CPSC 340:
Machine Learning and Data Mining

More PCA

Admin

- My office hours:
  - I've had to reschedule them from today to tomorrow 3-4pm
  - Sorry for the short notice
  - If you haven't picked up your midterm yet, I'll have them in my office hours
- Assignment 4:
  - Due Sunday.
- Assignment 5:
  - Coming soon.
- Stop/go lecture pacing tool
  - Nobody has used it for the last several lectures so I'll stop using it.
  - It sounds good in theory but maybe it's just an extra thing to think about?

The 10 Algorithms Machine Learning Engineers Need to Know

1. Decision trees
2. Naïve Bayes classification
3. Ordinary least squares regression
4. Logistic regression
5. Support vector machines
6. Ensemble methods
7. Clustering algorithms
8. Principal component analysis
9. Singular value decomposition
10. Independent component analysis

http://www.kdnuggets.com/2016/08/10-algorithms-machine-learning-engineers.html

Last time: Principal Component Analysis (PCA)

- PCA is a linear model for unsupervised learning.
- Represents features as linear combination of latent factors:

$$x_{ij} = w_j^T z_i$$

↳ column 'j'

$$x_i = W^T z_i = z_{i1} w_1 + z_{i2} w_2 + \cdots + z_{ik} w_k$$

↳ row 1, ↳ row 2, row k

the means/factors

  - But we're learning the latent factors 'W' and latent features z_i.
- Can also be viewed as an approximate matrix factorization:

$$X \approx ZW$$

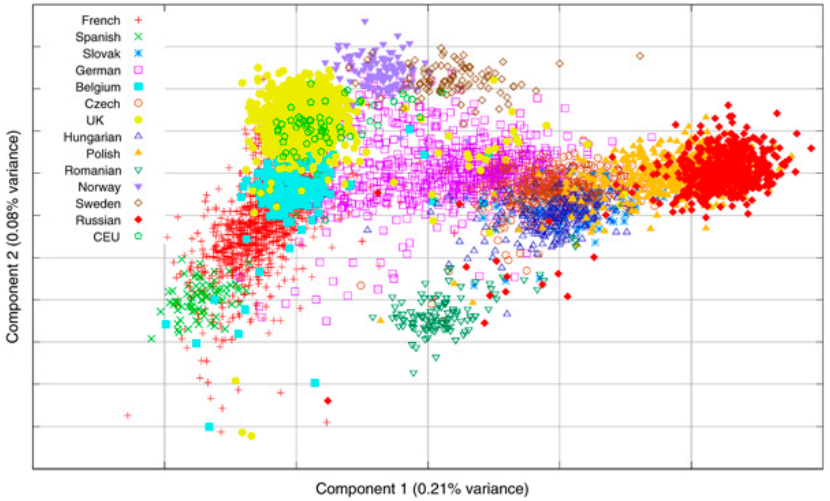
$$\underset{n \times d}{[\quad]} \approx \underset{n \times k}{[\quad]} \underset{k \times d}{[\quad]}$$

Last time: Principal Component Analysis (PCA)

- PCA is a linear model for unsupervised learning.
- Represents features as linear combination of latent factors:

$$x_{ij} = w_j^T z_i \qquad x_i = W^T z_i$$

- Uses: dimensionality reduction, visualization, factor discovery.



Trait	Description
Openness	Being curious, original, intellectual, creative, and open to new ideas.
Conscientiousness	Being organized, systematic, punctual, achievement-oriented, and dependable.
Extraversion	Being outgoing, talkative, sociable, and enjoying social situations.
Agreeableness	Being affable, tolerant, sensitive, trusting, kind, and warm.
Neuroticism	Being anxious, irritable, temperamental, and moody.

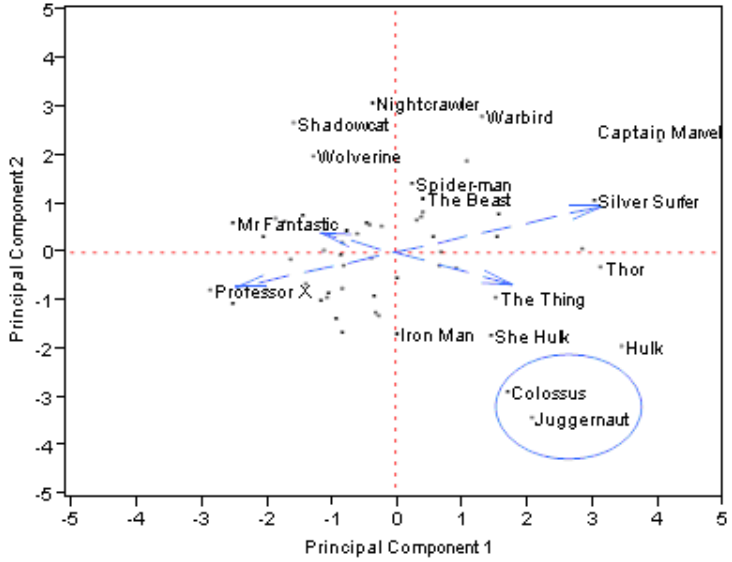
http://infoproc.blogspot.ca/2008/11/european-genetic-substructure.html
https://new.edu/resources/big-5-personality-traits

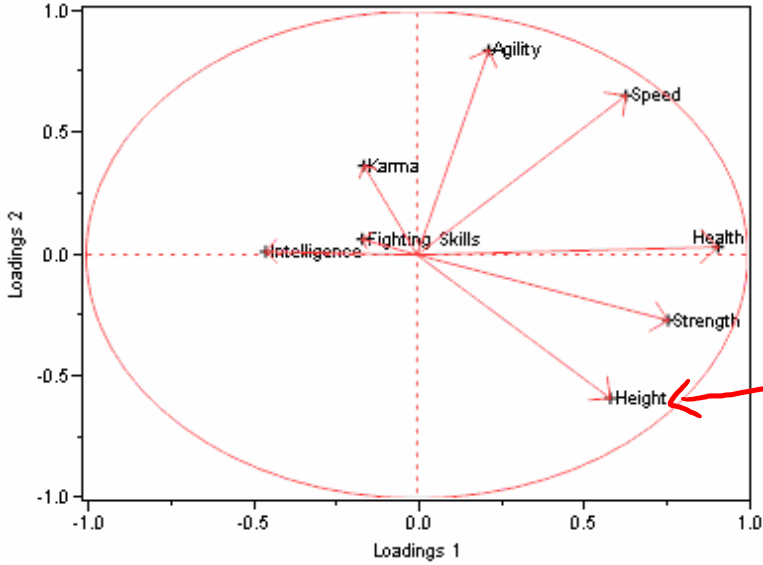
Last time: Principal Component Analysis (PCA)

- PCA is a linear model for unsupervised learning.
- Represents features as linear combination of latent factors:

$$x_{ij} = w_j^T z_i \qquad x_i = W^T z_i$$

- Uses: dimensionality reduction, visualization, factor discovery.



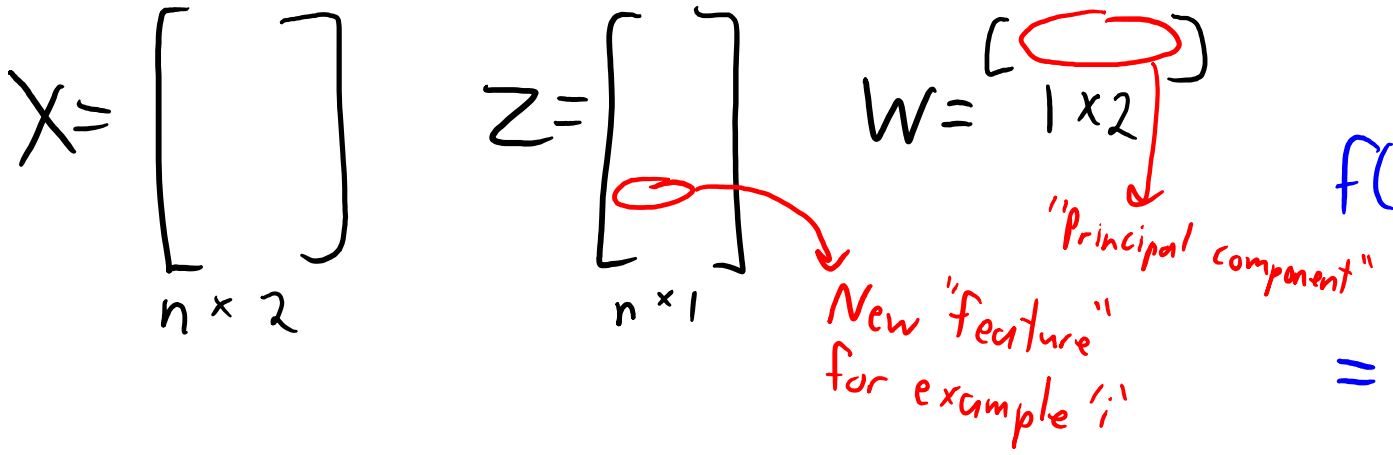


z_i value when $x_i = [1\ 0\ 0\ 0\ 0\ 0]$

http://www.prismtc.co.uk/superheroes-pca/

PCA with d=2 and k =1

- So simplest interesting case is d=2 and k=1:



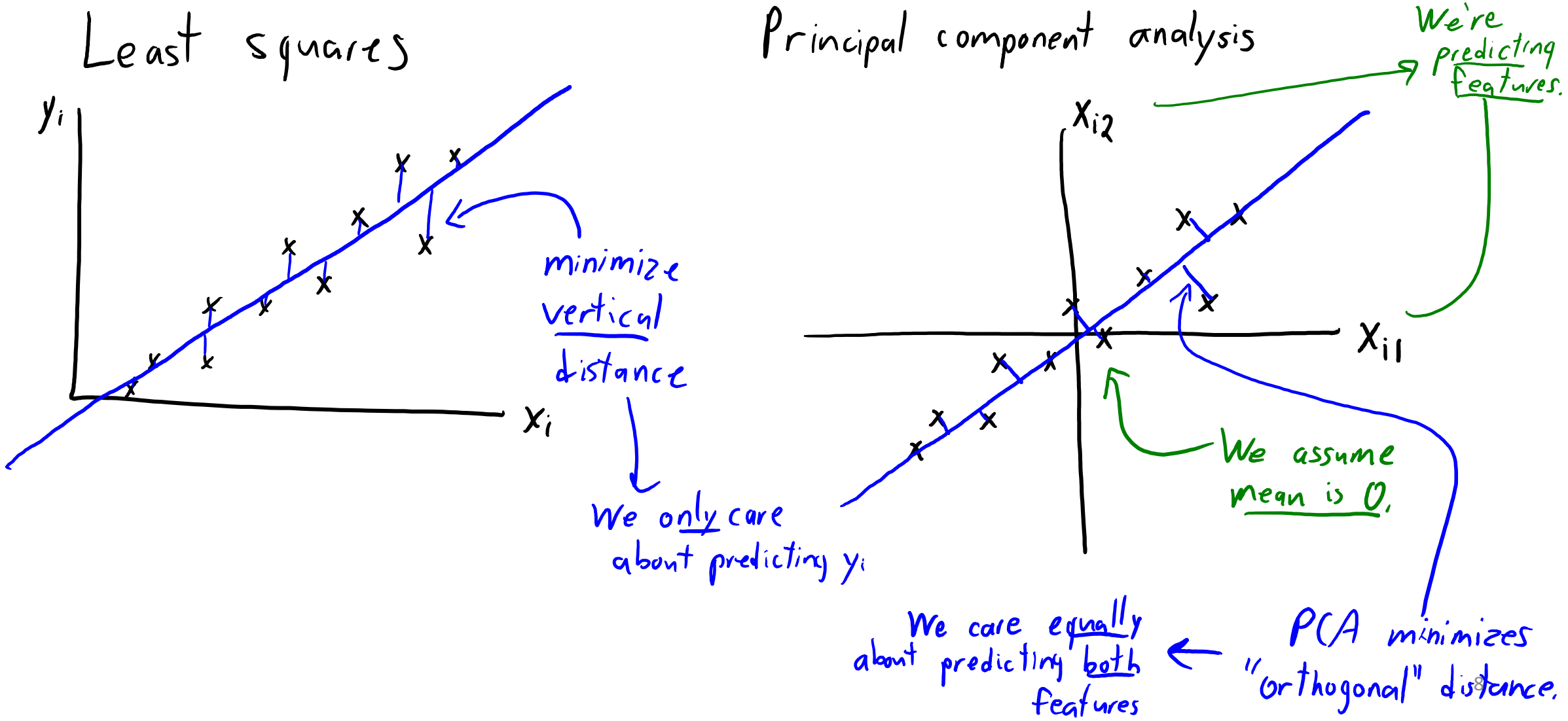
PCA objective:

