CS 461: Machine Learning Principles

Class 7: Sept. 26

Bayesian Regression and Summary

Instructor: Diana Kim

Bayesian Regression

$$y = \phi\left(\vec{d}(x)\right)\vec{w} + \varepsilon$$

Bayesian Regression:
MAP (**M**aximum **A** **P**osteriori estimation)

- $w^* = argmax\, p(\vec{w} \mid \vec{y}, \Phi) = \dfrac{p\left(y \middle| \vec{w}, \Phi\right) \boldsymbol{p(\vec{w})}}{p(\Phi)}$ (MAP)

Bayesian Regression:
MAP (**M**aximum **A** **P**osteriori estimation)

For the derivation of Gaussian posterior density, please check Bishop 2.3.1 - 2.3.3

- $w^* = argmax\, p(\vec{w} \mid \vec{y}, \Phi) = \dfrac{p(y \mid \vec{w}, \Phi)\, p(\vec{w})}{p(\Phi)}$ (MAP)

likelihood

$\sim \mathcal{N}_{\vec{y}}\ (\Phi \cdot \vec{w}, \sigma_\epsilon^2 I)$

prior

$\sim \mathcal{N}_{\vec{w}}\ (0, \sigma_w^2 I)$

+ when likelihood and prior are both Gaussian, posterior also Gaussian. The maximal point of the Gaussian is expected value, so MAP solution is expected value of posterior.

posterior

$$\sim \mathcal{N}_{\vec{w}}\ \left(\frac{1}{\sigma_\epsilon^2} \cdot \left(\frac{1}{\sigma_w^2} I + \frac{1}{\sigma_\epsilon^2} \Phi^t \Phi\right)^{-1} \Phi y, \left(\frac{1}{\sigma_w^2} I + \frac{1}{\sigma_\epsilon^2} \Phi^t \Phi\right)^{-1}\right)$$

MAP (Maximum A Posteriori estimation)

$$w^* = argmax\, p(\vec{w} \mid \vec{y}, \Phi)$$

$$\sim \mathcal{N}_{\vec{w}} \boxed{(\frac{1}{\sigma_\epsilon^2} \cdot (\frac{1}{\sigma_w^2} I + \frac{1}{\sigma_\epsilon^2} \Phi^t \Phi)^{-1} \Phi y,} (\frac{1}{\sigma_w^2} I + \frac{1}{\sigma_\epsilon^2} \Phi^t \Phi)^{-1})$$

$$w^* = \boxed{\frac{1}{\sigma_\epsilon^2} \cdot (\frac{1}{\sigma_w^2} I + \frac{1}{\sigma_\epsilon^2} \Phi^t \Phi)^{-1} \Phi y}$$

Gaussian density has a mode at the mean.
Q: When σ_w $is\ large\ (close\ to\ unform)\ then\ w*$ close to what?

[1] Observation in Bayesian Regression (MAP)

$$w^* = \frac{1}{\sigma_\epsilon^2} \cdot \left(\frac{1}{\sigma_w^2} I + \frac{1}{\sigma_\epsilon^2} \Phi^t \Phi\right)^{-1} \Phi y$$

+ large σ_W / prior density follows uniform,
the effect of prior becomes negligible and MAP becomes equivalent to ML solution!

- As σ_w *is large, the* prior density <u>does not gives much information about w.</u>

$$\lim_{\sigma_w \to \infty} \frac{1}{\sigma_\epsilon^2} \cdot \left(\frac{1}{\sigma_w^2} I + \frac{1}{\sigma_\epsilon^2} \Phi^t \Phi\right)^{-1} \Phi y = (\Phi^t \Phi)^{-1} \Phi y$$

This is the MLE solution!

[2] Observation in Bayesian Regression (MAP)

- $w^* = argmax\ p(\vec{w} \mid \vec{y}, \Phi) = \dfrac{p(y|\vec{w}, \Phi)p(\vec{w})}{p(\Phi)}$ (MAP)

likelihood × prior

$$\sim \mathcal{N}_{\vec{y}}\left(\Phi \cdot \vec{w}, \sigma_\epsilon I\right) \times \sim \mathcal{N}_{\vec{w}}\left(0, \sigma_w^2 I\right)$$

+ MAP is equivalent to Ridge Regression when $\lambda : \frac{\sigma_\epsilon^2}{\sigma_w^2}$. Ridge Regression is the special case of MAP (prior and likelihood are Gaussian)

$$w* = \arg\min_w \frac{1}{2\sigma_\epsilon^2}||\vec{y} - \Phi\vec{w}||^2 + \frac{1}{2\sigma_w^2}||\vec{w}||^2$$

This is the objective function of Ridge Regression

[3] Observation in Bayesian Regression (MAP) [from Bishop Figure 3.7]
: how likelihood and posterior will be updated as we collect the more data?

+ As collecting more data, likelihood and posterior are updated.

- $w^* = argmax\ p(\vec{w} \mid \vec{y}, \Phi) = \dfrac{p\left(y \middle| \vec{w}, \Phi\right) p(\vec{w})}{p(\Phi)}$ (MAP)

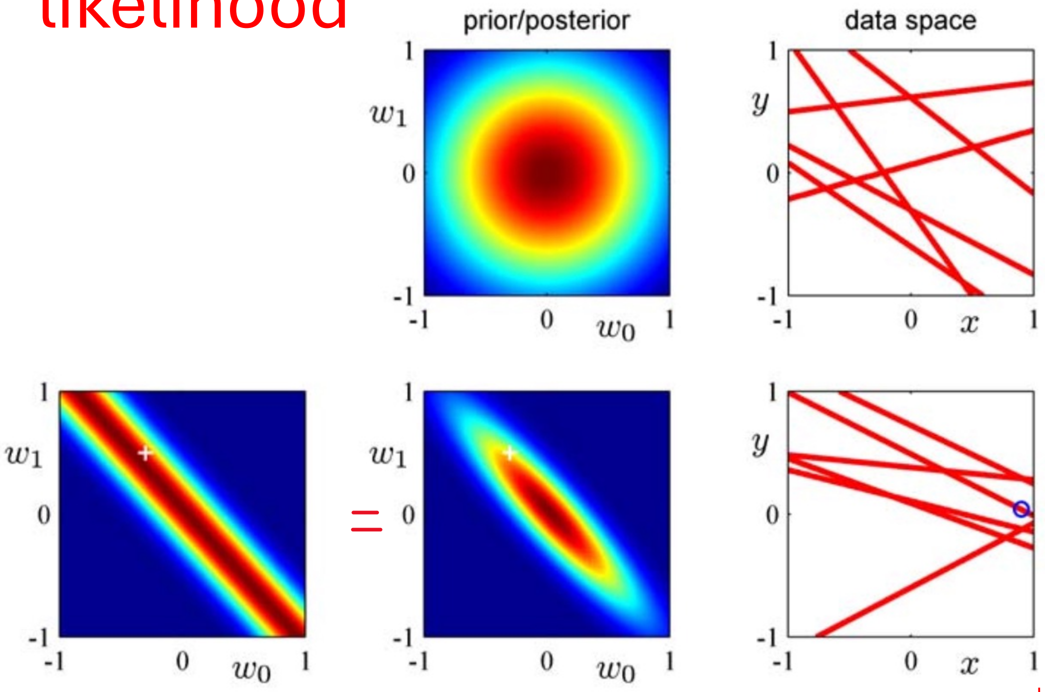
[3] Observation in Bayesian Regression (MAP) [from Bishop Figure 3.7]
: how likelihood and posterior will be updated as we collect the more data?

Suppose we collect data from $t(x) = \boldsymbol{w_0} + w_1 x + \varepsilon$.

[no data yet]

likelihood

Prior/ Posterior



- No data so no likelihood
- $$f(\vec{w}) = \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp - \frac{||\vec{w}||^2}{2\sigma_w^2}$$

+no data:
no likelihood and prior is isotropic

- $$f(x_1, t_1|\vec{w}) = \frac{1}{\sqrt{2\pi\sigma_\epsilon^2}} \exp - \frac{(t_1 - w_0 - w_1 x_1)^2}{2\sigma_\epsilon^2}$$
- $$f(\vec{w}|x_1, t_1) = f(\vec{w}) \cdot f(x_1, t_1|\vec{w})$$

[The update by the first data sample (x_1, t_1)]

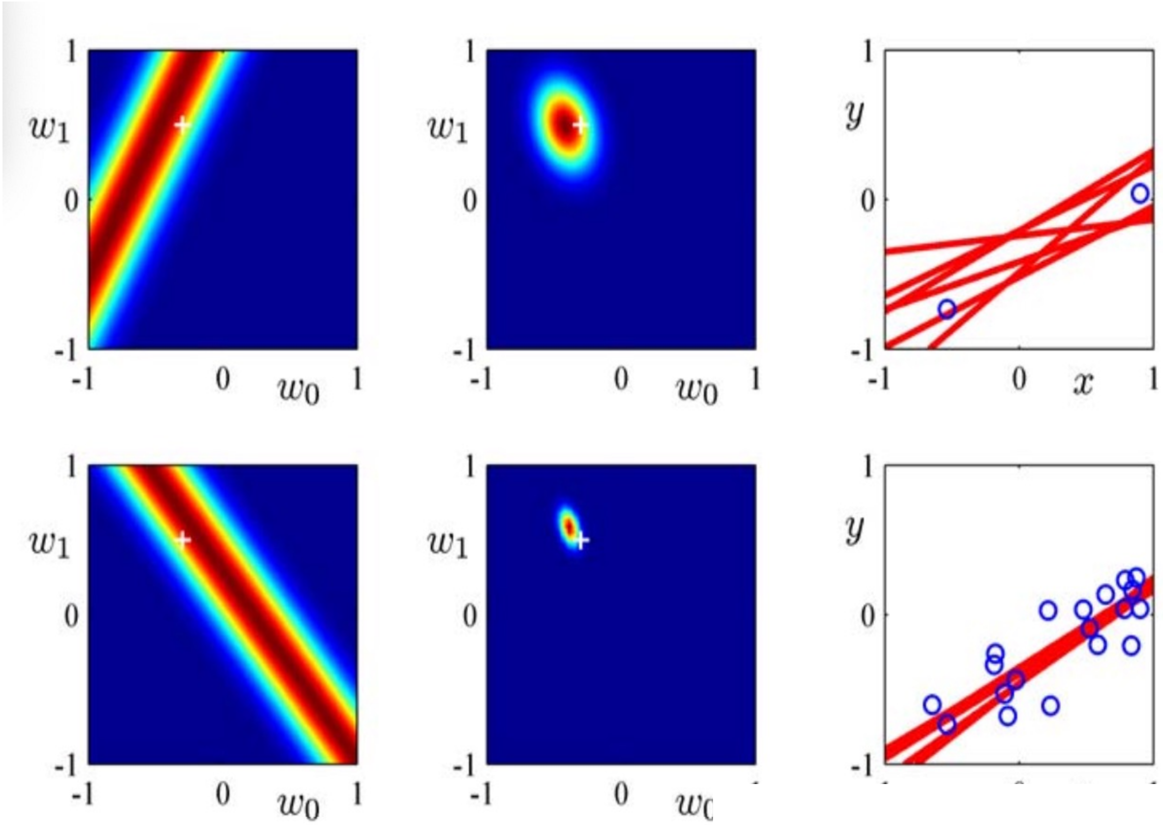
+one data point:
likelihood becomes hyperbolic cylinder highest $t_1 - w_0 - w_1 x_1 = 0$
and prior becomes non-isotropic.

