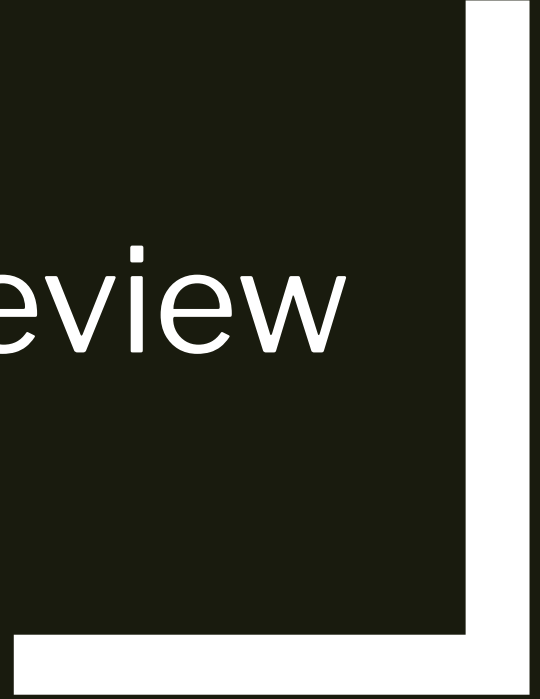


# PCA Review



# Refresher Question

- PCA always produces a unique solution (True/False)

# Refresher Question

- PCA always produces a unique solution (True/False)
- Answer: False. PCA does not produce a unique solution since solutions are equivalent under scalar multiplication.

# Refresher Question

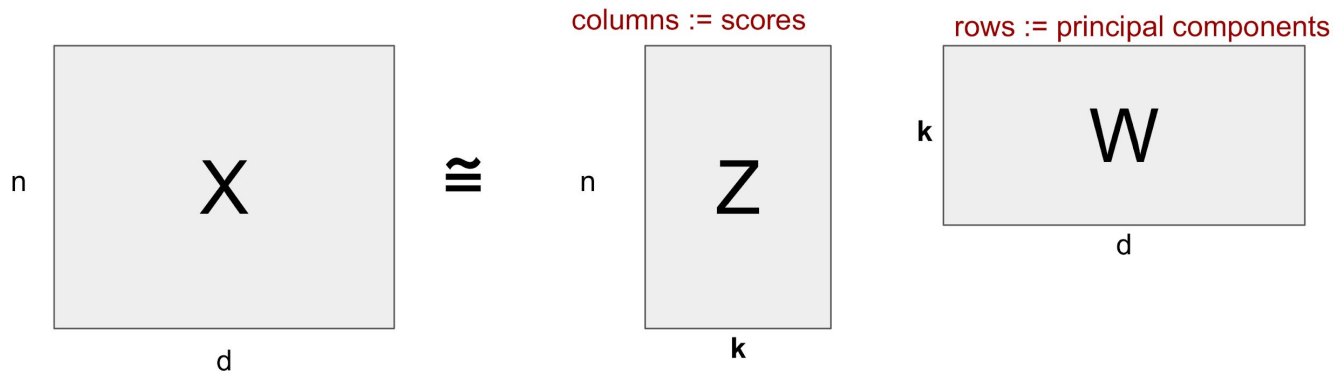
- What modifications do we make to PCA to make our solutions unique?

# Refresher Question

- What modifications do we make to PCA to make our solutions unique?
- Answer: Normalization, orthogonality, sequential fitting

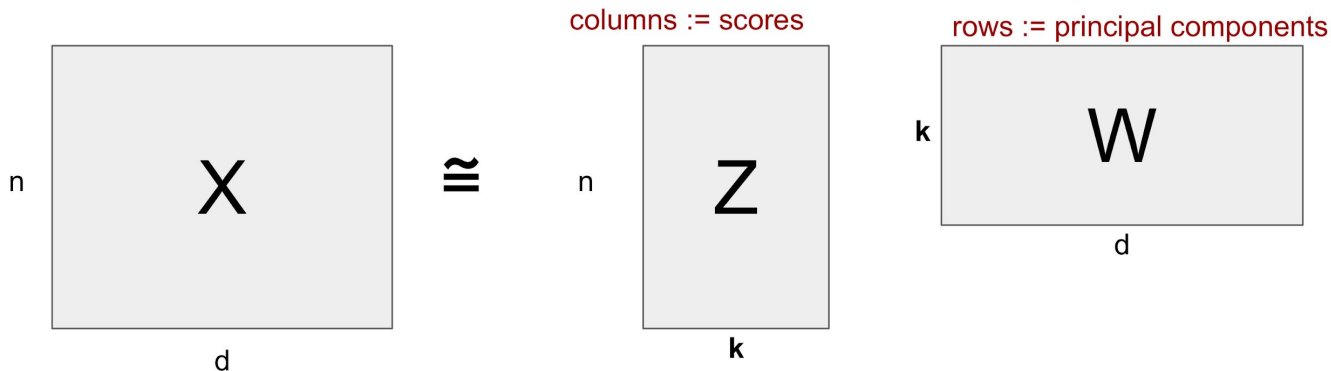
# Reminder: PCA

- **P**roincipal **C**omponent **A**nalysis
  - Factor our data into "components" that best describe variance observed in the features
- Data is factored into two matrices:
  - $W$  contains the principal components
    - Each principal component can be thought of as a vector
  - $Z$  contains the scores
    - How much of each component to mix to recover our example
- Thus we are trying to break down our data into a set of vectors that best describe the variance, and the linear combination of those vectors that we have to take to obtain our example in that basis



# Reminder: PCA

- In particular when  $k < d$ , PCA can be thought of as examining our data in a lower dimensional feature space
  - Each example is described by  $k$  features in the matrix  $Z$  as opposed to  $d$  scores in the matrix  $X$
- PCA is a *latent factor model*
  - Output of PCA is a basis that factors our data



