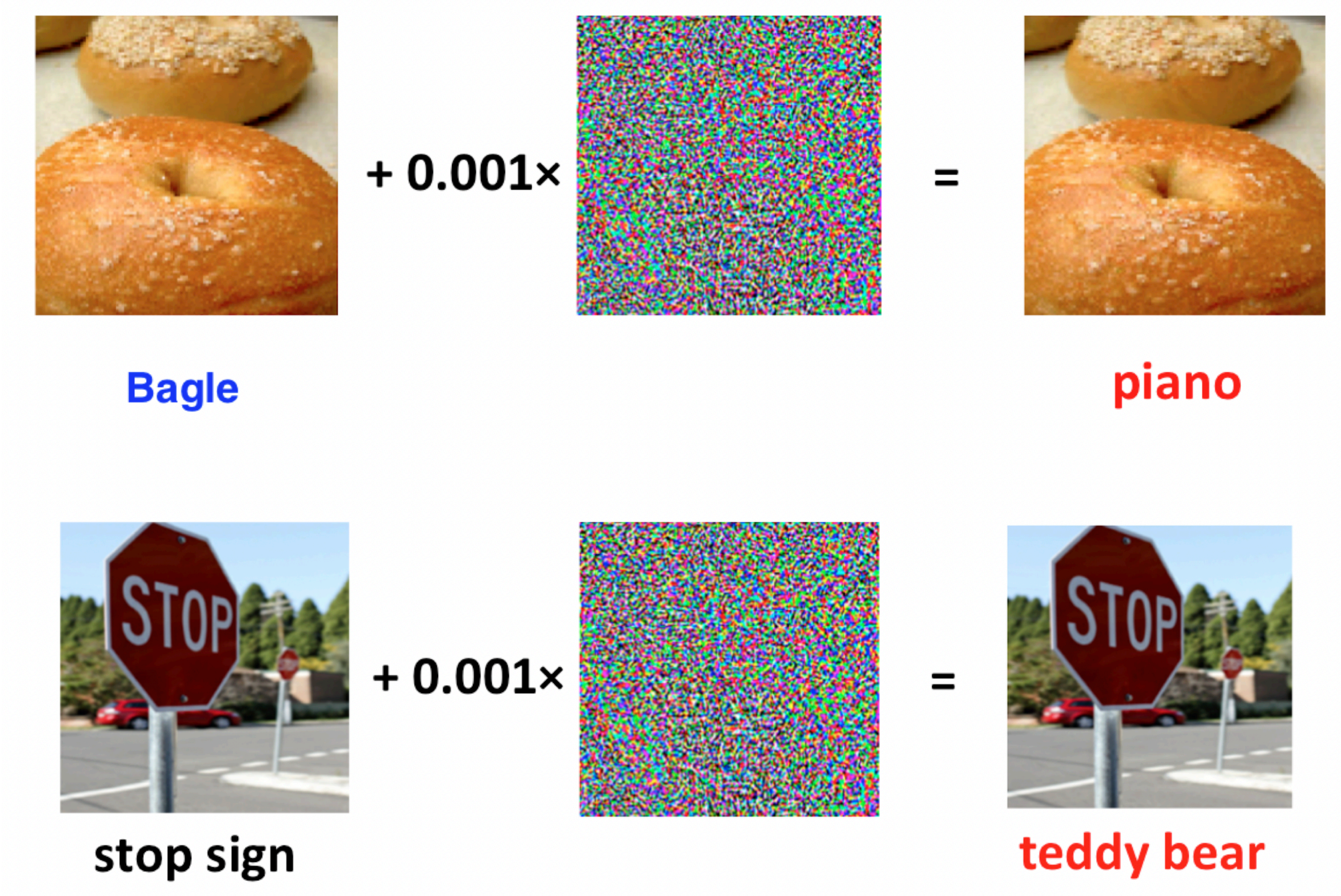
Open in Colab URLIS.NET/tm

Lipschitz constant of a convolutional layer in neural network

It was observed, that small perturbation in Neural Network input could lead to significant errors, i.e. misclassifications. Picture below from the article.



