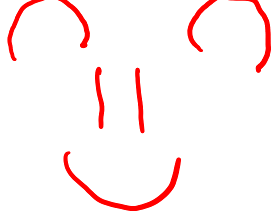
Warm-up as you log in



1. https://www.sporcle.com/games/MrChewypoo/minimalist_disney
2. https://www.sporcle.com/games/Stanford0008/minimalist-cartoons-slideshow
3. https://www.sporcle.com/games/MrChewypoo/minimalist

Announcements

Assignments

- HW9
  - Due Wed, 12/9, 11:59 pm
  - The two slip days are free (last possible submission Fri, 12/11, 11:59 pm)

Final Exam

- Mon, 12/14
- Stay tuned to Piazza for more details

Wrap-up Clustering

Clustering slides

Introduction to Machine Learning

Dimensionality Reduction and PCA

Instructor: Pat Virtue

Learning Paradigms

Paradigm	**Data**
Supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N \qquad \mathbf{x} \sim p^*(\cdot)$ and $y = c^*(\cdot)$
$\hookrightarrow$ Regression	$y^{(i)} \in \mathbb{R}$
$\hookrightarrow$ Classification	$y^{(i)} \in \{1, \ldots, K\}$
$\hookrightarrow$ Binary classification	$y^{(i)} \in \{+1, -1\}$
$\hookrightarrow$ Structured Prediction	$\mathbf{y}^{(i)}$ is a vector
Unsupervised	$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N \qquad \mathbf{x} \sim p^*(\cdot)$
Semi-supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N_1} \cup \{\mathbf{x}^{(j)}\}_{j=1}^{N_2}$
Online	$\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), (\mathbf{x}^{(3)}, y^{(3)}), \ldots\}$
Active Learning	$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$ and can query $y^{(i)} = c^*(\cdot)$ at a cost
Imitation Learning	$\mathcal{D} = \{(s^{(1)}, a^{(1)}), (s^{(2)}, a^{(2)}), \ldots\}$
Reinforcement Learning	$\mathcal{D} = \{(s^{(1)}, a^{(1)}, r^{(1)}), (s^{(2)}, a^{(2)}, r^{(2)}), \ldots\}$

Outline

Dimensionality Reduction

- High-dimensional data
- Learning (low dimensional) representations

Principal Component Analysis (PCA)

- Examples: 2D and 3D
- PCA algorithm
- PCA objective and optimization
- PCA, eigenvectors, and eigenvalues

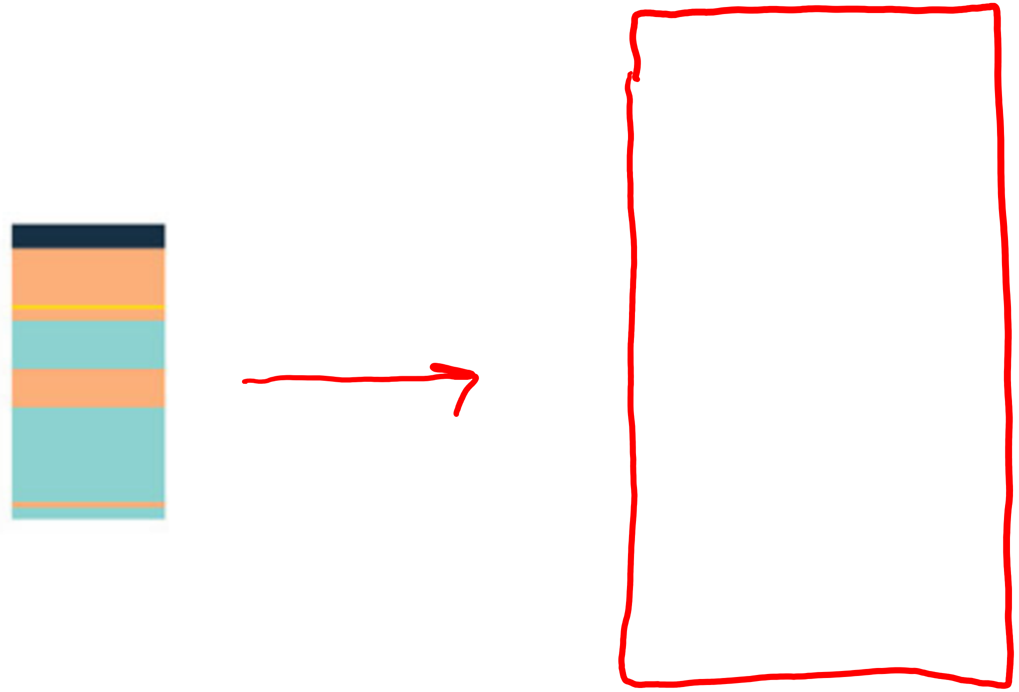
Warm-up as you log in

1. https://www.sporcle.com/games/MrChewypoo/minimalist_disney

2. https://www.sporcle.com/games/Stanford0008/minimalist-cartoons-slideshow

3. https://www.sporcle.com/games/MrChewypoo/minimalist

Dimensionality Reduction



Dimensionality Reduction

$x' \in \mathbb{R}^{10,000}$

$z \in \mathbb{R}^{30}$

Dimensionality Reduction

$x \in \mathbb{R}^{10000}$

$\xrightarrow{f(x)}$

$z \in \mathbb{R}^{30}$

Dimensionality Reduction

$x \in \mathbb{R}^{10000}$

$f(x)$

$z \in \mathbb{R}^{30}$

$g(x)$

$x' \in \mathbb{R}^{10000}$

$\|x - x'\|_2$

Dimensionality Reduction

For each $\vec{x}^{(i)} \in \mathbb{R}^M$ find representation $\vec{z}^{(i)} \in \mathbb{R}^K$ where $K \ll M$

$$\vec{z} = f(\vec{x})$$

$$x' = g(\vec{z})$$

$$\|x - x'\|_2^2$$

Reconstruction Error

$$\min_{f,g} \frac{1}{N} \sum_{i=1}^{N} \|x^{(i)} - x'^{(i)}\|_2^2$$

$$x' = g(f(x))$$

High Dimension Data

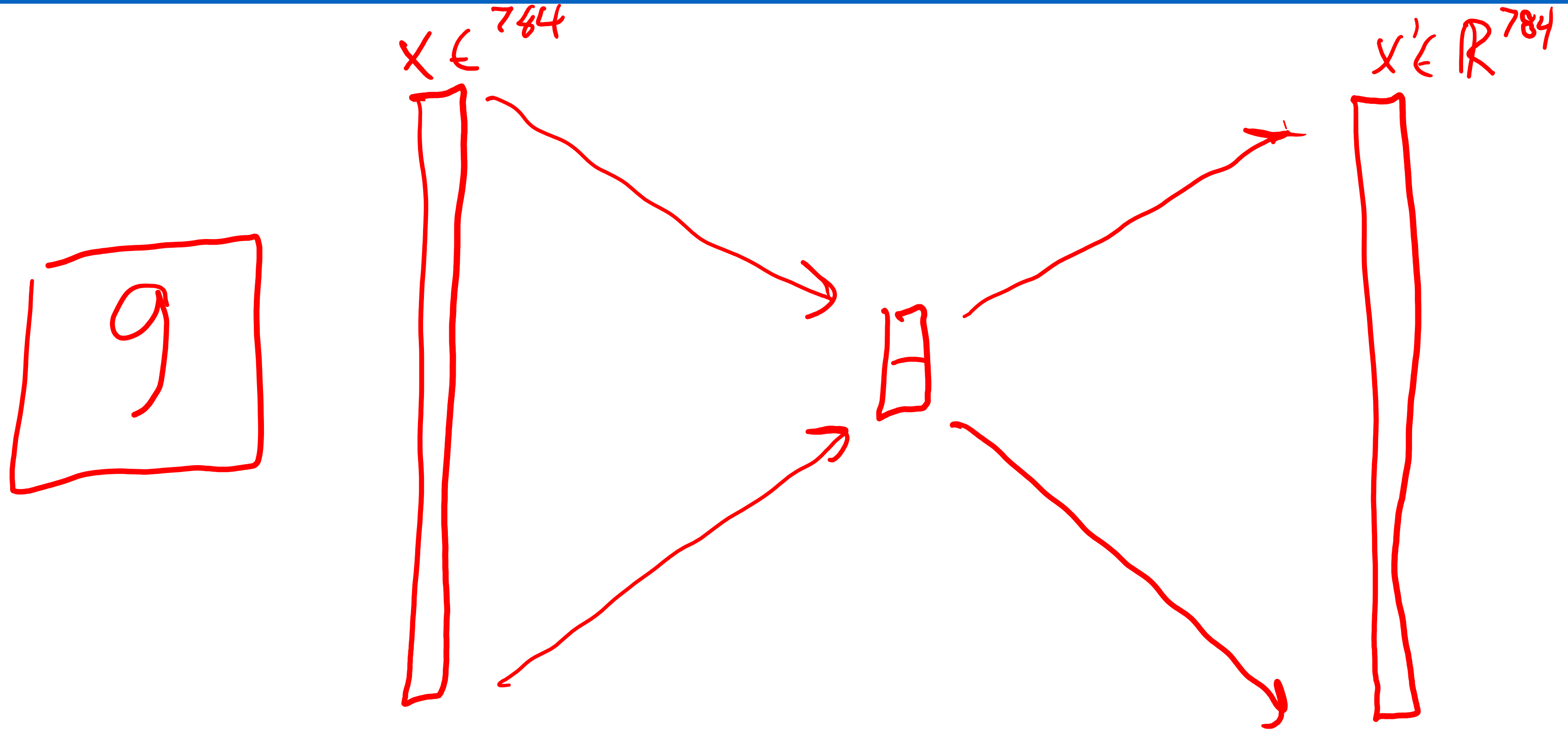
Examples of high dimensional data:

– High resolution images (millions of pixels)

