《人工智能与Python程序设计》——机器学习模型训练与测试

人工智能与Python程序设计 教研组

人工智能的发展历史：诞生和初期

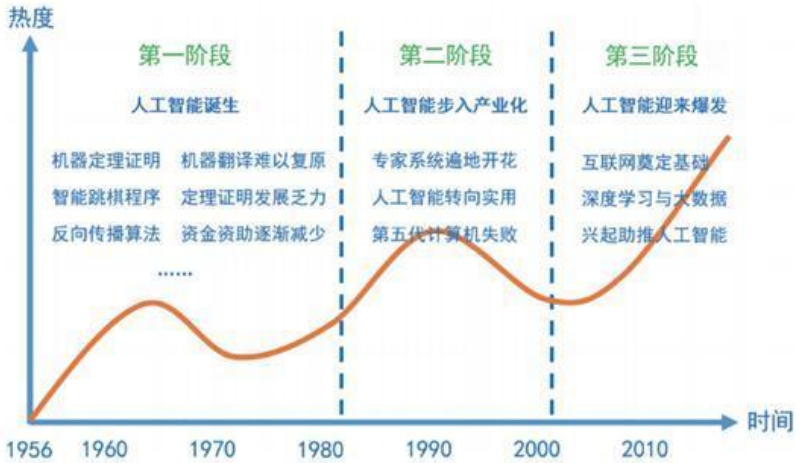
- 人工智能的诞生（20世纪40～50年代）
  - 1950年：**图灵测试**
  - 1956年：第一次**人工智能研讨会**首次提出“人工智能”概念
  - 1957：Frank Rosenblatt提出**感知机**模型
- 人工智能的第一次黄金时代（20世纪50～70年代）
  - 1966年~1972年：首台人工智能机器人Shakey诞生
  - 1966年：世界上第一个聊天机器人ELIZA发布
- 人工智能的第一次低谷（20世纪70～80年代）
  - 1969：马文·明斯基撰写了名为《感知器》一书，发现感知机的局限性
  - 1973: 小丘报告（Lighthill report）否定人工智能作为独立学科存在的必要性

人工智能的发展历史：知识时代

- 人工智能的繁荣期（1980年~1987年）
  - 以**专家系统**为代表：采用（基于规则的）知识表示和知识推理技术来模拟通常由领域**专家**才能解决的复杂问题
  - 1986：**反向传播算法**（Backpropogation algorithm）出现
- 人工智能的第二次冬天（1987年~1993年）
  - 专家系统难以升级扩展，鲁棒性不够，导致高昂的维护成本
  - 神经网络效果未能超过统计的方法，资源消耗大，无法解决大规模问题
  - 日本第五代计算机计划失败
  - “AI之冬”

人工智能的发展历史：机器学习时代

- 人工智能真正的春天（1993年至今）
  - 统计机器学习时代
    - 基于AI技术的搜索引擎获得巨大的成功
  - 深度学习崛起
    - 2006：Geoffrey Hinton发表Science论文
    - 2009：ImageNet数据集发布，2010年举行首次ImageNet竞赛
    - 2016：AlphaGo战胜围棋世界冠军李世石



机器学习训练与测试

提纲

- 机器学习
- 模型训练与性能测试
- 模型评价

机器学习是人工智能的核心

- 维基百科中对机器学习的定义

机器学习有下面几种定义：

- 机器学习是一门人工智能的科学，该领域的主要研究对象是人工智能，特别是如何在经验学习中改善具体算法的性能。
- 机器学习是对能通过经验自动改进的计算机算法的研究。
- 机器学习是用数据或以往的经验，以此优化计算机程序的性能标准。

- 一种经常引用的英文定义：A computer program is said to learn from **experience E** with respect to some class of **tasks T** and performance **measure P**, if its performance at tasks in T, as measured by P, improves with experience E.

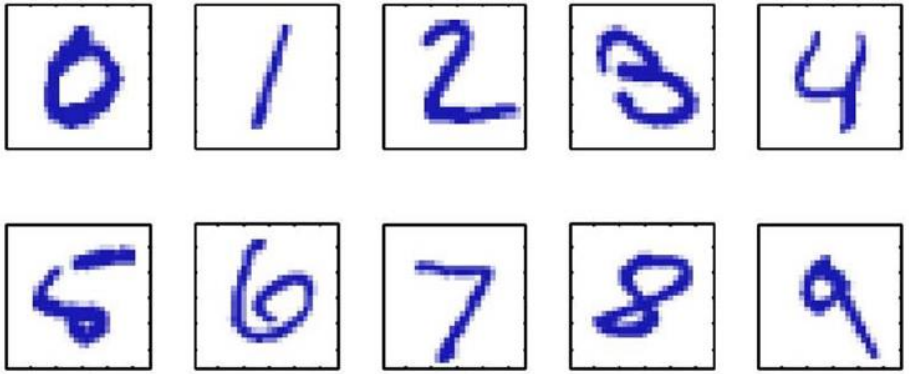
机器学习的定义

- 研究一类算法，使之
  - 在某些**任务**上(task)
  - 通过已有的观测**经验**(数据)(experience)
  - 提升算法**效果**(performance)
- 机器学习任务举例：网页分类

