Machine Learning Principles

Class6 : Sept. 22

Linear Regression III

Instructor: Diana Kim

Today's Lecture

1. Convex Optimization Theory

    - necessary & sufficient condition for optimality

    - equality constraint problem

    - inequality constraint problem

2. Regularization

    - three interpretations of MMSE with regularization

    - setting a right regularization parameter $\lambda*$: cross validation

3. Overfitting and Underfitting

- Optimization Theory:

  Solving a convex optimization problem by using a Langrangian function

$$x^* = \min_{g(x) \le 0} f(x)$$

[1] Local and Global Minimum (why optimization theory?)

In an ML problem, we need to solve an optimization problem, finding local / global minimum (suboptimal/optimal): MLE / MAP

- regression with constraint (regularization)

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

- regression without constraint

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

$$\text{subject to} \quad ||\vec{w}||^2 \leq C$$

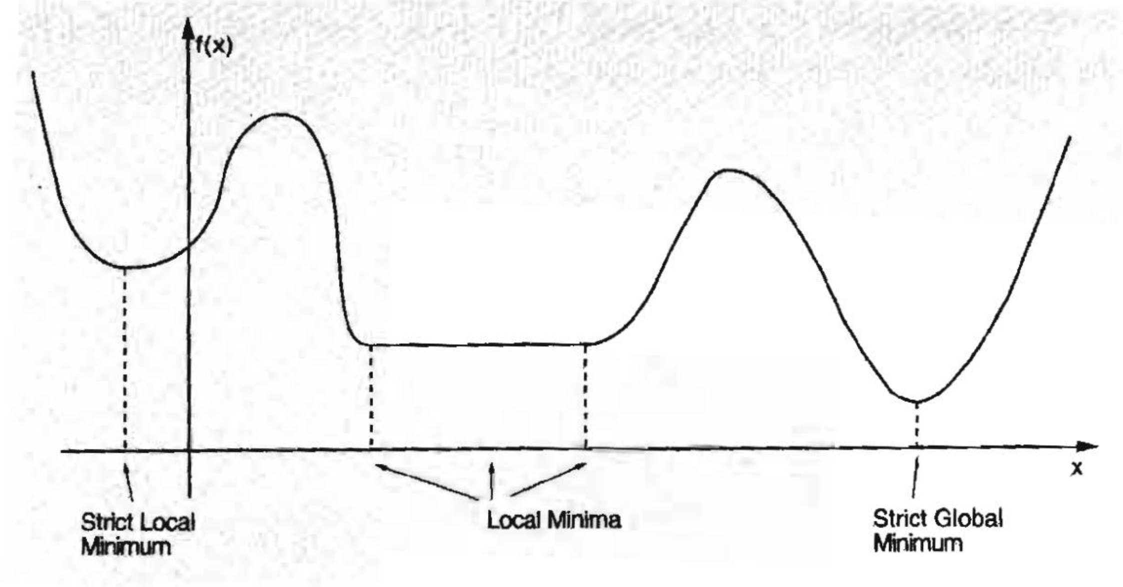
[2] Local and Global Minimum

Local Minimum x^*

$$f(x*) \leq f(x), \quad \exists\epsilon \quad s.t \quad ||x - x*|| < \epsilon \quad \forall x$$

Global Minimum x^*

$$f(x*) \leq f(x) \quad \forall x$$



[3] Local and Global Minimum (**necessary conditions** for local minimum)

- By Taylor series

if x^* is a local optimal,
then the Taylor approximation is non-negative:

$$f(x*+\Delta x) - f(x*) \approx \nabla f(x*)^t \Delta x + \frac{1}{2}\Delta x^t \nabla^2 f(x*)\Delta x \geq 0$$

[5] Local and Global Minimum (Gradient Vector)

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \frac{\partial f(x)}{\partial x_3} \\ \cdots \end{bmatrix}$$

Gradient of $f(x_1, x_2, x_3 \ldots) = y \quad \in \mathbb{R}$

- this is a vector.
- the collection of the first partial derivatives
- the direction of the greatest change of a scalar function

[6] Local and Global Minimum (Hessian Matrix)

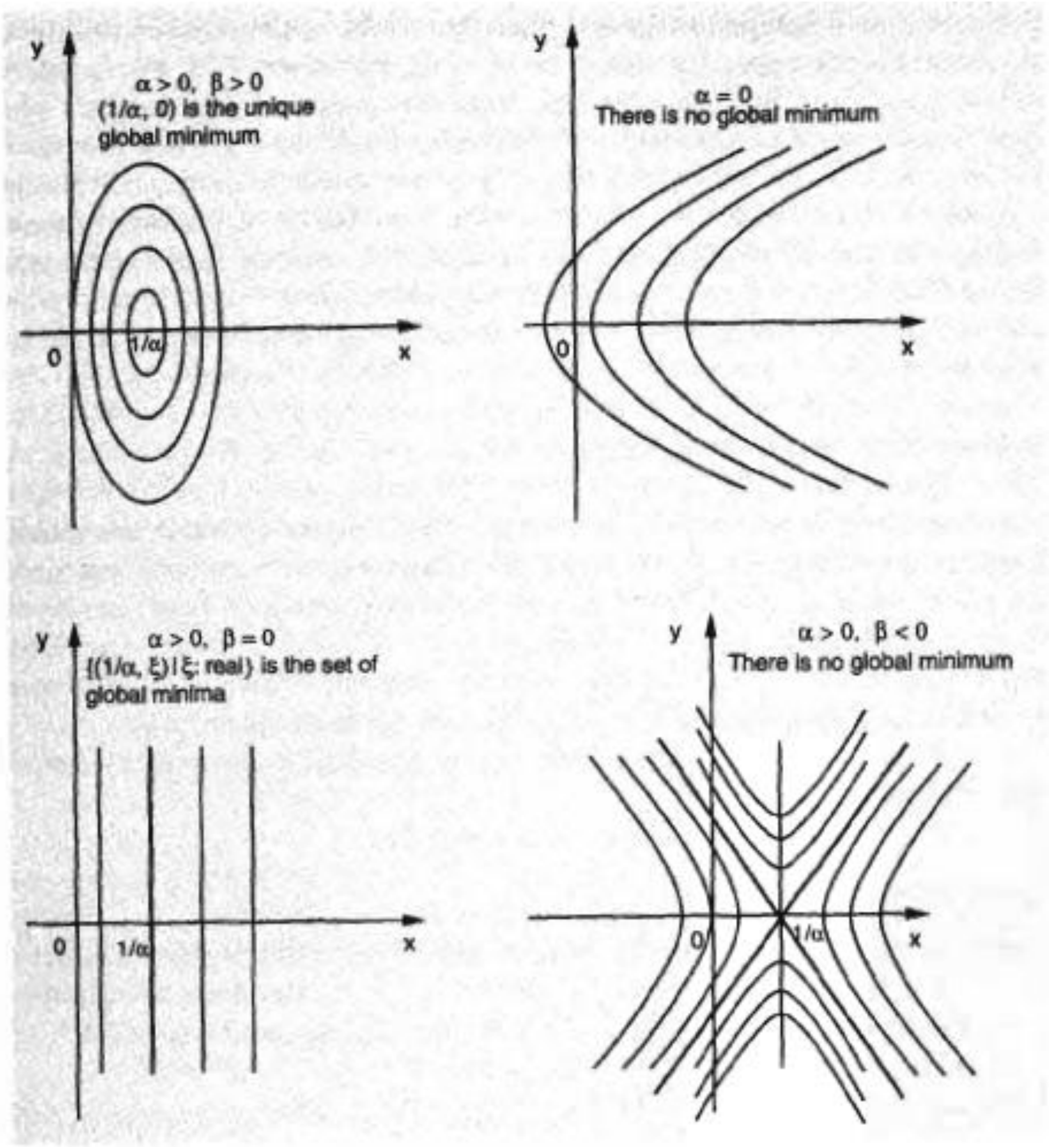
$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial^2 x_1} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_3} & \cdots \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial^2 x_2} & \frac{\partial^2 f(x)}{\partial x_2 \partial x_3} & \cdots \\ \frac{\partial^2 f(x)}{\partial x_3 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_3 \partial x_2} & \frac{\partial^2 f(x)}{\partial x_3 \partial x_3} & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

Hessian Matrix of $f(x_1, x_2, x_3 \ldots) = y \in \mathbb{R}$

- this is a matrix.
- the collection of all second partial derivatives
- Concavity of a function at the point x^*
- $\lambda_{min} \leq \Delta x^t \nabla^2 f(x *) \Delta x \leq \lambda_{max}$ where $||\Delta x|| = 1$
- The eigenvalues provide the concavity of principal axes (maximum/ minimum)

[7] Local and Global Minimum (Hessian Matrix)

$$f(x,y) = \frac{1}{2}\left(\alpha x^2 + \beta y^2\right) - x$$



- Equality Constraint Problem

[1] Equality Constraint Problem (example)

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & h_i(x) = 0 \quad i = 1, ..., m \end{aligned}$$

ex] $$\begin{aligned} \min_{x} \quad & x_1 + x_2 \\ \text{s.t.} \quad & x_1^2 + x_2^2 = 2 \end{aligned}$$

[2] Equality Constraint Problem (necessary conditions)

$$\min_{x} \quad f(x)$$

$$\text{s.t.} \quad h_i(x) = 0 \quad i = 1, ..., m$$

- **Condition1**: let x^* be a local minimum of f s.t $h_i(x) = 0$ and $\nabla h_i(x^*) \ldots \nabla h_i(x^*)$ are linearly independent. then, there exist a unique vector $\lambda^* = (\lambda_1^*, \lambda_2^*, \ldots. \lambda_m^*)$ s.t

$$\nabla f(x*) + \sum_{i=1}^{m} \lambda_i^* \nabla h_i(x*) = 0$$

- **Condition2**: $y^t \{\nabla^2 f(x*) + \sum_{i=1}^{n} \lambda_i^* \nabla^2 h_i(x*))\} y \geq 0 \quad y \in V(x*)$

$$V(x*) = \{y | \nabla h_i(x*)^t y = 0 \quad \forall i = 1, ..., m\}$$

[3] Equality Constraint Problem (Lagrangian Multiplier Theorem)

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & h_i(x) = 0 \quad i = 1, \ldots, m \end{aligned}$$

- Condition1: let x^* be a local minimum of f s.t $h_i(x) = 0$ and $\nabla h_i(x^*) \ldots \nabla h_i(x^*)$ are linearly independent. then, there exist a unique vector $\lambda^* = (\lambda_1^*, \lambda_2^*, \ldots \lambda_m^*)$ s.t

$$\nabla f(x*) + \sum_{i=1}^{m} \lambda_i^* \nabla h_i(x*) = 0$$

- Condition2: $y^t\{\nabla^2 f(x*) + \sum_{i=1}^{m} \lambda_i^* \nabla^2 h_i(x*))\}y \geq 0 \quad y \in V(x*)$

$$V(x*) = \{y | \nabla h_i(x*)^t y = 0 \quad \forall i = 1, \ldots, m\}$$

Q: What if we define a new function? $L(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i h_i(x)$

[4] Equality Constraint Problem (Lagrangian function)

