法
- 多项式回归
- **过拟合现象及损失函数的正则化处理**
- 逻辑回归

过拟合与损失函数的正则化

什么是线性回归

向量范数，假设向量有N个元素。[…， …， …]

- L1-范数：即向量元素绝对值之和。

$$\|x\|_1 = \sum_{i=1}^{N}|x_i|$$

- L2-范数：Euclid范数（欧几里得范数，常用计算向量长度），即向量元素绝对值的平方和再开方。

$$\|x\|_2 = \left(\sum_{i=1}^{N}|x_i|^2\right)^{\frac{1}{2}}$$

- ∞-范数：即所有向量元素绝对值中的最大值。

$$\|x\|_\infty = \max_i |x_i|$$

- -∞-范数：即所有向量元素绝对值中的最小值。

$$\|x\|_{-\infty} = \min_i |x_i|$$

- p-范数：即向量元素绝对值的p次方和的1/p次幂。

$$\|x\|_p = \left(\sum_{i=1}^{N}|x_i|^p\right)^{\frac{1}{p}}$$

矩阵范数

假设矩阵A为m*n，即m行，n列。$\begin{bmatrix} & \cdots & \\ \vdots & \ddots & \vdots \\ & \cdots & \end{bmatrix}$

- L1-范数：列和范数，即矩阵的所有列向量元素绝对值之和的最大值。

  $\|A\|_1 = \max_j \sum_{i=1}^{m} |a_{ij}|$

- L2-范数：谱范数，即A^TA矩阵的最大特征值的开平方。

  $\|A\|_2 = \sqrt{\lambda_1}$，$\lambda_1$为$A^TA$的最大特征值

- ∞-范数：行和范数，即矩阵的所有行向量元素绝对值之和的最大值。

  $\|A\|_\infty = \max_i \sum_{j=1}^{n} |a_{ij}|$

- F-范数：Frobenius范数，即矩阵元素绝对值的平方和再开平方。

  $\|A\|_F = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2\right)^{\frac{1}{2}}$

线性回归的正则化

- 应对过拟合(Overfitting)。因为在某些情况下，学习得到的模型在训练集上也许误差较小。但是对于测试集中之前未见样本的预测却未必有效。为此可以在损失函数中加入正则化项。以线性回归为例，

$$J(\theta)=\frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2+\alpha\sum_{j=1}^{n}\theta_j^2\right]$$

正则化项

其中α是正则化参数（regularization parameter），用于控制两个不同的目标的平衡。

I. 第一个目标是使假设更好地拟合训练数据。

II. 第二个目标是要正则化处理，使得模型不要太复杂。

线性回归正则化后的梯度更新方法

- 新的损失函数$J(\theta)=\frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^i)-y^i)^2+\boldsymbol{\alpha}\sum_{k=1}^{n}\boldsymbol{\theta_k}^2\right]$
- 新的梯度更新公式:

$$\theta_j'=\theta_j-L\frac{\partial J(\theta)}{\partial\theta_j}$$

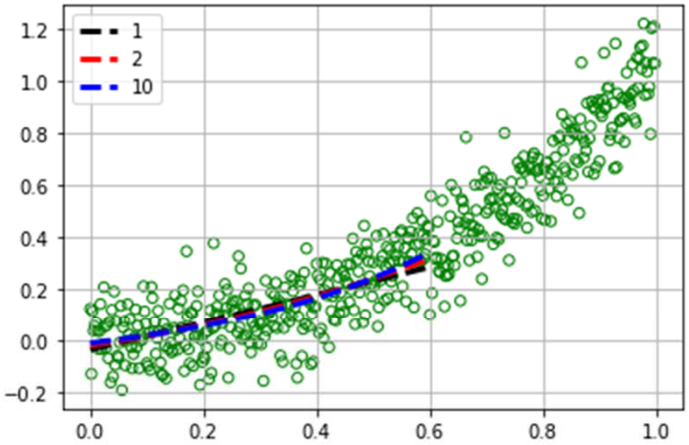
$$=\theta_j-L\cdot\left[\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^i)-y^i)x_j^i-\frac{\boldsymbol{\alpha}}{2m}\cdot 2\cdot\boldsymbol{\theta_j}\right]$$

$$=\theta_j-L\cdot\left[\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^i)-y^i)x_j^i-\frac{\boldsymbol{\alpha}}{2m}\cdot 2\cdot\boldsymbol{\theta_j}\right]$$

