Principal Component Analysis

Exercises prefixed with **T** are discussed in the tutorial. The **H** prefix is for the homework questions.

Exercise T1.1: Dimensionality Reduction via PCA (tutorial)

(a) What are the characteristics of a covariance matrix?

(b) Measure the mean squared error (MSE) when reconstructing observations from a subset of the observed variables.

(c) Describe how PCA rotates the observed variables to minimize the MSE.

(d) What is a scree plot?

(e) Measure MSE when reconstructing observations after they have been projected onto the space spanned by a subset of PCs.

Exercise H1.1: PCA: 2-dimensional Toy Data (homework, 2 points)

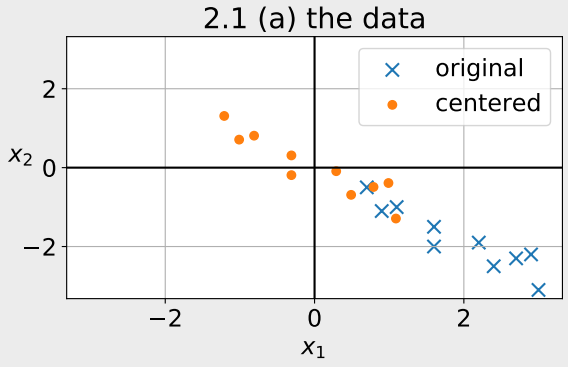
(a) Load the dataset `pca-data-2d.dat` and create a scatter plot of the *centered* data.

(b) Determine the Principal Components (PCs) and create another scatter plot of the same data points in the coordinate system spanned by the 2 PCs.

(c) PCA can be used to compress data e.g. using only information contained in the first M out of N PCs. Plot the reconstruction of the data in the original coordinate system when using (i) only the first or (ii) only the second PC for reconstruction.

Solution

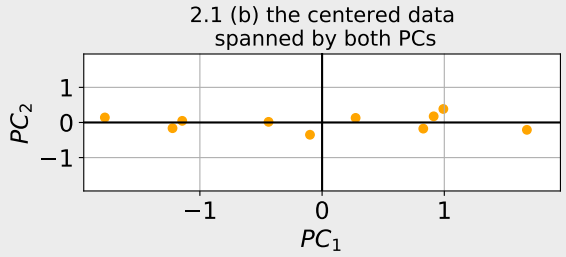
(a)



(b)
Let $\underline{\mathbf{X}} \in \mathbb{R}^{N \times p}$ be the centered data with N dimensions and p points.
$\underline{\mathbf{M}} := \left(\underline{\mathbf{e}}_1, \underline{\mathbf{e}}_2\right)^\top$ represents the Eigenbasis, where each column contains $\underline{\mathbf{e}}_a$ represents the normalized eigenvector of the a^{th} PC.
The data is projected onto the space spanned by the PCs using

$$\underline{\mathbf{u}}^{(\alpha)} = \underline{\mathbf{M}}^\top \underline{\mathbf{x}}^{(\alpha)}, \alpha = 1, \ldots, p \quad \Longrightarrow \quad \underline{\mathbf{U}} = \underline{\mathbf{M}}^\top \underline{\mathbf{X}}.$$

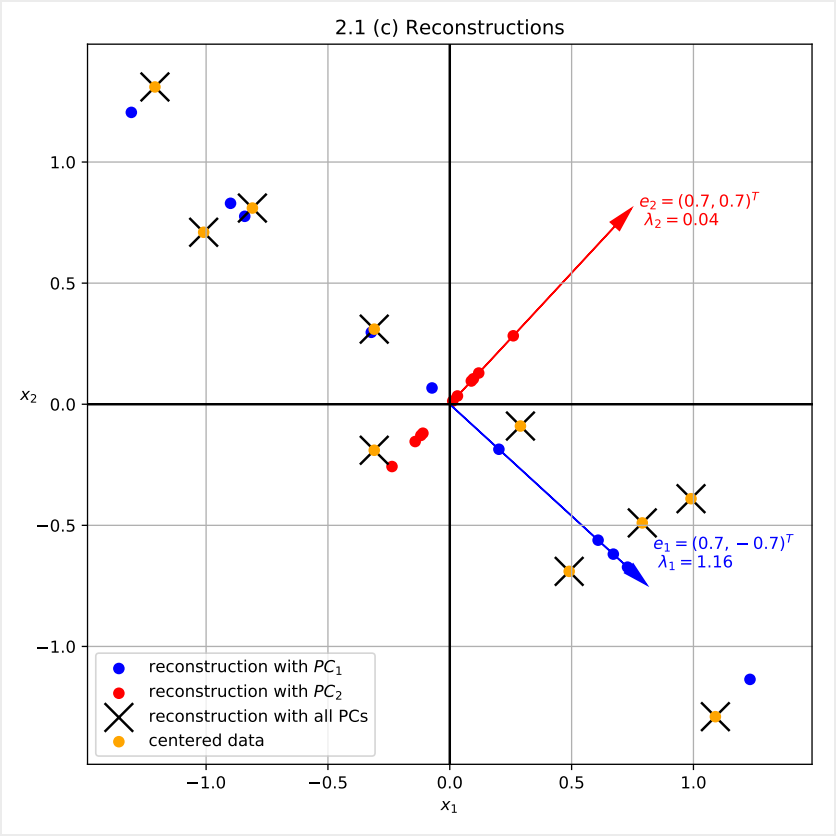
Note:
In Python 3 you can multiply matrices A and B (2-D numpy arrays) using A @ B.



