

# 尝试用 ChatGPT、谷歌 Genimi 和知乎直答 学习 SVD 相关知识

首先都从询问大模型什么是矩阵 svd 分解？

我觉得大模型把定义说的都很好很对。GPT 由于没有追问选项我又问了它快速算法是什么。其他两个按照其第一个追问选项追问。

## GPT



矩阵的奇异值分解 (Singular Value Decomposition, SVD) 是一种在矩阵分解中非常重要的工具。SVD 将矩阵分解为三个矩阵的乘积，这种分解在数据降维、图像压缩和信号处理等领域有着广泛的应用。

对于一个任意的  $m \times n$  矩阵  $A$ ，SVD 可以将其表示为：

$$A = U \Sigma V^T$$

其中：

1.  $U$ ：是一个  $m \times m$  的正交矩阵，列向量被称为左奇异向量。
2.  $\Sigma$ ：是一个  $m \times n$  的对角矩阵，对角线上元素是  $A$  的奇异值，其他元素为零。奇异值按降序排列。
3.  $V^T$ ：是一个  $n \times n$  的正交矩阵，列向量称为右奇异向量。

### 主要性质

1. **奇异值**：矩阵  $\Sigma$  的对角线元素即为奇异值，是  $A$  的非负实数。
2. **秩**：矩阵  $A$  的非零奇异值的数量等于  $A$  的秩。
3. **正交矩阵**： $U$  和  $V$  都是正交矩阵，这意味着  $U^T U = I$  和  $V^T V = I$ ，其中  $I$  是单位矩阵。
4. **分解的唯一性**：若  $A$  的奇异值是唯一的，那么相应的奇异值分解也是唯一的（忽略奇异值的符号）。

### 应用

- **数据降维**：通过截取最大的奇异值，可以用较少的特征来近似原矩阵，从而减少维度。
- **图像压缩**：保留奇异值较大的部分，可以压缩图像数据而保留主要信息。
- **最小二乘拟合**：在求解方程组时，SVD 可以用于求解最小二乘解，即便方程组不满秩。

SVD 是一种极为强大的分解方法，适用于不一定是方阵的矩阵分析，具有很高的稳定性和计算效率。

GPT 现在比较喜欢偷懒不给我追问选项，于是我询问他有没有快速算法。

## 1. 随机化 SVD (Randomized SVD)

随机化 SVD 是一种基于随机投影的近似算法，它通过随机矩阵将高维数据投影到较低维空间，在保持矩阵的奇异值结构的同时减少了计算量。其基本步骤如下：

- 步骤 1: 生成一个随机矩阵  $\Omega$  (例如，高斯分布的随机矩阵)。
- 步骤 2: 计算矩阵  $Y = A\Omega$ ，将高维矩阵  $A$  转换为低维空间中的近似矩阵  $Y$ 。
- 步骤 3: 对  $Y$  进行正交分解，得到  $Q$ 。
- 步骤 4: 计算  $B = Q^T A$ ，然后对  $B$  做标准 SVD，得到近似的奇异值和向量。

随机化 SVD 是一种很高效的算法，尤其在大型稀疏矩阵中表现良好，且可以通过增加随机投影的维数来提高精度。

## 2. Lanczos 方法

Lanczos 方法是一种针对对称矩阵的迭代算法，主要用于计算大型矩阵的前  $k$  个特征值和特征向量。这一方法可以扩展到计算 SVD，因为我们可以通过对  $A^T A$  或  $AA^T$  应用 Lanczos 方法来求解前几个奇异值和奇异向量。

该方法的优点在于只需存储少量的中间结果，因此在存储和计算上都更高效。缺点是 Lanczos 方法对数值稳定性要求较高，通常需要重新正交化来保证精度。

## 3. Power 迭代法 (Power Iteration)

Power 迭代法是一种简单的迭代方法，通常用于计算矩阵的最大奇异值及其对应的奇异向量。

- 步骤:
  - 初始化一个随机向量  $x$ 。
  - 不断地计算  $y = Ax$  并将  $y$  归一化为新的  $x$ 。
  - 当  $x$  收敛时，最终得到的  $x$  将近似为最大的奇异向量，而  $Ax$  的模则是对应的最大奇异值。

