Function fitting

Lecture 4; Reading: ISLR sections 2.1, 3.2.1, 3.5

IFT6758, Fall 2020

Leveraging Input → Output relationship in data

Leveraging Input → Output relationship in data

Examples:

- Genetic profile → Chance of developing disease
- Person's characteristics → Whether they'll vote
- Marketing plan → Total sales amount
- Image pixel values → What's in the image

Leveraging Input → Output relationship in data

Examples:

- Genetic profile → Chance of developing disease
- Person's characteristics → Whether they'll vote
- Marketing plan → Total sales amount
- Image pixel values → What's in the image

Mathematical expression

- $x_i = (x_{i_1}, \ldots, x_{i_p})$ ← **inputs**
- $\boldsymbol{y_i} = \boldsymbol{f}(\boldsymbol{x_i})$ ← **inputs to output relationship**

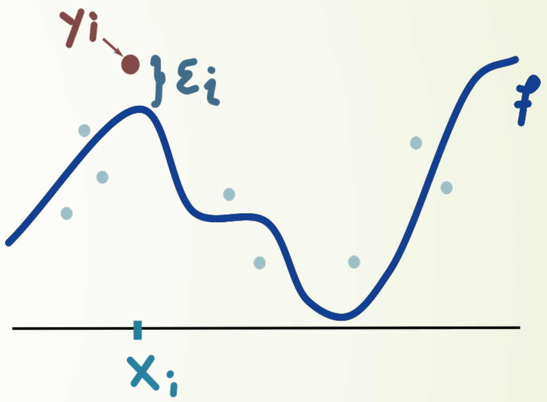
Observed and predicted values

Observed and predicted values

- $y_i = f(x_i) + \epsilon_i$

Observed and predicted values

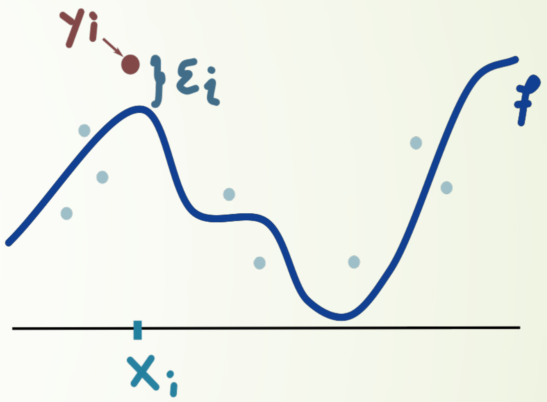
- $y_i = f(x_i) + \epsilon_i$



Observed and predicted values

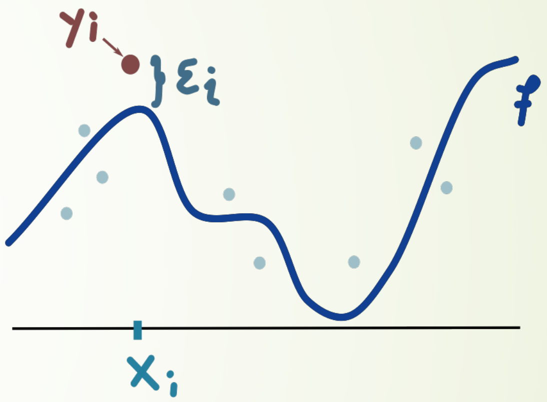
- $y_i = f(x_i) + \epsilon_i$

Why is this decomposition?



Observed and predicted values

- $y_i = f(x_i) + \epsilon_i$

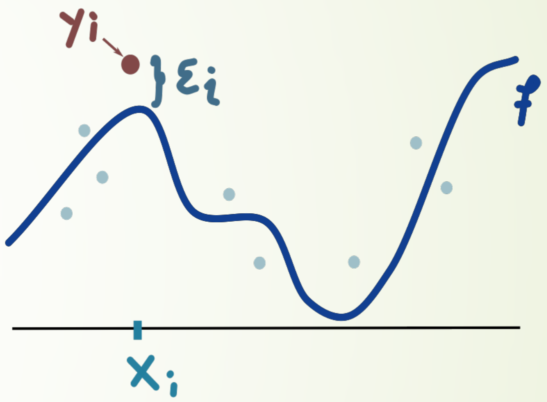


Why is this decomposition?

- $\boldsymbol{f}$ describes **systematic variation** in $\boldsymbol{y_i}$

Observed and predicted values

- $y_i = f(x_i) + \epsilon_i$



Why is this decomposition?

- $\boldsymbol{f}$ describes **systematic variation** in $\boldsymbol{y_i}$
- $\boldsymbol{\epsilon_i}$ reflects variations whose source is unknown

to us. Consider coin tosses.

Why care?

Why care?

- **Why not just visualize the data we have?**

Why care?

- **Why not just visualize the data we have?**
- **Reason 1: Prediction**
  - We may want the $\boldsymbol{y_i}$ corresponding to an input x_i
  - Inputs may be much easier to collect than outputs
- **Reason 2: Inference**
  - We may care about the form of $\boldsymbol{f}$, for personal understanding
  - e.g., is a particular input relevant at all?

Why care?

