周志华 著

MACHINE LEARNING

机器学习

清华大学出版社

崔磊

QQ：362626744
E-Mail: leicui@nwu.edu.cn
办公室: 信息学院院楼912

本章课件致谢：李绍园 刘冲

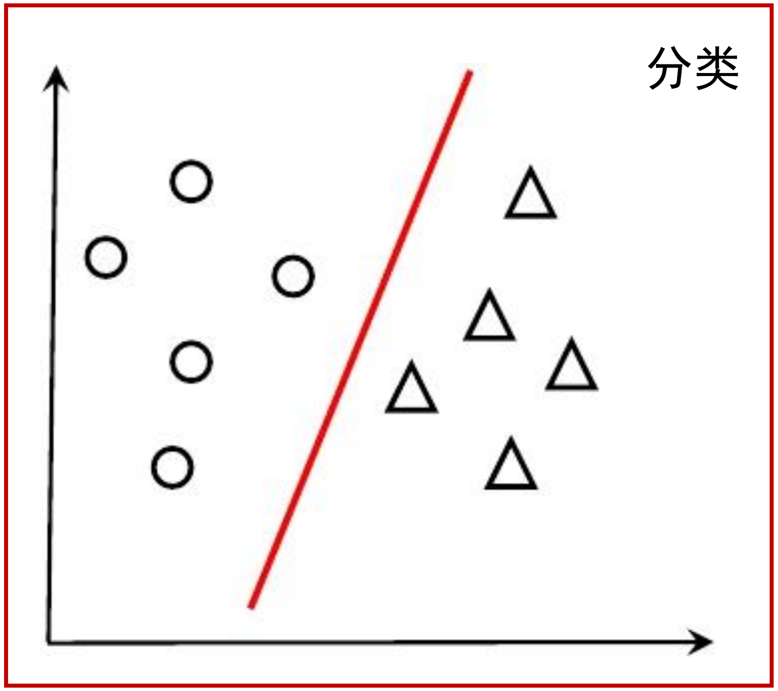
第三章：线性模型

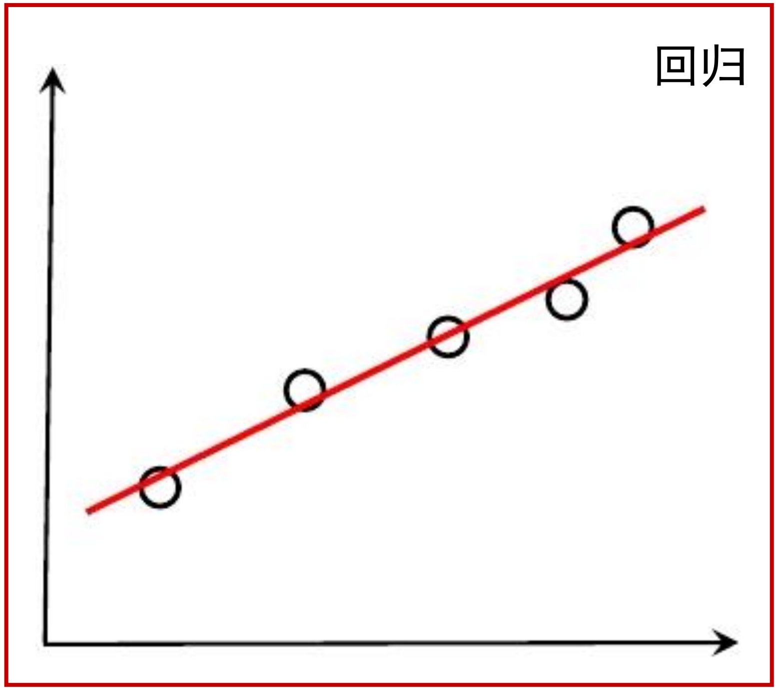
目录

目录

- 基本形式
- 线性回归
- 对数几率回归
- 线性判别分析
- 多分类学习
- 类别不平衡

基本形式





线性模型(linear model)试图学得一个通过属性的线性组合来进行预测的函数

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

$\boldsymbol{x} = (x_1; x_2; \ldots; x_d)$ 是由属性描述的示例，其中x_i是$\boldsymbol{x}$在第 i 个属性上的取值

向量形式：$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x} + b$

线性模型优点

- 形式简单、易于建模
- 可解释性
- 是非线性模型的基础

  可以在线性模型的基础上通过引入层级结构或高维映射而得

- 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

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

目录

线性回归—单元线性回归

线性回归试图学得一个线性模型以尽可能准确地预测实际输出标记

$f(x) = wx_i + b$ 使得 $f(x_i) \simeq y_i$

令均方误差最小化，有

$$
\begin{aligned}
(w^*, b^*) &= \underset{(w,b)}{\arg\min} \sum_{i=1}^{m} \left(f(x_i) - y_i\right)^2 \\
&= \underset{(w,b)}{\arg\min} \sum_{i=1}^{m} \left(y_i - wx_i - b\right)^2
\end{aligned}
$$

对 $E_{(w,b)} = \sum_{i=1}^{m} \left(y_i - wx_i - b\right)^2$ 进行最小二乘参数估计

线性回归—单元线性回归

分别对 w 和 b 求偏导：

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

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

令导数为 0, 得闭式解：

$$w = \frac{\sum\limits_{i=1}^{m} y_i\,(x_i - \bar{x})}{\sum\limits_{i=1}^{m} x_i^2 - \frac{1}{m}\left(\sum\limits_{i=1}^{m} x_i\right)^2} \qquad b = \frac{1}{m}\sum_{i=1}^{m}(y_i - wx_i)$$

线性回归—多元线性回归

- 给定数据集

$$D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\ldots,(\boldsymbol{x}_m,y_m)\}$$

$$\boldsymbol{x}_i=(x_{i1};x_{i2};\ldots;x_{id})\quad y_i\in\mathbb{R}$$

- 多元线性回归目标

$$f(\boldsymbol{x}_i)=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b \text{ 使得} f(\boldsymbol{x}_i)\simeq y_i$$