Power 迭代法的缺点是只能计算出最大的奇异值和奇异向量。为了计算次大的奇异值，通常需要先去掉最大的分量，重复这个过程。

## 4. 截断 SVD (Truncated SVD)

截断 SVD 直接利用矩阵的前  $k$  个奇异值和对应的奇异向量来近似矩阵  $A$ 。这是在数据降维中常用的方法。一般步骤如下：

- 首先计算  $A$  的前  $k$  个奇异值和奇异向量。
- 构造一个较小的对角矩阵  $\Sigma_k$  和对应的  $U_k$  和  $V_k$ 。
- 最终近似矩阵  $A \approx U_k \Sigma_k V_k^T$ 。

截断 SVD 可以直接在稀疏矩阵库 (如 SciPy) 中快速计算，尤其在降维情况下极为高效。

## 5. 分块 SVD (Block SVD)

分块 SVD 是一种将大型矩阵分成小块，对每个小块分别进行 SVD，然后将结果合并的方法。这种方法可以在分布式系统或多核计算环境下进行计算，从而提高效率。

- 将矩阵  $A$  分为多个小块  $A_1, A_2, \dots, A_n$ 。
- 对每个小块  $A_i$  进行 SVD，得到  $U_i, \Sigma_i, V_i$ 。
- 将每个小块的结果聚合到全局奇异值分解中。

分块 SVD 可以有效处理超大型矩阵，但需要设计合适的合并策略以保持分解的精度。

GPT 相比于谷歌和知乎的要详尽一些，但是感觉知乎的回答更加深入易懂可能是知乎中文更好，知乎数学回答更加专业，加上该模型在内测中。

常用的快速SVD算法

- **随机SVD算法 (Randomized SVD) :**
  - **核心思想:** 通过随机投影将原始矩阵投影到一个低维空间，然后对投影后的矩阵进行SVD分解，从而得到原始矩阵的近似SVD分解。
  - **优点:** 计算速度快，适用于大规模稀疏矩阵。
- **幂法 (Power Method) :**
  - **核心思想:** 通过反复迭代的方式计算矩阵的最大奇异值和对应的奇异向量。
  - **优点:** 实现简单，适合计算最大的几个奇异值。
- **Krylov子空间方法:**
  - **核心思想:** 将矩阵投影到一个低维的Krylov子空间上，然后在这个子空间上进行SVD分解。
  - **优点:** 适用于求解部分奇异值和奇异向量。
- **分块算法:**
  - **核心思想:** 将大矩阵分成若干个小块，分别对每个小块进行SVD分解，然后将结果合并。
  - **优点:** 适合并行计算，可以充分利用多核处理器。

参考来源

1 矩阵的SVD (奇异值分解) 在算法实现中是怎样的? 有...

2 SVD分解为什么是最好的? QR分解和SVD比较? LU呢...

3 一种快速的截断式矩阵分解算法 (SVD)

查看更多 > 19 条来源

回答

是的，奇异值分解 (SVD) 有快速算法，特别是针对大型矩阵，其中随机化SVD (Randomized SVD) 是一种广泛使用的高效方法。

**随机化SVD (Randomized SVD) :**

- **优点:** 它通过使用随机投影来近似原始矩阵的范围 (即列空间)，从而减少计算复杂度。这种方法特别适合于大数据集，因为它可以利用并行计算资源，如GPU，且对数据的不精确性相对不敏感。
- **算法步骤:**
  1. **范围估计:** 首先，通过与随机生成的小矩阵相乘并进行多次迭代 (LU或QR分解)，来估计原始矩阵的范围，得到一个较小的正交矩阵  $Q$ 。
  2. **构造小矩阵并分解:** 然后，将原始矩阵投影到这个较小的维度空间，形成一个新的矩阵  $B = Q^T A$ 。这个矩阵的大小远小于原矩阵，对  $B$  进行标准SVD分解。

