

机器学习基础

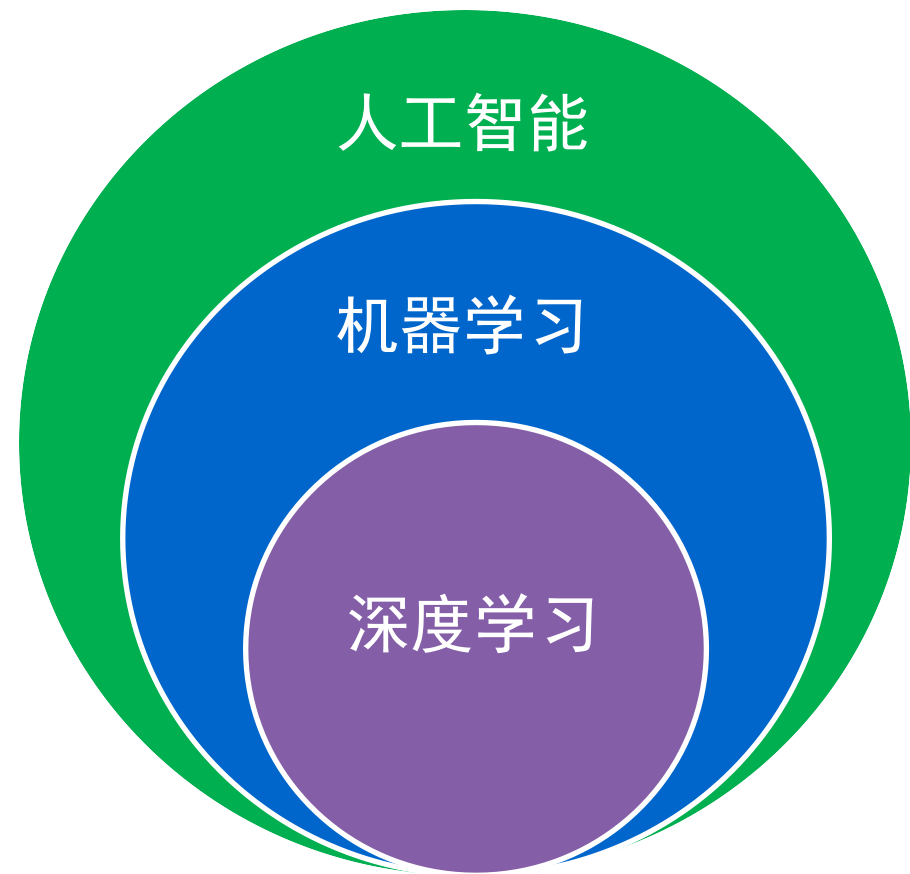
2022.05.05

目录

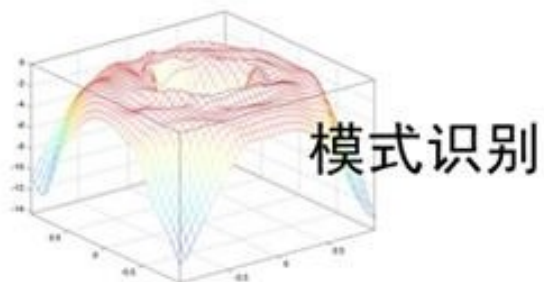
- 1. 机器学习概述
- 2. 机器学习的类型
- 3. 机器学习的背景知识
- 4. 机器学习相关概念
- 5. 机器学习的开发流程
- 6. 机器学习算法--KNN

1. 机器学习概述

- **人工智能：** 机器展现的人类智能
- **机器学习：** 计算机利用已有的数据(经验)，得出了某种模型，并利用此模型预测未来的一种方法
- **深度学习：** 实现机器学习的一种技术



