Machine Learning Principles

Class3 : Sept. 11

Linear Algebra for ML

Instructor: Diana Kim

Today's Lecture

0. Why Linear Algebra?
1. Useful Techniques of Linear Algebra for ML

   - vector space
   - spectral decomposition

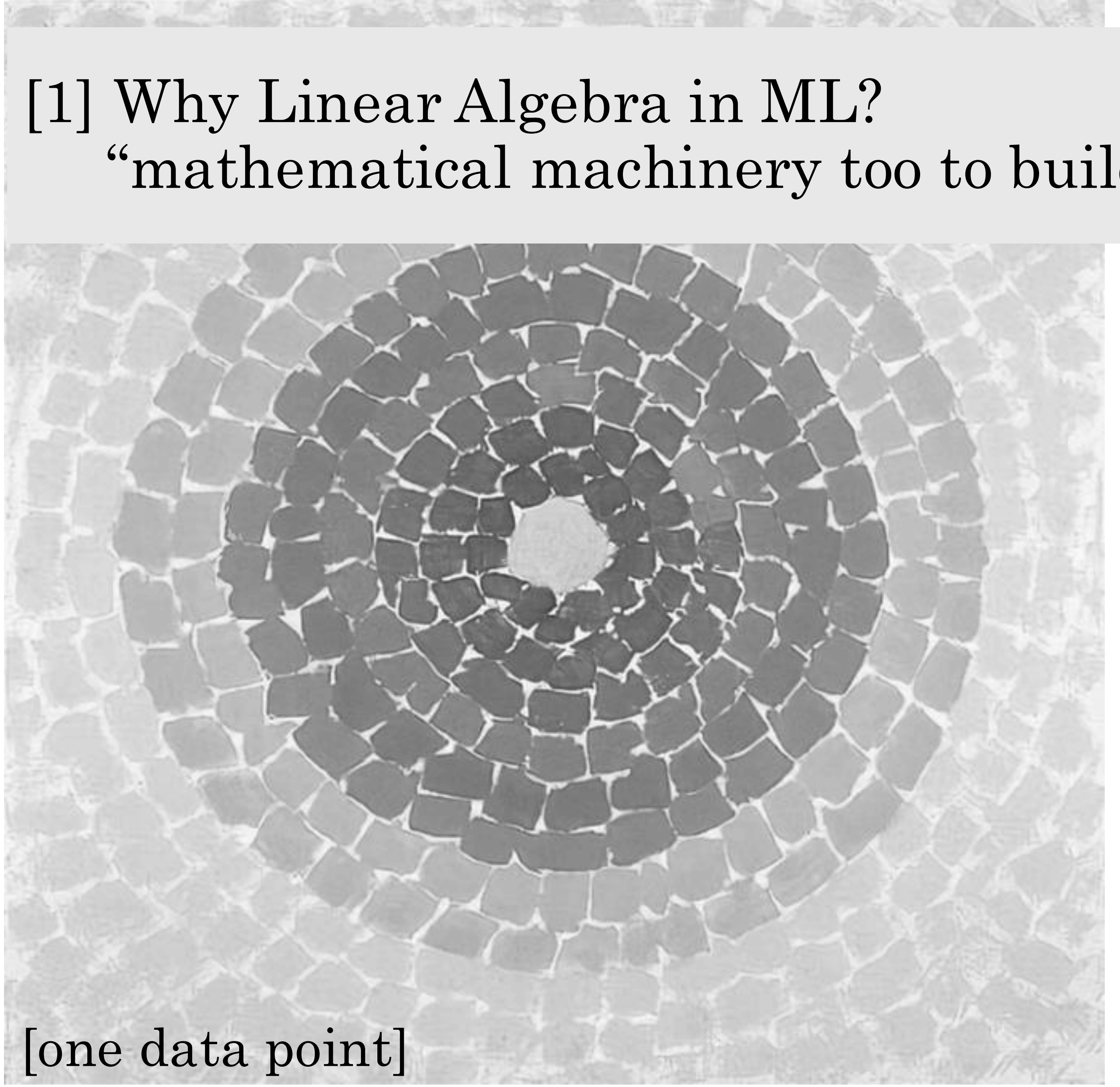
     (1) decorrelation and whitening

     (2) PCA

   - singular vector decomposition
   - pseudo inverse matrix.

[1] Why Linear Algebra in ML?

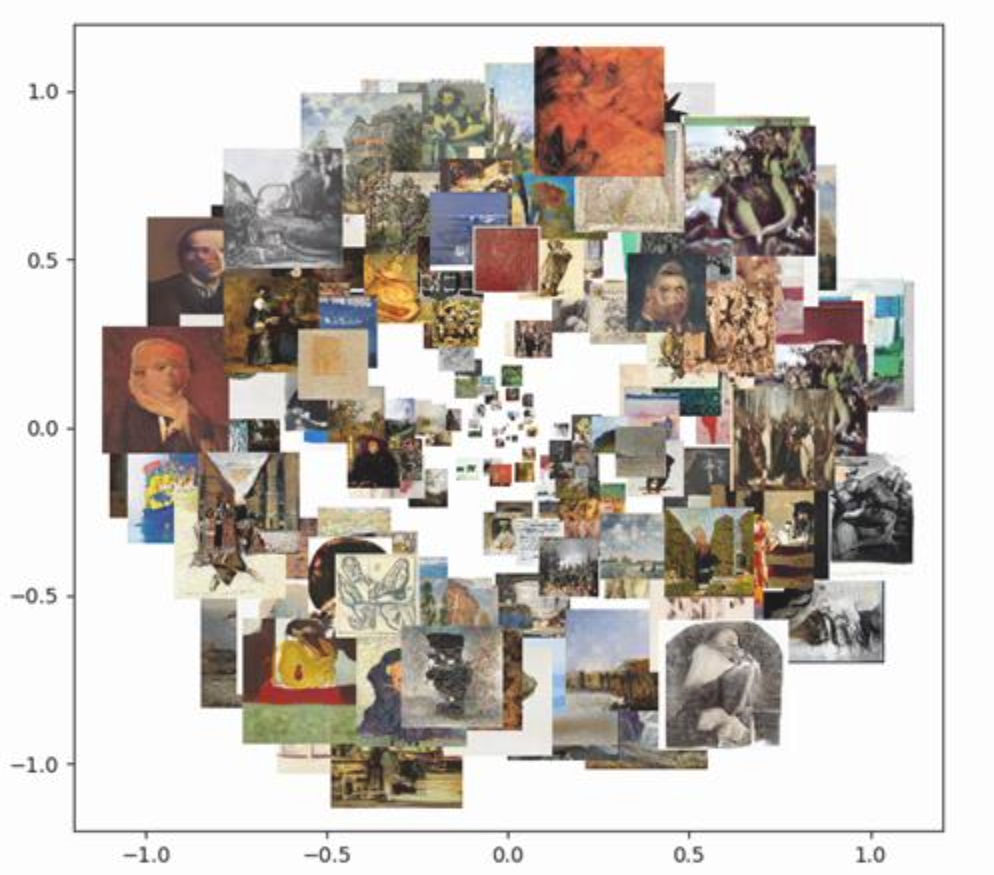
“mathematical machinery too to build ML system with data”



[one data point]

