SVD

奇异值分解(Singular Value Decomposition，以下简称 SVD)是在机器学习领域广泛应用的算法，它不光可以用于 PCA 降维算法中的特征分解，还可以用于推荐系统，以及自然语言处理等领域,是很多机器学习算法的基石。

1. 回顾特征值和特征向量

首先回顾下特征值和特征向量的定义如下：

$$\mathbf{AX} = \boldsymbol{\lambda X}$$

其中 A 是一个 n*n 矩阵， X 是一个 n 维向量，则 $\boldsymbol{\lambda}$是矩阵 A 的一个特征值，而 X 是矩阵 A 的特征值 $\boldsymbol{\lambda}$ 所对应的特征向量。

求出特征值和特征向量之后，我们可以将矩阵 A 特征分解。如果我们求出了矩阵 A 的 n 个特征值 ，以及这 n 个特征值所对应的特征向量 $W_1, W_2, \ldots\ldots W_n$ ，

那么矩阵 A 就可以用下式的特征分解表示：

$$\mathrm{A} = \mathrm{W}\sum\mathrm{W}^{-1} \qquad (1)$$

其中 W 是这 n 个特征向量所张成的 n×n 维矩阵，而 Σ 为这 n 个特征值为主对角线的 n×n 维矩阵。一般我们会把 W 的这 n 个特征向量标准化,即满足$W^T \cdot W = E$，此时 W 的 n 个特征向量为标准正交基，满足$W^T = W^{-1}$ ，也就是说 W 为正交矩阵。

这样我们的特征分解表达式（1）可以写成

$$\mathrm{A} = \mathrm{W}\sum\mathrm{W}^{\mathrm{T}} \qquad (2)$$

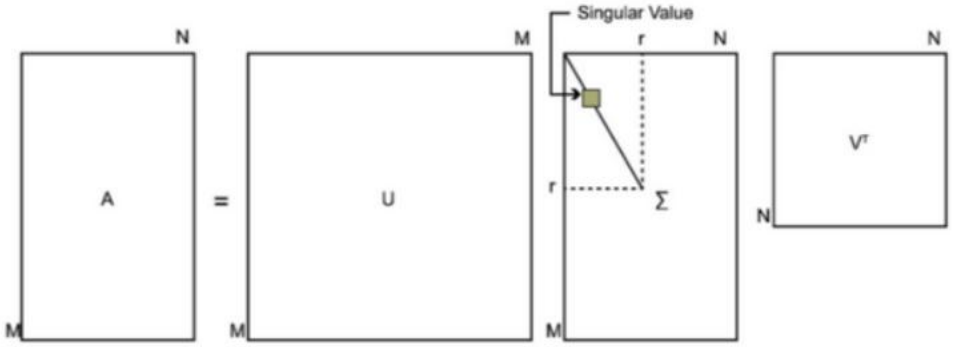
注意到要进行特征分解，矩阵 A 必须为方阵。那么如果 A 不是方阵，即行和列不相同时，我们还可以对矩阵进行分解吗？答案是可以，此时我们的 SVD 登场了。

2. SVD 的定义

SVD 也是对矩阵进行分解，但是和特征分解不同，SVD 并不要求要分解的矩阵为方阵。假设我们的矩阵 A 是一个 m×n 的矩阵，那么我们定义矩阵 A 的 SVD 为：

$$A = U\sum V^T \tag{3}$$

其中 U 是一个 m*m 的矩阵，$\sum$ 是一个 m*n 的矩阵，除了主对角线上的元素以外全为 0，主对角线上的每个元素都称为奇异值， V 是一个 n*n 的矩阵。 U 和 V 都是正交矩阵，即满足$U^T \cdot U = E$，$V^T \cdot V = E$，如下图所示



那么我们如何求出 SVD 分解后的 U,Σ,V 这三个矩阵呢？

如果我们将 A 的转置和 A 做矩阵乘法，那么会得到 n×n 的一个方阵$A^T \cdot A$ 。既然 $A^T \cdot A$ 是方阵，那么我们就可以进行特征分解，得到的特征值和特征向量满足下式：

$$(A^T A)v_i = \lambda_i v_i$$

这样我们就可以得到矩阵 $A^T \cdot A$ 的 n 个特征值和对应的 n 个特征向量 v 了。将 $A^T \cdot A$ 的所有特征向量张成一个 n×n 的矩阵 V，就是我们 SVD 公式里面的 V 矩阵了。一般我们将 V 中的每个特征向量叫做 A 的右奇异向量。

