



本课件仅用于教学使用。未经许可，任何单位、组织和个人不得将课件用于该课程教学之外的用途(包括但不限于盈利等)，也不得上传至可公开访问的网络环境

1

数据科学导论

Introduction to Data Science

第四章 数据挖掘基础

黄振亚，陈恩红

Email: huangzhy@ustc.edu.cn, cheneh@ustc.edu.cn

课程主页：

<http://staff.ustc.edu.cn/~huangzhy/Course/DS2025.html>

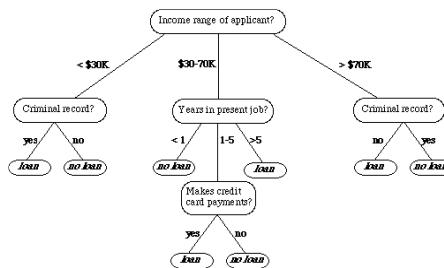


数据建模基础

11

□ 数据挖掘——四个任务有哪些常用方法？

分类与预测



关联分析



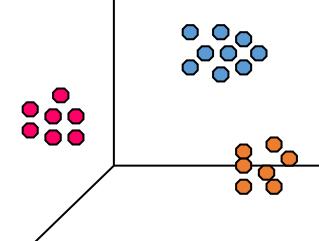
数据

	T		H		P	
	L	H	L	H	L	H
J	-6.0	8.8	60	100	986	1044
F	-2.8	10.9	48	100	973	1025
M	-5.6	17.7	34	100	976	1037
A	-1.2	22.2	27	100	996	1036
M	-0.8	27.8	25	100	1003	1034
J	5.2	29.1	26	100	998	1030
J	9.8	30.6	23	99	997	1027
A	5.6	26.1	31	100	992	1029
S	5.2	24.8	35	100	998	1028
O	-0.4	21.3	42	100	990	1031
N	-7.6	17.3	55	100	963	1023
D	-10.4	9.2	53	100	987	1039

table 17a

2010 monthly weather variation, Cambridge (UK)

聚类





分类与预测

16

□ 分类与预测 – 有监督学习

- 已知：一组数据（训练集） (X, Y)
- 如右图，每一条记录表示为 (x, y)

- x : 数据特征/属性（如收入）
- y : 类别标记（是否有借款）

- 任务：
 - 学习一个模型，利用每一条记录的特征 x 去预测它对应的类别 y

即：输入未标记的数据（含特征 x ），
预测数据的类别 y

分类 / 数值预测 取决于 类别标签是
离散型 / 数值型

3个特征：

- 是否有住房
- 婚姻状态
- 年收入

类别：
是否拖欠贷款

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



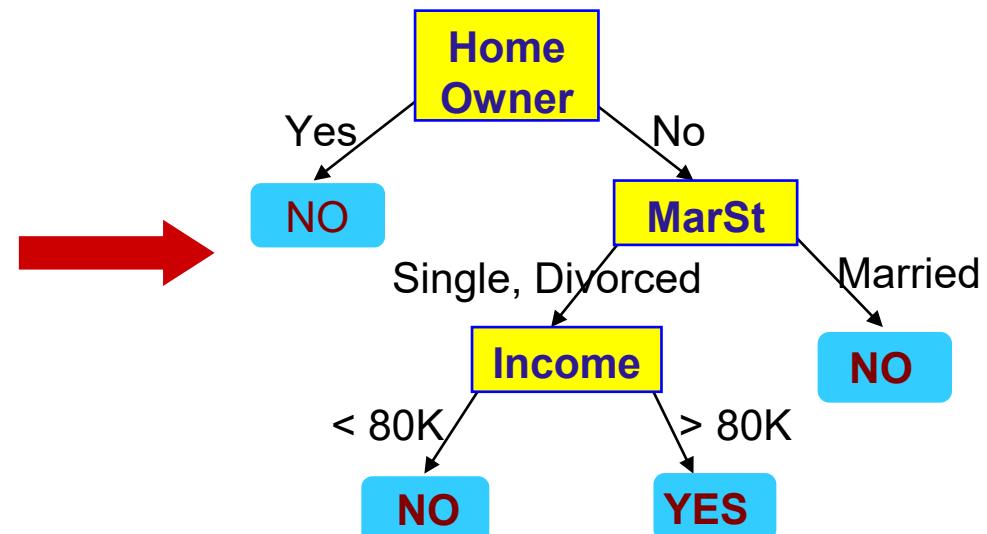
分类：决策树

34

□ 如何建立决策树？

- 基本的决策树学习过程，可以归纳为以下三个步骤：
- 1. 特征选择：选取对于训练数据有着较强区分能力的特征
- 2. 生成决策树：基于选定的特征，逐步生成完整的决策树
- 3. 决策树剪枝：简化部分枝干，避免过拟合因素影响

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes





分类与预测

61

- 有监督学习：分类与预测
- 常用方法
 - 规则方法
 - 决策树
 - 最近邻方法
 - 支持向量机 (SVM)
 - 集成方法
- 类不平衡问题
- 分类的评价指标

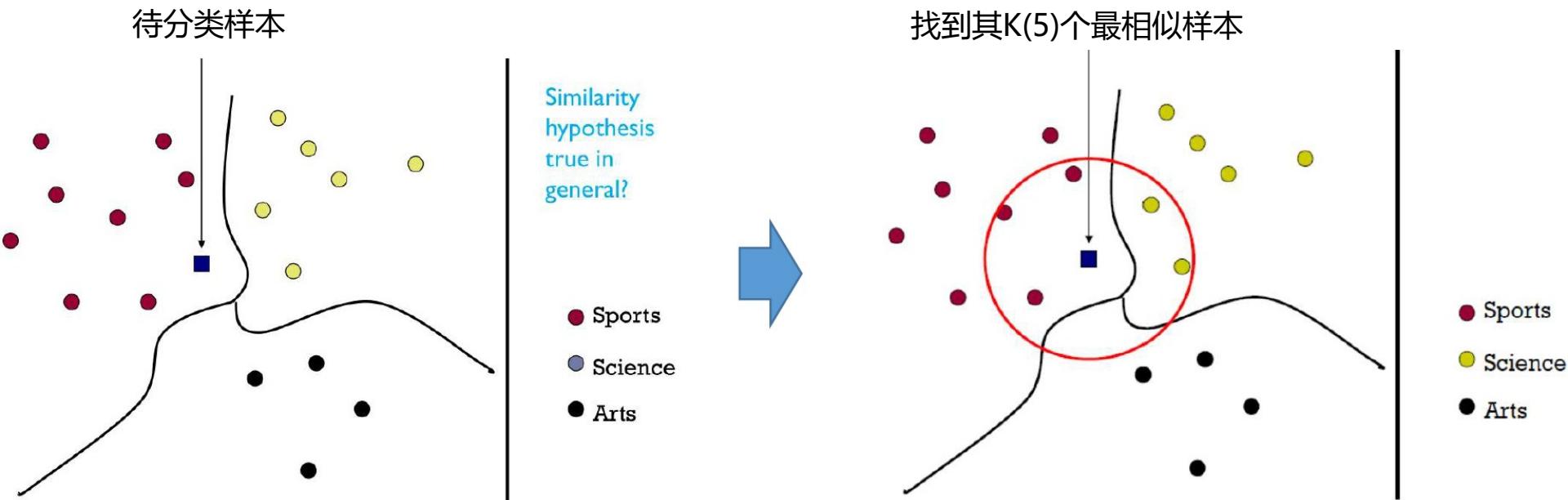


分类：K近邻方法

63

□ 分类——K近邻方法

- 对数据空间内的样本，可提出相似样本假设
 - 表征上相近的样本应该属于同一个类别
- K近邻思想：用K个最相似样本的类别来预测未知样本的类别(投票方法)
- 核心问题：距离度量、K的取值