The figure is drawn based on the data simulation as follows:

$$t(x) = 0.5\,x \; - 0.3 \; + \; \varepsilon$$
$$\varepsilon \sim N(0, \sigma = 0.2)$$

$(x_1 t_1), (x_2 t_2), (x_3 t_3), (x_4 t_4), \ldots$

[3] Observation in MAP
: how likelihood and posterior will be updated as we collect the more data?



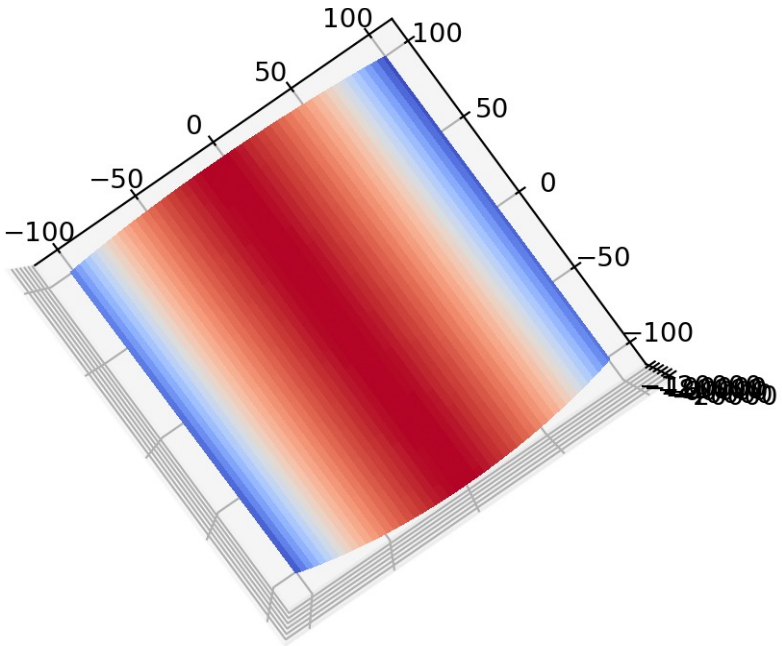
- $$f(x_2, t_2|\vec{w}) = \frac{1}{\sqrt{2\pi\sigma_\epsilon^2}} \exp - \frac{(t_2 - w_0 - w_1 x_2)^2}{2\sigma_\epsilon^2}$$
- $$f(\vec{w}|x_1, t_1, x_2, t_2) = f(\vec{w}) \cdot f(x_1, t_1, x_2, t_2|\vec{w})$$

Q) Why posterior density getting sharper as we have more data, finally pinpointing the ground truth W* = (-0.3, 0.5)?

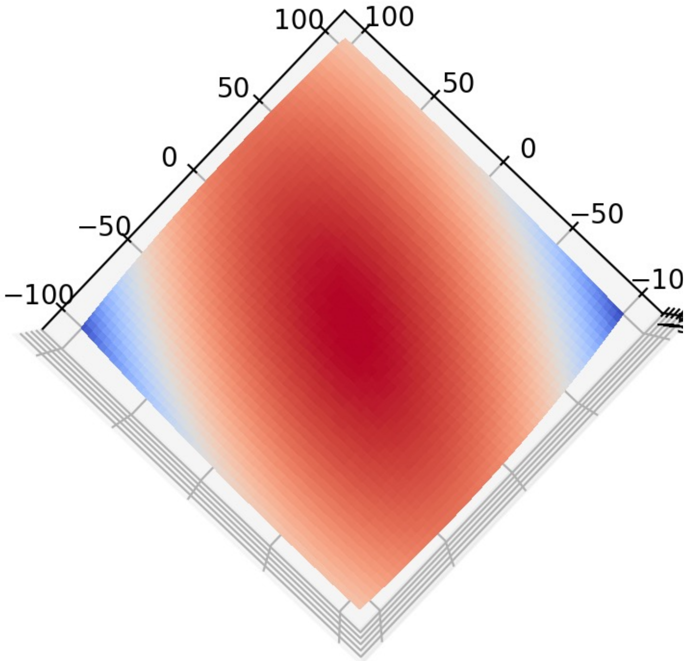
$$\sim \mathcal{N}_{\vec{w}} \left(\frac{1}{\sigma_\epsilon^2} \cdot \left(\frac{1}{\sigma_w^2} I + \frac{1}{\sigma_\epsilon^2} \Phi^t \Phi\right)^{-1} \Phi y, \boxed{\left(\frac{1}{\sigma_w^2} I + \frac{1}{\sigma_\epsilon^2} \Phi^t \Phi\right)^{-1}}\right)$$

Likelihood Function ($\vec{w}$) for the different # of data points

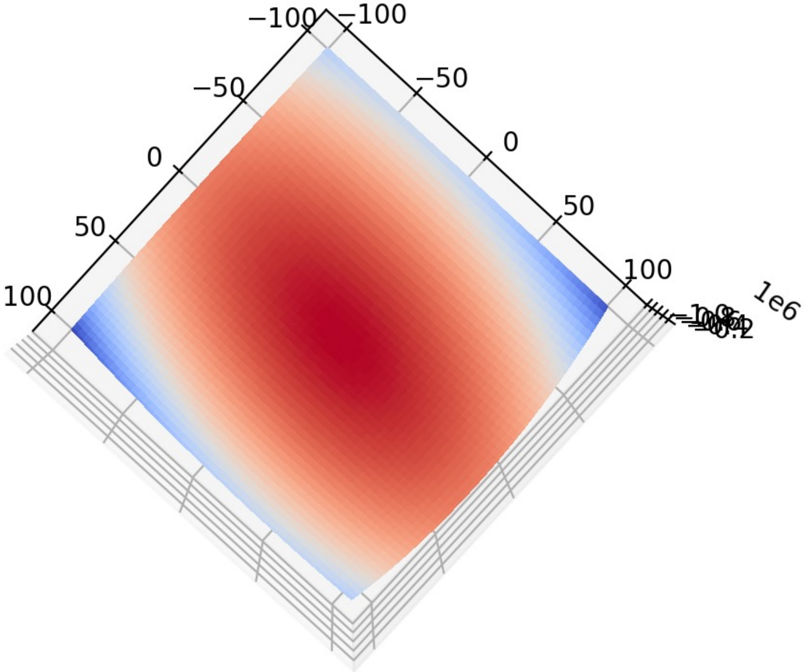
+textbook (Bishop Fig 3.7) showed likelihood only for the case: a single data point (hyperbolic cylinder).
But likelihood function becomes elliptical paraboloids as having data points more than two.



data points: 1
Parabolic Cylinder



data points: 2
Elliptical paraboloids



data points: 5

[3] Observation in MAP

: the variance of posteriori density gets smaller as N (# data) goes large.

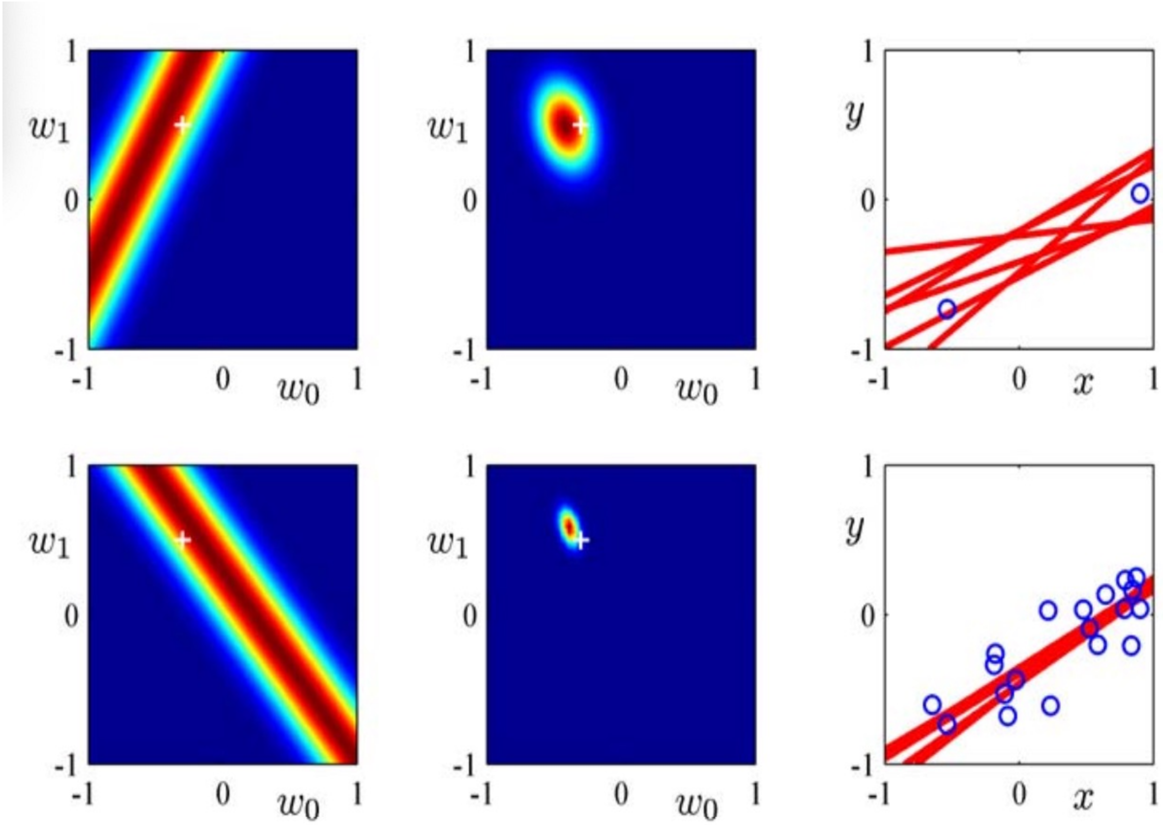
$$N * COV(X,X)$$

$$\left(\frac{1}{\sigma_w^2}I + \frac{1}{\sigma_\epsilon^2}\Phi^t\Phi\right)^{-1} =$$

$$= \left(\frac{1}{\sigma_w^2}VIV^t + \frac{1}{\sigma_\epsilon^2}V\begin{bmatrix} N\lambda_1 & 0 & 0 & \dots & 0 \\ 0 & N\lambda_2 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & \dots & \dots & N\lambda_M \end{bmatrix}V^t\right)^{-1}$$

+N goes to large, inverse matrix will have small eigenvalues.