1.66116878e-02	1.43487025e-02	2.32550185e-02	1.19144898e-02
9.63837430e-02	8.31691474e-02	3.06324158e-02	2.04893108e-02
2.02936288e-02	1.33205568e-02	1.15464339e-02	8.20994973e-02
1.77106280e-02	8.52256175e-03	1.17111830e-02	1.45943370e-02
9.04859081e-02	4.89737056e-02	4.16163765e-02	1.04203597e-02
2.58007385e-02	4.69408296e-02	4.55647372e-02	1.24655450e-02
2.41902079e-02	2.42145211e-02	4.81210562e-04	8.90940428e-03
6.85443357e-02	4.78595048e-02	1.18027581e-02	1.17037995e-02
1.90981105e-02	1.00829145e-02	5.23220561e-03	3.84746492e-02
8.81355628e-02	3.36198583e-02	5.35092168e-02	4.87123579e-02
8.99140537e-03	5.39787523e-02	4.79393527e-02	2.99579669e-02
1.71675645e-02	3.76332477e-02	7.88647458e-02	3.79528590e-02
4.51750029e-03	9.38767791e-02	3.61216962e-02	1.98117495e-02
3.94294709e-02	1.14793226e-01	1.48017062e-02	6.40132884e-03
8.16167146e-03	6.94637448e-02	3.43800858e-02	8.04584008e-03
1.55022442e-01	2.59600277e-03	3.20520252e-02	1.00370266e-01
6.82575488e-03	3.21909995e-03	6.07831627e-02	5.22131985e-03
4.10482734e-02	5.29111736e-02	4.37461957e-02	4.37461808e-02
4.10865359e-02	9.59158130e-03	4.68185917e-02	7.04082549e-02
5.19240461e-03	9.47480425e-02	1.72703192e-02	1.32609099e-01
1.84857957e-02	2.34019849e-03	2.21508313e-02	3.19128227e-03
1.03731174e-02	7.90489465e-02	3.40001471e-02	2.08658073e-02
3.63909267e-03	2.93061193e-02	1.79619715e-02	3.92507110e-03
1.22312911e-01	4.27385271e-02	4.02529091e-02	6.87315594e-03
1.79619640e-02	1.44496362e-03	3.47868539e-04	2.03075245e-01
2.45202169e-01	1.26138151e-01	1.07999377e-01	1.46901429e-01
9.70007405e-02	1.03836969e-01	1.09804377e-01	1.04106106e-01
8.70869756e-02	8.81577432e-02	7.79228508e-02	9.59928930e-02
2.06121951e-01	2.38734394e-01	1.37491360e-01	7.11895898e-02
9.10348147e-02	1.08147562e-01	8.93435627e-02	8.45326930e-02
8.54639262e-02	7.94288218e-02	7.84831643e-02	6.98279142e-02
6.67123348e-02	7.02826679e-02	1.02719694e-01	1.04542613e-01
1.12103589e-01	8.02482218e-02	1.26211137e-01	1.22251317e-01
1.18328705e-01	9.65996012e-02	9.47735459e-02	8.21543038e-02
7.41177499e-02	1.03439212e-01	1.14290312e-01	1.15447372e-01
1.28355548e-01	1.06327742e-01	7.30694234e-02	6.20305464e-02
1.06132567e-01	7.94187784e-02	8.86070132e-02	8.47868249e-02
1.07920051e-01	8.36525112e-02	6.55624866e-02	7.80229717e-02
8.64467472e-02	8.55527893e-02	1.10759147e-01	1.32106051e-01
7.44441077e-02	5.27140088e-02	1.08958408e-01	8.06024447e-02
9.61078107e-02	9.56790447e-02	1.04670644e-01	8.01085979e-02
6.94930553e-02	7.93032944e-02	9.49410051e-02	7.71025643e-02
1.05781302e-01	1.46627113e-01	6.05126023e-02	4.13953587e-02
1.23981662e-01	1.08231083e-01	1.34232193e-01	1.18365087e-01
1.09341882e-01	8.85261148e-02	8.09772611e-02	8.30637813e-02
1.13967851e-01	8.66363198e-02	1.12511240e-01	1.40123367e-01
6.65690452e-02	4.43194956e-02	1.57082587e-01	1.04566351e-01
9.03969407e-02	1.14839226e-01	1.21048279e-01	9.68690664e-02
9.22555625e-02	1.10619672e-01	1.21558294e-01	8.79304186e-02
1.04587585e-01	1.42198473e-01	8.80973265e-02	4.42507043e-02

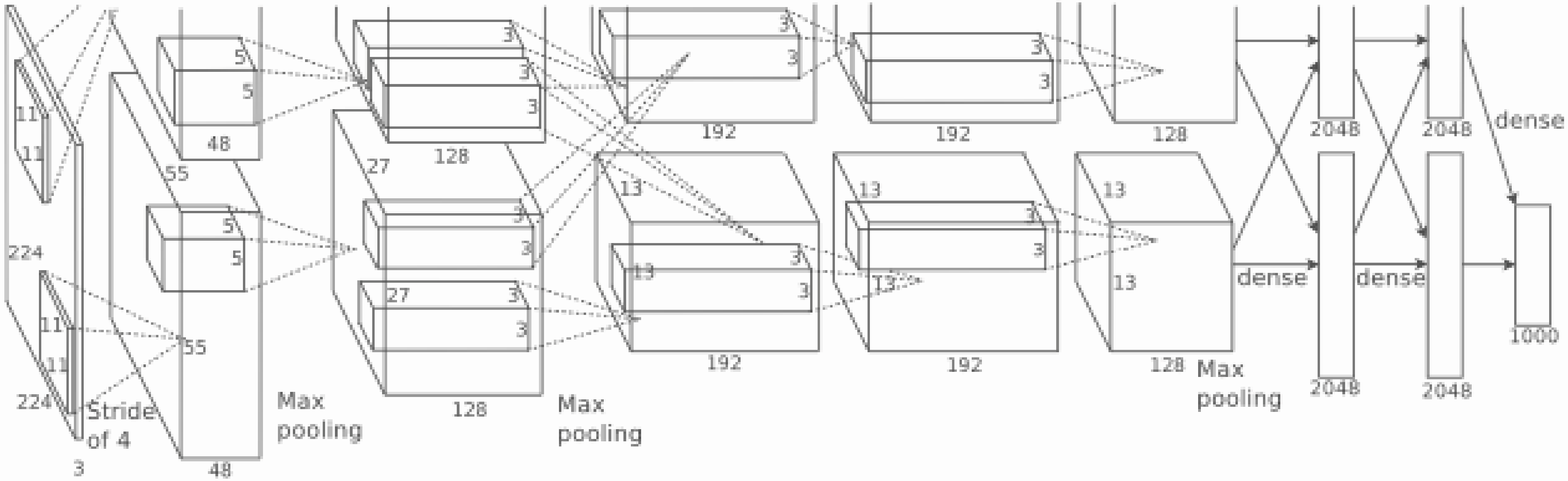
[2] Why Linear Algebra in ML? (data set)



[wikiArt data, ~80,000 images]

- How can you define a random variable to represent the data?

[3] Why Linear Algebra in ML? (transformation)

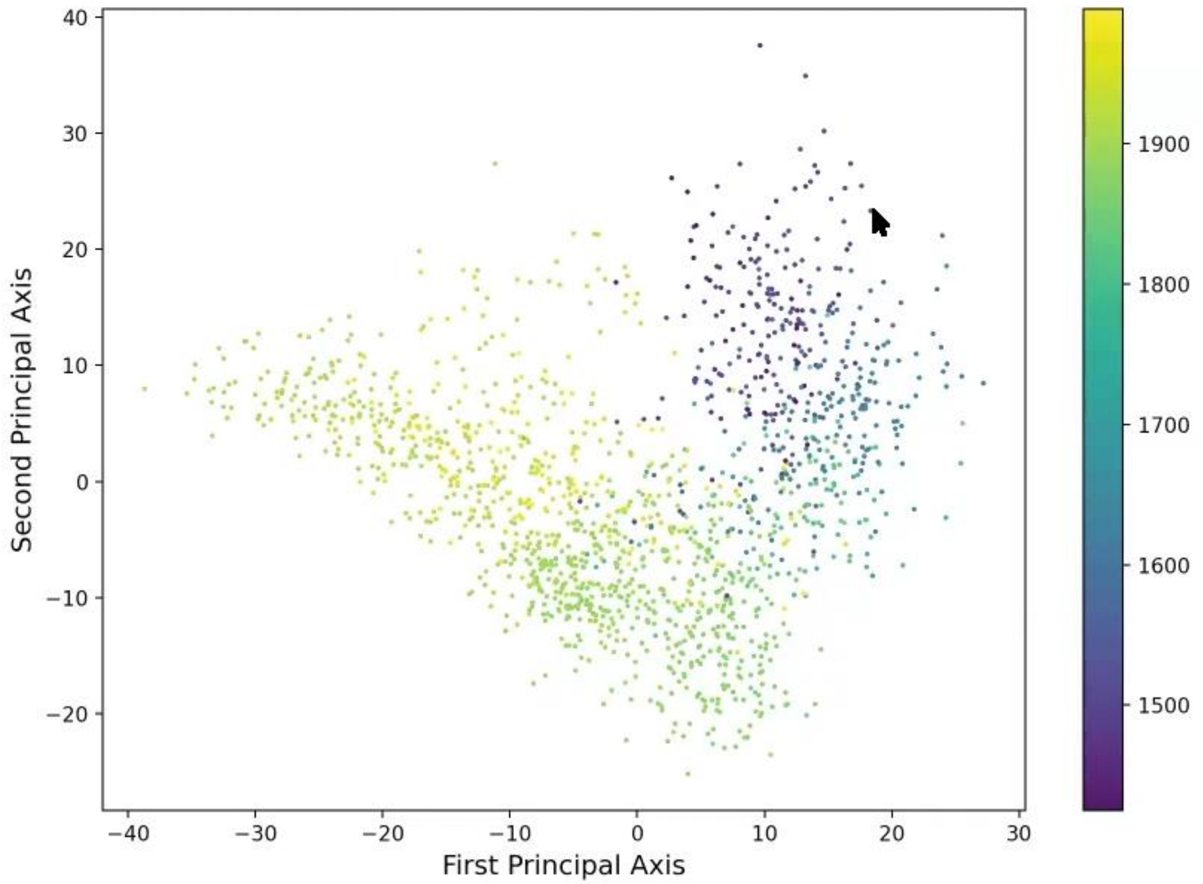


- ML system with data : linear / non-linear transformation.

[4] Why Linear Algebra in ML?

- data representation/transformation in a vector space (high dimensional)
- provides tools to understand the geometric shape of high dimensional data (decorrelation: whitening & data compression: PCA)
- linear regression problem can be translated into solving a linear system.
- Linear Algebra plays a role in every stage of machine learning from (1) data preprocessing to (2) modeling and learning.

Demo: Embedding of the Last Hidden Layer of a Deep-CNN Style Classifier (First Two Axes)



[5] Why Linear Algebra in ML? (Linear Regression)

- Linear Regression Model: $aX_1 + bX_2 + cX_3 = Y$
- a, b, c by solving the linear system below

	X_1	X_2	X_3
#1	1.4	2.7	3.2
#2	2.2	3.5	3.3
#3	3.1	5.2	1.2
#4	1.7	1.0	0.2
#5	4.6	1.1	0.9
#6	2.2	4.3	2.7
#7	1.2	2.3	7.6
#8	0.3	0.2	3.2
#9	2.5	0.5	0.9
#10	1.8	1.9	1.1

$$\times \begin{bmatrix} a \\ b \\ c \end{bmatrix} =$$

	Y
#1	1
#2	1.3
#3	0.8
#4	0.3
#5	1
#6	4
#7	3.6
#8	2.9
#9	2.5
#10	0

- Linear Algebra 101: Vector Space

[1] **Vector Space** & Subspace

A vector space ($\mathbb{R}^N$) is a set of vectors that is closed by linear combinations (vector addition and multiplication by real number).
The vector addition & multiplication by real number follows the eight rules.

