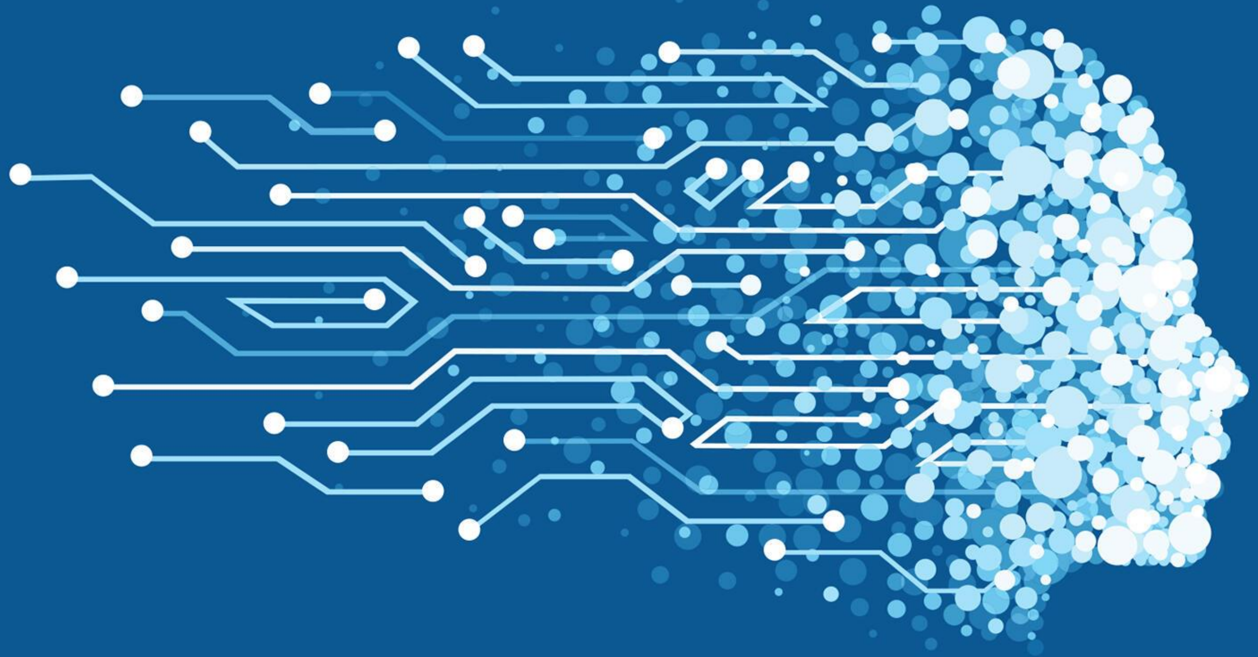


《机器学习》第三章 逻辑回归

2023年10月

上节回顾

监督学习

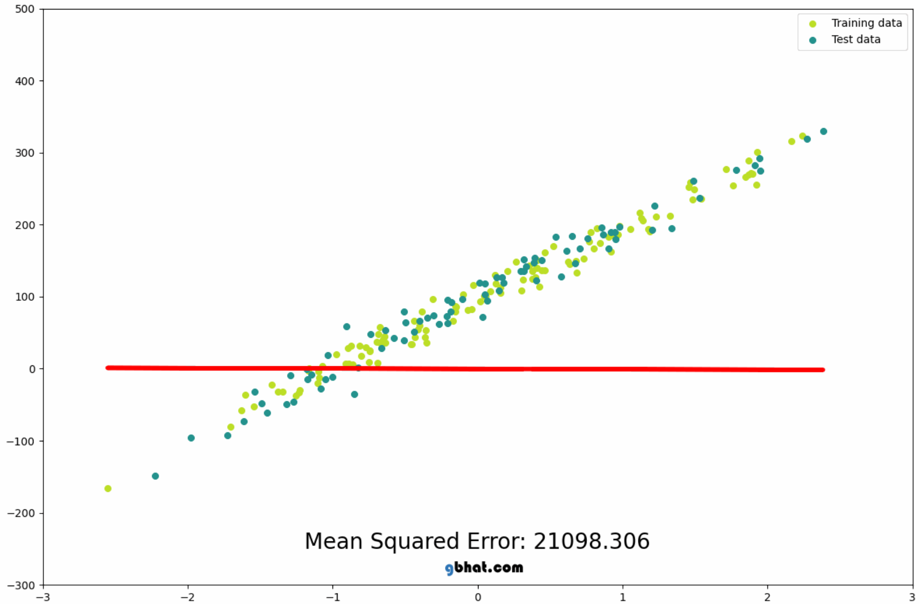
回归

Regression

分类

Classification

线性回归 $h(x) = w_0 + w_1x_1 + w_2x_2 + ... + w_nx_n$



本章目录

1.分类问题

1.分类问题

(a) 公共安全

(b) 金融风控

(c) 媒体监管

分类预测 **离散标签**

1.分类问题

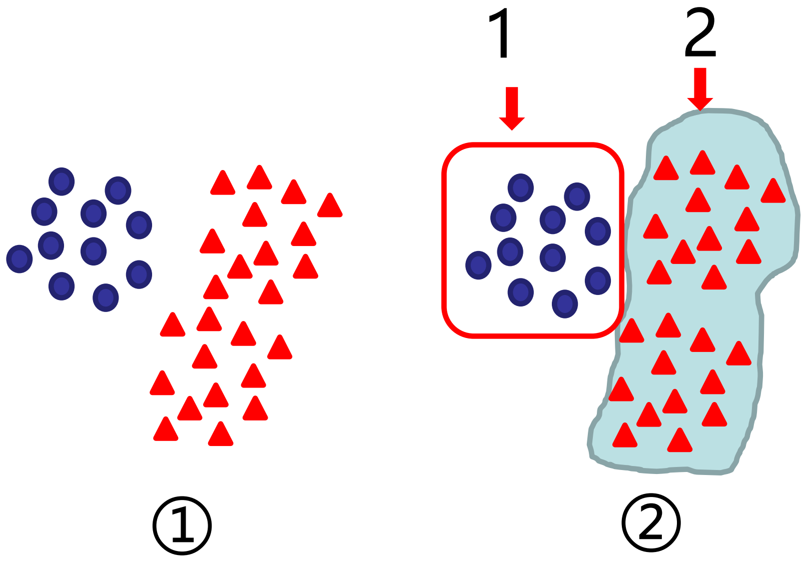
二分类

我们先从用蓝色圆形数据定义为类型1，

其余数据为类型2；

只需要分类1次

步骤：①->②



二分类

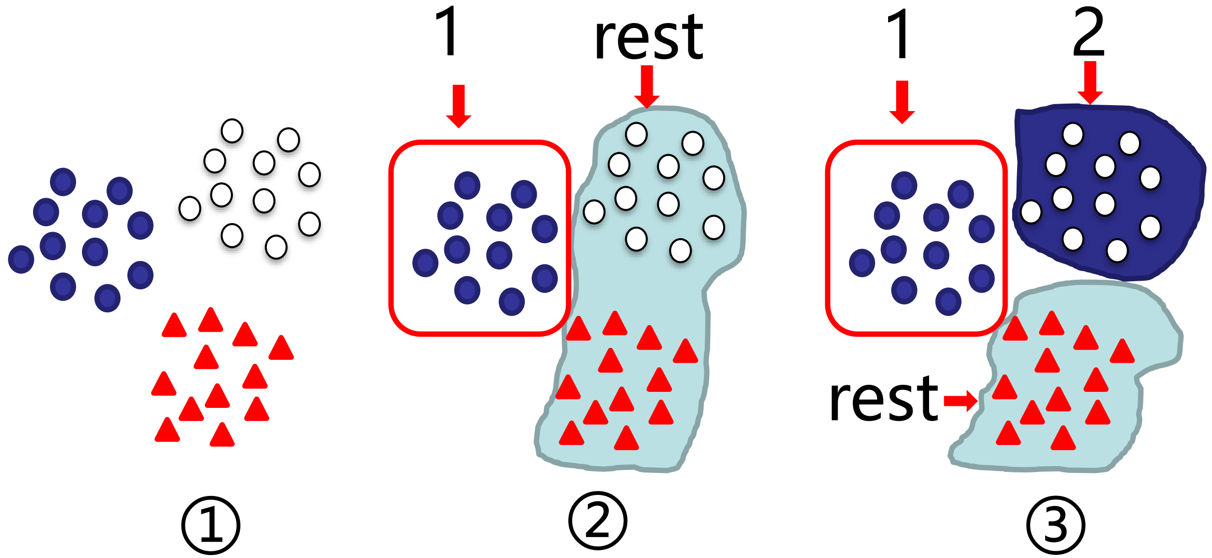
1.分类问题

多分类

我们先定义其中一类为类型1（正类），其余数据为负类（rest）；

接下来去掉类型1数据，剩余部分再次进行二分类，分成类型2和负类；如果有n类，那就需要分类n-1次

步骤：①->②->③->……

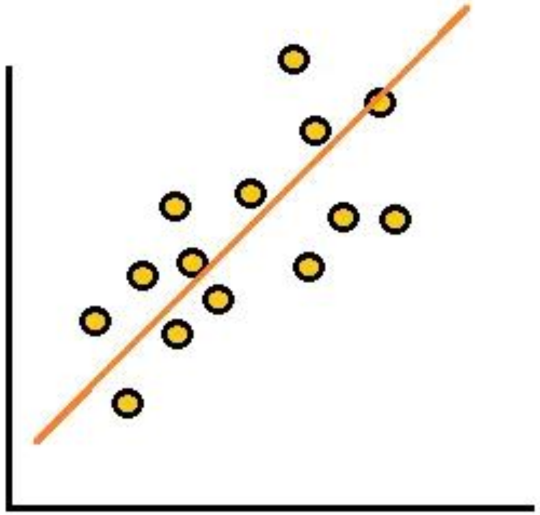


One-vs-All (One-vs-Rest)

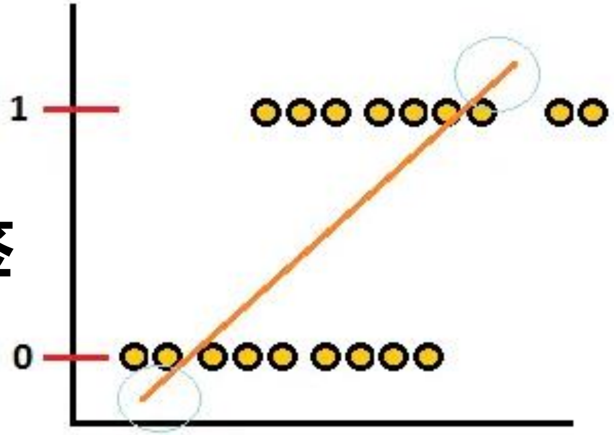
一对多 (一对余)

1.分类问题

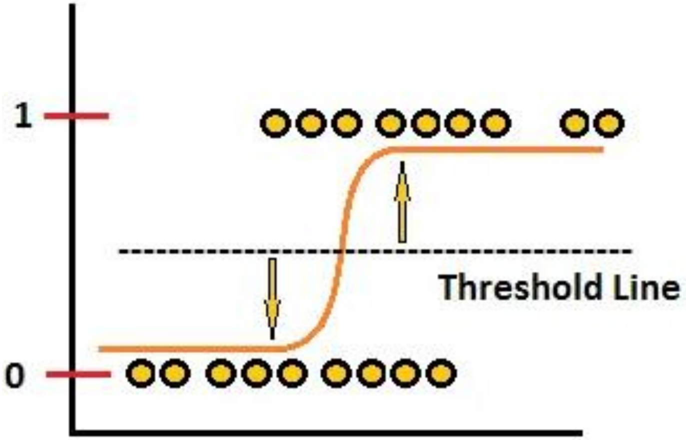
连续标签



离散标签



如何用线性模型解决分类问题?



