# 附录

## 1、向量-线性回归

### 1.1、符号

$\boldsymbol{x}$ 列向量

$\boldsymbol{x}^T$ 行向量

$\boldsymbol{X}$ 矩阵

$\boldsymbol{I}$ 单位矩阵

$\boldsymbol{\mathcal{X}}$ 张量

$||\cdot||_2$ L2范数

### 1.2、Least Squares Regression

传统线性回归模型可以写作：

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{X}) + \boldsymbol{\varepsilon}$$

使用线性模型拟合得到的估计值模型为:

$$\hat{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{\beta} = \beta_0\boldsymbol{1} + \beta_1\boldsymbol{x}_1 + \cdots + \beta_k\boldsymbol{x}_k$$

其中

$\boldsymbol{y} \in \mathbb{R}^n$

$\boldsymbol{X} \in \mathbb{R}^{n\times(k+1)}$

$\boldsymbol{\varepsilon} \in \mathbb{R}$

$\boldsymbol{\beta} \in \mathbb{R}^{k+1}$

注：下文$\boldsymbol{k+1}$简写为$\boldsymbol{k}$

使用最小二乘损失函数获取权重参数

$$
\begin{aligned}
&\mathcal{L} \\
&= Loss\ Function \\
&= RSS(\beta) \\
&= ||y - X\beta||_2^2 \\
&= (y - X\beta)^T (y - X\beta) \\
&= y^T y - (X\beta)^T y - y^T X\beta + (X\beta)^T (X\beta) \\
&= (X\beta)^T (X\beta) - 2(X\beta)^T y + y^T y \\
&= \beta^T X^T X\beta - 2\beta^T X^T y + y^T y
\end{aligned}
$$

其中 $y^T X\beta = scalar = (X\beta)^T y$

令 $A = X^T X \in \mathbb{R}^{k \times k}$

$$
\begin{aligned}
&\frac{\partial \beta^T X^T X\beta}{\partial \beta} = \frac{\partial \beta^T A\beta}{\partial \beta} \\
&= (A + A^T)\beta \\
&= (X^T X + (X^T X)^T)\beta \\
&= 2X^T X\beta
\end{aligned}
$$

$$
\begin{aligned}
&\frac{\partial \beta^T X^T y}{\partial \beta} = \\
&\frac{\partial \beta^T}{\partial \beta}(X^T y) + \beta^T \frac{\partial X^T y}{\partial y} \\
&= I(X^T y + \beta^T 0) \\
&= X^T y
\end{aligned}
$$

$$
\frac{\partial \mathcal{L}}{\partial \beta} = 2X^T X\beta - 2X^T y
$$

由于这是凸优化问题，令 $\frac{\partial J}{\partial \beta} = 0$，则得到权重参数 $\beta$ 的估计值

$$
\hat{\beta} = (X^T X)^{-1} X^T y
$$

最小二乘回归根据预测变量拟合响应变量为

$$
\hat{y} = X\hat{\beta} = X(X^T X)^{-1} X^T y
$$

## 1.3、 Proof: $\frac{\partial \beta^T A\beta}{\partial \beta} = (A + A^T)\beta$

- 微分法

设 $u(\beta) = \beta^T A\beta,\ h = \Delta\beta$

$$
\begin{aligned}
&u(\beta + h) \\
&= (\beta + h)^T A(\beta + h) \\
&= \beta^T A\beta + h^T A\beta + \beta^T Ah + h^T Ah \\
&= u(\beta) + \beta^T (A + A^T)h + u(h) \\
&= u(\beta) + h^T (A + A^T)\beta + u(h)
\end{aligned}
$$

其中 $h^T A\beta = 1 \times 1\ matrix = scalar = \beta^T A^T h$

设$r_\beta(h)=u(h)=h^TAh$，当$h\to 0$则$r_\beta(h)=o(\|h\|)$这证明$u$在$\beta$处的微分是线性函数。则

$$\because\ u(\beta+\Delta\beta)-u(\beta)=h^T(A+A^T)\beta+u(h)$$

$$\therefore\ \nabla u(\beta)=I(A+A^T)\beta+o(\|h\|)$$

即 $\frac{\partial\beta^TA\beta}{\partial\beta}=(A+A^T)\beta$

- 元素法

设$A\in\mathbb{R}^{k\times k}$ 即 $A=[a_{i,j}]$ , $\beta\in\mathbb{R}^k$

$$u=\beta^TA\beta=\sum_{i=1}^{k}\sum_{j=1}^{k}a_{i,j}\beta_i\beta_j$$

根据元素对列向量求导规则

$$\frac{\partial y_p}{\partial x}=[\frac{\partial y_p}{\partial x_1},\cdots,\frac{\partial y_p}{\partial x_k}]^T$$