Lipschitz constant bounds the magnitude of the output of a function, so it cannot change drastically with a slight change in the input

$$\|NN(image) - NN(image + \varepsilon)\| \leq L\|\varepsilon\|$$

In this notebook we will try to estimate Lipschitz constant of some convolutional layer of a Neural Network.

размеры $h \cdot w \cdot C_{in}$

$h \cdot w \cdot C_{out}$

Power method for the singular value estimation of matrix W:

W - матрица линейного слоя $224 \times 224 \times 64 \approx$

$\approx$ много

$x_{in} \rightarrow \boxed{} \rightarrow y_{out}$

conv2d

$y_{out} = W \cdot x_{in} + b$

conv layer

(3, 224, 224) → (64, 224, 224)

conv_t_layer

(64, 224, 224) → (3, 224, 224)

$$x_{k+1} = \frac{W^\top W x_k}{\|W^\top W x_k\|}$$

$$\sigma_{k+1} = \frac{\|W x_k\|}{\|x_k\|}.$$

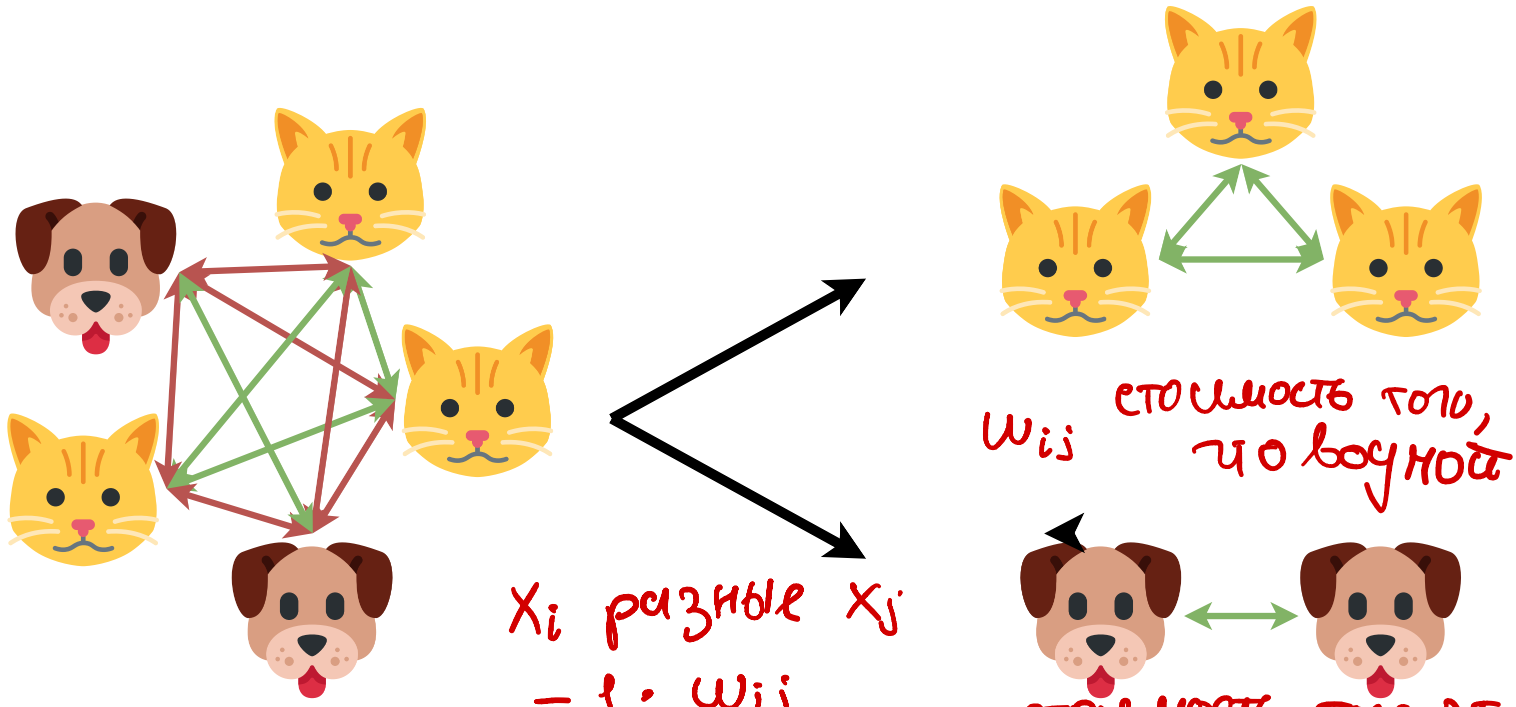
matvec

forward pass

1. $X_k \to W x_k$
2. $W^T(W x_k) = W^T W x_k$
3. норм.

Two way partitioning problem

