CS 461: Machine Learning Principles

Class 4: Sept. 16

Data, Feature Extraction, PCA

Instructor: Diana Kim

+ Regression and Classification models are basically linear on the top of feature space.
+To optimize for the linear models, careful feature engineering is needed
to reflect the nature of the target tasks well

Data (Experience, Experiment Outcomes)

- Output of observations: images in digital format, sequence of DNA, piece of texts, time sampled signals.

- Data can be thought as Random Vectors' realization.

Matrix Form $\vec{D} = (D_1, D_2, \ldots, D_M)$

Feature: M

data: N

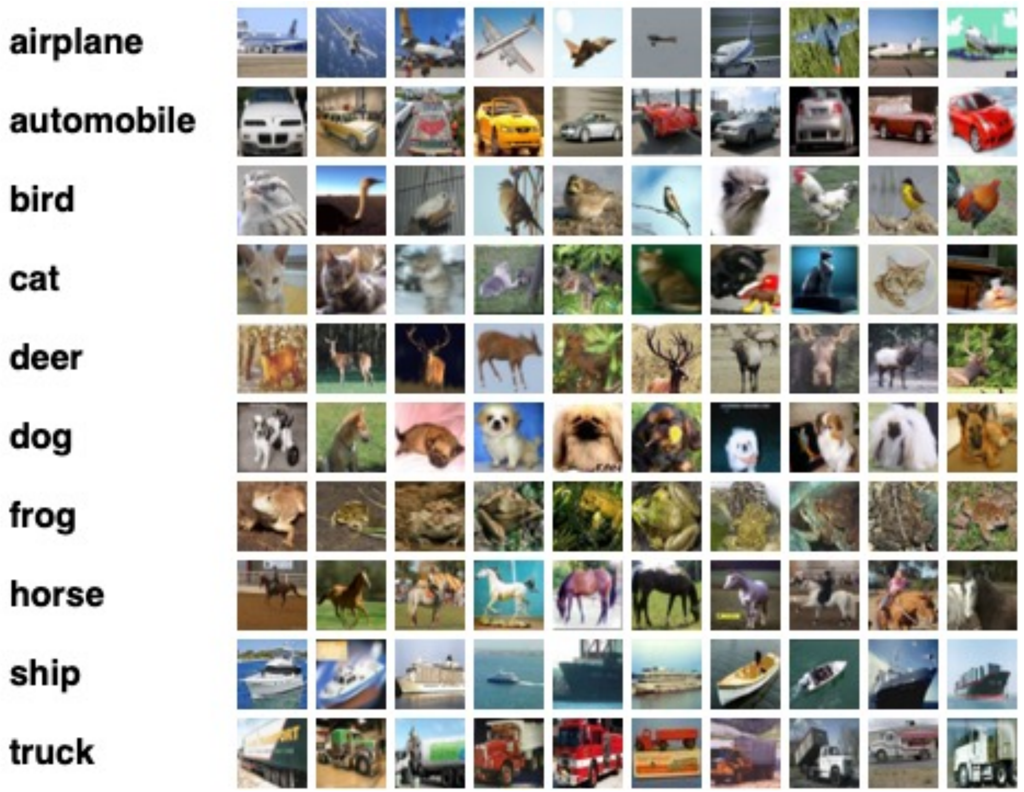
d_11	d_12	…	d_1M
d_21	d_22	…	d_2M
…			
…			
…			
…			
d_N1	d_N2	…	d_NM

- N times realization of $\vec{D}$
- N data points in M dimensional space
- M feature points in N dimensional space

Various Data Types

Various Data Types (1) : Images

From https://yann.lecun.com/exdb/mnist/index.html



https://www.cs.toronto.edu/~kriz/cifar.html

- Image Data
- MNIST (28 x 28 pixels = 784 –D vectors), ImageNet, CIFAR10 /100 WikiArt

Various Data Types (2): Time Signal

From the paper
"EEG-Based Emotion Classification Using Long Short-Term Memory Network with Attention Mechanism" by Kim et al.

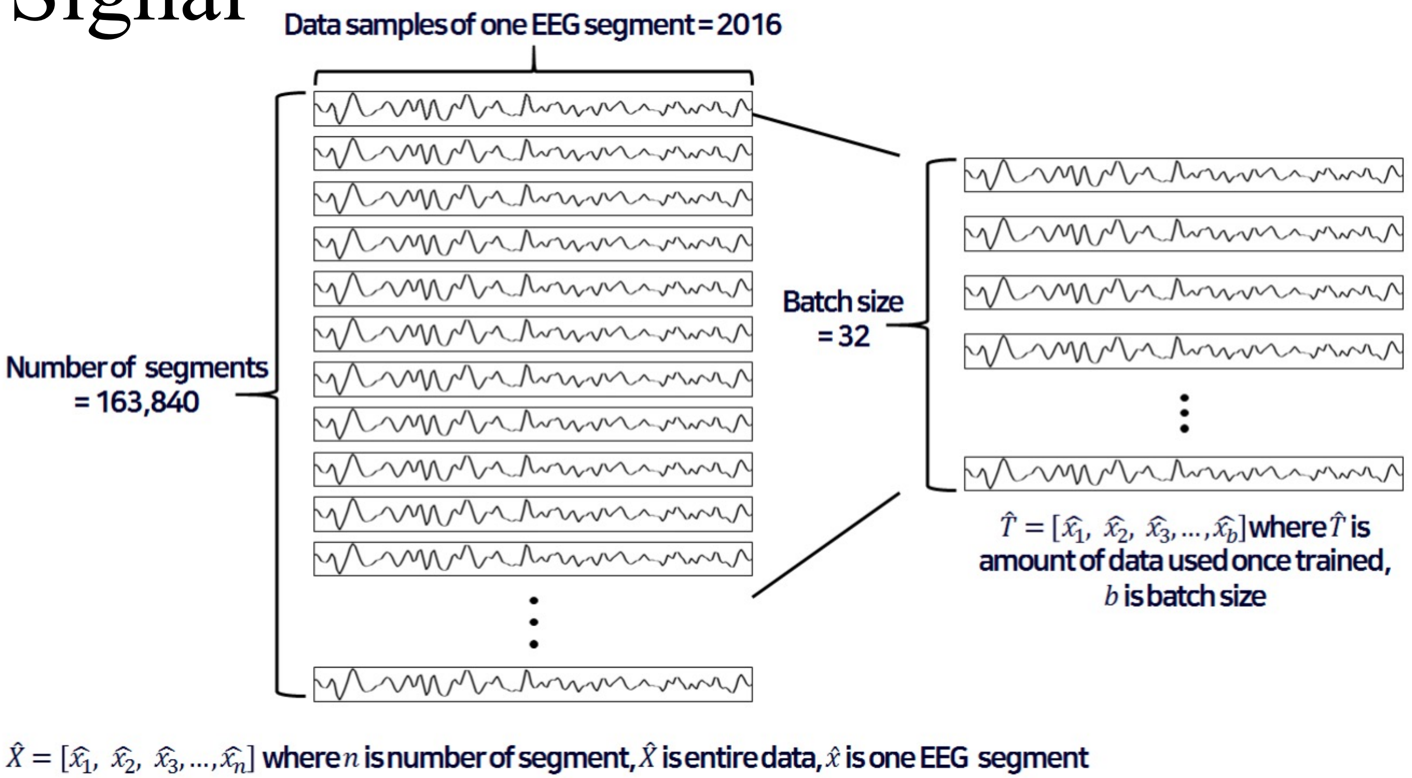
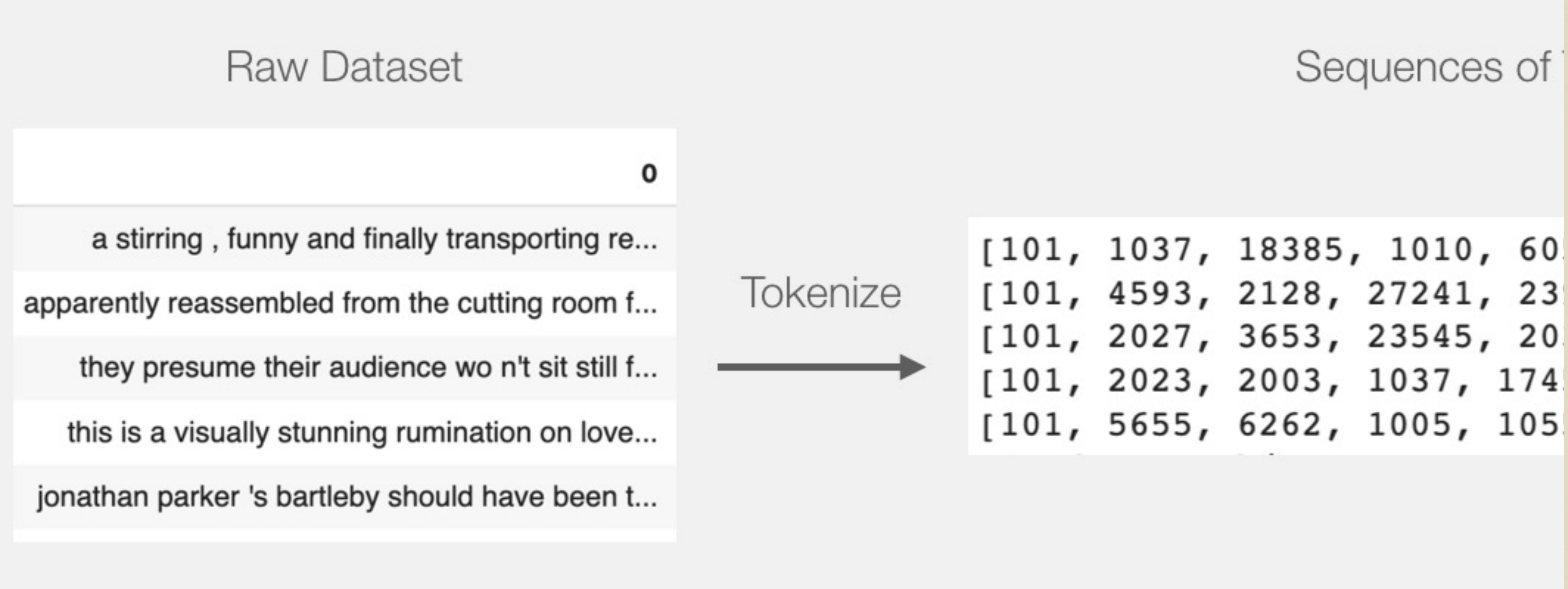