2.Sigmoid函数

2.Sigmoid函数

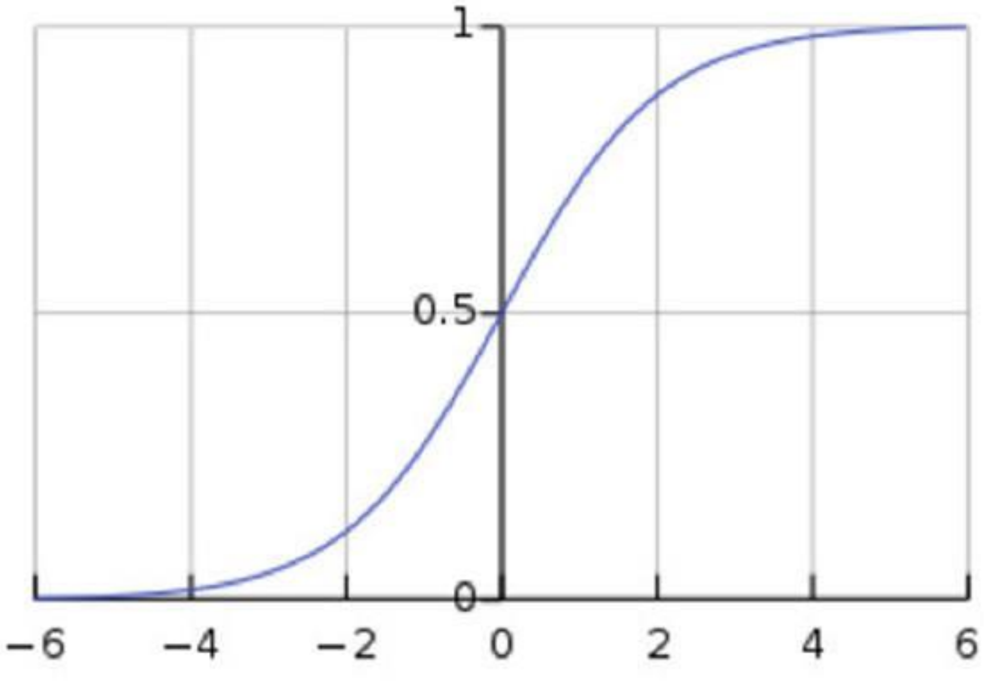
Sigmoid 函数

$g(z)$代表一个常用的S形逻辑函数（Sigmoid function）：

$$g(z) = \frac{1}{1+e^{-z}}$$

和线性模型 z=$w^T x + b$ 合起来，得到逻辑回归模型的假设函数:

$$\hat{y} = h(x) = \frac{1}{1+e^{-(w^T x+b)}}$$



当$g(z)$大于等于0.5时，预测 y=1

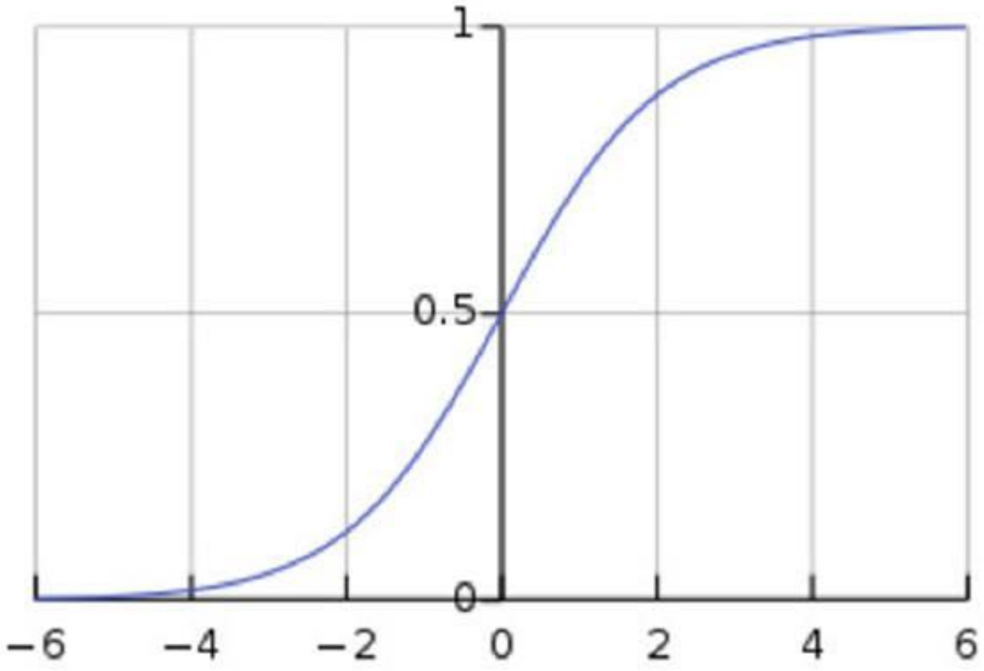
当$g(z)$小于0.5时，预测 y=0

注意：若表达式 $h(x) = z = w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n + b = w^Tx + b$，则$b$可以融入到$w_0$，即：$z$=$w^Tx$

2.Sigmoid函数

Sigmoid函数 $g(z)=\frac{1}{1+e^{-z}}$ 易求导，适合梯度下降更新：

$$
\begin{aligned}
g'(z) &= \left(\frac{1}{1+e^{-z}}\right)' \\
&= \frac{e^{-z}}{(1+e^{-z})^2} \\
&= \frac{1}{1+e^{-z}}\frac{e^{-z}}{1+e^{-z}} \\
&= g(z)\big(1-g(z)\big)
\end{aligned}
$$



$g(z)$

3.逻辑回归求解

3.逻辑回归求解

假设一个二分类模型：

$$p(y=1|x;w)=h(x)$$
$$p(y=0|x;w)=1-h(x)$$

则预测的似然估计如下：

$$p(y|x;w)=(h(x))^{y}(1-h(x))^{1-y}$$

3.逻辑回归求解

损失函数为交叉熵：

$$L(\hat{y},y) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$$

$\hat{y}$ 表示预测值$h(x)$

y 表示真实值

为了衡量算法在全部训练样本上的表现如何，我们需要定义一个算法的代价函数，算法的代价函数是对m个样本的损失函数求和然后除以m:

代价函数

$$J(w) = \frac{1}{m}\sum_{i=1}^{m} L\left(\hat{y}^{(i)}, y^{(i)}\right) = \frac{1}{m}\sum_{i=1}^{m}\left(-y^{(i)}\log\hat{y}^{(i)} - (1-y^{(i)})\log(1-\hat{y}^{(i)})\right)$$

3.逻辑回归求解

梯度下降更新方法：

$$w_j := w_j - \alpha \frac{\partial J(w)}{\partial w}$$

梯度下降求解：

$$\frac{\partial}{\partial w_j} J(w) = \frac{1}{m} \sum_{i=1}^{m} \left(h\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$$

得到模型参数更新方法

$$w_j := w_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$$

4.逻辑回归代码实现

4.逻辑回归代码实现

Sigmoid 函数

$$g(z) = \frac{1}{1+e^{-z}}$$

```
def sigmoid(z):
    return 1 / (1 + np.exp(-z))
```

代价函数

$$J(w) = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)}\log(h(x^{(i)})) + (1-y^{(i)})\log(1-h(x^{(i)})))$$

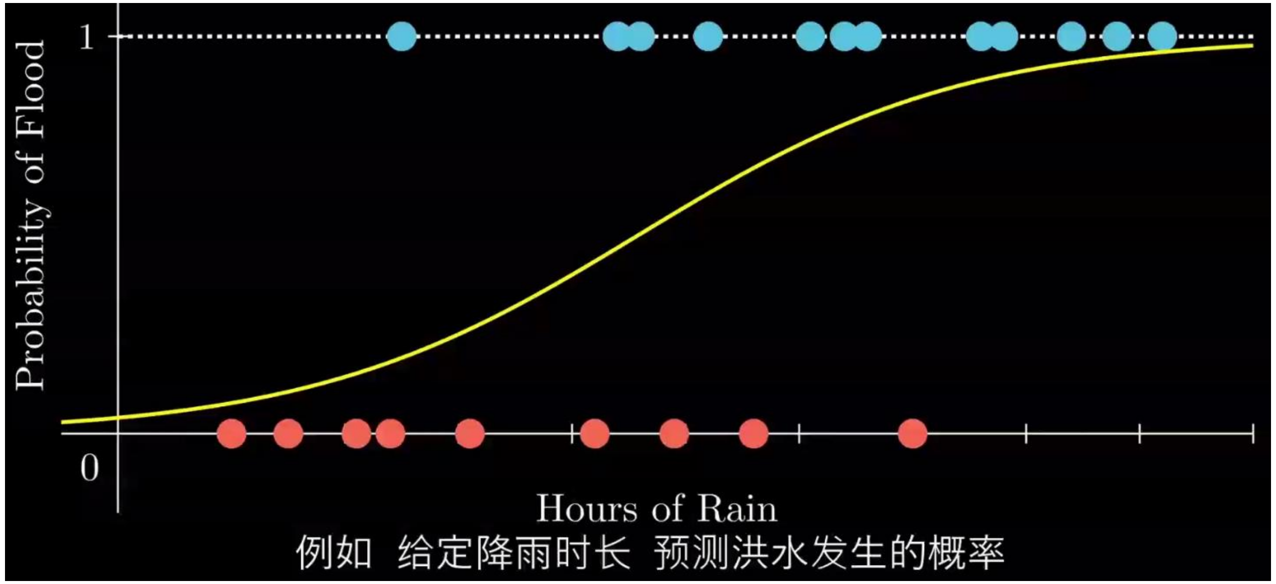
```
def cost(w, X, y):
    w = np.matrix(w)
    X = np.matrix(X)
    y = np.matrix(y)
    first = np.multiply(-y, np.log(sigmoid(X * w.T)))
    second = np.multiply((1 - y), np.log(1 - sigmoid(X * w.T)))
    return np.sum(first - second) / (len(X))
```

知识小结

1. Sigmoid函数
2. 逻辑回归推导——最大似然估计
3. 逻辑回归代码实现

作业和拓展

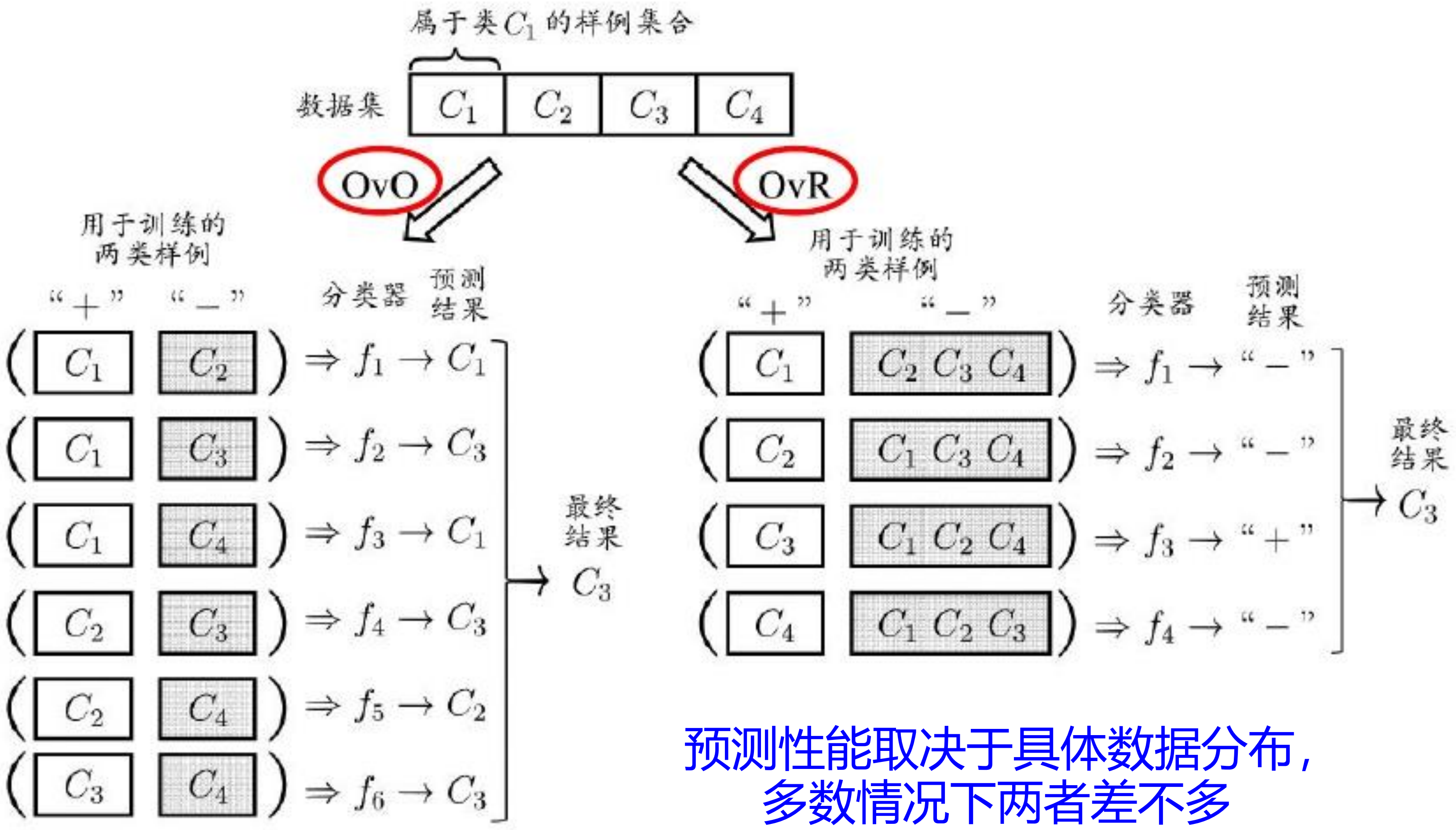
1. 课后练习：实现完整的逻辑回归代码，解决洪水预测问题。训练数据集见学习通APP。



2. 拓展思考：Sigmoid函数是如何构造出来的?