f

类别
个人主页
公司主页
科研机构主页
新闻网站主页
其它网页

机器学习应用举例——手写字符识别

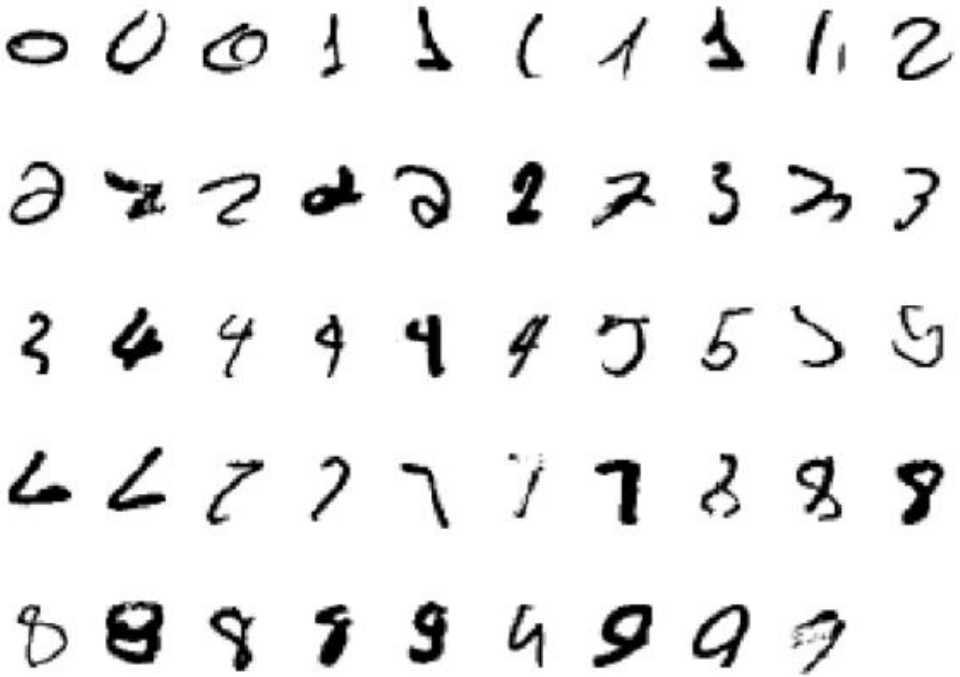


Images are 28 x 28 pixels

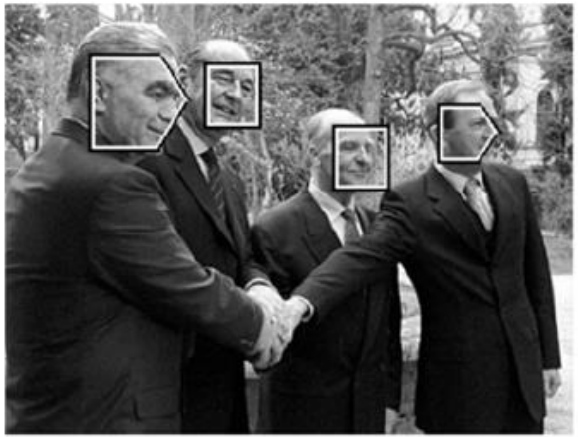
- 输入：每一个图像表示为28*28维的向量$\mathrm{x} \in R^{784}$
- 学习一个分类器$f: \mathbf{x} \mapsto \{0,1,2,3,4,5,6,7,8,9\}$

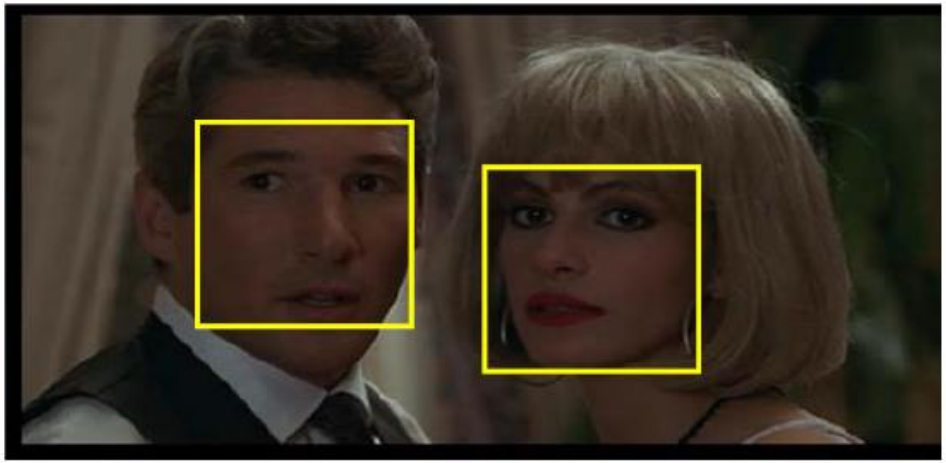
机器学习应用举例——手写字符识别（续）

- 1. 搜集大量的“训练”数据
  - （图像，标签）对
- 2. 训练（学习）分类器f
- 3. 来了新的图像，应用分类器f，得到预测的标签



机器学习应用举例——人脸识别





- 输入：每一个窗口的图像$\mathbf{x}$
- 学习一个分类器$f: \mathbf{x} \mapsto \{$人物A，人物B，人物C$\}$

机器学习应用举例——人脸识别（续）

- 1. 搜集大量的“训练”数据
  - （窗口图像，标签）对
- 2. 训练（学习）分类器f
- 3. 来了新的图像，应用分类器f，得到预测的标签

机器学习应用举例——垃圾信息分类


From: Fannie Fritz <guadalajarae1@aspenrealtors.com>
Subject: **US $ 119.95 Viagra 50mg x 60 pills**
Date: March 31, 2008 7:24:53 AM PDT (CA)

