Recap

- (Ch 10) Data in high dimensions
  - Visualizing data
  - Summarizing data

Today

- (Ch 10) Data in high dimensions
  - Dimensionality reduction
  - Principal components analysis

Covariance matrix of multidimensional data

- Given a dataset $\{\mathbf{x}\}$ of N d-dimensional vectors $\mathbf{x}_i$, the covariance matrix is a $d \times d$ matrix

$$\text{Covmat}(\{\mathbf{x}\}) = \frac{\sum_i (\mathbf{x}_i - \text{mean}(\{\mathbf{x}\}))(\mathbf{x}_i - \text{mean}(\{\mathbf{x}\}))^T}{N}$$

- Properties
  - The (j, k) entry of $\text{Covmat}(\{\mathbf{x}\})$ is $\text{cov}\left(\{\mathbf{x}^{(j)}\}, \{\mathbf{x}^{(k)}\}\right)$
  - The (j, j) entry of $\text{Covmat}(\{\mathbf{x}\})$ is $\text{var}\left(\{\mathbf{x}^{(j)}\}\right)$
  - $\text{Covmat}(\{\mathbf{x}\})$ is symmetric

Translation properties

- Translating the data translates the mean

$$\text{mean}(\{\mathbf{x} + \mathbf{c}\}) = \text{mean}(\{\mathbf{x}\}) + \mathbf{c}$$

- Translating the data leaves the covariance matrix unchanged

$$\text{Covmat}(\{\mathbf{x} + \mathbf{c}\}) = \text{Covmat}(\{\mathbf{x}\})$$

Linear transformation properties

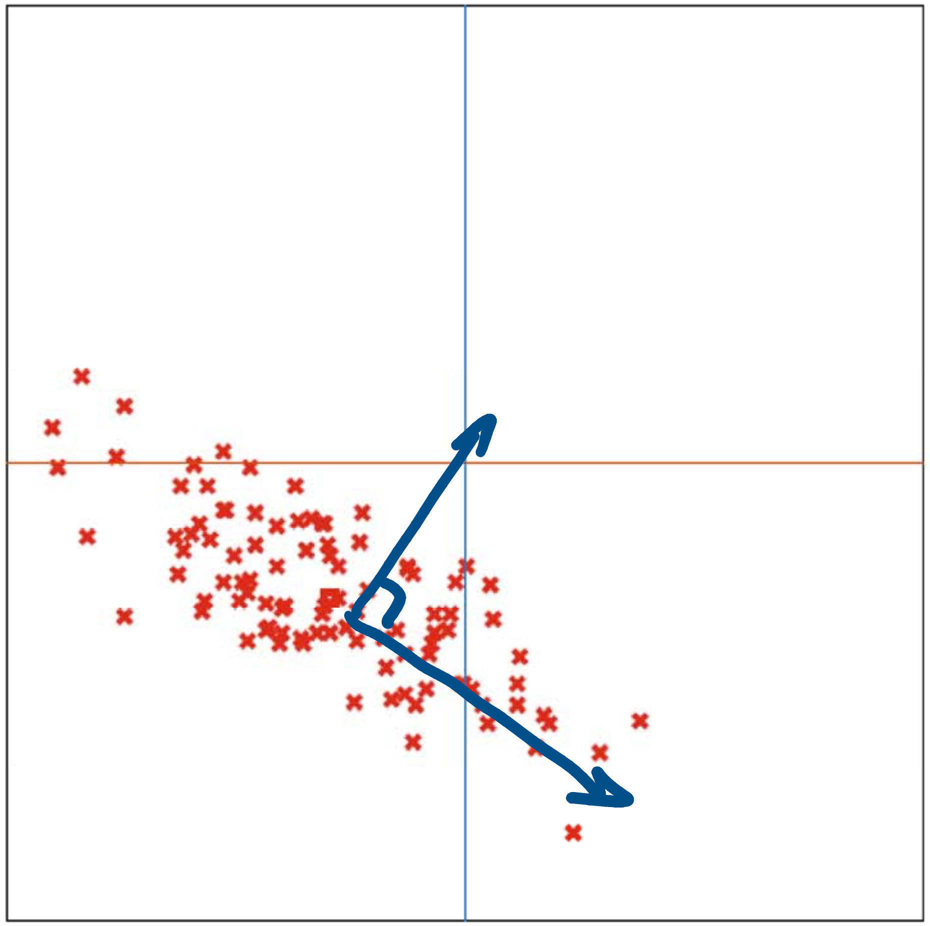
- Linearly transforming the data linearly transforms the mean