线性回归—多元线性回归

- 为了方便矩阵运算，把 $\boldsymbol{w}$ 和 b 合并成一个列向量，则数据可以表示为如下

$$\hat{\omega} = (\omega; b) = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \dots \\ \omega_d \\ b \end{pmatrix}$$

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} & 1 \\ x_{21} & x_{22} & \cdots & x_{2d} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{md} & 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1^{\mathrm{T}} & 1 \\ \boldsymbol{x}_2^{\mathrm{T}} & 1 \\ \vdots & \vdots \\ \boldsymbol{x}_m^{\mathrm{T}} & 1 \end{pmatrix}$$

线性回归—多元线性回归

$$X*\hat{\omega}=\begin{pmatrix} x_{11} & x_{12} & \dots & x_{1d} & 1 \\ x_{21} & x_{22} & \dots & x_{2d} & 1 \\ \dots & \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{md} & 1 \end{pmatrix} * \begin{pmatrix} \omega_1 \\ \omega_2 \\ \dots \\ \omega_d \\ b \end{pmatrix} = \begin{pmatrix} \omega_1 x_{11}+\omega_2 x_{12}+\dots\omega_d x_{1d}+b \\ \omega_1 x_{21}+\omega_2 x_{22}+\dots\omega_d x_{2d}+b \\ \dots \\ \omega_1 x_{m1}+\omega_2 x_{m2}+\dots\omega_d x_{md}+b \end{pmatrix} = \begin{pmatrix} f(\mathrm{x}_1) \\ f(\mathrm{x}_2) \\ \dots \\ f(\mathrm{x}_m) \end{pmatrix}$$

$$\boldsymbol{y}=(y_1;y_2;\dots;y_m)$$

线性回归—多元线性回归

同样采用最小二乘法求解，有

$$\hat{\boldsymbol{w}}^* = \underset{\hat{\boldsymbol{w}}}{\arg\min} \left(\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}}\right)^{\mathrm{T}} \left(\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}}\right)$$

令 $E_{\hat{\boldsymbol{w}}} = \left(\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}}\right)^{\mathrm{T}} \left(\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}}\right)$ ，对 $\hat{\boldsymbol{w}}$ 求导:

$$\frac{\partial E_{\hat{\boldsymbol{w}}}}{\partial \hat{\boldsymbol{w}}} = 2\mathbf{X}^{\mathrm{T}} \left(\mathbf{X}\hat{\boldsymbol{w}} - \boldsymbol{y}\right)$$

令其为零可得 $\hat{\boldsymbol{w}}$

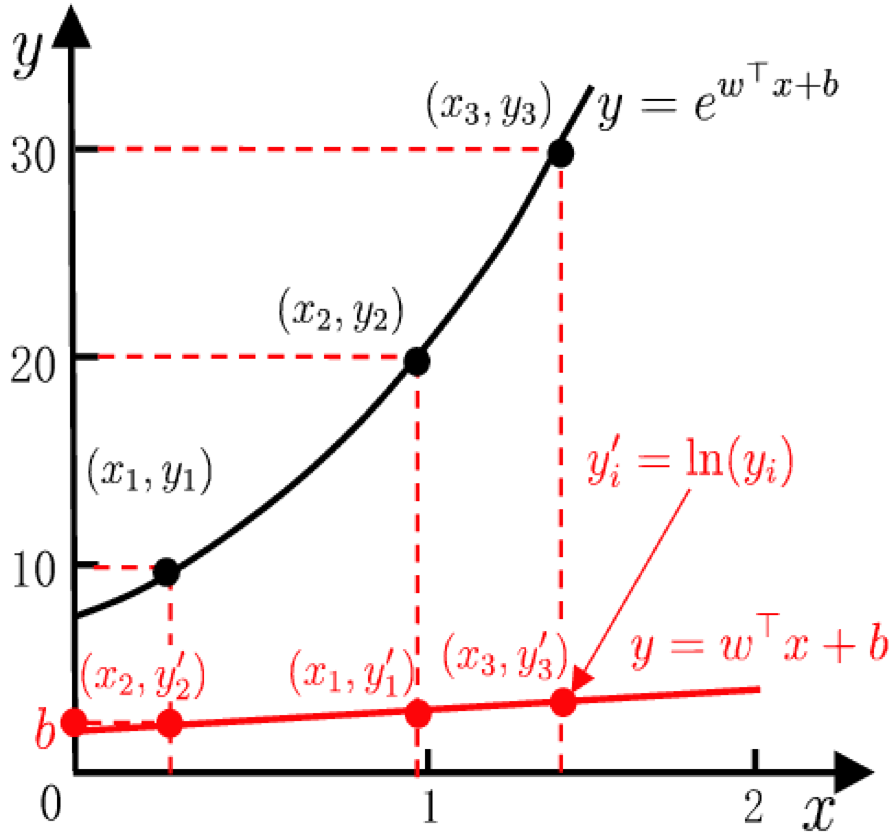
然而，麻烦来了：涉及矩阵求逆!

- 若 $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ 满秩或正定，则 $\hat{\boldsymbol{w}}^* = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1} \mathbf{X}^{\mathrm{T}}\boldsymbol{y}$
- 若 $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ 不满秩，则可解出多个 $\hat{\boldsymbol{w}}$

此时需求助于归纳偏好，或引入 正则化 (regularization)

对数线性回归

- 输出标记的对数为线性模型逼近的目标



$$\ln y = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$$

$$y = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$$

线性回归 - 广义线性模型

- 更一般形式

$$y = g^{-1}\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b\right)$$

- $g(\cdot)$ 称为联系函数（link function）

- 对数线性回归是 $g(\cdot) = \ln(\cdot)$ 时广义线性模型的特例

目录

- 基本形式
- 线性回归
- 对数几率回归
- 线性判别分析
- 多分类学习
- 类别不平衡

二分类任务

- 预测值与输出标记

$$z = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b \qquad y \in \{0, 1\}$$