Dimensionality Reduction

http://timbaumann.info/svd-image-compression-demo/

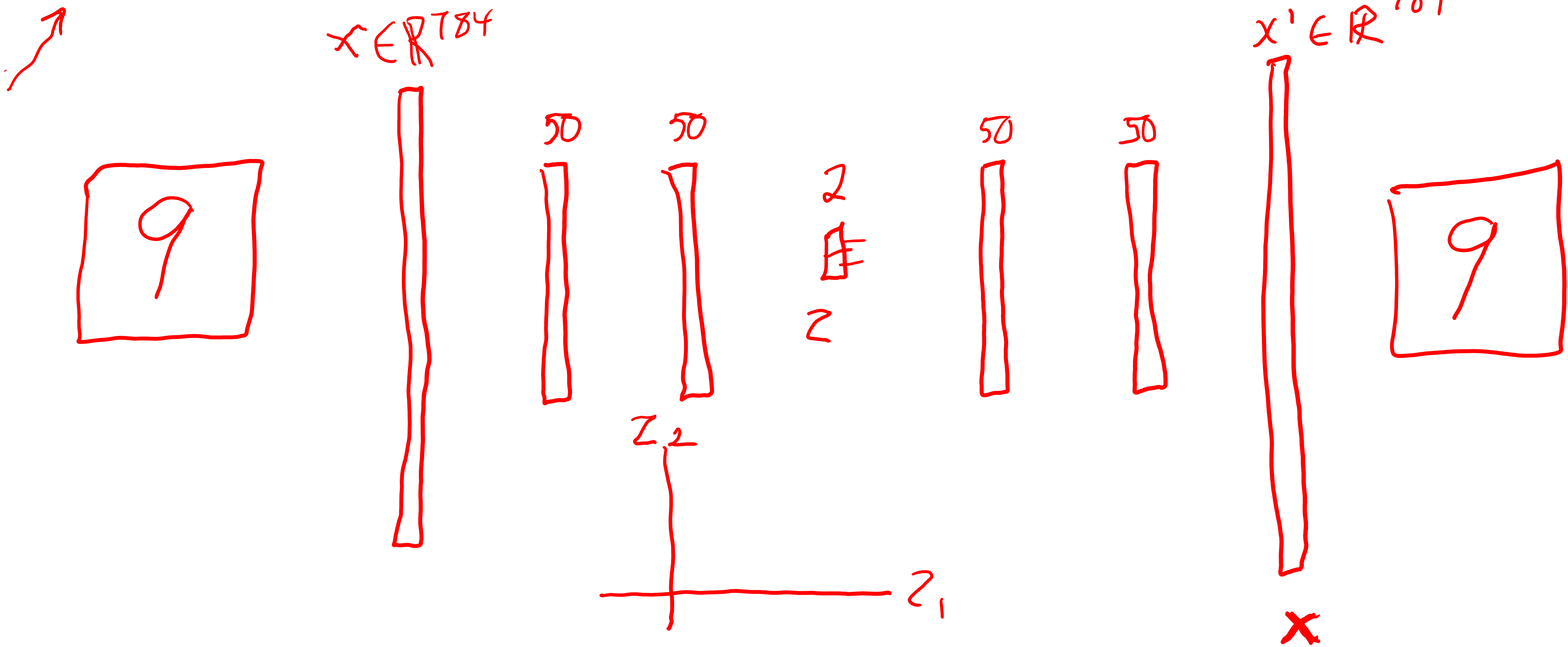
https://cs.stanford.edu/people/karpathy/convnetjs/demo/autoencoder.html



Dimensionality Reduction

http://timbaumann.info/svd-image-compression-demo/

https://cs.stanford.edu/people/karpathy/convnetjs/demo/autoencoder.html



High Dimension Data

Examples of high dimensional data:

– Brain Imaging Data (100s of MBs per scan)

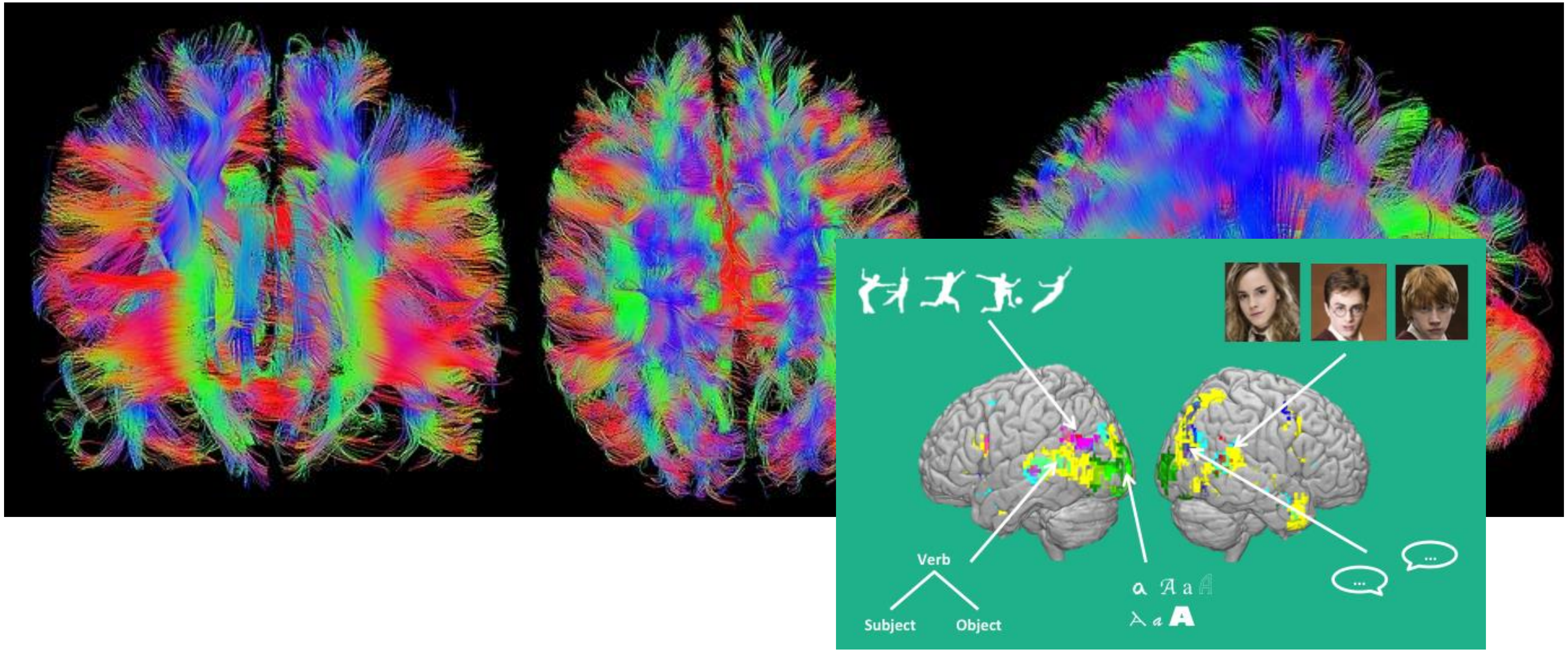


Image from (Wehbe et al., 2014)

Image from https://pixabay.com/en/brain-mrt-magnetic-resonance-imaging-1728449/

Learning Representations

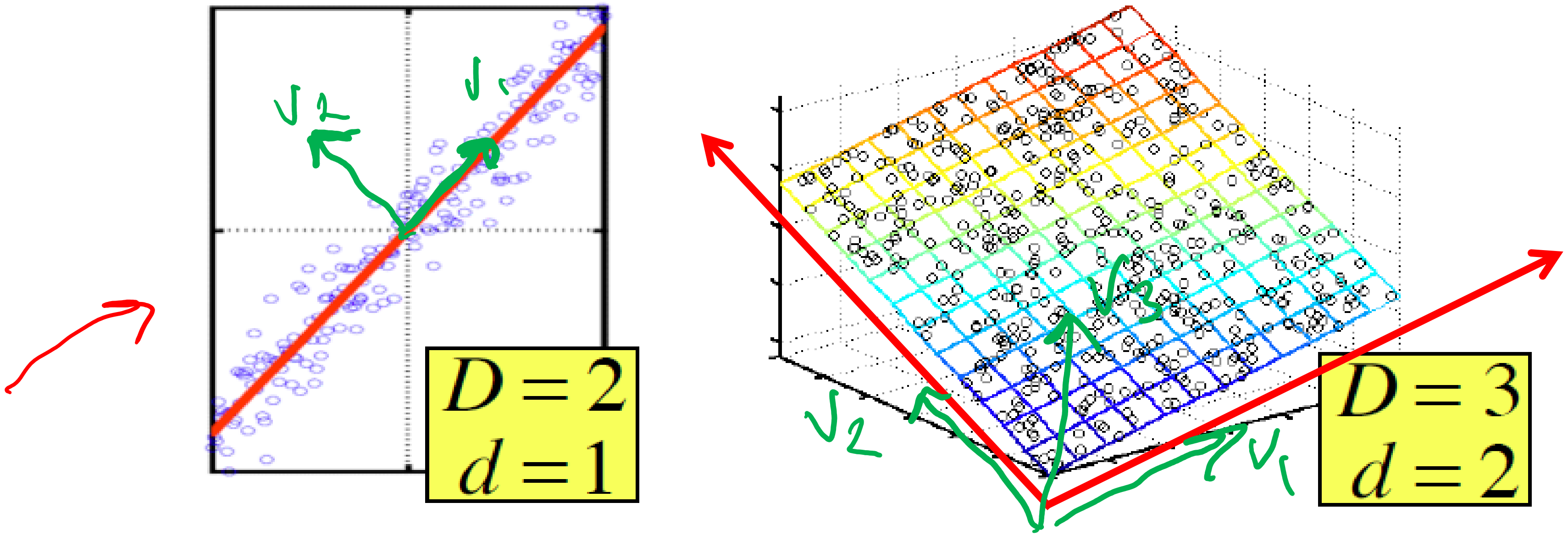
PCA, Kernel PCA, ICA: Powerful unsupervised learning techniques for extracting hidden (potentially lower dimensional) structure from high dimensional datasets.

Useful for:

- Visualization
- More efficient use of resources (e.g., time, memory, communication)
- Statistical: fewer dimensions → better generalization
- Noise removal (improving data quality)
- Further processing by machine learning algorithms

PRINCIPAL COMPONENT ANALYSIS (PCA)

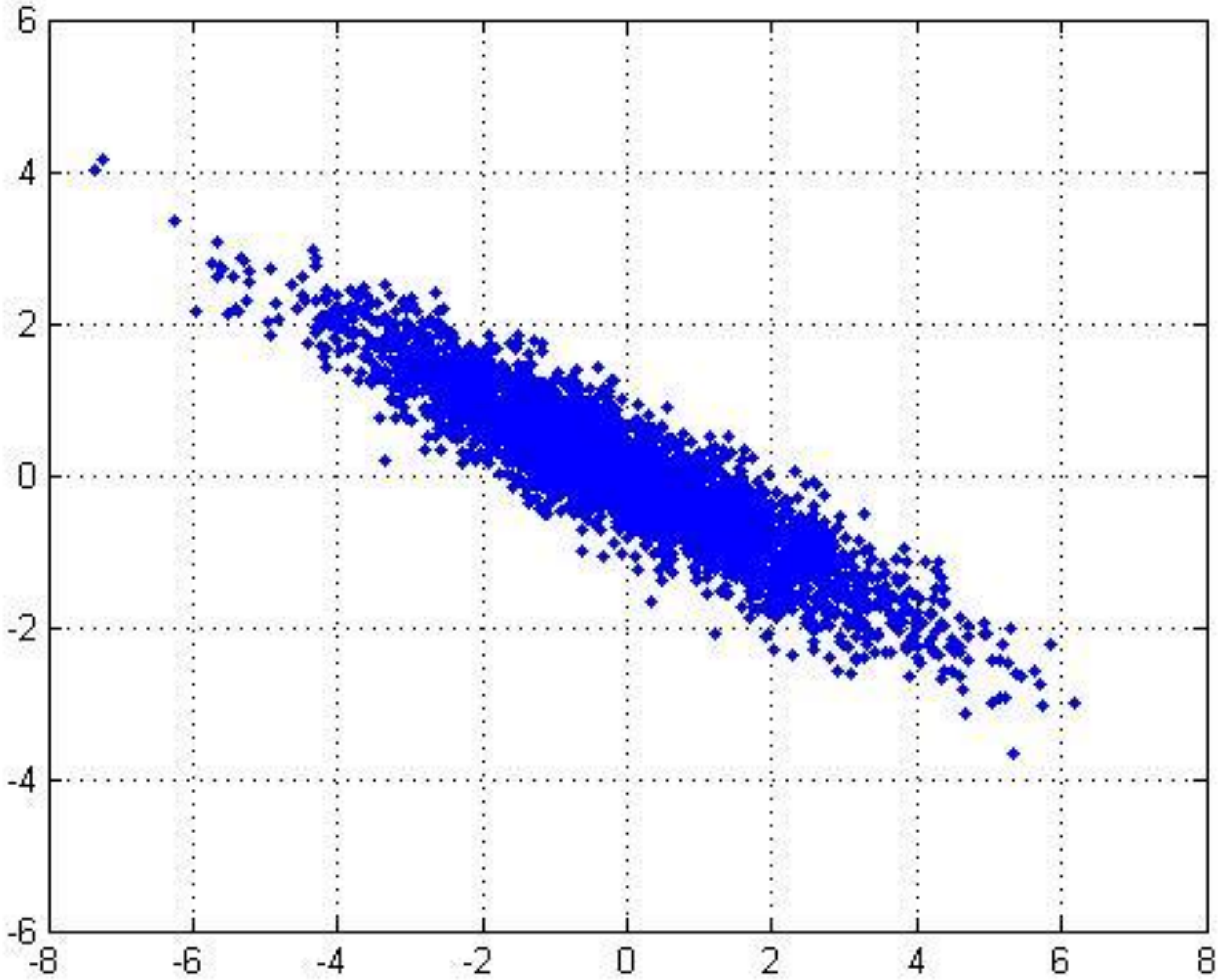
Principal Component Analysis (PCA)



In case where data lies on or near a low d-dimensional linear subspace, axes of this subspace are an effective representation of the data.

