BUSS 3620.人工智能导论

机器学习

刘佳璐

安泰经济与管理学院

上海交通大学

此前 vs 之后

此前

- 有关于如何执行任务的详细步骤
- 无需外界的数据
- 按照步骤完成任务

之后

- 没有关于如何执行任务的明确说明
- 以数据的形式提供信息
- AI自身决定如何完成任务

BUSS 3620.人工智能导论

#1. 监督学习

刘佳璐

安泰经济与管理学院

上海交通大学

监督学习 Supervised learning

- 基于一个包含输入-输出对的数据集，学习一个将输入映射到输出的函数

AI任务	输入	输出
垃圾邮件过滤	电子邮件	垃圾邮件(0/1)
语音识别	音频	具体的文字
机器翻译	英语	中文
广告投放	广告/用户信息	是否点击(0/1)
自动驾驶	图像/雷达信息	汽车的位置
图片检查	图片	是否缺陷(0/1)

分类 Classification

- 学习将输入点映射到离散类别的函数的监督学习任务
- 输出：离散类别

下雨

不下雨

数据 Data

	输入		输出/标签
日期	湿度 (相对湿度)	气压 (mb)	是否下雨
1 月 1 日	93%	999.7	下雨
1 月 2 日	49%	1015.5	不下雨
1 月 3 日	79%	1031.1	不下雨
1 月 4 日	65%	984.9	下雨
1 月 5 日	90%	975.2	下雨

基于一些**输入**（湿度、气压）的数据，计算机能否预测明天的天气应该与什么**标签**相关联？

更数学化的表示

f(湿度, 气压)

真实的函数关系是未知的、隐藏的

$$f(93, 999.7) = \text{下雨}$$

$$f(49, 1015.5) = \text{不下雨}$$

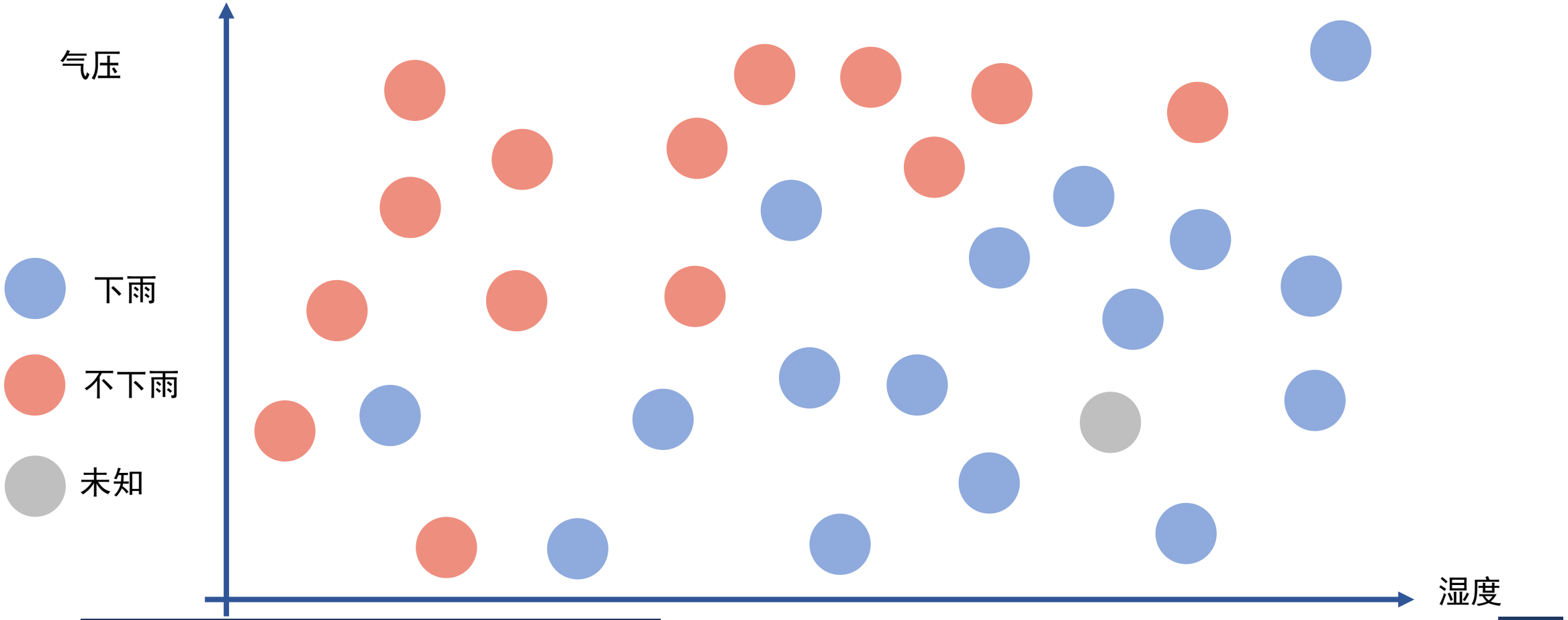
$$f(79, 1031.1) = \text{不下雨}$$

h(湿度, 气压)

假设的函数关系 Hypothesis function

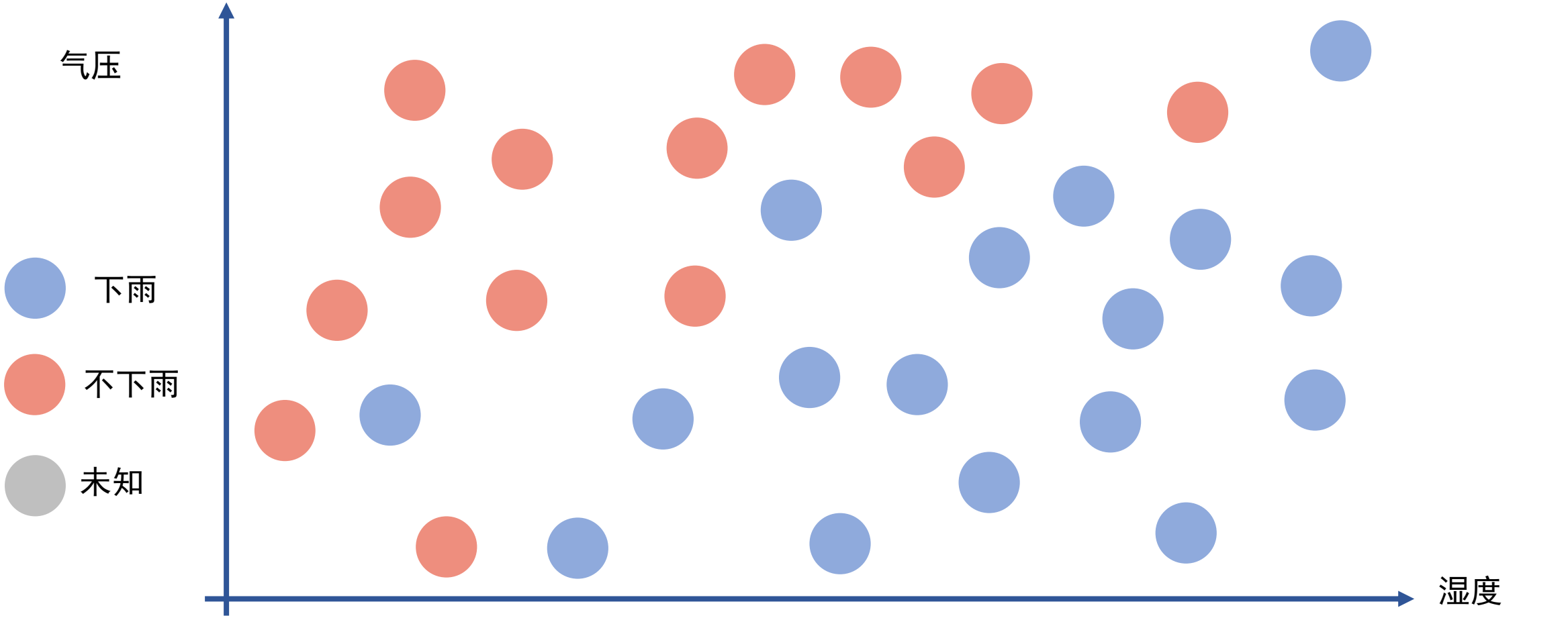
分类任务：这个 $h()$ 是什么样子的?

从哪里开始呢?



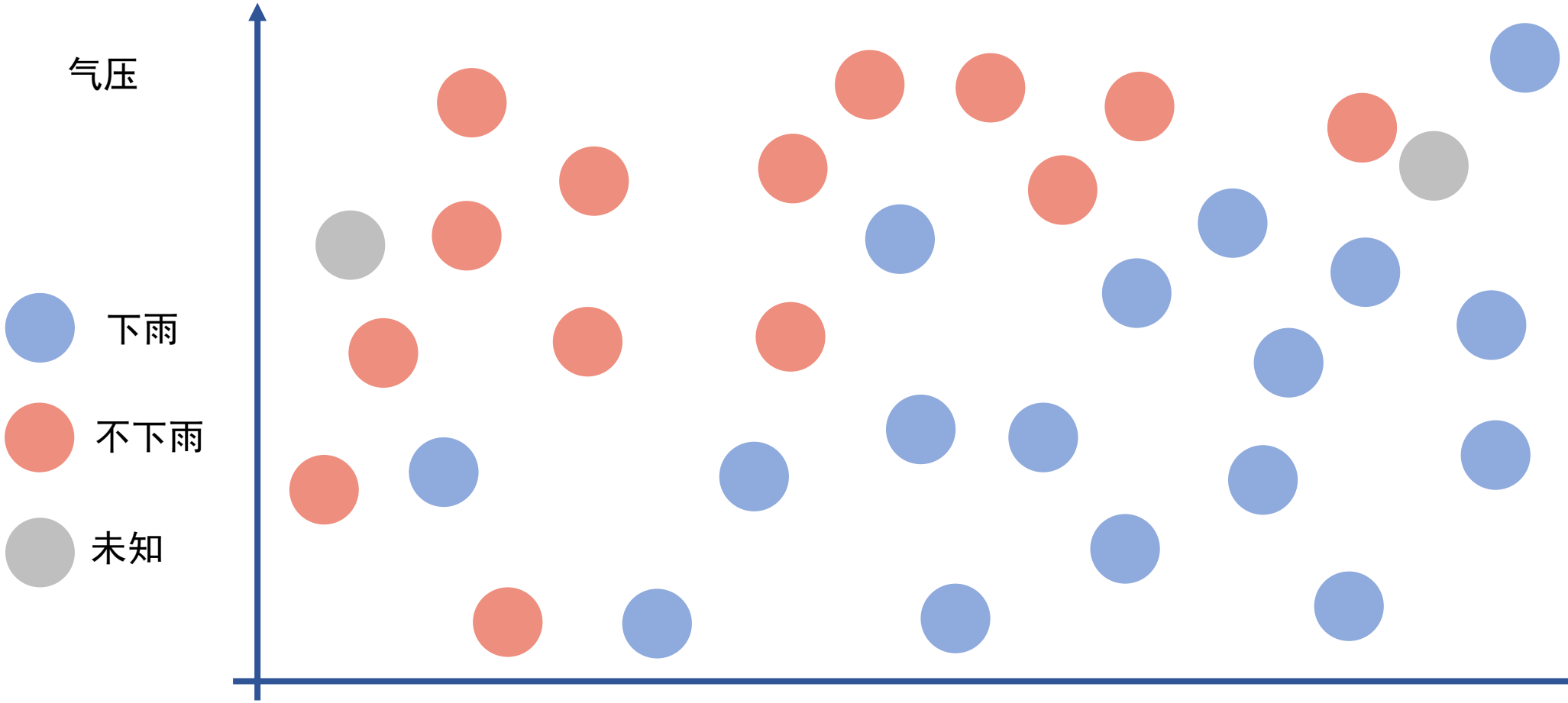
近邻法分类 Nearest – neighbor classification

- 将被识别样本分类为**离他最近的数据点**的类别的算法



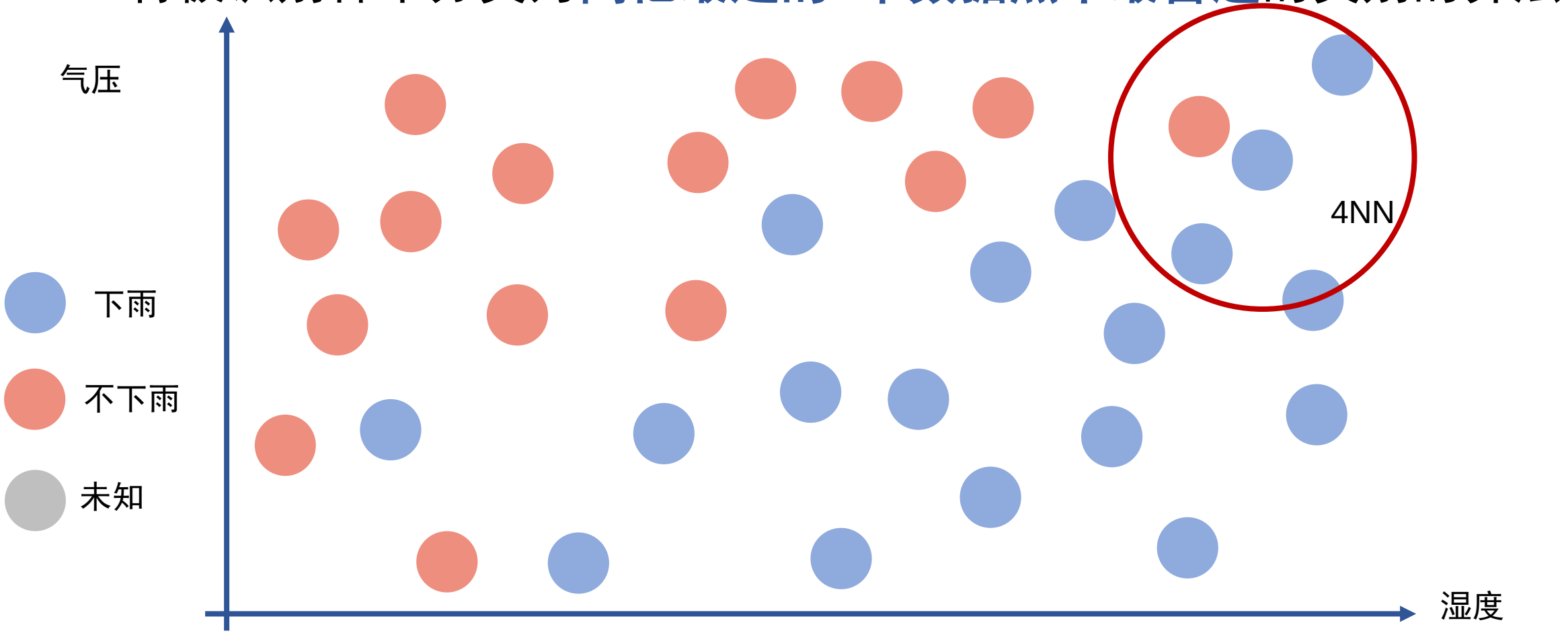
练习 #1

- 使用近邻法对两个灰色的点进行分类



k-最近邻算法 k – nearest – neighbor classification

- 将被识别样本分类为**离他最近的*k*个数据点中最普遍**的类别的算法



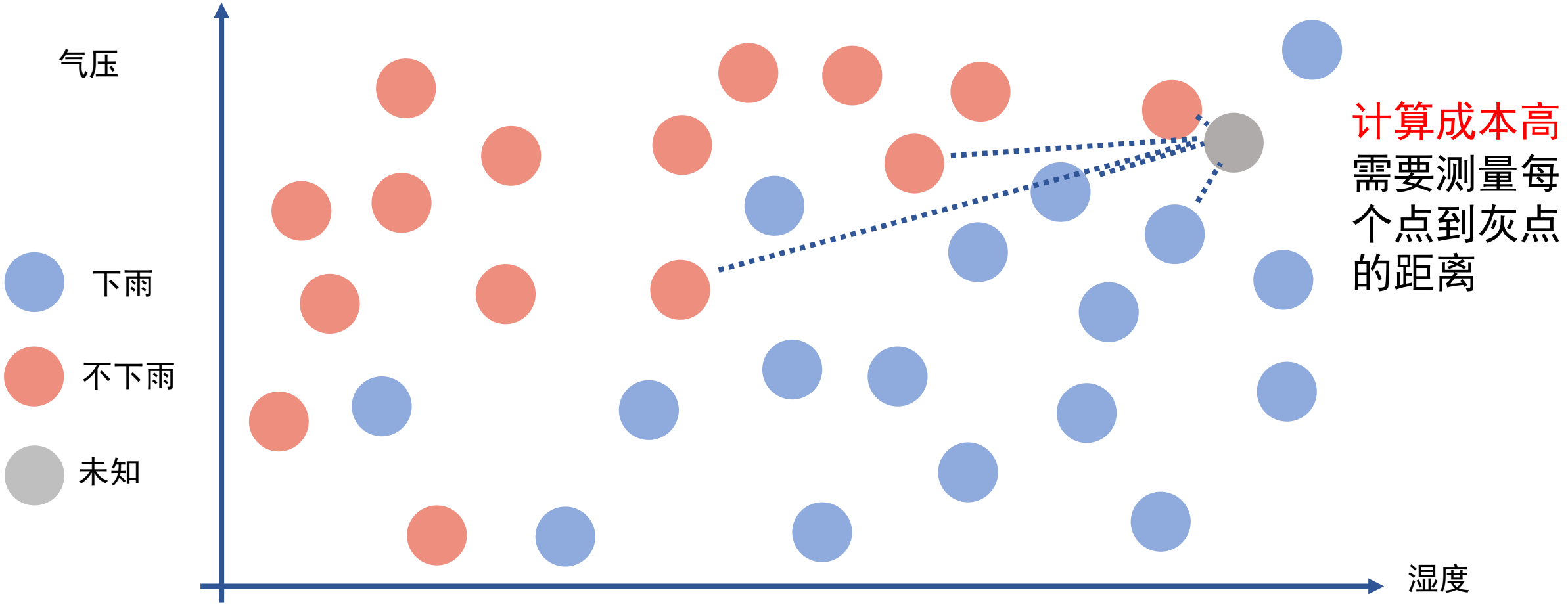
练习 #2

- 根据下面的数据，使用3NN对新纸巾样本进行分类
  - 新样本：耐酸性=3，强度=7
  - 距离使用曼哈顿距离$d = |x_1 - x_2| + |y_1 - y_2|$

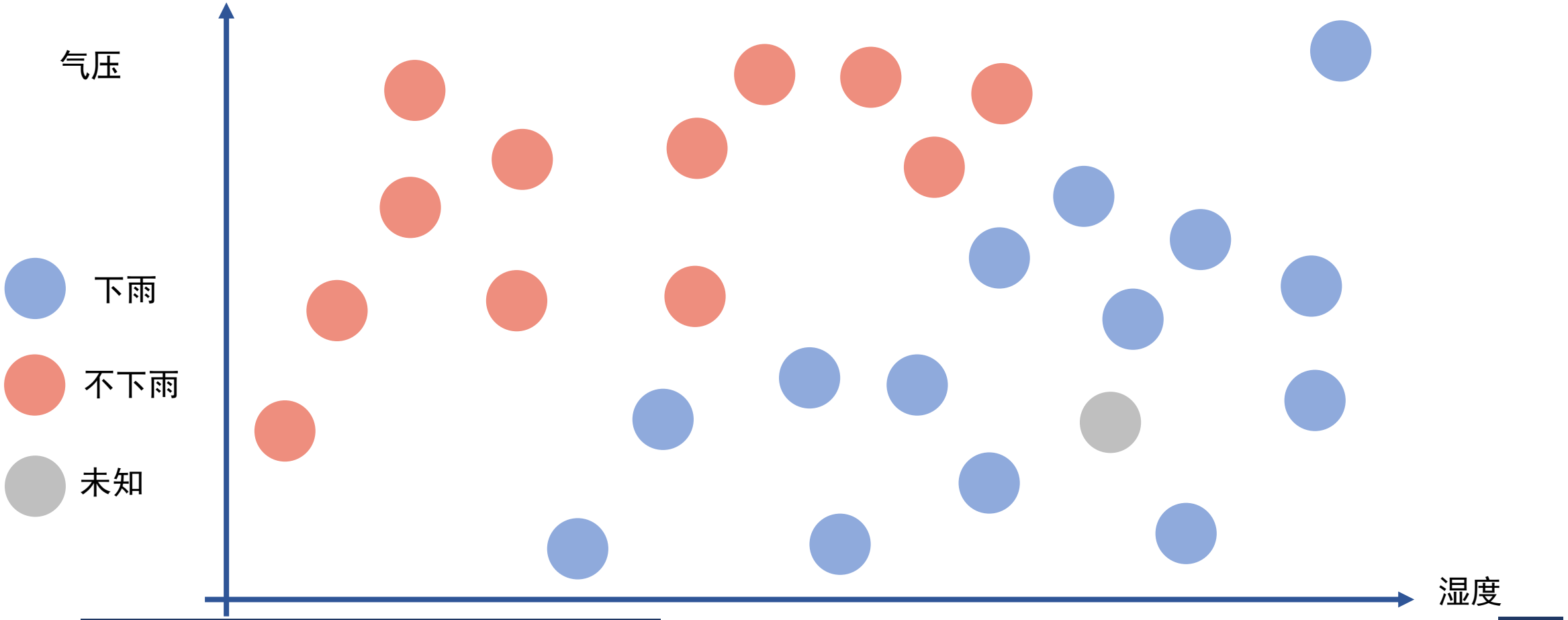
耐酸性(秒)	强度(kg/m²)	纸巾好坏
7	7	坏
7	4	坏
6	7	坏
3	5	好
3	4	好
1	4	好

k-最近邻算法 k – nearest – neighbor classification

- 将被识别样本分类为**离他最近的*k*个数据点中最普遍**的类别的算法

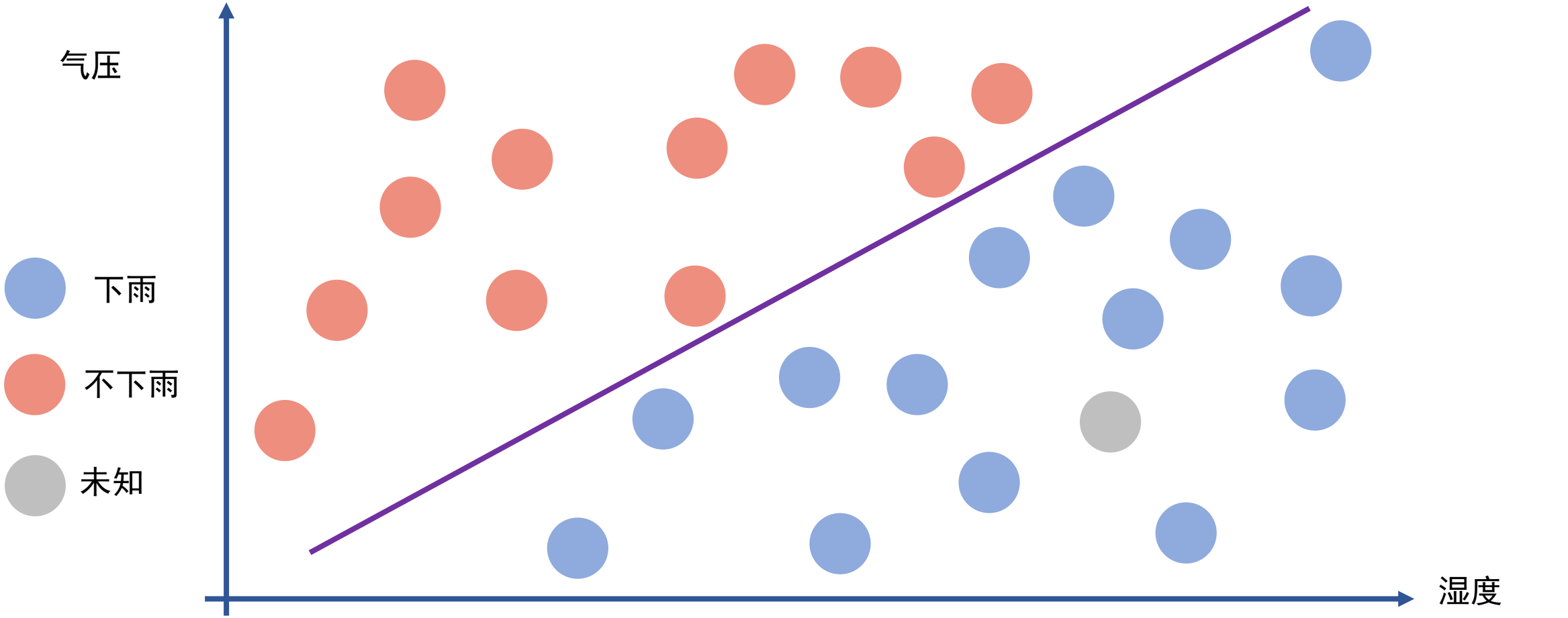


除近邻法之外，还可以?