Figure 1. Example of an electroencephalogram (EEG) input segment with a window of about 15 s (exactly 15.75 s).

+ The length of time series entries can vary. Recurrent modeling like RNN can handle the case.

- Sampled time series
- The record of electric activity in brain (EEG)
- Weather data, Stock prices

Various Data Types (3) : Texts/ Tokens



From https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

- Texts
- Tokenized based on Vocabulary

```
went
january
october
our
august
april
york
12
few
2012
2008
east
show
member
college
2009
father
public
##us
come
men
five
set
station
church
##c
next
former
november
room
party
located
december
2013
```

(a) (b) (c)

Figure 1.1: Three types of Iris flowers: Setosa, Versicolor and Virginica. Used with kind permission of Dennis Kramb and SIGNA.

Various Data Types
: the high-level features

index	sl	sw	pl	pw	label
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
	...				
50	7.0	3.2	4.7	1.4	Versicolor
	...				
149	5.9	3.0	5.1	1.8	Virginica

Table 1.1: A subset of the Iris design matrix. The features are: sepal length, sepal width, petal length, petal width. There are 50 examples of each class.

Kepler's Empirical Discovery Planetary Motion

	D	P	D^2	P^3
Mercury	0.24	0.39	0.058	0.059
Venus	0.62	0.72	0.38	0.39
Earth	1.00	1.00	1.00	1.00
Mars	1.88	1.53	3.53	3.58
Jupiter	11.90	5.31	142.00	141.00
Saturn	29.30	9.55	870.00	871.00

From Kernel Methods for Pattern Analysis by John Shawe-Talyor

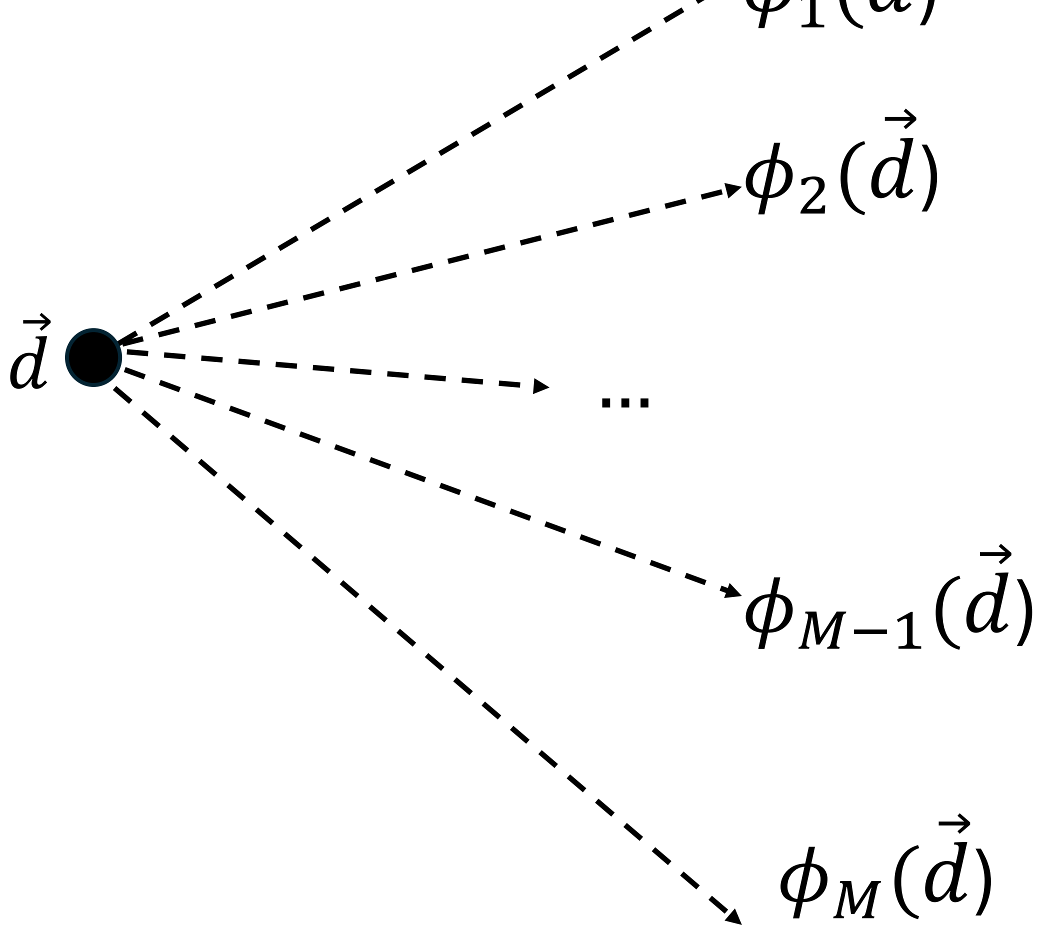
- **P**eriod : the time of one revolution around the sun
- **D**istance: the average distance D from the sun

- $P^3 = D^2$
- If there is a pattern among the features, we can predict one features from the remaining ones.

The raw/ original data as is may not be good for automatic learning

- needs feature extraction/transformation
- the raw data is embedded into the feature space (higher/smaller-D)
- must reflect the nature/essence of our target problems