1. 机器学习概述—机器学习范围



数据挖掘



机器学习

语音识别



统计学习



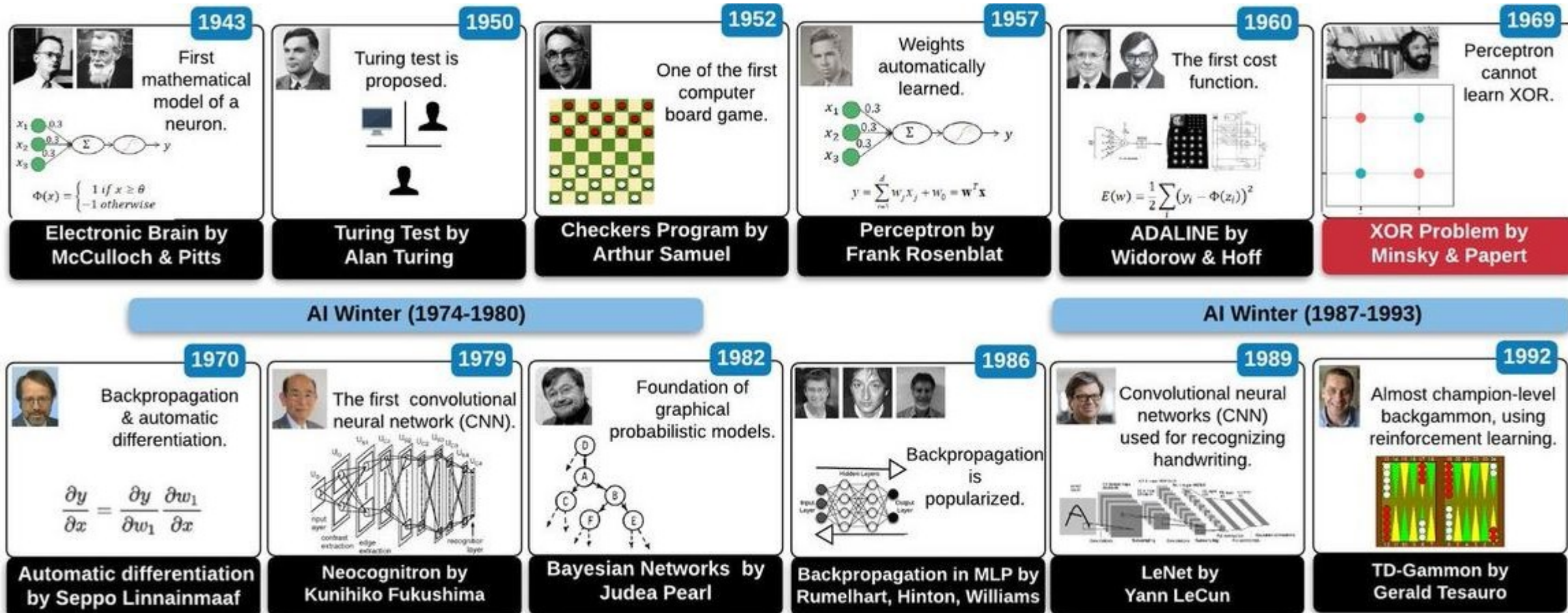
自然语言处理



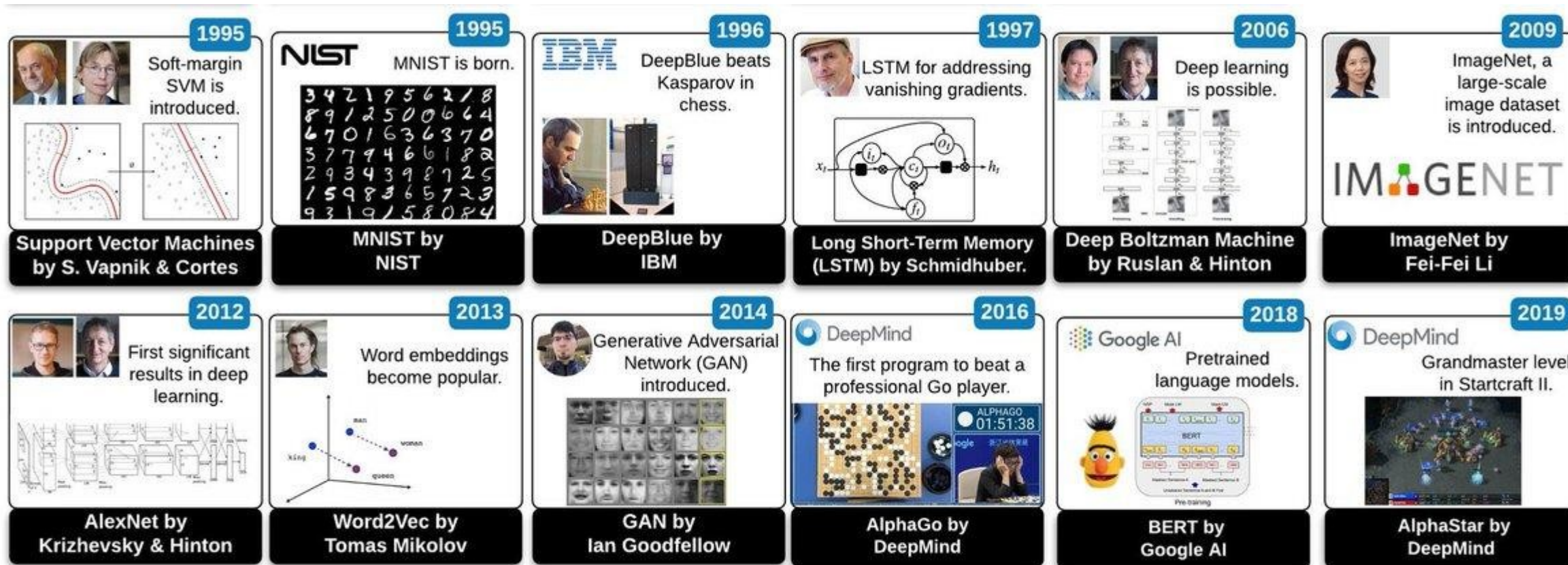
1. 机器学习概述—可以解决什么问题？

- 给定数据的预测问题
 - ✓ 数据清洗/特征选择
 - ✓ 确定算法模型/参数优化
 - ✓ 结果预测
- 不能解决什么
 - ✓ 大数据存储/并行计算
 - ✓ 做一个机器人

1. 机器学习概述—发展史



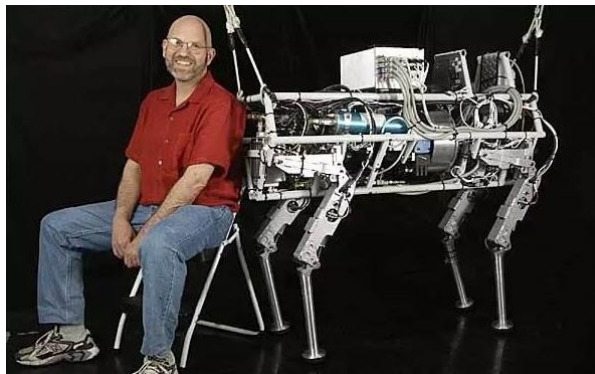
1. 机器学习概述—发展史



1. 机器学习概述—不同视角的机器学习



不同行业的人以为我做的事情



父母以为我做的事情



朋友以为我做的事情

$$\frac{\partial}{\partial w} L(w, b, \alpha) = w - \sum \alpha_i y_i x_i = 0, \quad w = \sum \alpha_i y_i x_i$$

$$\frac{\partial}{\partial b} L(w, b, \alpha) = \sum \alpha_i y_i = 0$$

代入 $L(w, b, \alpha)$

