Introduction to Supervised Learning

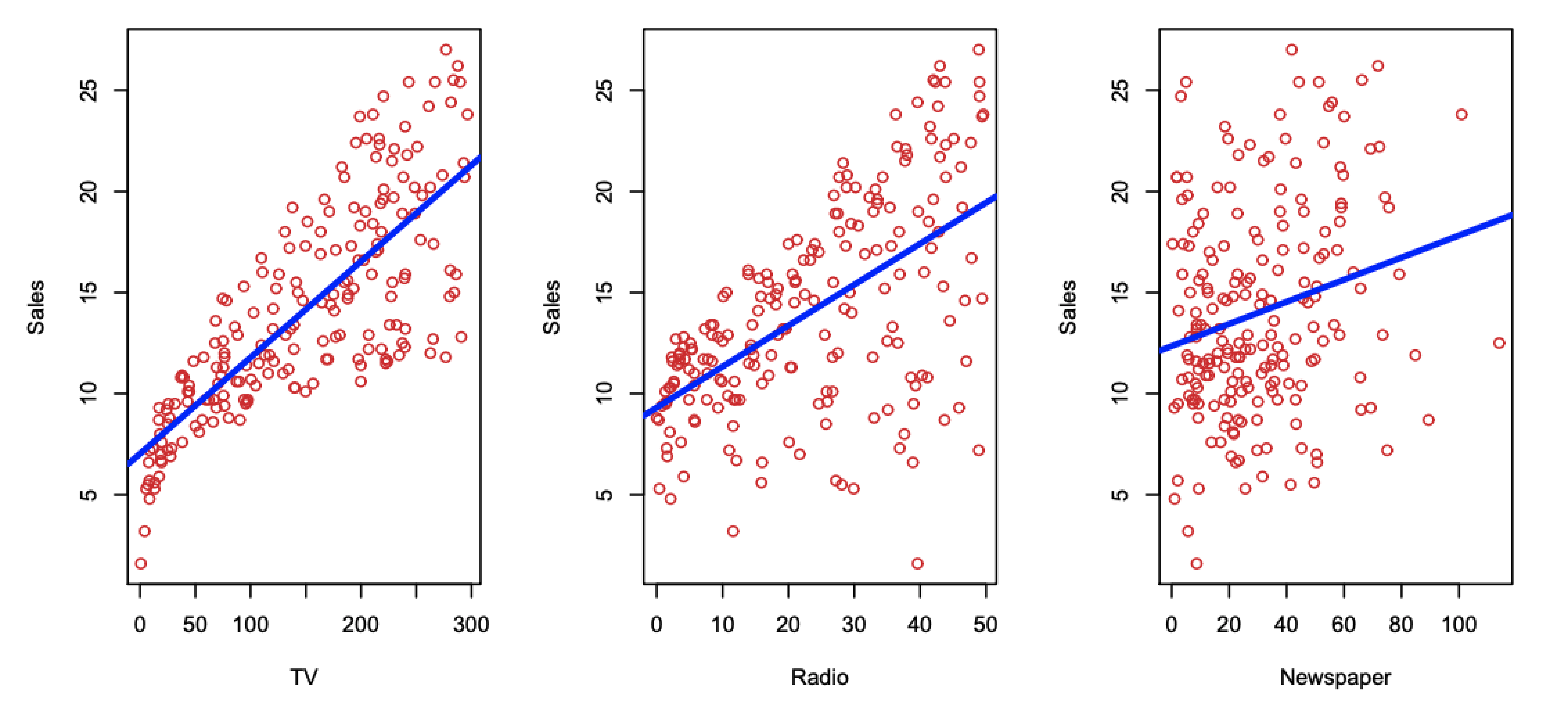
金融投资学

Instructor: Haoran LEI

Hunan University

A marketing example

- Bob founded a company selling baby shoes. He advertised the company products via TV, Radio and Newspaper.
- Given the history data of **Sales** and advertising expense on **TV**, **Radio** and **Newspaper**, can we
  - predict "Sales" using these three inputs?
  - know how can Bob advertise his goods better given a budget set on advertising?
- What's your advice to Bob?


Sales
TV
Sales
Radio
Sales
Newspaper

Bob can do better using a **model**:

$$\texttt{Sales} \approx f(\texttt{TV}, \texttt{Radio}, \texttt{Newspaper}).$$

- `Sales` is a **response** or **target** that Bob wishes to predict
- `TV` is a **feature**, or **input**, or **predictor**. We name it X_1
- Likewise, `Radio` is named as X_2, and so on.

$$X = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}, X \text{ is called the input vector.}$$

- Bob can write his model as:

$$Y = f(X) + \epsilon$$

- where ϵ captures measurement errors and other discrepancies.
- What can Bob do with f?

With a good f, Bob can

- make predictions of sales (Y) at new points $X = x$;
- understand which components of X are important in explaining Y, and which are irrelevant;
    - In the salary case, "Seniority" and "Years of Education" are have a big impact on Income, but "Marital Status" typically does not.
- Depending on the complexity of f, we may be able to understand how each component X_j of X affects Y.

The ideal $f(x)$, expected value and regression function

- Is there an ideal $f(X)$?
- What is a good value for $f(X)$ given a specific value of X, say $X = 4$?

The ideal $f(x)$ and expected value

- Is there an ideal $f(X)$?
- What is a good value for $f(X)$ given a specific value of X, say $X = 4$?

Theoretically, a **very good** value will be

$$f(4) = \mathbb{E}[Y|X = 4]$$

- pronounced as "the expected value of Y given X being 4".

So, the ideal f is $f(x) = \mathbb{E}(Y|X = x)$, the **regression function**.