1. $x + y = y + x$.
2. $x + (y + z) = (x + y) + z$.
3. There is a unique "zero vector" such that $x + 0 = x$ for all x. [additive identity]
4. For each x there is a unique vector $-x$ such that $x + (-x) = 0$.
5. $1x = x$.
6. $(c_1 c_2)x = c_1(c_2 x)$.
7. $c(x + y) = cx + cy$.
8. $(c_1 + c_2)x = c_1 x + c_2 x$.

[2] **Vector Space** & Subspace

- vector space: closed under linear combination

$\leftrightarrow$ if V *is* a vector space,
then if $v_1 \in V$ and $v_2 \in V$ then $av_1 + bv_2 \in V, \forall\ a$ and $b \in \mathrm{R}$.

Q: simplest vector space?

Q: $V = \{[0], [1]\}$ is this a vector space?

[3] Vector Space & **Subspace**

A subspace of a vector space is a subset that satisfies the requirements for a vector space.

ex] the examples of subspace

(1) any line through (0,0,0) in $\mathbb{R}^3$
(2) any plane through (0,0,0) in $\mathbb{R}^3$
(3) a line does not pass through (0,0,0) in $\mathbb{R}^3$?

- Linear Algebra 101: Matrix’s Vector Space
- column & row & null space

[1] Column & Row Space of a Matrix

The column (row) space contains
all linear combinations of the column (row) of matrix A.

ex] describe the **column / row** space of the matrix A.

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

(1) column space: $\mathbb{R}^2$
(2) row space: a plane spanned by [1 0 1] and [0 1 1]
: $x + y - z = 0$
: a subspace in $\mathbb{R}^3$

[2] **Dimension** of the column/ row space

The dimension of row / columns space = # independent vectors.
The dimension is called **the "rank" of a matrix.**
The dimension of row space and column space of a matrix is the same!

ex] describe the column / row space of the matrix A.

$$\mathrm{A} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

(1) column space: $\mathbb{R}^2$
(2) row space: a plane spanned by [1 0 1] and [0 1 1]
: $x + y - z = 0$
: a subspace in $\mathbb{R}^3$

[3] Null Space of a Matrix

The null space of a matrix A consists of all vectors x s.t $Ax = 0$
Null space is orthogonal to the row space of A.
rank (row) + rank (null) = $\boldsymbol{N}$ when row space in $\mathbb{R}^N$

ex] describe the null space of A. (recitation)

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 5 & 4 & 9 \\ 2 & 4 & 6 \end{bmatrix}$$

- Linear Algebra in ML
  (1) solving a linear system ($Ax = b$)

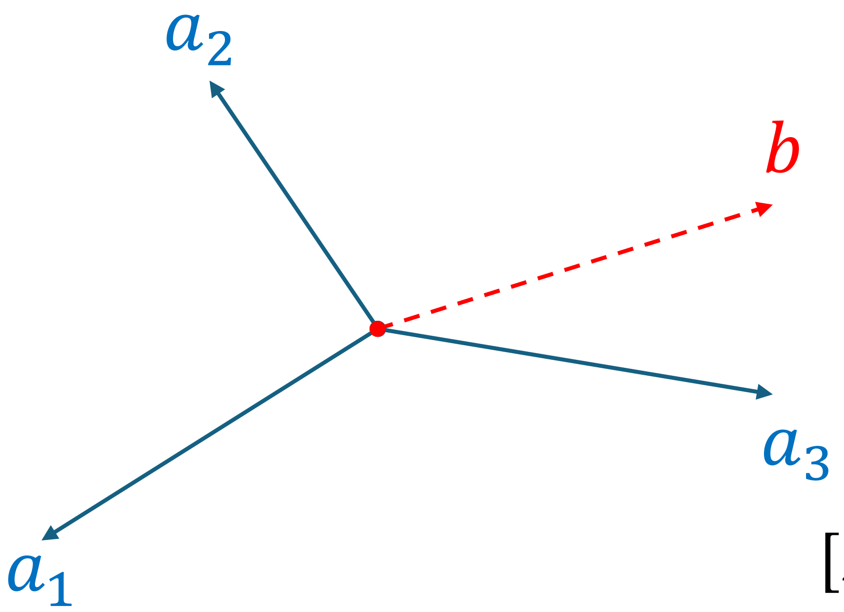
[1] Solution of a Linear System (Existence of Solution)

A solution / multiple solutions of a linear system ($Ax = b$) exist if the vector b is on the column space of A.

ex] $\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$

Q: how can we make this equation to have a solution?
how many solutions exist?

[2] Solution of a Linear System (Existence of Multiple Solutions)



[2d - space]

$$\begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = b,$$ infinite many possible solutions $[x, y, z]$

infinite many soluitons exist if b is on the column space of $[a_1 \quad a_2 \quad a_3]$ and the column vectors $a_1 \quad a_2 \quad a_3$ are not independent.

[3] Solution of a Linear System (no solution)

no solution of a linear system ($Ax = b$) exist if the vector b is on the column space of A.

ex] in linear regression, we aim to solve $[a, b, c]$

	X_1	X_2	X_3
#1	1.4	2.7	3.2
#2	2.2	3.5	3.3
#3	3.1	5.2	1.2
#4	1.7	1.0	0.2
#5	4.6	1.1	0.9
#6	2.2	4.3	2.7
#7	1.2	2.3	7.6
#8	0.3	0.2	3.2
#9	2.5	0.5	0.9
#10	1.8	1.9	1.1
…			

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} =$$

	Y
#1	1
#2	1.3
#3	0.8
#4	0.3
#5	1
#6	4
#7	3.6
#8	2.9
#9	2.5
#10	0
…	

- the vector for Y is likely be outside of the three-dimensional column space in $\mathbb{R}^M$ (M: $\#data$)

- Linear Algebra in ML
  (2) Spectral Decomposition of Symmetric Matrix (Covariance)

*symmetric matrices are decomposed into eigenvectors and eigenvalues matrices.

[1] Spectral Decomposition of Symmetric Matrix

A symmetric matrix (Σ) can be represented
by the matrix of eigenvectors (E) and eigenvalues (Λ).

$$\Sigma = \begin{bmatrix} & | & \\ e_1, e_2, & \ldots & e_n \\ & | & \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & & & \\ 0 & \ldots & \ldots & \lambda_n \end{bmatrix} \cdot \begin{bmatrix} e_1^t \\ e_2^t \\ \ldots \\ e_n^t \end{bmatrix}$$

**convention: descending order

$$\Sigma = \lambda_1 \cdot e_1 e_1^t + \lambda_2 \cdot e_2 e_2^t + \ldots + \lambda_n \cdot e_n e_n^t$$

- e_i: eigenvectors, ${e_i}^t \cdot e_j = 0 \;\; and \; ||e_i|| = 1$ (orthonormal)
- λ_i: eigenvalues (# non zeros = #rank)

[2] Spectral Decomposition of Symmetric Matrix (proof)

proof] $[\Sigma e_1 \quad \Sigma e_2 \quad \dots \quad \Sigma e_n] = [\lambda_1 e_1 \quad \lambda_1 e_2 \quad \dots \quad \lambda_n e_n]$

$$= [e_1 \quad e_2 \quad \dots \quad e_n] \begin{bmatrix} \lambda_1 & \dots & 0 \\ \dots & \dots & 0 \\ 0 & 0 & \lambda_n \end{bmatrix}$$

$$\Sigma E = E\Lambda$$

$$\Sigma = E\Lambda E^t \text{ (Spectral Decomposition)}$$

two eigenvectors for the distinct eigenvalues of a symmetric matrix are orthogonal

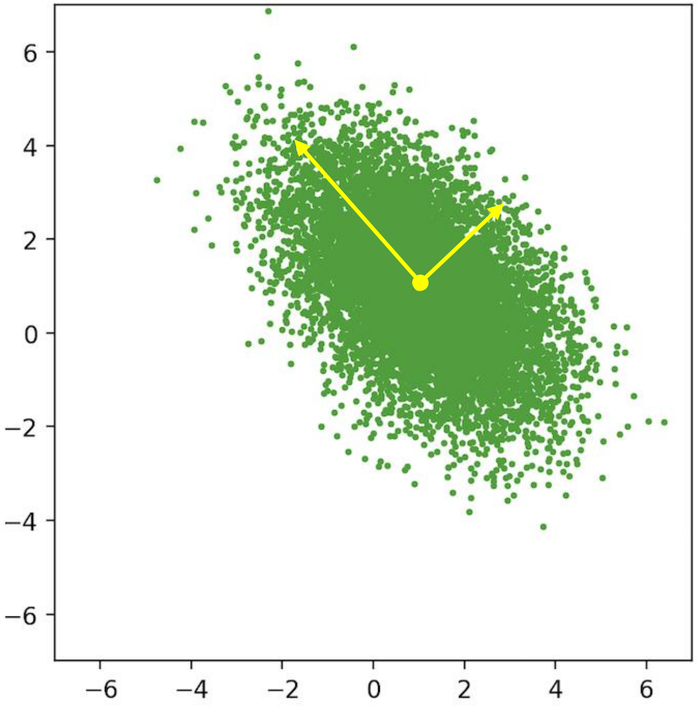
$$\begin{aligned} e_1^t \Sigma e_2 &= e_1^t \Sigma^t e_2 \\ &= (\Sigma e_1)^t e_2 \\ &= \lambda_1 e_1^t e_2 \\ \lambda_2 e_1^t e_2 &= \lambda_1 e_1^t e_2 \end{aligned}$$