- **Why not just visualize the data we have?**
- **Reason 1: Prediction**
  - We may want the $\boldsymbol{y_i}$ corresponding to an input x_i
  - Inputs may be much easier to collect than outputs
- **Reason 2: Inference**
  - We may care about the form of $\boldsymbol{f}$, for personal understanding
  - e.g., is a particular input relevant at all?

We want **quantitative estimates**, not visual summaries

Estimates and Predictions

Estimates and Predictions

- In reality, we won't know $\boldsymbol{f}$

Estimates and Predictions

- In reality, we won't know $\boldsymbol{f}$

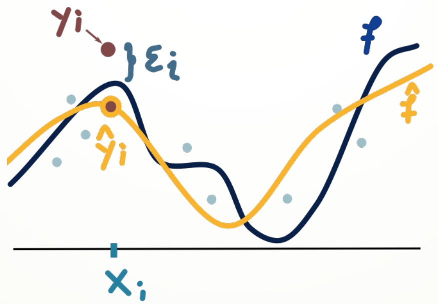
- We estimate it from data and call the result $\hat{f}$

Estimates and Predictions

- In reality, we won't know $\boldsymbol{f}$
- We estimate it from data and call the result $\hat{\boldsymbol{f}}$
- To predict $\boldsymbol{y_i}$ for some input $\boldsymbol{x_i}$, we'd use $\hat{\boldsymbol{y}} = \hat{\boldsymbol{f}}(\boldsymbol{x_i})$

Estimates and Predictions

- In reality, we won't know $\boldsymbol{f}$
- We estimate it from data and call the result $\hat{\boldsymbol{f}}$
- To predict $\boldsymbol{y_i}$ for some input $\boldsymbol{x_i}$, we'd use $\hat{\boldsymbol{y}} = \hat{\boldsymbol{f}}(\boldsymbol{x_i})$



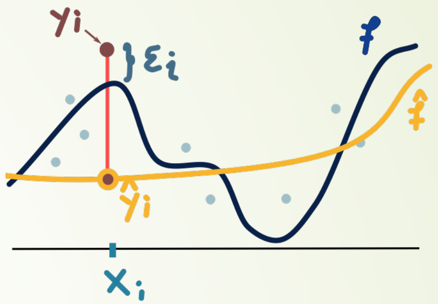
Sources of error

Sources of error

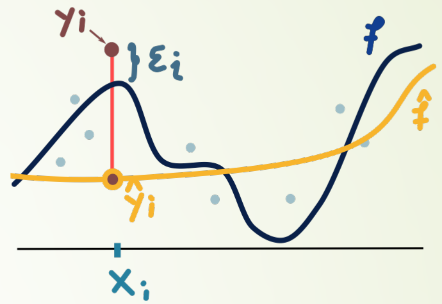
- **Approximation error:** $\hat{f}$ isn't close to f
  - This error is *reducible* (use a better algorithm)

Sources of error

- **Approximation error:** $\hat{f}$ isn't close to f
  - This error is *reducible* (use a better algorithm)



Sources of error

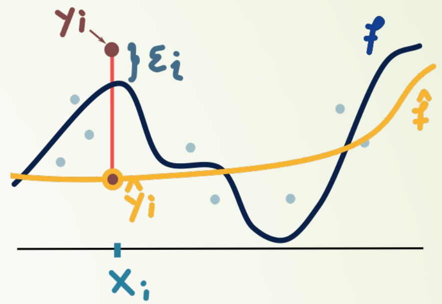


- **Approximation error:** $\hat{f}$ isn't close to f
  - This error is *reducible* (use a better algorithm)

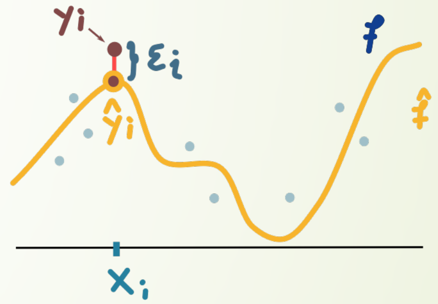
- **Irreducible error:** y_i isn't close to $f(x_i)$
  - Incur this error even if f were perfectly known

Sources of error

- **Approximation error:** $\hat{f}$ isn't close to f
  - This error is *reducible* (use a better algorithm)