buy now Viagra (Sildenafil) 50mg x 30 pills
http://fullgray.com

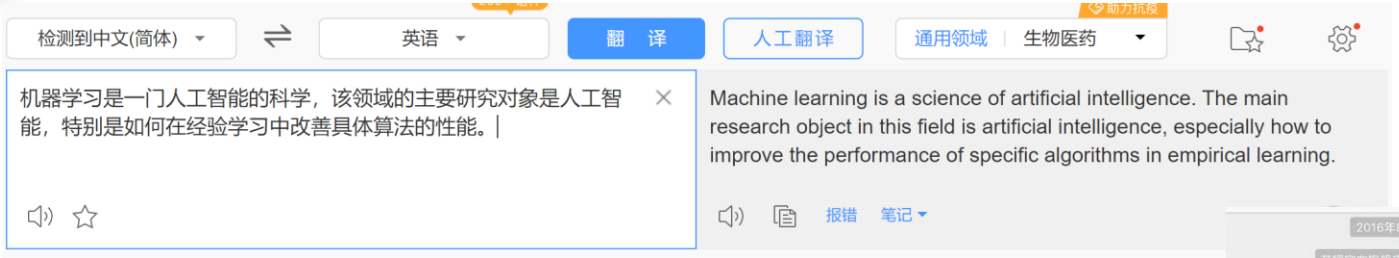
- 输入：每一个邮件，表示为一个向量$\mathbf{x}$
- 学习一个分类器$f: \mathbf{x} \mapsto \{$正常邮件，垃圾邮件$\}$

机器学习应用举例——垃圾信息分类（续）

- 1. 搜集大量的“训练”数据
  - （垃圾信息，标签）对
- 2. 训练（学习）分类器f
- 3. 来了新的信息，应用分类器f，得到预测的标签

标签	短信内容
0	商业秘密的秘密性那是维系其商业价值和垄断地位的前提条件之一
1	南口阿玛施新春第一批限量春装到店啦· 春暖花开淑女裙、冰蓝色公主衫·气质粉小西装、冰丝女王长半裙
0	带给我们大常州一场壮观的视觉盛宴
0	有原因不明的泌尿系统结石等
0	23年从盐城拉回来的麻麻的嫁妆
1	感谢致电杭州萧山全金釜韩国烧烤店，本店位于金城路xxx号。韩式烧烤等，价格实惠、欢迎惠顾【全金釜韩国烧烤店】
0	这款UVe智能杀菌机器人是扫地机的最佳伴侣
1	一次价值xxx元王牌项目；可充值xxx元店内项目卡一张；可以参与V动好生活百分百抽奖机会一次！预约电话：xxxxxxxxxx
0	此类皮肤特别容易招惹粉刺、黑头等
1	(长期诚信在本市作各类资格职称（以及印 /章、牌、 .....等。祥：x x x x x x x x x x 李伟%