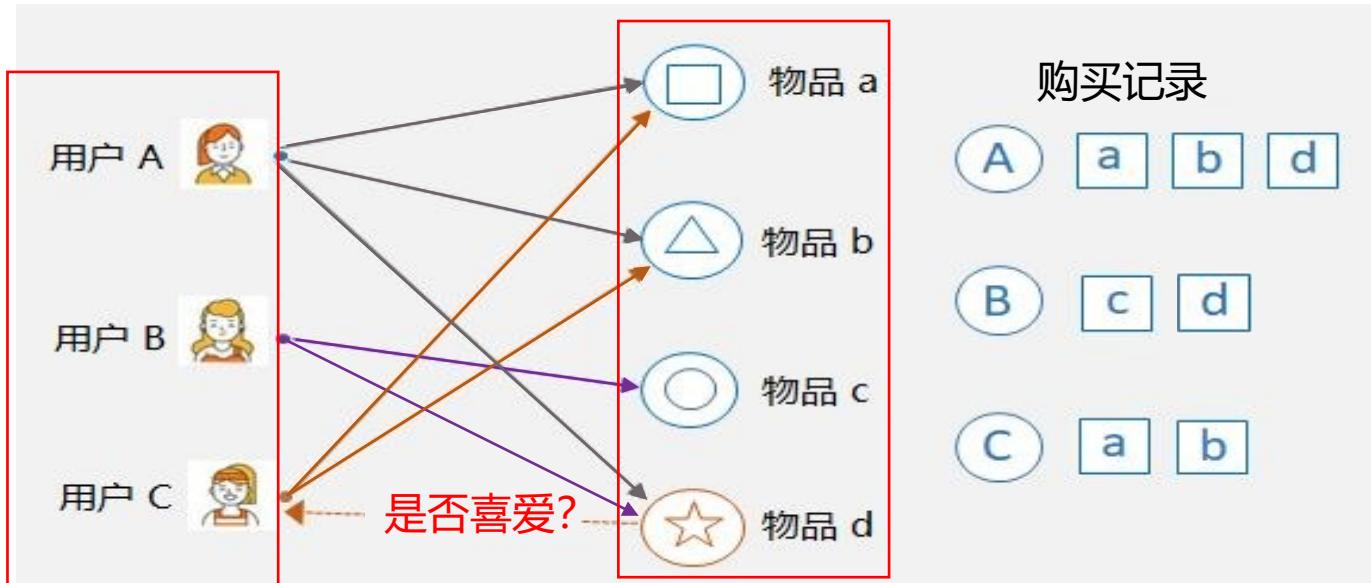




K近邻方法实例

67

□ K近邻思想的应用实例：推荐系统的UCF、ICF模型



UCF: 基于K个相似用户对物品的评分

- 用户A、B购买过物品d，且与C相似，可用他们对d的平均喜爱程度作为C对d的喜爱程度

ICF: 基于用户对K个相似物品的评分

- C购买过a,b，若物品d与a,b相似，可用C对他们的平均喜爱程度作为对物品d的喜爱程度

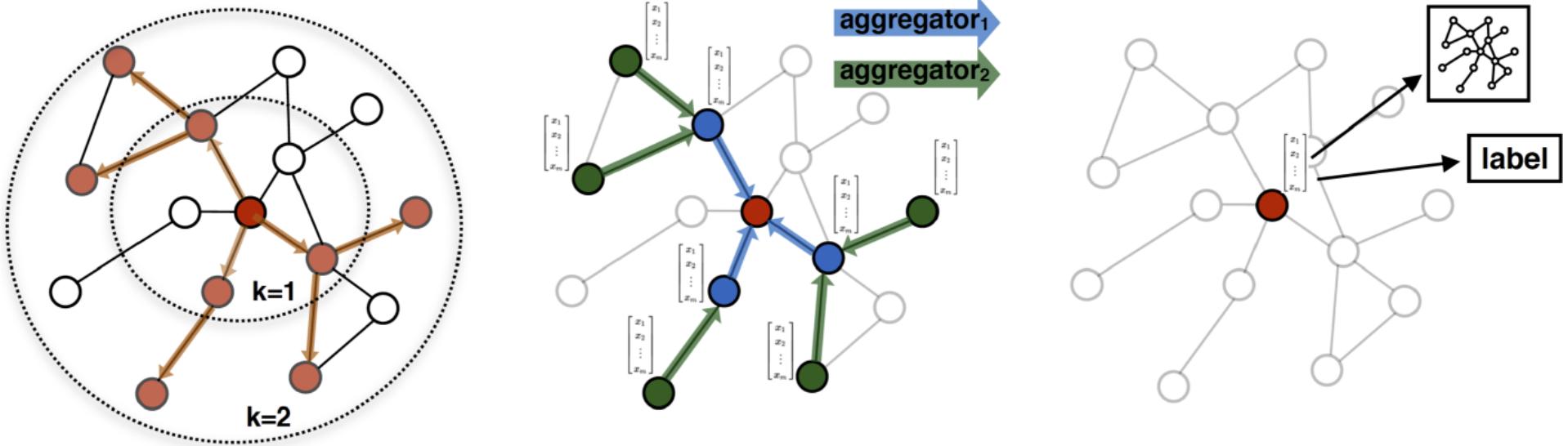
- Sarwar, Badrul, et al. Item-based collaborative filtering recommendation algorithms. WWW 2001.
➤ Amazon recommendations: Item-to-item collaborative filtering. IEEE Internet computing, 2003



K近邻方法实例

68

- K近邻思想的应用实例：图神经网络中的近邻
 - 基本思想：将K个邻居节点的信息传播到当前节点
 - 距离度量：基于注意力机制计算(GAT模型)



➤ Hamilton, Will, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs." NeurIPS 2017



KNN与Transformer

69

□ Transformer

- Google Brain 2017的提出的一篇工作
- 针对RNN的弱点进行重新设计，解决了RNN效率问题和传递中的缺陷
- **Original paper:** Vaswani et al. *Attention Is All You Need*. In *NIPS*, 2017.

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

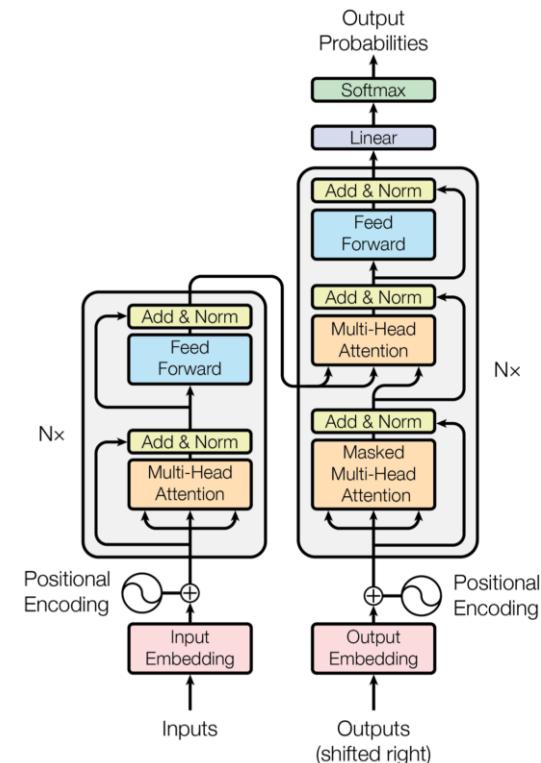
Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Lukasz Kaiser*
Google Brain
lukasz.kaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com





分类与预测

70

□ 有监督学习：分类与预测

□ 常用方法

- 规则方法
- 决策树
- 最近邻方法
- 支持向量机 (SVM)
- 集成方法

□ 分类的评价指标

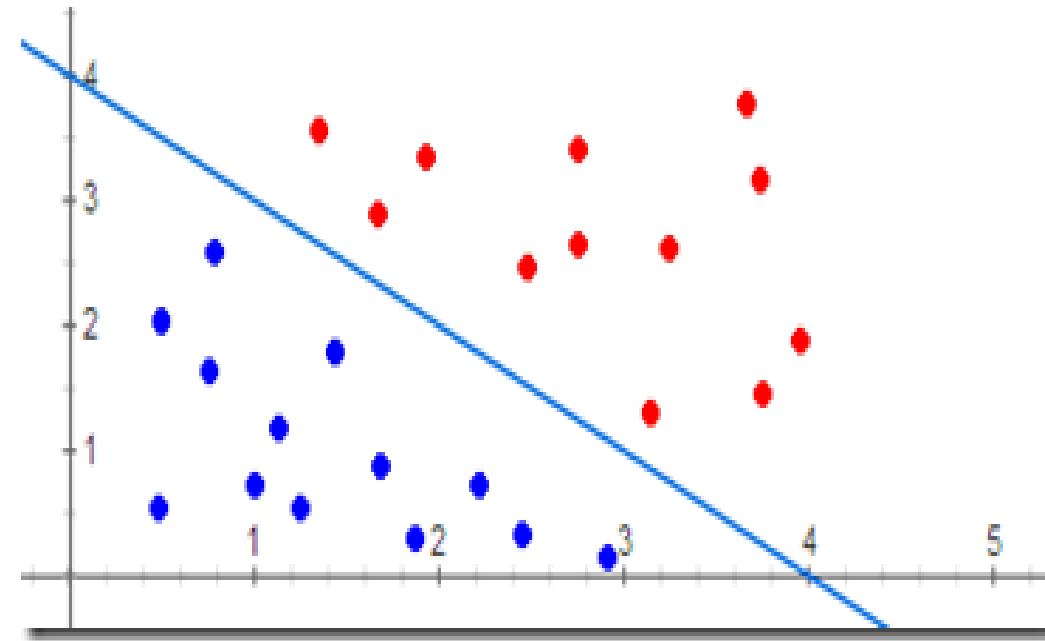


分类：感知机

71

□ 分类——感知机 (perceptron)

- 目标：寻找一条直线 S （高维时是超平面）划分不同类别的数据
- 输入：样本的特征向量 $X=\{x\}, x \in R^d$
- 输出：样本类别 $y \in \{-1, +1\}$



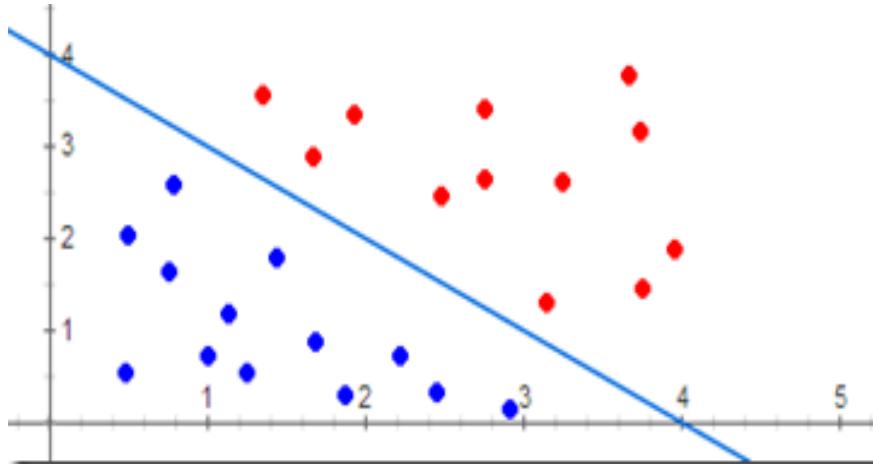


分类：感知机

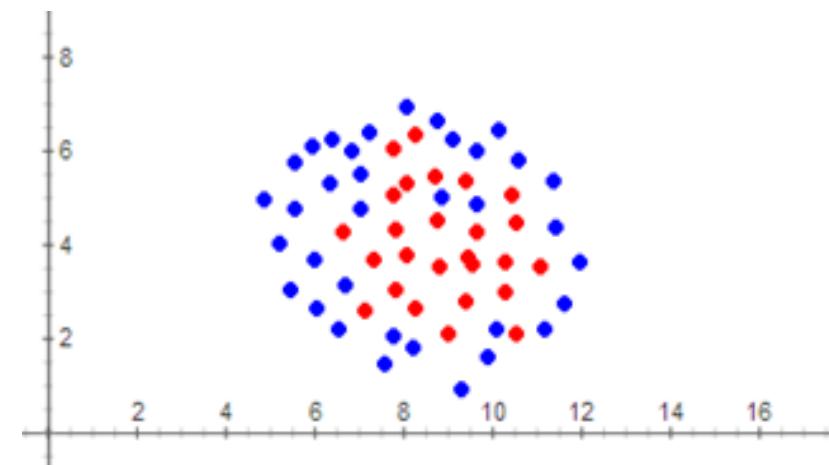
72

□ 分类——感知机 (perceptron)