- Lagrangian function/ unconstrained function

$$L(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i \, h_i(x)$$

- two necessary optimality conditions for $L(x, \lambda)$

$$\left\{ \begin{array}{l} \nabla f(x*) + \sum_{i=1}^{m} \lambda_i^* \nabla h_i(x*) = 0 \qquad h_i(x*) = 0 \quad \forall i = 1, 2..., m \\ \\ y^t \{\nabla^2 f(x*) + \sum_{i=1}^{m} \lambda_i^* \nabla^2 h_i(x*))\} y \geq 0 \quad y \in V(x*) \end{array} \right.$$

$$V(x*) = \{y | \nabla h_i(x*)^t y = 0 \quad \forall i = 1, ..., m\}$$

[5] Equality Constraint Problem (Lagrangian function)

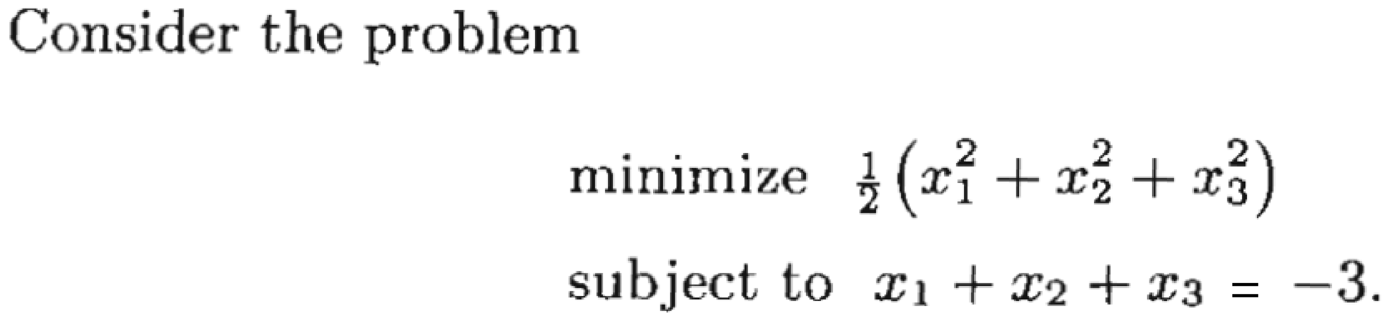
[Lagrangian Function/ unconstrained function]

$$L(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i h_i(x)$$

The necessary conditions for optimal points for the unconstrained function (Lagrangain) is equivalent to the constrained primary problem.

Therefore, we could fine optimal solutions for the primal problem by solving the necessary conditions on Lagrangian function $L(x, \lambda)$

[6] Equality Constraint Problem (Lagrangian example)



Consider the problem

$$\begin{aligned} &\text{minimize} && \tfrac{1}{2}\left(x_1^2 + x_2^2 + x_3^2\right) \\ &\text{subject to} && x_1 + x_2 + x_3 = -3. \end{aligned}$$

Q: Lagrangian function?

- Inequality Constraint Problem

[1] Inequality Constraint Problem (necessary conditions)

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0 \quad i = 1, \ldots, m \end{aligned}$$

* pay attention! the direction of inequality!

[2] Inequality Constraint Problem (necessary conditions)

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0 \quad i = 1, ..., m \end{aligned}$$

- two possible cases for x^*

(1) x^* inside of $g_i(x) < 0$

(2) x^* on the boundary of $g_i(x) = 0$

X

[3] Inequality Constraint Problem (necessary conditions)

$$\min_{x} \quad f(x)$$

$$\text{s.t.} \quad g_i(x) \leq 0 \quad i = 1, \ldots, m$$

- two possible cases and their necessary conditions

(1) x^* inside of the manifold by $g_j(x) < 0$

$$\rightarrow \nabla f(x*) = 0$$

(2) x^* on the boundary of $g_h(x) = 0$ there exist λ_h

X

$$\rightarrow \nabla f(x*) + \lambda_h \nabla g_h(x*) = 0$$

Q: sign λ_h ?

[4] Inequality Constraint Problem (necessary conditions)

[5] Inequality Constraint Problem (KKT necessary conditions)

[**K**arush–**K**uhn–**T**ucker conditions]

Let x^*be a local minimum of the problem

$$\min_x \quad f(x)$$

$$\text{s.t.} \quad g_i(x) \leq 0 \quad i = 1, \dots, m$$

important to set the inequality this form (less than or equal to)!

Then, there exist $\lambda_i, \ i = 1, \dots, m$ such that

(1) $$\nabla f(x*) + \sum_{i=1}^{m} \lambda_i \nabla g_m(x*) = 0$$

- stationary condition

- complementary slackness condition $\lambda_i \cdot g_i(x^*) = 0$

(2) $$\begin{cases} \lambda_j \geq 0 & j = 1, \dots, r \\ \lambda_j = 0 & \forall j \notin A(x*) \end{cases}$$

$A(x^*)$ is the set of active constraints at x^*

[6] Inequality Constraint Problem (KKT necessary conditions)

Let x^*be a local minimum of the problem

$$\begin{aligned} &\min_{x} \quad f(x) \\ &\text{s.t.} \quad g_i(x) \leq 0 \quad i = 1, \ldots, m \end{aligned}$$

Then, there exist $\lambda_i,\ i = 1, \ldots, m$ such that

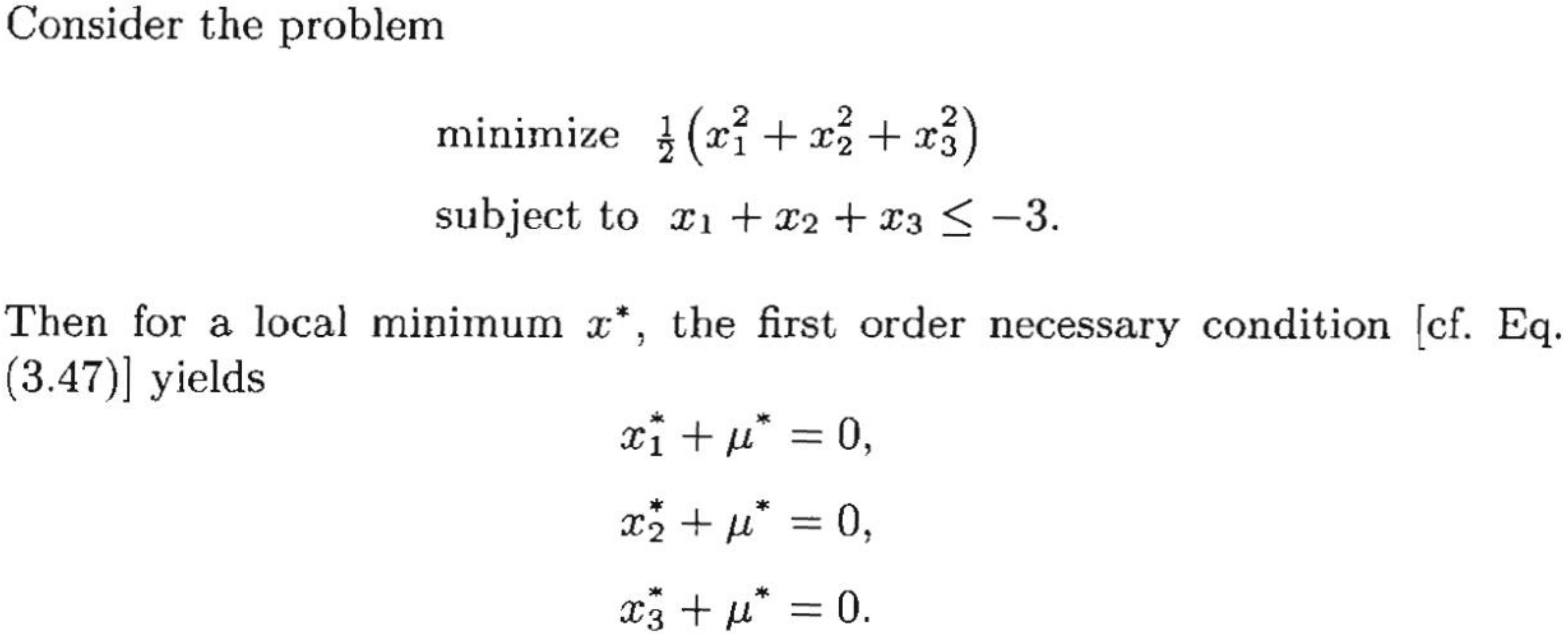
$$(1) \quad \nabla f(x*) + \sum^{m} \lambda_i \nabla g_m(x*) = 0$$

$$(2) \quad \begin{cases} \lambda_j \geq 0 & j = 1, \ldots, r \\ \lambda_j = 0 & \forall j \notin A(x*) \end{cases}$$

$$(3) \quad g(x^*) \leq 0$$

- Primary feasibility

[7] Inequality Constraint Problem (example 1)



Consider the problem

$$\begin{aligned} &\text{minimize} \quad \tfrac{1}{2}\left(x_1^2 + x_2^2 + x_3^2\right) \\ &\text{subject to} \quad x_1 + x_2 + x_3 \leq -3. \end{aligned}$$

Then for a local minimum x^*, the first order necessary condition [cf. Eq. (3.47)] yields

$$x_1^* + \mu^* = 0,$$

$$x_2^* + \mu^* = 0,$$

$$x_3^* + \mu^* = 0.$$

From Nonlinear Programming, Bertsekas Example 3.3.1

- Regularization as an optimization problem

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

[1] Regularization by an optimization problem

In a ML problem, we need to solve an optimization problem, finding local / global minimum (suboptimal/optimal).

- regression <u>without</u> constraint

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

- regression <u>with</u> constraint (regularization)

$$\arg\min_{\vec{w}} ||\vec{y} - \Phi \cdot \vec{w}||^2$$
$$\text{subject to} \quad ||\vec{w}||^2 \leq C$$

[2] Regularization by an optimization problem (Lagrangian form)

$$\begin{aligned}&\arg\min_{\vec{w}} ||\vec{y} - \Phi \cdot \vec{w}||^2 \\ &\quad \text{subject to} \quad ||\vec{w}||^2 \leq C\end{aligned} \quad \longleftrightarrow \quad \arg\min_{\vec{w}} ||\vec{y} - \Phi \cdot \vec{w}||^2 + \lambda^* (||\vec{w}||^2 - C)$$

- $C \infty \frac{1}{\lambda*}$ “but does not need to know the exact value of C”
- an example problem will be discussed in recitation #3 Sept. 24th.

[3] Regularization by an optimization problem (Lagrangian form)

$$\begin{aligned} &\arg\min_{\vec{w}} ||\vec{y} - \Phi \cdot \vec{w}||^2 \\ &\quad \text{subject to} \quad ||\vec{w}||^2 \leq C \end{aligned} \qquad \longleftrightarrow \qquad \arg\min_{\vec{w}} ||\vec{y} - \Phi \cdot \vec{w}||^2 + \lambda^* (||\vec{w}||^2 - C)$$

- according to C we define,
  optimal Lagrangian $\lambda *$ will be different!

