特征学习与稀疏学习

汇报人：杜昭武

2024/11/3

1.子集搜索与评价

相关特征：对当前的学习任务有用的属性称为“相关特征”

特征选择：从给定特征集合中选择出相关特征子集的过程

冗余特征：在机器学习问题中，某些特征特供的信息与其他特征已经提供的信息高度相似或重复，对模型的性能提升很小或者没有贡献。这样的特征不会为模型带来额外的信息，却可能增加模型的复杂性，降低模型的可解释性，甚至在某些情况下可能会导致过拟合。

假定：本章数据中不涉及冗余特征，并且假定初始的特征集合包含了所有重要信息

思路：产生一个候选子集

如何 根据评价结果获取下一个候选特征子集?

如何评价候选特征子集的好坏?

1.子集搜索与评价

- 子集搜索（subset search）
  - 前向搜索

给定特征集合 $\{a_1, a_2, \ldots, a_d\}$, 将每个特征看作一个候选子集, 对这 d 个候选单特征子集进行评价, 假定 $\{a_2\}$ 最优, 于是将 $\{a_2\}$ 作为第一轮的选定集; 然后, 在上一轮的选定集中加入一个特征, 构成包含两个特征的候选子集, 假定在这 $d-1$ 个候选两特征子集中 $\{a_2, a_4\}$ 最优, 且优于 $\{a_2\}$, 于是将 $\{a_2, a_4\}$ 作为本轮的选定集; ……假定在第 $k+1$ 轮时, 最优的候选 $(k+1)$ 特征子集不如上一轮的选定集, 则停止生成候选子集, 并将上一轮选定的 k 特征集合作为特征选择结果.

  - 后向搜索

若从完整的特征集合开始，每次尝试去掉一个无关特征

  - 双向搜索

每一轮逐渐增加选定相关特征（这些特征在后续轮中不会被去除）、同时减少无关特征

1.子集搜索与评价

- 子集评价(subset evaluation)

给定数据集 D, 假定 D 中第 i 类样本所占的比例为 p_i $(i = 1, 2, \ldots, |\mathcal{Y}|)$.
对属性子集 A, 假定根据其取值将 D 分成了 V 个子集 $\{D^1, D^2, \ldots, D^V\}$,
每个子集中的样本在 A 上取值相同, 于是我们可计算属性 子集 A 的信息增益

其中信息熵定义为

$$\mathrm{Gain}(A) = \mathrm{Ent}(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} \mathrm{Ent}(D^v) \,,$$

$$\mathrm{Ent}(D) = -\sum_{i=1}^{|\mathcal{Y}|} p_k \log_2 p_k$$

- 信息增益Gain越大， 意味着特征子集A包含的有助于分类的信息越多

常见的特征选择方法大致分为三类:

- 过滤式filter
- 包裹式wrapper
- 嵌入式embedding

1.子集搜索与评价

假设有一个简单的数据集如下：

天气	玩耍
晴天	是
阴天	是
雨天	否
晴天	是
晴天	否
阴天	否
阴天	是
雨天	否

1. 计算数据集的熵 $H(D) = -(\frac{4}{8}\log_2\frac{4}{8} + \frac{4}{8}\log_2\frac{4}{8}) = 1$

2. 计算 D 的条件熵 $H(D|天气) = \sum_{V\in\{晴,阴,雨\}} \frac{|D_V|}{|D|} H(D_V)$

① $|D_{晴}| = 3$，且 $H(D_{晴}) = -(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}) \approx 0.918$

② $|D_{阴}| = 3$ 且 $H(D_{阴}) = -(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}) \approx 0.918$

③ $|D_{雨}| = 2$ 且 $H(D_{雨}) = -(0\log_2 0 + 1\log_2 1) = 0$

$H(D|天气) = (\frac{3}{8}\times 0.918 + \frac{3}{8}\times 0.918 + \frac{2}{8}\times 0) \approx 0.688$

3. 计算信息增益 $Gain(D, 天气) = H(D) - H(D|天气) = 1 - 0.688 = 0.312$

2.过滤式选择

- 过滤式方法先对数据集进行特征选择，然后再训练学习器，特征选择过程与后续学习器无关。这相当于先用特征选择过程对初始特征进行 “过滤”，再用过滤后的特征来训练模型。换而言之，过滤式特征选择是一种在特征选择过程中独立于具体机器学习模型的方法。
- Relief： 设计了一个“相关统计量”来度量特征的重要性