$$\begin{aligned} \min L(w, b, \alpha) &= \frac{1}{2} \|w\|^2 + \sum_{i=1}^m \alpha_i (-y_i (w^T x_i + b) + 1) \\ &= \frac{1}{2} w^T w - \sum_{i=1}^m \alpha_i y_i w^T x_i - b \sum_{i=1}^m \alpha_i y_i + \sum_{i=1}^m \alpha_i \\ &= \frac{1}{2} w^T \sum \alpha_i y_i x_i - \sum_{i=1}^m \alpha_i y_i w^T x_i + \sum_{i=1}^m \alpha_i \\ &= \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \alpha_i y_i w^T x_i \\ &= \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x_i x_j) \end{aligned}$$

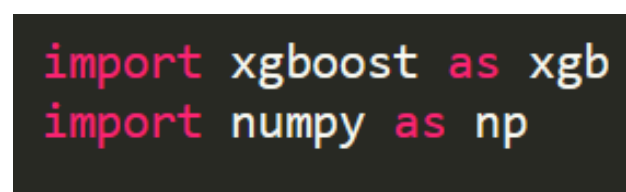
再把 max 问题转成 min 问题:

$$\begin{aligned} \max \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x_i x_j) &= \min \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x_i x_j) - \sum_{i=1}^m \alpha_i \\ \text{s.t. } \sum_{i=1}^m \alpha_i y_i &= 0, \end{aligned}$$