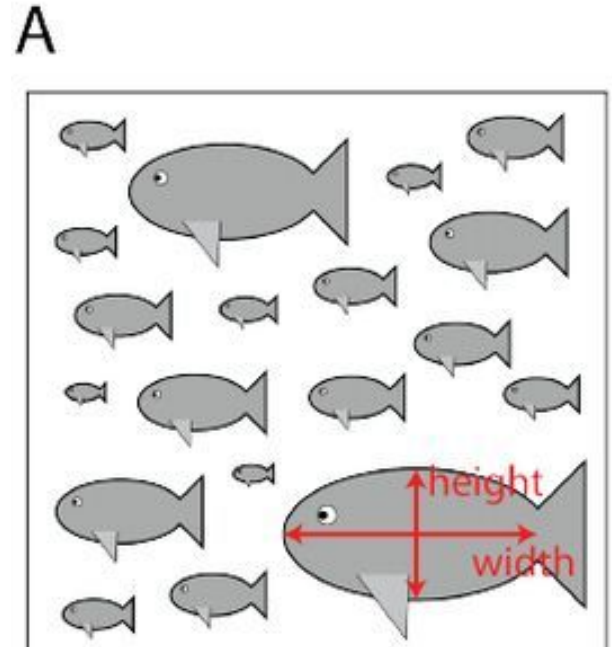
# PCA Example Questions





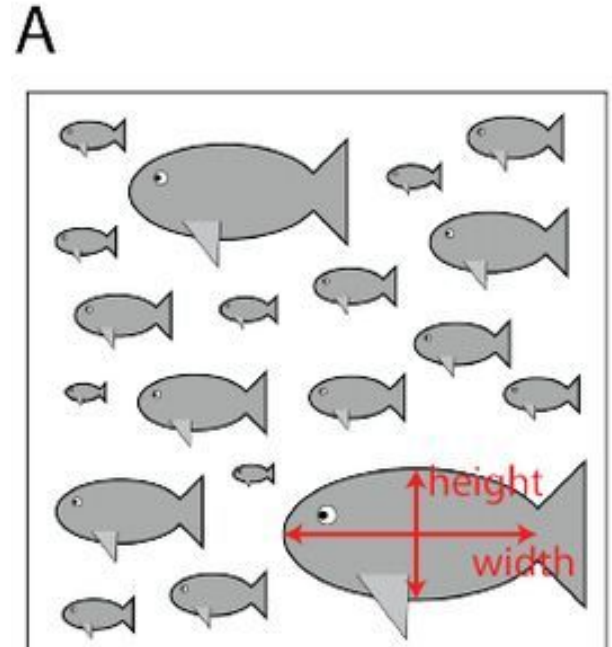
# PCA Example

- Suppose we have a population of fish, and our features for each fish consist of its width and height
- Suppose we want to reduce the dimensionality of our data
- How might PCA be applied to this dataset?



# PCA Example

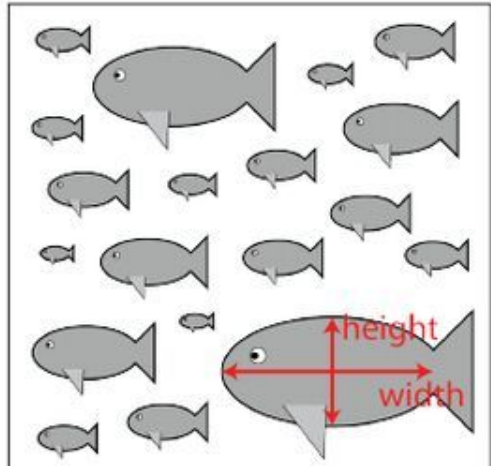
- Suppose we have a population of fish, and our features for each fish consist of its width and height
- Suppose we want to reduce the dimensionality of our data
- How might PCA be applied to this dataset?
- **Answer: We might use PCA to describe our fish population in a single dimension rather than 2 (reduce the dimensionality of our data)**



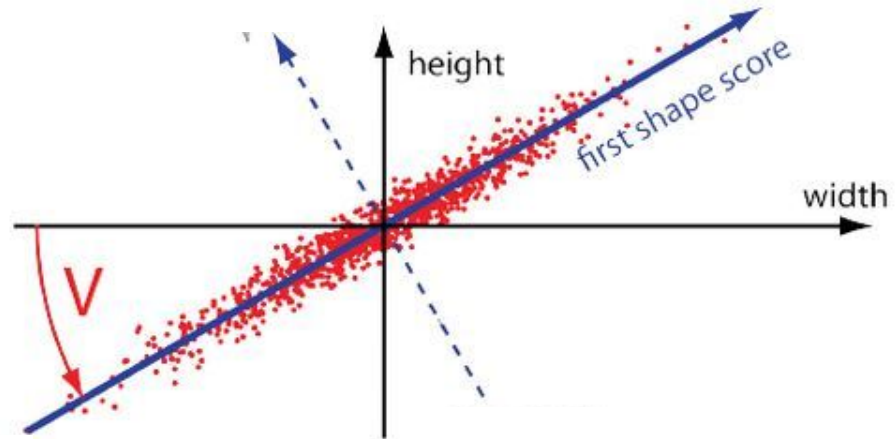
# Example: Axis Finding

- If there is a relation (correlation) between height and width, then a large part of the variance of our data can be described by a single number (1 dimension)
- PCA finds the "best representation" in this lower dimension (i.e. reduces dimensionality)

A



B



# Reminder: PCA

- To clarify what we mean by "best representation":
  - In our original feature space, what are we trying to minimize?

