

Andrey Nikishaev

Jul 2, 2019 · 4 min read · Listen

Linear Algebra. SVD and PCA

Want become an expert in Computer Vision & Object Detection?

Subscribe on New Practical Course

SVD

$$\mathbf{X}_{m \times n} = \mathbf{U}_{m \times m} \mathbf{S}_{m \times n} \mathbf{V}_{n \times n}^T$$

Geometric meaning

The main idea of SVD is that matrix A is basically a linear transformation. And SVD decompose it into 3 factors Rotation(U)*Scaling/Shear(S)*Rotation(V). This gives you interesting possibilities like use it for dimension reduction(that i will show in PCA example), or for point matching in space

Nice interactive demo: <https://www.geogebra.org/m/mrey8VJX>

How to compute SVD

1. We have matrix A for which we want to compute SVD

$$\mathbf{A} = \begin{bmatrix} 4 & 0 \\ 3 & -5 \end{bmatrix}$$

2. We need to compute A.T and gram(A) = A.T * A

$$\text{Since } \mathbf{A}^T = \begin{bmatrix} 4 & 3 \\ 0 & -5 \end{bmatrix} \text{ then, } \mathbf{A}^T \mathbf{A} = \begin{bmatrix} 4 & 3 \\ 0 & -5 \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 3 & -5 \end{bmatrix} = \begin{bmatrix} 25 & -15 \\ -15 & 25 \end{bmatrix}$$

3. From gram(A) we can compute eigenvalues and singular values which will be real, cause gram(A) is NxN size.

$$\mathbf{A}^T \mathbf{A} - c \mathbf{I} = \begin{bmatrix} 25 - c & -15 \\ -15 & 25 - c \end{bmatrix} \quad \begin{array}{l} \text{characteristic equation} \longrightarrow (25 - c)(25 - c) - (-15)(-15) = 0 \\ \text{The quadratic equation gives two values.} \\ \text{In decreasing order, these are} \longrightarrow |40| > |10| \\ \text{eigenvalues} \longrightarrow c_1 = 40 \quad c_2 = 10 \end{array}$$
$$\text{singular values} \longrightarrow s_1 = \sqrt{40} = 6.3245 > s_2 = \sqrt{10} = 3.1622$$

Note that singular values should be sorted from biggest to smallest

$$s_1 \geq s_2 \geq \dots \geq s_n \geq 0$$

4. Construct S matrix

$$\mathbf{S} = \begin{bmatrix} 6.3245 & 0 \\ 0 & 3.1622 \end{bmatrix}$$

5. Now we need find matrix V. For this we need to solve equation $\mathbf{gram(A)} - c_n \mathbf{I}$ for each eigenvalue, where I — is identity matrix

$$\begin{array}{l} \text{for } c_1 = 40 \\ \mathbf{A}^T \mathbf{A} - c \mathbf{I} = \begin{bmatrix} 25 - 40 & -15 \\ -15 & 25 - 40 \end{bmatrix} = \begin{bmatrix} -15 & -15 \\ -15 & -15 \end{bmatrix} \\ (\mathbf{A}^T \mathbf{A} - c \mathbf{I}) \mathbf{x}_1 = \mathbf{0} \\ \begin{bmatrix} -15 & -15 \\ -15 & -15 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{array}{l} -15x_1 + -15x_2 = 0 \\ -15x_1 + -15x_2 = 0 \end{array} \\ \text{Solving for } x_2 \text{ for either equation: } x_2 = -x_1 \\ \mathbf{x}_1 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ -x_1 \end{bmatrix} \\ \text{Dividing by its length,} \\ L = \sqrt{x_1^2 + x_2^2} = x_1 \sqrt{2} \\ \mathbf{x}_1 = \begin{bmatrix} x_1 / L \\ -x_1 / L \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 0.7071 \\ -0.7071 \end{bmatrix} \end{array} \quad \begin{array}{l} \text{for } c_2 = 10 \\ \mathbf{A}^T \mathbf{A} - c \mathbf{I} = \begin{bmatrix} 25 - 10 & -15 \\ -15 & 25 - 10 \end{bmatrix} = \begin{bmatrix} 15 & -15 \\ -15 & 15 \end{bmatrix} \\ (\mathbf{A}^T \mathbf{A} - c \mathbf{I}) \mathbf{x}_2 = \mathbf{0} \\ \begin{bmatrix} 15 & -15 \\ -15 & 15 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{array}{l} 15x_1 + -15x_2 = 0 \\ -15x_1 + 15x_2 = 0 \end{array} \\ \text{Solving for } x_2 \text{ for either equation: } x_2 = x_1 \\ \mathbf{x}_2 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_1 \end{bmatrix} \\ \text{Dividing by its length,} \\ L = \sqrt{x_1^2 + x_2^2} = x_1 \sqrt{2} \\ \mathbf{x}_2 = \begin{bmatrix} x_1 / L \\ x_1 / L \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 0.7071 \\ 0.7071 \end{bmatrix} \end{array}$$