(c)
We can undo the projection (reconstruct the observations) using

$$\widetilde{\underline{\mathbf{X}}} = \left(\underline{\mathbf{M}}^\top\right)^{-1} \underline{\mathbf{U}} \overset{\underline{\mathbf{M}} \text{ is orth.}}{=} \underline{\mathbf{M}}' \, \underline{\mathbf{U}}',$$

where $\underline{\mathbf{M}}', \underline{\mathbf{U}}'$ can contain all or a subset of the eigenvectors and projected components, respectively.
In the case of $\underline{\mathbf{U}}' = \underline{\mathbf{U}}$, we have a perfect reconstruction (i.e. $\widetilde{\underline{\mathbf{X}}} = \underline{\mathbf{X}}$ and MSE = 0).


2.1 (c) Reconstructions
$e_2 = (0.7, 0.7)^T$
$\lambda_2 = 0.04$
$e_1 = (0.7, -0.7)^T$
$\lambda_1 = 1.16$
reconstruction with PC_1
reconstruction with PC_2
reconstruction with all PCs
centered data
x_1
x_2
1.0
0.5
0.0
−0.5
−1.0

Exercise H1.2: PCA: 3-dimensional Toy Data (homework, 2 points)

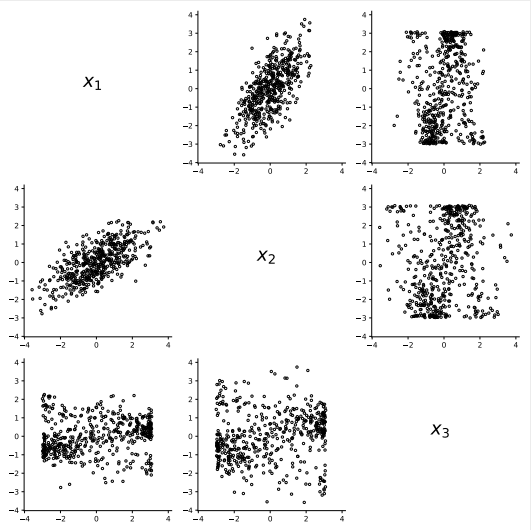
(a) Load the dataset `pca-data-3d.txt`, center it, and show the scatter plot matrix.

(b) Determine the PCs and make the analogous scatter plot matrix for the 2d-coordinate systems spanned by the different pairs of PCs.

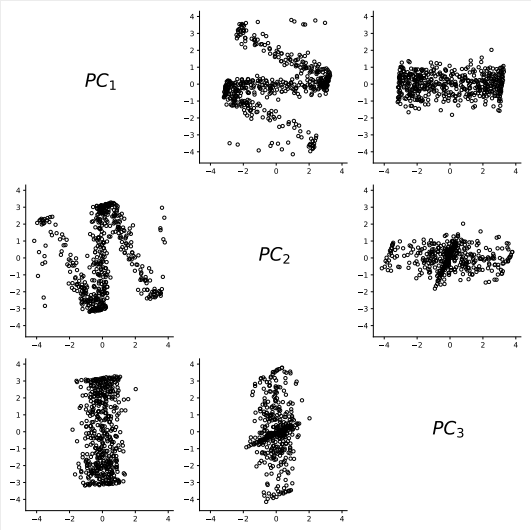
(c) Examine the 3d-reconstruction of the data in the original coordinate systems when using only (i) the first, (ii) the first two or (iii) all three PCs for reconstruction. Discuss how useful these principal directions are.