- 寻找函数将分类标记与线性回归模型输出联系起来

找 z 和 y 的
联系函数

- 最理想的函数——单位阶跃函数

$$y = \begin{cases} 0, & z < 0; \\ 0.5, & z = 0; \\ 1, & z > 0, \end{cases}$$

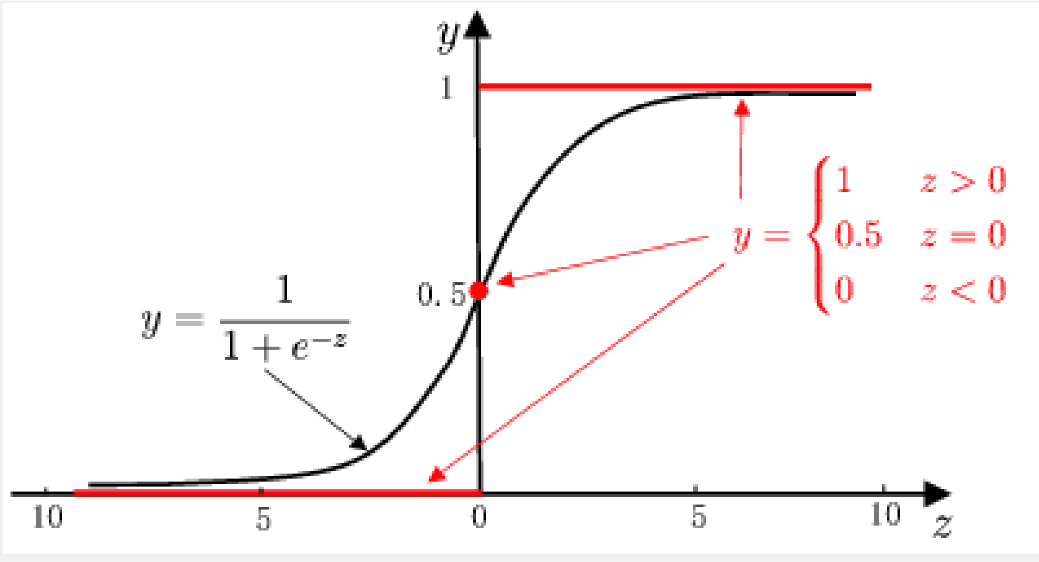
  - 预测值大于零就判为正例，小于零就判为反例，预测值为临界值零则可任意判别

二分类任务

- 单位阶跃函数缺点
  - 不连续
- 替代函数——对数几率函数（logistic function）
  - 单调可微、任意阶可导

$$y = \frac{1}{1+e^{-z}}$$

单位阶跃函数与对数几率函数的比较



对数几率回归

以对数几率函数为联系函数:

$$y = \frac{1}{1+e^{-z}} \quad \text{变为} \quad y = \frac{1}{1+e^{-(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b)}}$$

即: $\ln \frac{y}{1-y} = \boldsymbol{\omega}^T \boldsymbol{x} + \boldsymbol{b}$ 　　y为样本x为正例的可能性

“对数几率”
(log odds, 亦称 logit)

$\frac{y}{1-y}$：几率(odds), 反映了 x 作为正例的相对可能性

“对数几率回归” (logistic regression)
简称“对率回归”

- 无需事先假设数据分布
- 可得到“类别”的近似概率预测
- 可直接应用现有数值优化算法求取最优解

注意：它是
分类学习算法!

对数几率回归

若将 y 看作类后验概率估计 $p(y=1 \mid \boldsymbol{x})$，则

$$\ln \frac{y}{1-y} = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b \quad \text{可写为} \quad \ln \frac{p(y=1 \mid \boldsymbol{x})}{p(y=0 \mid \boldsymbol{x})} = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$$

于是，可使用“极大似然法”

(maximum likelihood method)

给定数据集 $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$，最大化样本属于其真实标记的概率

最大化“对数似然”(log-likelihood)函数

$$\ell(\boldsymbol{w}, b) = \sum_{i=1}^{m} \ln p(y_i \mid \boldsymbol{x}_i; \boldsymbol{w}, b)$$

对数几率回归

令 $\boldsymbol{\beta} = (\boldsymbol{w}; b)$ $\hat{\boldsymbol{x}} = (\boldsymbol{x}; 1)$，则 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$ 可简写为 $\boldsymbol{\beta}^{\mathrm{T}}\hat{\boldsymbol{x}}$

再令 $p_1(\hat{\boldsymbol{x}}_i; \boldsymbol{\beta}) = p(y = 1 \mid \hat{\boldsymbol{x}}; \boldsymbol{\beta}) = \dfrac{e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}}{1 + e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}}$

$p_0(\hat{\boldsymbol{x}}_i; \boldsymbol{\beta}) = p(y = 0 \mid \hat{\boldsymbol{x}}; \boldsymbol{\beta}) = 1 - p_1(\hat{\boldsymbol{x}}_i; \boldsymbol{\beta}) = \dfrac{1}{1 + e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}}$

则似然项可重写为 $p(y_i \mid \boldsymbol{x}_i; \boldsymbol{w}_i, b) = y_i p_1(\hat{\boldsymbol{x}}_i; \boldsymbol{\beta}) + (1 - y_i) p_0(\hat{\boldsymbol{x}}_i; \boldsymbol{\beta})$