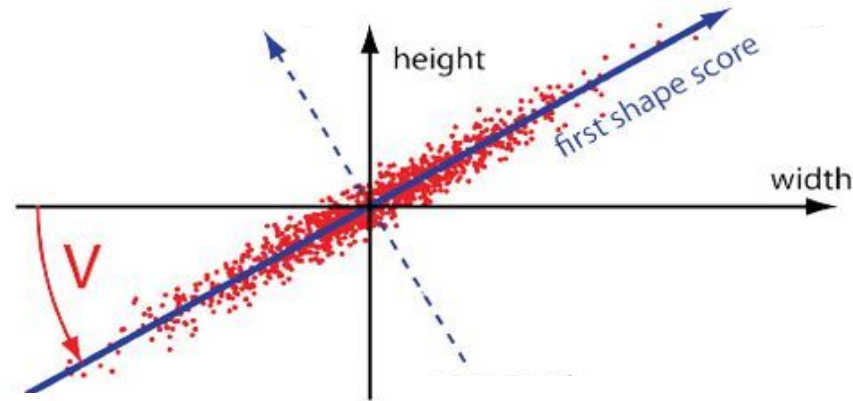
PCA objective:

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

minimize

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

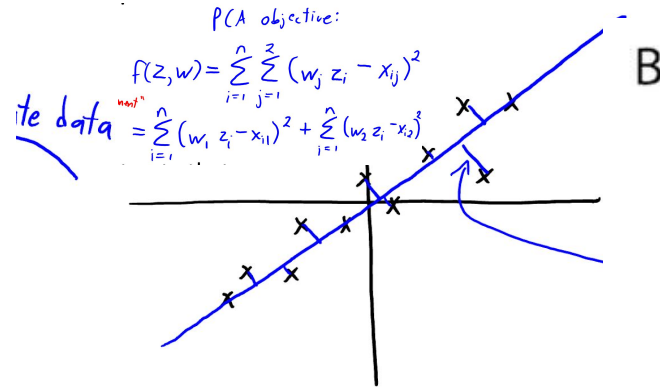
B



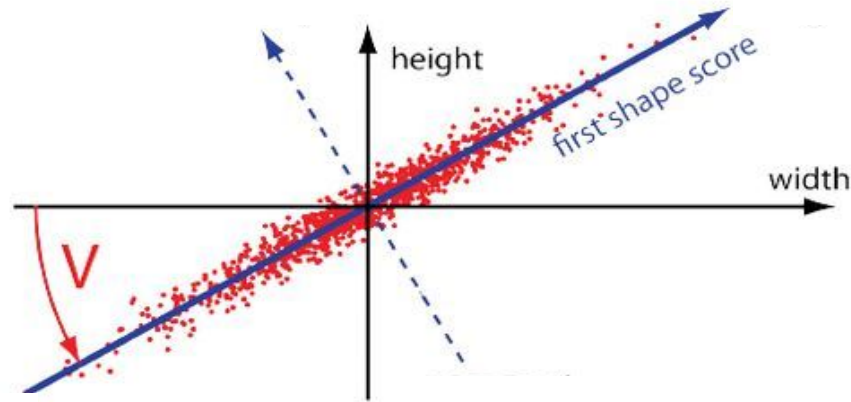
# Reminder: PCA

- In our original feature space, what are we trying to minimize?
- **Answer: PCA can be interpreted as minimizing the orthogonal distance between our axis and data in the original feature space.** So we define "best representation" as being that which

Principal component analysis



interpreted as (turned a 2d problem into a 1d problem) "orthogonal" PC



# Reminder: PCA

- In our original feature space, what are we trying to minimize?
- **Answer 2: Considering the objective function, PCA picks a first PC in the direction that maximizes the variance. This is the statistical interpretation of how PCA is describing the data.**

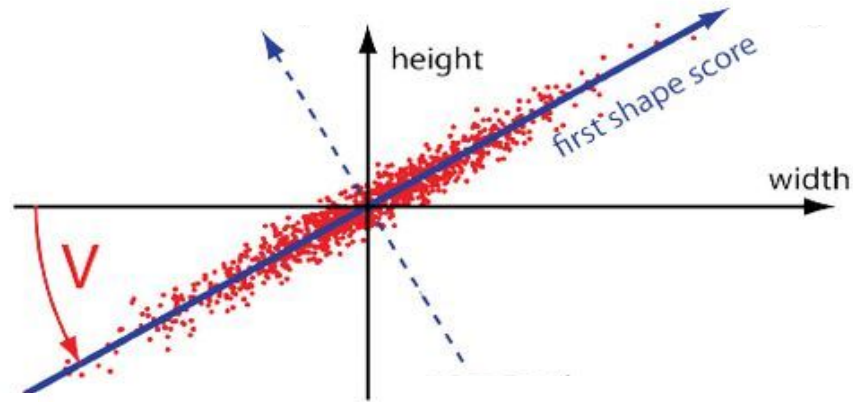
PCA objective:

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

minimize

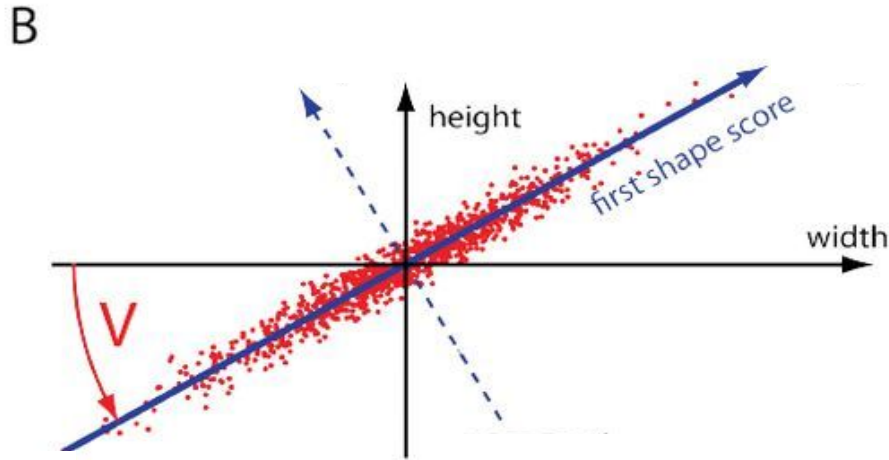
$$= \sum_{i=1}^n (w_1 z_i - x_{i1})^2 + \sum_{i=1}^n (w_2 z_i - x_{i2})^2$$

