模式识别第二次作业

201300086 史浩男 人工智能学院

K-均值聚类. 聚类是无监督学习中经典的例子, 而 K-均值聚类 (K-means clustering) 则可能是聚类任务中使用最广泛的问题.
给定一组样本 $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\}$, 其中 $\boldsymbol{x}_j \in \mathbb{R}^d$ $(1 \leqslant j \leqslant M)$, K-均值聚类问题试图将

㊀ 如果你无法获得 Matlab 软件, 也可以使用 Octave 接口作为替换.

这 M 个样本分成 K 组, 每组的样本彼此相似 (即属于相同组的一对样本之间的距离很小).
令 γ_{ij} $(1 \leqslant i \leqslant K, 1 \leqslant j \leqslant M)$ 表示组指示器, 即如果 $\boldsymbol{x}_j$ 被分到了第 i 组, 则 $\gamma_{ij} = 1$; 否则 $\gamma_{ij} = 0$. 请注意, 对于任意 $1 \leqslant j \leqslant M$, 有

$$\sum_{i=1}^{K} \gamma_{ij} = 1.$$

令 $\boldsymbol{\mu}_i \in \mathbb{R}^d$ $(1 \leqslant i \leqslant K)$ 为第 i 组的代表.

(a) 证明下述优化公式对 K–均值的目标进行了形式化.

$$\underset{\gamma_{ij}, \boldsymbol{\mu}_i}{\arg\min} \sum_{i=1}^{K} \sum_{j=1}^{M} \gamma_{ij} \|\boldsymbol{x}_j - \boldsymbol{\mu}_i\|^2 . \tag{3.5}$$

(b) 发现 γ_{ij} 和 $\boldsymbol{\mu}_i$ 的值, 使其为公式 (3.5) 的全局解, 这个问题是 NP 难的. 在实际中, 通常使用 Lloyd 算法来确定 K- 均值的解. 在对 γ_{ij} 和 $\boldsymbol{\mu}_i$ 进行初始化之后, 这一方法会在以下两步之间进行反复迭代, 直至收敛:

i. 固定 $\boldsymbol{\mu}_i$ (对于所有的 $1 \leqslant i \leqslant K$), 找到 γ_{ij} 使得损失函数最小化. 这一步将所有样本重新分配到各组;

ii. 固定 γ_{ij} (对于所有的 $1 \leqslant i \leqslant K, 1 \leqslant j \leqslant M$), 找到 $\boldsymbol{\mu}_i$ 使得损失函数最小化. 这一步重新计算了每组的代表.

分别推导上述两步中 γ_{ij} 和 $\boldsymbol{\mu}_i$ 的更新规则. 当 $\boldsymbol{\mu}_i$ (对于所有的 $1 \leqslant i \leqslant K$) 被固定时, 你应该找到 γ_{ij} 使得公式 (3.5) 最小化的值, 反之亦然.

(c) 证明 Lloyd 算法能够收敛.

一、（3.2）K-means

(a)公式抽象

不妨假设μ_i为聚类中心，我们的目标就是把数据点根据到中心距离分类，形式化目标函数如下：

$$D=\sum_{\mathrm{j}=1}^{\mathrm{M}}\sum_{\mathrm{i}=1}^{\mathrm{K}}\gamma_{\mathrm{ij}}\|\boldsymbol{x}_{\mathrm{j}}-\boldsymbol{\mu}_{\mathrm{i}}\|^2 \tag{1}$$

因此只需要最小化这个目标函数，就能求出对应的γ_{ij}和$\boldsymbol{\mu}_{\mathrm{i}}$，从而完成聚类

$$\underset{\gamma_{\mathrm{ij}},\boldsymbol{\mu}_{\mathrm{i}}}{\arg\min}\sum_{\mathrm{i}=1}^{\mathrm{K}}\sum_{\mathrm{j}=1}^{\mathrm{M}}\gamma_{\mathrm{ij}}\|\boldsymbol{x}_{\mathrm{j}}-\boldsymbol{\mu}_{\mathrm{i}}\|^2 \tag{2}$$

(b)迭代规则

固定μ_{i}：

此时只需找到每个数据点距离最近的中心是哪个，所有中心都是不变的，因此表达式为：

$$\gamma_{ij}=\begin{cases}1, & \text{if } i=\arg\min_i\|\boldsymbol{x}_j-\boldsymbol{\mu}_i\|^2\\ 0, & \text{otherwise}\end{cases} \tag{3}$$