[3] Spectral Decomposition of Symmetric Matrix (Geometric Shape)

Covariance matrix shows the extent and direction of data sample dispersion.
The direction and degree can be computed by using spectral decomposition.

$$\Sigma = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \quad \mu = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

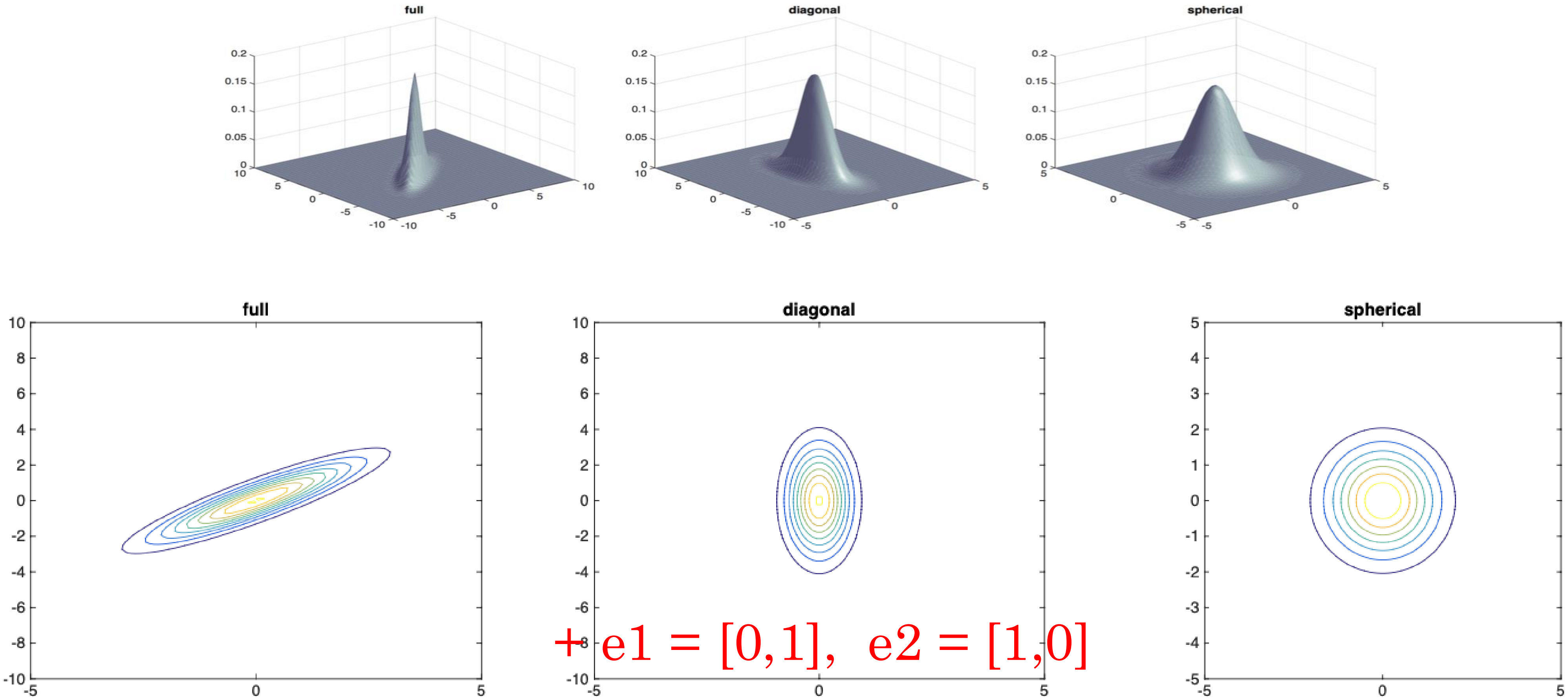
$$\Sigma = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}$$



[2-d Gaussian Data Samples]

[4] Spectral Decomposition of Symmetric Matrix (Geometric Shape)

From Murphy Figure 3.5 and 3.6. “An Introduction”

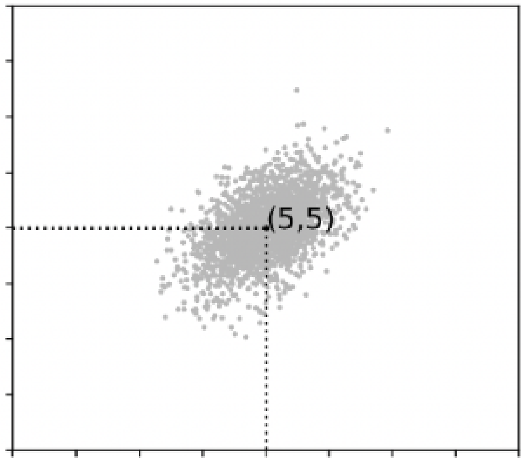


+e1 = [0,1], e2 = [1,0]

- can we guess the eigenvector & eigenvalues from the gaussian contours?

[5] Spectral Decomposition of Symmetric Matrix (Geometric Shape)

[Quiz Example]

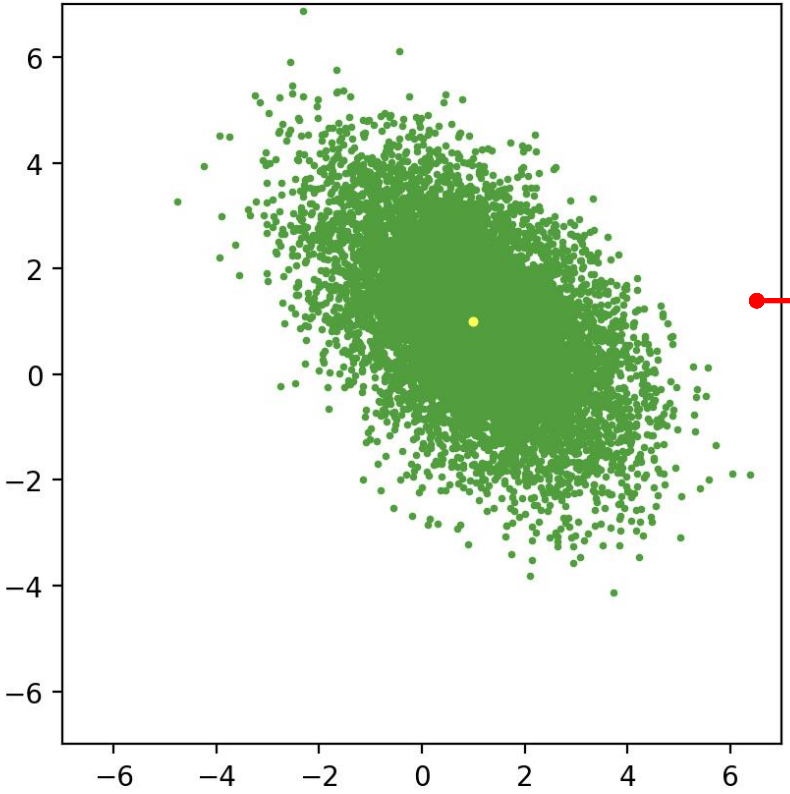


$$e1 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \qquad e2 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$$

$$E[X] = \begin{bmatrix} 5 \\ 5 \end{bmatrix} \quad COV[X,X] = \begin{bmatrix} \quad & \quad \\ \quad & \quad \end{bmatrix} \cdot \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \quad & \quad \\ \quad & \quad \end{bmatrix}$$

- Linear Algebra in ML
  (2) Spectral Decomposition of Symmetric Matrix (Covariance)
  - -whitening (decorrelation)
  - -PCA (data dimensionality reduction)

ex] how could we whiten the data samples (decorrelation)?