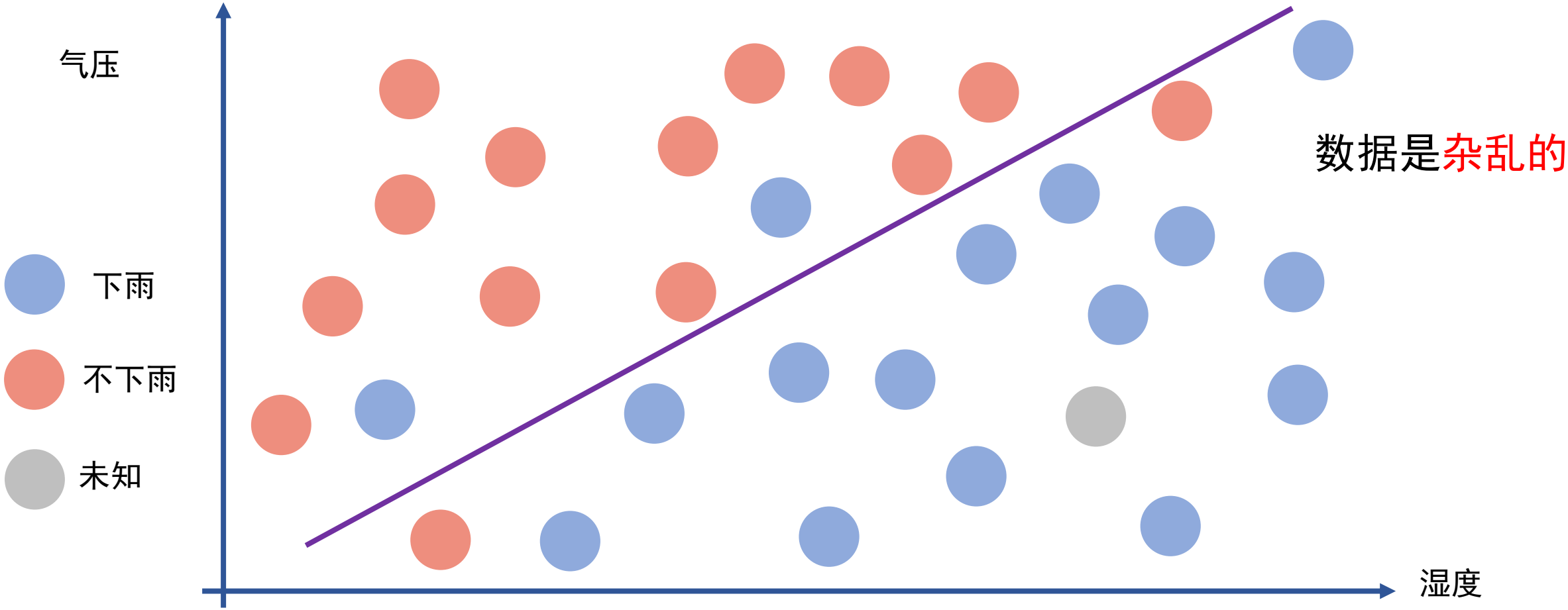
画出决策边界 decision boundary

- 感知机模型 Perceptron Model / 线性回归 Linear Regression



画出决策边界 decision boundary

- 感知机模型 Perceptron Model / 线性回归 Linear Regression

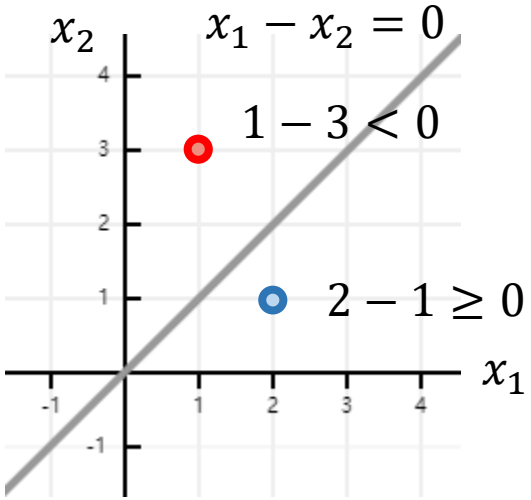


感知机模型 / 线性回归

$x_1 =$ 湿度　　$x_2 =$ 气压

$$h(x_1, x_2) = \begin{cases} \text{下雨 if } w_0 + w_1 x_1 + w_2 x_2 \geq 0 \\ \text{不下雨 if } w_0 + w_1 x_1 + w_2 x_2 < 0 \end{cases}$$

$$h(x_1, x_2) = \begin{cases} 1 \text{ if } w_0 + w_1 x_1 + w_2 x_2 \geq 0 \\ 0 \text{ if } w_0 + w_1 x_1 + w_2 x_2 < 0 \end{cases}$$



输入向量 Input vector **x**: $(1, x_1, x_2)$

权重向量 Weight vector **w**: (w_0, w_1, w_2)

w·**x**: $w_0 + w_1 x_1 + w_2 x_2$

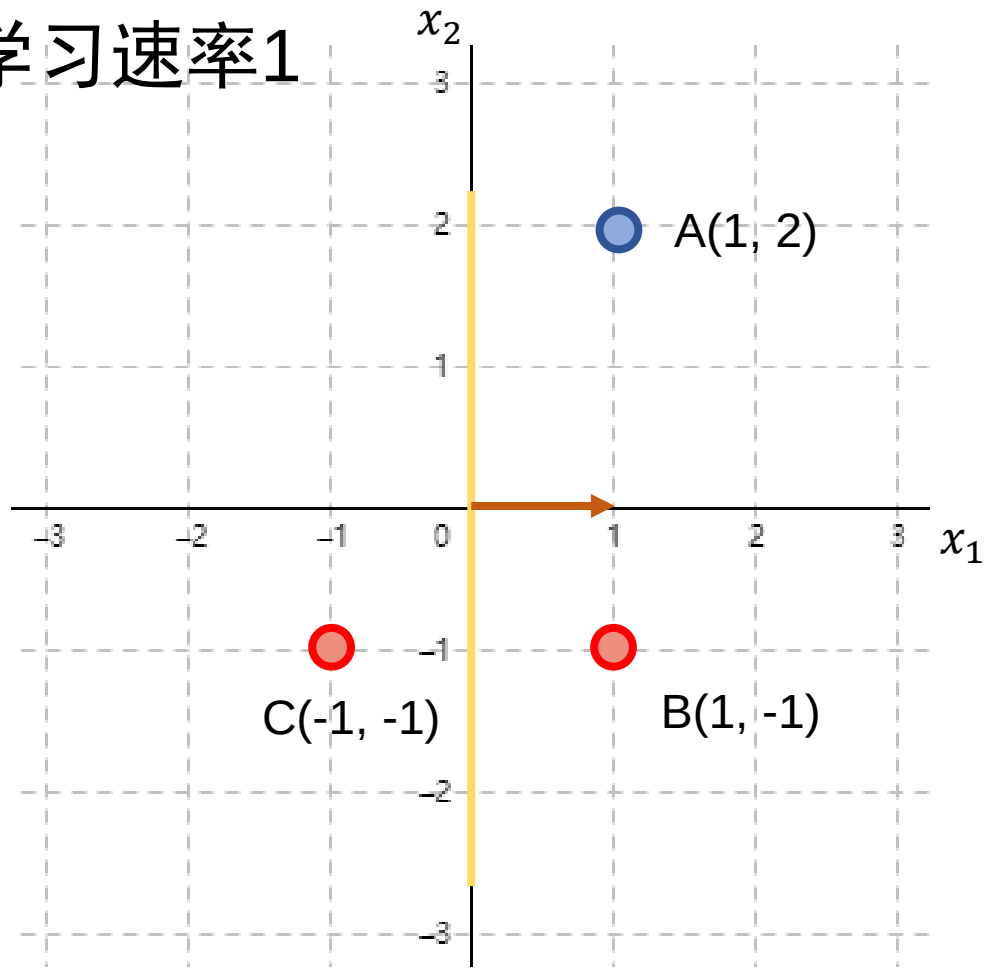
$$h_{\mathbf{w}}(\mathbf{x}) = \begin{cases} 1 \text{ if } \mathbf{w} \cdot \mathbf{x} \geq 0 \\ 0 \text{ if } \mathbf{w} \cdot \mathbf{x} < 0 \end{cases}$$

感知机模型的学习规则

- 给定数据点$(\mathbf{x}, \mathrm{y})$,根据以下方式更新权重:
  - $w_i = w_i + \alpha(y - h_w(\mathbf{x})) \times x_i$
  - $w_i = w_i + \alpha(\text{真实值} - \text{估计值}) \times x_i$
    - 如果 真实值 – 估计值 $= 0$,不更新
    - 如果 真实值 – 估计值 > 0,增加w_i
    - 如果 真实值 – 估计值 < 0,减少w_i
    - α:学习速率 learning rate – AI 更新权重的值的速度有多快

感知机模型学习示例

- 数据点A，B，C，蓝色1，红色0, 学习速率1
- 初始决策边界$1 \times x_1 + 0 \times x_2 = 0$
  - 初始权重向量$(1, 0)$
- 学习第一个点A
  - 估计值：$1 \times 1 + 0 \times 2 > 0 \Rightarrow 1$
  - 分类正确，不更新

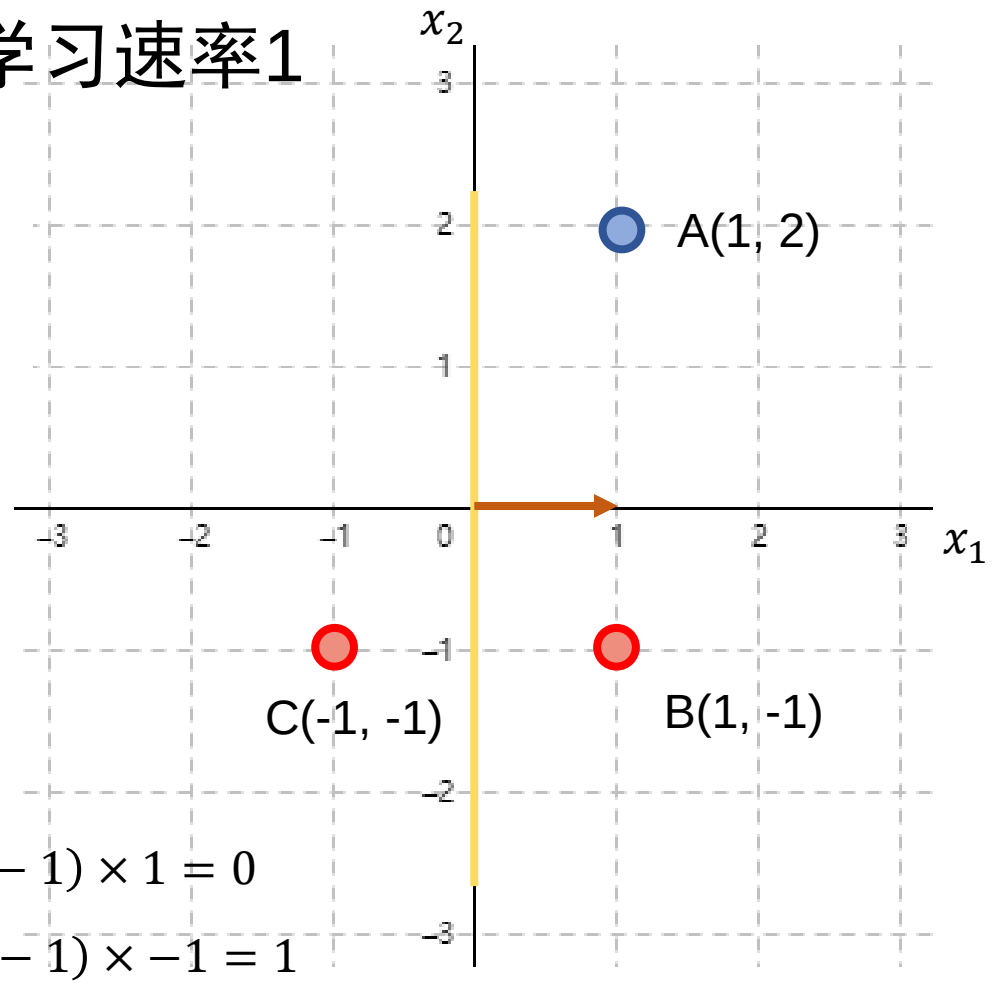


感知机模型学习示例

- 数据点A，B，C，蓝色1，红色0, 学习速率1
- 新决策边界$0 \times x_1 + 1 \times x_2 = 0$
  - 新的权重向量$(0, 1)$
- 学习第二个点B
  - 估计值：$1 \times 1 + 0 \times -1 > 0 \Rightarrow 1$
  - 分类不正确，更新

$$w_1 = w_1 + \alpha(\text{真实值} - \text{估计值}) \times x_1 = 1 + 1 \times (0 - 1) \times 1 = 0$$

$$w_2 = w_2 + \alpha(\text{真实值} - \text{估计值}) \times x_2 = 0 + 1 \times (0 - 1) \times -1 = 1$$

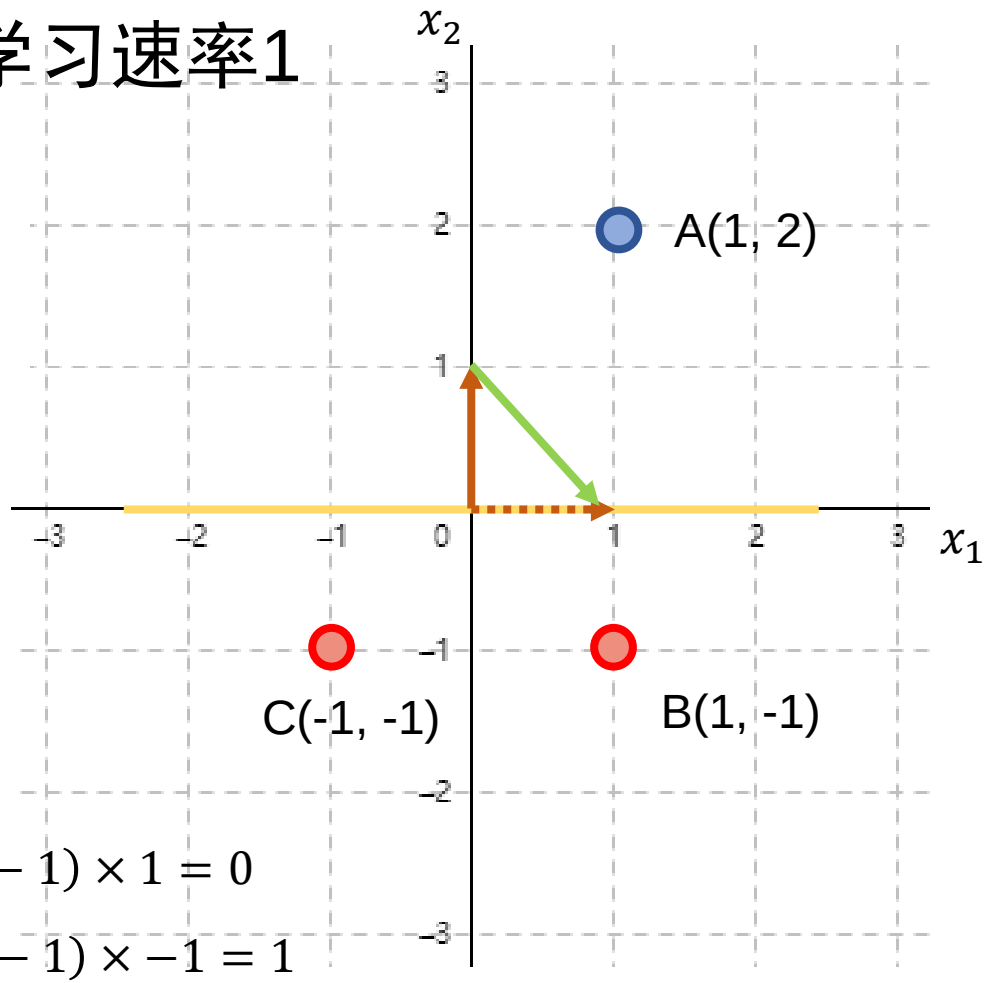


感知机模型学习示例

- 数据点A，B，C，蓝色1，红色0, 学习速率1
- 新决策边界$0 \times x_1 + 1 \times x_2 = 0$
  - 新的权重向量$(0, 1)$
- 学习第二个点B
  - 估计值：$1 \times 1 + 0 \times -1 > 0 \Rightarrow 1$
  - 分类不正确，更新

$$w_1 = w_1 + \alpha(\text{真实值} - \text{估计值}) \times x_1 = 1 + 1 \times (0 - 1) \times 1 = 0$$

$$w_2 = w_2 + \alpha(\text{真实值} - \text{估计值}) \times x_2 = 0 + 1 \times (0 - 1) \times -1 = 1$$

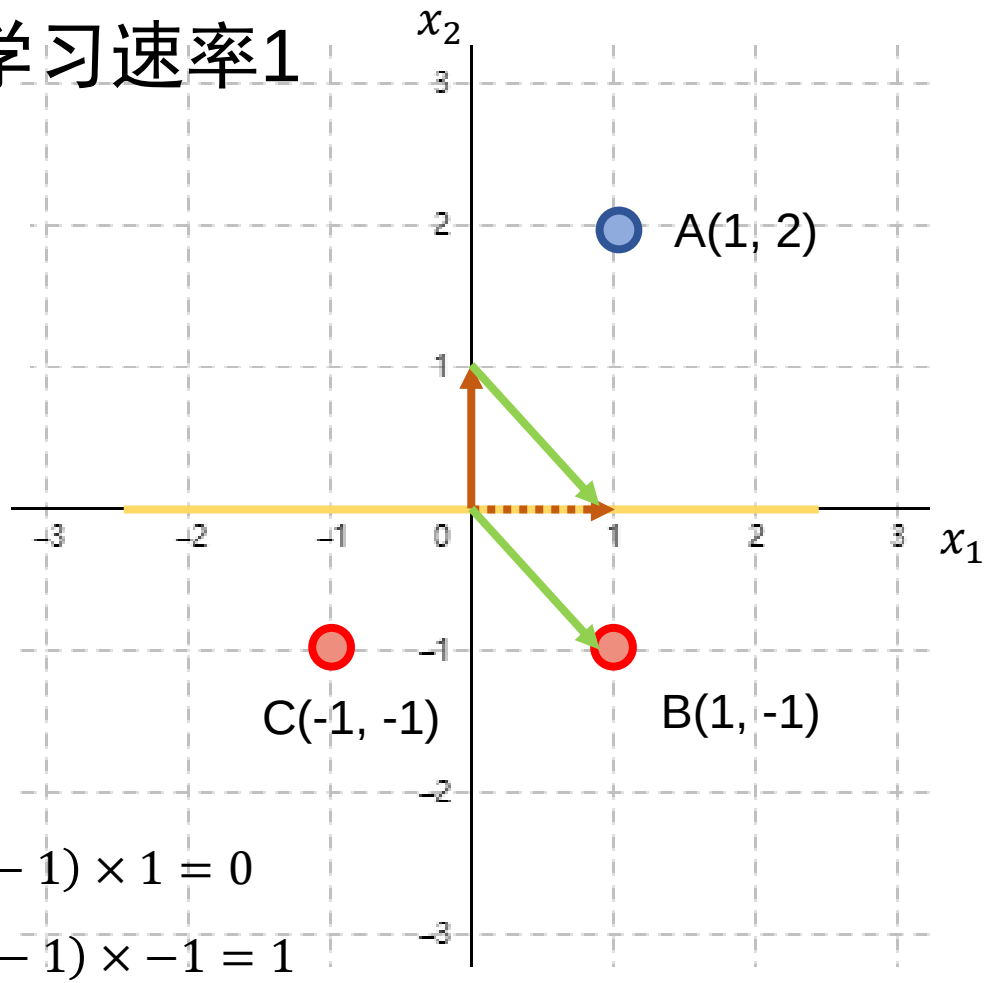


感知机模型学习示例

- 数据点A，B，C，蓝色1，红色0, 学习速率1
- 新决策边界$0 \times x_1 + 1 \times x_2 = 0$
  - 新的权重向量$(0, 1)$
- 学习第二个点B
  - 估计值：$1 \times 1 + 0 \times -1 > 0 \Rightarrow 1$
  - 分类不正确，更新

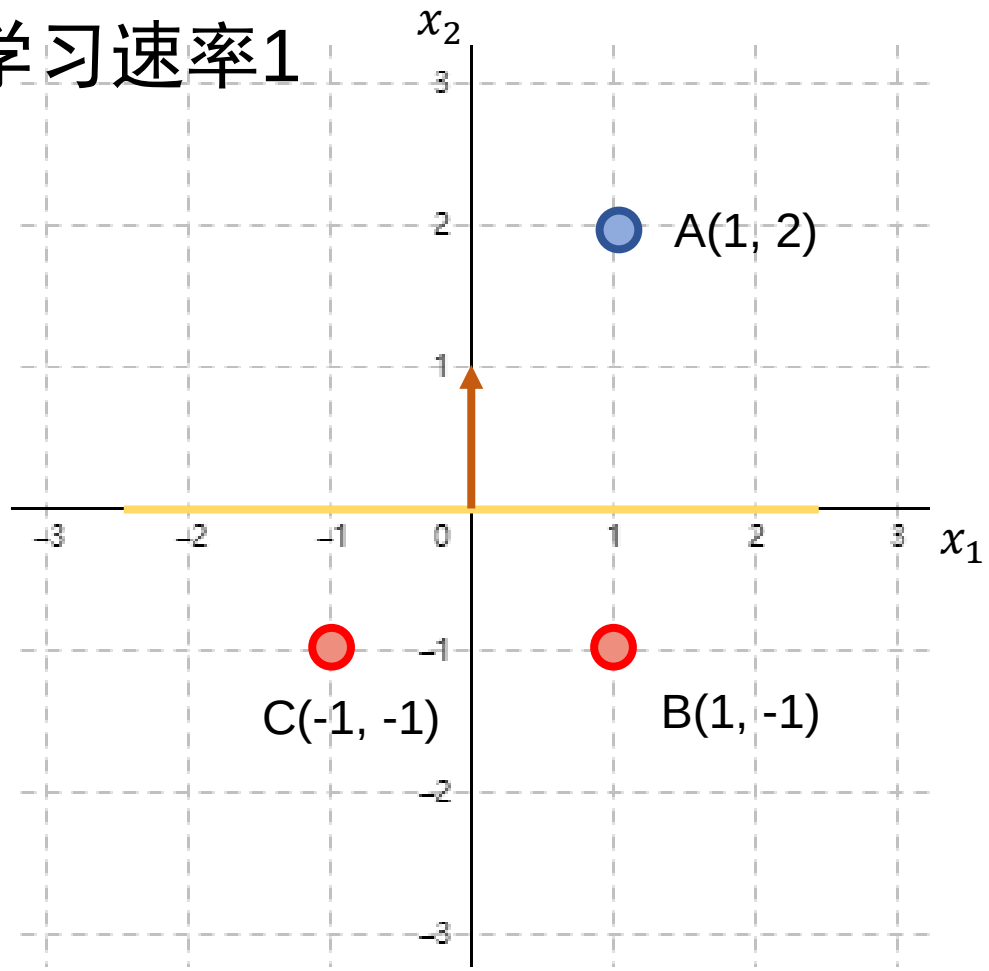
$$w_1 = w_1 + \alpha(\text{真实值} - \text{估计值}) \times x_1 = 1 + 1 \times (0 - 1) \times 1 = 0$$

$$w_2 = w_2 + \alpha(\text{真实值} - \text{估计值}) \times x_2 = 0 + 1 \times (0 - 1) \times -1 = 1$$

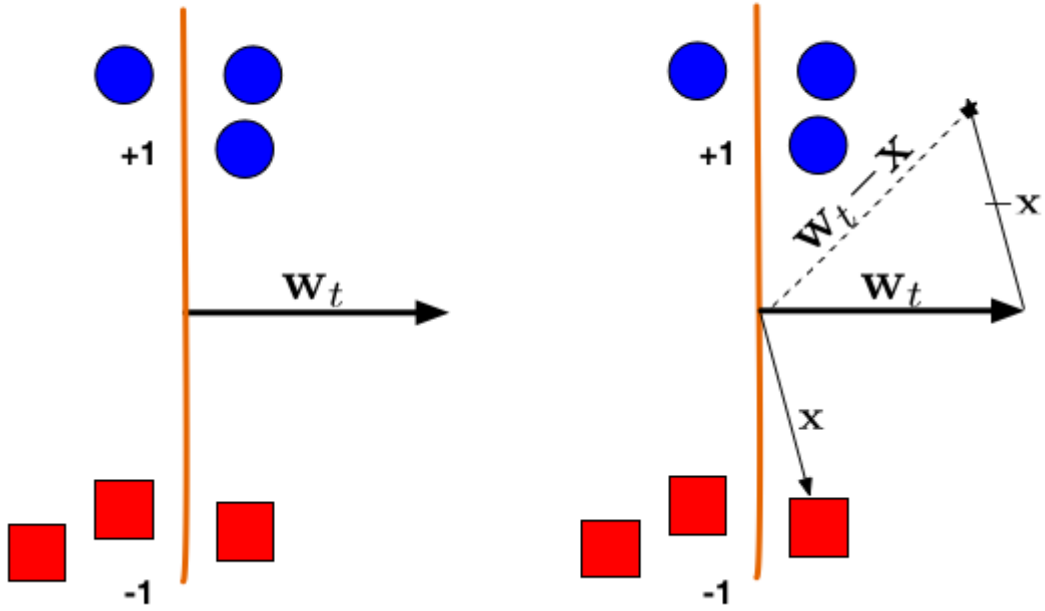


感知机模型学习示例

- 数据点A，B，C，蓝色1，红色0, 学习速率1
- 新决策边界$0 \times x_1 + 1 \times x_2 = 0$
  - 新的权重向量$(0, 1)$
- 学习第三个点C
  - 估计值：$0 \times -1 + 1 \times -1 < 0 \Rightarrow 0$
  - 分类正确，不更新



感知机模型学习图示



来源：Cornell CS4780

练习 #3

- 用感知机模型学习右侧的数据，权重为？
  - 输入向量**x**: $(1, x_1, x_2, x_3)$
  - 权重向量**w**: (w_0, w_1, w_2, w_3), 初始权重向量为$(1, 0, 0, 0)$
  - 学习速率 $\alpha = 1$　　学习公式：$w_i = w_i + \alpha(y - h_w(\mathbf{x})) \times x_i$

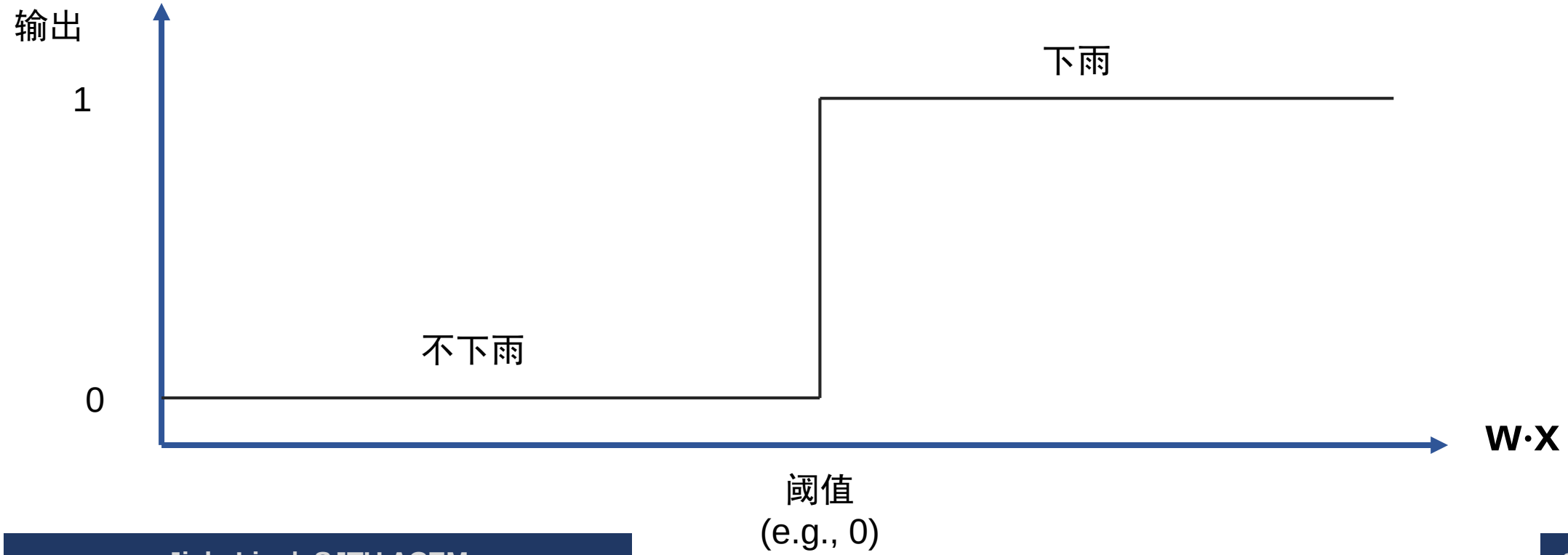
x_1	x_2	x_3	y
4	3	6	0
2	-2	3	1
1	1	0	0

步骤	w_0	w_1	w_2	w_3	w·x	分类正确？
0	1	0	0	0		
1						
2						
3					--	--

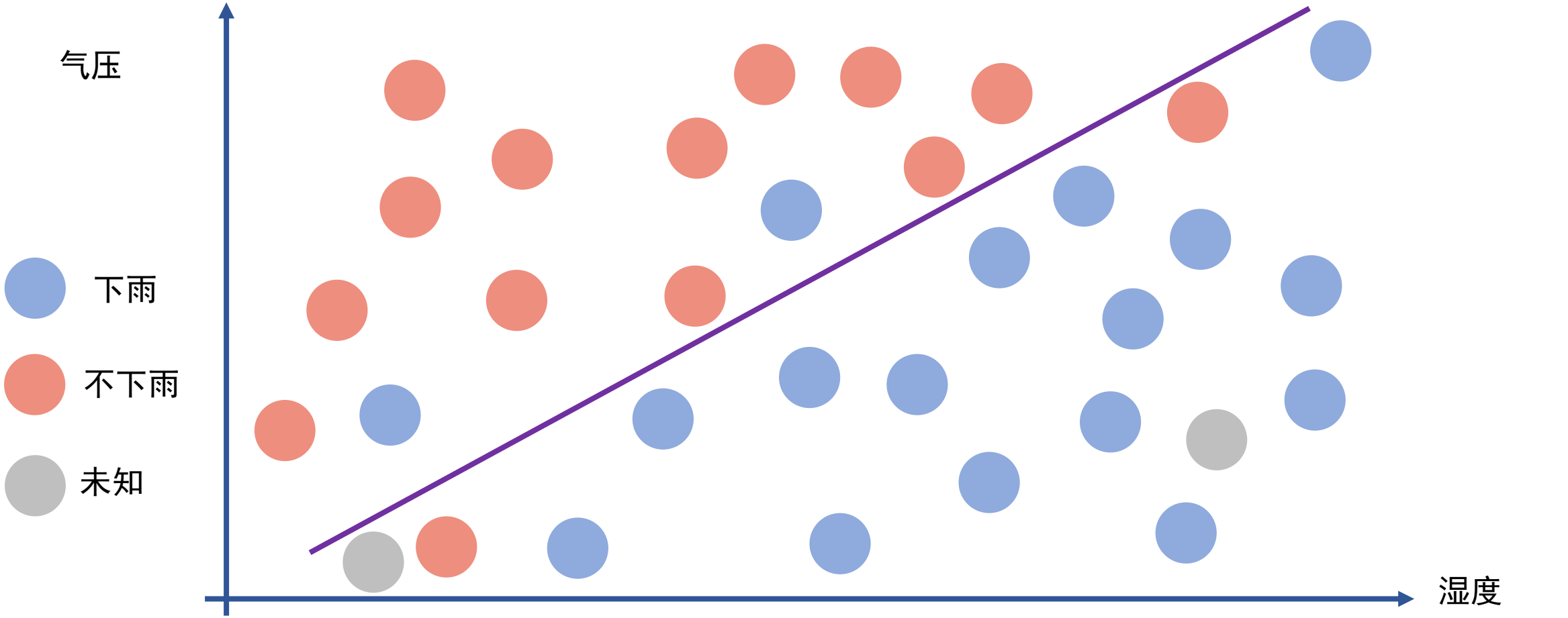
阈值函数 Threshold function

- 硬阈值 Hard threshold
  - 输出只有两种可能的值

$$h_{\mathbf{w}}(\mathbf{x}) = \begin{cases} 1 \text{ if } \mathbf{w} \cdot \mathbf{x} \geq 0 \\ 0 \text{ if } \mathbf{w} \cdot \mathbf{x} < 0 \end{cases}$$



