人工智能实验

2022年春季

k-近邻算法

目录

- **1 *k*–近邻（*k*-NN）算法**
  - **有监督学习**
  - *k*-NN处理分类问题
  - *k*-NN参数设置
- 2 实验任务与要求

k-NN与有监督学习

- k-NN是有监督的机器学习模型

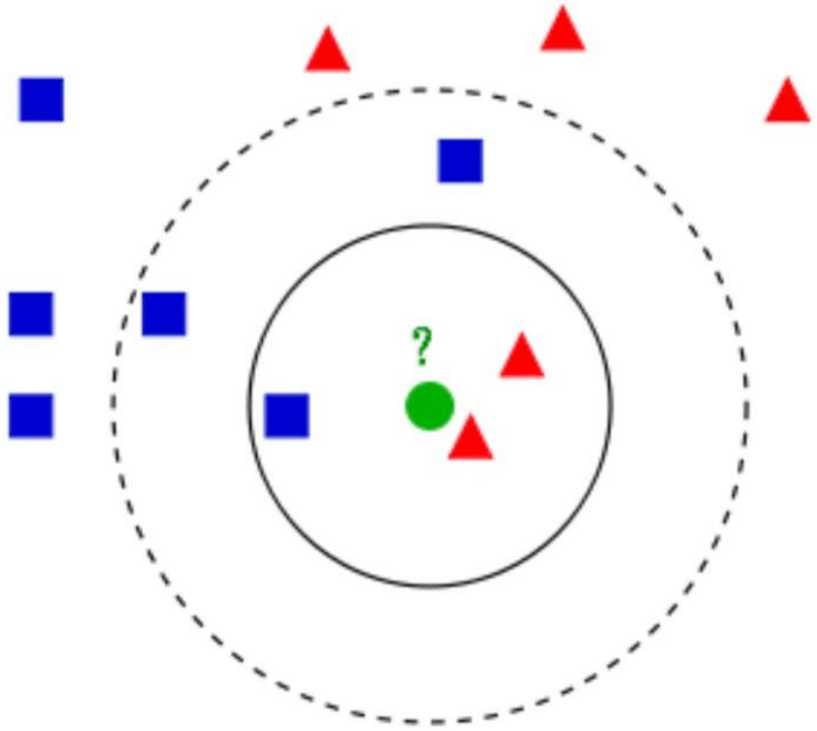
- 有监督学习的基本步骤：上课—考试
  - 给出带标签的训练数据
  - 用训练数据训练模型至一定程度
  - 用训练好的模型预测不带标签的数据的标签

- 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（如预测明天气温是多少度）

目录

- **1 *k*–近邻（*k*-NN）算法**
  - 有监督学习
  - ***k*-NN处理分类问题**
  - *k*-NN参数设置
- 3 实验任务与要求

k-NN处理分类问题



半径大小 表示 K值大小

- k-nearest neighbours **classifier**:

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

- 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
  - $maj(\cdot)$: 返回众数

k-NN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	?

k-NN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	?

k-NN处理分类问题：步骤

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	?

2. 相似度计算：计算test1与每个train的距离

- 欧氏距离：

$$d(train1, test1) = \sqrt{(1-0)^2 + (1-0)^2 + \cdots + (0-1)^2} = \sqrt{6};$$

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

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

（也可以使用其他距离度量方式）

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。

目录

- **1 *k*–近邻（*k*-NN）算法**
  - 有监督学习
  - *k*-NN处理分类问题
  - ***k*-NN参数设置**
- 2 实验任务与要求

k-NN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

不同距离度量方式

- 距离公式：

  L_p距离(所有距离的总公式)：

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

  - $p=1$：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 $x_1=(1,1)^{\mathrm{T}}$, $x_2=(5,1)^{\mathrm{T}}$, $x_3=(4,4)^{\mathrm{T}}$，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：

  $\cos\left(\vec{A},\vec{B}\right)=\frac{\vec{A}\cdot\vec{B}}{\left|\vec{A}\right|\left|\vec{B}\right|}$，其中$\vec{A}$和$\vec{B}$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

k-NN算法的效率

- 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)。

目录

- 1 k–近邻（k-NN）算法
  - 有监督学习
  - *k*-NN处理分类问题
  - k-NN参数设置
- **2 实验任务与要求**

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k-NN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
  - 加分项：
    - 距离度量
    - 算法效率优化