[3] Observation in MAP
: how likelihood and posterior will be updated as we collect the more data?



- $f(x_2, t_2|\vec{w}) = \frac{1}{\sqrt{2\pi\sigma_\epsilon^2}} \exp - \frac{(t_2 - w_0 - w_1 x_2)^2}{2\sigma_\epsilon^2}$
- $f(\vec{w}|x_1, t_1, x_2, t_2) = f(\vec{w}) \cdot f(x_1, t_1, x_2, t_2|\vec{w})$

+ collecting more data,
posterior density gets sharper.
we have consistent estimate for $\vec{w}$.

For Regression Problem

We learned two approaches.

- MLE
- MAP

- Linear Regression Problem

$$y = \phi\left(\vec{d}(x)\right)\vec{w} + \varepsilon$$

- Algorithms: MLE, MAP, Ridge Regression (Gaussian prior case MAP)

MLE

$$W* = (\Phi^t\Phi)^{-1}\Phi y$$

data points matter.

MAP

$$w^* = \frac{1}{\sigma_\epsilon^2}\cdot\left(\frac{1}{\sigma_w^2}I + \frac{1}{\sigma_\epsilon^2}\Phi^t\Phi\right)^{-1}\Phi y$$

prior knowledge is expensive.

Evaluation

The goal of ML: building a robust system for unseen data

Computing Empirical MSE (Mean Square Error)

Given data = $\{(x_1, t_1),\ (x_2, t_2), (x_3, t_3), \ldots, (x_N, t_N)\}$

$$L = \frac{1}{N}\sum_{i}^{N}\{y(x_i; D) - t_i\}^2$$

Model Prediction on x Ground Truth Value

Evaluation on Training Data

✓ Underfitting

- poor performance on train set

- need to change feature map
  - changing hypothetical space
  - increasing model complexity

Evaluation on Test Data

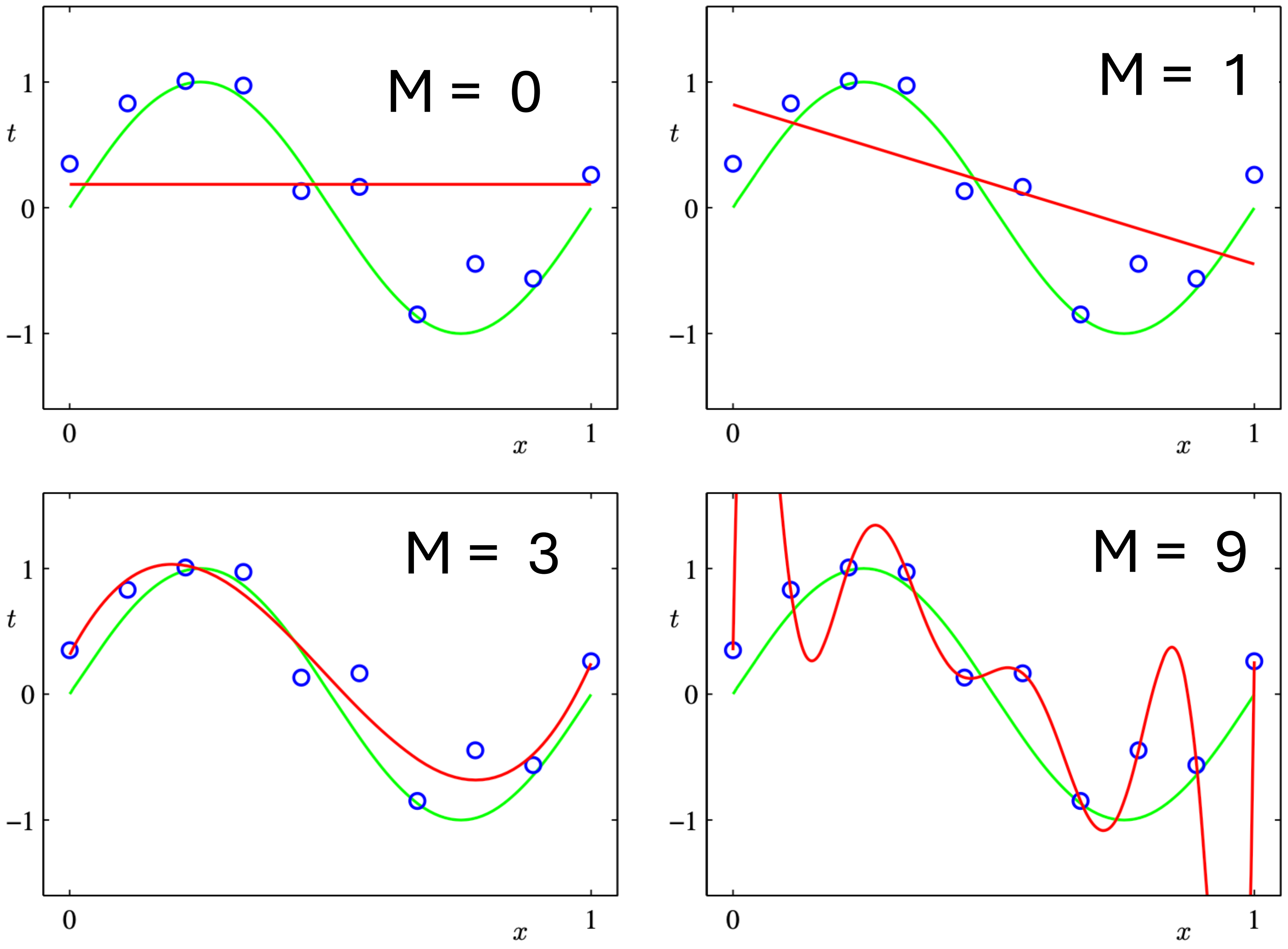
✓ Overfitting

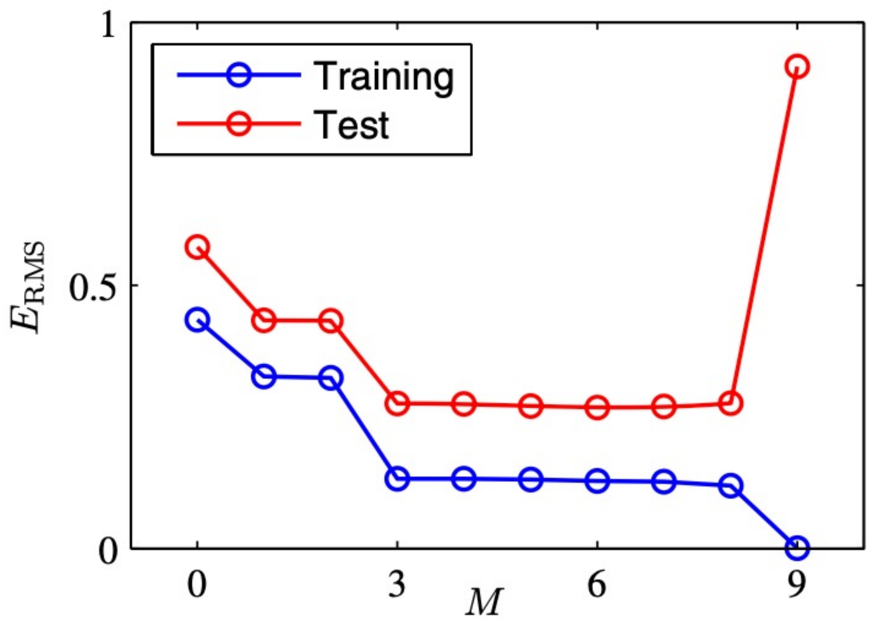
- the performance gap between test and train set

- reducing model complexity (need to consider underfitting possibility)
- collect more data

- regularization

Underfitting and Overfitting Example [from Bishop Figure 1.4 and 1.5]