$$f(Z,W) = \sum_{i=1}^{n}\sum_{j=1}^{2}(w_j z_i - x_{ij})^2$$

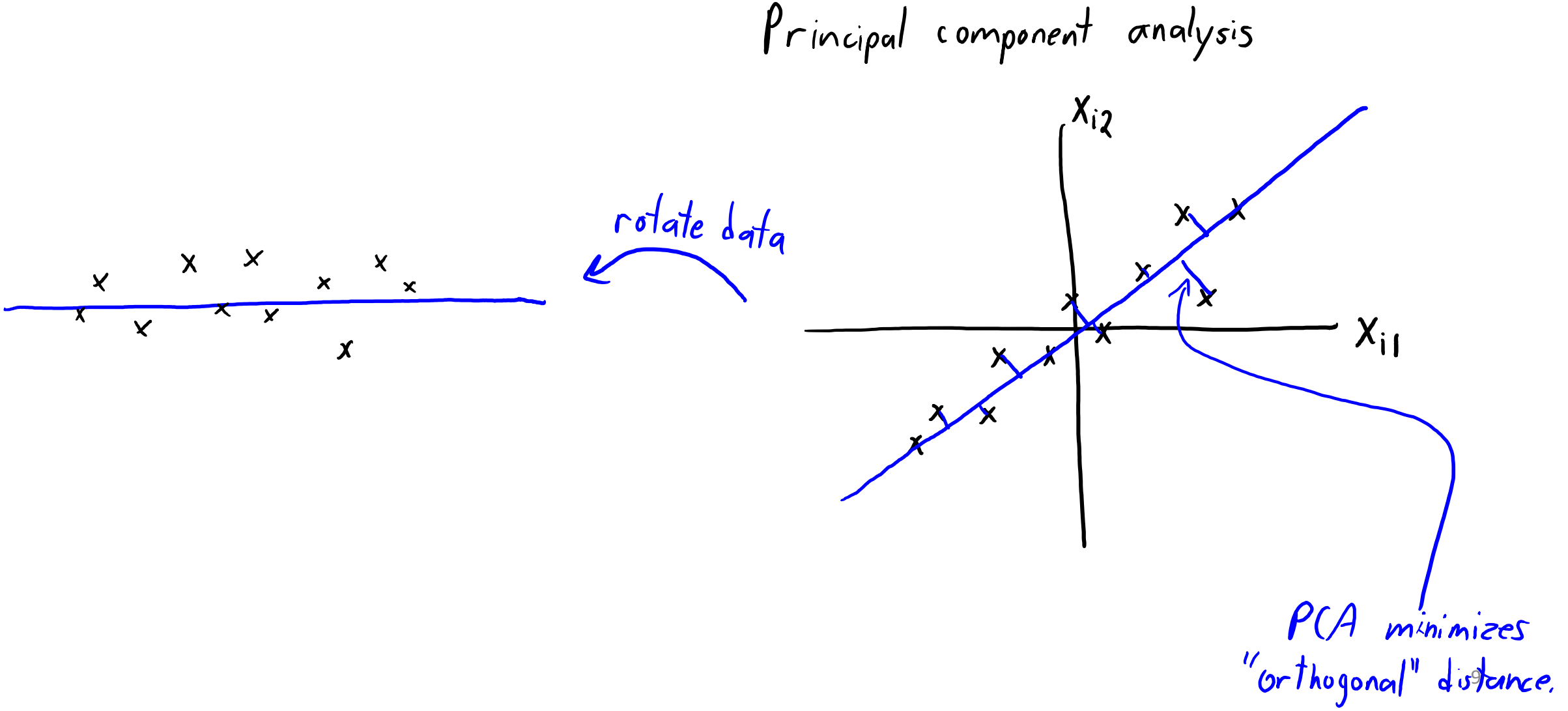
$$= \sum_{i=1}^{n}(w_1 z_i - x_{i1})^2 + \sum_{j=1}^{n}(w_2 z_i - x_{i2})^2$$

- Very similar to a least squares problem, but note that:
  - We have no 'y_i', we are trying to predict each feature x_{ij} from the single z_i.
  - But features 'z_i' are also variables, we are learning the features z_i too.
- Side note: in PCA we assume features have a mean of 0.
  - You can subtract mean or add bias variable if this is not true.

PCA with d=2 and k =1

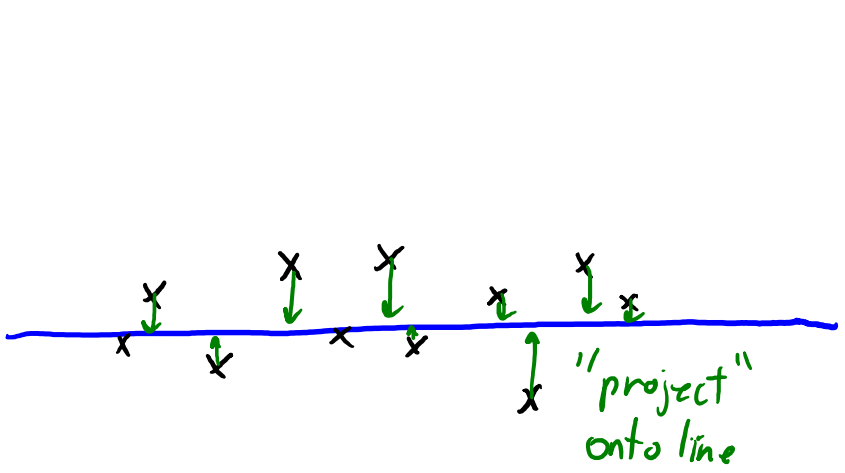


PCA with d=2 and k =1

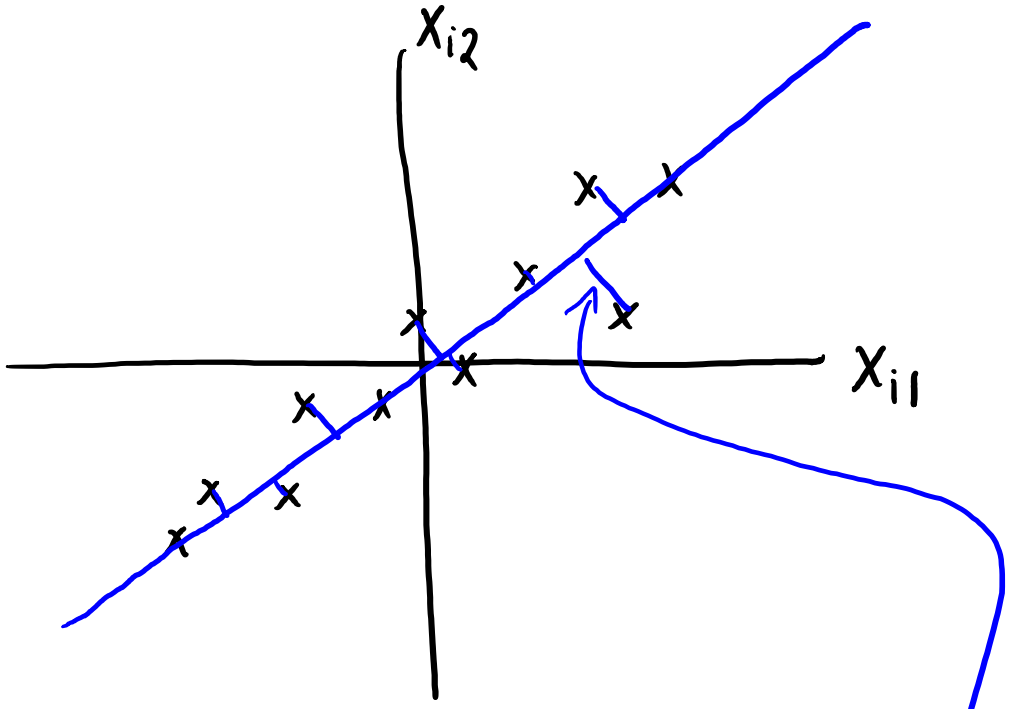


PCA with d=2 and k =1

Principal component analysis



rotate data

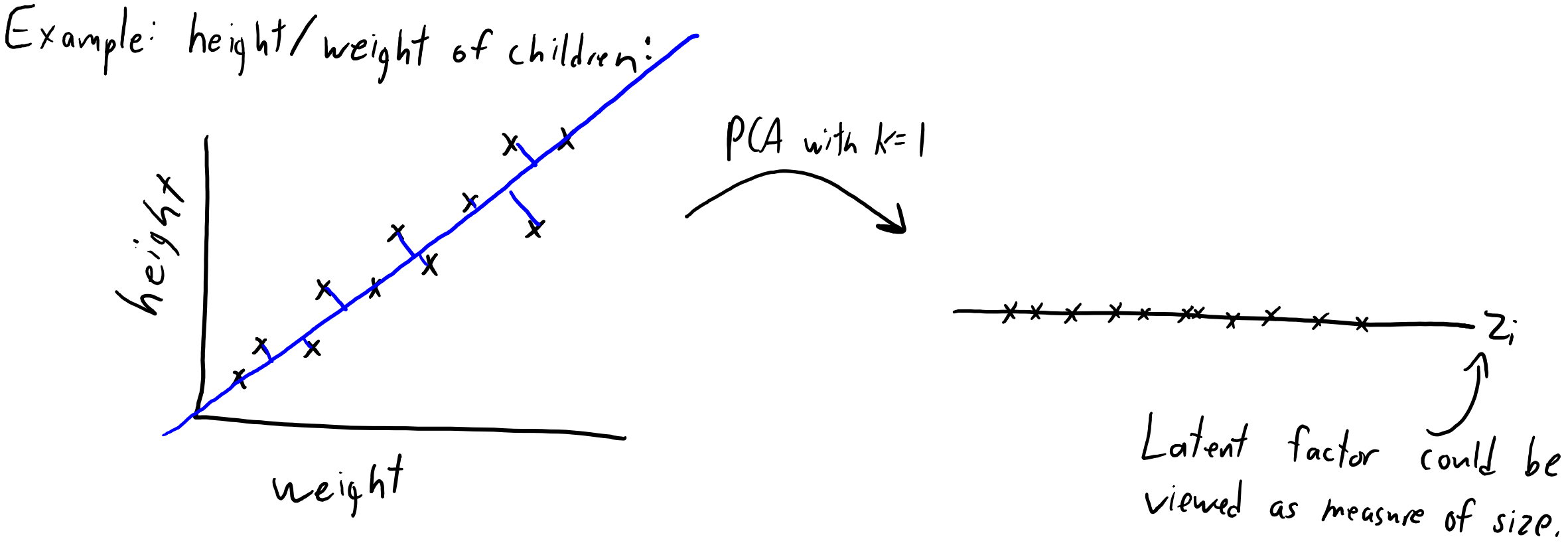


z_i can be interpreted as position along the line.

(turned a 2d problem into a 1d problem)

PCA minimizes "orthogonal" distance.

PCA with d=2 and k =1



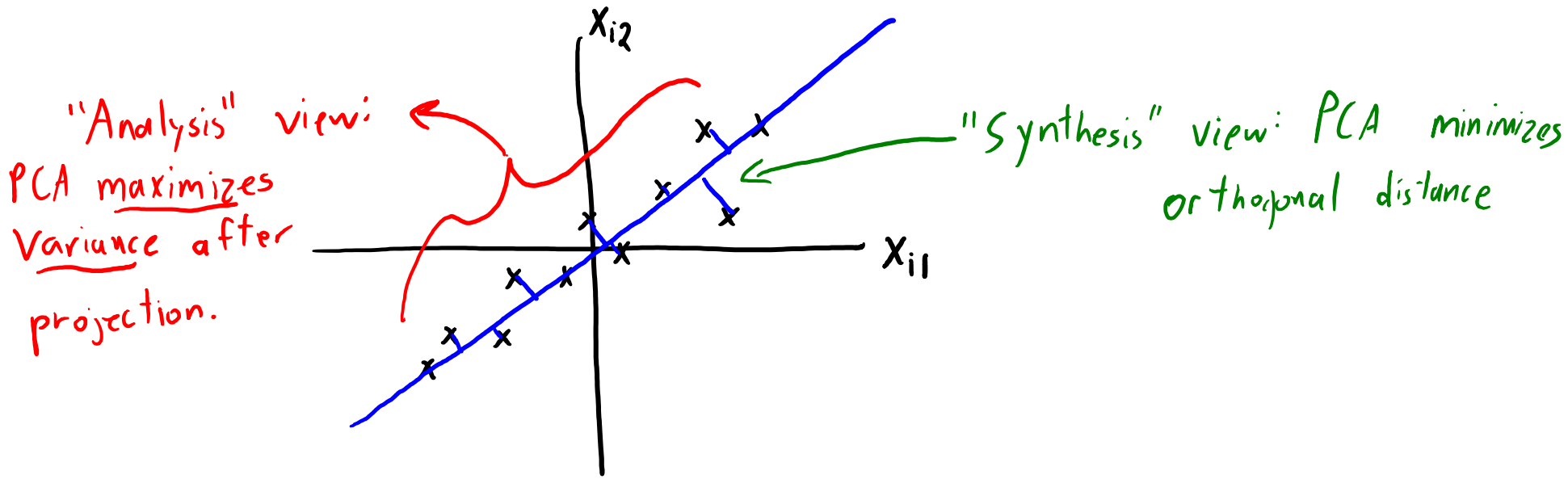
- The PCA objective with general 'd' and 'k':

$$f(W, Z) = \sum_{i=1}^{n} \sum_{j=1}^{d} (w_j^T z_i - x_{ij})^2 = \|ZW - X\|_F^2$$

  - Where we've subtracted mean μ_j from each feature.