- constant addition/subtraction won't change x^*

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

[4] Regularization by an optimization problem (Lagrangian form)

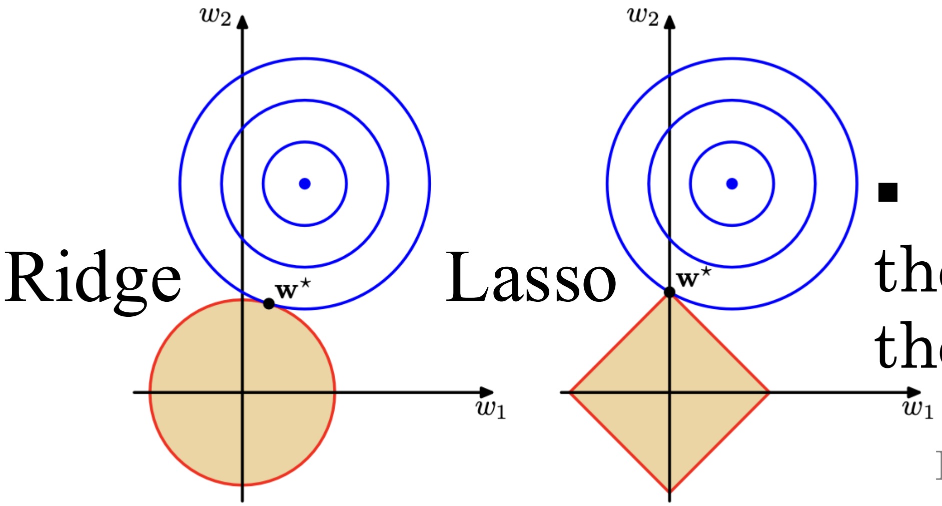
$$\left\{\begin{array}{l} \arg\min\limits_{\vec{w}} ||\vec{y} - \Phi \cdot \vec{w}||^2 \\ \quad \text{subject to} \quad ||\vec{w}||^2 \leq C \\ \\ \\ \arg\min\limits_{\vec{w}} ||\vec{y} - \Phi \cdot \vec{w}||^2 + \lambda^*(||\vec{w}||^2) \end{array}\right.$$

- in regularized regression learning, we will change λ^* and test its performance to find a good λ^* (empirically)

[5] Regularization by an optimization problem (Ridge & Lasso)

$$\arg\min_{\vec{w}} ||\vec{y} - \Phi \cdot \vec{w}||^2 + \lambda^*(||\vec{w}||^2)$$ [Ridge regularization]

$$\arg\min_{\vec{w}} ||\vec{y} - \Phi \cdot \vec{w}||^2 + \lambda^*(||\vec{w}||)$$ [Lasso regularization]



- the constraints regulate the magnitude of *w* (parameters), the model complexity. Lasso gives a sparse solution.

From Bishop Chap Figure 3.4

- How could we set $\lambda*$? (controlling C)

[1] selection of $\lambda *$

- we have data points $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_N, y_N)\}$
- split the data into train/ test
- set basis functions (polynomial / Gaussian)

[2] selection of $\lambda *$

- we have data points $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_N, y_N)\}$
- split the data into train/ validation/ test
- set hypothetical basis functions (polynomial / Gaussian)

[train]

- PCA of train data for dim-reduction and whitening.
  (save the PCA blocks computed using training set)
- set the normal equations for the various $\lambda *$ to estimate w

[validation]

- PCA of validation data and whitening with the same block in training.
- compute error between the ground truth y and prediction y': $||y - y'||^2$
- choose the best $\lambda *$

[3] selection of $\lambda *$ (validation set)

Validation set is a separate hold out set.
One example ratio = train: validation: test = 80: 10 : 10

- to select regularization parameters
- to finalize basis set
- to finalize feature map (# features dim)

[**recall] Linear Regression by MAP (solving the optimization problem)

$$
\begin{aligned}
\nabla J(W) &= -2\Phi^t \cdot \vec{Y} + 2\Phi^t \cdot \Phi \cdot \vec{W} + 2\lambda * \cdot \vec{W} = 0 \\
&\leftrightarrow \Phi^t \cdot \Phi \cdot \vec{W} + \lambda * \cdot \vec{W} = \Phi^t \cdot \vec{Y} \\
&\leftrightarrow V \begin{bmatrix} \lambda_1 + \lambda * & 0 & ... & 0 \\ 0 & \lambda_2 + \lambda * & ... & 0 \\ ... & ... & ... & ... \\ 0 & ... & ... & \lambda_m + \lambda * \end{bmatrix} V^t \cdot \vec{W} = V^t \lambda^{1/2} E^t \vec{Y} \\
&\leftrightarrow \vec{W} = V \begin{bmatrix} \frac{1}{\lambda_1 + \lambda *} & 0 & ... & 0 \\ 0 & \frac{1}{\lambda_2 + \lambda *} & ... & 0 \\ ... & ... & ... & ... \\ 0 & ... & ... & \frac{1}{\lambda_m + \lambda *} \end{bmatrix} E^t \vec{Y}
\end{aligned}
$$

[by the λ^* we can avoid the case parameters gets too large.]

[4] selection of $\lambda *$

[train]

- PCA of train data for dim-reduction and whitening.
  (save the PCA blocks computed using training set)
- set normal equation with various $\lambda *$ to estimate w

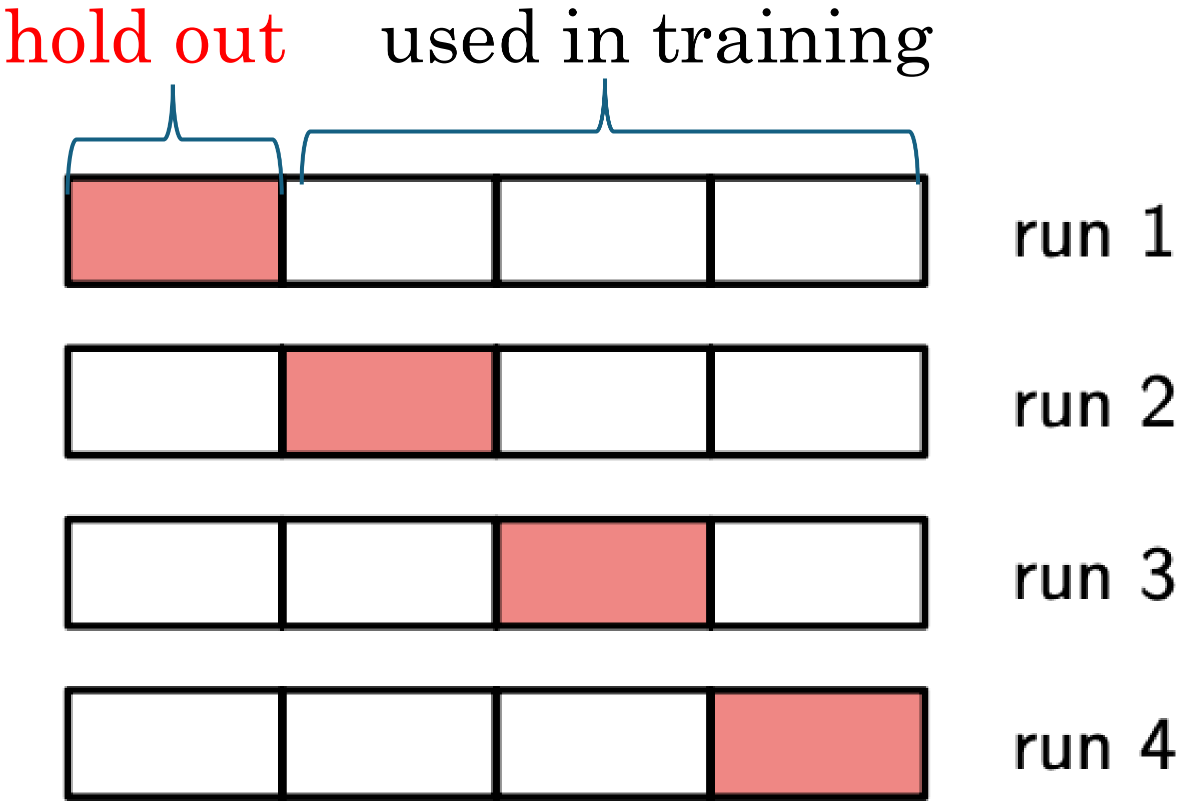
[validation]

- PCA of validation data and whitening with the same block in training.
- compute error between the groud turth y and prediction y': $||y - y'||^2$
- Choose the best $\lambda *$

[test: report the final test results!]

- PCA of test data and whitening with the same block in
- compute error between the groud turth y and prediction y': $||y - y'||^2$

[5] selection of $\lambda*$ (cross validation)



hold out

used in training

run 1

run 2

run 3

run 4

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

The example for the case $S = 4$

- **Overfitting & Underfitting**
  (variance vs. bias)

**recall slide: what is the linear regression problem? (feature engineering)

effective feature engineering is the key to linear regression problem.

- choice of basis functions: nature of target tasks
- # (num) features: complexity but need to consider # data points
- no collinearity.

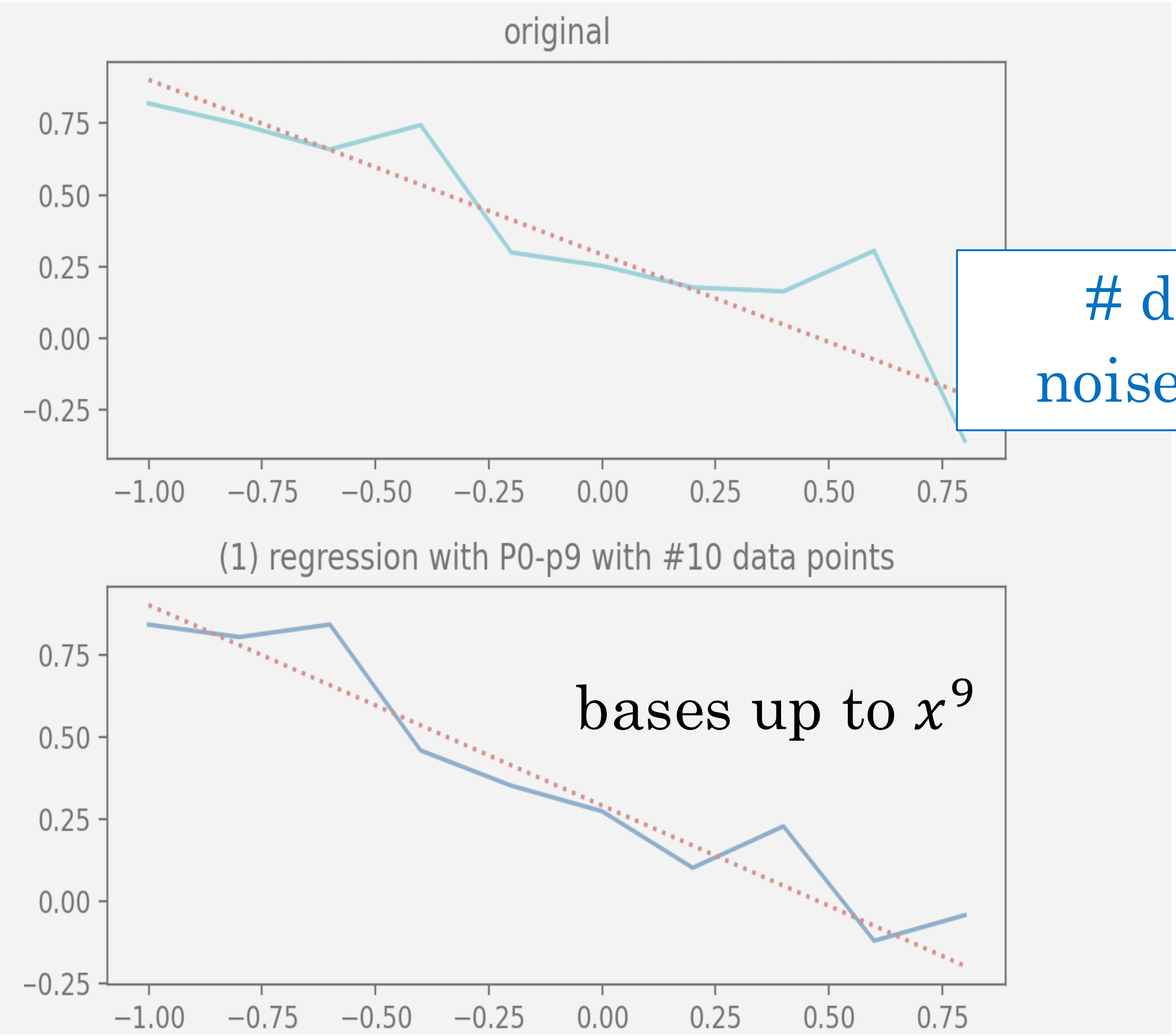
The higher polynomials are expressive, but we should be careful when using them when with a limited number of data points.

