SVD 方法应用于矩阵近似

2021 年 1 月 19 日

1 基本事实

对于任何数域 K 上的一个 $m \times n$ 矩阵 A，设 $r = \min\{m, m\}$，则总是存在矩阵 $U_{m\times r}$，对角矩阵 $D_{r\times r}$，酉矩阵[1]$V_{n\times r}$，使得

$$A = U\ D\ V^{\star} \tag{1.1}$$

式中，$V^{\star}$ 表示对矩阵 V 取共轭转置．经过整理，还能使得 D 的主对角线元素的绝对值大小依次递减．

正交矩阵[2]是特殊的酉矩阵．

2 直觉表述

设 A 是 $n \times m$ 的，对 A 做 SVD 操作，可以得到一个 $n \times r$ 的酉矩阵 U，一个 $r \times r$ 的对角矩阵 D，并且 D 的主对角元的绝对值依次递减，以及一个 $m \times r$ 的酉矩阵 V．设 $\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_r$ 是矩阵 U 的列向量，设 $d_1, d_2, \cdots, d_r$ 是矩阵 D 的对角元，设 $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_r$ 是矩阵 $V^{\top}$ 的行向量，则 A 可表为

$$A = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_r \end{bmatrix} \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ d_1 & 0 & \cdots & d_r \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_r \end{bmatrix} \tag{2.1}$$

经过计算可知

$$A = \sum_{k=1}^{r} d_k \boldsymbol{u}_k \boldsymbol{v}_k = d_1 \boldsymbol{u}_1 \boldsymbol{v}_1 + d_2 \boldsymbol{u}_2 \boldsymbol{v}_2 + \cdots + d_r \boldsymbol{u}_r \boldsymbol{v}_r \tag{2.2}$$

这样一来，我们就知道了：$\boldsymbol{u}_1\boldsymbol{v}_1, \boldsymbol{u}_2\boldsymbol{v}_2, \cdots, \boldsymbol{u}_r\boldsymbol{v}_r$ 是构成矩阵 A 的 r 个「成分」，而 $d_1, d_2, \cdots, d_r$ 则分别对应各个成分的「重要性」．我们不禁猜想：或许，去掉 A 的几个「不重要」的成分，只保留 A 的几个「较为重要」的成分，再用这些保留下来的「成分」来重新组成 A 的近似，或许误差不大？

事实上，设 k 是一个整数且满足 $1 \le k \le r$，那么我们定义

$$A(k) \overset{\text{def}}{=} \sum_{i=1}^{k} d_i \boldsymbol{u}_i \boldsymbol{v}_i \tag{2.3}$$

并且称：$A(k)$ 是 A 的一个近似．

---

[1]如果一个矩阵与自身的共轭转置做矩阵乘法运算，所得的矩阵为单位矩阵，则称这个矩阵是一个酉矩阵 (Unitary matrix)．

[2]与自身的转置做矩阵乘法运算，所得结果为单位矩阵的实矩阵，称为正交矩阵．

3 数值实验

设矩阵

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 10 & 12 & 14 & 15 \\ 3 & 5 & 7 & 11 \end{bmatrix} \tag{3.1}$$

借助于数学软件，我们求得：

$$U = \begin{bmatrix} 0.180347 & 0.279568 & -0.943036 \\ 0.864615 & -0.502165 & 0.0164804 \\ 0.468952 & 0.818335 & 0.332282 \end{bmatrix}, \quad D = \begin{bmatrix} 29.7461 & & \\ & 3.74586 & \\ & & 0.373717 \end{bmatrix} \tag{3.2}$$

以及

$$V = \begin{bmatrix} 0.344024 & -0.61056 & 0.584976 \\ 0.43975 & -0.367116 & -0.0719649 \\ 0.535476 & -0.123672 & -0.728906 \\ 0.633666 & 0.690758 & 0.348311 \end{bmatrix} \tag{3.3}$$

并且容易验证这些值满足关系式

$$A = U\ D\ V^{\mathsf{T}} \tag{3.4}$$

取 $k = 2$，那么

$$A - A(k) = 0.373717 \times \begin{bmatrix} -0.943036 \\ 0.0164804 \\ 0.332282 \end{bmatrix} \times \begin{bmatrix} 0.584976 & -0.0719649 & -0.728906 & 0.348311 \end{bmatrix} \tag{3.5}$$

$$= \begin{bmatrix} -0.206162 & 0.0253625 & 0.256887 & -0.122755 \\ 0.00360287 & -0.000443232 & -0.00448934 & 0.00214525 \\ 0.072642 & -0.00893656 & -0.0905151 & 0.0432531 \end{bmatrix} \tag{3.6}$$

再对 $A - A(k)$ 按元素平方、求和并开根号得绝对误差：

$$\sqrt{\mathrm{Sum}\ \{A\ .\hat{}\ 2\}} = 0.139665 \tag{3.7}$$

而

$$\frac{\mathrm{Sum}\ \{A\}}{12} = 7.25 \tag{3.8}$$

平均到 A 的每一个元素，只有 $0.139665/7.25 = 0.0192907$ 的绝对误差.

