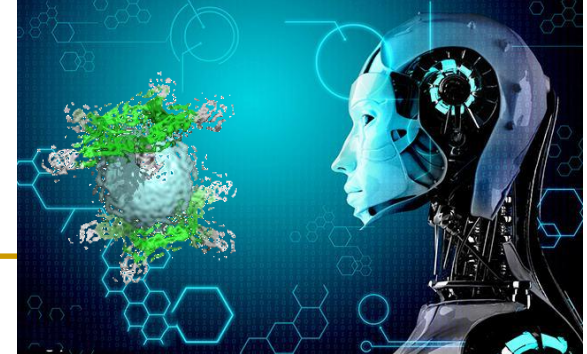


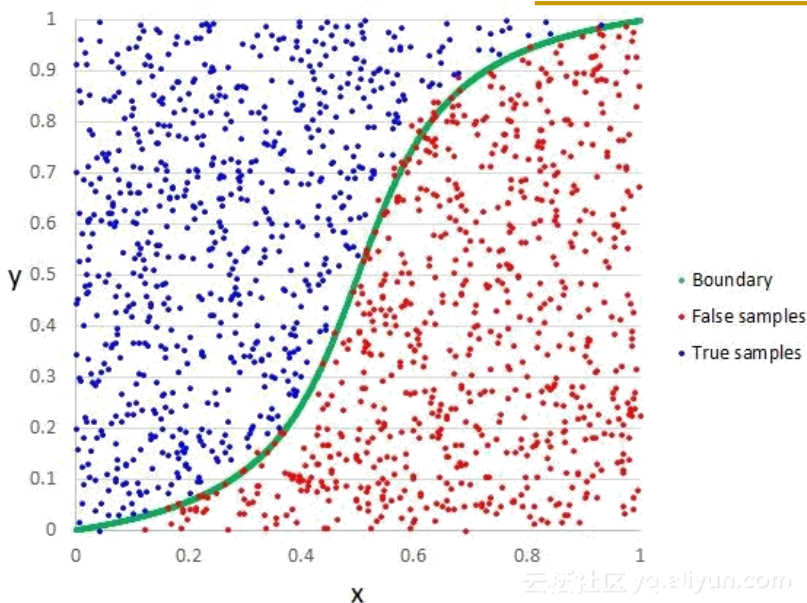
《机器学习及其在化学中的应用》2025年课程



Sec. 3

分类的线性方法 (逻辑回归)

Logistic Regression Example



刘志荣 (LiuZhiRong@pku.edu.cn)

北京大学化学学院

2025.9.29

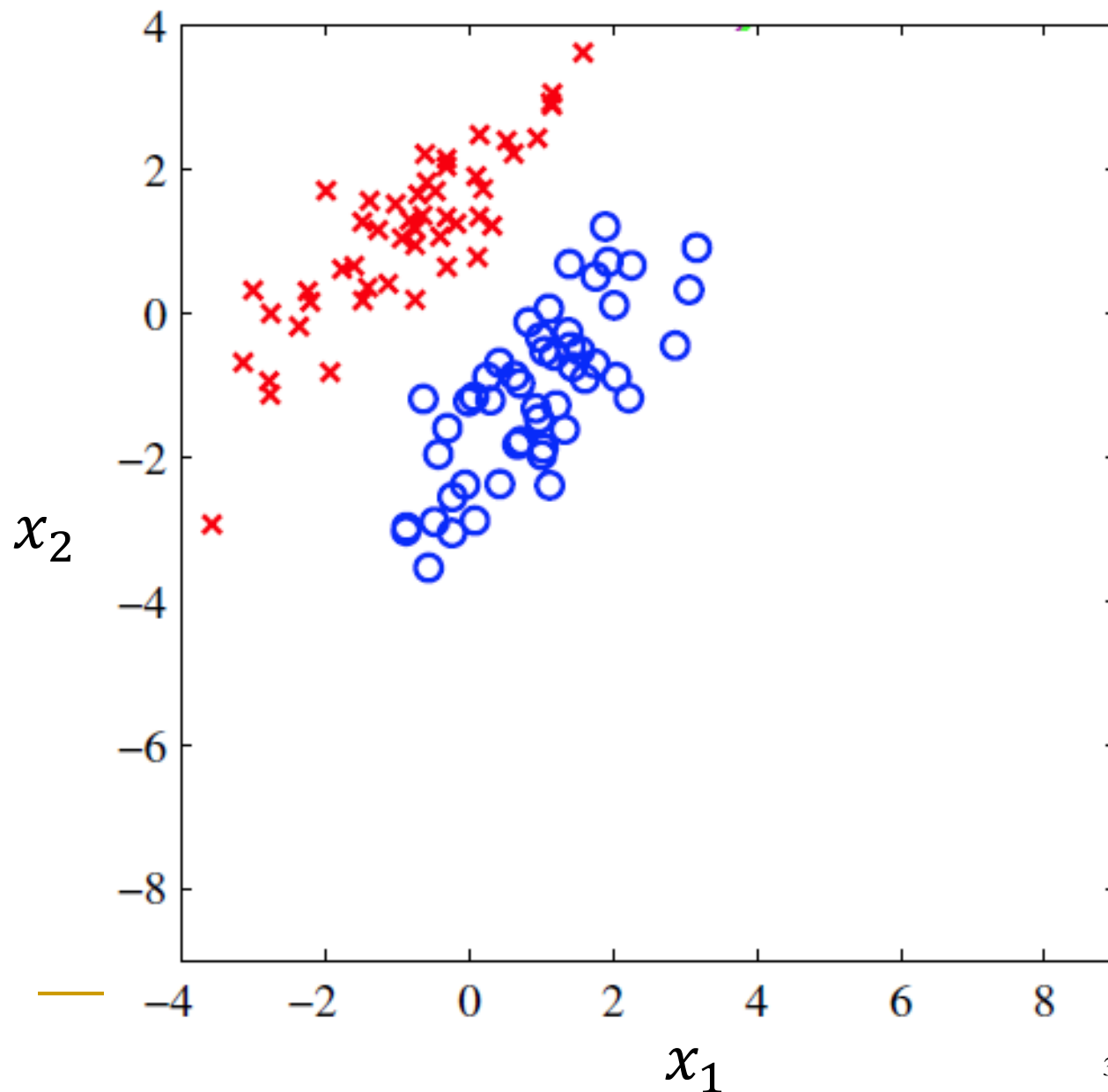
内容提要

- 简单的思路：线性拟合
- 感知器
- 逻辑回归
 - 激活函数；
 - 误差函数；
 - 梯度下降法；
 - 基的使用；正则化；
 - 多分类。
- 应用例子：点击率、金属价态

