

# Machine Learning for Data Science (CS4786)

## Lecture 12

Canonical Correlation Analysis & Kernel PCA

Course Webpage :

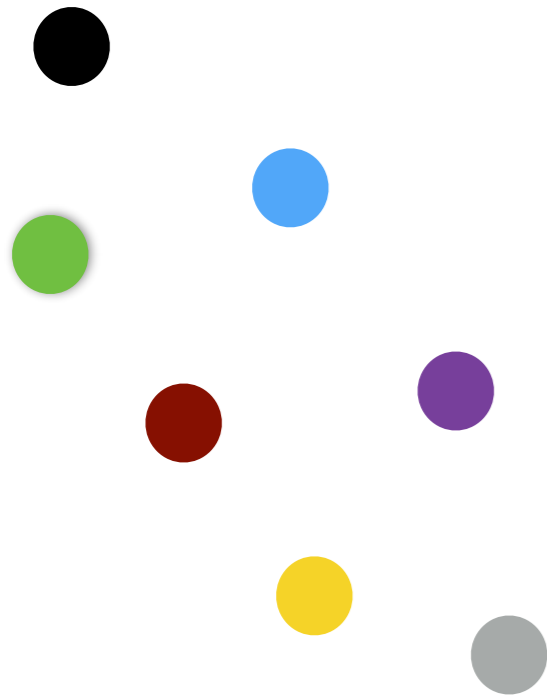
<http://www.cs.cornell.edu/Courses/cs4786/2017fa/>

# How do we get the right direction? (say $K = 1$ )

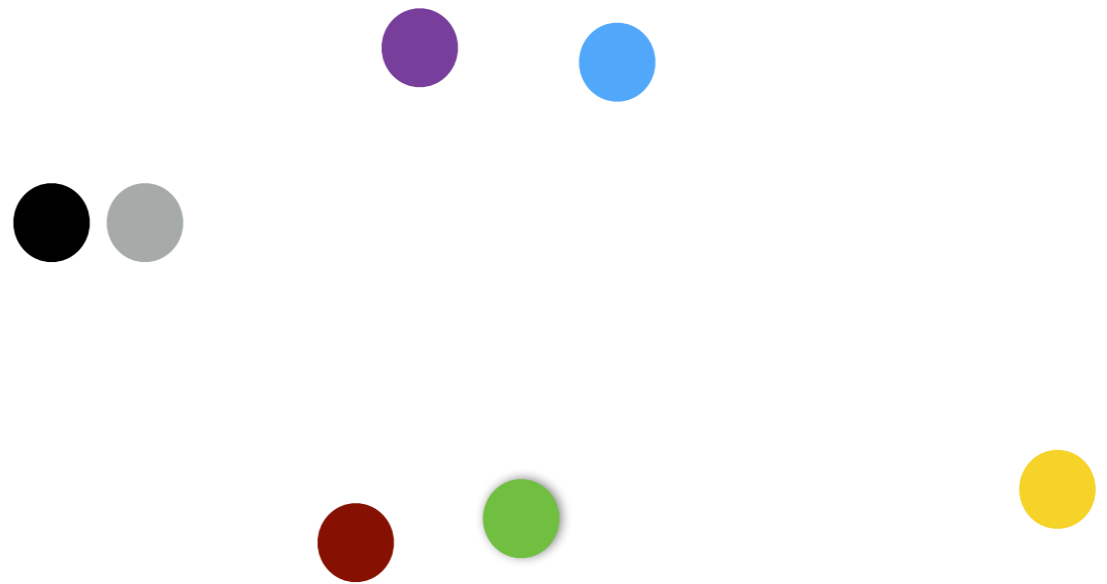


Age  
+ Gender  
Angle

# WHICH DIRECTION TO PICK?



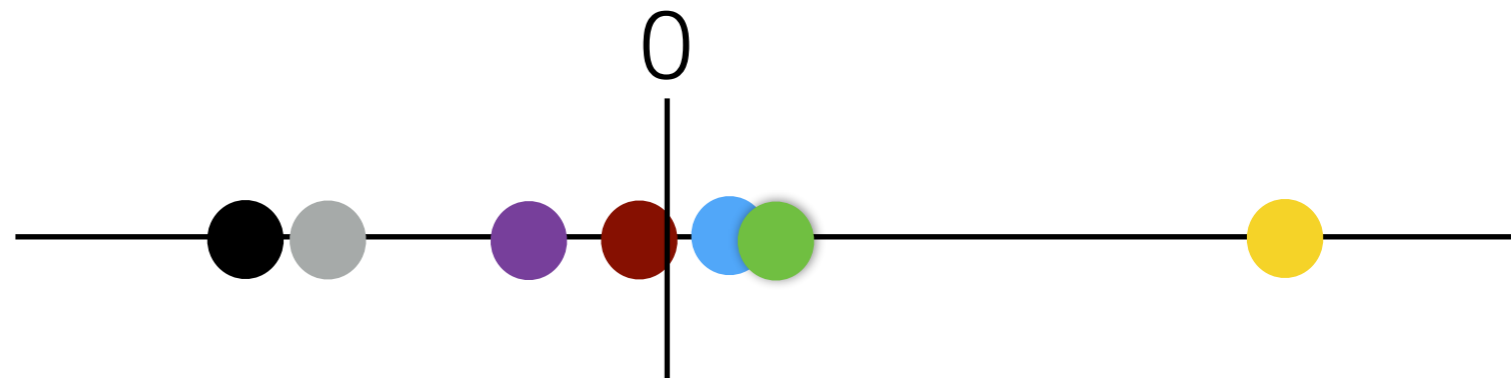
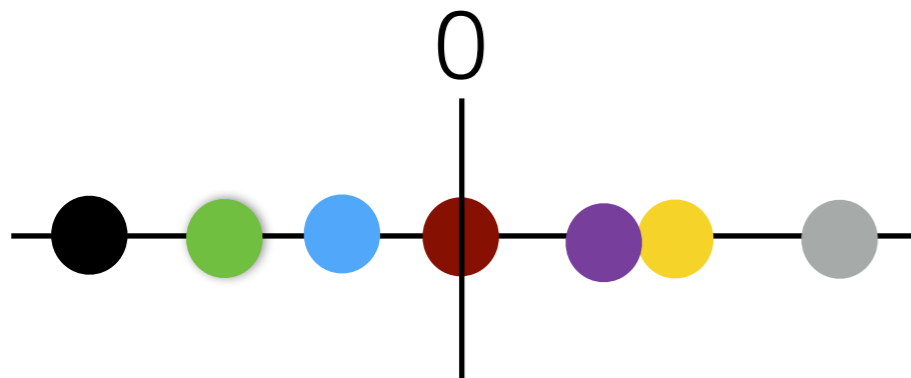
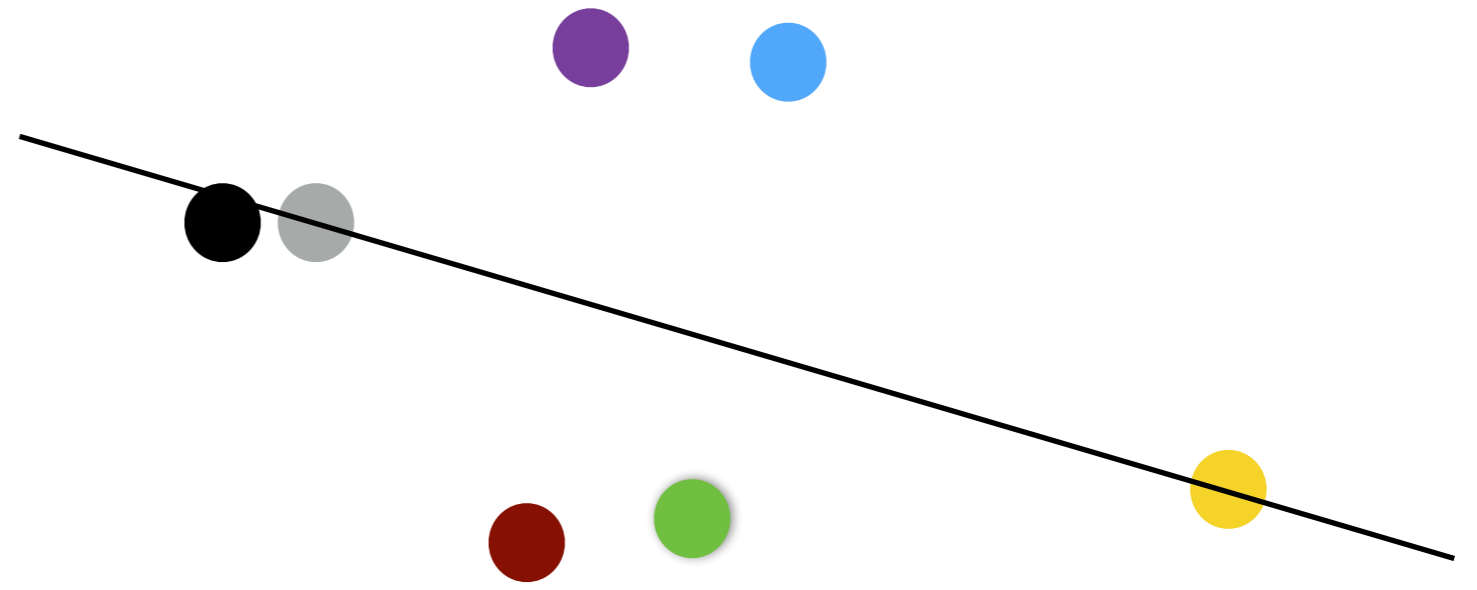
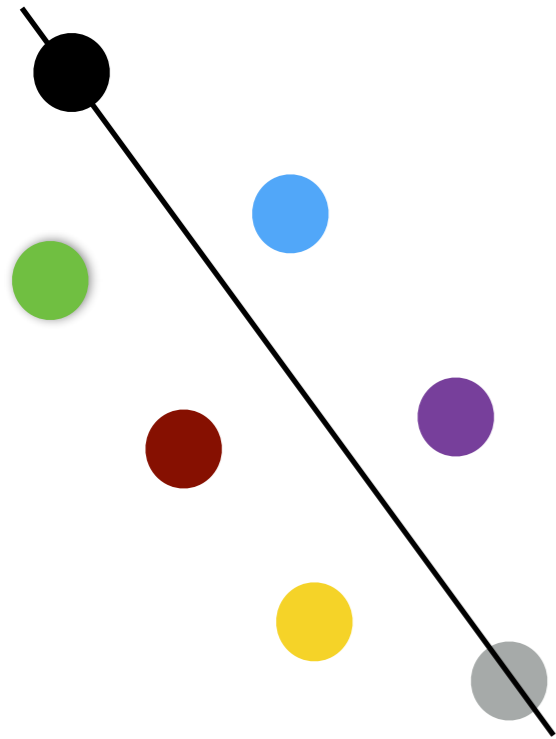
View I