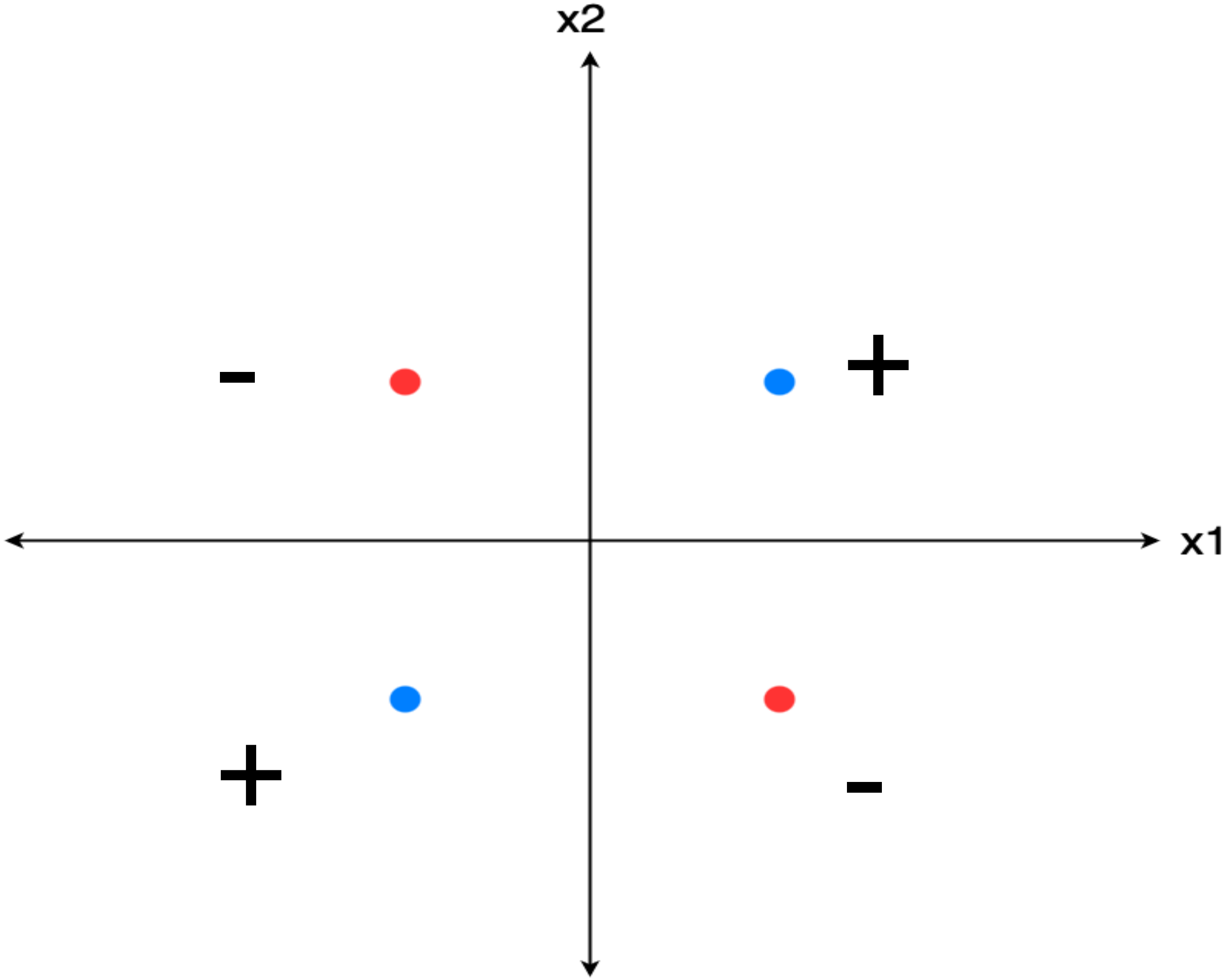
Feature Extraction is a mapping ϕ: $\overrightarrow{D} \rightarrow R^M$ (data embedding)



XOR Problem

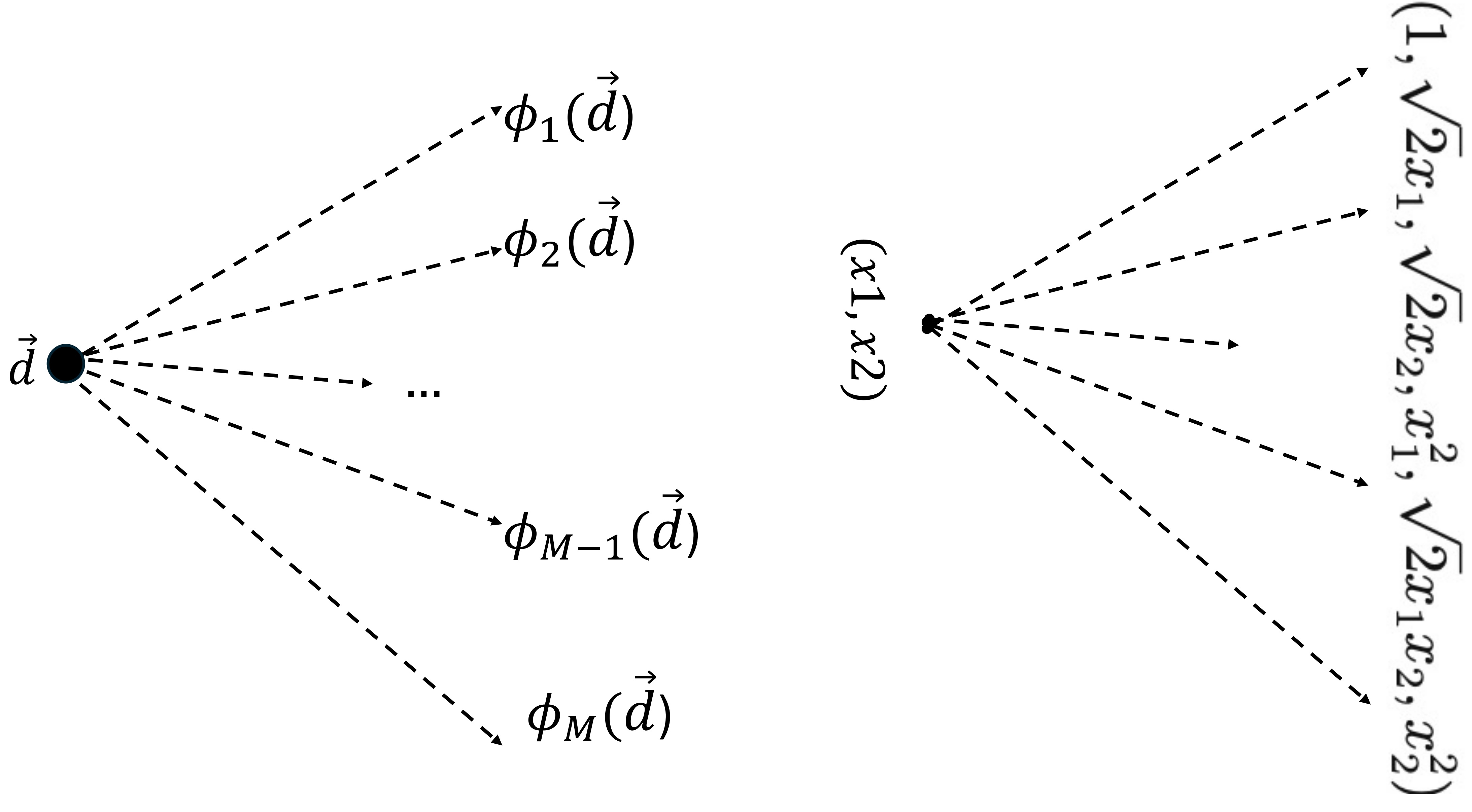
(x1,x2) ----------> (x1,x2, x1*x2)

If the (x1*x2) is added to the feature space, then the space becomes linearly separable.



Q: How would you create $\mathbf{X}_3$ to make the feature space to be linearly separable?

Feature Extraction (1) : Feature Engineering



Feature Extraction (1) : Feature Engineering

data embeddings

$\phi_1(\vec{d})$

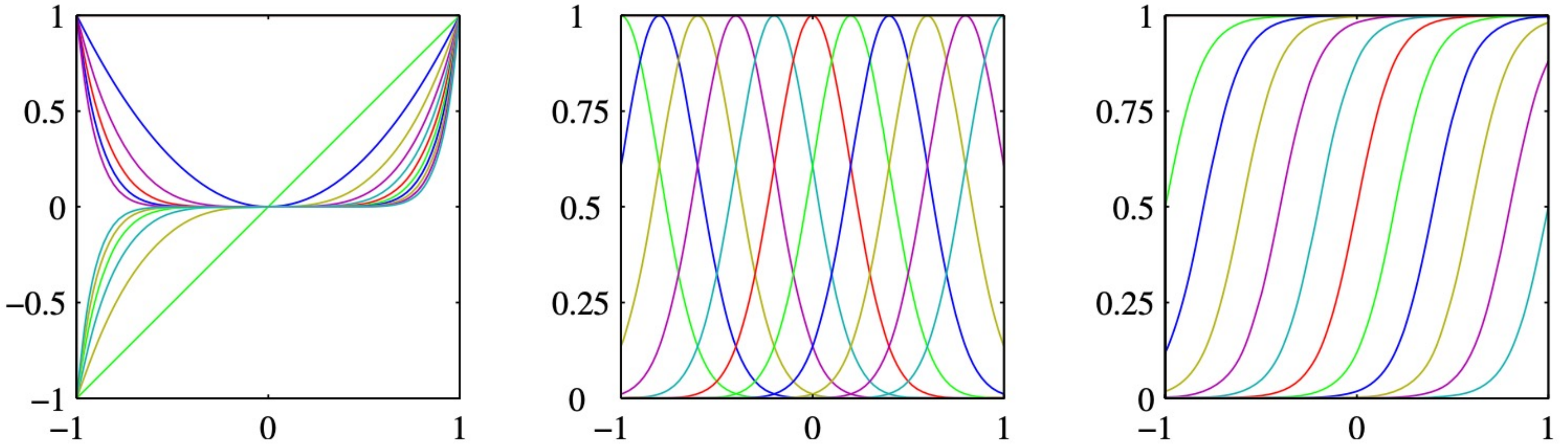
$\phi_2(\vec{d})$

$\vec{d}$

...