$$\mathbf{V} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} 0.7071 & 0.7071 \\ -0.7071 & 0.7071 \end{bmatrix}$$

$$\mathbf{V}^T = \begin{bmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \end{bmatrix}$$

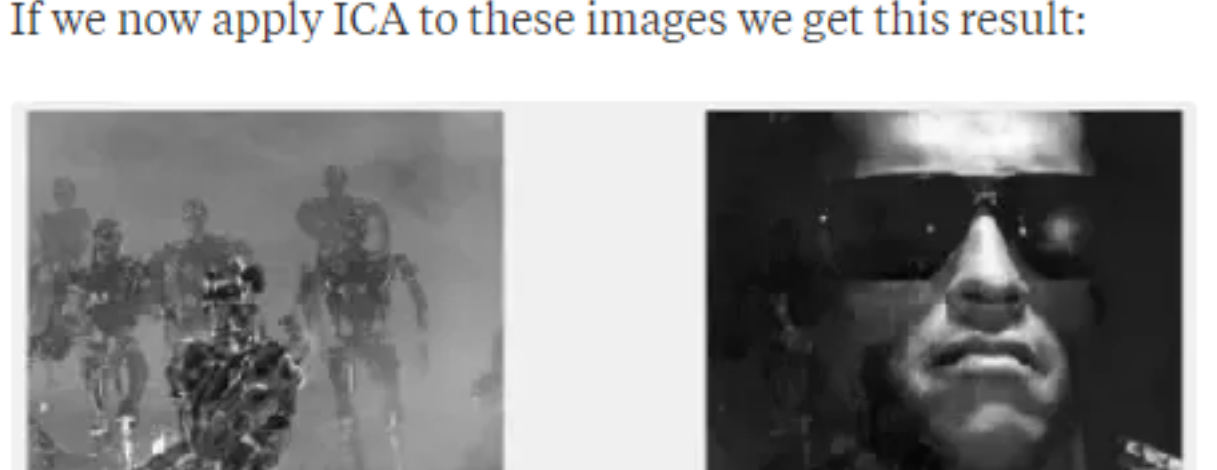
6. Now we need to find matrix U. If determinant(diag(S)) != 0 we can get U from equation $\mathbf{U} = \mathbf{A}^* \mathbf{V} * \text{inv}(\mathbf{S})$. In other situation when determinant(diag(S)) != 0 and matrix is Singular (not invertible) we can use formula $\mathbf{un} = (\mathbf{I} / \mathbf{sn}) * \mathbf{A} * \mathbf{vn}$, where vn is V[:,n] vector. $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_n]$