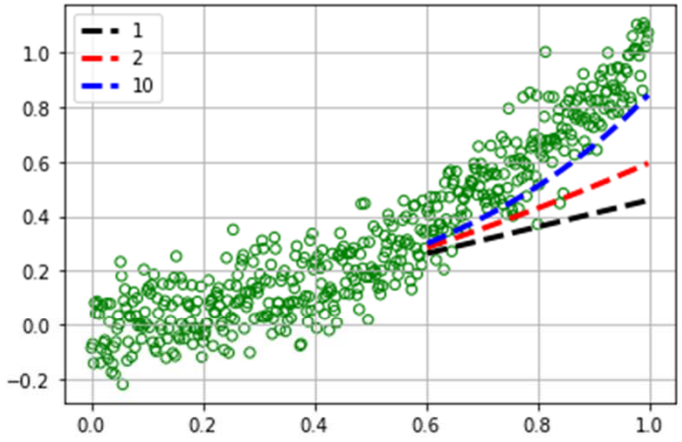
$$=\theta_j(1-L\frac{\alpha}{m})-L\cdot\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^i)-y^i)x_j^i$$

过拟合现象及损失函数的正则化处理

岭回归克服过拟合现象的示例



(a) 使用岭回归拟合后的模型在前 300 个横坐标值（训练集）上绘制的曲线



（b）使用岭回归拟合后的模型在后 200 个横坐标值（测试集）上绘制的曲线

下面给出一个岭回归克服过拟合现象的具体示例。

数据集。如图4.11所示，数据集为二维平面上的500个数据点。每个点的横坐标x为从0到1范围内步长为0.002等距离划分形成的500个数值，纵坐标y为其横坐标x的平方加上一个随机数，即$y = x^2 + r$。其中$r \sim \mathcal{N}(0,\ 0.01)$，即随机数服从期望为0，标准差为0.1的正态分布。因此本例中的数据集是用一个二次函数加上随机扰动形成的500个数据样本。将前300个点作为训练集，将后200个点作为测试集。

回归模型。逐渐增加多项式回归的次数，例如本例中设置多项式回归的次数为1（线性回归）、2和10。图4.11中的（a）图和（b）图中的曲线为使用岭回归拟合后的模型在前300个横坐标值（训练集）和后200个横坐标值（测试集）上绘制的曲线。

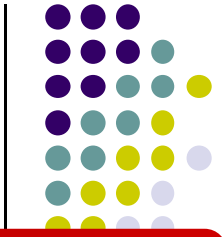
过拟合现象及损失函数的正则化处理

岭回归克服过拟合现象的示例

表4.6 有无正则化的回归性能对比

		(a) 没有正则化处理		(b) 岭回归（即L2正则化）	
评价数据集	回归模型	RMSE	R方	RMSE	R方
训练集	线性	0.11	0.45	0.11	0.43
	二次回归	0.10	0.52	0.11	0.47
	十次回归	0.10	0.53	0.10	0.49
测试集	线性	0.28	-0.68	0.30	-1.08
	二次回归	0.17	0.37	0.20	0.09
	十次回归	5431.18	$-6.14*10^8$	0.10	0.76

分类及回归问题



本章内容简介

- 分类与回归问题概述
- 分类问题的常见性能度量方法
- 典型的浅层机器学习分类方法
- 回归问题的常见性能度量方法
- 线性回归及基于梯度下降的线性回归模型参数求解方法
- 多项式回归
- 过拟合现象及损失函数的正则化处理
- 逻辑回归

分类及回归问题

本章内容简介

- 分类与回归问题概述
- 分类问题的常见性能度量方法
- 典型的浅层机器学习分类方法
- 回归问题的常见性能度量方法
- 线性回归及基于梯度下降的线性回归模型参数求解方法
- 多项式回归
- 过拟合现象及损失函数的正则化处理
- **逻辑回归**