Solution

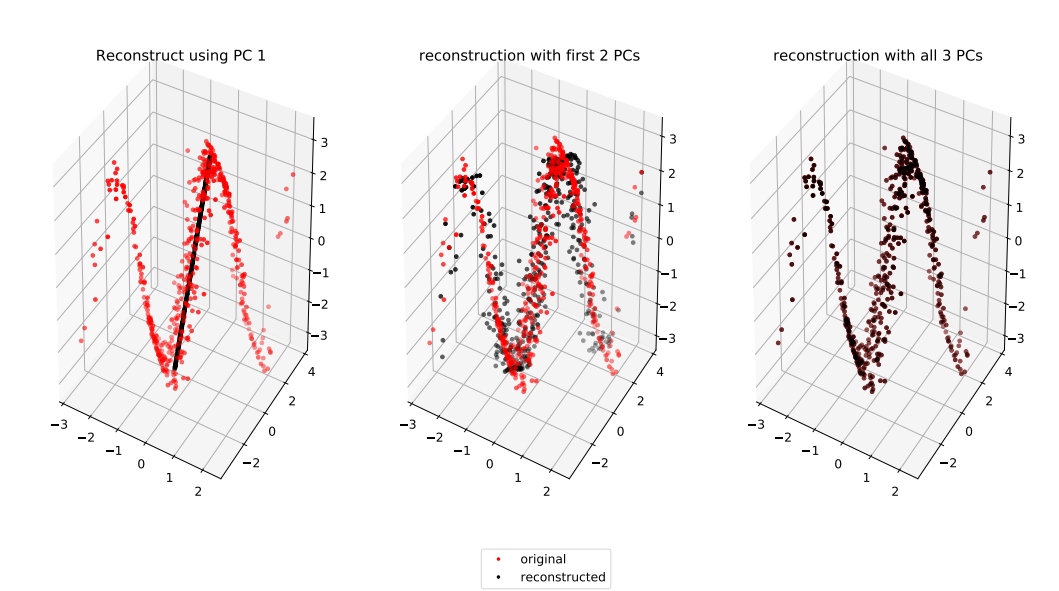
(a)



(b)



(c) Reconstructions:



Exercise H1.3: Projections of a dynamical system (homework, 3 points)

Using the data from the file `expDat.txt`, we can interpret the data as describing the process of a "large" system with $N = 20$ dimensions at each timepoint (cf. Exercise Sheet 00b).

(a) Find the 20 Principal Components of this dataset.

(b) Plot the temporal evolution of the system projected onto the *first two PCs* by making:

(i) a *scatter plot* of the 100 datapoints in the 2d-coordinate system spanned by the first two PCs. Color each point by its time index.

(ii) a *line/scatter plot* of the 100 data point projections onto the first PC and onto the second PC (i.e., two lines in one plot, one line for the projections along each PC and where the x-axis shows the time index). Use the same color scale as before to indicate the time index in order to highlight the (temporal) relationship in both plots.

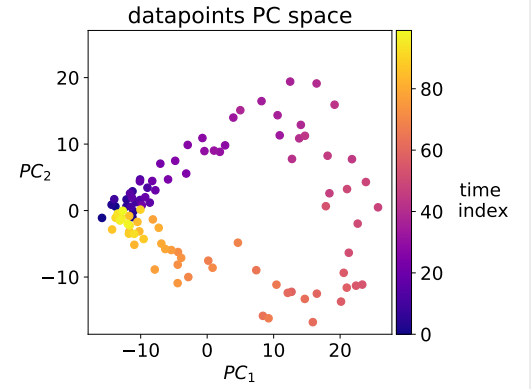
(c) Create a new dataset by shuffling the data (i.e. reorder *each* of the 20 columns the 100 data point components in a different random sequence).

(d) Plot the covariance matrices and scree plots for both the original and the scrambled data and interpret your results.