给定训练集 $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$

  - 猜中近邻（near-hit）：在Xi的同类样本中寻找其最近$\boldsymbol{x}_{i,\mathrm{nh}}$
  - 猜错近邻（near-miss）：从异类样本中寻找其近邻$\boldsymbol{x}_{i,\mathrm{nm}}$
  - 相关统计量对应于属性j的分量为

$$\delta^j = \sum_i -\mathrm{diff}(x_i^j, x_{i,\mathrm{nh}}^j)^2 + \mathrm{diff}(x_i^j, x_{i,\mathrm{nm}}^j)^2$$

2.过滤式选择

$$\delta^j = \sum_i -\mathrm{diff}(x_i^j, x_{i,\mathrm{nh}}^j)^2 + \mathrm{diff}(x_i^j, x_{i,\mathrm{nm}}^j)^2$$

x_a^j 表示样本 $\boldsymbol{x}_a$ 在属性 j 上的取值

$\mathrm{diff}(x_a^j, x_b^j)$ 取决于属性 j 的类型: 若属性 j 为离散型, 则 $x_a^j = x_b^j$ 时 $\mathrm{diff}(x_a^j, x_b^j) = 0$, 否则为 1;
若属性 j 为连续型, 则 $\mathrm{diff}(x_a^j, x_b^j) = |x_a^j - x_b^j|$

- 若 $\boldsymbol{x}_i$ 与其猜中近邻 $\boldsymbol{x}_{i,\mathrm{nh}}$ 在属性 j 上的距离小于 $\boldsymbol{x}_i$ 与 其猜错近邻 $\boldsymbol{x}_{i,\mathrm{nm}}$ 的距离,
则说明属性 j 对区分同类与异类样本是有益的, 于是 增大属性 j 所对应的统计量分量;
- 若 $\boldsymbol{x}_i$ 与其猜中近邻 $\boldsymbol{x}_{i,\mathrm{nh}}$ 在属性 j 上的距离大于 $\boldsymbol{x}_i$ 与其猜错近邻 $\boldsymbol{x}_{i,\mathrm{nm}}$ 的距离,
则说明属性 j 起负面作用, 于是减小属性 j 所对应的统计量分量.

3.包裹式选择

- 包裹式特征选择直接把最终将要使用的学习器的性能作为特征子集的评价准则。换言之，包裹式特征选择的目的是为给定学习器选择最有利于其性能，“量身定做”的特征子集
- LVW（Las Vegas Wrapper）.在拉斯维加斯方法的框架下使用随机策略来进行子集搜索

输入： 数据集 D;
特征集 A;
学习算法 $\mathfrak{L}$;
停止条件控制参数 T.

过程：

```
E = ∞;
d = |A|;
A* = A;
t = 0;
while t < T do
    随机产生特征子集 A';
    d' = |A'|;
    E' = CrossValidation(L(D^A'));
    if (E' < E) ∨ ((E' = E) ∧ (d' < d)) then
        t = 0;
        E = E';
        d = d';
        A* = A'
    else
        t = t + 1
    end if
end while
```

输出： 特征子集 A^*

在数据集 D 上，使用交叉验证法来估计学习器 $\mathfrak{L}$ 的误差

4.嵌入式选择与L1正则化

- 嵌入式特征选择是将特征选择过程与学习器训练过程融为一体，即在学习器训练过程中自动进行了特征选择

给定数据集 $D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\ldots,(\boldsymbol{x}_m,y_m)\}$, 其中 $\boldsymbol{x}\in\mathbb{R}^d, y\in\mathbb{R}$.