4.逻辑回归代码实现

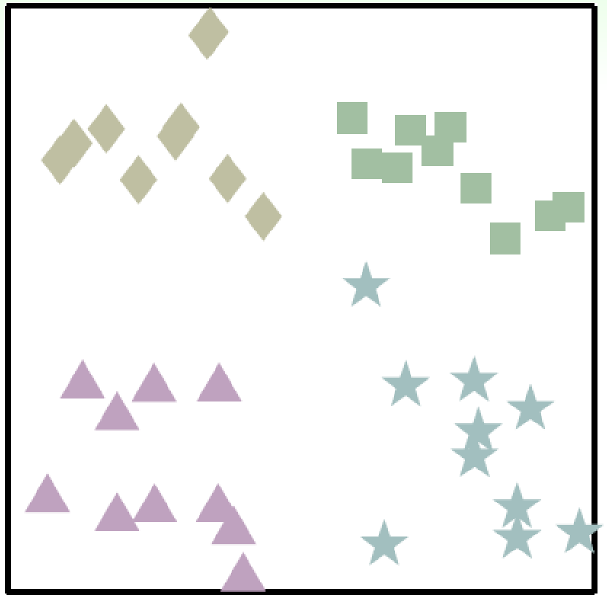
多分类学习



预测性能取决于具体数据分布，
多数情况下两者差不多

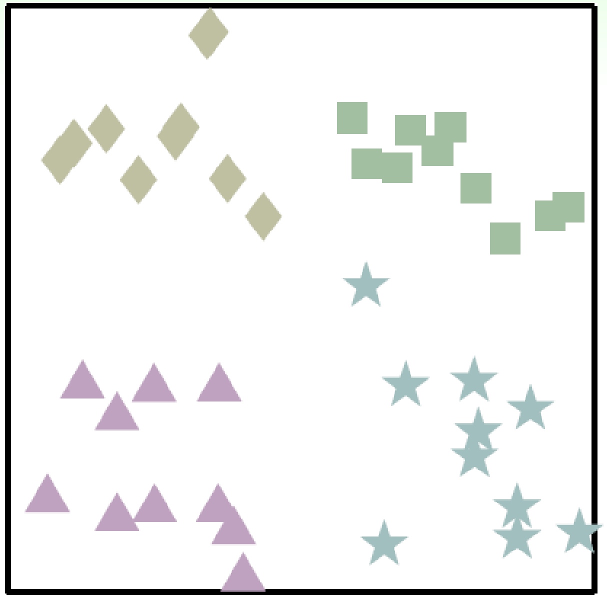
22

Multiclass Classification



- $\mathcal{Y} = \{\square, \Diamond, \triangle, \star\}$
  (4-class classification)
- **many applications** in practice, especially for 'recognition'

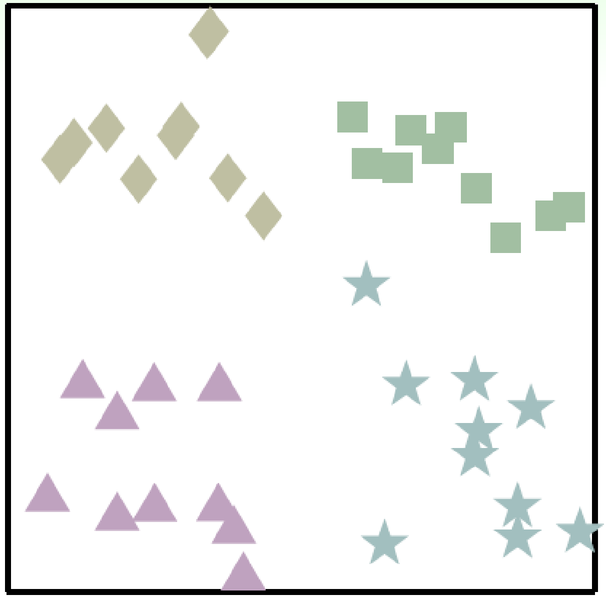
Multiclass Classification



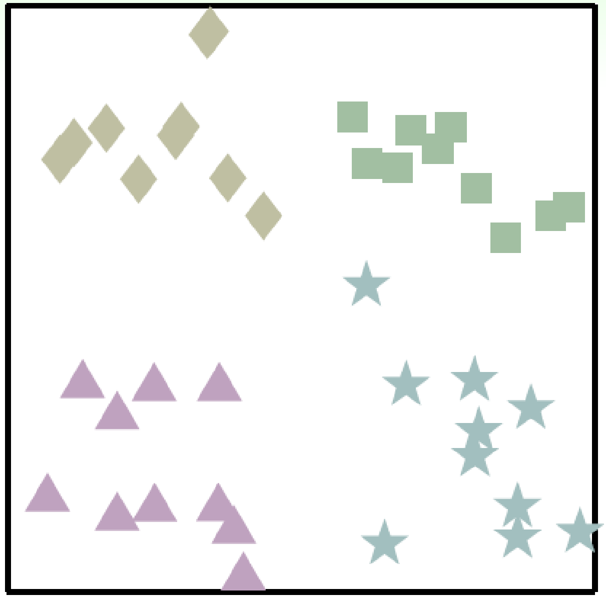
- $\mathcal{Y} = \{\square, \diamond, \triangle, \star\}$ (4-class classification)
- **many applications** in practice, especially for 'recognition'

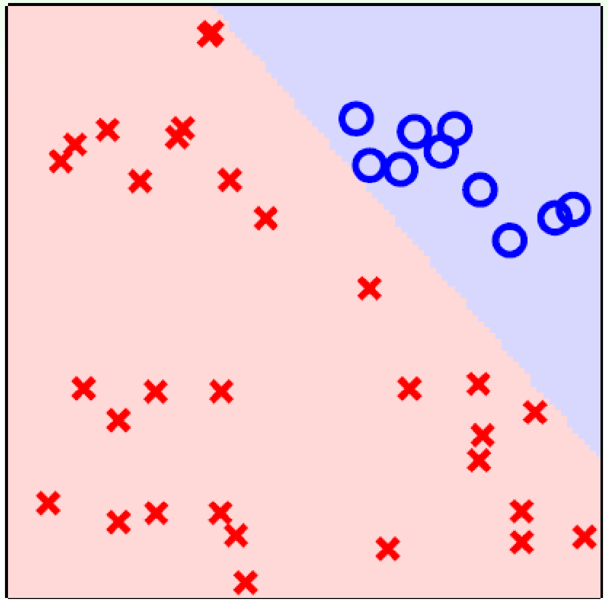
next: use **tools for $\{\times, \circ\}$ classification** to $\{\square, \diamond, \triangle, \star\}$ classification

One Class at a Time



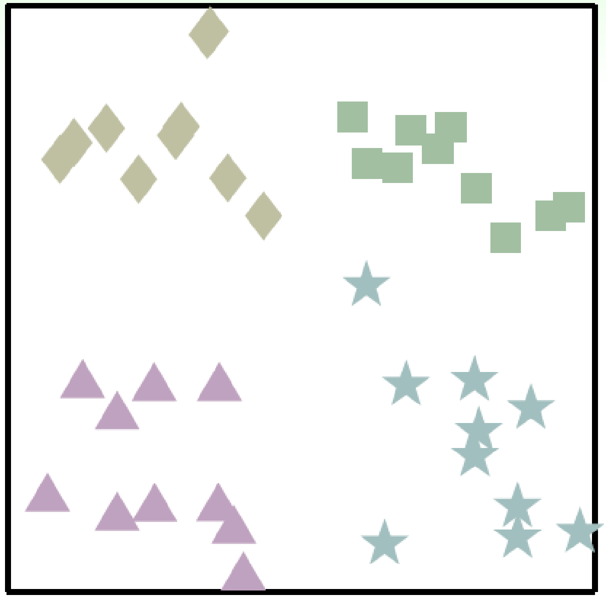
One Class at a Time

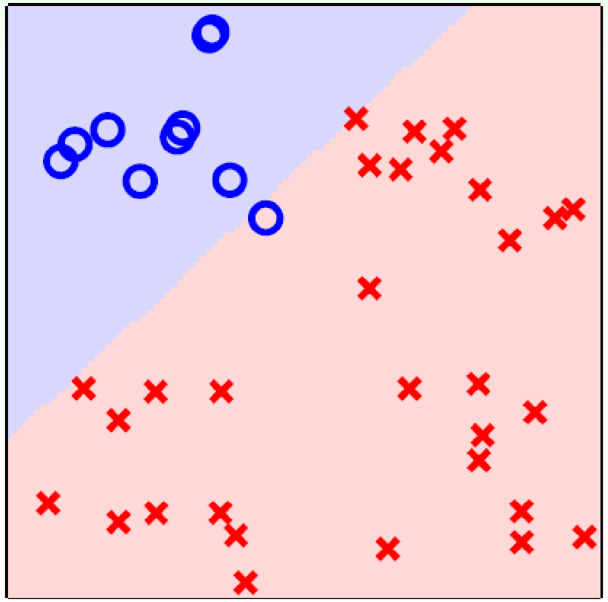




$\square$ or not? $\{\square = \circ, \Diamond = \times, \triangle = \times, \star = \times\}$

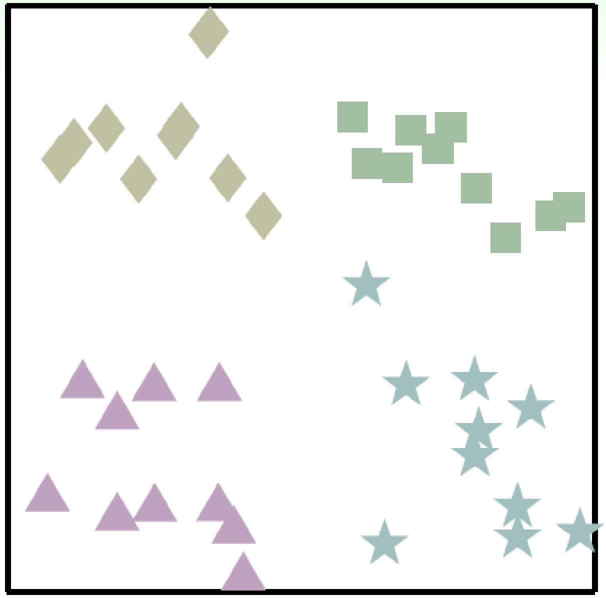
One Class at a Time

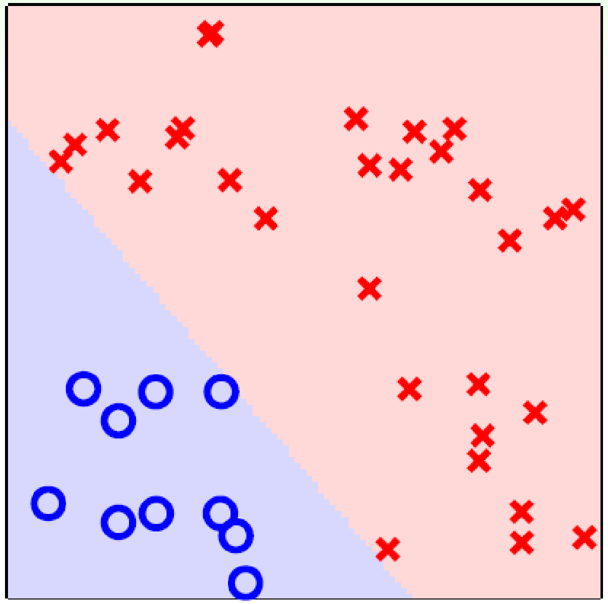




$\Diamond$ or not? $\{\Box = \times, \Diamond = \circ, \triangle = \times, \star = \times\}$