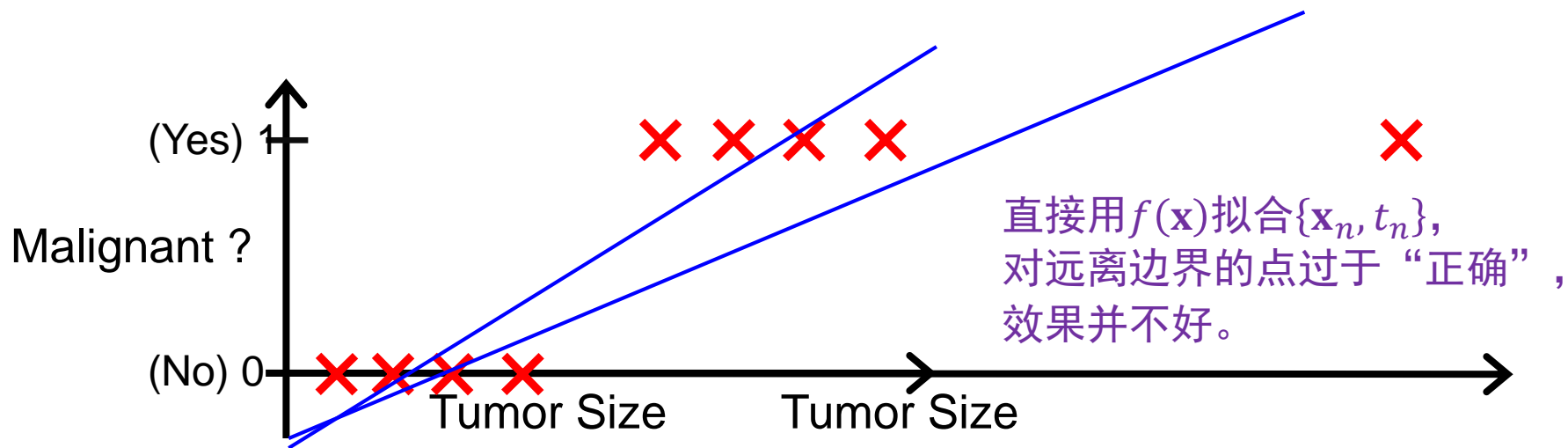
分类问题

- 预测数据（输入） \mathbf{x} 所属的类别（class）（输出）：

$$t = \begin{cases} 1, & \text{属于第一类} \\ 0, & \text{属于第二类} \end{cases}$$



一个简单的思路：线性拟合

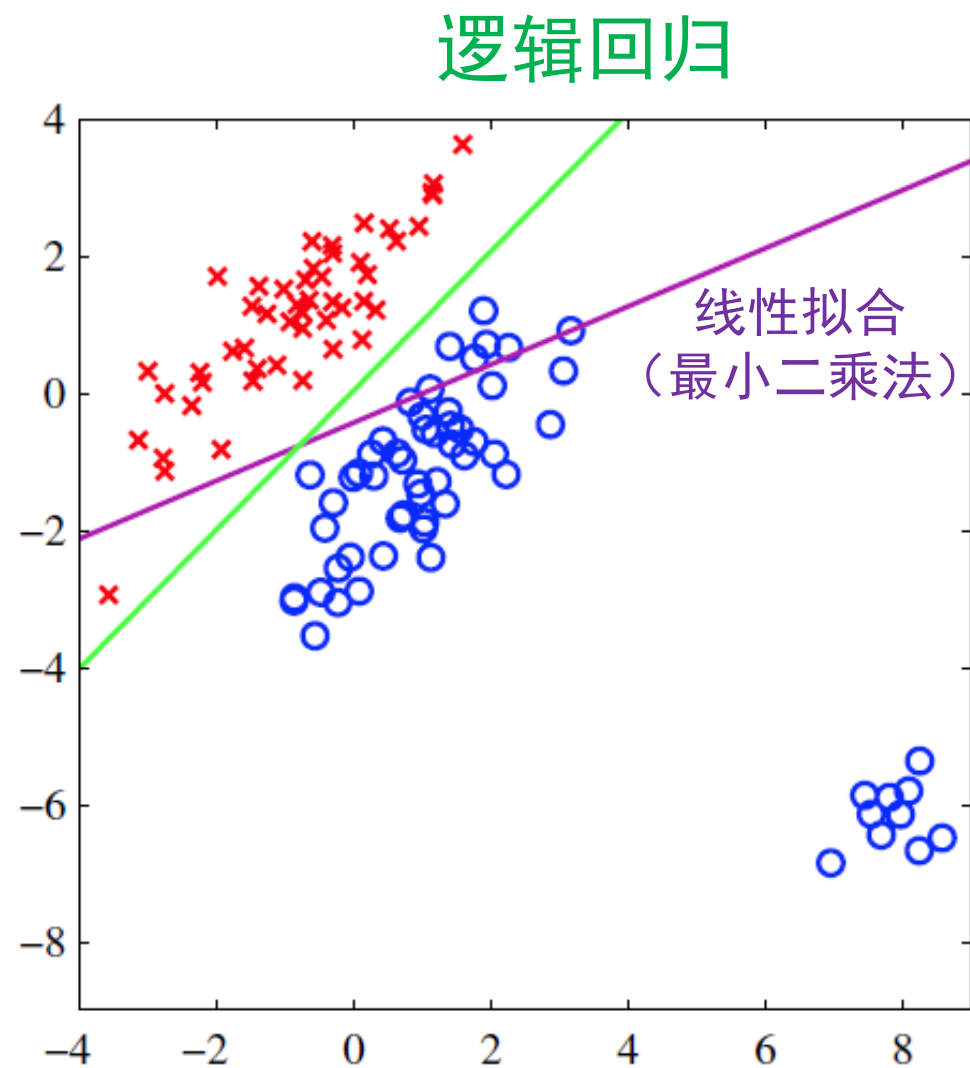
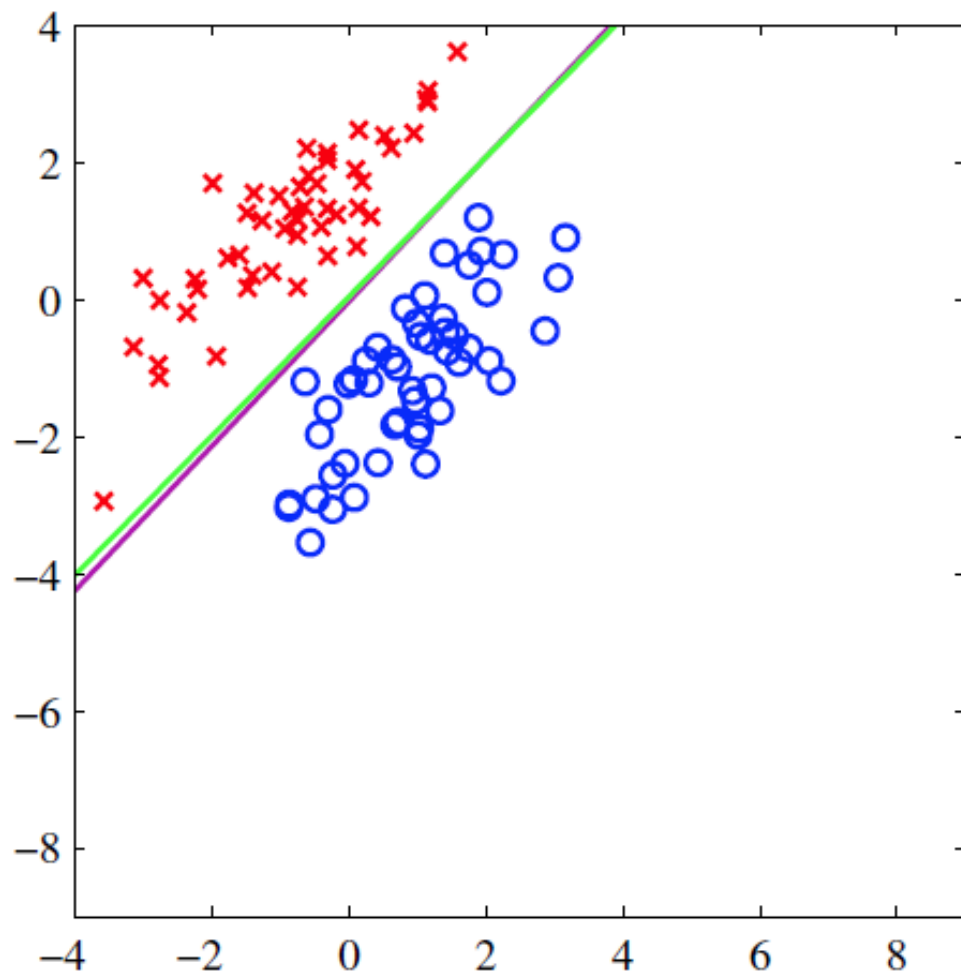


$$f(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_D x_D = \mathbf{w}^T \mathbf{x}$$

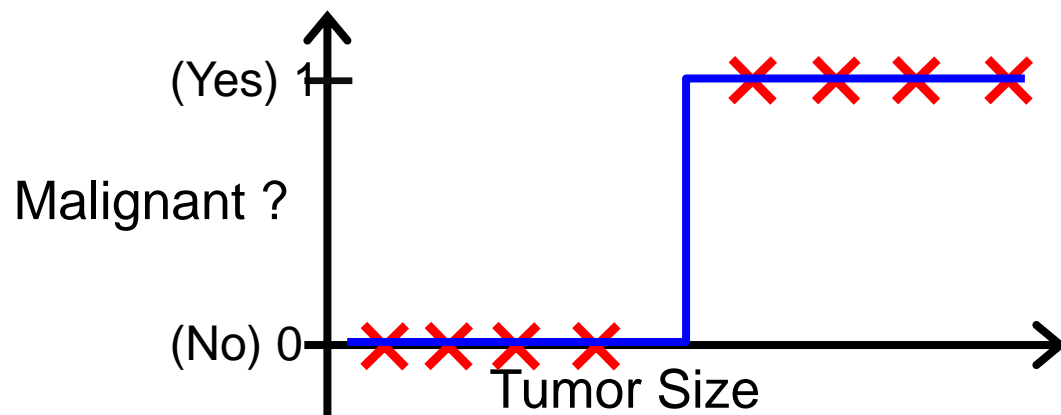
(其中定义 $x_0 \equiv 1$)

$$\text{预测值} = \begin{cases} 0 & (\text{if } f(\mathbf{x}, \mathbf{w}) \leq 0.5) \\ 1 & (\text{if } f(\mathbf{x}, \mathbf{w}) > 0.5) \end{cases}$$

二维例子



感知器 (perceptron)



■ Frank Rosenblatt (1962)

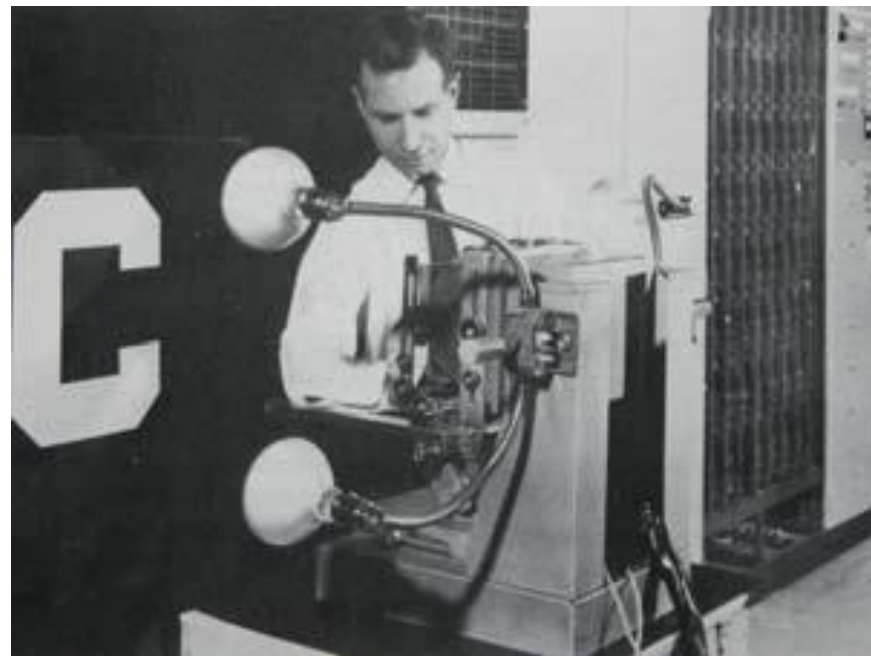
$$y(\mathbf{x}) = f(\mathbf{w}^T \mathbf{x})$$