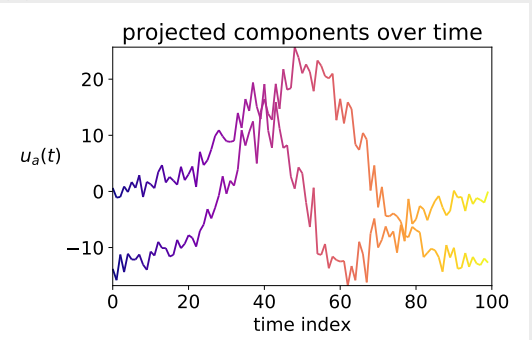
(e) What would be the result if we use them same shuffling for all columns, i.e. randomizing only the row order? You don't need to program anything to answer this question.

Solution

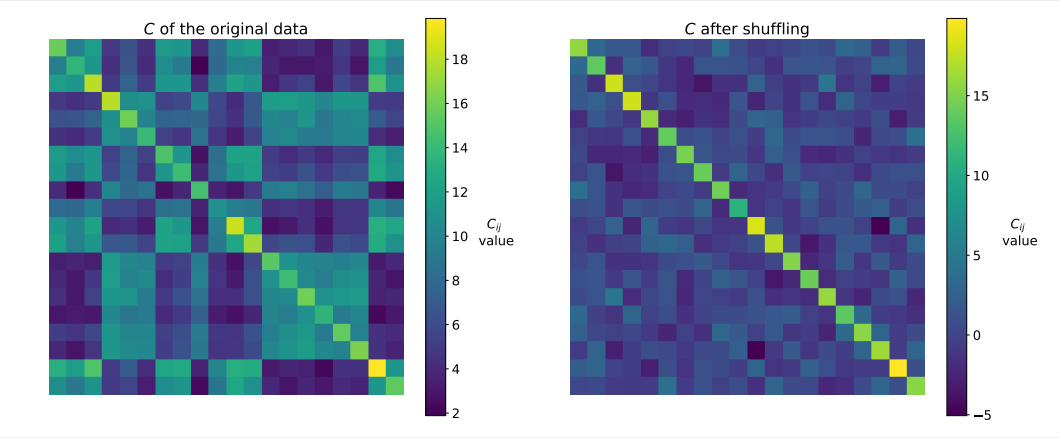
(b)
(i)

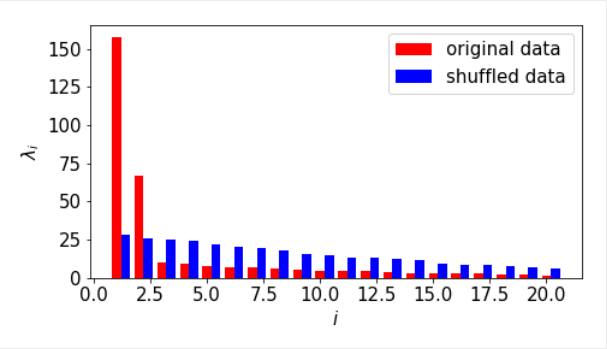


(ii)



(d)





Scree plots show how much variance is carried by which PC. We can see that, in the original data, the first 2 PCs carry a high percentage of the total variance, whereas in the shuffled data the variance is more evenly distributed over the PCs.

(e) If the data points were shuffled in the same sequence for all columns, this would just change the sequence of the data points (in the rows). Therefore, the resulting principle components would be identical, but the temporal sequence of the data points would be different.

Exercise H1.4: Image data compression and reconstruction(homework, 3 points)

The file `imgpca.zip` contains training images from two different categories, namely nature and buildings. The prefix `n` is used for the nature images, while the prefix `b` is used for the images of buildings.

(a) For each category separately: Randomly sample at least $p = 5000$ patches (e.g. 500 per image) of $16 \times 16 = 256$ pixels from this set of images and assemble them in a $256 \times p$ matrix.

(b) Calculate the PCs of these image patches and visualize the direction of the first 24 PCs as 16×16 images. Are there differences between the PC "images" of buildings vs. nature?

(c) Answer using a scree plot: How many PCs should you keep for each of the two categories? What are the resulting respective compression ratios (qualitatively)?

(d) Comparing image statistics: How similar are the statistics of nature vs building images? One way to go about this, is to reconstruct the projected image patches back into image space while alternating which basis we use for reconstruction. Whether we

A. use the basis with which the patches were originally projected (i.e. using the PCs that match the category of the image)
OR

B. reconstruct the patches using the "wrong" basis (i.e. the PCs of the other image category).

Procedure: Pick any 3 images from each category. For each image:

(i) Project all non-overlapping[1] 16×16-patches onto the first M PCs of that image's category (e.g. if it's a building image, project it using the first M building PCs)
for $M \in \{1, 2, 4, 8, 16, 100\}$.