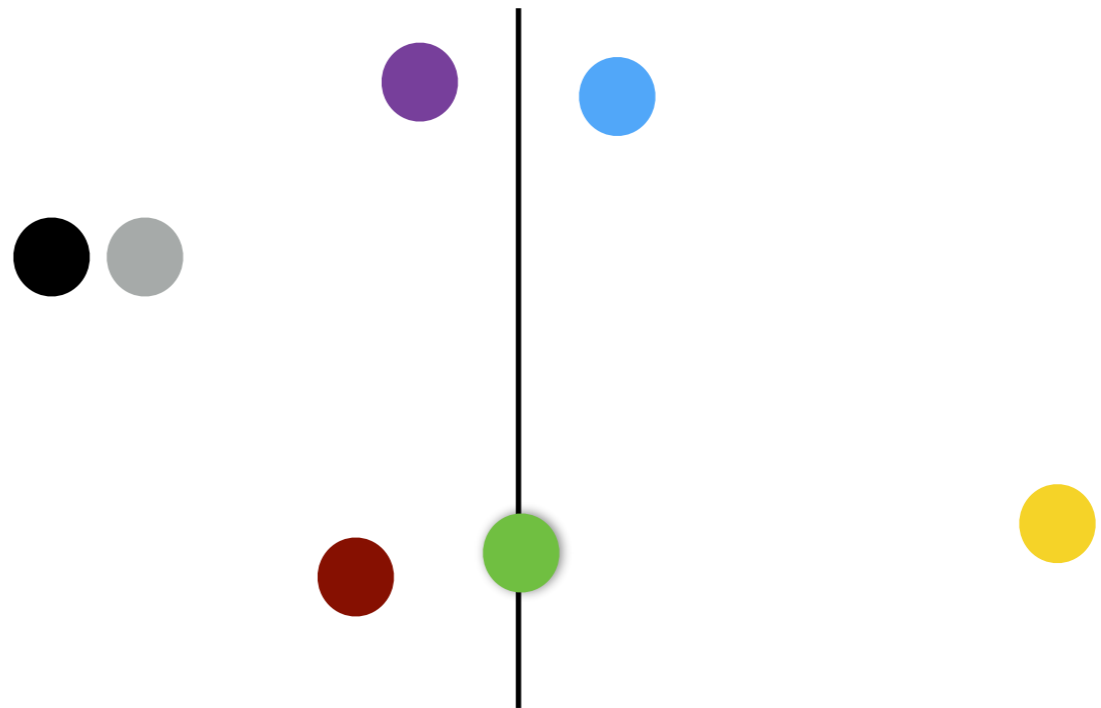
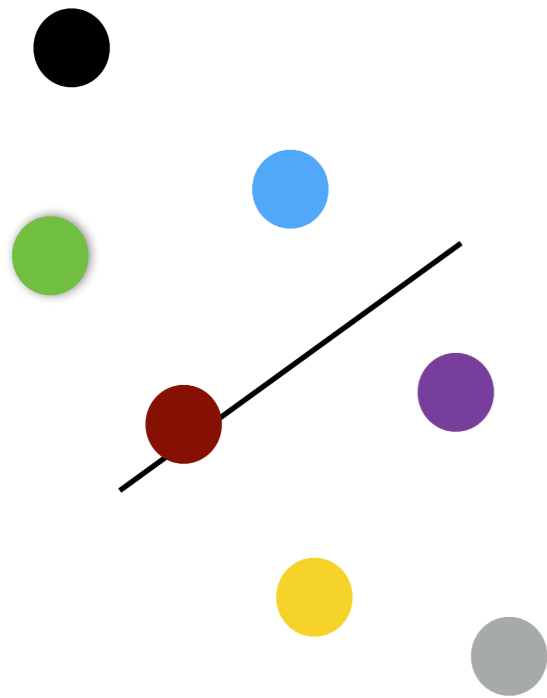
View II

# WHICH DIRECTION TO PICK?

PCA direction



# WHICH DIRECTION TO PICK?



Direction has large covariance

How do we pick the right direction to project to?

# MAXIMIZING CORRELATION COEFFICIENT

- Say  $\mathbf{w}_1$  and  $\mathbf{v}_1$  are the directions we choose to project in views 1 and 2 respectively we want these directions to maximize,

$$\frac{1}{n} \sum_{t=1}^n \left( \mathbf{y}_t[1] - \frac{1}{n} \sum_{t=1}^n \mathbf{y}_t[1] \right) \cdot \left( \mathbf{y}'_t[1] - \frac{1}{n} \sum_{t=1}^n \mathbf{y}'_t[1] \right)$$

where  $\mathbf{y}_t[1] = \mathbf{w}_1^\top \mathbf{x}_t$  and  $\mathbf{y}'_t[1] = \mathbf{v}_1^\top \mathbf{x}'_t$

What is the problem  
with the above?



# WHY NOT MAXIMIZE COVARIANCE

$$\text{Say } \frac{1}{n} \sum_{t=1}^n \mathbf{x}_t[2] \cdot \mathbf{x}'_t[2] > 0$$

Scaling up this coordinate we can blow up covariance

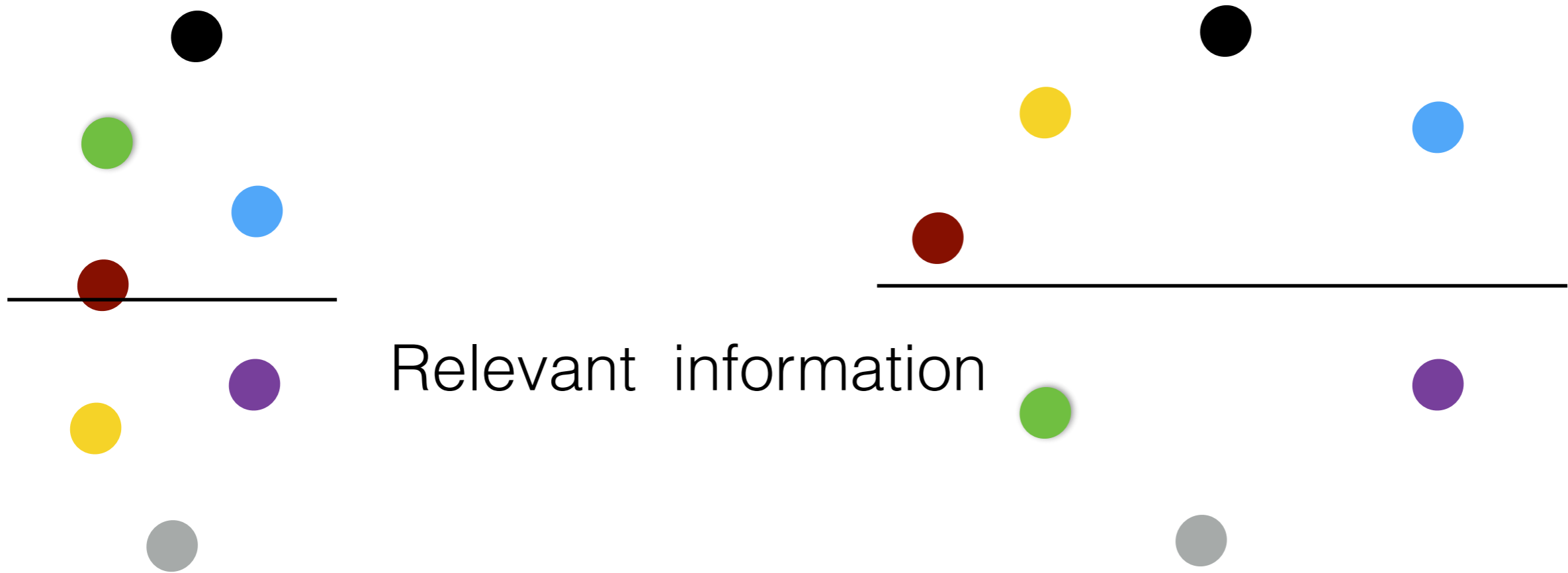
# WHY NOT MAXIMIZE COVARIANCE



$$\text{Say } \frac{1}{n} \sum_{t=1}^n \mathbf{x}_t[2] \cdot \mathbf{x}'_t[2] > 0$$