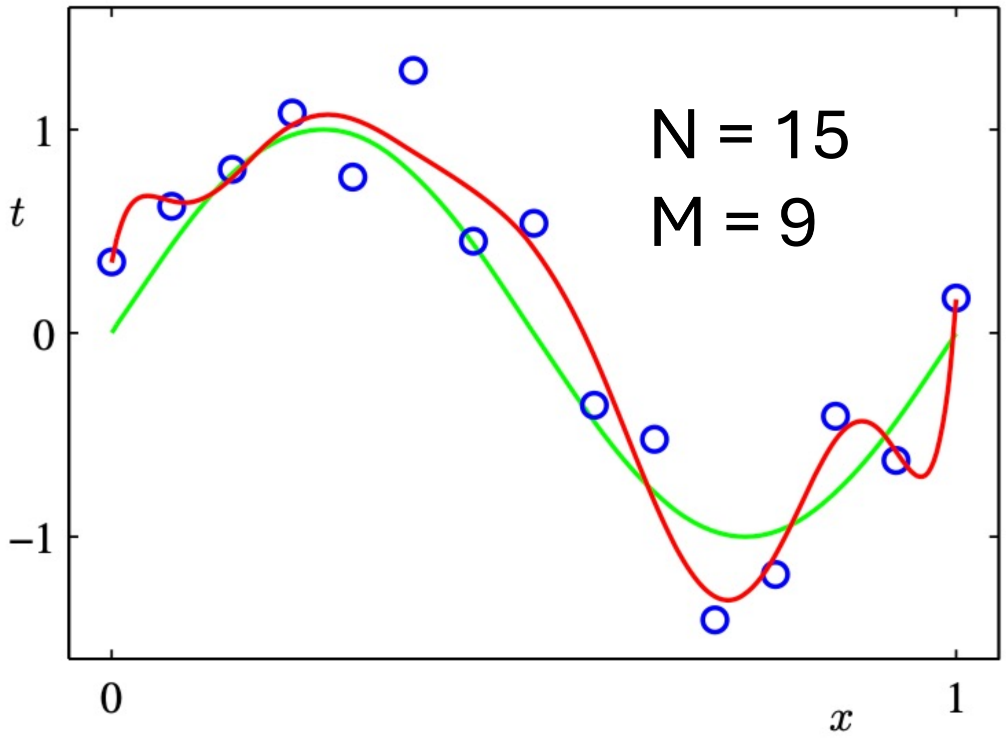


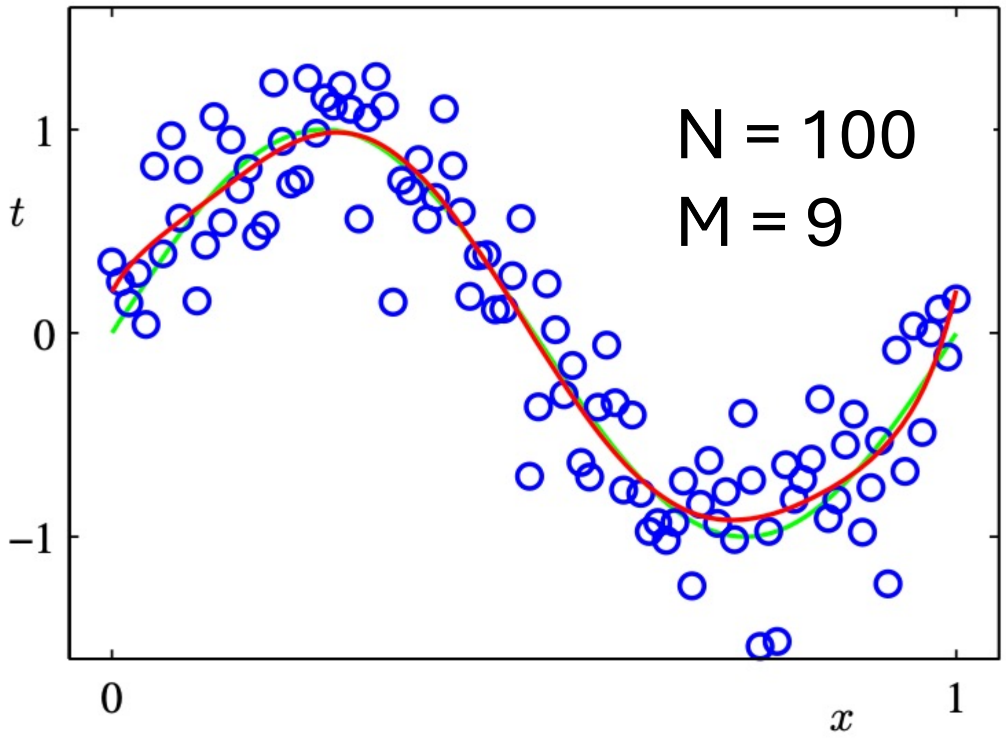


+ for M=0 and M=3, underfitting
  high error on both train and test set
+ M= 9, overfitting:
  large performance gap between train and test set

Overfitting becomes less severe as the size of the data set increases.

[from Bishop Figure] 1.6





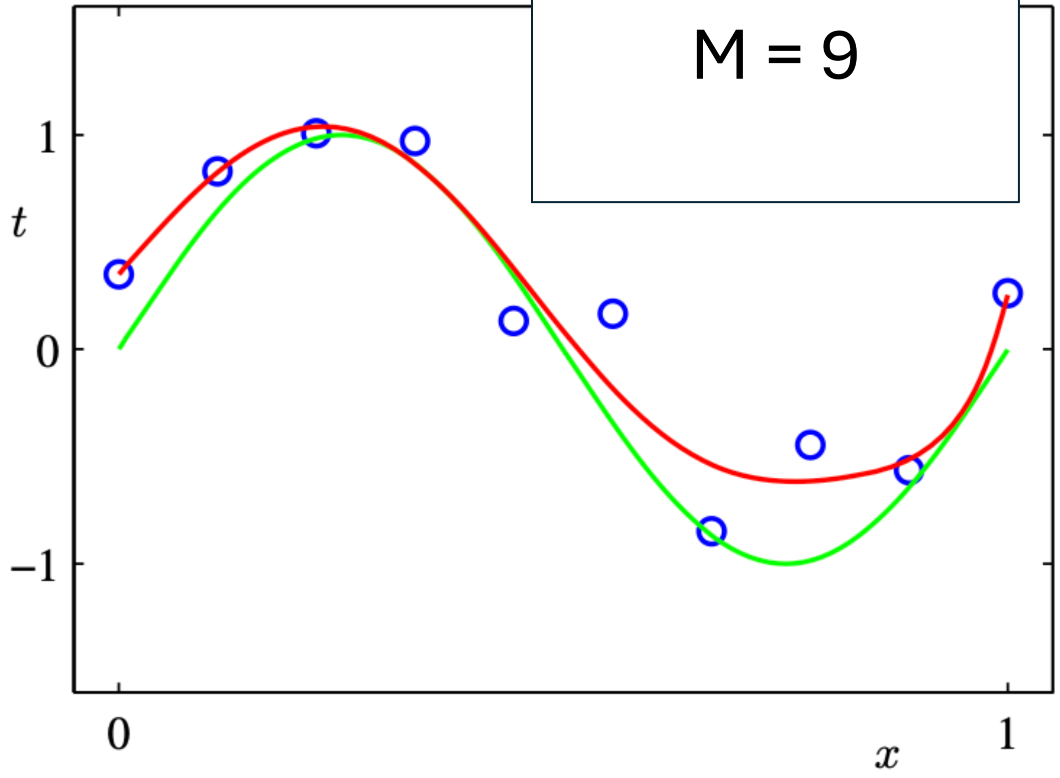
As we have the limited number of data samples,
the overfitting problem can be avoided by adopting Bayesian approach.
For example, regularized regression is the case.

$$\arg\min_{\vec{w}} ||\vec{y} - \Phi \cdot \vec{w}||^2 + \lambda^*(||\vec{w}||^2)$$ **“Ridge Regression”**

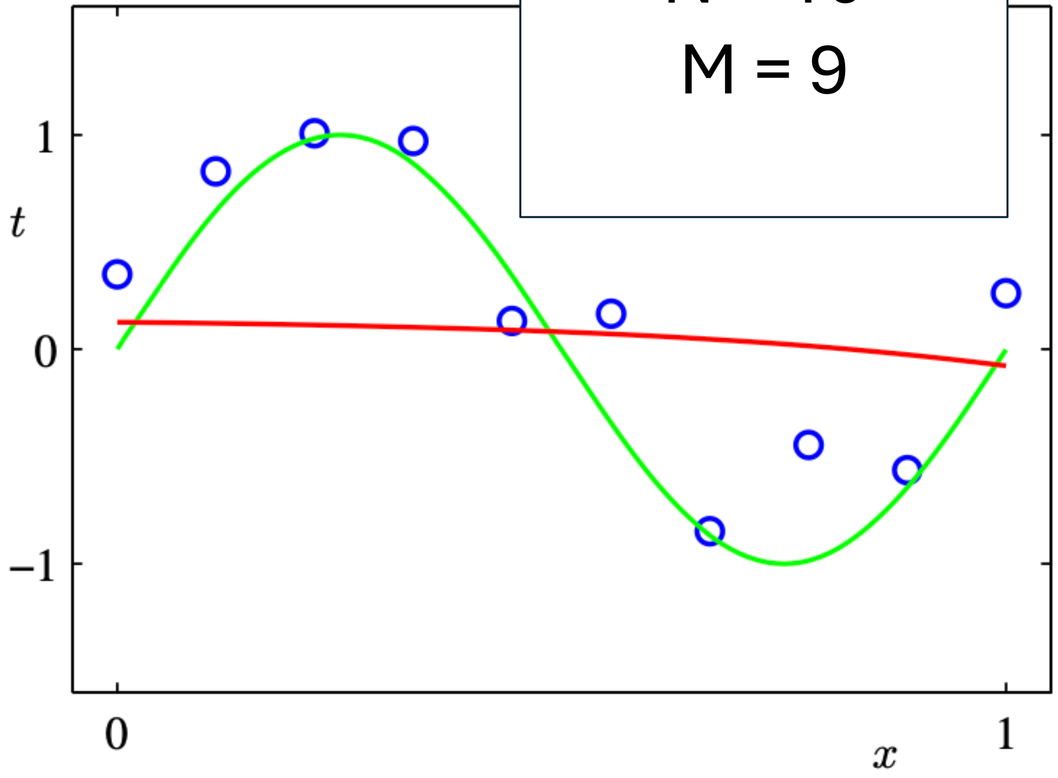
$$\arg\min_{\vec{w}} ||\vec{y} - \Phi \cdot \vec{w}||^2 + \lambda^*(||\vec{w}||)$$ “Lasso Regression”