Maximizing Variance vs. Minimizing Error

- Our "synthesis" view that PCA minimizes approximation error.
  - Makes connection to k-means and our tricks for linear regression.



- Classic "analysis" view: PCA maximizes variance in z_i space.
  - You pick 'W' to explain as much variance in the data as possible.

Choosing Number of Latent Factors

- Common approach to choosing 'k':
  - Compute error with k=0: $\|X\|_F^2 = n * var(x_{ij})$ (remember that columns have mean of zero)
  - Compare to error with non-zero 'k':

$$\frac{\|ZW - X\|_F^2}{\|X\|_F^2}$$

  - Gives a number between 0 and 1, giving how much "variance remains".
    - If you want to explain 90% of variance, choose smallest 'k' where ratio is < 0.10.

PCA Computation

- 3 common ways to solve this problem:
  - Singular value decomposition: classic non-iterative approach (bonus slide).
  - Alternating minimization:
    1. Start with random initialization.
    2. Optimize 'W' with 'Z' fixed (solve gradient with respect to 'W' equals to 0).

(writing gradient as a matrix) $\nabla_W f(W,Z) = Z^T ZW - Z^T X$ so $W = (Z^T Z)^{-1}(Z^T X)$

    3. Optimize 'Z' with 'W' fixed (solve gradient with respect to 'Z' equals to 0).

$\nabla_Z f(W,Z) = ZWW^T - XW^T$ so

$Z = XW^T(WW^T)^{-1}$

    4. Go back to 2.
  - Stochastic gradient: gradient descent based on random 'i' and 'j'.
    - (Or just plain gradient descent).

PCA Computation

- The PCA objective with general 'd' and 'k':

$$f(W, Z) = \sum_{i=1}^{n} \sum_{j=1}^{d} (w_j^T z_i - x_{ij})^2 = \|ZW - X\|_F^2$$

  - Where we've subtracted mean μ_j from each feature.

- At test time, to find optimal 'Z' given 'W' for new data:

Given factors 'W' and test data $\hat{X}$:

Subtract training mean μ_j for each feature 'j': $\hat{x}_i \leftarrow \hat{x}_i - \mu$

Solve for 'Z' given 'W': $Z = \hat{X} W^T (WW^T)^{-1}$

(If $k=1$ then $z_i = \frac{w_c^T x_i}{w_c^T w_c}$)

PCA Non-Convexity

- The PCA objective with general ‘d’ and ‘k’:

$$f(W, Z) = \sum_{i=1}^{n} \sum_{j=1}^{d} (w_j^T z_i - x_{ij})^2$$

- This objective is not jointly convex in ‘W’ and ‘Z’.
  - This is why iterative methods need random initialization.
    - If you initialize with z1 = z2, then they stay the same.
  - But it’s possible to show that all “stable” local optima are global optima.
    - So alternating minimization and stochastic gradient give global optima in practice.

PCA Non-Uniqueness

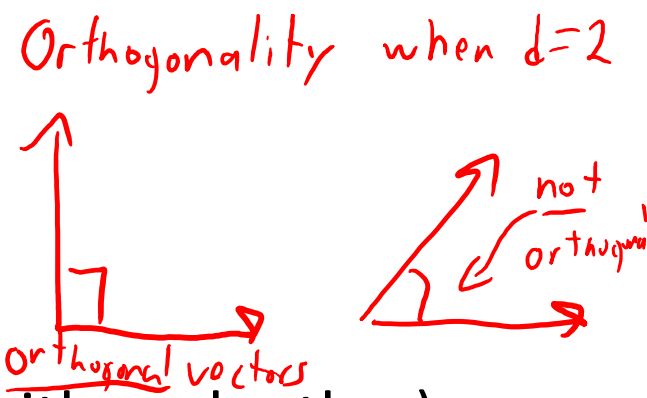
- Last time we discussed the scaling problem:

We get same $f(W, Z)$ if you replace 'W' by αW
and 'Z' by $(\frac{1}{\alpha})Z$ for any $\alpha \neq 0$

$$(\tfrac{1}{\alpha} Z)(\alpha W) = ZW$$

A standard fix: require that $\|w_c\| = 1$ for all factors 'c'.
↳ row 'c' of 'W'

PCA Non-Uniqueness



- But with multiple PCs, we have new problems:
  - Factors could be non-orthogonal (components interfere with each other):

For d=2 and k=2 an optimal solution is $W = \begin{bmatrix} 1 & 0.99 \\ 0.99 & 1 \end{bmatrix}$ ← factors are almost identical

But with d=2 and k=2 we could equivalently take $W = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ Very interpretable

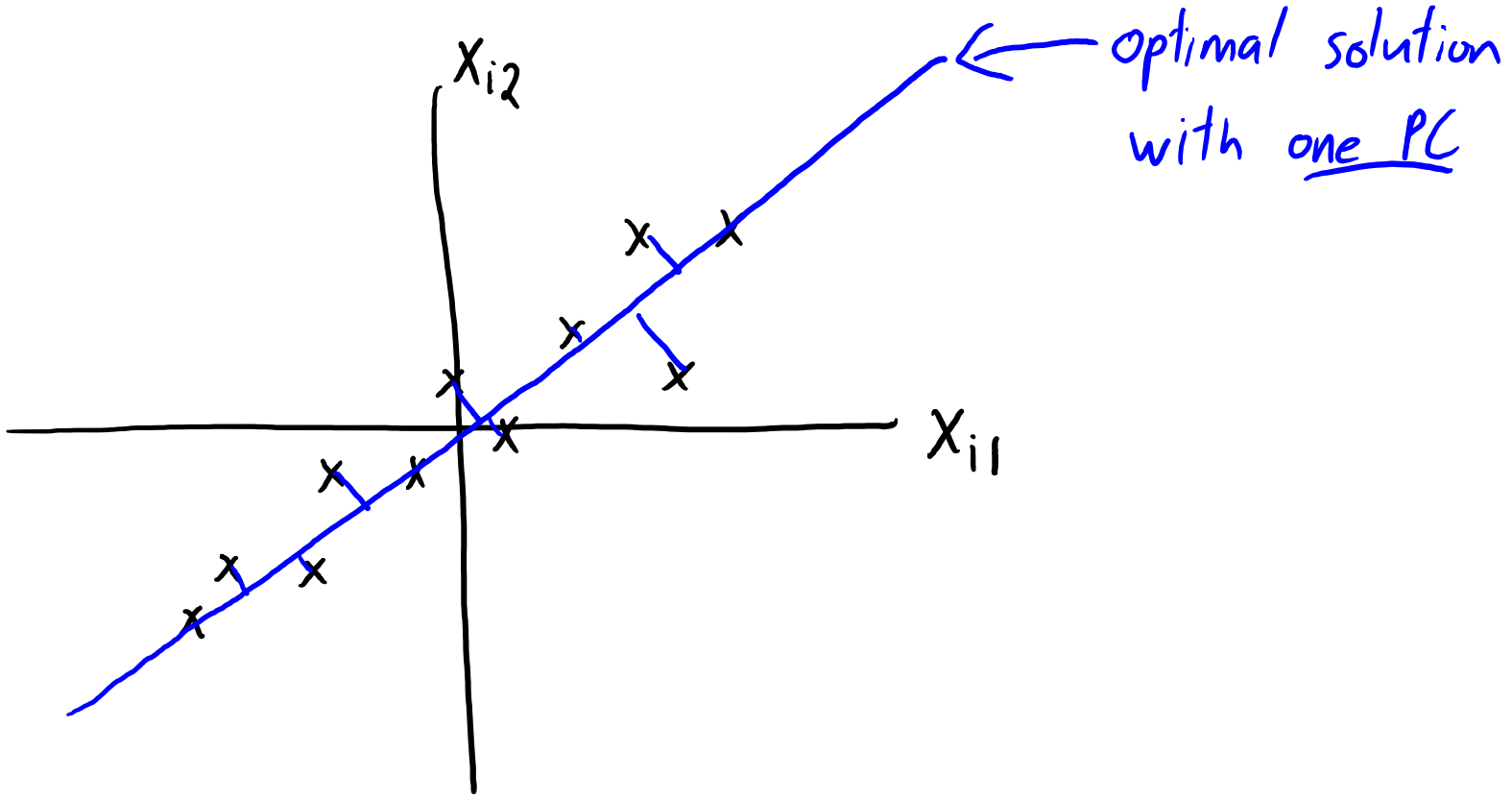
The standard fix is requiring orthogonal factors: $w_c^T w_{c'} = 0$ when $c \neq c'$

  - You can still "rotate" the factors and also have label switching.
    - A fix is to fit the PCs sequentially (can be done with SVD approach):

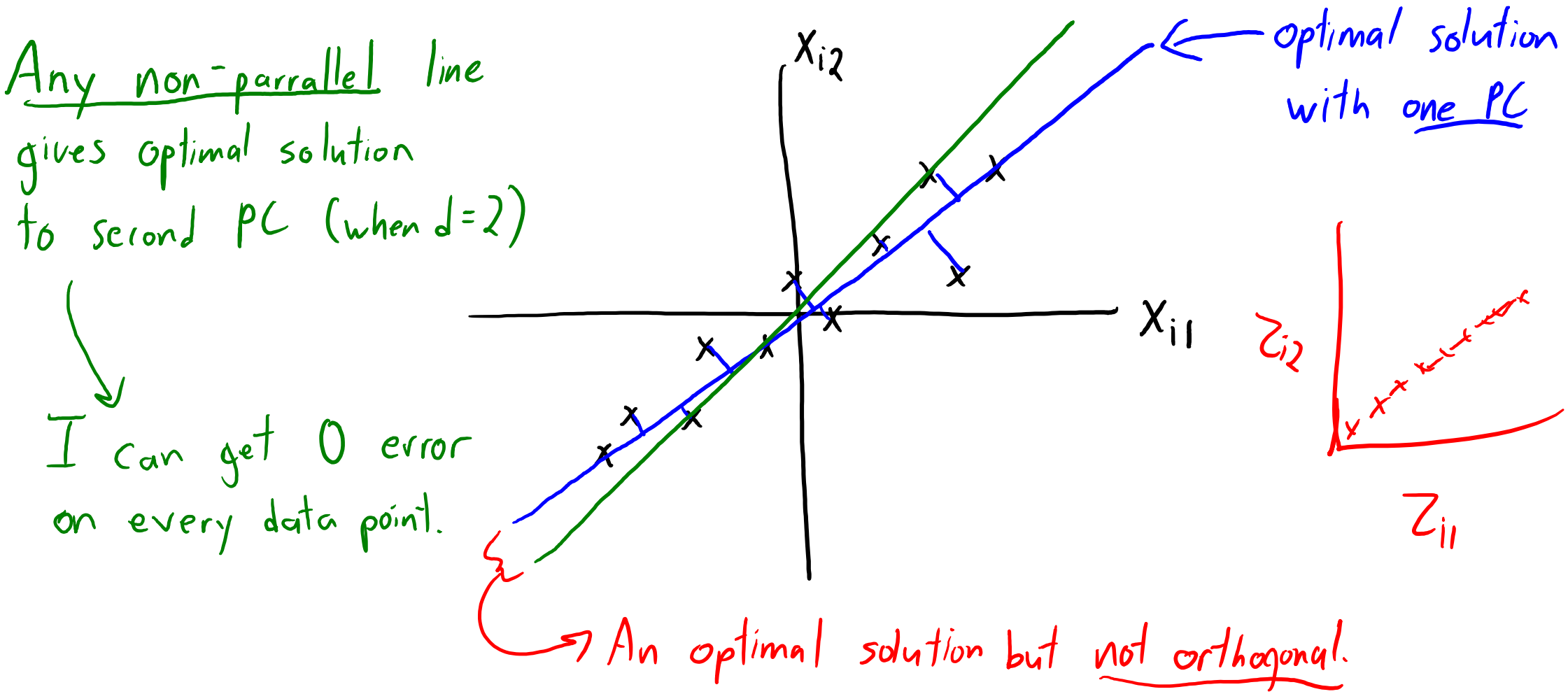
1. Find "first" PC w_c that minimizes $\|zw_c^T - X\|_F^2$ (PCA with k=1)
2. Fix "first" PC w_1 and find w_c minimizing $\|ZW - X\|_F^2$ where $w_1^T w_c = 0$
3. Fix "first" and "second" PC and find w_c with $w_1^T w_c = 0$ and $w_2^T w_c = 0$

