Singular Value Decomposition (SVD) and Principal Component Analysis (PCA)

Jaehyuk Choi

March 23, 2017

Eigen(spectral) decomposition

For a matrix $\boldsymbol{A}$, eigenvalue λ_k and eigenvector $\boldsymbol{v}_k$ satisfy

$$\boldsymbol{A}\boldsymbol{v}_k = \lambda_k \boldsymbol{v}_k.$$

The matrix $\boldsymbol{A}$ can be decomposed into

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{-1},$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix with values λ_k and $\boldsymbol{Q} = (\boldsymbol{v}_1 \cdots \boldsymbol{v}_n)$, i.e., $\boldsymbol{Q}_{*j} = \boldsymbol{v}_j$.
When $\boldsymbol{A}$ is real and symmetric, $\boldsymbol{Q}$ is an orthogonal matrix, $\boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{I}$,

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T,$$

Singular Value Decomposition (SVD)

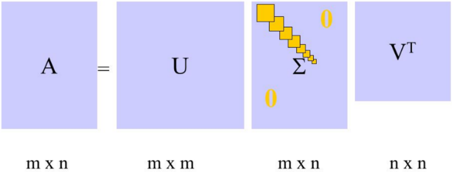
The single most useful practical concept in linear algebra:

- Any matrix (even rectangular) has a SVD.
- SVD tells everything on a matrix.

For any $m \times n$ matrix A, there is a unique decomposition:

$$A = USV^T, \quad \text{where}$$

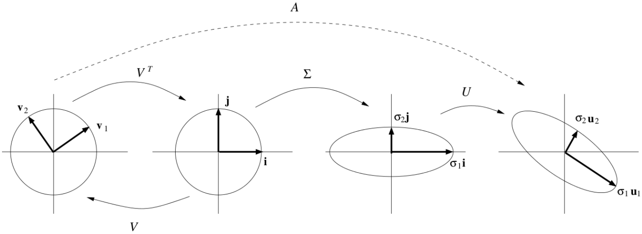
- U $(m \times m)$: orthogonal $(UU^T = U^T U = I)$
- S $(m \times n)$: diagonal. Singular values, $s_k \geq 0$, are in decreasing order for $1 \leq k \leq \min(m, n)$
- V $(n \times n)$: orthogonal $(VV^T = V^T V = I)$

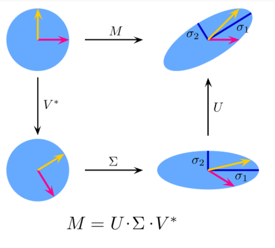


SVD: Intuition

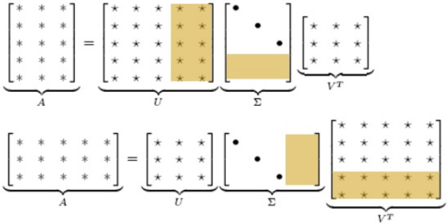
Linear transformation A is decomposed into

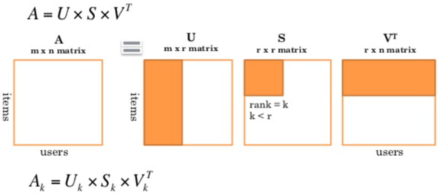
- a rotation by V^T
- a scaling by S
- a rotation by U





SVD: Compact Form, Low Rank Approximation





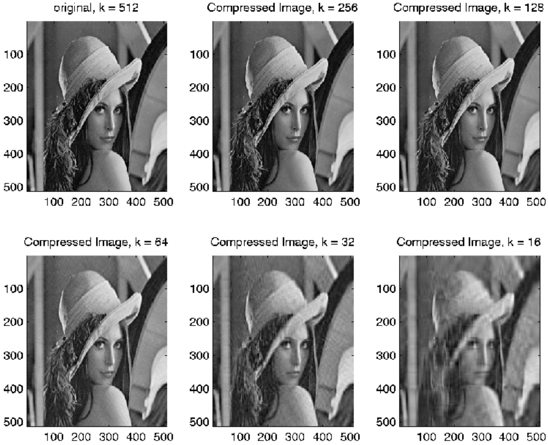
- For a non-square matrix, a compact form is enough:
  $U\ (m\times r)$, $S\ (r\times r)$, $V\ (n\times r)$ where $r=\min(m,n)$.
- If the rank is $k\ (\le r)$, $s_{j>k}=0$:
  $U\ (m\times k)$, $S\ (k\times k)$, $V\ (n\times k)$
- Using the first $j\ (\le k)$ biggest singular values,

$$A_j = U_j S_j V_j^T = \sum_{i=1}^{j} \boldsymbol{u}_i s_i \boldsymbol{v}_i^T, \quad U_j\,(m\times j),\ S_j\,(j\times j),\ V_j\,(n\times j)$$

is the best approximation with rank j minimizing the norm $\|A-A_j\|_F$

SVD: Image Compression

An image file is nothing but a matrix, so the low-rank approximation of SVD works as an image compression method. The storage is reduced from mn to $(m+n+1)k$.