其中

$$f(a) = \begin{cases} 0 & (\text{if } a \leq 0) \\ 1 & (\text{if } a > 0) \end{cases}$$

用 $y(\mathbf{x})$ 拟合 $\{\mathbf{x}_n, t_n\}$

缺点：阶跃函数，不容易计算导数及进行优化求解



逻辑回归

这是一种非常强大，甚至可能是世界上使用最广泛的一种分类算法。

——吴恩达

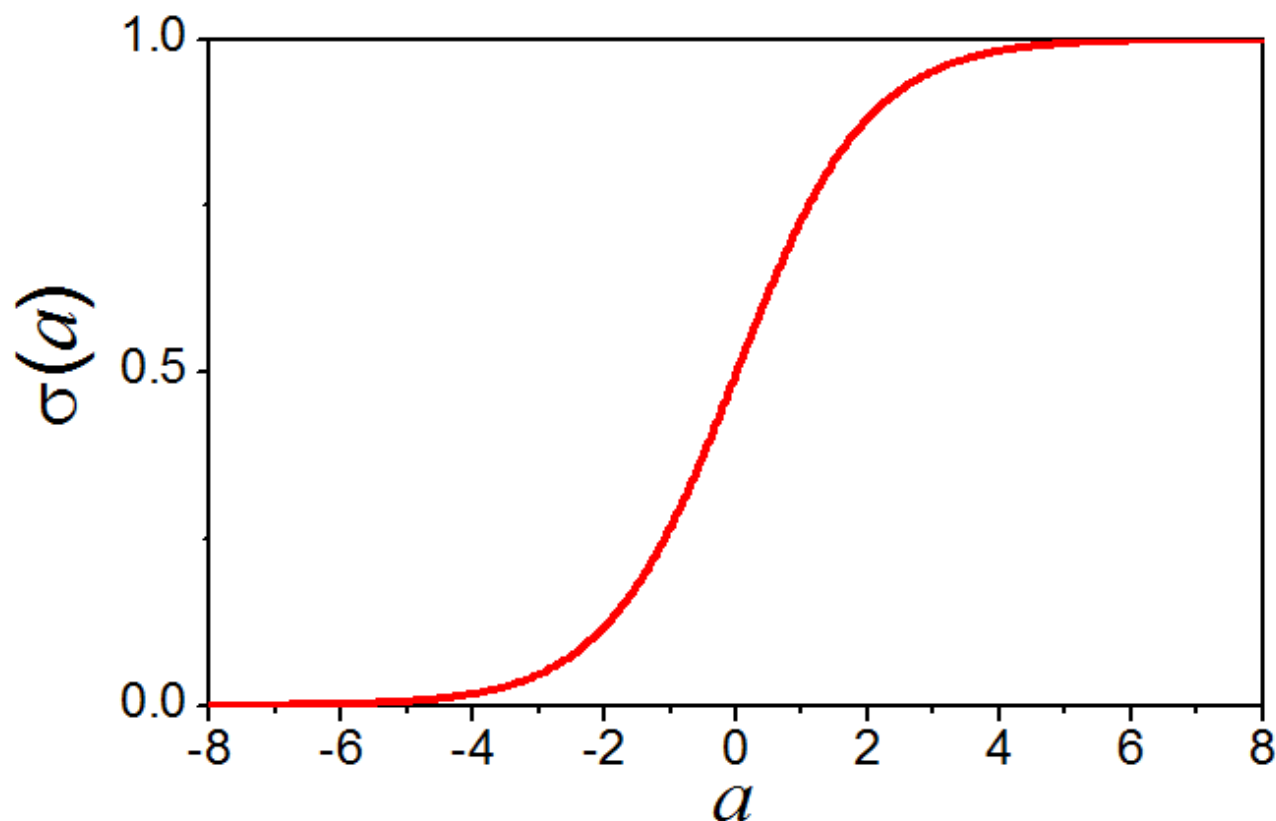
$$y(\mathbf{x}) = f(\mathbf{w}^T \mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x})$$

其中激活函数（activation function）用下式表示：

$$\sigma(a) = \frac{1}{1 + e^{-a}}$$

被称为S型（sigmoid）函数
或逻辑（logistic）函数。

- 进一步把 $\sigma(\mathbf{w}^T \mathbf{x})$ 解读为
数据点 \mathbf{x} 属于第一类的概率。



题外：逻辑函数/模型的前世今生...

- Verhulst—Pearl (1838)

- 简单的人口增长模型：

$$\frac{dy(t)}{dt} = ay(t)$$

解为

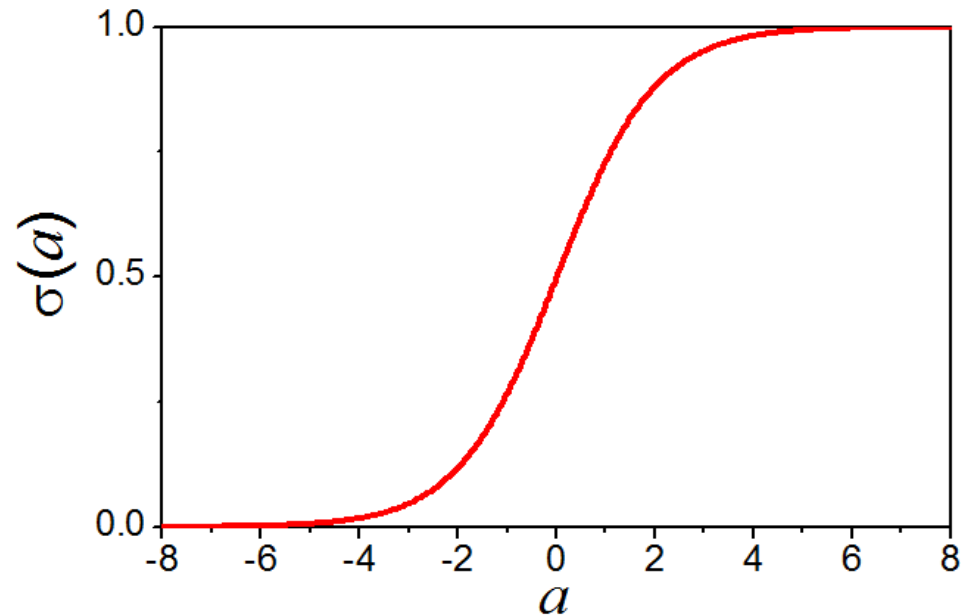
$$y(t) = y_0 e^{at}$$

- 资源有限时，

$$\frac{dy(t)}{dt} = a(y)y = (a_0 - ry)y$$

解为

$$y(t) = \frac{a_0/r}{1 + \left(\frac{a_0}{ry_0} - 1\right)e^{-rt}} = \frac{y_{\max}}{1 + \left(\frac{y_{\max}}{y_0} - 1\right)e^{-rt}}$$



应用：产品周期、
行业趋势、...

题外：围棋与电子游戏中的等级与对战匹配...

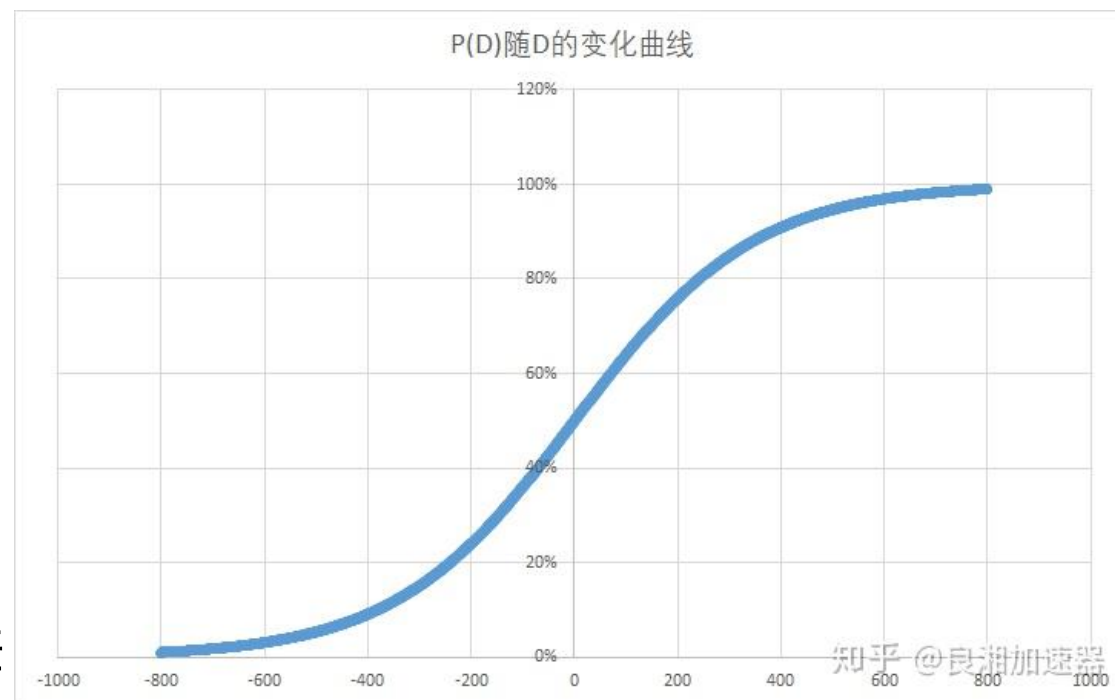
■ ELO算法

- 假设选手的表现在其实力（真实等级分数 x ）附近一定范围内波动（噪声），符合正态分布。
- 如果等级分数比对方高 Δx ，则获胜概率与高斯误差函数有关，一般近似成如下简单公式：

