Numerical Methods I
Singular Value Decomposition

Aleksandar Donev
Courant Institute, NYU[1]
donev@courant.nyu.edu

[1] MATH-GA 2011.003 / CSCI-GA 2945.003, Fall 2014

October 9th, 2014

Outline

Formal definition of the SVD

Every matrix has a **singular value decomposition**

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\star} = \sum_{i=1}^{p} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\star}$$

$$[m \times n] = [m \times m]\,[m \times n]\,[n \times n]\,,$$

where **U** and **V** are **unitary matrices** whose columns are the left, $\mathbf{u}_i$, and the right, $\mathbf{v}_i$, **singular vectors**, and

$$\boldsymbol{\Sigma} = \text{Diag}\,\{\sigma_1, \sigma_2, \ldots, \sigma_p\}$$

is a **diagonal matrix** with real positive diagonal entries called **singular values** of the matrix

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0,$$

and $p = \min(m, n)$ is the maximum possible rank of the matrix.

Comparison to eigenvalue decomposition

- Recall the eigenvector decomposition for diagonalizable matrices

$$\mathbf{AX} = \mathbf{X\Lambda}.$$

- The singular value decomposition can be written similarly to the eigenvector one

$$\mathbf{AV} = \mathbf{U\Sigma}$$
$$\mathbf{A}^{\star}\mathbf{U} = \mathbf{V\Sigma}$$

and they both **diagonalize A**, but there are some important **differences**:

1. The SVD exists for any matrix, not just diagonalizable ones.
2. The SVD uses different vectors on the left and the right (different basis for the domain and image of the linear mapping represented by **A**).
3. The SVD always uses orthonormal basis (unitary matrices), not just for unitarily diagonalizable matrices.

Relation to Hermitian Matrices

- For **Hermitian (symmetric) matrices**,
$$\mathbf{X} = \pm\mathbf{U} = \pm\mathbf{V}$$
and
$$\boldsymbol{\Sigma} = |\Lambda|,$$
so there is **no fundamental difference** between the SVD and eigenvalue decompositions.
- The squared singular values are **eigenvalues of the normal matrix**:
$$\sigma_i(\mathbf{A}) = \sqrt{\lambda_i(\mathbf{A}\mathbf{A}^\star)} = \sqrt{\lambda_i(\mathbf{A}^\star\mathbf{A})}$$
since
$$\mathbf{A}^\star\mathbf{A} = (\mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^\star)(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\star) = \mathbf{V}\boldsymbol{\Sigma}^2\mathbf{V}^\star$$
is a similarity transformation.
Similarly, the singular vectors are the corresponding eigenvectors up to a sign.

Rank-Revealing Properties

- Assume the rank of the matrix is r, that is, the dimension of the range of $\mathbf{A}$ is r and the dimension of the null-space of $\mathbf{A}$ is $n - r$ (recall the fundamental theorem of linear algebra).
- The SVD is a **rank-revealing** matrix factorization because only r of the singular values are nonzero,

$$\sigma_{r+1} = \cdots = \sigma_p = 0.$$

- The left singular vectors $\{\mathbf{u}_1, \ldots, \mathbf{u}_r\}$ form an **orthonormal basis for the range** (column space, or image) of $\mathbf{A}$.
- The right singular vectors $\{\mathbf{v}_{r+1}, \ldots, \mathbf{v}_n\}$ form an **orthonormal basis for the null-space** (kernel) of $\mathbf{A}$.

The matrix pseudo-inverse

- For square non-singular systems, $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$.
  Can we generalize the matrix inverse to non-square or rank-deficient matrices?
- Yes: **matrix pseudo-inverse** (Moore-Penrose inverse):

$$\mathbf{A}^{\dagger} = \mathbf{V}\mathbf{\Sigma}^{\dagger}\mathbf{U}^{\star},$$

  where

$$\mathbf{\Sigma}^{\dagger} = \text{Diag}\left\{\sigma_1^{-1}, \sigma_2^{-1}, \ldots, \sigma_r^{-1}, 0, \ldots, 0\right\}.$$

- In numerical computations very small singular values should be considered to be zero (see homework).
- Theorem: The **least-squares solution** to over- or under-determined linear systems $\mathbf{A}\mathbf{x} = \mathbf{b}$ is

$$\mathbf{x} = \mathbf{A}^{\dagger}\mathbf{b}.$$

Proof of Least-Squares (1)

$$\min_x \|Ax - b\|_2 \leftarrow \text{LEAST SQUARES}$$

SUCH THAT $\|x\|_2$ IS MINIMAL

$$(Ax-b)^*(Ax-b) = x^*(A^*A)x - 2x^*A^*b + \ldots$$

USING SVD: $x^*A^*Ax = x^*V\Sigma^2 \underline{\underline{V^*x}}$

AND $x^*A^*b = b^*(Ax) = (U^*b)\Sigma \underline{\underline{(V^*x)}}$

DENOTING
$$\begin{cases} V^*x = w & \leftarrow \text{NEW VARIABLE} \\ U^*b = c & \leftarrow \text{CONSTANT} \end{cases}$$