(ii) Reconstruct the projected image patches by using:

"matching" PCs (i.e. building PCs for reconstructing building image patches and nature PCs for nature image reconstruction)
as well as

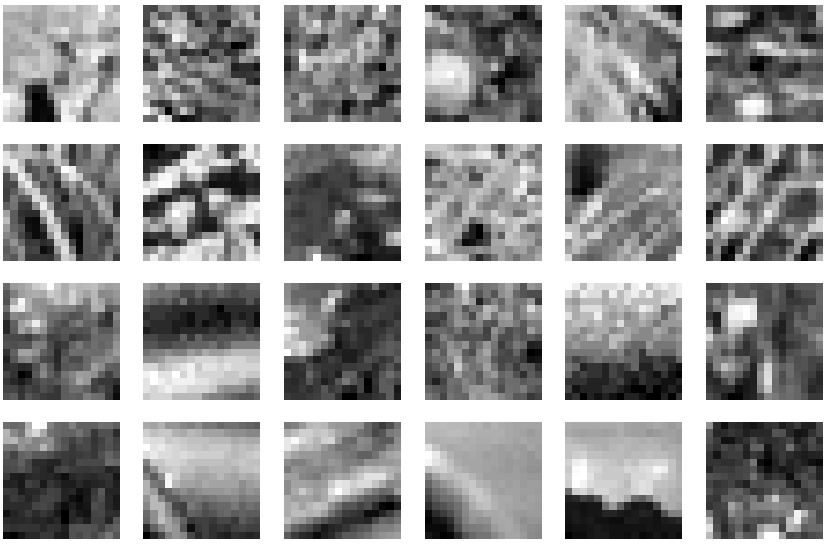
the "wrong" PCs (i.e. building PCs for nature image reconstruction, nature PCs for building image reconstruction).

Deliverables: Choose a compact way to visualize this comparison,
e.g. show the reconstructed patches reassembled into an image within a subplot. The subplot next to it shows the reassembled reconstructions using a larger number of M PCs. The last subplot in this row of subplots shows the original image. Use one row for reconstructions using "matching" PCs and another row for "wrong" PCs.
$\rightarrow$ Interpret the results.