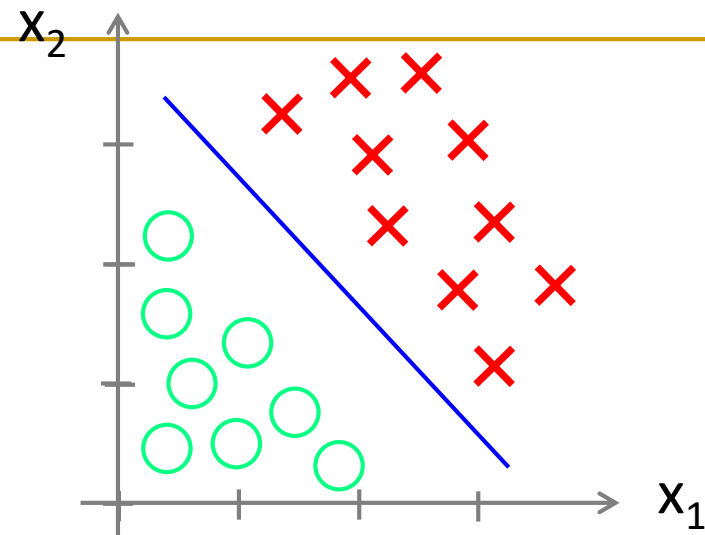
$$P(\Delta x) = \frac{1}{1 + e^{-\alpha \Delta x}}$$

- α 是个参数。
- 通过对比比赛结果的观察来推断真实水平并进行等级调整。

国际象棋中分数差
对胜率的影响



逻辑回归：决策面



$$y(\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x})$$

$\sigma(\mathbf{w}^T \mathbf{x}) \geq 0.5$ 时，预测 \mathbf{x} 属于第一类；否则预测其属于第二类。

- 因此，所预测的第一类与第二类的边界（决策面，decision surface或decision boundary），由下式所描述：

$$\sigma(\mathbf{w}^T \mathbf{x}) = 0.5$$

$$\mathbf{w}^T \mathbf{x} = 0$$

是 \mathbf{x} 空间中的一个线性平面。

- 由于 σ 的存在， y 对 \mathbf{w} 不再是线性函数。因此，优化求解比线性回归复杂。

逻辑回归：似然函数

- 已知 $\{\mathbf{x}_n, t_n\}$ ，求 $y(\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x})$ 中参数 \mathbf{w} 的最佳值。
- 当 $t_n = 1$ （第一类）时，预测结果与 t_n 一致的概率是 $y_n \equiv y(\mathbf{x}_n) = \sigma(\mathbf{w}^T \mathbf{x}_n)$
- 当 $t_n = 0$ （第二类）时，预测结果与 t_n 一致的概率是 $1 - y_n$ 。
- 综合起来，可写成： $y_n^{t_n}(1 - y_n)^{1-t_n}$
- 似然函数

$$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^N y_n^{t_n}(1 - y_n)^{1-t_n}$$

逻辑回归：误差函数

- 基于似然函数，定义（交叉熵）误差函数
 - 关于交叉熵，参考Bishop 1.6.1或刘志荣书稿第3章

$$E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w}) = -\sum_{n=1}^N [t_n \ln y_n + (1 - t_n) \ln (1 - y_n)]$$

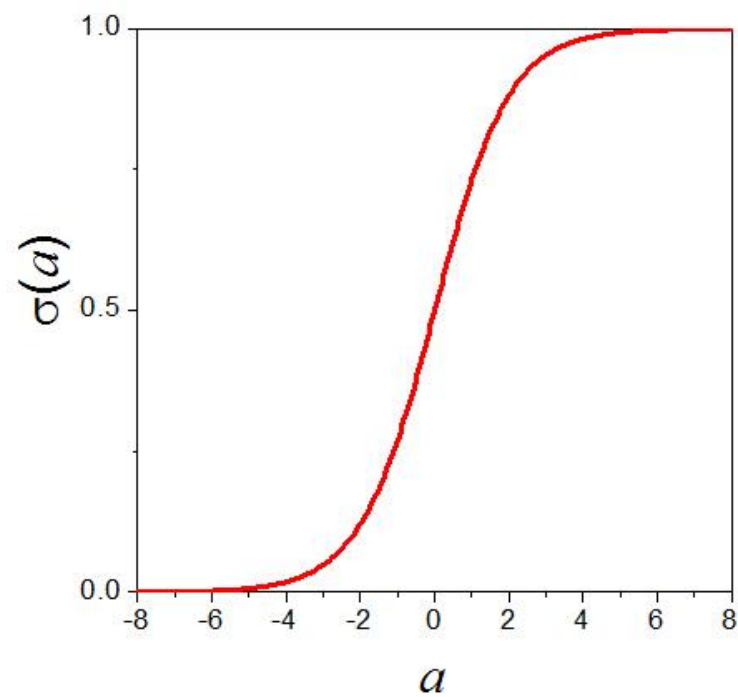
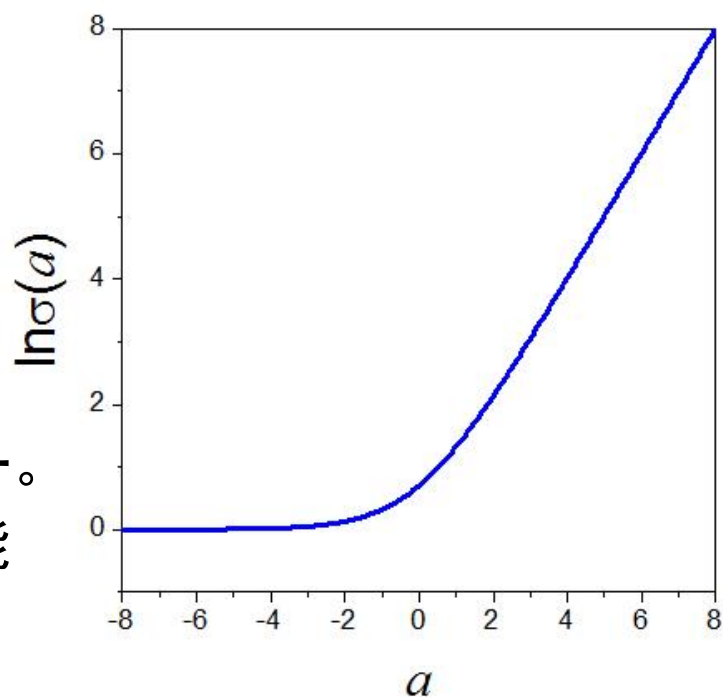
- 题外：规则溶液的混合熵为：

$$\Delta S_{\text{mixing}} = -Nk_B [x \ln x + (1 - x) \ln (1 - x)]$$

额外的好处...

- 上面定义的误差函数是凸(convex)函数，具有良好的数学性质，容易求解最大/最小值。
- 而按通常的最小二乘法定义得到的误差函数则是非凸(non-convex)函数，数值求解困难。

凸函数的最小值存在且唯一。
沿着某种方向持续下降总能找到最小点。



梯度

- $\sigma(a) = \frac{1}{1+e^{-a}}$
- $\frac{d\sigma(a)}{da} = \frac{-e^{-a}}{-(1+e^{-a})^2} = \sigma(a)[1 - \sigma(a)]$
- $E(\mathbf{w}) = -\sum_{n=1}^N [t_n \ln y_n + (1 - t_n) \ln(1 - y_n)]$
- $\frac{dE(\mathbf{w})}{d\mathbf{w}} = -\sum_{n=1}^N \left[\frac{t_n}{y_n} - \frac{(1-t_n)}{1-y_n} \right] \frac{dy_n}{d\mathbf{w}} = -\sum_{n=1}^N \left[\frac{t_n}{y_n} - \frac{(1-t_n)}{1-y_n} \right] \frac{d\sigma(\mathbf{w}^T \mathbf{x}_n)}{d\mathbf{w}}$
 $= -\sum_{n=1}^N \left[\frac{t_n}{y_n} - \frac{(1-t_n)}{1-y_n} \right] y_n(1 - y_n) \mathbf{x}_n = \sum_{n=1}^N (y_n - t_n) \mathbf{x}_n$

梯度下降法

- 用于求解能使代价函数最小的参数。
- $\frac{dE(\mathbf{w})}{d\mathbf{w}} \equiv \nabla_{\mathbf{w}}E(\mathbf{w})$ 给出了 $E(\mathbf{w})$ 随 \mathbf{w} 变化最快的（未归一化）方向。
- 迭代求解：

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla_{\mathbf{w}}E(\mathbf{w})$$

- η 是步长或学习率，可取固定值或（通过一维方法得到）优化值。
- 这种方法在机器学习的优化（训练）中广泛使用。
- 更高级的优化方法：共轭梯度法、质牛顿法（如BFGS）

在线学习

$$E(\mathbf{w}) = - \sum_{n=1}^N [t_n \ln y_n + (1 - t_n) \ln(1 - y_n)]$$

- 当数据个数太多时，可采用：

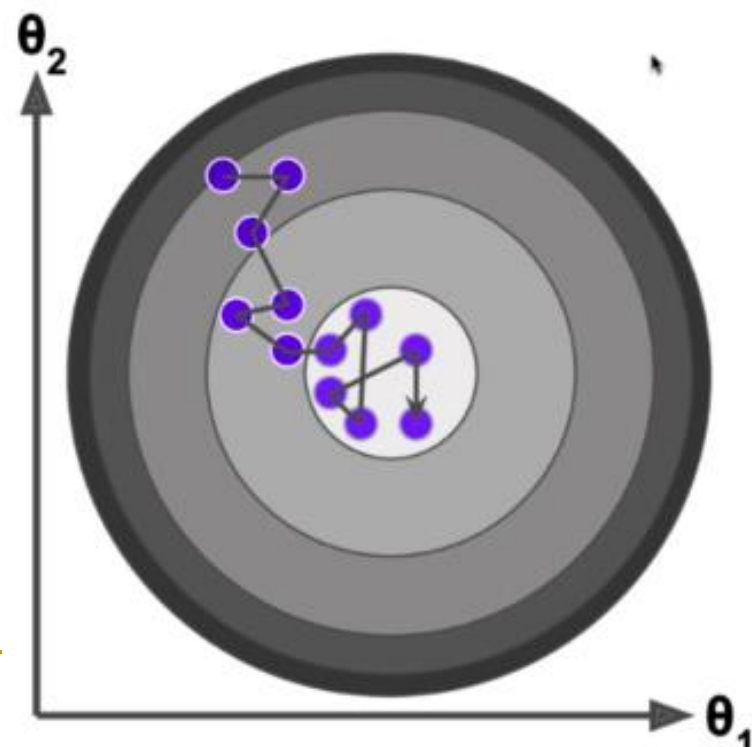
$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla_{\mathbf{w}} E_n(\mathbf{w})$$