$$\begin{bmatrix} 0.4472 & 0.8944 \\ 0.8944 & -0.4472 \end{bmatrix}$$

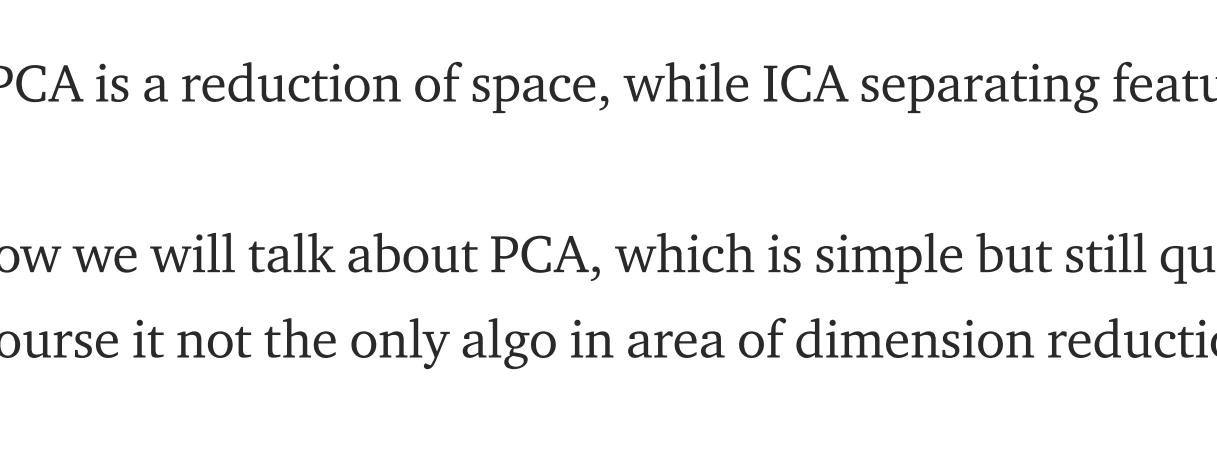
PCA

First of all i want to say that PCA has a brother ICA, and the difference between them can be shown on this image example:

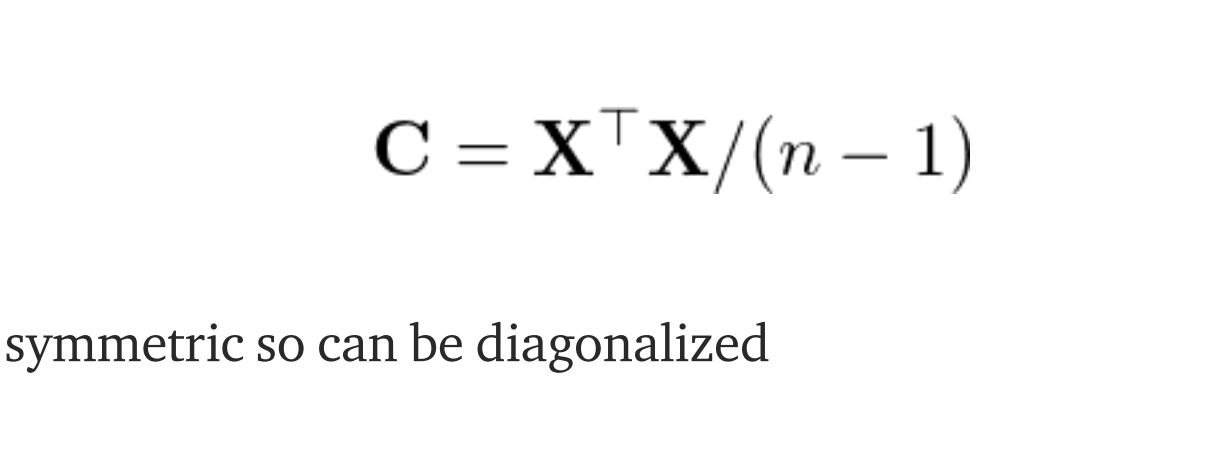
As an example of ICA consider these two images:



I mixed them in different proportions producing these two mixes:



If we now apply ICA to these images we get this result:



<https://www.quora.com/What-is-the-difference-between-PCA-and-ICA/answer/Luis-Argerich>

So idea of PCA is a reduction of space, while ICA separating features of that space. But right now we will talk about PCA, which is simple but still quite useful things(of course it not the only algo in area of dimension reduction)

How to compute PCA

So we have matrix X. First we will need center it by $\mathbf{X} / \text{std} - \text{mean}$ (by columns)

Then the covariance of X equal

$$\mathbf{C} = \mathbf{X}^T \mathbf{X} / (n - 1)$$

C matrix is symmetric so can be diagonalized

$$\mathbf{C} = \mathbf{V} \mathbf{L} \mathbf{V}^T,$$

where V — matrix of eigenvectors, L — diagonal matrix of eigenvalue. The eigenvectors are called **principal axes** and projection of the data on them **principal components (XV)**