继续追问 最后，通过最初的正交矩阵  $Q$  和  $B$  的SVD结果来重构原矩阵的SVD，得到  $U, \Sigma, V$ 。

**Truncated SVD:** 对于需要降维的情况，通常只需要矩阵的前几个最大的奇异值和对应的奇异向量，这称为截断SVD。虽然不是一种全新的快速算法，但通过只计算前几个奇异值和向量，可以显著减少计算量。

**MATLAB中的快速实现:**

- MATLAB提供了 `svd` 函数，它专门用于部分奇异值分解，允许用户指定需要的奇异值的数量，从而在大矩阵上更高效地工作。

然后又尝试了让 GPT 给我写一个 SVD 分解的代码，他正确调用了 np 库。并在 vscode 内部的插件 cursor 让其验证一下结果的正确性，并可视化绘图出来。

```
def thermodynamic (A_reconstructed):  
  
    sns.heatmap(A_reconstructed,  
  
annot=True, cmap='coolwarm')  
  
# 定义一个矩阵 A  
  
A = np.array([ [1,1,1,1],  
  
                [0,1,0,0],  
  
                [1,1,1,0],  
  
                [0,1,1,0],  
  
                [0,1,0,0],  
  
                [1,1,0,0]])  
  
thermodynamic(A)  
  
# 使用 NumPy 的 linalg.svd 函数进行 SVD 分解  
  
U, s, Vt = np.linalg.svd(A)
```

```

tt=2

s = np.diag(s[:4-tt])

s = np.pad(s, ((0, 2+tt), (0, 2)), mode='constant')

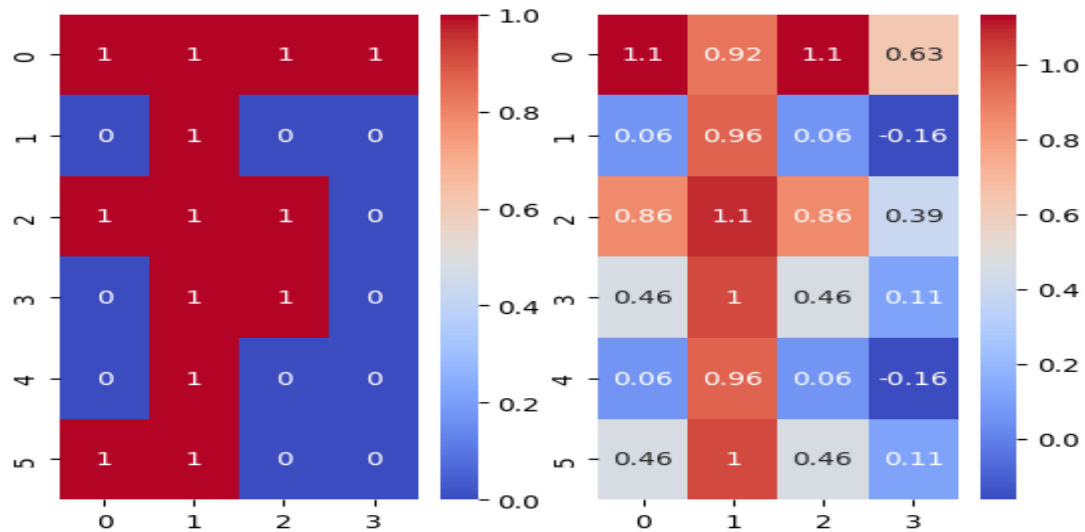
print(s)

# 验证 SVD 分解

# 重构矩阵 A，检查是否与原矩阵 A 相同

A_reconstructed = U @ s @ Vt

```



# 知乎

## 回答

奇异值分解（SVD）是一种矩阵分解方法，将一个矩阵分解为三个矩阵的乘积。具体来说，对于一个  $m \times n$  的矩阵  $A$ ，其 SVD 分解可以表示为：

$$A = U \Sigma V^T$$

其中：

- $U$  是一个  $m \times m$  的正交矩阵，其列向量是  $A$  的左奇异向量。
- $\Sigma$  是一个  $m \times n$  的对角矩阵，其对角线上的元素是  $A$  的奇异值，按从大到小的顺序排列。
- $V$  是一个  $n \times n$  的正交矩阵，其列向量是  $A$  的右奇异向量。

SVD 分解在数据压缩、图像处理、推荐系统等领域中有广泛的应用。例如，通过保留矩阵  $\Sigma$  的前  $k$  个奇异值及其对应的左奇异向量和右奇异向量，可以得到矩阵  $A$  的一个低秩近似，从而实现数据压缩。

SVD 分解的计算通常使用数值方法，如 QR 算法或分而治之算法。在实际应用中，可以使用如 NumPy、MATLAB 等数学软件包提供的 SVD 分解函数来实现。

因此，矩阵 SVD 分解是  $A = U \Sigma V^T$ 。

复制

反馈

重答

## Q 相关追问

SVD 分解与 PCA 分解之间有什么区别和联系？

矩阵 SVD 分解的数学原理是什么？

在机器学习中，SVD 分解有哪些应用场景？

很奇怪为什么没有结合知乎特点，然后我就点击了第一个追问 svd 分解和 pca 之间有什么区别和联系

# SVD 分解与 PCA 分解之间有什么区别和联系？

## 参考来源

收起

1 PCA和SVD的联系和区别？  
392 赞同 · 822 收藏

Alex · 关注  
1.1 万关注 · 4581 赞同

2 各位高手好！请求PCA与SVD的关系？求解！！  
2541 浏览

SofaSofa.io · 关注  
2631 关注 · 1597 赞同

3 机器学习中SVD和PCA一直没有搞的特别清楚，应该...  
496 赞同 · 769 收藏

重含章 · 关注  
数字等 2 个话题下的优...