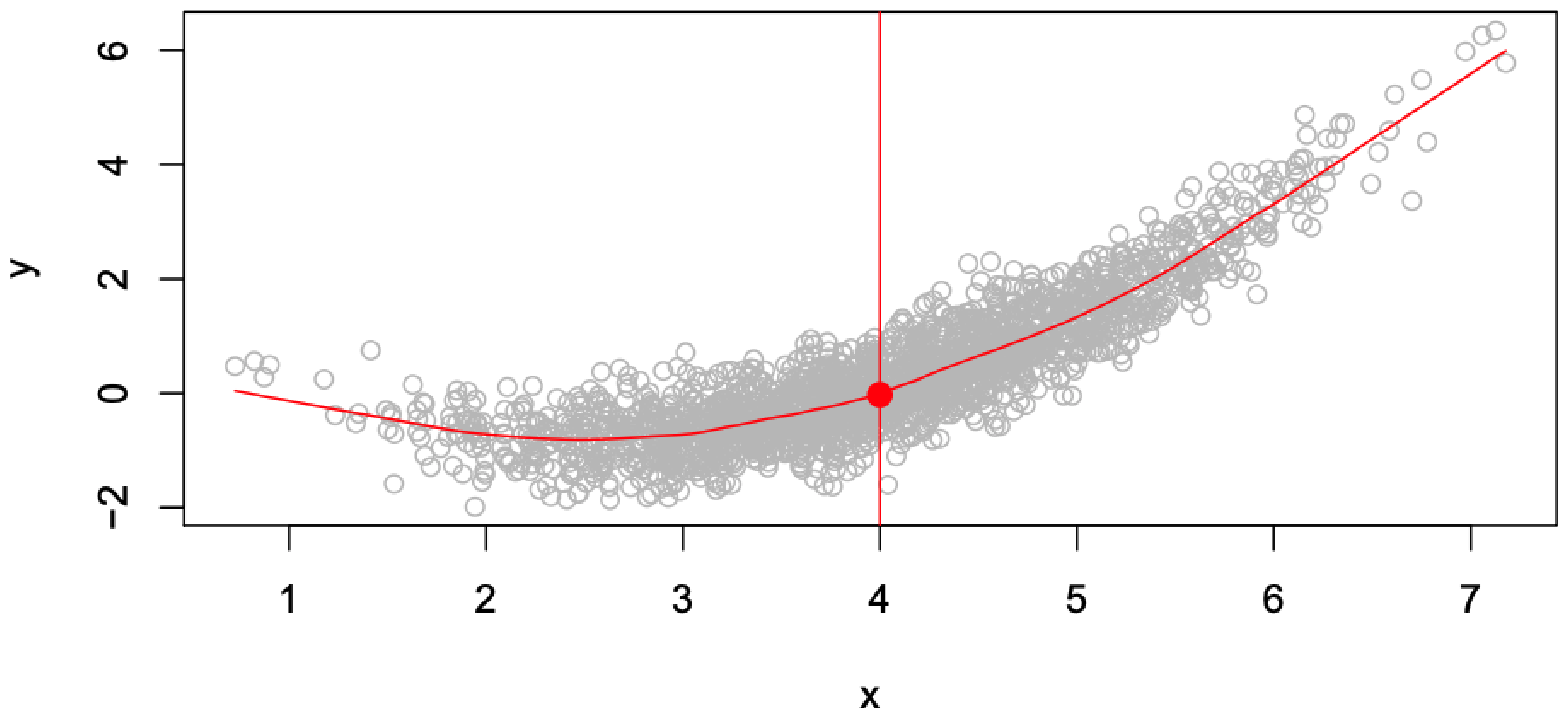
Regression function

- $f(x) = \mathbb{E}[Y|X = x]$ is formally called the **regression function**.

To make a **prediction**, we are calculating the **conditional expectations** of Y given X:

$$f(x) = f(x_1, x_2, x_3) = E[Y|X_1 = x_1, X_2 = x_2, X_3 = x_3]$$

Example: Use $f(4)$ as a prediction for Y given $X = 4$.



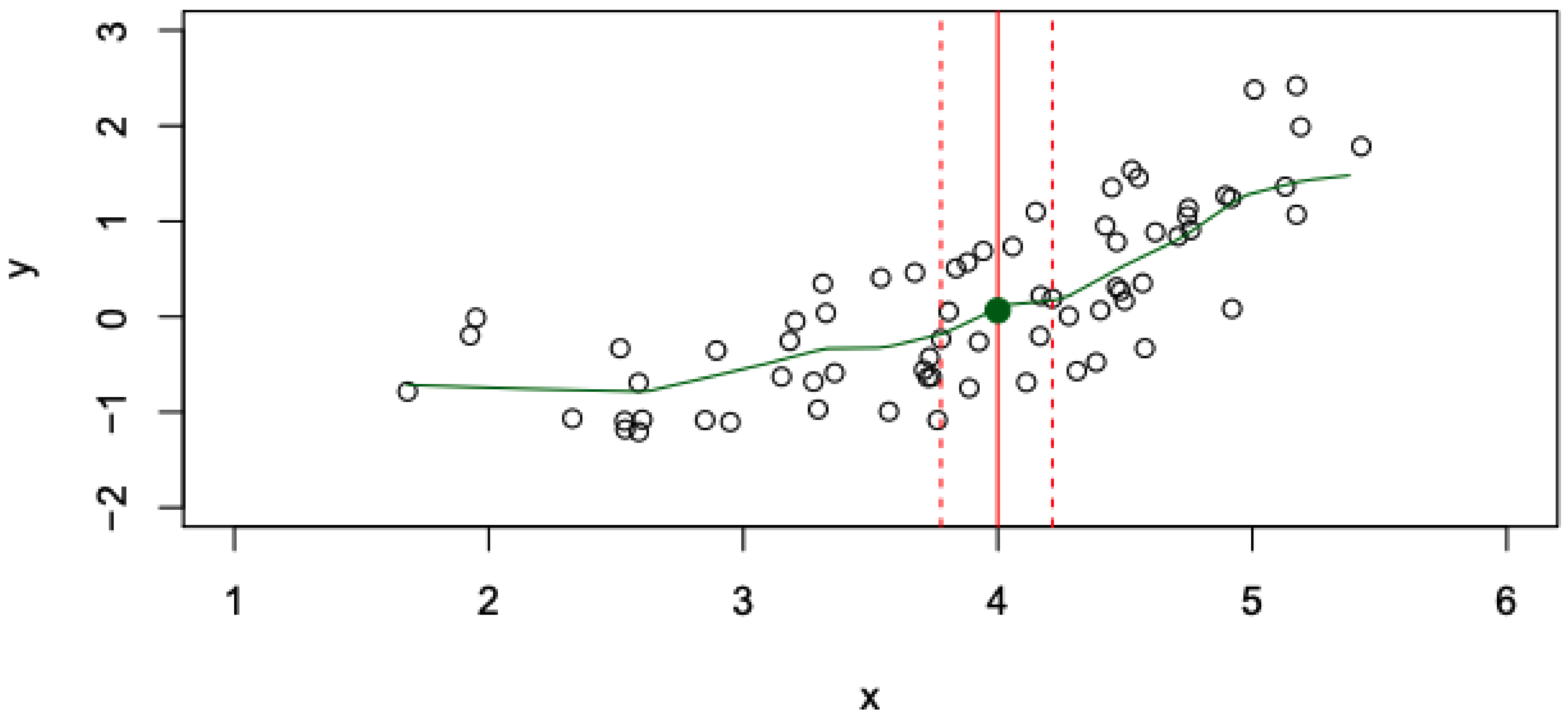
How to estimate f

- Typically, Bob has few (if any) data points with $X = 4$ exactly.
- So we cannot compute $E[Y|X = x]$!
- A good estimate $\hat{f}$ of f at x is

$$\hat{f}(x) = \text{Ave}(Y|X \in \mathcal{N}(x))$$

where $\mathcal{N}(x)$ is some **neighborhood** of x.

Example: estimate $\hat{f}(4)$



Nearest neighbor: pros and cons

Nearest neighbor averaging can be pretty good for small p – ie, $p \leq 4$ and large N

- We'll discuss smoother version, such as kernel and spline smoothing later.