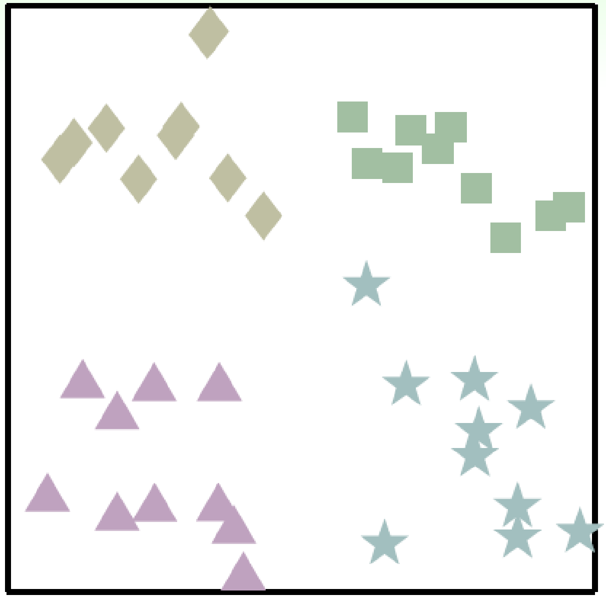
One Class at a Time

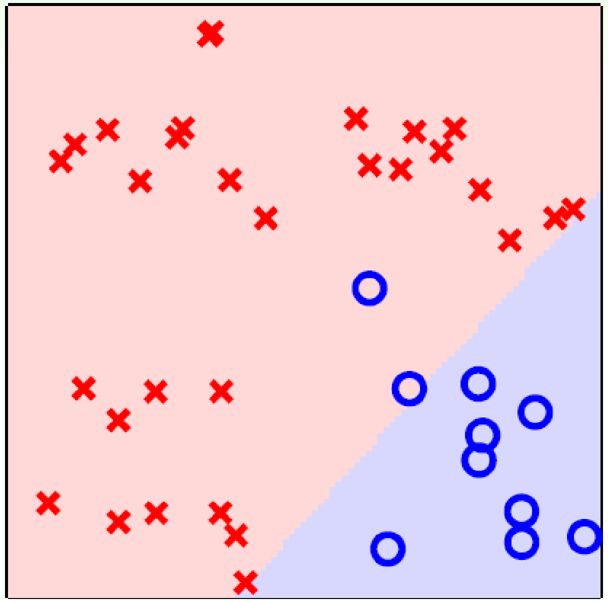




$\triangle$ or not? $\{\square = \times, \Diamond = \times, \triangle = \circ, \star = \times\}$

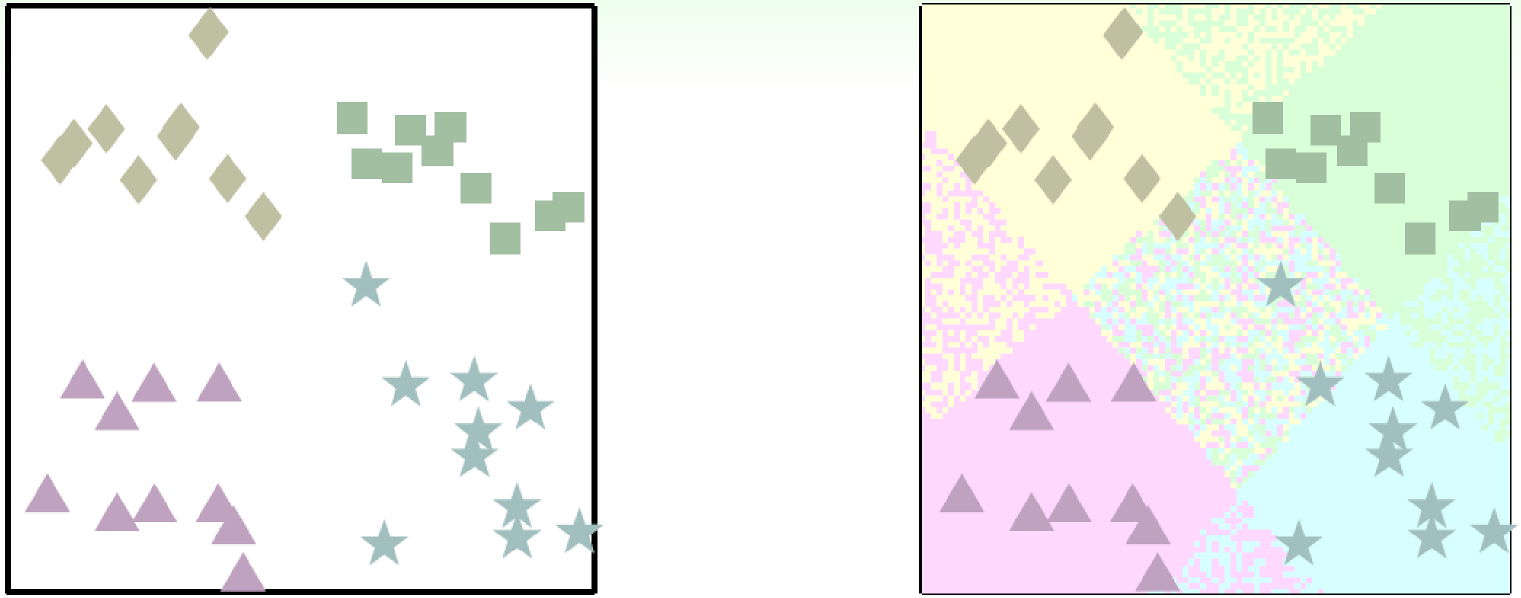
One Class at a Time

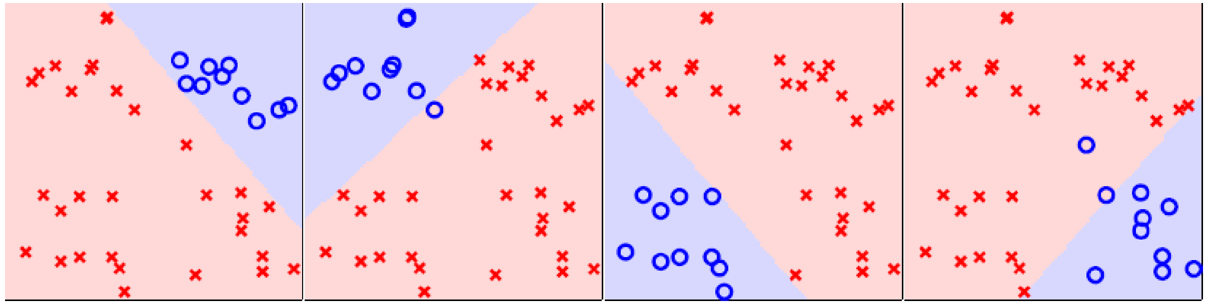




$\star$ or not? $\{\square = \times, \Diamond = \times, \triangle = \times, \star = \circ\}$

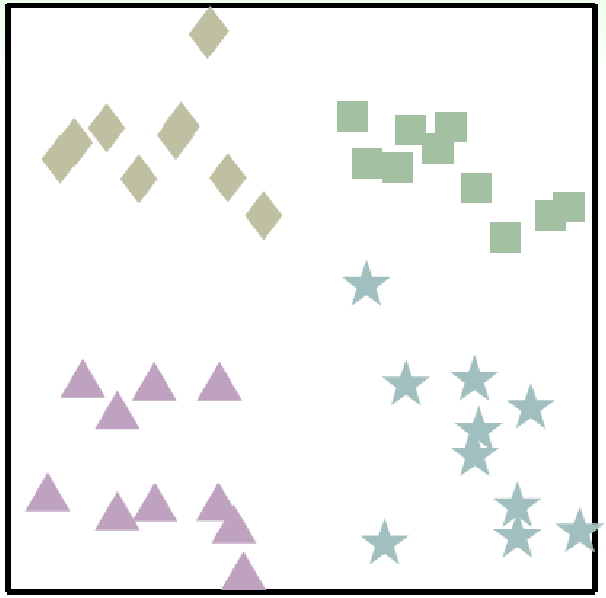
Multiclass Prediction: Combine Binary Classifiers

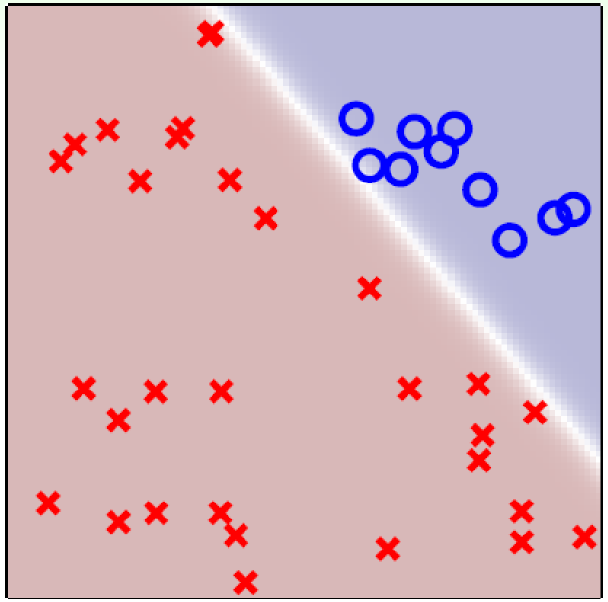


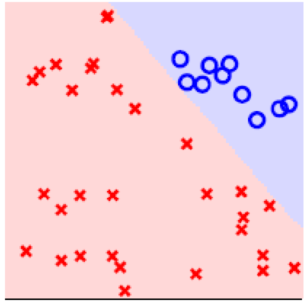


but **ties? :-)**

One Class at a Time **Softly**

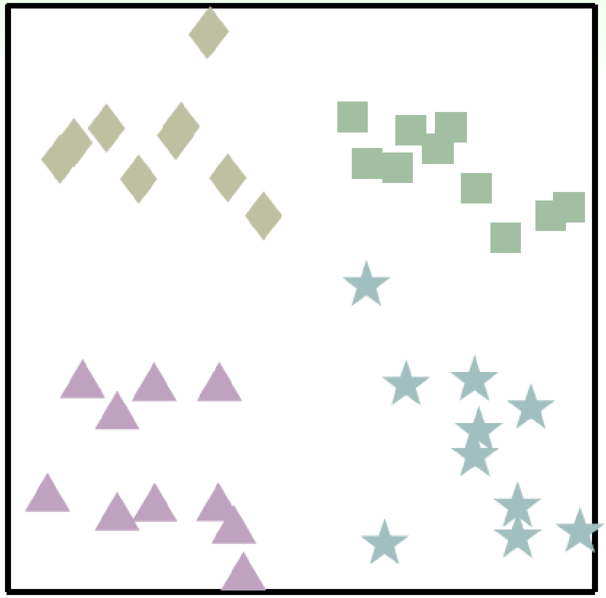


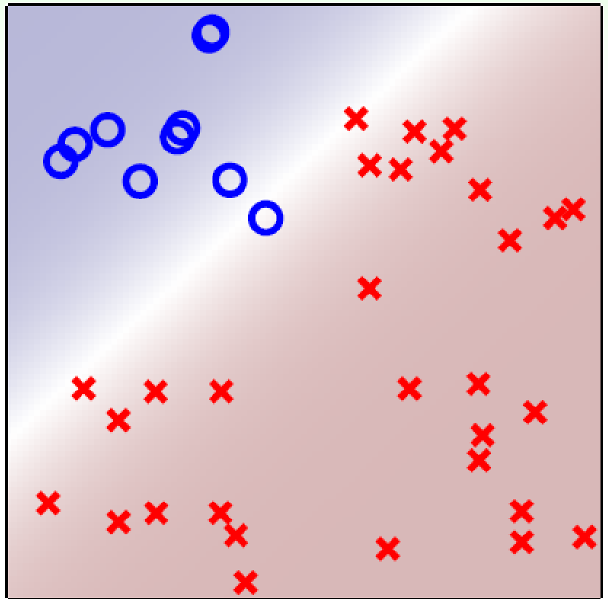


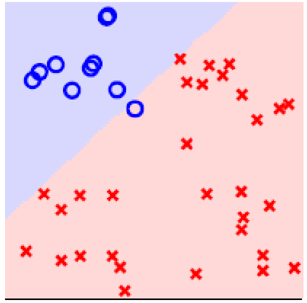


$$P(\square|\mathbf{x})?\ \{\square = \circ, \Diamond = \times, \triangle = \times, \star = \times\}$$