如果我们将 A 和 A 的转置做矩阵乘法，那么会得到 m×m 的一个方阵 $A \cdot A^T$ 。既然 $A \cdot A^T$ 是方阵，那么我们就可以进行特征分解，得到的特征值和特征向量满足下式：

$$(AA^T)u_i = \lambda_i u_i$$

这样我们就可以得到矩阵 $A \cdot A^T$ 的 m 个特征值和对应的 m 个特征向量 u 了。将 $A \cdot A^T$ 的所有特征向量张成一个 m×m 的矩阵 U，就是我们 SVD 公式里面的 U 矩阵了。一般我们将 U 中的每个特征向量叫做 A 的左奇异向量。

U 和 V 我们都求出来了，现在就剩下奇异值矩阵 Σ 没有求出了.由于 Σ 除了对角线上是奇异值其他位置都是 0，那我们只需要求出每个奇异值 σ 就可以了。

我们注意到（最后两步示意图可能更清晰）：

$$A = U\Sigma V^T \Rightarrow AV = U\Sigma V^T V \Rightarrow AV = U\Sigma \Rightarrow Av_i = \sigma_i u_i \Rightarrow \sigma_i = Av_i / u_i$$

这样我们可以求出我们的每个奇异值，进而求出奇异值矩阵 Σ。上面还有一个问题没有讲，就是我们说 $A^T \cdot A$ 的特征向量组成的就是我们 SVD 中的 V 矩阵，而

$A \cdot A^T$ 的特征向量组成的就是我们 SVD 中的 U 矩阵，这有什么根据吗？这个其实很容易证明，我们以 V 矩阵的证明为例。

$$A = U\Sigma V^T \Rightarrow A^T = V\Sigma U^T \Rightarrow A^T A = V\Sigma U^T U\Sigma V^T = V\Sigma^2 V^T$$

可以看出 $A^T \cdot A$ 的特征向量组成的的确就是我们 SVD 中的 V 矩阵。类似的方法可以得到 $A^T \cdot A$ 的特征向量组成的就是我们 SVD 中的 U 矩阵。
进一步我们还可以看出我们的特征值矩阵等于奇异值矩阵的平方，也就是说特征值和奇异值满足如下关系：

$$\sigma_i = \sqrt{\lambda_i}$$

这样也就是说，我们可以不用 $\frac{A \cdot v_i}{u_i}$ 来计算奇异值，也可以通过求出 $\sqrt{\lambda_i}$ 的特征值取平方根来求奇异值。

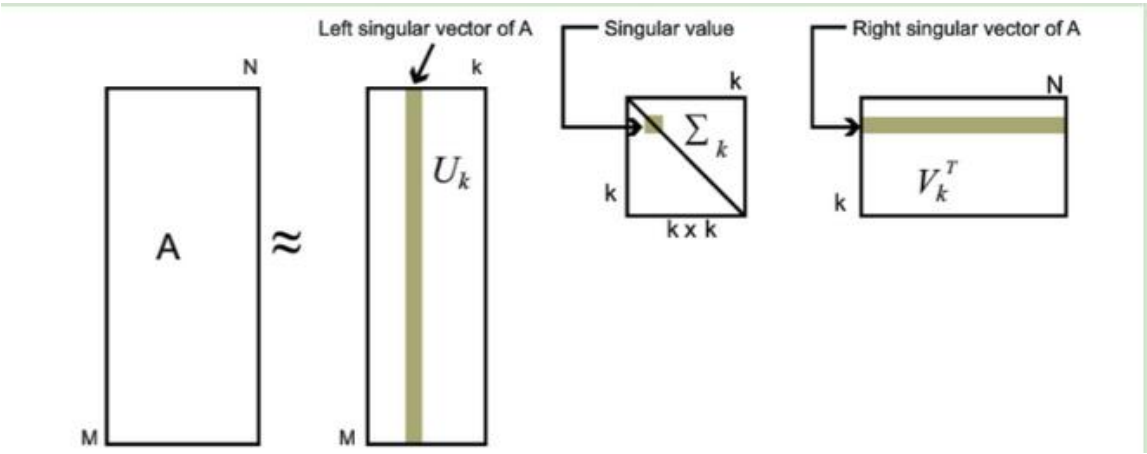
3. SVD 的特质

对于奇异值,它跟我们特征分解中的特征值类似，在奇异值矩阵中也是按照从大到小排列，而且奇异值的减少特别的快，在很多情况下，前 10%甚至 1%的奇异值的和就占了全部的奇异值之和的 99%以上的比例。也就是说，我们也可以用最大的 k 个的奇异值和对应的左右奇异向量来近似描述矩阵。