即迭代时每次只利用一个训练数据点。

这也被称为随机梯度下降法（Stochastic gradient descent，简称SGD）。

- 可能有利于逃离鞍点

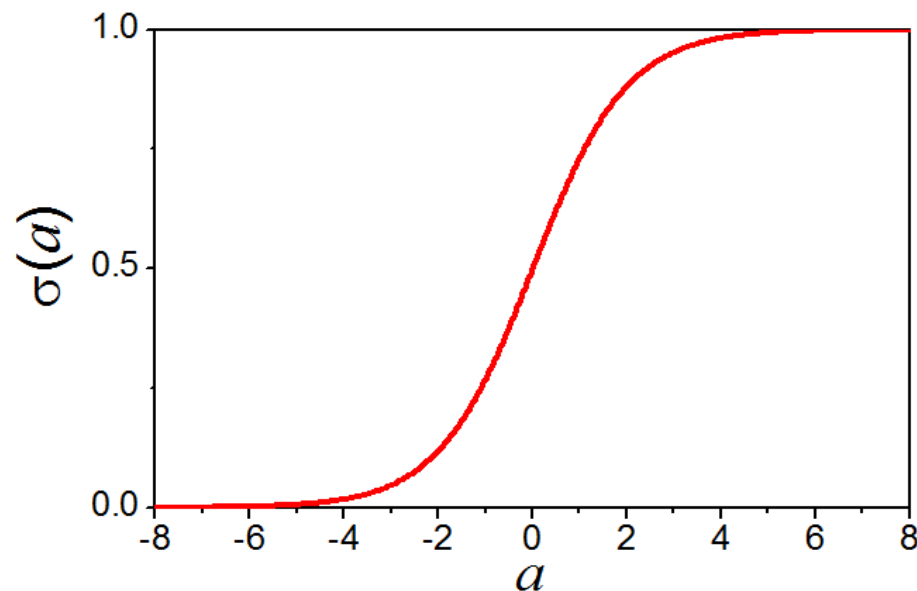
随机梯度下降法 Stochastic



逻辑回归的直观图像

$$y(\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x})$$

- <https://www.jiqizhixin.com/articles/2018-10-19-15>
达观数据王子豪：这5个例子，小学生都能秒懂分类算法.mht
- “上课讲话扣1分，不交作业扣2分，比赛得奖加5分”
(纪律委员)
- 得分 $\mathbf{w}^T \mathbf{x}$ 越高，属于“表现好”的概率 y 就越大。
- 思考：课程成绩的等级制。



基函数的使用

- $y(\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x})$
- 与线性回归类似，可引入基函数。

$$y(\mathbf{x}) = \sigma \left(\sum_{j=0}^M w_j \phi_j(\mathbf{x}) \right) = \sigma(\mathbf{w}^T \boldsymbol{\Phi}(\mathbf{x}))$$

- 此时 \mathbf{w} 的数目不再与 \mathbf{x} 的分量数目相关，而是等于基函数的数目。

$$\frac{dE(\mathbf{w})}{d\mathbf{w}} = - \sum_{n=1}^N \left[\frac{t_n}{y_n} - \frac{(1 - t_n)}{1 - y_n} \right] \frac{d\sigma}{d\mathbf{w}} = \sum_{n=1}^N (y_n - t_n) \boldsymbol{\Phi}(\mathbf{x}_n)$$

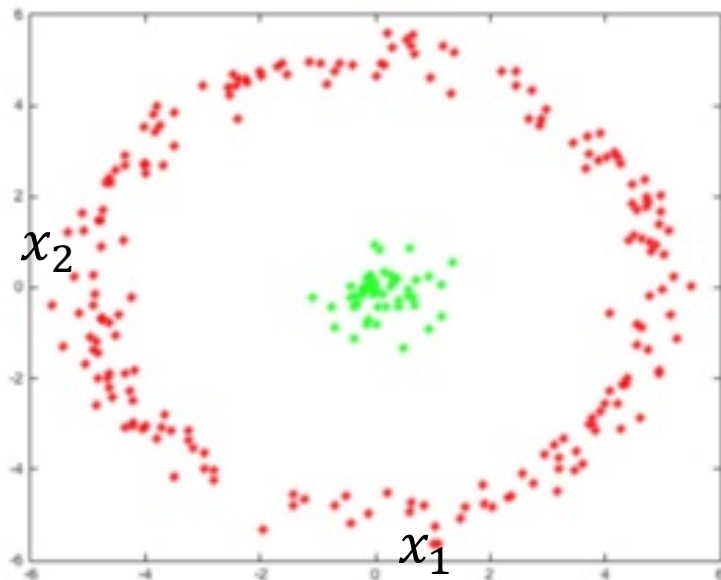


Figure 1: Original Data

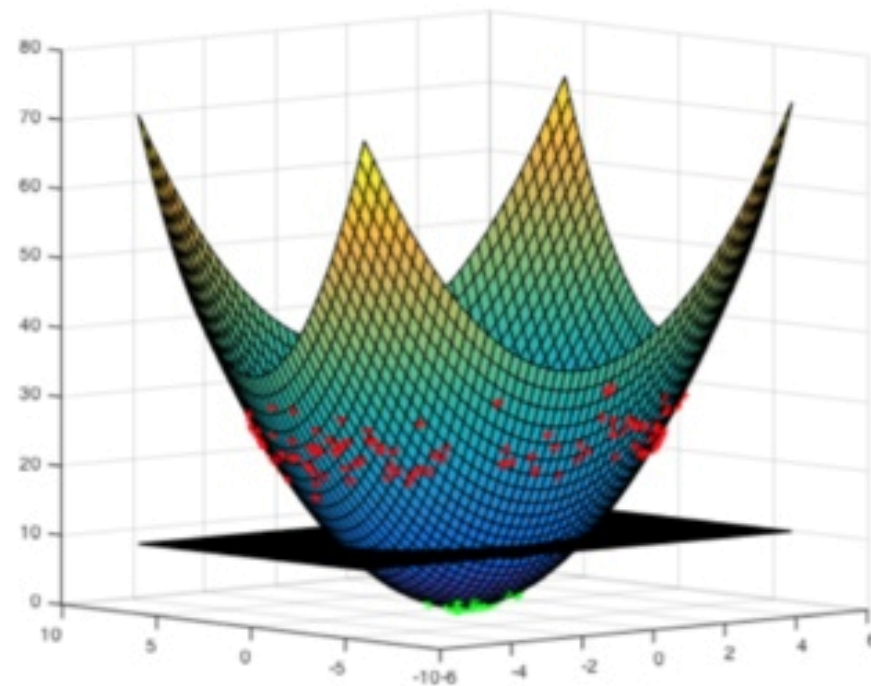


Figure 2: Data on feature space

- 基函数:

$$1, \quad x_1, \quad x_2, \quad x_1^2, \quad x_1x_2, \quad x_2^2$$

- 低维线性不可分的数据投影（变换）到高维空间后很可能变得线性可分。

正则化的使用

$$E(\mathbf{w}) = - \sum_{n=1}^N [t_n \ln y_n + (1 - t_n) \ln(1 - y_n)] + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w}$$

- 相当于最大化后验概率。

贝叶斯逻辑回归

- 完整的贝叶斯观点（略，参见Bishop 4.5）

多类别分类 (Multi-class classification)

- 例子

- Email标签/整理:

工作Work、朋友Friends、家庭Family、爱好Hobby



$t = 1$



2



3



4

- 天气预报: 晴、多云、雨



$t = 1$



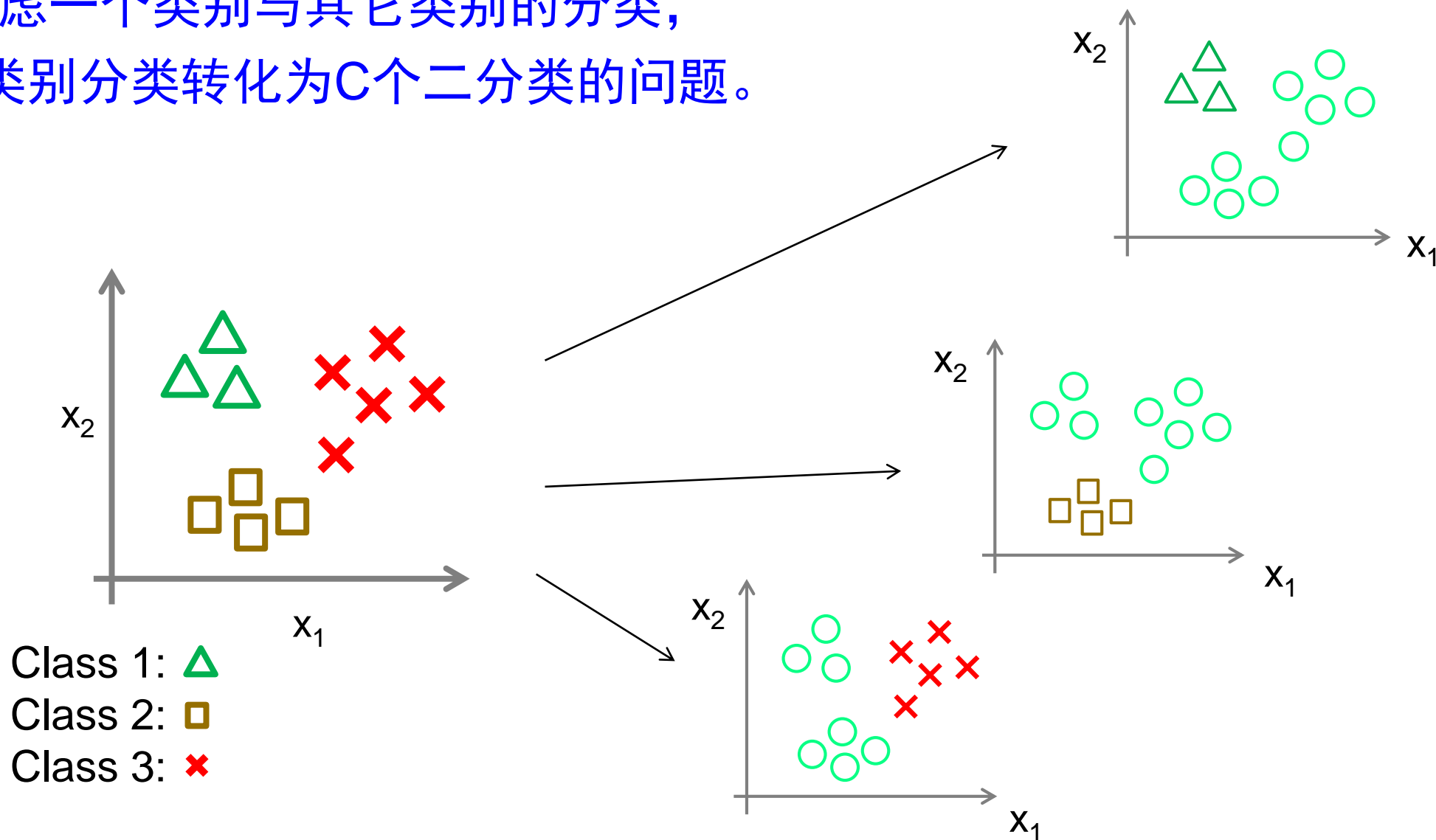
2



3

多类别分类的方法1：one-vs-rest

- 每次考虑一个类别与其它类别的分类，
将多类别分类转化为C个二分类的问题。



一种简单的做法：

- 对每个类别 i ，训练一个逻辑回归分类器 $y^{(i)}(\mathbf{x})$ 以区分类别 i 与其它类别；
- 对于一个新的 \mathbf{x} ，选择一个让 $y^{(i)}(\mathbf{x})$ 最大的 i 作为多类别的预测结果。

多类别分类的方法2: softmax

- 应用更多。
- 对每个类别 i ，定义

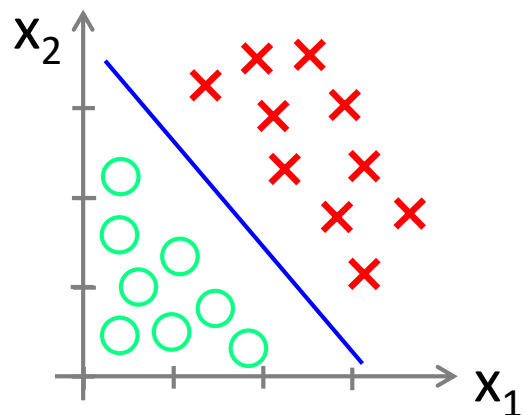
$$z_i(\mathbf{x}) = \mathbf{w}_{(i)}^T \mathbf{x}$$

- 利用softmax函数来计算数据点 \mathbf{x} 属于类别 i 的概率：

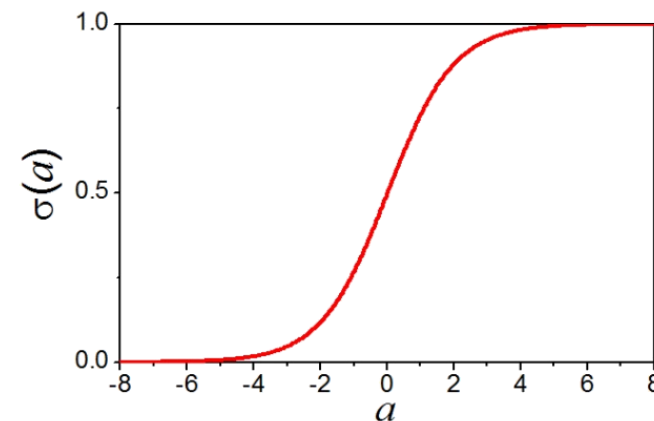
$$y^{(i)}(\mathbf{x}) = \frac{e^{z_i}}{\sum_i e^{z_i}} = \frac{e^{\mathbf{w}_{(i)}^T \mathbf{x}}}{\sum_i e^{\mathbf{w}_{(i)}^T \mathbf{x}}}$$

- 这适用于类别互斥的情形。

小结：逻辑回归



$$y(\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x})$$



其中激活函数： $\sigma(a) = \frac{1}{1+e^{-a}}$ ，可理解成概率。

- 似然函数： $p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^N y_n^{t_n} (1 - y_n)^{1-t_n}$
- 交叉熵误差函数： $E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w}) = -\sum_{n=1}^N [t_n \ln y_n + (1 - t_n) \ln(1 - y_n)]$
- 梯度： $\frac{dE(\mathbf{w})}{d\mathbf{w}} = \sum_{n=1}^N (y_n - t_n) \mathbf{x}_n$
- 梯度下降法、随机梯度下降法。

应用例子1： 点击率预测

- H. Brendan McMahan, Gary Holt, ..., Jeremy Kubica. Ad Click Prediction: a View from the Trenches. Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining, 1222-1230 (2013).
- H. B. McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and L1 regularization. In AISTATS, 2011

学习资料： [p1222-mcmahan.pdf](#), [mcmahan11b.pdf](#)

背景

- 互联网广告，或者个性化推荐，甚至一般的互联网产品，一个重要指标就是点击率（click through rate, CTR）。
- 线上来的每一个广告请求，都提取出相应的特征，再根据模型的参数，计算一个点击某广告的概率。
- 网站收入：
$$\text{展示次数} * \text{点击率} * \text{广告价格}$$

Baidu 百度 机器学习课程

网页 资讯 视频 图片 知道 文库 贴吧 采购 地图 更多»

百度为您找到相关结果约7,600,000个

机器学习 和深度学习的区别 机器学习理论+实战 可分期

机器学习 和深度学习的区别, 高端机器学习培训选青鸟, 与名企合作, 入学签署就业协议, 理论+实战, 分期付款, 先试听后入学, 点击马上咨询在线老师.

itpc.bdqwin.com 2019-09 - 评价 广告

【优达学城】机器学习课程 来自硅谷优达学城的机器学习公开课

超人气口碑硅谷机器学习课程, 5年良心出品, 9个不同实战项目, 立抢限量席位. 硅谷导师一对一直播辅导, 4个月快速获得实战项目经验, 初级与进阶课程可选.

cn.udacity.com 2019-09 - 评价 广告

机器学习课程 免费公开课

针对企事业单位实际, 中科信软提供高级机器学习课程, 可按需求培训, 上门内训, 公开课培训专业机器学习课程.

www.info-soft.cn 2019-09 - 评价 广告

【51cto】51CTO学院-唐老师机器学习课程 实战视频课程

51CTO, 技术用户遍布全国. 机器学习课程 免费试看, 适用于机器学习/人工智能爱好者机器学习课程 将通俗易懂的原理讲解与实例代码结合带大家学习机器学习的经典算法.

edu.51cto.com 2019-09 - 评价 广告

【北风网】机器学习课程——过来人的经验机器学习课程

获得网友: 90%好评 744条评价 “服务周到 | 老师好 | 内容丰富”

机器学习课程 我选北风网! 总监实战教学机器学习课程, 0元入门到精通, 项目不断. 20万学员..

www.beifeng.org.cn 2019-09 - 评价 广告

吴恩达机器学习 - 网易云课堂

本门课程是 Coursera 上的第一门课, 也是吴恩达(Andrew Ng)老师的经典之作, 并授权网易汉化发布. 从2011年上线到2017年, 本门课程已经在全球

https://study.163.com/course/i... - 百度快照

机器学习课程 - 机器学习 | Coursera

Learn 机器学习 from 斯坦福大学. Machine learning is the science of getting computers to act without being explicitly programmed. In the past decade, ...

www.coursera.or... - 百度快照

【经典】吴恩达《机器学习》课程 - Beyond 2016的博客 - CSDN博客

2018年9月2日 -

在线学习的挑战

- 为了保证快速的更新，训练样本是一条一条地过来的，每来一个样本，模型的参数对这个样本进行一次迭代，从而保证了模型的及时更新。
- 只能扫描一次样本，而且样本是一条一条地过来的。
- 为了减小误差，以及增加稀疏性，人们提出了多个版本的算法
 - FOBOS (Forward Backward Splitting)
 - 微软的 RDA
 - Google 的 FTRL-Proximal。Follow-the-regularized-leader

If we wish to model our problem using logistic regression, we can use the following online framework. On round t , we are asked to predict on an instance described by feature vector $\mathbf{x}_t \in \mathbb{R}^d$; given model parameters \mathbf{w}_t , we predict $p_t = \sigma(\mathbf{w}_t \cdot \mathbf{x}_t)$, where $\sigma(a) = 1/(1 + \exp(-a))$ is the sigmoid function. Then, we observe the label $y_t \in \{0, 1\}$, and suffer the resulting LogLoss (logistic loss), given as

$$\ell_t(\mathbf{w}_t) = -y_t \log p_t - (1 - y_t) \log(1 - p_t), \quad (1)$$

the negative log-likelihood of y_t given p . It is straightforward to show $\nabla \ell_t(\mathbf{w}) = (\sigma(\mathbf{w} \cdot \mathbf{x}_t) - y_t)\mathbf{x}_t = (p_t - y_t)\mathbf{x}_t$, and this gradient is all we will need for optimization purposes.