Scaling up this coordinate we can blow up covariance

# WHY NOT MAXIMIZE COVARIANCE



$$\text{Say } \frac{1}{n} \sum_{t=1}^n \mathbf{x}_t[2] \cdot \mathbf{x}'_t[2] > 0$$

Scaling up this coordinate we can blow up covariance

# MAXIMIZING CORRELATION COEFFICIENT

- Say  $\mathbf{w}_1$  and  $\mathbf{v}_1$  are the directions we choose to project in views 1 and 2 respectively we want these directions to maximize,

$$\frac{\frac{1}{n} \sum_{t=1}^n (\mathbf{y}_t[1] - \frac{1}{n} \sum_{t=1}^n \mathbf{y}_t[1]) \cdot (\mathbf{y}'_t[1] - \frac{1}{n} \sum_{t=1}^n \mathbf{y}'_t[1])}{\sqrt{\frac{1}{n} \sum_{t=1}^n (\mathbf{y}_t[1] - \frac{1}{n} \sum_{t=1}^n \mathbf{y}_t[1])^2} \sqrt{\frac{1}{n} \sum_{t=1}^n (\mathbf{y}'_t[1] - \frac{1}{n} \sum_{t=1}^n \mathbf{y}'_t[1])^2}}$$

# BASIC IDEA OF CCA

- Normalize variance in chosen direction to be constant (say 1)
- Then maximize covariance
- This is same as maximizing “correlation coefficient”

# COVARIANCE VS CORRELATION

- $\text{Covariance}(A, B) = \mathbb{E}[(A - \mathbb{E}[A]) \cdot (B - \mathbb{E}[B])]$

Depends on the scale of  $A$  and  $B$ . If  $B$  is rescaled, covariance shifts.

- $\text{Correlation}(A, B) = \frac{\mathbb{E}[(A - \mathbb{E}[A]) \cdot (B - \mathbb{E}[B])]}{\sqrt{\text{Var}(A)}\sqrt{\text{Var}(B)}}$

Scale free.

# MAXIMIZING CORRELATION COEFFICIENT

- Say  $\mathbf{w}_1$  and  $\mathbf{v}_1$  are the directions we choose to project in views 1 and 2 respectively we want these directions to maximize,

$$\frac{1}{n} \sum_{t=1}^n \left( \mathbf{y}_t[1] - \frac{1}{n} \sum_{t=1}^n \mathbf{y}_t[1] \right) \cdot \left( \mathbf{y}'_t[1] - \frac{1}{n} \sum_{t=1}^n \mathbf{y}'_t[1] \right)$$

where  $\mathbf{y}_t[1] = \mathbf{w}_1^\top \mathbf{x}_t$  and  $\mathbf{y}'_t[1] = \mathbf{v}_1^\top \mathbf{x}'_t$

# MAXIMIZING CORRELATION COEFFICIENT

- Say  $\mathbf{w}_1$  and  $\mathbf{v}_1$  are the directions we choose to project in views 1 and 2 respectively we want these directions to maximize,

$$\frac{1}{n} \sum_{t=1}^n \left( \mathbf{y}_t[1] - \frac{1}{n} \sum_{t=1}^n \mathbf{y}_t[1] \right) \cdot \left( \mathbf{y}'_t[1] - \frac{1}{n} \sum_{t=1}^n \mathbf{y}'_t[1] \right)$$

$$\text{s.t. } \frac{1}{n} \sum_{t=1}^n \left( \mathbf{y}_t[1] - \frac{1}{n} \sum_{t=1}^n \mathbf{y}_t[1] \right)^2 = \frac{1}{n} \sum_{t=1}^n \left( \mathbf{y}'_t[1] - \frac{1}{n} \sum_{t=1}^n \mathbf{y}'_t[1] \right)^2 = 1$$

where  $\mathbf{y}_t[1] = \mathbf{w}_1^\top \mathbf{x}_t$  and  $\mathbf{y}'_t[1] = \mathbf{v}_1^\top \mathbf{x}'_t$



# CANONICAL CORRELATION ANALYSIS

- Hence we want to solve for projection vectors  $\mathbf{w}_1$  and  $\mathbf{v}_1$  that

$$\text{maximize } \frac{1}{n} \sum_{t=1}^n \mathbf{w}_1^\top (\mathbf{x}_t - \boldsymbol{\mu}) \cdot \mathbf{v}_1^\top (\mathbf{x}'_t - \boldsymbol{\mu}')$$

$$\text{subject to } \frac{1}{n} \sum_{t=1}^n (\mathbf{w}_1^\top (\mathbf{x}_t - \boldsymbol{\mu}))^2 = \frac{1}{n} \sum_{t=1}^n (\mathbf{v}_1^\top (\mathbf{x}'_t - \boldsymbol{\mu}'))^2 = 1$$

$$\text{where } \boldsymbol{\mu} = \frac{1}{n} \sum_{t=1}^n \mathbf{x}_t \text{ and } \boldsymbol{\mu}' = \frac{1}{n} \sum_{t=1}^n \mathbf{x}'_t$$

# CANONICAL CORRELATION ANALYSIS

- Hence we want to solve for projection vectors  $\mathbf{w}_1$  and  $\mathbf{v}_1$  that

$$\text{maximize } \mathbf{w}_1^\top \Sigma_{1,2} \mathbf{v}_1$$

$$\text{subject to } \mathbf{w}_1^\top \Sigma_{1,1} \mathbf{w}_1 = \mathbf{v}_1^\top \Sigma_{2,2} \mathbf{v}_1 = 1$$

# CANONICAL CORRELATION ANALYSIS

- Hence we want to solve for projection vectors  $\mathbf{w}_1$  and  $\mathbf{v}_1$  that

$$\text{maximize } \mathbf{w}_1^\top \Sigma_{1,2} \mathbf{v}_1$$

$$\text{subject to } \mathbf{w}_1^\top \Sigma_{1,1} \mathbf{w}_1 = \mathbf{v}_1^\top \Sigma_{2,2} \mathbf{v}_1 = 1$$

$$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} = \text{COV} \left( \begin{matrix} X & X' \end{matrix} \right)$$

# SOLUTION

# SOLUTION

$$W_1 = \text{eigs}\left(\Sigma_{11}^{-1} \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21}, K\right)$$

$$W_2 = \text{eigs}\left(\Sigma_{22}^{-1} \Sigma_{21} \Sigma_{11}^{-1} \Sigma_{12}, K\right)$$

# CCA ALGORITHM

# CCA ALGORITHM

$$1. \quad X = \begin{pmatrix} n & \begin{matrix} X_1 \\ d_1 \end{matrix}, & \begin{matrix} X_2 \\ d_2 \end{matrix} \end{pmatrix}$$

# CCA ALGORITHM

$$1. \quad X = \begin{pmatrix} n & \begin{matrix} X_1 \\ d_1 \end{matrix}, & \begin{matrix} X_2 \\ d_2 \end{matrix} \end{pmatrix}$$

$$2. \quad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} = \text{COV} \left( \begin{matrix} X \end{matrix} \right)$$



# CCA ALGORITHM

$$1. \quad X = \begin{pmatrix} n & \begin{matrix} X_1 \\ d_1 \end{matrix}, & \begin{matrix} X_2 \\ d_2 \end{matrix} \end{pmatrix}$$

$$2. \quad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} = \text{COV} \left( \begin{matrix} X \end{matrix} \right)$$

$$3. \quad W_1 = \text{eigs} \left( \Sigma_{11}^{-1} \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21}, K \right)$$