$$\begin{aligned}
&\frac{\partial u}{\partial\beta_p}\\
&=\frac{\sum_{i=1}^{k}\sum_{j=1}^{k}\partial a_{i,j}\beta_i\beta_j}{\partial\beta_p}\\
&=\frac{\sum_{i=1,i\neq p}^{k}\sum_{j=1}^{k}\partial a_{i,j}\beta_i\beta_j}{\partial\beta_p}+\frac{\sum_{j=1}^{k}\partial a_{p,j}\beta_p\beta_j}{\partial\beta_p}\\
&=\frac{\sum_{i=1,i\neq p}^{k}\sum_{j=1,j\neq p}^{k}\partial a_{i,j}\beta_i\beta_j}{\partial\beta_p}+\frac{\sum_{i=1,i\neq p}^{k}\partial a_{i,p}\beta_i\beta_p}{\partial\beta_p}+\frac{\sum_{j=1,j\neq p}^{k}\partial a_{p,j}\beta_p\beta_j}{\partial\beta_p}+\frac{\partial a_{p,p}\beta_p\beta_p}{\partial\beta_p}\\
&=0+\sum_{i=1,i\neq p}^{k}a_{i,p}\beta_i+\sum_{j=1,j\neq p}^{k}a_{p,j}\beta_j+2a_{p,p}\beta_p\\
&=\sum_{i=1}^{k}a_{i,p}\beta_i+\sum_{j=1}^{k}a_{p,j}\beta_j\\
&=(\beta^TA)_p+(A\beta)_p\\
&=(A^T\beta)_p+(A\beta)_p
\end{aligned}$$

其中$(\beta^TA)_p$ 是行向量$\beta^TA$的第$p$个分量,$(A\beta)_p$ 是列向量$A\beta$的第$p$的元素。

## 1.4、Ridge Regression

当输入因子共线性严重的时候适用岭回归，岭回归是对最小二乘回归的一种补充，它损失了无偏性，来换取高的数值稳定性，从而得到较高的计算精度。

在最小二乘回归中最小化损失函数

$$\mathcal{L}=\sum(y-(\beta_0+\sum_{i=1}^{k}\beta_ix_i))^2$$

求得$\beta$权重系数。在岭回归中最小化目标函数则是

$$\mathcal{L} = \sum(y - (\beta_0 + \sum_{i=1}^{k} \beta_i x_i))^2 + \alpha \sum_{i=1}^{k} \beta_i^2$$

通过添加对权重参数的2范数惩罚项，抑制$\beta$过大数值，即抑制响应变量对单个预测变量的敏感性。

$$\begin{aligned}
&J^{ridge} \\
&= RSS(\beta^{ridge}) \\
&= \min_{\beta} \|y - X\beta\|^2 + \lambda \|W\|_2^2 \\
&= (y - X\beta)^T (y - X\beta) + \lambda \beta^T \beta \\
&= \beta^T X^T X \beta - 2\beta^T X^T y + y^T y + \lambda \beta^T \beta
\end{aligned}$$

令$\frac{\partial J^{ridge}}{\partial \beta} = 2X^T X\beta - 2X^T y + 2\lambda\beta = 0$

则$\hat{\beta}^{ridge} = (X^T X + \lambda I)^{-1} X^T y$

对比 $\hat{\beta}^{ols} = (X^T X)^{-1} X^T y$

对$OLS$当$X$不是列满秩（列满秩：因子矩阵列线性无关）或输入因子存在较强的多重共线性，$X^T X$的行列式接近于0，即$X^T X$接近奇异。计算$(X^T X)^{-1}$误差较大。当$X^T X$的行列式接近于0时，ridge regression 将其主对角元素都加上一个$\lambda$数，可以降低矩阵奇异风险。当$\lambda$（收缩量或惩罚系数）增加时，偏差增加，方差减小。

常数项使用$L2$惩罚项将导致回归出现错误。将输入因子矩阵按列标准化之后，截距估计为$\beta_0 = \bar{y}$ 。所以通常使用ridge regression 回归的时候先对输入因子矩阵$X$ 标准化处理，并不包含截距（偏置）输入。对于分类独立预测变量（哑变量），$ridge\ regression$ 和$lass\ regression$ 同样有效。

岭回归通过对权重系数的大小施加惩罚来规范线性回归。权重系数朝着零向彼此收缩，但是，当这种情况发生时，如果预测变量没有相同的尺度，那么收缩是不公平的。两个具有不同尺度的预测变量将对惩罚项有不同的贡献，因为惩罚项是所有系数的平方和。为了避免这种问题，很多时候，独立变量的中心和缩放是为了具有方差1。但是二进制变量（哑变量）不一定代表高斯/正态分布。当将它们转换为“normal”值，其中mean = 0和std.dev = 1时，不会创建基本的正态分布，您可以应用可能不符合的假设。