程序员以为我做的事情

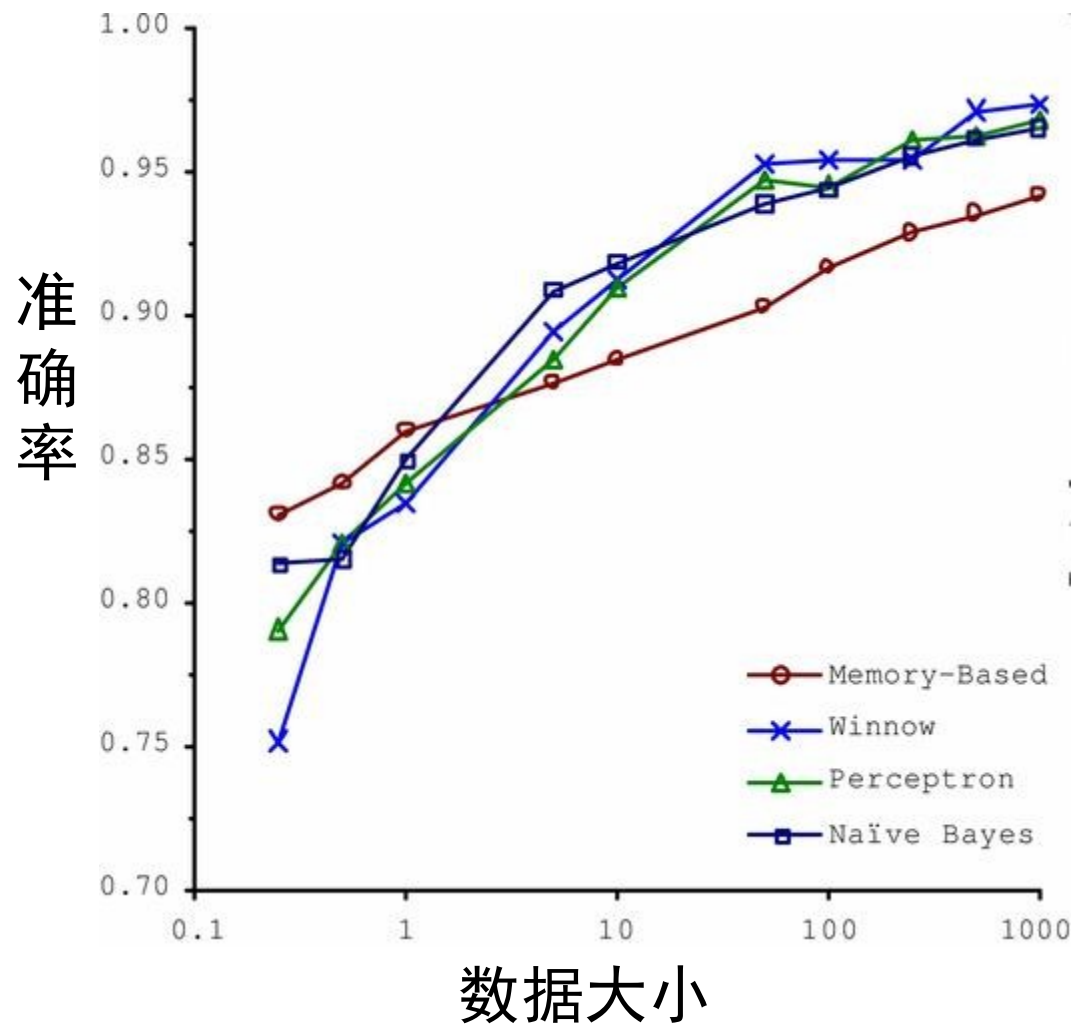


我自己以为我做的事情



实际上我做的事情

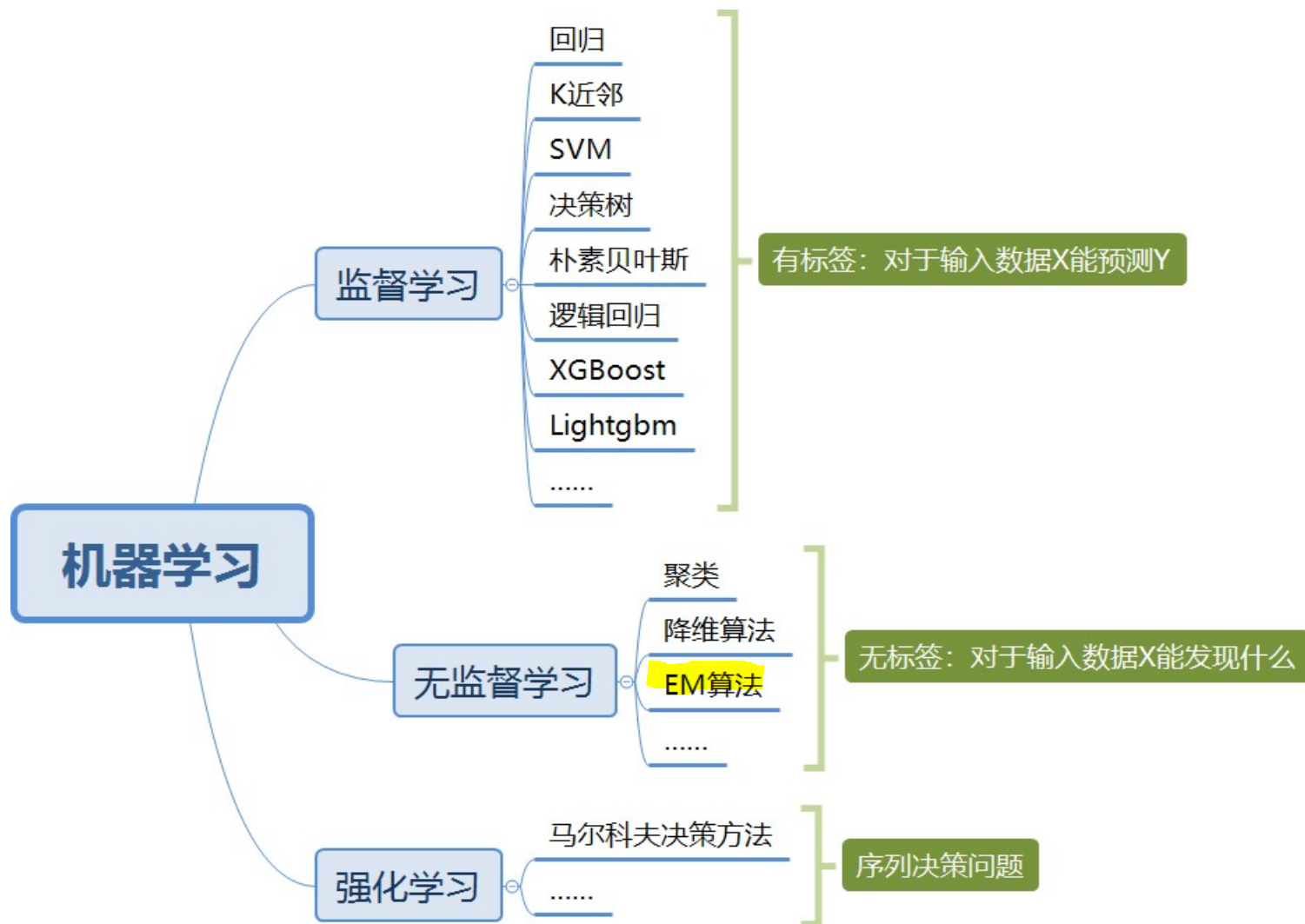
1. 机器学习概述—数据决定一切



通过这张图可以看出，各种不同算法在输入的数据量达到一定级数后，都有相近的高准确度。于是诞生了机器学习界的名言：

成功的机器学习应用不是拥有最好的算法，而是拥有最多的数据！

2. 机器学习的类型



2. 机器学习的类型—监督学习

✓ 分类 (Classification)

- ✓ 身高1.65m, 体重100kg的男人肥胖吗?
- ✓ 根据肿瘤的体积、患者的年龄来判断良性或恶性?

✓ 回归 (Regression、Prediction)

- ✓ 如何预测上海浦东的房价?
- ✓ 未来的股票市场走向?

2. 机器学习的类型—无监督学习

- ✓ 聚类 (**Clustering**) 没有标签, 划分
 - ✓ 如何将教室里的学生按爱好、身高划分为5类?
- ✓ 降维 (**Dimensionality Reduction**)
 - ✓ 如何将原高维空间中的数据点映射到低维度的空间中?

2. 机器学习的类型—强化学习

- ✓ 强化学习（Reinforcement Learning）
 - ✓ 用于描述和解决智能体（agent）在与环境的交互过程中通过学习策略以达成回报最大化或实现特定目标的问题。

3. 机器学习背景知识

- 数学基础

- ✓ 高等数学：导数、微分、泰勒公式…
- ✓ 线性代数：向量、矩阵…
- ✓ 概率论与数理统计：概率基本公式、常见分布、期望、协方差…

- Python基础