更多机器学习应用案例



机器翻译

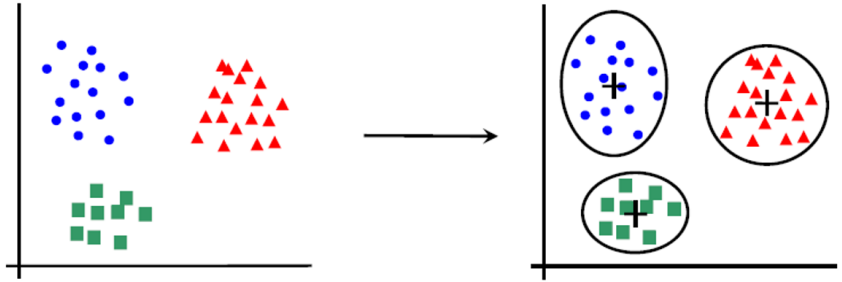
商品推荐

网页搜索

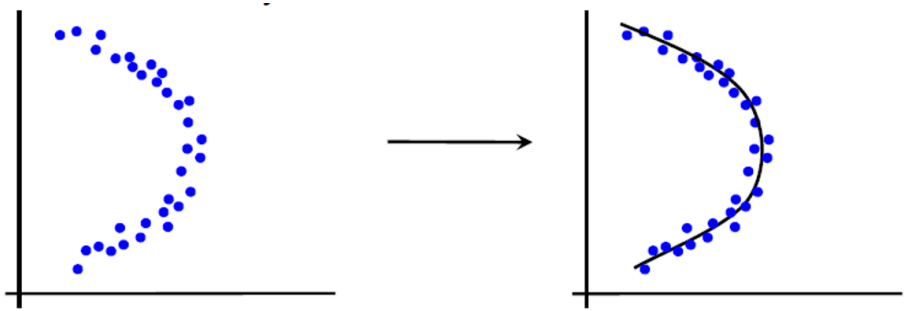
机器人客服

机器学习：无监督学习问题

- 聚类
  - 相似的数据尽可能聚集到一起、不同的数据尽量分离

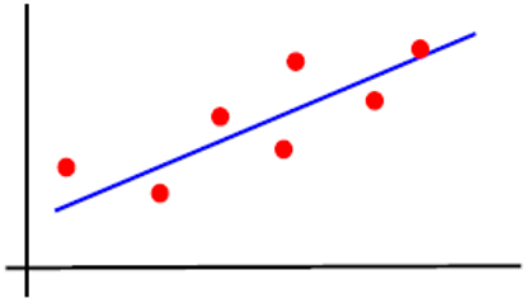


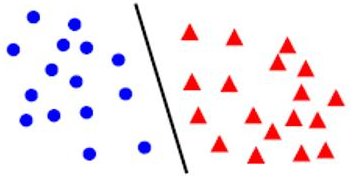
- 维度缩减
  - 预测数据内部关系，从高维表达压缩到低维表达

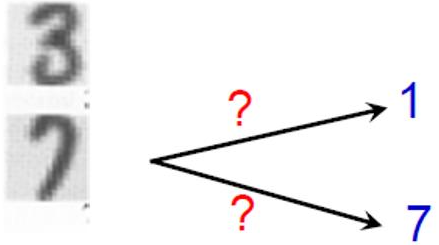


机器学习：监督学习问题

- 回归
  - 预测数值，例如高度、重量、温度等
- 分类
  - 预测类别，例如手写数字类

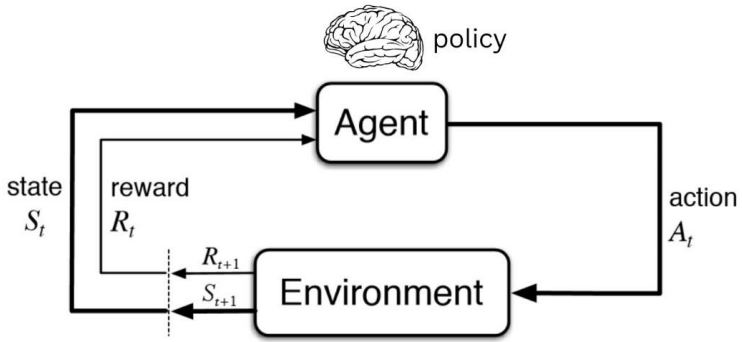






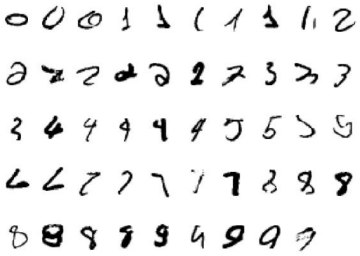
机器学习：强化学习问题

- 学习在**环境**中按照一定策略进行**行动**，以获取最大化的预期收益
  - 学习最优策略
    - 例如：如何下围棋；如何回答人类问题
  - 最大化预期收益
    - 赢棋+1分
    - 正确回答问题+1分
  - 在于环境交互过程中收集经验
    - 数据是动态的
    - 需要平衡探索与利用



监督学习中的分类问题

- 接下来，我们聚焦于监督学习中的**分类问题**
- 研究一类算法，使之
  - 在某些**任务**上（如：把手写数字图片转换为0-9十个字符，即学习一个识别函数：$f: \mathbf{x} \mapsto \{0,1,2,3,4,5,6,7,8,9\}$）
  - 通过已有的观测**经验**(数据)（如：6000个图片-字符对）



  - 提升算法**效果**(performance)（如：在测试集合上的识别准确率）

机器学习流程

提纲

- 机器学习
- **模型训练与性能测试**
- 模型评价

一个典型的分类器学习流程

训练样本

标签

科研机构

个人

公司

科研机构

标签1

测试样本

标签2

标签3

训练/学习

f

预测

评价

分类效果

预测标签1
预测标签2
预测标签3

构建分类器的流程

数据准备

- 数据标注
- 训练集/验证集/测试集分割
- 特征提取

模型训练

- 分类损失函数
- 损失函数优化和参数调优

模型测试

- 性能评价指标

数据准备——数据标注

不带标签的数据
商业秘密的秘密性那是维系其商业价值和垄断地位的前提条件之一
南口阿玛施新春第一批限量春装到店啦· 春暖花开淑女裙、冰蓝色公主衫·气质粉小西装、冰丝女王长半裙
带给我们大常州一场壮观的视觉盛宴
有原因不明的泌尿系统结石等
23年从盐城拉回来的麻麻的嫁妆
感谢致电杭州萧山全金釜韩国烧烤店，本店位于金城路xxx号。韩式烧烤等，价格实惠、欢迎惠顾【全金釜韩国烧烤店】
这款UVe智能杀菌机器人是扫地机的最佳伴侣
一次价值xxx元王牌项目；可充值xxx元店内项目卡一张；可以参与V动好生活百分百抽奖机会一次！预约电话：xxxxxxxxxx
此类皮肤特别容易招惹粉刺、黑头等
(长期诚信在本市作各类资格职称（以及印 /章、牌、 .....等。祥：xxxxxxxxxx 李伟%

（人工）标注

标签 (0：正常短信，1：垃圾短信)
0
1
0
0
0
1
0
1
0
1

- 不带标签的数据往往很容易获取（例如：抓取网页）
- 带（人工）标签的数据集构建需要耗费大量的人力物力

数据分割

- **模型训练**和**模型评价**都需要带标签的数据，这两部分数据不能重合
  - 原因：在实际场景中，模型需要应用于未观测数据
- 操作方法：把整体标注数据随机分成两份
  - 训练集合（如：70%） + 测试集合（如：30%）