# 2、张量-线性回归

## 2.1 符号

$x_i$ 标量

$x$ 列向量

$x^T$ 行向量

$\boldsymbol{X}$ 矩阵

$\boldsymbol{I}$ 单位矩阵

张量 $\boldsymbol{\mathcal{X}} = \mathbb{R}^{I_1 \times \cdots I_M}$

张量的d-mode矩阵 $\boldsymbol{mat_d(\mathcal{X}) = X_{(d)} \in \mathbb{R}^{I_d \times (I_1 \cdots I_{d-1} I_{d+1} \cdots I_M)}}$

张量向量化$\boldsymbol{vec(\mathcal{X})}$

## 2.2 Tensor Ridge Regression

矢量空间中的经典线性预测器由下式给出

- $\boldsymbol{y_i = f(x) = \langle x, \beta \rangle + b}$

这里是响应变量为标量的线性回归模型。将上述经典线性预测器从矢量空间扩展到张量空间

- $\boldsymbol{y_i = f(\mathcal{X}) = \langle \mathcal{X}, \mathcal{W} \rangle + b}$

当使用张量向量化处理如 $\boldsymbol{\langle vec(\mathcal{X}, vec(\mathcal{W})) \rangle}$的时候上述两式完全相同，然而在向量预测器中对于多维度或者多信息源的预测变量将输入数据简单的拼接成向量，容易出现过拟合和高计算复杂度的问题。尽管可以通过使用无监督维度降低算法（如PCA）处理输入因子向量，但这通常导致难以建立明晰的预测变量对响应变量的解释。这里使用张量CP分解方法，通过将权重参数张量$\boldsymbol{\mathcal{W}}$的分解为多个秩一张量相加来执行特征选择或权重系数的维数降低并同时捕获多维度预测变量之间的深层抽象关系。通过对权重张量的CP分解可以线性预测器可以改写为：

- $$\begin{aligned} y_i &= \langle \mathcal{X}, \mathcal{W} \rangle + b \\ &= \left\langle \mathcal{X}, \sum_{r=1}^{R} u_r^{(1)} \circ u_r^{(2)} \circ \cdots \circ u_r^{(M)} \right\rangle + b \\ &= \sum_{r=1}^{R} \left\langle \mathcal{X}, u_r^{(1)} \circ u_r^{(2)} \circ \cdots \circ u_r^{(M)} \right\rangle + b \\ &= \sum_{r=1}^{R} \langle \mathcal{X}, \mathcal{U}_r \rangle + b \end{aligned}$$

这里使用CP分解将权重参数张量$\boldsymbol{\mathcal{W}}$分解为R个秩一张量$\boldsymbol{\mathcal{U}_r}$，如果将这R个秩一张量看做一组坐标基，则每个秩一张量都可以视为一个坐标基（如右手坐标系的一个维度），这样对R个秩一张量的累加和就可以看做类比于空间几何投影概念，也就是样本张量$\boldsymbol{\mathcal{X}}$ 投影到具有R个坐标基的空间中。类似于$\boldsymbol{PCA}$，使用多个投影的减少沿着一个方向进行投影时发生的信息丢失。

这种方法可以有效降低估计权重参数数量， 例如输入预测变量为的形状设定为[30, 30, 10]的形状，则对应的权重张量需要估计$\boldsymbol{30 \times 30 \times 10 = 9000}$ 个权重参数（标量），使用CP分解权重张量之后，例如分解为4个坐标基（秩一张量）累加和则只需要估计$\boldsymbol{(30 + 30 + 10) \times 4 = 280}$ 个权重参数。

## 2.3 内积 Inner Product

向量$\boldsymbol{x}, \boldsymbol{w} \in \mathbb{R}^N$

- $\langle \boldsymbol{x}, \boldsymbol{w} \rangle = \sum_{i=1}^{N} x_i w_i = \boldsymbol{x}^T \boldsymbol{w}$

矩阵 $\boldsymbol{X}, \boldsymbol{W} \in \mathbb{R}^{m,n}$

- $\langle \boldsymbol{X}, \boldsymbol{W} \rangle$
  $= \sum_{i=1}^{m} \sum_{j=1}^{n} x_{i,j} w_{i,j}$
  $= vec(\boldsymbol{X})^T vec(\boldsymbol{W})$
  $= vec(\boldsymbol{W})^T vec(\boldsymbol{X})$
  $= \langle vec(\boldsymbol{X}), vec(\boldsymbol{W}) \rangle$

张量$\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{W}} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$