$$\text{mean}(\{A\mathbf{x}\}) = A\ \text{mean}(\{\mathbf{x}\})$$

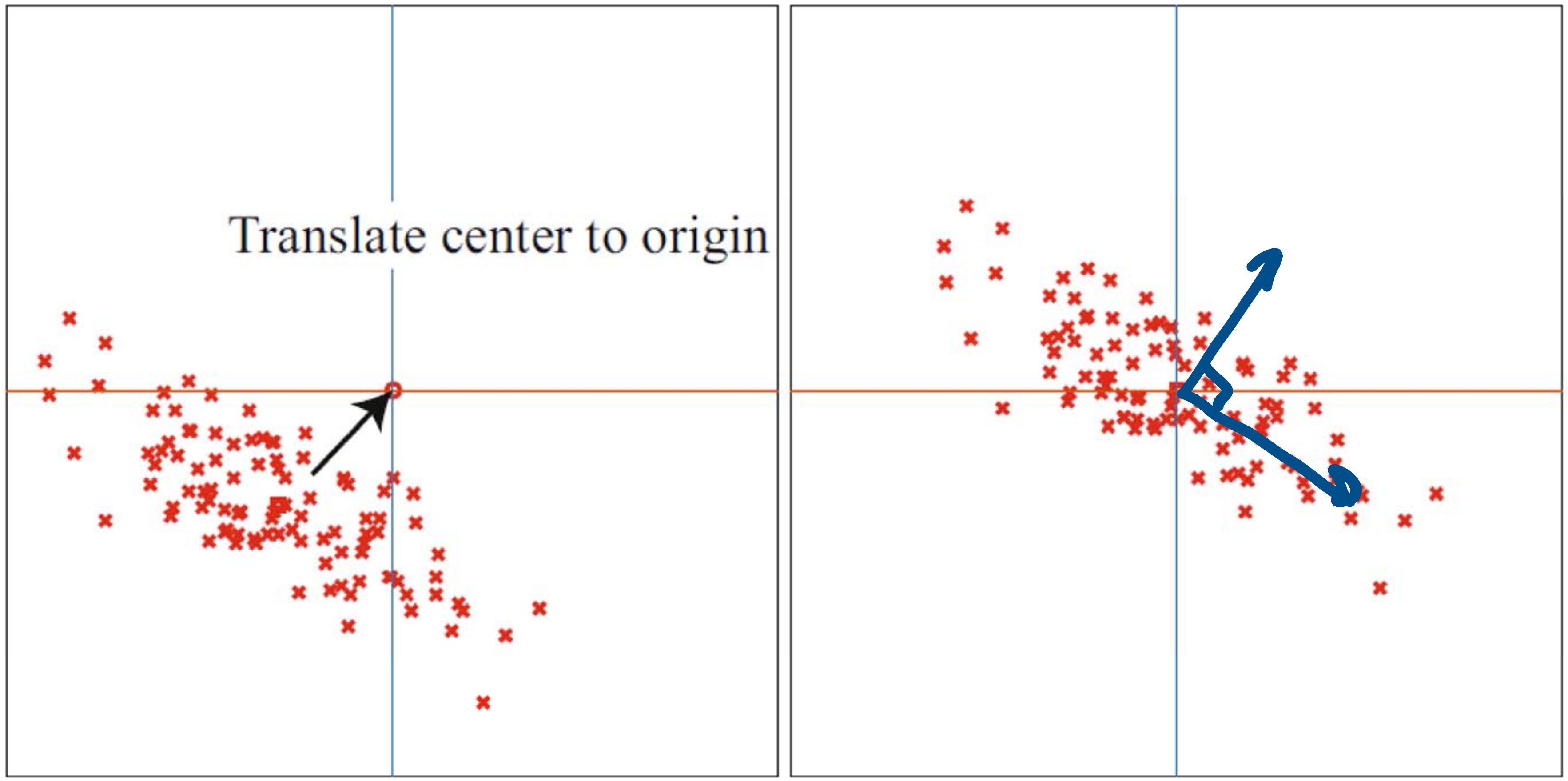
- Linearly transforming the data changes the covariance matrix

$$\text{Covmat}(\{A\mathbf{x}\}) = A\ \text{Covmat}(\{\mathbf{x}\})\ A^{\boldsymbol{T}}$$