标签	短信内容
0	商业秘密的秘密性那是维系其商业价值和垄断地位的前提条件之一
1	南口阿玛施新春第一批限量春装到店啦· 春暖花开淑女裙、冰蓝色公主衫·气质粉小西装、冰丝女王长半裙
0	带给我们大常州一场壮观的视觉盛宴
0	有原因不明的泌尿系统结石等
0	23年从盐城拉回来的麻麻的嫁妆
1	感谢致电杭州萧山全金釜韩国烧烤店，本店位于金城路xxx号。韩式烧烤等，价格实惠、欢迎惠顾【全金釜韩国烧烤店】
0	这款UVe智能杀菌机器人是扫地机的最佳伴侣
1	一次价值xxx元王牌项目；可充值xxx元店内项目卡一张；可以参与V动好生活百分百抽奖机会一次！预约电话：xxxxxxxxxx
0	此类皮肤特别容易招惹粉刺、黑头等
1	(长期诚信在本市作各类资格职称（以及印 /章、牌、 ……等。祥：ｘｘｘｘｘｘｘｘｘｘｘ李伟%

训练集合

测试集合

标签	短信内容
0	商业秘密的秘密性那是维系其商业价值和垄断地位的前提条件之一
1	南口阿玛施新春第一批限量春装到店啦· 春暖花开淑女裙、冰蓝色公主衫·气质粉小西装、冰丝女王长半裙
0	有原因不明的泌尿系统结石等
0	这款UVe智能杀菌机器人是扫地机的最佳伴侣
1	一次价值xxx元王牌项目；可充值xxx元店内项目卡一张；可以参与V动好生活百分百抽奖机会一次！预约电话：xxxxxxxxxx
1	(长期诚信在本市作各类资格职称（以及印 /章、牌、 ……等。祥：ｘｘｘｘｘｘｘｘｘｘｘ李伟%

标签	短信内容
0	带给我们大常州一场壮观的视觉盛宴
0	23年从盐城拉回来的麻麻的嫁妆
1	感谢致电杭州萧山全金釜韩国烧烤店，本店位于金城路xxx号。韩式烧烤等，价格实惠、欢迎惠顾【全金釜韩国烧烤店】
0	此类皮肤特别容易招惹粉刺、黑头等

数据分割（续）

- 训练集合用于确定模型的参数，包括一般参数（机器学习算法确定）和**超参数（研发人员手工调优）**
  - 不同的机器学习模型有不同的超参数，可能的超参：学习步长、正则项权重等
- （可选操作）为了方便调节超参，在训练集合中**随机**划分出**验证集合**

特征抽取

- 实际应用中的数据多种多样：文本、图像、音视频……
- 机器学习算法：大多要求数据为**定长的实数向量（特征向量）**
- 特征抽取：将数据表达为机器学习算法要求的特征向量的过程

标签	短信内容
0	商业秘密的秘密性那是维系其商业价值和垄断地位的前提条件之一
1	南口阿玛施新春第一批限量春装到店啦· 春暖花开淑女裙、冰蓝色公主衫·气质粉小西装、冰丝女王长半裙
0	带给我们大常州一场壮观的视觉盛宴
0	有原因不明的泌尿系统结石等
0	23年从盐城拉回来的麻麻的嫁妆
1	感谢致电杭州萧山全金釜韩国烧烤店，本店位于金城路xxx号。韩式烧烤等，价格实惠、欢迎惠顾【全金釜韩国烧烤店】
0	这款UVe智能杀菌机器人是扫地机的最佳伴侣
1	一次价值xxx元王牌项目；可充值xxx元店内项目卡一张；可以参与V动好生活百分百抽奖机会一次！预约电话：xxxxxxxxxx
0	此类皮肤特别容易招惹粉刺、黑头等
1	(长期诚信在本市作各类资格职称（以及印 /章、牌、 .....等。祥：xxxxxxxxxxx李伟%

特征抽取函数ϕ

标签	特征向量
0	$[1.1, 2, 0, 0, 0, 3, 0, 0, 0, 0, 1]$
1	$[0, 2, 0, 0, 0, 3, 1, 0, 0, 0, 0]$
0	$[1, 2, 0, 0, 0, 3, 0, 0,1\ 0, 0, 1]$
0	$[0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0]$
0	$[0, 1, 0, 0, 0, 3, 0, 0, 0, 0, 0]$
1	$[3.0, 1, 0, 1, 0, 2, 0, 0, 0, 0, 1]$
0	$[0, 2, 0, 0, 0, 1, 0, 0, 0, 0, 0]$
1	$[0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 1]$
0	$[0, 2, 0, 0, 0, 3, 0, 0, 0, 0, 0]$
1	$[0, 2, 0, 0, 0, 2, 0, 0, 0, 0, 1]$

常用的特征举例

- 需不同的数据和分类任务设计不同的特征
  - 例如：文本分类常用的Bag of Words特征
    - 每一个单词对应特征向量的一个维度，出现为1，不出现为0

人工智能/与/Python/程序设计

如何/学习/Python/程序设计

人民大学/高瓴/人工智能/学院

人工智能	与	Python	程序设计	如何	学习	人民大学	高瓴	学院	…
1	1	1	1	0	0	0	0	0	
0	0	1	1	1	1	0	0	0	
1	0	0	0	0	0	1	1	1	