查看更多 >  
20 条来源

Sofa

## 回答

奇异值分解（SVD）和主成分分析（PCA）在数据降维中都是重要的工具，它们之间既有联系也有区别。

### 联系：

- 矩阵分解基础：**PCA实际上是SVD的一种应用。当对一个中心化（即均值为零）的数据矩阵进行PCA时，其本质是通过计算该矩阵的协方差矩阵的特征值分解来完成的。但是，这个协方差矩阵的特征值分解等价于原始数据矩阵经过转置后的SVD分解中的右奇异向量和奇异值的平方根。
- 降维原理：**两者都旨在找到数据的主要方向，以减少数据维度同时保留尽可能多的信息。在PCA中，这些主要方向是协方差矩阵的特征向量，而在SVD中，用于降维的“主要方向”是右奇异向量。
- 奇异值与特征值：**在PCA中，协方差矩阵的特征值对应于数据方差，而SVD中矩阵  $\Sigma$  的奇异值是这些特征值的平方根。因此，PCA的主成分对应的方差是SVD奇异值的平方。

### 区别：

- 适用性：**SVD适用于任意大小的矩阵，不局限于方阵，而PCA通常应用于方阵或数据矩阵，要求数据先进行中心化处理。
- 直接性：**PCA直接从协方差矩阵出发，寻找方差最大的投影方向；而SVD直接对原始数据矩阵操作，但通过转换可以实现相同目的。
- 稳定性与效率：**SVD在数值稳定性上通常优于PCA，特别是在数据矩阵接近奇异时。SVD的计算方法有时更高效，尤其是在利用特定算法避免直接计算协方差矩阵时，因为协方差矩阵可能会非常大且计算密集。
- 信息获取：**PCA关注的是数据方差最大化，而SVD提供了一种更通用的分解方式，不仅用于降维，还可以用于图像压缩、噪声消除等多种应用。

总结：尽管PCA可以通过SVD来实现，它们的出发点不同。PCA侧重于数据的方差解释和降维，而SVD是一种更广泛的线性代数工具，用于矩阵的分解和重构。在实际应用中，选择哪种方法取决于具体需求，如数据的特性、计算资源和对稳定性的要求。

复制 反馈 置答

## Q 相关追问

- 为什么奇异值分解（SVD）能够用于执行主成分分析（PCA）？
- 在特征向量和奇异向量的计算中，两者有什么本质区别？
- 奇异值分解和主成分分析在数据降维中如何选择使用？

果然知乎的有点就体现出来了，大模型推给了我很多优秀的答案，比如该优秀答主的回答 (99+ 封私信 / 90 条消息) 机器学习中 SVD 和 PCA 一直没有搞的特别清楚，应该如何理解呢？ - 知乎

# 谷歌

(点赞可以直接导出为 Google 文档[唯一一个可以简便粘贴的](#))  
好的，很乐意为你讲解 SVD 分解。

## SVD 分解是什么？

奇异值分解（**Singular Value Decomposition, SVD**）是一种重要的矩阵分解方法，它可以将任意一个矩阵分解为三个矩阵的乘积：

$$A = U \Sigma V^T$$

其中：

- **A** 是要分解的原始矩阵。
- **U** 是一个左奇异矩阵，是一个  $m \times m$  的正交矩阵。
- **$\Sigma$**  是一个对角矩阵，称为奇异值矩阵，其对角线上的元素称为奇异值。
- **$V^T$**  是右奇异矩阵，是一个  $n \times n$  的正交矩阵，**V** 是  **$V^T$**  的转置。

形象地理解：

你可以把矩阵 **A** 看作是一个线性变换，它将一个向量空间中的向量映射到另一个向量空间中。**SVD** 分解就是将这个线性变换分解为三个简单的变换：

1.  **$V^T$** ：将原始向量旋转到一个新的坐标系。
2.  **$\Sigma$** ：在新的坐标系中对每个坐标轴进行缩放。
3. **U**：将缩放后的向量旋转回原来的坐标系。