- 1957年由Rosenblatt提出，是神经网络与支持向量机的基础
- 感知机的前提：样本空间线性可分
 - 左例中，可以用一条直线将+1类和-1类完美分开，称这个样本空间是线性可分的
 - 右例的样本是线性不可分的，感知机不能处理这种情况



线性可分



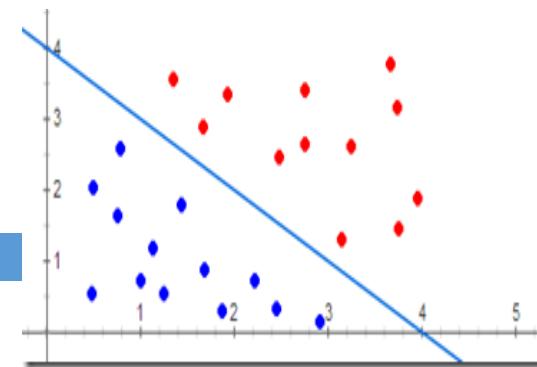
线性不可分



分类：感知机

73

□ 感知机的基本概念



- 模型：符号函数 $f(x) = \text{sign}(w \cdot x + b) = \begin{cases} +1, & w \cdot x + b \geq 0 \\ -1, & w \cdot x + b < 0 \end{cases}$
- 分界超平面 S : $w \cdot x + b = 0$
- 学习目标：最小化 误分类点 到 超平面 的 总距离
 - 点 (x_0, y_0) 到超平面 S : $w \cdot x + b = 0$ 的距离 $\frac{1}{\|w\|} |w \cdot x_0 + b|$ 推导过程省略
 - 误分类点 (x_i, y_i) 到超平面 S : $w \cdot x_i + b = 0$ 的距离为 $-\frac{1}{\|w\|} y_i (w \cdot x_i + b)$

x_i 错分时，若 y_i 为-1，则计算的 $(w \cdot x_i + b) > 0$
若 y_i 为+1，则计算的 $(w \cdot x_i + b) < 0$

- 损失函数: $\underset{w,b}{\operatorname{argmin}} L(w, b) = - \sum_{x_i \in M} y_i (w \cdot x_i + b),$
- 学习策略：找到参数 w 和 b ，使得损失函数最小

它是连续可导的，这就使得我们比较容易求得其最小值



分类：感知机

74

□ 感知机学习算法：梯度下降 — 课后学习

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

- 随机初始化 w_0 和 b_0
- 梯度下降不断地极小化损失函数
 - 每次随机选取一个误分类点对 w 和 b 进行更新。
 - 设误分类点为 (x_i, y_i) ，那么损失函数 $L(w, b)$ 的梯度为：

$$\nabla_w L(w, b) = -y_i \cdot x_i$$

$$\nabla_b L(w, b) = -y_i$$

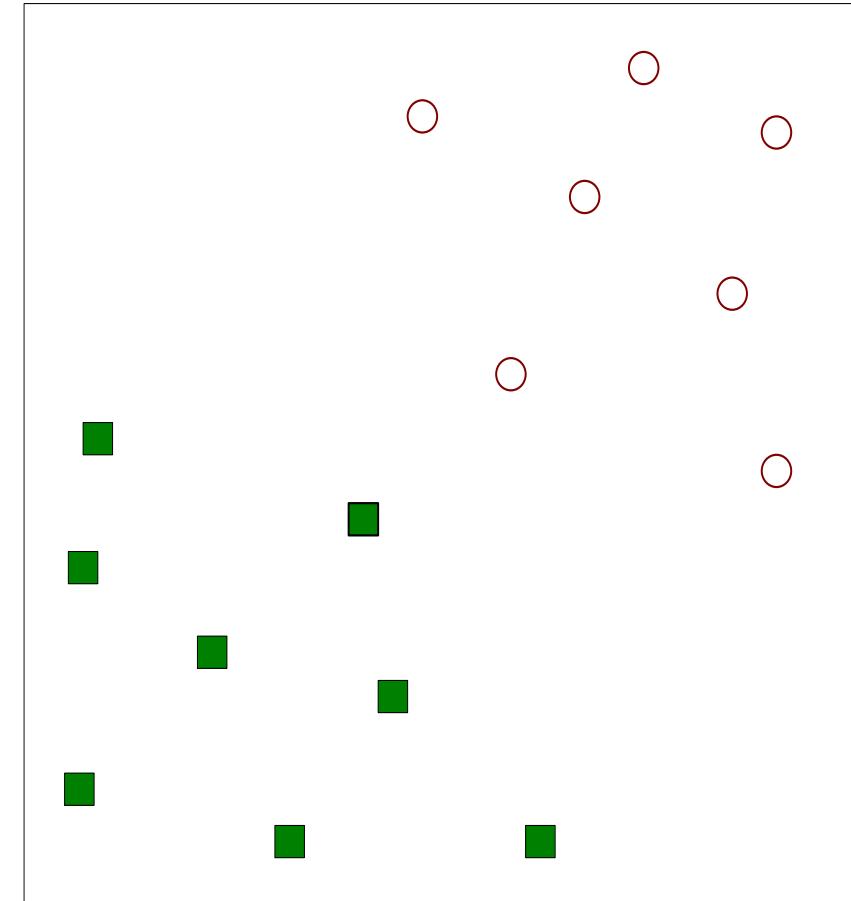
- 接下来对 w, b 进行更新 $w \leftarrow w + \gamma y_i \cdot x_i, b \leftarrow b + \gamma y_i$ ，其中 $\gamma (0 \leq \gamma \leq 1)$ 为步长，也称为学习速率 (learning rate)。
- 通过迭代，直到损失函数为 0 (无误分点)



分类：支持向量机

75

□ 分类——支持向量机 (Support Vector Machine)



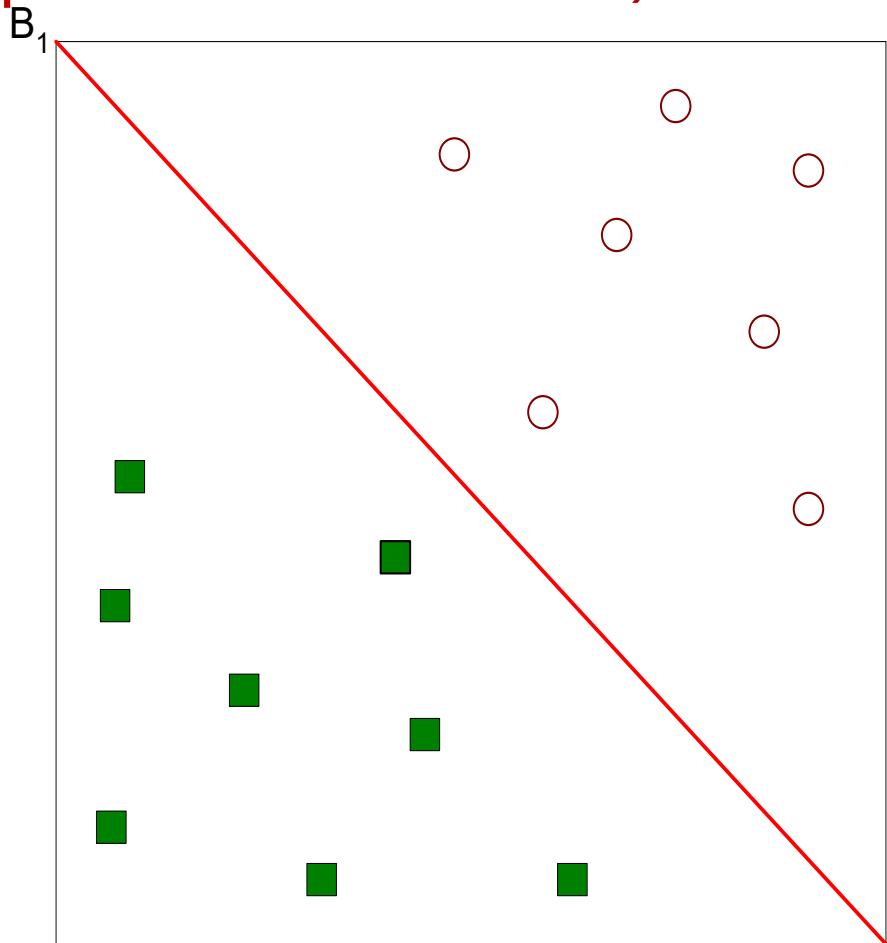


分类：支持向量机

76

□ 分类——支持向量机 (Support Vector Machine)