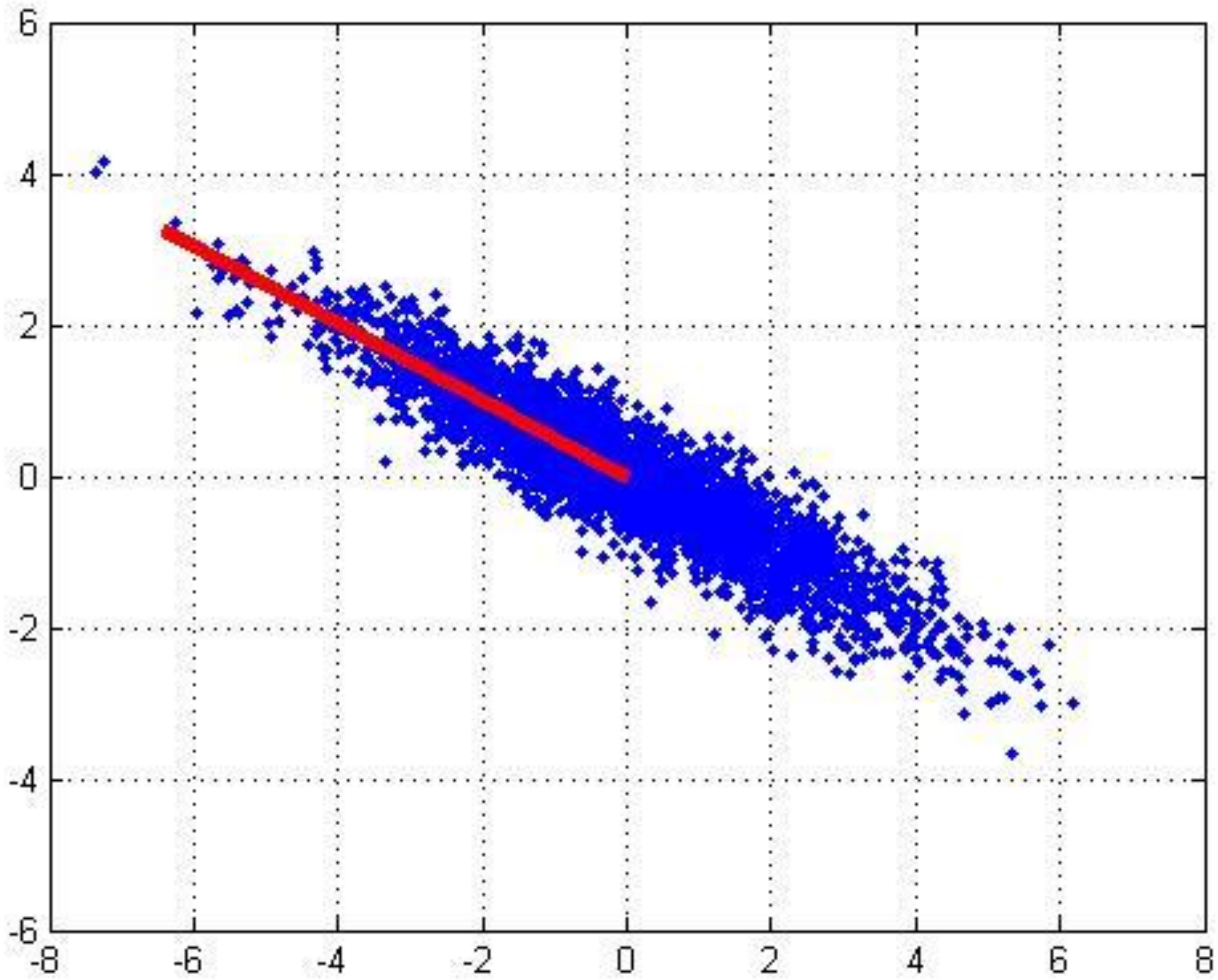
Identifying the axes is known as Principal Components Analysis, and can be obtained by using classic matrix computation tools (Eigen or Singular Value Decomposition).