- $\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{W}} \rangle$
  $= \sum_{i_1=1}^{I_1} \cdots \sum_{i_M=1}^{I_M} x_{i_1,\cdots,i_M} w_{i_1,\cdots,i_M}$
  $= vec(\boldsymbol{\mathcal{X}})^T vec((\boldsymbol{W})$
  $= \langle vec(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{W}}) \rangle$

根据元素计算可得

- $\langle vec(\boldsymbol{\mathcal{X}}), vec(\boldsymbol{\mathcal{W}}) \rangle = \left\langle \boldsymbol{X}_{(d)}, \boldsymbol{W}_{(d)} \right\rangle = \langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{W}} \rangle$

## 2.4 矩阵内积与迹运算

设$\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{m,n}$， $\boldsymbol{B} = (b_{ij}) \in \mathbb{R}^{m,n}$， $\boldsymbol{C} = \boldsymbol{B}^T \boldsymbol{A} = (c_{ij}) \in \mathbb{R}^{n,n}$ 。

- $(c)_{ij} = \sum_{k=1}^{m} b_{ki} a_{kj}$
- $\mathrm{tr}(\boldsymbol{B}^T \boldsymbol{A}) = \sum_{i=1}^{n} c_{ii} = \sum_{i=1}^{n} \sum_{k=1}^{m} b_{ki} a_{ki}$

则$\mathrm{tr}(\boldsymbol{B}^T \boldsymbol{A}) = \langle \boldsymbol{A}, \boldsymbol{B} \rangle = \mathrm{tr}((\boldsymbol{B}^T \boldsymbol{A})^T) = \mathrm{tr}(\boldsymbol{A}^T \boldsymbol{B})$

设$\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{m,n}$， $\boldsymbol{B} = (b_{ij}) \in \mathbb{R}^{m,n}$， $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{B}^T = (c_{ij}) \in \mathbb{R}^{m,m}$ 。

- $(c)_{ij} = \sum_{k=1}^{n} b_{ki} a_{kj}$
- $\mathrm{tr}(\boldsymbol{A}\boldsymbol{B}^T) = \sum_{i=1}^{m} c_{ii} = \sum_{i=1}^{m} \sum_{k=1}^{n} b_{ki} a_{ki}$

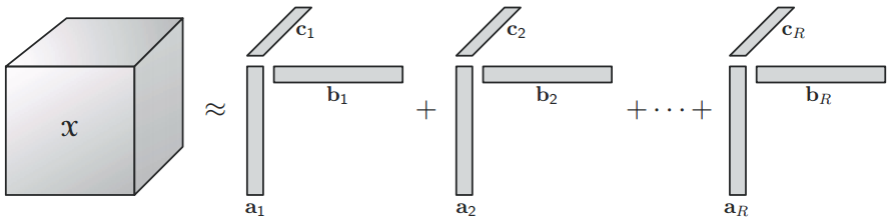
则$\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \mathrm{tr}(\boldsymbol{B}^T \boldsymbol{A}) = \mathrm{tr}((\boldsymbol{B}^T \boldsymbol{A})^T) = \mathrm{tr}(\boldsymbol{A}^T \boldsymbol{B})$
$= \mathrm{tr}(\boldsymbol{A}\boldsymbol{B}^T) = \mathrm{tr}((\boldsymbol{A}\boldsymbol{B}^T)^T) = \mathrm{tr}(\boldsymbol{A}\boldsymbol{B}^T)$

## 2.5 Frobenius norm

- $\|\boldsymbol{A}\| = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{m} \|a_{i,j}\|^2}$
- $\|\boldsymbol{A}\|^2 = \sum_{i=1}^{m} \sum_{j=1}^{m} \|a_{i,j}\|^2 = \langle \boldsymbol{A}, \boldsymbol{A} \rangle = trace(\boldsymbol{A}\boldsymbol{A}^T)$

## 2.6 CP分解

**canonical polyadic decomposition**




- $\mathcal{W}$
  $\approx [[U^{(1)}, U^{(2)}, \cdots U^{(M)}]]$
  $\triangleq \sum_{r=1}^{R} u_r^{(1)} \circ u_r^{(2)} \circ \cdots \circ u_r^{(M)}$

其中 $U^{(K)} = [u_1^{(k)}, \cdots, u_r^{(k)}]$ 对张量$\mathcal{W}$的$mode$ d展开矩阵$W_{(d)}$，CP分解转换为：

- $W_{(d)} = U^{(d)} U^{(-d)T}$

其中

- $U^{(d)} \in \mathbb{R}^{I_d \times R}$
- $U^{(-d)} \in \mathbb{R}^{(I_1 \cdots I_{d-1} I_{d+1} \cdots I_M) \times R} = \mathbb{R}^{S \times R}$
- $U^{(-k)} = (U^M \odot \cdots \odot U^{(d+1)} \odot U^{(d-1)} \odot \cdots \odot U^{(1)})$