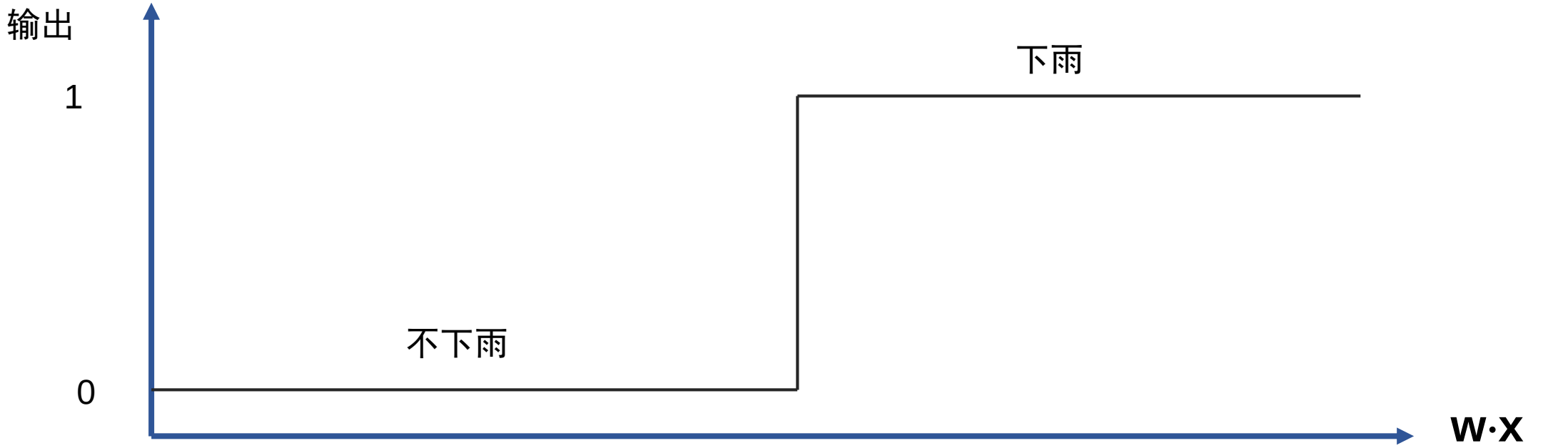
你对自己的分类结果有多自信?



阈值函数 Threshold function

- 硬阈值 Hard threshold
  - 不能表明对分类结果的自信程度
  - 很难求导

$$h_{\mathbf{w}}(\mathbf{x}) = \begin{cases} 1 \text{ if } \mathbf{w} \cdot \mathbf{x} \geq 0 \\ 0 \text{ if } \mathbf{w} \cdot \mathbf{x} < 0 \end{cases}$$

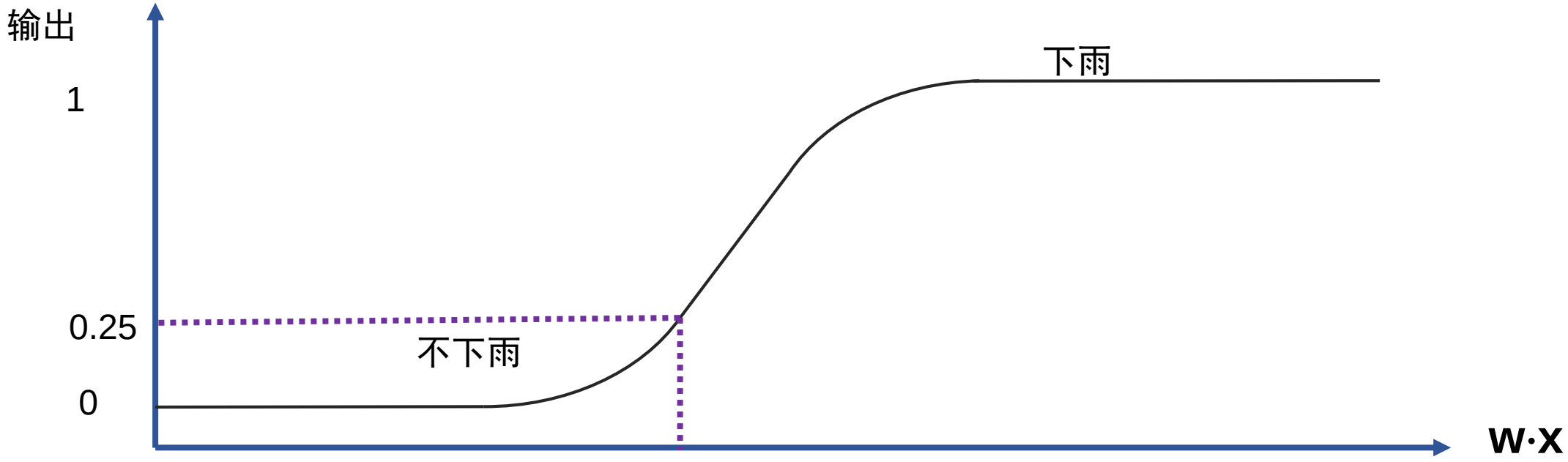


阈值函数 Threshold function

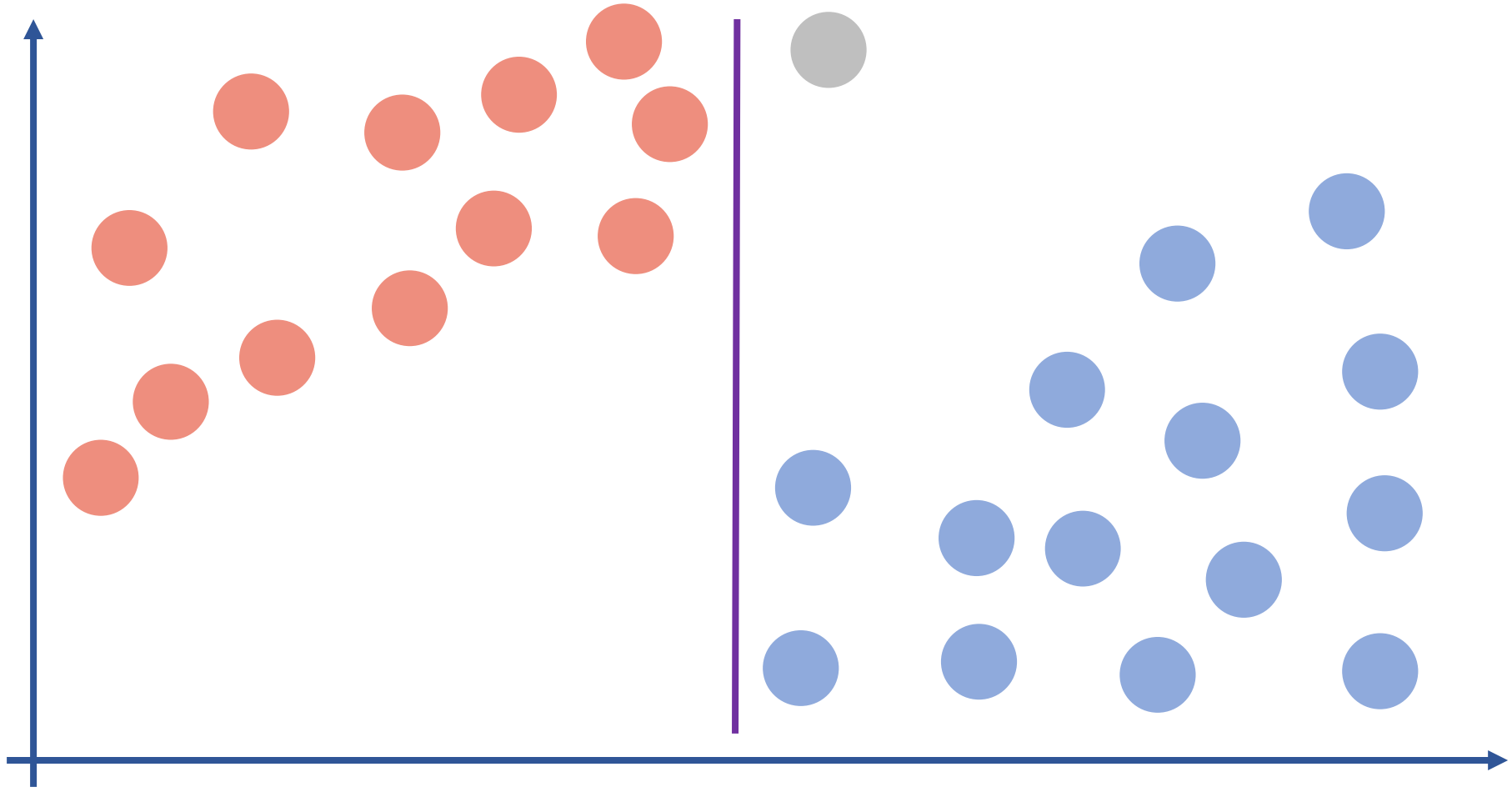
- 软阈值 Soft threshold
  - 输出介于 0 – 1 之间的实数
    - 反映属于哪一类别的概率

$$h_{\mathbf{w}}(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}\cdot\mathbf{x}}}$$