Overfitting can be avoided by adopting Bayesian Approach

$ln\lambda = -18$
N = 10
M = 9

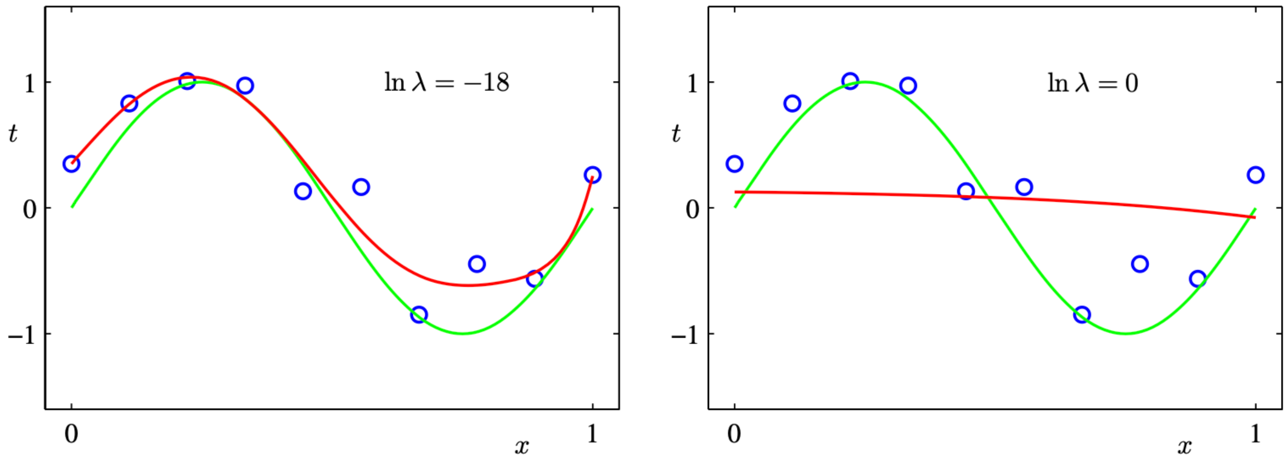


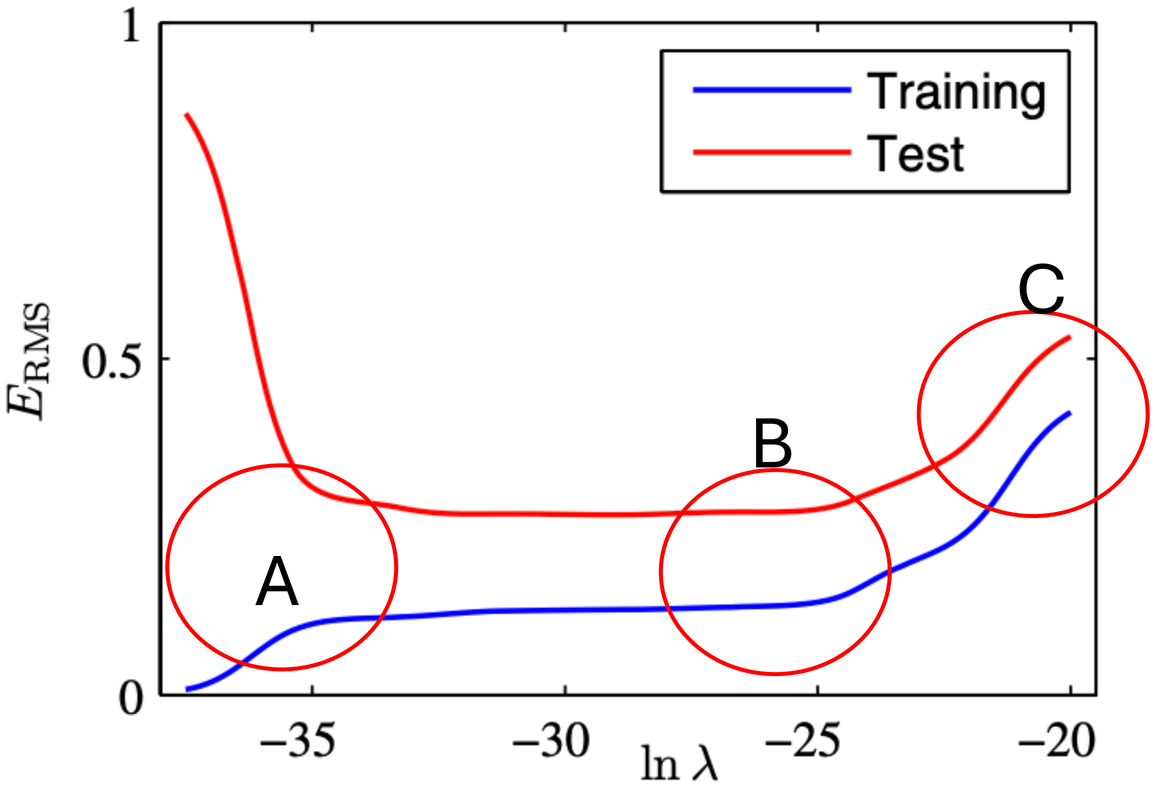
$ln\lambda = 0$
N = 10
M = 9



Ridge/Lasso regression regulates complexity.

Overfitting can be avoided by adopting Bayesian Approach





- Ridge regression regulates complexity.
- We need to choose regularization parameter:
- Q: which λ *would you choose A or B*?

\+ In the lecture, I mentioned that the performance gap between train and test should be considered for the selection lambda so the answer was B instead of A (both test performance are similar)

\+ But I realized that this may not be true (we only consider test error). Let me revisit this problem again and open for discussion in the next lecture.

It is not a good idea to tune our model based on test set!

Evaluation on Training Data

✓Underfitting

- poor performance on train set
- need to change feature map
  - changing hypothetical space
  - increasing model complexity

Evaluation on Test Data

✓Overfitting

- the performance gap between test and train set
- reducing model complexity (but be careful!)
- collect more data
- regularization.

Validation set
we need a hold-out set (a separate validation set)

- to finalize basis set
- to finalize feature map
- <u>to select regularization parameters</u>

Computing Empirical MSE error

Given data = $\{(x_1, t_1),\ (x_2, t_2), (x_3, t_3), \ldots, (x_N, t_N)\}$

Evaluation on Training	Validation	Evaluation on Test

✓Underfitting

✓Overfitting

- Adjusting feature map
- Adjusting basis functions
- Collecting more data
- Testing Regularization Parameters

- reporting test performance

S-Fold Cross Validation

$$L = \frac{1}{S}\sum_{i=1}^{S} L_s$$

The example for the case S = 4

S = 10 is common.

Summary

Quiz1: Sept. 30 1:00 PM

Class #1 Sept. 5:

- ML principles

Machine Learning Principles:

1. Define the target task:
   regression, classification, density estimation, learning latent information

2. Functional modeling:
   discriminative (generative), parametric (non-parametric), complexity decision

3. Data collection and feature extraction

4. Learning algorithms:
   empirical performance metric, batch or online, optimization methods

5. Evaluation
   underfitting and overfitting, # data points and model complexity

Class #2 Sept. 9:

- Bayes Rule
- Covariance Matrix of Random Vector

Bayes Rule in Machine Learning as an Inference Method

$$P(w|D) = \frac{p(w,D)}{P(D)} = \frac{p(D|w){\color{red}p(w)}}{p(D)}$$

Bayesian Probability (MAP)	Frequentist Probability (ML)
+ quantification uncertainty + prior density (expert knowledge)	+ relative frequency as # trials goes ∞ + w exists as a fixed point
Q: The chances of detecting life on Mars? Q: Which one is data sensitive (# of data, intrinsic noise in data)?	

Random Vector $\vec{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_D \end{bmatrix}$

- Mean vector: $E[\overrightarrow{X}] = \begin{bmatrix} E[X_1] \\ E[X_2] \\ \vdots \\ E[X_D] \end{bmatrix}$
- Covariance Matrix $\mathrm{Cov}[\boldsymbol{x}] \triangleq \mathbb{E}\left[(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])^{\mathsf{T}}\right] \triangleq \boldsymbol{\Sigma}$

$$= \begin{pmatrix} \mathbb{V}[X_1] & \mathrm{Cov}[X_1, X_2] & \cdots & \mathrm{Cov}[X_1, X_D] \\ \mathrm{Cov}[X_2, X_1] & \mathbb{V}[X_2] & \cdots & \mathrm{Cov}[X_2, X_D] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}[X_D, X_1] & \mathrm{Cov}[X_D, X_2] & \cdots & \mathbb{V}[X_D] \end{pmatrix}$$

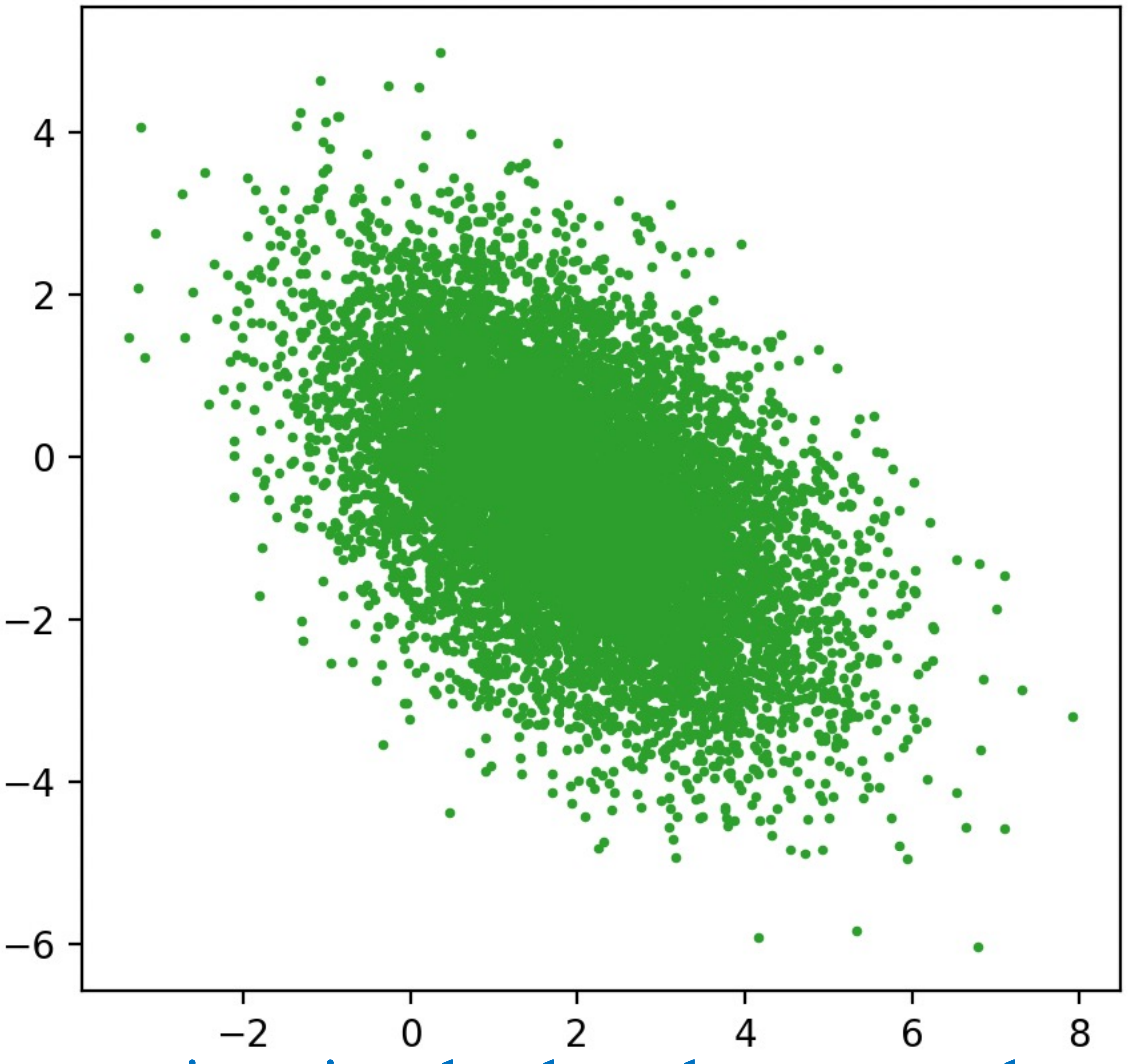
Q: The R.Vs are independent and identical how can we express the Covariance Matrix?
Q: How would you examine data shape as the data follows a certain covariance matrix?