# CCA ALGORITHM

$$1. \quad X = \begin{pmatrix} n & \begin{matrix} X_1 \\ d_1 \end{matrix}, & \begin{matrix} X_2 \\ d_2 \end{matrix} \end{pmatrix}$$

$$2. \quad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} = \text{COV} \left( \begin{matrix} X \end{matrix} \right)$$

$$3. \quad W_1 = \text{eigs} \left( \begin{matrix} \Sigma_{11}^{-1} \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21} \end{matrix}, K \right)$$

$$4. \quad Y_1 = \begin{matrix} X_1 - \mu_1 \\ \end{matrix} \times W_1$$

CCA DEMO



i can't believe how awful is this movie i was expecting it to be really good especially with the actors that were in the cast this is depressing i'm so bummed that they ruined such a good plot



bummed to see such a bad game what an awful performance by everyone on the team as if everyone played to loose need to improve hitters more but fielders were also worse today one of the worst performance in the history of baseball



oh man this war movie was just too depressing for me some scenes were simply awful even though the plot closely follows the novel which i've read i was bummed at the end and had to secretly go cry



i will tell you what is wrong with it it is dead that's what is wrong with about enough of this that team is definitely deceased tired and shag after a long game you say look matey not a single soul in that lineup a single ball even if i put 4000-volts through them they are bleeding they are not pitching they passed on period plus pretty sure they m awful elderberries after that game you should be depressed like me



this was so hilarious that's the best movie i've seen in a while i didn't know this actor before but he is so funny i was laughing from start to finish



it was hilarious to see playing these kids against experts throughout the game they were just running here and there and trying to get to the ball which they couldn't even once this was funny for viewers but organizers should ensure that inexperienced teams don't play against the experienced ones to keep the game interesting



dude that movie was so funny right i was laughing in like fits during some of the scenes i know the plot is supposed to be thought-provoking but i found it hilarious i really should stop laughing all the time but who cares right



now what seems to be the problem he says after leaning on the coach's limb body after a fast pitch struck him during the game his face was icy serious not laughing at all unlike everyone else jen said 'it is the coach he is not moving at all is he dead he said slowly course not we answered laughing again thank god



well that was a funny movie i enjoyed the plot with all those twists you never knew what was going to happen especially in this last scene i wasn't expecting this outcome at all haha



was it a game at all i felt as if everyone was just trying to stay warm by making as little move as possible laziness of fielders was making it appear as if they were running in 0.5x speed mode haha strikers made good use of pitch they got and it was an easy win



lol i can't even sit properly now i have a tummy ache because of all the rofling that actor's head looked like a volcano haha i swear it looked like it was about to erupt and his brains would spill out haha



fans at the game are encouraged to get out of their seats stretch a bit and sing take me out to the ball game that is the closest baseball gets to a halftime haha



really love that movie we saw yesterday i was really excited since i knew it was going to be released this week and i haven't been disappointed at all i especially enjoyed the acting of the actors they were so good



what an awesome game it was dwight evans set the path to unprecedented victory when he made his very first strike on the pitch he alone made the whole game enjoyable excited for the next match



omg i totally loved yesterday's movie we were all so excited to finally catch the third movie after months of scouring the fan pages for the plot there are mixed opinions on the acting but i think the actors did a brilliant job overall



80 years old and was still playing the game stuff like this keeps you excited motivated you know yes he did break his back walking to the pitch to take the strike but you know everyone has to expire and go to their maker at some point he was lucky to do it while doing something he loved i am sure he enjoyed every second of it we should learn to enjoy this game too like him and reflect that on our strikes

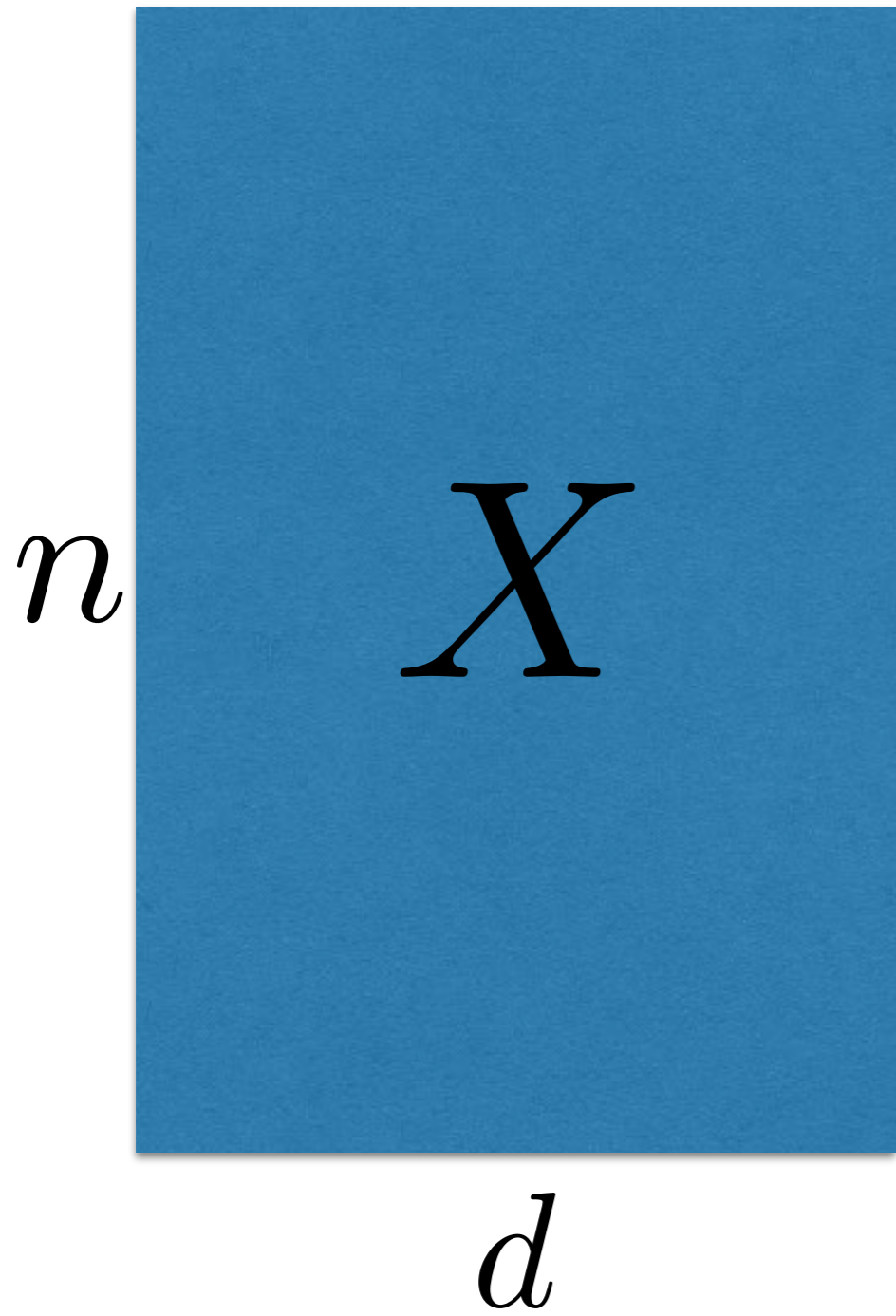
# Kernel PCA

(non-linear projections)

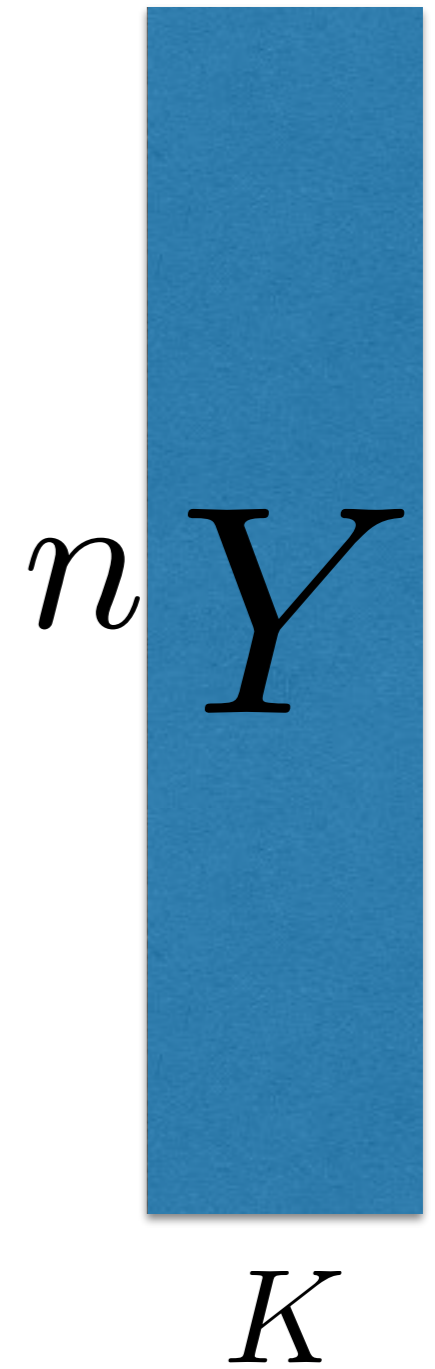
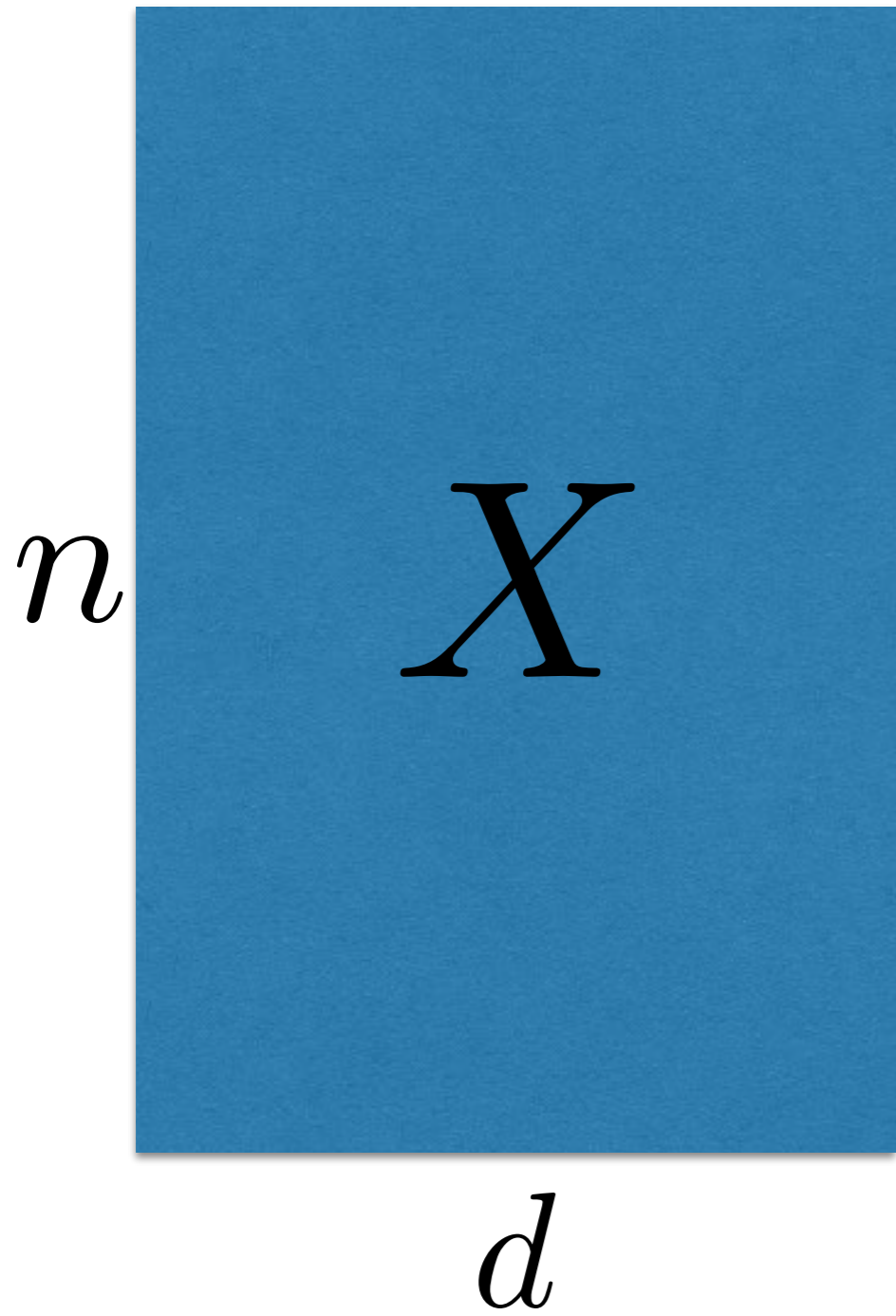