## 2.7 TRR目标函数

对一组给定的有标签训练集$\{\mathcal{X}_i, y_i\}_{i=1}^N$ 其中$\mathcal{X}_i = \mathbb{R}^{I_1 \times \cdots I_M}$ 为$M-mode$ 张量，$y_i$ 为对应的标量标签。评估权重参数的目标函数可以写作

- $\mathcal{L}(\mathcal{W}, b) = \frac{1}{2} \sum_{i=1}^{N} l(y_i, f(\mathcal{X}_i, \Theta)) + \frac{1}{2} \psi(\Theta)$

$l(\cdot)$ 偏差损失函数，$\psi(\cdot)$ 惩罚函数。当使用响应变量和标签差最小平方和作为损失函数的时候就是张量形式的最小二乘估计，添加2范数作为惩罚函数便得到张量形式的岭回归。

使用CP分解将权重参数张量转换为一组矩阵

- $\mathcal{W} \approx [[U^{(1)}, U^{(2)}, \cdots U^{(M)}]]$

则目标函数转换为

- $\mathcal{L}(\mathcal{W}, b)$
  $= \mathcal{L}(\{U^{(1)}, \cdots, U^{(M)}\}, b) + \Phi(\mathcal{W})$
  $= \frac{1}{2} \sum_{i=1}^{N} (y_i - \langle \mathcal{X}_i, [[U^{(1)}, \cdots, U^{(M)}]] \rangle - b)^2$
  $+ \frac{\lambda}{2} \| [[U^{(1)}, \cdots, U^{(M)}]] \|_{Fro}^2$

使用坐标下降法最小化这个目标函数，在每次迭代中，固定除了$U^{(j)}$之外的其他权重矩阵，解决相对于权重参数矩阵集的一个子集$U^{(j)}$的凸优化问题。在每次迭代中，求解与模式投影相关联的权重张量分解参数矩阵$U^{(j)}$，同时保持其它模式的投影对应的参数矩阵$\{U^{(k)}\}|_{k=1,k\neq j}^{M}$固定。也就是在每迭代步最小化如下子目标函数：

- $$\begin{aligned}&\mathcal{L}_d(U^{(d)},b)\\&=l_d(U^{(d)},b)+\Phi_d(U^{(d)})\\&=\frac{1}{2}\sum_{i=1}^{N}(y_i-\langle X_{i(d)},W_{(d)}\rangle-b)^2+\frac{\lambda}{2}\|W_{(d)}\|_{Fro}^2\\&=\frac{1}{2}\sum_{i=1}^{N}(y_i-\langle X_{i(d)},U^{(d)}U^{(-d)T}\rangle-b)^2\\&+\frac{\lambda}{2}\langle U^{(d)}U^{(-d)T},U^{(d)}U^{(-d)T}\rangle\\&=\frac{1}{2}\sum_{i=1}^{N}(y_i-\langle U^{(d)}U^{(-d)T},X_{i(d)}\rangle-b)^2\\&+\frac{\lambda}{2}\langle U^{(d)}U^{(-d)T},U^{(d)}U^{(-d)T}\rangle\\&=\frac{1}{2}\sum_{i=1}^{N}(y_i-\mathrm{Tr}(U^{(d)}U^{(-d)T}X_{i(d)}^T)-b)^2\\&+\frac{\lambda}{2}\mathrm{Tr}(U^{(d)}U^{(-d)T}U^{(-d)}U^{(d)T})\end{aligned}$$

其中权重参数张量和CP分解矩阵关系为 $W_{(d)}=U^{(d)}U^{(-d)T}$

将$\mathcal{L}$ 中的 $L2-norm$ 范数惩罚项按$mode-d$ 模式矩阵拆分，即将

- $\Phi(\mathcal{W})=\frac{\lambda}{2}\|[[U^{(1)},\cdots,U^{(M)}]]\|_{Fro}^2$

修改为

- $\Phi(\mathcal{W})=\frac{\lambda}{2}\sum_{d=1}^{M}\|U^{(d)}\|_{Fro}^2$
- $\frac{\lambda}{2}\|U^{(d)}\|_{Fro}^2=\frac{\lambda}{2}\|vec(U^{(d)})\|^2$