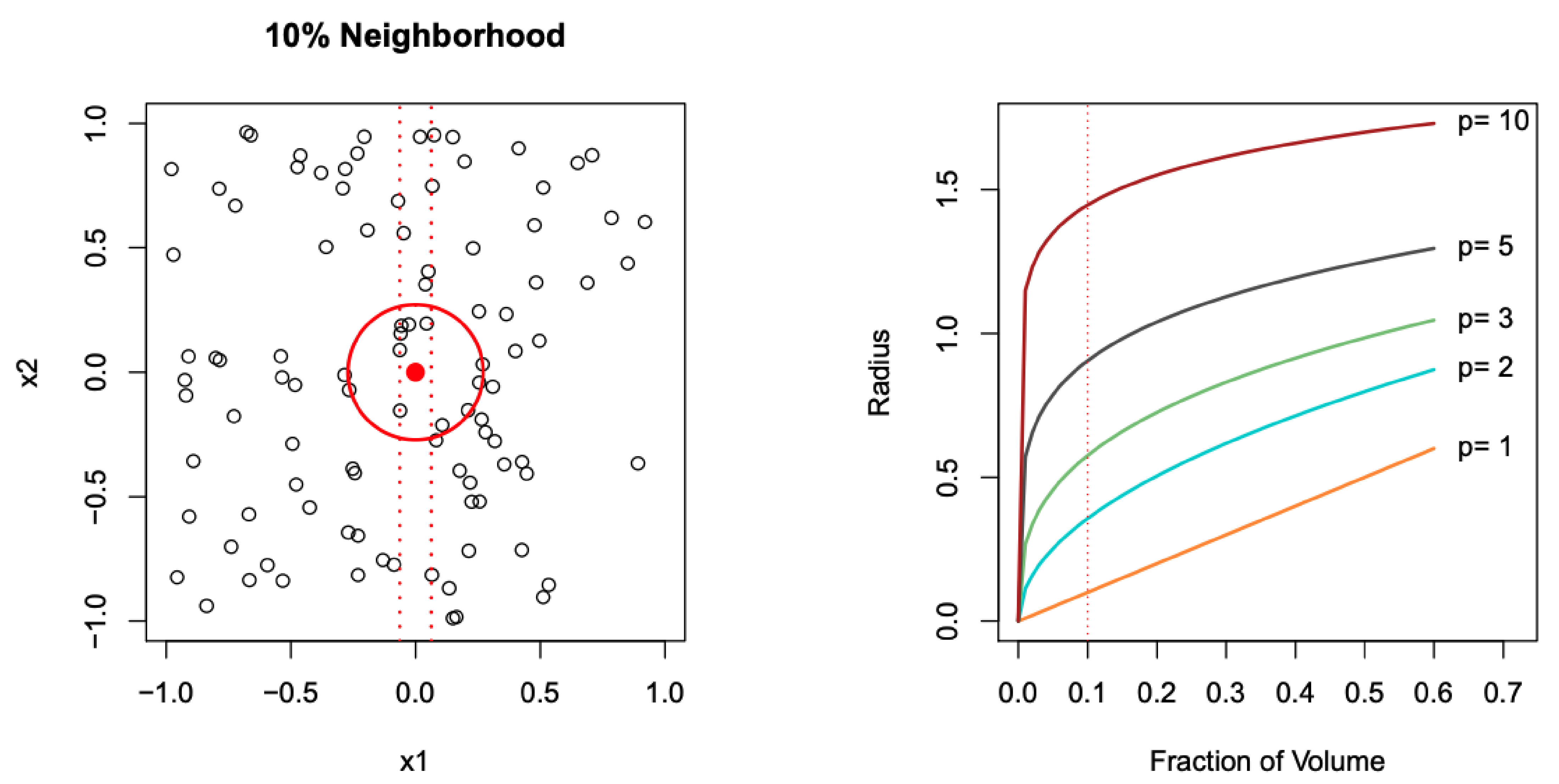
Nearest neighbor method is likely to perform poorly when *p* is large.

- **The curse of dimensionality**.

Nearest neighbor and the curse of dimensionality

Nearest neighbors estimates, $\hat{f}(x)$, tend to be **far away** in high dimensions.

- We need to get a reasonable fraction of the N values of y_i to average to bring the variance down, say 10%
- However, a 10% neighborhood in high dimensions need no longer be local, so we lose the spirit of estimating $\mathbb{E}(Y|X = x)$ by local averaging.



The curse of dimensionality: illustration

Parametric and structured models

The linear model is an important example of a parametric model:

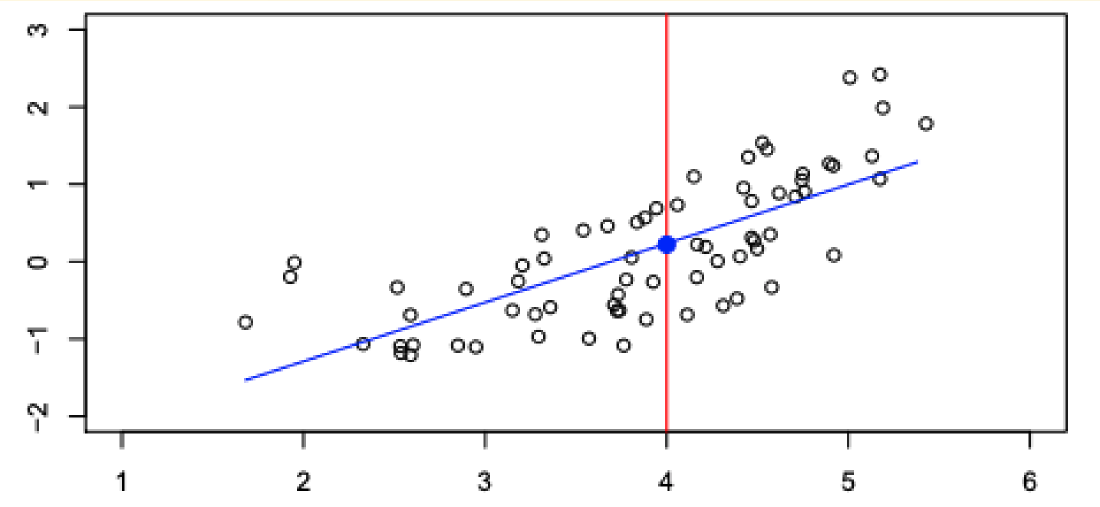
$$f_L(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

- A linear model is specified by $p + 1$ parameters: the β's
- Estimate the parameters by fitting the model to **training data**.

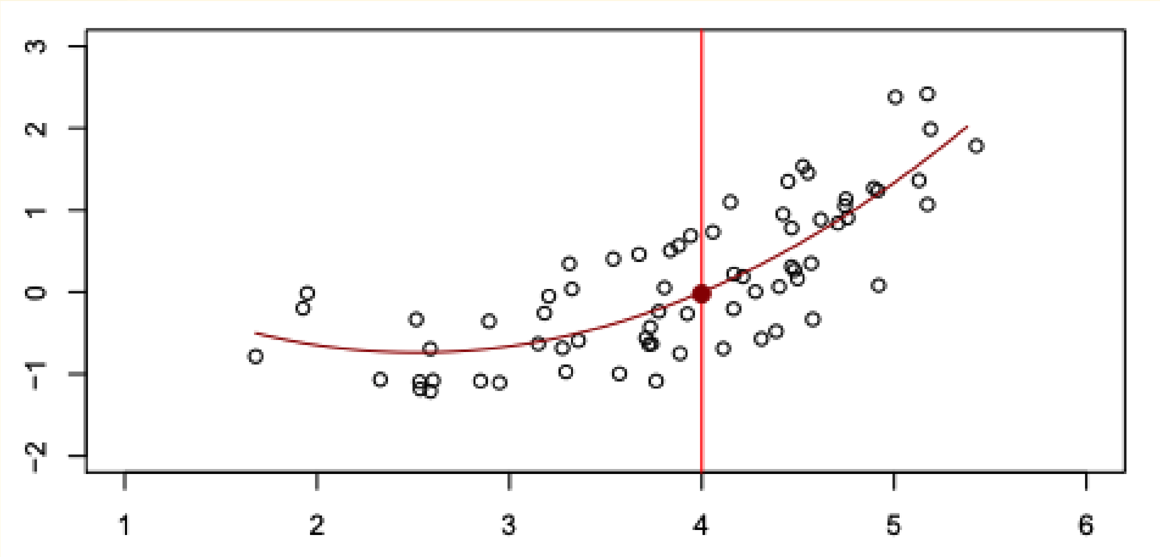
Why linear model

- Linear models are almost never "correct"...
  - Being correct means that the true relationship $f(X)$ is linear in $X_1, \ldots, X_p$
- However, a linear model $\hat{f}_L(X)$ often serves as a good and **interpretable** approximation to the unknown $f(X)$.
- In many real usages, linear models are good enough.