One Class at a Time **Softly**

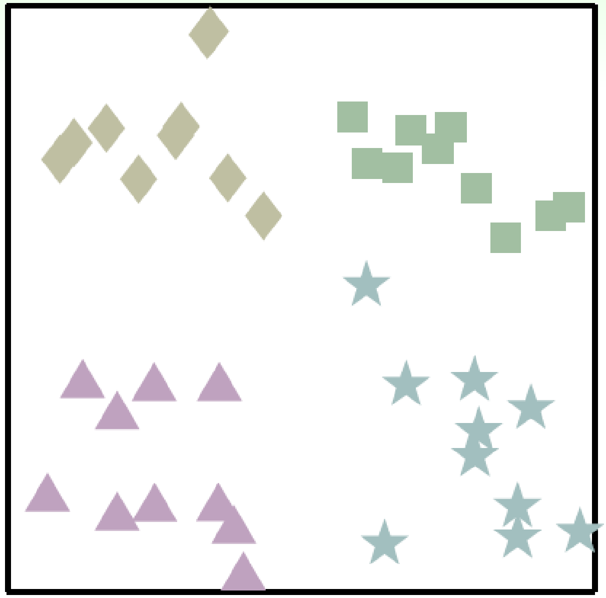


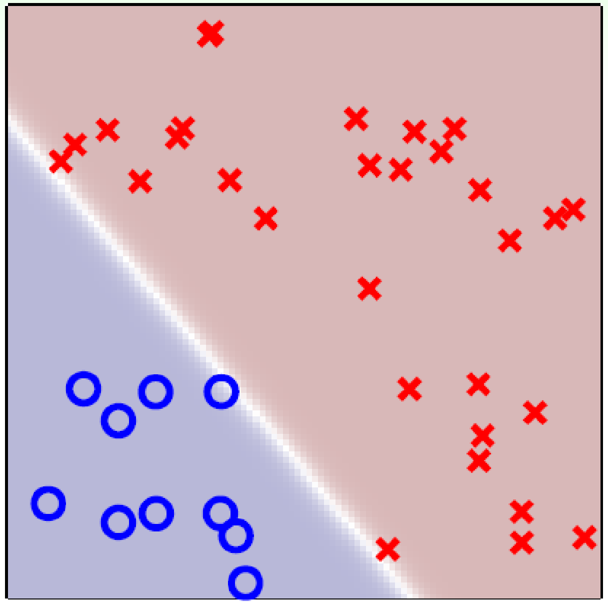


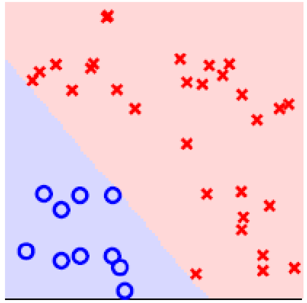


$P(\Diamond|\mathbf{x})?\ \{\square = \times, \Diamond = \circ, \triangle = \times, \star = \times\}$

One Class at a Time **Softly**

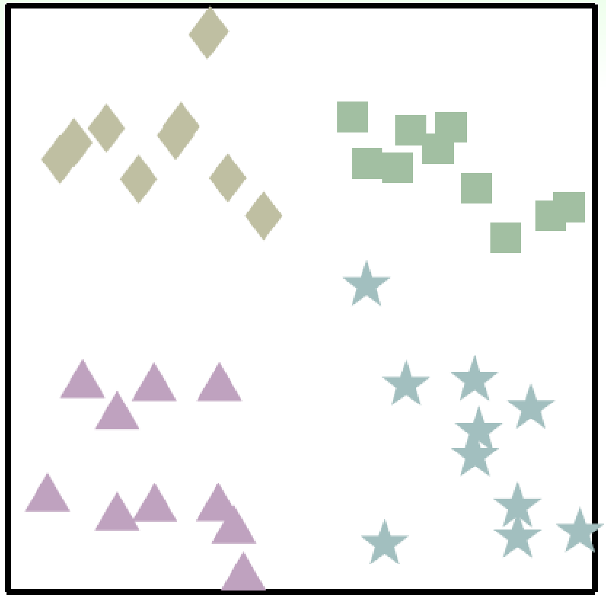


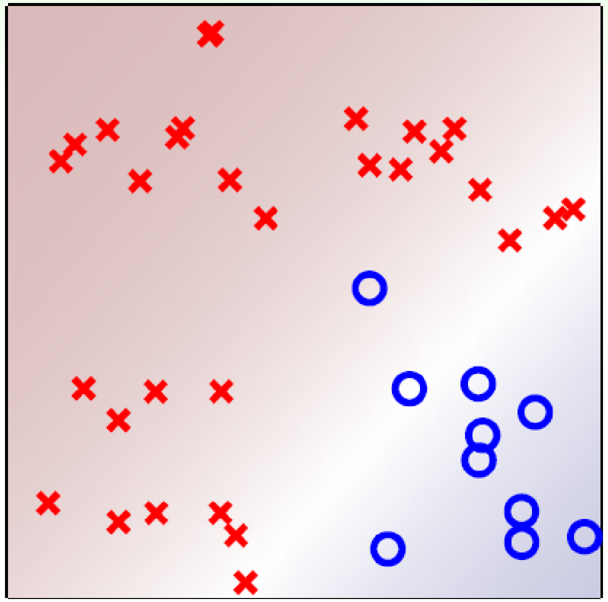


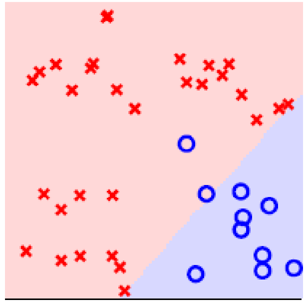


$$P(\triangle|\mathbf{x})?\ \{\square = \times, \lozenge = \times, \triangle = \circ, \star = \times\}$$

One Class at a Time **Softly**

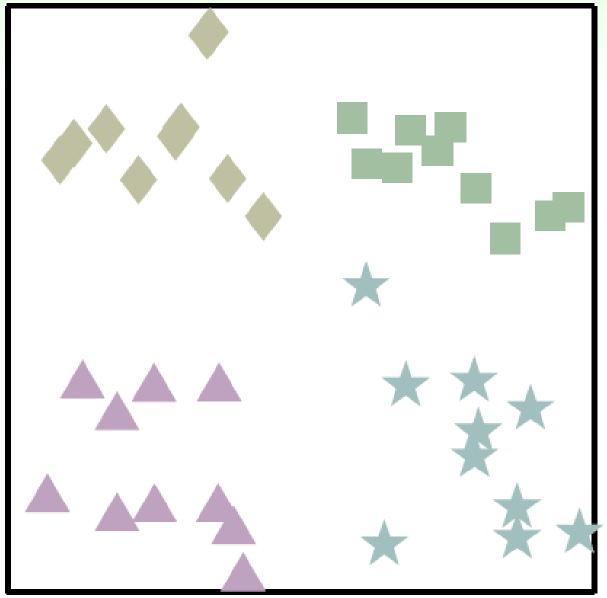


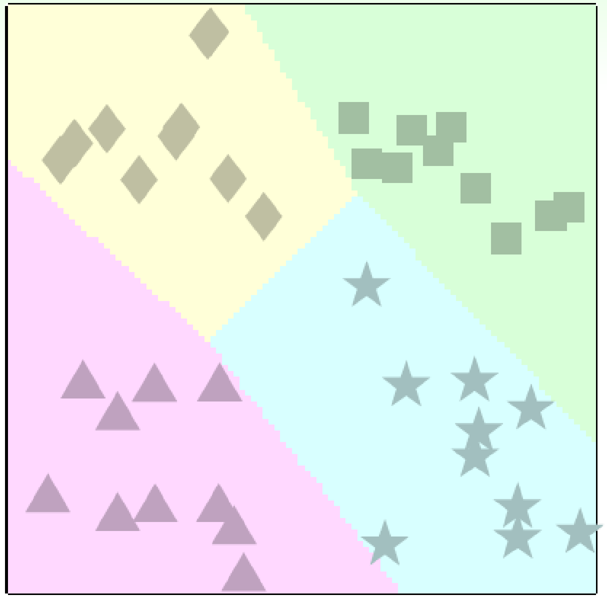


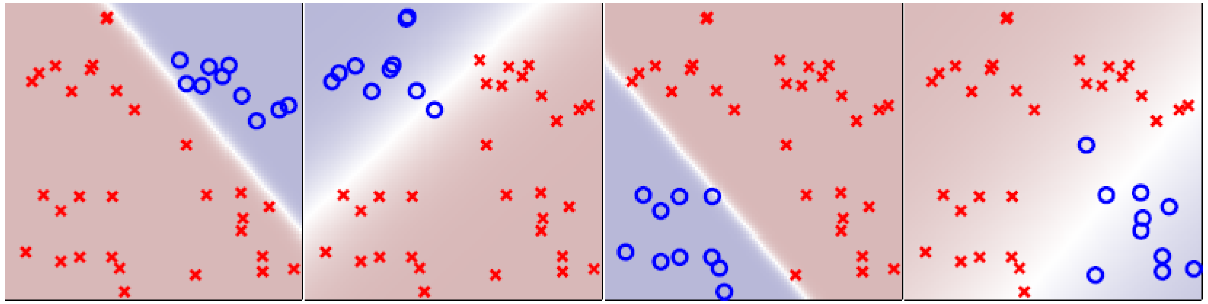


$P(\star|\mathbf{x})$? $\{\square = \times, \diamond = \times, \triangle = \times, \star = \circ\}$

