**奇异值分解 SVD 的数学解释**

<https://blog.csdn.net/u010099080/article/details/68060274>

奇异值分解（Singular Value Decomposition，SVD）是一种矩阵分解（Matrix Decomposition）的方法。除此之外，矩阵分解还有很多方法，例如特征分解（Eigendecomposition）、LU分解（LU decomposition）、QR分解（QR decomposition）和极分解（Polar decomposition）等。这篇文章主要说下奇异值分解，这个方法在机器学习的一些算法里占有重要地位。

# 相关概念

参考自维基百科。

* **正交矩阵**：若一个方阵其行与列皆为正交的单位向量，则该矩阵为正交矩阵，且该矩阵的转置和其逆相等。两个向量正交的意思是两个向量的内积为 0
* **正定矩阵**：如果对于所有的非零实系数向量 *z*

，都有 *zTAz*>0，则称矩阵 *A*

* 是正定的。正定矩阵的行列式必然大于 0， 所有特征值也必然 > 0。相对应的，半正定矩阵的行列式必然 ≥ 0。

# 定义

下面引用 [SVD 在维基百科中的定义](https://en.wikipedia.org/wiki/Singular_value_decomposition)。

In linear algebra, the **singular value decomposition** (**SVD**) is a factorization of a real or complex matrix. It is the generalization of the eigendecomposition of a positive semidefinite normal matrix (for example, a symmetric matrix with positive eigenvalues) to any *m*×*n*

matrix via an extension of polar decomposition.

也就是说 SVD 是线代中对于实数矩阵和复数矩阵的分解，将特征分解从 半正定矩阵 推广到任意 *m*×*n*

矩阵。

**注意：本篇文章内如未作说明矩阵均指实数矩阵。**

假设有 *m*×*n*

的矩阵 *A* ，那么 SVD 就是要找到如下式的这么一个分解，将 *A*

分解为 3 个矩阵的乘积：

*Am*×*n*=*Um*×*m*Σ*m*×*nVTn*×*n*

其中，*U*

和 *V*

都是**正交矩阵** （Orthogonal Matrix），在复数域内的话就是酉矩阵（Unitary Matrix），即

*UTU*=*Em*×*m*

*VTV*=*En*×*n*

换句话说，就是说 *U*

的转置等于 *U* 的逆，*V* 的转置等于 *V*

的逆：

*UT*=*U*−1

*VT*=*V*−1

而 Σ

就是一个非负实对角矩阵。

那么 *U*

和 *V* 以及 Σ

是如何构成的呢？

# 求解

*U*

和 *V* 的列分别叫做 *A* 的 **左奇异向量**（left-singular vectors）和 **右奇异向量**（right-singular vectors），Σ 的对角线上的值叫做 *A*

的奇异值（singular values）。

其实整个求解 SVD 的过程就是求解这 3 个矩阵的过程，而求解这 3 个矩阵的过程就是求解特征值和特征向量的过程，问题就在于 **求谁的特征值和特征向量**。

* *U*

的列由 *AAT*

 的单位化过的特征向量构成

 *V* 的列由 *ATA*

 的单位化过的特征向量构成

 Σ 的对角元素来源于 *AAT* 或 *ATA*

* 的特征值的平方根，并且是按从大到小的顺序排列的

知道了这些，那么求解 SVD 的步骤就显而易见了：

1. 求 *AAT*

的特征值和特征向量，用单位化的特征向量构成 *U*

  求 *ATA* 的特征值和特征向量，用单位化的特征向量构成 *V*

  将 *AAT* 或者 *ATA* 的特征值求平方根，然后构成 Σ

# 举例

假设

*A*=⎛⎝⎜⎜⎜⎜21004300⎞⎠⎟⎟⎟⎟

那么可以计算得到

*AAT*=⎛⎝⎜⎜⎜⎜20140014100000000000⎞⎠⎟⎟⎟⎟

接下来就是求这个矩阵的特征值和特征向量了

*AATx*=*λx*

(*AAT*−*λE*)*x*=0

要想该方程组有非零解（即非零特征值），那么系数矩阵 *AAT*−*λE*

的行列式必须为 0

∣∣∣∣∣∣∣20−*λ*14001410−*λ*0000−*λ*0000−*λ*∣∣∣∣∣∣∣=0

求解这个行列式我就不再赘述了，这个直接使用行列式展开定理就可以了，可以得到 *λ*1≈29.86606875，*λ*2≈0.13393125，*λ*3=*λ*4=0

，有 4 个特征值，因为特征多项式 |*AAT*−*λE*|

是一个 4 次多项式。对应的单位化过的特征向量为

⎛⎝⎜⎜⎜⎜0.817415560.5760484400−0.576048440.817415560000100001⎞⎠⎟⎟⎟⎟

这就是矩阵 *U*

了。

同样的过程求解 *ATA*

的特征值和特征向量，求得 *λ*1≈0.13393125，*λ*2≈29.86606875

，将**特征值降序排列后**对应的单位化过的特征向量为

(0.404553580.9145143−0.91451430.40455358)

这就是矩阵 *V*

了。

而矩阵 Σ

根据上面说的为特征值的平方根构成的对角矩阵

⎛⎝⎜⎜⎜⎜5.464985700000.3659661900⎞⎠⎟⎟⎟⎟

到此，SVD 分解就结束了，原来的矩阵 *A*

就被分解成了 3 个矩阵的乘积。

*A*4×2=*U*4×4Σ4×2*VT*2×2

⎛⎝⎜⎜⎜⎜21004300⎞⎠⎟⎟⎟⎟=⎛⎝⎜⎜⎜⎜0.817415560.5760484400−0.576048440.817415560000100001⎞⎠⎟⎟⎟⎟⎛⎝⎜⎜⎜⎜5.464985700000.3659661900⎞⎠⎟⎟⎟⎟(0.404553580.9145143−0.91451430.40455358)*T*

# Numpy 实现

Python 中可以使用 numpy 包的 linalg.svd() 来求解 SVD。

import numpy as np

A = np.array([[2, 4], [1, 3], [0, 0], [0, 0]])

print(np.linalg.svd(A))

* 1
* 2
* 3
* 4

输出

(array([[-0.81741556, -0.57604844, 0. , 0. ],

[-0.57604844, 0.81741556, 0. , 0. ],

[ 0. , 0. , 1. , 0. ],

[ 0. , 0. , 0. , 1. ]]),

array([ 5.4649857 , 0.36596619]),

array([[-0.40455358, -0.9145143 ],

[-0.9145143 , 0.40455358]]))

* 1
* 2
* 3
* 4
* 5
* 6
* 7

# END