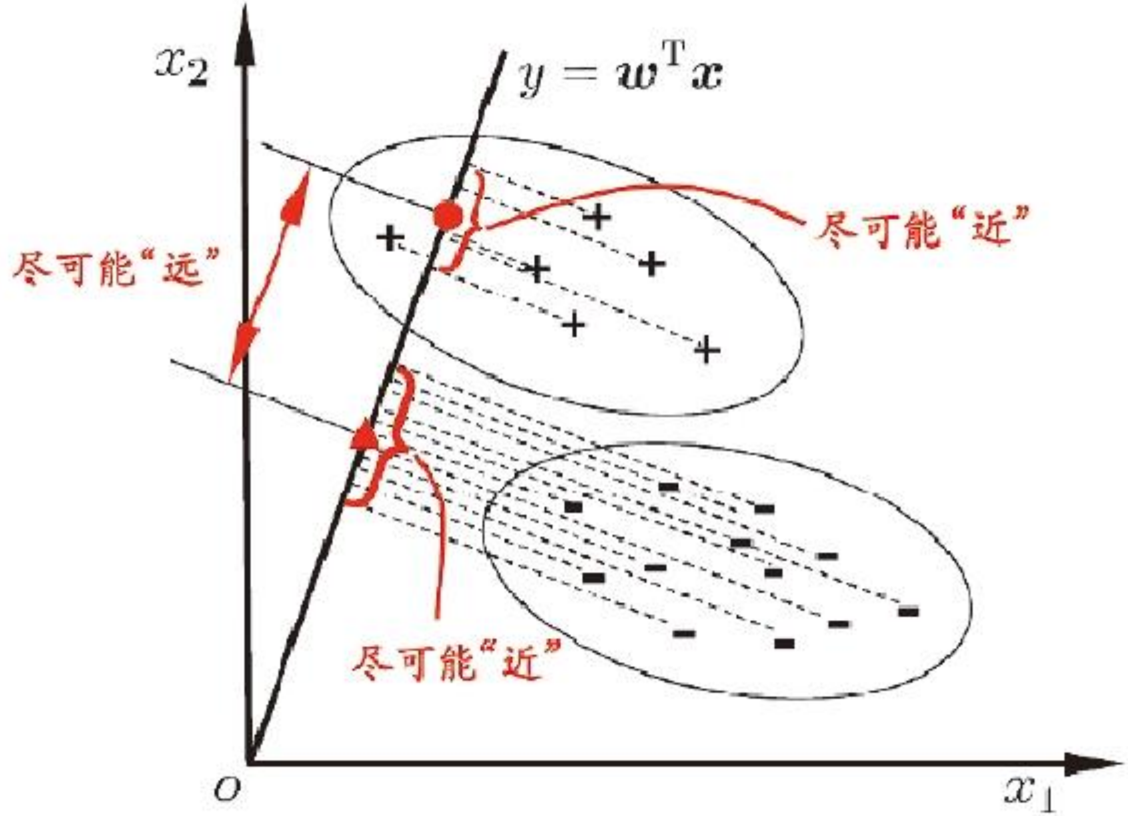
于是，最大化似然函数 $\ell(\boldsymbol{w}, b) = \sum_{i=1}^{m} \ln p(y_i \mid \boldsymbol{x}_i; \boldsymbol{w}, b)$

等价为最小化 $\ell(\boldsymbol{\beta}) = \sum_{i=1}^{m} \left(-y_i \boldsymbol{\beta}^{\mathrm{T}}\hat{\boldsymbol{x}}_i + \ln\left(1 + e^{\boldsymbol{\beta}^{\mathrm{T}}\hat{\boldsymbol{x}}_i}\right)\right)$

高阶可导连续凸函数，可用经典的数值优化方法
如梯度下降法/牛顿法 [Boyd and Vandenberghe, 2004]

目录

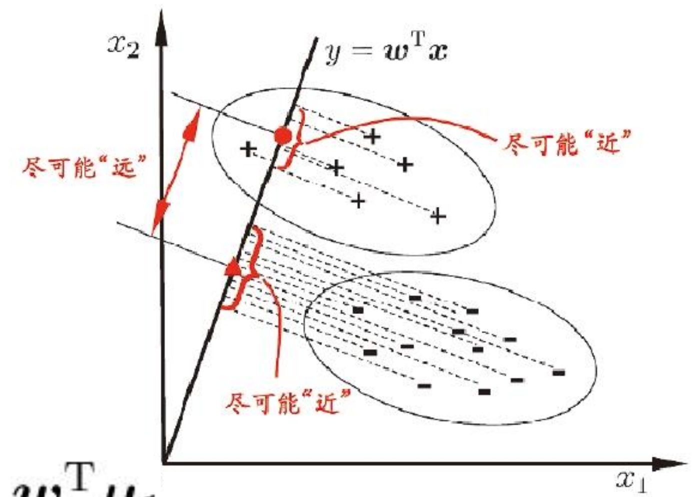
线性判别分析（LDA)



LDA 的思想: 给定训练样例集设法将样例投影到一条直线上，使得同类样例的投影点尽可能接近、异类样例的投影点尽可能远离; 在对新样本进行分类时，将其投影到同样的这条直线上，再根据投影点的位置来确定新样本的类别.

线性判别分析（LDA）

给定数据集 $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$



第 i 类示例的集合 X_i

第 i 类示例的均值向量 $\boldsymbol{\mu}_i$

第 i 类示例的协方差矩阵 $\boldsymbol{\Sigma}_i$

两类样本的中心在直线上的投影：$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0$ 和 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1$

两类样本的协方差：$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_0\boldsymbol{w}$ 和 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_1\boldsymbol{w}$

同类样例的投影点尽可能接近 → $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_0\boldsymbol{w} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_1\boldsymbol{w}$ 尽可能小

异类样例的投影点尽可能远离 → $\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0 - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1\|_2^2$ 尽可能大

于是，最大化

$$J = \frac{\left\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0 - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1\right\|_2^2}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_0\boldsymbol{w} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_1\boldsymbol{w}} = \frac{\boldsymbol{w}^{\mathrm{T}}\left(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1\right)\left(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1\right)^{\mathrm{T}}\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\left(\boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1\right)\boldsymbol{w}}$$

线性判别分析（LDA）

令 $\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w}=1$，最大化广义瑞利商等价形式为

$$\min_{\boldsymbol{w}} \ -\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_b\boldsymbol{w}$$
$$\text{s.t. } \boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w}=1$$

运用拉格朗日乘子法，有 $\mathbf{S}_b\boldsymbol{w}=\lambda\mathbf{S}_w\boldsymbol{w}$

$\mathbf{S}_b\boldsymbol{w}$ 的方向恒为 $\boldsymbol{\mu}_0-\boldsymbol{\mu}_1$，不妨令 $\mathbf{S}_b\boldsymbol{w}=\lambda(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)$