则

$$\begin{aligned}&\mathcal{L}_d(U^{(d)},b)\\&=\frac{1}{2}\sum_{i=1}^{N}(y_i-\mathrm{Tr}(U^{(d)}\tilde{X}_{i(d)}^T)-b)^2+\frac{\lambda}{2}\|U^{(d)}\|_{Fro}^2\\&=\frac{1}{2}\sum_{i=1}^{N}(y_i-vec(\tilde{X}_{i(d)})^Tvec(U^{(d)})-b)^2+\frac{\lambda}{2}\|vec(U^{(d)})\|^2\\&=\frac{1}{2}\|y-[vec(\tilde{X}_{1(d)}),\cdots,vec(\tilde{X}_{N(d)})]^Tvec(U^{(d)})-N*b\|^2+\frac{\lambda}{2}\|vec(U^{(d)})\|^2\\&=\frac{1}{2}\|y-[b,vec(\tilde{X}_{1(d)}),\cdots,vec(\tilde{X}_{N(d)})]^T[1,vec(U^{(d)})^T]^T\|^2+\frac{\lambda}{2}\|vec(U^{(d)})\|^2\\&=\frac{1}{2}\|y-\Phi\hat{\beta}^{(d)}\|^2+\frac{\lambda}{2}\|\hat{\beta}^{(d)}\|^2\end{aligned}$$

注：为了获得目标函数坐标下降法的封闭解，这里添加了对常数偏置项的惩罚，这样获得的偏置系数估计是不准确的，在使用的时候需要额外的处理。其中

$$\Phi=[vec(\tilde{X}_{1(d)}),\cdots,vec(\tilde{X}_{N(d)})]^T\in\mathbb{R}^{N\times I_dR}$$

$\hat{\beta}^{(d)} = [1, vec(U^{(d)})^T]^T \in \mathbb{R}^{I_d R+1}$

$U^{(d)} \in \mathbb{R}^{I_d \times R}$

$U^{(-d)} \in \mathbb{R}^{S_d \times R}$

$X_{i(d)} U^{(-d)} \in \mathbb{R}^{I_d \times S_d}$

$\tilde{X}_{i(d)} = X_{i(d)} U^{(-d)} \in \mathbb{R}^{I_d \times R}$

这样关于权重参数张量的对其使用CP分解的矩阵的目标函数就转换为标准的岭回归问题了，则

- $\hat{\beta}^{(d)} = (\Phi^T \Phi + \lambda I)^{-1} \Phi^T y$
- $U^{(d)} = reshape(\hat{\beta}^{(d)}, [-1, R])$
- 使用$[[U^{(1)}, U^{(2)}, \cdots U^{(M)}]]$重构$\mathcal{W}$

在不使用惩罚项修正权重参数的时候，即张量形式的最小二乘回归中偏置$b$直接使用上式得出，在使用$L2-norm$修正权重参数的时候由于偏置权重参数同时被修正，这导致常数偏置项的错误。所以TRR的常数偏置应该使用无偏置回归的残差平均值获得，也就是使用无偏置回归模型的残差平均值计算偏置，即：

- $b = \frac{1}{N}(y_i - \langle \mathcal{X}, \mathcal{W} \rangle)$

## 3、Global Dimensionality Reduction

### 3.1 符号

$x_i$ 标量

$x$ 列向量

$x^T$ 行向量

$X$ 矩阵

$I$ 单位矩阵

张量 $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$

### 3.2 Tensor Transformation

Tucker Decomposition

- $\mathcal{X}_i = \mathcal{C}_i \times_1 U_1^i \times_2 U_2^i \times U_3^i$

其中

$\mathcal{X}_i \in \mathbb{R}^{I_1 \times I_2 \times I_3}$

$\mathcal{C}_i \in \mathbb{R}^{J_1 \times J_2 \times J_3}$

$U_k^i \in \mathbb{R}^{I_k \times R_k}$

并假设每个因子矩阵$U_k^i$描述一个信息源或信息模式，如企业特定、事件特定和情绪特定。核心张量$\mathcal{C}_1$表示由张量$\mathcal{X}_i$三中模型之间的相关程度。在Tucker分解之后，最小化如下目标函数获得用于修正$U_k^i$相关矩阵$V_k \in \mathbb{R}^{I_k \times J_k}$

- $\min\limits_{V_k} J(V_k) = \frac{\sum\limits_{i=1}^{N}\sum\limits_{j=i}^{N} \|V_k^T U_k^i - V_k^T U_k^j\|^2 w_{i,j}}{\sum\limits_{i=1}^{N} \|V_k^T U_K^i\|^2 d_{i,i}}$

其中 $W$是一个加权的上三角矩阵，捕获张量序列$\mathcal{X}_i$的多维关联，其中元素$w_{i,j}$表征两个相似收益率模式的训练集：

- $w_{i,j} = \begin{cases} 1, \ if \ \ i \le j \ and \ \|y_i - y_j\| \le 5\% \\ 0, \ otherwise \end{cases}$