$\phi_{M-1}(\vec{d})$

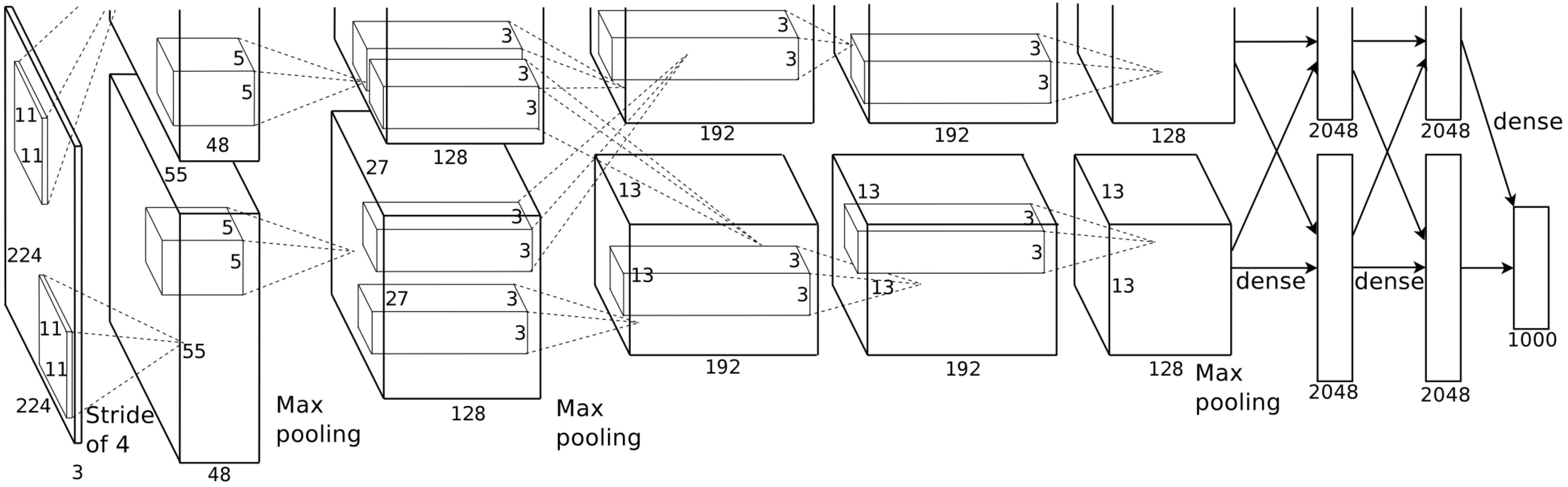
$\phi_M(\vec{d})$



The Various Types of Basis Functions (Polynomial, Gauss, Logistic Sigmoid)

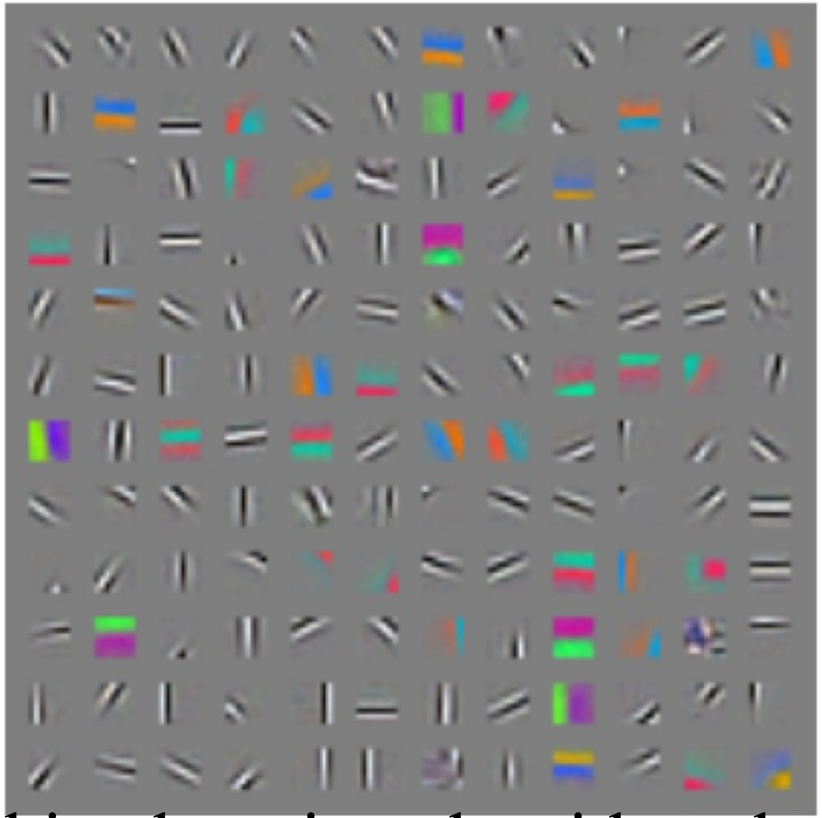
From Figure 3.1 Textbook Bishop

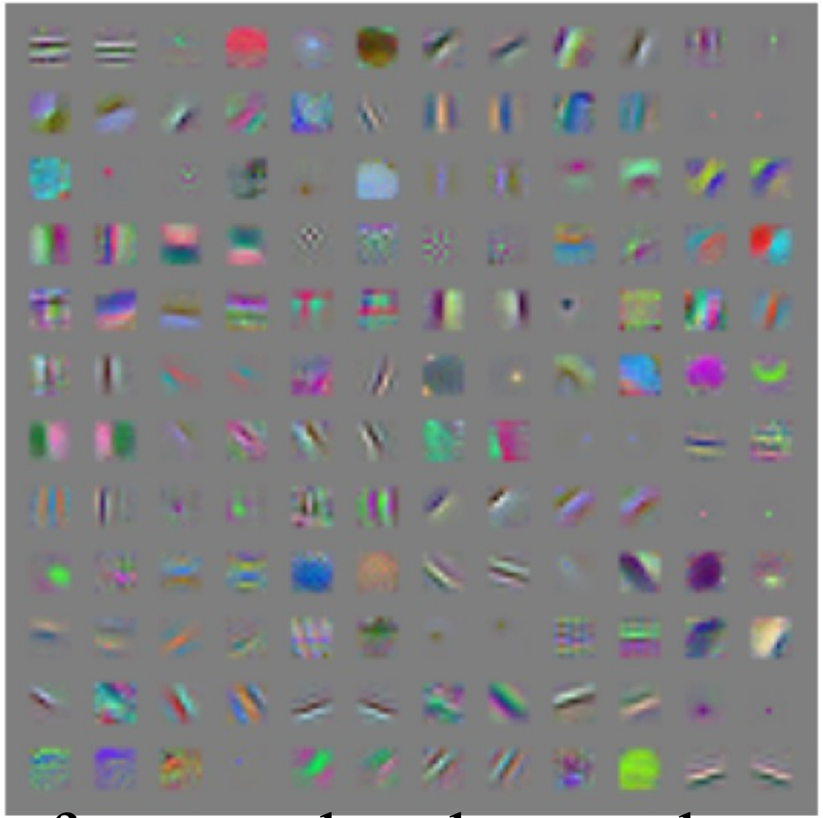
Feature Extraction (2): Learned Features



Learning the low-level features to the high-level features through the hierarchical structure of deep-CNN.

Feature Extraction (2): Learned Features





Many machine learning algorithms learn features that detect edges. These feature detectors are reminiscent of the Gabor functions known to be present in primary visual cortex. (Left) Weights learned by an unsupervised learning algorithm and (Right) Convolution kernels learned by the first layer of a fully supervised convolutional maxout network. Figure 19. 19 from Deep Learning by Ian Goodfellow

Linear relations will be sought among the images of the data items in the feature space.