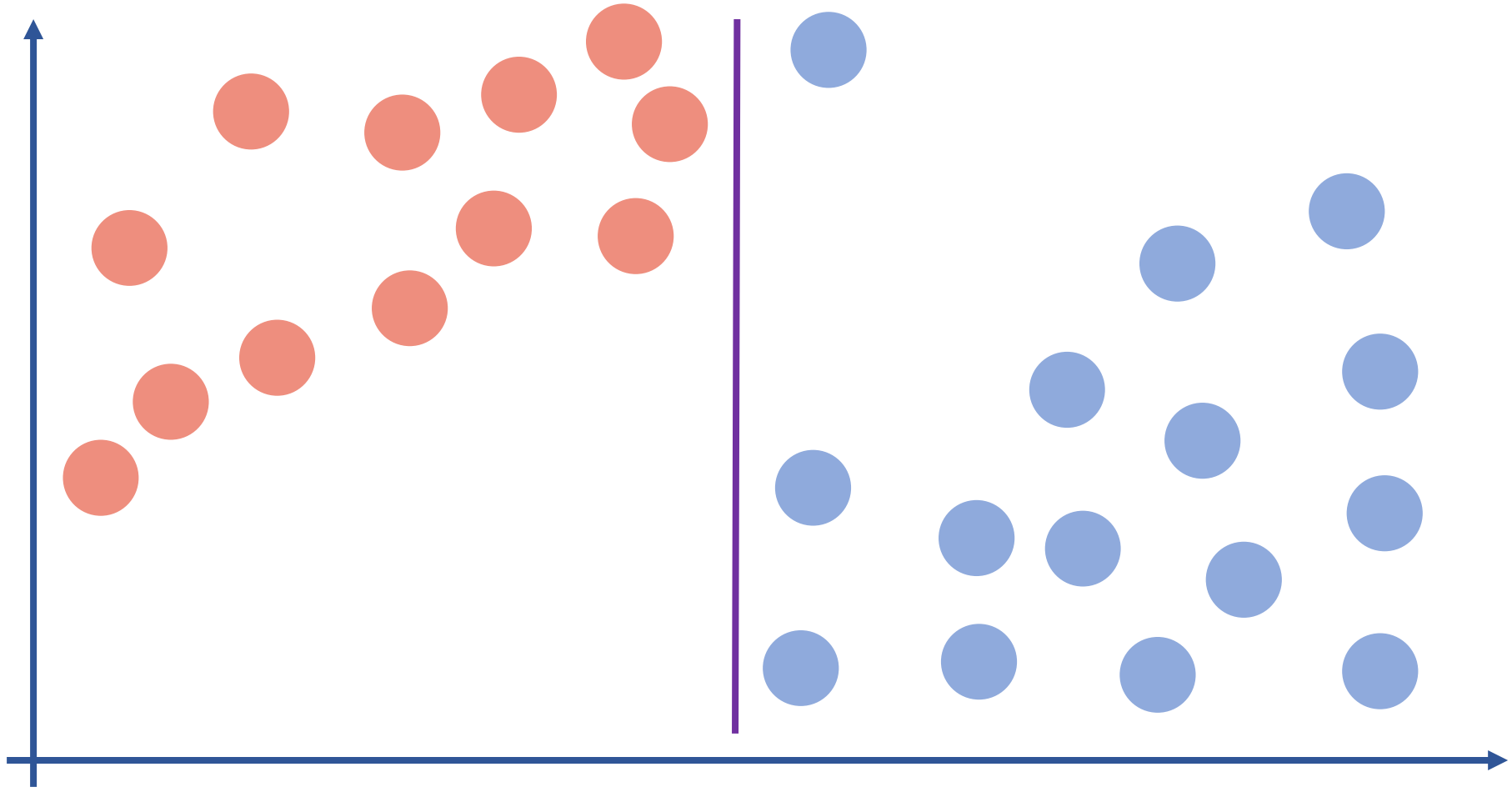
Logistic函数/Sigmoid函数



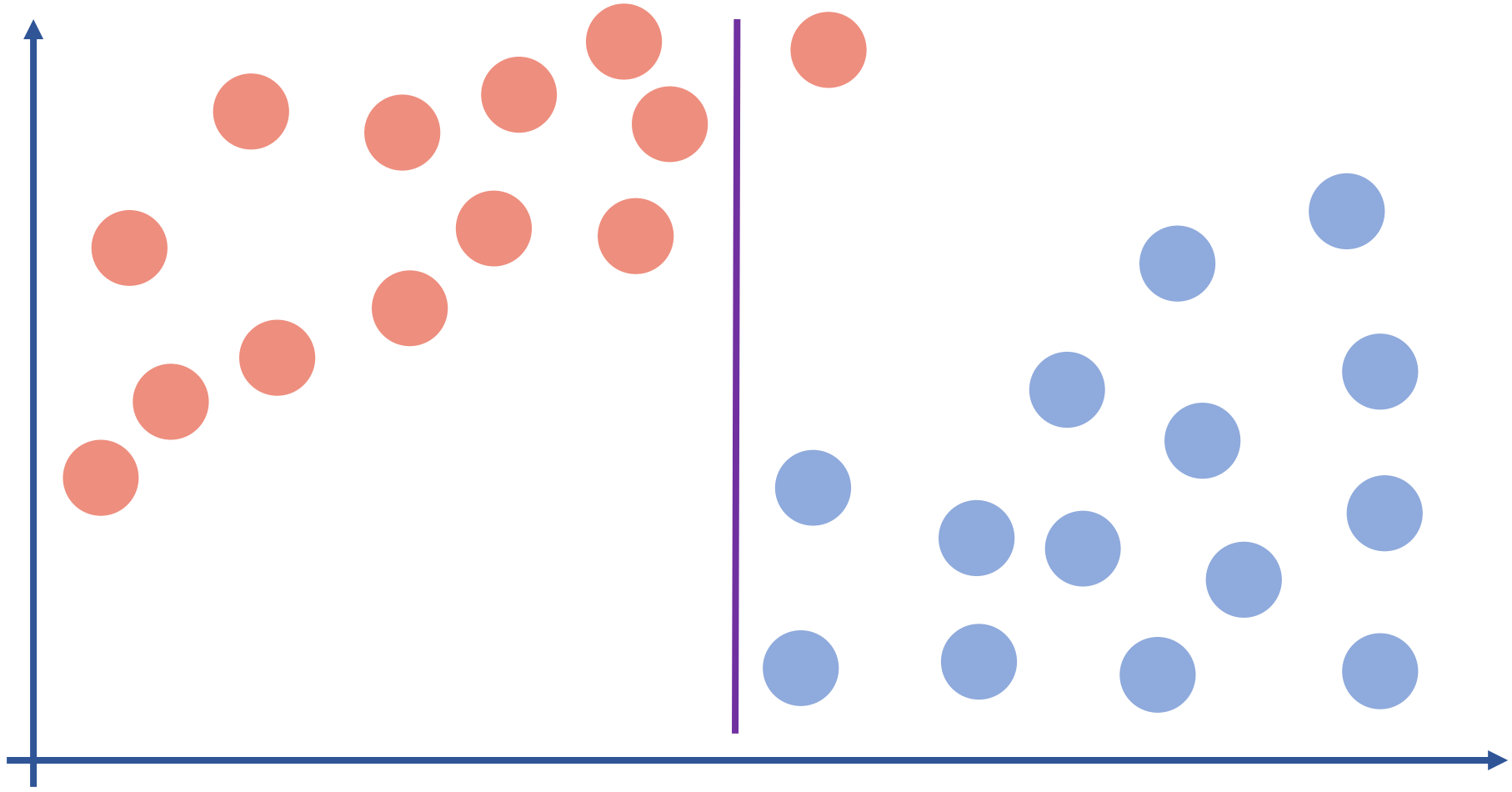
画决策边界可能会……



画决策边界可能会……

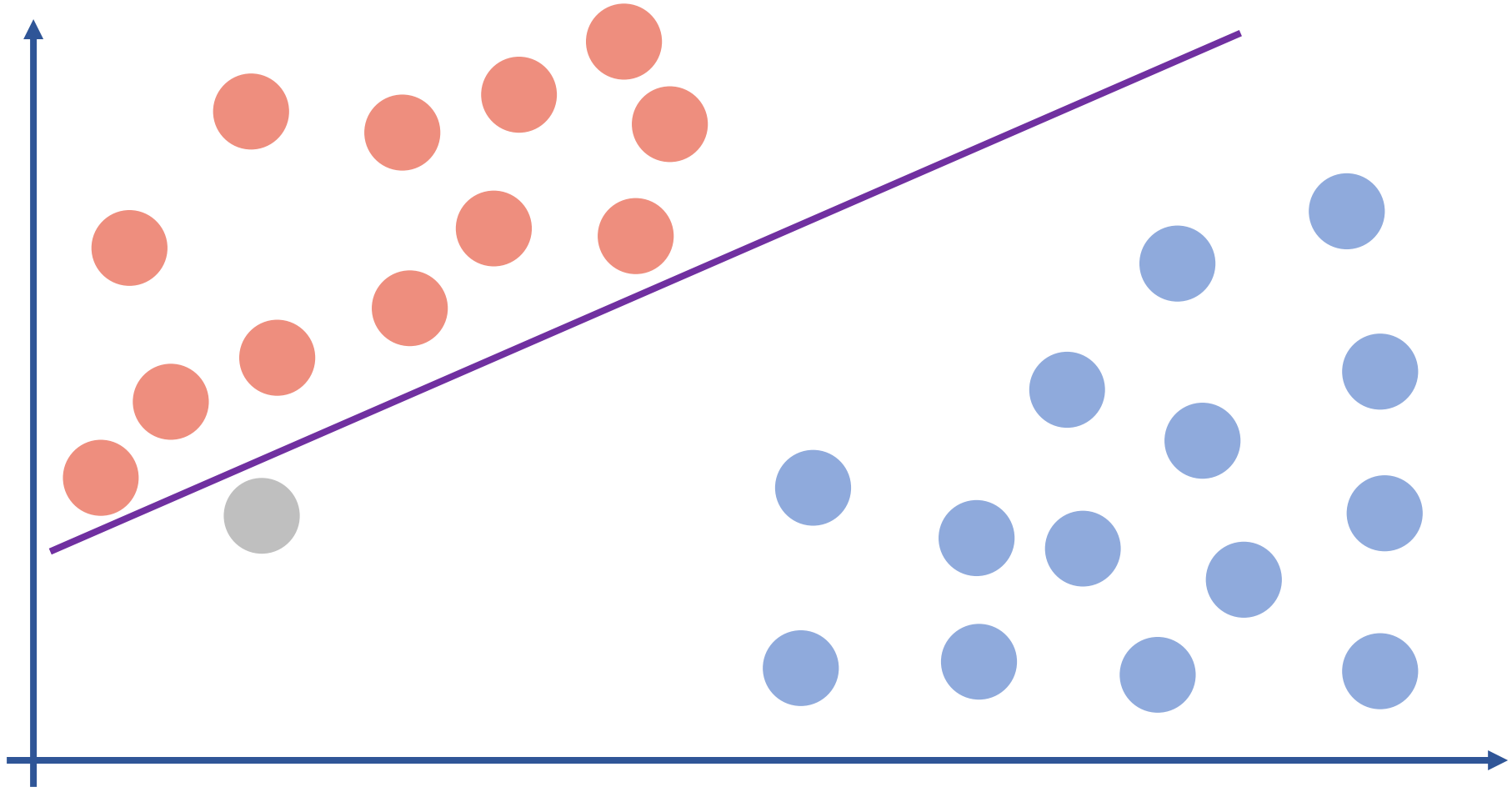


画决策边界可能会……

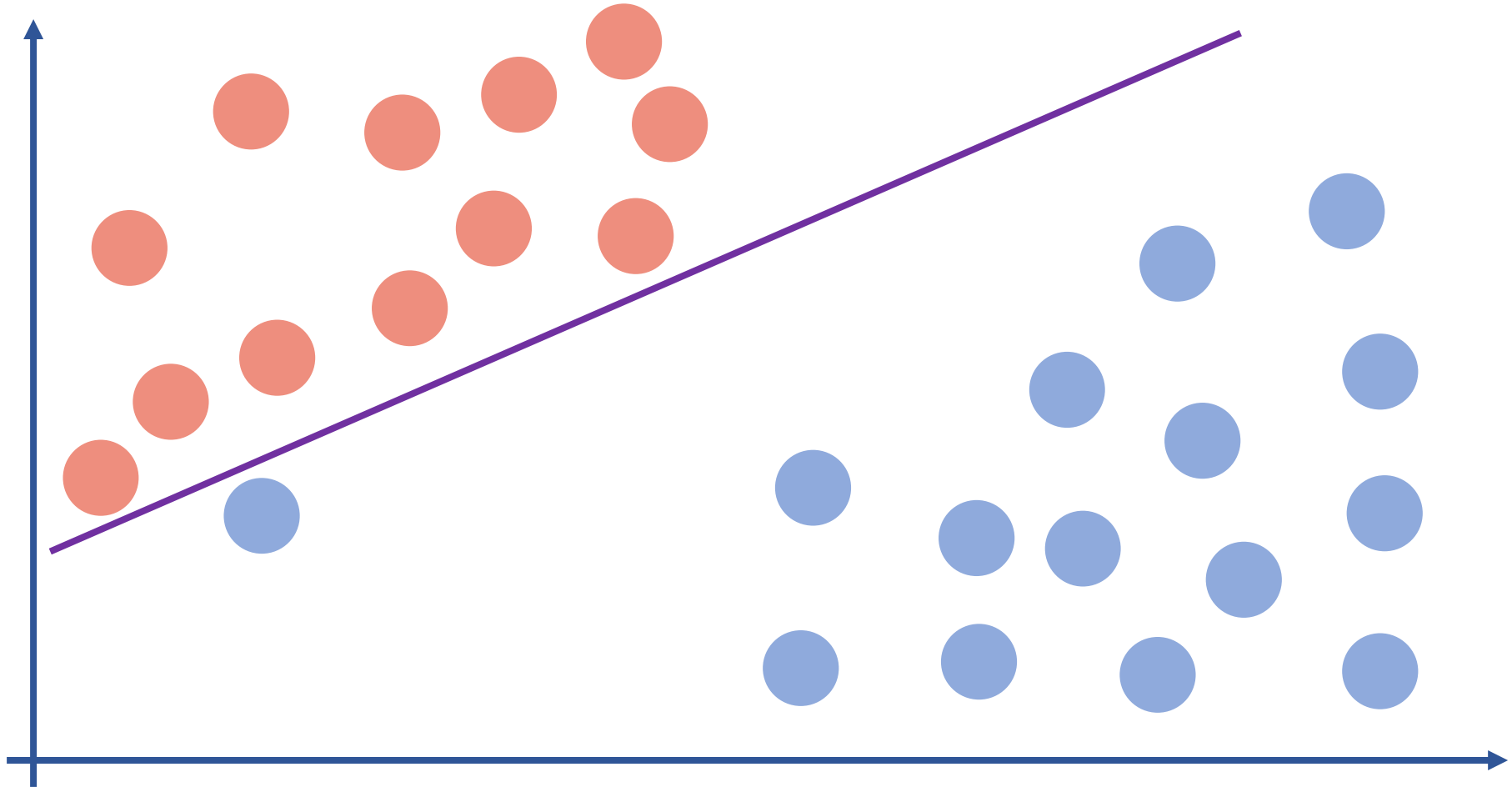


可能这不是一个好的决策边界

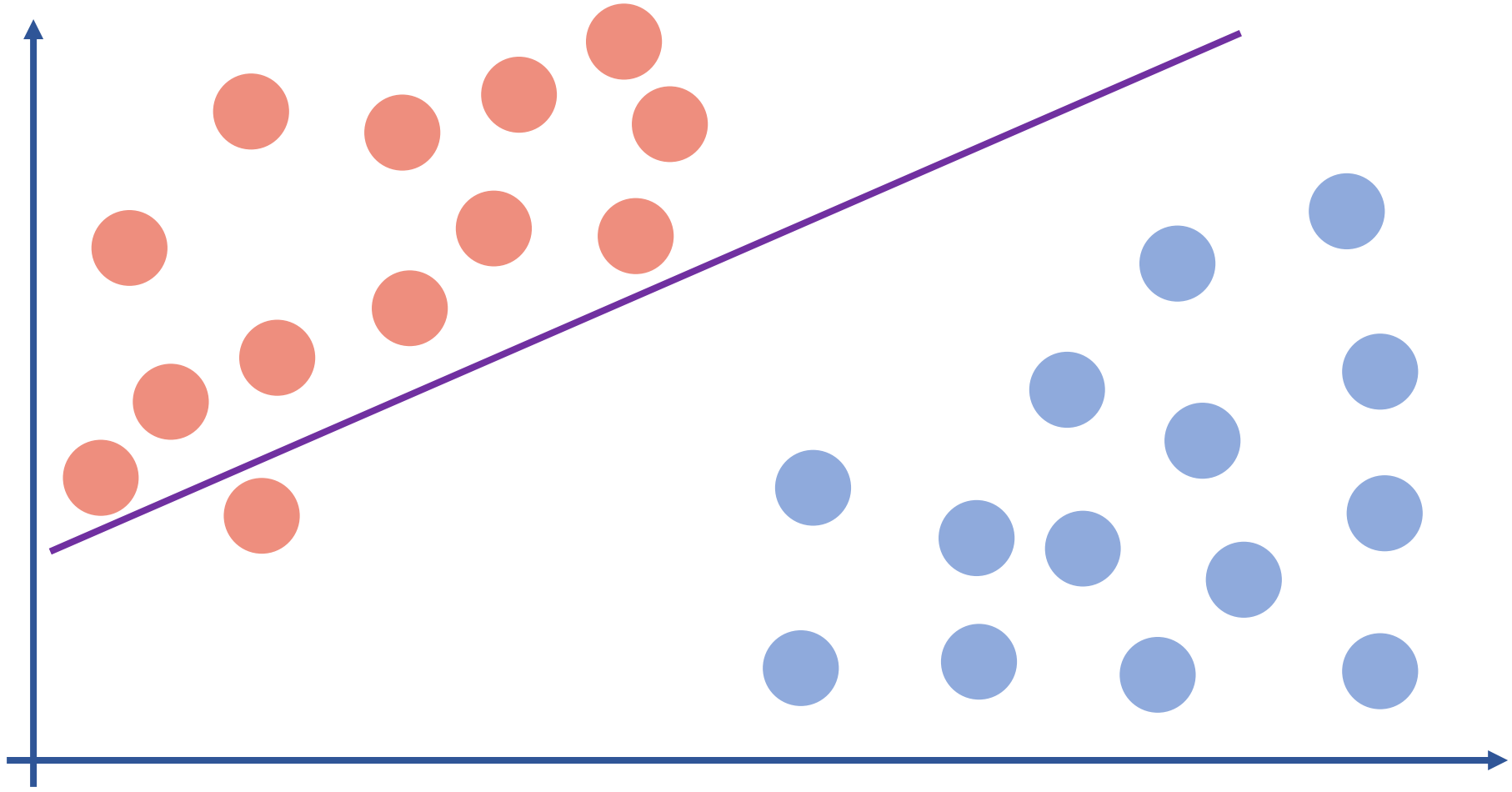
画决策边界可能会……



画决策边界可能会……

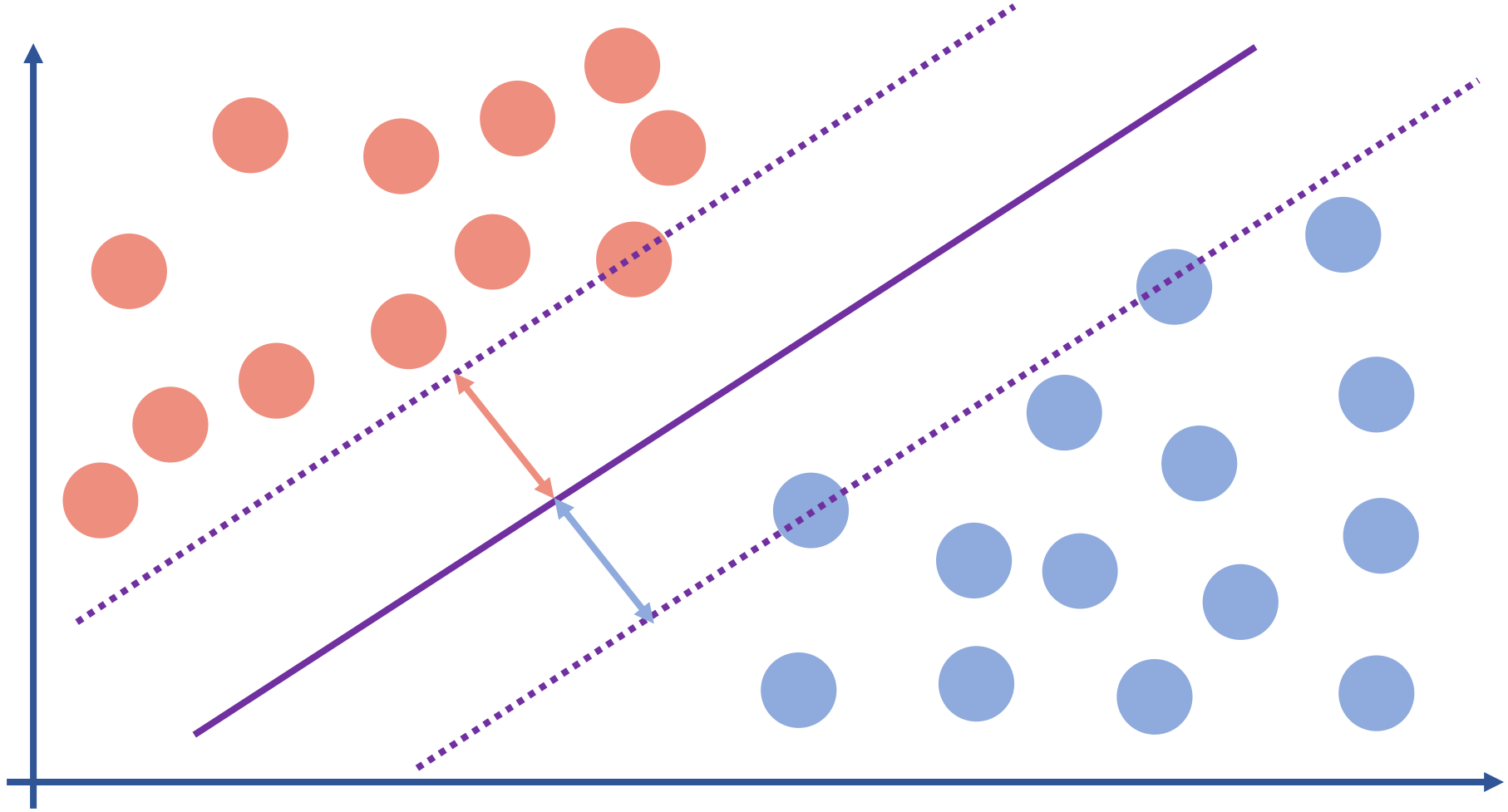


画决策边界可能会……



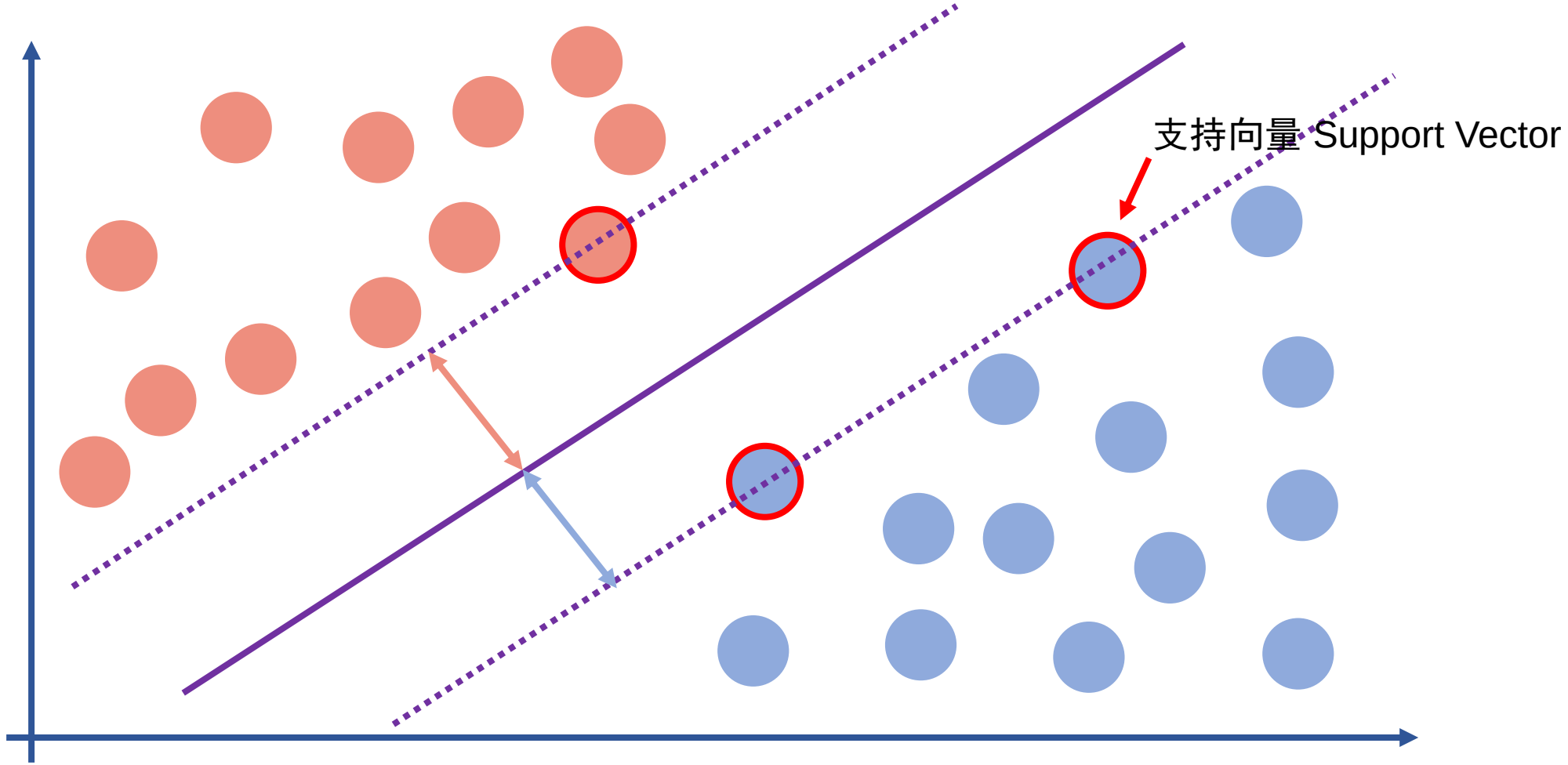
可能这不是一个好的决策边界

支持向量机 Support vector machine



这条线似乎与红点和蓝点的距离是一样远的

支持向量机 Support vector machine

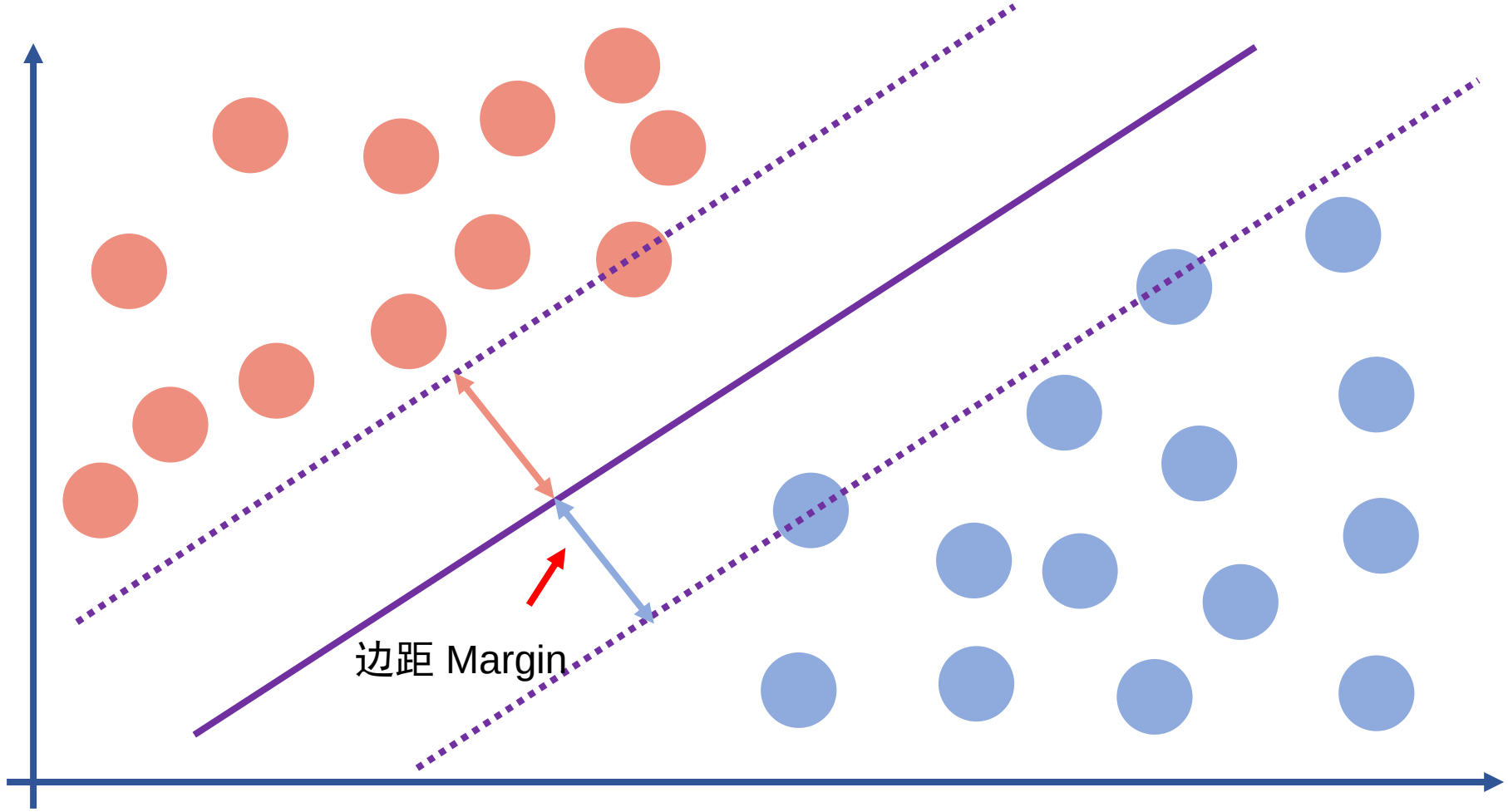


这条线似乎与红点和蓝点的距离是一样远的

支持向量机 Support vector machine

- 支持向量 Support vector
  - 离决策边界最近的点叫做支持向量
  - 这些点是很难被分类的点
  - 直接决定了超平面的位置

支持向量机 Support vector machine



这条线似乎与红点和蓝点的距离是一样远的

支持向量机 Support vector machine

- 支持向量 Support vector
  - 离决策边界最近的点叫做支持向量
  - 这些点是很难被分类的点
  - 直接决定了超平面的位置
- 最大边距分割边界 Maximum margin separator
  - 使得距离决策边界最近的点(支持向量)到决策边界的距离**最大**

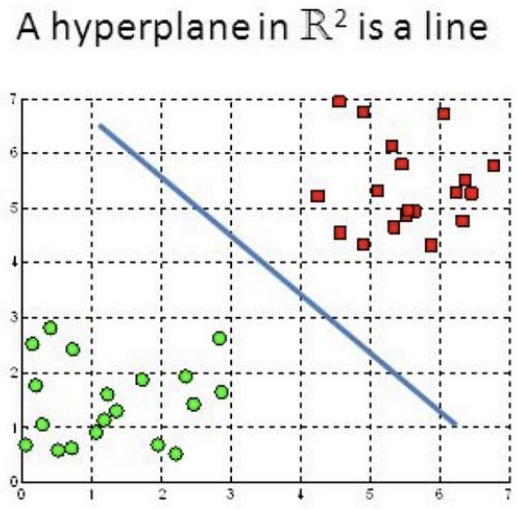
练习 #4

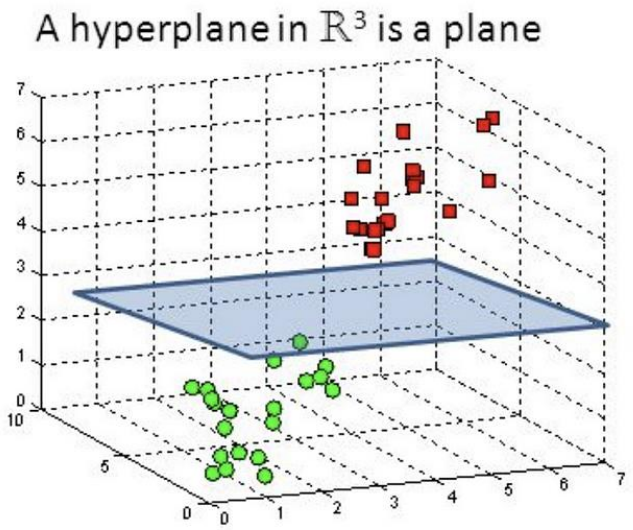
- 在如下所示的数据中，哪些点是支持向量？SVM的决策边界是什么呢？

x_1	x_2	y
1	9	0
5	5	0
9	5	1
13	1	1
13	9	1
1	1	0

支持向量机 Support vector machine

- 可以处理高维数据

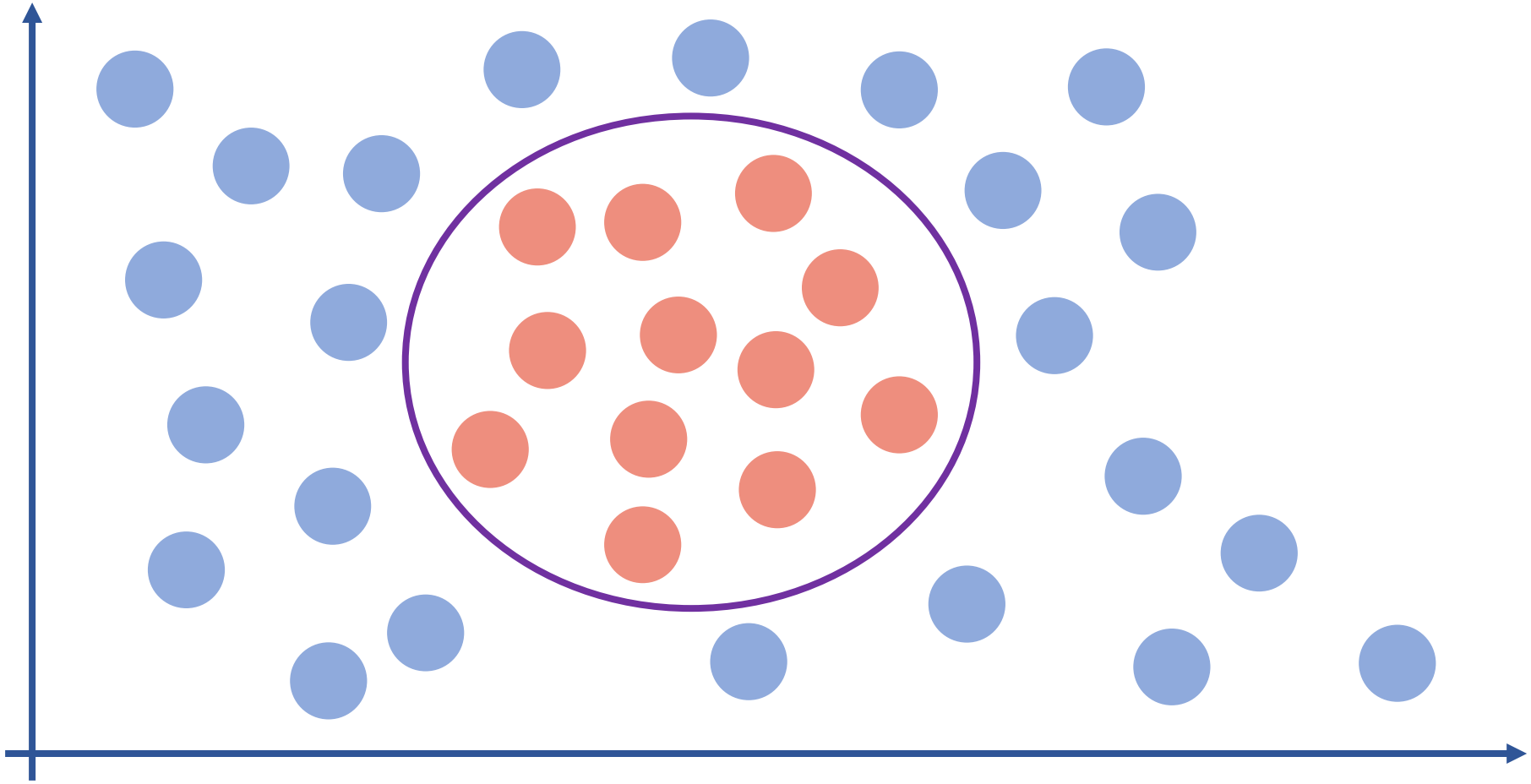




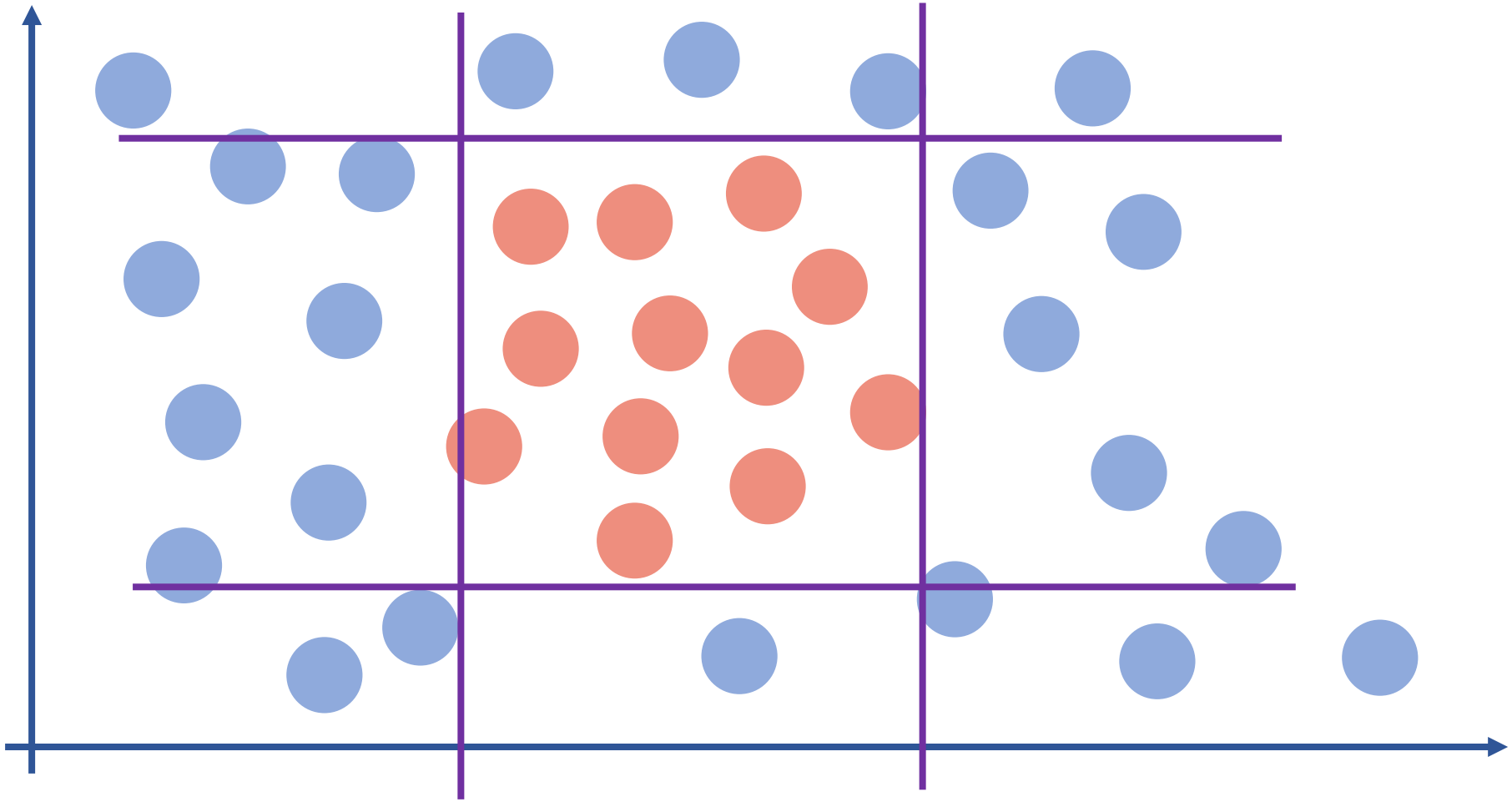
来源：by Rohith Gandhi from *Towards Data Science*