B



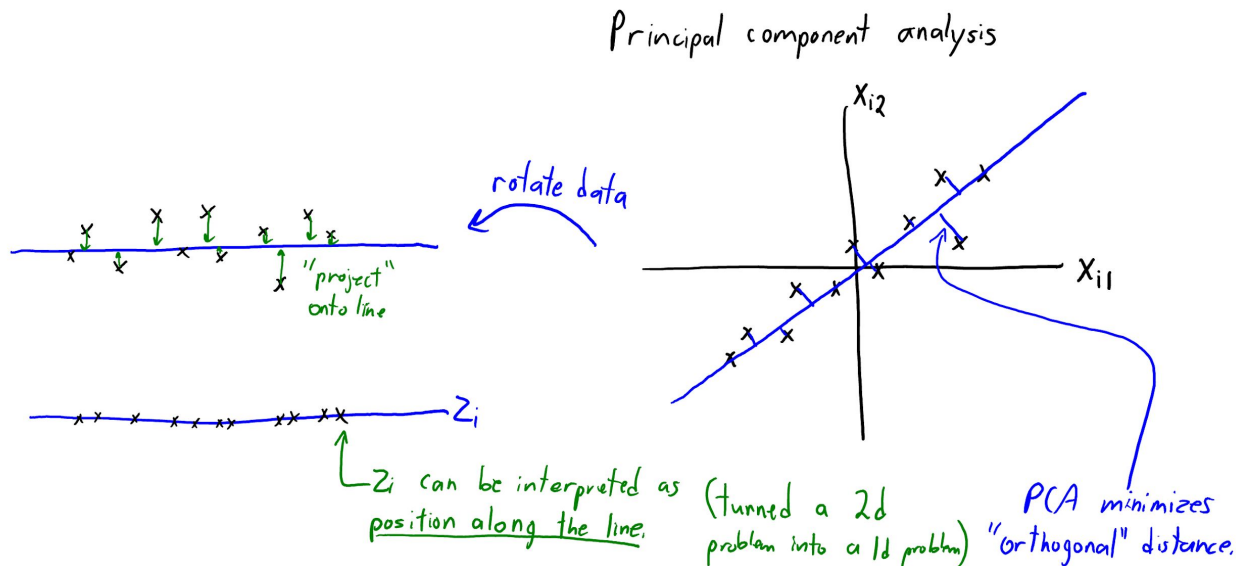
# Reminder: PCA

- In this simple example, how might you interpret the first principal component score ( $z_i$ ) of an example?



# Reminder: PCA

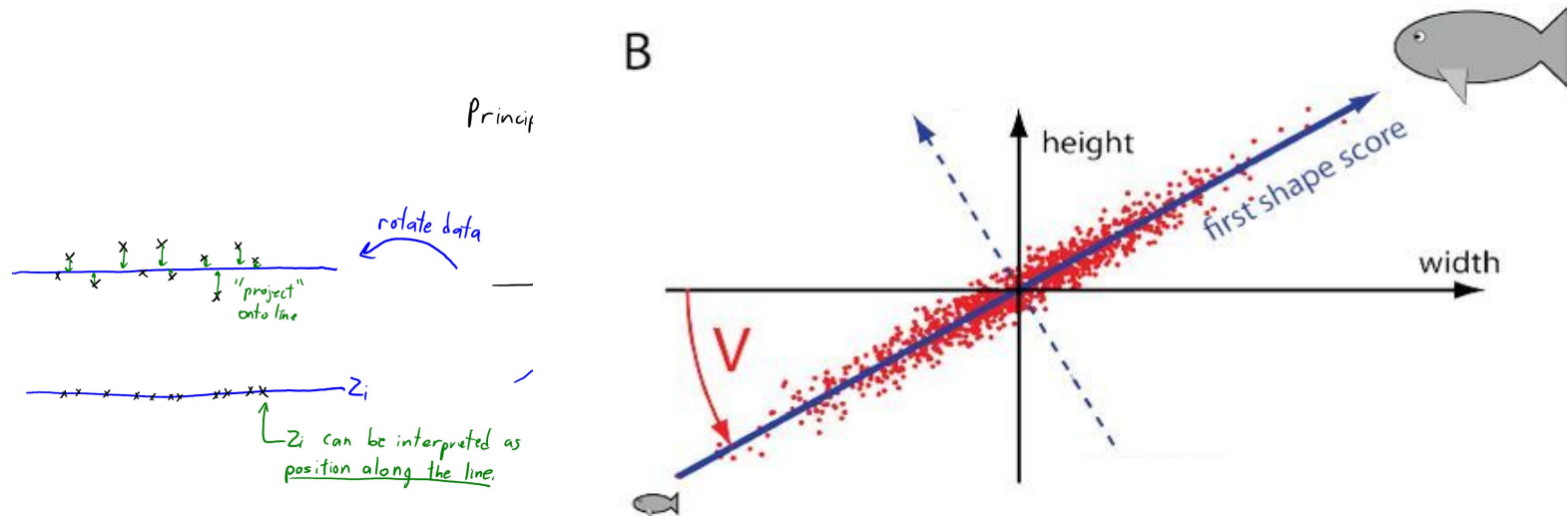
- In this simple example, how might you interpret the first principal component score ( $z_i$ ) of an example?
- Recall that our PC score  $z_i$  can be thought of as position along a line