Principal Component Analysis (PCA)

If $\boldsymbol{X}$ is a matrix of n samples of p features ($n \times p$), the covariance matrix is

$$\boldsymbol{\Sigma} = \frac{1}{n}\boldsymbol{X}^T\boldsymbol{X} : (p \times p) \text{ symmetric matrix}$$

The covariance matrix of the transformed space $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{W}$ is

$$\mathrm{Cov}(\boldsymbol{Z}) = \frac{1}{n}(\boldsymbol{XW})^T(\boldsymbol{XW}) = \frac{1}{n}\boldsymbol{W}^T(\boldsymbol{X}^T\boldsymbol{X})\boldsymbol{W} = \boldsymbol{W}^T\Sigma\boldsymbol{W}$$

If we pick $\boldsymbol{W}$ to be the orthogonal transformation of *SVD*, i.e., $\boldsymbol{\Sigma} = \boldsymbol{W}\boldsymbol{S}\boldsymbol{W}^T$,

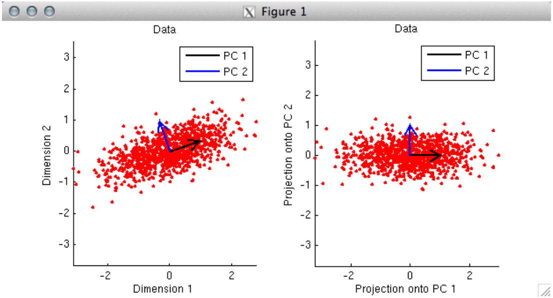
$$\mathrm{Cov}(\boldsymbol{Z}) = \boldsymbol{S} = \mathrm{diag}(S_{11}, \cdots, S_{pp}).$$

Notice that $\mathrm{Cov}(Z_i, Z_j) = \boldsymbol{W}_{*i}^T\boldsymbol{\Sigma}\boldsymbol{W}_{*j} = S_{ij}$ is zero if $i \neq j$, so the extracted features are orthogonal.

Process of finding W

Let $\boldsymbol{W} = (\boldsymbol{W}_{*1}\ \boldsymbol{W}_{*2}\ \cdots \boldsymbol{W}_{*p})$.

- Find $\boldsymbol{W}_{*1}$ such that $|\boldsymbol{W}_{*1}| = 1$ and $|\boldsymbol{W}_{*1}^T \boldsymbol{\Sigma} \boldsymbol{W}_{*1}|$ is maximized.
- Find $\boldsymbol{W}_{*2}$ such that $|\boldsymbol{W}_{*2}| = 1$, $|\boldsymbol{W}_{*2}^T \boldsymbol{\Sigma} \boldsymbol{W}_{*2}|$ is maximized and $\boldsymbol{W}_{*1}^T \boldsymbol{W}_{*2} = 0$.
- $\cdots$
- Find $\boldsymbol{W}_{*k}$ such that $|\boldsymbol{W}_{*k}| = 1$, $|\boldsymbol{W}_{*k}^T \boldsymbol{\Sigma} \boldsymbol{W}_{*k}|$ is maximized and $\boldsymbol{W}_{*k}$ is orthogonal to $\{\boldsymbol{W}_{*j}\}$ for $j < k$.



Total and Explained Variance

The total variance is the variance of all original features. Under PCA,

$$\sum_{k=1}^{p} \mathrm{Var}(X_k) = \sum_{k=1}^{p} S_{kk}.$$

Therefore the ratio

$$\frac{\sum_{j=1}^{k} S_{jj}}{\sum_{j=1}^{p} S_{jj}}$$

indicates how much of the total variance is *explained* by the first k PCA factors. Extracting features from PCA is an unsupervised learning, NOT supervised learning, because the response variable is not associated.

PCA vs Simple Linear Regression for (x, y)

PCA is not same as Simple Linear regression (OLS)!

- **Linear Regression** minimize the the (squared) distance in y-axis.
- **PCA** (1st component) minimize the (squared) shortest distance.