Intuition



w_{ij} стоимость того, что вдумной

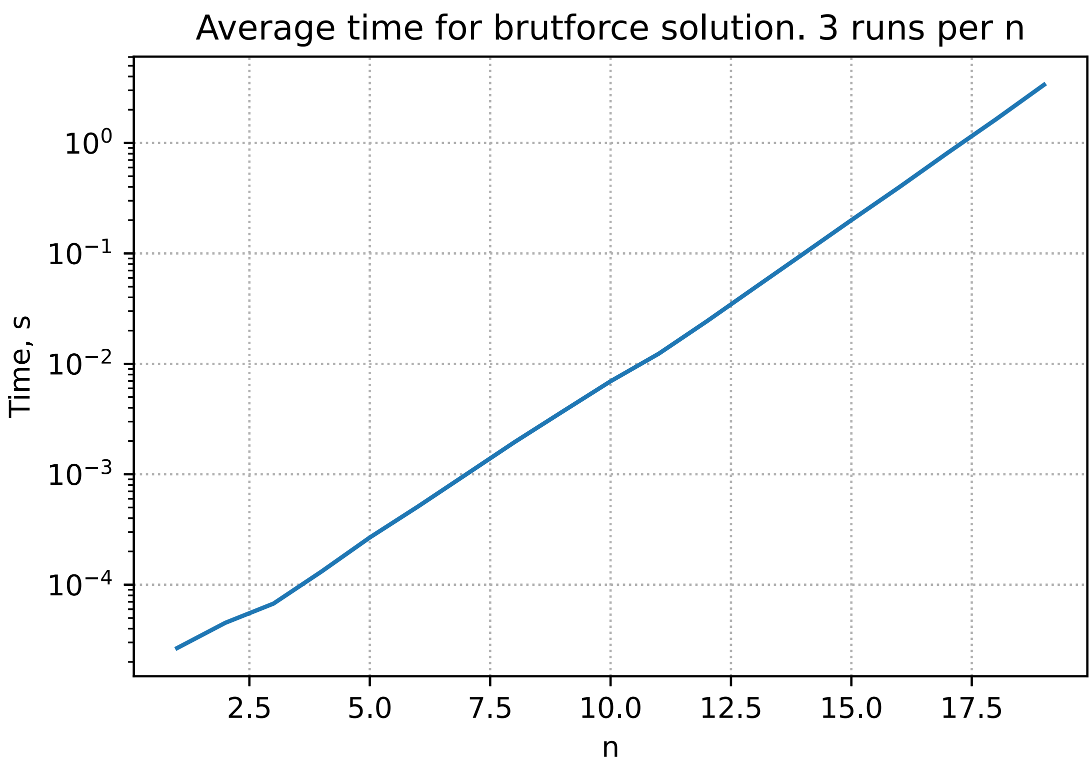
X_i разные X_j

$-1 \cdot w_{ij}$

— стоимость того, что в разных

Suppose, we have a set of n objects, which are needed to be splitted into two groups. Moreover, we have information about the preferences of all possible pairs of objects to be in the same group. this information could be presented in the matrix form: $W \in \mathbb{R}^{n \times n}$, where $\{w_{ij}\}$ is the cost of having i-th and j-th object in the same partitions. It is easy to see, that the total number of partitions is finite and eqauls to 2^n. So this problem can in principle be solved by simply checking the objective value of each feasible point. Since the number of feasible points grows exponentially, however, this is possible only for small problems (say, with $n \leq 30$). In general (and for n larger than, say, 50) the problem is very difficult to solve.