于是 $\boldsymbol{w}=\mathbf{S}_w^{-1}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)$

实践中通常是进行奇异值分解 $\mathbf{S}_w=\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{T}}$ $\longrightarrow$ 附录 A

然后 $\mathbf{S}_w^{-1}=\mathbf{V}\boldsymbol{\Sigma}^{-1}\mathbf{U}^{\mathrm{T}}$

目录

多分类学习

- 多分类学习方法
  - 二分类学习方法推广到多类
  - 利用二分类学习器解决多分类问题（常用）
    - 对问题进行拆分，为拆出的每个二分类任务训练一个分类器
    - 对于每个分类器的预测结果进行集成以获得最终的多分类结果
- 拆分策略
  - 一对一（One vs. One, OvO）
  - 一对其余（One vs. Rest, OvR）
  - 多对多（Many vs. Many, MvM）

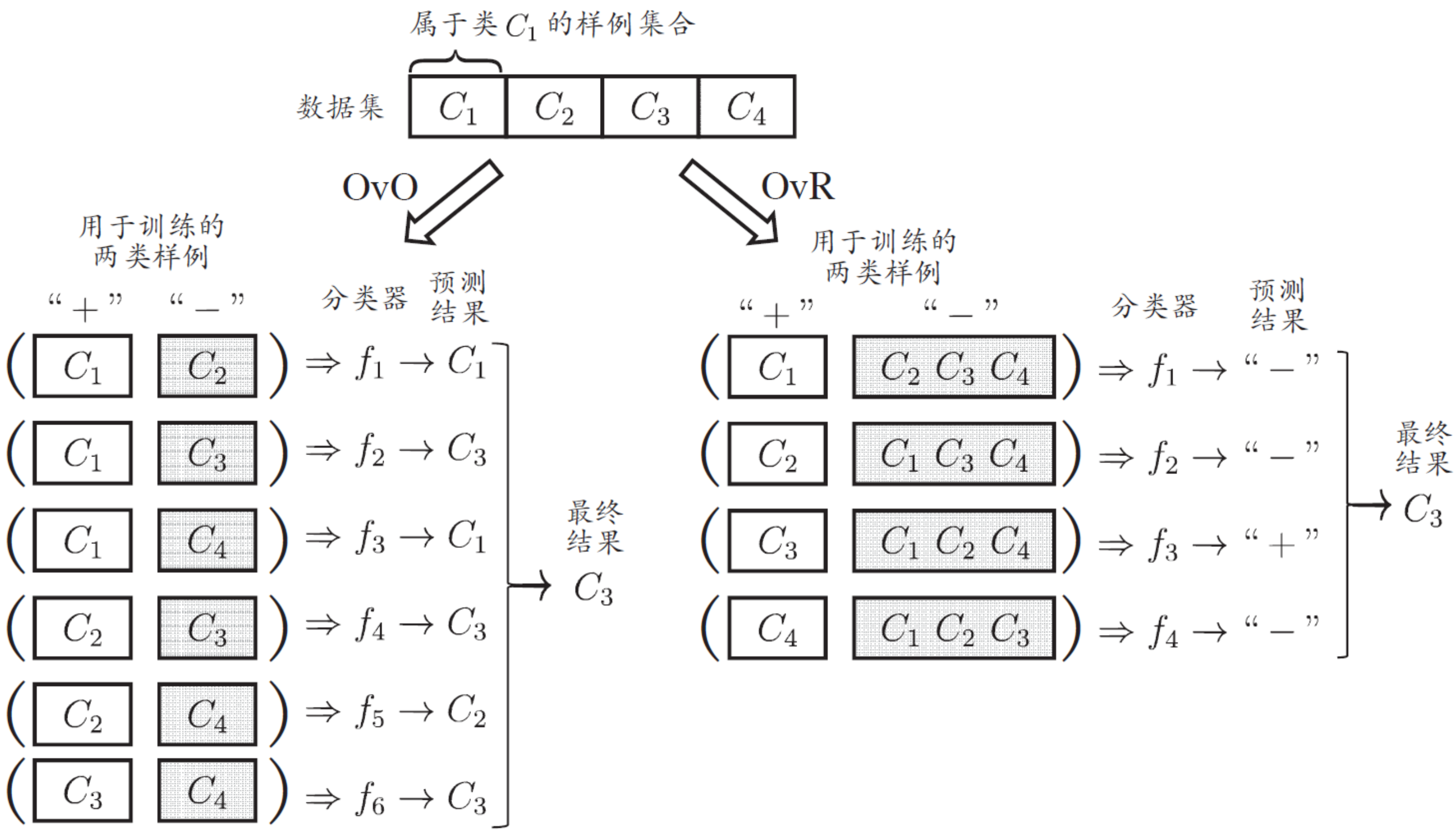
多分类学习- 一对一

- 拆分阶段
  - N个类别两两配对
    - N(N-1)/2 个二分类任务
  - 各个二分类任务学习分类器
    - N(N-1)/2 个二类分类器
- 测试阶段
  - 新样本提交给所有分类器预测
    - N(N-1)/2 个分类结果
  - 投票产生最终分类结果
    - 被预测最多的类别为最终类别

多分类学习- 一对其余

- 任务拆分
  - 某一类作为正例，其他反例
    - N 个二分类任务
  - 各个二类任务学习分类器
    - N 个二分类分类器
- 测试阶段
  - 新样本提交给所有分类器预测
    - N 个分类结果
  - 比较各分类器预测置信度
    - 置信度最大类别作为最终类别