在Tucker分解中，因子矩阵$U_k^i$仅保留张量$\mathcal{X}_i$内的各种信息模式之间的固有关联。 为了捕获各种模式间的动态关系，对目标函数$J(V_k)$进行优化，以确定校正因子$V_k$以调整$U_i^k|_{i=1}^N$。目标函数试图降低相似收益率的因子张量的$mode\ k$因子矩阵差异，同时最大程度保留$mode\ k$因子矩阵的原始信息。
$\|V_k^T U_k^i - V_k^T U_k^j\|^2 w_{i,j}$ 通过最小化具最大相似程度(95%)因子矩阵$U_k^i$ 和$U_k^j$ 的获得修正张量分解因子矩阵的修正矩阵$V_k^T$。使用$w_{i,j}$ 定义收益率相似股票$y_i$ 和 $y_j$之间的差异来决断因子矩阵$U_k^i$ 和$U_k^j$ 之间的差异。为了避免过度修正特性模式因子矩阵$U_k^i$ 这里最大化因子矩阵的方差。

- $var(x) = \sum (x_i - \mu)^2 p_i$
- $\mu = \sum x_i p_i$

可以使用图谱论**(Spectral Graph Theory)** 从对角矩阵$D$估计概率$p_i$ 最大化方差可以进行如下改写，即目标函数的分母。

- $var(V_k^T U_k) = \sum\limits_{i=1}^{N} \|V_k^T U_K^i\|^2 d_{i,i}$

假设$V_k^T U_k$是张量子空间中的一个均值为0随机变量矩阵，令$A^i = V_k^T U_k^i$ 则信息模式$mode\ k$的目标函数改写为：

- $$\begin{aligned} J(V_k) &= \frac{\sum\limits_{i=1}^{N}\sum\limits_{j=i}^{N} \|A^i - A^j\|^2 w_{i,j}}{\sum\limits_{i=1}^{N} \|A^i\|^2 d_{i,i}} \\ &= \frac{\sum\limits_{i=1}^{N}\sum\limits_{j=i}^{N} trace((A^i - A^j)(A^i - A^j)^T) w_{i,j}}{\sum\limits_{i=1}^{N} trace(A^i A^{iT}) d_{i,i}} \\ &= \frac{\sum\limits_{i=1}^{N}\sum\limits_{j=i}^{N} trace(A^i A^{iT} + A^j A^{jT} - A^i A^{jT} - A^j A^{iT}) w_{i,j}}{\sum\limits_{i=1}^{N} trace(A^i A^{iT}) d_{i,i}} \\ &= \frac{trace(\sum\limits_{i=1}^{N} A^i A^{iT} d_{i,i} - \sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N} A^i A^{jT} w_{i,j})}{trace(\sum\limits_{i}^{N} A A^{iT} d_{i,i})} \end{aligned}$$

其中$\|A\|^2=\sum_{i=1}^{m}\sum_{j=1}^{m}\|a_{i,j}\|^2=\langle A,A\rangle=trace(AA^T)$

令

- $D_U=\sum_{i=1}^{N}d_{i,i}U^iU^{iT}$
- $W_U=\sum_{i=1}^{N}\sum_{j=1}^{N}w_{i,j}U^iU^{jT}$

则目标函数可以改写为：

- $J(V_k)$
  $=\frac{trace(\sum_{i=1}^{N}A^iA^{iT}d_{i,i}-\sum_{i=1}^{N}\sum_{j=1}^{N}A^iA^{jT}w_{i,j})}{trace(\sum_{i}^{N}AA^{iT}d_{i,i})}$
  $=\frac{trace(\sum_{i=1}^{N}V_k^TU^iU^{iT}V_kd_{i,i}-\sum_{i=1}^{N}\sum_{j=i}^{N}V_k^TU^iU^{jT}V_kw_{i,j})}{trace(\sum_{i}^{N}V_k^TU^iU^{iT}V_kd_{i,i})}$
  $=\frac{trace(V_k^T(\sum_{i=1}^{N}d_{i,i}U^iU^{iT})V_k-V_k^T(\sum_{i=1}^{N}\sum_{j=1}^{N}w_{i,j}U^iU^{j,T})V_k)}{trace(V_k^T(\sum_{i}^{N}d_{i,i}U^iU^{iT})V_k)}$
  $=\frac{trace(V_k^TD_UV_k-V_k^TW_UV_k)}{trace(V_k^TD_UV_k)}$

添加约束条件$trace(V^TD_UV)=1$获得目标函数的唯一优化结果，这是获取因子矩阵对应的修正矩阵的优化问题转化为：

- $min\ J(V)=trace(V^TD_UV-V^TW_UV)$
- $s.t.\ trace(V^TD_UV)=1$

使用拉格朗日乘子处理这个凸优化问题：

- $L(V)=trace(V^TD_UV-V^TW_UV)-\lambda(trace(V^TD_UV)-1)$
- $\frac{dL(V_k)}{dV_k}$
  $=(V_k^T(D_U-W_U))^T+(D_U-W_U)V_k-\lambda((V^TD_U)+D_UV)$
  $=((D_U-W_U)^T+D_U-W_U)V_k-\lambda(D_U^T+D_U)^T$
  $=2(D_U-W_U)V_k-2\lambda D_UV_k$