- 可以处理非线性可分(non-linearly separable)的数据

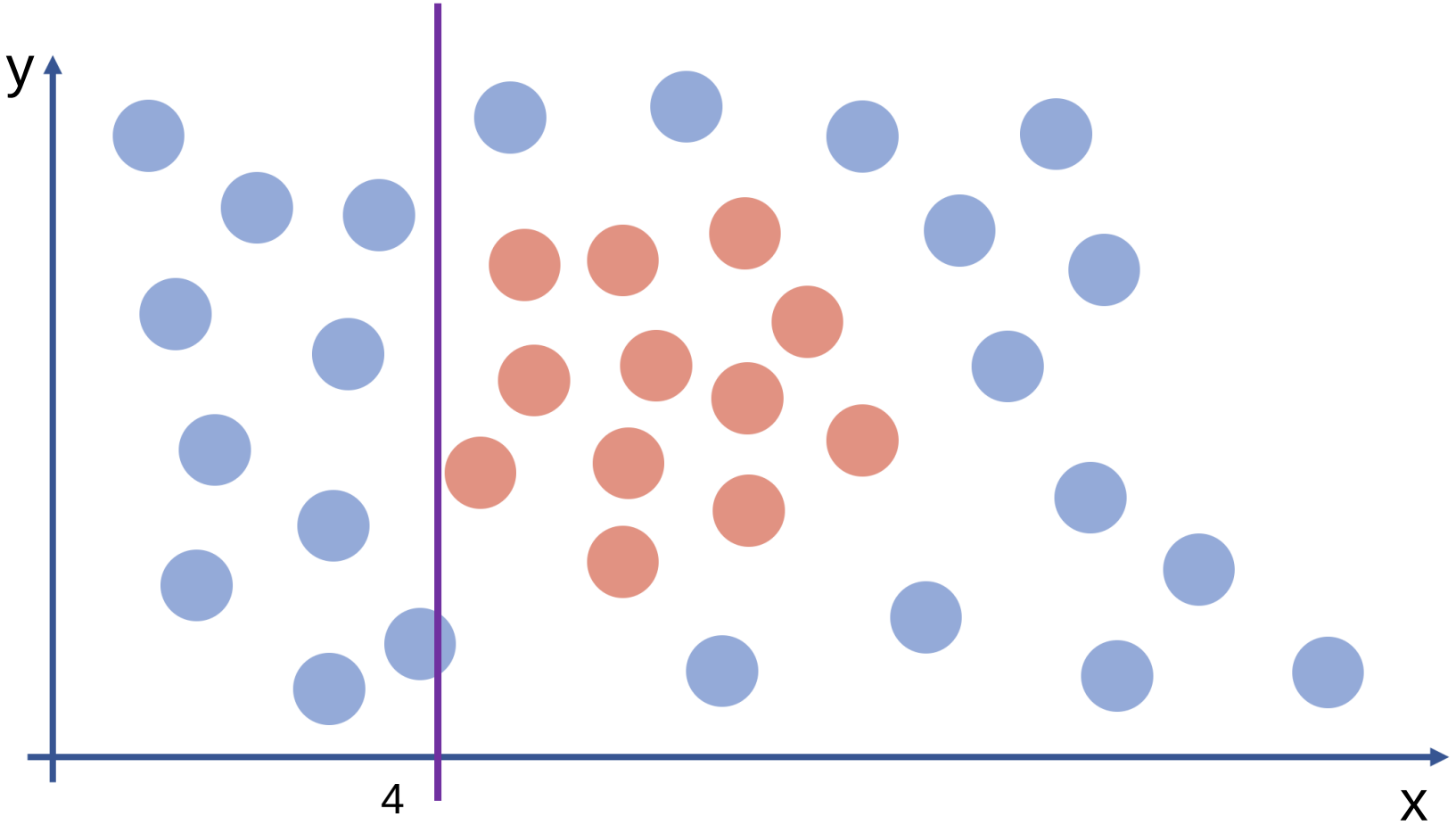
非线性可分数据 Non-linearly separable data

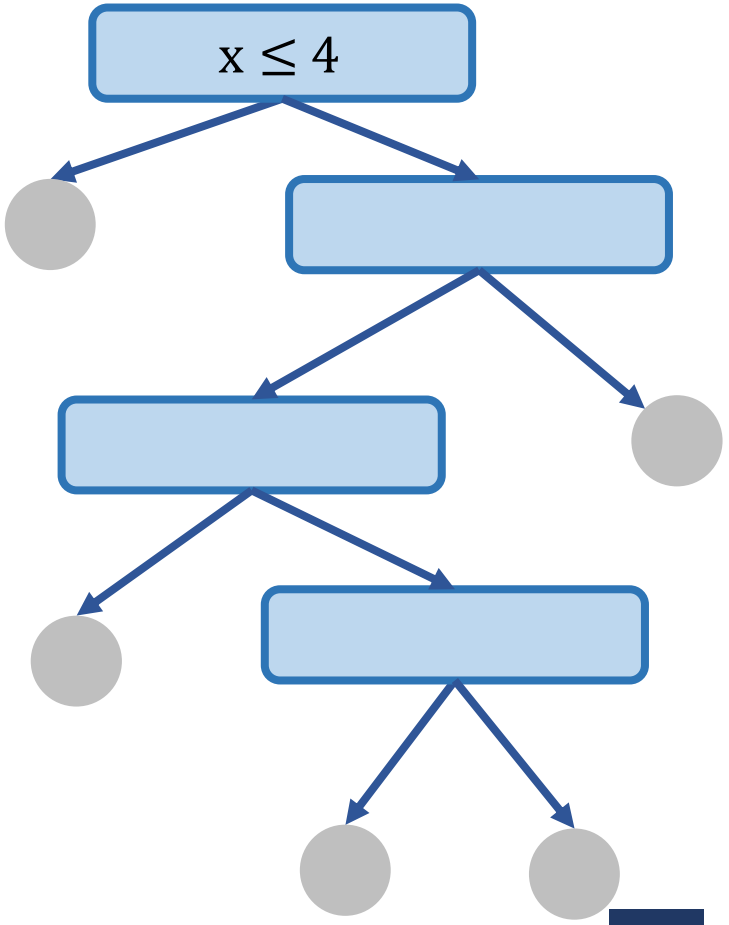


决策树 Decision tree

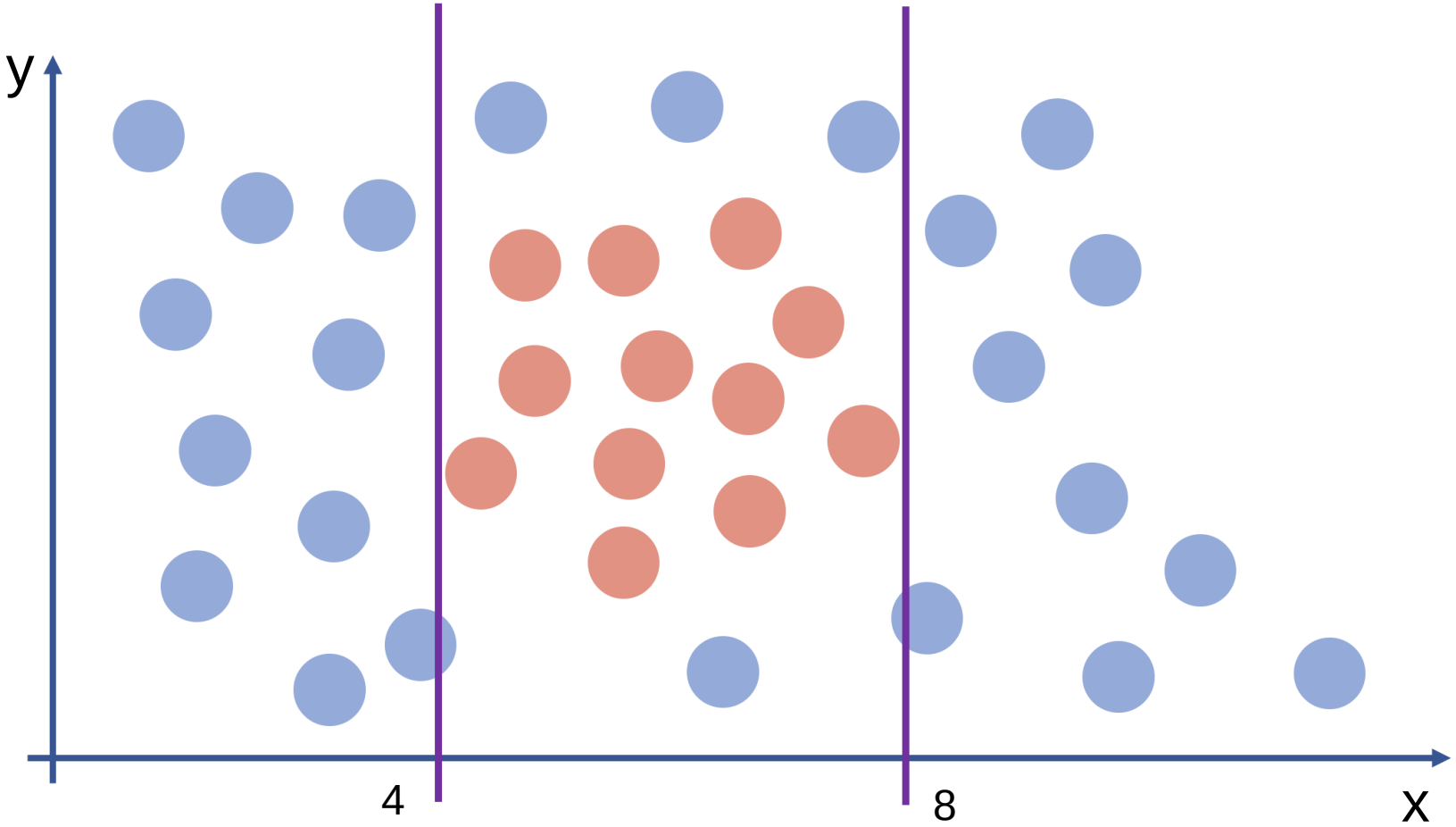


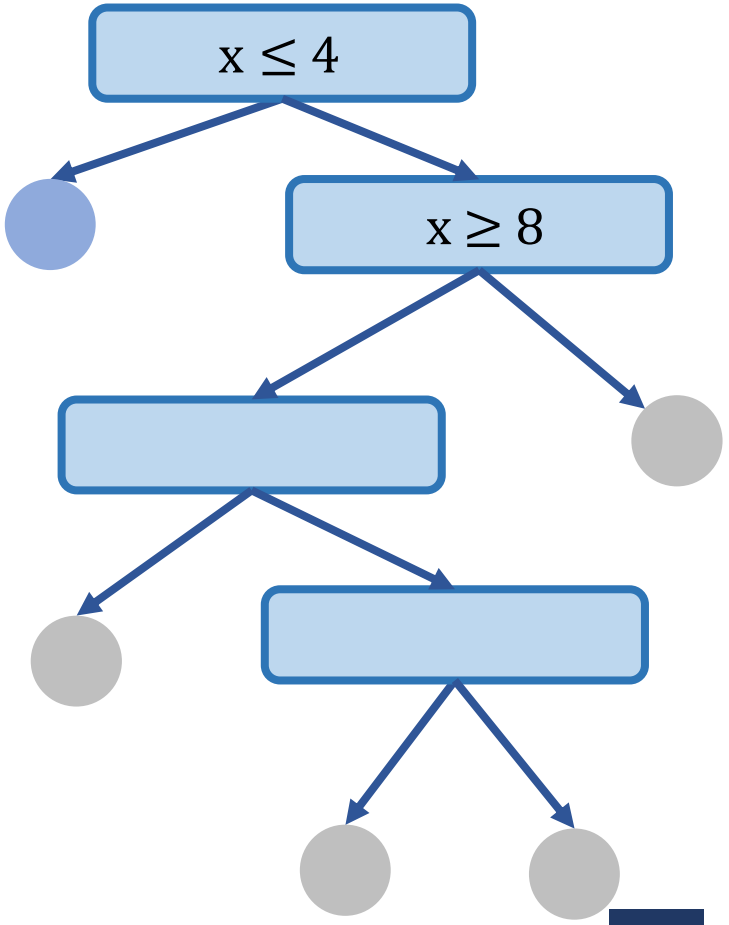
决策树 Decision tree



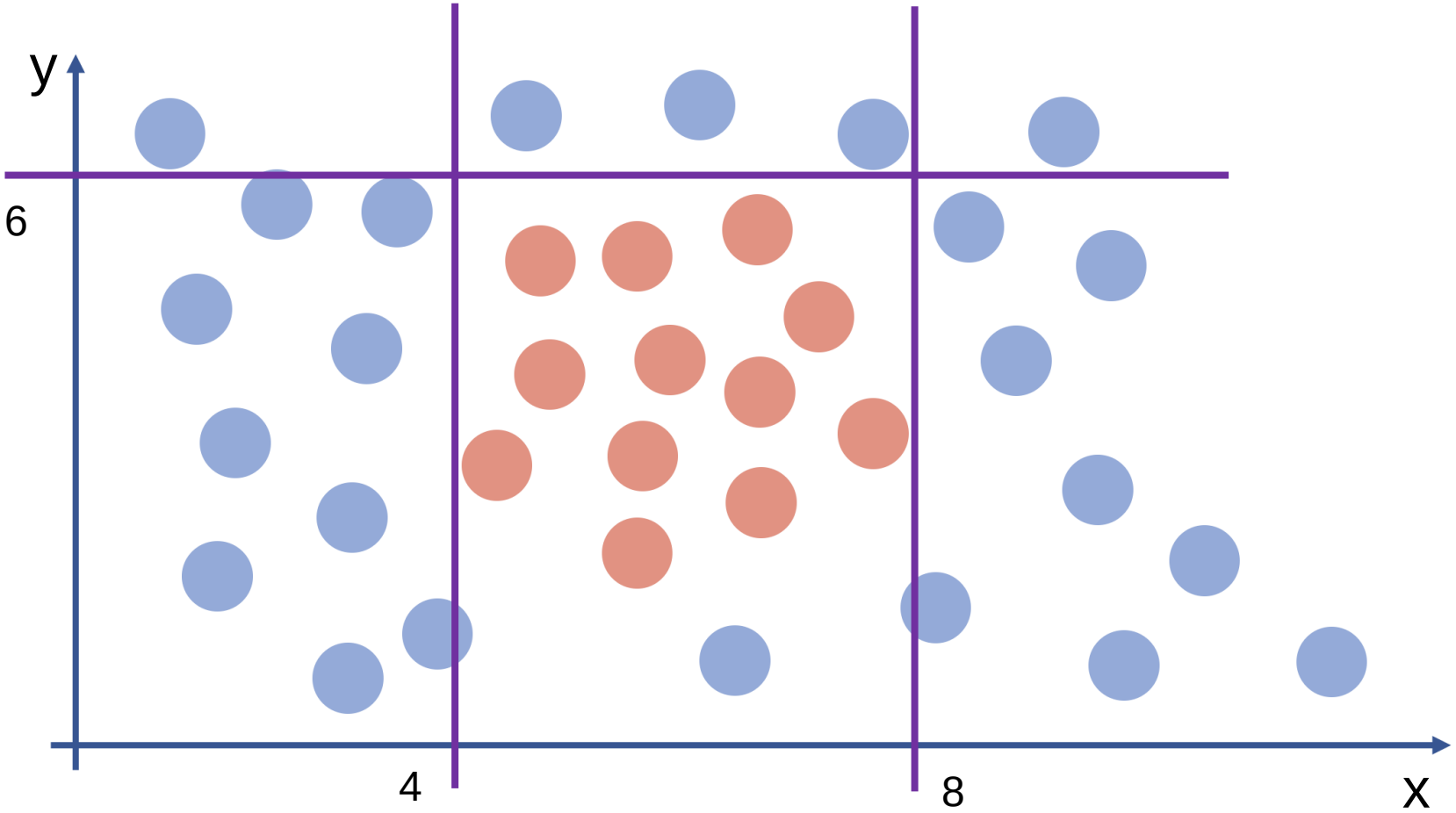


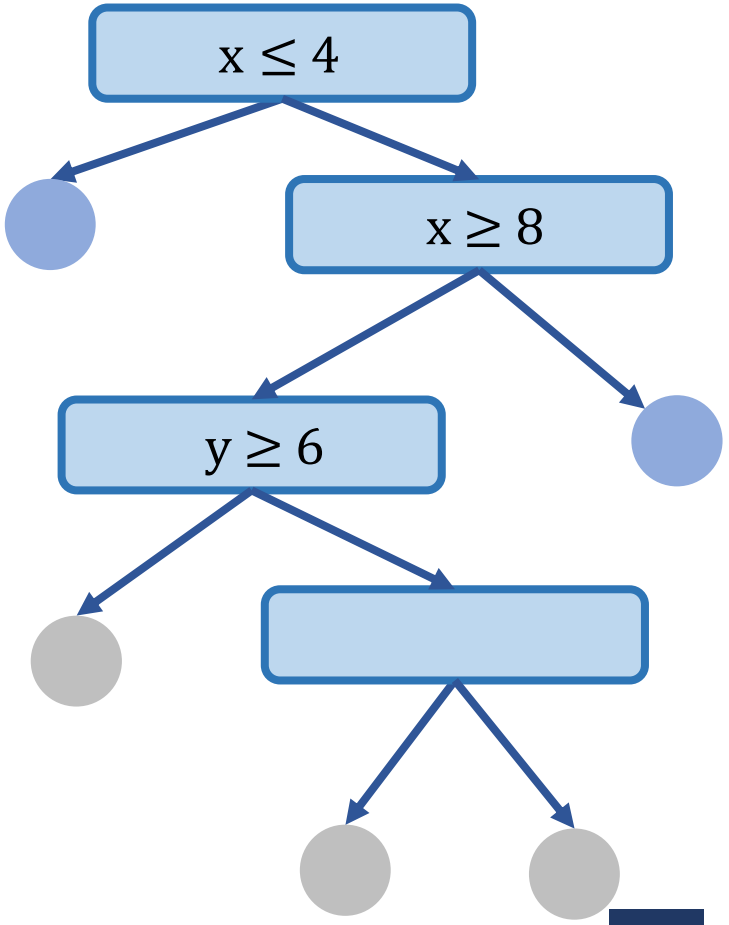
决策树 Decision tree



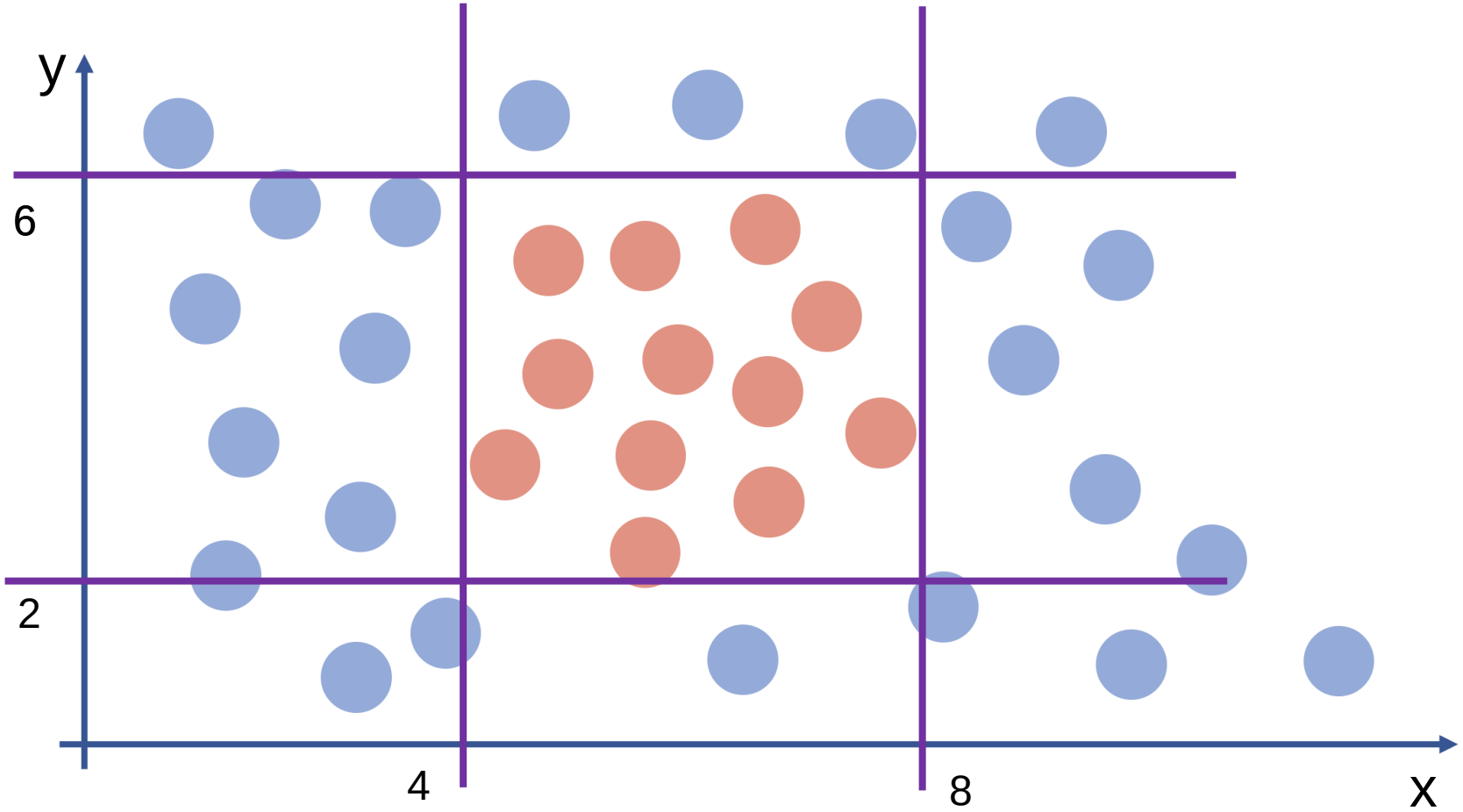


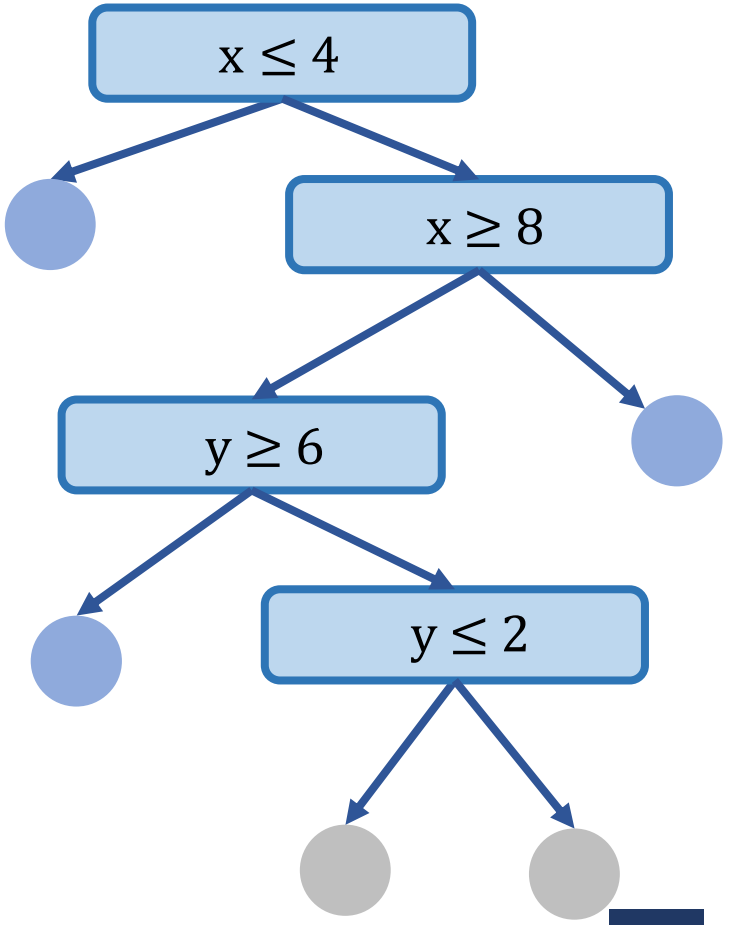
决策树 Decision tree





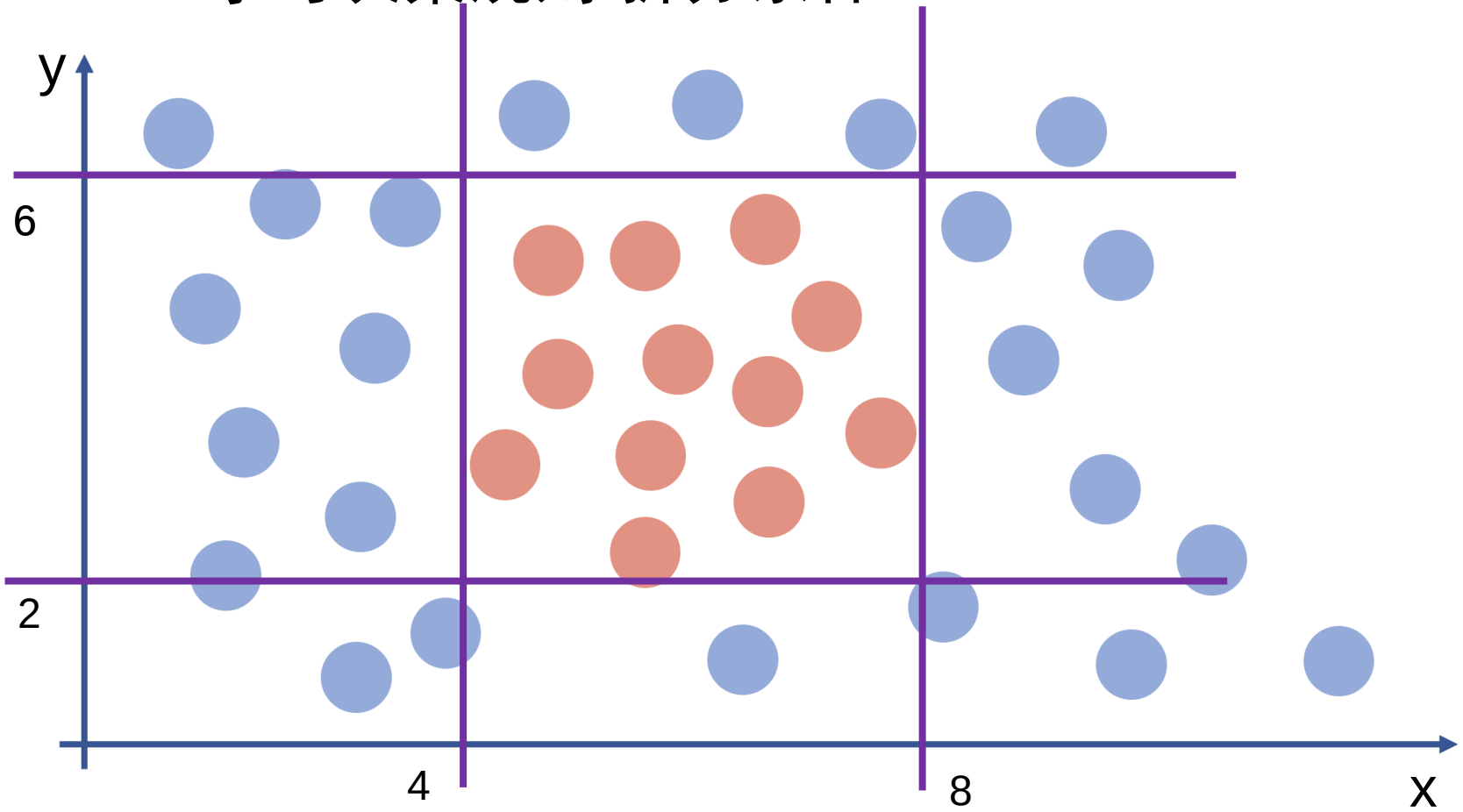
决策树 Decision tree

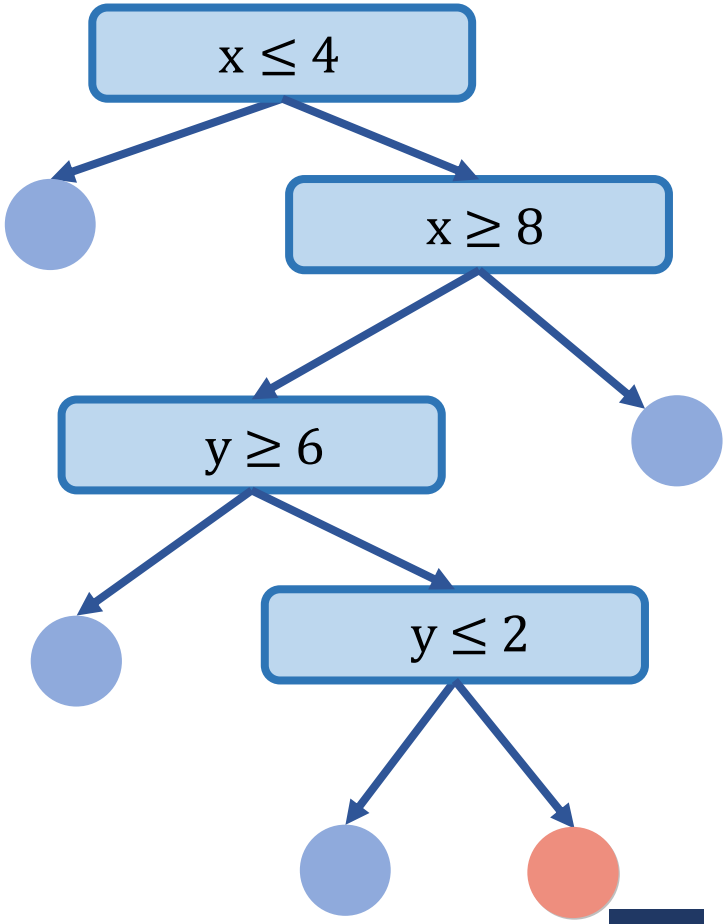




决策树 Decision tree

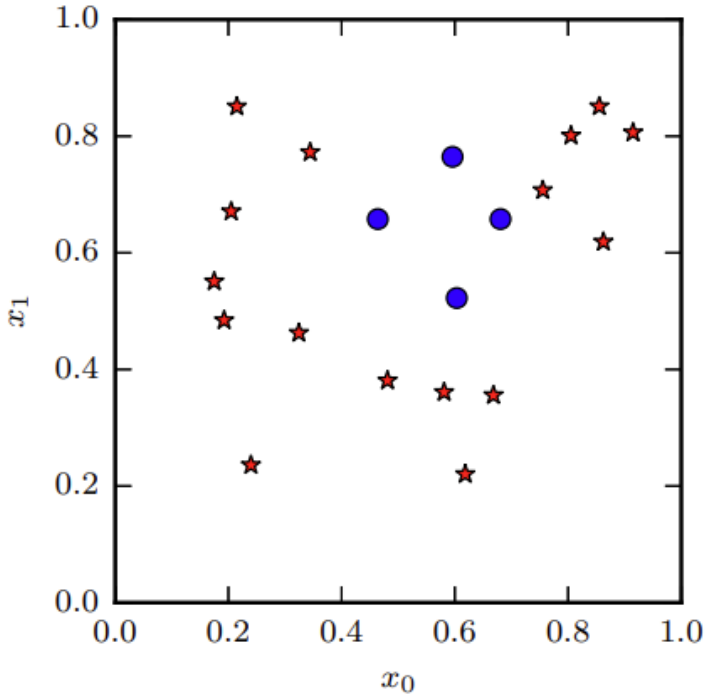
- 学习决策规则/拆分条件





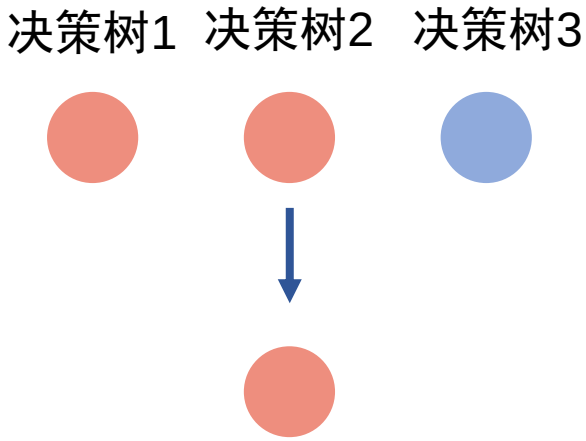
练习 #5

- 有如下所示的数据集，请提议一个决策树的模型，并根据你提议的决策树将(0.4, 1), (0.6, 1), (0.6, 0)进行分类



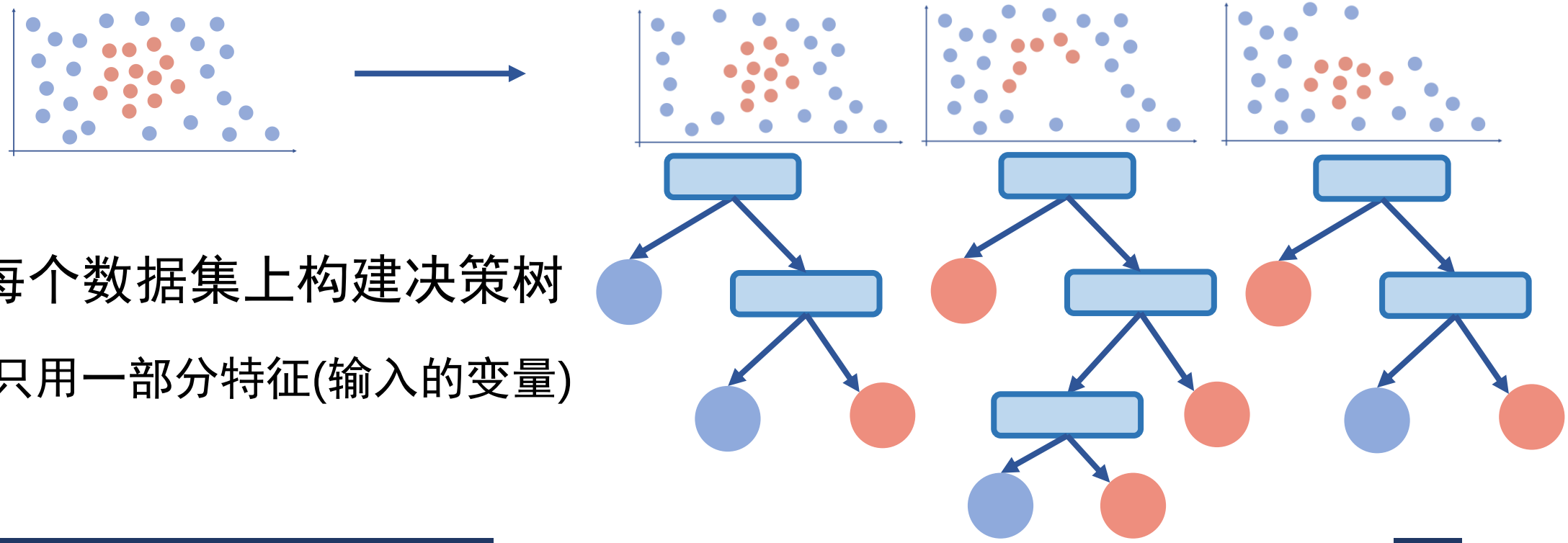
集成方法 Ensemble methods

- 创建多个分类模型，然后将它们组合在一起，从而增进分类结果
- 随机森林 Random Forest
  - 构建多个决策树模型
  - 将不同模型的结果集成在一起
    - 少数服从多数 (Majority Vote)



随机森林 Random Forest

- 自助法 Bootstrapping
  - 有放回的抽样方法



- 在每个数据集上构建决策树
  - 只用一部分特征(输入的变量)

练习 #6

- 假设有一组数据(X, Y)，Y取两个值Yes和No。我们使用了集成方法进行分类。首先采用自助法(Bootstrapping)抽取了10个新的数据集，分辨学习了10个分类模型。基于x，分类模型输出对$(P(Y = Yes|x))$的估计

0.1, 0.15, 0.2, 0.2, 0.55, 0.6, 0.6, 0.65, 0.7, 0.75

① 如果采取少数服从多数(Majority Vote)的方法，分类结果为?

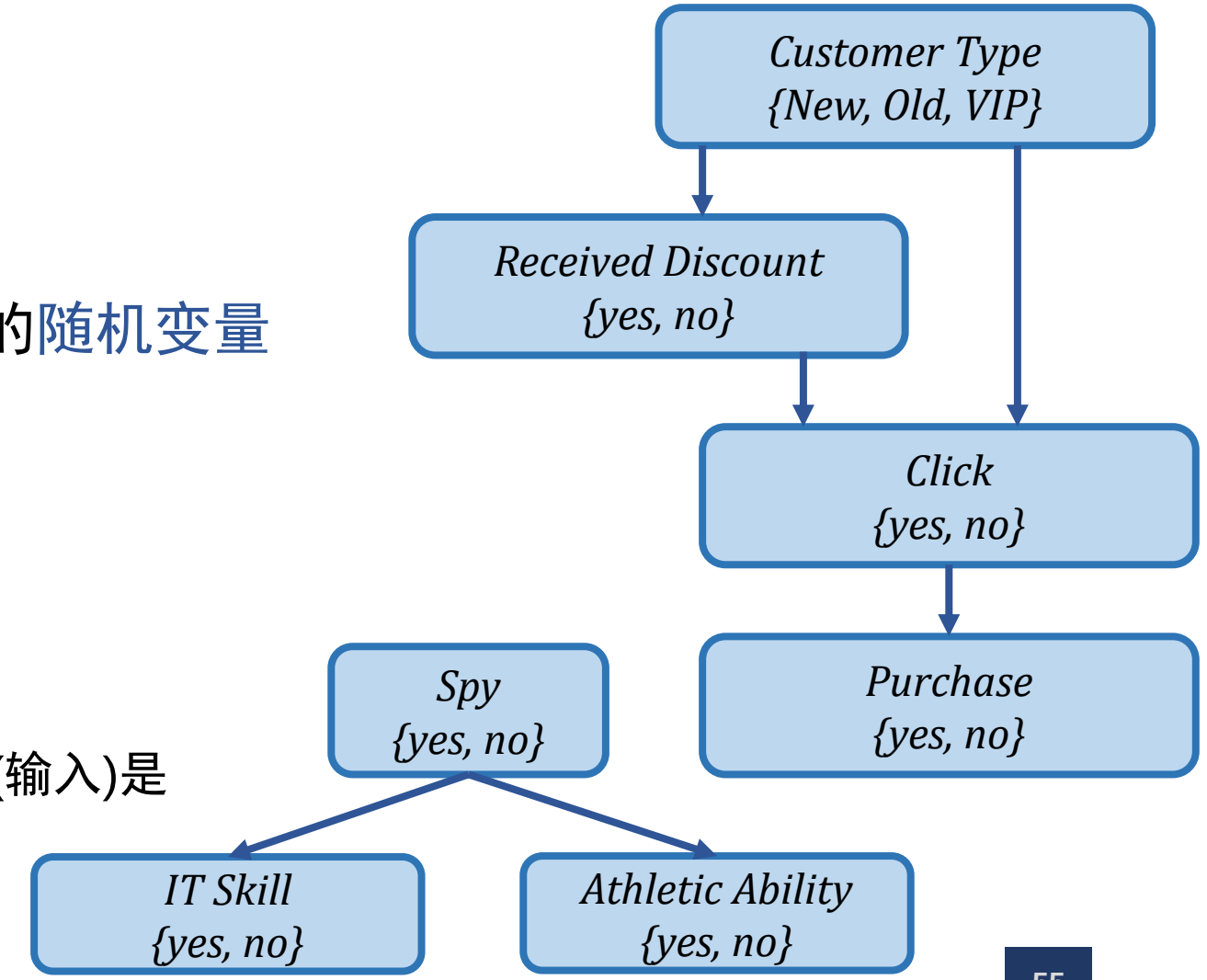
② 如果分类结果取决于平均概率，那分类为?

集成方法 Ensemble methods

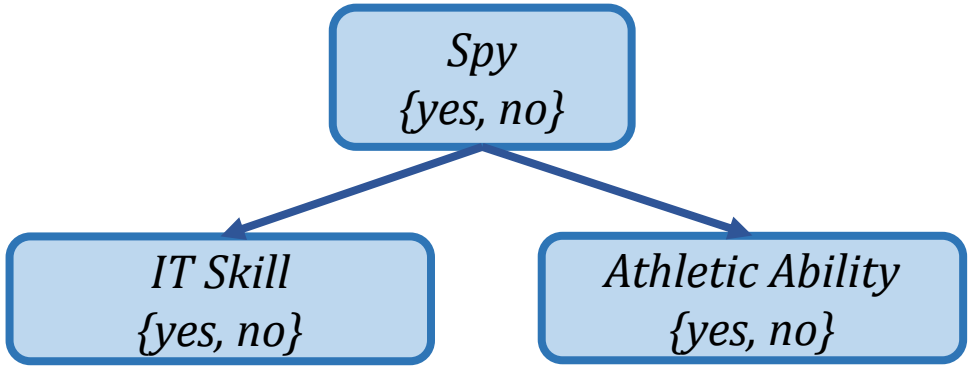
- 创建多个分类模型，然后将它们组合在一起，从而增进分类结果
  - 构建多个机器学习的模型
  - 将不同模型的结果集成在一起
    - 硬投票(Hard vote): 根据模型输出的分类来继承
    - 软投票(Soft vote): 根据模型预测的概率来集成

基于模型的方法 Model-based approach

- 模型：贝叶斯网络
- 朴素贝叶斯 Naïve Bayes
  - 输入和输出均为贝叶斯网络中的随机变量