多分类学习– 两种策略比较



多分类学习- 两种策略比较

一对一

- 训练N(N-1)/2个分类器，存储开销和测试时间大
- 训练只用两个类的样例，训练时间短

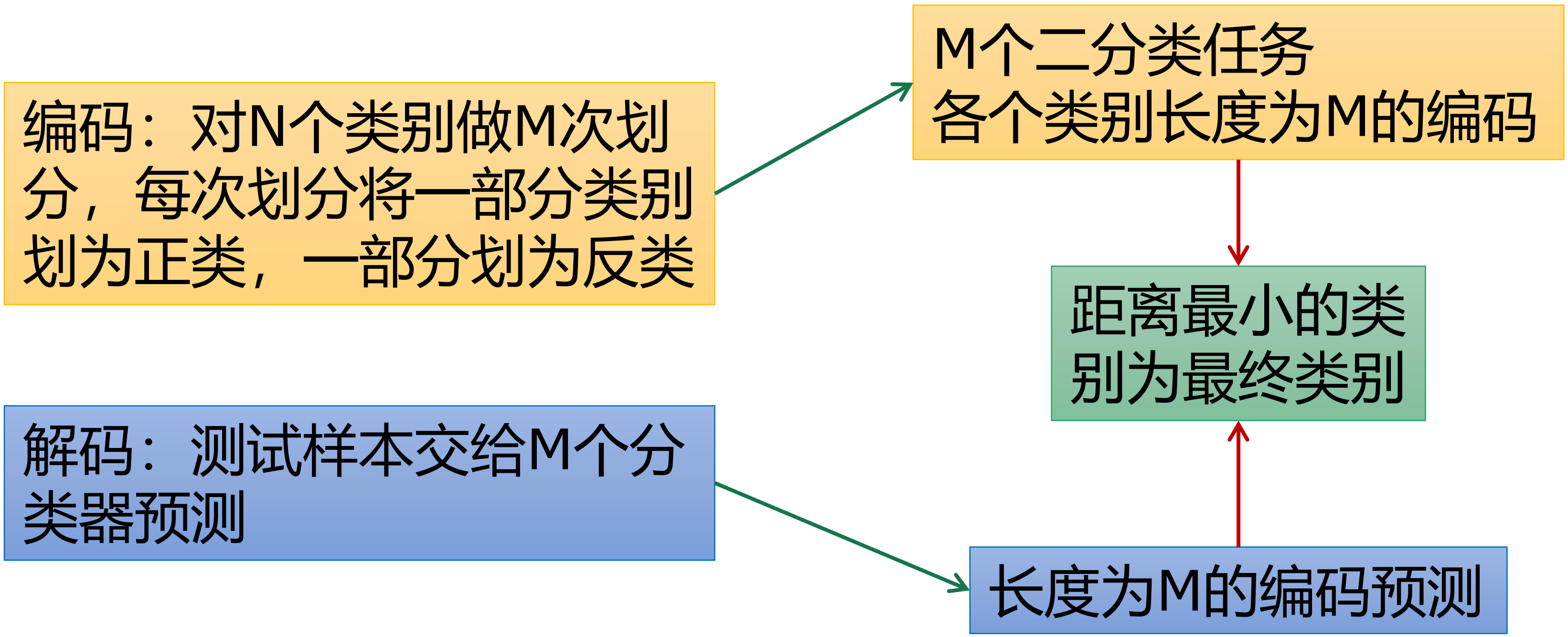
一对其余

- 训练N个分类器，存储开销和测试时间小
- 训练用到全部训练样例，训练时间长

预测性能取决于具体数据分布，多数情况下两者差不多

多分类学习– 多对多

- 多对多（Many vs Many, MvM）
  - 若干类作为正类，若干类作为反类
- 纠错输出码（Error Correcting Output Code, ECOC）



多分类学习- 多对多

- 纠错输出码(Error Correcting Output Code, ECOC)

	f_1	f_2	f_3	f_4	f_5	海明距离	欧氏距离
C_1	−1	+1	−1	+1	+1	3	$2\sqrt{3}$
C_2	+1	−1	−1	+1	−1	4	4
C_3	−1	+1	+1	−1	+1	1	2
C_4	−1	−1	+1	+1	−1	2	$2\sqrt{2}$
测试示例	−1	−1	+1	−1	+1		

(a) 二元 ECOC 码

[Dietterich and Bakiri,1995]

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	海明距离	欧氏距离
C_1	−1	−1	+1	+1	−1	+1	+1	4	4
C_2	−1	0	0	0	+1	−1	0	2	2
C_3	+1	+1	−1	−1	−1	+1	−1	5	$2\sqrt{5}$
C_4	−1	+1	0	+1	−1	0	+1	3	$\sqrt{10}$
测试示例	−1	+1	+1	−1	+1	−1	+1		