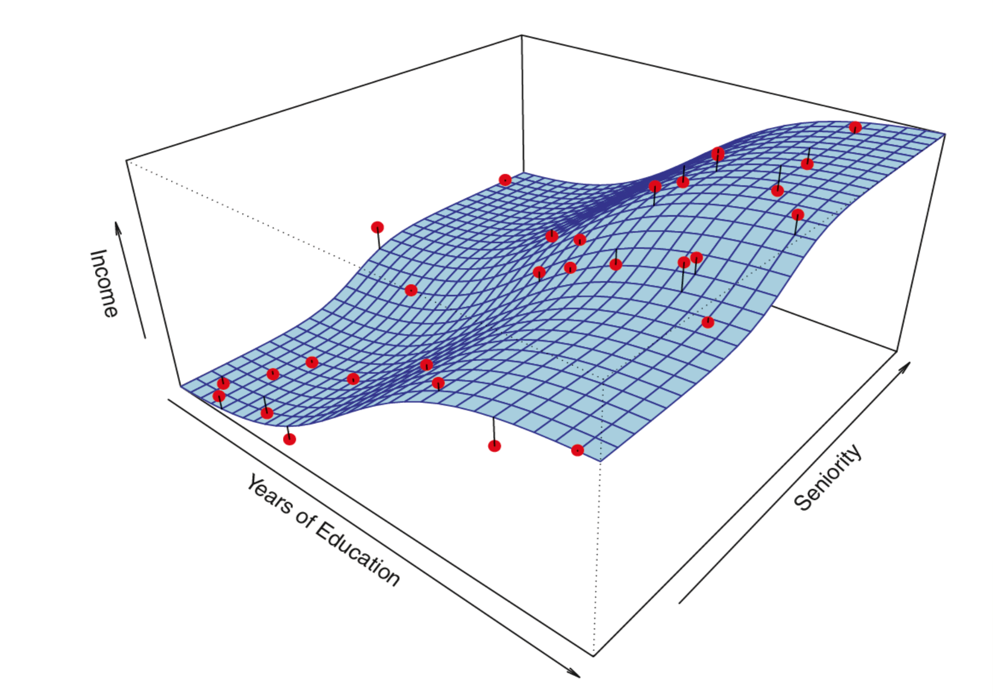


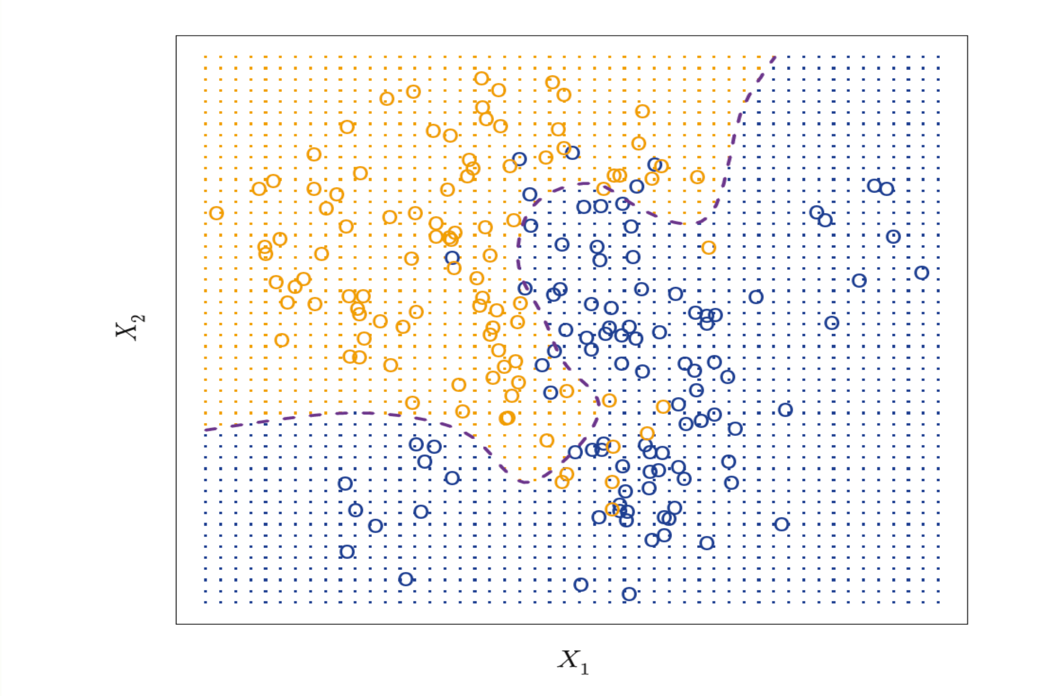
- **Irreducible error:** y_i isn't close to $f(x_i)$
  - Incur this error even if f were perfectly known



Extendable to high dimension

Extendable to high dimension





How to find $\hat{f}$?

How to find $\hat{f}$?

- Propose a model family F : e.g., set of all linear functions of x_i

How to find $\hat{f}$?

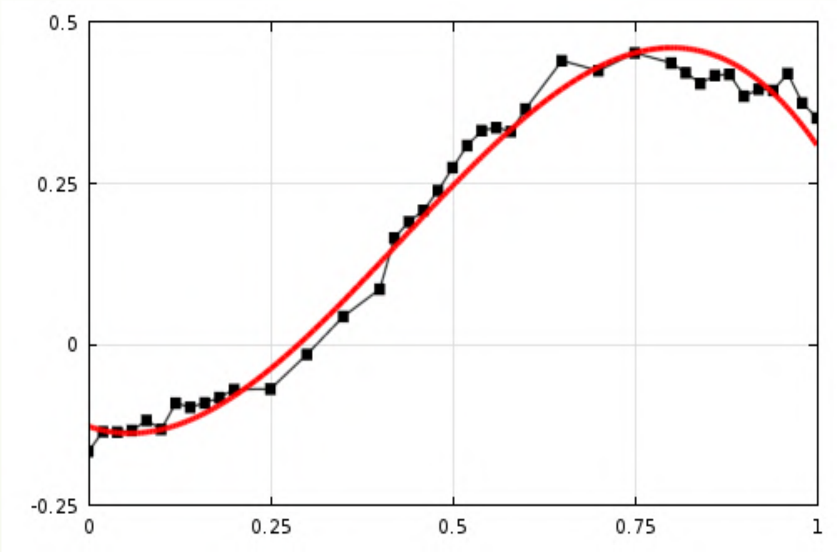
- Propose a model family F : e.g., set of all linear functions of x_i
- Define a procedure to choose $\hat{f} \in F$ based on the data

How to find $\hat{f}$?