2D Gaussian dataset



Slide from Barnabas Poczos

1st PCA axis

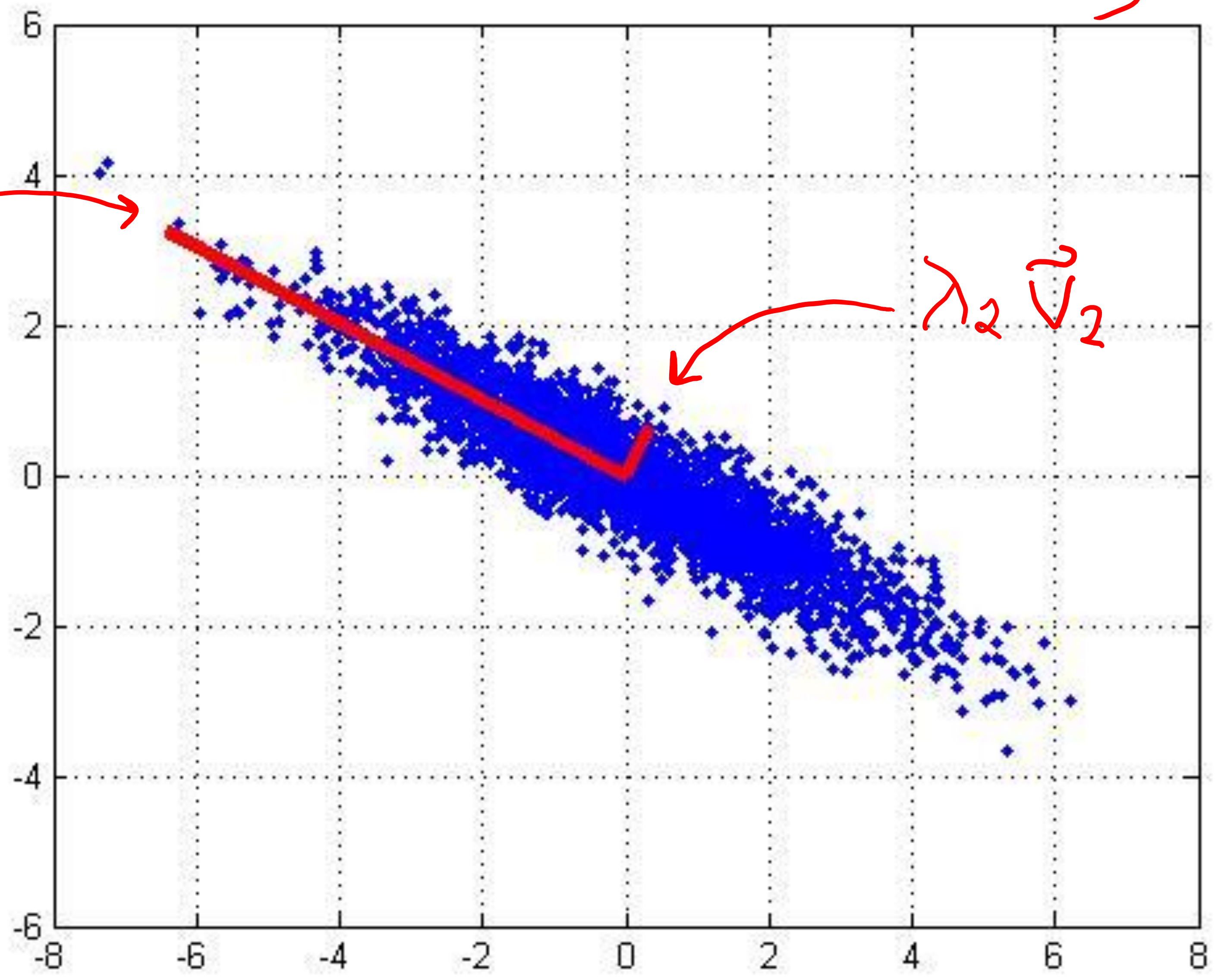


Slide from Barnabas Poczos

2^{nd} PCA axis

$eig(X^T X)$

$\lambda_1 \vec{v}_1$

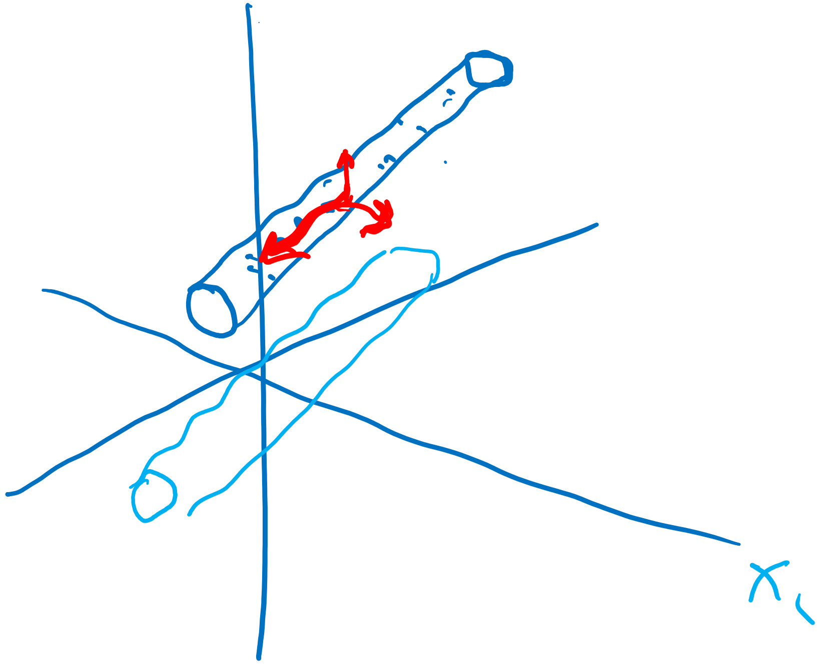


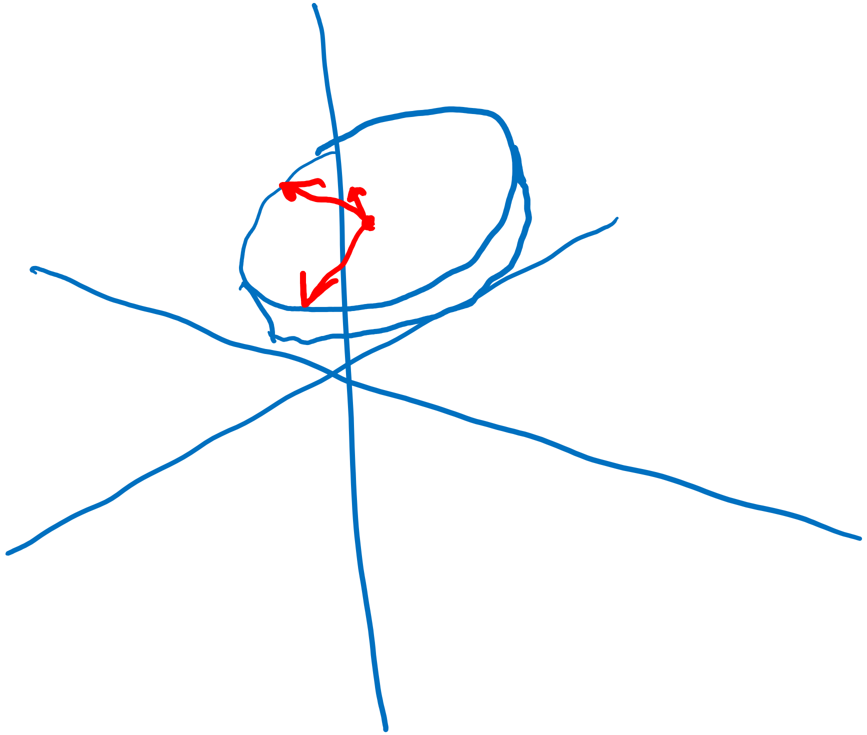
$\lambda_2 \vec{v}_2$

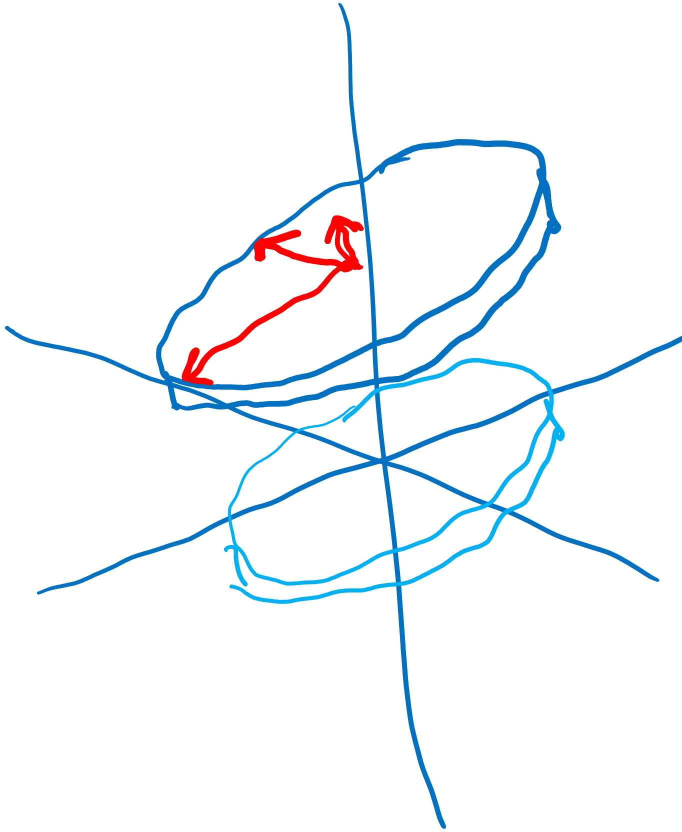
Slide from Barnabas Poczos

$N = 100$
$M = 3$

PCA Axes







Data for PCA

$$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$$

$$\mathbf{X} = \begin{bmatrix} (\mathbf{x}^{(1)})^T \\ (\mathbf{x}^{(2)})^T \\ \vdots \\ (\mathbf{x}^{(N)})^T \end{bmatrix}$$

We assume the data is **centered**

$$\mu = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}^{(i)} = \mathbf{0}$$

Q: What if your data is **not** centered?

A: Subtract off the sample mean

Slide from Matt Gormley

Sample Covariance Matrix

The sample covariance matrix is given by:

$$\Sigma_{jk} = \frac{1}{N} \sum_{i=1}^{N} (x_j^{(i)} - \mu_j)(x_k^{(i)} - \mu_k)$$

Since the data matrix is centered, we rewrite as:

$$\mathbf{\Sigma} = \frac{1}{N} \mathbf{X}^T \mathbf{X}$$

$$[M \times N] \cdot [N \times M]$$

$$\mathbf{X} = \begin{bmatrix} (\mathbf{x}^{(1)})^T \\ (\mathbf{x}^{(2)})^T \\ \vdots \\ (\mathbf{x}^{(N)})^T \end{bmatrix}$$

PCA Algorithm

V $\quad K < M$

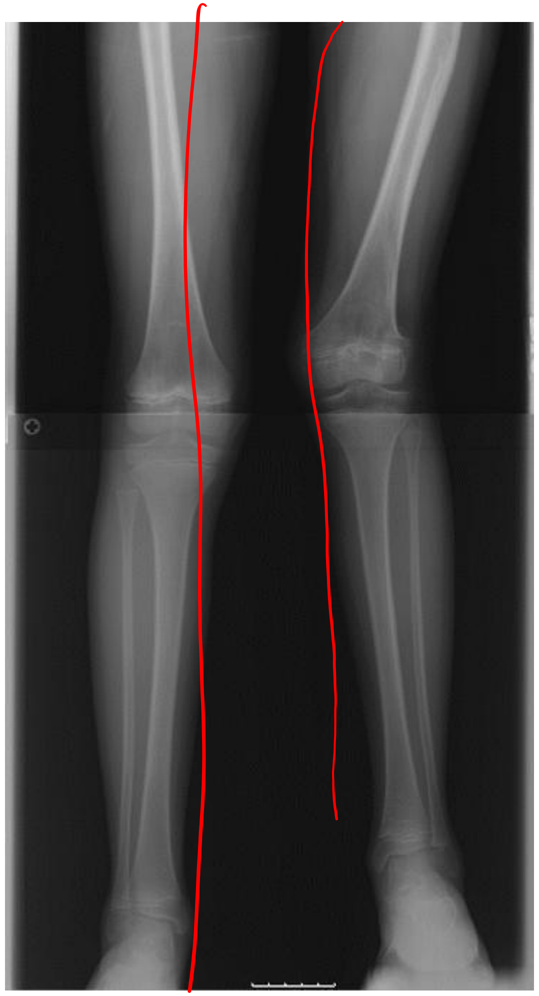
Input: $\boldsymbol{X}, \boldsymbol{X}_{test}, K$

1. Center data (and scale each axis) based on training data → $\boldsymbol{X}, \boldsymbol{X}_{\boldsymbol{test}}$
2. $\boldsymbol{V}$ = eigenvectors$(\boldsymbol{X}^T\boldsymbol{X})$
3. Keep only the top K eigenvectors: $\boldsymbol{V}_K$
4. $\mathbf{Z}_{\text{test}} = \boldsymbol{X}_{test}\boldsymbol{V}_K$

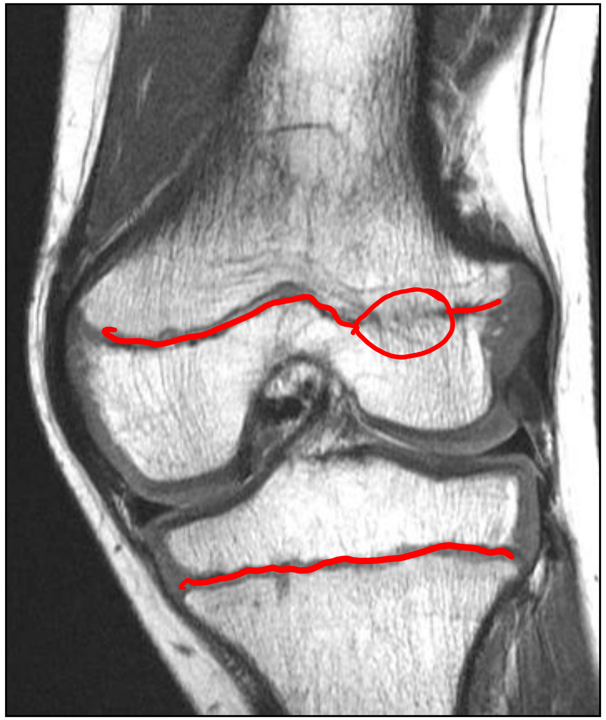
Optionally, use $\boldsymbol{V}_K^T$ to rotate $\mathbf{Z}_{\text{test}}$ back to original subspace $\mathbf{X}'_{\text{test}}$ and uncenter

Growth Plate Imaging

Growth Plate Disruption and Limb Length Discrepancy



8 year-old boy with previous fracture and
4cm leg length discrepancy



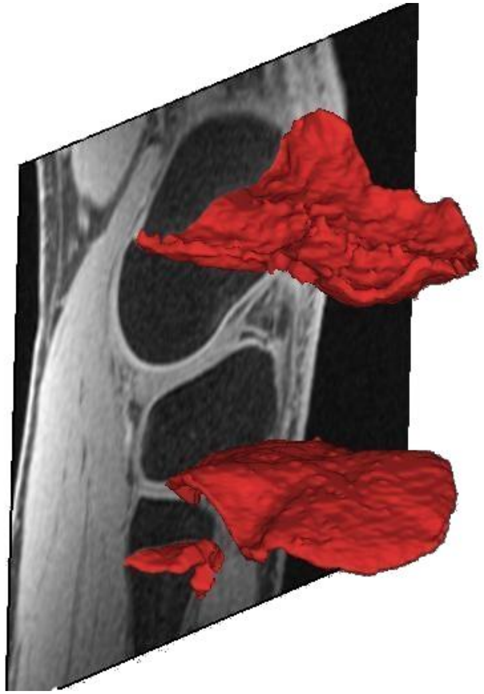
Images Courtesy
H. Potter, H.S.S.