文本数据

对应的特征向量

常用的特征举例

- 需不同的数据和分类任务设计不同的特征
  - 例如：手写字符识别常用的像素特征
    - 每一个像素点对应特征向量的一个维度，数值为灰度值

(0,0)	(0,1)	(0,2)	…	(0,27)	(1,0)	…	…	(27,27)
0. 1	0.2	0.4	0.7	0	1.1			0.01
0.2	0.1	0.1	0.6	0.1	0			0.1

28*28图像数据

对应的特征向量

构建分类器的流程

数据准备

- 数据标注
- 训练集/验证集/测试集分割
- 特征提取

模型训练

- 分类损失函数
- 损失函数优化和参数调优

模型测试

- 性能评价指标
- 交叉验证

机器学习模型训练

- 输入：带人工标签的训练数据集合$D = \{(\mathbf{x}_i, y_i)\}$
  - $\mathbf{x}_i \in R^N$: N维特征向量
  - $y_i \in \{+1, -1\}$: 二值标签
- 输出：预测模型f的参数
  - 例如线性模型：$f(\mathbf{x}; \mathbf{w}, b) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b;\ \hat{y} = \begin{cases} +1, \mathbf{w}^{\mathrm{T}}\mathbf{x} + b > 0 \\ -1, \mathbf{w}^{\mathrm{T}}\mathbf{x} + b \leq 0 \end{cases}$
    - 输出参数为向量$\mathbf{w}$和偏置值b

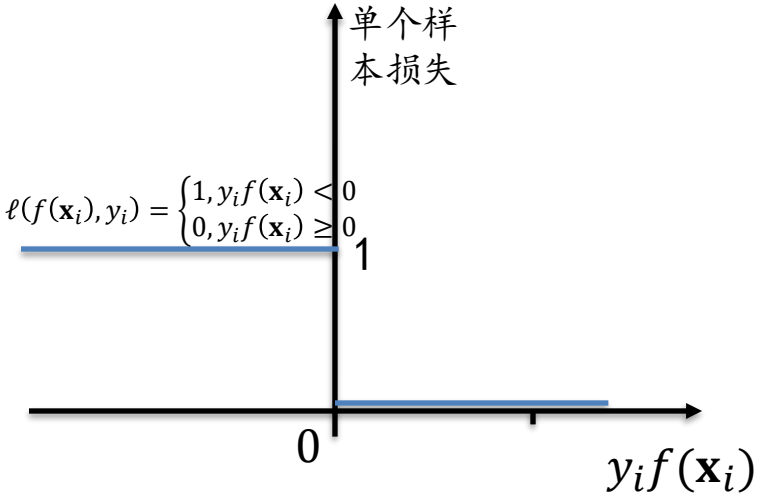
- $\mathbf{w}$和b取什么样的值是最好的？
  - 直觉上：尽量精确地预测训练集合D中所有的数据标签：$\forall i: \hat{y}_i \approx y_{\mathrm{i}}$
  - 这是一个优化问题，最优参数值就是一个函数（损失函数）的最优值点

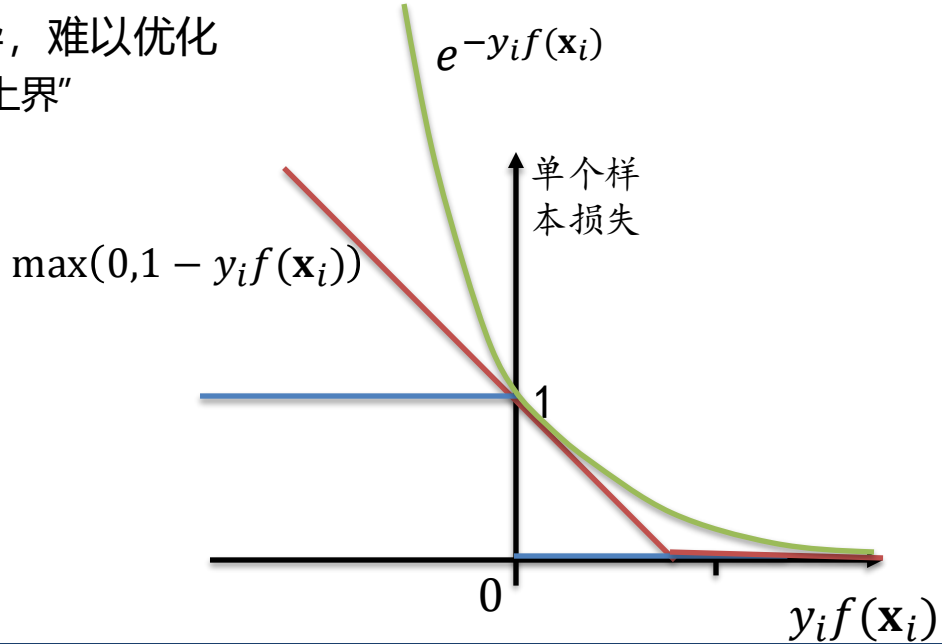
如何构造损失函数?