- 考虑最简单的线性回归模型，以平方误差为损失函数，则优化目标为

$$\min_{\boldsymbol{w}}\sum_{i=1}^{m}(y_i-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)^2 .$$

- 为了缓解过拟合问题，引入正则化项目，若使用L1范数正则化，则

$$\min_{\boldsymbol{w}}\sum_{i=1}^{m}(y_i-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)^2+\lambda\|\boldsymbol{w}\|_1 .$$

4.嵌入式选择与L1正则化

- 采用L1范数，则

$$\min_{\boldsymbol{w}} \sum_{i=1}^{m} (y_i - \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i)^2 + \lambda \|\boldsymbol{w}\|_1 \ .$$

L_1 正则化问题的求解可使用近端梯度下降 (Proximal Gradient Descent, 简称 PGD) [Boyd and Vandenberghe, 2004]. 具体来说, 令 ∇ 表示微分算子, 对优化目标

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \lambda \|\boldsymbol{x}\|_1 \ , \tag{11.8}$$

若 $f(\boldsymbol{x})$ 可导, 且 ∇f 满足 L-Lipschitz 条件, 即存在常数 $L > 0$ 使得

$$\left\|\nabla f(\boldsymbol{x}') - \nabla f(\boldsymbol{x})\right\|_2^2 \leqslant L \left\|\boldsymbol{x}' - \boldsymbol{x}\right\|_2^2 \quad (\forall \boldsymbol{x}, \boldsymbol{x}') \ , \tag{11.9}$$

4.嵌入式选择与L1正则化

$\min_x f(x) + \lambda \|x\|_1$

$\Rightarrow$ 泰勒: $f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots$

即: $f(x) \approx f(x_k) + \nabla f(x_k)(x - x_k) + \frac{L}{2}\|x - x_k\|^2$

$= \frac{L}{2}\|x - (x_k - \frac{1}{L}\nabla f(x_k))\|_2^2 + \text{const}$

则在 $\boldsymbol{x}_k$ 附近可将 $f(\boldsymbol{x})$ 通过二阶泰勒展式近似为

$$\begin{aligned}\hat{f}(\boldsymbol{x}) &\simeq f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{x}_k\|^2 \\ &= \frac{L}{2}\left\|\boldsymbol{x} - \left(\boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)\right)\right\|_2^2 + \text{const}, \end{aligned} \tag{11.10}$$

于是, 若通过梯度下降法对 $f(\boldsymbol{x})$ 进行最小化, 则每一步梯度下降迭代实际上等价于最小化二次函数 $\hat{f}(\boldsymbol{x})$. 将这个思想推广到式(11.8), 则能类似地得到其每一步迭代应为

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}} \frac{L}{2}\left\|\boldsymbol{x} - \left(\boldsymbol{x}_k - \frac{1}{L}\nabla f(\boldsymbol{x}_k)\right)\right\|_2^2 + \lambda\|\boldsymbol{x}\|_1 , \tag{11.12}$$

4.嵌入式选择与L1正则化

即在每一步对 $f(\boldsymbol{x})$ 进行梯度下降迭代的同时考虑 L_1 范数最小化.

对于式(11.12), 可先计算 $\boldsymbol{z}=\boldsymbol{x}_k-\frac{1}{L}\nabla f(\boldsymbol{x}_k)$, 然后求解

$$\boldsymbol{x}_{k+1}=\arg\min_{\boldsymbol{x}}\frac{L}{2}\|\boldsymbol{x}-\boldsymbol{z}\|_2^2+\lambda\|\boldsymbol{x}\|_1\ . \tag{11.13}$$

$x_{k+1}^{i}=\arg\min_{x^i}\frac{L}{2}(x^i-z^i)^2+\lambda|x^i|$

$\frac{L}{2}x^2-Lzx+\frac{L}{2}z^2+\lambda|x|$

$x>0$: $\frac{L}{2}x^2+(\lambda-Lz)x+\frac{L}{2}z^2$ $\quad x^*=\frac{zL-\lambda}{L}$

$x<0$: $\frac{L}{2}x^2+(\lambda-Lz)x+\frac{L}{2}z^2$ $\quad x^*=\frac{zL+\lambda}{L}$

令 x^i 表示 $\boldsymbol{x}$ 的第 i 个分量, 将式(11.13)按分量展开可看出, 其中不存在 x^ix^j $(i\neq j)$ 这样的项, 即 $\boldsymbol{x}$ 的各分量互不影响, 于是式(11.13)有闭式解

$$x_{k+1}^i=\begin{cases}z^i-\lambda/L, & \lambda/L<z^i\ ;\\ 0, & |z^i|\leqslant\lambda/L\ ;\\ z^i+\lambda/L, & z^i<-\lambda/L\ ,\end{cases} \tag{11.14}$$