[1] Overfitting & Underfitting

Given a set of basis functions and data,
we can have the two problematic cases (if having inappropriate complexity):

(1) **Overfitting** / (2) **Underfitting**

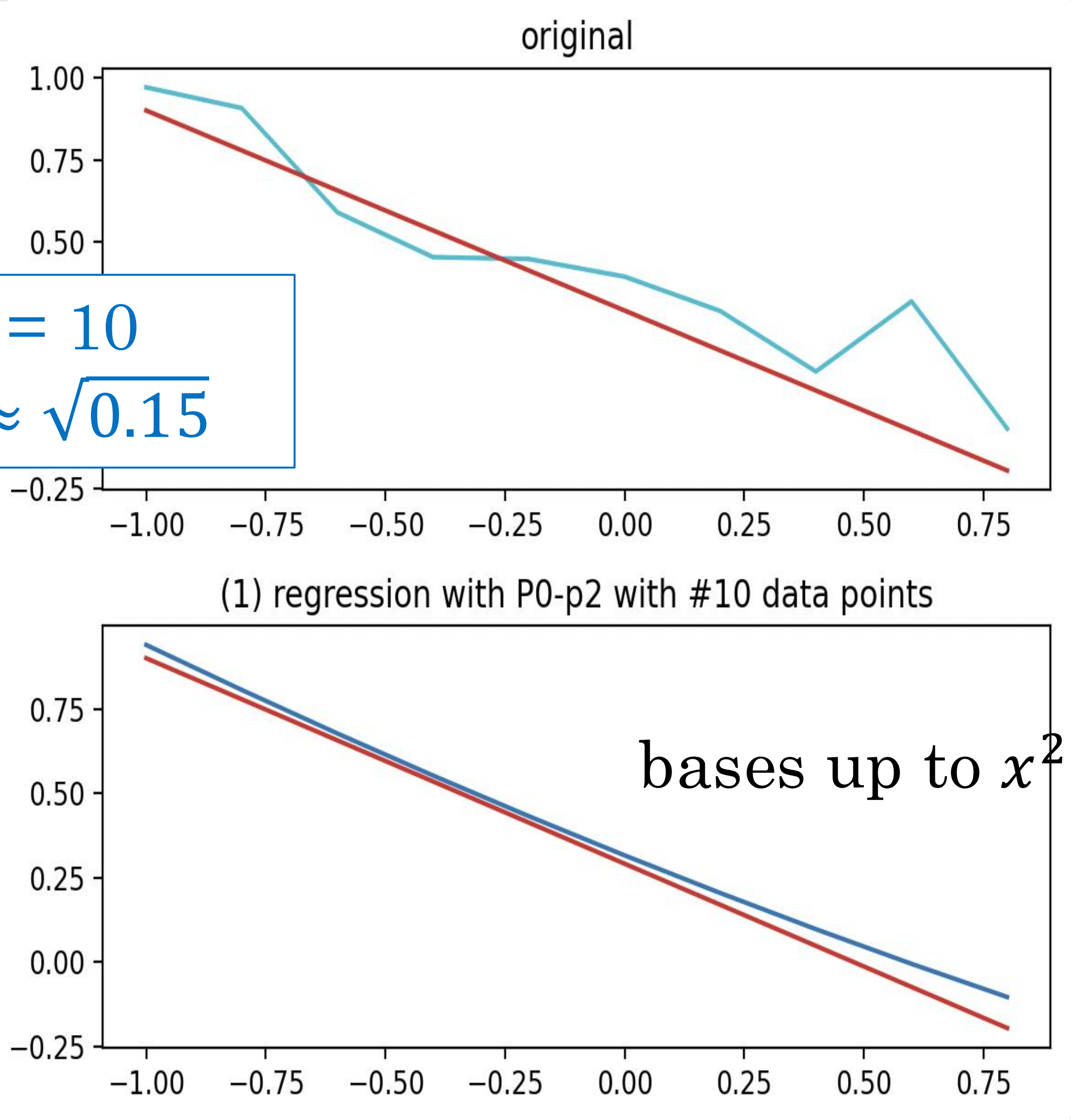
[2] Overfitting & Underfitting (example of overfitting)



bases up to x^9

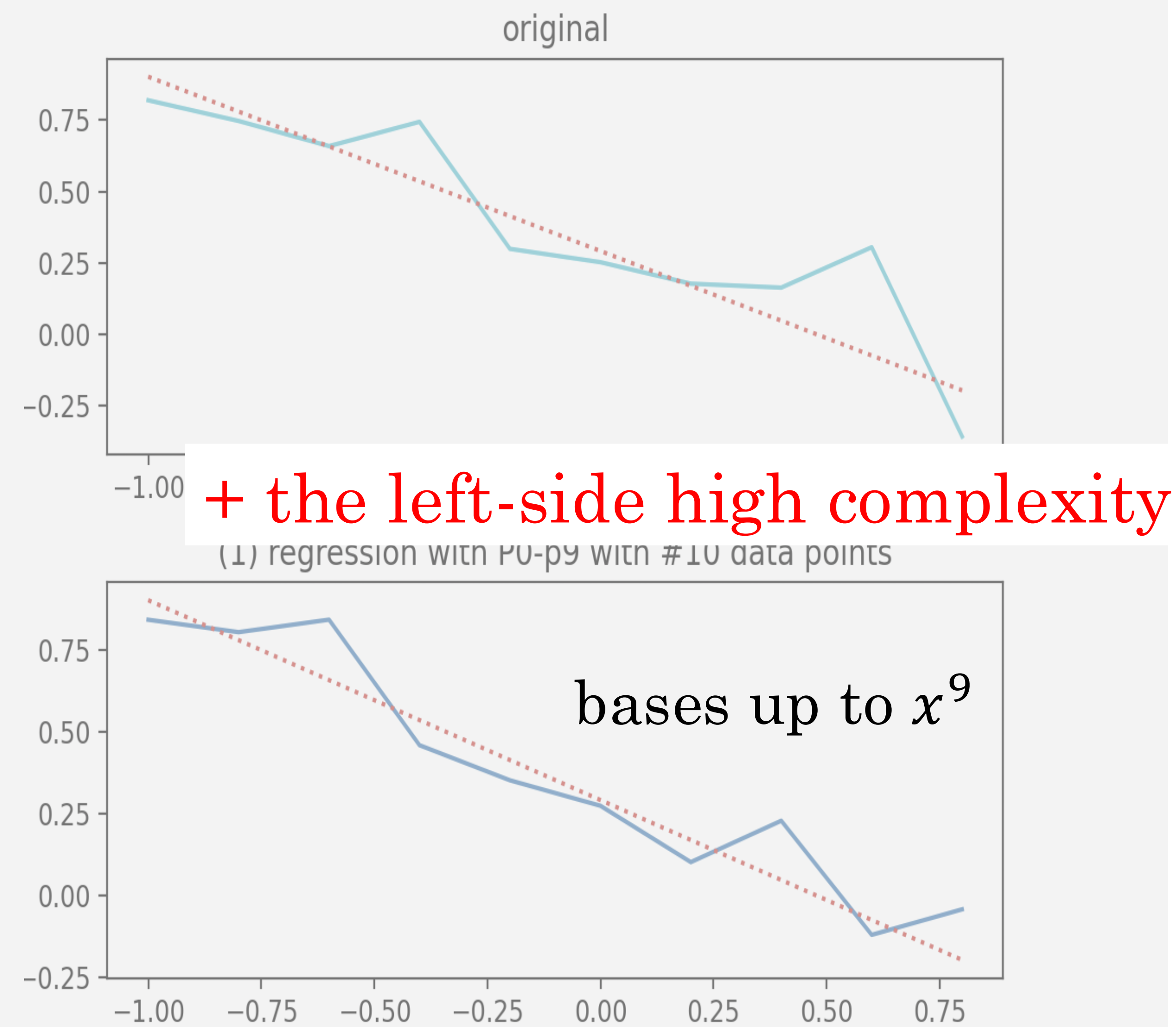
# data = 10

noise $\sigma \approx \sqrt{0.15}$

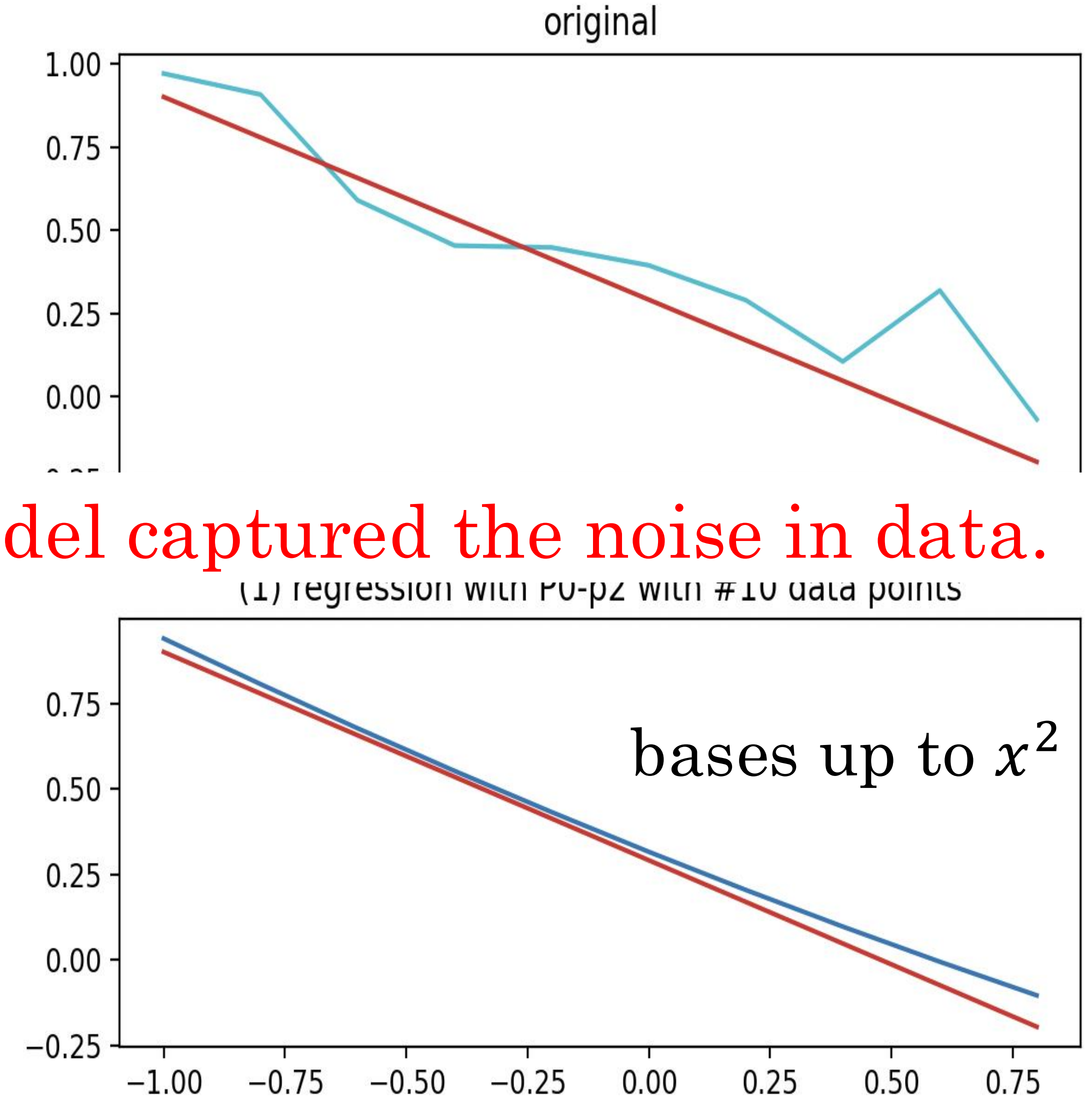


bases up to x^2

[3] Overfitting & Underfitting (overfitting)



bases up to x^9



bases up to x^2

+ the left-side high complexity model captured the noise in data.