- 给定:
  - 训练集$D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{M}$中有M个训练样本
  - f中参数$\mathbf{w}, b$的某一具体取值
- 总体的损失 = 单个样本上的损失和（或者平均值）
  - 总体损失：$L(D; f) = \sum_{i=1}^{M} \ell(f(\mathbf{x}_i), y_i)$
  - 单个样本0-1损失：$\ell(f(\mathbf{x}_i), y_i) = \begin{cases} 1, & y_i f(\mathbf{x}_i) < 0 \\ 0, & y_i f(\mathbf{x}_i) \geq 0 \end{cases}$
    - 注意$y_i = 1$或者-1，$y_i f(\mathbf{x}_i) < 0$表示$f(\mathbf{x}_i)$的预测与真实标签y_i**不一致**

如何构造损失函数？

- 总体损失：$L(D;f)=\sum_{i=1}^{M}\ell(f(\mathbf{x}_i),y_i)$
- 单个样本0-1损失：$\ell(f(\mathbf{x}_i),y_i)=\begin{cases}1, & y_i f(\mathbf{x}_i)<0\\ 0, & y_i f(\mathbf{x}_i)\geq 0\end{cases}$
- 但是，上述损失函数不是凸函数不可导，难以优化
  - 转而优化上述0-1损失的平滑、凸的“上界”





优化损失函数

$$L(D;\mathbf{w},b)=\sum_{i=1}^{M}\ell(f(\mathbf{x}_i;\mathbf{w},b),y_i)$$

- 机器学习算法可以归结为最优化损失函数

$$(\mathbf{w}^*,b^*)=\mathrm{argmin}_{\mathbf{w},b}\,L(D;\mathbf{w},b)$$

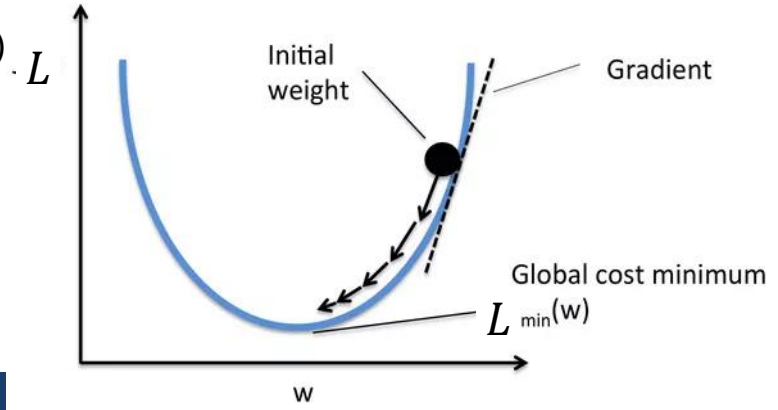
- 常选用的优化算法
  - 梯度下降（Gradient Descent, GD）
  - 随机梯度下降（Stochastic Gradient Descent, SGD）
  - ……

- 选择不同的损失函数和不同的优化算法
  →不同的机器学习算法

提问：线性回归和逻辑斯蒂回归的损失函数分别是什么？

$$L(\boldsymbol{w})=\frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_i\boldsymbol{w}-y_i)^2$$

$$L(\boldsymbol{w})=-\frac{1}{N}\sum_{i=1}^{N}[y_i\ln p_i+(1-y_i)\ln(1-p_i)]$$



构建分类器的流程

数据准备

- 数据标注
- 训练集/验证集/测试集分割
- 特征提取

模型训练

- 分类损失函数
- 损失函数优化和参数调优

模型测试

- 性能评价指标

模型测试的目标

- 评测一个模型在实际应用环境的性能
  - 评估不同机器学习模型、不同参数设置的优劣
  - 在线应用模型前对预测精度进行估计
- 评测方式
  - 在线评测：将模型上线，让真正的用户使用并搜集数据进行评测
    - 如：A/B testing
    - 代价高，数据不能复用
    - 工业界产品上线前的最后一步
  - 离线评测：利用提前标注好的数据（测试集合）进行评测
    - 可以计算不同的评价指标
    - 数据可以重复使用，同时对多种不同的模型进行评价
    - 机器学习研究常用的评测方式

二值分类问题的评价指标

- 离线评价
  - 给定一个**测试集合**$D_t = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$
  - 给定一个待测试的模型$f(\mathbf{x})$
  - 对比人工标签y_i和对应的预测值$\mathrm{sgn}(f(\mathbf{x}_i))$

标注标签 y_i
-1
1
-1
-1
-1
1
-1
1
-1
1

对比

预测标签 $\mathrm{sgn}(f(x_i))$
+1
1
-1
+1
-1
-1
-1
1
-1
1

	预测为正样本	预测为负样本
标注为正样本	TP (true positive)	FN (false negative)
标注为负样本	FP (false positive)	TN (true negative)

混淆矩阵

基于混淆矩阵的评价指标

- 最容易想到的指标：预测正确率Accuracy
  - $\text{Accuracy}=\dfrac{\text{正确预测的样本数}}{\text{总样本数}}=\dfrac{TP+TN}{TP+FN+FP+TN}$

	预测为正样本	预测为负样本
标注为正样本	TP (true positive)	FN (false negative)
标注为负样本	FP (false positive)	TN (true negative)

标注标签 y_i	预测标签 $\text{sgn}(f(x_i))$
-1	+1
1	1
-1	-1
-1	+1
-1	-1
1	-1
-1	-1
1	1
-1	-1
1	1

对比

$$\text{Accuracy}=\frac{10-3}{10}=0.7$$

Accuracy潜在的问题

	预测为正样本	预测为负样本	
标注为正样本	0	10	10
标注为负样本	0	990	990

$$\text{Accuracy}=\frac{0+990}{0+10+0+990}=0.99$$

	预测为正样本	预测为负样本	
标注为正样本	490	5	495
标注为负样本	5	500	505

$$\text{Accuracy}=\frac{490+500}{490+5+5+500}=0.99$$

提问：两个模型的Accuracy相等（0.99），它们真实的性能也是一样的吗?