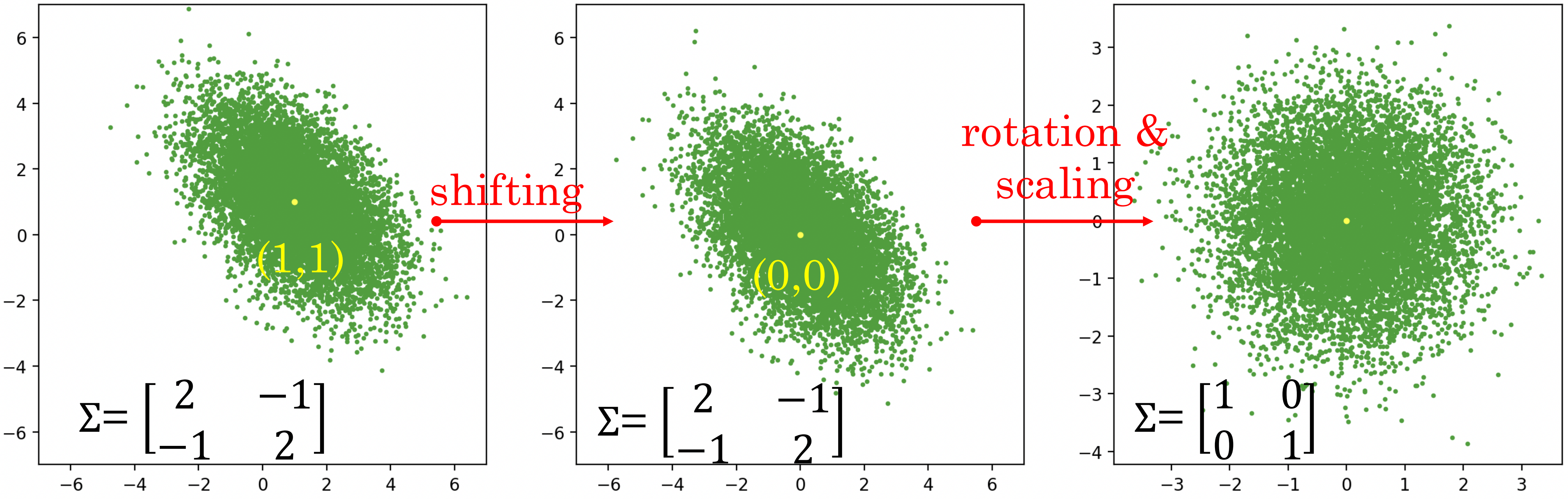


what transform?

$$\Sigma = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, \mu = [1,1]$$

[2] Spectral Decomposition of Symmetric Matrix (Whitening: mean=0 & COV=I)

ex] how could we whiten the data samples?



[3] Spectral Decomposition of Symmetric Matrix (Whitening)

Suppose random vector X's E[X] = μ and COV(X) = $\Sigma = E\Lambda E^t$
Define Affine transformation W = $AX + b$ where $E[W] = 0$ and COV(W) = I.

sol)

$$E[W] = E[Ax + b] = 0$$

$$b = -A\mu_x$$

$$W = A(x - \mu_x)$$

- [shifting]

$$\begin{aligned} COV(W) &= E[WW^t] = I \\ &= E[A(x-\mu_x)(x-\mu_x)^t A^t] \\ &= AE[(x-\mu_x)(x-\mu_x)^t]A^t \\ &= A \cdot COV(X)A^t = I \end{aligned}$$

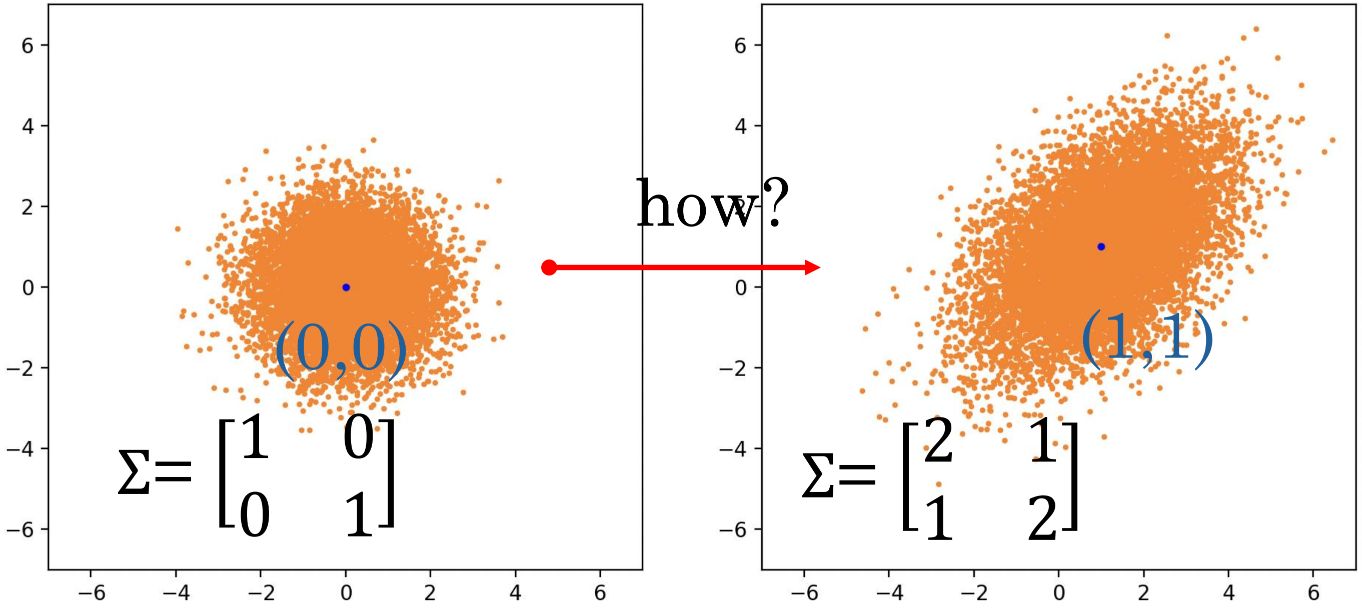
$$A = \Lambda^{-1/2}E^t$$

- [rotation & scaling]

[4] Spectral Decomposition of Symmetric Matrix (Gaussian Generation, **recitation**)

ex] how can we generate the Gaussian random vector $X \sim N(\mu, \Sigma)$ from $W \sim N\,(0,\, I)$?

$$\Sigma^* = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \qquad \mu = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$



$A = A\,W + b$, A? b?

This problem will be covered in detail during the recitation.

Q: How do eigenvectors determine the direction of maximum / minimum variance of data?

1. we know spectral decomposition (eigenvectors and eigenvalues)

$$\Sigma = \begin{bmatrix} | \\ e_1, e_2, \dots e_n \\ | \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & & & \\ 0 & \dots & \dots & \lambda_n \end{bmatrix} \cdot \begin{bmatrix} e_1^t \\ e_2^t \\ \dots \\ e_n^t \end{bmatrix}$$

2. for $n = 2$, data variance at a certain direction can be computed by

$$\mathrm{E}[P^t \cdot X \cdot X^t \cdot P] = P^t E[X \cdot X^t]\, P = P^t COV(X)\, P \text{ where } P = a\overrightarrow{e_1} + b\overrightarrow{e_2}$$

and $||P|| = 1$ $(a^2 + b^2 = 1)$

$$= a^2\lambda_1 + b^2\lambda_2$$

3. $\lambda_{min} \le a^2\lambda_1 + b^2\lambda_2 \le \lambda_{max}$

*Spectral Decomposition of Symmetric Matrix (Singular)

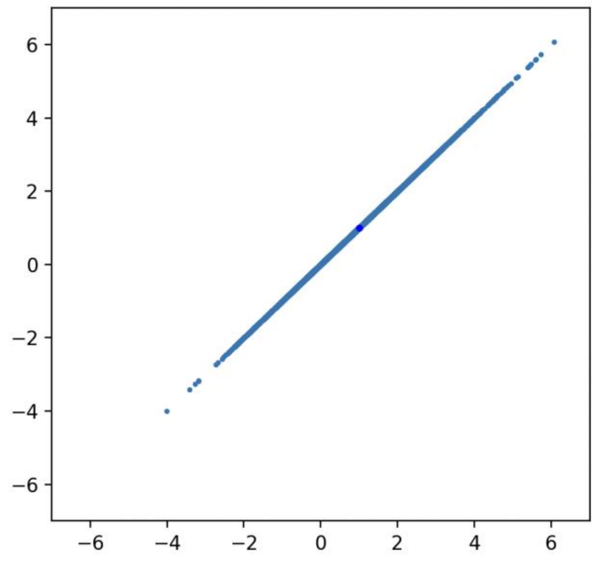
ex] how would the shape of a 2d scatter plot look like when the covariance matrix singular?

$$\Sigma = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} 3 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \qquad \mu = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

* Spectral Decomposition of Symmetric Matrix (Singular)

ex] how would the shape of a 2d scatter plot look like when the covariance matrix singular?

$$\Sigma = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} 3 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \qquad \mu = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$



- Principal Component Analysis (PCA)

  representation of data manifolds in a high-dimensional space

  into a subspace (“approximation”)

[**] previous slide: Spectral Decomposition of Symmetric Matrix

Suppose random vector X's E[X] = μ and COV(X) = $\Sigma = E\Lambda E^t$
Define Affine transformation W = $AX + b$ where $E[W] = 0$ and COV(W) = I.

sol)

$$E[W] = E[Ax + b] = 0$$

$$b = -A\mu_x$$

$$W = A(x - \mu_x)$$

- [shifting]

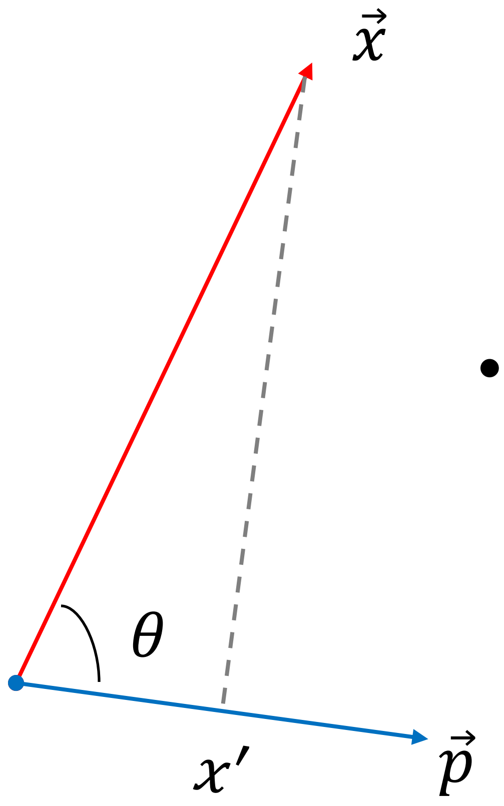
$$\mathrm{W} = \Lambda^{-\frac{1}{2}} E^t (X - \mu_x)$$

$$\begin{aligned} COV(W) &= E[WW^t] = I \\ &= E[A(x-\mu_x)(x-\mu_x)^t A^t] \\ &= AE[(x-\mu_x)(x-\mu_x)^t]A^t \\ &= A \cdot COV(X)A^t = I \end{aligned}$$

$$A = \Lambda^{-1/2} E^t$$

- [rotation & scaling]