- ✓ 环境安装：Anaconda、Jupyter Notebook、Pycharm、VSCode
- ✓ Python基础
- ✓ Python库：numpy、pandas、scipy、matplotlib、scikit-learn…

4. 机器学习相关概念

- ✓ 机器学习方法
 - ✓ 模型
 - ✓ 损失函数
 - ✓ 优化算法
 - ✓ 模型评估指标

4. 机器学习相关概念—模型

机器学习首先要考虑使用什么样的模型。

模型的类别，大致有两种：一是概率模型(Probabilistic Model)和非概率模型(Non-Probabilistic Model)。

在监督学习中，概率模型可被表示为 $P(y|x)$ ，非概率模型则为 $y = f(x)$ 。其中， x 是输入， y 是输出。

在无监督学习中，概率模型可被表示为 $P(z|x)$ ，非概率模型则为 $z = f(x)$ 。其中， x 是输入， z 是输出。

4. 机器学习相关概念—模型

决策树、朴素贝叶斯、隐马尔科夫模型、高斯混合模型属于**概率模型**。

感知机、支持向量机、KNN、AdaBoost、K-means以及神经网络均属于**非概率模型**。

直观理解：拟合数据的分布函数

假设数据的特征是 $X = (x_1, x_2, \dots, x_n)$ W 各个维度的权重

我们需要找到一个函数能表示所有数据： $y = WX = w_1x_1 + w_2x_2 + \dots + w_nx_n$

例如，猫的特征是 $X = (10, 20, 30)$ ， $y = WX = w_1x_1 + w_2x_2 + w_3x_3 = w_1 \times 10 + w_2 \times 20 + w_3 \times 30 = \begin{cases} 0 \\ 1 \end{cases}$