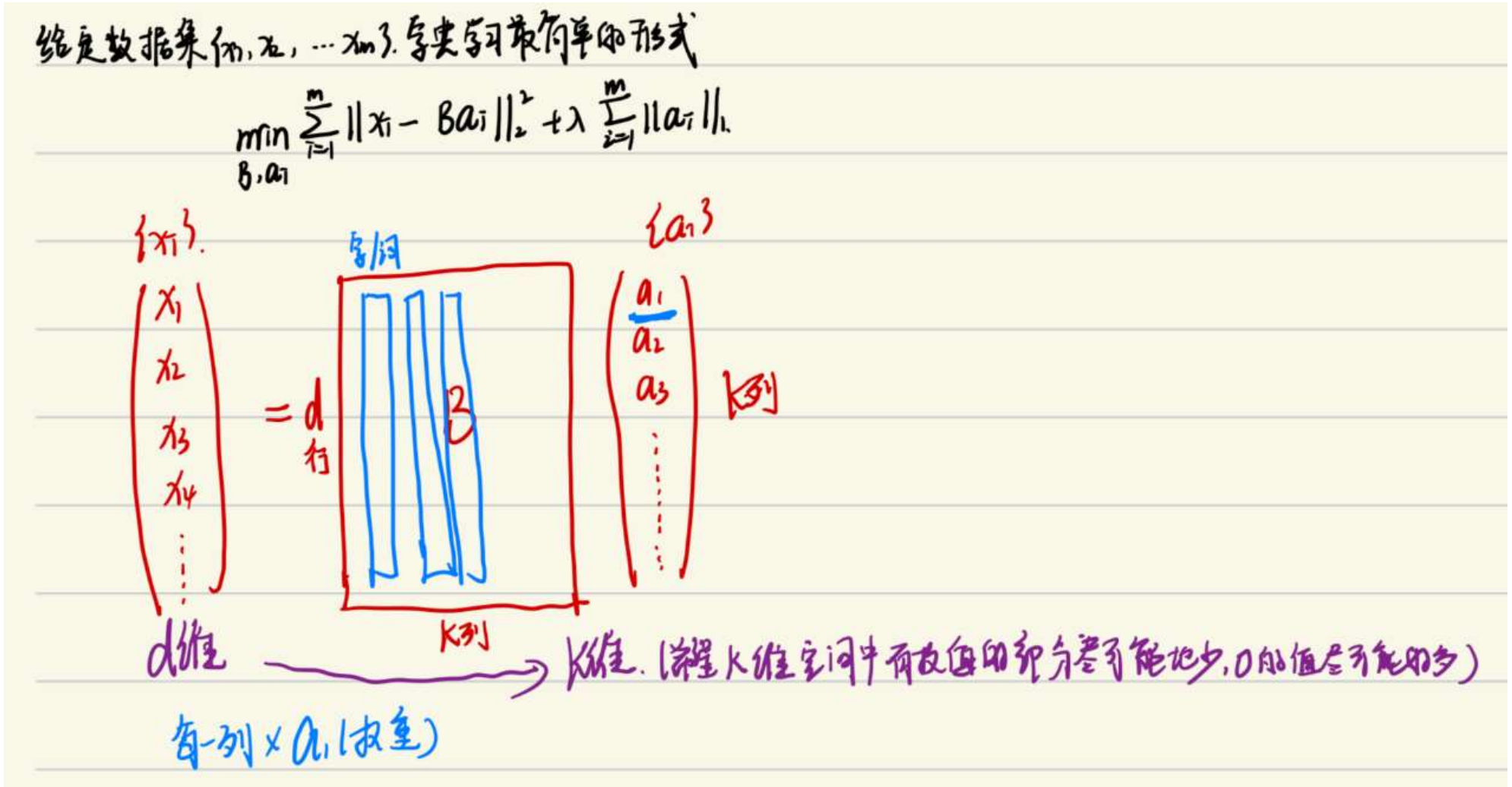
5.稀疏表示与字典学习

- 考虑另外一种稀疏性：数据集D所对应的矩阵中存在很多零元素，但这些零元素并不是以整列整行的形式存在
- 字典学习（稀疏编码）：为普通稠密表达的样本找到合适的字典，将样本转化为合适的稀疏表达形式，从而使学习任务得以简化，模型的复杂度得以降低。
- 本质
  - 对庞大数据集的降维表示
  - 尝试学习蕴藏在样本背后最质朴的特征