For example, bruteforce solution on MacBook Air with M1 processor without any explicit parallelization will take more, than a universe lifetime for $n = 62$.



Despite the hardness of the problems, there are several ways to approach it.

Problem

We consider the (nonconvex) problem

$$\min_{x \in \mathbb{R}^n} x^\top W x,$$
$$\text{s.t. } x_i^2 = 1,\ i = 1, \ldots, n$$

where $W \in \mathbb{R}^n$ is the symetric matrix. The constraints restrict the values of x_i to 1 or -1, so the problem is equivalent to finding the vector with components ± 1 that minimizes $x^\top W x$. The feasible set here is finite (it contains 2^n points), thus, is non-convex.

The objective is the total cost, over all pairs of elements, and the problem is to find the partition with least total cost.

Simple lower bound with duality

We now derive the dual function for this problem. The Lagrangian is

$$L(x,\nu) = x^\top W x + \sum_{i=1}^n \nu_i (x_i^2 - 1) = x^\top (W + \mathrm{diag}(\nu))x - \mathbf{1}^\top \nu.$$

We obtain the Lagrange dual function by minimizing over x:

$g(\nu) = \inf_{x \in \mathbb{R}^n} L(x, \nu)$

$\forall \nu$

двойств. функции

$p^* \geq g(\nu)$

опт. стоимость

$$g(\nu) = \inf_{x \in \mathbb{R}^n} x^\top (W + diag(\nu))x - \mathbf{1}^\top \nu = $$
$$= \begin{cases} \mathbf{1}^\top \nu, & \boxed{W + \mathrm{diag}(\nu) \succeq 0} \\ -\infty, & \text{otherwise} \end{cases}$$

$\in \mathbb{R}^n$ такие ν:

$W + \gamma \cdot I$, $\gamma \in \mathbb{R}$

$W + (-\lambda_{min}(W)) \cdot I \succeq 0$

sa

This dual function provides lower bounds on the optimal value of the difficult problem. For example, we can take any specific value of the dual variable

$$\boxed{\nu = -\lambda_{min}(W)\mathbf{1},}$$

SVD $O(n^3)$

This yields the bound on the optimal value p^*:

$$\boxed{p^* \geq g(\nu) \geq -\mathbf{1}^\top \nu = n\lambda_{min}(W)}$$

Question Can you obtain the same lower bound without knowledge of duality, but using the iddea of eigenvalues?

функции $\lambda_{max} = \rho$

спектр исх. матрицы

W:

$\lambda_{min}(W)$ 0 $\lambda_{max}(W)$

Code

Open in Colab

$W - \rho \cdot I$

степенной метод получает $\max_i |\lambda_i|$

References

$\lambda_{min}(W) - \rho$ 0 $\lambda_{max}(W) - \rho$ SHIFT

- Convex Optimization book by Stephen Boyd and Lieven Vandenberghe.