一个可行解



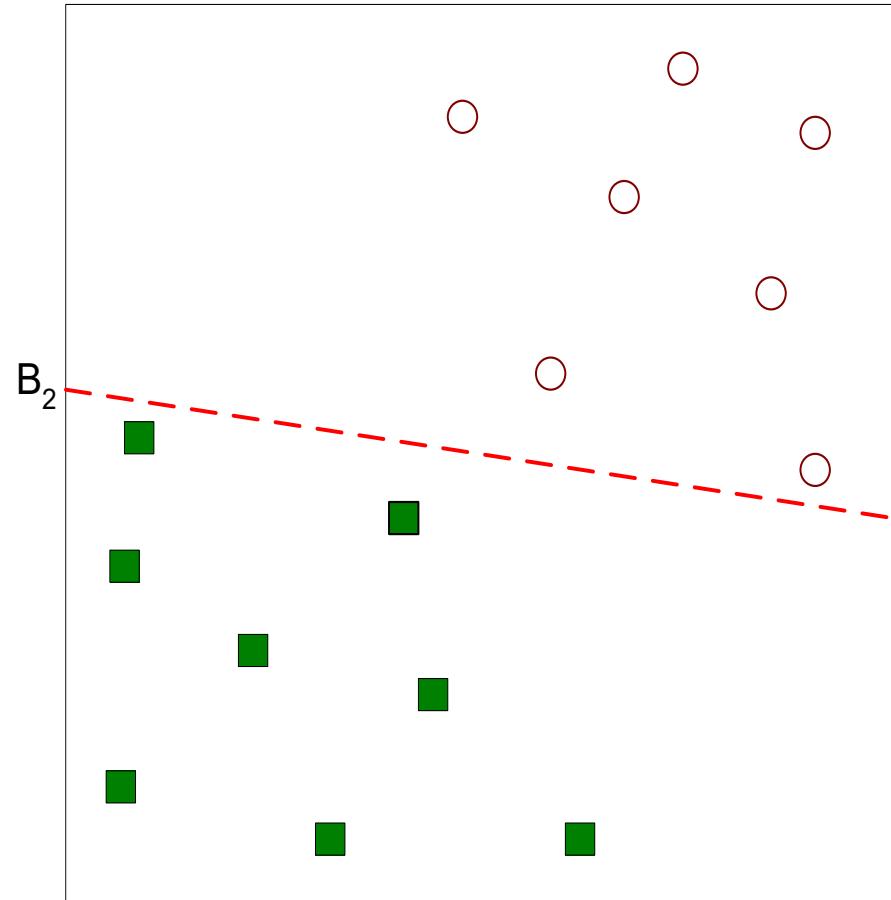


分类：支持向量机

77

□ 分类——支持向量机 (Support Vector Machine)

另一个可行解



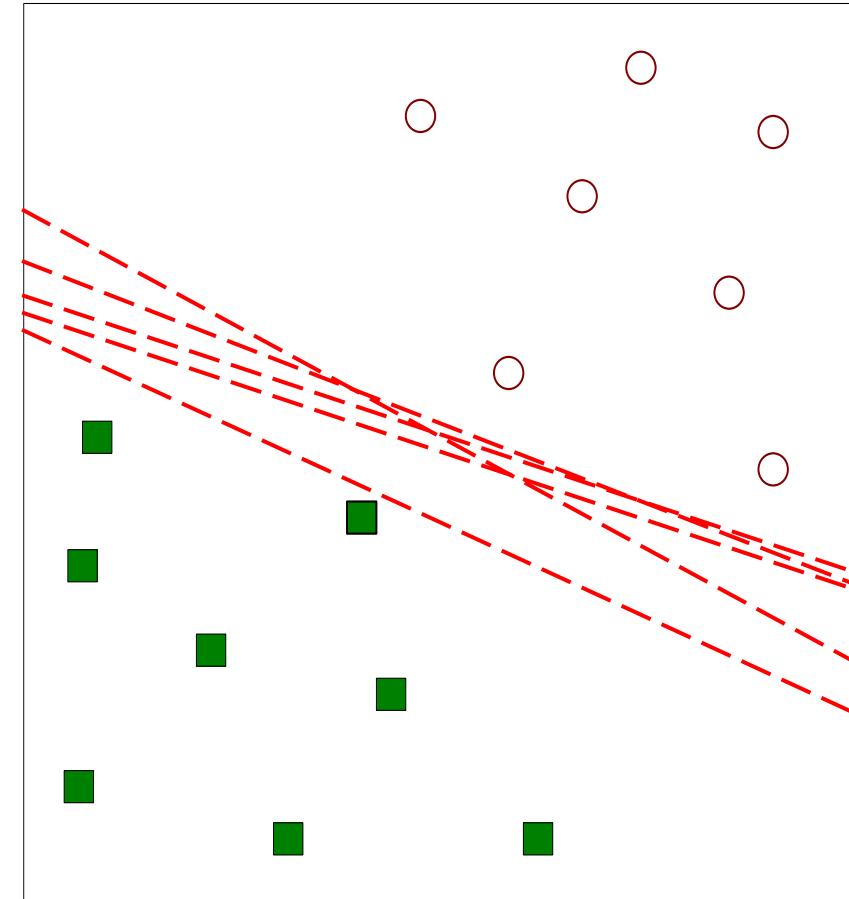


分类：支持向量机

78

□ 分类——支持向量机 (Support Vector Machine)

其他可行解





分类：支持向量机

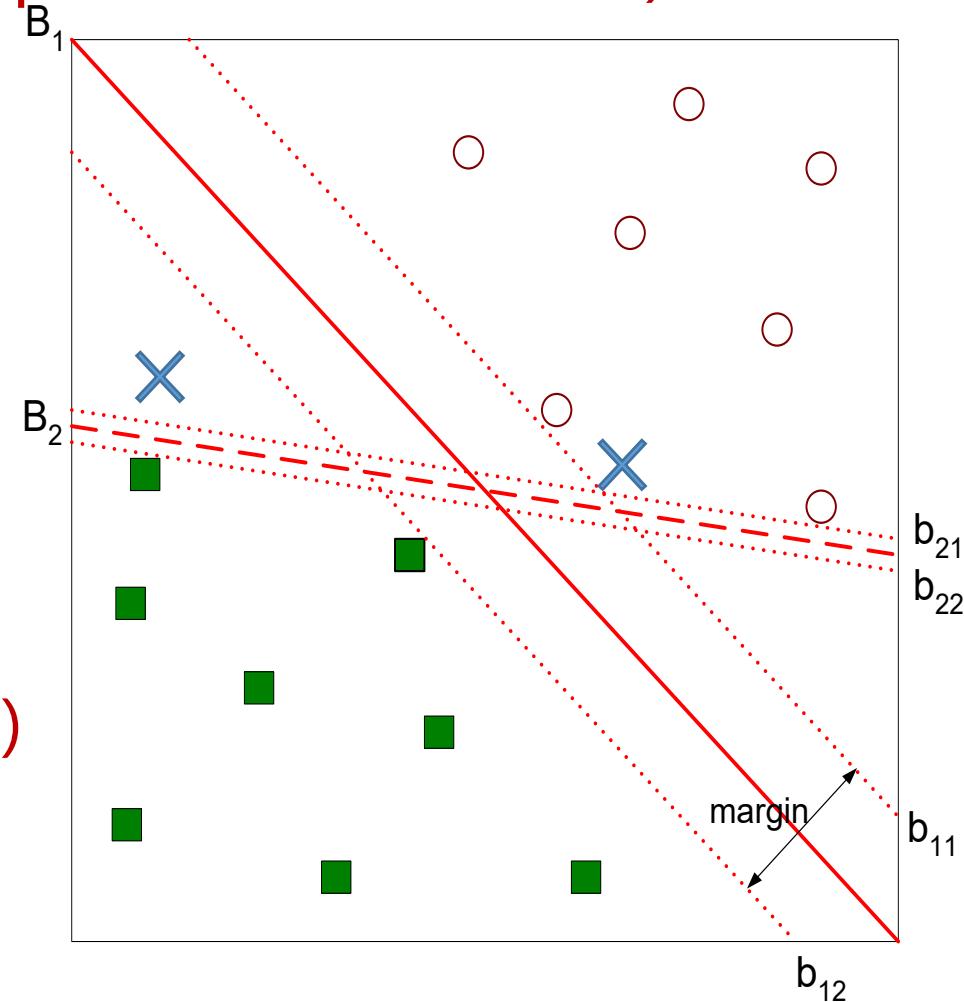
79

□ 分类——支持向量机 (Support Vector Machine)

B1与B2，哪个更好？

B1 保证分类正确（区分）

分类间隔大（更容易区分）





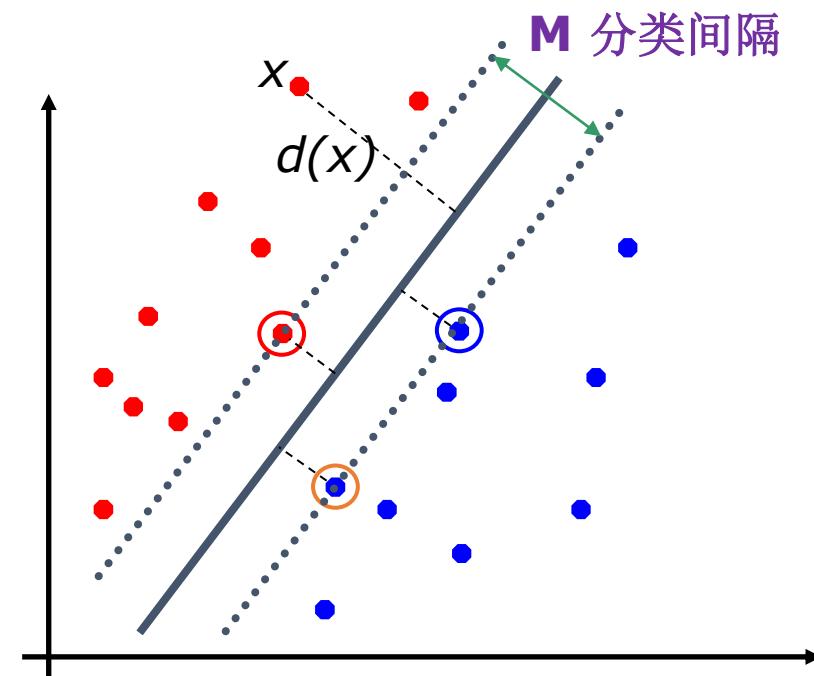
分类：支持向量机

80

□ 分类——支持向量机

- 因此，应该选择“正中”的最大间隔超平面（分类间隔最大）
 - 容忍性好，泛化能力强
 - 在线性可分的条件下，符合这样条件的超平面“存在且唯一”
- 问题：如何找到最优的超平面？