固定γ_{ij}：

此时所有类别包含哪些点已经确定，只需在每个类中找到类中点最近的中心位置，可以解出：

$$\boldsymbol{\mu}_i=\frac{\sum_{j=1}^{M}\gamma_{ij}\boldsymbol{x}_j}{\sum_{j=1}^{M}\gamma_{ij}} \tag{4}$$

(c)证明收敛

只需证明，（b）中的两个迭代步骤，都会使目标函数D不增（单调递减有下界的函数必然收敛。而如果两个步骤都不增不减，说明已经收敛。如果至少有一个是递减的，那么满足条件单调递减有下界，一定会最终收敛）

固定μ_{i}：

$$\begin{aligned}D'-D&=\sum_{i=1}^{K}\sum_{j=1}^{M}\gamma'_{ij}\|\boldsymbol{x}_j-\boldsymbol{\mu}_i\|^2-\sum_{i=1}^{K}\sum_{j=1}^{M}\gamma_{ij}\|\boldsymbol{x}_j-\boldsymbol{\mu}_i\|^2\\&=\sum_{i=1}^{K}\sum_{j=1}^{M}\left(\gamma'_{ij}-\gamma_{ij}\right)\|\boldsymbol{x}_j-\boldsymbol{\mu}_i\|^2\\&=\sum_{i=1}^{K}\left(\|\boldsymbol{x}_j-\boldsymbol{\mu}'_i\|^2-\|\boldsymbol{x}_j-\boldsymbol{\mu}_i\|^2\right)\\&\leq 0\end{aligned} \tag{5}$$

固定γ_{ij}：

$$
\begin{aligned}
D_j'-D_j &= \sum_{i=1}^{K}\gamma_{ij}\|\boldsymbol{x}_j-\boldsymbol{\mu}_i'\|^2-\sum_{i=1}^{K}\gamma_{ij}\|\boldsymbol{x}_j-\boldsymbol{\mu}_i\|^2 \\
&= \sum_{i=1}^{K}\gamma_{ij}\left(\|\boldsymbol{x}_j-\boldsymbol{\mu}_i'\|^2-\|\boldsymbol{x}_j-\boldsymbol{\mu}_i\|^2\right) \\
&= \sum_{i=1}^{K}\gamma_{ij}\big(\boldsymbol{x}_j-\boldsymbol{\mu}_i'+\boldsymbol{x}_j-\boldsymbol{\mu}_i\big)^{\mathrm{T}}\left(\boldsymbol{x}_j-\boldsymbol{\mu}_i'-\boldsymbol{x}_j+\boldsymbol{\mu}_i\right) \\
&= \sum_{i=1}^{K}\gamma_{ij}\big(\boldsymbol{\mu}_i-\boldsymbol{\mu}_i'\big)^{\mathrm{T}}\left(2\boldsymbol{x}_j-\boldsymbol{\mu}_i'-\boldsymbol{\mu}_i\right) \\
&= \left(\boldsymbol{\mu}_i-\boldsymbol{\mu}_i'\right)^{\mathrm{T}}\left(2\left(\sum_{i=1}^{K}\gamma_{ij}\boldsymbol{x}_j\right)-\left(\sum_{i=1}^{K}\gamma_{ij}\right)\left(\boldsymbol{\mu}_i'+\boldsymbol{\mu}_i\right)\right) \\
&= \left(\sum_{i=1}^{K}\gamma_{ij}\right)\big(\boldsymbol{\mu}_i-\boldsymbol{\mu}_i'\big)^{\mathrm{T}}\left(2\frac{\sum_{i=1}^{K}\gamma_{ij}\boldsymbol{x}_j}{\sum_{i=1}^{K}\gamma_{ij}}-\boldsymbol{\mu}_i'-\boldsymbol{\mu}_i\right) \\
&= \left(\sum_{i=1}^{K}\gamma_{ij}\right)\big(\boldsymbol{\mu}_i-\boldsymbol{\mu}_i'\big)^{\mathrm{T}}\left(2\boldsymbol{\mu}_i'-\boldsymbol{\mu}_i'-\boldsymbol{\mu}_i\right) \\
&\leq 0
\end{aligned}
\tag{6}
$$

因此我们简洁地证明了，两个迭代步骤都使目标函数D不增。

综上，一定会最终收敛

4.2 (线性回归) 考虑一个 n 个样本 $(\boldsymbol{x}_i, y_i)$ $(1 \leqslant i \leqslant n)$ 的集合, 其中 $\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$. 对于任一样本 $(\boldsymbol{x}, y)$, **线性回归** (linear regression) 模型假设

$$y = \boldsymbol{x}^T \boldsymbol{\beta} + \epsilon,$$

其中 ϵ 是一个随机变量, 用来对回归误差进行建模, $\boldsymbol{\beta} \in \mathbb{R}^d$ 是模型的参数. 对于第 i 个样本, 我们有 $\epsilon_i = y_i - \boldsymbol{x}_i^T \boldsymbol{\beta}$.