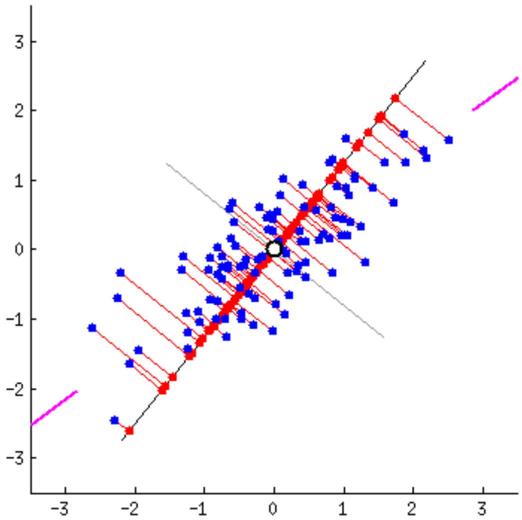
Eigenfaces

PCA recap

Intuition

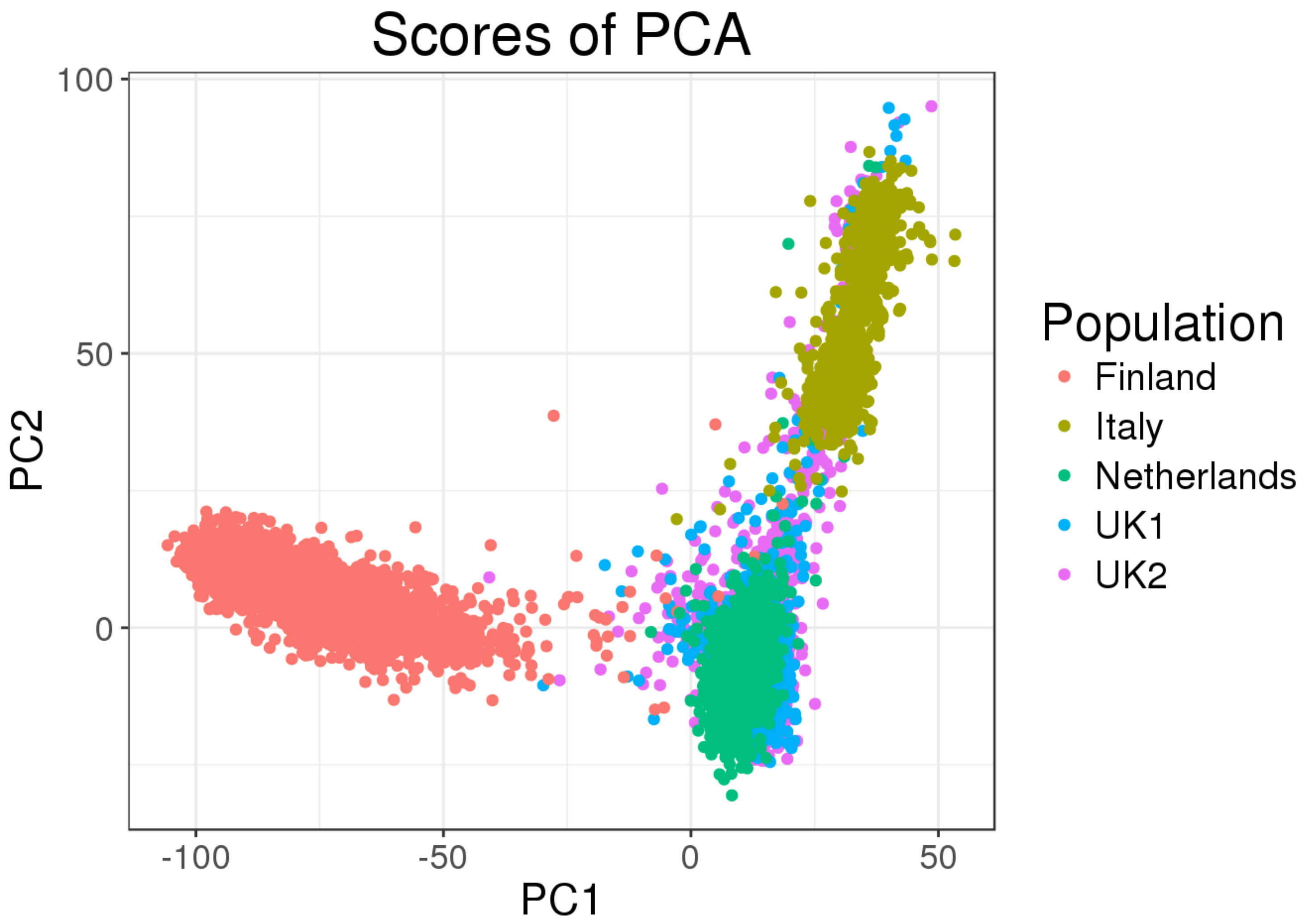
Imagine, that you have a dataset of points. Your goal is to choose orthogonal axes, that describe your data the most informative way. To be precise, we choose first axis in such a way, that maximize the variance (expressiveness) of the projected data. All the following axes have to be orthogonal to the previously chosen ones, while satisfy largest possible variance of the projections.