□ 最大化分类间隔





分类：支持向量机

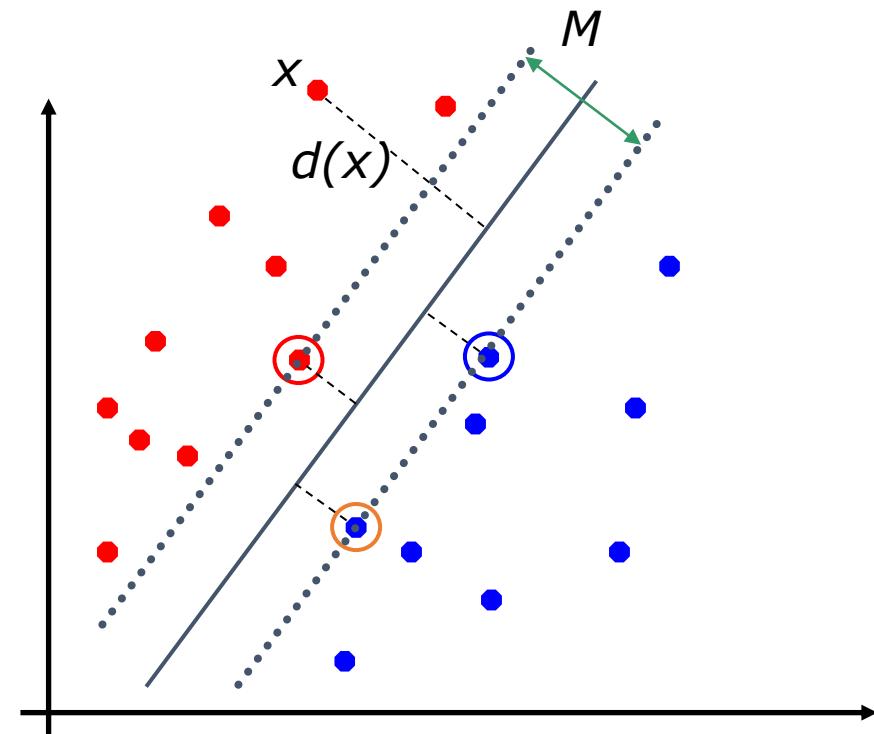
81

□ 分类——支持向量机

□ 理解：最大化分类间隔

- 区分能力更强
- 概率的角度：最难的点置信度最大
- 即使我们在选边界的时候犯了小错误，使得边界有偏移，仍然有很大概率保证可以正确分类绝大多数样本
- 很容易实现交叉验证，因为边界只与极少数的样本点有关
- 有一定的理论支撑
- 实验结果验证了其有效性

保证分类正确性—当前
保证分类质量—未来





分类：支持向量机

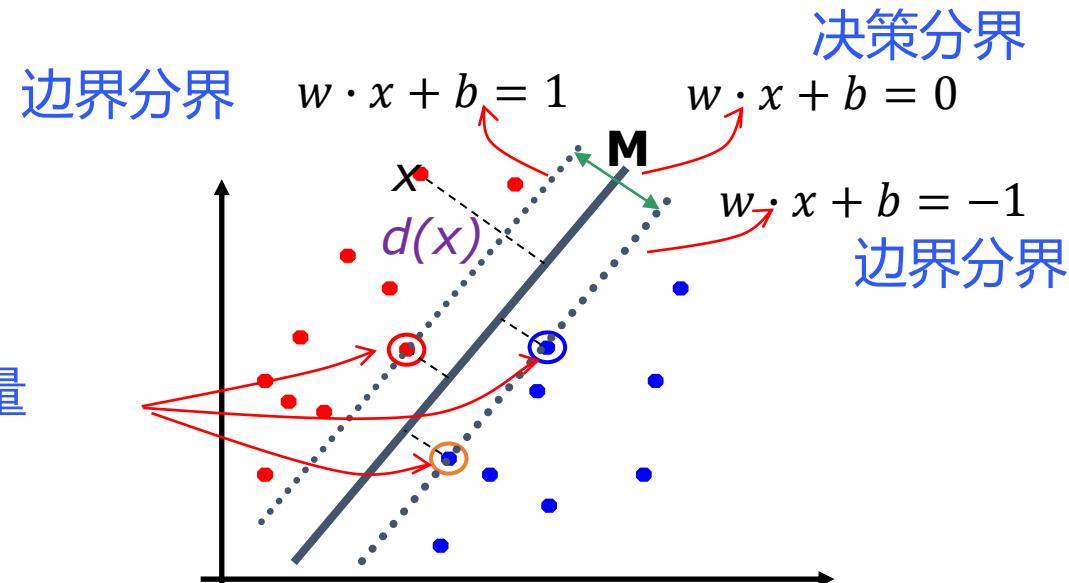
82

□ 支持向量机的基本概念

- 模型：符号函数 $y = \text{sign}(w \cdot x + b) = \begin{cases} +1, & w \cdot x + b \geq 0 \\ -1, & w \cdot x + b < 0 \end{cases}$
- 决策分界面(Decision Boundary): $w \cdot x + b = 0$
- 边界分界面(Margin Boundary) : $w \cdot x + b = \pm 1$
- 支持向量(Support Vectors): 满足 $w \cdot x + b = \pm 1$ 的样本

两个边界之间的距离 $d(x)$

支持向量





分类：支持向量机

83

□ 支持向量机

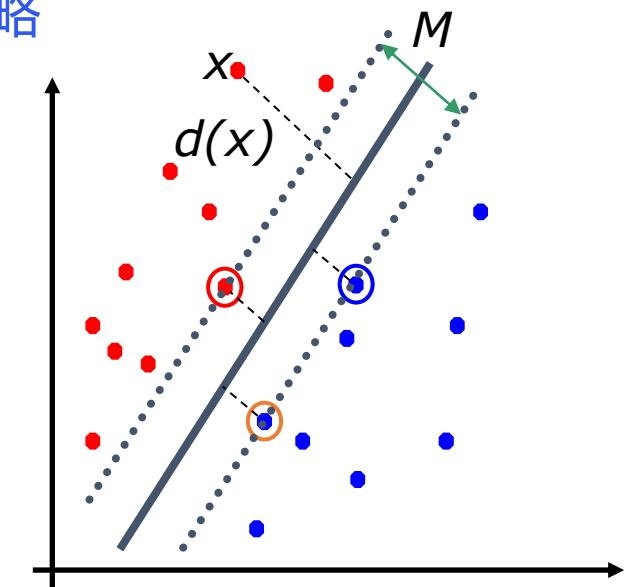
□ 两个边界之间的距离 $d(x)$: 支持向量到决策分界面的距离

- 点 x_i 到平面 $w \cdot x + b = 0$ 的距离为 $\frac{1}{\|w\|} y_i (w \cdot x_i + b)$
- 支持向量到决策平面的距离为 $\frac{1}{\|w\|}$

最大化 ■ 两个边界之间的距离: $\frac{2}{\|w\|}$ 推导过程省略

- 目标函数: $\operatorname{argmax}_{w,b} L(w, b) = \frac{2}{\|w\|}$
- 学习策略: 找到参数 w 和 b , 使得目标最大

$$\operatorname{argmax}_{w,b} L(w, b) = \frac{2}{\|w\|}$$
$$s. t. \quad y_i (w^T \cdot x_i + b) \geq 1, \quad i = 1, 2, \dots, m$$





分类：支持向量机

84

□ 支持向量机学习算法

□ 优化任务转化

$$\underset{w,b}{\operatorname{argmax}} \ L(w,b) = \frac{2}{\|w\|}$$
$$s.t. \quad y_i(w^T \cdot x_i + b) \geq 1, \quad i = 1, 2, \dots, m$$

优化目标：平方和系数都是为了求导方便



等价

$$\underset{w,b}{\operatorname{argmin}} \ L(w,b) = \frac{1}{2} \|w\|^2$$
$$s.t. \quad y_i(w^T \cdot x_i + b) \geq 1, \quad i = 1, 2, \dots, m$$

- (课后学习)注意到，该形式符合凸二次规划问题特征，可以借助拉格朗日对偶性，通过求解对偶问题加以求解
- (课后学习)具体而言，求解方式可采用序列最小优化算法 (SMO)

from sklearn.svm import LinearSVC



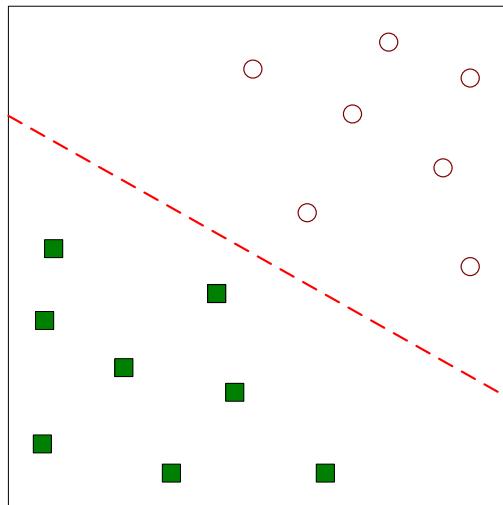
分类：支持向量机

85

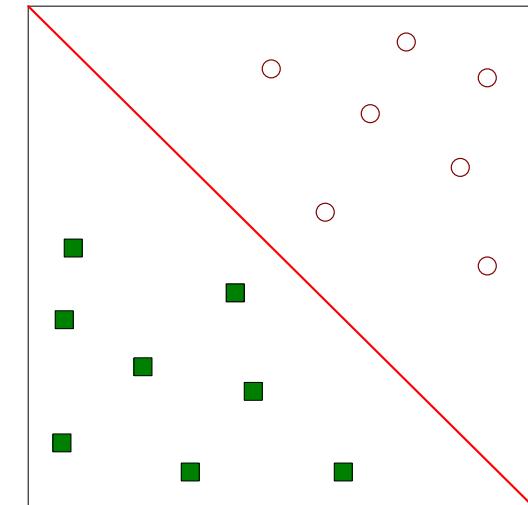
□ 感知机与支持向量机对比

- 相同点：均采用模型 $f(x) = \text{sign}(w \cdot x + b)$
- 不同点：采用不同的优化目标

感知机



SVM



优化目标：

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

$$\begin{aligned} \min_{w,b} L(w, b) &= \frac{1}{2} \|w\|^2 \\ s.t. y_i (w^T x_i + b) &\geq 1 \end{aligned}$$