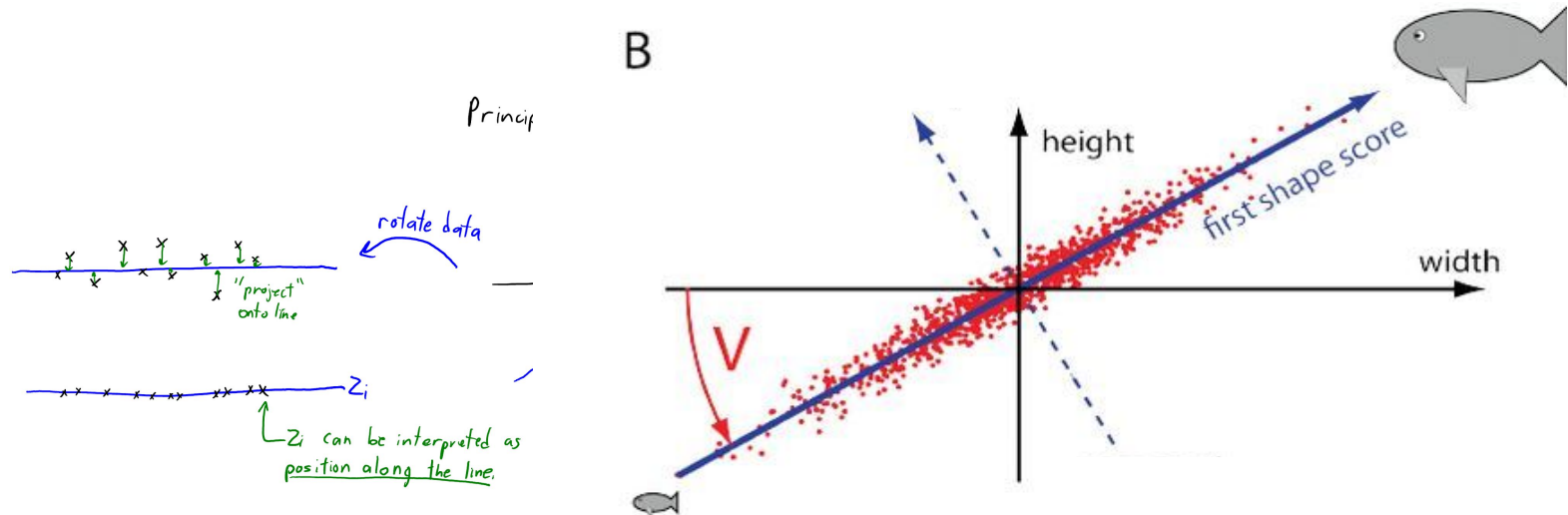
# Reminder: PCA

- The first PC here describes width and height both increasing linearly with the value of our first PC
  - We can interpret our first PC score as roughly describing "size"
- Fish with large first PC scores will have larger width and height than small first PC scores



# Reminder: PCA

- Thus we have gone from describing our data in terms of a width and a height to a single dimension roughly described as size
- Our first PC here reduces the dimensionality well, and so is likely to be useful



# Question

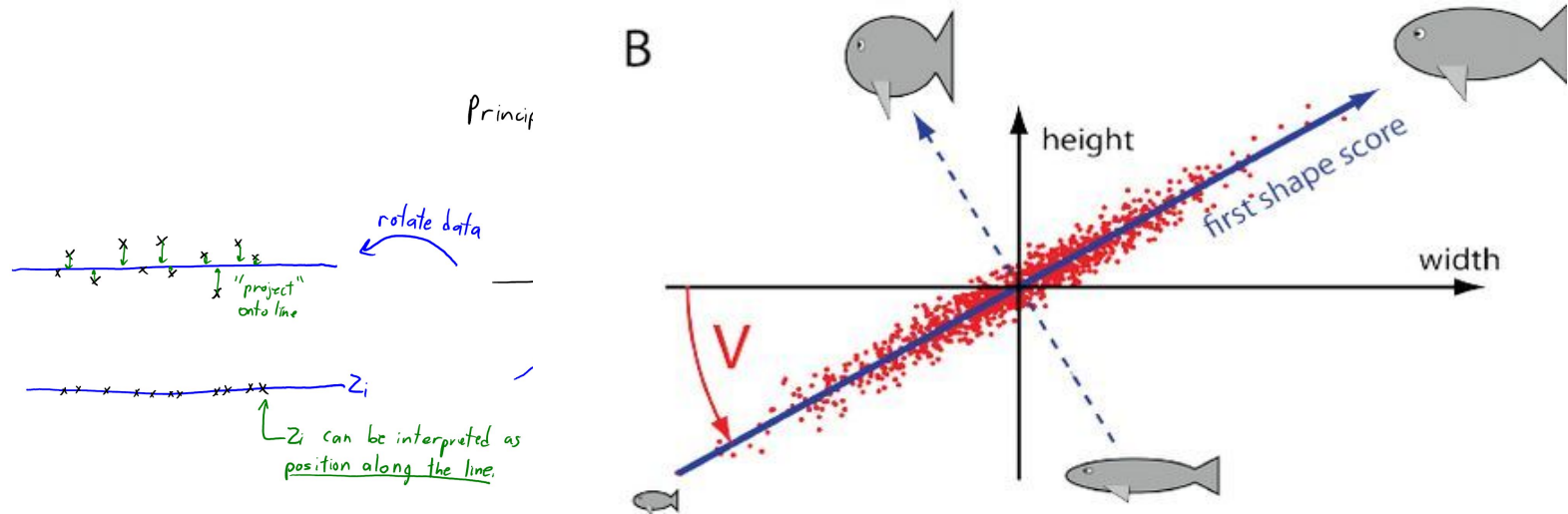
- Could you use PCA if  $k = 2$ ? What about  $k > 2$ ?

# Question

- Could you use PCA if  $k = d$ ? What about  $k > d$ ?
- Answer:
  - For  $k = 2$ , you can, but any 2D solution will be capable of fully describing our data, and will be a degenerate solution of the PCA objective
  - For  $k > 2$  the components of the solution set  $z_i$  will be linearly dependent, and will not provide any added depth

# Reminder: PCA

- Consider fitting a second principal component
- If we enforce that the second principal component must be orthogonal to the first, then it will describe the data's variation about the first principal component
- With two PCs we can completely describe the data, and so even though we have defined a unique SVD based solution, there are infinitely many other possible solutions which also minimize the PCA objective in 2D