# LINEAR PROJECTIONS

# LINEAR PROJECTIONS



# LINEAR PROJECTIONS



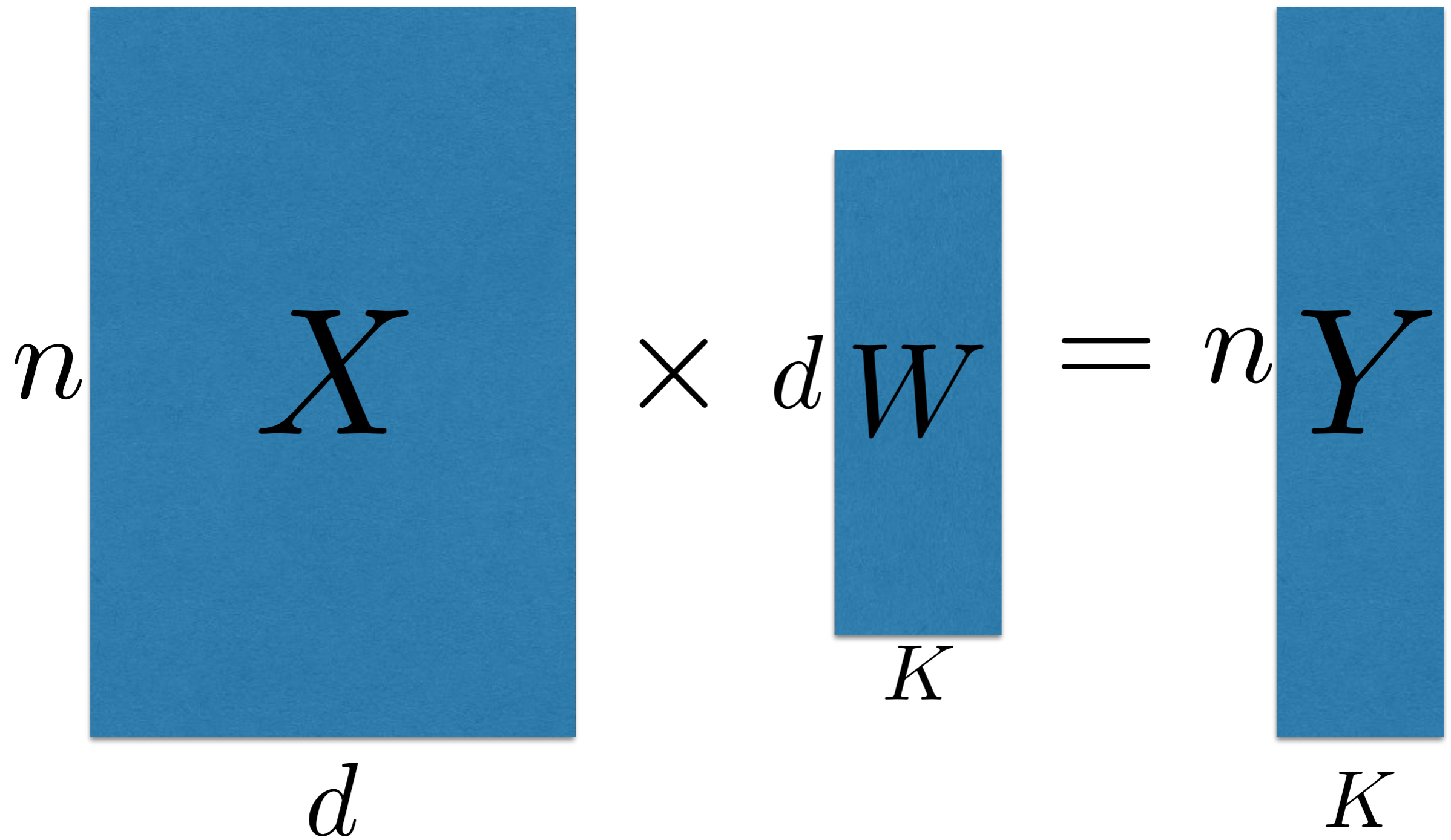
# LINEAR PROJECTIONS

The diagram illustrates the multiplication of two matrices to produce a third matrix. On the left is a large blue rectangle representing matrix  $X$ , with the dimension  $n$  labeled to its left and  $d$  labeled below it. To the right of  $X$  is a multiplication symbol  $\times$ . Next is a smaller blue rectangle representing matrix  $W$ , with the dimension  $d$  labeled to its left and  $K$  labeled below it. To the right of  $W$  is an equals sign  $=$ . Finally, on the right, is a blue rectangle representing matrix  $Y$ , with the dimension  $n$  labeled to its left and  $K$  labeled below it.

$$\begin{matrix} n \\ \times \\ X \\ d \end{matrix} \times \begin{matrix} d \\ W \\ K \end{matrix} = \begin{matrix} n \\ Y \\ K \end{matrix}$$