- Propose a model family F : e.g., set of all linear functions of x_i
- Define a procedure to choose $\hat{f} \in F$ based on the data
  - E.g., the choice $\hat{f}$ that minimizes $\sum_i (y_i - \hat{f}(x_i))^2$

How to find $\hat{f}$?

- Propose a model family F : e.g., set of all linear functions of x_i
- Define a procedure to choose $\hat{f} \in F$ based on the data
  - E.g., the choice $\hat{f}$ that minimizes $\sum_i (y_i - \hat{f}(x_i))^2$



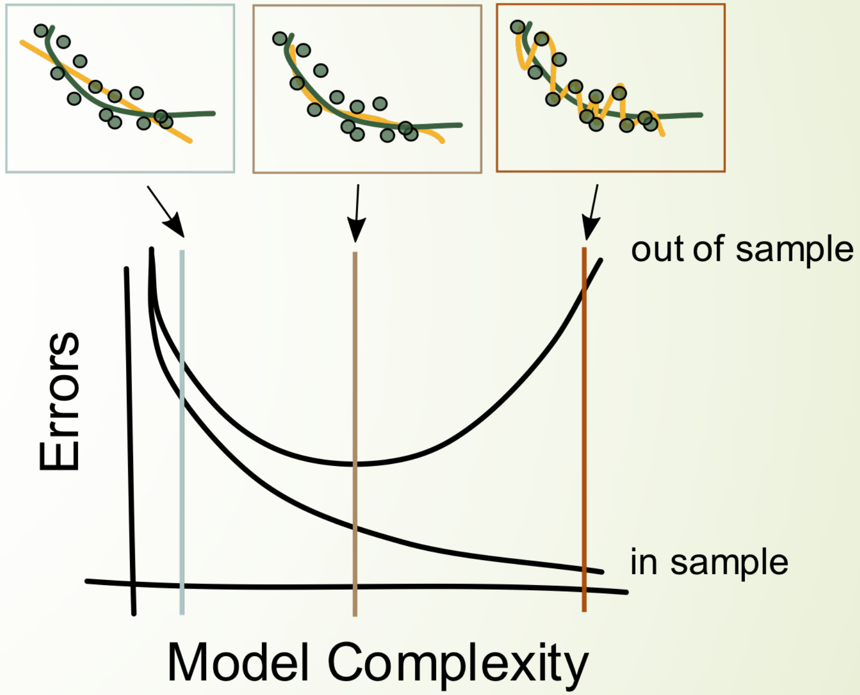
Find a ‘good enough’ fit

Find a ‘good enough’ fit

- Ultimately, you want your **model to perform well on out-of-sample data**

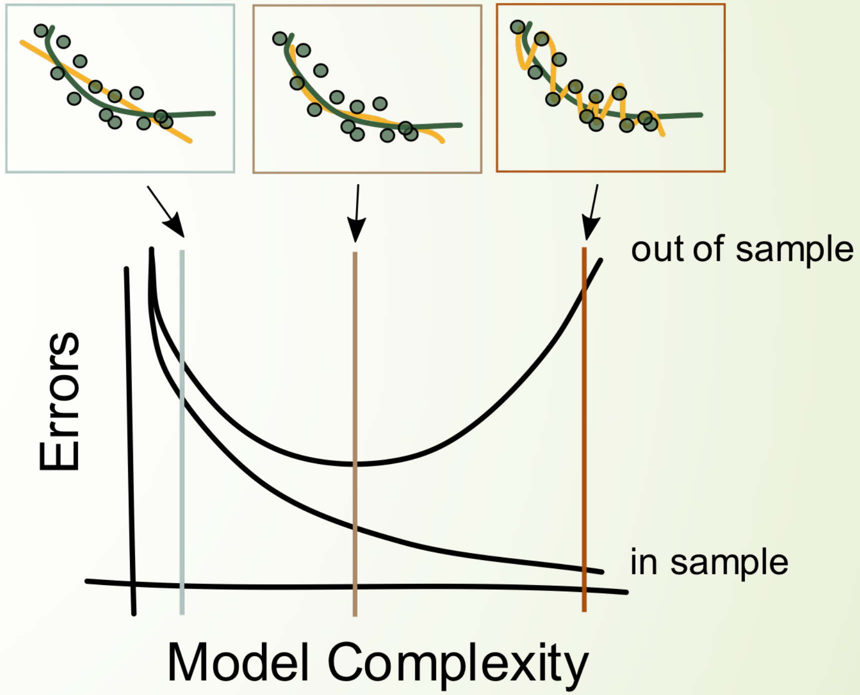
Find a 'good enough' fit

- Ultimately, you want your **model to perform well on out-of-sample data**



Find a 'good enough' fit

- Ultimately, you want your **model to perform well on out-of-sample data**
- **Bias-Variance trade-off:**
  - **Bias ≡ model inaccuracy**
  - **Variance: Different samples ⟶ Different $\hat{f}$**