⋮

Basis, Orthogonality, Sequential Fitting

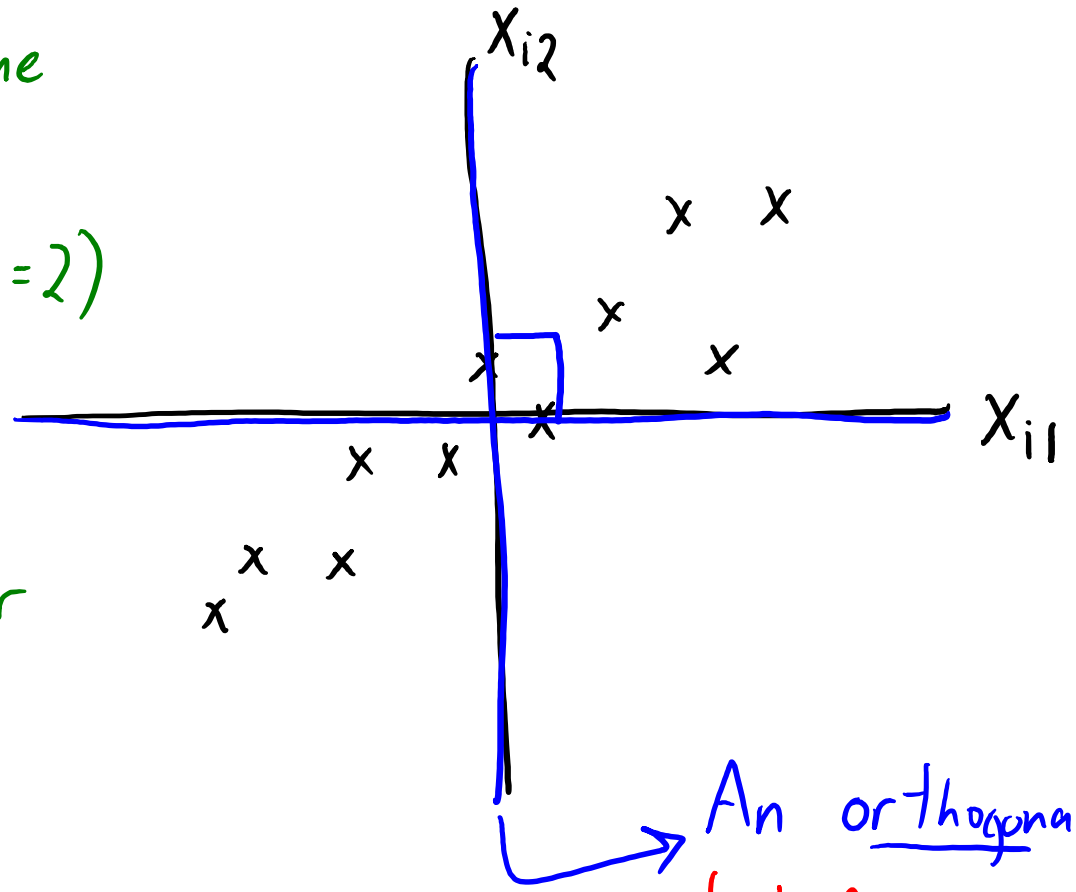


Basis, Orthogonality, Sequential Fitting



Basis, Orthogonality, Sequential Fitting

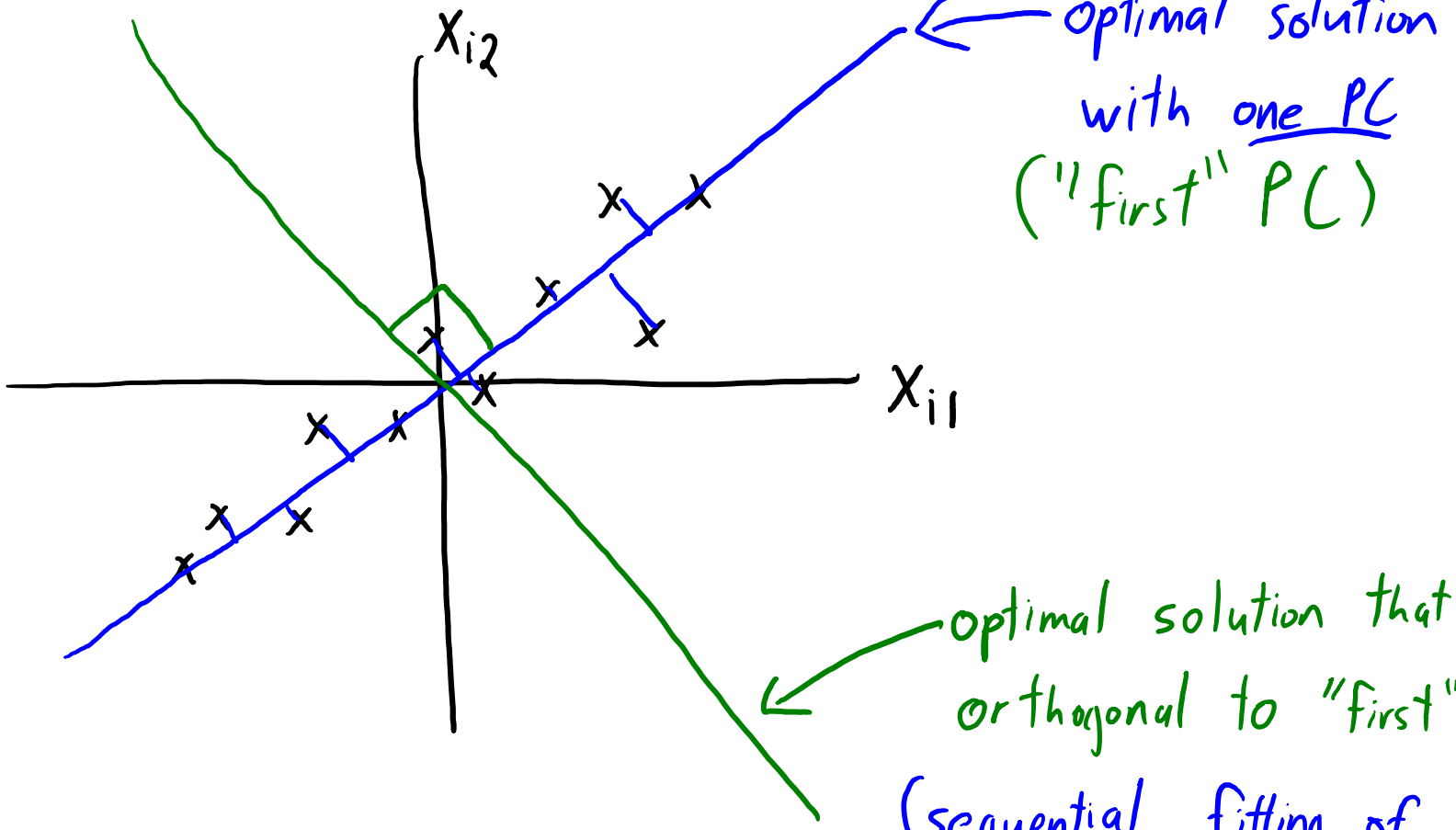
Any non-parrallel line gives optimal solution to second PC (when d=2)



I can get 0 error on every data point.

An orthogonal solution (PCs are not redundant)
but PCs have nothing to do with data

Basis, Orthogonality, Sequential Fitting



http://setosa.io/ev/principal-component-analysis

Notice that if we require $\|w_c\| = 1$ then $w_c^T w_c = 1$

and if we also require orthogonality $w_c^T w_{c'} = 0$ for $c \neq c'$ then:

$$WW^T = \begin{bmatrix} — & w_1^T & — \\ — & w_2^T & — \\ & \vdots & \\ — & w_K^T & — \end{bmatrix} \begin{bmatrix} | & | & & | \\ w_1 & w_2 & \cdots & w_K \\ | & | & & | \end{bmatrix} = \begin{bmatrix} w_1^T w_1 & w_1^T w_2 & \cdots & w_1^T w_K \\ w_2^T w_1 & w_2^T w_2 & \cdots & w_2^T w_K \\ \vdots & & & \vdots \\ w_K^T w_1 & w_K^T w_2 & \cdots & w_K^T w_K \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & & \vdots \\ 0 & 0 & 1 & \ddots & \\ \vdots & & & & 0 \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} = I$$

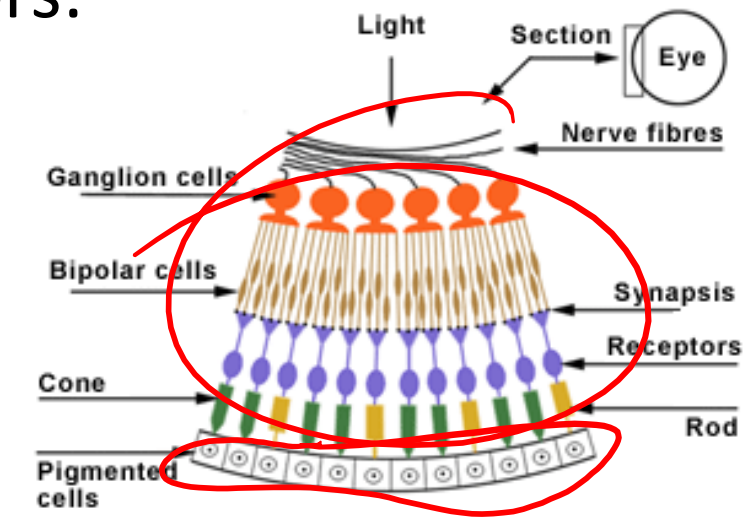
So finding Z simplifies: $Z = XW^T(WW^T)^{-1} = XW^T$

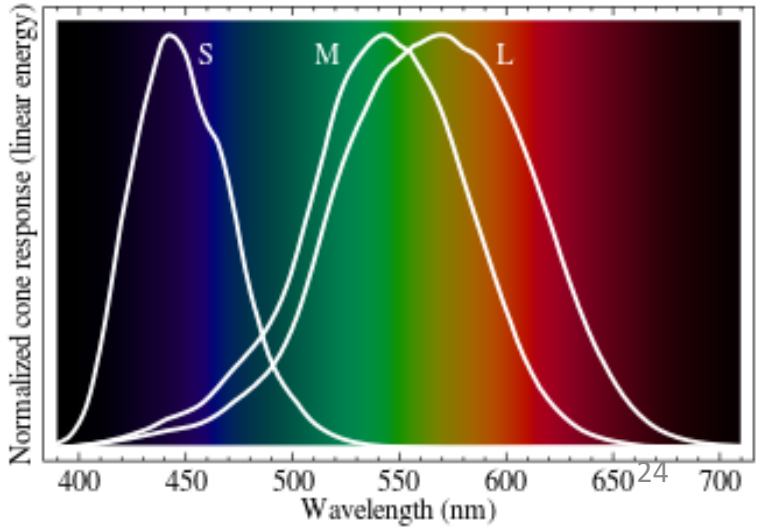
(If $K=1$ then $z_i = \underbrace{w_c^T x_i}_{w_c^T w_c} = w_c^T x_i$)

Do I need all this math?

Colour Opponency in the Human Eye

- Classic model of the eye is with 4 photoreceptors:
  - Rods (more sensitive to brightness).
  - L-Cones (most sensitive to red).
  - M-Cones (most sensitive to green).
  - S-Cones (most sensitive to blue).
- Two problems with this system:
  - Correlation between receptors (not orthogonal).
    - Particularly between red/green.
  - We have 4 receptors for 3 colours.