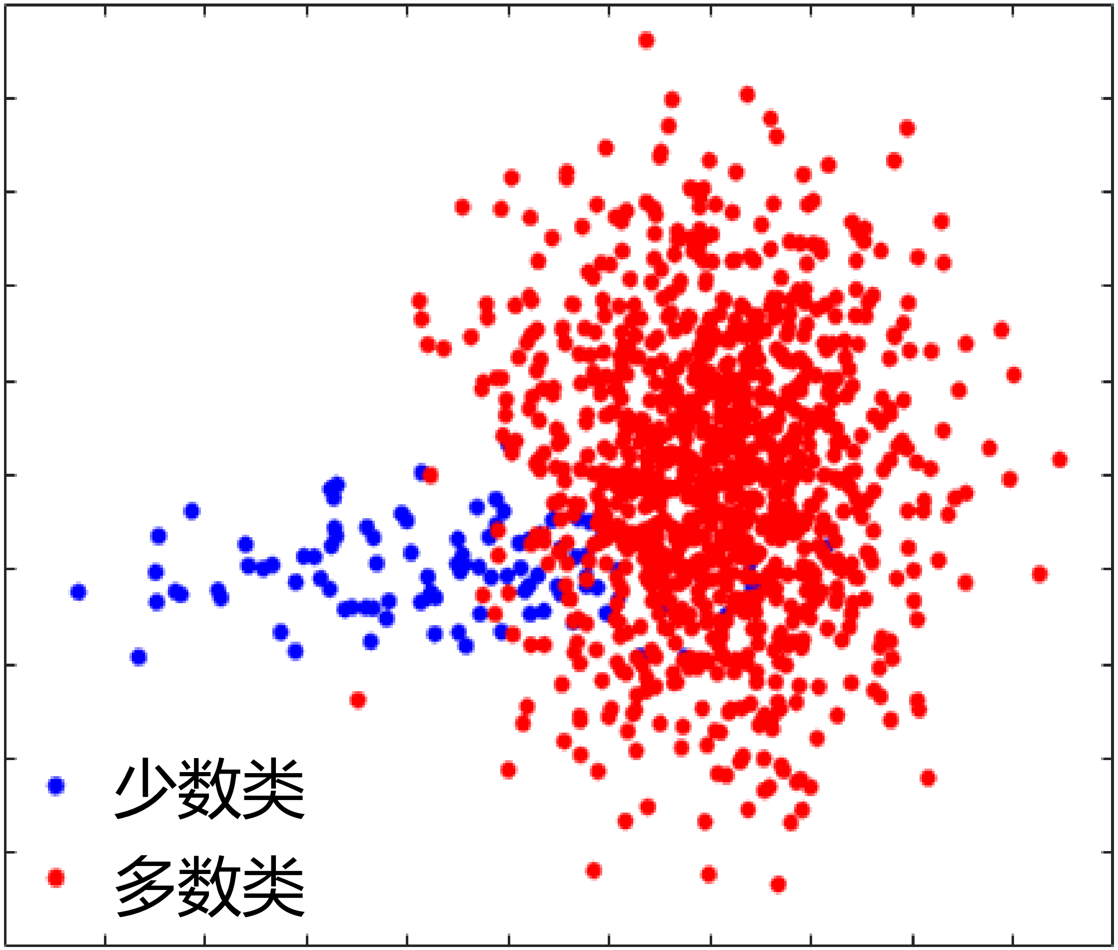
(b) 三元 ECOC 码

[Allwein et al. 2000]

- ECOC编码对分类器错误有一定容忍和修正能力，编码越长、纠错能力越强
- 对同等长度的编码，理论上来说，任意两个类别之间的编码距离越远，则纠错能力越强

目录

类别不平衡问题



多数类：

拥有较多样本数量的类别

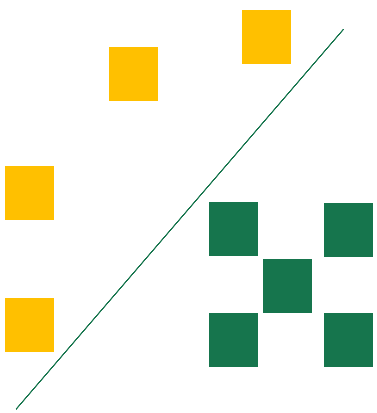
少数类：

样本数量少的类别

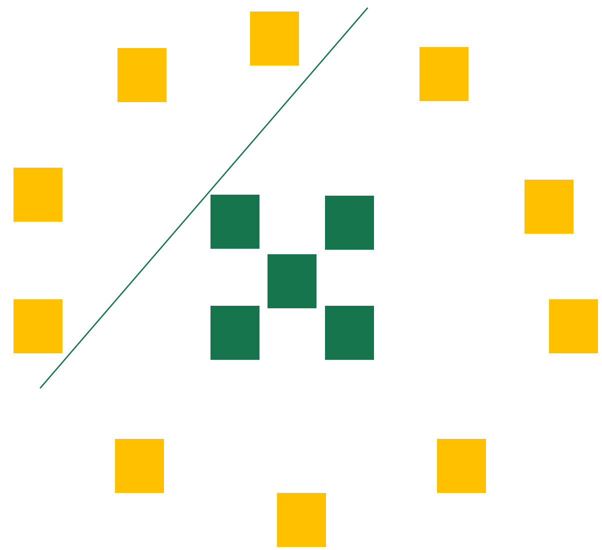
类别不平衡问题指不同类别的训练样本数目相差很大

类别不平衡问题

如果使用传统分类问题的方法去解决带有类别不平衡的问题时，会出现怎样的情况?



采集的数据分布



真实的数据分布

少数类样本将会很容易被误分！！！

类别不平衡问题

数据角度 → ① 欠采样　②过采样

算法角度 → 集成学习（第8章内容）

类别不平衡问题

- 类别不平衡（class imbalance）
  - 不同类别训练样例数相差很大情况（正类为小类）

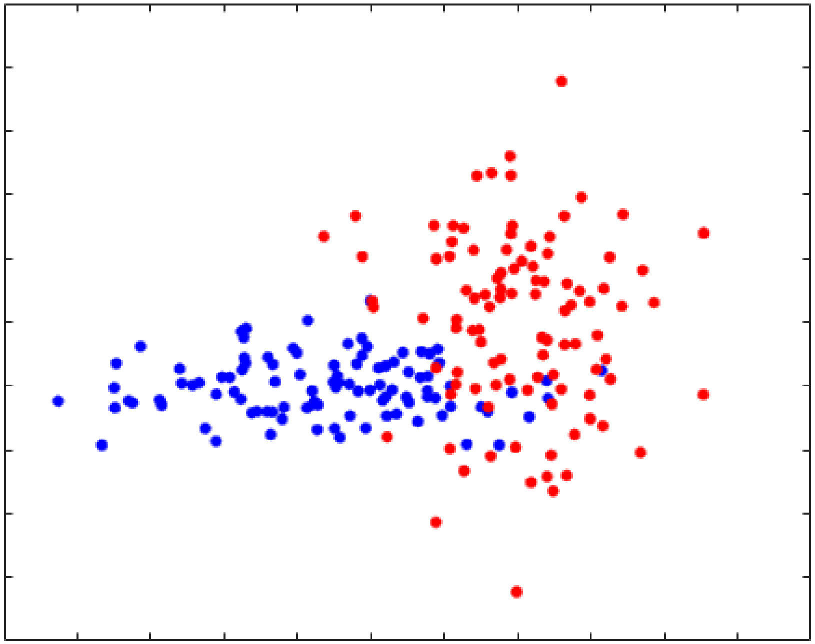
类别平衡正例预测 $\frac{y}{1-y} > 1$ ⇒ $\frac{y}{1-y} > \frac{m^+}{m^-}$ 正负类比例