# LINEAR PROJECTIONS



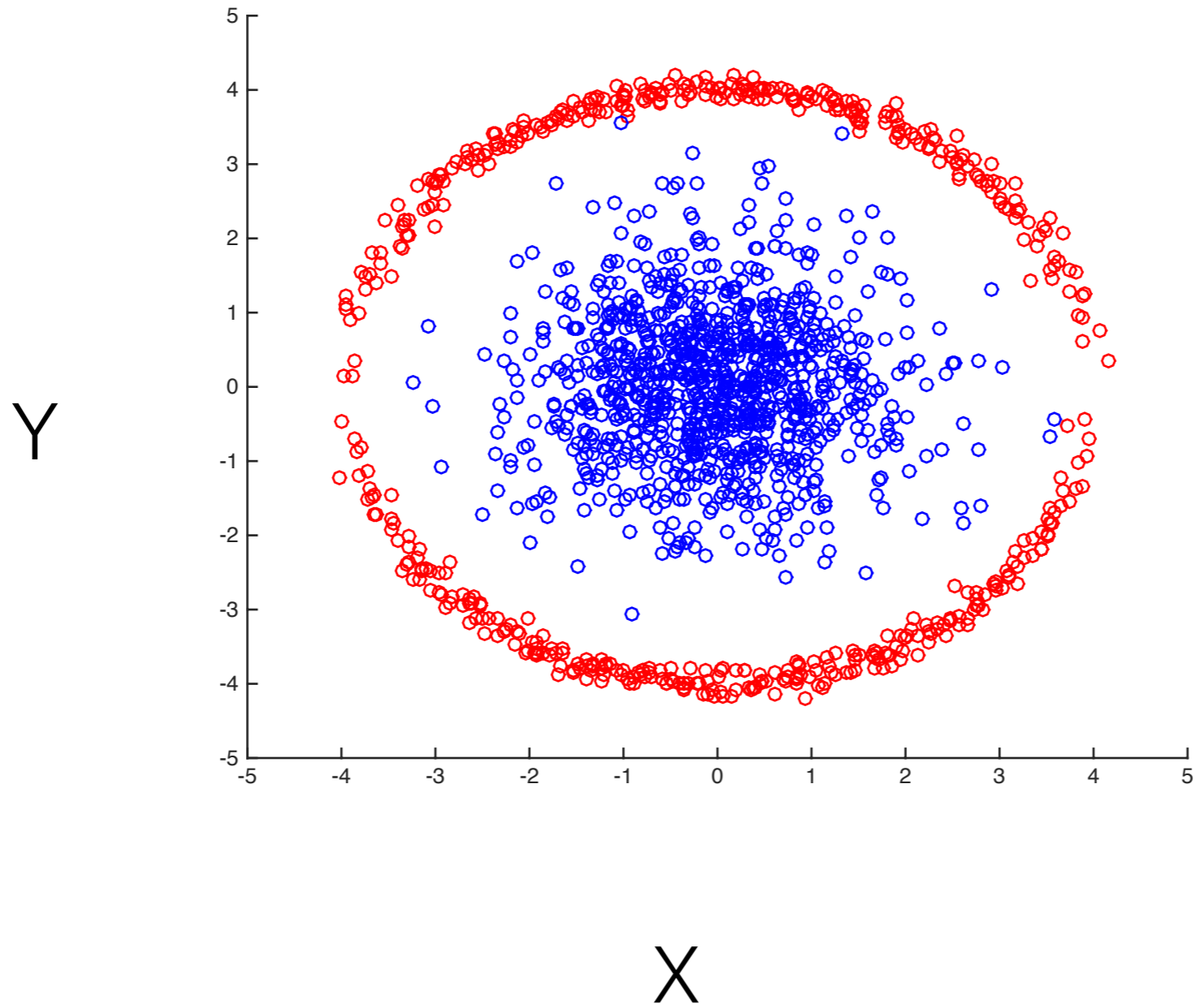
The diagram illustrates the concept of linear projection using matrix multiplication. It shows three matrices represented by blue rectangles:

- A large square matrix  $X$  with dimensions  $n$  (height) and  $d$  (width).
- A smaller vertical rectangular matrix  $W$  with dimensions  $d$  (height) and  $K$  (width).
- A vertical rectangular matrix  $Y$  with dimensions  $n$  (height) and  $K$  (width).

The relationship between these matrices is shown as  $X \times W = Y$ . The multiplication symbol  $\times$  is placed between  $X$  and  $W$ , and the equals sign  $=$  is placed between  $W$  and  $Y$ . The dimensions  $n$ ,  $d$ , and  $K$  are labeled next to their respective matrices.

Works when data lies in a low dimensional linear sub-space

# EXAMPLE



# LINEAR PROJECTIONS (RIGHT CO-ORDINATES)

Demo

# A FIRST CUT

- Given  $\mathbf{x}_t \in \mathbb{R}^d$ , the feature space vector is given by mapping

$$\Phi(\mathbf{x}_t) = (\mathbf{x}_t[1], \dots, \mathbf{x}_t[d], \mathbf{x}_t[1] \cdot \mathbf{x}_t[1], \mathbf{x}_t[1] \cdot \mathbf{x}_t[2], \dots, \mathbf{x}_t[d] \cdot \mathbf{x}_t[d], \dots)^\top$$



# A FIRST CUT

- Given  $\mathbf{x}_t \in \mathbb{R}^d$ , the feature space vector is given by mapping

$$\Phi(\mathbf{x}_t) = (\mathbf{x}_t[1], \dots, \mathbf{x}_t[d], \mathbf{x}_t[1] \cdot \mathbf{x}_t[1], \mathbf{x}_t[1] \cdot \mathbf{x}_t[2], \dots, \mathbf{x}_t[d] \cdot \mathbf{x}_t[d], \dots)^\top$$

- Enumerating products up to order  $K$  (ie. products of at most  $K$  coordinates) we can get degree  $K$  polynomials.

# A FIRST CUT

- Given  $\mathbf{x}_t \in \mathbb{R}^d$ , the feature space vector is given by mapping

$$\Phi(\mathbf{x}_t) = (\mathbf{x}_t[1], \dots, \mathbf{x}_t[d], \mathbf{x}_t[1] \cdot \mathbf{x}_t[1], \mathbf{x}_t[1] \cdot \mathbf{x}_t[2], \dots, \mathbf{x}_t[d] \cdot \mathbf{x}_t[d], \dots)^\top$$

- Enumerating products up to order  $K$  (ie. products of at most  $K$  coordinates) we can get degree  $K$  polynomials.
- However dimension blows up as  $d^K$

# A FIRST CUT

- Given  $\mathbf{x}_t \in \mathbb{R}^d$ , the feature space vector is given by mapping

$$\Phi(\mathbf{x}_t) = (\mathbf{x}_t[1], \dots, \mathbf{x}_t[d], \mathbf{x}_t[1] \cdot \mathbf{x}_t[1], \mathbf{x}_t[1] \cdot \mathbf{x}_t[2], \dots, \mathbf{x}_t[d] \cdot \mathbf{x}_t[d], \dots)^\top$$

- Enumerating products up to order  $K$  (ie. products of at most  $K$  coordinates) we can get degree  $K$  polynomials.
- However dimension blows up as  $d^K$
- Is there a way to do this without enumerating  $\Phi$ ?

# KERNEL TRICK

# KERNEL TRICK

- Essence of Kernel trick:
  - If we can write down an algorithm only in terms of  $\Phi(\mathbf{x}_t)^\top \Phi(\mathbf{x}_s)$  for data points  $\mathbf{x}_t$  and  $\mathbf{x}_s$

# KERNEL TRICK

- Essence of Kernel trick:
  - If we can write down an algorithm only in terms of  $\Phi(\mathbf{x}_t)^\top \Phi(\mathbf{x}_s)$  for data points  $\mathbf{x}_t$  and  $\mathbf{x}_s$
  - Then we don't need to explicitly enumerate  $\Phi(\mathbf{x}_t)$ 's but instead, compute  $k(\mathbf{x}_t, \mathbf{x}_s) = \Phi(\mathbf{x}_t)^\top \Phi(\mathbf{x}_s)$  (even if  $\Phi$  maps to infinite dimensional space)

# KERNEL TRICK

- Essence of Kernel trick:
  - If we can write down an algorithm only in terms of  $\Phi(\mathbf{x}_t)^\top \Phi(\mathbf{x}_s)$  for data points  $\mathbf{x}_t$  and  $\mathbf{x}_s$
  - Then we don't need to explicitly enumerate  $\Phi(\mathbf{x}_t)$ 's but instead, compute  $k(\mathbf{x}_t, \mathbf{x}_s) = \Phi(\mathbf{x}_t)^\top \Phi(\mathbf{x}_s)$  (even if  $\Phi$  maps to infinite dimensional space)
- Example: RBF kernel  $k(\mathbf{x}_t, \mathbf{x}_s) = \exp(-\sigma \|\mathbf{x}_t - \mathbf{x}_s\|_2^2)$ , polynomial kernel  $k(\mathbf{x}_t, \mathbf{x}_s) = (\mathbf{x}_t^\top \mathbf{y}_t)^p$

# KERNEL TRICK

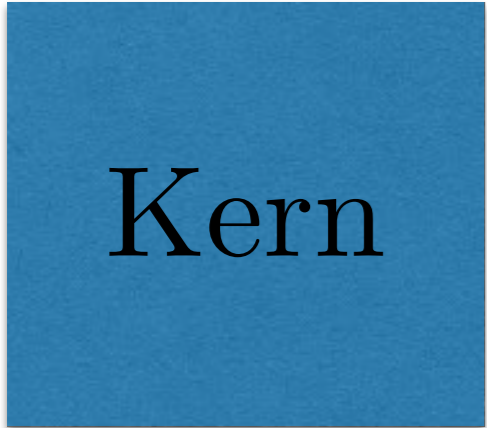
- Essence of Kernel trick:
  - If we can write down an algorithm only in terms of  $\Phi(\mathbf{x}_t)^\top \Phi(\mathbf{x}_s)$  for data points  $\mathbf{x}_t$  and  $\mathbf{x}_s$
  - Then we don't need to explicitly enumerate  $\Phi(\mathbf{x}_t)$ 's but instead, compute  $k(\mathbf{x}_t, \mathbf{x}_s) = \Phi(\mathbf{x}_t)^\top \Phi(\mathbf{x}_s)$  (even if  $\Phi$  maps to infinite dimensional space)
- Example: RBF kernel  $k(\mathbf{x}_t, \mathbf{x}_s) = \exp(-\sigma \|\mathbf{x}_t - \mathbf{x}_s\|_2^2)$ , polynomial kernel  $k(\mathbf{x}_t, \mathbf{x}_s) = (\mathbf{x}_t^\top \mathbf{y}_t)^p$
- Kernel function measures similarity between points.



# KERNEL PCA

# KERNEL PCA

1.