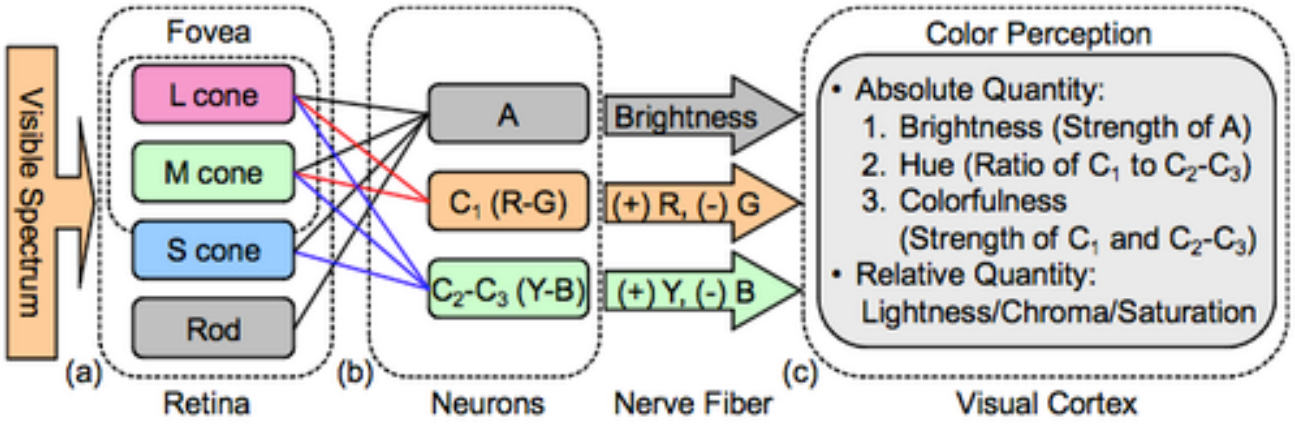


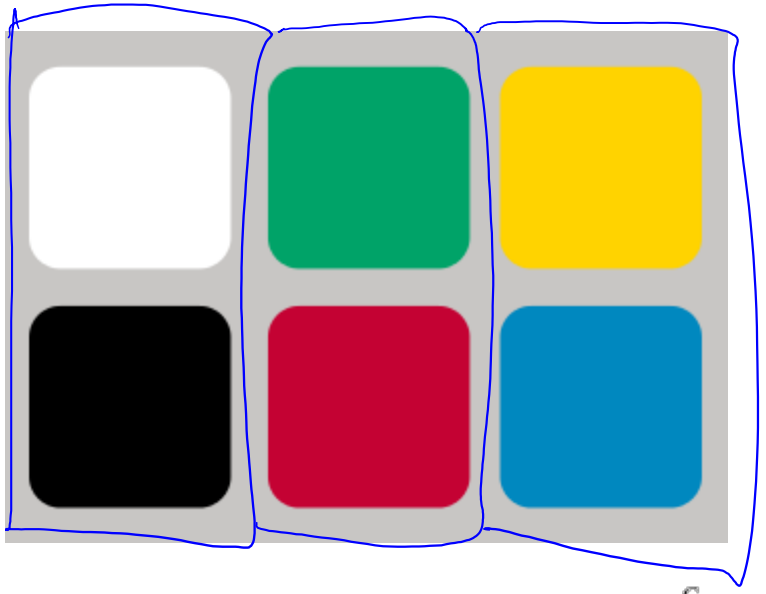


http://oneminuteastronomer.com/astro-course-day-5/
https://en.wikipedia.org/wiki/Color_visio

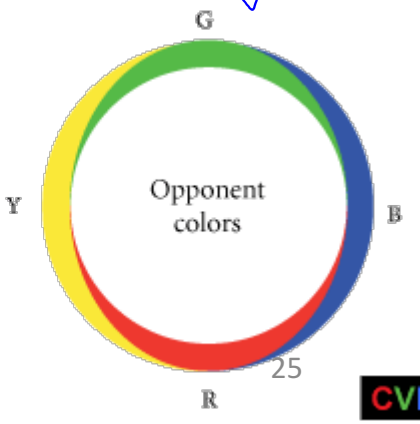
Colour Opponency in the Human Eye

- Bipolar and ganglion cells seem to code using “opponent colors”:
  - 3-variable orthogonal basis:





- This is similar to PCA (d = 4, k = 3).



http://oneminuteastronomer.com/astro-course-day-5/
https://en.wikipedia.org/wiki/Color_visio
http://5sensesnews.blogspot.ca/

Colour Opponency Representation

For this pixel, eye gets 4 signals

Can represent 4 original values with these 3 z_i values and matrix 'W'

$= w_1$

First row of W (First PC)

Analogous to means in k-means.

brightness

$+ w_2$

Second row (4×1)

red/green

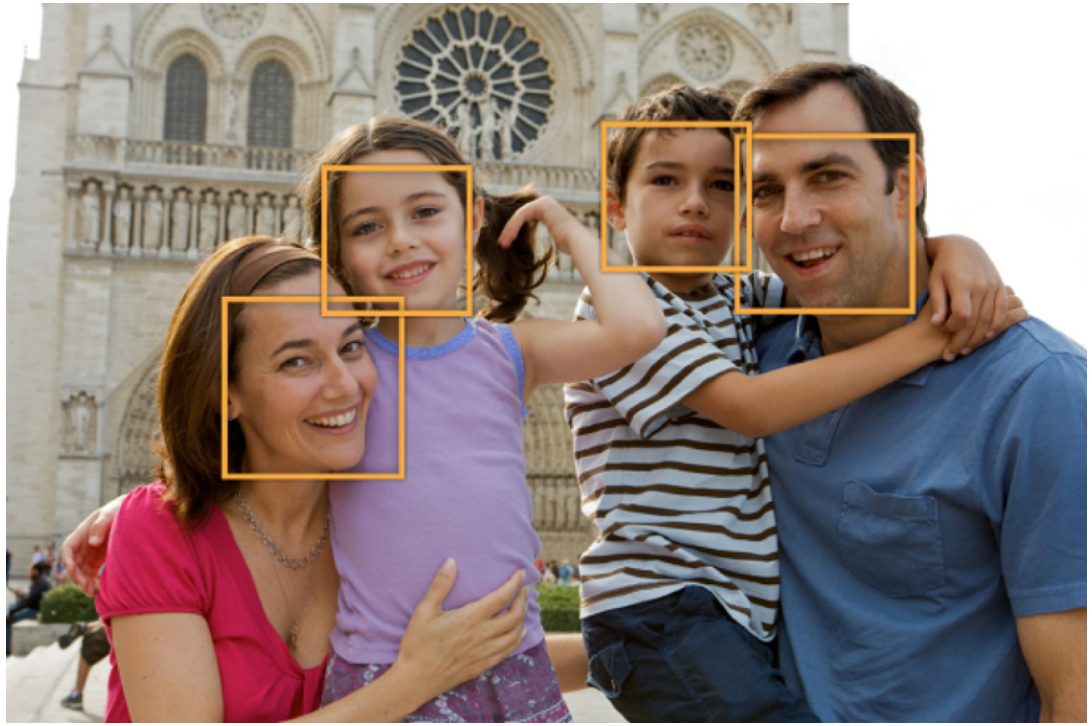
$+ w_3$

Third row (4×1)

blue/yellow

Application: Face Detection

- Consider problem of face detection:

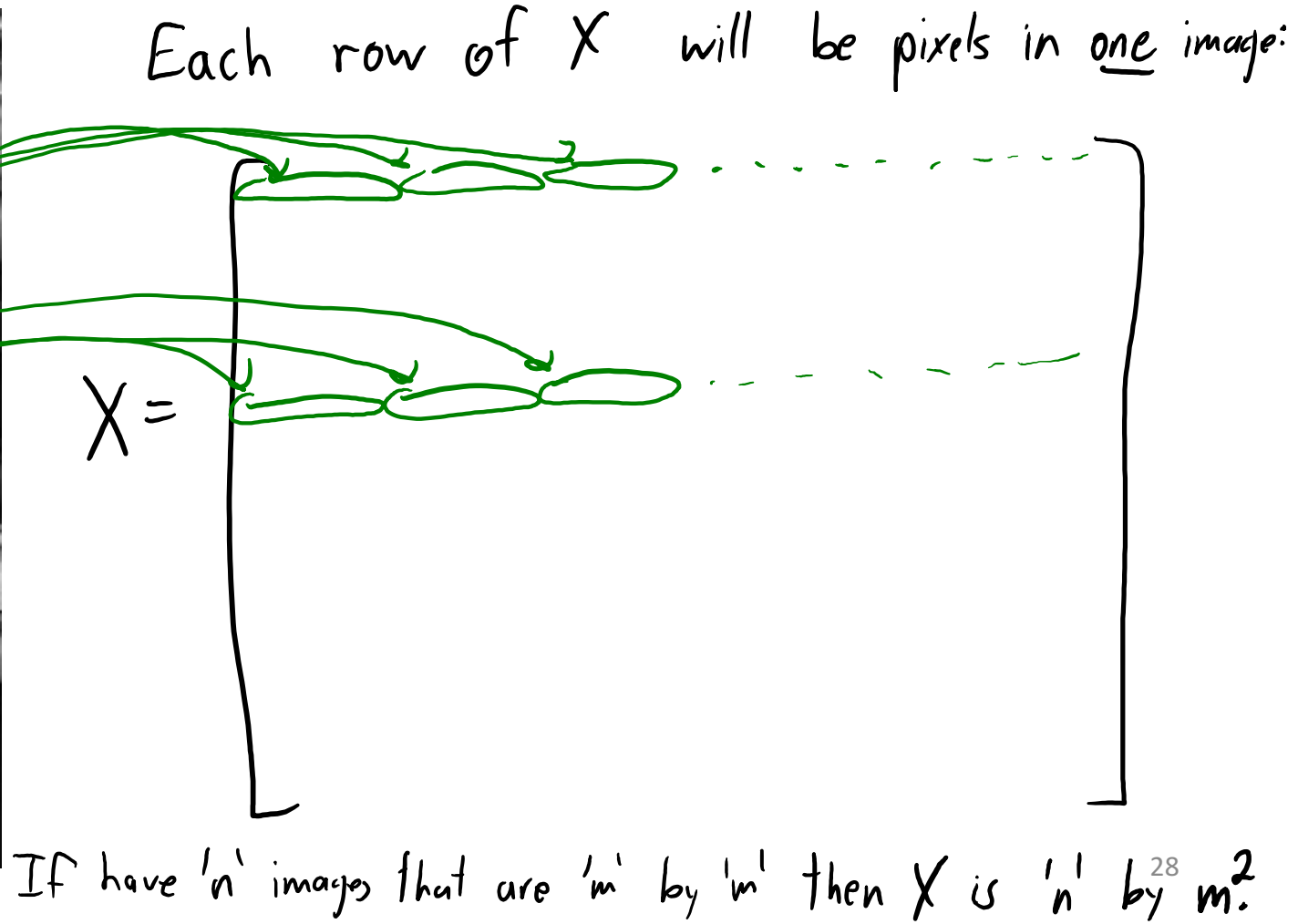


- Classic methods use “eigenfaces” as basis:
  - PCA applied to images of faces.

https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_detect_faces/ci_detect_faces.html

Eigenfaces

- Collect a bunch of images of faces under different conditions:



Eigenfaces

Compute mean μ_j of each column.

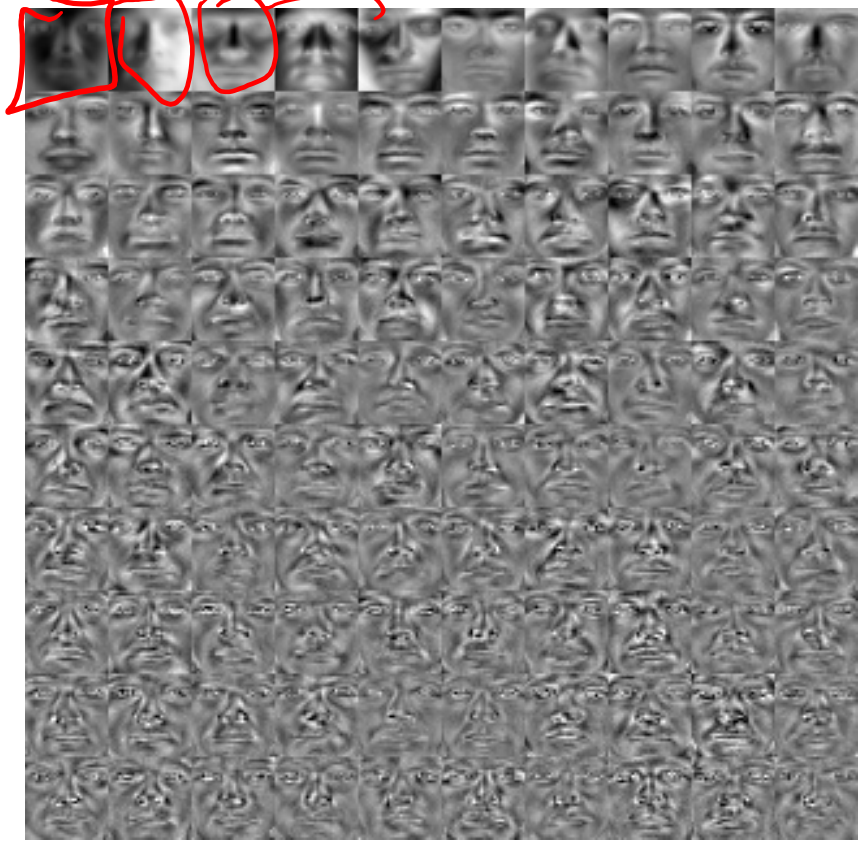
Each row of X will be pixels in one image:

$$X = \begin{bmatrix} — & x_1 - \mu & — \\ — & x_2 - \mu & — \\ & \vdots & \\ — & x_n - \mu & — \end{bmatrix}$$