Multiclass Prediction: Combine **Soft** Classifiers

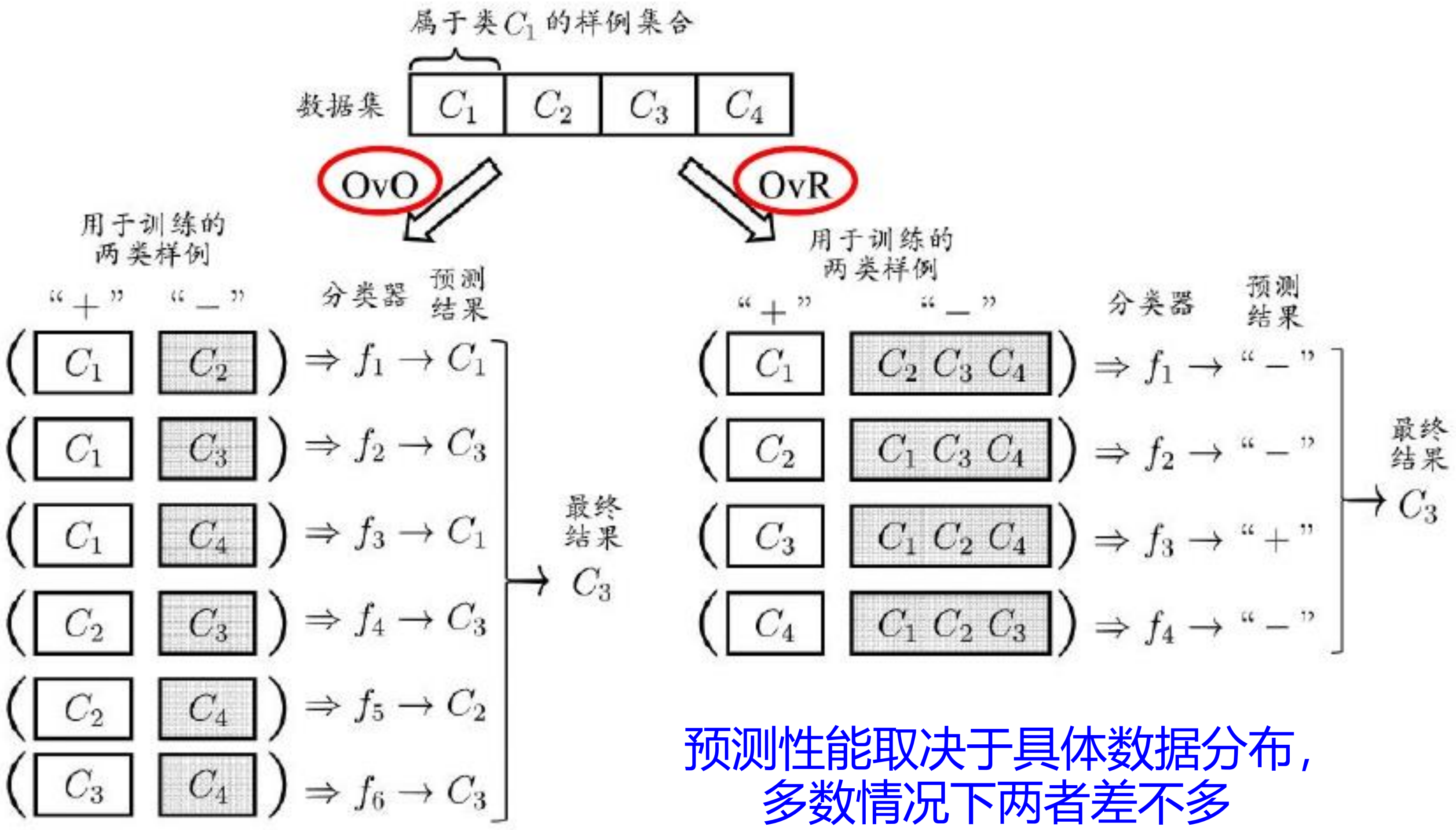






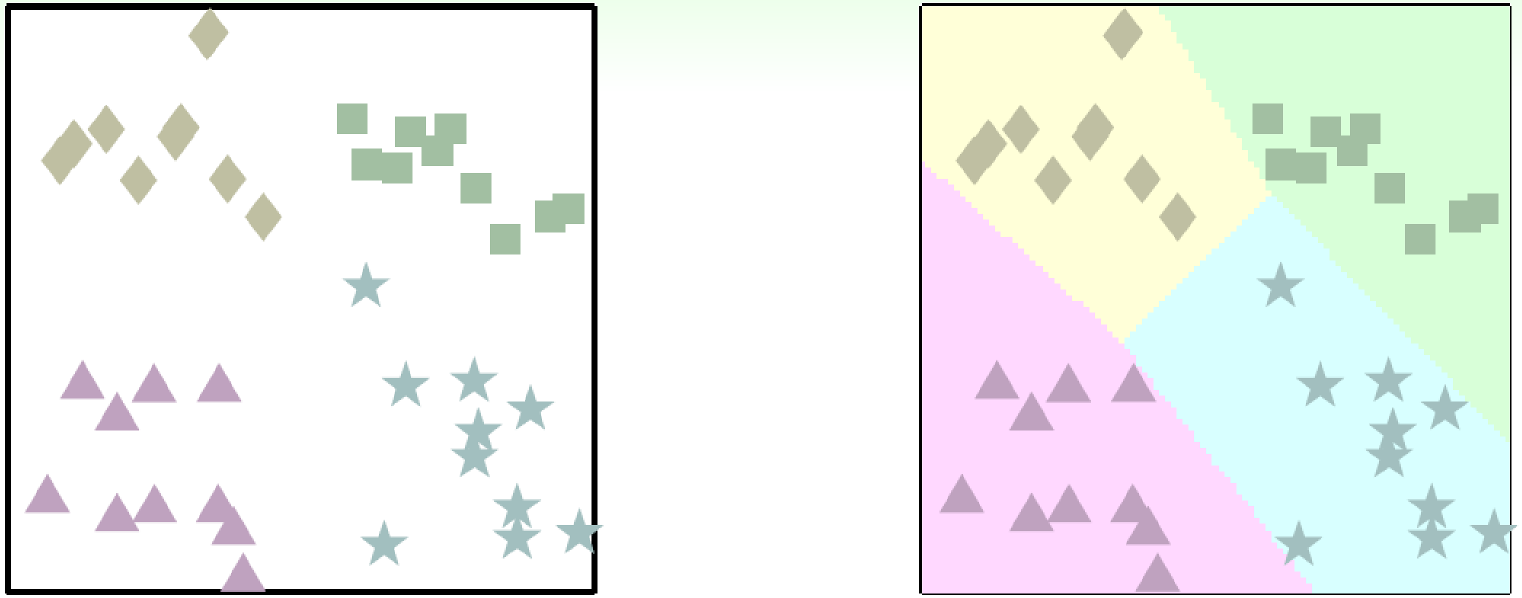
$$g(\mathbf{x}) = \text{argmax}_{k \in \mathcal{Y}}\, \theta\left(\mathbf{w}_{[k]}^T \mathbf{x}\right)$$

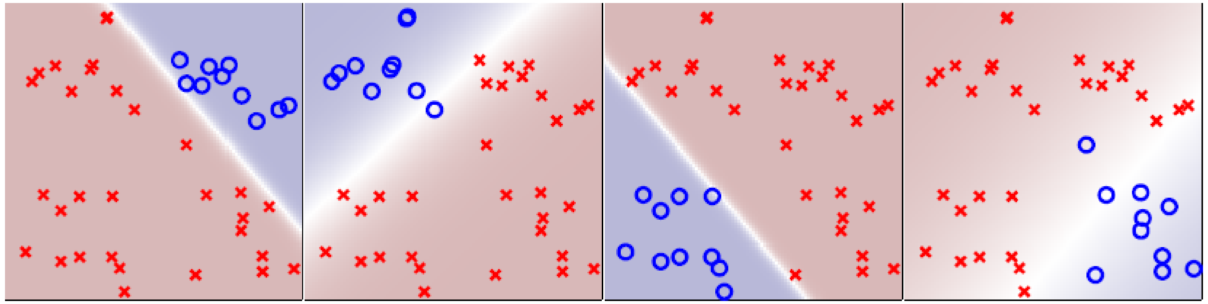
多分类学习



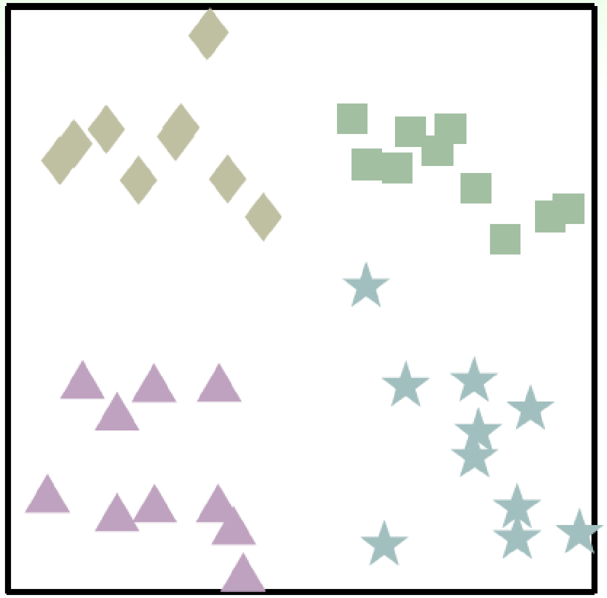
预测性能取决于具体数据分布，
多数情况下两者差不多

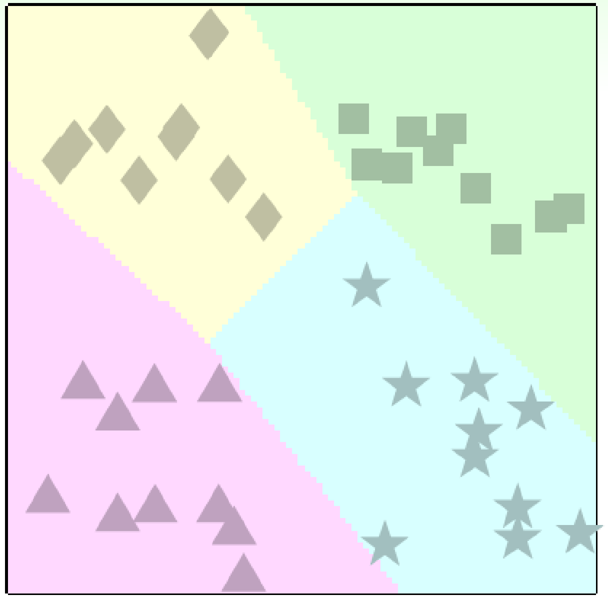
Source of **Unbalance**: One versus **All**

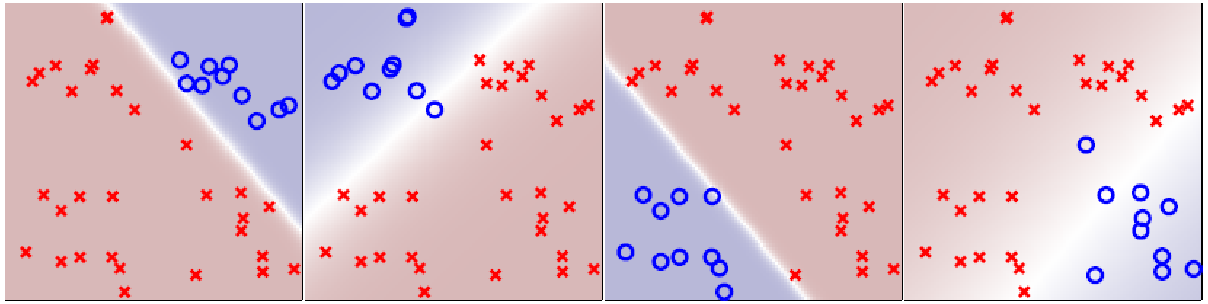




Source of **Unbalance**: One versus **All**

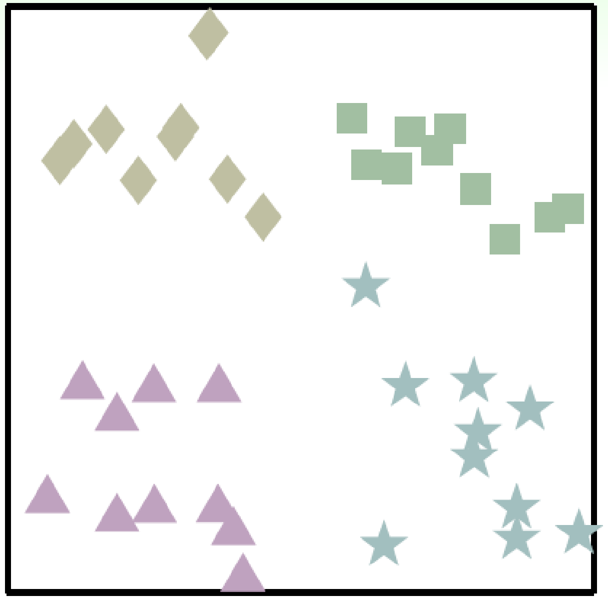


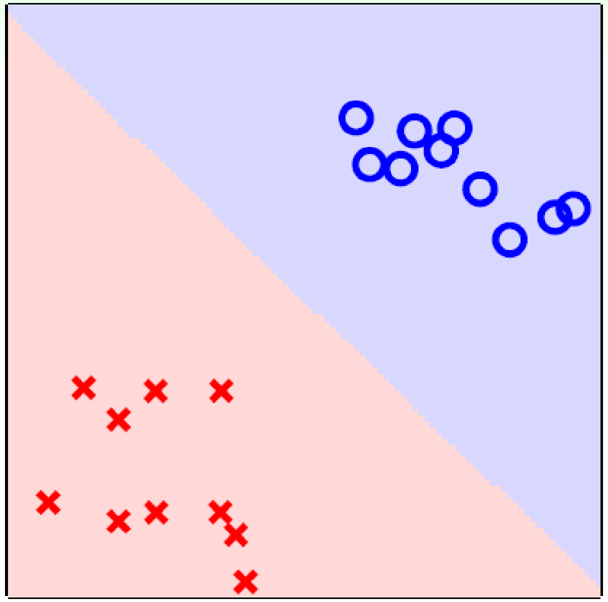




idea: make binary classification problems more **balanced** by one versus **one**