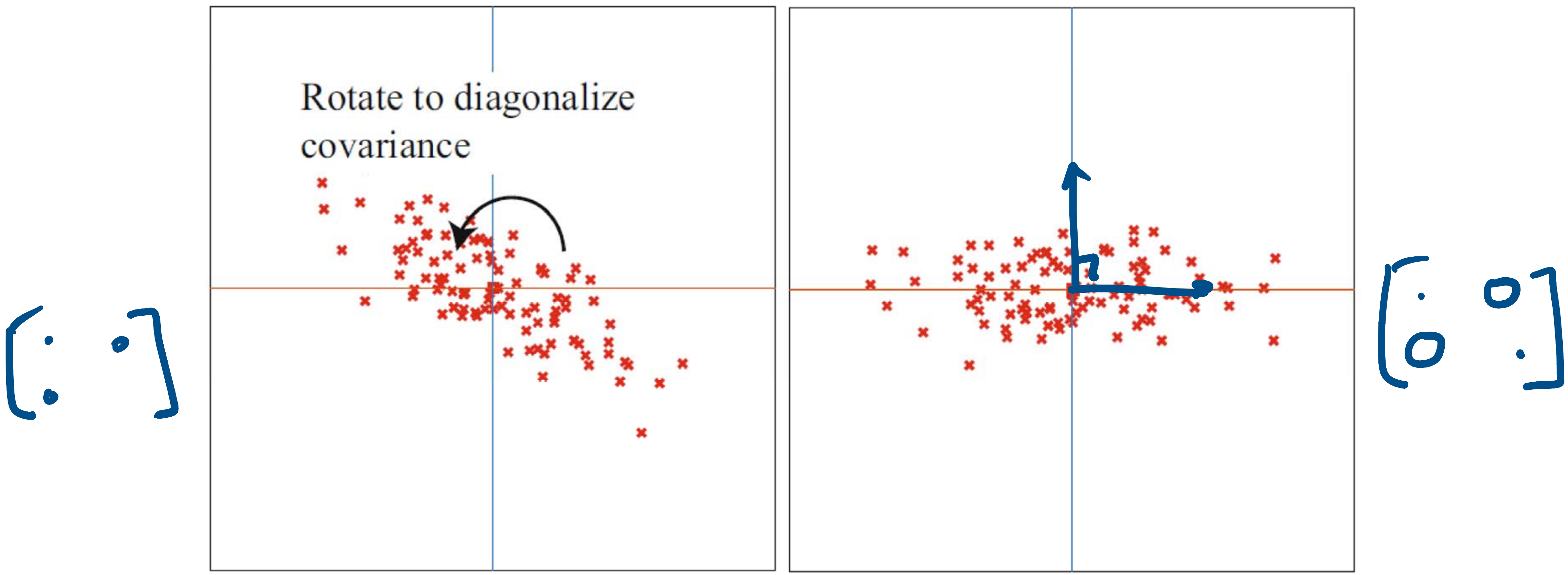
Dimensionality reduction: 2D to 1D example