Replace each x_{ij} by $x_{ij} - \mu_j$

Eigenfaces

Compute top 'k' PCs on centered data:

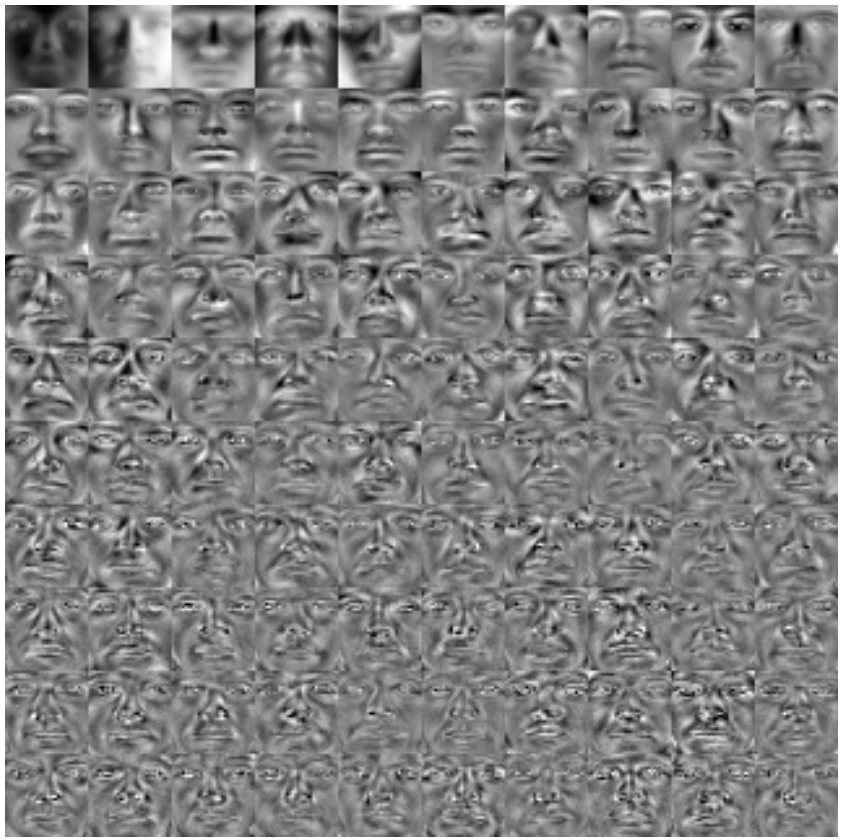


Each row of X will be pixels in one image:

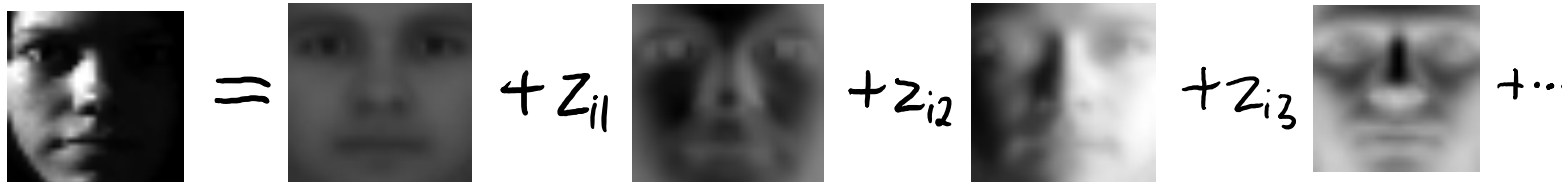
$$X = \begin{bmatrix} — & x_1 - \mu & — \\ — & x_2 - \mu & — \\ & \vdots & \\ — & x_n - \mu & — \end{bmatrix}$$

Eigenfaces

Compute top 'k' PCs on centered data:



"Eigenface" representation:



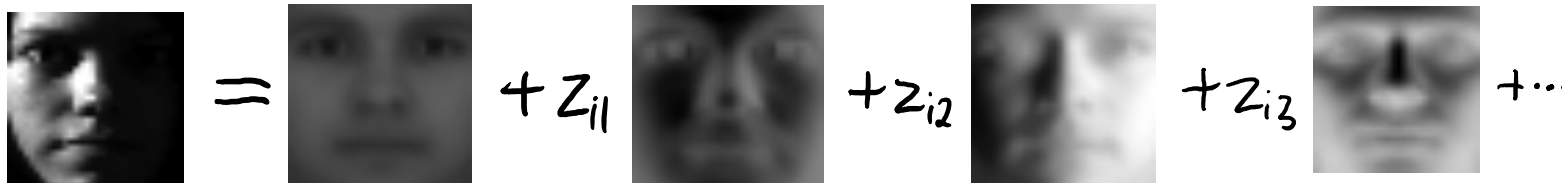
x_i μ PC1 PC2 PC3

(first row of W)

Eigenfaces

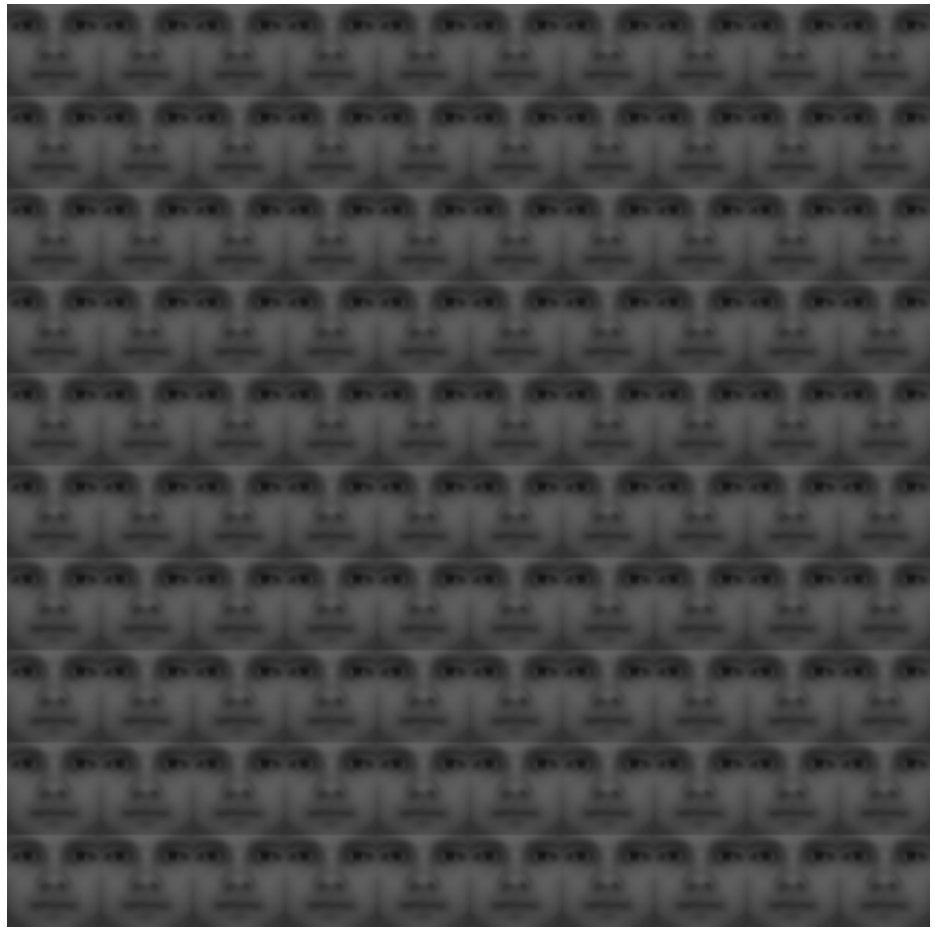
100 of the original faces:

"Eigenface" representation:



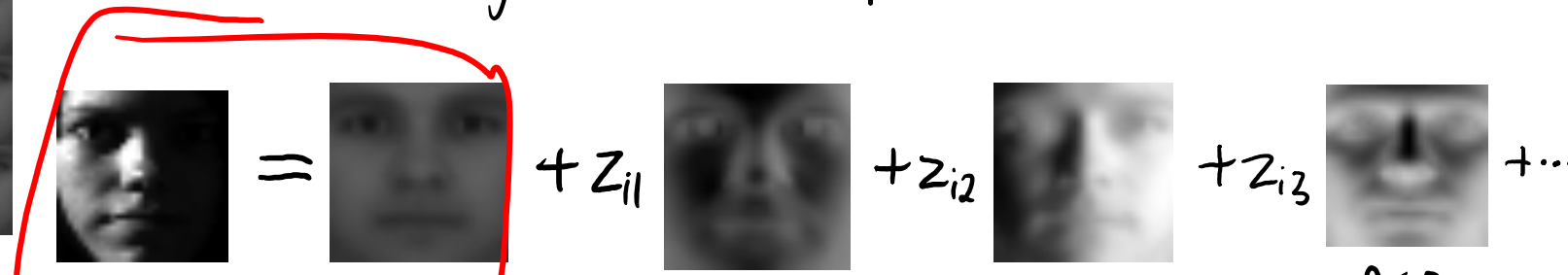
Eigenfaces

Reconstruction with $k = 0$



Variance explained: 0%

"Eigenface" representation:

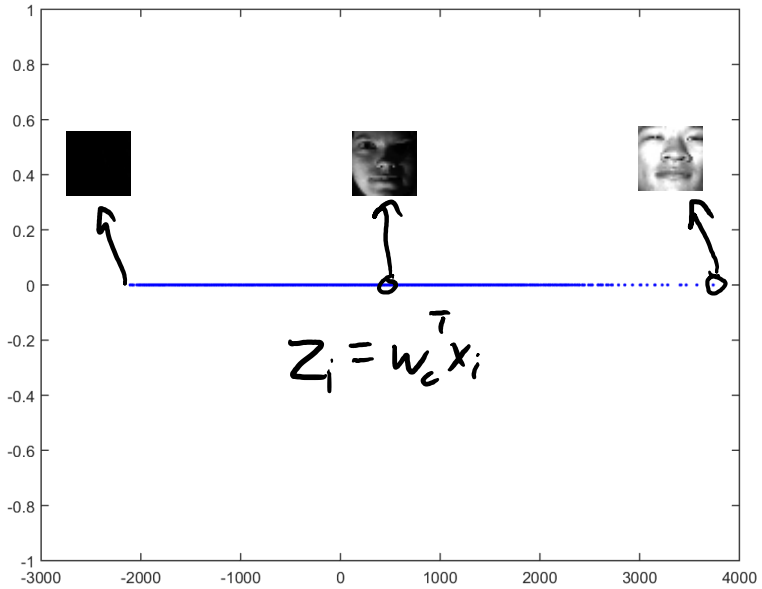


Eigenfaces

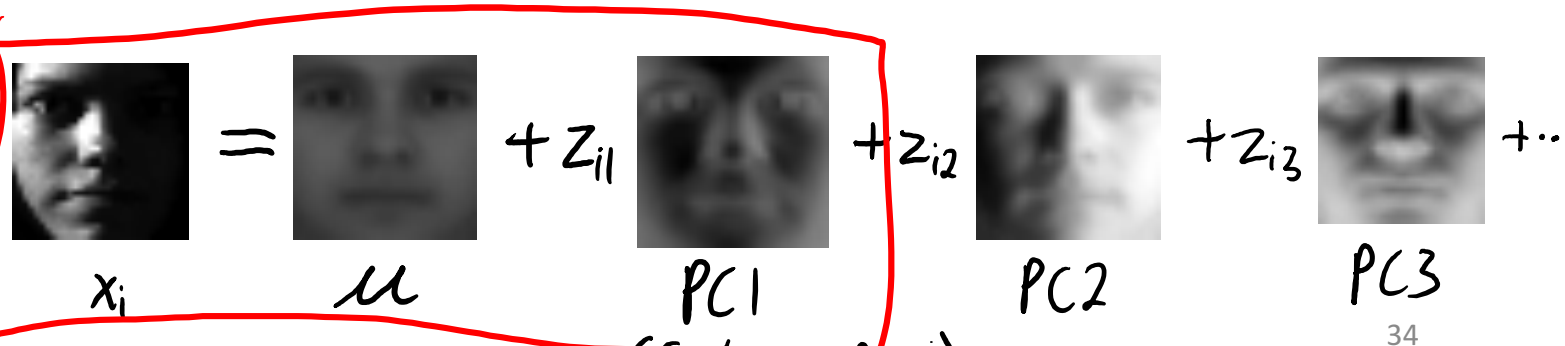
Reconstruction with $k=1$

Variance explained: 34%

PCA Visualization:



"Eigenface" representation:

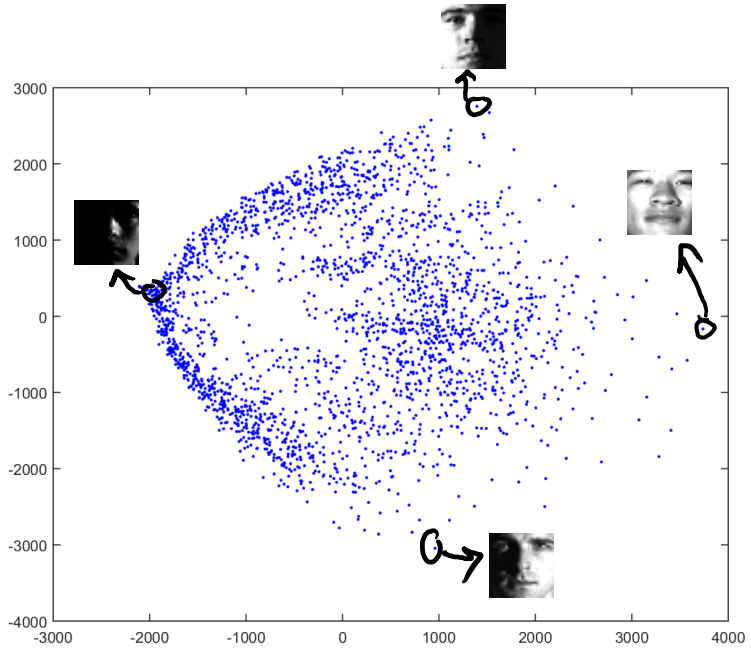


Eigenfaces

Reconstruction with $k = 2$

Variance explained: 71%

PCA Visualization:



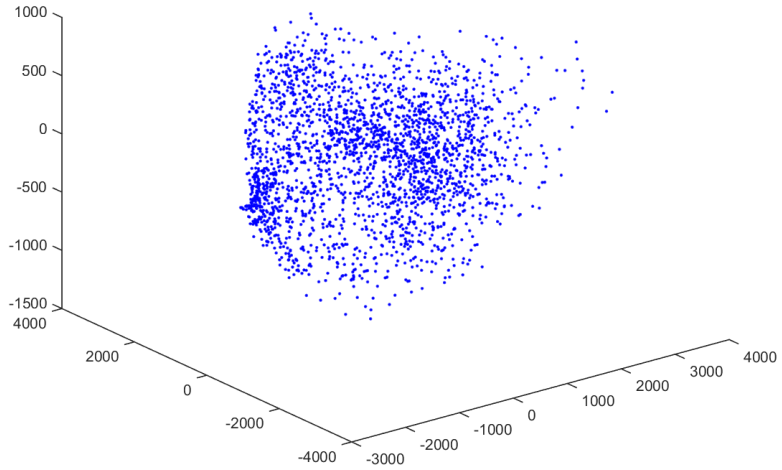
"Eigenface" representation:

Eigenfaces

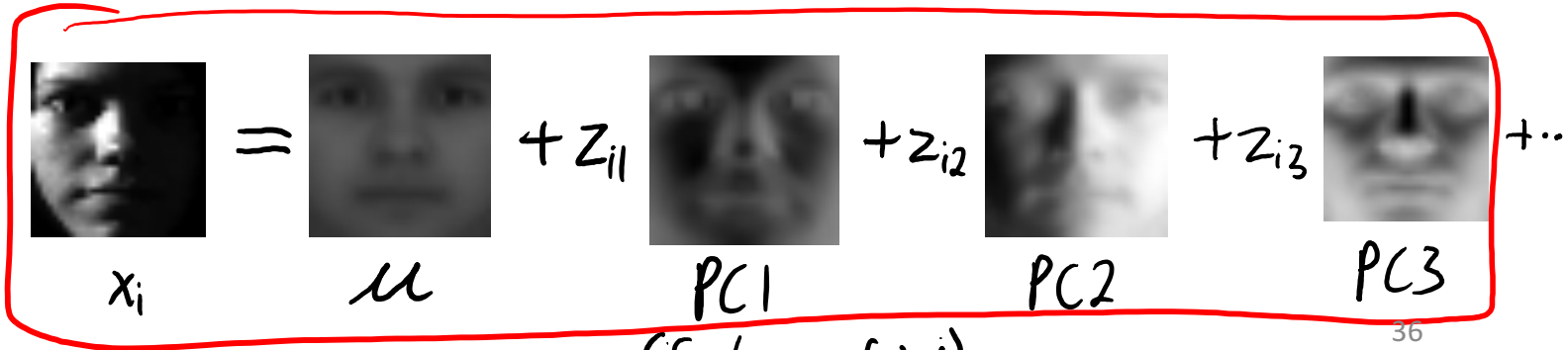
Reconstruction with $k = 3$

Variance explained: 76%

PCA Visualization:



"Eigenface" representation:

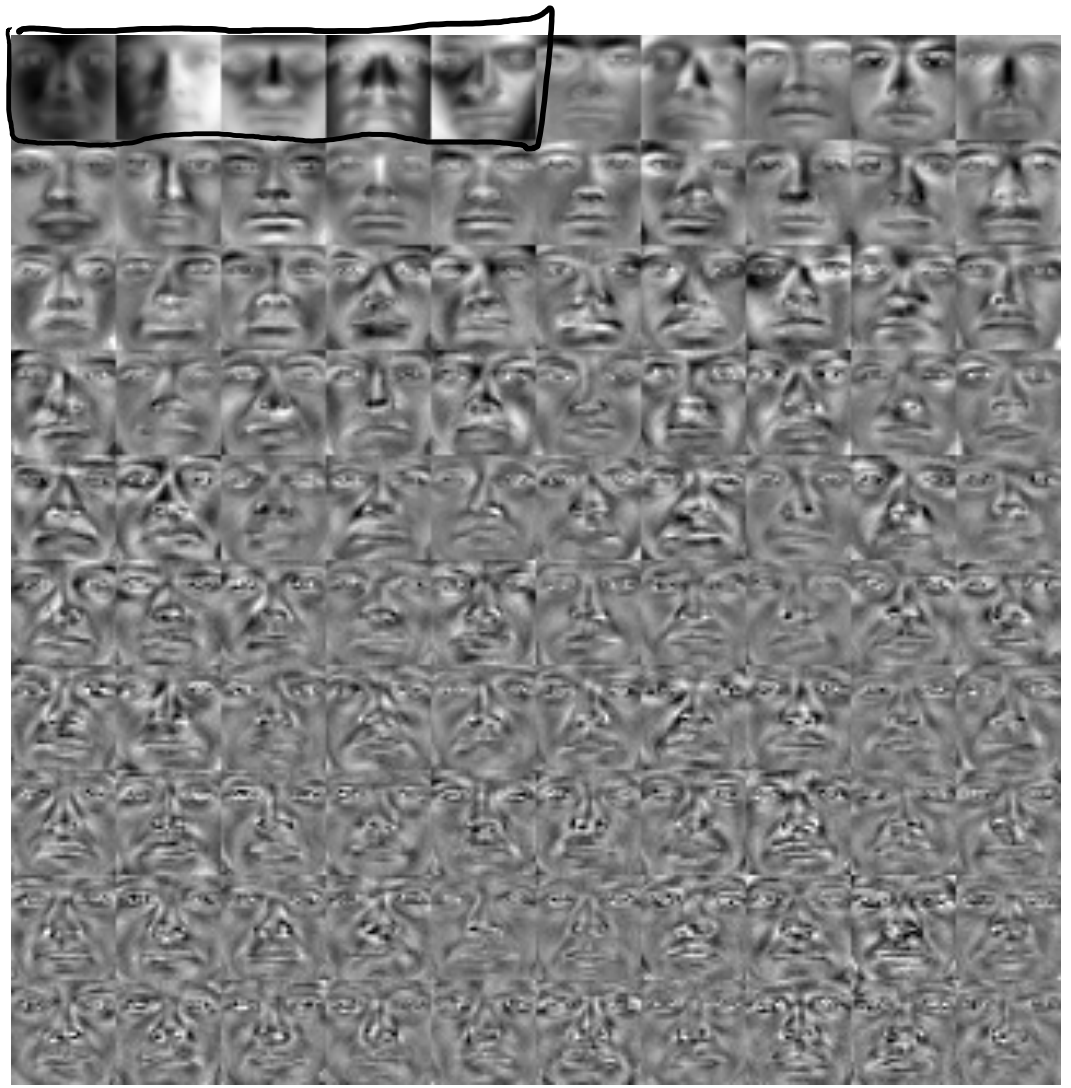


(first row of W)