- Linear Regression
- Linear Classification

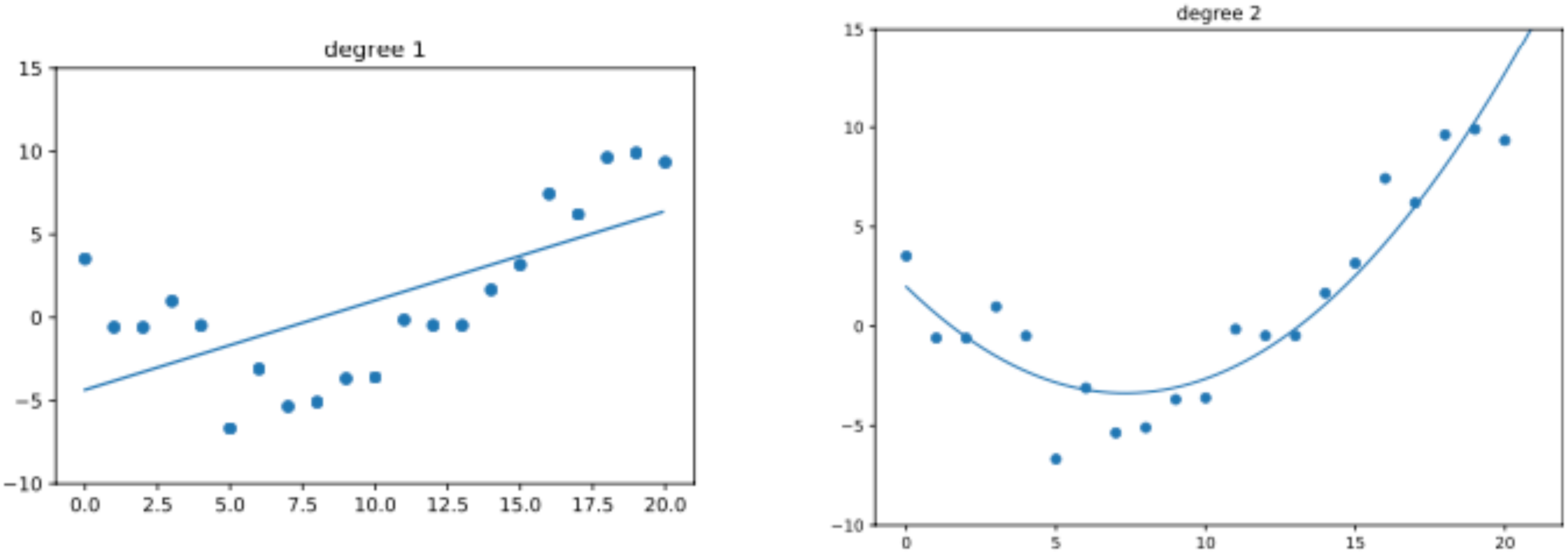
- **Linear Regression Modeling:**
- $\boldsymbol{D \cdot w = \; y'}$

$$\text{\# data: N}\left\{\overbrace{\begin{bmatrix} d_{11} & d_{12} & \dots & d_{1M} \\ d_{21} & d_{22} & \dots & d_{2M} \\ \dots & & & \\ \dots & & & \\ \dots & & & \\ \dots & & & \\ d_{N1} & d_{N2} & \dots & d_{NM} \end{bmatrix}}^{\text{\# Feature: M}}\right. \times \begin{bmatrix} W_1 \\ W_2 \\ \dots \\ \\ W_M \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ \\ \\ \\ y_n \end{bmatrix}$$

- **Binary Classification Modeling (Learning a Decision Rule): $\boldsymbol{D \cdot w \gtreqless 0}$**

$$\text{\# data: N}\left\{\overbrace{\begin{bmatrix} d_{11} & d_{12} & \dots & d_{1M} \\ d_{21} & d_{22} & \dots & d_{2M} \\ \dots & & & \\ \dots & & & \\ \dots & & & \\ \dots & & & \\ d_{N1} & d_{N2} & \dots & d_{NM} \end{bmatrix}}^{\text{\# Feature: M}}\right. \times \begin{bmatrix} W_1 \\ W_2 \\ \dots \\ \\ W_M \end{bmatrix} \overset{+}{\underset{-}{\gtreqless}} \begin{bmatrix} 0 \\ 0 \\ \dots \\ \\ \\ \\ 0 \end{bmatrix}$$

The goal of Regression



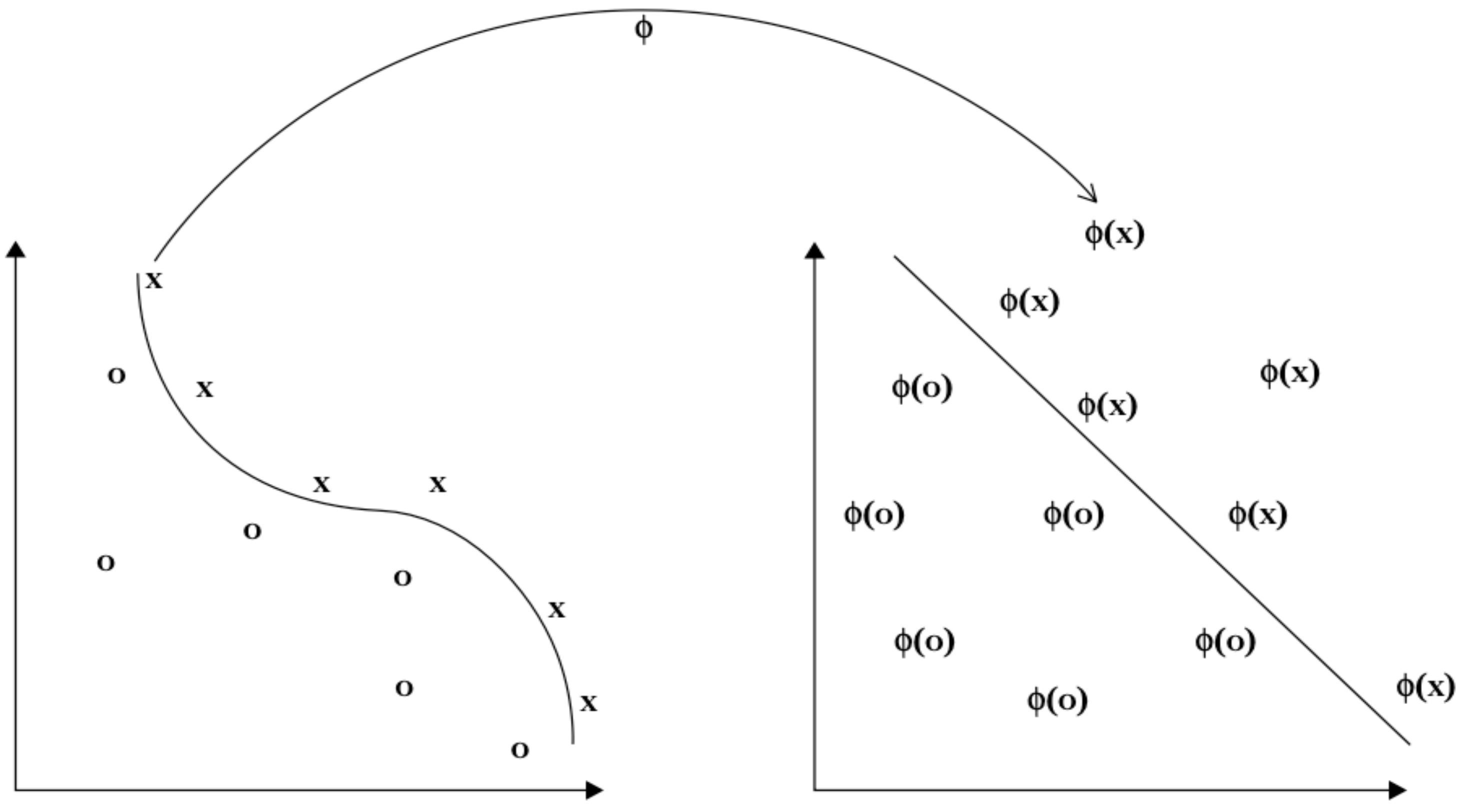
(a) (b)

From Textbook Murphy

Figure 11.1: Polynomial of degrees 1 and 2 fit to 21 datapoints. Generated by linreg_poly_vs_degree.ipynb.