(a) 使用训练样本、参数 $\boldsymbol{\beta}$ 和平方误差 ($\sum_{i=1}^n \epsilon_i^2$, 它是 MSE 乘以样本个数) 将线性回归任务表示为训练集上的优化问题.

(b) 我们可以将训练样本 $\boldsymbol{x}_i$ 组织成一个 $n \times d$ 的矩阵 X, 其第 i 行是向量 $\boldsymbol{x}_i^T$. 类似地, 我们可以将 y_i 组织成向量 $\boldsymbol{y} \in \mathbb{R}^n$, y_i 在其第 i 行中. 使用 X 和 $\boldsymbol{y}$ 重写 (a) 中的优化问题.

(c) 找到 $\boldsymbol{\beta}$ 的最佳值. 暂时假设 $X^T X$ 是可逆的. 该解被称为**普通线性回归** (ordinary linear regression) 解.

(d) 当维度大于样本个数, 即 $d > n$ 时, $X^T X$ 是否可逆?

(e) 如果我们在线性回归中增加一个带有权衡参数 λ $(\lambda > 0)$ 的正则项

$$\mathcal{R}(\boldsymbol{\beta}) = \boldsymbol{\beta}^T \boldsymbol{\beta},$$

该正则项会带来什么样的影响? 使用该正则项的线性回归被称为**岭回归** (ridge regression), 该正则项是 *Tikhonov* **正则化** (Tikhonov regularization) 的特例.

(f) 使用 X、$\boldsymbol{y}$、$\boldsymbol{\beta}$ 和 λ 写出岭回归中的优化问题. 找到其解.

(g) 当 $X^T X$ 不可逆时, 普通线性回归将遇到困难. 岭回归在这方面有何帮助?

(h) 如果 $\lambda = 0$, 岭回归的解是什么? 如果 $\lambda = \infty$ 呢?

(i) 我们可否将 λ 视为普通参数 (而不是超参数) 来学习得到一个好的 λ 值呢? 也就是说, 通过在训练集上联合优化 λ 和 $\boldsymbol{\beta}$ 来最小化岭回归损失函数.

4.3 (多项式回归) 多项式回归模型 $y = f(x) + \epsilon$ 假设映射 f 是一个多项式. 一个 d 阶多项式具有如下形式:

二、（4.2）LR

(a，b)优化问题与重写

$$\arg\min_{\beta} (y_i - x_i^T \boldsymbol{\beta})^2 \tag{7}$$

矩阵表示重写：可以展开简化

$$\arg\min_{\beta} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^{\mathrm{T}} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \tag{8}$$

(c)求解

$$\frac{\partial E}{\partial \boldsymbol{\beta}} = 2\boldsymbol{X}^{\mathrm{T}} (\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}) = \mathbf{0} \tag{9}$$

由于 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$可逆，解出：

$$\boldsymbol{\beta}^* = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1} \boldsymbol{X}^{\mathrm{T}} \boldsymbol{y} \tag{10}$$

(d)维度大于样本导致不可解

由矩阵的性质可知：$\operatorname{rank}\left(\boldsymbol{X}^{\mathrm{T}} \boldsymbol{X}\right)=\operatorname{rank}(\boldsymbol{X}) \leq n<d$ ，而$\boldsymbol{X}^{\mathrm{T}} \boldsymbol{X}$是一个 $d \times d$的矩阵, 不满秩的矩阵必然不可逆

(e)正则化项作用

正则化项度量了模型复杂度，是用于对抗过拟合的关键手段。正则化表示了对模型的一种偏好, 可以对模型的复杂度进行约束, 因此可以在性能相同的模型中，选择出模型复杂度最低的一个。

(f)求解岭回归优化问题

$$\begin{aligned} & \arg \min_{\beta}(\boldsymbol{y}-\boldsymbol{X} \boldsymbol{\beta})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X} \boldsymbol{\beta})+\lambda \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{\beta} \\ & \frac{\partial E}{\partial \boldsymbol{\beta}}=2 \boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X} \boldsymbol{\beta}-\boldsymbol{y})+2 \lambda \boldsymbol{\beta}=\mathbf{0} \end{aligned} \tag{11}$$