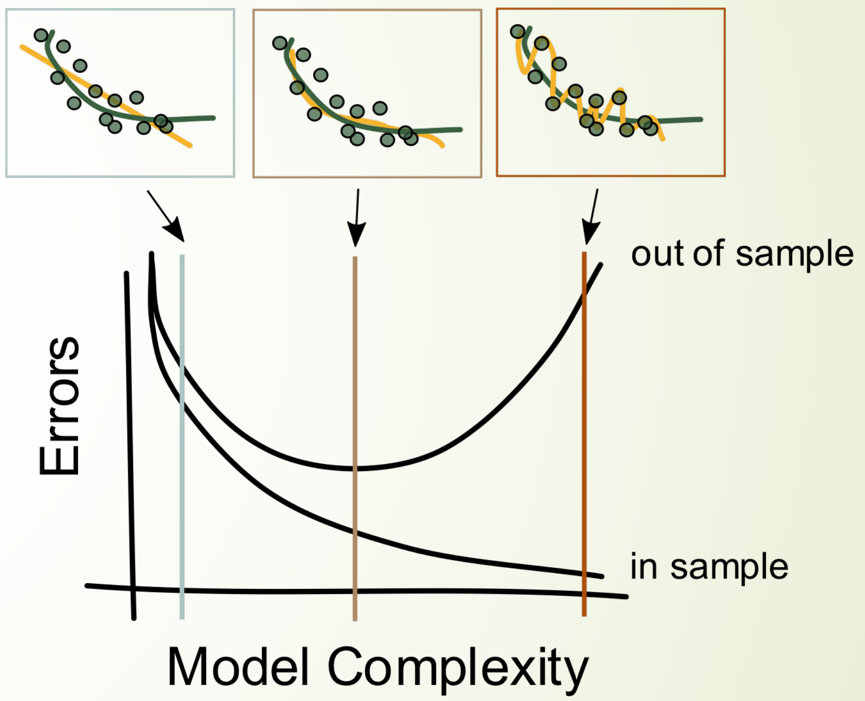


Find a 'good enough' fit

- Ultimately, you want your **model to perform well on out-of-sample data**

- **Bias-Variance trade-off:**
  - **Bias ≡ model inaccuracy**
  - **Variance: Different samples** $\longrightarrow$ **Different $\hat{f}$**

- **Incurring some bias can** lead to **better stability** and overall **better predictions**



Model Flexibility - Takeaways

Model Flexibility - Takeaways

- If you **don't have too many samples**, you should prefer a **simpler model**

Model Flexibility - Takeaways

- If you **don't have too many samples**, you should prefer a **simpler model**

- If you have **many samples**, you can afford a **more complex model**

Model Flexibility - Takeaways

- If you **don't have too many samples**, you should prefer a **simpler model**
- If you have **many samples**, you can afford a **more complex model**
- We'll need **some sort of mechanism** to tell which regime we're in

Model Flexibility - Takeaways

- If you **don't have too many samples**, you should prefer a **simpler model**
- If you have **many samples**, you can afford a **more complex model**
- We'll need **some sort of mechanism** to tell which regime we're in
- Examples of different model families

Model Flexibility - Takeaways

- If you **don't have too many samples**, you should prefer a **simpler model**
- If you have **many samples**, you can afford a **more complex model**
- We'll need **some sort of mechanism** to tell which regime we're in
- Examples of different model families
  - Useful tutorial: pyGAM

Model Flexibility - Takeaways

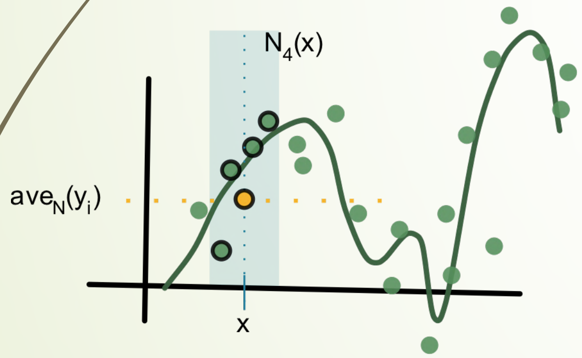
- If you **don't have too many samples**, you should prefer a **simpler model**
- If you have **many samples**, you can afford a **more complex model**
- We'll need **some sort of mechanism** to tell which regime we're in
- Examples of different model families
  - Useful tutorial: pyGAM
- **Reading**: ISLR 2.1, 3.2.1, 3.5

Algorithms: K-Nearest Neighbors (KNN) Regression

- K fixed and given
- **Samples:** $(x_i, \mathrm{y_i})_{\{\mathrm{i=1}\}}^{\mathrm{N}}$
- **Estimate data generating function:** $\hat{f}(x) = \frac{1}{K}\sum_{i \in N_K(x)} y_i$
- N_K: K **nearest neighbors of** x within the **training set**