Class #3 Sept. 12:

- Spectral Decomposition: the shape of data
- Designing R.V transformation

to earn a desired covariance matrix

Gaussian Samples & its Covariance Matrix



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

+ the covariance matrix shows
how the data is dispersed (which direction/ variation)

Q: Can you imagine the data shape as we have different eigenvector/ eigenvalue matrices?

Q: Can we design a new Gaussian random vector $Y \sim N(0, \Sigma^*)$ from $X \sim N(0, I)$?

- Y = AX

- COV (Y) =

- What A will be? + A = $E\ \Lambda^{1/2}$ *where* $\Sigma^* = E\ \Lambda^{1/2}\Lambda^{1/2}E^t$

Class #4 Sept. 16:

- PCA application: Compression and Whitening
- Non-linear Feature Mapping to accommodate linear modeling

PCA Applications

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

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

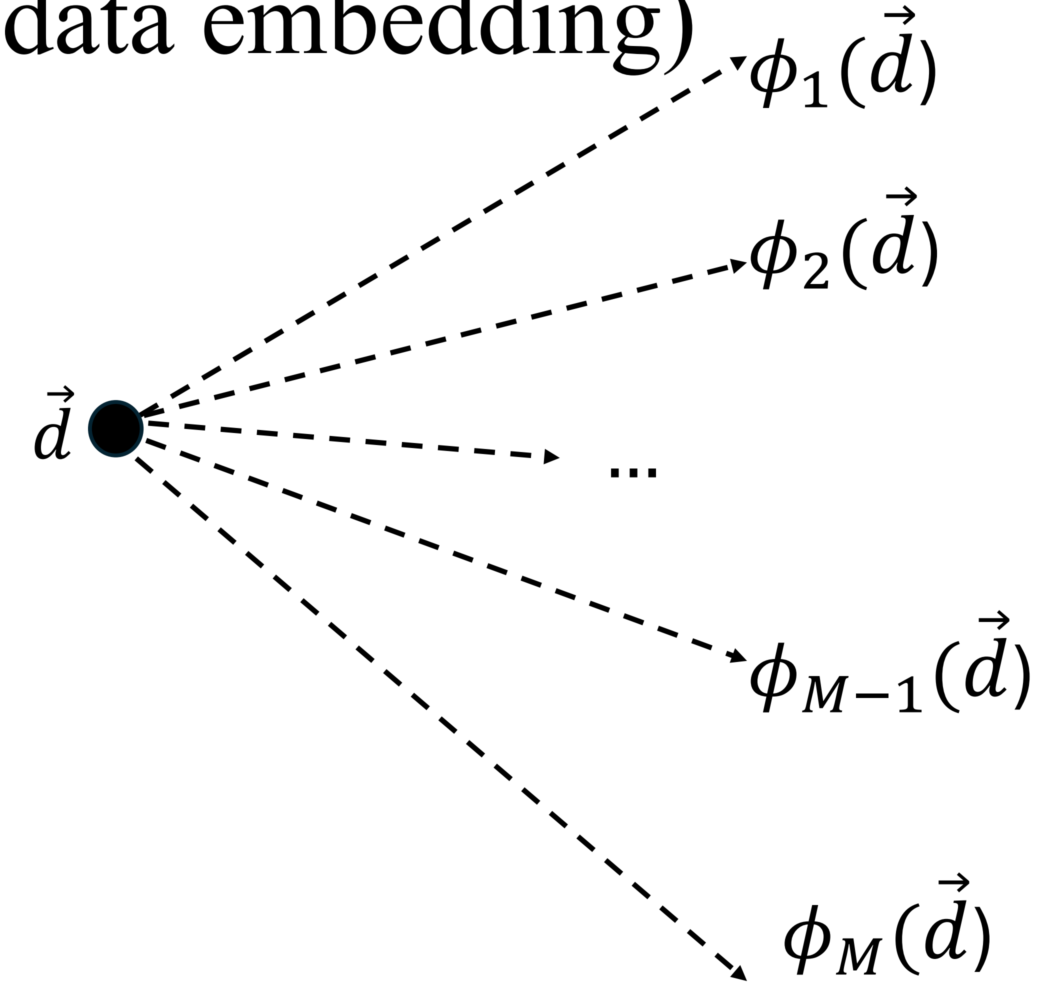
- Whitening & Dimensionality Reduction

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

Q: Can you define U_M and $\bar{x}$ to reduce the dimension of data space?

Q: What is the difference between $U_M^t(x_n - \bar{x})$ $vs.$ $\bar{x} + U_M U_M^t(x_n - \bar{x})$

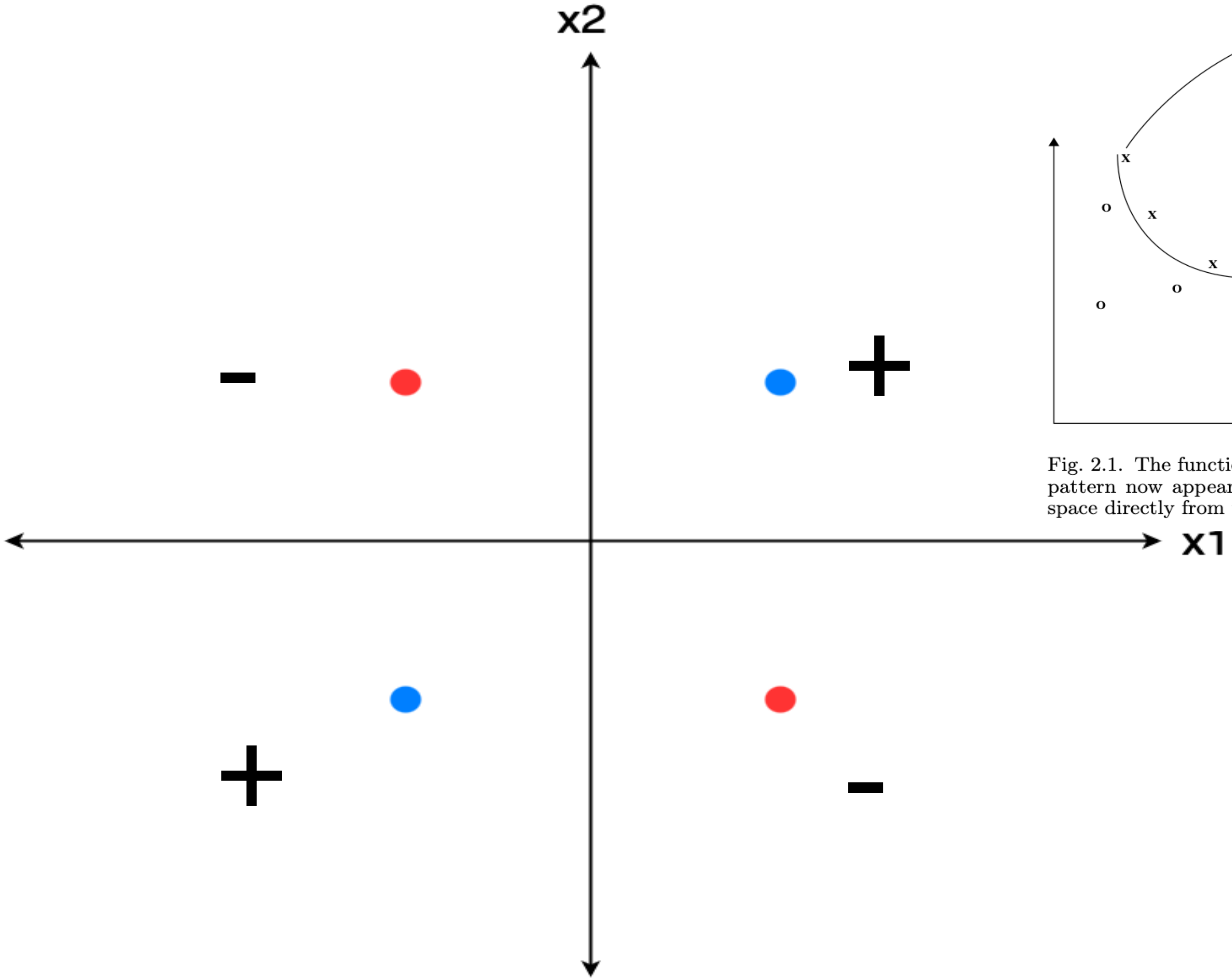
Feature Extraction is a mapping
$\phi\colon \overrightarrow{D} \rightarrow R^M$ (data embedding)



XOR Problem

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

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



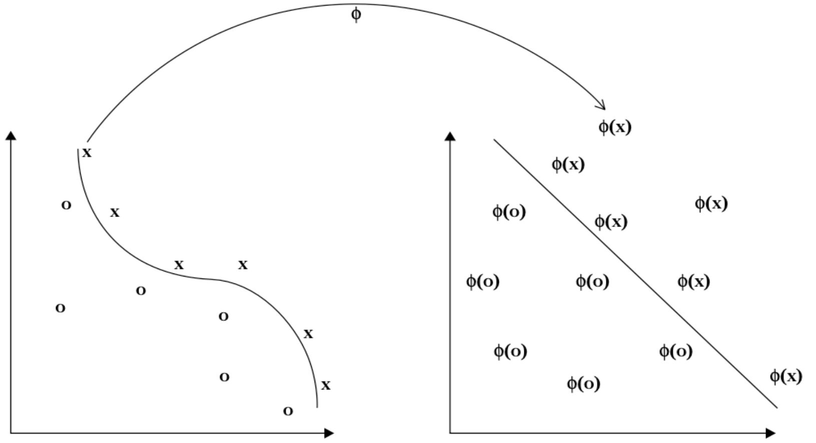


Fig. 2.1. The function ϕ embeds the data into a feature space where the nonlinear pattern now appears linear. The kernel computes inner products in the feature space directly from the inputs.