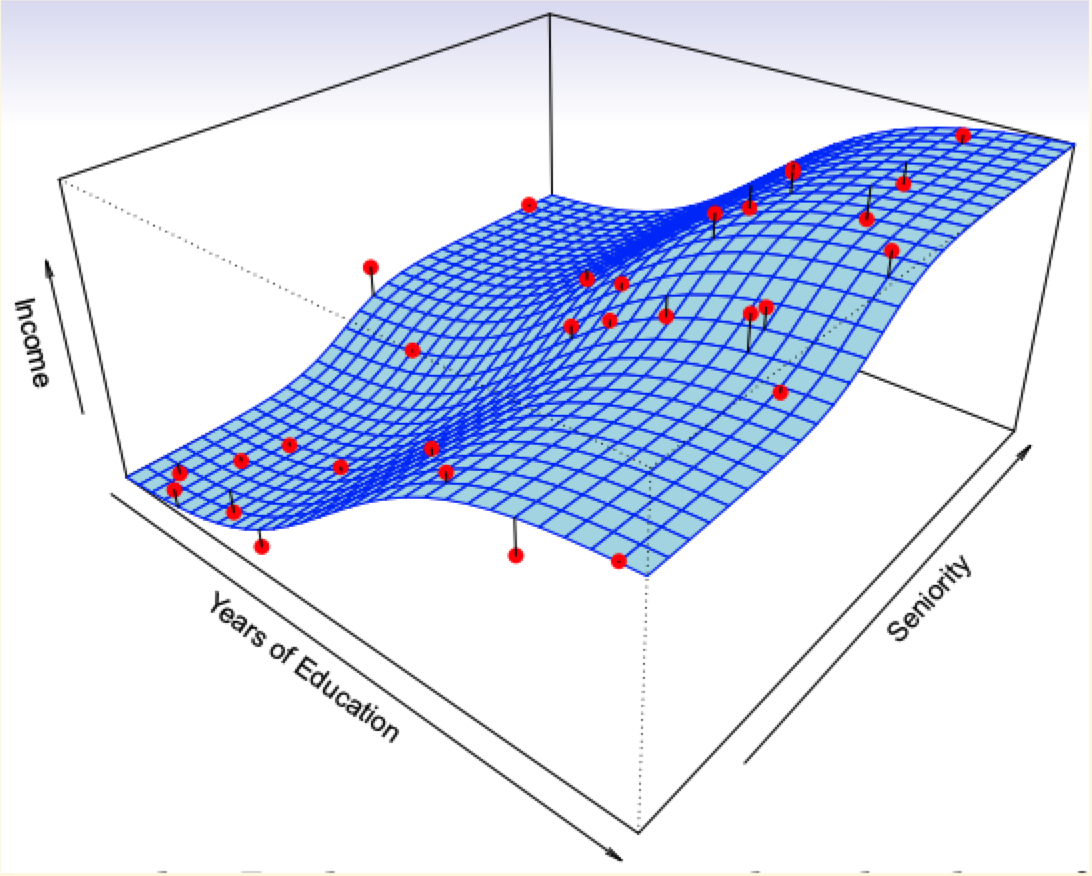
A linear model $\hat{f}_L(X) = \hat{\beta}_0 + \hat{\beta}_1 X$ gives a reasonable fit:



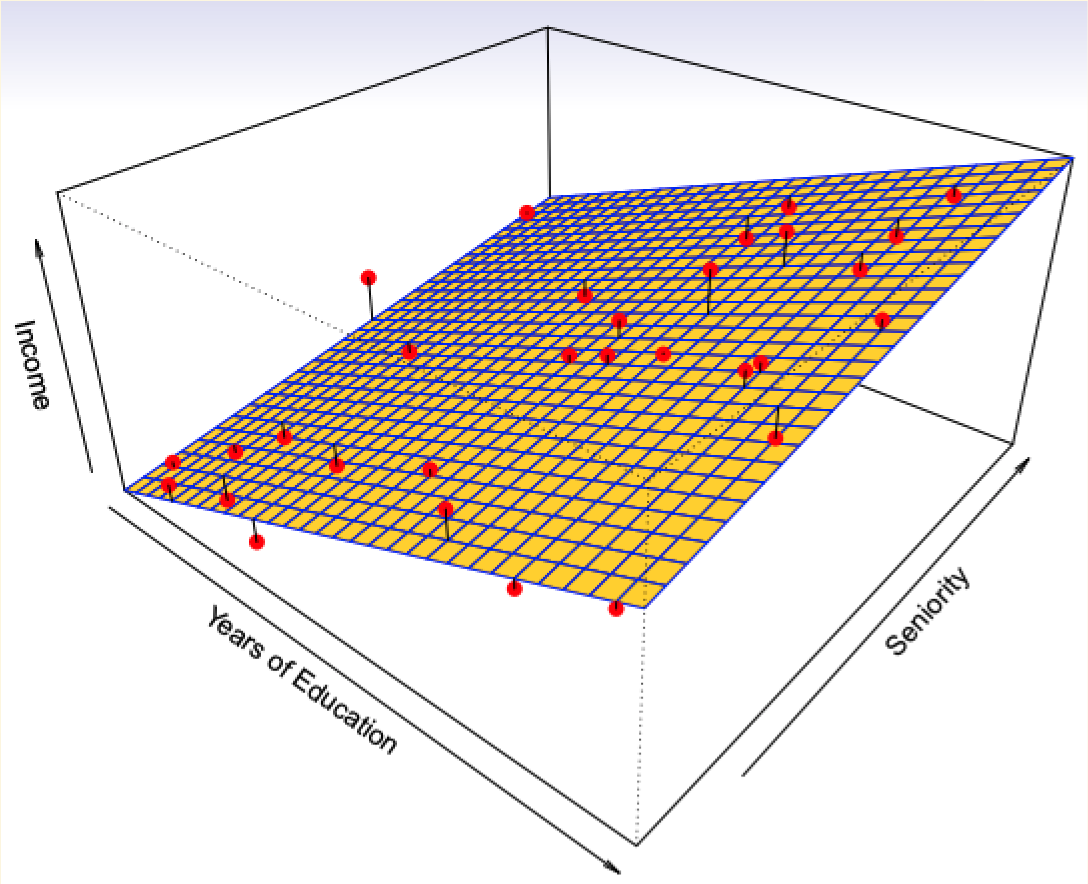
A quadratic model $\hat{f}_Q(X) = \hat{\beta}_0 + \hat{\beta}_1 X + \hat{\beta}_2 X^2$ fits slightly better:



Simulated example. Red points are simulated values for income from the model (**the blue surface**):

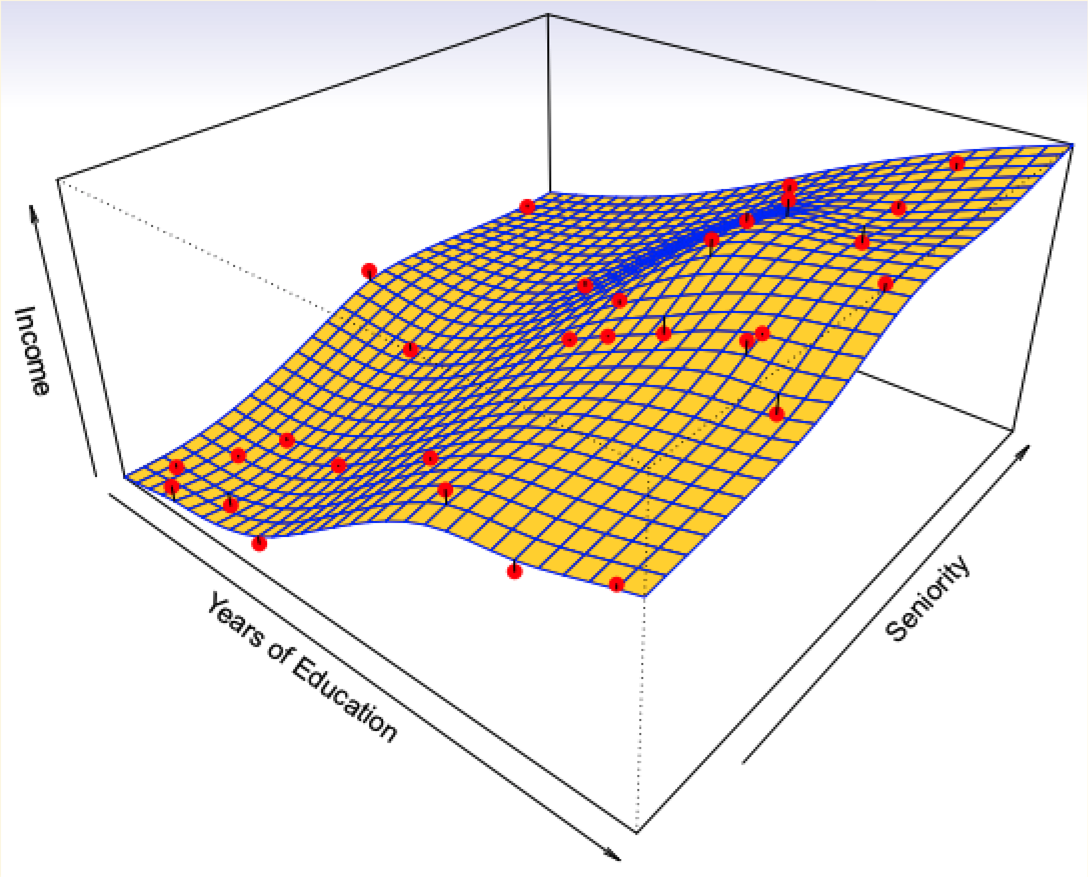


Linear regression model fit to the simulated data:

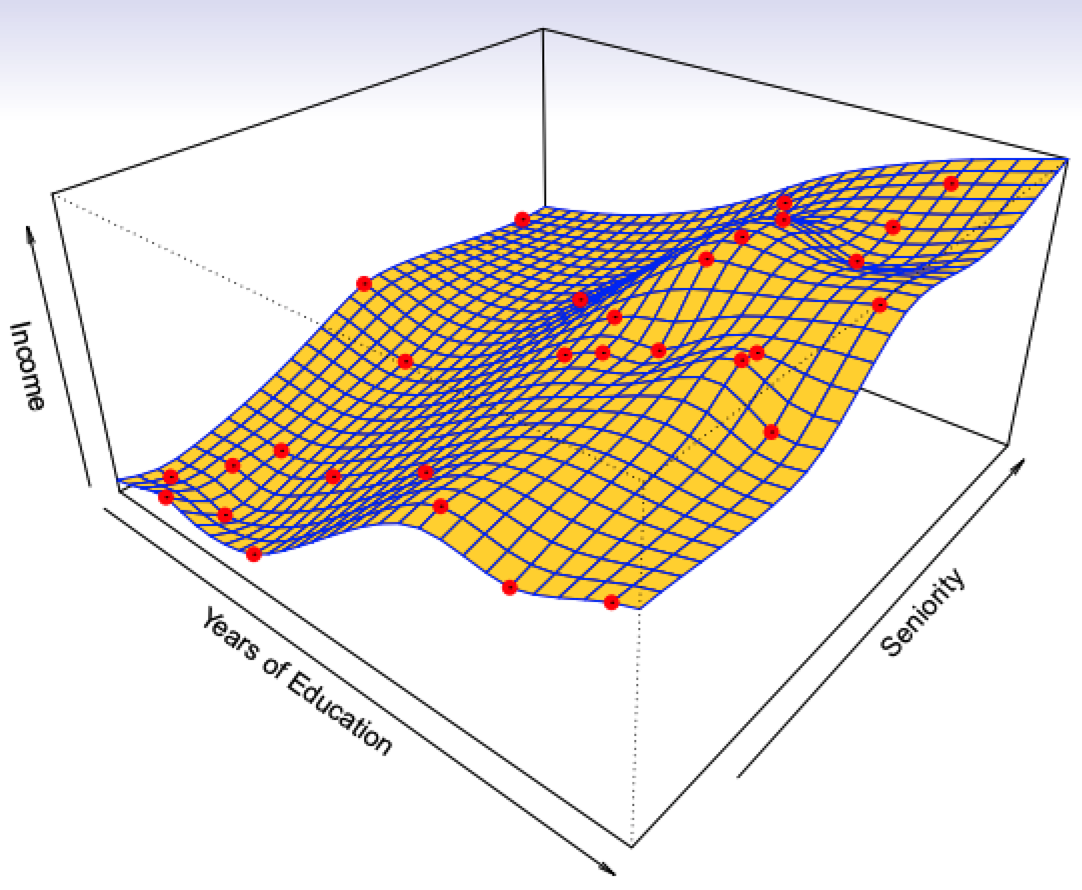


$$\hat{f}_L = \hat{\beta}_0 + \hat{\beta}_1 \times \text{education} + \hat{\beta}_2 \times \text{seniority}$$

More flexible regression model $\hat{f}_S$(education, seniority) fit to the simulated data: **thin-plate spline**.



Indeed, we can fine-tune the roughness of the spline fit. So the fitted model makes no errors on all the training data! (**Overfitting**)



Trade-offs

1. Prediction **accuracy** versus **interpretability**.
   - Linear models are easy to interpret; thin-plate splines are not.
2. **Good fit** versus **over-fit** or **under-fit**.
   - How do we know when the fit is just right?
3. **Parsimony** versus **black-box**
   - In general, a simpler model involving fewer variables is better than a black-box predictor involving them all.