**projection ** (a vector to a scalar/ a vector)



- $\vec{x}$ projection to $\vec{p} = \vec{p}^t \cdot \vec{x}$ (to a scalar)

$$= \frac{||\vec{p}|| \cdot ||\vec{x}|| cos\,\theta}{||\vec{p}||^2} \vec{p}$$

[1] Principal Component Analysis (Data Compression)

PCA is to compute **data approximation** $\widetilde{\boldsymbol{x_n}}$ that minimize

$$J = \frac{1}{N}\sum_{n=1}^{N} ||x_n - \widetilde{x_n}||^2$$

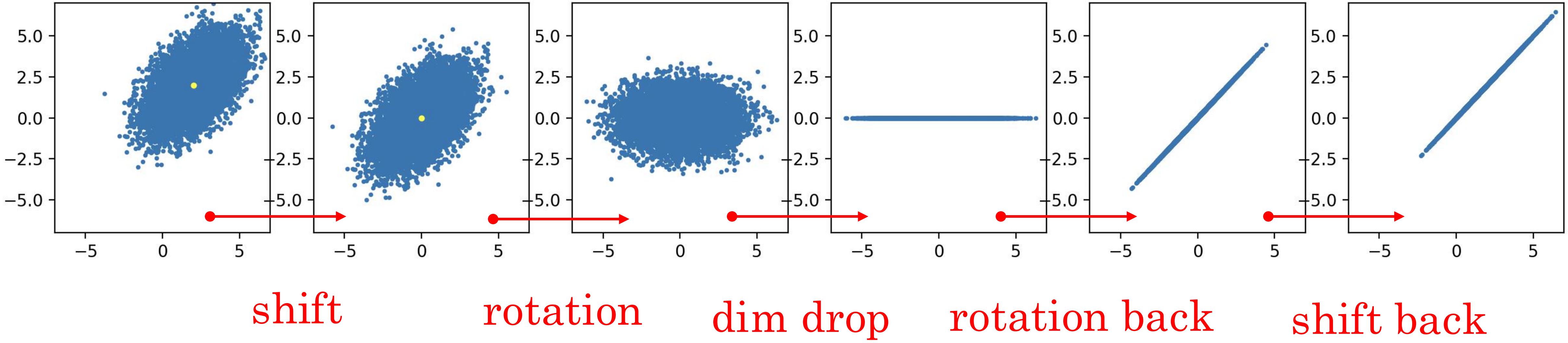
$$\widetilde{X_n} = \underbrace{\bar{x} + U_M \underbrace{U_M^t \underbrace{(x_n - \bar{x})}_{\textbf{shift}\text{?}}}_{\substack{\text{projection \& make variance zero} \\ \text{for the small eigenvalue direction}}}}_{\text{linear combination and shift back}}$$

Q: what if $M = N$?

[2] Principal Component Analysis (Data Compression)

$$\widetilde{X_n} = \bar{x} + U_M U_M^t (x_n - \bar{x})$$

$$\Sigma^* = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \qquad \mu = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$



[PCA process: still in 2d but 1d subspace]

[3] Principal Component Analysis (Data Compression)

[Different PCA Approximation for M = 1, M = 10, M =50, M = 250]

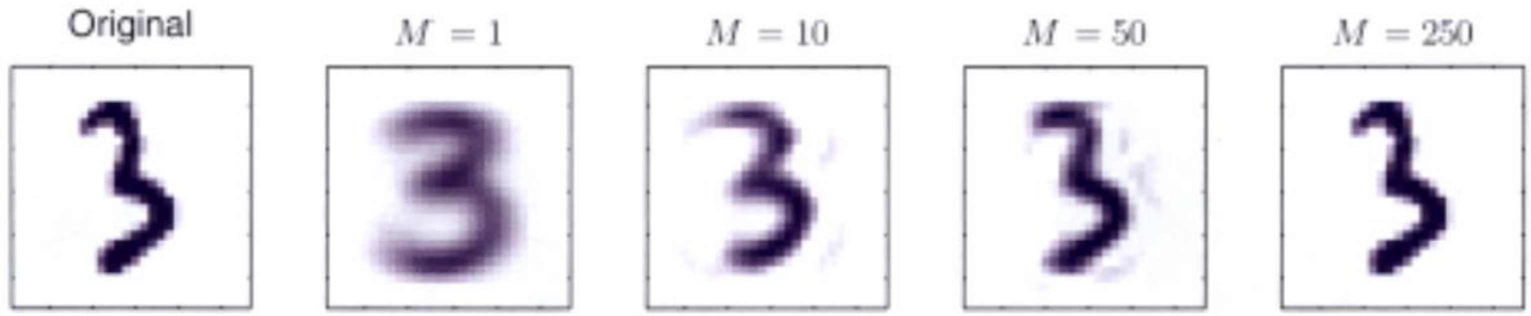


Figure 12.5 An original example from the off-line digits data set together with its PCA reconstructions obtained by retaining M principal components for various values of M. As M increases the reconstruction becomes more accurate and would become perfect when $M = D = 28 \times 28 = 784$.

From Bishop Chap. 12

$$\widetilde{X_n} = \bar{x} + U_M U_M^t (x_n - \bar{x})$$

[Visualization of Eigenvectors]

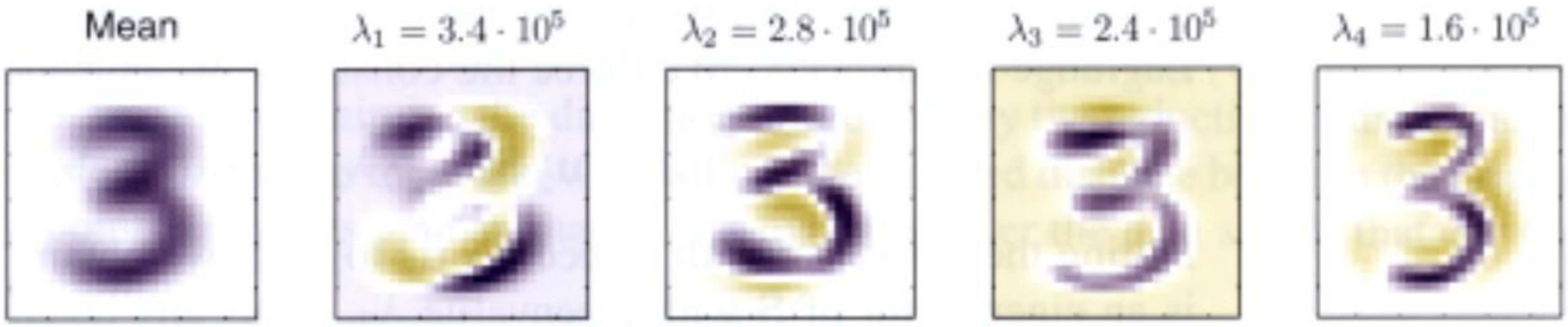
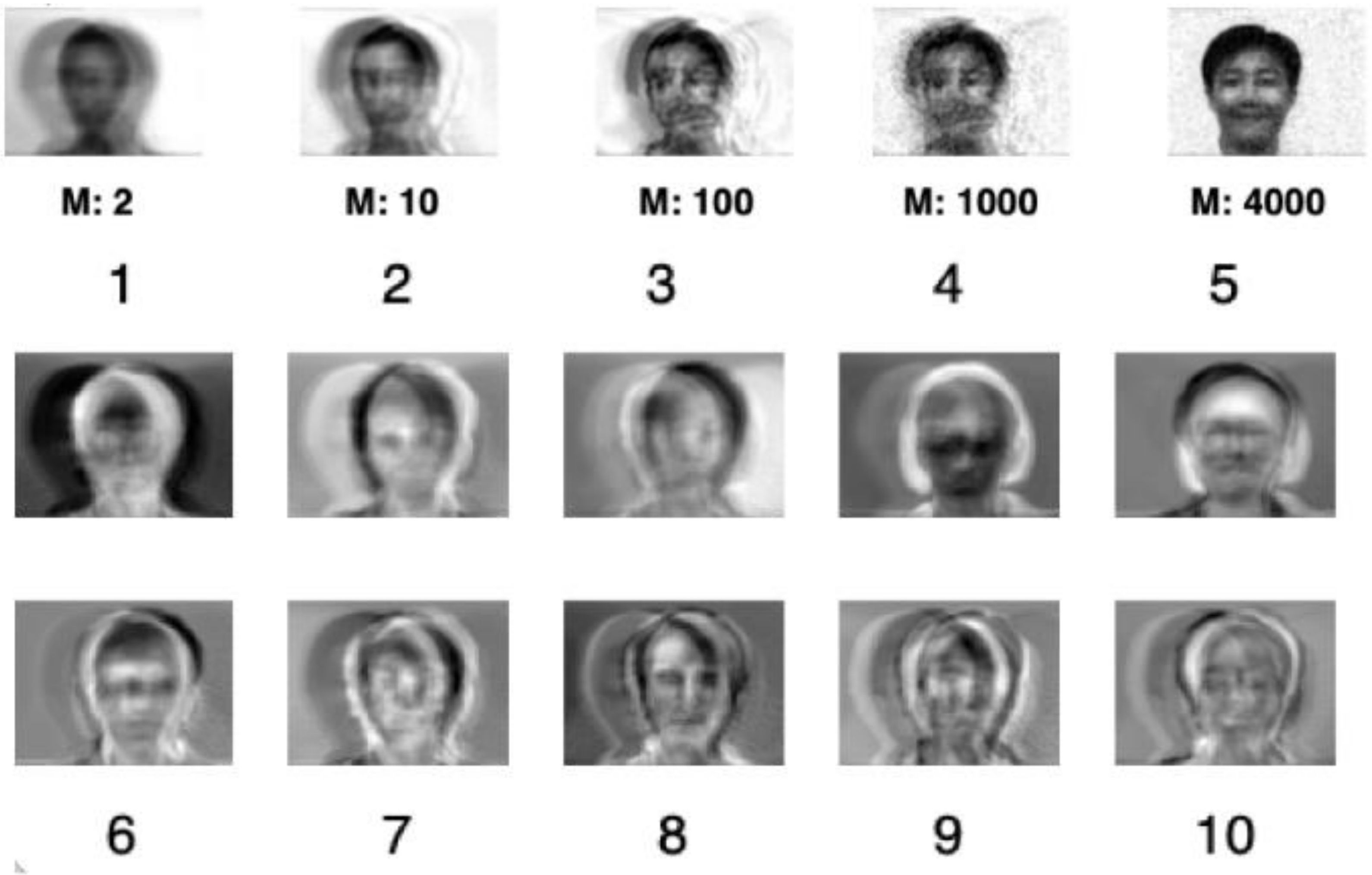