解得最优

$$\boldsymbol{\beta}^{*}=\left(\boldsymbol{X}^{\mathrm{T}} \boldsymbol{X}+\lambda \boldsymbol{I}\right)^{-1} \boldsymbol{X}^{\mathrm{T}} \boldsymbol{y} \tag{12}$$

(g)正则化在可逆方面的作用

加入岭回归正则项后，当λ足够大时(可解)，$\boldsymbol{X}^{\mathrm{T}} \boldsymbol{X}+\lambda \boldsymbol{I}$ 几乎总是可逆, 总能求解，解决了无法获得唯一的模型参数的问题。

同时正则化是用于对抗过拟合的关键手段

(h)极端λ的影响

如果$\lambda=0$, 岭回归退化为普通线性回归
如果$\lambda=\infty$, 则优化问题变为 $\arg \min_{\beta} \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{\beta}$, 解出 $\boldsymbol{\beta}=\mathbf{0}$

(i)λ为什么必须是超参数

因为正则化项$\lambda \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{\beta}$恒正，目标函数中只有正则化项中出现了λ，最优化目标函数时一定会将**λ优化为0或β优化为0**，失去了正则化的意义

4.5 (AUC-PR 和 AP) 我们尚未讨论过 AUC-PR 度量的计算细节. 对于二分类任务而言, 我们假设每个样本 $\boldsymbol{x}$ 都有一个得分 $f(\boldsymbol{x})$, 并按照这些得分对测试样本进行降序排序. 然后, 对于每个样本, 我们将分类阈值设置为当前样本的得分 (即只有当前样本以及之前的样本会被分为正类). 在该阈值处可以计算得到一对查准率和查全率. PR 曲线是通过连接相邻的点绘制得到的. AUC-PR 就是 PR 曲线下的面积.

令 (r_i, p_i) 表示第 i 对查全率和查准率 $(i = 1, 2, \ldots)$. 在计算面积时, r_i 和 r_{i-1} 之间的面积是使用*梯形插值* $(r_i - r_{i-1})\dfrac{p_i + p_{i-1}}{2}$ 计算得到的, 其中 $r_i - r_{i-1}$ 表示在 x 轴上的长度, p_i 和 p_{i-1} 是两根垂直线段在 y 轴上的长度. 针对所有 i 的值求和, 我们得到了 AUC-PR 值. 请注意, 我们假设第一对 $(r_0, p_0) = (0, 1)$, 这是对应于阈值 $+\infty$ 的一个伪匹配对.

(a) 对于表 4.3 所示的 10 个测试样本 (下标从 1 到 10), 当阈值设为当前样本的 $f(x_i)$ 值时, 计算查准率 (p_i) 和查全率 (r_i) 的值. 令类别 1 为正类, 补全表 4.3 中的其他

值. 将梯形近似值 $(r_i - r_{i-1})\dfrac{p_i + p_{i-1}}{2}$ 填入第 i 行的“AUC-PR”一列; 将其总和填入最后一行.

表 4.3 AUC-PR 和 AP 的计算

下标	类别标记	得分	查准率	查全率	AUC-PR	AP
0			1.0000	0.0000	-	-
1	1	1.0				
2	2	0.9				
3	1	0.8				
4	1	0.7				
5	2	0.6				
6	1	0.5				
7	2	0.4				
8	2	0.3				
9	1	0.2				
10	2	0.1				
					(?)	(?)

(b) *平均精度* (Average Precision, AP) 是另外一种能将 PR 曲线概括为数字的方法. 与 AUC-PR 类似, AP 使用矩形来近似 r_i 和 r_{i-1} 之间的面积, 为 $(r_i - r_{i-1})p_i$. 将此近似值填入第 i 行的“AP”一列中; 并将其总和填入最后一行. AUC-PR 和 AP 都是对 PR 曲线的总结, 因此它们的值应该彼此相似. 是吗?

(c) AUC-PR 和 AP 都对标记的顺序很敏感. 如果交换一下第 9 行和第 10 行的类别标记, 那么新的 AUC-PR 和 AP 是多少?

(d) 基于类别标记、得分和正类, 编程计算 AUC-PR 和 AP 的值. 使用表 4.3 中的测试样本集来验证你程序的正确性.

