数学课程

课前复习

- 课件用法
  - Markdown
  - Jupyter notebook
  - 编辑器：VSCode
    - Python 扩展
    - Markdown 扩展
- 课上教材
  - 李航，统计学习方法。https://item.jd.com/12522197.html
    - 课程前半段，小班同学
    - 机器学习基础理论
    - 传统的算法
    - 面向人群：统计，互联网公司从业人员
    - 传统机器学习算法可以完成所有任务
    - 只不过传统机器学习算法对特征工程要求比较高
  - Goodfellow，深度学习。https://item.jd.com/12128543.html
    - 课程后半段
    - 主要分为：CV（计算机视觉），NLP（自然语言处理）
    - 面向：高新技术企业
    - 所谓的AI（人工智能）有60%指的是深度学习->深度神经网络
  - 阅读、补充材料，有条件有时间可以阅读。
    - Python学习手册，https://item.jd.com/12452929.html
      - 面向对Python编程不熟悉的人员
      - 课程主要使用的是Python 、
      - 或者看Python文档
      - Python并不是机器学习专属语言
      - 机器学习并不依赖于Python
        - 现在机器学习主要是高性能计算
        - Python大多用于现有库的调用，但是高性能的部分需要C/C++
        - 如果有条件**学习一门高性能的编程语言**，这需要后期补充，面试可能不需要这么深入
    - 算法导论，https://item.jd.com/11311645.html
      - 理解图算法、贪心算法、动态规划等一系列算法。

        - 如果想短期提升算法能力，可以刷题。
        - 面试经常问到。
        - 所谓的AI也包括传统算法。
    - 信号、图像处理
        - 用于机器学习的特征工程
        - 面试CV方向可能会问到
        - 信号处理
        - 为深度学习打基础。
        - 代码：第一章-索贝尔滤波.py
        - 代码：第一章-声音.py
    - 最优化理论，https://item.jd.com/12402926.html
        - 买任何一本类似的书都可以
    - 现代几何学

思考:什么是机器学习

- 机器学习是从数据中发现规律，并指导生产的学科。
- 机器学习是让机器学习人类知识的过程。
- 机器学习=数据（样本）+模型+优化
    - 数据需要人工采集并标注
    - 模型是对数据真实分布的简化，以及人经验的体现。
    - 优化是求解模型中的参数
- 假设一系列数据点 (x, y)，求x和y的关系?
    - 首先获取数据，也是获取样本。
    - 假设xy的关系是（建模）：y=ax+b
    - 优化过程：使用MSE作为优化函数，进行迭代，称为最小二乘。

概率与统计

- 离散和连续类型随机变量
    - 有监督机器学习算法分为：分类算法和回归算法，分类算法就是离散类型的，连续类型的是回归算法。
    - 例子：两枚硬币，分别抛了1000次。A硬币正面出现了800次，B硬币正面出现了200次（获取的样本，有2000个）。此时抛硬币出现了反面，思考抛的是A还是B。算法叫做贝叶斯算法（简化的）
        - 代码：第二章-抛硬币统计.py
        - 机器学习可以从概率角度理解
        - 机器学习算法并不难。

- 统计概念
  - 期望：
    - $\mathbb{E}(x)=\int_{-\infty}^{+\infty}xp(x)dx$(连续类型)
    - $\mathbb{E}(x)=\sum_i x_ip(x_i)=\mu(x)$(离散类型)
    - 概率应当是研究样本所得
  - 统计量：样本均值
    - $\bar{x}=\sum_i x_i/N$
    - x_i是样本。
  - 方差
    - $Var(x)=\int_{-\infty}^{+\infty}(x-\mu)^2p(x)dx$
    - 方差无偏估计 $\frac{\sum_i(x_i-\bar{x})^2}{N-1}$
    - 标准差：$\sigma=\sqrt{Var}$
- 常见分布
  - 高斯分布，正态分布：$N(\mu,\sigma)=\frac{1}{\sqrt{2\pi}\sigma}e^{-(x-\mu)^2/(2\sigma^2)}$
  - 实例分析：
    - 代码：第二章-高斯分布随机数分类问题.py
    - 思路s1：统计x0附近，哪一类数据点比较多。
      - 统计过程会遍历样本点，速度很慢。
    - 思路2：（机器学习思路），假设样本符合高斯分布，求解高斯分布分布参数
      - 所谓参数就是样本均值和方差。
      - 通过建模简化了计算过程
      - 机器学习建模均是对数据分布的简化
      - 并不是所有样本均符合假设，但是假设能一定程度完成分类任务。
      - 机器学习是一个近似过程。
- 协方差
  - 代码:第二章-协方差与线性相关系数.py
  - 假设x,y的关系是y=ax+b（线性关系），意味着xy数据存在冗余。
  - 是否存在线性关系，通过协方差判断。
- 信息熵
  - 衡量系统的混乱程度
  - 仅需要记住数学形式即可
  - e.g.:抛硬币，正0.5反0.5
  - 在知道均值和方差的前提下，高斯分布是最大熵分布。
  - 求高斯分布的熵