Figure 12.3 The mean vector $\overline{\mathbf{x}}$ along with the first four PCA eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_4$ for the off-line digits data set, together with the corresponding eigenvalues.

$$\widetilde{X_n} = \bar{x} + U_M U_M^t (x_n - \bar{x})$$

**Principal Component Analysis (eigenface in assignment#2)



[5] Principal Component Analysis (Data Visualization in 2D or 3D)

- "two ways to use PCA"
  - approximation
  - dimensionality reduction

$$\widetilde{X_n} = \bar{x} + U_M \boxed{U_M^t (x_n - \bar{x})}$$

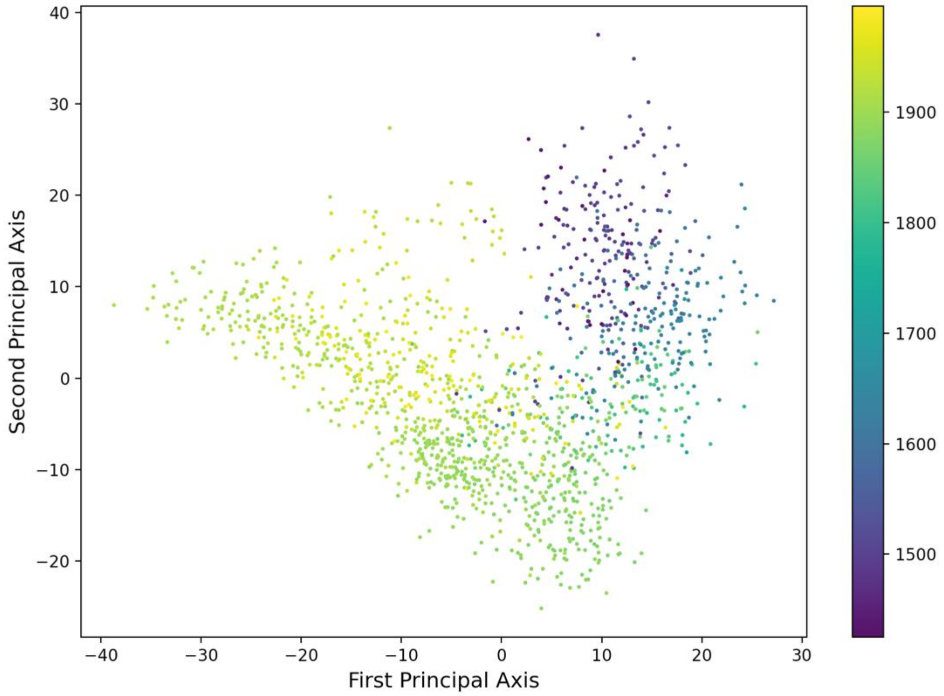
stop in projection!

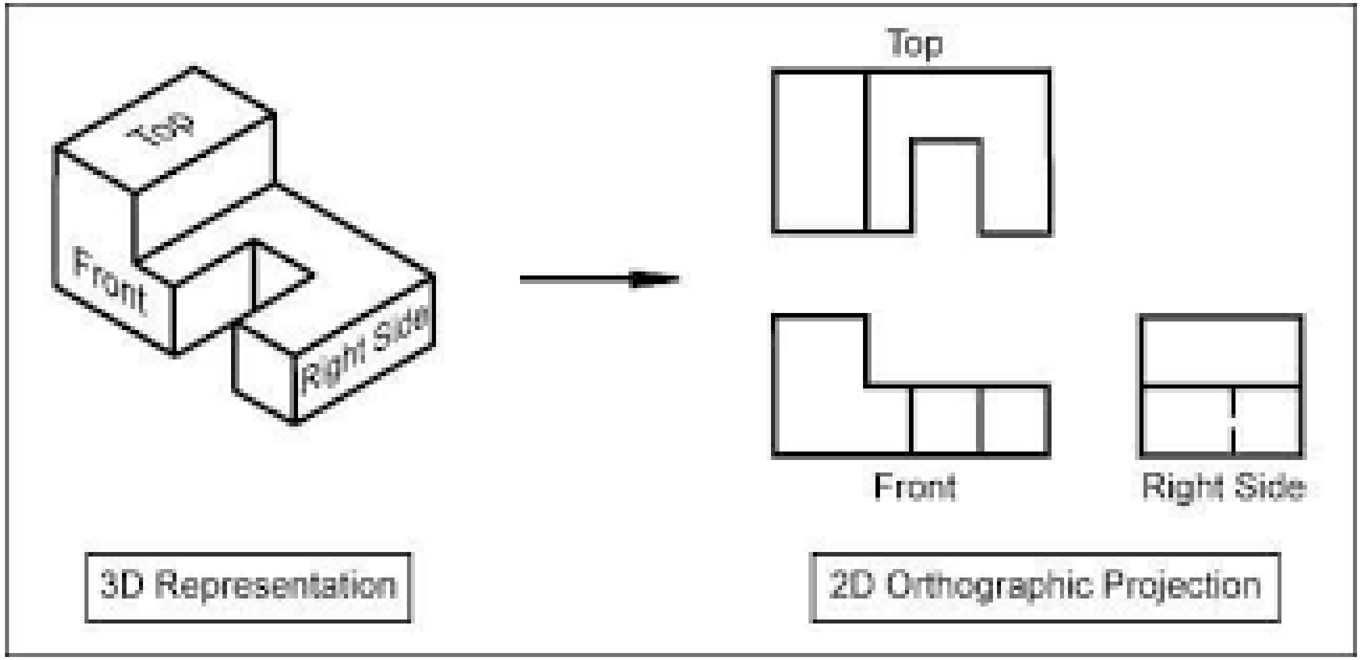
$$\tilde{X}_n = U_M^t (x_n - \bar{x})$$

the high dimensional data points (for example, 2048-d) can be represented by 2d/3d data.

[6] Principal Component Analysis (Data Visualization Example)

[visualization (1) (the projection of high dimensional data to 3D or 2D)]





The last hidden layer embedding (2048D) of a Deep-CNN Style Classifier is projected to the top principal axes (the eigenvectors corresponding to the first and second largest eigenvalues). The samples are color-coded by year of made.

- Visualization (2) (projection of high dimensional data to 3D or 2D)

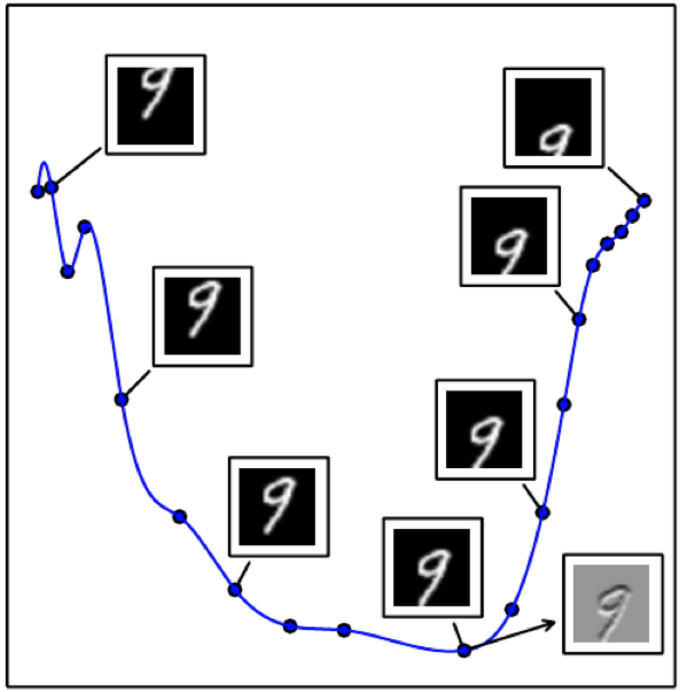


Figure 14. 6 from Deep Learning by Ian Goodfellow

One dimensional manifold that traces out a curved path for vertical shift of digit “9”. The manifold in the high dimensional space is projected into 2D.