给定数据集 $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$, 字典学习最简单的形式为

$$\min_{\mathbf{B},\boldsymbol{\alpha}_i} \sum_{i=1}^{m} \|\boldsymbol{x}_i - \mathbf{B}\boldsymbol{\alpha}_i\|_2^2 + \lambda \sum_{i=1}^{m} \|\boldsymbol{\alpha}_i\|_1 \ , \tag{11.15}$$

5.稀疏表示与字典学习


给定数据集 $\{x_1, x_2, \cdots x_m\}$ 字典学习最简单的形式
$\min_{B,a_i} \sum_{i=1}^{m} \|x_i - Ba_i\|_2^2 + \lambda \sum_{i=1}^{m} \|a_i\|_1$
$\{x_i\}$
x_1
x_2
x_3
x_4
$= d$ 行
字典
B
K列
d维
$\{a_i\}$
a_1
a_2
a_3
K列
K维 (降维K维空间中有效值的个数可能比少，0的值可能比多)
每一列×a_i (权重)

5.稀疏表示与字典学习

- 采用变量交替优化的策略来求解（字典B和稀疏矩阵ai）
- 第一步：固定住字典B

$$\min_{\boldsymbol{\alpha}_i} \|\boldsymbol{x}_i - \mathbf{B}\boldsymbol{\alpha}_i\|_2^2 + \lambda\|\boldsymbol{\alpha}_i\|_1 \ . \tag{11.16}$$

- 第二步：固定住ai来更新字典B

$$\min_{\mathbf{B}} \|\mathbf{X} - \mathbf{B}\mathbf{A}\|_F^2, \tag{11.17}$$

  - 基于逐列更新策略的KSVD　令 $\boldsymbol{b}_i$ 表示字典矩阵 $\mathbf{B}$ 的第 i 列，$\boldsymbol{\alpha}^i$ 表示稀疏矩阵 $\mathbf{A}$ 的第 i 行，式(11.17)可重写为

$$\begin{aligned}
\min_{\mathbf{B}} \|\mathbf{X} - \mathbf{B}\mathbf{A}\|_F^2 &= \min_{\boldsymbol{b}_i} \left\| \mathbf{X} - \sum_{j=1}^{k} \boldsymbol{b}_j \boldsymbol{\alpha}^j \right\|_F^2 \\
&= \min_{\boldsymbol{b}_i} \left\| \left(\mathbf{X} - \sum_{j \neq i} \boldsymbol{b}_j \boldsymbol{\alpha}^j \right) - \boldsymbol{b}_i \boldsymbol{\alpha}^i \right\|_F^2 \\
&= \min_{\boldsymbol{b}_i} \|\mathbf{E}_i - \boldsymbol{b}_i \boldsymbol{\alpha}^i\|_F^2 \ .
\end{aligned} \tag{11.18}$$