  - 目标： P(output | input) ?
  - 朴素(天真幼稚) Naïve
    - 结构简单，Output影响Input
    - 在类别确定时，假设其各个特征(输入)是相互独立的



朴素贝叶斯 Naïve Bayes



Spy	Probability
Yes	0.25
No	0.75

Spy	IT Skill	Probability
yes	yes	0.8
yes	no	0.2
no	yes	0.47
no	no	0.53

Spy	Athletic Ability	Probability
yes	yes	0.6
yes	no	0.4
no	yes	0.33
no	no	0.67

- 练习 #7：
- 一个人: IT skill = yes, Athletic Ability=no
- 分类任务: spy or not?
- 即：求$\mathrm{P}(\boldsymbol{Spy}|IT\ Skill = yes, Atheletic\ Ability = no)$?

分类 Classification

- 输出是离散值
  - 近邻法 Nearest neighbor
  - 感知机模型 Perceptron / 线性回归 Linear regression
  - 支持向量机 Support vector machine
  - 决策树 Decision trees
  - 集成方法 Ensemble Learning
    - 随机森林 Random forest
  - 朴素贝叶斯 Naïve Bayes

回归 Regression

- 学习将输入点映射到**连续值**的函数的监督学习任务

f(广告投入)　　真实的函数关系是未知的、隐藏的

$$f(1200) = 5800$$

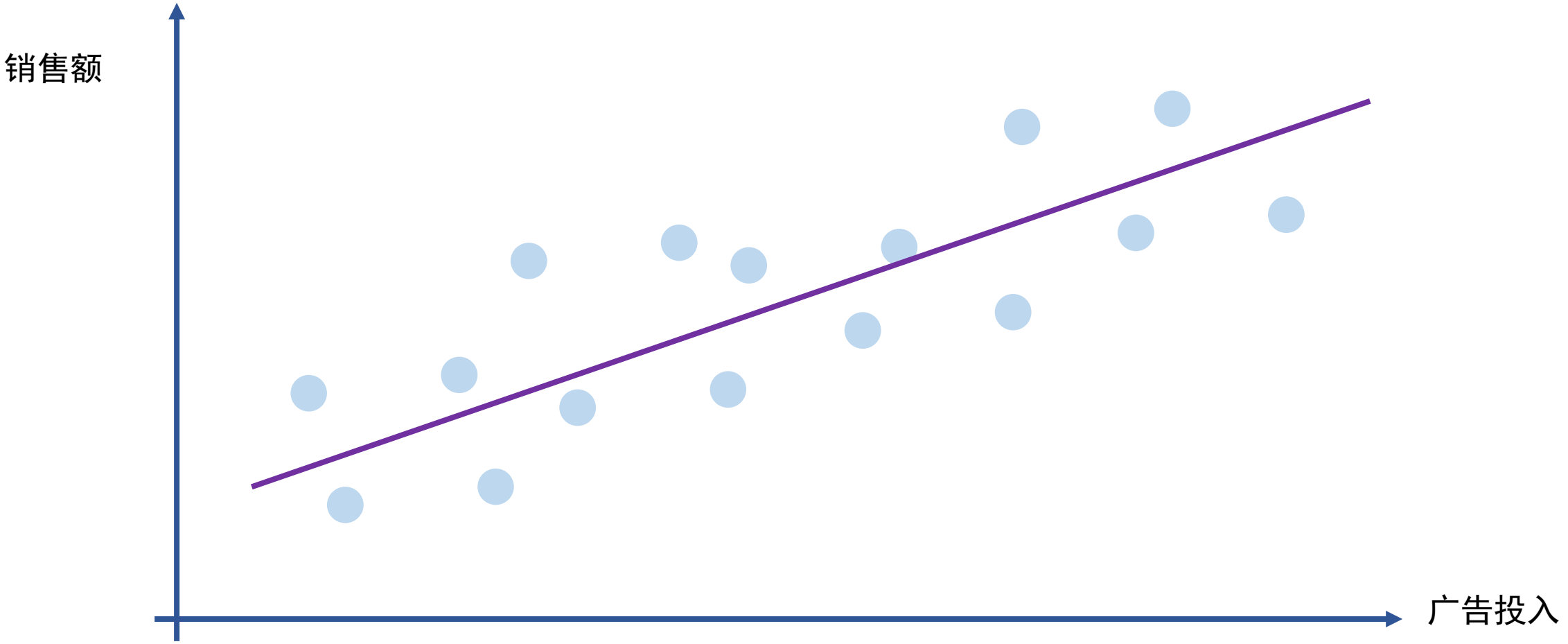
$$f(2800) = 13400$$

$$f(1800) = 8400$$

h(广告投入)　　假设的函数关系 Hypothesis function

预测任务: 这个 $h()$ 是什么样子的?

画图



找到一条线来估计广告投入和销售额之间的关系

有问题吗?

- 请随时举手提问。

BUSS 3620.人工智能导论

#2. 评估假设

刘佳璐

安泰经济与管理学院

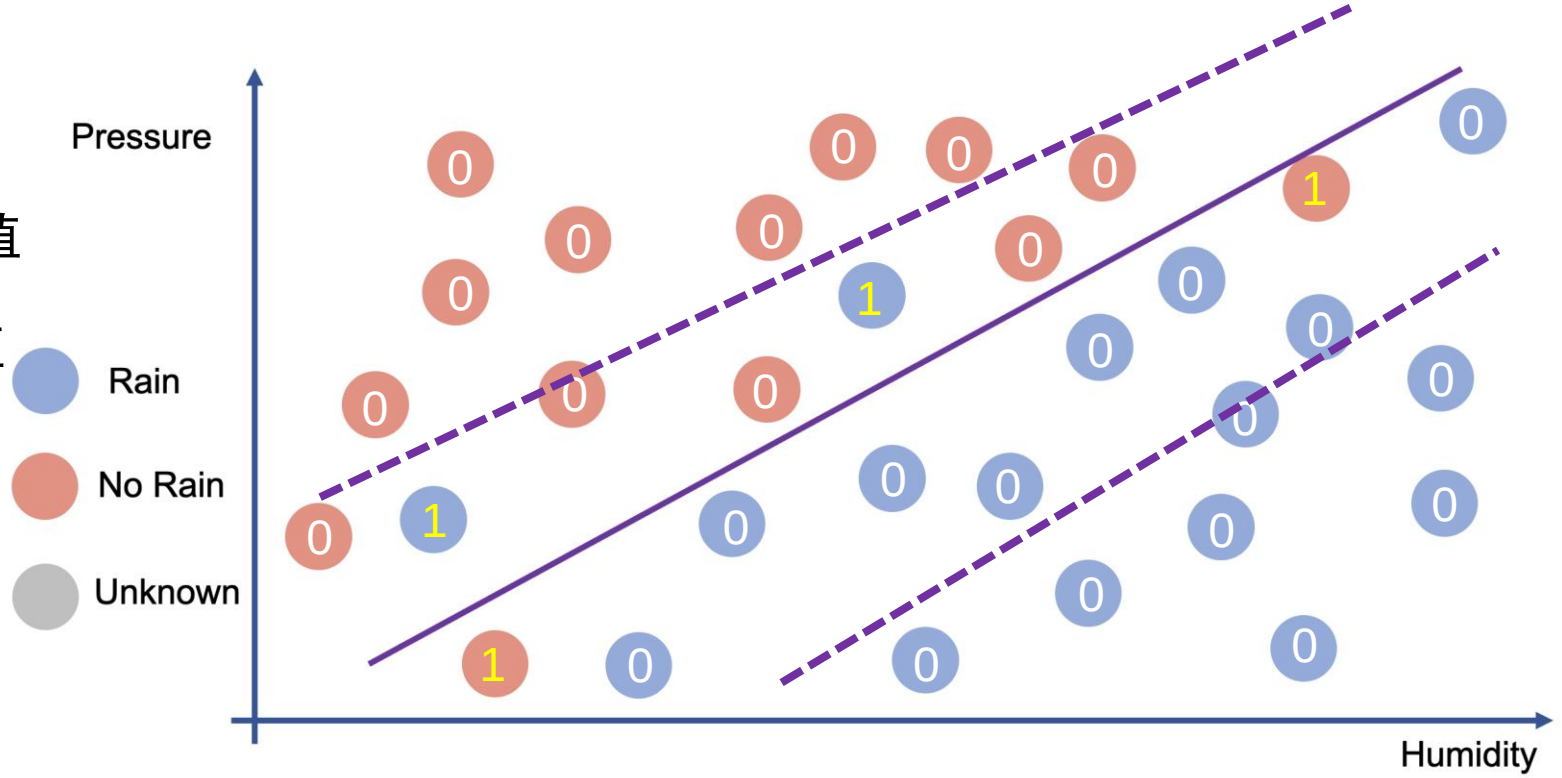
上海交通大学

和优化类似...

- 最小化损失函数 loss function
- 损失函数 loss function
  - 衡量假设的函数关系有多差的函数
  - 只要我们分类/预测出错，就会造成损失

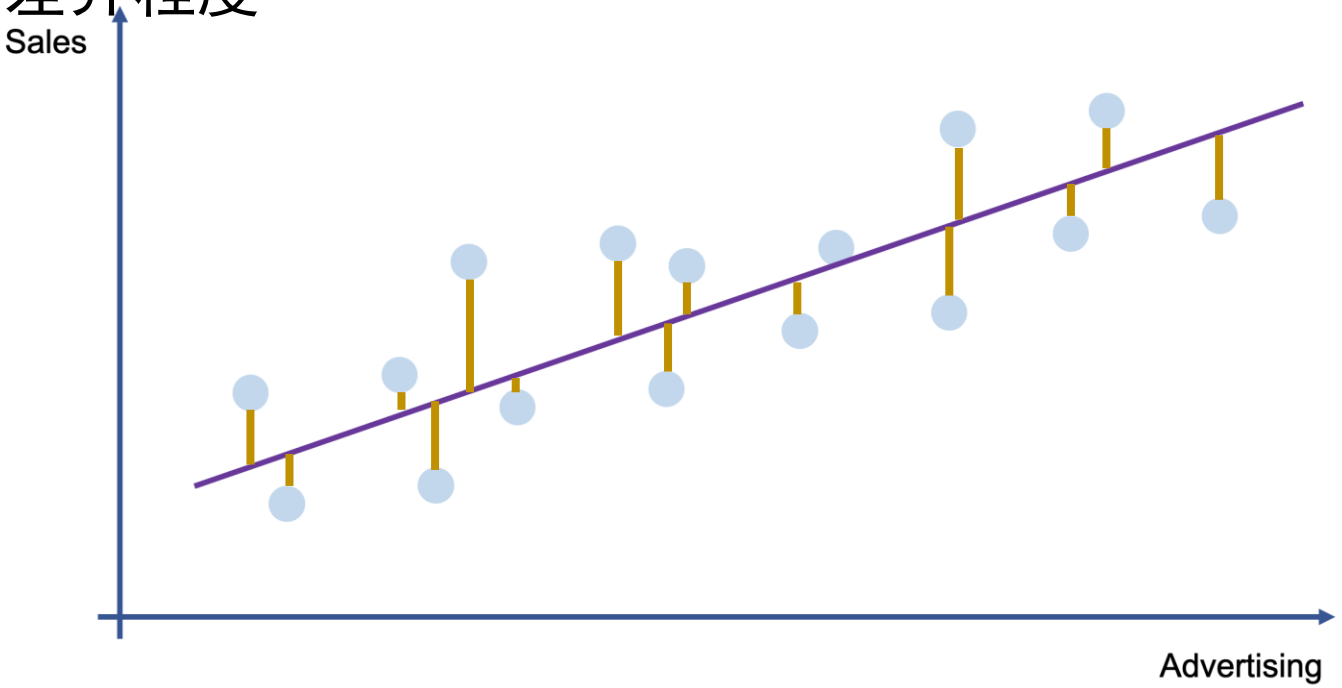
一些热门的损失函数

- 0-1 损失函数
  - 适用于**离散**值
  - L(真实值, 预测值) =
    - 0 如果 真实值 = 预测值
    - 1如果 真实值 ≠ 预测值



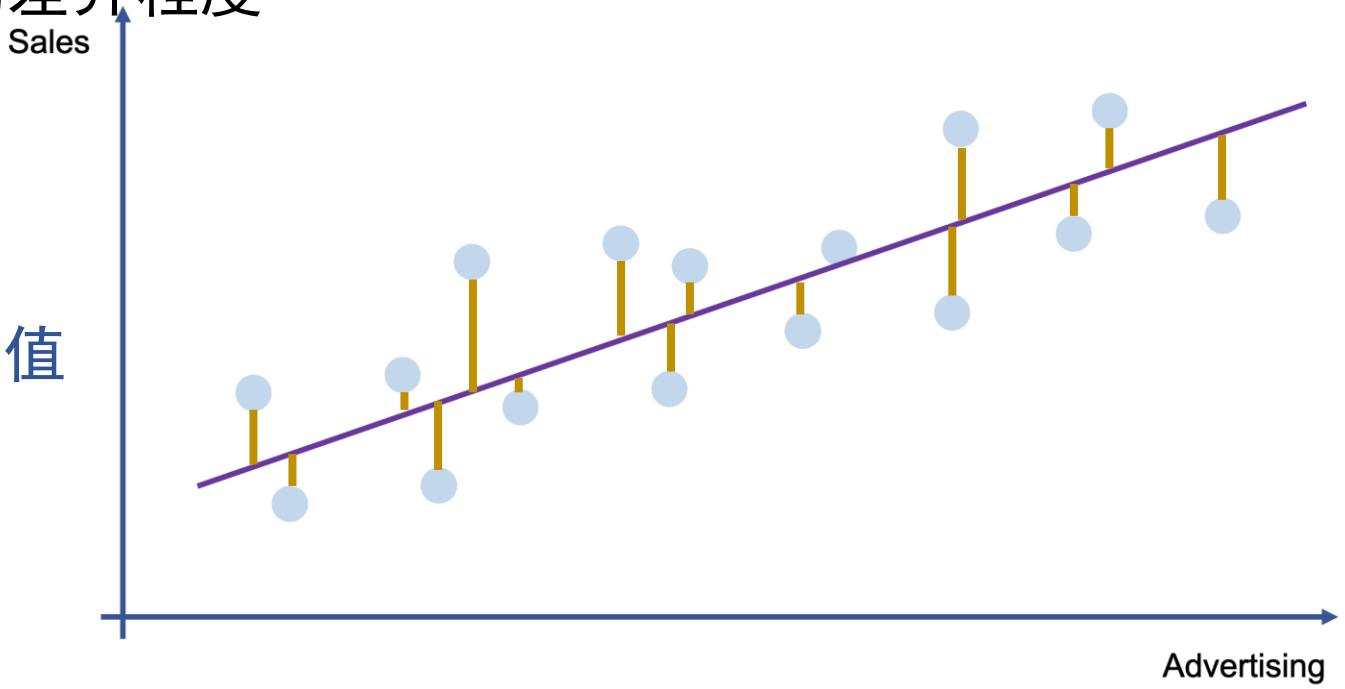
一些热门的损失函数

- L_1损失函数
  - 适用于**连续**值
  - 衡量每个预测值与真实值的差异程度
  - $L(\text{真实值}, \text{预测值}) =$
    - |真实值 − 预测值|



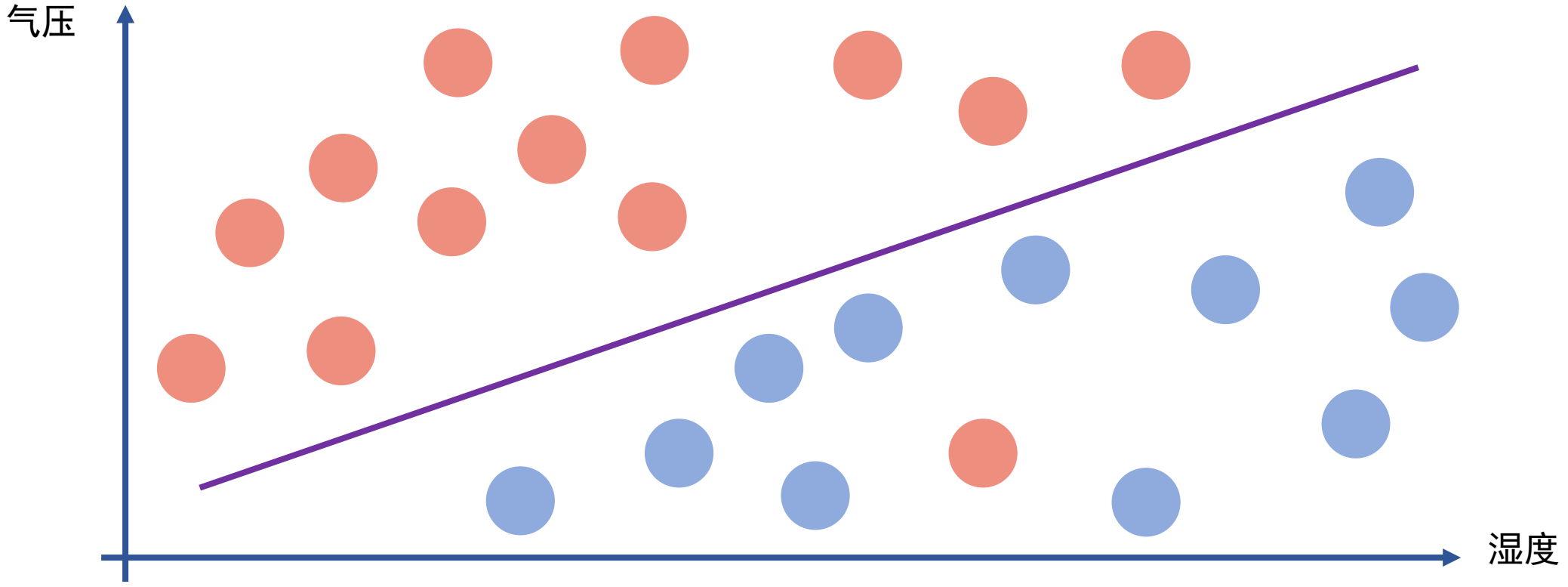
一些热门的损失函数

- L_2损失函数
  - 适用于**连续**值
  - 衡量每个预测值与真实值的差异程度
  - $L(\text{真实值}, \text{预测值}) =$
    - $(\text{真实值} - \text{预测值})^2$
  - L_2损失函数更严厉惩罚异常值