- Linear Regression: $f_w\ (x, y) = \ \ 0$
- Given data, how can we find the function $f_w\ (x, y) = \ \ 0$ matched the best to data based on a predefined metric: MMSE/ MAP

The goal of Classification



From Kernel Methods for Pattern Analysis by John Shawe-Talyor

Fig. 2.1. The function ϕ embeds the data into a feature space where the nonlinear pattern now appears linear. The kernel computes inner products in the feature space directly from the inputs.

- Linear Classification: $f_w\ (x, y) \gtreqless 0$
- Given data, how can we find the hyperplane $f_w\ (x) = \ \ 0$ perfectly separates +/ - samples? Through feature transformation

The Linear Modeling
demands very expressive feature space.
But, there is a thing we must consider.
Q?

The Linear Modeling
demands very expressive feature space.
There is a thing we must consider: # data points.
But, what if there exist an algorithm that can identify
the data points that define the maximum margin hyperplanes?

If an algorithm can identify boundary active samples, those samples can define an optimal hyperplane (separates the space with the best margin). This implies that we can be free from the constraints by #data. The algorithm is SVM, but we will cover it later.

Feature Selection Rules

Feature Selection Rules

1. Hypothetical Space: enough capacity but not too complex
   - Ridge Regression (Regularization)
   - Selection based on Cross Validation

2. No collinearity effect
   - may be okay for prediction performance but hard to interpret the results. and results in large variation.
   - can be reduced by whitening preprocessing

The Effect of Collinearity

- $y \ = w_1 x_1 \ + \ w_2 x_2 + \mathrm{b}$
- If x_1 and x_2 are correlated then $x_1 = \alpha \ x_2$
- Then the possible MMSE solutions are infinitely many.

- Singular cases

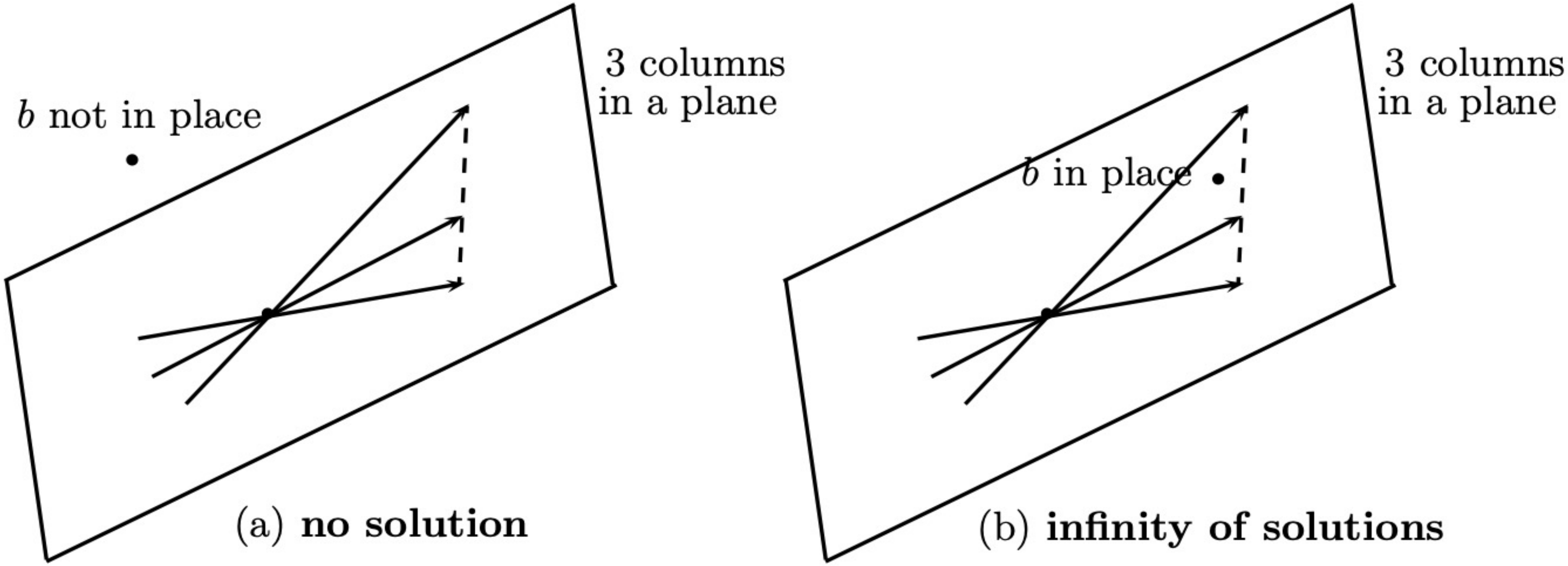


Figure 1.6: Singular cases: b outside or inside the plane with all three columns.

From Linear Algebra and Its Application by Gilbert Strang

$$D^t D \cdot w = D^t b$$

$$where\ D\ (n \times m)\ \text{ is data matrix}$$

- $\text{what if } \mathrm{D^tD} \text{ does not have inverse?}$
- $\text{i. e it}'\text{s spectral decomposition contians zero eigenvalues}$
- $(\text{Pseudo} - \text{Inverse})$

Pseudo Inverse $D^t \cdot D$

$$D^t \cdot D = V \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & & & \\ 0 & & \lambda_{m-1} & 0 \\ 0 & \dots & \dots & 0 \end{bmatrix} V^t$$

$$(D^t \cdot D)^\dagger = V \begin{bmatrix} \frac{1}{\lambda_1} & 0 & \dots & 0 \\ 0 & \frac{1}{\lambda_2} & \dots & 0 \\ \vdots & & & \\ 0 & & \frac{1}{\lambda_{m-1}} & 0 \\ 0 & \dots & \dots & 0 \end{bmatrix} V^t$$

Pseudo Inverse D (SVD)

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

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

$\mathrm{D}\cdot D^{\dagger}$ vs. $D^{\dagger} \cdot D$

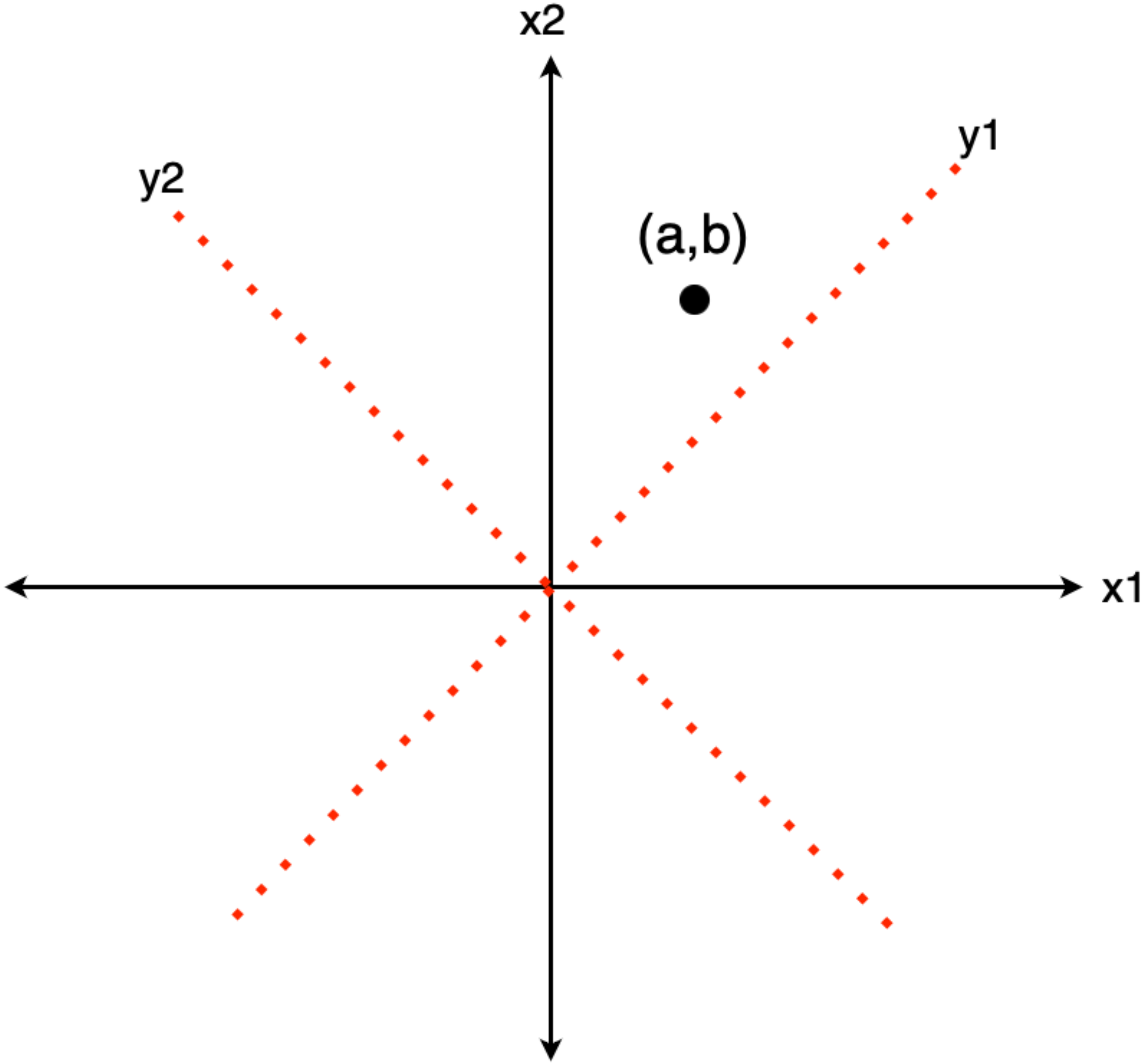
$$
\mathrm{D} = \begin{bmatrix} & | & \\ u_1, & u_2, & \ldots u_n \\ & | & \end{bmatrix} \cdot \begin{bmatrix} \sqrt{\lambda_1} & 0 & \ldots & 0 \\ 0 & \sqrt{\lambda_2} & \ldots & 0 \\ \vdots & & & \\ 0 & & \sqrt{\lambda_{m-1}} & 0 \\ 0 & \ldots & \ldots & \sqrt{\lambda_m} \\ 0 & \ldots & \ldots & 0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