三、（4.5）AUC

(a)

下标	标记	得分	P	R	AUC-PR	AP
0			1	0		
1	1	1	1	0.2	0.2	0.2
2	2	0.9	0.5	0.2	0	0
3	1	0.8	0.67	0.4	0.1167	0.1333
4	1	0.7	0.75	0.6	0.1417	0.15
5	2	0.6	0.6	0.6	0	0
6	1	0.5	0.67	0.8	0.1267	0.1333
7	2	0.4	0.57	0.8	0	0
8	2	0.3	0.5	0.8	0	0
9	1	0.2	0.56	1	0.1056	0.111
10	2	0.1	0.5	1	0	0

(b)AP&PR

相似是正常的，而且AP比PR总是稍微大一点点

原因是他们的计算方式只有细微区别：

$$
\begin{aligned}
AP - PR &= \sum_{i=1}^{n}(r_i - r_{i-1})p_i - \sum_{i=1}^{n}(r_i - r_{i-1})\frac{p_i + p_{i-1}}{2} \\
&= \sum_{i=1}^{n}\frac{1}{2}(r_i - r_{i-1})\left(p_i - p_{i-1}\right)
\end{aligned} \tag{13}
$$

(c)

交换了第 9 行和第 10 行的类别标记之后, AUC-PR=0.6794, AP=0.7167.

(d)

代码：

```
from collections import Counter

v = [1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1]
label = [1, 2, 1, 1, 2, 1, 2, 2, 1, 2]#[1, 2, 1, 1, 2, 1, 2, 2, 2, 1]

P = [1.0]
R = [0.0]
TPR = [0.0]
FPR = [0.0]
for i in range(1, len(v) + 1):
    pos_count = Counter(label[:i])
    neg_count = Counter(label[i:])
```

```
        TP = pos_count.get(1, 0)
        FP = pos_count.get(2, 0)
        FN = neg_count.get(1, 0)
        TN = neg_count.get(2, 0)
        P.append(TP / (TP + FP))
        R.append(TP / (TP + FN))
AUC_PR = [0.5 * (R[i] - R[i - 1]) * (P[i] + P[i - 1]) for i in range(1, len(R))]
AP = [(R[i] - R[i - 1]) * P[i] for i in range(1, len(R))]

print('P:', [*P])
print('R:', [*R])
print('AUC_PR:', [*AUC_PR])
print('AP:', [*AP])
```

我们可以使用 k-NN 方法来做回归任务. 令 $D=\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ 为训练集, 其中标记 $y \in \mathbb{R}$ 是由 $y=F(\boldsymbol{x})+\epsilon$ 生成的, 其真正的回归函数 F 在生成标记 y 时被噪声 ϵ 所污染. 我们假设随机噪声 ϵ 独立于其他任何东西, $\mathbb{E}[\epsilon]=0$ 且 $\operatorname{Var}(\epsilon)=\sigma^2$.

对于任一测试样本 $\boldsymbol{x}$, k-NN 方法在数据集 D 中查找其 k 近邻 (k 是正整数), 记为 $\boldsymbol{x}_{nn(1)}, \boldsymbol{x}_{nn(2)}, \ldots, \boldsymbol{x}_{nn(k)}$, 其中 $1 \leqslant nn(i) \leqslant n$ 是第 i 个最近邻的索引. 那么, 对 $\boldsymbol{x}$ 的预测结果为

$$f(\boldsymbol{x}; D)=\frac{1}{k} \sum_{i=1}^k y_{nn(i)} \cdot$$

(a) 对 $\mathbb{E}[(y-f(\boldsymbol{x}; D))^2]$ 的偏置-方差分解是什么? 其中 y 是 $\boldsymbol{x}$ 的标记. 不要使用缩写 (公式 (4.28) 使用了缩写, 例如 $\mathbb{E}[f]$ 应该为 $\mathbb{E}_D[f(\boldsymbol{x}; D)]$.) 使用 $\boldsymbol{x}$、y、F、f、D 和 σ 来描述该分解.

(b) 使用 $f(\boldsymbol{x}; D)=\frac{1}{k} \sum_{i=1}^k y_{nn(i)}$ 来计算 $\mathbb{E}[f]$ (从这里开始可以使用缩写).

(c) 使用 $\boldsymbol{x}$ 和 y 来代替分解公式中的 f 那一项.

(d) 方差项是多少? 当 k 改变时, 它会如何变化?