Let's take a look at the simple 2d data. We have a set of blue points on the plane. We can easily see that the projections on the first axis (red dots) have maximum variance at the final position of the animation. The second (and the last) axis should be orthogonal to the previous one.



source

This idea could be used in a variety of ways. For example, it might happen, that projection of complex data on the principal plane (only 2 components) bring you enough intuition for clustering. The picture below plots projection of the labeled dataset onto the first to principal components (PCs), we can clearly see, that only two vectors (these PCs) would be enogh to differ Finnish people from Italian in particular dataset (celiac disease (Dubois et al. 2010))



source

Problem

The first component should be defined in order to maximize variance. Suppose, we've already normalized the data, i.e. $\sum_i a_i = 0$, then sample variance will become the sum of all squared projections of data points to our vector $\mathbf{w}_{(1)}$, which implies the following optimization problem:

$$\mathbf{w}_{(1)} = \underset{\|\mathbf{w}\|=1}{\arg\max} \left\{ \sum_i \left(\mathbf{a}_{(i)}^\top \cdot \mathbf{w} \right)^2 \right\}$$

or

$$\mathbf{w}_{(1)} = \underset{\|\mathbf{w}\|=1}{\arg\max} \{ \|\mathbf{A}\mathbf{w}\|^2 \} = \underset{\|\mathbf{w}\|=1}{\arg\max} \left\{ \mathbf{w}^\top \mathbf{A}^\top \mathbf{A}\mathbf{w} \right\}$$

since we are looking for the unit vector, we can reformulate the problem:

$$\mathbf{w}_{(1)} = \arg\max \left\{ \frac{\mathbf{w}^\top \mathbf{A}^\top \mathbf{A}\mathbf{w}}{\mathbf{w}^\top \mathbf{w}} \right\}$$

It is known, that for positive semidefinite matrix $A^\top A$ such vector is nothing else, but eigenvector of $A^\top A$, which corresponds to the largest eigenvalue. The following components will give you the same results (eigenvectors).

So, we can conclude, that the following mapping:

$$\underset{n\times k}{\Pi} = \underset{n\times d}{A} \cdot \underset{d\times k}{W}$$

describes the projection of data onto the k principal components, where W contains first (by the size of eigenvalues) k eigenvectors of $A^\top A$.

Now we'll briefly derive how SVD decomposition could lead us to the PCA.

Firstly, we write down SVD decomposition of our matrix:

$$A = U\Sigma W^\top$$

and to its transpose:

$$\begin{aligned} A^\top &= (U\Sigma W^\top)^\top \\ &= (W^\top)^\top \Sigma^\top U^\top \\ &= W\Sigma^\top U^\top \\ &= W\Sigma U^\top \end{aligned}$$

Then, consider matrix $AA^\top$:

$$\begin{aligned} A^\top A &= (W\Sigma U^\top)(U\Sigma V^\top) \\ &= W\Sigma I\Sigma W^\top \\ &= W\Sigma\Sigma W^\top \\ &= W\Sigma^2 W^\top \end{aligned}$$

Which corresponds to the eigendecomposition of matrix $A^\top A$, where W stands for the matrix of eigenvectors of $A^\top A$, while Σ^2 contains eigenvalues of $A^\top A$.

At the end:

$$\begin{aligned} \Pi &= A \cdot W = \\ &= U\Sigma W^\top W = U\Sigma \end{aligned}$$