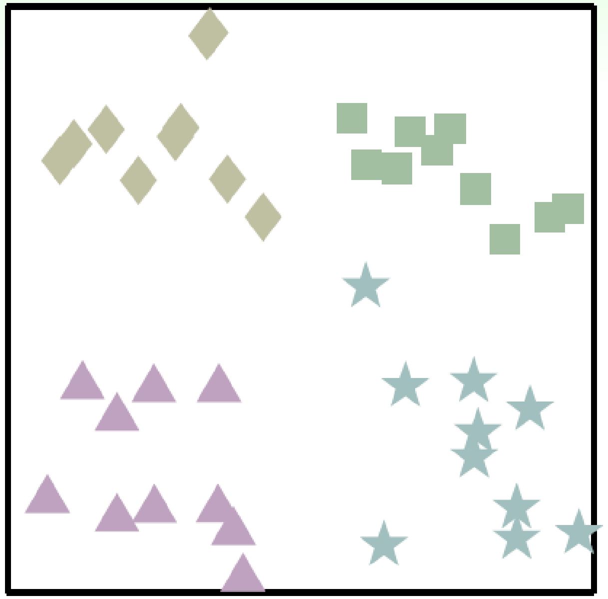
One versus One at a Time

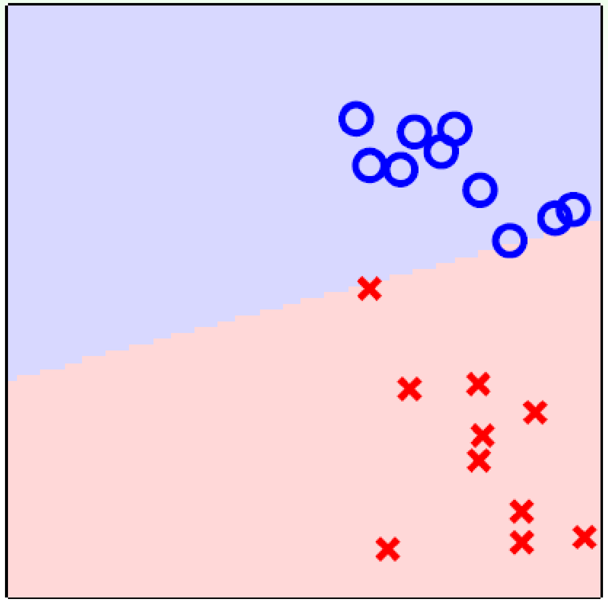




□ or △? {□ = ∘, ◇ = nil, △ = ×, ⋆ = nil}

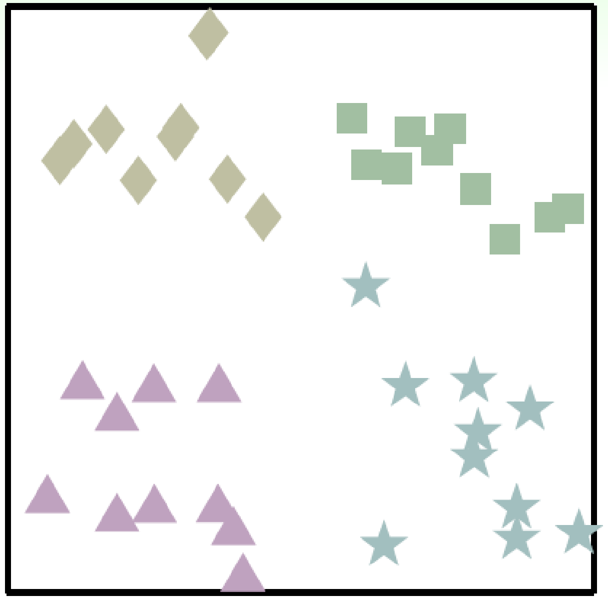
One versus One at a Time

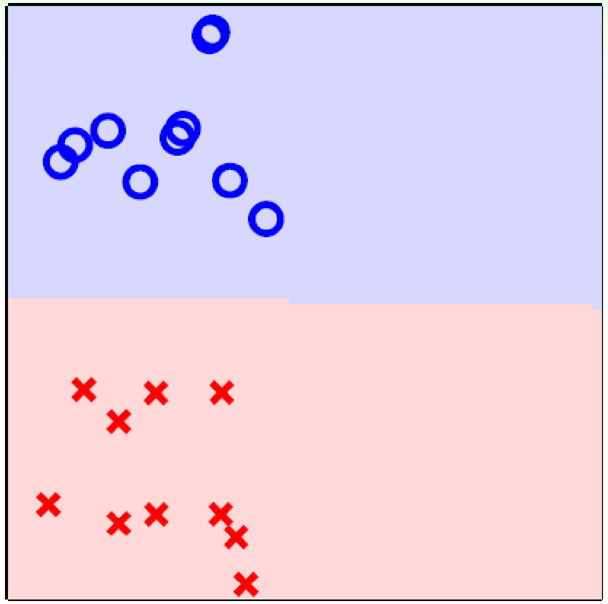




□ or ⋆? {□ = ○, ◊ = nil, △ = nil, ⋆ = ×}

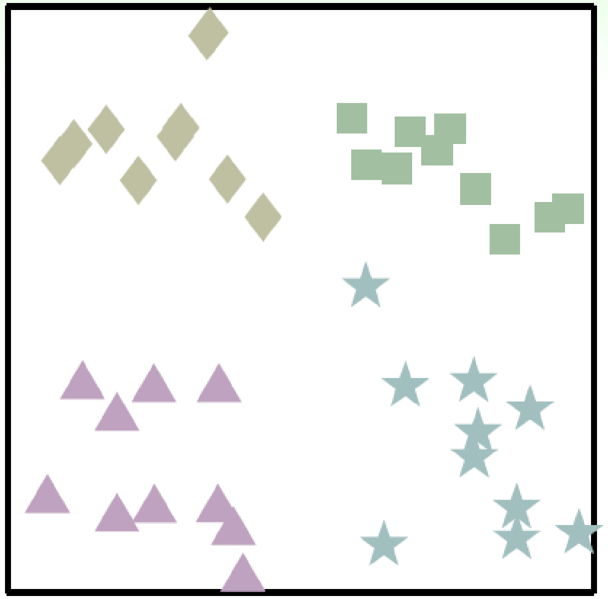
One versus One at a Time

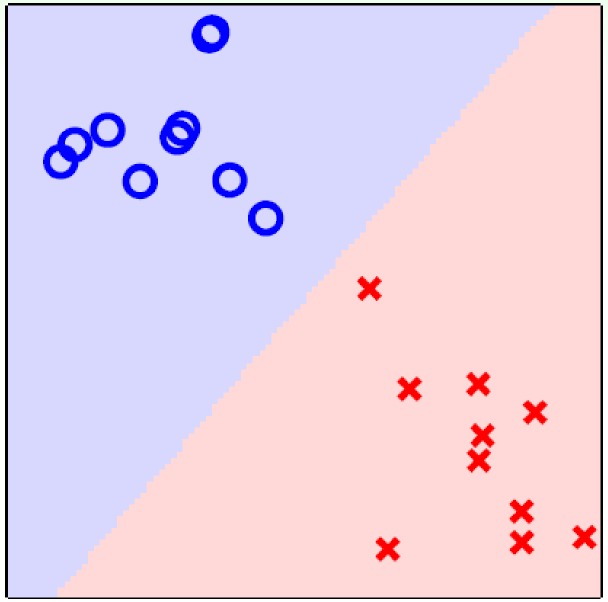




$\Diamond$ or $\triangle$? $\{\square = \text{nil}, \Diamond = \circ, \triangle = \times, \star = \text{nil}\}$

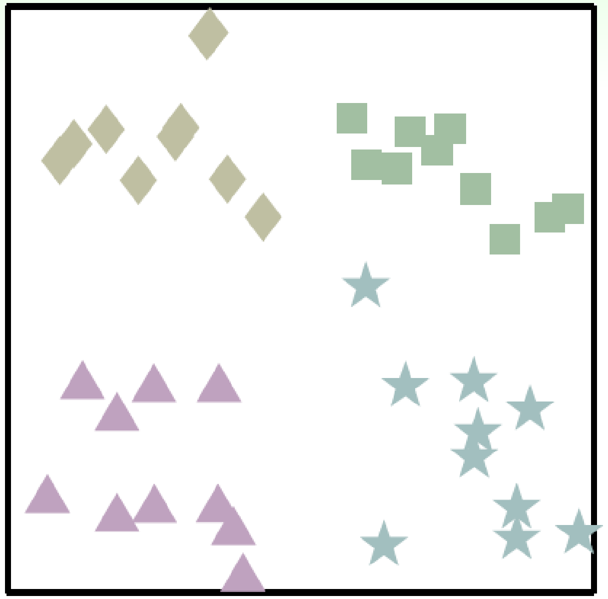
One versus One at a Time

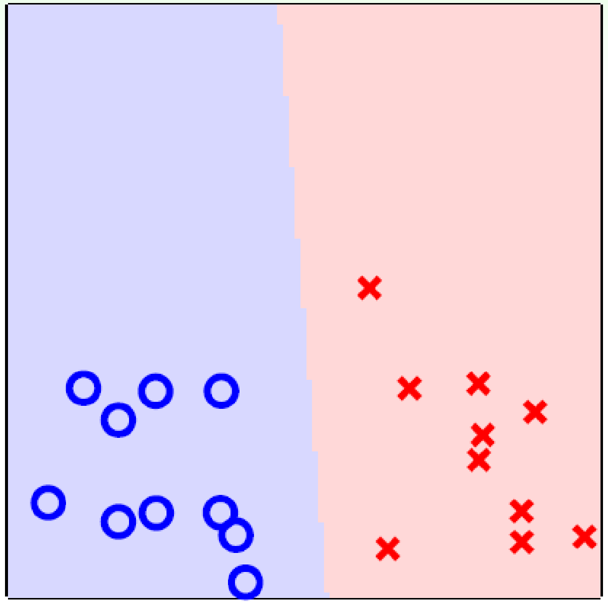




$\diamond$ or $\star$? $\{\square = \text{nil}, \diamond = \circ, \triangle = \text{nil}, \star = \times\}$

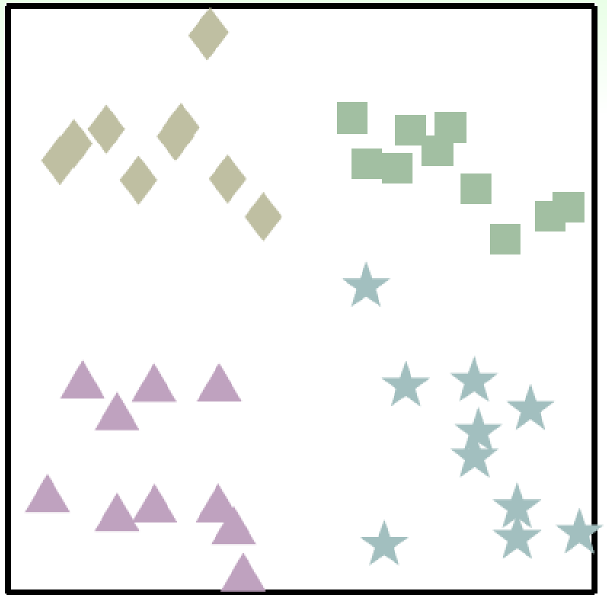
One versus One at a Time

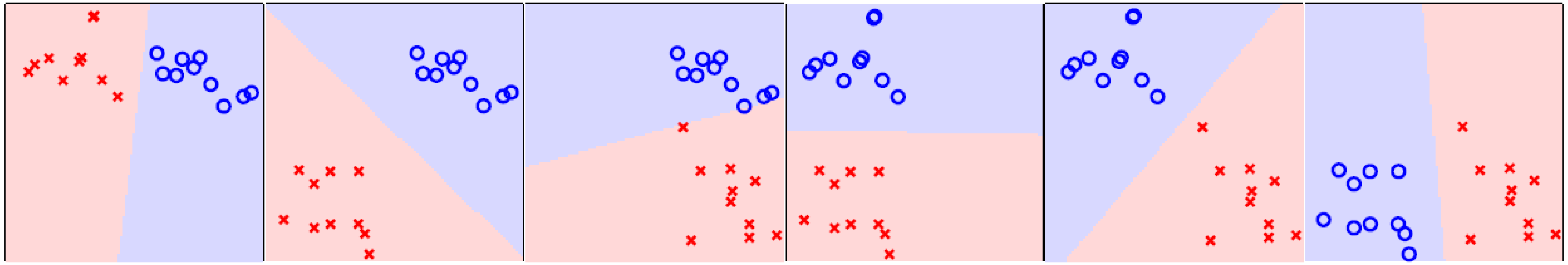




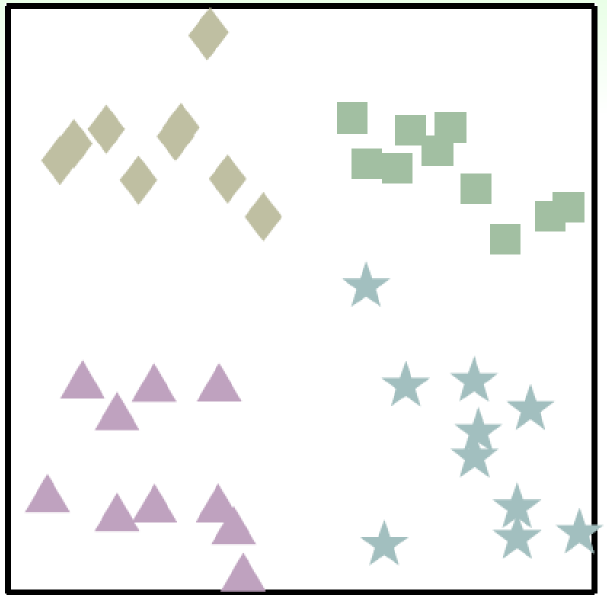
△ or ⋆? {□ = nil, ◊ = nil, △ = ∘, ⋆ = ×}