(1)

Proof of Least-Squares (2)

$$\|Ax-b\|_2^2 = w^*\Sigma^2 w - 2c^*\Sigma w + \ldots$$

$$= \sum_{i=1}^{r} \sigma_i^2 |w_i|^2 - 2\sum_{i=1}^{r} (\sigma_i w_i) c_i^*$$

$$= \sum_{i=1}^{r} \left|\sigma_i w_i - c_i\right|^2 + \text{constants}$$

WHICH IS MINIMIZED IF

$$\sigma_i w_i = c_i \Rightarrow w_i = \frac{c_i}{\sigma_i}$$

or

$$\boxed{(V^*x)_i = \frac{(U^*b)_i}{\sigma_i}, \quad i \le r}$$

(2)

Proof of Least-Squares (3)

How about $w_{r+1}, \dots, w_m$?

$$\|x\|_2^2 = \|Vw\|_2^2 = w^*(V^*V)w^* = \|w\|_2^2 = \sum |w_i|^2$$

So the norm of x is minimized if the norm of w is minimized.

$$\Rightarrow \boxed{w_{r+1} = \dots = 0}$$

$$\Rightarrow x = Vw = V\Sigma^{\dagger}c = (V\Sigma^{\dagger}U^*)b = A^{\dagger}b$$

QED (3)

Sensitivity (conditioning) of the SVD

- Since unitary transformations preserve the 2-norm,

$$\|\delta\Sigma\|_2 \approx \|\delta A\|_2 .$$

- The SVD computation is always **perfectly well-conditioned**!
- However, this refers to absolute errors: The **relative error** of small singular values will be large.
- The **power of the SVD** lies in the fact that it always exists and can be computed stably...but it is **expensive to compute**.

Computing the SVD

- The SVD can be computed by performing an eigenvalue computation for the **normal matrix $\mathbf{A}^{\star}\mathbf{A}$** (a positive-semidefinite matrix).
- This squares the condition number for small singular values and is **not numerically-stable**.
- Instead, one can compute the eigenvalue decomposition of the **symmetric indefinite** $2m \times 2m$ **block matrix**

$$\mathbf{H} = \begin{bmatrix} \mathbf{0} & \mathbf{A}^{\star} \\ \mathbf{A} & \mathbf{0} \end{bmatrix}.$$

- The cost of the calculation is $\sim O(mn^2)$, of the same order as eigenvalue calculation, but in practice **SVD is more expensive**, at least for well-conditioned cases.

Reduced SVD

The **full (standard) SVD**

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\star} = \sum_{i=1}^{p} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\star}$$

$$[m \times n] = [m \times m]\,[m \times n]\,[n \times n]\,,$$

is in practice often computed in **reduced (economy) SVD** form, where $\boldsymbol{\Sigma}$ is $[p \times p]$:

$$[m \times n] = [m \times n]\,[n \times n]\,[n \times n] \quad \text{for} \quad m > n$$

$$[m \times n] = [m \times m]\,[m \times m]\,[m \times n] \quad \text{for} \quad n > m$$

This contains all the information as the full SVD but can be **cheaper to compute** if $m \gg n$ or $m \ll n$.

In MATLAB

- $[U, \Sigma, V] = svd(A)$ for **full SVD**, computed using a QR-like method.
- $[U, \Sigma, V] = svd(A,' econ')$ for **economy SVD**.
- For rank-defficient or under-determined systems the backslash operator (*mldivide*) gives a **basic solution**.
  Basic means $\mathbf{x}$ has at most r non-zeros (not unique).
- The **least-squares solution** can be computed using *svd* or *pinv* (pseudo-inverse, see homework).
- A rank-q approximation can be computed efficiently for **sparse matrices** using

$$[U, S, V] = svds(A, q).$$

Low-rank approximations

- The SVD is a decomposition into **rank-1 outer product matrices**:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\star} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\star} = \sum_{i=1}^{r} \mathbf{A}_i$$

- The rank-1 components $\mathbf{A}_i$ are called **principal components**, the most important ones corresponding to the larger σ_i.
- Ignoring all singular values/vectors except the first q, we get a **low-rank approximation**:

$$\mathbf{A} \approx \hat{\mathbf{A}}_q = \mathbf{U}_q \mathbf{\Sigma}_q \mathbf{V}_q^{\star} = \sum_{i=1}^{q} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\star}.$$

- Theorem: This is the **best approximation** of rank-q in the Euclidian and Frobenius norm:

$$\left\| \mathbf{A} - \hat{\mathbf{A}}_q \right\|_2 = \sigma_{q+1}$$

Applications of SVD/PCA

- **Statistical analysis** (e.g., DNA microarray analysis, clustering).
- Data **compression** (e.g., image compression, explained next).
- **Feature extraction**, e.g., face or character recognition (see Eigenfaces on Wikipedia).
- **Latent semantic indexing** for context-sensitive searching (see Wikipedia).
- **Noise reduction** (e.g., weather prediction).
- One example concerning language analysis given in homework.