[4] Overfitting & Underfitting (overfitting: too complex)

- overfitting

when model complexity is too high relatively to # training data,
then the complex model fits to noise/ undefined by data. one phenomenon is the large gap between train vs. test; we say the model has a high variance.

Q: what if we have enough training data?

[5] Overfitting & Underfitting (overfitting: too complex)

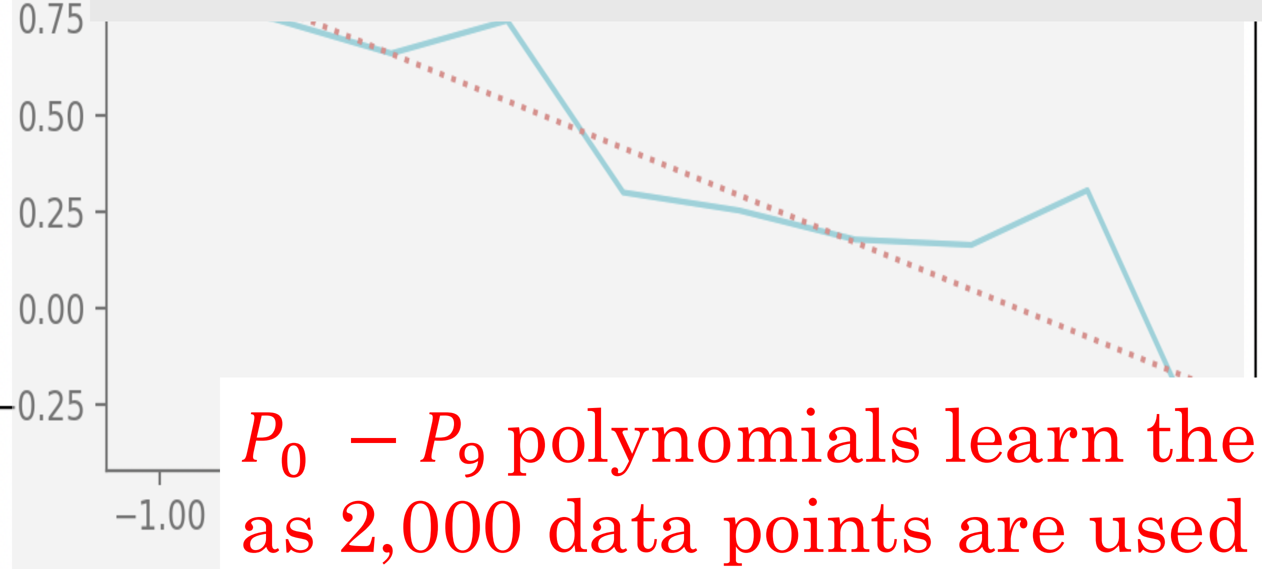
- overfitting

when model complexity is too high relatively to # training data,
then the complex model fits to noise/ undefined by data. one phenomenon is the large gap between train vs. test; we say the model has a high variance.

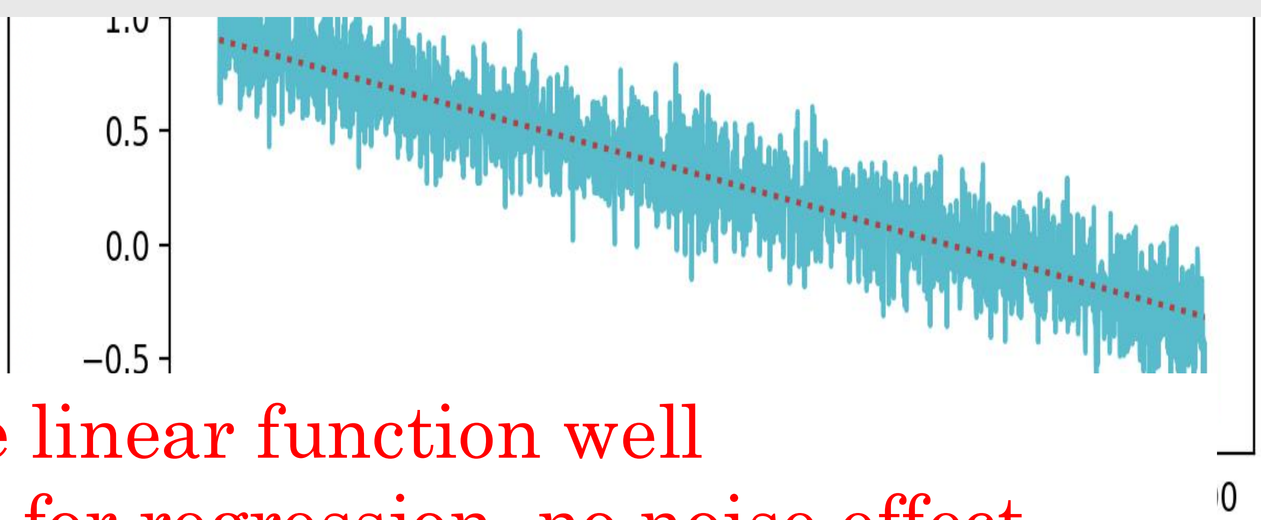
Q: what if we have enough training data?
$\sum_{i=1}^{\infty} \varepsilon_i = 0$ noise effect diminishes.

[6] Overfitting & Underfitting (with large amount of data)

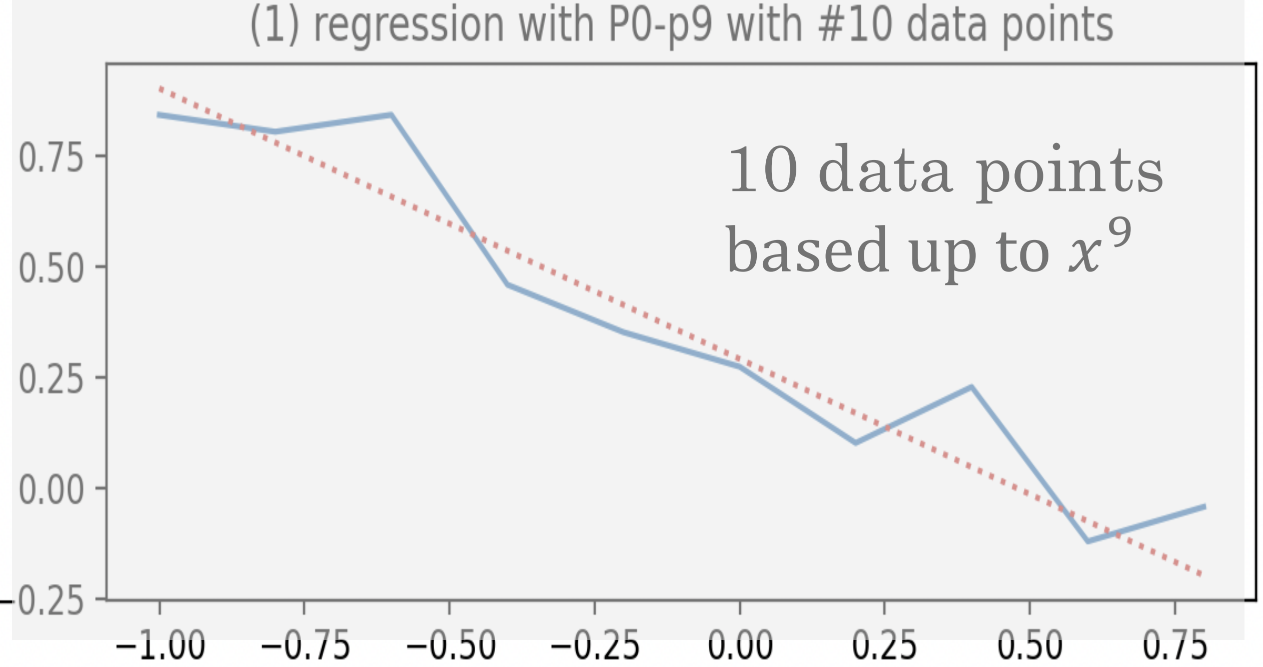
$P_0 - P_9$ polynomials learn the linear function well as 2,000 data points are used for regression. no noise effect.



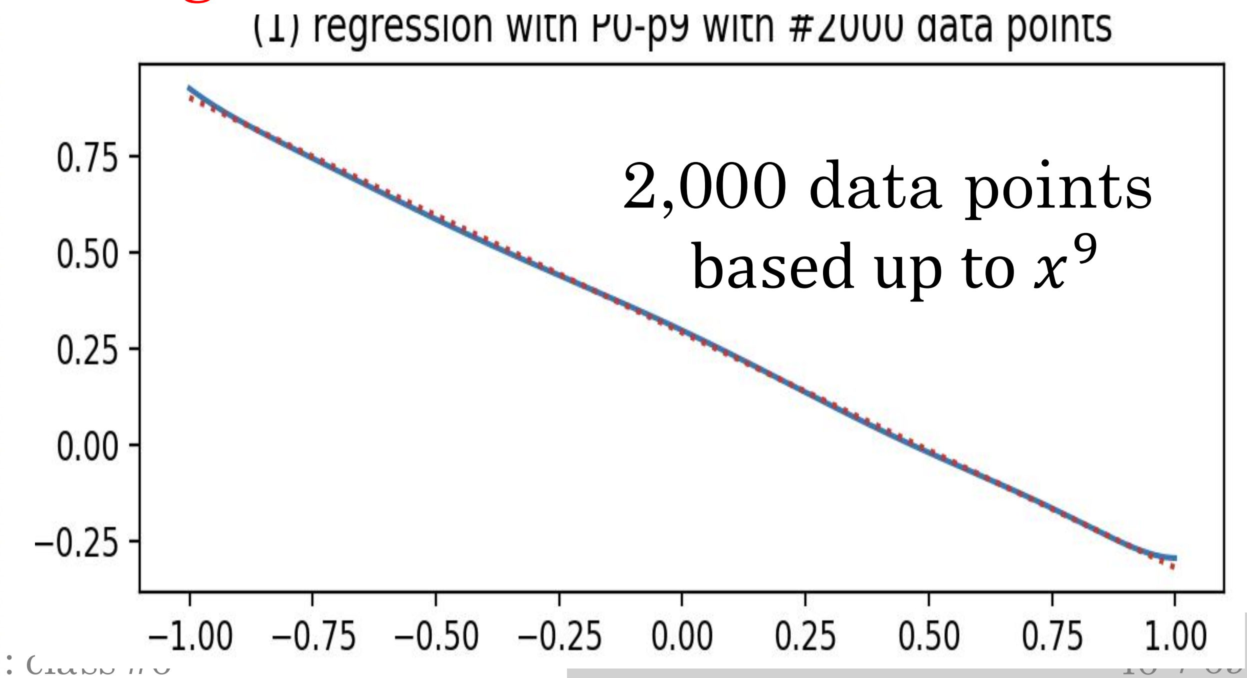
(1) regression with P0-p9 with #10 data points



(1) regression with P0-p9 with #2000 data points



10 data points based up to x^9



2,000 data points based up to x^9

[7] Overfitting & Underfitting (with large amount of data)

[from Bishop p 9.]

the larger data data set,
the more complex the model that we can afford to fit to the data.
One rough heuristic that is sometimes advocated that the number of data points should be no less than some multiple of the number of adaptive parameters.