其中$D_U^T=\sum_{i=1}^{N}d_{i,i}(U^iU^{iT})^T=\sum_{i=1}^{N}d_{i,i}U^iU^{iT}=D_U$

则修正矩阵$V_k$可以由下式得到：

- $(D_U-W_U)V_k=\lambda D_UV_k$

通过如下算法即可获得修正矩阵，通过使用Tucker分解和修正矩阵将原始输入因子矩阵体积进行压缩$\mathcal{X}\in\mathbb{R}^{I_1\times I_2\times I_3}\rightarrow\tilde{\mathcal{X}}\in\mathbb{R}^{J_1\times J_2\times J_3}$

Table III. Algorithm: Global Dimensionality Reduction of a Tensor Stream

| | |
|---|---|
| *Input*: | The training tensor stream $\mathcal{X}_i\|_{i=1}^N \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ and the associated indicators $y_i\|_{i=1}^N \in \mathbb{R}$. |
| *Output*: | The mapped tensor steam $\bar{\mathcal{X}}_i\|_{i=1}^N \in \mathbb{R}^{J_1 \times J_2 \times J_3}$, where $J_k \leq I_k$. |
| **Step 1**: | Calculate the weight matrix $\mathbf{W}$; |
| **Step 2**: | From $k = 1$ to 3 |
| **Step 3**: | From $i = 1$ to $N$ |
| **Step 4**: | Decompose the original tensor $\mathcal{X}_i$ into $\mathcal{C}_i \times_1 \mathbf{U}_1^i \times_2 \mathbf{U}_2^i \times_3 \mathbf{U}_3^i$ by Tucker decomposition; |
| **Step 5**: | End |
| **Step 6**: | $\mathbf{D}_{U_k} = \sum_{i=1}^{N} d_{i,i} \mathbf{U}_k^i \mathbf{U}_k^{iT}$; |
| **Step 7**: | $\mathbf{W}_{U_k} = \sum_{i=1}^{N} \sum_{j=i}^{N} w_{i,j} \mathbf{U}_k^i \mathbf{U}_k^{iT}$; |
| **Step 8**: | Obtain $\mathbf{V}_k$ by solving $(\mathbf{D}_{U_k} - \mathbf{W}_{U_k})\mathbf{V}_k = \lambda \mathbf{D}_{U_k} \mathbf{V}_k$; |
| **Step 9**: | End; |
| **Step 10**: | From $i = 1$ to $N$ |
| **Step 11**: | $\bar{\mathcal{X}}_i = \mathcal{C}_i \times_1 (\mathbf{V}_1^T \mathbf{U}_1^i) \times_2 (\mathbf{V}_2^T \mathbf{U}_2^i) \times_3 (\mathbf{V}_3^T \mathbf{U}_3^i)$; |
| **Step 12**: | End. |

# 引用


Li X, Zhou H, Li L. Tucker tensor regression and neuroimaging analysis[J]. arXiv preprint arXiv:1304.5637, 2013.

Petersen K B, Pedersen M S. The matrix cookbook[J]. Technical University of Denmark, 2008, 7: 15.

Li Q, Jiang L L, Li P, et al. Tensor-Based Learning for Predicting Stock Movements[C]//AAAI. 2015: 1784-1790.

Kolda T G, Bader B W. Tensor decompositions and applications[J]. SIAM review, 2009, 51(3): 455-500.

Guo W, Kotsia I, Patras I. Tensor learning for regression[J]. IEEE Transactions on Image Processing, 2012, 21(2): 816-827.

Tibshirani R. Modern regression 1: Ridge regression[C]//Data Mining. 2013, 36: 462-662.

Figueiredo M A T. Lecture Notes on Linear Regression[J].

Nel D G. On matrix differentiation in statistics[J]. South African Statistical Journal, 1980, 14(2): 137-193.

Dullemond K, Peeters K. Introduction to Tensor calculus[J]. Kees Dullemond and Kasper Peeters, 1991.

Zhao Q, Caiafa C F, Mandic D P, et al. Higher order partial least squares (HOPLS): a generalized multilinear regression method[J]. IEEE transactions on pattern analysis and machine intelligence, 2013, 35(7): 1660-1673.

Li Q, Chen Y, Jiang L L, et al. A tensor-based information framework for predicting the stock market[J]. ACM Transactions on Information Systems (TOIS), 2016, 34(2): 11.