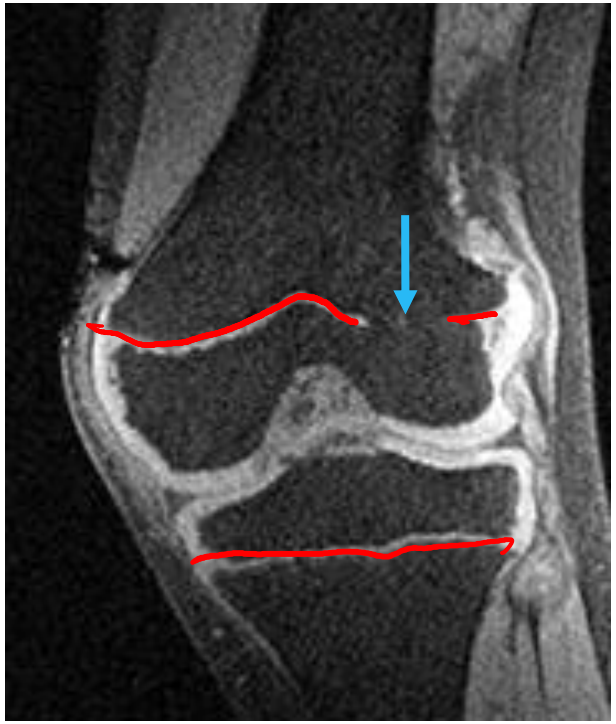
imagination at work

Growth Plate Imaging

Growth Plate Disruption and Limb Length Discrepancy

8 year-old boy with previous fracture and
4cm leg length discrepancy

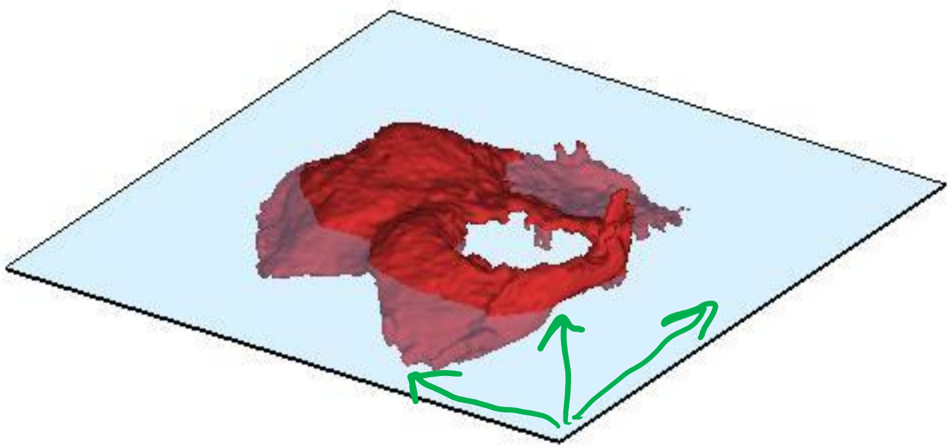




Images Courtesy
H. Potter, H.S.S.

Growth Plate Imaging

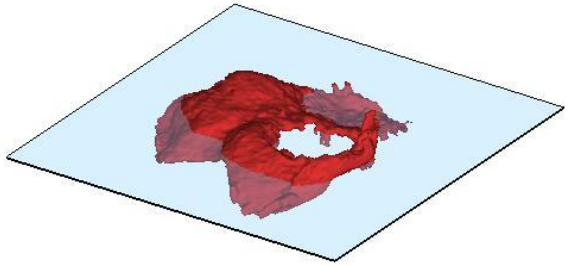
Area Measurement

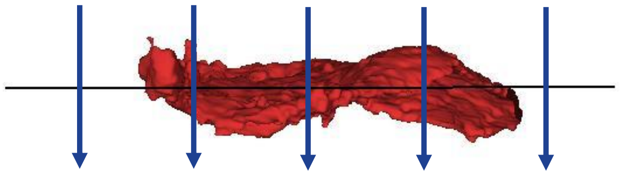


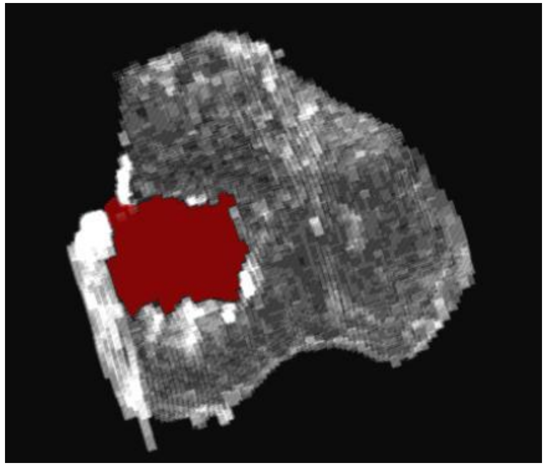
imagination at work

Growth Plate Imaging

Area Measurement







Flatten Growth Plate to Enable 2D Area Measurement

Piazza Poll 1

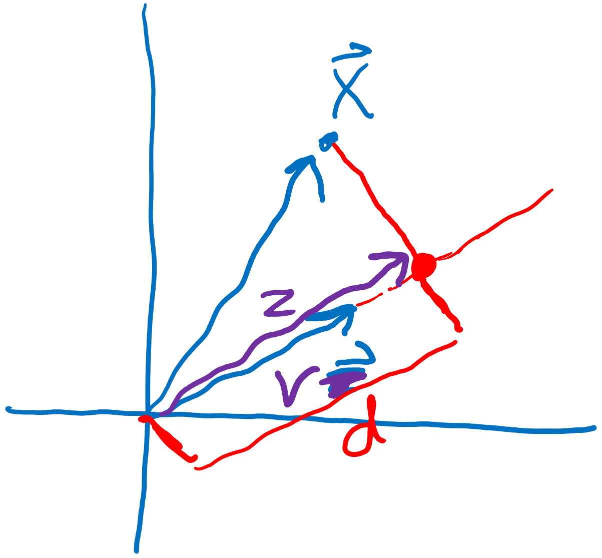
What is the projection of point $\boldsymbol{x}$ onto vector $\boldsymbol{v}$, assuming that $\|\boldsymbol{v}\|_2 = 1$?

~~A.~~ $\boldsymbol{v}\boldsymbol{x}$

B. $\boldsymbol{v}^T\boldsymbol{x}$

C. $(\boldsymbol{v}^T\boldsymbol{x})\boldsymbol{v}$

~~D.~~ $\boldsymbol{v}^T\boldsymbol{x}\,\boldsymbol{x}^T\boldsymbol{v}$



$$d = \frac{v^T x}{\|v\|_2}$$

$$\vec{z} = d\,\vec{v}$$

Rotation of Data (and back)

$$V = \begin{bmatrix} \leftarrow v_1^T \rightarrow \\ \leftarrow v_2^T \rightarrow \\ \vdots \end{bmatrix}$$

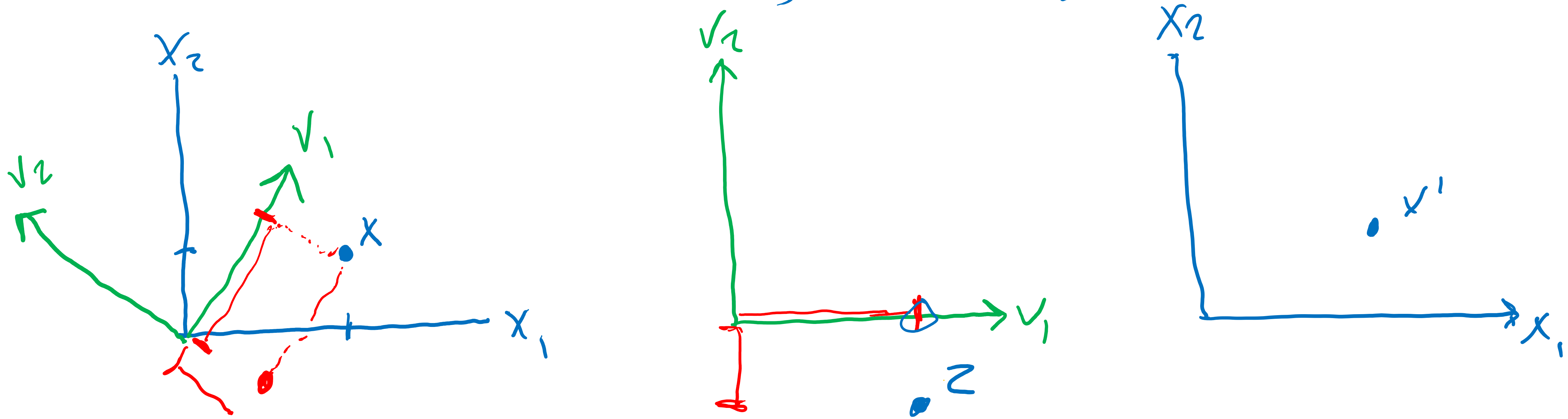
1. For any orthogonal matrix $\boldsymbol{V} \in \mathbb{R}^{M \times M}$
2. Rotate to new space: $\quad \boldsymbol{z}^{(i)} = \boldsymbol{V}\boldsymbol{x}^{(i)} \quad \forall i$
3. (Un)rotate back: $\quad \boldsymbol{x}'^{(i)} = \boldsymbol{V}^T \boldsymbol{z}^{(i)}$

$V^T V = I$

$V^T = V^{-1}$

Vx

$V^T z$

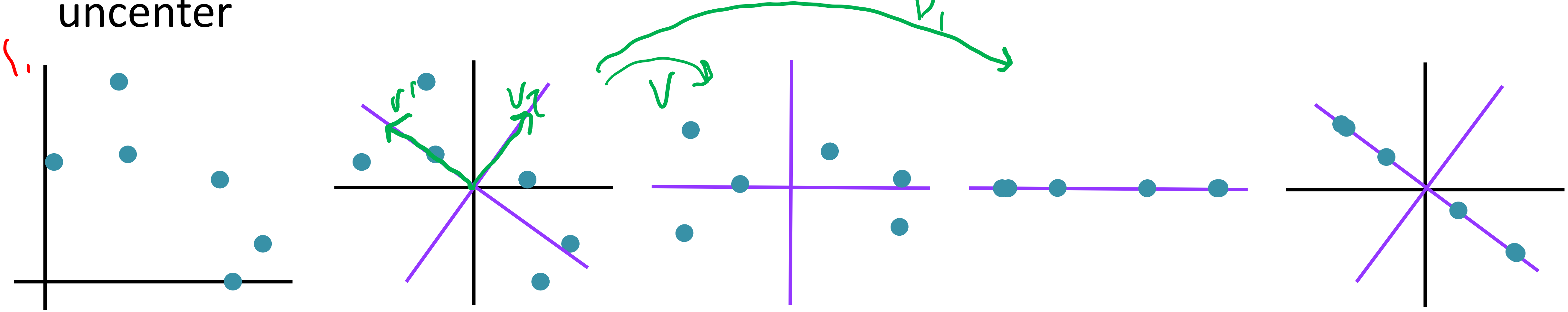


PCA Algorithm

Input: $\boldsymbol{X}, \boldsymbol{X}_{test}, K$

1. Center data (and scale each axis) based on training data → $\boldsymbol{X}, \boldsymbol{X}_{test}$
2. $\boldsymbol{V}$ = eigenvectors($\boldsymbol{X}^T\boldsymbol{X}$)
3. Keep only the top K eigenvectors: $\boldsymbol{V}_K$
4. $\mathbf{Z}_{\text{test}} = \boldsymbol{X}_{test}\boldsymbol{V}_K$

Optionally, use $\boldsymbol{V}_K^T$ to rotate $\mathbf{Z}_{\text{test}}$ back to original subspace $\mathbf{X}'_{\text{test}}$ and uncenter



Outline

Dimensionality Reduction

- High-dimensional data
- Learning (low dimensional) representations

Principal Component Analysis (PCA)

- Examples: 2D and 3D
- PCA algorithm
- PCA objective and optimization
- PCA, eigenvectors, and eigenvalues

Sketch of PCA

1. Select "best" $\boldsymbol{V} \in \mathbb{R}^{K \times M}$
2. Project down: $\boldsymbol{z}^{(i)} = \boldsymbol{V}\boldsymbol{x}^{(i)} \quad \forall i$
3. Reconstruct up: $\boldsymbol{x}'^{(i)} = \boldsymbol{V}^T \boldsymbol{z}^{(i)}$