Accuracy潜在的问题——正负样本不平衡

	预测为正样本	预测为负样本	
标注为正样本	0	10	10
标注为负样本	0	990	990

$$\text{Accuracy}=\frac{0+990}{0+10+0+990}=0.99$$

分类器做出了毫无意义的判断：所有输入样本都是负例。
Accuracy仍然认为分类性能很不错！

	预测为正样本	预测为负样本	
标注为正样本	490	5	495
标注为负样本	5	500	505

$$\text{Accuracy}=\frac{490+500}{490+5+5+500}=0.99$$

如何避免上述错误？

- 改用更加合理的评价指标：Precision(精确率)/Recall(召回率)
  - 更加关注于对正样本的预测
  - 区分了两类错误

$$\text{Precision}=\frac{\text{正确预测的正样本数}}{\text{预测为正例的样本数}}=\frac{TP}{TP+FP}$$

$$\text{Recall}=\frac{\text{正确预测的正样本数}}{\text{标注的正样本数}}=\frac{TP}{TP+FN}$$

	预测为正样本	预测为负样本
标注为正样本	TP (true positive)	FN (false negative)
标注为负样本	FP (false positive)	TN (true negative)

Precision/Recall的计算

$$\text{Precision}=\frac{\text{正确预测的正样本数}}{\text{预测为正例的样本数}}=\frac{TP}{TP+FP},\ \text{Recall}=\frac{\text{正确预测的正样本数}}{\text{标注的正样本数}}=\frac{TP}{TP+FN}$$

标注标签 y_i	预测标签 $\text{sgn}(f(x_i))$
-1	+1
1	1
-1	-1
-1	+1
-1	-1
1	-1
-1	-1
1	1
-1	-1
1	1

对比

	预测为正样本	预测为负样本
标注为正样本	TP =3	FN = 1
标注为负样本	FP =2	TN = 4

$$\text{Precision}=\frac{3}{3+2}=0.6$$

$$\text{Recall}=\frac{3}{3+1}=0.75$$

F1：Precision和Recall的平均数

- F1值为Precision和Recall值的**调和平均数**
  - $F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{1}{\frac{1}{2}\left(\frac{1}{Precision} + \frac{1}{Recall}\right)}$

- 类比：以速度P上山，以速度R下山，平均速度F是多少？
  - $F = \frac{\text{上下山总里程}}{\text{上下山总时间}} = \frac{2S}{\frac{S}{P} + \frac{S}{R}} = \frac{2PR}{P+R}$
  - 平均速度更多取决于较慢的上山速度

- ➔ F1更加接近于Precision和Recall中较小的那个数字 (soft minimum)

F1值的计算

$$\text{Precision}=\frac{\text{正确预测的正样本数}}{\text{预测为正例的样本数}}=\frac{TP}{TP+FP},\ \text{Recall}=\frac{\text{正确预测的正样本数}}{\text{标注的正样本数}}=\frac{TP}{TP+FN}$$

标注标签 y_i	预测标签 $\text{sgn}(f(x_i))$
-1	+1
1	1
-1	-1
-1	+1
-1	-1
1	-1
-1	-1
1	1
-1	-1
1	1

对比

	预测为正样本	预测为负样本
标注为正样本	TP =3	FN = 1
标注为负样本	FP =2	TN = 4

$$\text{Precision}=\frac{3}{3+2}=0.6$$

$$\text{Recall}=\frac{3}{3+1}=0.75$$

$$F1=\frac{2\times 0.6\times 0.75}{0.6+0.75}=0.6667$$

总结：构建分类器的流程

数据准备

- 数据标注
- 训练集/验证集/测试集分割
- 特征提取

模型训练

- 分类损失函数
- 损失函数优化和参数调优

模型测试

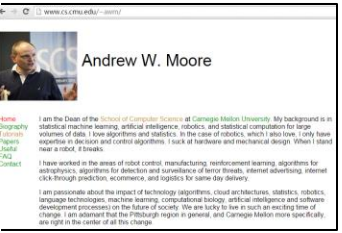
- 性能评价指标

总结：构建分类器的流程

训练样本

标签

1



个人

公司

科研机构

2

f

3

预测标签1
预测标签2
预测标签3

测试样本

人工标签1

1

3

分类效果

练习：实现二分类问题评价指标

```python
import numpy as np

# Evaluation类
class EvaluationMetrics(object):

    def __confusion_matrix(self, y_true, y_pred):...

    def accuracy(self, y_true, y_pred):
        ...

    def precision(self, y_true, y_pred):
        ...

    def recall(self, y_true, y_pred):
        ...

    def f1(self, y_true, y_pred):
        ...

# 测试样例1
y_t = np.asarray([0, 1, 1, 0, 1, 1, 0, 0, 0, 0])
y_p = np.asarray([1, 1, 0, 1, 1, 1, 0, 0, 0, 0])
metric = EvaluationMetrics()
```

- y_true和y_pred均为numpy向量，分别代表标注值和预测值
  - 1：正例
  - 0：负例
- 实现函数
  - __confusion_matrix
  - accuracy
  - precision
  - recall
  - f1
- 注意
  - 对y_true和y_pred的大小和数值做合法性检查
  - 对边界情况进行判断

谢谢！