## SVD 分解有什么用？

SVD 分解在很多领域都有广泛的应用，比如：

- **数据降维**：通过保留最大的奇异值对应的特征向量，可以将高维数据降到低维，同时保留大部分信息。
- **图像压缩**：利用 SVD 分解可以将图像矩阵分解为三个较小的矩阵，从而实现图像压缩。
- **推荐系统**：SVD 可以用于分析用户和物品之间的潜在关系，从而实现个性化推荐。
- **自然语言处理**：SVD 可以用于文本主题模型，发现文本中的潜在语义。

## SVD 分解的计算

SVD 分解的计算过程比较复杂，一般使用现成的数学库来实现。

基本步骤：

1. 计算  $A^TA$  和  $AA^T$  的特征值和特征向量。
2. 将  $A^TA$  的特征向量构成矩阵  $V$ ， $AA^T$  的特征向量构成矩阵  $U$ 。
3. 将  $A^TA$  或  $AA^T$  的特征值的平方根构成对角矩阵  $\Sigma$ 。

## SVD 分解的例子

假设我们有一个矩阵  $A$ ：

```
A = [[1, 2],  
      [3, 4]]
```

通过 SVD 分解，我们可以得到：

```
U = [[0.4046, -0.9149],  
      [0.9149, 0.4046]]
```

```
Σ = [[5.4649, 0],  
      [0, 0.3651]]
```

```
V^T = [[0.5760, -0.8192],  
        [0.8192, 0.5760]]
```

## 总结

SVD 分解是一种强大的数学工具，在机器学习、数据挖掘、图像处理等领域都有广泛的应用。通过理解 SVD 分解的原理和应用，可以帮助我们更好地解决实际问题。

想更深入地学习 SVD 分解，可以参考以下资源：

- 维基百科：<https://zh.wikipedia.org/zh-cn/%E5%A5%87%E5%BC%82%E5%80%BC%E5%88%86%E8%A7%A3>
- 刘建平 Pinard 博客：<https://www.cnblogs.com/pinard/p/6251584.html>

如果你还有其他问题，欢迎随时提问！



想更深入地学习，我们可以从以下方面展开：

- **SVD** 分解的几何意义
- **SVD** 分解与 **PCA** 的关系
- **SVD** 分解在推荐系统中的应用
- **SVD** 分解在自然语言处理中的应用
- **SVD** 分解的算法实现

你对哪个方面更感兴趣呢？

然后点击 **SVD 分解的几何意义**，这个问题很有助于理解 **svd** 的定义。

**SVD** 分解的几何意义

**SVD** 分解的几何意义 可以形象地理解为：将一个复杂的线性变换分解为一系列简单的几何变换。

## 核心思想

- 任意一个矩阵 都可以看作是一个线性变换，它将一个向量空间中的向量映射到另一个向量空间中。
- **SVD** 分解 则将这个复杂的线性变换分解为三个简单的变换：旋转、缩放和再旋转。

## 具体解释

- **$V^T$** : 这个变换相当于将原始向量旋转到一个新的坐标系中。这个新的坐标系是由矩阵 **A** 的右奇异向量构成的。
- **$\Sigma$** : 这个对角矩阵表示在新的坐标系中对每个坐标轴进行缩放。对角线上的元素（奇异值）表示在对应方向上的缩放比例。
- ****U****: 这个变换相当于将缩放后的向量旋转回原来的坐标系。这个新的坐标系是由矩阵 **A** 的左奇异向量构成的。

## 几何解释的意义

- **降维**: 通过选择最大的几个奇异值对应的特征向量，可以将高维数据映射到低维空间，同时保留大部分信息。这相当于在原始数据中找到最重要的几个方向。
- **数据压缩**: 通过舍弃较小的奇异值，可以对数据进行压缩，减少存储空间。
- **噪声去除**: 较小的奇异值往往对应着数据中的噪声，通过去除这些奇异值，可以达到降噪的目的。

## 形象化理解

想象一个二维平面上的圆。经过一个矩阵变换后，这个圆可能会变成一个椭圆。SVD 分解就是找到这个椭圆的长轴和短轴，以及将圆变换成椭圆所需的旋转角度。

## 总结

SVD 分解的几何意义不仅有助于我们直观地理解 SVD 分解的过程，而且为我们提供了许多实际应用的思路。通过将复杂的线性变换分解为简单的几何变换，我们可以更好地分析和处理数据。