Online gradient descent¹ (OGD) has proved very effective for these kinds of problems, producing excellent prediction accuracy with a minimum of computing resources. However, in practice another key consideration is the size of the final model; since models can be stored sparsely, the number of non-zero coefficients in \mathbf{w} is the determining factor of

Given a sequence of gradients $\mathbf{g}_t \in \mathbb{R}$, OGD performs the update

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathbf{g}_t,$$

where η_t is a non-increasing learning-rate schedule, e.g., $\eta_t = \frac{1}{\sqrt{t}}$. The FTRL-Proximal algorithm instead uses the update

$$\mathbf{w}_{t+1} = \arg \min_{\mathbf{w}} \left(\mathbf{g}_{1:t} \cdot \mathbf{w} + \frac{1}{2} \sum_{s=1}^t \sigma_s \|\mathbf{w} - \mathbf{w}_s\|_2^2 + \lambda_1 \|\mathbf{w}\|_1 \right),$$

where we define σ_s in terms of the learning-rate schedule such that $\sigma_{1:t} = \frac{1}{\eta_t}$. On the surface, these updates look very different, but in fact when we take $\lambda_1 = 0$, they produce an *identical* sequence of coefficient vectors. However, the FTRL-Proximal update with $\lambda_1 > 0$ does an excellent job of inducing sparsity (see experimental results below).

can re-write the update as the argmin over $\mathbf{w} \in \mathbb{R}^d$ of

$$\left(\mathbf{g}_{1:t} - \sum_{s=1}^t \sigma_s \mathbf{w}_s \right) \cdot \mathbf{w} + \frac{1}{\eta_t} \|\mathbf{w}\|_2^2 + \lambda_1 \|\mathbf{w}\|_1 + (\text{const}).$$

Thus, if we have stored $\mathbf{z}_{t-1} = \mathbf{g}_{1:t-1} - \sum_{s=1}^{t-1} \sigma_s \mathbf{w}_s$, at the beginning of round t we update by letting $\mathbf{z}_t = \mathbf{z}_{t-1} + \mathbf{g}_t + (\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}})\mathbf{w}_t$, and solve for \mathbf{w}_{t+1} in closed form on a per-coordinate bases by

$$w_{t+1,i} = \begin{cases} 0 & \text{if } |z_{t,i}| \leq \lambda_1 \\ -\eta_t(z_{t,i} - \text{sgn}(z_{t,i})\lambda_1) & \text{otherwise.} \end{cases}$$

Thus, FTRL-Proximal stores $\mathbf{z} \in \mathbb{R}^d$ in memory, whereas OGD stores $\mathbf{w} \in \mathbb{R}^d$. Algorithm 1 takes this approach, but

Algorithm 1 Per-Coordinate FTRL-Proximal with L_1 and L_2 Regularization for Logistic Regression

With per-coordinate learning rates of Eq. (2).

Input: parameters $\alpha, \beta, \lambda_1, \lambda_2$

$(\forall i \in \{1, \dots, d\})$, initialize $z_i = 0$ and $n_i = 0$

for $t = 1$ **to** T **do**

 Receive feature vector \mathbf{x}_t and let $I = \{i \mid x_i \neq 0\}$

 For $i \in I$ compute

$$w_{t,i} = \begin{cases} 0 & \text{if } |z_i| \leq \lambda_1 \\ -\left(\frac{\beta + \sqrt{n_i}}{\alpha} + \lambda_2\right)^{-1} (z_i - \text{sgn}(z_i)\lambda_1) & \text{otherwise.} \end{cases}$$

 Predict $p_t = \sigma(\mathbf{x}_t \cdot \mathbf{w})$ using the $w_{t,i}$ computed above

 Observe label $y_t \in \{0, 1\}$

for all $i \in I$ **do**

$g_i = (p_t - y_t)x_i$ *#gradient of loss w.r.t. w_i*

$\sigma_i = \frac{1}{\alpha} \left(\sqrt{n_i + g_i^2} - \sqrt{n_i} \right)$ *#equals $\frac{1}{\eta_{t,i}} - \frac{1}{\eta_{t-1,i}}$*

$z_i \leftarrow z_i + g_i - \sigma_i w_{t,i}$

$n_i \leftarrow n_i + g_i^2$

end for

end for

应用例子2: MOFs中金属离子的价态

- Kevin Maik Jablonka, Daniele Ongari, Seyed Mohamad Moosavi and Berend Smit. Using collective knowledge to assign oxidation states of metal cations in metal–organic frameworks. *Nature Chemistry* **13**, 771-777 (2021).

学习资料: nc717.pdf

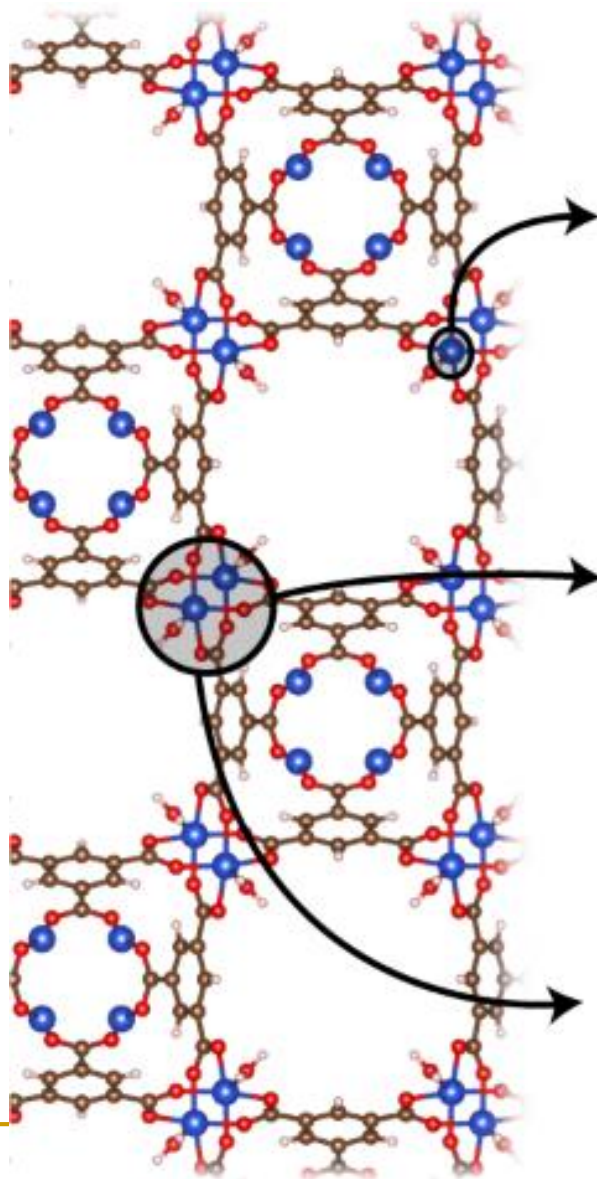
媒体介绍: <https://www.jiqizhixin.com/articles/2021-09-17-9>

背景

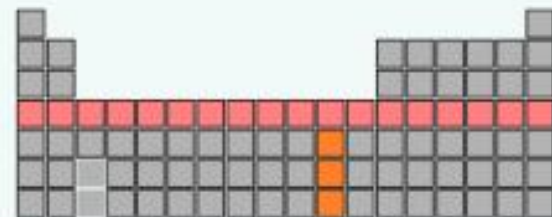
- 氧化态对于平衡氧化还原反应，帮助化学家系统化和解释（氧化还原）反应性以及光谱特性等至关重要。例如，高锰酸盐 (Mn(vii)) 具有很强的氧化性，通常呈紫色，而 Mn(ii) 化合物的反应性较低，通常是无色的。
- 对于复杂的化合物，金属离子的价态并不容易确定。
 - 例如，除了原子距离以外，金属配合物的几何形状也很重要。