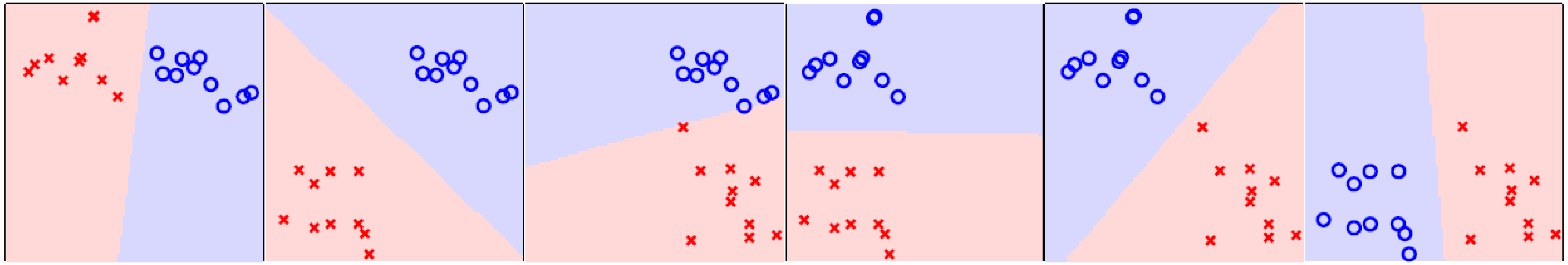
Multiclass Prediction: Combine **Pairwise** Classifiers





Multiclass Prediction: Combine **Pairwise** Classifiers

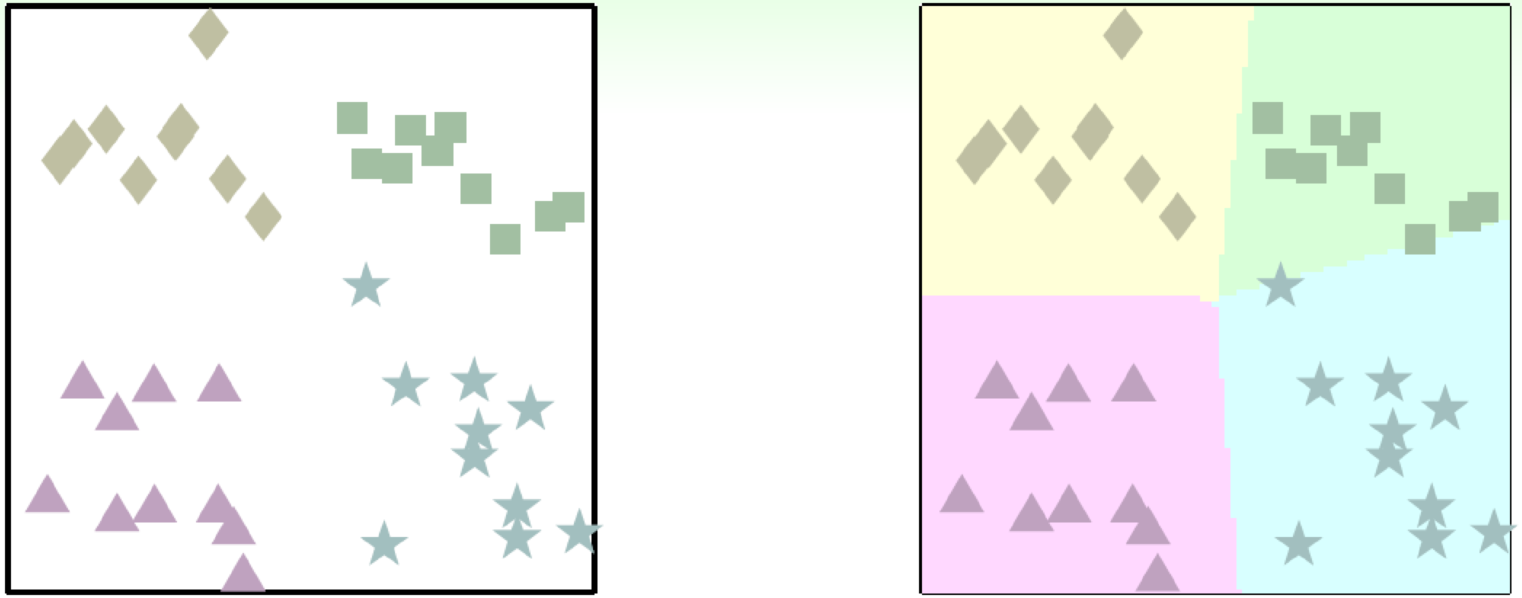


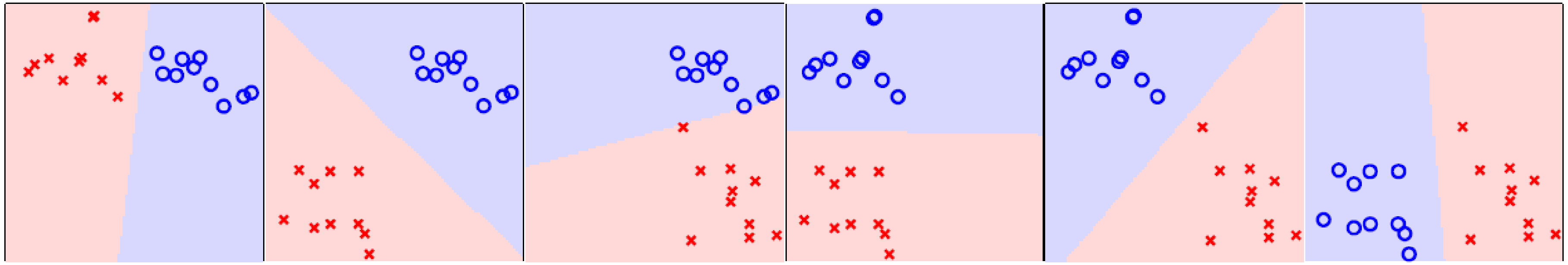


$$g(\mathbf{x}) = \text{tournament champion}\left\{\mathbf{w}_{[k,\ell]}^T\mathbf{x}\right\}$$

(voting of classifiers)

Multiclass Prediction: Combine **Pairwise** Classifiers



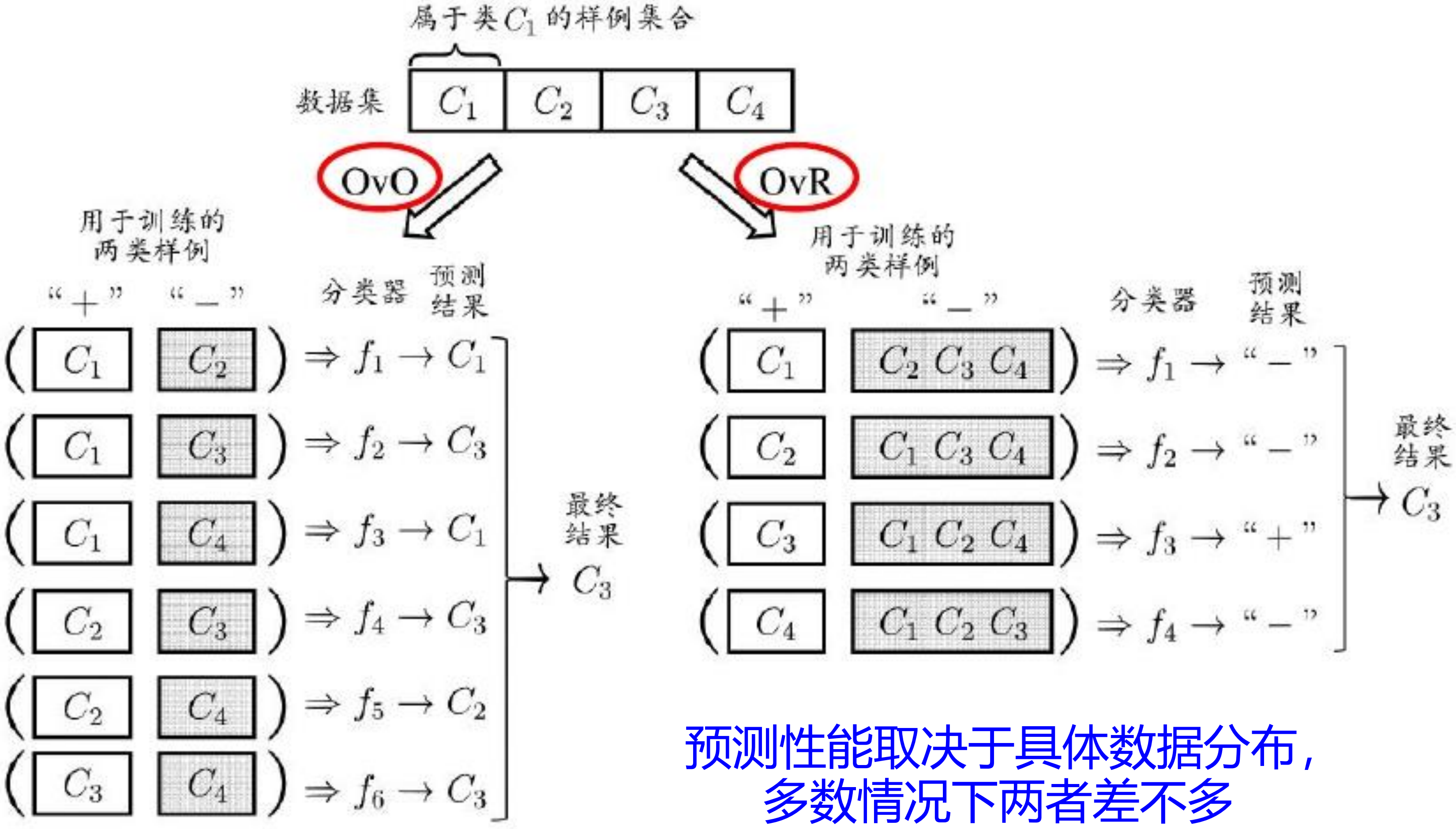


$g(\mathbf{x}) = \text{tournament champion} \left\{ \mathbf{w}_{[k,\ell]}^T \mathbf{x} \right\}$

(voting of classifiers)

多分类学习

- 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



预测性能取决于具体数据分布，
多数情况下两者差不多

- 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

参考文献

1. Prof. Andrew Ng. Machine Learning. Stanford University
2. 《统计学习方法》，清华大学出版社，李航著，2019年出版
3. 《机器学习》，清华大学出版社，周志华著，2016年出版
4. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag, 2006
5. Hsuan-Tien Lin（林轩田), Machine Learning Foundations, https://www.csie.ntu.edu.tw/~htlin/mooc/

谢 谢！