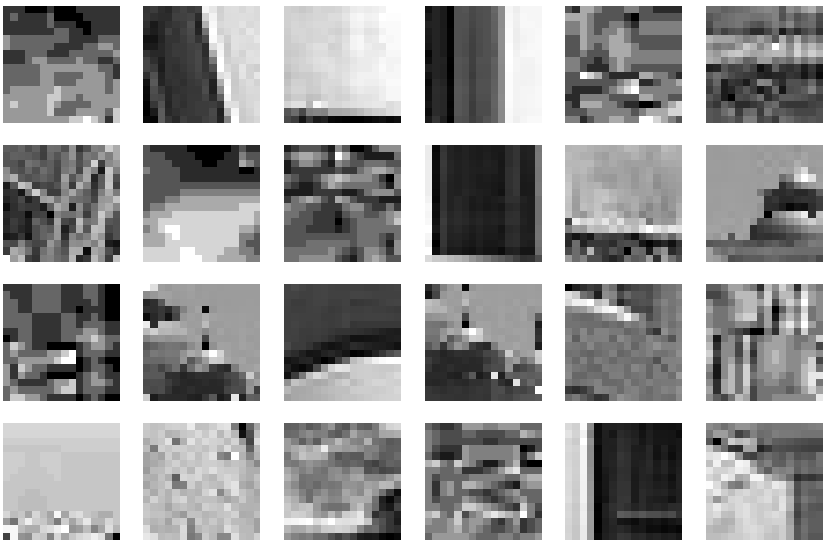
Solution

(a)
Random patches of natural images

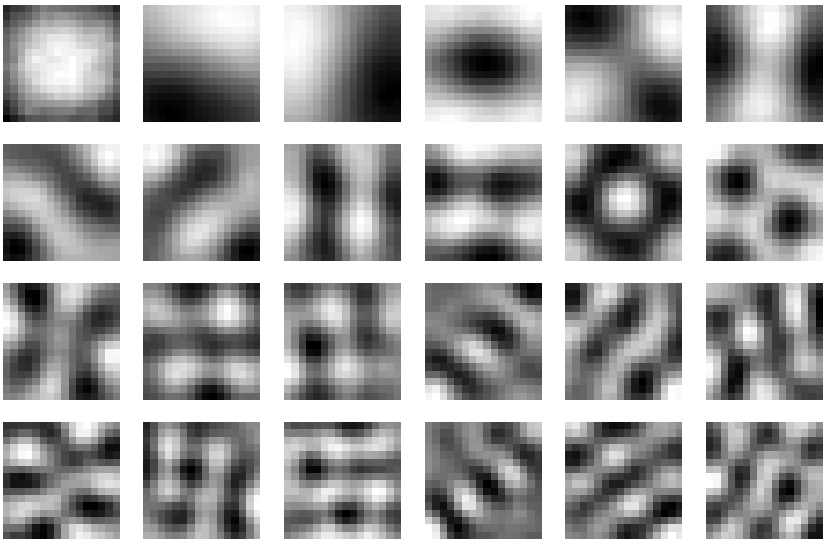
[1] For boundary regions it might be required to overlap the patches such that no pixels of the image are neglected.