Now lets perform SVD over X

$$\mathbf{X} = \mathbf{U} \mathbf{S} \mathbf{V}^T,$$

Pasting this into formula of covariance of X will gives us this

$$\mathbf{C} = \mathbf{V} \mathbf{S} \mathbf{U}^T \mathbf{U} \mathbf{S} \mathbf{V}^T / (n - 1) = \mathbf{V} \frac{\mathbf{S}^2}{n - 1} \mathbf{V}^T$$

where singular values are related to the eigenvalues of covariance matrix as

$$\lambda_i = s_i^2 / (n - 1)$$

Principal components are given by

$$\mathbf{T} = \mathbf{X} \mathbf{V} = \mathbf{U} \mathbf{S} \mathbf{V}^T \mathbf{V} = \mathbf{U} \mathbf{S}$$

So:

1. Use SVD on A* and get U, V and S.
2. Set new N dimension to which you want to reduce your data. And get first N values from U and S.
 $\mathbf{Ur} = \mathbf{U}[:, :n], \mathbf{Sr} = \mathbf{S}[:, n:n]$
- 3) **Reduced(A, N) = Ur * Sr**

Of course it's only one way of computing PCA, but as i know almost all current algorithms use SVD(different variations of it) for that, so it most popular.

Code

[You can run all this code in Google Colab](#)

Sources

- <https://math.berkeley.edu/~hutching/teach/54-2017/svd-notes.pdf>
- http://math.mit.edu/classes/18.095/2016IAP/lec2/SVD_Notes.pdf
- https://en.wikipedia.org/wiki/Singular_value_decomposition

Support

Become a Patron and support our community to make more interesting articles & tutorials

More cool things about Linear Algebra

Linear Algebra. Points matching with SVD in 3D space

Problem

towardsdatascience.com

Linear Algebra. Polynomials. Interpolation. Least squares

In this part of lectures we will talk about simplest "ml models" — polynomials.

towardsdatascience.com

78

Andrey Nikishaev · Jul 1, 2019

Linear Algebra. Polynomials. Interpolation. Least squares

Want become an expert in Computer Vision & Object Detection? Subscribe on New Practical Course In this part of lectures we will talk about simples...

Machine Learning · 3 min read

Post a quick thought or a long story. It's easy and free.

Write with the app

Andrey Nikishaev · Jun 30, 2019

Linear Algebra. Points matching with SVD in 3D space

Want become an expert in Computer Vision & Object Detection? Subscribe on New Practical Course Problem We need to find best rotation & translation params between two sets of points in 3D space. This type of...

Data Science · 3 min read

Andrey Nikishaev · Mar 6, 2019

Deploying on AWS Free Tier with Docker and Fabric

In this article i want to summarize all things that u will need to make good dev environment and deployment for a small application. To make this happen we will use AWS Free Tier and Docker containers and orchestrati...

Docker · 7 min read

Andrey Nikishaev · Aug 29, 2018

23 Best Popular Science YouTube Channels

One man said "To become a genius you need to learn new thing every day". So instead watching stupid TV-shows, TV-series or new SUPER DUPER BLOCKBUSTER with 3 lines of scenario, it's much better to spent this tim...

YouTube · 2 min read

Andrey Nikishaev · Apr 23, 2018

Useful snippets for Google Colaboratory. Free GPU included.

For people who are new or still didn't use it — Google Colab is Jupiter service with dedicated server(with GPU) that you get free for about 12...

Google Colaboratory · 3 min read

Read more from Machine Learning World

Upgrade

Search

Andrey Nikishaev

4.2K Followers

Machine Learning and Computer Vision Researcher. Founder LearnMLToday

Follow

Related

Machine Learning isn't Magic! How to properly train a r...

XGBoost Technique and Ensemble Methods are related? How?

Introduction to Principle Component Analysis(PCA)

Objective Metrics and Gradient Descent Algorithms for...

Help Status Writers Blog Careers Privacy Terms About Knowable