如何学习这些权重 $W = (w_1, w_2, \dots, w_n)$ ？

度量预测的 y 和真实的 \bar{y} 之间的差异，即计算loss值，通过使得loss最小化使得差异最小化

4. 机器学习相关概念—损失函数 loss

1. 0-1损失函数(0-1 Loss Function)

$$L(Y, f(X)) = \begin{cases} 1, & Y \neq f(X) \\ 0, & Y = f(X) \end{cases}$$

2. 平方损失函数(Quadratic Loss Function)

$$L(Y, f(X)) = (Y - f(X))^2$$

3. 绝对损失函数(Absolute Loss Function)

$$L(Y, f(X)) = |Y - f(X)|$$

4. 对数损失函数(Logarithmic Loss Function)

$$L(Y, P(Y|X)) = -\log P(Y|X)$$

4. 机器学习相关概念—损失函数

根据上述损失函数模型，我们可知，损失函数值越小，模型性能越好。给定一个数据集，我们将训练数据集的平均损失称为经验风险。基于经验风险最小化原则，可构建全局损失函数求解最优化问题：

$$\min_f \frac{1}{N} \sum_{n=1}^N L(y_n, f(x_n))$$

当样本数量足够大时，根据大数定理，经验风险会近似于模型的期望风险。此时，经验风险最小化能确保有好的学习性能。然而，当样本数量不足时，单单利用经验风险最小化可能会导致“**过拟合**”的问题。为此，我们再原有基础上加上用于控制模型复杂度的正则项 (Regularizer)，得到结构最小化准则。

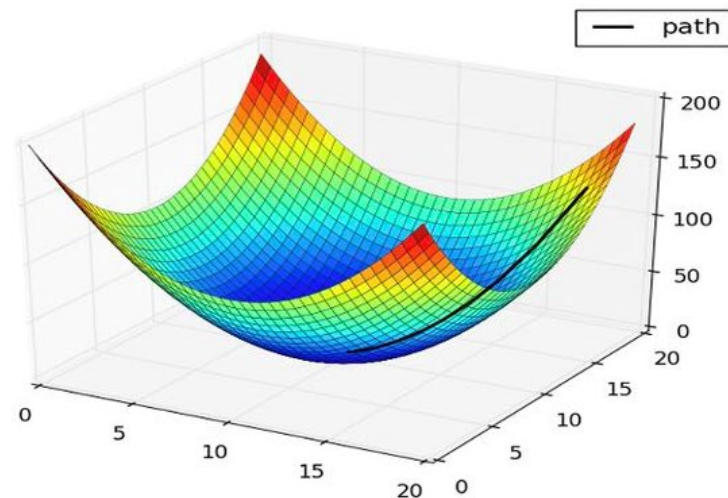
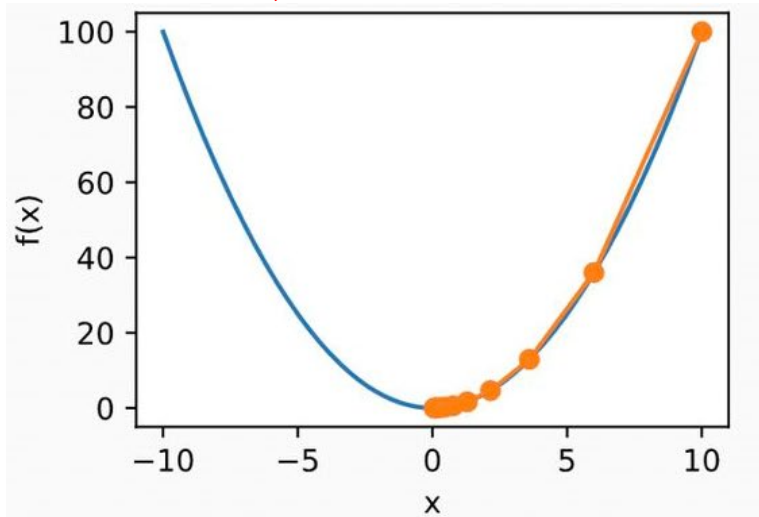
4. 机器学习相关概念—优化算法