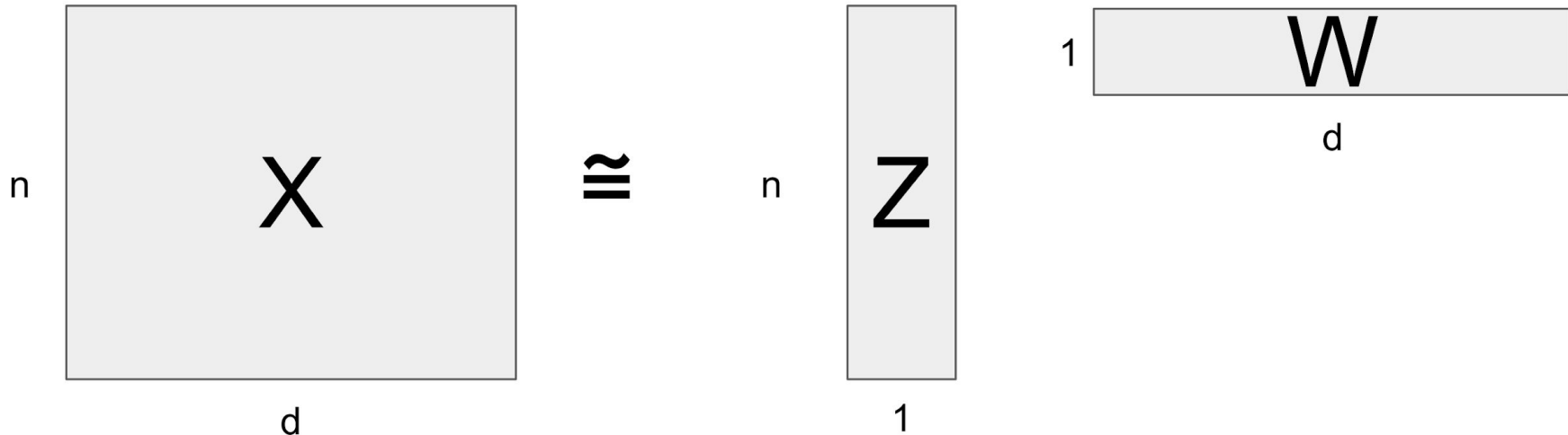


# PCA Calculation Questions



# K = 1 Case

- We'll consider conceptually the case where  $k = 1$
- How do we find  $W$  (aka the first Principal Component)?



# PCA Computation: SVD

- How do we fit with normalization/orthogonality/sequential-fitting?
  - It can be done with the “singular value decomposition” (SVD).
  - Take CPSC 302.
- 4 lines of Python code:
  - `mu = np.mean(X,axis=0)`
  - `X -= mu`
  - `U, s, Vh = np.linalg.svd(X)`
  - `W = Vh[:k]`
- Computing Z is cheaper now:

$$Z = X W^T (W W^T)^{-1} = X W^T$$

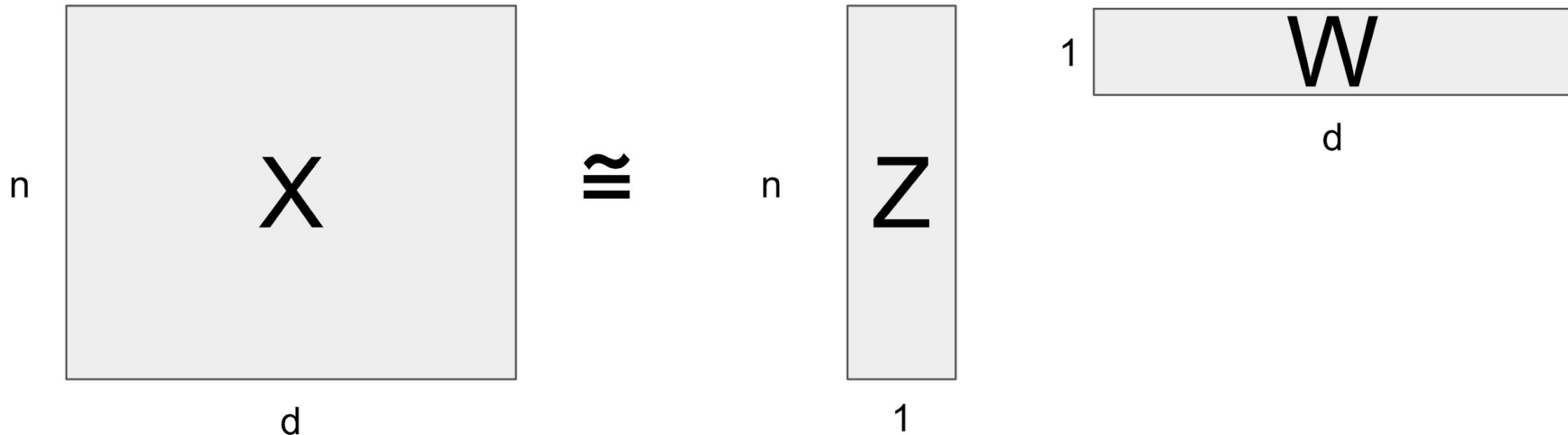
$$W W^T = \begin{bmatrix} -w_1- \\ -w_2- \\ \vdots \\ -w_k- \end{bmatrix} \begin{bmatrix} | & | & \dots & | \\ w_1^T & w_2^T & \dots & w_k^T \\ | & | & \dots & | \end{bmatrix}$$