Image Compression

```
>> A=rgb2gray(imread('basket.jpg'));
>> imshow(A);
>> [U,S,V]=svd(double(A));
>> r=25; % Rank-r approximation
>> Acomp=U(:,1:r)*S(1:r,1:r)*(V(:,1:r))';
>> imshow(uint8(Acomp));
```

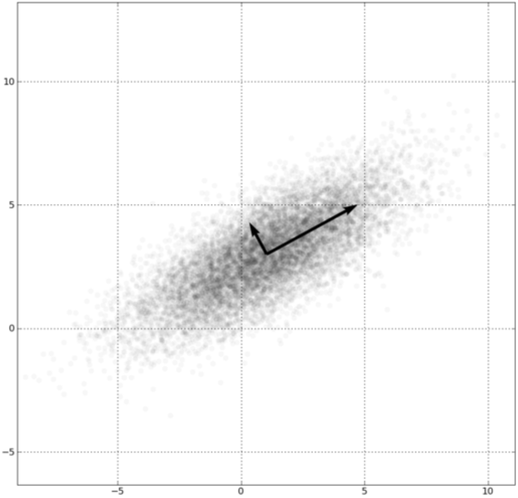
Compressing an image of a basket

We used only 25 out of the ~ 400 singular values to construct a rank 25 approximation:

Principal Component Analysis

- **Principal Component Analysis** (PCA) is a term used for low-rank approximations in statistical analysis of data.
- Consider having m empirical data points or **observations** (e.g., daily reports) of n **variables** (e.g., stock prices), and put them in a **data matrix** $\mathbf{A} = [m \times n]$.
- Assume that each of the variables has **zero mean**, that is, the empirical mean has been subtracted out.
- It is also useful to choose the units of each variable (normalization) so that the **variance is unity**.
- We would like to find an **orthogonal transformation** of the original variables that accounts for as much of the variability of the data as possible.
- Specifically, the first principal component is the direction along which the variance of the data is largest.

PCA and Variance



PCA and SVD

- The **covariance matrix** of the data tells how correlated different pairs of variables are:
$$\mathbf{C} = \mathbf{A}^T\mathbf{A} = [n \times n]$$
- The largest eigenvalue of $\mathbf{C}$ is the direction (line) that minimizes the sum of squares of the distances from the points to the line, or equivalently, **maximizes the variance** of the data projected onto that line.
- The SVD of the data matrix is $\mathbf{A} = \mathbf{U\Sigma V}^\star$.
- The eigenvectors of $\mathbf{C}$ are in fact the columns of $\mathbf{V}$, and the eigenvalues of $\mathbf{C}$ are the squares of the singular values,
$$\mathbf{C} = \mathbf{A}^T\mathbf{A} = \mathbf{V\Sigma}\left(\mathbf{U}^\star\mathbf{U}\right)\mathbf{\Sigma V}^\star = \mathbf{V\Sigma}^2\mathbf{V}^\star.$$

Note: the singular values necessarily real since $\mathbf{C}$ is positive semi-definite.

Clustering Analysis

- Given a new data point $\mathbf{x}$, we can **project** it onto the basis formed by the principal component directions as:

$$\mathbf{V}\mathbf{y} = \mathbf{x} \quad \Rightarrow \quad \mathbf{y} = \mathbf{V}^{-1}\mathbf{x} = \mathbf{V}^{\star}\mathbf{x},$$

  which simply amounts to taking dot products $y_i = \mathbf{v}_i \cdot \mathbf{x}$.
- The first few y_i's often provide a good a **reduced-dimensionality representation** that captures most of the variance in the data.
- This is very useful for **data clustering** and analysis, as a tool to understand empirical data.
- The PCA/SVD is a linear transformation and it **cannot capture nonlinearities**.

Conclusions/Summary

- The **singular value decomposition** (SVD) is an alternative to the eigenvalue decomposition that is **better for rank-defficient and ill-conditioned matrices** in general.
- Computing the SVD is **always numerically stable** for any matrix, but is typically more expensive than other decompositions.
- The SVD can be used to compute **low-rank approximations** to a matrix via the principal component analysis (PCA).
- PCA has many practical applications and usually **large sparse matrices** appear.