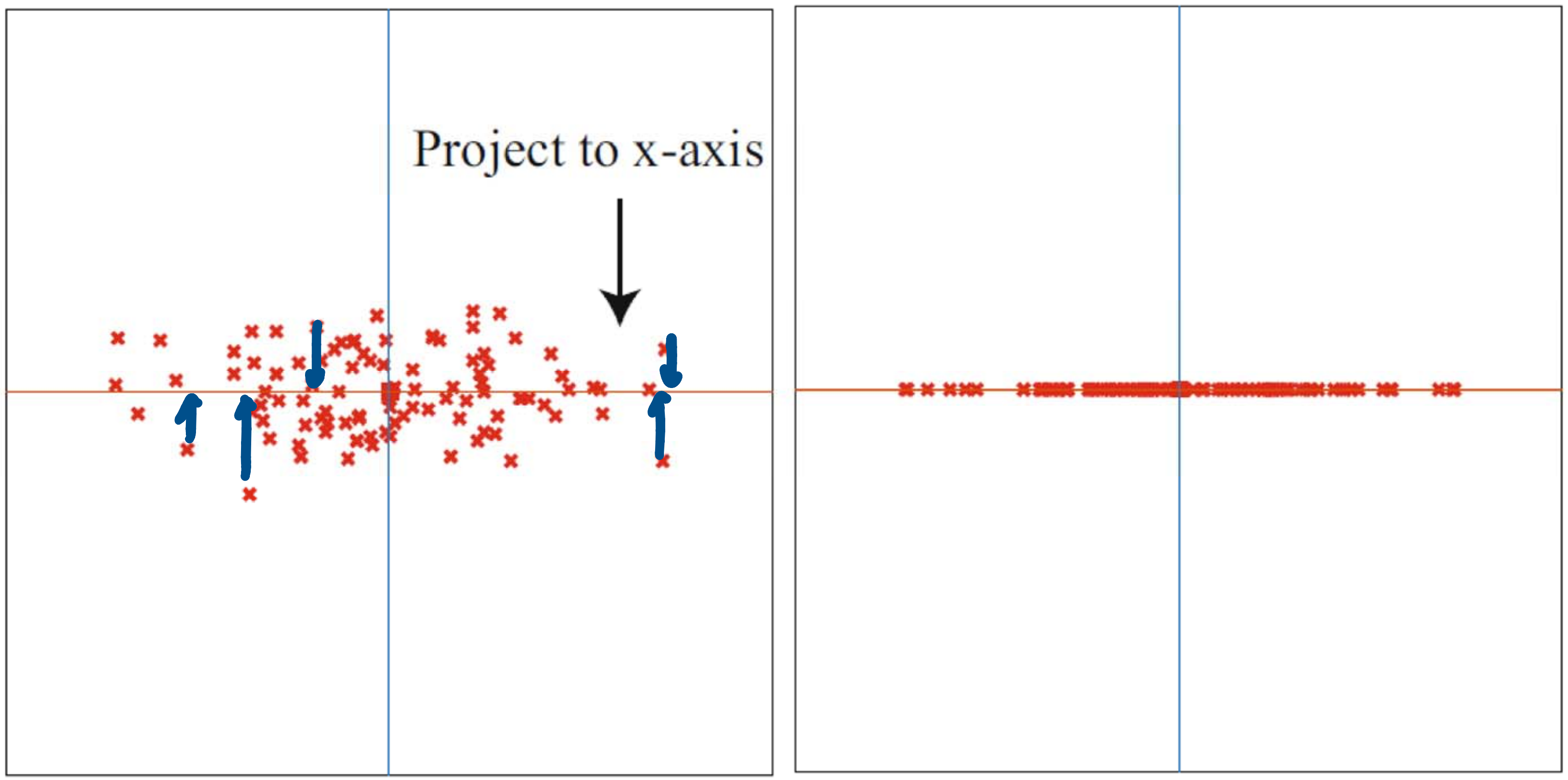
Step 1: subtract mean



Step 2: apply linear transformation



Step 3: drop component(s)



Principal components analysis (PCA)

We will reduce the dimensionality of dataset $\{\mathbf{x}\}$ from d to s

- Step 1: define $\{\mathbf{m}\}$ such that $\mathbf{m}_i = \mathbf{x}_i - \mathrm{mean}(\{\mathbf{x}\})$

- Step 2: define $\{\mathbf{r}\}$ such that $\mathbf{r}_i = U^T \mathbf{m}_i$
  where $\Lambda = U^T \mathrm{Covmat}(\{\mathbf{x}\}) U$ is the diagonalization of $\mathrm{Covmat}(\{\mathbf{x}\})$ with the eigenvalues sorted in decreasing order

- Step 3: define $\{\mathbf{p}\}$ such that each $\mathbf{p}_i$ is $\mathbf{r}_i$ with the last $d - s$ components zeroed out (or discarded)

What happens to the mean?