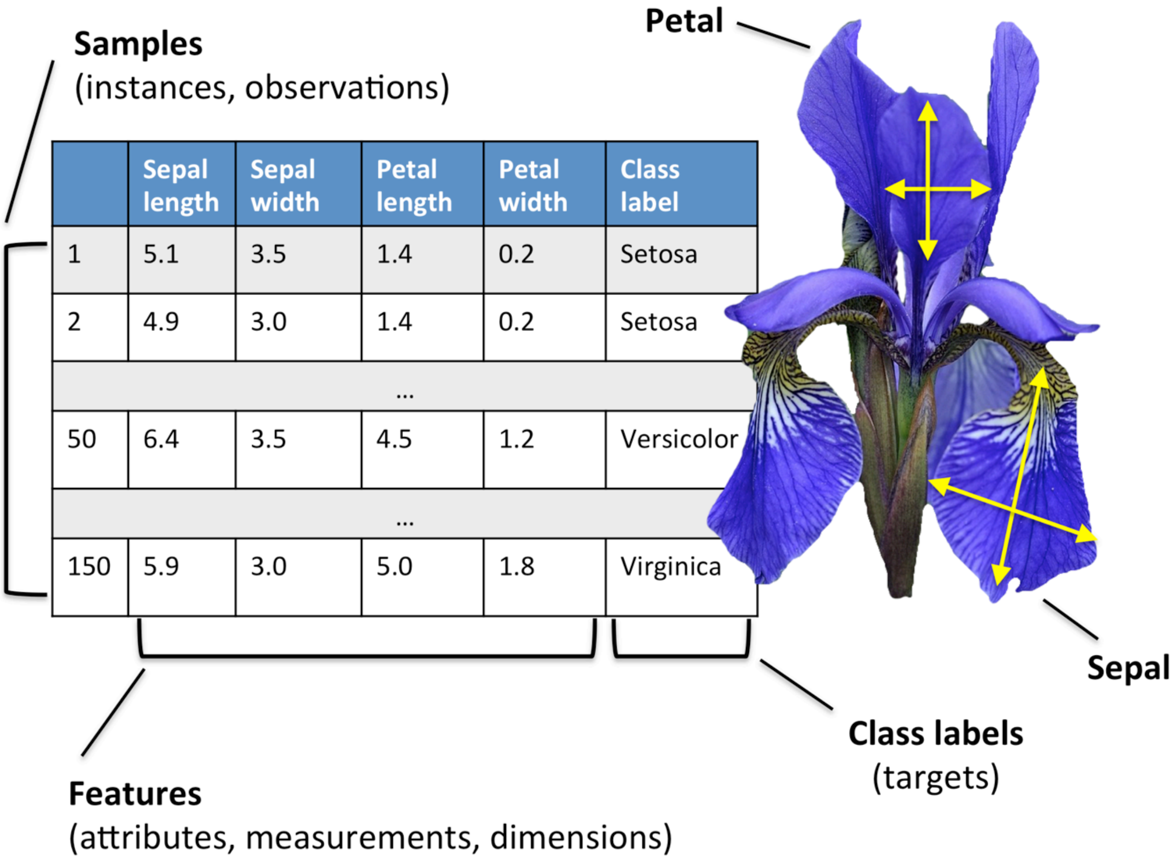
The latter formula provide us with easy way to compute PCA via SVD with any number of principal components:

$$\Pi_r = U_r \Sigma_r$$

Examples

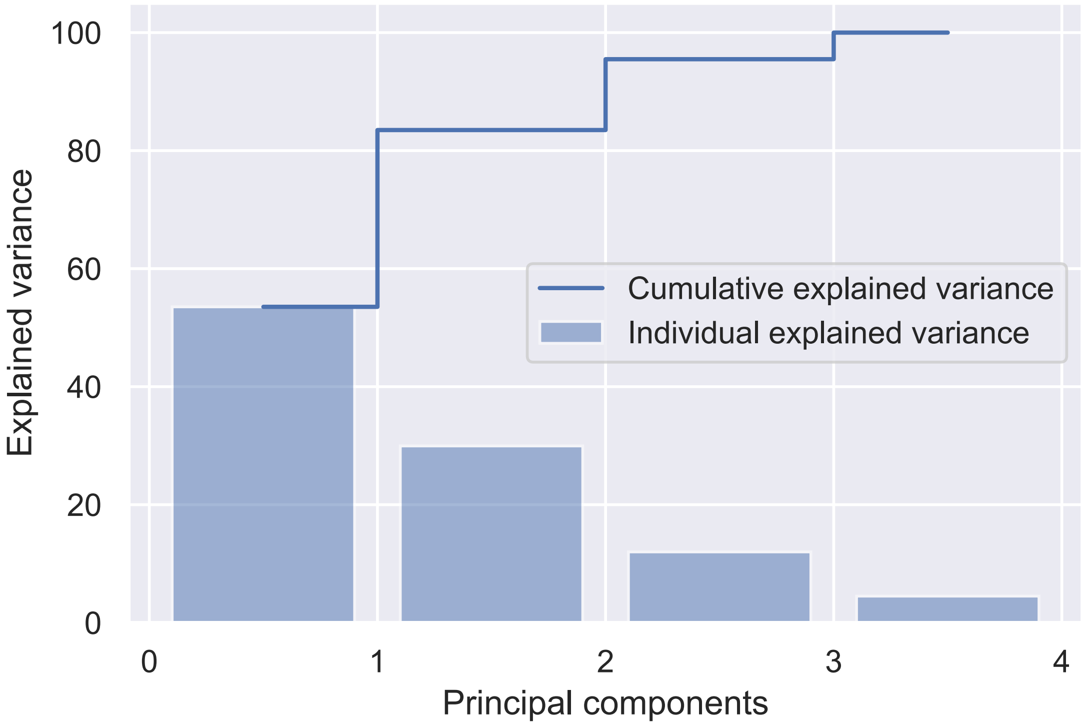
🌼 Iris dataset

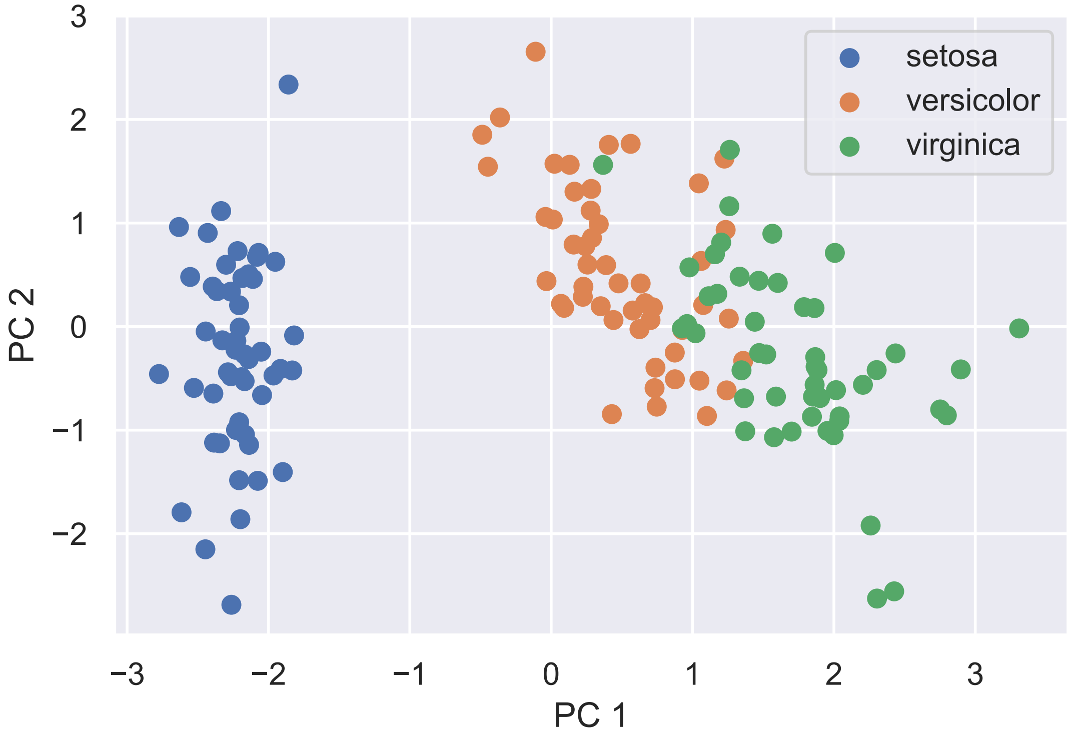
Consider the classical Iris dataset



source

We have the dataset matrix $A \in \mathbb{R}^{150 \times 4}$





Code

Open in Colab

Related materials

- Wikipedia

- Blog post
- Blog post