$DD^{+}(N \text{ by } N)$ *has zero eigenvalus on diagonal*
while$D^{+}D$ $(M \text{ by } M)$ *will have all positive eigenvalues.*

Principal Component Analysis (PCA)

We want to make data (N-D) moves around within a subspace (M-D), (N>M)

Vector $\vec{x} = (a, b)$ can be represented by different orthonormal bases.



- X_n

When $x_n \in R^D$ and u_i where $i = 1, 2, ..., D$ are are abnormal basis,

$$x_n = \sum_{i=1}^{D} \alpha_{ni} u_i \quad \text{and} \quad \alpha_{ni} =< x_n, u_i >$$

- $\widetilde{X_n}$

We want to approximate x_n on the subspace of the first M basis,

$$\tilde{x_n} = \sum_{i=1}^{M} z_{ni} u_i + \sum_{i=M+1}^{D} b_i u_i$$

Q: What is the optimal values for Z_{ni} and b_i minimizing $\mathrm{J} = \| X_n - \widetilde{X_n} \|^2$

We want to minimize the averaged square error between x_n and $\tilde{x_n}$

$$\arg\min_{(z_{ni},b_i)} J = \frac{1}{N}\sum_{n=1}^{N}(< x_n, u_i > \cdot u_i^t - \sum_{i=1}^{M} z_{ni}u_i^t - \sum_{i=M+1}^{D} b_i u_i^t)\cdot$$

$$(< x_n, u_i > \cdot u_i - \sum_{i=1}^{M} z_{ni}u_i - \sum_{i=M+1}^{D} b_i u_i)$$

- Respect to z_{nk}

$$\frac{\partial J}{\partial z_{nk}} = (-2u_k^t)\cdot(< x_n, u_i > \cdot u_i - \sum_{i=1}^{M} z_{ni}u_i - \sum_{i=M+1}^{D} b_i u_i)$$

$$= -2 < x_n, u_k > +2z_{nk} = 0$$

- Respect to b_r

$$\frac{\partial J}{\partial b_r} = \frac{1}{N}\sum_{n=1}^{N}(-2u_r^t)\cdot(< x_n, u_i > \cdot u_i - \sum_{i=1}^{M} z_{ni}u_i - \sum_{i=M+1}^{D} b_i u_i)$$

$$= \frac{1}{N}\sum_{n=1}^{N}(-2 < x_n, u_r > +2b_r)$$

- Rewrite J

$$J = ||x_n - \tilde{x_n}||^2 = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=M+1}^{D}((x_n - \bar{x})^t u_i)^2$$

$$= \sum_{i=M+1}^{D} u_i^t \cdot \boxed{\frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})(x_n - \bar{x})^t} \cdot u_i$$

$$= \sum_{i=M+1}^{D} u_i^t \Sigma u_i$$

Estimation of COV (X,X)

- Lagrangian Function for the Constraint ${u_i}^t u_i = 1$

Lagrangian function for the constraint $||u_i|| = 1$

$$J(\lambda) = \sum_{i=M+1}^{D} u_i^t \Sigma u_i + \lambda(1 - u_i^t u_i)$$

$$\frac{\partial J}{\partial u_i} = \Sigma u_i - \lambda^* u_i = 0$$

Q: What the optimal solution indicate about u_i?

- Go back to J

$$J = ||x_n - \tilde{x_n}||^2 = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=M+1}^{D}((x_n - \bar{x})^t u_i)^2$$

$$= \sum_{i=M+1}^{D} u_i^t \cdot \boxed{\frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})(x_n - \bar{x})^t} \cdot u_i$$

$$= \sum_{i=M+1}^{D} u_i^t \Sigma u_i$$

Estimation of COV (X,X)

Q: To minimize J?

Now, we are ready to define $\widetilde{X_n}$ $(PCA\ Approximation)$

$$
\begin{aligned}
\tilde{x_n} &= \sum_{i=1}^{M}(x_n^t u_i)u_i + \sum_{i=M+1}^{D}(\bar{x}^t u_i)u_i \\
&= \bar{x} - \bar{x} + \sum_{i=1}^{M}(x_n^t u_i)u_i + \sum_{i=M+1}^{D}(\bar{x}^t u_i)u_i \\
&= \bar{x} - \sum_{i=1}^{M}(\bar{x}^t u_i)u_i + \sum_{i=1}^{M}(x_n^t u_i)u_i \\
&= \bar{x} + \sum_{i=1}^{M}((x_n^t - \bar{x}^t)u_i)u_i \\
&= \bar{x} + U_M U_M^t (x_n - \bar{x})
\end{aligned}
$$

$\widetilde{X_n}$ is not full dimension.
Depending on how we select U_M, we can define different approximations.

- Variance of $\widetilde{x_n}$

$$\tilde{x_n} - \bar{x} = u_j^t(x_n - \bar{x})u_j$$

$$\frac{1}{N}(\tilde{x_n} - \bar{x})^t(\tilde{x_n} - \bar{x})^t = \frac{1}{N}u_j^t(x_n - \bar{x})u_j u_j^t(x_n - \bar{x})^t u_j$$

$$var(\tilde{x_n}) = \lambda_i$$

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 Mean and Eigenvectors
The image can be represented by sum of mean and the linear combinations of eigenvectors

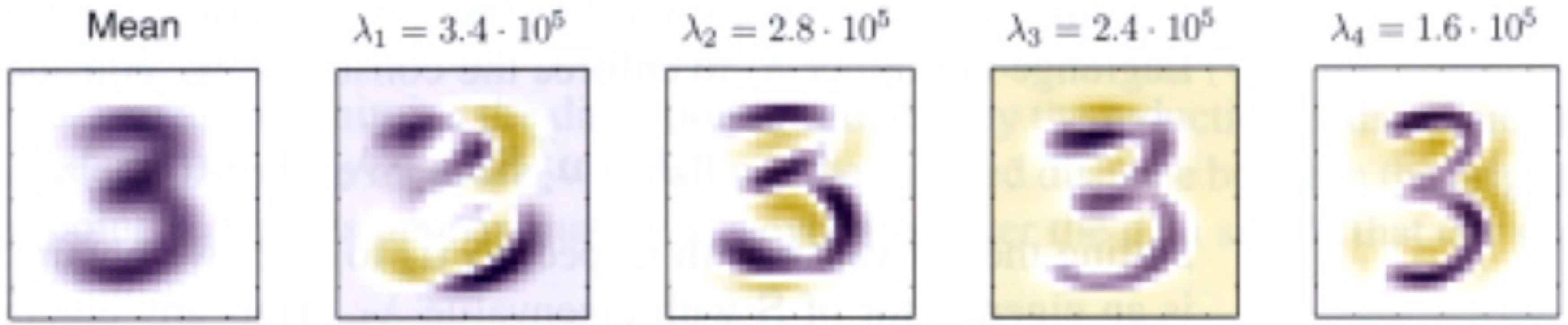


Figure 12.3 The mean vector $\overline{\mathbf{x}}$ along with the first four PCA eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_4$ for the off-line digits data set, together with the corresponding eigenvalues.