- Step 1:

$$\mathrm{mean}(\{\mathbf{m}\}) = \mathrm{mean}(\{\mathbf{x}\}) - \mathrm{mean}(\{\mathbf{x}\}) = \mathbf{0}$$

- Step 2:

$$\mathrm{mean}(\{\mathbf{r}\}) = U^T \mathrm{mean}(\{\mathbf{m}\}) = U^T \mathbf{0} = \mathbf{0}$$

- Step 3:

$$\mathrm{mean}(\{\mathbf{p}\}) = \mathrm{mean}(\{\mathbf{r}\}) = \mathbf{0}$$

What happens to the covariance matrix?

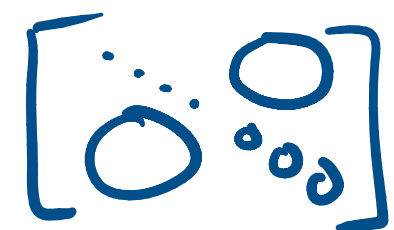
- Step 1:

$$\text{Covmat}(\{\mathbf{m}\}) = \text{Covmat}(\{\mathbf{x}\})$$

- Step 2:

$$\text{Covmat}(\{\mathbf{r}\}) = U^T \text{Covmat}(\{\mathbf{m}\}) U = \Lambda$$

- Step 3:



eigenvalues

Covmat($\{\mathbf{p}\}$) is Λ with the last $d - s$ diagonal terms zeroed out

Mean square error of the projection (step 3)

$$\frac{1}{N}\sum_{i}\|\mathbf{r}_i - \mathbf{p}_i\|^2 = \frac{1}{N}\sum_{i}\sum_{j=s+1}^{d}\left(r_i^{(j)}\right)^2 = \sum_{j=s+1}^{d}\underbrace{\frac{1}{N}\sum_{i}\left(r_i^{(j)}\right)^2}$$

$$= \sum_{j=s+1}^{d} \mathrm{var}\left(\left\{r^{(j)}\right\}\right)$$ since mean$(\{r\}) = 0$

$$= \sum_{j=s+1}^{d} \lambda_j$$ since the variances are diagonal entries of Covmat$(\{r\}) = \Lambda$

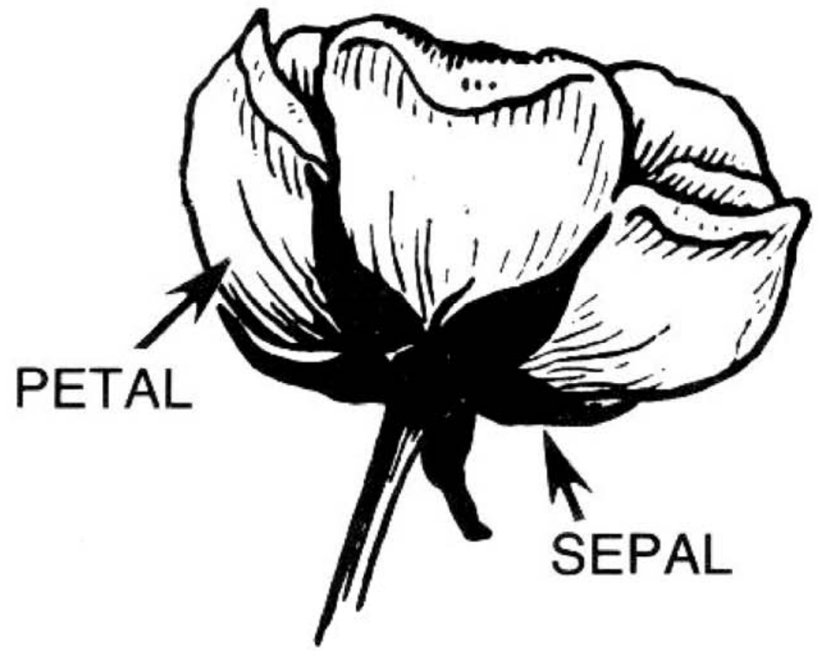
The mean square error is the sum of the smallest $d - s$ eigenvalues in Λ

PCA: iris dataset example

- The Iris dataset is a famous dataset consisting of measurements of three different varieties of iris flowers
  - Iris-setosa
  - Iris-versicolor
  - Iris-virginica

- There are 4 measurements per item
  - Sepal length (cm)
  - Sepal width (cm)
  - Petal length (cm)
  - Petal width (cm)