函数和优化

- 函数的展开

    - $f(x_0+\Delta x)=f(x_0)+f'(x_0)dx+\frac{1}{n!}f^{(n)}(x)\Delta x^n$
    - $f(x_0+\Delta x)\approx f(x_0)+f'(x_0)\Delta x$
    - $f(x_0+\Delta x)-f(x_0)=f'(x_0)\Delta x$
    - 如果想让f(x)不断减小，只需要让$\Delta x=-f'(x)$，迭代称为梯度下降法。
    - 带入后变为了
        - $f(x_0+\Delta x)-f(x_0)=-(f'(x_0))^2\leq 0$
        - 所以$f(x_0+\Delta x)\leq f(x_0)$
        - 所以迭代可以逐渐减少。
    - f(x)=x^2+2x 求函数极小值。e.g.
        - 代码：第三章-梯度下降法-1d.py
        - 代码：第三章-梯度下降法-1d.py
- 梯度下降法
    - 给定函数初始值x_0
    - 给定学习率 η
        - 如果学习率过大，会导致迭代发散。
        - 如果学习率选择过小，会导致收敛缓慢。
    - 计算梯度（导数）$g=\nabla f=(\frac{\partial f}{\partial x_i})$
    - 执行$x_t=x_{t-1}-\eta g$
    - 直到最大迭代次数，或结果不再变化。
    - eg. 求函数f(x1, x2)=x1$^{2+2x1+x2}$2+x2的极小值。
    - 现有课程优化主要是基于梯度的。
- 第一个机器学习问题：假设一系列数据点 (x,d)，求x和d的关系?
    - 首先假设关系是$y=ax+b$
    - 定义y和d相似度评价函数，$L=\sum_i(y_i-d_i)^2/N$，此时称为最小二乘法。
    - 目标求解合适的ab使得loss最小
    - $g=(\frac{\partial L}{\partial a},\frac{\partial L}{\partial b})$
    - 执行梯度下降法
    - 批学习
        - 每次从样本中选择一部分进行训练
    - 问题：如何提升精度?
        - 假设关系为：$y=ax^2+bx+c$
- 名词
    - 梯度下降法：求解函数梯度并进行优化
    - 最小二乘法：损失函数是 (y-d)^2，y模型输出，d标签 ，是回归问题。
    - 批学习：每次输入多个样本进行训练。
        - 批尺寸越大理论上最终结果精度越高
        - 但是速度越慢。

线性代数

- 矩阵和其运算
  - 当成代数工具
  - 简化公式的书写
  - 加减乘除均是代数计算
  - 矩阵求导参考文献：The Matrix Cookbook
  - 概念
    - 对角矩阵：仅在对角线元素有值
    - 方阵：行列相等
    - 逆矩阵：AB=I，此时AB互为逆矩阵
    - 对称矩阵：$a_{ij} = a_{ji}, A^T = A$
- 举例：鸢尾花数据
  - 有150个样本，每个样本有四个属性
  - 所以数据X是一个矩阵，[150, 4]
  - 有三个类，此时标签长度为3
  - 建立模型就是Y=XW+b，也是线性模型
- 仿射变换和特征值分解
  - 空间如何进行仿射变换
    - 代码：第四章-仿射变换.py
  - Y=XW+b为线性变换，也是仿射变换。
  - 仿射变换可以用于图像的旋转
  - 特征值分解：$A = E\Lambda E^{-1}$
    - 对于方阵而言的
    - 实对称矩阵特征值分解：$A = E\Lambda E^{\top}$
    - 分解后E代表空间的旋转，而Λ代表空间的拉伸
  - SVD分解
    - 对于任意实数矩阵而言的
    - 首先假设矩阵B可以分解为$B = MQN, M^TM = I$
    - 取$A = B^TB$，此时A是对称方阵
    - 对A进行特征值分解：$A = B^TB = N^TQ^TM^TMQN = N^T = N^TQ^2N = E\Lambda E^T$
    - 所以$N = E^T$，很容易得M
    - 代码：第四章-SVD图像压缩.py
  - 了解内容
    - 矩阵QR分解
    - 酉矩阵
    - 广义逆

API参考

- Numpy：矩阵运算库
  - np.random.normal:产生正态分布随机数
    - 均匀分布通过BoxMuller转换为正态分布。
  - np.random.randint:等概率的产生整形数字
  - np.random.random:产生均匀分布随机数
    - 算法：线性同余发生器
  - np.mean:统计均值
  - np.std:统计标准差
  - 假设x是numpy矩阵
    - x[idx]相当于取以idx为索引的数字。
  - linalg.svd：奇异值分解
  - np.savez()数据保存
  - np.load() 数据加载
    - 参考官方文档
- Matplotlib：绘图库
  - scatter:绘制散点图
  - hist:绘制柱状图
  - plot
- scipy：优化和数据处理库，numpy拓展
- sympy：符号计算库
- LaTeX语法
  - $$公式
  - $\beta\gamma\frac{b}{a}\mathbf{B},\vec{a}$
- Mathematica：计算机代数系统，符号计算用的软件，付费软件。

重点

- 概率论
  - 均值、方差等概念
- 函数和优化
  - 梯度迭代算法
- 线性代数
  - 矩阵的运算
- 如果不懂
  - 可以选择性跳过。

- 后面学习完后，再复习。
- 参考书籍《深度学习》前四章。