Quantify model accuracy

- Suppose we fit a model $\hat{f}(x)$ to some training data $\mathsf{Tr} = \{x_i, y_i\}_{i=1}^N$. We want to know how well it performs.
- We could compute the average squared prediction error over Tr

$$\mathrm{MSE}_{\mathsf{Tr}} = \mathrm{Ave}_{i \in \mathsf{Tr}} \left[y_i - \hat{f}(x_i) \right]^2$$

Quantify model accuracy

- Suppose we fit a model $\hat{f}(x)$ to some training data $\textsf{Tr} = \{x_i, y_i\}_{i=1}^{N}$. We want to know how well it performs.
- We could compute the average squared prediction error over $\textsf{Tr}$

$$\mathrm{MSE}_{\textsf{Tr}} = \mathrm{Ave}_{i \in \textsf{Tr}}\left[y_i - \hat{f}(x_i)\right]^2$$

- Of course, simply looking at $\mathrm{MSE}_{\textsf{Tr}}$ is biased in favor for more overfit models

Quantify model accuracy

- To overcome overfitting, we should compute MSE using *fresh* **test data**: $\mathsf{Te} = \{x_i, y_i\}_{i=1}^{M}$

$$\mathrm{MSE}_{\mathsf{Te}} = \mathrm{Ave}_{i \in \mathsf{Te}} \left[y_i - \hat{f}(x_i) \right]^2$$

Quantify model accuracy

- To overcome overfitting, we should compute MSE using fresh **test data**: $\mathsf{Te} = \{x_i, y_i\}_{i=1}^{M}$

$$\mathrm{MSE}_{\mathsf{Te}} = \mathrm{Ave}_{i \in \mathsf{Te}} \left[y_i - \hat{f}(x_i) \right]^2$$

Principle: use *different* datasets for *training* and *testing*!

- In the industry practice, a standard workflow involves three separate datasets: **training data**, **validation data** and **test data**.

Training / Validation/ Test data

Suppose we have three candidate models:

1. $f_1(x_1) = \beta_0 + \beta_1 x_1$
2. $f_2(x_1) = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2$
3. $f_3(x_1) = \beta_0 + \beta_1 x_1 + \beta_2 \sqrt{x_1}$

Training / Validation/ Test data

Suppose we have three candidate models:

1. $f_1(x_1) = 1.2 + 0.5x_1$
2. $f_2(x_1) = 1.5 + 0.2x_1 + 0.2x_1^2$
3. $f_3(x_1) = 1.0 + 0.6x_1 - 0.2\sqrt{x_1}$

Step 1: use training data to **train** all three models.
Usually, this is to choose $\beta_0, \beta_1, \beta_2$ to minimize $\mathrm{MSE}_{\mathsf{Tr}}$.

Training / Validation/ Test data

Suppose we have three candidate models:

1. $f_1(x_1) = 1.2 + 0.5x_1 \implies \text{MSE}_{\text{Val}} = 30$
2. $f_2(x_1) = 1.5 + 0.2x_1 + 0.2x_1^2 \implies \text{MSE}_{\text{Val}} = 20$ (Winner)
3. $f_3(x_1) = 1.0 + 0.6x_1 - 0.2\sqrt{x_1} \implies \text{MSE}_{\text{Val}} = 40$

Step 2: use validation data to **select** the "best" one with minimal MSE_{Val}.

Training / Validation/ Test data

Suppose we have three candidate models:

1. $f_1(x_1) = 1.2 + 0.5x_1$
2. $f_2(x_1) = 1.5 + 0.2x_1 + 0.2x_1^2 \implies \text{MSE}_{\text{Te}} = 28$
3. $f_3(x_1) = 1.0 + 0.6x_1 - 0.2\sqrt{x_1}$

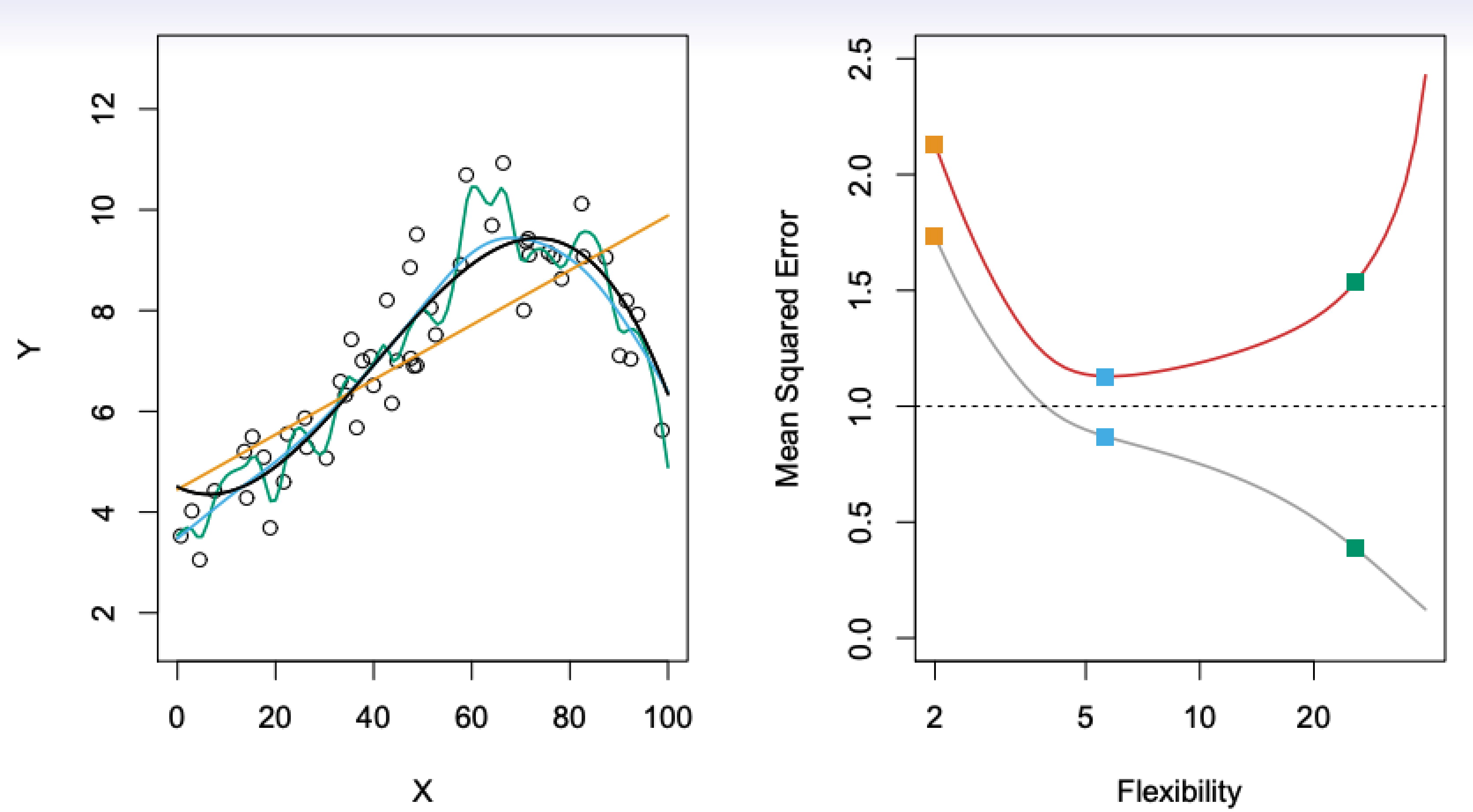
Step 3: use test data to **assess** model performance on new data. The reported model performance should be based on the new test data.