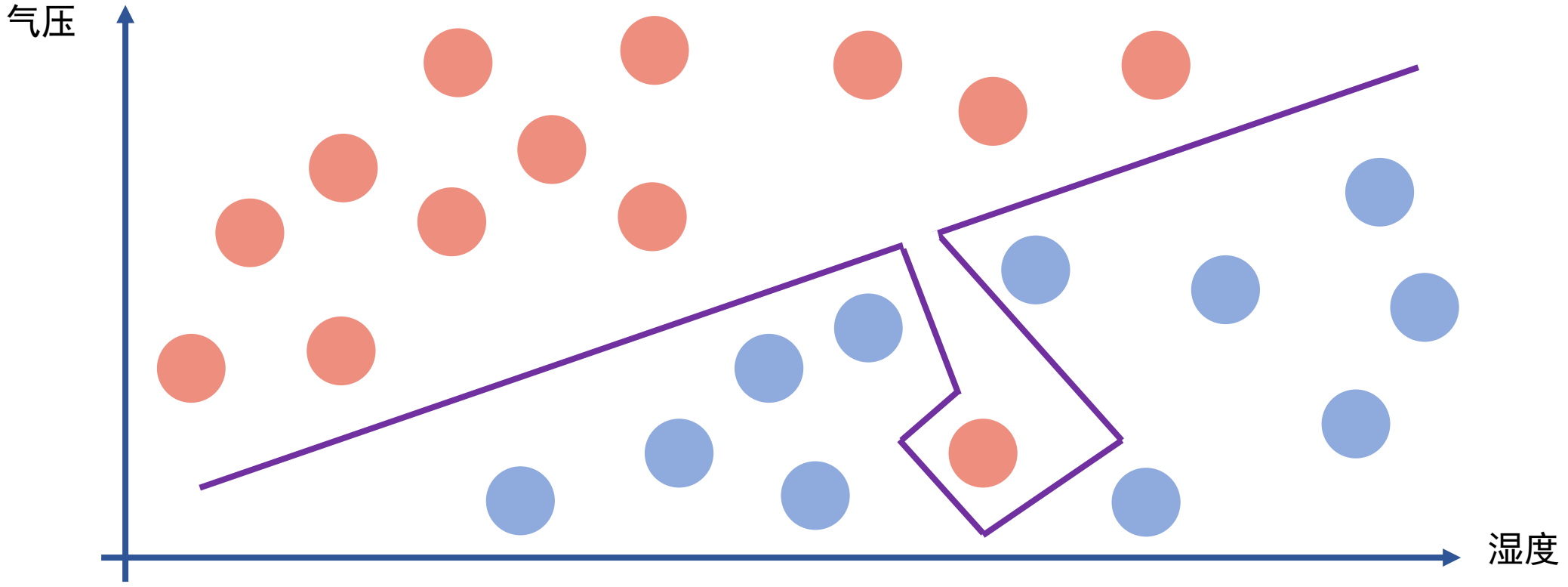
过拟合 Overfitting

- 模型过于适配特定数据集，因此可能无法泛化到其他的数据
  - 分类



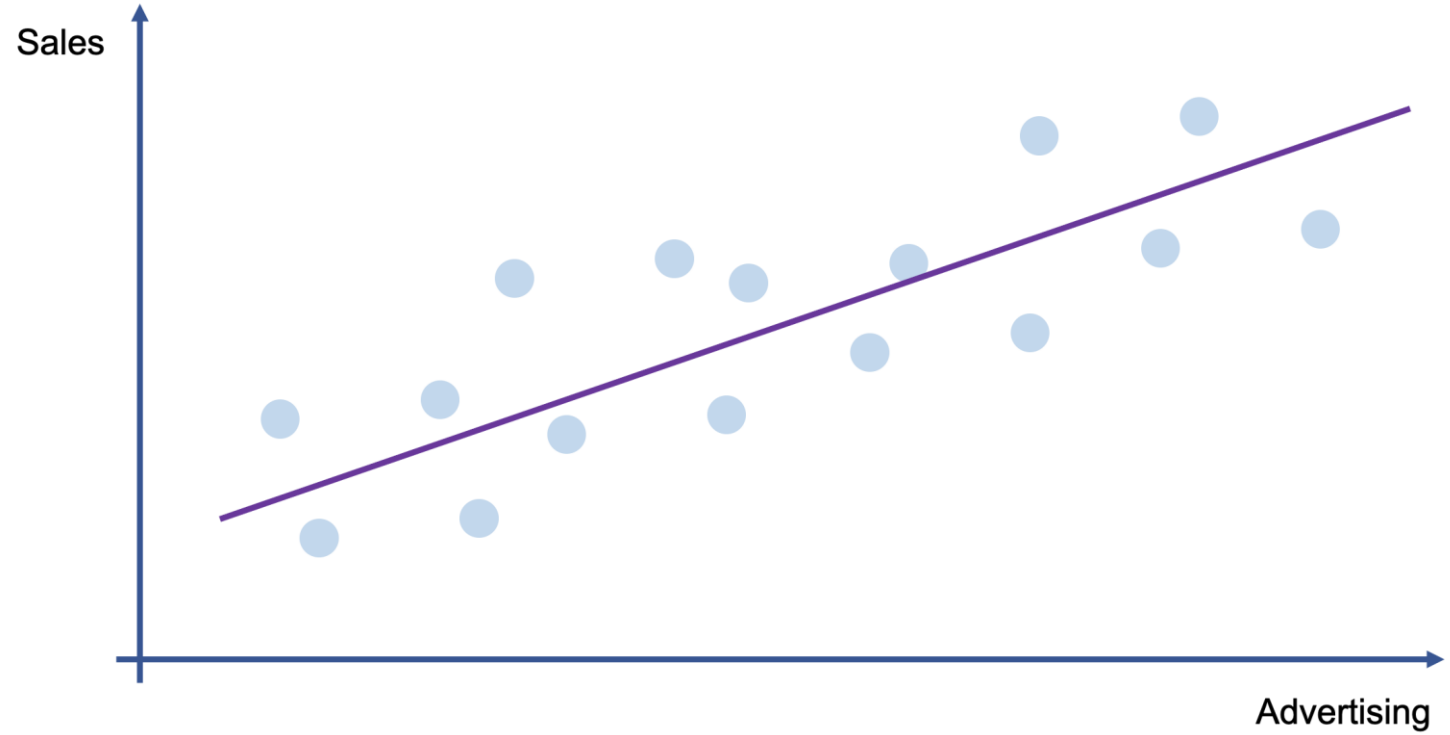
过拟合 Overfitting

- 模型过于适配特定数据集，因此可能无法泛化到其他的数据
  - 分类



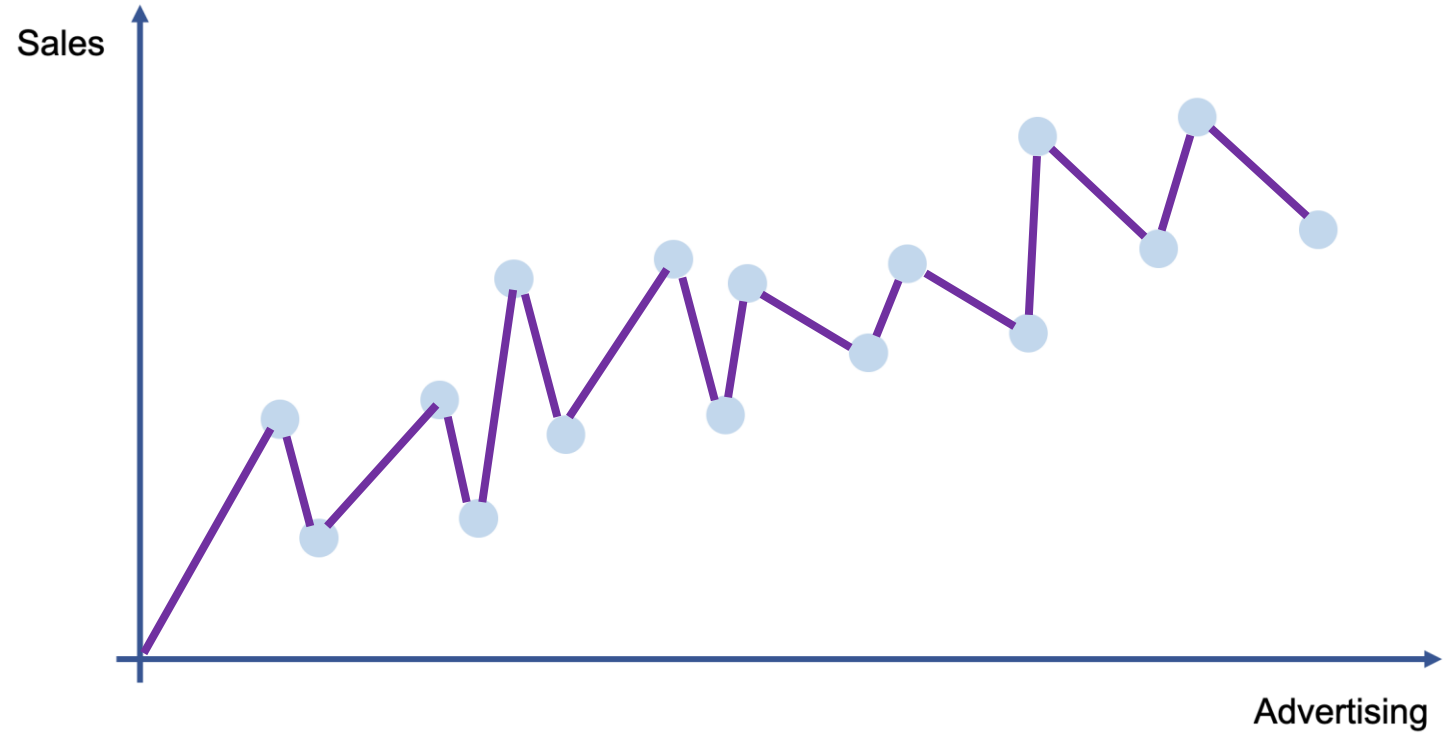
过拟合 Overfitting

- 模型过于适配特定数据集，因此可能无法泛化到其他的数据
  - 回归



过拟合 Overfitting

- 模型过于适配特定数据集，因此可能无法泛化到其他的数据
  - 回归



如何避免过拟合?

- 正则化 Regularization
  - 惩罚更复杂的假设，更喜欢简单、普适的假设

    $cost(\mathrm{h}) = loss(\mathrm{h}) + \lambda\, complexity(\mathrm{h})$

  - 如果λ很大，意味着惩罚更复杂的假设

如何避免过拟合?

- 留出法交叉验证 Holdout cross-validation
  - 将数据分成训练集(training set)和测试集(test set)。在训练集上学习模型，在测试集上评估模型
  - 缺点：未使用足够的数据来训练模型

如何避免过拟合?

- k次交叉验证 k-fold cross-validation
  - 将数据分成k个集合，并实验k次。每次实验将其中一个集合作为测试集，并将剩余数据当做训练集
    - 例：3次交叉验证 3-fold cross-validation

训练集 Training set	训练集 Training set	测试集 Testing set
训练集 Training set	测试集 Testing set	训练集 Training set
测试集 Testing set	训练集 Training set	训练集 Training set

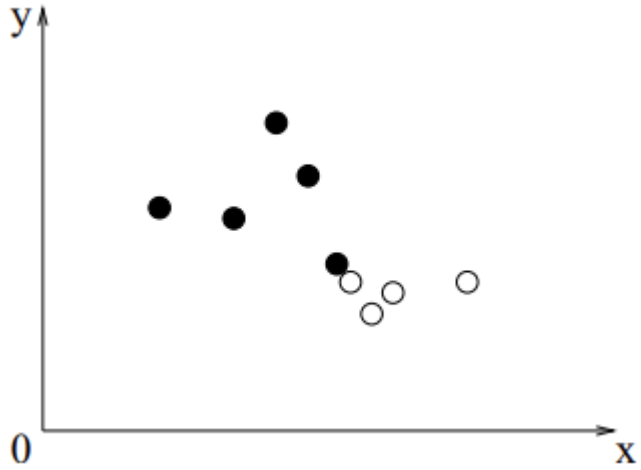
  - 误差Error：k次测试误差的平均值

练习 #8.1

- 有如下图所示的数据点。我们将使用KNN对他们进行分类。黑色的点的标记(label)为1，白色的点为0。采用0-1 损失函数。

① 如果我们采用1-NN，交叉验证的时候，采用留一法(leave-one-out)，即每次将1个数据点作为测试集，那么误差为多少？

② 如果我们采用3-NN，交叉验证的时候，采用留一法(leave-one-out)，那么误差为多少？



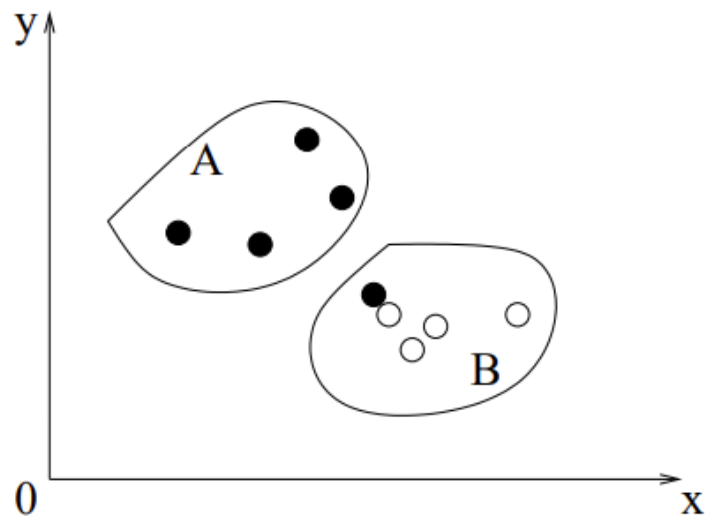
误差：错误次数/测试几次（是一个分数）

练习 #8.2

- 有如下图所示的数据点。我们将使用KNN对他们进行分类。黑色的点的标记(label)为1，白色的点为0。采用0-1 损失函数。我们使用2次交叉验证，将数据集分为A,B两个数据集。

① 如果我们采用1-NN，那么误差为多少？

② 如果我们采用3-NN，那么误差为多少？



k次交叉验证 k-fold cross-validation

- k比较小
  - 损失了大量的训练数据(1/K)，可能误差比较大
- k比较大(例如，留一法)
  - 基本没有损失什么数据，效率慢(需要学习N次)

衡量测试效果

- 损失函数
  - 如果数据点很多，loss自然就很大，不可比
- 其他指标：

① 灵敏性 Sensitivity / 查全率 Recall /真阳性率 True positive rate

$$\frac{N(\text{预测值} = 1,\ \text{真实值} = 1)}{N(\text{真实值} = 1)}$$

		预测值	
		+	-
真实值	+	True Positive	False Negative
	-	False Positive	True Negative

② 特异性 Specificity / 真阴性率 True negative rate

$$\frac{N(\text{预测值} = 0, \text{真实值} = 0)}{N(\text{真实值} = 0)}$$

③ 查准率 Precision / 阳性预测值 Positive predicted value

$$\frac{N(\text{预测值} = 1, \text{真实值} = 1)}{N(\text{预测值} = 1)}$$

衡量测试效果

- F_1 值
  - 真阳性率 precision 和 阳性预测值 recall 的调和平均值

$$F_1 = \frac{2}{\frac{1}{recall} + \frac{1}{precision}} = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

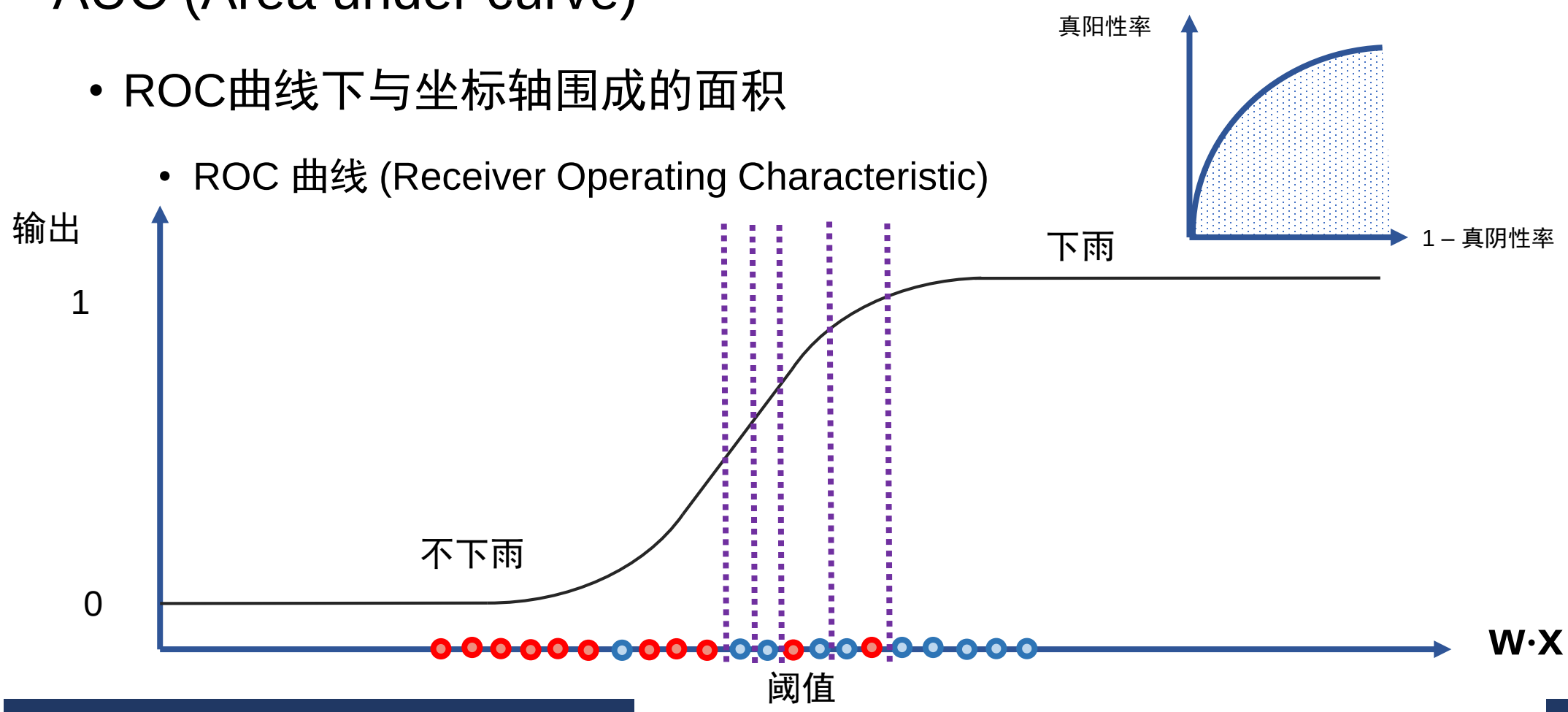
  - 衡量模型的整体表现：取值范围0~1

F_1 值	含义
>0.9	非常好
0.8 – 0.9	好
0.5 – 0.8	还行
<0.5	不好

来源: https://stephenallwright.com/interpret-f1-score/

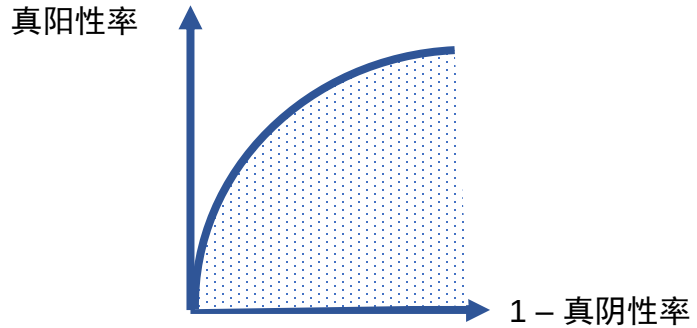
衡量测试效果

- AUC (Area under curve)
  - ROC曲线下与坐标轴围成的面积
    - ROC 曲线 (Receiver Operating Characteristic)



衡量测试效果

- AUC (Area under curve)
  - ROC曲线下与坐标轴围成的面积
    - ROC 曲线 (Receiver Operating Characteristic)
  - 衡量模型准确分类的能力：取值范围0~1
    - 1 很好, 0.5 随机分类



AUC	含义
>0.8	非常好
0.7 – 0.8	好
0.5 – 0.7	还行
0.5	和随机选择一样
<0.5	错误的分类

https://stephenallwright.com/interpret-auc-score/

有问题吗?

- 请随时举手提问。

BUSS 3620.人工智能导论

#3. 代码示例：识别假币

刘佳璐

安泰经济与管理学院

上海交通大学

背景与数据集说明

- 警官
  - 收缴了一批假币，假币的数据记录在 banknotes_history.csv
  - 近日，有一批新的纸币需要辨认是否为假币
    - 有一些纸币是假币
    - 新的纸币数据记录在 banknotes_new.csv
    - 目标：根据旧的假币数据，预测哪一张纸币是假币
  - 一个月以后，权威机构指出这批新的纸币哪些是假币
    - 数据记录在banknotes_new_result.csv
    - 可以用来评估我们预测的结果

代码框架

- 读取用于训练的历史数据
- 生成和训练一个机器学习模型
- 预测
- 衡量模型表现

读取数据

- 用 pandas 包

```
# Read historical data -- for training
data = pd.read_csv('banknotes_history.csv',index_col=0)
```

- 处理数据
  - NA值

```
data = data.dropna(subset=['label'])  # remove banknotes that contains NA value
data = data.fillna(data.mean()) # fill in the banknotes with the column mean
```

  - 将输入与输出分隔开
  - 定比(Scale)数据
    - 帮助算法快速找到损失函数的最小值

```
# Prepare the outcome y, and feature dataframe
label = data.pop('label')
train_data=pd.DataFrame(preprocessing.scale(data))
```

处理数据的小技巧

- 缺失值 Missing Value
  - 删除含有缺失值的数据
  - 用平均值填补缺失值
- 分类值 Categorical Value (例如，星期一、星期二)
  - 更改为整数
- 布尔值 Boolean values (True或False)
  - 更改为整数

生成和训练一个机器学习模型

- https://scikit-learn.org/stable/user_guide.html

Examples

```
>>> from sklearn.datasets import load_digits
>>> from sklearn.linear_model import Perceptron
>>> X, y = load_digits(return_X_y=True)
>>> clf = Perceptron(tol=1e-3, random_state=0)
>>> clf.fit(X, y)
Perceptron()
>>> clf.score(X, y)
0.939...
```

Methods

`decision_function`(X)	Predict confidence scores for samples.
`densify`()	Convert coefficient matrix to dense array format.
`fit`(X, y[, coef_init, intercept_init, ...])	Fit linear model with Stochastic Gradient Descent.
`get_params`([deep])	Get parameters for this estimator.
`partial_fit`(X, y[, classes, sample_weight])	Perform one epoch of stochastic gradient descent on given samples.
`predict`(X)	Predict class labels for samples in X.
`score`(X, y[, sample_weight])	Return the mean accuracy on the given test data and labels.
`set_params`(**params)	Set the parameters of this estimator.
`sparsify`()	Convert coefficient matrix to sparse format.

生成和训练一个机器学习模型

- https://scikit-learn.org/stable/user_guide.html

```
# Build and train a machine learning model
model = KNeighborsClassifier(n_neighbors=1) #nearest neighbor
model = Perceptron() #Perceptron
model = svm.SVC() # support vector machine
model = tree.DecisionTreeClassifier() # decision trees
model = RandomForestClassifier(n_estimators=10) # random forest
model = GaussianNB() #Naive Bayes


model.fit(train_data, label)
```

预测

- 读取需要预测的数据集(只有X没有Y)
- 处理数据
- 预测
- 记录并匹配纸币序号

```
# Make prediction
pred = pd.read_csv('banknotes_new.csv',index_col=0)
index = pred.index

pred=pd.DataFrame(preprocessing.scale(pred))
pred.index=index
```

```
#make prediction
outcome=pd.DataFrame(model.predict(pred))
# model.decision_function(pred)
outcome.index=index
outcome.columns=['predicted']
```

衡量模型表现

- 读取测试集的真实分类
- 与预测结果数据合并
- 计算各种模型指标并输出结果

```
## Evaluate the performance
actual=pd.read_csv('banknotes_new_result.csv',index_col=0)
outcome=outcome.merge(actual,left_index=True,right_index=True,how='inner')
```

```
# Compute how well we performed
correct = 0
incorrect = 0
total = 0
for actual, predicted in zip(outcome.label, outcome.predicted):
    total += 1
    if actual == predicted:
        correct += 1
    else:
        incorrect += 1

sensitivity=recall_score(outcome.label, outcome.predicted, pos_label=1)
specificity=recall_score(outcome.label, outcome.predicted, pos_label=0)
precision = precision_score(outcome.label, outcome.predicted)
F1Score = f1_score(outcome.label, outcome.predicted)
auc = roc_auc_score(outcome.label, outcome.predicted)
```

k次交叉验证 k-fold cross-validation

- 将训练数据分成训练集和测试集

```
### if want to split training and testing dataset

X_training, X_testing, y_training, y_testing = train_test_split(
    data, label, test_size=0.4
)
X_training
y_training
```

- k次交叉验证 k-fold cross-validation

```
## k-fold cross validation
scores = cross_val_score(model, train_data, label, cv=5) # five-fold cross validation
scores # accuracy
```

有问题吗?