SVM总结

86

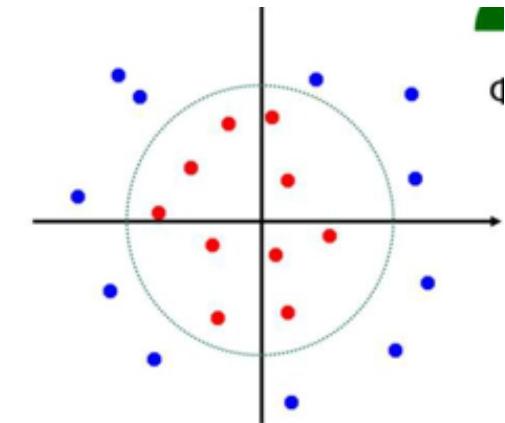
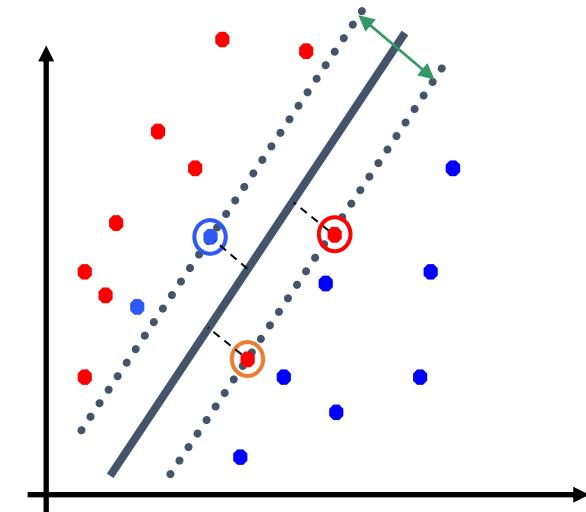
□ 支持向量机

□ 优点：强分类器

- 区分分类结果
- 最大化间隔：更容易区分分类结果
- 有数学理论保证
- 只有支持向量在影响，优化简单

□ 缺点： Hard Margin SVM

- 只能处理线性可分问题
- 线性不可分
 - 软间隔：soft margin SVM (课后学习)
- 线性完全不可分





线性不可分问题

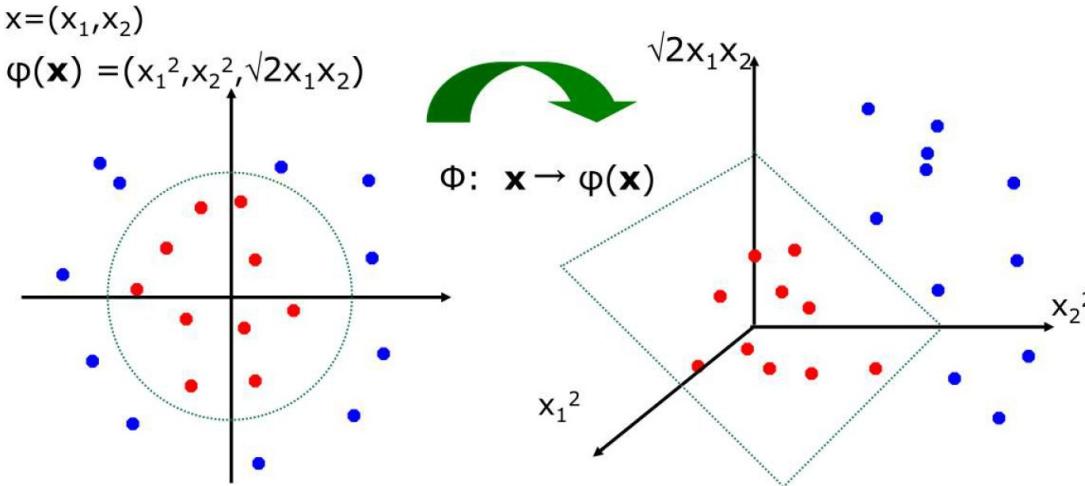
87

□ 线性不可分问题

- 如果数据集线性不可分，不存在这样一个超平面，怎么办？
 - 解决方法：将样本映射到一个更高维的特征空间，使得在这个特征空间线性可分

□ 核函数：

- 核函数的目的，在于将高维空间下的SVM求解时需要的内积运算转化为低维空间下的核函数计算，从而避免可能遇到的“维度灾难”问题





核函数

88

- 线性不可分问题与核函数——（课后学习）
- 常见的核函数如下表所示

名称	表达式	参数
线性核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$	
多项式核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j)^d$	$d \geq 1$ 为多项式的次数
高斯核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\ \mathbf{x}_i - \mathbf{x}_j\ ^2}{2\delta^2}\right)$	$\delta > 0$ 为高斯核的带宽(width)
拉普拉斯核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\ \mathbf{x}_i - \mathbf{x}_j\ }{\delta}\right)$	$\delta > 0$
Sigmoid核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta \mathbf{x}_i^\top \mathbf{x}_j + \theta)$	\tanh 为双曲正切函数, $\beta > 0, \theta < 0$

- 其中，线性核函数与高斯核函数（径向基）是最常用的
- 常见挑选核函数方法一般为以下两种：
 - 穷举法：一个个试过来，选择效果最好的一种
 - 混合法：将多个不同的核函数混合起来使用



分类与预测

89

- 有监督学习：分类与预测
- 常用方法
 - 规则方法
 - 决策树
 - 最近邻方法
 - 感知机，支持向量机（SVM）
 - 集成方法
- 分类的评价指标
- 类不平衡问题



分类：集成学习

90

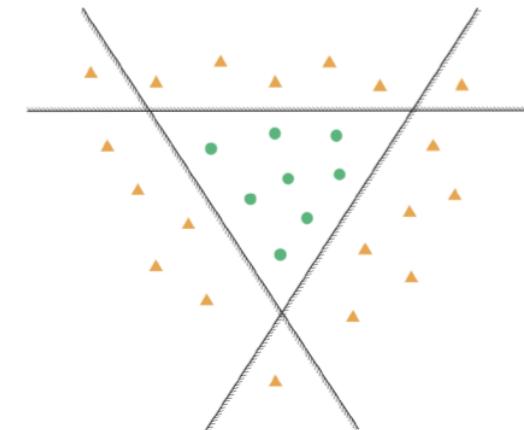
□ 分类——集成学习

- 思想：集成多个模型的能力，得到比单一模型更好的效果
- 为什么能够提升效果？

- 增强模型的表达能力
 - 单个感知机无法正确分类数据能用三个感知机完成
- 降低误差
 - 假设单个分类器误差 p ，有 T 个独立的分类器采用投票进行预测，得到集成模型 H
 - 集成分类器误差为

$$Error_H = \sum_{k \leq \frac{T}{2}} C_T^k \cdot p^{T-k} \cdot (1-p)^k$$

- $T = 5, p = 0.1$ 时， $Error_H < 0.01$



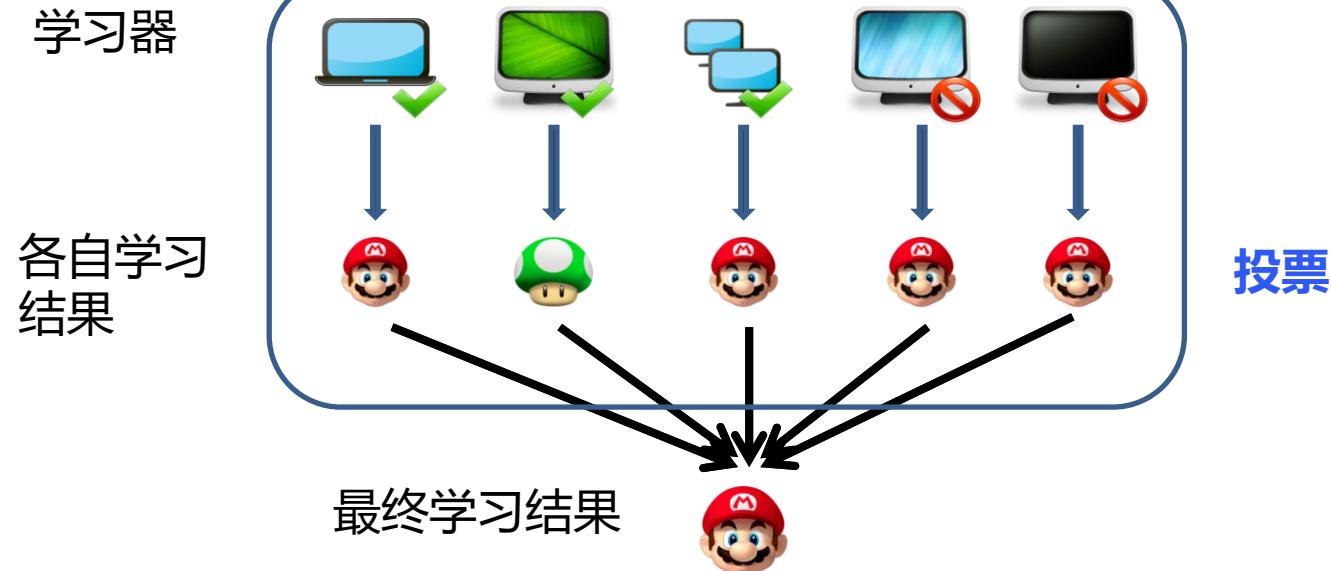


分类：集成学习

91

□ 分类——集成学习

□ 集成过程



- 此类方法经常用在数据挖掘竞赛中(如KDD CUP, CCF-BDCI)
 - KDD Cup2021 MAG240M-LSC赛道第一名集成了30个模型
 - KDD Cup2021 WikiKG90M-LSC赛道第三名集成了15个模型