如何使得loss最小化？

算法指的是模型学习中的具体计算方法。一般来说，基于参数模型构建的统计学习问题都为最优化问题，它们都具有显式的解析解。

现有的优化方法主要有：梯度下降法、牛顿法、拟牛顿法、ADAM等等。

梯度下降法：一阶导数为0的点，不断下降寻找极小值



4. 机器学习相关概念—模型评估

当损失函数给定时，我们将基于模型训练数据的误差(Training Error)和测试数据的误差(Testing Error)作为模型评估的标准。

测试误差的具体定义为：
$$E_{test} = \frac{1}{N'} \sum_{n=1}^{N'} L(y_n, \hat{f}(x_n))$$

其中， N' 为测试数据数量， $L(y_n, \hat{f}(x_n))$ 是损失函数， y_n 代表真实标签， $\hat{f}(x_n)$ 代表预测标签。

一般来说，若我们模型学习的效果好，则训练误差和测试误差接近一致。

5. 机器学习的开发流程



5. 机器学习的开发流程

数据搜集



数据清洗



特征工程



数据建模



6. 机器学习算法--KNN

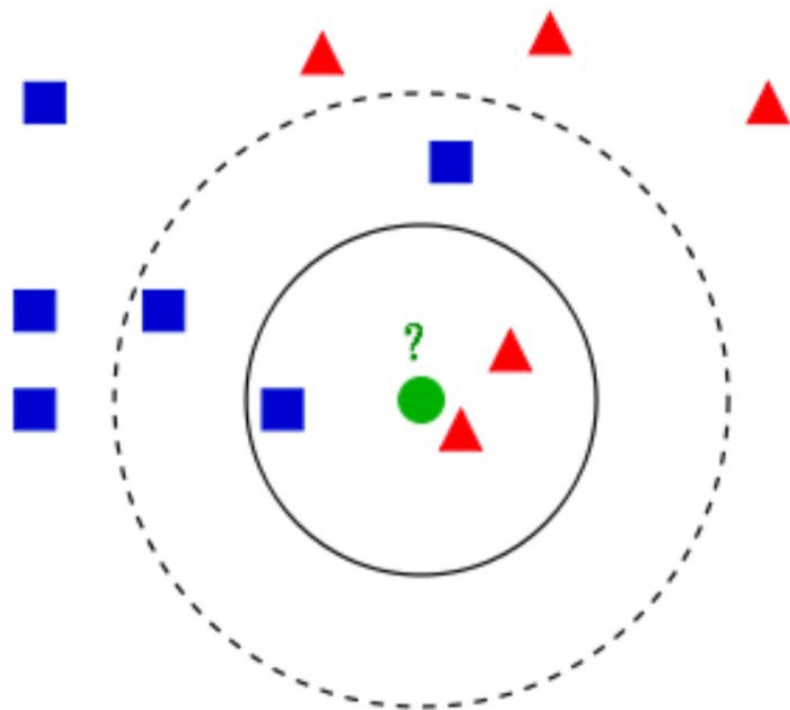
- 6.1 K-近邻（KNN）算法
 - 有监督学习
 - K-NN处理分类问题
 - K-NN实验参数设置
- 6.2 实验任务与要求

6. 机器学习算法--KNN

- 6.1 K-近邻（KNN）算法——有监督学习
- k -NN是有监督的机器学习模型
- 有监督学习的基本步骤：上课——考试
 - 给出带标签的训练数据
 - 用训练数据训练模型至一定程度
 - 用训练好的模型预测不带标签的数据的标签
- 常见的有监督学习问题：
 - 分类问题：预测离散值的问题（如预测明天是否会下雨）
 - 回归问题：预测连续值的问题（如预测明天天气温是多少度）

6. 机器学习算法--KNN

• 6.1 K-近邻 (KNN) 算法—KNN处理分类问题



对每个训练数据的距离进行排序，
最近的3(k)个中有2个是红色三角形，
所以将绿色圆圈分类为红色三角形

半径大小 表示 K值大小

k-nearest neighbours classifier:

$$f(q) = \text{maj} \left(g \left(\Phi_{X,k}(q) \right) \right)$$

其中:

$\Phi_{X,k}(q)$: 返回训练集 X 中距离 q 最近的
 k 个样本

$g(\cdot)$: 返回 (训练) 样本的标签

$\text{maj}(\cdot)$: 返回众数

6. 机器学习算法--KNN

- 6.1 K-近邻（KNN）算法—KNN处理分类问题：例子

给定文本的情感分类任务：

输入：文本

输出：类标签

分类：多数投票原则

Document number	The sentence words	emotion
train 1	I buy an apple phone	happy
train 2	I eat the big apple	happy
train 3	The apple products are too expensive	sadnesss
test 1	My friend has an apple	?

6. 机器学习算法--KNN

• 6.1 K-近邻（KNN）算法—KNN处理分类问题：步骤

Document number	The sentence words	emotion
train 1	I buy an apple phone	happy
train 2	I eat the big apple	happy
train 3	The apple products are too expensive	sadnesss
test 1	My friend has an apple	?