[8] Overfitting & Underfitting

Q: how could we prevent the overfitting as we have limited data?

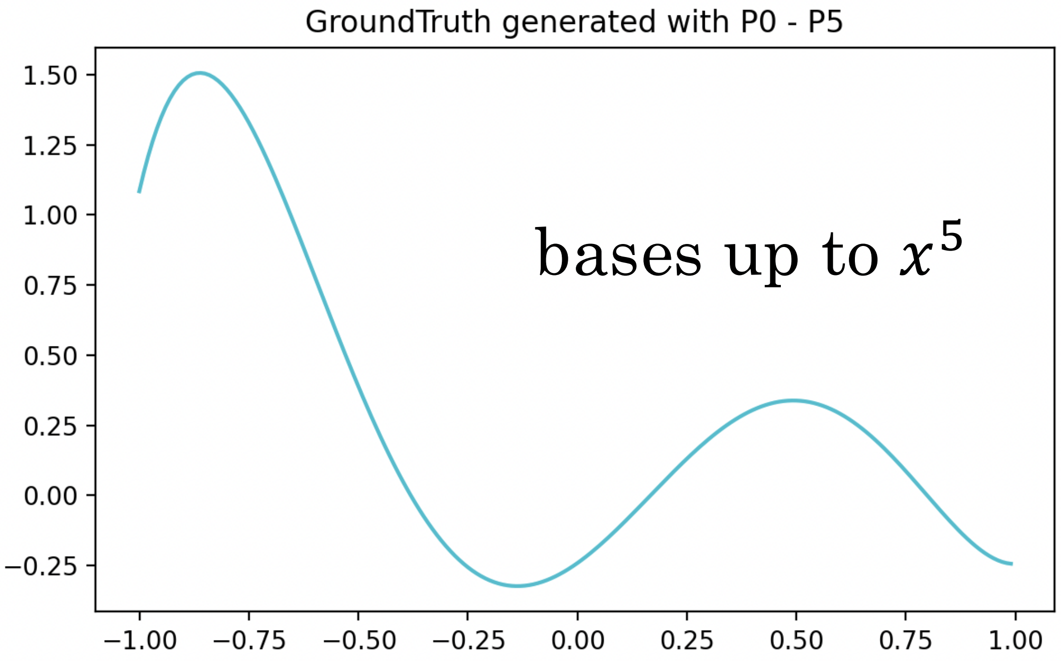
do we need to collect more data points?

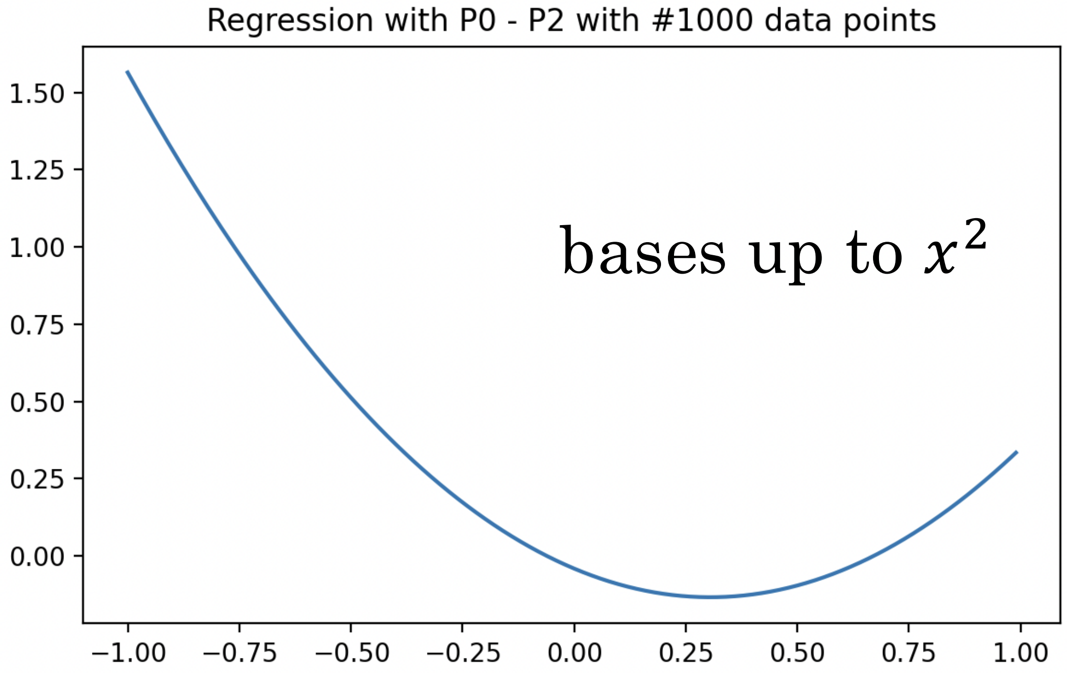
do we need to reduce the number of features?

[9] Overfitting & Underfitting (underfitting)

when model complexity
is not enough for the ground truth model,
no way to learn even when we have enough data. (no hope to learn)

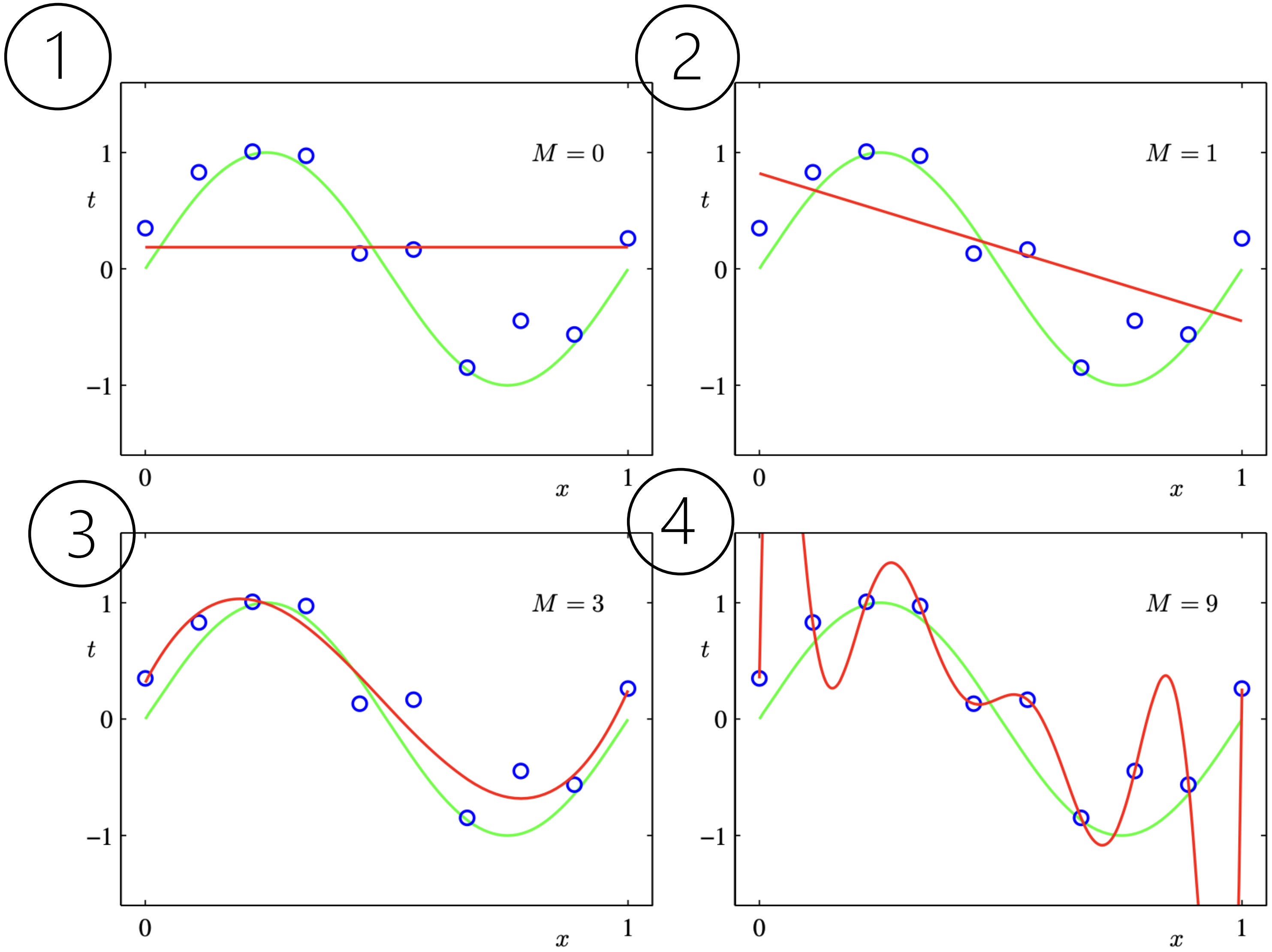
[10] Overfitting & Underfitting (example of underfitting)



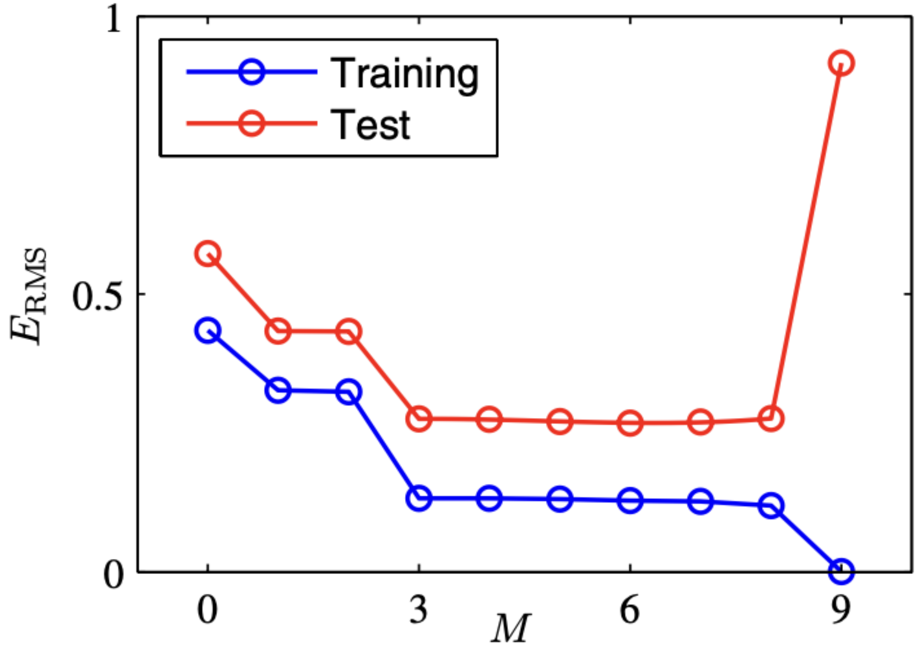


- underfitting example:
  # data is 1,000 but as the basis functions are limited by $P_0 - P_2$
  no way to capture the original function generated by $P_0 - P_5$

[11] Overfitting & Underfitting (textbook examples)



[from Bishop Figure 1.4 and 1.5]



Q: which one shows overfitting/ underfitting?