分类：集成学习

92

□ 常见集成学习方法

□ Bagging (Bootstrap Aggregating)

- 对样本或特征随机取样，学习产生多个独立的模型，然后平均所有模型的预测值
- 主要减小方差
- 典型代表：随机森林

□ Boosting

- 串行训练多个模型，后面的模型是基于前面模型的训练结果（误差）
- 主要减小偏差
- 典型代表： AdaBoost



随机森林

93

□ 集成学习算法：随机森林

- 最典型的Bagging算法：[算法思想](#)
- 随机：每棵树，保证各棵树之间的独立性，采用两到三层的随机性
 - 随机有放回的抽取样本作为训练集
 - 随机选取m个特征作为树节点的划分特征
 - 随机选择特征取值进行分割 (不遍历特征所有取值)
- [样本和特征尽可能不同](#)
- 森林：多颗决策树集成
 - 假设使用三棵决策树组合成随机森林，每各不相同且预测结果相互独立，每棵树的预测错误率为 40%。那么两棵树以及上预测错误的概率下降为：三三棵全部错误+两棵树错误一个正确= $0.4^3 + 3 * 0.4^2 * (1 - 0.4) = 0.352$

[sklearn.ensemble.RandomForestClassifier](#)



AdaBoost

94

□ 集成学习算法：AdaBoost

- 最有代表性的Boosting算法
- 算法思想：利用同一训练样本的不同加权版本，训练一组弱分类器，然后把这些弱分类器以加权的形式集成起来，形成一个最终的强分类器：
 - 在每一步迭代过程中，会给训练集中的样本赋予一个权重 $w_1, w_2 \dots, w_n$
 - 样本的初始权重都一样，设置为 $\frac{1}{n}$ ；
 - 在每一步迭代过程中，
 - 被当前弱分类器分错的样本的权重会相应得到提高
 - 被当前弱分类器分对的样本的权重则会相应降低；
 - 弱分类器的权重则根据当前分类器的加权错误率来确定。

```
from sklearn.ensemble import AdaBoostClassifier
```



分类与预测

95

□ 有监督学习：分类与预测

□ 常用方法

- 规则方法
- 决策树
- 最近邻方法
- 支持向量机 (SVM)
- 集成方法

□ 分类的评价指标



分类模型的评价

96

□ 如何评价分类模型的效果？——以二分类为例

□ 基本概念

- T/F: True or False, 表示二分类结果的正确与否
- P/N: Positive or Negative, 表示算法对样本的判断

□ 四种简写的含义：

- 真正(True Positive, TP): 样本为正例，预测为正，(正确)
- 假负(False Negative, FN): 样本为正例，预测为负，(错误)
- 假正(False Positive, FP): 样本为负例，预测为正，(错误)
- 真负(True Negative, TN): 样本为负例，预测为负，(正确)



分类模型的评价

97

- 如何评价分类模型的效果? —— 指标 Accuracy
- 通常用混淆矩阵 表示: TP、 FN、 FP、 TN

		PREDICTED (预测) CLASS	
		Class=Yes	Class>No
ACTUAL (真实) CLASS	Class=Yes	a (TP)	b (FN)
	Class>No	c (FP)	d (TN)

- 评价指标1: $\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$



分类模型的评价

98

□ 如何评价分类模型的效果？——指标Accuracy

- 举例：假设一共有200个测试数据样本，以下是使用分类模型得到的分类结果，请问Accuracy是多少？

		PREDICTED (预测) CLASS	
ACTUAL (真实) CLASS		Class=Yes	Class>No
	Class=Yes	60 (TP)	40 (FN)
	Class>No	20 (FP)	80 (TN)

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{60 + 80}{60 + 80 + 20 + 40} = 0.7$$



分类模型的评价

99

□ 如何评价分类模型的效果？— Accuracy的局限性

□ 样本不均衡时，评估结果可能不合理

□ 例子：考虑2分类问题

- 假设10000个数据样本中，类别0的样本数为 9990，类别1的样本数为 10

- 假设模型将所有样本均预测为0

- 计算 $\text{Accuracy} = 9990/10000 = 99.9\%$

Accuracy很高，
表明模型很好？

结论：模型不好

原因：10个正例均预测错误

分析：这个例子中，显然我们关注类别为1的样本，但Accuracy被类别为0的样本影响了

ACTUAL CLASS	Count	PREDICTED CLASS	
		Class=Yes	Class>No
Class=Yes	a	b	
Class>No	c	d	



分类模型的评价

100

□ 如何评价分类模型的效果？——分类问题的常用指标

- 准确率(查准率) $Precision(p) = \frac{a}{a+c} = \frac{TP}{TP+FP}$ ：预测为Yes中正确的比例
 - 正确预测的个体总数 / 预测出的个体总数
- 召回率(查全率) $Recall(r) = \frac{a}{a+b} = \frac{TP}{TP+FN}$ ：真实为Yes中被预测正确的比例
 - 正确预测的个体总数 / 测试集中Yes的个体总数

Count		PREDICTED CLASS	
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	a (TP)	b (FN)
	Class>No	c (FP)	d (TN)



分类模型的评价

101

□ 分类模型的其他指标——分类问题的常用指标

□ $F\text{值} = \frac{2PR}{P+R} = \frac{2a}{2a+b+c}$: 正确率和召回率的调和平均值

□ 意义：不同应用场景中，对于准确率和召回率有着不同的侧重

- 邮件分类：宁愿放过一些垃圾邮件，也不能错杀正常邮件
 - 牺牲召回率，保证较高准确率
- 智慧医疗：宁愿多判断一些疑似患者，不能漏掉一个病人
 - 牺牲准确率，保证较高召回率

ACTUAL CLASS	Count	PREDICTED CLASS	
		Class=Yes	Class>No
		a	b
		c	d



分类模型的评价

102

□ 分类问题的常用指标——课堂练习

- 在一次垃圾邮件检测中，使用某分类模型认为有100篇邮件是垃圾邮件，后经过专家判定，其中真是垃圾邮件的为60篇，其余的40篇为误分类，那么请问本次分类的准确率Precision就等于_____。
- 假如专家发现邮件样本集里还有90篇垃圾邮件，由于各种原因而未被检出（漏检），那么按照上述公式，本次分类的查全率Recall就等于_____，F1值等于_____。

$$\text{Precision}(P) = \frac{a}{a+c}$$

$$\text{Recall}(R) = \frac{a}{a+b}$$

$$F\text{值} = \frac{2PR}{P+R} = \frac{2a}{2a+b+c}$$

		PREDICTED CLASS	
		Class=Yes	Class>No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class>No	c (FP)	d (TN)



分类模型的评价

103

□ 分类模型的其他指标—ROC与AUC

□ ROC(Receiver Operating Characteristic)与AUC(Area Under the ROC curve)

- 背景：发展于20世纪50年代的信号检测理论，用于分析噪声信号
- 两者的关系：ROC曲线的面积就是AUC

□ 基本概念

■ 真正例率TPR(True Positive Rate) = $TP/(TP+FN)$

- 预测为正且实际为正的样本占所有正样本的比例

■ 假正例率FPR(False Positive Rate)= $FP/(TN+FP)$

- 预测为正但实际为负的样本占所有负样本的比例

		PREDICTED CLASS	
		Class=Yes	Class>No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class>No	c (FP)	d (TN)