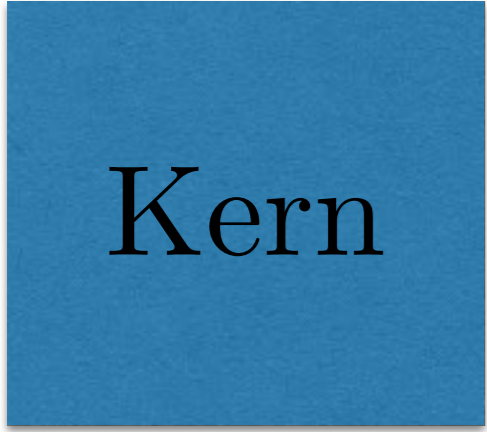
n

n

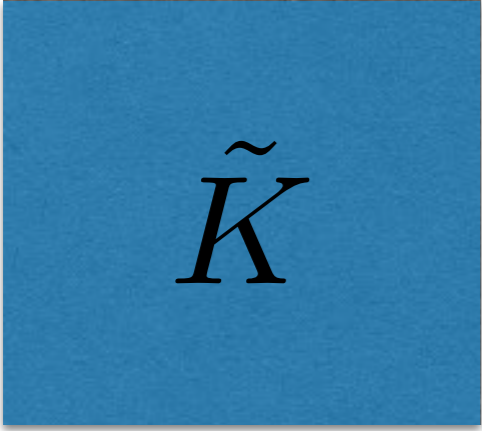
$$= \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_n) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ k(x_{n-1}, x_1) & k(x_{n-1}, x_2) & \dots & k(x_{n-1}, x_n) \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, x_n) \end{bmatrix}$$

# KERNEL PCA

1.


$$= \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_n) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ k(x_{n-1}, x_1) & k(x_{n-1}, x_2) & \dots & k(x_{n-1}, x_n) \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, x_n) \end{bmatrix}$$

2.


$$= \text{Kern} - \frac{1}{n} (\mathbf{1} \text{ Kern} + \text{Kern} \mathbf{1}) + \frac{1}{n^2} \mathbf{1} \text{ Kern} \mathbf{1}$$

# KERNEL PCA

# KERNEL PCA

$$3. \left[ \begin{array}{c} n \\ \mathbf{P} \\ K \end{array} , \gamma \right] = \text{eigs} \left( \begin{array}{c} \tilde{K} \\ K \end{array} \right)$$

# KERNEL PCA

$$3. \left[ \begin{array}{c} n \\ \mathbf{P} \\ K \end{array} ; \gamma \right] = \text{eigs} \left( \begin{array}{c} \tilde{K} \\ K \end{array} \right)$$

$$4. \begin{array}{c} n \\ \mathbf{a} \\ K \end{array} = n \begin{array}{c} \vdots \quad \vdots \\ \frac{P_1 \dots P_K}{\sqrt{n\gamma_1} \sqrt{n\gamma_K}} \\ \vdots \quad \vdots \\ K \end{array}$$

# KERNEL PCA

$$3. \begin{bmatrix} n \\ P \\ K \end{bmatrix}, \gamma = \text{eigs} \left( \begin{bmatrix} \tilde{K} \\ K \end{bmatrix} \right)$$

$$4. \begin{bmatrix} n \\ \alpha \\ K \end{bmatrix} = n \begin{bmatrix} P_1 \dots P_K \\ \sqrt{n\gamma_1} \dots \sqrt{n\gamma_K} \\ K \end{bmatrix}$$

$$5. \begin{bmatrix} n \\ Y \\ K \end{bmatrix} = n \begin{bmatrix} \tilde{K} \\ n \end{bmatrix} \times \begin{bmatrix} n \\ \alpha \\ K \end{bmatrix}$$

# LETS REWRITE PCA

- $k^{\text{th}}$  column of  $W$  is eigenvector of covariance matrix



# LETS REWRITE PCA

- $k^{\text{th}}$  column of  $W$  is eigenvector of covariance matrix  
That is,  $\lambda_k W_k = \Sigma W_k$ . Rewriting, for centered  $X$

# LETS REWRITE PCA

- $k^{\text{th}}$  column of  $W$  is eigenvector of covariance matrix  
That is,  $\lambda_k W_k = \Sigma W_k$ . Rewriting, for centered  $X$

$$\lambda_k W_k = \frac{1}{n} \left( \sum_{t=1}^n \mathbf{x}_t \mathbf{x}_t^{\top} \right) W_k = \frac{1}{n} \sum_{t=1}^n (\mathbf{x}_t^{\top} W_k) \mathbf{x}_t$$

# LETS REWRITE PCA

- $k^{\text{th}}$  column of  $W$  is eigenvector of covariance matrix  
That is,  $\lambda_k W_k = \Sigma W_k$ . Rewriting, for centered  $X$

$$\lambda_k W_k = \frac{1}{n} \left( \sum_{t=1}^n \mathbf{x}_t \mathbf{x}_t^{\top} \right) W_k = \frac{1}{n} \sum_{t=1}^n (\mathbf{x}_t^{\top} W_k) \mathbf{x}_t$$

$W_k$ 's can be written as linear combination of  $\mathbf{x}_t$ 's, as

$$W_k = \sum_{t=1}^n \alpha_k[t] \mathbf{x}_t$$

where  $\alpha_k[t] = \frac{1}{\lambda_k n} (\mathbf{x}_t^{\top} W_k)$

# LETS REWRITE PCA

- We have that  $W_k = \sum_{s=1}^n \alpha_k[s] \mathbf{x}_s$  and that  $\alpha_k[t] = \frac{1}{\lambda_k n} (\mathbf{x}_t^\top W_k)$ .

# LETS REWRITE PCA

- We have that  $W_k = \sum_{s=1}^n \alpha_k[s] \mathbf{x}_s$  and that  $\alpha_k[t] = \frac{1}{\lambda_k n} (\mathbf{x}_t^\top W_k)$ .
- Hence:

$$\alpha_k[t] = \frac{1}{\lambda_k n} \left( \mathbf{x}_t^\top \left( \sum_{s=1}^n \alpha_k[s] \mathbf{x}_s \right) \right) = \frac{1}{\lambda_k n} \sum_{s=1}^n \alpha_k[s] \mathbf{x}_t^\top \mathbf{x}_s$$

# LETS REWRITE PCA

- We have that  $W_k = \sum_{s=1}^n \alpha_k[s] \mathbf{x}_s$  and that  $\alpha_k[t] = \frac{1}{\lambda_k n} (\mathbf{x}_t^\top W_k)$ .
- Hence:

$$\alpha_k[t] = \frac{1}{\lambda_k n} \left( \mathbf{x}_t^\top \left( \sum_{s=1}^n \alpha_k[s] \mathbf{x}_s \right) \right) = \frac{1}{\lambda_k n} \sum_{s=1}^n \alpha_k[s] \mathbf{x}_t^\top \mathbf{x}_s$$

- Let  $\tilde{K}$  be a matrix such that  $\tilde{K}_{s,t} = \mathbf{x}_t^\top \mathbf{x}_s$ . Hence,  $\alpha_k[t] = \frac{1}{\lambda_k n} \alpha_k^\top \tilde{K}_t$  and

$$\alpha_k = \frac{1}{\lambda_k n} \tilde{K} \alpha_k$$

where  $\tilde{K}_t$  is the  $t$ 'th column of  $\tilde{K}$ .

# LETS REWRITE PCA

- We have that  $W_k = \sum_{s=1}^n \alpha_k[s] \mathbf{x}_s$  and that  $\alpha_k[t] = \frac{1}{\lambda_k n} (\mathbf{x}_t^\top W_k)$ .
- Hence:

$$\alpha_k[t] = \frac{1}{\lambda_k n} \left( \mathbf{x}_t^\top \left( \sum_{s=1}^n \alpha_k[s] \mathbf{x}_s \right) \right) = \frac{1}{\lambda_k n} \sum_{s=1}^n \alpha_k[s] \mathbf{x}_t^\top \mathbf{x}_s$$

- Let  $\tilde{K}$  be a matrix such that  $\tilde{K}_{s,t} = \mathbf{x}_t^\top \mathbf{x}_s$ . Hence,  $\alpha_k[t] = \frac{1}{\lambda_k n} \alpha_k^\top \tilde{K}_t$  and

$$\alpha_k = \frac{1}{\lambda_k n} \tilde{K} \alpha_k$$

where  $\tilde{K}_t$  is the  $t$ 'th column of  $\tilde{K}$ .

- Hence  $\alpha_k$  is in the direction of eigen vector of  $\tilde{K}$

# LETS REWRITE PCA

- Further, since  $W_k$  is unit norm,

$$1 = \|W_k\|_2^2 = \left( \sum_{t=1}^n \alpha_k[t] \mathbf{x}_t \right)^\top \left( \sum_{s=1}^n \alpha_k[s] \mathbf{x}_s \right) = \alpha_k^\top \tilde{K} \alpha_k = n \gamma_k \alpha_k^\top \alpha_k$$

Hence  $\|\alpha_k\|^2 = \frac{1}{n \gamma_k}$  where  $\gamma_k$  is the  $k$ 'th eigen value of matrix  $\tilde{K}$



# LETS REWRITE PCA

- However  $W_k$  itself is in feature space and has the same dimensionality of  $\Phi(x)$  (which is possibly infinite)!