(e) 偏置的平方项是多少? 它会如何随 k 变化? (提示: 考虑 $k=n$)

四、 (4.6) KNN

(a)偏置-方差分解

首先给出误差表达式

$$\mathbb{E}_D[(y-f(\boldsymbol{x}; D))]=\mathbb{E}_D\left[(F(\boldsymbol{x})-f(\boldsymbol{x}; D)+\epsilon)^2\right]=\mathbb{E}_D\left[(F(\boldsymbol{x})-f(\boldsymbol{x}; D))^2\right]+\sigma^2 \tag{14}$$

展开可得

$$\mathbb{E}_D\left[(F(\boldsymbol{x})-f(\boldsymbol{x}; D))^2\right]=\left(\mathbb{E}_D[F(\boldsymbol{x})-f(\boldsymbol{x}; D)]\right)^2+\operatorname{Var}(F(\boldsymbol{x})-f(\boldsymbol{x}; D)) \tag{15}$$

由于 $F(\boldsymbol{x})$ 是确定的, 与训练集 D 无关, 即 $\mathbb{E}_D[F(\boldsymbol{x})]=F(\boldsymbol{x})$, 则上式进一步简化为:

$$\begin{aligned} & \left(\mathbb{E}_D[F(\boldsymbol{x})-f(\boldsymbol{x}; D)]\right)^2+\operatorname{Var}(F(\boldsymbol{x})-f(\boldsymbol{x}; D)) \\ & =\left(F(\boldsymbol{x})-\mathbb{E}_D[f(\boldsymbol{x}; D)]\right)^2+\operatorname{Var}(f(\boldsymbol{x}; D)) \\ & =\left(F(\boldsymbol{x})-\mathbb{E}_D[f(\boldsymbol{x}; D)]\right)^2+\mathbb{E}_D\left[\left(f(\boldsymbol{x}; D)-\mathbb{E}_D[f(\boldsymbol{x}; D)]\right)^2\right] \end{aligned} \tag{16}$$

综上,得到偏置-方差分解

$$\begin{aligned} & \mathbb{E}_D[(y-f(\boldsymbol{x}; D))] \\ = & \left(F(\boldsymbol{x})-\mathbb{E}_D[f(\boldsymbol{x}; D)]\right)^2+\mathbb{E}_D\left[\left(f(\boldsymbol{x}; D)-\mathbb{E}_D[f(\boldsymbol{x}; D)]\right)^2\right]+\sigma^2 \end{aligned} \tag{17}$$

(b)带入，缩写

最后一步

$$\mathbb{E}[f]=\mathbb{E}\left[\frac{1}{k}\sum_{i=1}^{k}y_{nn(i)}\right]=\frac{1}{k}\sum_{i=1}^{k}\mathbb{E}\left[F\left(\boldsymbol{x}_{nn(i)}\right)+\epsilon\right]=\frac{1}{k}\sum_{i=1}^{k}\mathbb{E}\left[F\left(\boldsymbol{x}_{nn(i)}\right)\right]=\frac{1}{k}\sum_{i=1}^{k}F\left(\boldsymbol{x}_{nn(i)}\right) \tag{18}$$

(c)x,y带入f

带入得到简化

$$\begin{aligned}
&(F(\boldsymbol{x})-\mathbb{E}_D[f(\boldsymbol{x};D)])^2+\mathbb{E}_D\left[(f(\boldsymbol{x};D)-\mathbb{E}_D[f(\boldsymbol{x};D)])^2\right]+\sigma^2\\
=&\left(F(\boldsymbol{x})-\frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_D\left[F\left(\boldsymbol{x}_{nn(i)}\right)\right]\right)^2+\mathbb{E}_D\left[\frac{1}{k}\left(\sum_{i=1}^{k}y_{nn(i)}-\frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_D\left[F\left(\boldsymbol{x}_{nn(i)}\right)\right]\right)^2\right]+\sigma^2\\
=&\left(F(\boldsymbol{x})-\frac{1}{k}\sum_{i=1}^{k}F\left(\boldsymbol{x}_{nn(i)}\right)\right)^2+\frac{1}{k^2}\mathbb{E}_D\left[\left(\sum_{i=1}^{k}\left(y_{nn(i)}-\mathbb{E}_D\left[F\left(\boldsymbol{x}_{nn(i)}\right)\right]\right)\right)^2\right]+\sigma^2\\
=&\left(F(\boldsymbol{x})-\frac{1}{k}\sum_{i=1}^{k}F\left(\boldsymbol{x}_{nn(i)}\right)\right)^2+\frac{\sigma^2}{k^2}+\sigma^2
\end{aligned} \tag{19}$$

(d)方差项与k

$$\frac{\sigma^2}{k^2} \tag{20}$$

k增大时，方差项系数变小，找到的最近邻更多，方差整体减小

(e)偏差平方项与k

$$\left(F(\boldsymbol{x})-\frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_D\left[F\left(\boldsymbol{x}_{nn(i)}\right)\right]\right)^2 \tag{21}$$

k增大时，这两项的差会越来越大，导致偏差增大。尤其是当k=n时，偏差达到最大，方差达到最小（0）

使用 Matlab 或 GNU Octave 完成以下实验. 编程实现 PCA 和白化变换 —— 你可以使用 eig 或 svd 等函数, 但不能使用可直接完成本任务的函数 (例如 princomp 函数).