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



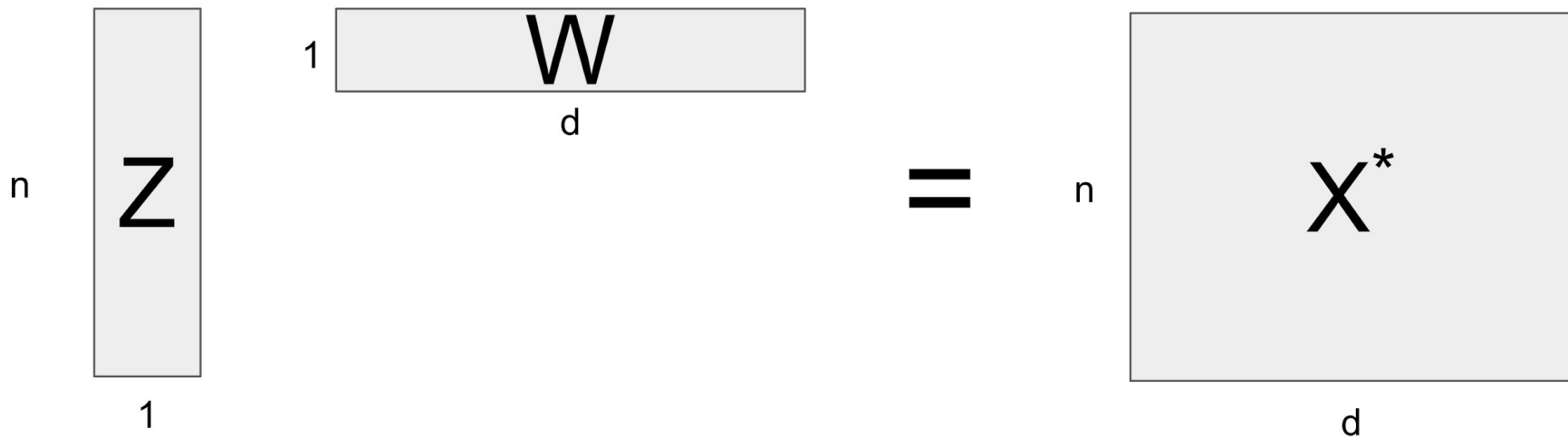
# K = 1 Case

- How do we find  $W$  (aka the first Principal Component)?
- Answer: Use a minimization algorithm :
  - Alternating Minimization
  - Stochastic Gradient Descent



# Reconstruction with $k = 1$

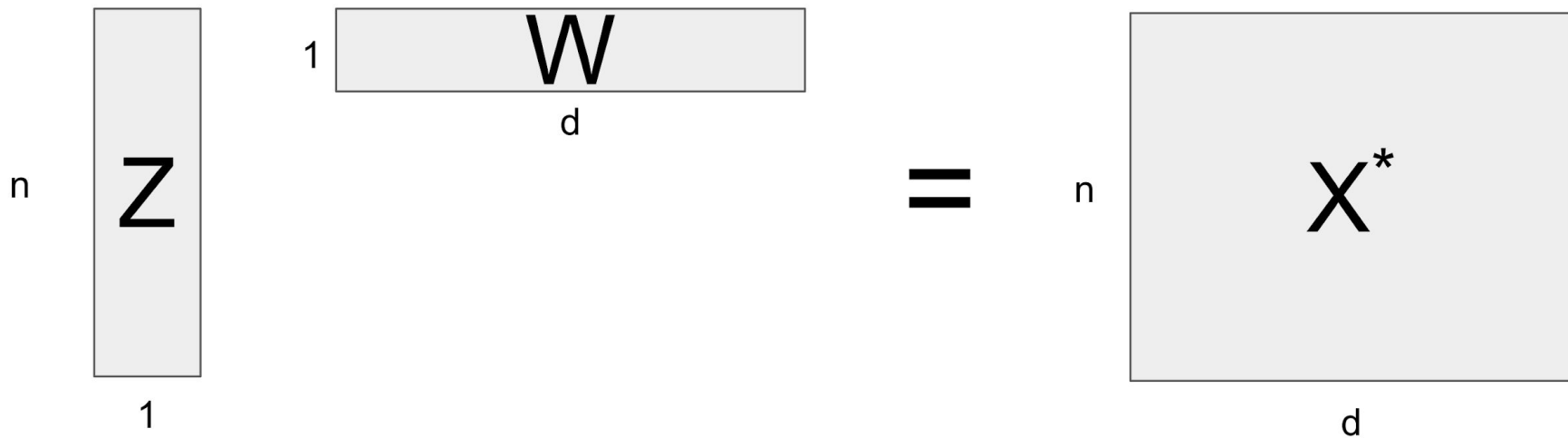
- Suppose we now take our principal components and multiply them out
- Is our output  $X^*$  the same as  $X$ ?



# Reconstruction with $k = 1$

- Suppose we now take our principal components and multiply them out
- Is our output  $X^*$  the same as  $X$ ?

Answer: If our data is completely describable with a single PC then yes, more generally no, since some information will be lost in the dimensionality reduction, our reconstructed  $X^*$  will be not quite the same as what we started with.



# Reconstruction with $k = 1$

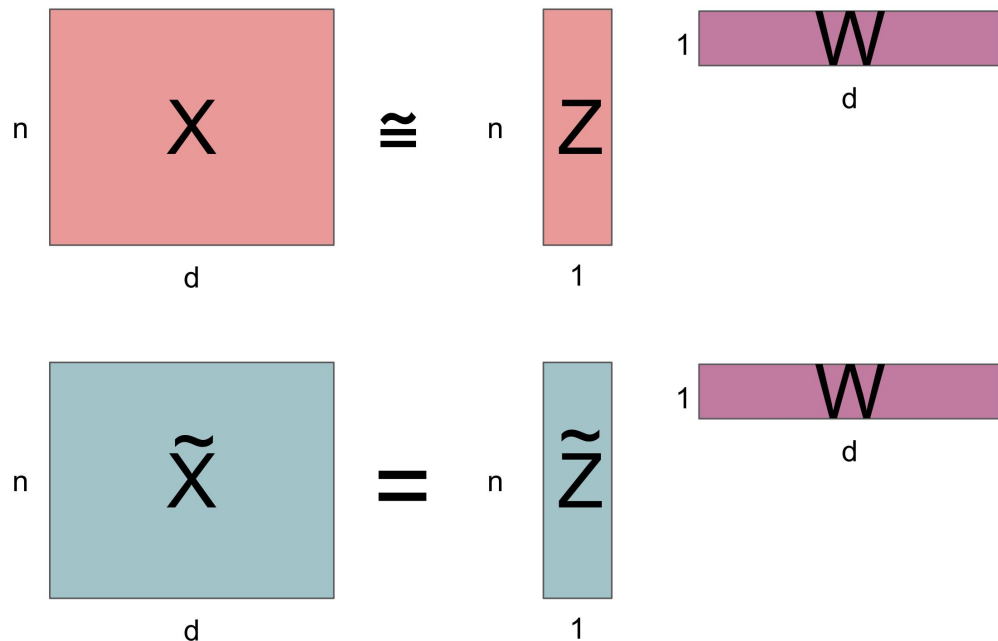
- Refresher: Suppose we train a linear classifier on the features we derived with PCA.
- i) Why might this fact ( $X \neq X^*$ ) be a good thing?
- ii) Why might it be a bad thing?