5.稀疏表示与字典学习

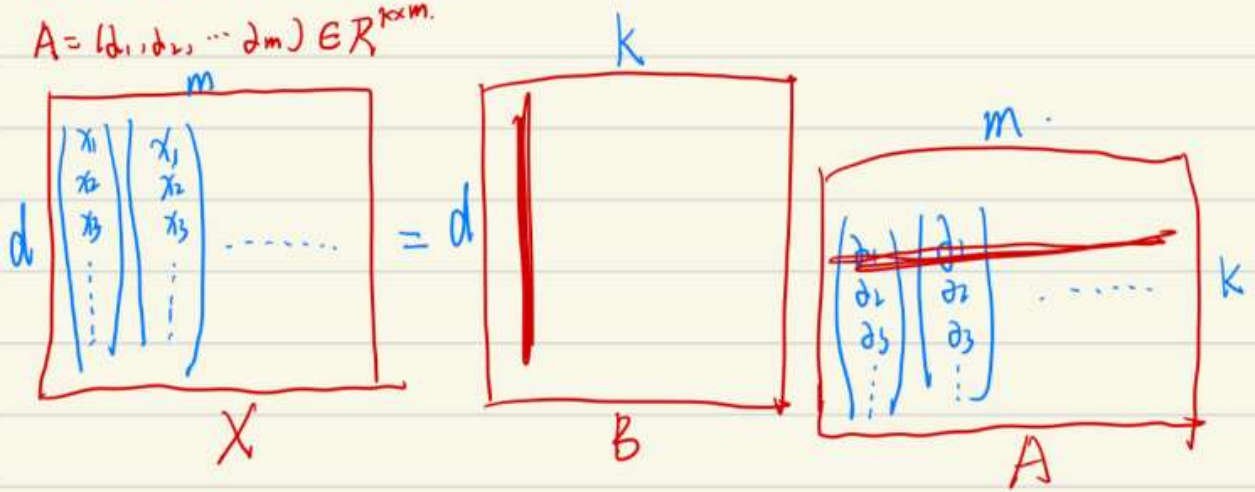
求解：1.固定B. $\min_{\alpha_i} \|x_i - B\alpha_i\|_2^2 + \lambda\|\alpha_i\|_1$

B：$d \times k$.　x_i：向量=范数的平方和

2. 固定α_i $\min_B \|X - BA\|_F^2$.　范数：每一项的平方和加起来

$X = (x_1, x_2, \cdots x_m) \in R^{d \times m}$

$A = (\alpha_1, \alpha_2, \cdots \alpha_m) \in R^{k \times m}$.



要使得所有元素的平方和最小，采用逐列更新策略

B的一列为一个基(词)，每次更新一列

b_i表示B中的第i列，α^j表示稀疏矩阵A的第j行

即 $\min\|X - BA\|_F^2 = \min_{b_i} \|X - \sum_{j=1}^{k} b_j \alpha^j\|_F^2$.

$d \times m$且秩为1的矩阵，k个矩阵相加

$= \min_{b_i} \|(X - \sum_{j \neq i} b_j \alpha^j) - b_i \alpha^i\|_F^2$　将第i列单独列出来

$= \min_{b_i} \|E_i - b_i \alpha^i\|_F^2$.

要更新的秩为1，近似最小.

对E_i进行奇异值分解，取得最大奇异值所对应的正交向量.

课后习题

11.1 试编程实现 Relief 算法, 并考察其在西瓜数据集 3.0 上的运行结果.

11.2 试写出 Relief-F 的算法描述.

Relief-F算法对δ^j的计算方式进行了扩展。

Relief-F算法步骤如下：

- 假定数据集D中样本共有|Y|个类别，
- 其中样本x_i属于第k类：|
- （1）先在第k类样本中寻找最近邻$x_{i,nh}$，称为"**猜中近邻**"；
- （2）再在第k类样本外寻找最近邻$x_{i,nm}$，称为"**猜错近邻**"；
- （3）计算**相关统计量**对应于**属性j的分量**：

$$\delta^j = \sum_i -\mathrm{diff}\left(x_i^j, x_{i,\mathrm{nh}}^j\right)^2 + p_l \cdot \mathrm{diff}\left(x_i^j, x_{i,l,\mathrm{nm}}^j\right)^2$$

其中，p_l为第l类样本在数据集D中所占比例。

- （4）将基于不同样本得到的估计结果取均值。

11.2.

输入：样本数据 X，类标记 Y

过程：计算 样本数 m，特征数 n，类别数 N=|y|，各类样本的占比例.

初始 δ = zeros(n)

for $i = 1, 2, \cdots m$

为所属类别 k

在 x_i 中的同类样本找 最近邻同类样本 $x_{i,nh}$

for $j = 1, 2, \cdots n$

$\delta_j = \delta_j - diff(x_i^j - x_{i,nh}^j)^2$

for $l = 1, 2, 3, \cdots, k-1, k+1, \cdots N$

在属于 l 类别的样本子集中找到 与 x_i 的最近邻样本 $x_{i,l,nm}$

for $j = 1, 2, \cdots n$;

$\delta_j = \delta_j + p_l * diff(x_i^j - x_{i,l,nm}^j)^2$

输出：相关统计量 δ.