- 请随时举手提问。

BUSS 3620.人工智能导论

#4. 无监督学习

刘佳璐

安泰经济与管理学院

上海交通大学

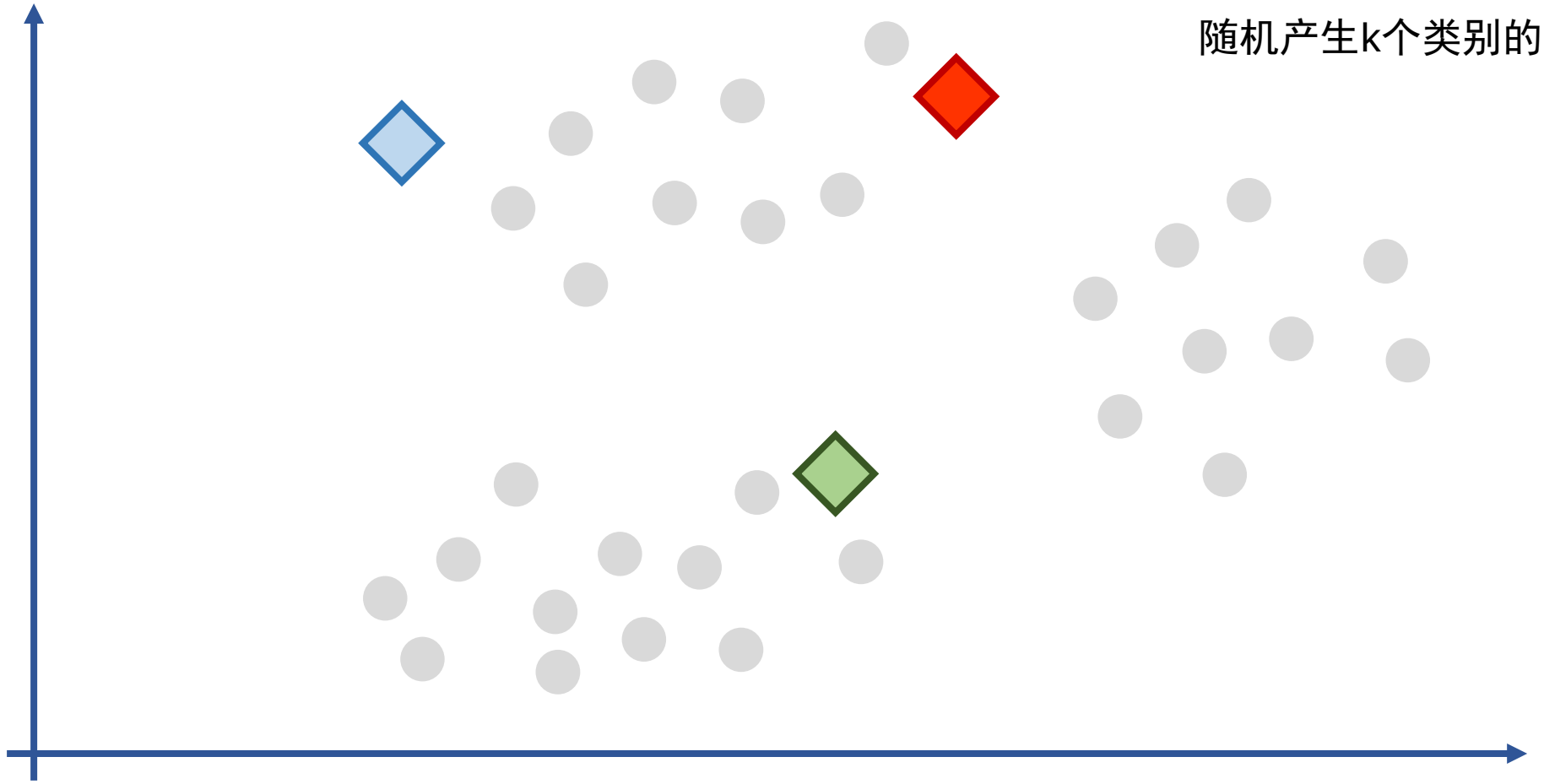
无监督学习 Unsupervised learning

- 基于输入数据，没有任何额外反馈，学习规律
- 聚类 Clustering
  - 将不同的东西分类，相似的东西分成同一类。与分类任务不同的是，没有提前标注好的类别。
  - 应用
    - 基因研究
    - 图像分割
    - 市场调查
    - 医学影像
    - 社交网络分析

k均值聚类算法 k-means clustering

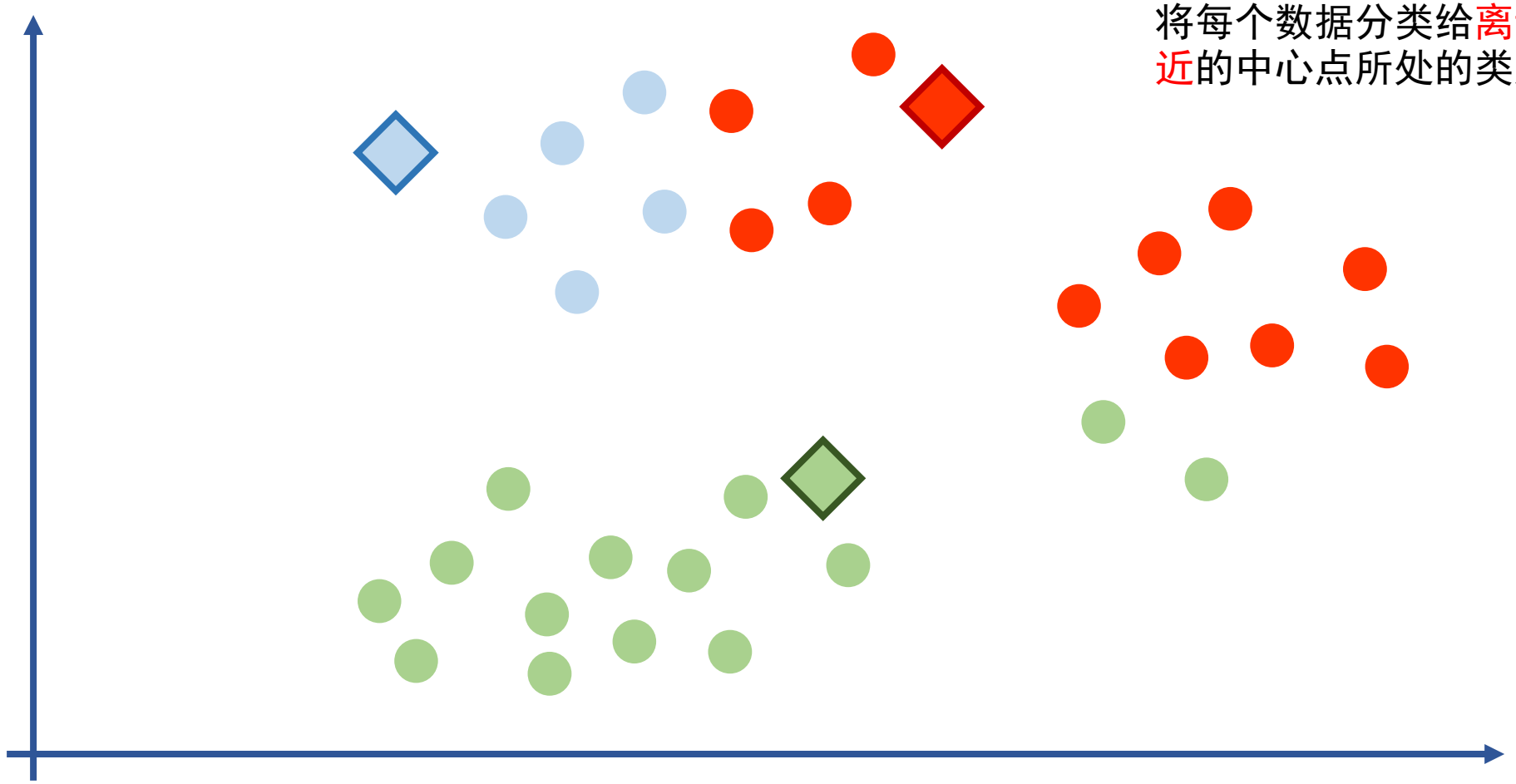
- 数据分类
  - 将数据分配给每一个类别
  - 更新类别的中心点
  - 重复这一过程

k均值聚类算法 k-means clustering



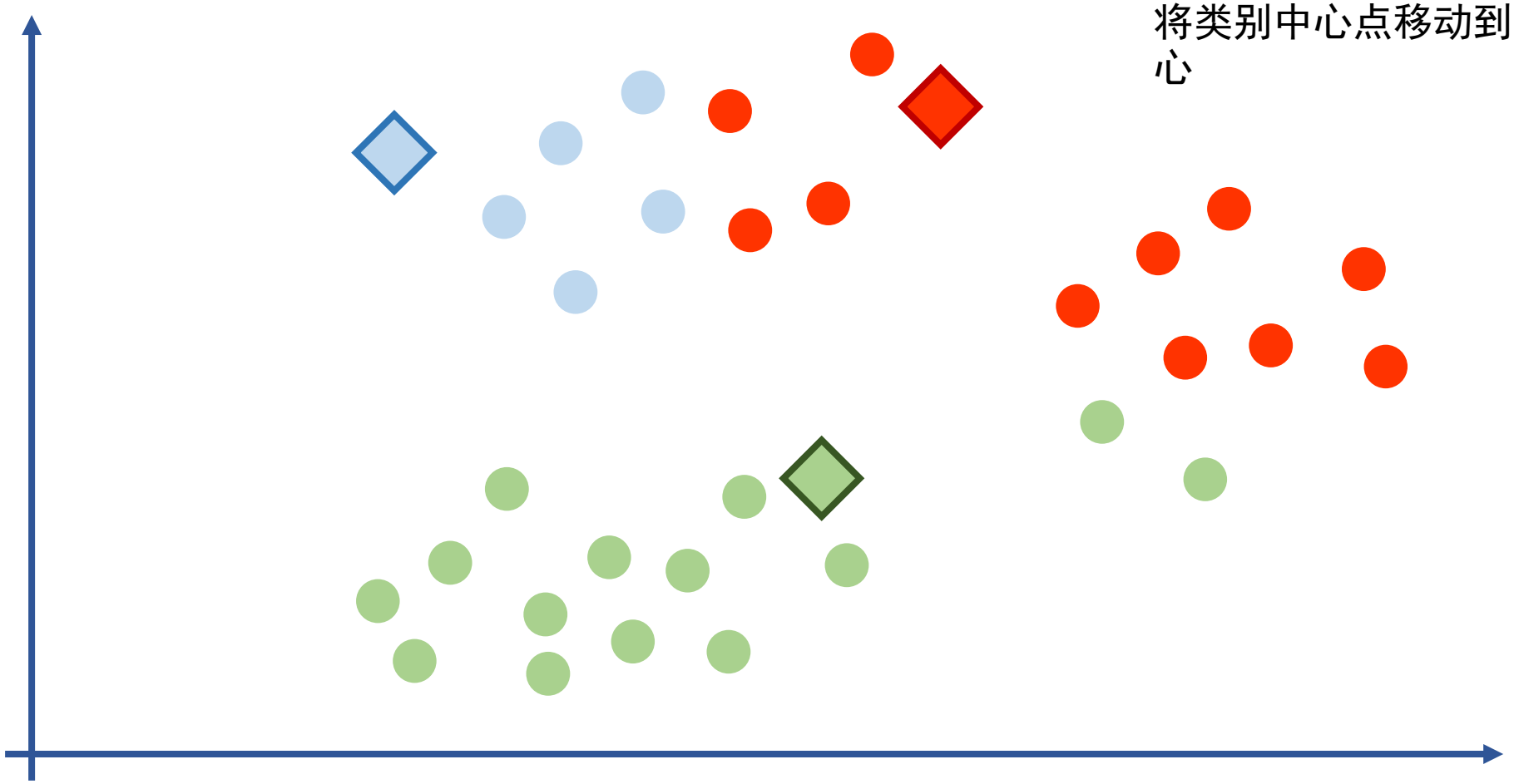
k均值聚类算法 k-means clustering

将每个数据分类给离该数据最近的中心点所处的类别



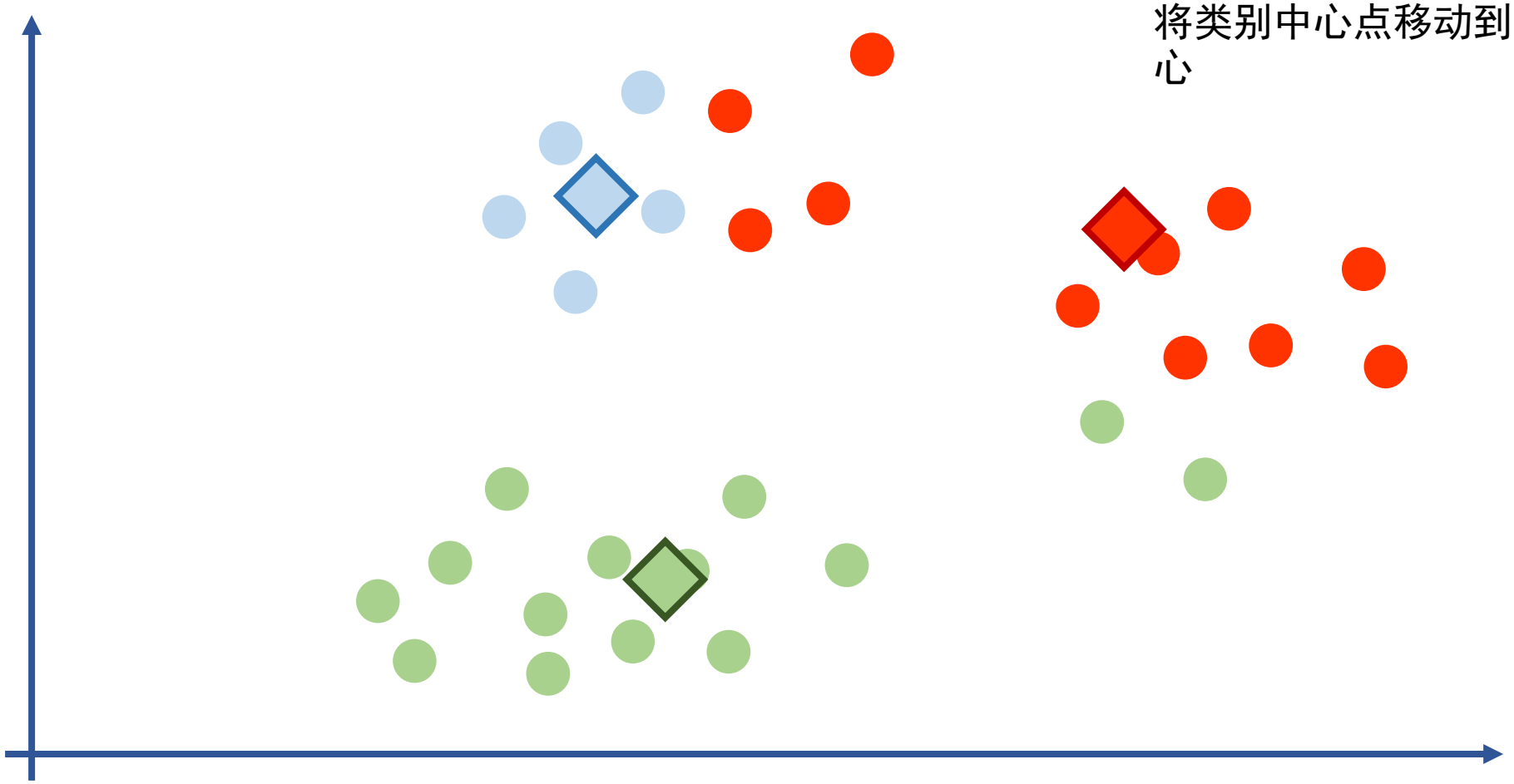
k均值聚类算法 k-means clustering

将类别中心点移动到类别的中心



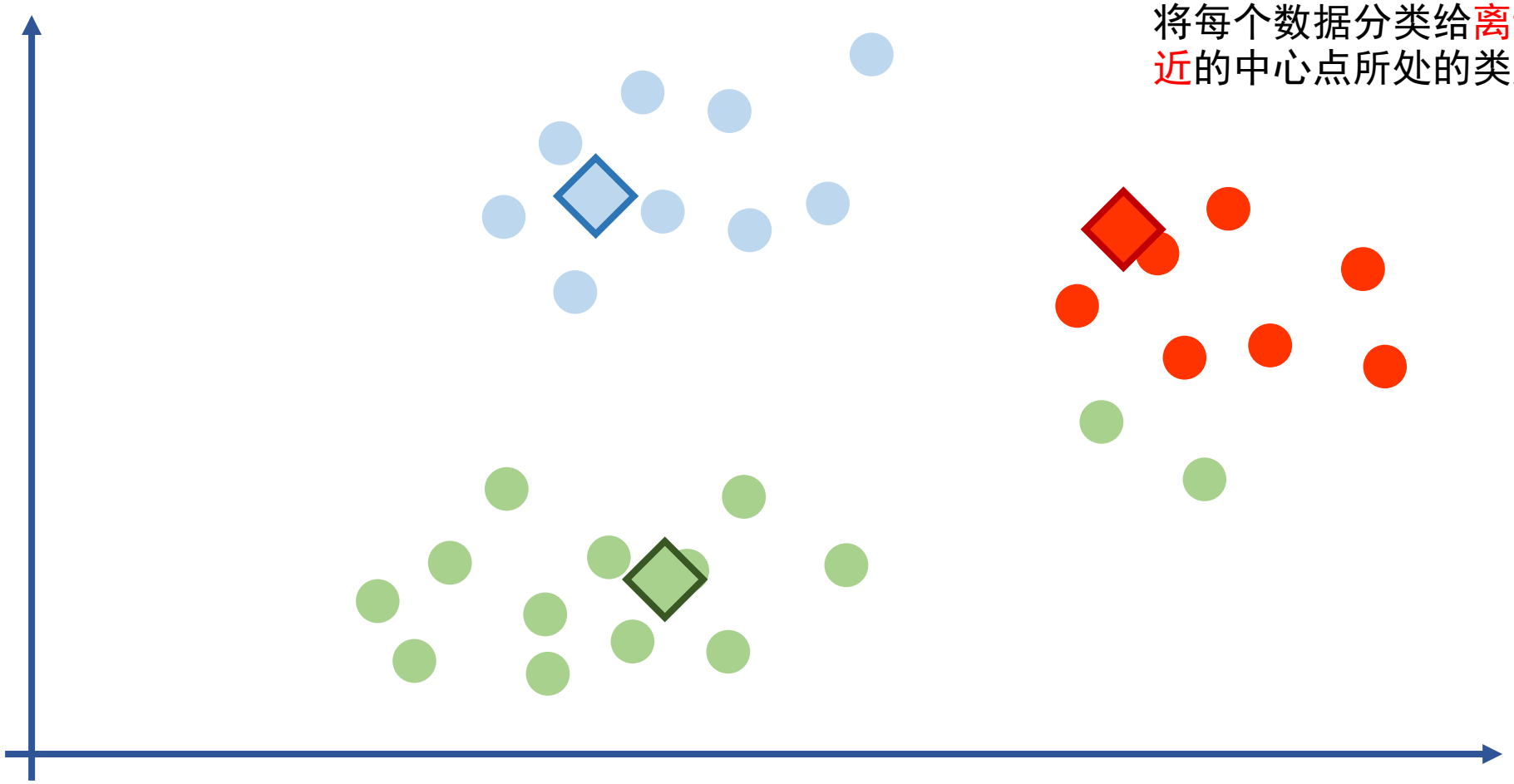
k均值聚类算法 k-means clustering

将类别中心点移动到类别的中心



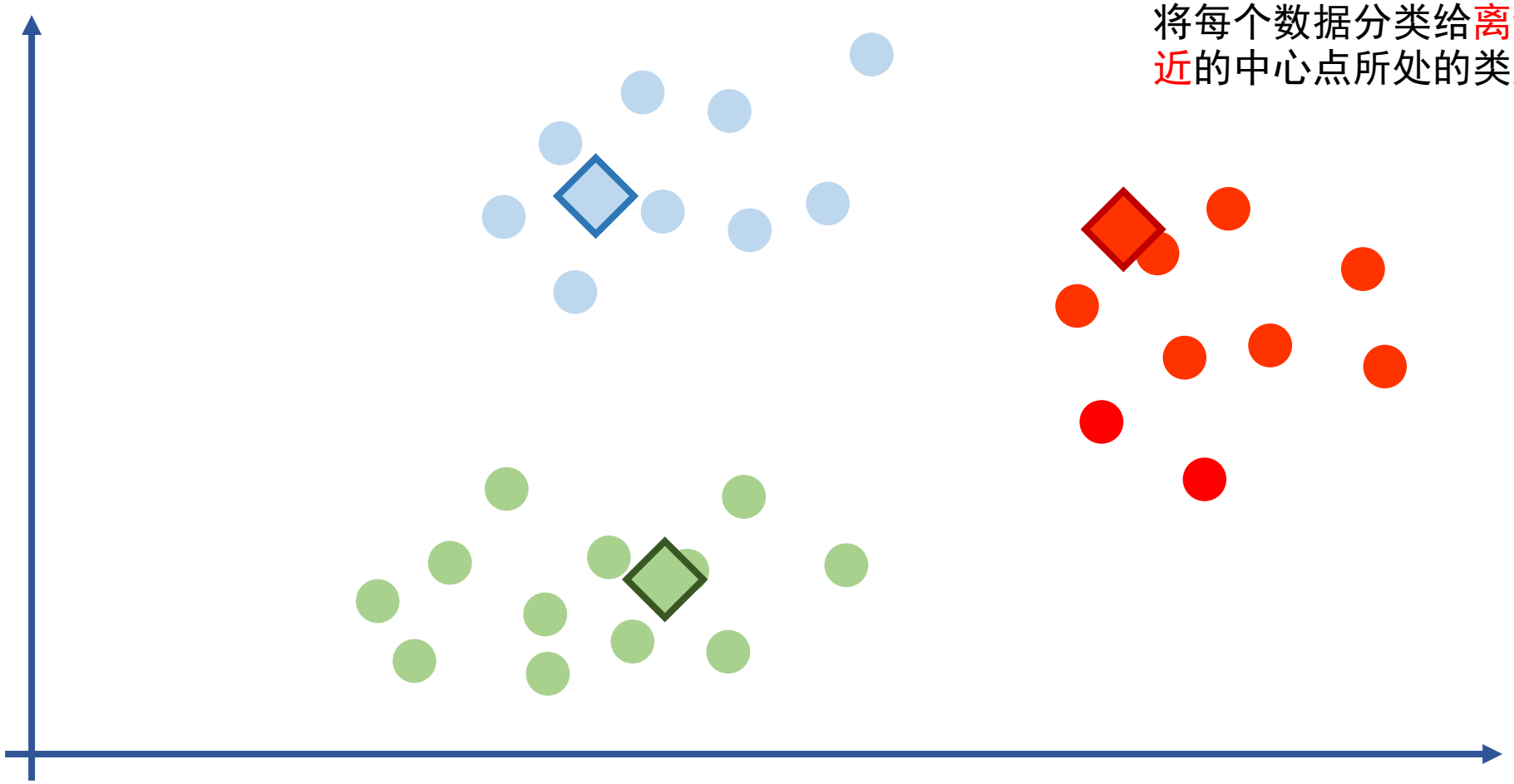
k均值聚类算法 k-means clustering

将每个数据分类给离该数据最近的中心点所处的类别



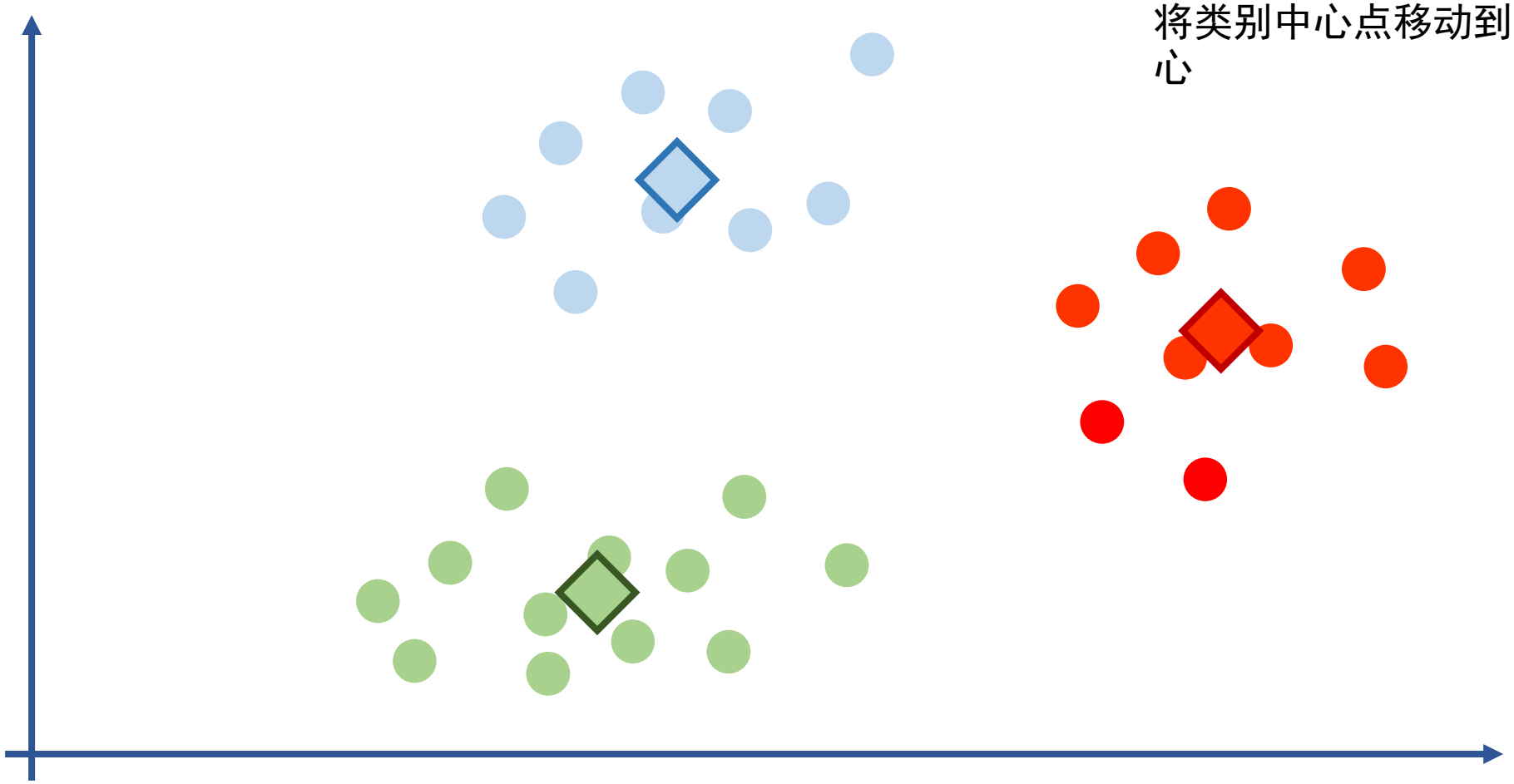
k均值聚类算法 k-means clustering

将每个数据分类给离该数据最近的中心点所处的类别



k均值聚类算法 k-means clustering

将类别中心点移动到类别的中心



练习 #9

- 使用K均值法(K-means)将下面8个数据点分成3类。距离函数为直线距离。3个类别C1,C2,C3的初始的3个中心点分别为A1, A4, A7。每个点之间的距离列在下表中。

	坐标	A1	A2	A3	A4	A5	A6	A7	A8
A1	(2, 10)	0	$\sqrt{25}$	$\sqrt{36}$	$\sqrt{13}$	$\sqrt{50}$	$\sqrt{52}$	$\sqrt{65}$	$\sqrt{5}$
A2	(2, 5)	$\sqrt{25}$	0	$\sqrt{37}$	$\sqrt{18}$	$\sqrt{25}$	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{20}$
A3	(8, 4)	$\sqrt{36}$	$\sqrt{37}$	0	$\sqrt{25}$	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{53}$	$\sqrt{41}$
A4	(5, 8)	$\sqrt{13}$	$\sqrt{18}$	$\sqrt{25}$	0	$\sqrt{13}$	$\sqrt{17}$	$\sqrt{52}$	$\sqrt{2}$
A5	(7, 5)	$\sqrt{50}$	$\sqrt{25}$	$\sqrt{2}$	$\sqrt{13}$	0	$\sqrt{2}$	$\sqrt{45}$	$\sqrt{25}$
A6	(6, 4)	$\sqrt{52}$	$\sqrt{17}$	$\sqrt{2}$	$\sqrt{17}$	$\sqrt{2}$	0	$\sqrt{29}$	$\sqrt{29}$
A7	(1, 2)	$\sqrt{65}$	$\sqrt{10}$	$\sqrt{53}$	$\sqrt{52}$	$\sqrt{45}$	$\sqrt{29}$	0	$\sqrt{58}$
A8	(4, 9)	$\sqrt{5}$	$\sqrt{20}$	$\sqrt{41}$	$\sqrt{2}$	$\sqrt{25}$	$\sqrt{29}$	$\sqrt{58}$	0

① 哪些点会被分配到C1,C2,C3?

② 一个回合以后，3个中心点更新为?

有问题吗?

- 请随时举手提问。