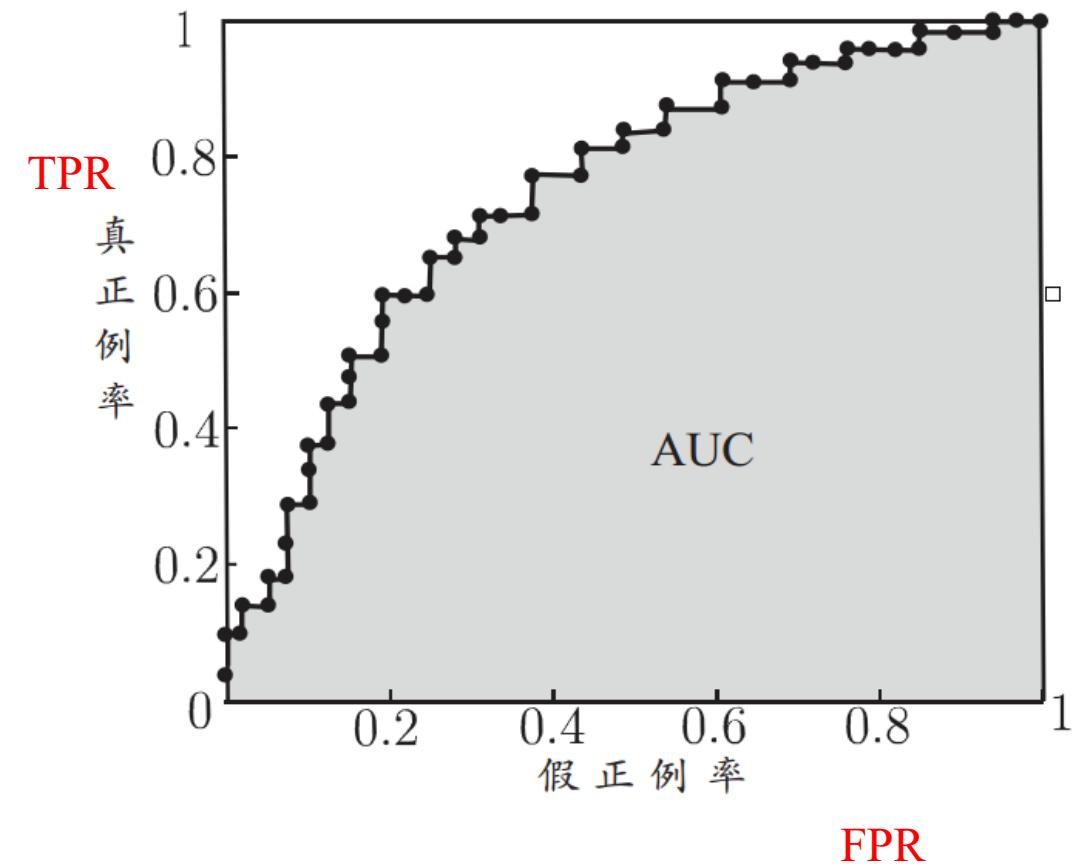
分类模型的评价

104

□ ROC曲线

特点：

- 对角线表示区分能力为0，即随机猜测
- 在对角线上端越远，效果越好
- 低于对角线的结果无意义（无区分度）



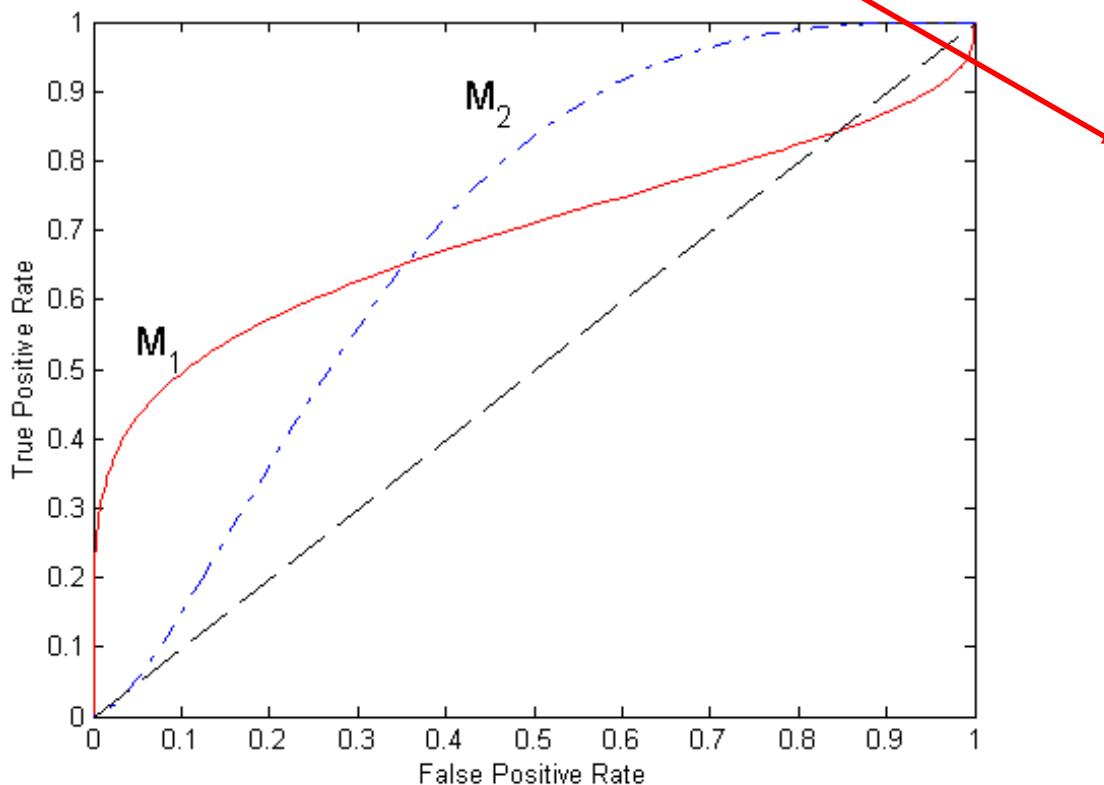


分类模型的评价

105

□ 如何基于ROC曲线比较模型好坏？

- 一般而言，模型A的ROC曲线将模型B的完全包围，则模型A更好
- 但往往并不会完全包围



对比ROC曲线发现，两个模型都不是一直表现得好

- M_1 在FPR较小时表现好
- M_2 在FPR较大时表现好



分类模型的评价

106

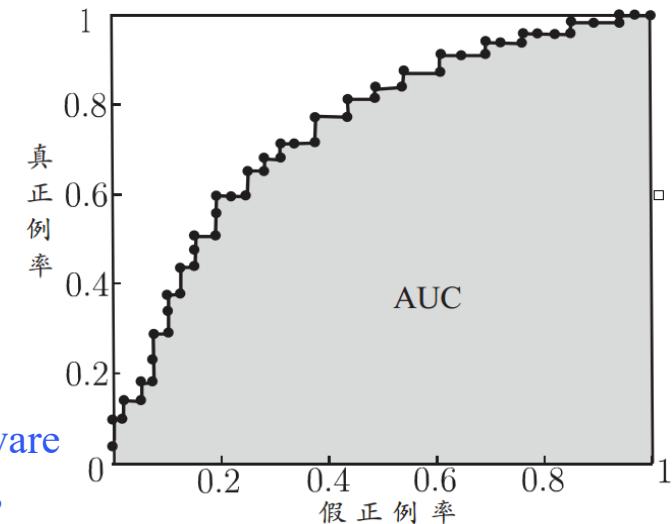
□ ROC不能量化——AUC量化指标

- AUC定义为ROC曲线的面积，可以直接计算如下
- 假设ROC曲线由 $\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m), x_1 = 0, x_m = 1\}$ 的点按序连接而形成，则AUC为：

$$AUC = \frac{1}{2} \sum_{i=1}^{m-1} (x_{i+1} - x_i) \cdot (y_{i+1} + y_i)$$

AUC越大，结果越好

AUC衡量了样本预测的排序质量





分类模型的评价

107

□ 如何评价分类模型的效果？一分类模型的比较方法

□ 关于性能比较：

- 测试性能并不等于泛化性能
- 测试性能随着测试集的变化而变化
- 很多机器学习算法本身有一定的随机性

■ 例如，对两个模型：

- M1: accuracy = 85%, 在30个样本上测试
- M2: accuracy = 75%, 在5000个样本上测试

□ 常常进行**假设检验**，判断不同模型的性能差别是否具有统计意义

- 假设检验为学习器性能比较提供了重要依据，基于其结果我们可以推断出：若在测试集上观察到学习器A比B好，则A的泛化性能是否在统计意义上优于B，以及这个结论的把握有多大。

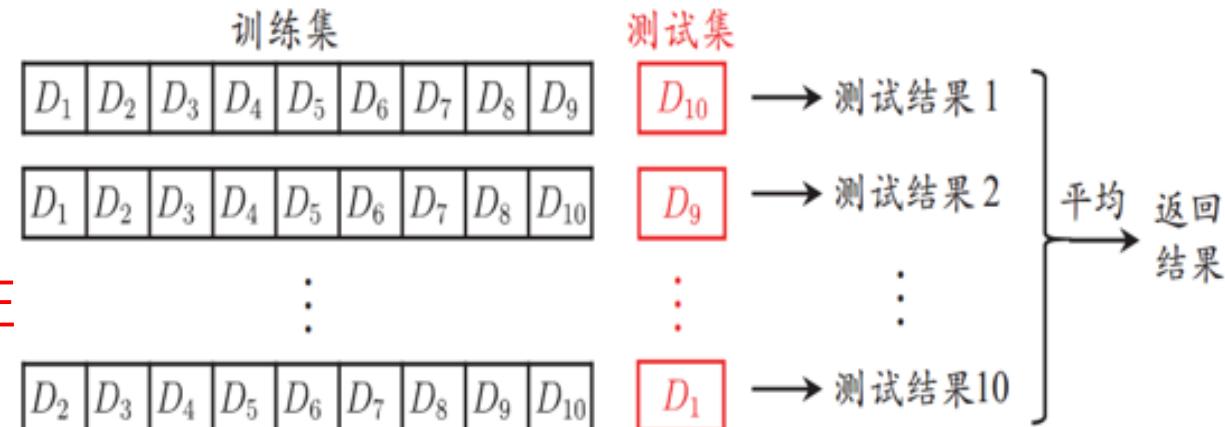
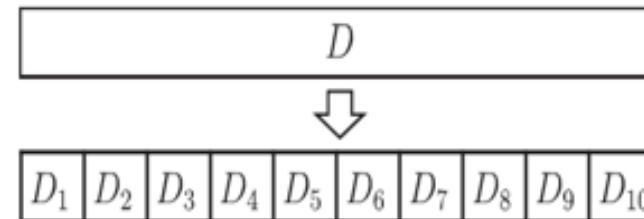


分类模型的验证

108

□ 分类模型的验证方法 —— 增加结果的可靠性

- 交叉验证法 (Cross Validation) : 将数据集分层采样划分为k个大小相似的互斥子集，每次用 $k-1$ 个子集的并集作为训练集，余下的子集作为测试集，最终返回k个测试结果的均值， k 最常用的取值是10.





分类与预测

109

□ 有监督学习：分类与预测

□ 常用方法

- 规则方法
- 决策树
- 最近邻方法
- 支持向量机 (SVM)
- 集成方法

□ 分类的评价指标