逻辑回归

逻辑回归的定义

线性回归模型的模型如下：$h_\theta(x)=\theta^T x$

逻辑回归的模型定义（需要借助Sigmoid函数）：$g(x)=\dfrac{1}{1+\exp^{-x}}$

将上述线性回归的模型带入到g(x)中，得到最终的逻辑回归的模型：

$$h_\theta(x)=g(\theta^T x)=\frac{1}{1+\exp^{(-\theta^T x)}}$$

假设该表达式为等于类 1 的概率，则类 0 的概率等于1减去等于类 1 的概率：

$$\begin{cases}P(c=1\mid x;\theta)=h_\theta(x)\\P(c=0\mid x;\theta)=1-h_\theta(x)\end{cases}$$

将上面两个式子整合为下面一个公式：

$$P(c=y\mid x;\theta)=(h_\theta(x))^y(1-h_\theta(x))^{1-y}$$

逻辑回归

逻辑回归的损失函数

那么似然函数为 $$L(\theta)=\prod_{i=1}^{m}(h_\theta(x))^{y}(1-h_\theta(x))^{1-y}$$

m表示样本个数，为了方便计算，取对数得

$$\log(L(\theta))=\sum_{i=1}^{m}[y^{(i)}\log h_\theta(x^{(i)})+(1-y^{(i)})\log(1-h_\theta(x^{(i)}))]$$

求上式的极大值，引入因子 -1/m，转化为求下式的极小值：

$$J(\theta)=-\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log h_\theta(x^{(i)})+(1-y^{(i)})\log(1-h_\theta(x^{(i)}))]$$

这就是逻辑回归的log损失函数，其中 $$h_\theta(x)=g(\theta^T x)=\frac{1}{1+\exp^{(-\theta^T x)}}$$

逻辑回归

逻辑回归的梯度下降更新方法

推导用到了sigmoid函数的导数公式，即：$g(x)=\frac{1}{1+e^{-x}}$

$$g'(x)=g(x)(1-g(x))$$

$$\theta_j=\theta_j-\alpha\cdot\frac{\partial}{\partial}J(\theta),(j=0,1,\cdots,n)\cdots\cdots\cdots(1)$$

推导偏导数：

$$\frac{\partial J(\theta)}{\partial\theta_J}=-\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\frac{1}{h_\theta(x^{(i)})}\frac{\partial h_\theta(x^i)}{\partial\theta_j}-(1-y^{(i)})\frac{1}{1-h_\theta(x^{(i)})}\frac{\partial h_\theta(x^i)}{\partial\theta_j}\right)\cdots\cdots\cdots\cdots\cdots(1)$$

$$=-\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\frac{1}{g(\theta^Tx^{(i)})}-(1-y^{(i)})\frac{1}{1-g(\theta^Tx^{(i)})}\right)\cdot\frac{\partial g(\theta^Tx^{(i)})}{\partial\theta_j}\cdots\cdots\cdots\cdots\cdots(2)$$

$$=-\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\frac{1}{g(\theta^Tx^{(i)})}-(1-y^{(i)})\frac{1}{1-g(\theta^Tx^{(i)})}\right)\cdot g(\theta^Tx^{(i)})(1-g(\theta^Tx^{(i)})x_j^{(i)}\cdots(3)$$

$$=-\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}(1-g(\theta^Tx^{(i)})-(1-y^{(i)})g(\theta^Tx^{(i)})\right)\cdot x_j^{(i)}\cdots\cdots\cdots\cdots\cdots\cdots\cdots(4)$$

$$=-\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}-g(\theta^Tx^{(i)})\right)\cdot x_j^{(i)}\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots(5)$$

$$=\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)})-y^{(i)}\right)\cdot x_j^{(i)}\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots(6)$$

分类及回归问题

本章小结

- 分类与回归问题概述
- 分类问题的常见性能度量方法
- 典型的浅层机器学习分类方法
- 回归问题的常见性能度量方法
- 线性回归及基于梯度下降的线性回归模型参数求解方法
- 多项式回归
- 过拟合现象及损失函数的正则化处理
- 逻辑回归