[12] Overfitting & Underfitting (symptoms)

- **overfitting**
  train performance good but test performance is bad
  the performance gap between test and train set (poor generalization/ high variance)

- **underfitting**
  both train and test performances are bad (biased/ high bias)

Q: how could we learn a model in right complexity?
for the primary goal in ML, high generalization!

- The Bias Variance Decomposition

  test error $= \text{variance} + bias^2 + \text{intrinsic error}$

[1] Bias & Variance Decomposition (Empirical MSE & Expected MSE)

- suppose we had a data set: *D*= *{*$(\overrightarrow{x_i}, t_i)$*, i=1,2,…n}*
- we had a set of basis function
- we learned a regression model $y(\vec{x})$ from D.

[2] Bias & Variance Decomposition (Empirical MSE & Expected MSE)

- suppose we had a data set: $D = \{(\vec{x}_i, t_i),\ i=1,2,\ldots N\}$
- we had a set of basis functions
- we learned a regression model $y(\vec{x}; D)$from D.

(1) empirical MSE error (this is the one we usually do.)

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

(2) expected MSE error, assuming we know the density $f(\vec{x}, t) = f(t|\vec{x})\, f(\vec{x})$

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

[3] Bias & Variance Decomposition (decomposition)

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

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

- $h(x)$ is an optimal function minimizing the modeling error given density $f(x,t)$

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

[4] Bias & Variance Decomposition (optimal $h(x) = E[T|x]$)

Suppose y(x) is our model, data follows the density $f(t, x) = f(t|x) \cdot f(x)$

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

$$J(y^*(x)) = \int_t (y^*(x) - t)^2 f(t|x) \, \mathrm{d}t$$

$$\frac{\partial J}{\partial y^*(x)} = \int_t (2y^*(x) - 2 \cdot t) f(t|x) \, \mathrm{d}t = 0$$

$$y^*(x) = \int_t t f(t|x) \, \mathrm{d}t = \boxed{E[T|x]}$$

$E[T|x]$ is a function of x

**for $f(x) \geq 0$, we can set up minimization only for $\int_t (y(x) - t)^2 f(t|x)$

[5] Bias & Variance Decomposition (decomposition)

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

$$\begin{aligned} E[L] &= \int_x \int_t (y(x;D) - h(x) + h(x) - t)^2 f(t|x) f(x)\,\mathrm{d}t\,\mathrm{d}x \\ &= \int_x \int_t \underbrace{(y(x;D) - h(x))^2}_{?} + \underbrace{(h(x) - t)^2}_{\epsilon} + \underbrace{2(y(x;D) - h(x))(h(x) - t)}_{\text{zero}}\,\mathrm{d}t\,\mathrm{d}x \end{aligned}$$

$t = h(x) + \epsilon$

$$\int_x \int_t 2(y(x;D) - h(x))(h(x) - t) f(t|x) f(x) dt dx$$

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

$$\int_x 2(y(x;D) - h(x))(\underbrace{E[T|x] - E[T|x]}_{\text{zero}}) f(x)\,\mathrm{d}x = 0$$

[6] Bias & Variance Decomposition (decomposition)

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

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

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

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

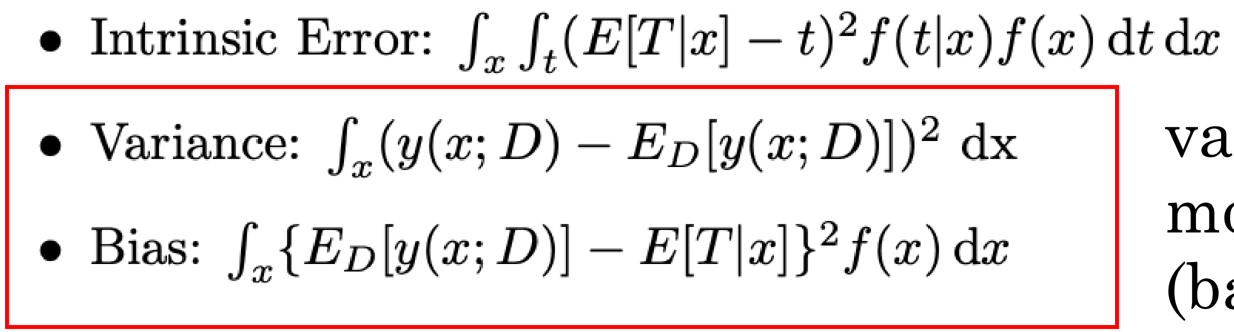
$$\int_x (y(x;D) - \boxed{E_D[y(x;D)]} + E_D[y(x;D)] - h(x))^2 f(x)\,\mathrm{d}x$$

expectation over different data sets

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

[7] Bias & Variance Decomposition (decomposition)

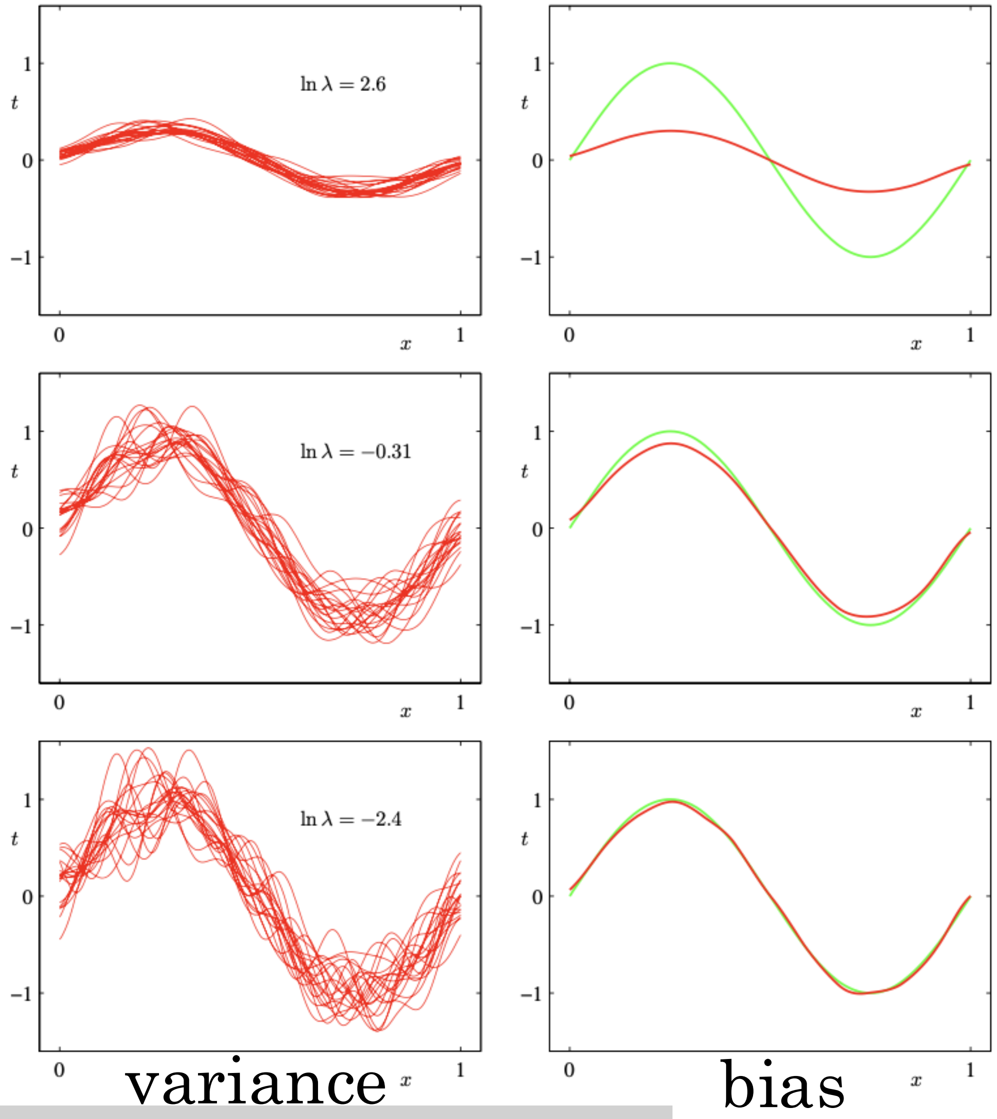
- suppose we had a data set: $D = \{(\vec{x_i}, t_i), i=1, 2, \ldots, N\}$
- we had a set of basis functions
- we learned a regression model $y(\vec{x}; D)$from D.
- the expected error is decomposed into three terms
  $E[L]$ = variance + $bias^2$ + intrinsic error



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

variance and bias vary depending on model complexity!
(basis set or regularization parameter)

[8] Bias & Variance Decomposition (trade off textbook example)

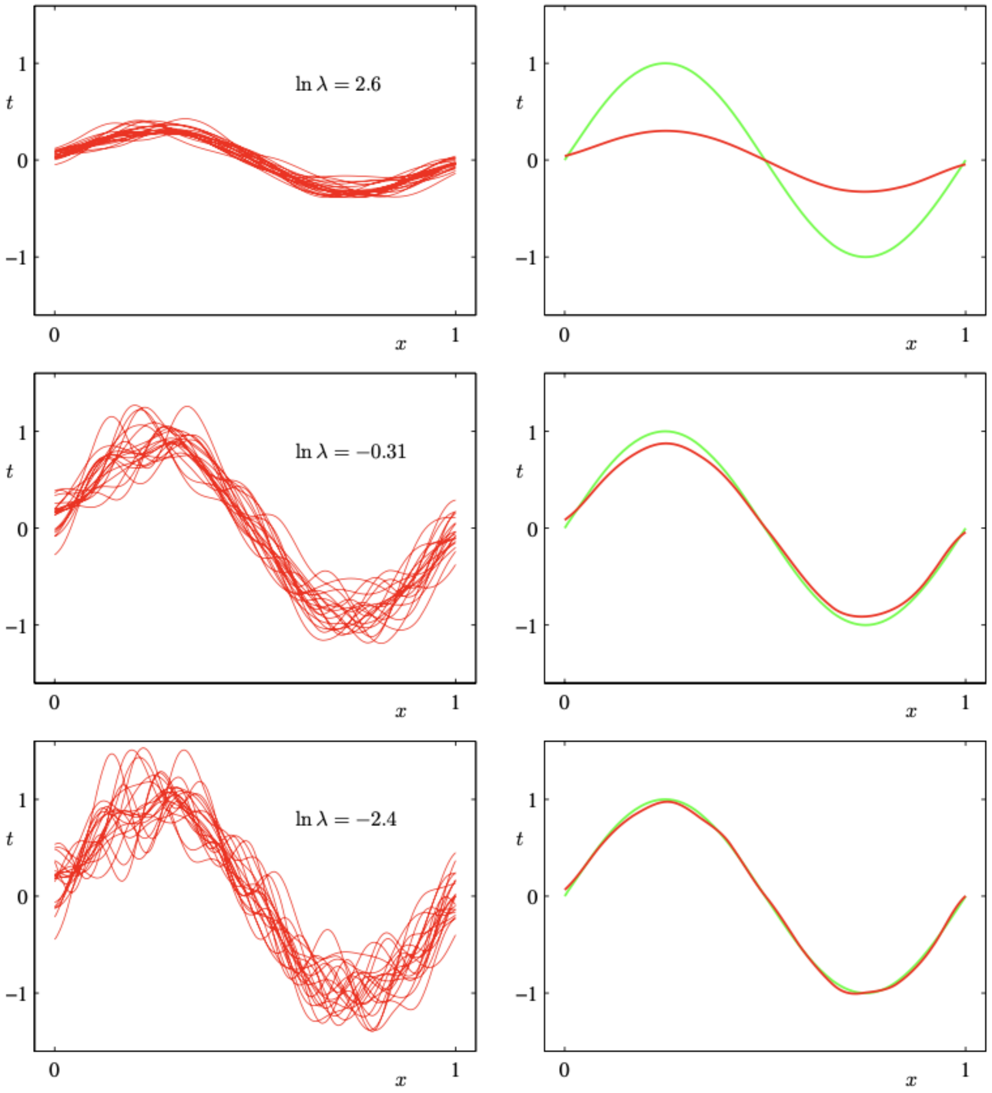


increasing complexity

[from Bishop p. 150]