(a) 使用 `x=randn(2000,2)*[2 1;1 2]` 生成 2000 个样本, 每个样本都是二维的. 使用 scatter 函数画出这 2000 个样本.

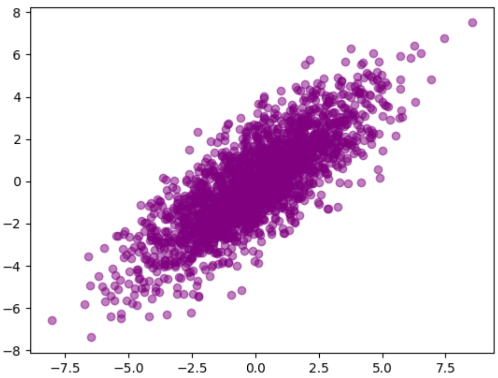
(b) 对这些样本进行 PCA 变换并保留所有的 2 个维度. 使用 scatter 函数画出 PCA 后的样本.

(c) 对这些样本进行白化变换并保留所有的 2 个维度. 使用 scatter 函数画出 PCA 后的样本.

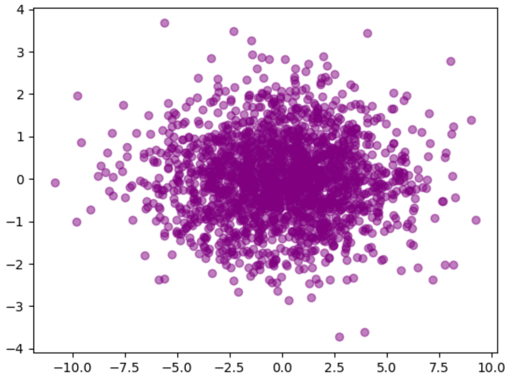
(d) 如果在 PCA 变换中保留所有的维度, 为什么 PCA 是数据 (在进行平移之后) 的一个旋转? 这一操作为什么会有用?

五、（5.3）编程：PCA&白化

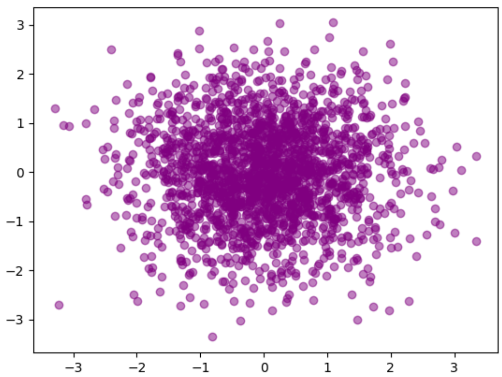
(a)



(b)



(c)



(d)PCA本质是旋转

因为PCA 本质是将数据视作了一个多维空间中的类球形, 并把这个球的各个轴按照各方差最大的方向，旋转对齐到坐标轴上。用数学方式解释，就是把数据乘上一个旋转矩阵，

PCA 旋转这一操作有效的原因： PCA数据降维的本质，就是在对齐到坐标轴上后，把短轴对应纬度去掉, 保留几个长轴对应的维度，进而得到新的降维后数据。由于已经进行旋转对齐，所以去除短轴这一过程很简单，只需比较轴长短即可。

(条件数) 给定任意矩阵范数 $\|\cdot\|$, 我们可为一个非奇异实方阵定义一个对应的条件数(condition number). 矩阵 X 的条件数定义为

$$\kappa(X)=\|X\|\|X^{-1}\|.$$

一个常用的条件数为 2-范数条件, 记为

$$\kappa_2(X)=\|X\|_2\|X^{-1}\|_2.$$

如果一个矩阵的条件数很大, 则称该矩阵为*病态的*(ill-conditioned).

(a) 如果你已经知道 X 的奇异值为 $\sigma_1 \geqslant \sigma_2 \geqslant \ldots \geqslant \sigma_n$, 那么条件数 $\kappa_2(X)$ 是多少?