- Singular Vector Decomposition of a Rectangular Matrix

[1] Singular Vector Decomposition of a **Rectangular** Matrix (A whose $n > m$)

$$A(n \times m) = \begin{bmatrix} & | & \\ u_1, u_2, & ... u_n \\ & | & \end{bmatrix} \cdot \begin{bmatrix} \sqrt{\lambda_1} & 0 & ... & 0... \\ 0 & \sqrt{\lambda_2} & ... & 0...0 \\ \vdots & & & \\ 0 & ... & ... & \sqrt{\lambda_m} \\ \vdots & & & \\ 0 & ... & ... & 0 \\ 0 & ... & ... & 0 \end{bmatrix} \cdot \begin{bmatrix} v_1^t \\ v_2^t \\ ... \\ v_m^t \end{bmatrix}$$

- u_i: eigenvectors of $AA^t = \mathrm{E}\Lambda E^t$ (n x n)
- v_i: eigenvectors of $A^tA = \mathrm{V}\Lambda V^t$ (m x m)
- λ_i: eigenvalues of both AA^t and A^tA

[2] Singular Vector Decomposition of a Rectangular Matrix (A^t)

$$A^t(m \times n) = \begin{bmatrix} & | & \\ v_1, v_2 & , & ...v_m \\ & | & \end{bmatrix} \cdot \begin{bmatrix} \sqrt{\lambda_1} & 0 & ... & 0... & 0 & 0 \\ 0 & \sqrt{\lambda_2} & ... & 0... & 0 & 0 \\ \vdots & & & & & \\ 0 & ... & ... & \sqrt{\lambda_m} & ... & 0 \end{bmatrix} \cdot \begin{bmatrix} u_1^t \\ u_2^t \\ ... \\ u_n^t \end{bmatrix}$$

- u_i: eigenvectors of AA^t
- v_i: eigenvectors of A^tA
- λ_i: eigenvalues of both AA^t and A^tA

[3] Singular Vector Decomposition of a Rectangular Matrix (AA^t and A^tA)

when A is a ($n \times m, n > m$) matrix, both AA^t and A^tA are symmetric.

$$AA^t = \begin{bmatrix} \vert & \\ u_1, u_2, \dots u_n \\ \vert & \end{bmatrix} \cdot \underbrace{\begin{bmatrix} \lambda_1 & 0 & \dots & 0 & \dots & 0 & 0 \\ 0 & \lambda_2 & \dots & 0 & \dots & 0 & 0 \\ \vdots & 0 & \dots & \lambda_m & & & \\ 0 & \dots & 0 & \dots & \dots & 0 & 0 \\ 0 & \dots & 0 & \dots & \dots & 0 & 0 \end{bmatrix}}_{[n \times n]} \cdot \begin{bmatrix} u_1^t \\ u_2^t \\ \dots \\ u_n^t \end{bmatrix}$$

filling up (n-m) zeros!

$$A^tA = \begin{bmatrix} \vert & \\ v_1, v_2, \dots v_m \\ \vert & \end{bmatrix} \cdot \underbrace{\begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & & & \\ 0 & \dots & \dots & \lambda_m \end{bmatrix}}_{[\mathrm{m} \times m]} \cdot \begin{bmatrix} v_1^t \\ v_2^t \\ \dots \\ v_m^t \end{bmatrix}$$

[4] Singular Vector Decomposition of a Rectangular Matrix (proof)

Proof]

$$A^t A = V \Lambda V^t$$

$$A^t A V = V \Lambda$$

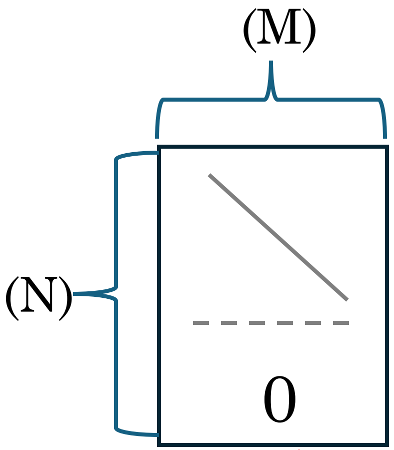
$$A A^t A V = A V \Lambda$$

- [Av is eigenvector of AA^t]

$$V^t A^t A V = \Lambda$$

$$(AV)^t (AV) = \Lambda$$

- [length of $||Av||^2 = \lambda$]



- [U is orthonormal and eigenvector matrix]
  - [Λ' $(n \times m)$]

$$AV = U \Lambda'^{1/2}$$

$$A = U \Lambda'^{1/2} V^t$$

- Pseudo-Inverse (using SVD)

  Generalization of the notion of inverse matrix.

[0] When do we need Pseudo-Inverse?

ex] $\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$

- no solution example

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 5 \end{bmatrix}$$

- projection to the column space
- finding an approximated solution

$$\begin{bmatrix} 5 & 10 \\ 10 & 20 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 11 \\ 22 \end{bmatrix}$$

- now we could find approximated solution
- rank:1 but two columns so redundant, ∞ sols
- no invertible (but we could find one of the ∞ solutions)

[1] Singular Vector Decomposition of a Rectangular Matrix (Pseudo-Inverse)

$$D = \underset{U}{\begin{bmatrix} \mid & \\ u_1, u_2, \ldots u_n \\ \mid & \end{bmatrix}} \cdot \begin{bmatrix} \sqrt{\lambda_1} & 0 & \ldots & 0 \\ 0 & \sqrt{\lambda_2} & \ldots & 0 \\ \vdots & & & \\ 0 & & \sqrt{\lambda_{m-1}} & 0 \\ 0 & \ldots & \ldots & 0 \\ 0 & \ldots & \ldots & 0 \\ 0 & \ldots & \ldots & 0 \end{bmatrix} \cdot \underset{V^t}{\begin{bmatrix} v_1^t \\ v_2^t \\ \ldots \\ v_m^t \end{bmatrix}}$$

$$D^\dagger = \underset{V}{\begin{bmatrix} \mid & \\ v_1, v_2, \ldots v_m \\ \mid & \end{bmatrix}} \cdot \begin{bmatrix} \frac{1}{\sqrt{\lambda_1}} & 0 & \ldots & 0 & \ldots & 0 \\ 0 & \frac{1}{\sqrt{\lambda_2}} & \ldots & 0 & \ldots & 0 \\ \vdots & & & & & \\ 0 & & \frac{1}{\sqrt{\lambda_{m-1}}} & 0 & \ldots & 0 \\ 0 & \ldots & \ldots & 0 & \ldots & 0 \end{bmatrix} \cdot \underset{U^t}{\begin{bmatrix} u_1^t \\ u_2^t \\ \ldots \\ u_n^t \end{bmatrix}}$$

pseudo-inverse

[2] Singular Vector Decomposition of a Rectangular Matrix (Pseudo-Inverse)

ex) compute the pseudo-inverse of the matrix below. [rectangular matrix]

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

[3] Singular Vector Decomposition of a Rectangular Matrix (Pseudo-Inverse)

ex) compute the pseudo-inverse of the matrix below. [square matrix]

$$\Sigma = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} 3 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}$$

for square and full rank matrix,
$[\Sigma^{\dagger} = \Sigma^{-1}]$

[4] Singular Vector Decomposition of a Rectangular Matrix (Linear System: D$x = b$)

no solution of a linear system (D$x = b$) exist
if the vector b is on the column space of D.

ex] in linear regression, we aim to solve $[a, b, c]$

	X_1	X_2	X_3
#1	1.4	2.7	3.2
#2	2.2	3.5	3.3
#3	3.1	5.2	1.2
#4	1.7	1.0	0.2
#5	4.6	1.1	0.9
#6	2.2	4.3	2.7
#7	1.2	2.3	7.6
#8	0.3	0.2	3.2
#9	2.5	0.5	0.9
#10	1.8	1.9	1.1
…			

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} =$$

	Y
#1	1
#2	1.3
#3	0.8
#4	0.3
#5	1
#6	4
#7	3.6
#8	2.9
#9	2.5
#10	0
…	

- the vector for Y is likely be outside of the three-dimensional column space in $\mathbb{R}^M$ (M: $\#data$)

[5] Singular Vector Decomposition of a Rectangular Matrix (Linear System: Dx $=$ b)

$Dx = b$

- no solution

$D^t D x = D^t b$

- projection to column space (approximated)
- exist solution (one / infinite many solution)

$(D^t D)^\dagger \, (D^t D) x = (D^t D)^\dagger \, D^t b$

x $= (D^t D)^\dagger \, D^t b$

- by using pseudo-inverse,
  find a solution in the subspace

We will study solving linear system using pseudo-inverse again
in Linear Regression (Sept. 15)