From Bishop Chap. 12

$$\widetilde{X}_n = \bar{x} + \underbrace{U_M \underbrace{U_M^t (x_n - \bar{x})}_{\text{A vector}}}_{\text{The linear combination of eigenvectors}}$$

PCA Applications

- Compression (small variance dimension does not help in learning)

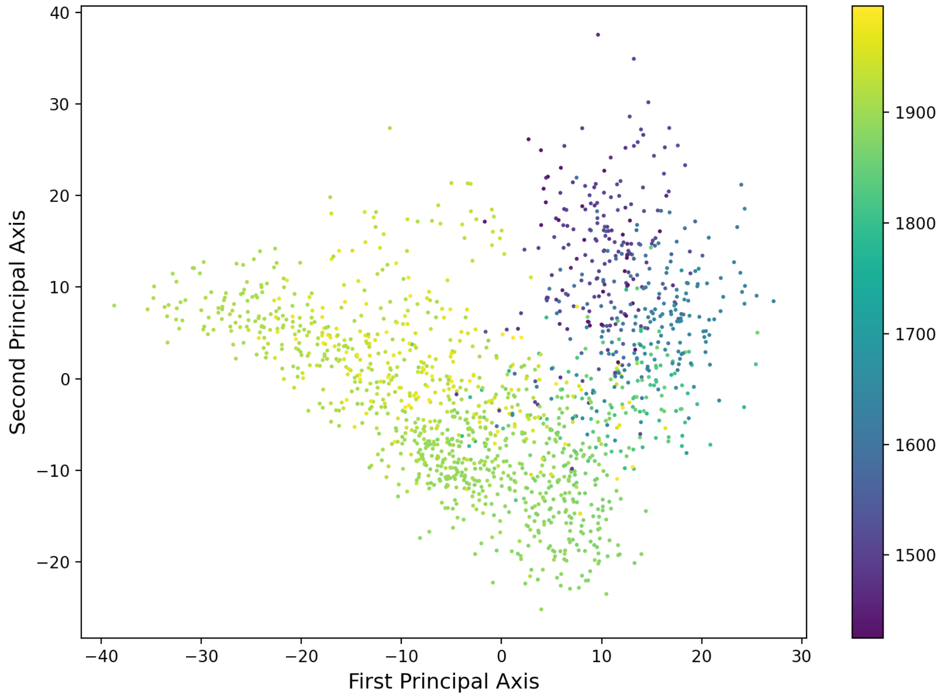
$$\widetilde{X_n} = \bar{x} + U_M U_M^t (x_n - \bar{x})$$

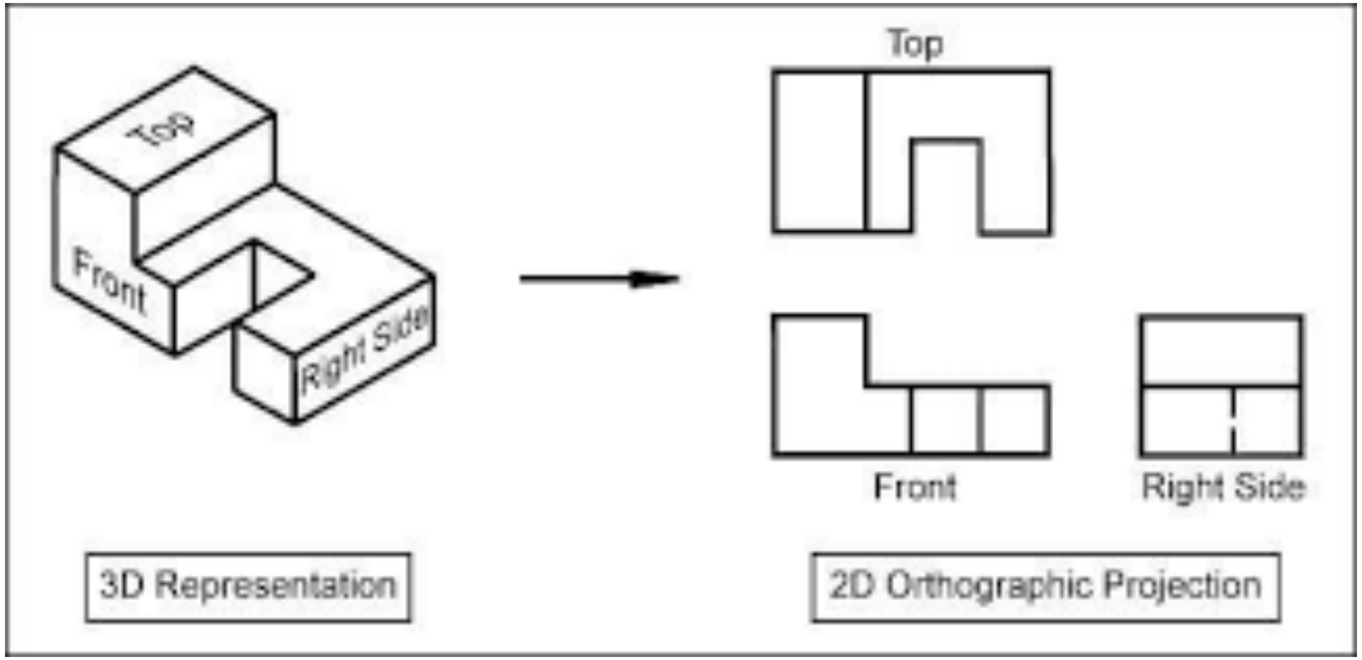
- Whitening (Rotation)

$$\tilde{x_n} = \Lambda^{-\frac{1}{2}} U_M^t (x_n - \bar{x})$$

PCA Applications

- Visualization (1) (the projection of high dimensional data to 3D or 2D)





The last hidden layer embedding 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.

PCA Applications

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

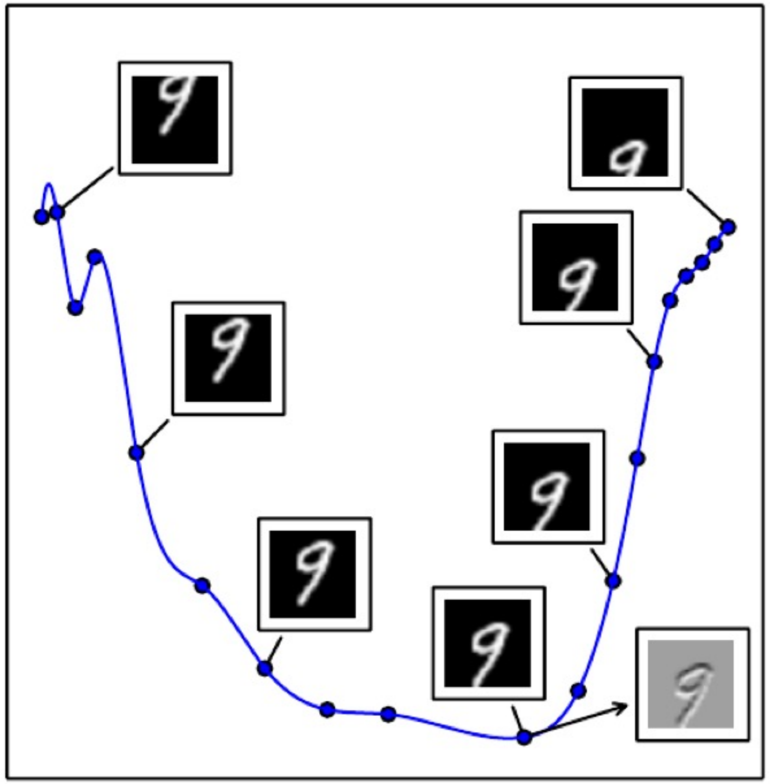


Figure 14. 6 from Deep Learning by Ian Goodfellow

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.