# Reconstruction with $k = 1$

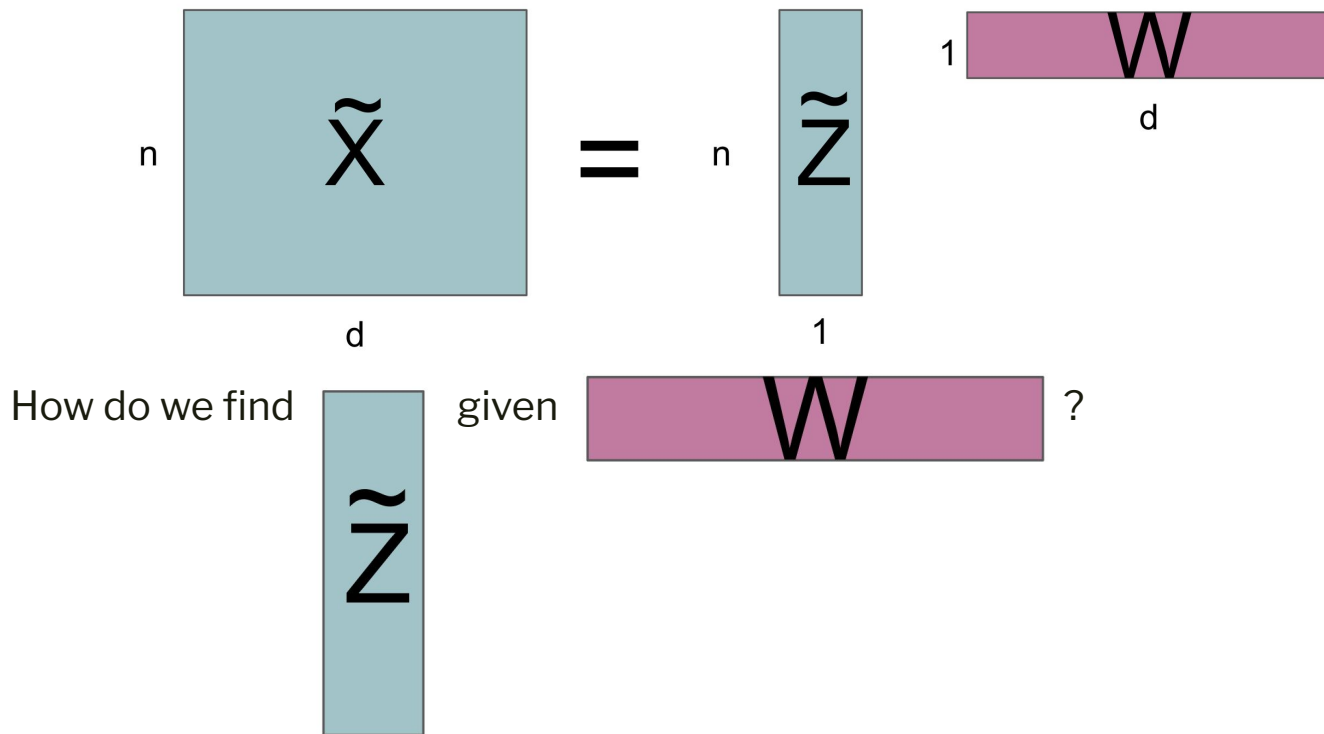
- Refresher: Suppose we train a linear classifier on the features we derived with PCA.
- i) Why might this fact ( $X \neq X^*$ ) be a good thing?
- ii) Why might it be a bad thing?
- Answer:
  - i) We've reduced the complexity of our description of our data while trying to retain the most relevant features. This might reduce overfitting by not giving the model overly specific data about training examples.
  - ii) We may have lost information relevant to our classification task (especially since we're only using  $k = 1$  here)

# Transforming Test Data in $k = 1$ Case

- Consider now our test set  $\tilde{X}$
- We would like to factor our test set into the same form as for our training set
- i.e. We would like to describe our test set in terms of the same  $z1$  (same  $W$ )
- This requires us to compute the class scores  $\tilde{Z}$



# Transforming Test Data in $k = 1$ Case



# Transforming Test Data in $k = 1$ Case

- To form  $Z$  given  $W$ :

$$X = ZW$$

$$XW^T = ZWW^T$$

$$XW^T = ZI_k$$

$$XW^T = Z$$

Rows of  $W$  are orthonormal

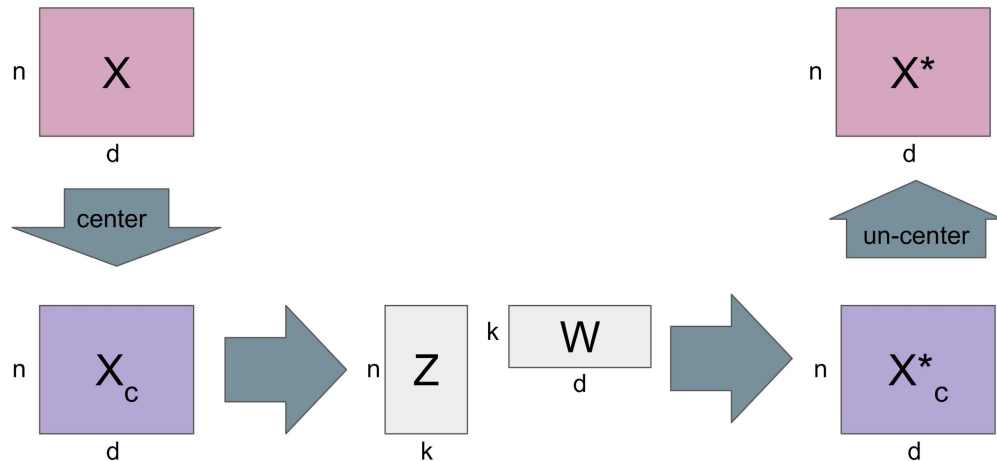




# Reminder: PCA with non-centered data

- The PCA objective assumes that the data have features with a mean of 0
- To use PCA on uncentered data, we first subtract the mean of the feature across the training set from each feature

→ Set  $\mu_j = \frac{1}{n} \sum_{i=1}^n x_{ij}$  (mean of column 'j')  
Replace each  $x_{ij}$  with  $(x_{ij} - \mu_j)$



# Questions