4 图像处理应用

既然按照 SVD 方法，矩阵可以被近似，那么同样地，将计算机中存储的图片视作矩阵，图片也可以被近似.

我们使用 Wolfram Mathematica 12.0[3]作为实验环境，首先加载图片 (图 4-1)：

```
spikey = Import["ExampleData/spikey.tiff"]
```

[3]一款数学软件

然后将图片转化为数值矩阵形式:

```
spikeyData = ImageData[spikey];
spikeyData1 = spikeyData[[;; , ;; , 1]];
spikeyData2 = spikeyData[[;; , ;; , 2]];
spikeyData3 = spikeyData[[;; , ;; , 3]];
```

值得说明的是, spikeyData 是 $155 \times 150 \times 3$ 的, 其中 155×150 对应着这张图片长宽各占 155 个与 150「像素」的长度, 而由于它又是彩色的, 所以每个像素需要用一个 3 元有序数对来确定显示的亮度和色彩等. 那么, spikeyData1, spikeyData2, 与 spikeyData3 都是 155×150 的, 分别由每个像素三元组的 1,2,3 分量构成.

图 4-1: 实验素材 spikey

这里的 spikeyData1, spikeyData2, 和 spikeyData3 分别是素材 spikey 的三个通道, 可把它们「画」出来 (图 4-2):

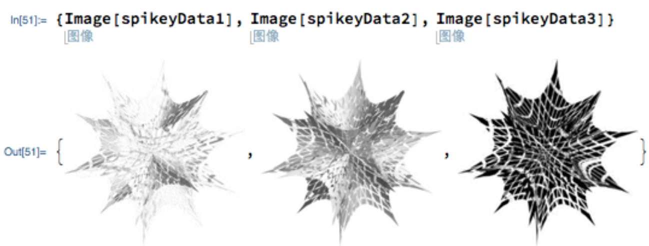

图 4-2: 实验素材 spikey 的三个通道

由于 spikeyData2 画出来效果比较明显, 所以我们就选它做 SVD, 因为这样也容易观察效果.

由于我们已经知道了怎样用 SVD 方法来近似矩阵, 所以可以直接写出代码:

```
approximate[matrix_, k_Integer] := Module[
    {matU, matD, matV},
    (
        {matU, matD, matV} = SingularValueDecomposition[matrix, k];
        matU.matD.Transpose[matV]
    )
];

Table[Image[approximate[spikeyData2, i]], {i, 5, 30, 5}]
```

效果如图 4-3:

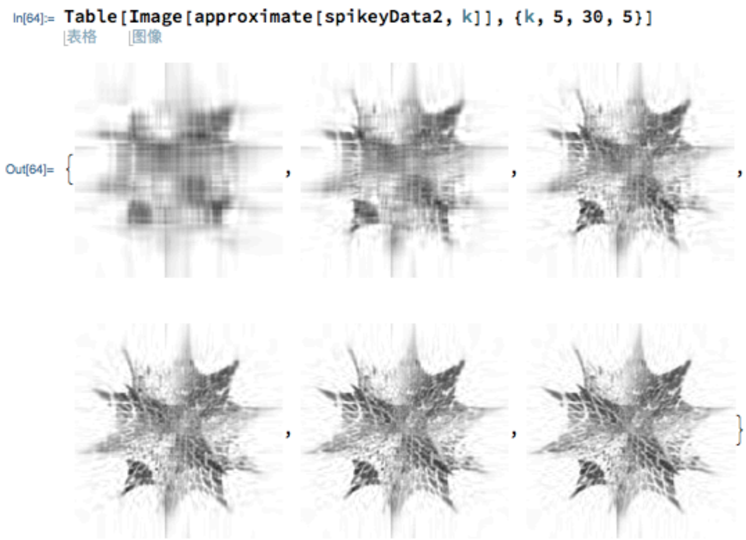

图 4-3: 分别取 $k=5,10,15,20,25,30$ 对图像数据矩阵进行近似

我们可以直观地看到：对于一个 155×150 的矩阵，k 仅仅取到 30，近似效果就已经很不错了.

5 总结

SVD 作为一种重要的数值方法在工程技术的各个领域中都有广泛应用，常见的数学软件基本上都内置了能进行 SVD 处理的函数，SVD 在统计、工程、图像处理、数值模拟、推荐系统、机器学习、文本处理等方面也都有应用. 而如今我们仅仅是将 SVD 在矩阵近似这一方面的应用展现给大家，但是还是希望感兴趣的读者坚持继续探索和思考.