Summary

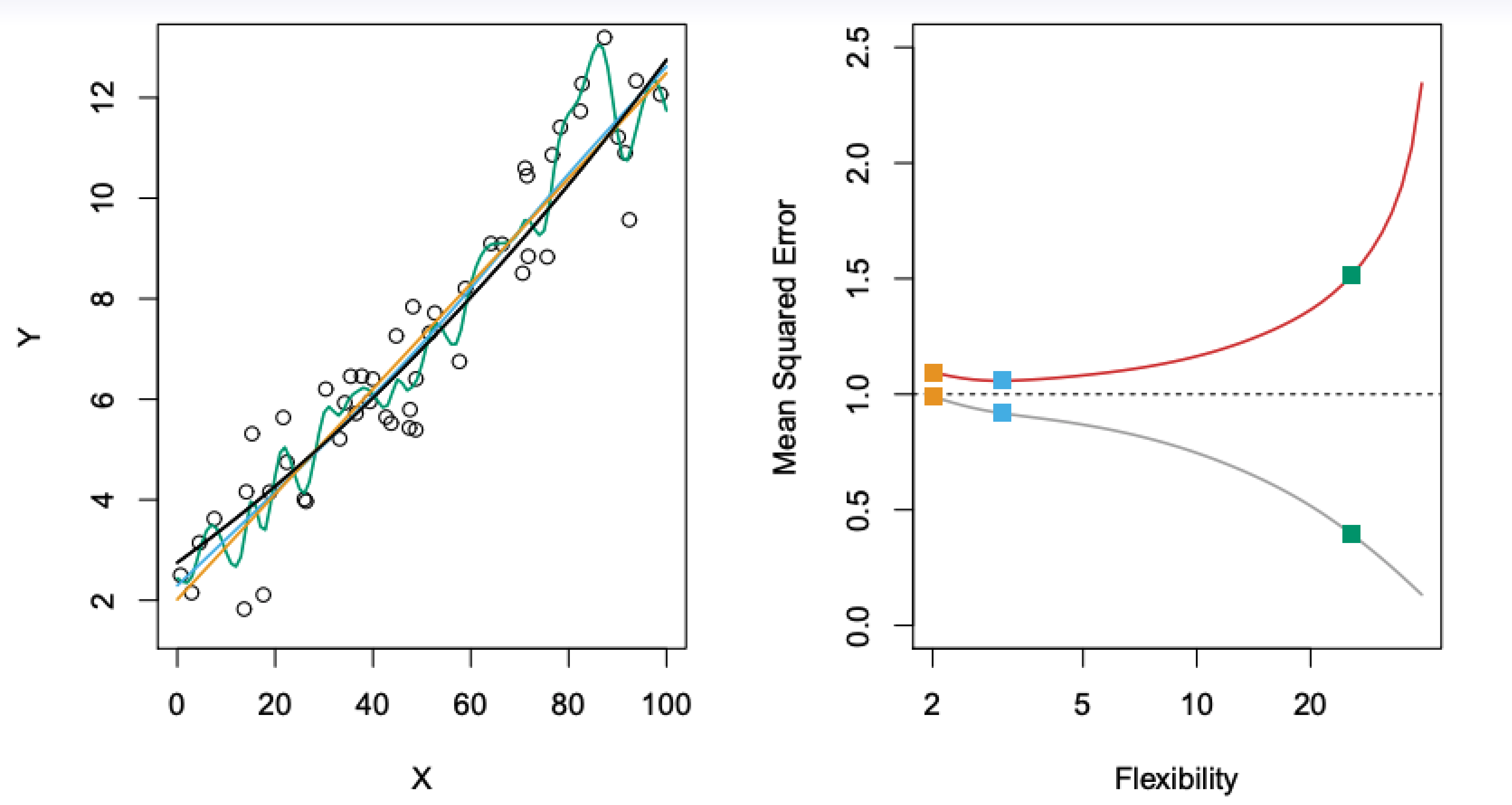
To overcome overfit, use fresh data for model selection (**Validation data**):

- Otheriwse, more flexible models (that are more likely to overfit data) will always win.

To have an unbiased evaluation of the selected model, use fresh data to evaluate model performance (**Test data**).