想更深入理解，可以参考以下方面：

- **SVD 与 PCA 的关系：**SVD 和 PCA 都可以用于降维，但它们从不同的角度出发。
- **SVD 在图像处理中的应用：**如图像压缩、去噪等。
- **SVD 在推荐系统中的应用：**如基于矩阵分解的推荐算法。

是否想更深入地探讨某个方面？比如，你可以问我：

- SVD 和 PCA 在降维方面有什么区别？
- SVD 在图像压缩中是如何实现的？
- 如何用代码实现 SVD 分解？

欢迎提出你的问题！

然后发现产生了和知乎一样的建议问题 [点击 SVD 和 PCA 在降维方面有什么区别？](#)

SVD 与 PCA 的关系：相辅相成的降维利器

SVD（奇异值分解）和 PCA（主成分分析）是线性代数中两个非常重要的概念，在数据降维、特征提取等领域有着广泛的应用。尽管它们在解决问题上有所不同，但两者之间存在着非常密切的关系。

## PCA 与 SVD 的基本概念

- **PCA（主成分分析）：**是一种统计方法，通过线性变换将原始数据变换为一组线性不相关的新变量，这些新变量按照它们对数据方差的贡献由大到小的顺序排列。第一个新变量称为第一主成分，它包含原始数据中方差最大的信息。
- **SVD（奇异值分解）：**是一种矩阵分解方法，可以将任意一个矩阵分解为三个矩阵的乘积。SVD 的几何意义是将一个线性变换分解为一系列的旋转、缩放和投影。

## SVD 与 PCA 的关系

- **SVD 是 PCA 的实现基础：**PCA 的目标是找到数据协方差矩阵的特征向量，而这些特征向量恰好是 SVD 分解中右奇异向量矩阵的列向量。因此，SVD 可以为 PCA 提供一种高效的计算方法。
- **SVD 可以同时行和列进行分析：**PCA 主要关注对数据样本（行）进行降维，而 SVD 则可以同时对行和列进行分析，这在一些应用场景下具有优势。
- **SVD 可以处理稀疏矩阵：**PCA 在计算协方差矩阵时，需要对数据进行中心化，这可能会破坏数据的稀疏性。而 SVD 可以直接对稀疏矩阵进行分解，因此在处理大规模稀疏数据时具有优势。

## 具体关系

### 1. PCA 的 SVD 实现：

- 计算数据矩阵  $X$  的协方差矩阵  $C$ 。
- 对  $C$  进行 SVD 分解： $C = USV^T$ 。
- $U$  的列向量就是 PCA 的特征向量，也就是主成分。

### 2. SVD 在 PCA 中的应用

- **降维：**通过选择前  $k$  个最大的奇异值对应的右奇异向量，可以将原始数据投影到一个  $k$  维的子空间中，实现降维。
- **噪声去除：**较小的奇异值往往对应着数据中的噪声，通过去除这些奇异值，可以达到降噪的目的。
- **特征提取：**每个奇异值对应一个特征，通过分析奇异值的大小，可以了解哪些特征对数据的影响最大。

## 总结

SVD 和 PCA 是紧密相关的两种技术。SVD 为 PCA 提供了一种高效的计算方法，而 PCA 则侧重于数据的降维和特征提取。在实际应用中，我们可以根据具体的问题选择合适的方法，或者将两者结合起来使用。

## 总结表格

特点	PCA	SVD
目标	降维，提取主成分	矩阵分解
计算方法	计算协方差矩阵，求特征向量	奇异值分解
应用场景	数据降维，特征提取，图像处理	数据降维，推荐系统，自然语言处理
优势	简单直观	可以处理稀疏矩阵，同时对行和列进行分析

何时选择 **PCA**，何时选择 **SVD**？

- 当主要目的是降维，且数据是稠密的，PCA 是一个不错的选择。
- 当需要同时对行和列进行分析，或者数据是稀疏的，SVD 是一个更好的选择。

希望这个回答对您有所帮助！

如果您还有其他问题，欢迎随时提出。

谷歌的表格绘制是很好的，但是感觉相比于知乎推荐的优秀答主的回答还是没有很清晰。谷歌的建议追问比知乎多一个几何表意义，然后才问了和知乎一样的 **svd** 与 **pca** 的关系。总体而言感觉知乎大模型学习数学的效果出乎意料的好，当然大模型的代码能力也是不容忽视的。