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

Class #5 Sept. 19:

- Linear Regression Problem
- Linear Regression Algorithm

Regression Problem

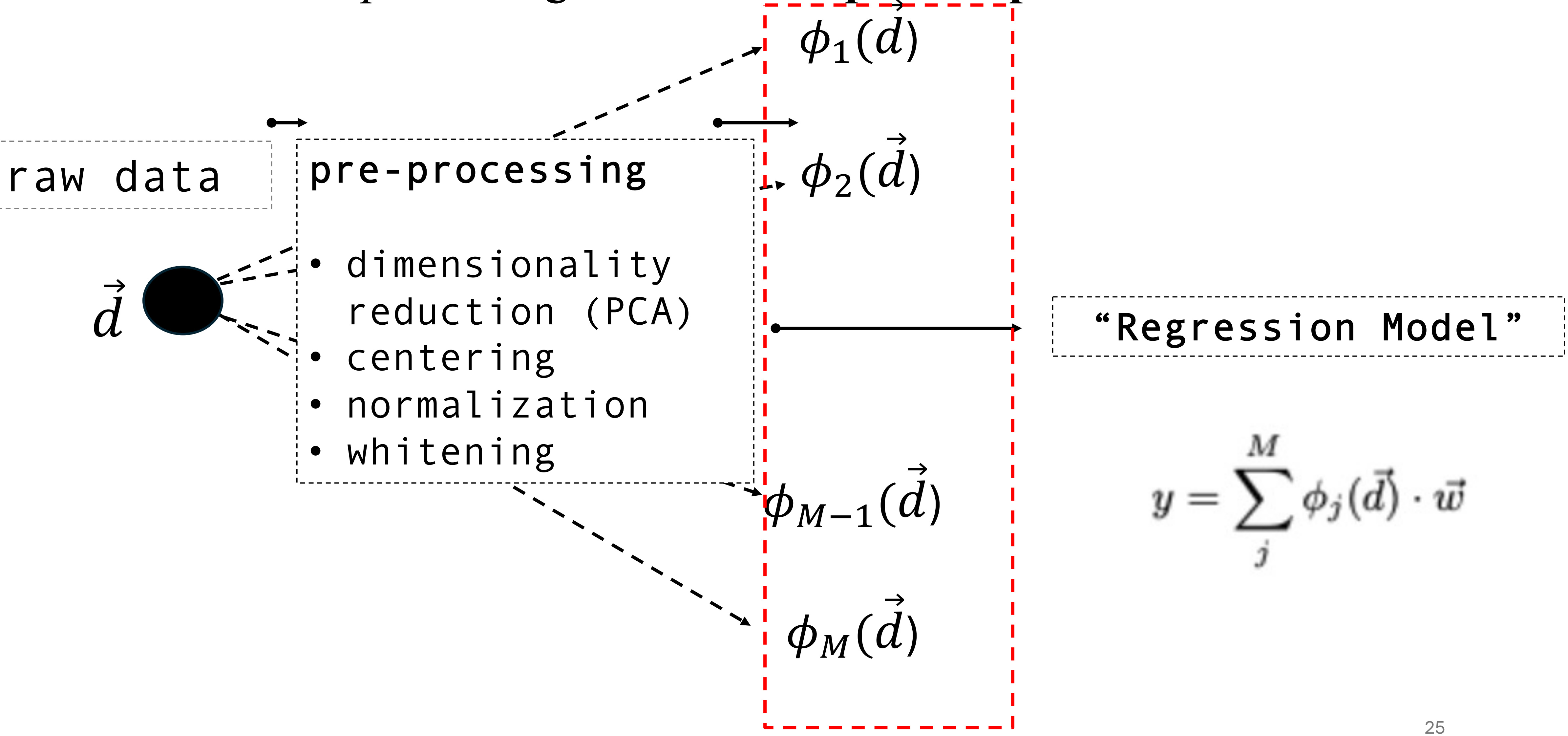
- Suppose we defined a proper feature map $\phi(\vec{d})$ for data point $(\vec{d}, y)$. then we can transform a data matrix D into Φ.
- We set the linear combination of $\phi(\vec{d})$ with $\vec{w}$ to predict the value y.

$$y = \Phi(\vec{d}) \cdot \vec{w} \quad + \varepsilon,\ \ \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

- We have observed data.
- We want to estimate $\vec{w}$

$$\begin{bmatrix} \phi_1\,(\mathrm{d1}) & \phi_2\,(\mathrm{d1}) & \dots & \phi_M\,(\mathrm{d1}) \\ \phi_1\,(\mathrm{d2}) & \phi_2\,(\mathrm{d2}) & \dots & \phi_M\,(\mathrm{d2}) \\ \dots & & & \\ \dots & & & \\ \dots & & & \\ \dots & & & \\ \phi_1\,(\mathrm{dN}) & \phi_2\,(\mathrm{dN}) & \dots & \phi_M\,(\mathrm{dN}) \end{bmatrix} \times \begin{bmatrix} W_1 \\ W_2 \\ \dots \\ \\ W_M \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ \\ \\ \\ y_n \end{bmatrix}$$

Raw data ⟶ Pre-processing ⟶ **Feature Space Expansion with Basis**



Regression Problem: Estimation Problem

we have observations: data Φ ($N \times M$) and $\vec{y}$

- $w^* = argmax\ p(\vec{y}|\vec{w}, \Phi)$: (MLE)

  $= argmax\ \mathcal{N}_y\left(\Phi(\vec{d}) \cdot \vec{w}\ \ , \sigma^2 \mathrm{I}\right)$

Ground Truth (data)

Prediction!

$$\arg\min_w ||\vec{y} - \Phi \cdot \vec{w}||^2$$

$$\boxed{\arg\min_w ||\vec{y} - \Phi \cdot \vec{w}||^2}$$

MLE becomes
Minimum Mean Square Error Problem

$$J(\vec{w}) = ||\vec{y} - \Phi \cdot \vec{w}||^2$$

$$J(\vec{w}) = (\vec{y}^t - \vec{w}^t \cdot \Phi^t) \cdot (\vec{y} - \Phi \cdot \vec{w})$$

$$\nabla J(\vec{w}) = -2 \cdot \Phi^t \cdot (\vec{y} - \Phi \cdot \vec{w}) = 0$$

$$\boxed{\Phi^t \cdot \Phi \cdot \vec{w} = \Phi^t \cdot \vec{y}}$$

Normal Equation

Q: How would you write normal equation and data matrix as data points and feature map is given?

Q: Can you express the optimal $\overrightarrow{w *}$ when $SVD\ of\ \Phi = \mathrm{E}\Lambda^{1/2}V^t\ \ is\ given$?

Class #6 Sept. 23 :

- Error Decomposition: Trade off between Variance and Bias

- Error Decomposition

$$E[L] = \int_x \int_t \{y(x;D) - t)\}^2 f(t|x) f(x)\, \mathrm{d}t\, \mathrm{d}x$$

$$E[L] = \int_x \int_t (y(x;D) - h(x) + h(x) - t)^2 f(t|x) f(x)\, \mathrm{d}t\, \mathrm{d}x$$

$$= \int_x \int_t \underbrace{(y(x;D) - h(x))^2}_{} + \underbrace{(h(x) - t)^2}_{\epsilon} + \underbrace{2(y(x;D) - h(x))(h(x) - t)}_{\text{zero}}\, \mathrm{d}t\, \mathrm{d}x$$

$$\int_x \int_t (y(x;D) - h(x))^2 f(t|x) f(x)\, \mathrm{d}t\, \mathrm{d}x$$

$$\int_x (y(x;D) - E_D[y(x;D)] + E_D[y(x;D)] - h(x))^2 f(x)\, \mathrm{d}x$$

$$\int_x \underbrace{(y(x;D) - E_D[y(x;D)])^2}_{\text{Variance}} + \underbrace{(E_D[y(x;D)] - h(x))^2}_{\text{Bias}} f(x)\, \mathrm{d}x$$

- Error Decomposition $E[L] = \underbrace{\text{Variance} + \text{Bias}}_{\text{Trade-off Relation}} + \text{Intrinstic Error}$

- Intrinsic Error: $\int_x \int_t (E[T|x] - t)^2 f(t|x) f(x)\, \mathrm{d}t\, \mathrm{d}x$
- Variance: $\int_x VAR_D[y(x;D)] f(x)\, \mathrm{d}x$
- Bias: $\int_x \{E_D[y(x;D)] - E[T|x]\}^2 f(x)\, \mathrm{d}x$