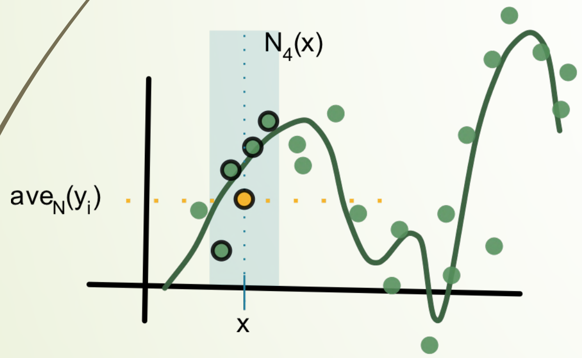
Algorithms: K-Nearest Neighbors (KNN) Regression

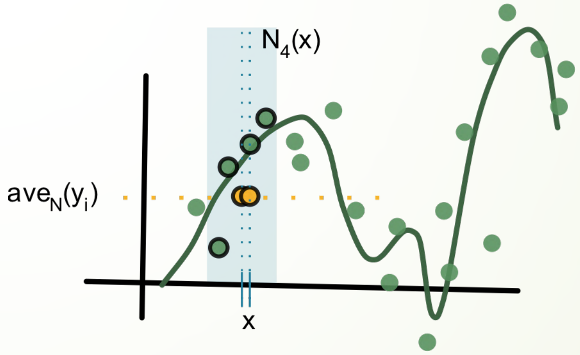
- K fixed and given
- **Samples:** $(x_i, \mathrm{y_i})_{\{i=1\}}^{\mathrm{N}}$
- **Estimate data generating function:** $\hat{f}(x) = \frac{1}{K}\sum_{i \in N_K(x)} y_i$
- N_K: K **nearest neighbors of** x within the **training set**



Algorithms: K-Nearest Neighbors (KNN) Regression

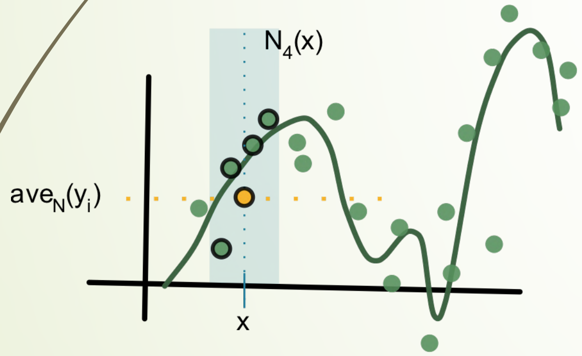
- K fixed and given
- **Samples:** $(x_i, y_i)_{\{i=1\}}^{N}$
- **Estimate data generating function:** $\hat{f}(x) = \frac{1}{K}\sum_{i \in N_K(x)} y_i$
- N_K: K **nearest neighbors of** x within the **training set**

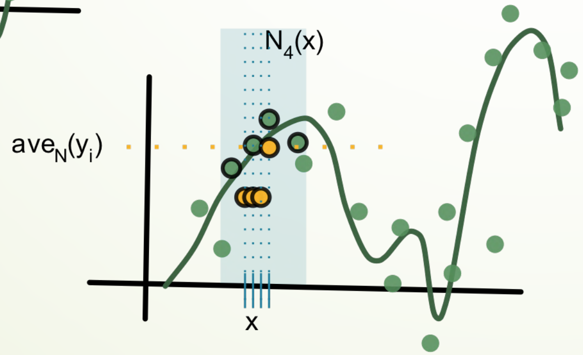


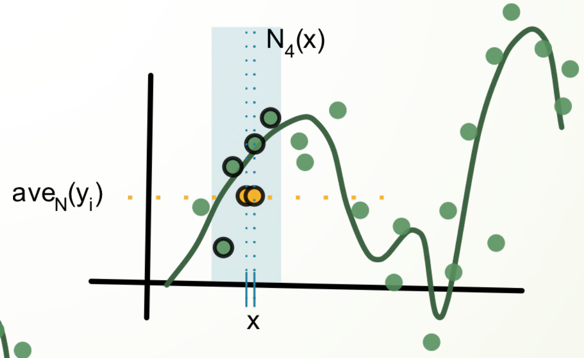


Algorithms: K-Nearest Neighbors (KNN) Regression

- K fixed and given
- **Samples:** $(x_i, \mathrm{y_i})_{\{\mathrm{i=1}\}}^{\mathrm{N}}$
- **Estimate data generating function:** $\hat{f}(x) = \frac{1}{K} \sum_{i \in N_K(x)} y_i$
- N_K: K **nearest neighbors of** x within the **training set**