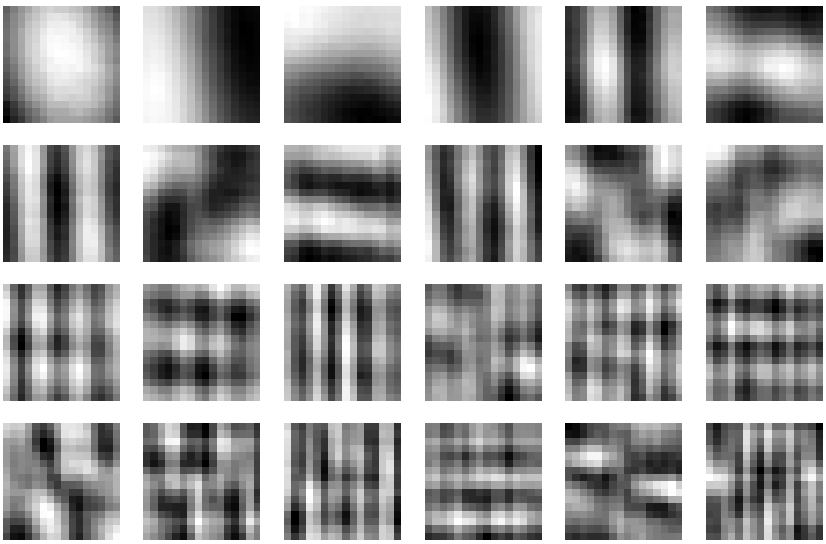
Random patches of building images



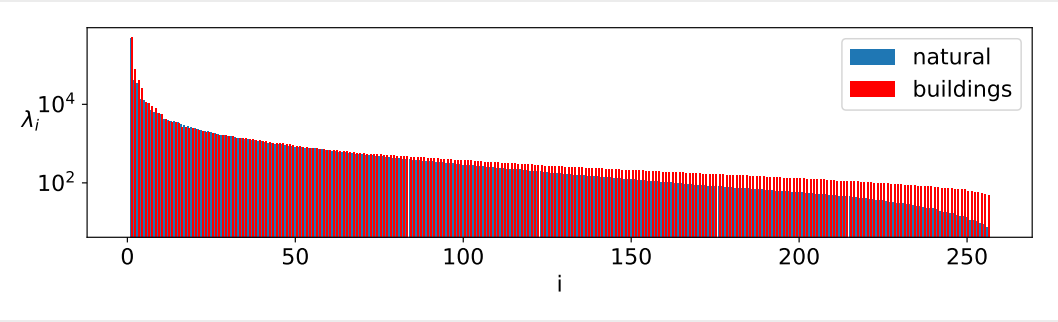
(b)
First 24 PCs of natural images



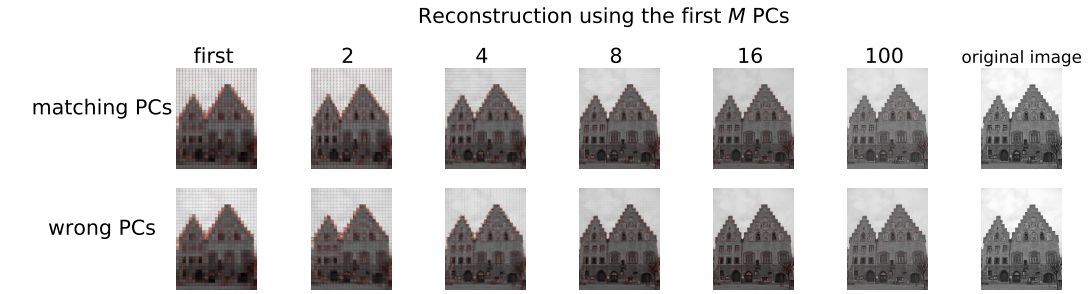
First 24 PCs of building images

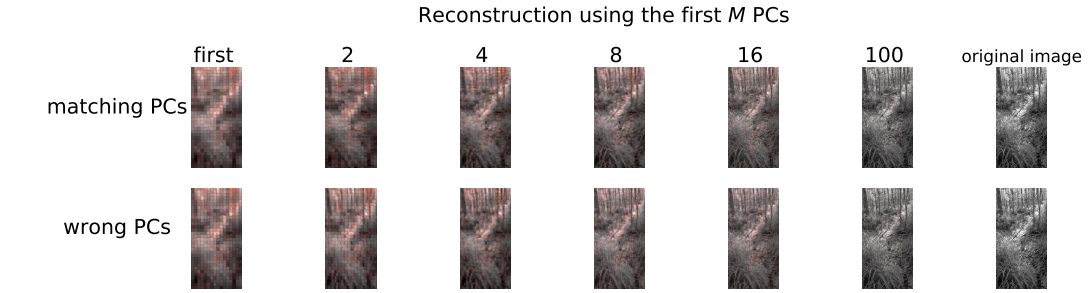


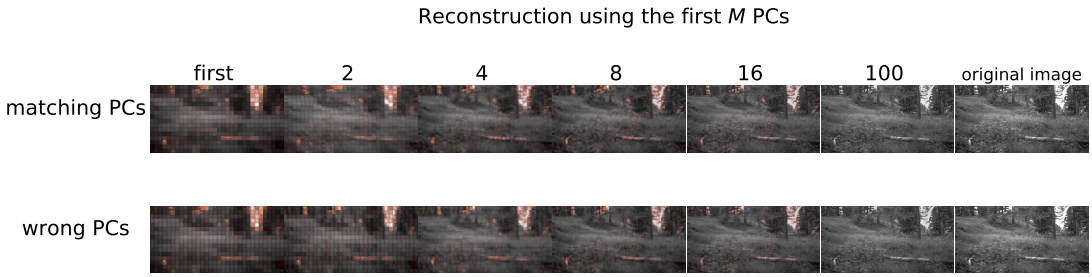
(c)
Scree plots



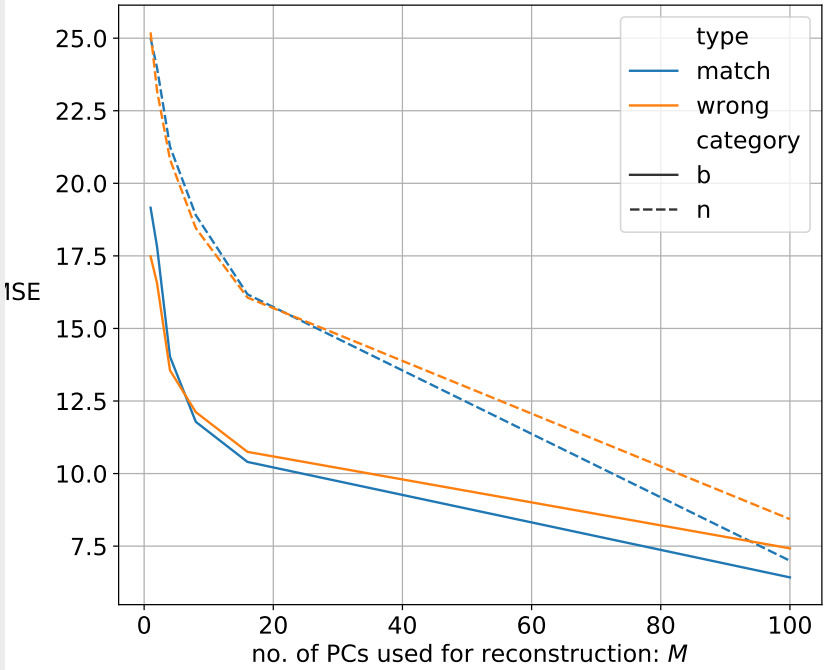
(d)Reconstruction of sample images:







Effect on Root mean squared error (RMSE):



Total 10 points.