1. 处理成one-hot矩阵

Document number	I	buy	an	apple	...	friend	has	emotion
train 1	1	1	1	1	...	0	0	happy
train 2	1	0	0	1	...	0	0	happy
train 3	0	0	0	1	...	0	0	sadness
test 1	0	0	1	1	...	1	1	?

6. 机器学习算法--KNN

• 6.1 K-近邻（KNN）算法—KNN处理分类问题：步骤

2. 相似度计算：计算test1与每个train的距离

欧氏距离：
$$d(train1, test1) = \sqrt{(1-0)^2 + (1-0)^2 + \dots + (0-1)^2} = \sqrt{6};$$

$$d(train2, test1) = \sqrt{(1-0)^2 + (1-0)^2 + \dots + (0-1)^2} = \sqrt{8};$$

$$d(train3, test1) = \sqrt{(0-0)^2 + (0-0)^2 + \dots + (0-1)^2} = \sqrt{9};$$

（也可以使用其他距离度量方式）

Document number	I	buy	an	apple	...	friend	has	emotion
train 1	1	1	1	1	...	0	0	happy
train 2	1	0	0	1	...	0	0	happy
train 3	0	0	0	1	...	0	0	sadness
test 1	0	0	1	1	...	1	1	?

3. 类别计算：最相似的k个样本之标签的众数

若k=1，test1的标签即为train1的标签happy；

若k=3，test1的标签为train1,train2,train3的标签中数量较多的，即为happy。

6. 机器学习算法--KNN

- 6.1 K-近邻（KNN）算法—KNN参数设置
- 采用不同的距离度量方式（见下一页）
- 通过验证集对参数（k值）进行调优
 - 如果k值取的过大，学习的参考样本更多，会引入更多的噪音，所以可能存在欠拟合的情况；
 - 如果k值取的过小，参考样本少，容易出现过拟合的情况
 - 关于k的经验公式：一般取 $k = \sqrt{N}$ ，N为训练集实例个数，大家可以尝试一下
- 权重归一化

Name	Formula	Explain
Standard score	$X' = \frac{X - \mu}{\sigma}$	μ is the mean and σ is the standard deviation
Feature scaling	$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$	X_{min} is the min value and X_{max} is the max value

6. 机器学习算法--KNN

• 6.1 K-近邻 (KNN) 算法—不同的度量公式

距离公式:

L_p 距离 (所有距离的总公式):

$$L_p(x_i, x_j) = \left\{ \sum_{l=1}^n |x_i^{(l)} - x_j^{(l)}|^p \right\}^{\frac{1}{p}}$$

$p = 1$: 曼哈顿距离;

$p = 2$: 欧氏距离, 最常见。

例 3.1 已知二维空间的 3 个点 $x_1 = (1, 1)^T$, $x_2 = (5, 1)^T$, $x_3 = (4, 4)^T$, 试求在 p 取不同值时, L_p 距离下 x_1 的最近邻点。

解 因为 x_1 和 x_2 只有第一维的值不同, 所以 p 为任何值时, $L_p(x_1, x_2) = 4$ 。而

$$L_1(x_1, x_3) = 6, \quad L_2(x_1, x_3) = 4.24, \quad L_3(x_1, x_3) = 3.78, \quad L_4(x_1, x_3) = 3.57$$

于是得到: p 等于 1 或 2 时, x_2 是 x_1 的最近邻点; p 大于等于 3 时, x_3 是 x_1 的最近邻点。 ■

余弦相似度:

$$\cos \left(\vec{A}, \vec{B} \right) = \frac{\vec{A} \cdot \vec{B}}{|\vec{A}| |\vec{B}|}, \quad \text{其中 } \vec{A} \text{ 和 } \vec{B} \text{ 表示两个文本特征向量;}$$

余弦值作为衡量两个个体间差异的大小的度量

为正且值越大, 表示两个文本差距越小, 为负代表差距越大, 请大家自行脑补两个向量余弦值

6. 机器学习算法--KNN

- 6.1 K-近邻 (KNN) 算法—KNN算法效率
- 假设训练集有 N 个样本，测试集有 M 个样本，每个样本是一个 V 维的向量。
- 如果使用线性搜索的话，那么 k -NN的时间花销就是 $O(N*M*V)$ 。
- 改善：KD树

6. 机器学习算法--KNN

- 6.2 实验任务与要求

- 在给定文本数据集完成文本情感分类训练，在测试集完成测试，计算准确率。
- 要求
 - 文本的特征可以使用TF或TF-IDF，对TF均使用拉普拉斯平滑技巧（可以使用sklearn库提取特征）
 - 利用KNN完成对测试集的分类，并计算准确率
 - 需要提交简要报告+代码
 - 压缩包：学号_姓名_作业编号.zip，如 20331234_张三_实验7.zip。
 - 截止日期：2022.5.19 23:59

Thanks!