特征

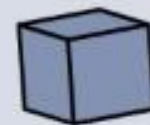
- 金属类型
- 配位环境的几何形状
- 化学环境
 - 例如电负性



Metal centre identity
Row
Column
Electronic configuration

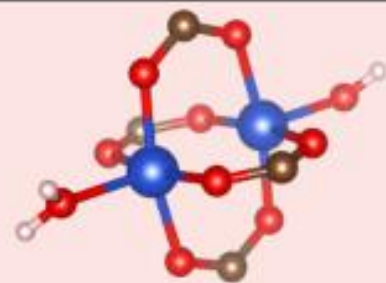


Local coordination
geometry



Local chemistry

Statistics of the properties
of neighbouring elements

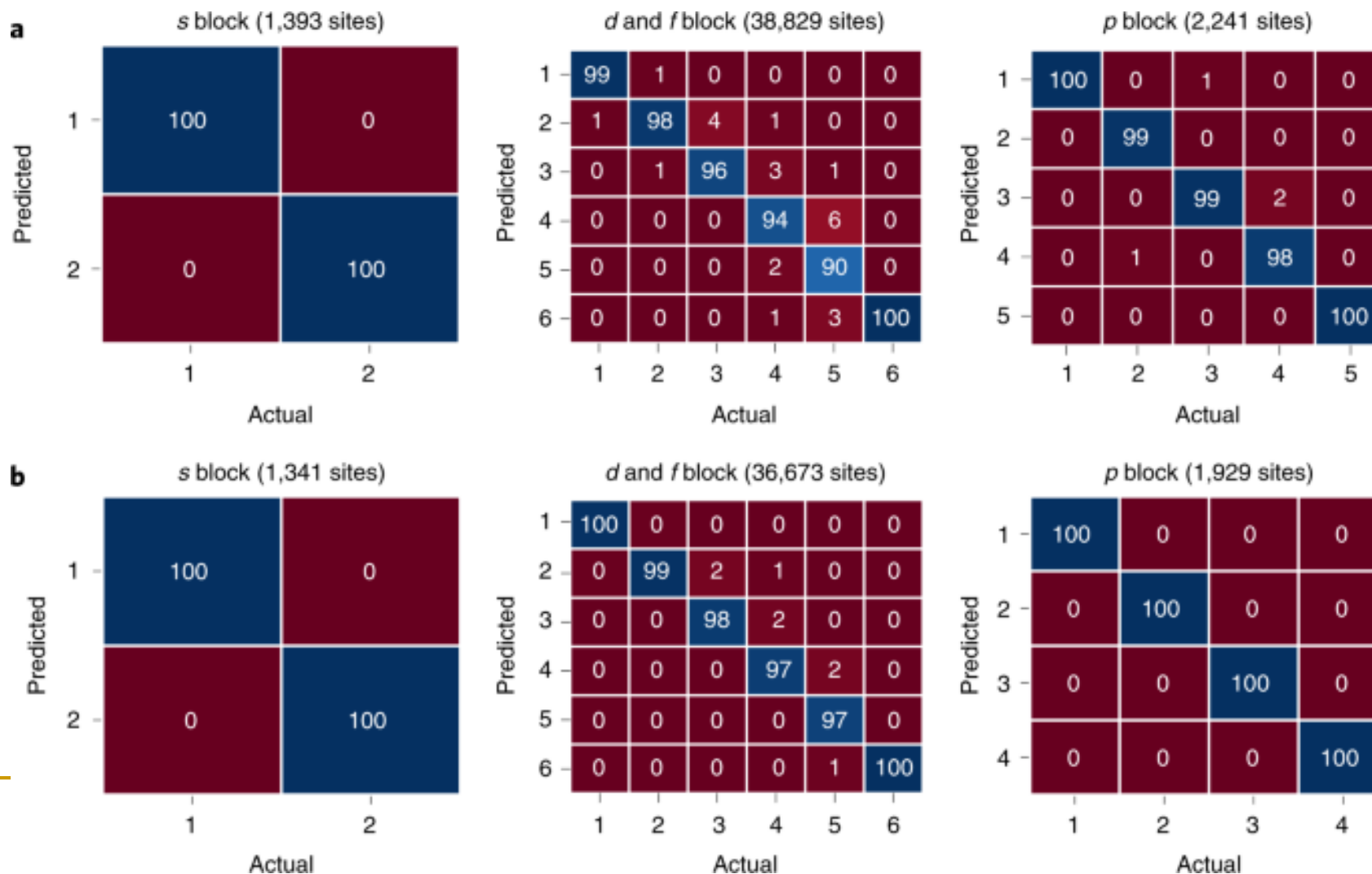


方法

- The ML model adopted in this work is a soft voting classifier using gradient boosting, k-nearest neighbours, **logistic regression** and an extra trees base classifier implemented in the **scikit-learn** library.

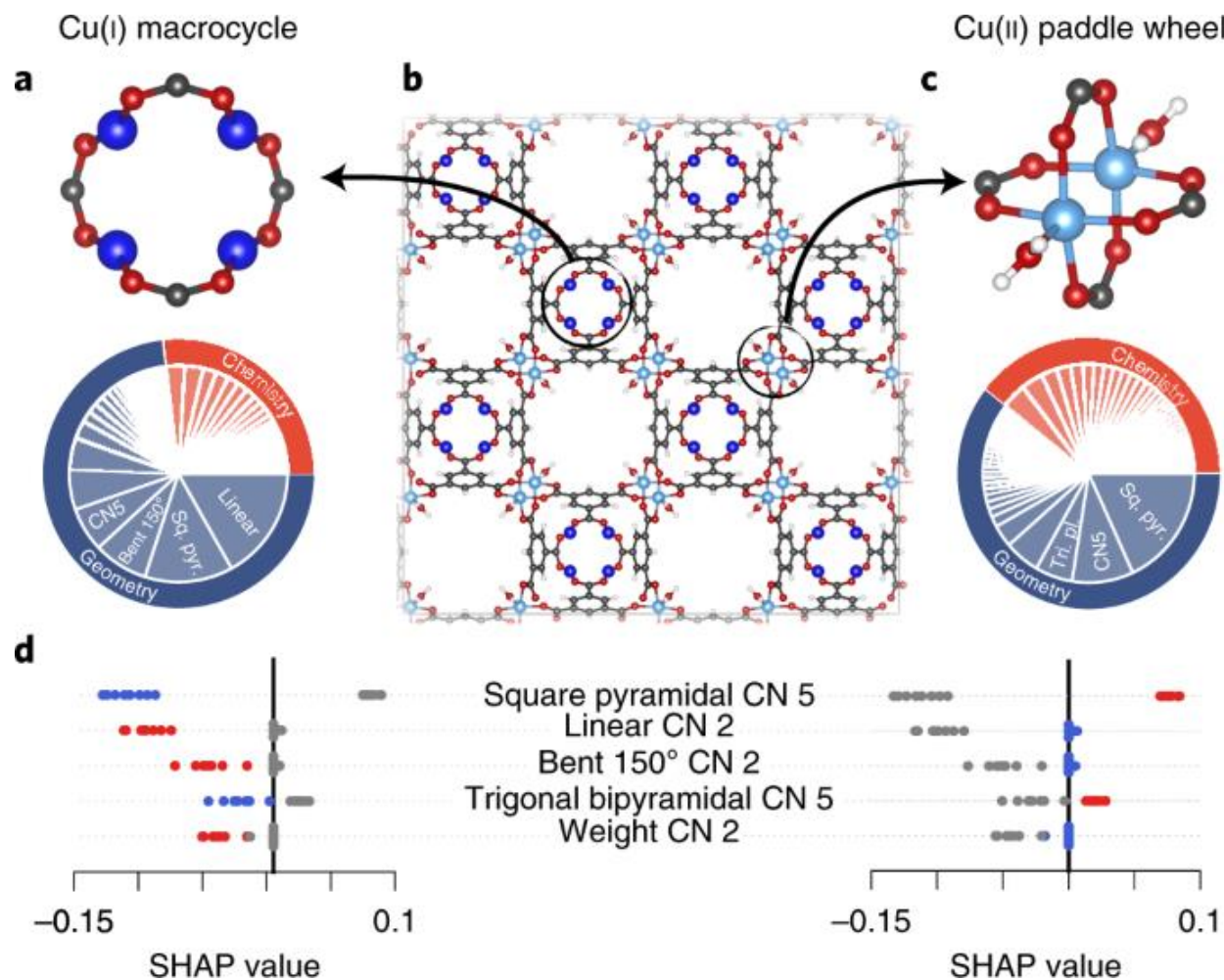
预测性能

- s区、p区、d区和f区金属。



实例研究

- 混合价MOF Cu(I/II) 苯-1,3,5-三羧酸酯 (BTC): 确定16 个金属位点中每一个的氧化态, 与实验数据一致。



Scikit-Learn相关内容

<https://scikit-learn.org/>

<https://sklearn.apacheecn.org/>

- **1.1.11. Logistic regression**
 - **linear_model.LogisticRegression**
- **1.1.12. Stochastic Gradient Descent – SGD**
- 1.5. Stochastic Gradient Descent**
 - **linear_model.SGDClassifier**
 - **linear_model.SGDRegressor**

■ Reference:

- ❑ 刘志荣³
- ❑ Bishop 4.0, 4.1.1-4.1.3, 4.2.1-4.2.3, 4.3.1-4.3.2;
- ❑ Elements 4;
- ❑ 实战⁵
- ❑ 吴恩达 ⁶

■ 扩展阅读：

□ <https://www.jiqizhixin.com/articles/2018-10-19-15>

达观数据王子豪：这5个例子，小学生都能秒懂分类算法.mht

□ <https://www.meihua.info/a/65329>

关于点击率模型，你知道这三点就够了.mht

□ <https://blog.csdn.net/guoziqing506/article/details/81328402>

逻辑回归（logistic regression）原理详解.mht

□ <https://www.pinlue.com/article/2017/08/2723/594294870044.html>

逻辑回归模型的前世今生.mht

它是一个简单的模型，却支撑着互联网新旧巨头最挣钱的核心模块；
它是一个经典的模型 行业内却很少有人能讲清它的前世今生

❑ <https://www.huxiu.com/article/400368.html>

在游戏里虐菜鸟，已经成为一种奢侈？.pdf

目前，游戏（例如DOTA2）的玩家匹配机制都在使用一个叫ELO的算法，这一算法从象棋而来。

❑ <https://zhuanlan.zhihu.com/p/57480433>

ELO算法的原理及应用.pdf

❑ https://bbs.gameres.com/thread_228018_1_1.html

ELO算法教程.mht

可体会逻辑函数的概率含义，以及学习率的使用与学习结果的收敛。

❑ <https://www.jiqizhixin.com/articles/2021-09-17-9>

元素周期表应加上氧化态？机器学习破解晶体结构的氧化态.pdf

谢谢大家！