Sketch of PCA

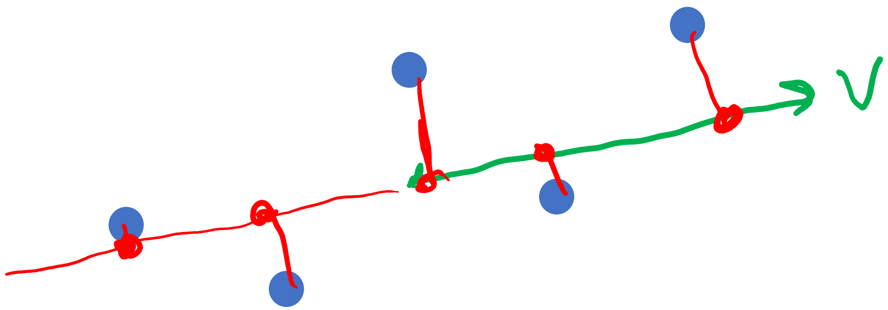
1. Select “best” $\boldsymbol{V} \in \mathbb{R}^{K \times M}$
2. Project down: $\boldsymbol{z}^{(i)} = \boldsymbol{V}\boldsymbol{x}^{(i)} \quad \forall i$
3. Reconstruct up: $\boldsymbol{x}'^{(i)} = \boldsymbol{V}^T \boldsymbol{z}^{(i)}$

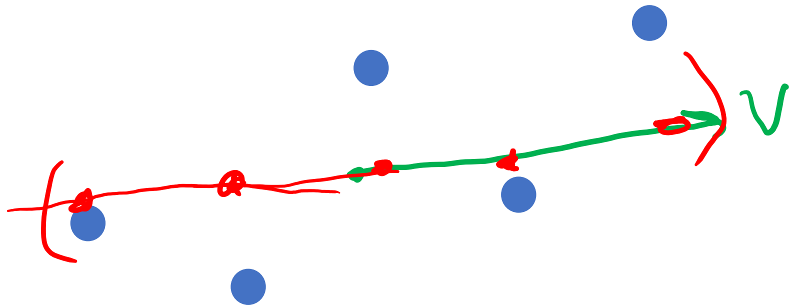
Definition of PCA

1. Select v_1 that best explains data
2. Select next v_j that
    i. Is orthogonal to $v_1, \dots, v_{j-1}$
    ii. Best explains remaining data
3. Repeat 2 until desired amount of data is explained

Select "Best" Vector

Reconstruction Error vs Variance of Projection





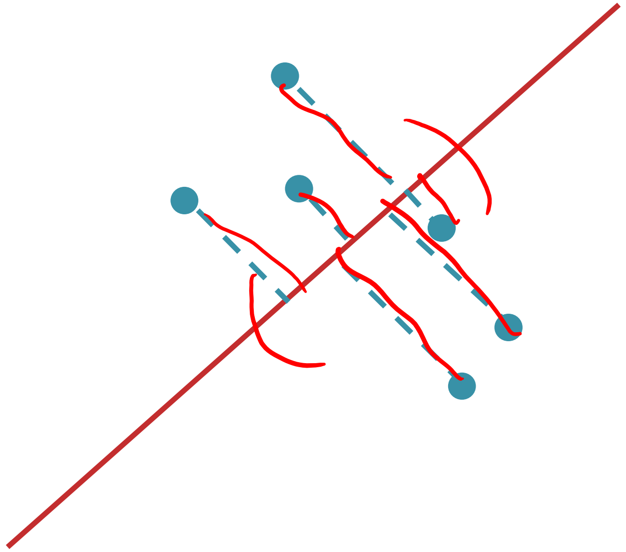
Piazza Poll 2 & Poll 3

Consider the two projections below

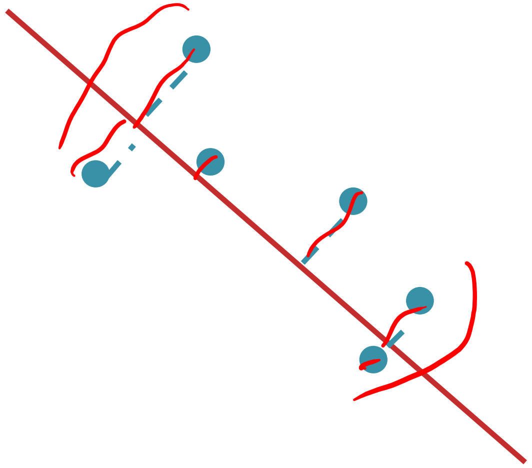
Poll 2: Which maximizes the variance?

Poll 3: Which minimizes the reconstruction error?

Option A

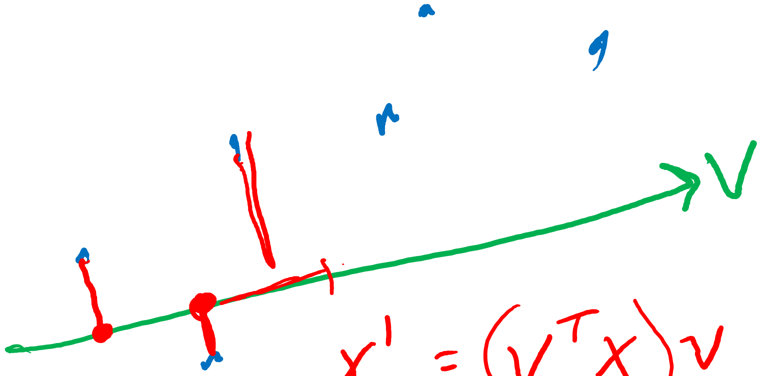


Option B

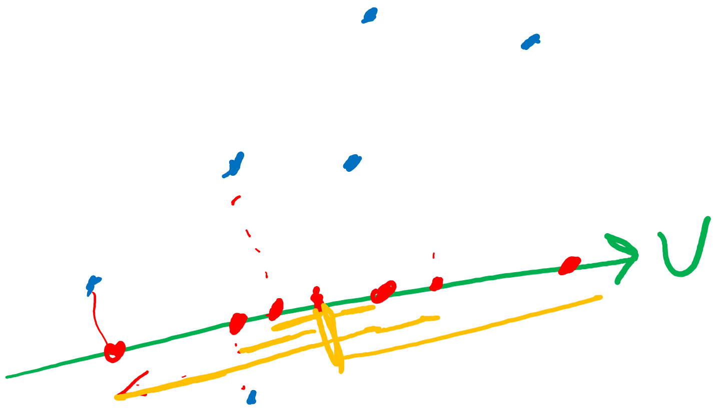


Select “Best” Vector

Reconstruction Error vs Variance of Projection



$x' = (v^T x)v$



Reconstruction error

$$\| x^{(i)} - x'^{(i)} \|_2^2$$

$$v^* = \underset{v}{\operatorname{argmin}} \| x^{(i)} - (v^T x)v \|_2^2$$

s.t. $\|v\|_2 = 1$

Variance of Projection

$$v^* = \underset{v}{\operatorname{argmax}} \sum_{i=1}^{N} \left(v^T x^{(i)}\right)^2$$

s.t. $\|v\|_2 = 1$

PCA

Equivalence of Maximizing Variance and Minimizing Reconstruction Error

Claim: Minimizing the reconstruction error is equivalent to maximizing the variance.

Proof: First, note that:

$$||\mathbf{x}^{(i)} - (\mathbf{v}^T\mathbf{x}^{(i)})\mathbf{v}||^2 = ||\mathbf{x}^{(i)}||^2 - (\mathbf{v}^T\mathbf{x}^{(i)})^2 \tag{1}$$

since $\mathbf{v}^T\mathbf{v} = ||\mathbf{v}||^2 = 1$.

Substituting into the minimization problem, and removing the extraneous terms, we obtain the maximization problem.

$$\mathbf{v}^* = \underset{\mathbf{v}:||\mathbf{v}||^2=1}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{x}^{(i)} - (\mathbf{v}^T\mathbf{x}^{(i)})\mathbf{v}||^2 \tag{2}$$

$$= \underset{\mathbf{v}:||\mathbf{v}||^2=1}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{x}^{(i)}||^2 - (\mathbf{v}^T\mathbf{x}^{(i)})^2 \tag{3}$$

$$= \underset{\mathbf{v}:||\mathbf{v}||^2=1}{\operatorname{argmax}} \frac{1}{N}\sum_{i=1}^{N} (\mathbf{v}^T\mathbf{x}^{(i)})^2 \tag{4}$$

(5)

Sketch of PCA

1. Select "best" $\boldsymbol{V} \in \mathbb{R}^{K \times M}$
2. Project down: $\boldsymbol{z}^{(i)} = \boldsymbol{V}\boldsymbol{x}^{(i)} \quad \forall i$
3. Reconstruct up: $\boldsymbol{x}'^{(i)} = \boldsymbol{V}^T \boldsymbol{z}^{(i)}$

Definition of PCA

1. Select v_1 that best explains data
2. Select next v_j that
   i. Is orthogonal to $v_1, \dots, v_{j-1}$
   ii. Best explains remaining data
3. Repeat 2 until desired amount of data is explained

PCA: the First Principal Component

To find the first principal component, we wish to solve the following constrained optimization problem (variance minimization).

$$\mathbf{v}_1 = \underset{\mathbf{v}:||\mathbf{v}||^2=1}{\text{argmax}}\ \mathbf{v}^T\mathbf{\Sigma}\mathbf{v} \tag{1}$$

So we turn to the method of Lagrange multipliers. The Lagrangian is:

$$\mathcal{L}(\mathbf{v}, \lambda) = \mathbf{v}^T\mathbf{\Sigma}\mathbf{v} - \lambda(\mathbf{v}^T\mathbf{v} - 1) \tag{2}$$

Taking the derivative of the Lagrangian and setting to zero gives:

$$\frac{d}{d\mathbf{v}}\left(\mathbf{v}^T\mathbf{\Sigma}\mathbf{v} - \lambda(\mathbf{v}^T\mathbf{v} - 1)\right) = 0 \tag{3}$$

$$\mathbf{\Sigma}\mathbf{v} - \lambda\mathbf{v} = 0 \tag{4}$$

$$\mathbf{\Sigma}\mathbf{v} = \lambda\mathbf{v} \tag{5}$$

Recall: For a square matrix $\mathbf{A}$, the vector $\mathbf{v}$ is an **eigenvector** iff there exists **eigenvalue** λ such that:

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v} \tag{6}$$

SVD for PCA

SVD matrix factorization

$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T,\ A \in \mathbb{R}^{N\times M}$

$\boldsymbol{U}$: $N \times$ N orthogonal matrix

- Columns of $\boldsymbol{U}$ are *left* singular vectors of $\boldsymbol{X}$
- Columns of $\boldsymbol{U}$ are eigenvectors of $\boldsymbol{X}\boldsymbol{X}^T$

$\boldsymbol{V}$: $M \times M$ orthogonal matrix

- Columns of $\boldsymbol{V}$ are *right* singular vectors of $\boldsymbol{X}$
- Columns of $\boldsymbol{V}$ are eigenvectors of $\boldsymbol{X}^T\boldsymbol{X}$

$\boldsymbol{S}$: $N \times M$ diagonal matrix

- Diagonal entries are singular values of $\boldsymbol{X}$, σ_k
- Each σ_k^2 are the eigenvalues of both $\boldsymbol{X}\boldsymbol{X}^T$ and $\boldsymbol{X}^T\boldsymbol{X}$!!