Eigenfaces

Reconstruction with $k=5$

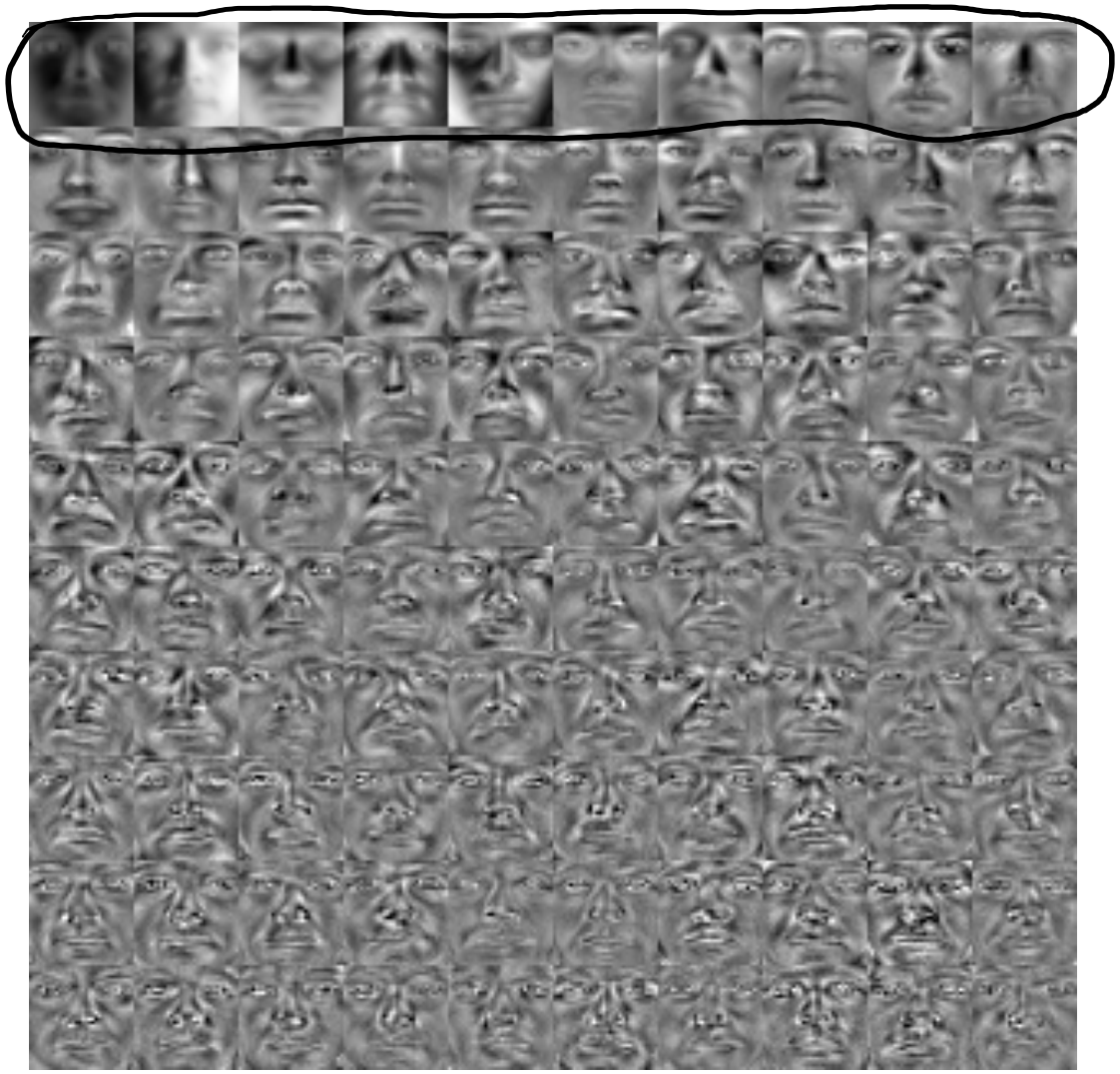
Variance explained: 80%



Eigenfaces

Reconstruction with k = 10

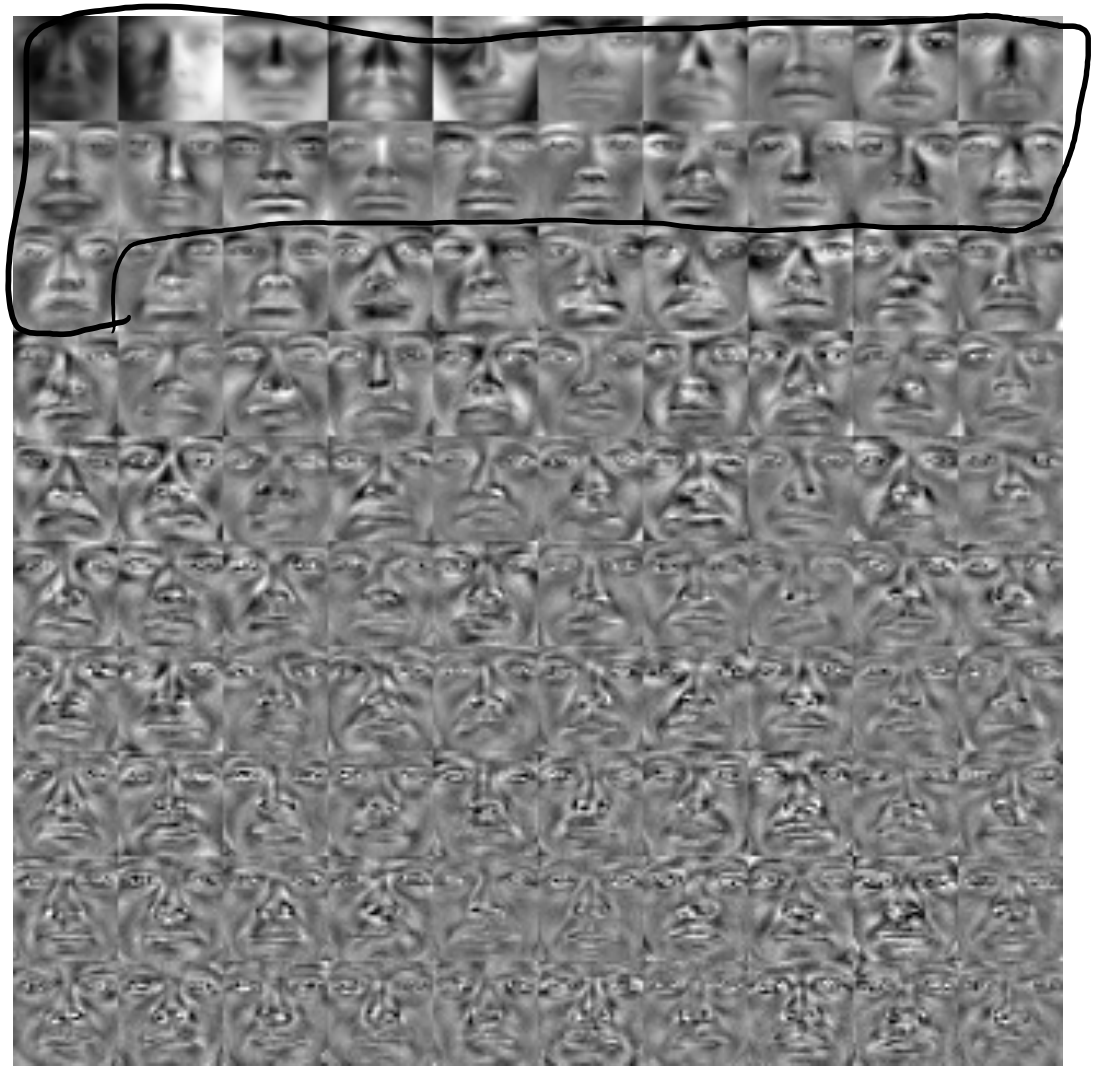
Variance explained: 85%



Eigenfaces

Reconstruction with k=21

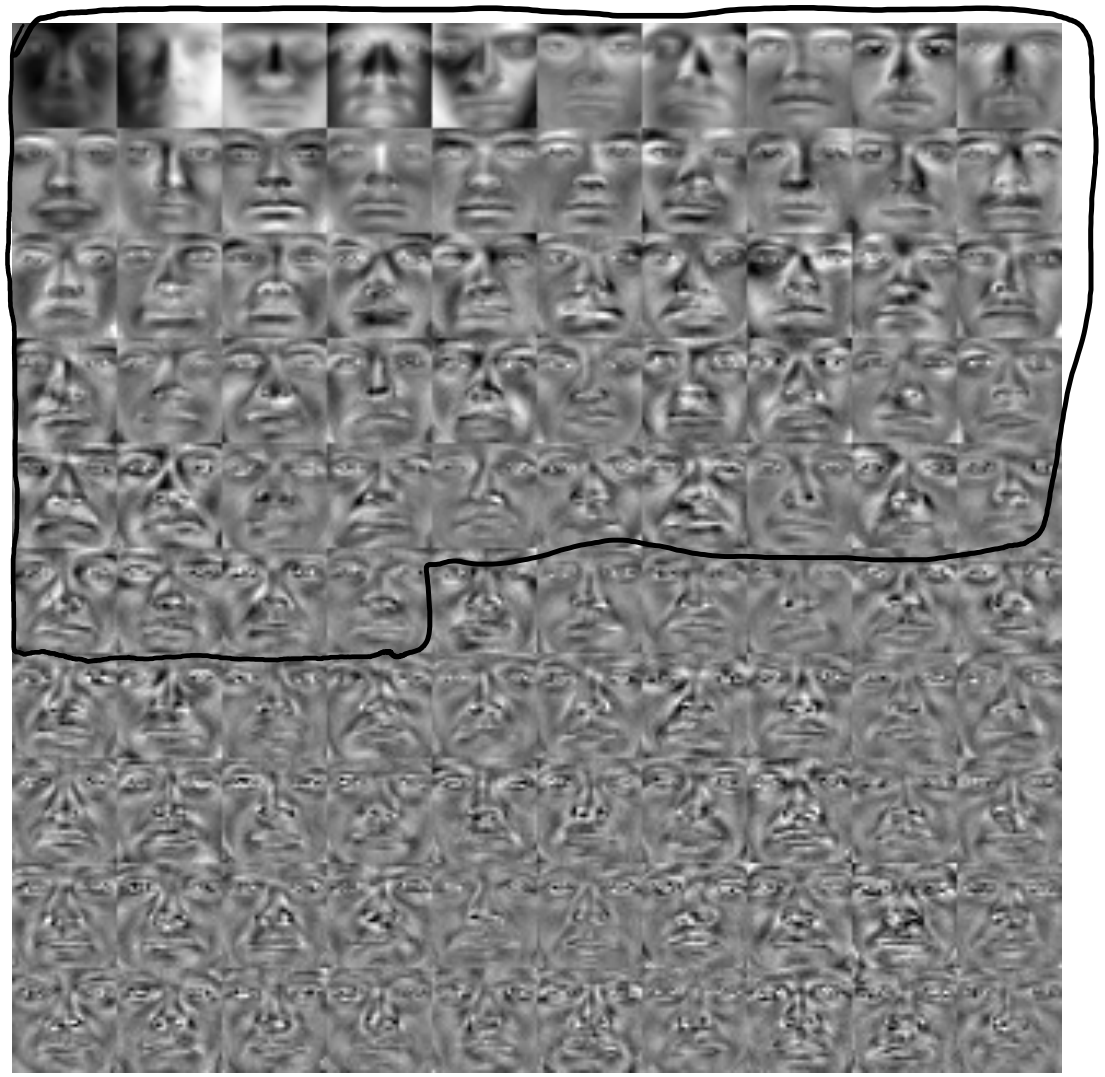
Variance explained: 90%



Eigenfaces

Reconstruction with k = 54

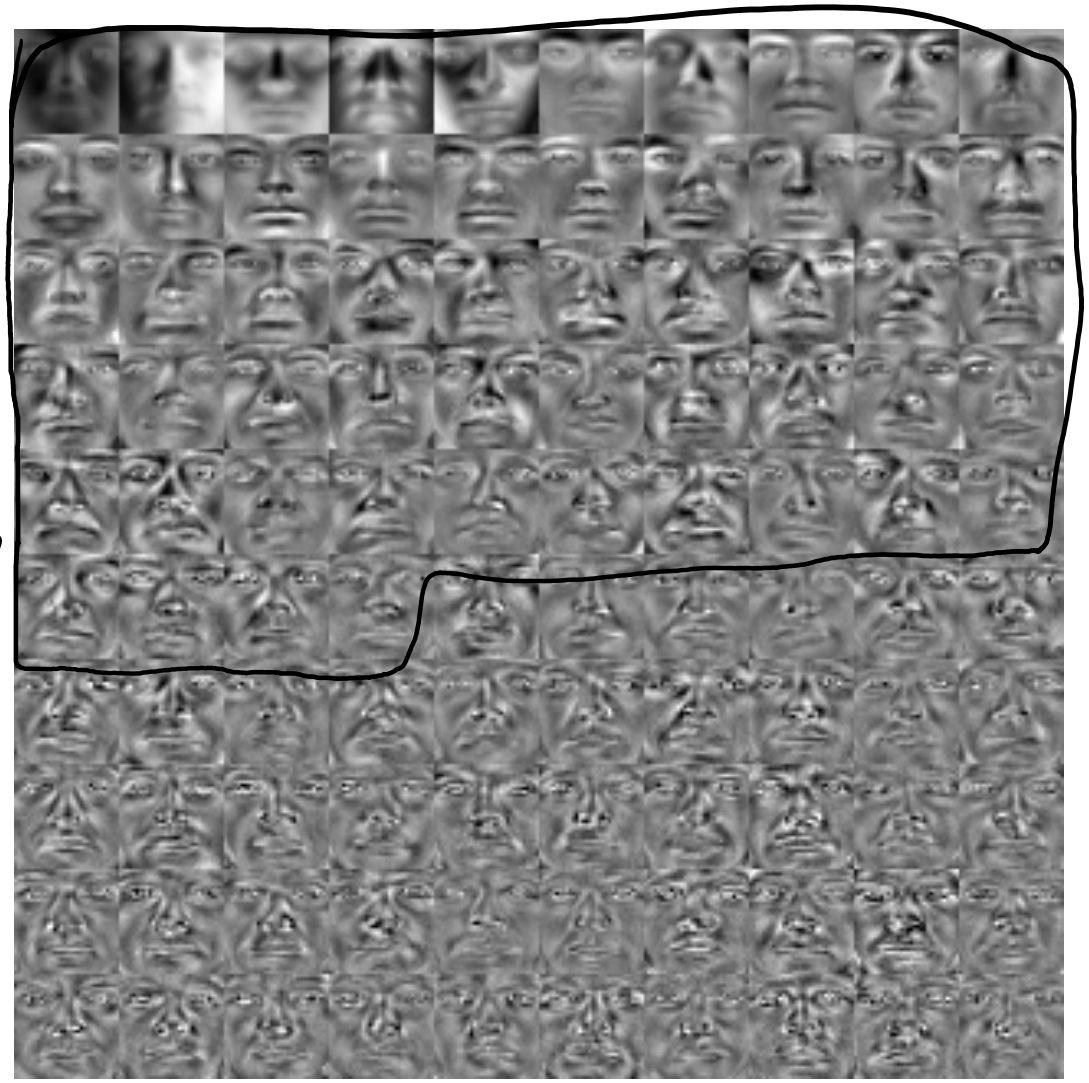
Variance explained: 95%



Eigenfaces

Original Images again:

Plus these "eigenfaces" and the mean,



We can replace 1024 x_i values by 54 z_i values

Summary

- Analysis view of PCA is that it maximizes variance.
  - We can choose ‘k’ to explain x% of the variance in the data.
- Orthogonal basis and sequential fitting of PCs:
  - Leads to non-redundant PCs with unique directions.
- Biological motivation for orthogonal and/or sparse latent factors.
- Next time: modifying PCA so it splits faces into ‘eyes’, ‘mouths’, etc.

Bonus Slide: PCA with Singular Value Decomposition

- Under constraints that $w_c^T w_c = 1$ and $w_c^T w_{c'} = 0$, use:

$$U \Sigma V^T = SVD(X)$$

$$W = V(:, 1:k)^T \qquad Z = XW^T$$

- You can also quickly get compressed version of new data:

$$\hat{Z} = \hat{X} W^T$$

- If W was not orthogonal, could get Z by least squares.
- In python, numpy.linalg.svd