- See today's Jupyter notebook



Reconstructing the data

- Given the projected dataset $\{\mathbf{p}\}$ and $\mathrm{mean}(\{\mathbf{x}\})$, we can approximately reconstruct the original dataset as $\{\hat{\mathbf{x}}\}$

$$\hat{\mathbf{x}}_i = U\mathbf{p}_i + \mathrm{mean}(\{\mathbf{x}\})$$

- Notice that each reconstructed data item $\hat{\mathbf{x}}_i$ is $\mathrm{mean}(\{\mathbf{x}\})$ plus a linear combination of the columns of U weighted by the entries in $\mathbf{p}_i$

- The columns of U are the normalized eigenvectors of $\mathrm{Covmat}(\{\mathbf{x}\})$ and are called the **principal components** of the data $\{\mathbf{x}\}$

End-to-end mean square error

- Each $\mathbf{x}_i$ becomes $\mathbf{r}_i$ by translation and rotation
- Each $\mathbf{p}_i$ becomes $\hat{\mathbf{x}}_i$ by the opposite rotation and translation
- Therefore, the end-to-end mean square error is

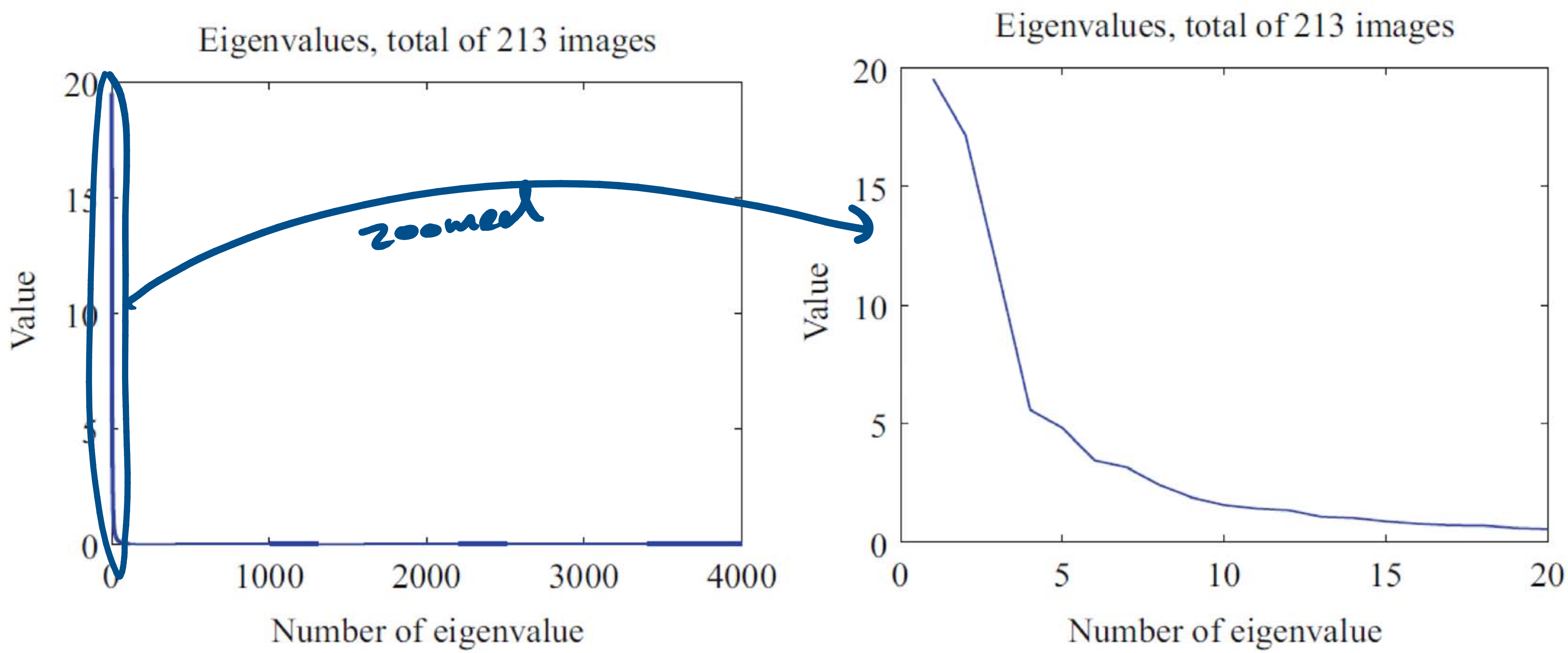
$$\frac{1}{N}\sum_i \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|^2 = \frac{1}{N}\sum_i \|\mathbf{r}_i - \mathbf{p}_i\|^2 = \sum_{j=s+1}^{d} \lambda_j$$