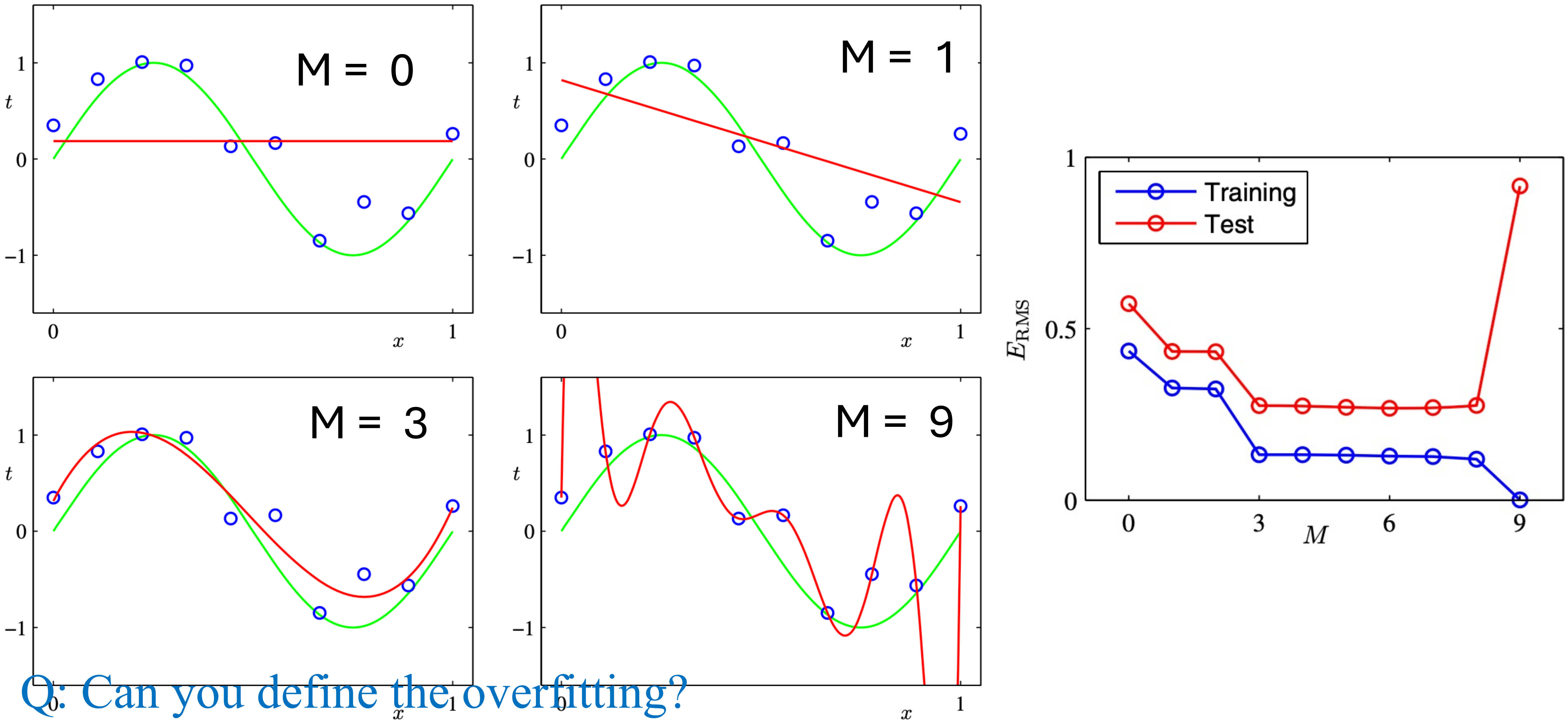
- Trade-Off between Variance & Bais

  Complex models : High Variance but Low Bias

  Simple model : Low Variance but High Bias

Class #7 Sept. 26 :
Generalization Performance in relation to complexity and #data points
Please Read Bishop 1.1

Underfitting and Overfitting Example [from Bishop Figure 1.4 and 1.5]

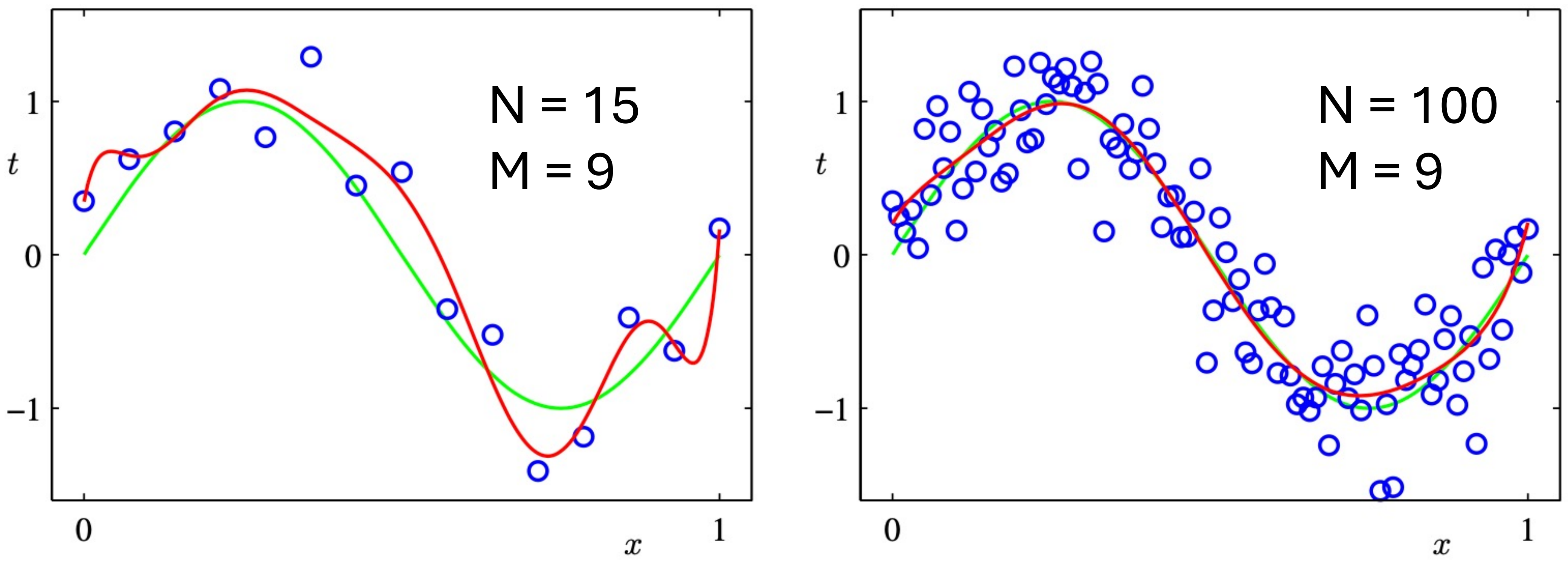


Q: Can you define the overfitting?

Q: What is the phenomenon as a model is experiencing overfitting?

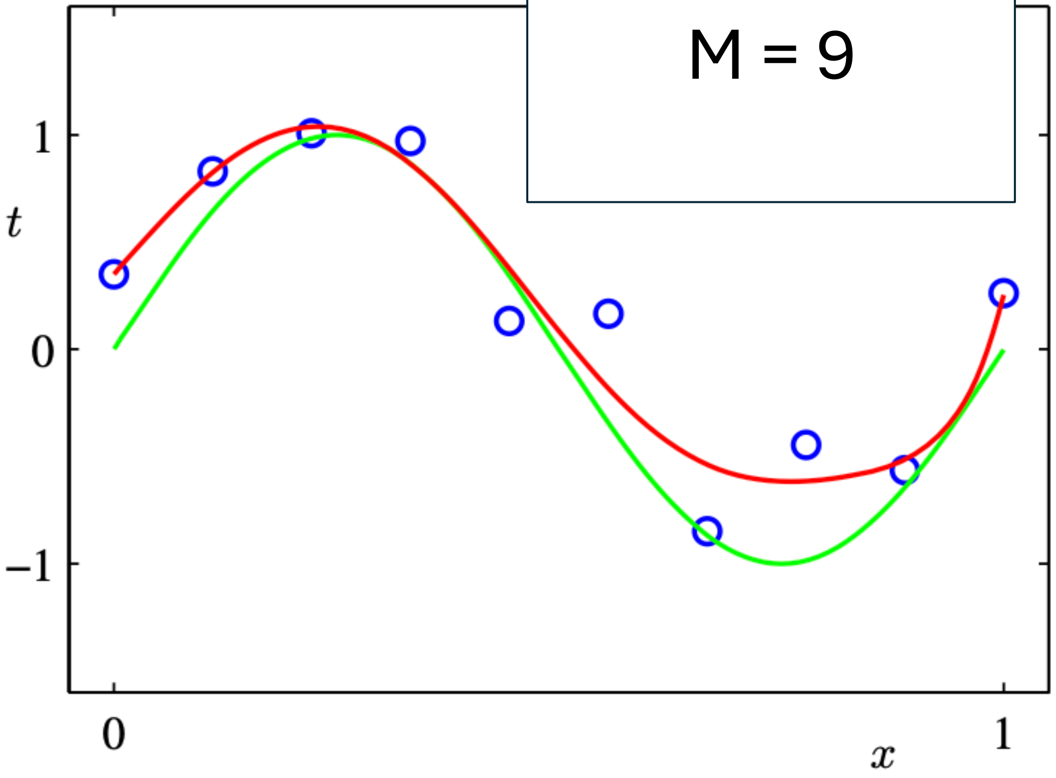
Overfitting becomes less severe as the size of the data set increases.

[from Bishop Figure] 1.6

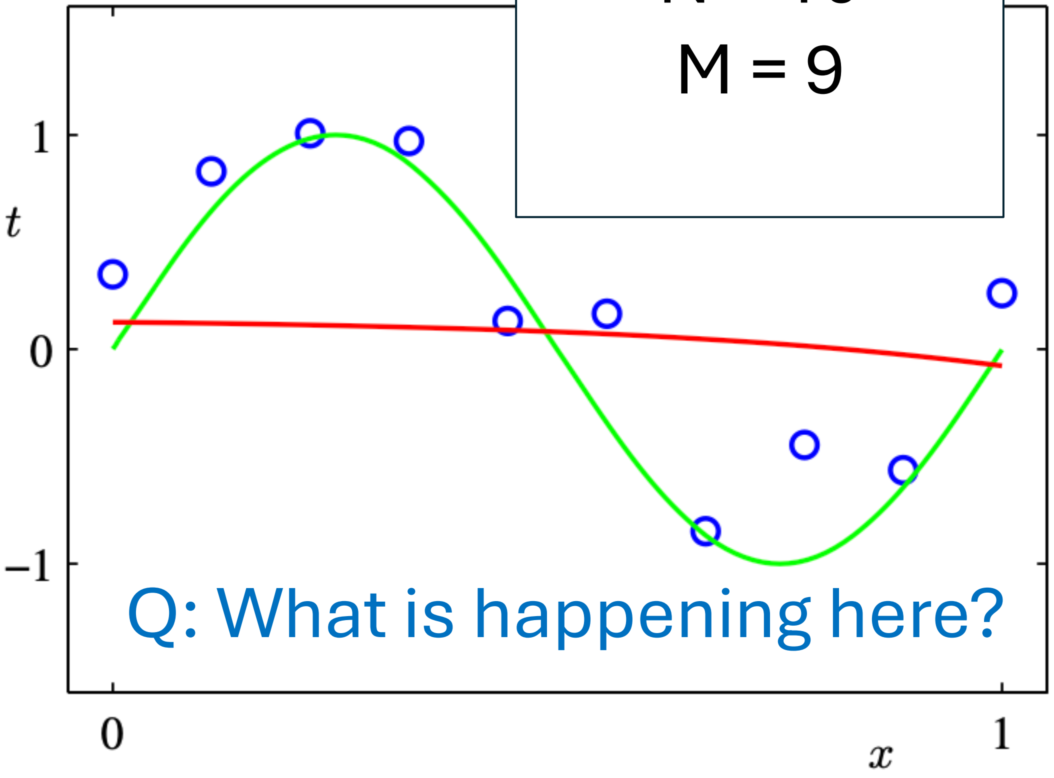


Overfitting can be avoided by adopting Bayesian Approach

$ln\lambda = -18$
N = 10
M = 9



$ln\lambda = 0$
N = 10
M = 9



Q: What is happening here?

Ridge Regression regulates complexity.

Q: How Ridge Regression helps to avoid overfitting?