也就是说：

$$A_{m\times n} = U_{m\times m}\,\Sigma_{m\times n}\,V^T_{n\times n} \approx U_{m\times k}\,\Sigma_{k\times k}\,V^T_{k\times n}$$

其中 k 要比 n 小很多，也就是一个大的矩阵 A 可以用三个小的矩阵来表示。如下图所示，现在我们的矩阵 A 只需要灰色的部分的三个小矩阵就可以近似描述了。



由于这个重要的性质，SVD 可以用于 PCA 降维，来做数据压缩和去噪。也可以用于推荐算法，将用户和喜好对应的矩阵做特征分解，进而得到隐含的用户需求来做推荐。同时也可以用于 NLP 中的算法，比如潜在语义索引（LSI）。

4. SVD 应用——PCA

PCA 降维，需要找到样本协方差矩阵 $X^T \cdot X$ 的最大的 d 个特征向量，然后用这最大的 d 个特征向量张成的矩阵来做低维投影降维。可以看出，在这个过程中需要先求出协方差矩阵 $X^T \cdot X$ ，当样本数多样本特征数也多的时候，这个计算量是很大的。

注意到我们的 SVD 也可以得到协方差矩阵 $X^T \cdot X$ 最大的 d 个特征向量张成的矩阵，但是 SVD 有个好处，有一些 SVD 的实现算法可以不求先求出协方差矩阵 $X^T \cdot X$ ，也能求出我们的右奇异矩阵 V。也就是说，我们的 PCA 算法可以不用做特征分解，而是做 SVD 来完成。这个方法在样本量很大的时候很有效。实际上，scikit-learn 的 PCA 算法的背后真正的实现就是用的 SVD，而不是我们认为的暴力特征分解。

另一方面，注意到 PCA 仅仅使用了我们 SVD 的右奇异矩阵，没有使用左奇异矩阵，那么左奇异矩阵有什么用呢？

假设我们的样本是 m×n 的矩阵 X，如果我们通过 SVD 找到了矩阵 $X \cdot X^T$ 最大的 d 个特征向量张成的 m×d 维矩阵 U，则我们如果进行如下处理：

$$X'_{d\times n} = \Sigma_{d\times d} \cdot U^T{}_{d\times m} \cdot X_{m\times n}$$

可以得到一个 d×n 的矩阵 X ‘，这个矩阵和我们原来的 m×n 维样本矩阵 X 相比，行数从 m 减到了 k，可见对行数进行了压缩。

左奇异矩阵可以用于行数的压缩。

右奇异矩阵可以用于列数即特征维度的压缩，也就是我们的 PCA 降维。

5. SVD 小结

SVD 作为一个很基本的算法，在很多机器学习算法中都有它的身影，特别是在现在的大数据时代，由于 SVD 可以实现并行化，因此更是大展身手。

SVD 的缺点是分解出的矩阵解释性往往不强，有点黑盒子的味道，不过这不影响它的使用。

SVD 是对数据进行有效特征整理的过程。首先，对于一个 m×n 矩阵 A，我们可以理解为其有 m 个数据，n 个特征，（想象成一个 n 个特征组成的坐标系中的 m 个点），然而一般情况下，这 n 个特征并不是正交的，也就是说这 n 个特征并不能归纳这个数据集的特征。SVD 的作用就相当于是一个坐标系变换的过程，从一个不标准的 n 维坐标系，转换为一个标准的 k 维坐标系，并且使这个数据集中的点，到这个新坐标系的欧式距离为最小值（也就是这些点在这个新坐标系中的投影方差最大化），其实就是一个最小二乘的过程。进一步，如何使数据在新坐标系中的投影最大化呢，那么我们就需要让这个新坐标系中的基尽可能的不相关，我们可以用协方差来衡量这种相关性。A^T·A 中计算的便是 n×n 的协方差矩阵，每一个值代表着原来的 n 个特征之间的相关性。当对这个协方差矩阵进行特征分解之后，我们可以得到奇异值和右奇异矩阵，而这个右奇异矩阵则是一个新的坐标系，奇异值则对应这个新坐标系中每个基对于整体数据的影响大小，我们这时便可以提取奇异值最大的 k 个基，作为新的坐标，这便是 PCA 的原理。