where $\lambda_{s+1}, \dots, \lambda_d$ are the smallest $d - s$ eigenvalues of $\mathrm{Covmat}(\{\mathbf{x}\})$

PCA: Japanese face dataset example

- The dataset consists of 213 images of Japanese women
- Each image is grayscale and has 64×64 resolution
- We can treat each image as a vector of dimension $d = 4096$

How quickly do the eigenvalues drop off?



What do the principal components look like?

Mean image from Japanese Facial Expression dataset

First sixteen principal components

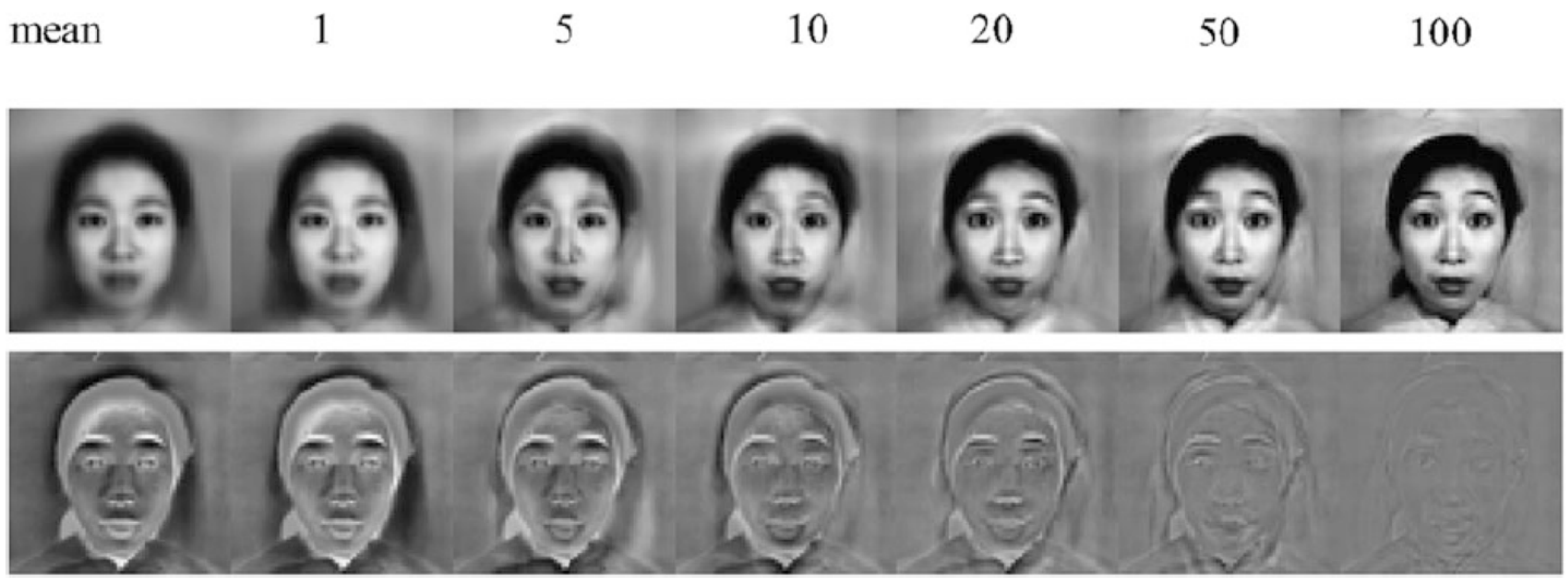
- The mean face is blurry
- The first few principal components capture
  - Shape of hair
  - Height of face
  - Height of eyebrows
  - Etc.

What do the reconstructions look like?

Original image

x_i

Number of principal components



Reconstructions

Error