- 再缩放（rescaling）
  - 阈值移动（threshold-moving）
  - 欠采样（undersampling）
    - 去除一些反例使正反例数目接近（EasyEnsemble [Liu et al.,2009]）
  - 过采样（oversampling）
    - 增加一些正例使正反例数目接近（SMOTE [Chawla et al.2002]）

欠采样

随机欠采样：

随机地删除一切多数类样本。



改进的欠采样：

有选择地去除一些对最终分类结果**影响不大**的多数类样本。（即删除远离分类边界或引起数据重叠的多数类样本）

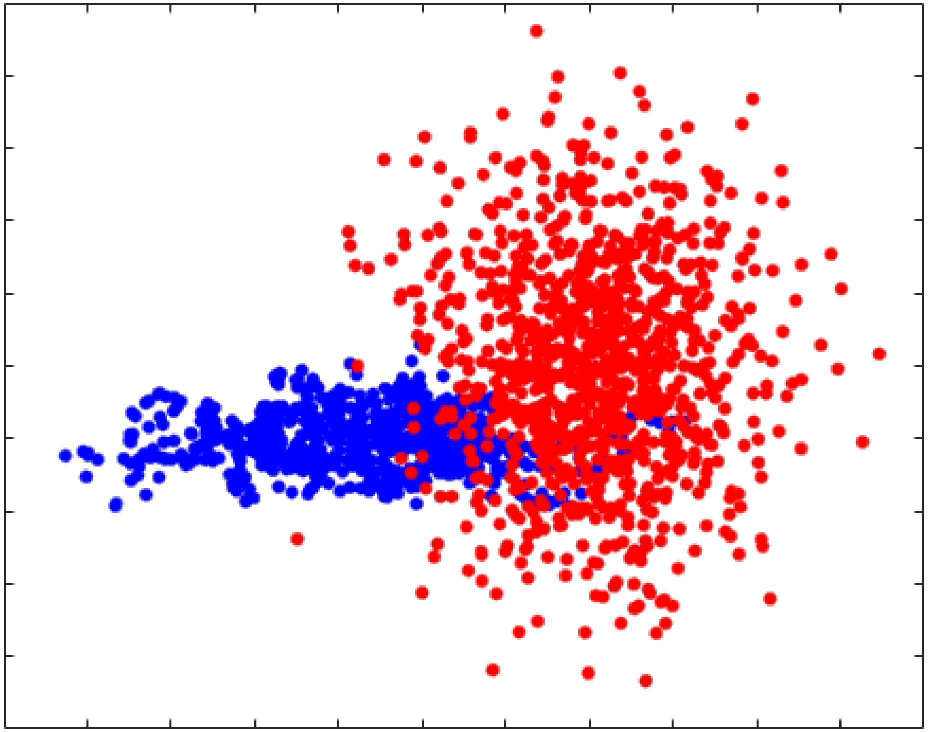
过采样

随机过采样：

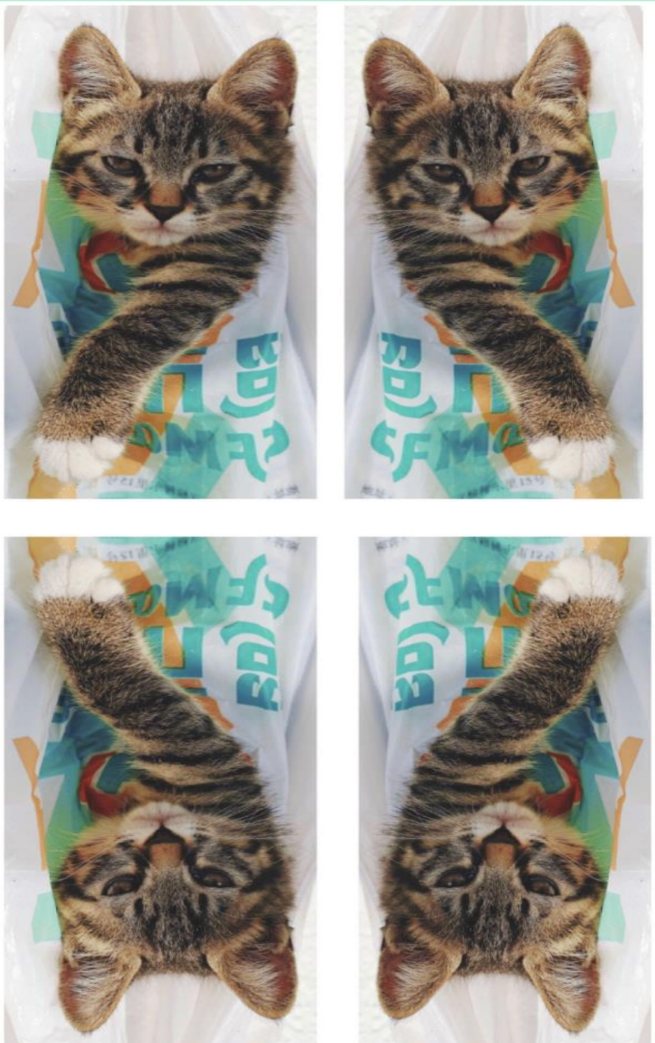
随机地复制一些少数类样本。

启发式过采样：

生成一些新的少数类样本。



图像方面经常采用的过采样方法



翻转

随机裁剪

图像方面经常采用的过采样方法

旋转

颜色扰动

总结

- 线性回归
  - 最小二乘法（最小化均方误差）
- 二分类任务
  - 对数几率回归
    - 单位阶跃函数、对数几率函数、极大似然法
  - 线性判别分析
    - 最大化广义瑞利商
- 多分类学习
  - 一对一
  - 一对其余
  - 多对多
    - 纠错输出码
- 类别不平衡问题
  - 基本策略：再缩放