# LETS REWRITE PCA

- However  $W_k$  itself is in feature space and has the same dimensionality of  $\Phi(x)$  (which is possibly infinite)!
- However, the projections are in  $K$  dimensions and we can hope to directly compute these as:

$$y_i[k] = \mathbf{x}_i^\top W_k = \sum_{t=1}^n \alpha_k[t] \tilde{K}_{t,i}$$

# REWRITING PCA

- We assumed centered data, what if its not,

$$\begin{aligned}\tilde{K}_{s,t} &= \left( \mathbf{x}_t - \frac{1}{n} \sum_{u=1}^n \mathbf{x}_u \right)^\top \left( \mathbf{x}_s - \frac{1}{n} \sum_{u=1}^n \mathbf{x}_u \right) \\ &= \mathbf{x}_t^\top \mathbf{x}_s - \left( \frac{1}{n} \sum_{u=1}^n \mathbf{x}_u \right)^\top \mathbf{x}_s - \left( \frac{1}{n} \sum_{u=1}^n \mathbf{x}_u \right)^\top \mathbf{x}_t \\ &\quad + \frac{1}{n^2} \left( \sum_{u=1}^n \mathbf{x}_u \right)^\top \left( \sum_{v=1}^n \mathbf{x}_v \right) \\ &= \mathbf{x}_t^\top \mathbf{x}_s - \frac{1}{n} \sum_{u=1}^n \mathbf{x}_u^\top \mathbf{x}_s - \frac{1}{n} \sum_{u=1}^n \mathbf{x}_u^\top \mathbf{x}_t + \frac{1}{n^2} \sum_{u=1}^n \sum_{v=1}^n \mathbf{x}_u^\top \mathbf{x}_v\end{aligned}$$

# REWRITING PCA

- Equivalently, if **Kern** is the matrix ( $\text{Kern}_{t,s} = x_t^\top x_s$ ),

$$\tilde{K} = \text{Kern} - \frac{(\mathbf{1}_{n \times n} \times \text{Kern})}{n} - \frac{(\text{Kern} \times \mathbf{1}_{n \times n})}{n} + \frac{(\mathbf{1}_{n \times n} \times \text{Kern} \times \mathbf{1}_{n \times n})}{n^2}$$

# PCA REWRITTEN

- Compute  $\tilde{K} = \text{Kern} - \mathbf{1} \text{Kern}/n - \text{Kern} \mathbf{1}/n + \mathbf{1} \text{Kern} \mathbf{1}/n^2$

# PCA REWRITTEN

- Compute  $\tilde{K} = \text{Kern} - \mathbf{1} \text{ Kern}/n - \text{Kern} \mathbf{1}/n + \mathbf{1} \text{ Kern} \mathbf{1}/n^2$
- Compute top  $K$  eigen vectors  $P_1, \dots, P_K$  along with eigen values  $\gamma_1, \dots, \gamma_K$  for the matrix  $\tilde{K}$

# PCA REWRITTEN

- Compute  $\tilde{K} = \text{Kern} - \mathbf{1} \text{ Kern}/n - \text{Kern} \mathbf{1}/n + \mathbf{1} \text{ Kern} \mathbf{1}/n^2$
- Compute top  $K$  eigen vectors  $P_1, \dots, P_K$  along with eigen values  $\gamma_1, \dots, \gamma_K$  for the matrix  $\tilde{K}$
- Rescale each  $P_k$  by the inverse of the square-root of corresponding eigen values ie.  $\alpha_k = P_k / \sqrt{n\gamma_k}$

# PCA REWRITTEN

- Compute  $\tilde{K} = \text{Kern} - \mathbf{1} \text{Kern}/n - \text{Kern} \mathbf{1}/n + \mathbf{1} \text{Kern} \mathbf{1}/n^2$
- Compute top  $K$  eigen vectors  $P_1, \dots, P_K$  along with eigen values  $\gamma_1, \dots, \gamma_K$  for the matrix  $\tilde{K}$
- Rescale each  $P_k$  by the inverse of the square-root of corresponding eigen values ie.  $\alpha_k = P_k / \sqrt{n\gamma_k}$
- Compute projections by setting

$$y_i[k] = \sum_{t=1}^n \alpha_k[t] \tilde{K}_{t,i}$$

or in other words  $Y = \tilde{K} \times [\alpha_1, \dots, \alpha_K]$



# KERNEL PCA

# KERNEL PCA

All we need to be able to compute, to perform PCA are  $\mathbf{x}_t^T \mathbf{x}_s$

# KERNEL PCA

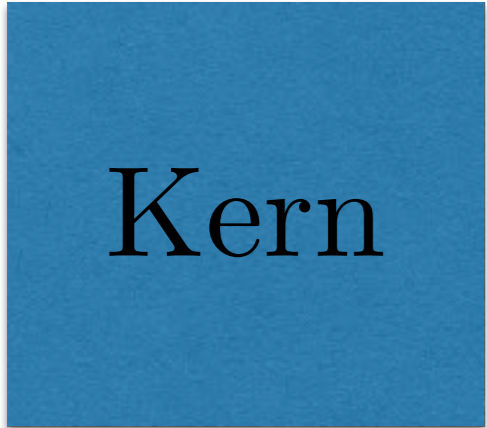
All we need to be able to compute, to perform PCA are  $\mathbf{x}_t^\top \mathbf{x}_s$

Replace  $\mathbf{x}_t^\top \mathbf{x}_s$  with  $\Phi(\mathbf{x}_t)^\top \Phi(\mathbf{x}_s) = k(x_t, x_s)$  to perform PCA  
in feature space

# KERNEL PCA

# KERNEL PCA

1.



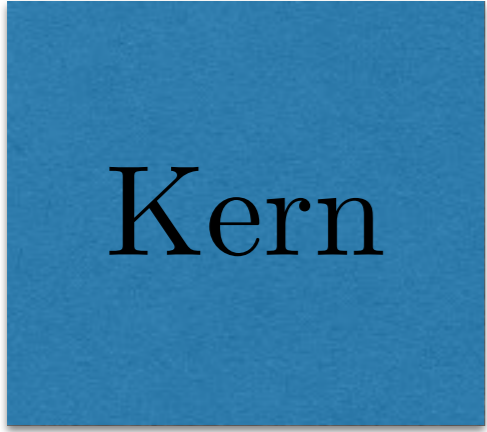
n

n

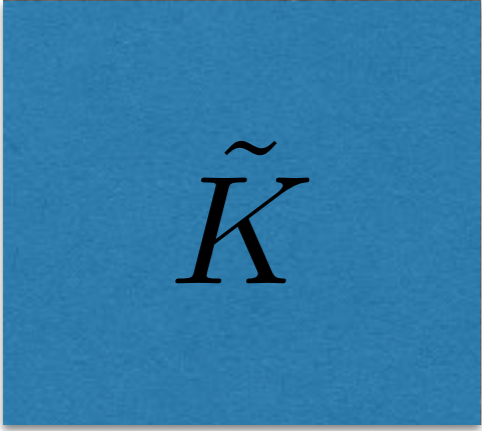
$$= \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_n) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ k(x_{n-1}, x_1) & k(x_{n-1}, x_2) & \dots & k(x_{n-1}, x_n) \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, x_n) \end{bmatrix}$$

# KERNEL PCA

1.


$$= \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_n) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ k(x_{n-1}, x_1) & k(x_{n-1}, x_2) & \dots & k(x_{n-1}, x_n) \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, x_n) \end{bmatrix}$$

2.


$$= \text{Kern} - \frac{1}{n} (\mathbf{1} \text{ Kern} + \text{Kern} \mathbf{1}) + \frac{1}{n^2} \mathbf{1} \text{ Kern} \mathbf{1}$$

# KERNEL PCA

# KERNEL PCA

$$3. \left[ \begin{array}{c} n \\ \color{red}{P} \\ K \end{array} , \gamma \right] = \text{eigs} \left( \begin{array}{c} \color{blue}{\tilde{K}} \\ K \end{array} \right)$$



# KERNEL PCA

$$3. \begin{bmatrix} n \\ P \\ K \end{bmatrix}, \gamma = \text{eigs} \left( \begin{bmatrix} \tilde{K} \\ K \end{bmatrix} \right)$$

$$4. \begin{bmatrix} n \\ \alpha \\ K \end{bmatrix} = n \begin{bmatrix} \vdots & \vdots \\ \frac{P_1}{\sqrt{n\gamma_1}} & \dots & \frac{P_K}{\sqrt{n\gamma_K}} \\ \vdots & \vdots \end{bmatrix}$$

# KERNEL PCA

$$3. \begin{bmatrix} n \\ P \\ \gamma \end{bmatrix} = \text{eigs} \left( \begin{bmatrix} \tilde{K} \\ K \end{bmatrix} \right)$$

$$4. \begin{bmatrix} n \\ \alpha \end{bmatrix} = n \begin{bmatrix} P_1 \dots P_K \\ \frac{1}{\sqrt{n\gamma_1}} \dots \frac{1}{\sqrt{n\gamma_K}} \end{bmatrix}$$

$$5. \begin{bmatrix} n \\ Y \end{bmatrix} = \begin{bmatrix} \tilde{K} \\ n \end{bmatrix} \times \begin{bmatrix} n \\ \alpha \end{bmatrix}$$

Demo