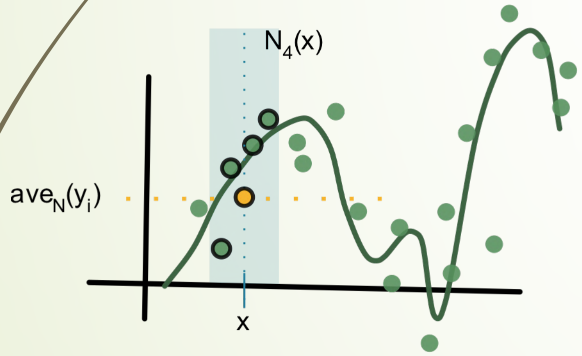


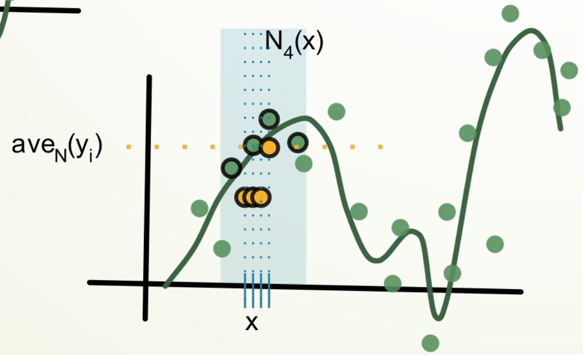


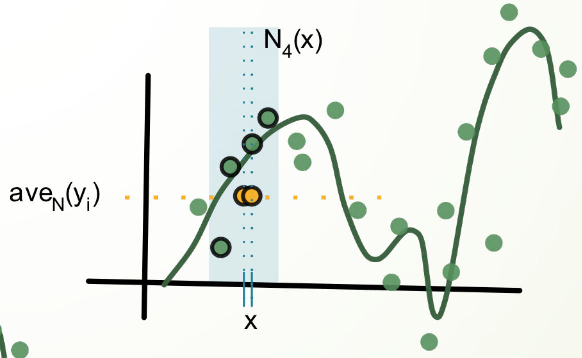


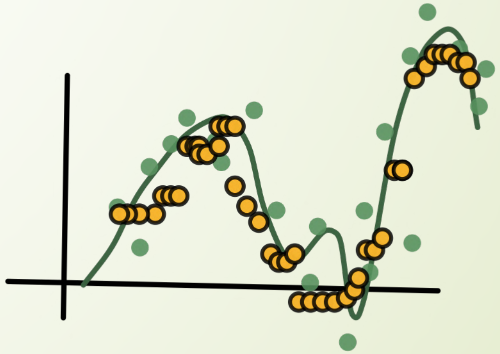
Algorithms: K-Nearest Neighbors (KNN) Regression

- K fixed and given
- **Samples:** $(x_i, \mathrm{y_i})_{\{\mathrm{i=1}\}}^{\mathrm{N}}$
- **Estimate data generating function:** $\hat{f}(x) = \frac{1}{K}\sum_{i \in N_K(x)} y_i$
- N_K: K **nearest neighbors of** x within the **training set**









Impact of $\boldsymbol{K}$: Bias-Variance Trade-off

Impact of K: Bias-Variance Trade-off

- **Model complexity** is controlled by the size of the neighborhood

Impact of $\boldsymbol{K}$: Bias-Variance Trade-off

- **Model complexity** is controlled by the size of the neighborhood
  - Large $\boldsymbol{K}$ $\longrightarrow$ Lower variance, larger bias
  - Small $\boldsymbol{K}$ $\longrightarrow$ Higher variance, smaller bias

Impact of $\boldsymbol{K}$: Bias-Variance Trade-off

- **Model complexity** is controlled by the size of the neighborhood
  - Large $\boldsymbol{K}$ $\longrightarrow$ Lower variance, larger bias
  - Small $\boldsymbol{K}$ $\longrightarrow$ Higher variance, smaller bias

  - Larger $\boldsymbol{K}$ learns smoother functions; smaller $\boldsymbol{K}$ can match more complex functions when the sampling density is high enough