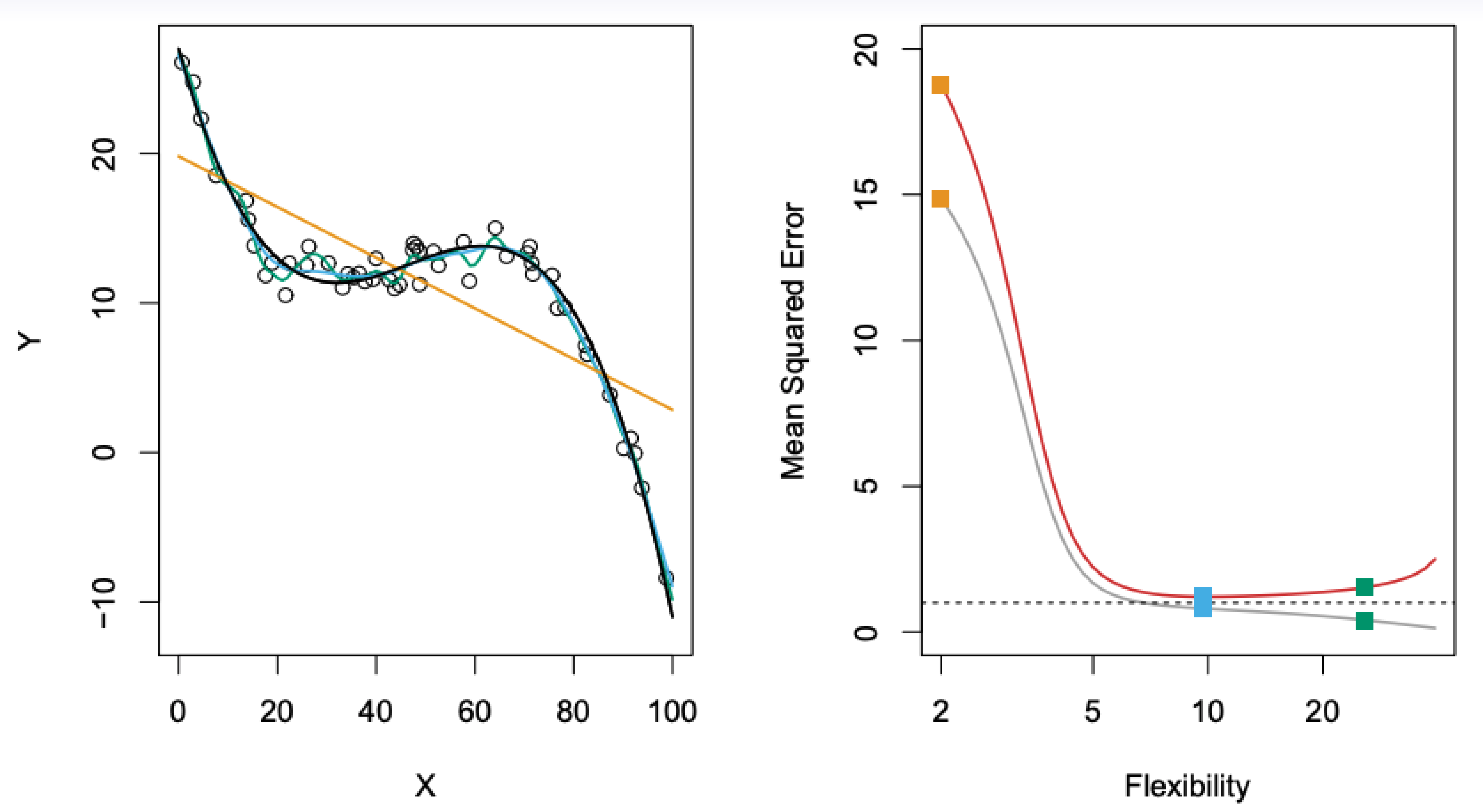


Example 1: Black curve is truth. Red curve (right) is MSE on $\mathcal{T}e$, grey curve is MSE on $\mathcal{T}r$. Orange, blue and green curves/squares correspond to fits of different flexibility.



Example 2: Here the truth is smoother.

So the smoother fit (blue) and linear model (orange) do well.



Example 3: Here the truth is wiggly and the noise is low.
So the most flexible fits (green) perform best.

Bias-Variance Trade-off

We have fit a model to some training data Tr.

Let (x_0, y_0) be a test observation drawn from the population.
If the true model is $Y = f(X) + \epsilon$, then

$$\mathbb{E}\left[(y_0 - \hat{f}(x_0))^2\right] = \mathrm{Var}[\hat{f}(x_0)] + \mathrm{Var}[\epsilon] + \left(\mathrm{Bias}[\hat{f}(x_0)]\right)^2$$

Bias-Variance Trade-off

We have fit a model to some training data Tr.

Let (x_0, y_0) be a test observation drawn from the population.
If the true model is $Y = f(X) + \epsilon$, then

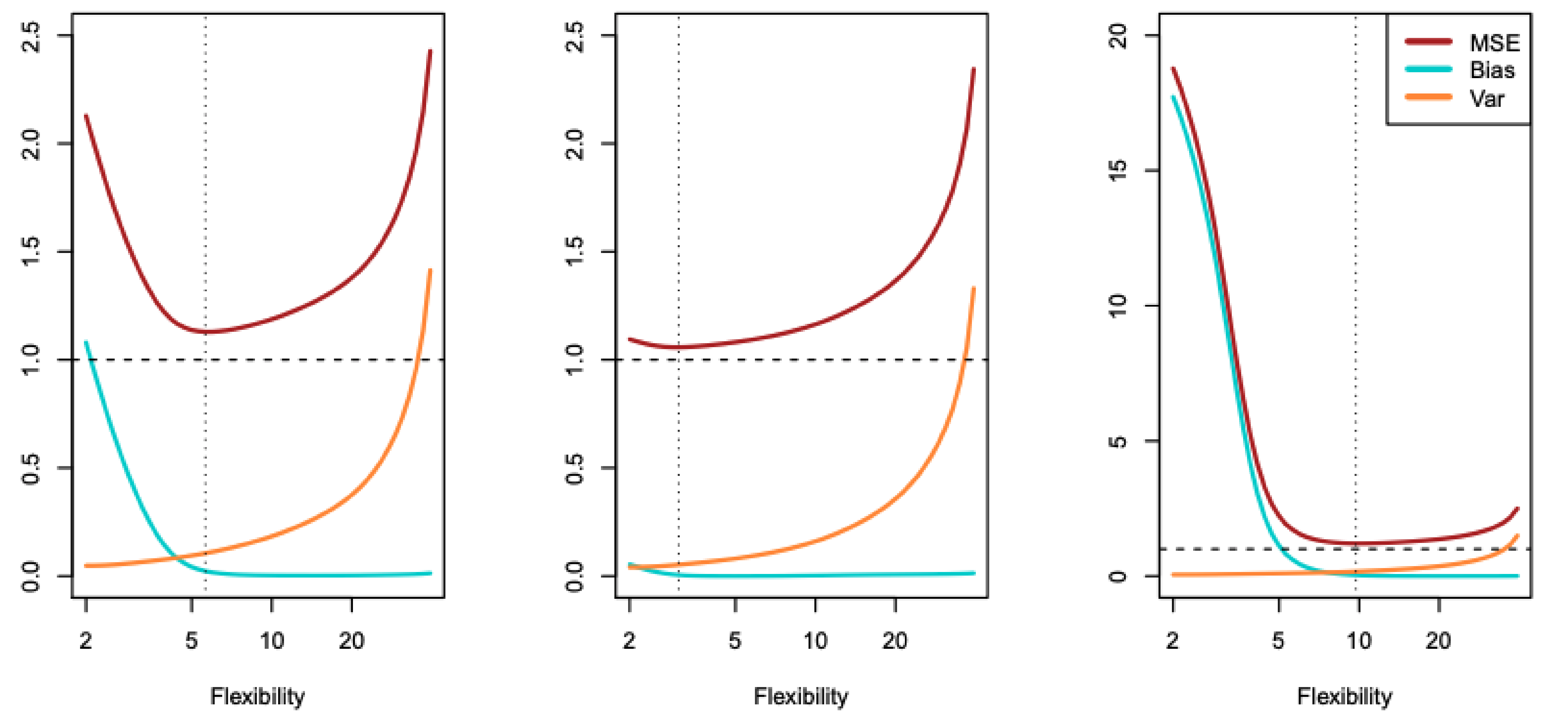
$$\mathbb{E}\left[(y_0 - \hat{f}(x_0))^2\right] = \text{Var}[\hat{f}(x_0)] + \text{Var}[\epsilon] + \Big(\text{Bias}[\hat{f}(x_0)]\Big)^2$$

- The expectation averages over the variability of y_0 and the variability in Tr.
- $\text{Bias}[\hat{f}(x_0)] = \mathbb{E}[\hat{f}(x_0)] - f(x_0)$

Bias-Variance Trade-off

$$\mathbb{E}\left[(y_0 - \hat{f}(x_0))^2\right] = \mathrm{Var}[\hat{f}(x_0)] + \mathrm{Var}[\epsilon] + \left(\mathrm{Bias}[\hat{f}(x_0)]\right)^2$$

- Typically as the *flexibility* of $\hat{f}$ increases, its **variance increases**, and its bias decreases.
- So choosing the flexibility based on average test error amounts to a *bias-variance trade-off*.
- Bias-Variance trade-off provides a new perspective to understand overfitting.



Bias-variance tradeoff for the three examples

Homework: Explain the three graphs in the previous slide

- I.e., explain the bias-variance tradeoff in the three cases in **your own words**. You do not need to use math in the explanations, but feel free to use some math if necessary.
- In the first plot, the true model is non-linear and almost quadratic; in the second plot, the true model is almost linear; in the third, the true model is non-linear but the noise is very small.
- Hint: You may read the Wiki on bias-variance tradeoff for inspirations.
- My Email: hlei@hnu.edu.cn