(b) 令 f 表示一个从 $\mathbb{X}$ 映射到 $\mathbb{Y}$ 的函数. 假设 $f(\boldsymbol{x})=\boldsymbol{y}$ 以及 $f(\boldsymbol{x}+\Delta\boldsymbol{x})=\boldsymbol{y}+\Delta\boldsymbol{y}$. 如果一个较小的 $\Delta\boldsymbol{x}$ 会导致一个较大的 $\Delta\boldsymbol{y}$, 我们称 f 为*病态函数*. 令 A 为一个 $\mathbb{R}^{n\times n}$ 中的满秩方阵 (即可逆矩阵) 并且 $\boldsymbol{b}\in\mathbb{R}^n$, 我们想要求解 $A\boldsymbol{x}=\boldsymbol{b}$. 如果 $\kappa_2(A)$ 很大, 说明该线性系统是病态的. (你不需要证明这个结论. 只需要给出一些直觉来说明为什么病态矩阵 A 是坏的.)

(c) 证明正交矩阵是*良态的*(well-conditioned) (即有较小的条件数).

六、（6.3）条件数

(a)矩阵 2-范数$=\sigma_{max}$

矩阵 2-范数等于其最大奇异值，可知 $\|\boldsymbol{X}\|_2=\sigma_1$, 且由矩阵的逆的性质可知 $\left\|\boldsymbol{X}^{-1}\right\|_2=\frac{1}{\sigma_n}$

$$\kappa_2(\boldsymbol{X})=\|\boldsymbol{X}\|_2\left\|\boldsymbol{X}^{-1}\right\|_2=\frac{\sigma_1}{\sigma_n} \tag{22}$$

(b)病态线性系统

我们想要解释的是，在$\kappa_2(\boldsymbol{A})$很大的情况下, 稍微改变 $\boldsymbol{A}$ 或 $\boldsymbol{b}$ 就会使 $\boldsymbol{x}$有很大的改变.

已知$\Delta\boldsymbol{x}=A^{-1}\Delta\boldsymbol{b}$, $\|\boldsymbol{b}\|\leq\|\boldsymbol{A}\|\|\boldsymbol{x}\|$, $\|\Delta\boldsymbol{x}\|\leq\left\|\boldsymbol{A}^{-1}\right\|\|\Delta\boldsymbol{b}\|$

相乘再除以 $\|\boldsymbol{b}\|\|\boldsymbol{x}\|$ 可得

$$\frac{\|\Delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|}\leq\|\boldsymbol{A}\|\left\|\boldsymbol{A}^{-1}\right\|\frac{\|\Delta\boldsymbol{b}\|}{\boldsymbol{b}}=\kappa_2(\boldsymbol{A})\frac{\|\Delta\boldsymbol{b}\|}{\|\boldsymbol{b}\|} \tag{23}$$

再进行扰动$\Delta\boldsymbol{A}$ 可得

$$\begin{gathered}(\boldsymbol{A}+\Delta\boldsymbol{A})(\boldsymbol{x}+\Delta\boldsymbol{x})=\boldsymbol{b}=\boldsymbol{A}\boldsymbol{x}\\ \boldsymbol{A}\Delta\boldsymbol{x}=-\Delta\boldsymbol{A}(\boldsymbol{x}+\Delta\boldsymbol{x})\\ \Delta\boldsymbol{x}=-\boldsymbol{A}^{-1}\Delta\boldsymbol{A}(\boldsymbol{x}+\Delta\boldsymbol{x})\end{gathered} \tag{24}$$

因此我们使用范数不等式并两边除以$\|\boldsymbol{x}+\Delta\boldsymbol{x}\|$ 有

$$\frac{\|\Delta\boldsymbol{x}\|}{\|\boldsymbol{x}+\Delta\boldsymbol{x}\|}\leq\|\boldsymbol{A}\|\left\|\boldsymbol{A}^{-1}\right\|\frac{\|\Delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|}=\kappa_2(\boldsymbol{A})\frac{\|\Delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|} \tag{25}$$

由此表达式发钱，即使当有较小的扰动$\Delta\boldsymbol{A}$ 或者 $\Delta\boldsymbol{b}$的时候, 也会带来较大的 $\Delta\boldsymbol{x}$, 小的输入变换就会导致较大的输出变化

这一定程度上说明了病态系统的原因

(c)良态正交矩阵

正交矩阵的逆等于其转置，有相同特征值

$$\kappa_2(\boldsymbol{X}) = \|\boldsymbol{W}\|_2 \left\|\boldsymbol{W}^{-1}\right\|_2 = \|\boldsymbol{W}\|_2 \left\|\boldsymbol{W}^{\mathrm{T}}\right\|_2 = (\|\boldsymbol{W}\|_2)^2 = 1 \tag{26}$$