PCA Algorithm

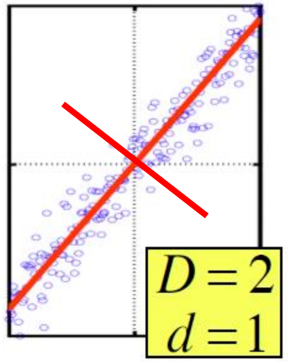
Input: $\boldsymbol{X}, \boldsymbol{X}_{test}, K$

1. Center data (and scale each axis) based on training data → $\boldsymbol{X}, \boldsymbol{X}_{\boldsymbol{test}}$
2. $\boldsymbol{V}$ = eigenvectors($\boldsymbol{X}^T\boldsymbol{X}$)
3. Keep only the top K eigenvectors: $\boldsymbol{V}_K$
4. $\mathbf{Z}_{\text{test}} = \boldsymbol{X}_{test}\boldsymbol{V}_K$

Optionally, use $\boldsymbol{V}_K^T$ to rotate $\mathbf{Z}_{\text{test}}$ back to original subspace $\mathbf{X}'_{\text{test}}$ and uncenter

Principal Component Analysis (PCA)

$(X^T X)v = \lambda v$, so v (the first PC) is the eigenvector of sample correlation/covariance matrix $X^T X$



Sample variance of projection $v^T X^T X\ v = \lambda v^T v = \lambda$

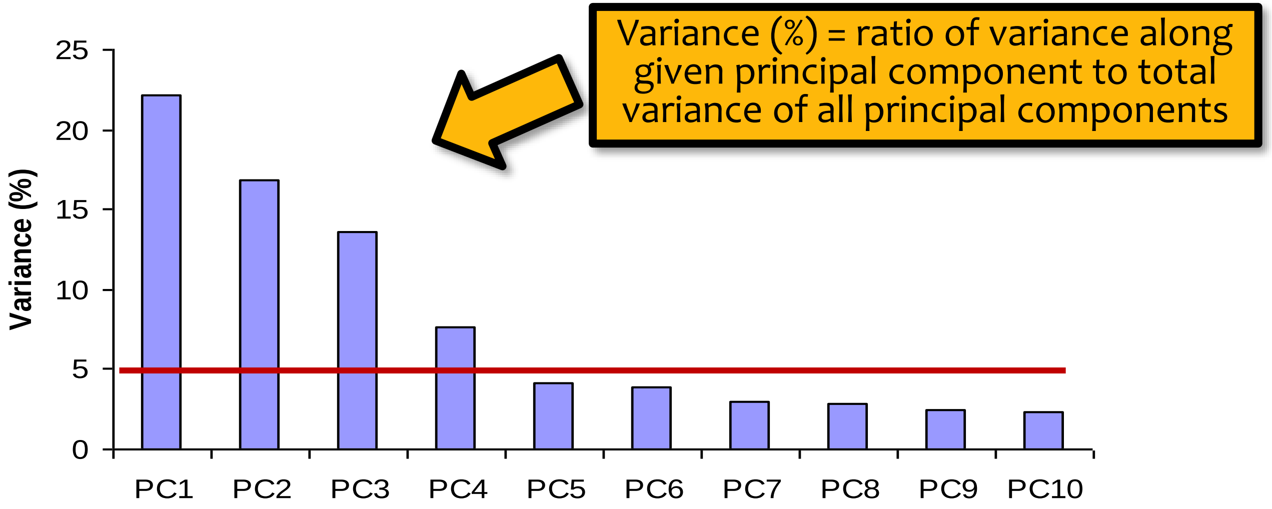
Thus, the eigenvalue λ denotes the amount of variability captured along that dimension (aka amount of energy along that dimension).

Eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots$

- The 1st PC v_1 is the eigenvector of the sample covariance matrix$X^T X$ associated with the largest eigenvalue
- The 2nd PC v_2 is the eigenvector of the sample covariance matrix$X^T X$ associated with the second largest eigenvalue
- And so on …

How Many PCs?

- For M original dimensions, sample covariance matrix is MxM, and has up to M eigenvectors. So M PCs.
- Where does dimensionality reduction come from?
  Can *ignore* the components of lesser significance.



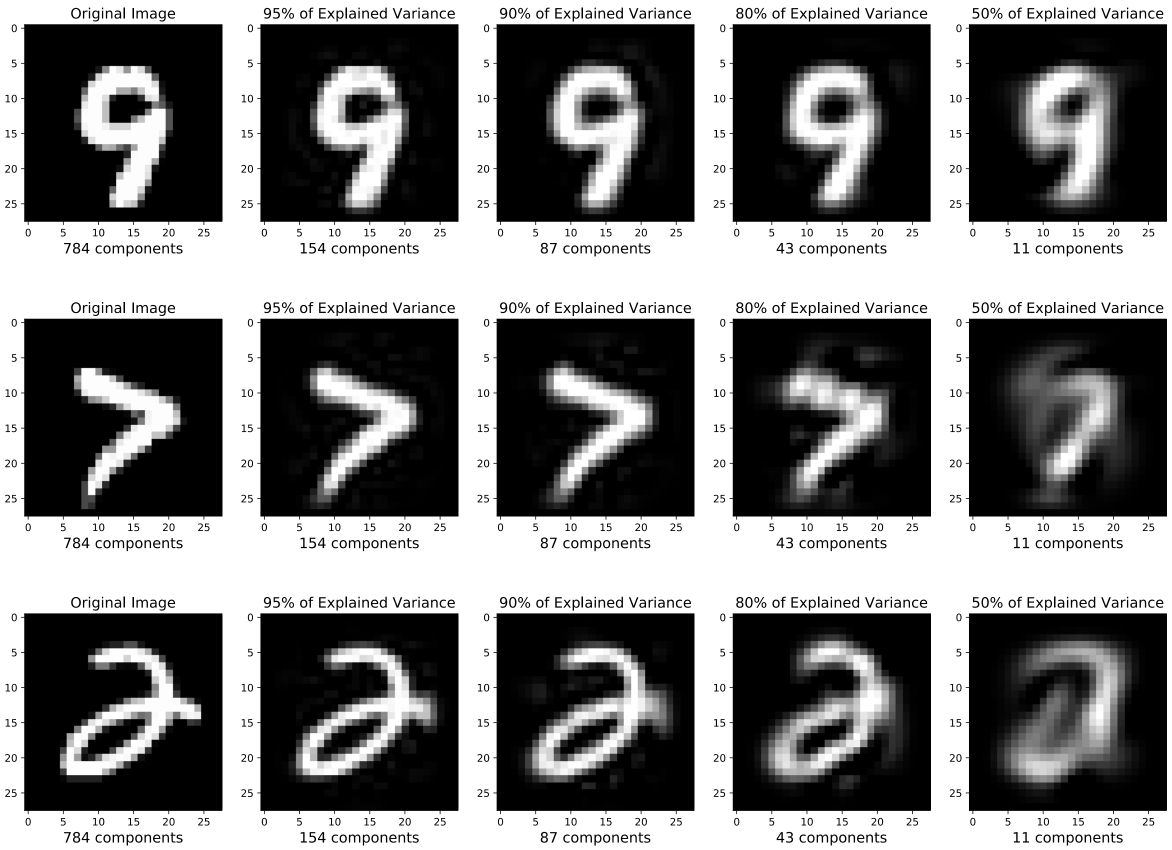
- You do lose some information, but if the eigenvalues are small, you don't lose much
  - M dimensions in original data
  - calculate M eigenvectors and eigenvalues
  - choose only the first D eigenvectors, based on their eigenvalues
  - final data set has only D dimensions

PCA EXAMPLES

Projecting MNIST digits

Task Setting:

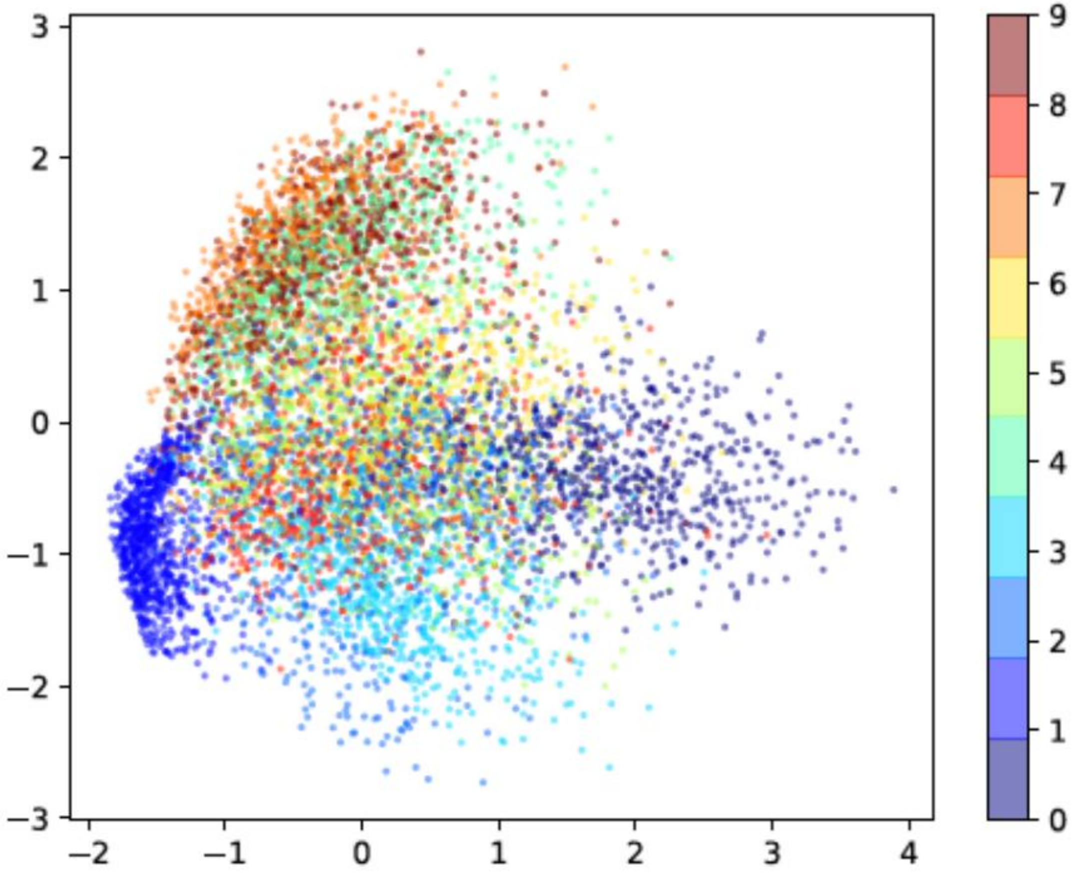
1. Take 28x28 images of digits and project them down to K components
2. Report percent of variance explained for K components
3. Then project back up to 28x28 image to visualize how much information was preserved



Projecting MNIST digits

Task Setting:

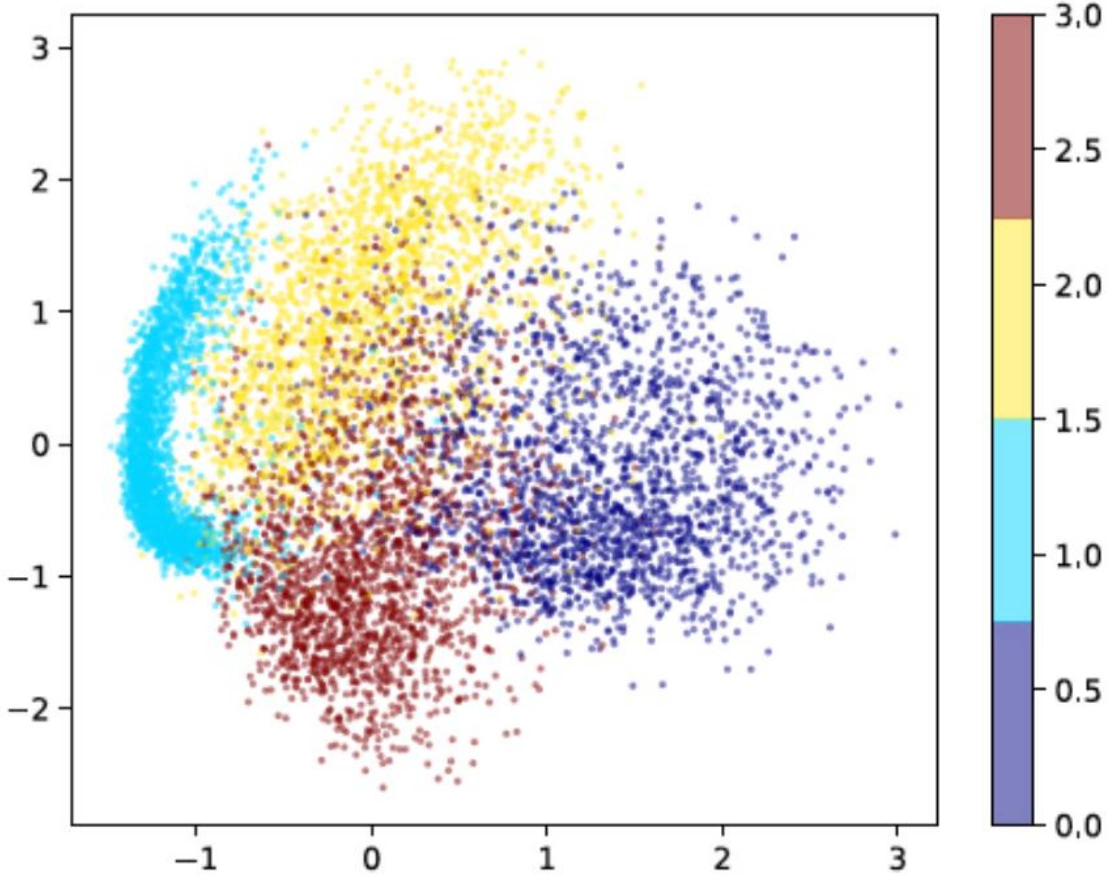
1. Take 28x28 images of digits and project them down to 2 components
2. Plot the 2 dimensional points



Projecting MNIST digits

Task Setting:

1. Take 28x28 images of digits and project them down to 2 components
2. Plot the 2 dimensional points

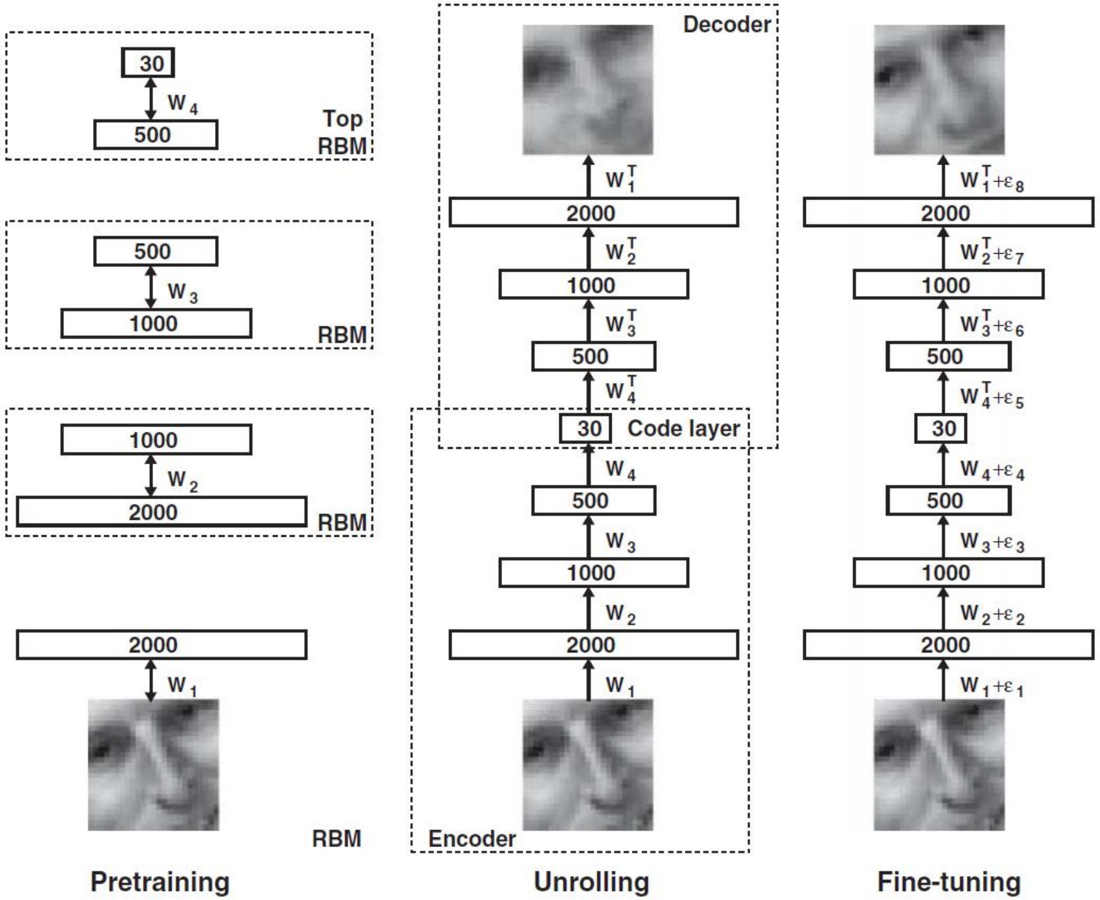


Dimensionality Reduction with Deep Learning

Hinton, Geoffrey E., and Ruslan R. Salakhutdinov.

"Reducing the dimensionality of data with neural networks."

Science 313.5786 (2006): 504-507.



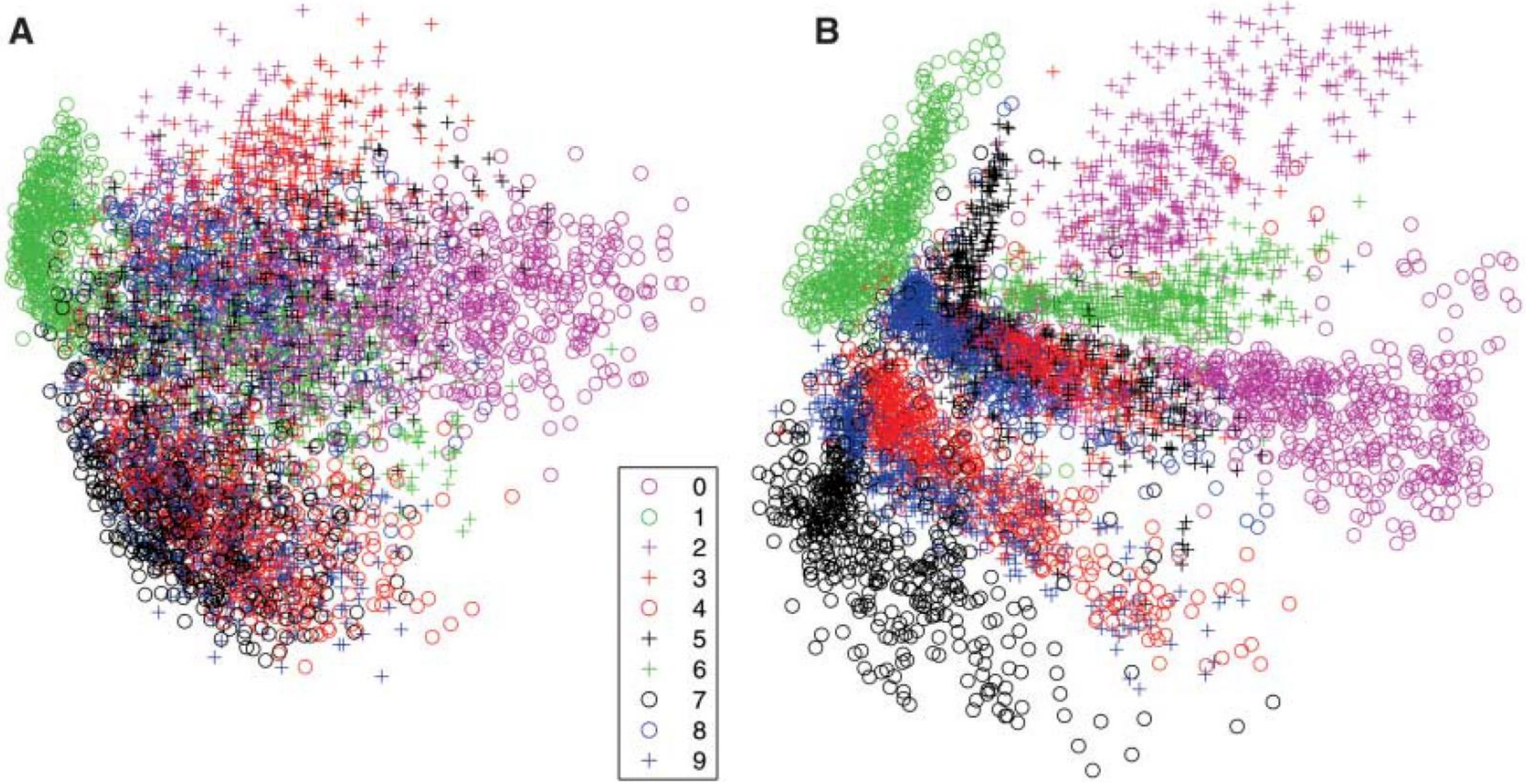
Dimensionality Reduction with Deep Learning

Hinton, Geoffrey E., and Ruslan R. Salakhutdinov.

"Reducing the dimensionality of data with neural networks."

Science 313.5786 (2006): 504-507.

PCA



Neural Network

A Huge Thanks to the Course Team!

Education Associates

Joshmin Ray
joshminr

Fatima Kizilkaya
fjeffrey

Brynn Edmunds
bedmunds

A Huge Thanks to the Course Team! Team

Teaching Assistants

Varsha
vkuppurr

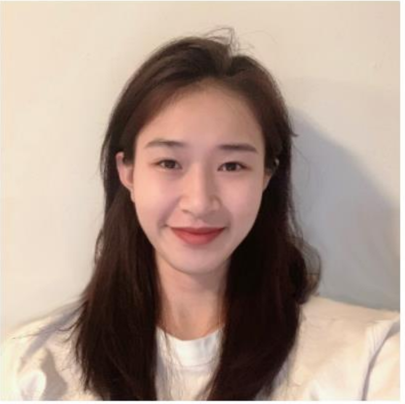
Hanyue
hanyuech

Andrew
andrewh1

Zhaomin
zhaominz

Everett
eknag

Alex
alexs1



Nan
nany

Adrian
akager

A Huge Thanks to the Course Team! Team

Teaching Assistants

Scott
sicongli

Laura
yurongl

Hongyi
hongyiz2

Daniel
seungwom

Young
youngwo1

Zhengyang
zhengyax

Ani
achowdh1

Eric
esliang

A Huge Thanks to the Course Team! Team

Students!!