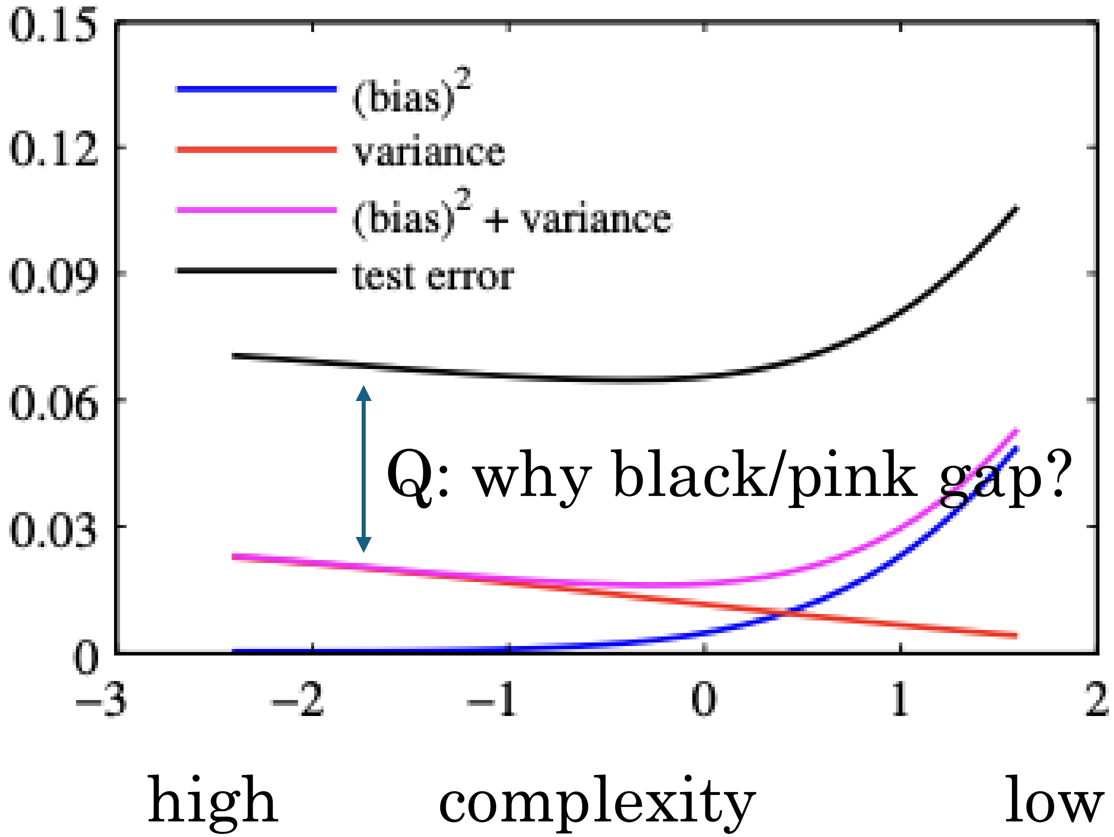
Q: how changes the variance & bias?

[9] Bias & Variance Decomposition (trade off textbook example)



increasing complexity

[from Bishop p. 150]



Q: why black/pink gap?

- Controlling Effective Model Complexity

  (strategy to avoid over/underfitting)

[1] Controlling Effective Model Complexity (strategies)

- **underfitting**

  both train and test performances are bad (biased/ high bias)

- **overfitting**

  train performance good but test performance is bad

  the performance gap between test and train set (poor generalization/ high variance)

Q: how could we learn a model in right complexity?

for the primary goal in ML, high generalization!

[2] Controlling Effective Model Complexity (colliding strategies)

- **underfitting**

  both train and test performances are bad (biased/ high bias)

  increasing the order of model hypothesis space

- **overfitting**

  train performance good but test performance is bad

  the performance gap between test and train set (poor generalization/ high variance)

  decreasing the order of model hypothesis space

Q: what will be the best strategies?

[2] Controlling Effective Model Complexity (colliding strategies)

- **Underfitting**
  both train and test performances are bad (biased/ high bias)
  increasing the order of model hypothesis space

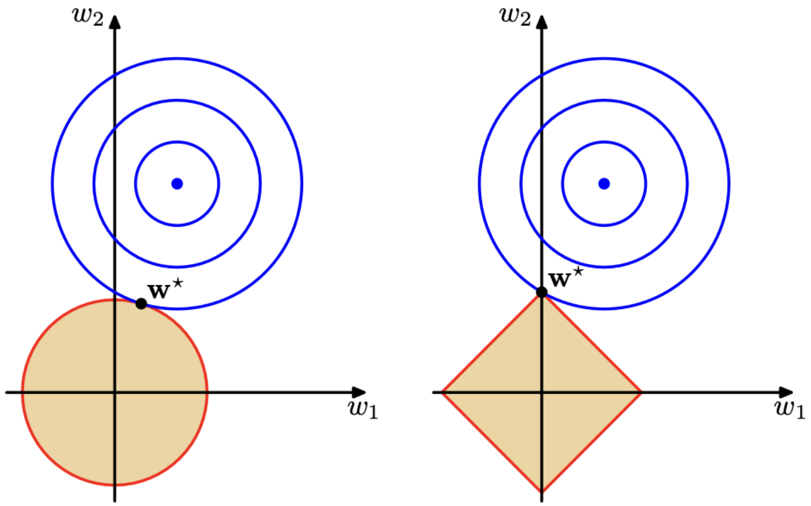
- **Overfitting**
  train performance good but test performance is bad
  the performance gap between test and train set (poor generalization/ high variance)
  decreasing the order of model hypothesis space

Q: what will be the best strategies?
collect many data samples but expansive. (ideal solution but not practical)
then, how do we control model complexity?

[3] Controlling Effective Model Complexity (regularization)

$$\arg\min_{\vec{w}} ||\vec{y} - \Phi \cdot \vec{w}||^2 + \lambda^*(||\vec{w}||^2)$$ [Ridge regularization]

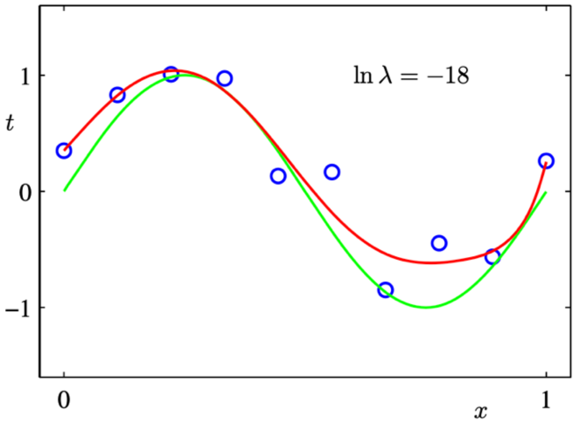
$$\arg\min_{\vec{w}} ||\vec{y} - \Phi \cdot \vec{w}||^2 + \lambda^*(||\vec{w}||)$$ [Lasso regularization]



- In Bayesian model, the effective number of parameters adapts automatically to the size of the data set.

[4] Controlling Effective Model Complexity (regularization by lambda)

overfitting suppressed

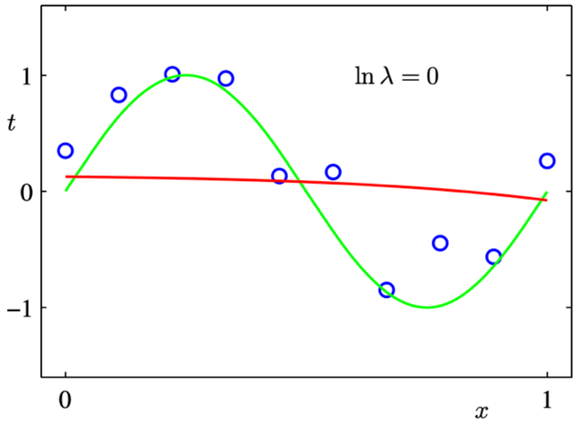


$ln\lambda = -18$

N = 10 (limited)

M = 9

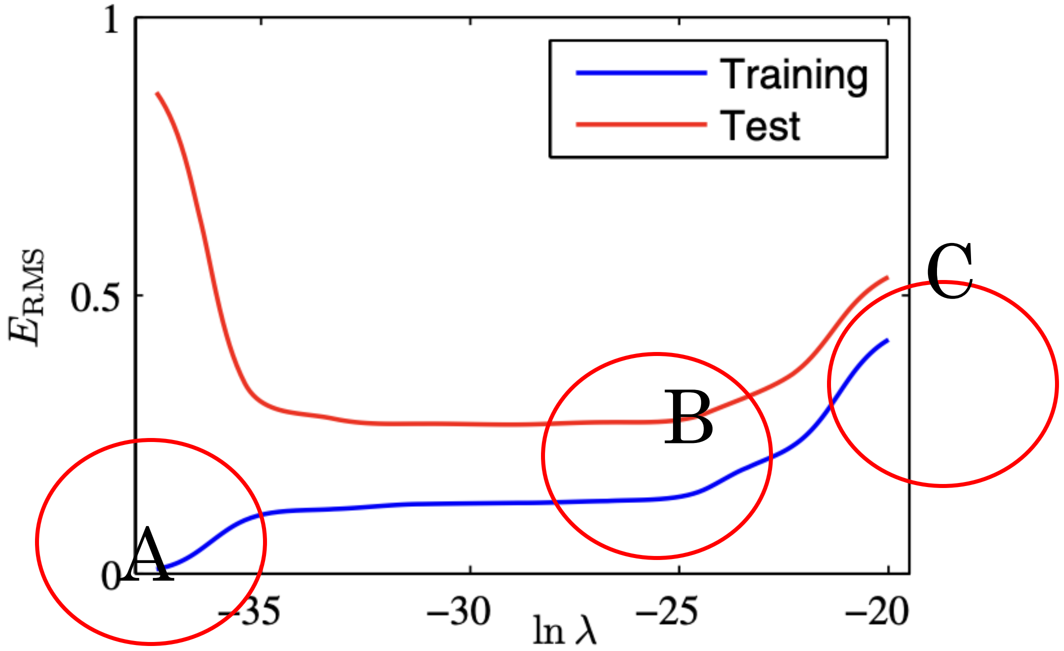
too simple



$ln\lambda = 0$

N = 10

M = 9



Q: which one would you choose A, B, C?