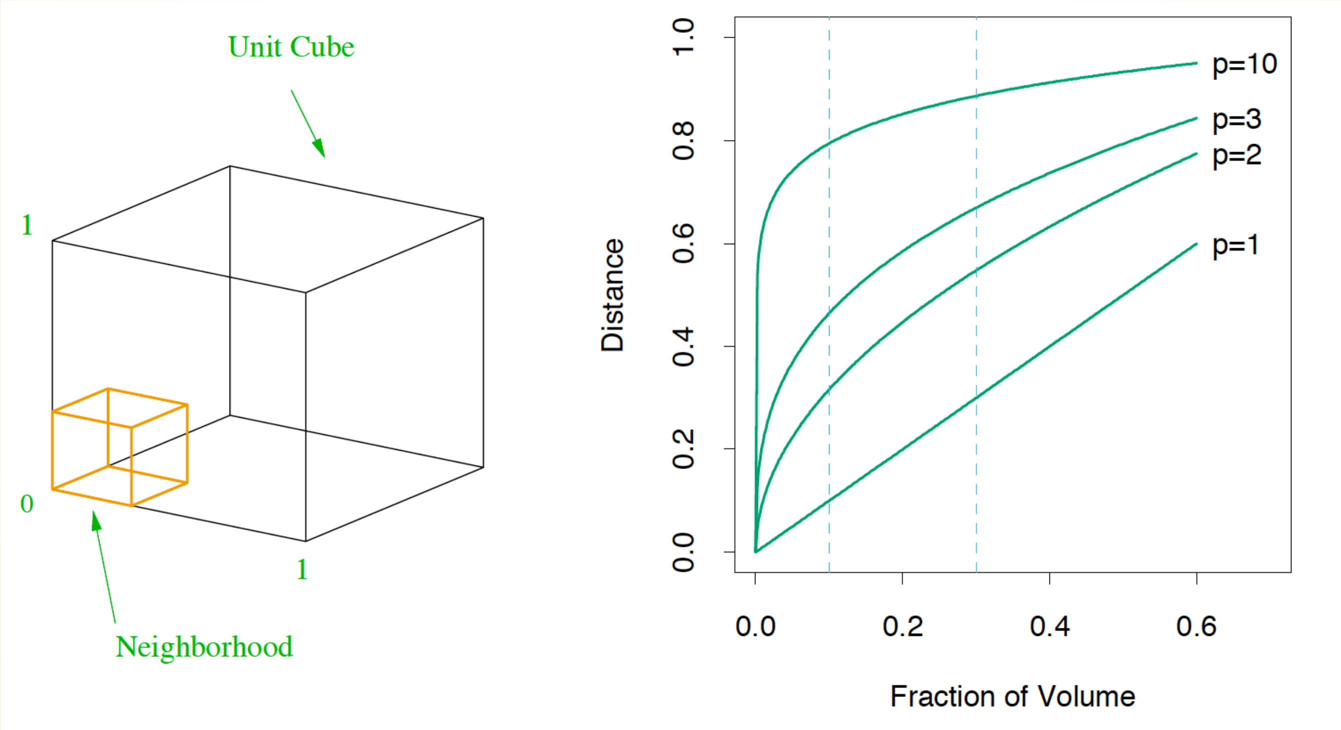
Curse of dimensionality

Curse of dimensionality

- The density of samples decreases with increase in dimension

Curse of dimensionality

- The density of samples decreases with increase in dimension



Curse of dimensionality

- The density of samples decreases with increase in dimension
- Lack of close neighbors causes increases in bias and variance

Curse of dimensionality

- The density of samples decreases with increase in dimension
- Lack of close neighbors causes increases in bias and variance
- Need to look at almost the whole space to make a prediction

Curse of dimensionality

- The density of samples decreases with increase in dimension
- Lack of close neighbors causes increases in bias and variance
- Need to look at almost the whole space to make a prediction
- It means you average over points that are quite different

KNN Classification

KNN Classification

- Categorical variables belonging to some classes

KNN Classification

- Categorical variables belonging to some classes
- The probability that a location $\boldsymbol{x}$ gets assigned to class $\boldsymbol{j}$ is approximated by

$$\hat{p}_k(x) = \frac{1}{K} \sum_{i \in N_K(x)} \mathbb{I}(y_i = j)$$

KNN Classification

- Categorical variables belonging to some classes
- The probability that a location $\boldsymbol{x}$ gets assigned to class $\boldsymbol{j}$ is approximated by

$$\hat{p}_k(x) = \frac{1}{K} \sum_{i \in N_K(x)} \mathbb{I}(y_i = j)$$

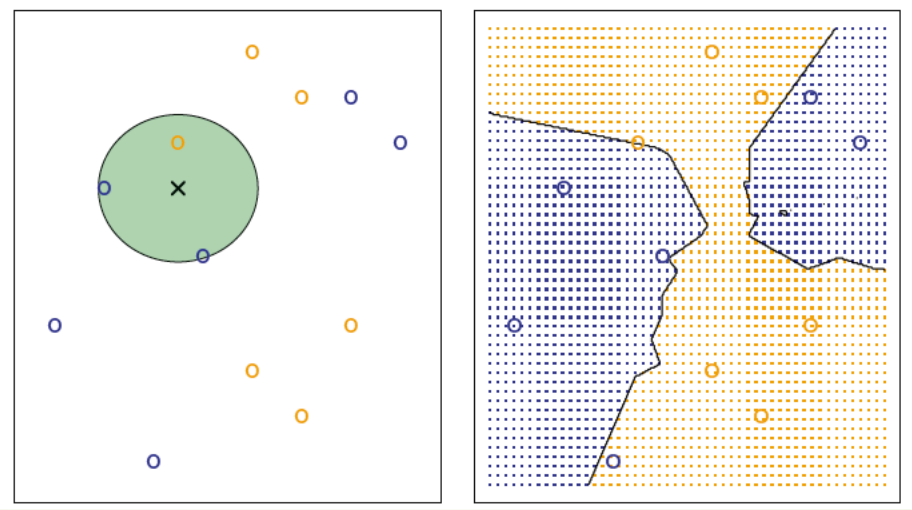
- Assign the class with highest probability. This is like taking majority votes.

KNN Classification

- Categorical variables belonging to some classes
- The probability that a location $\boldsymbol{x}$ gets assigned to class $\boldsymbol{j}$ is approximated by

$$\hat{p}_k(x) = \frac{1}{K} \sum_{i \in N_K(x)} \mathbb{I}(y_i = j)$$

- Assign the class with highest probability. This is like taking majority votes.



Next up: Linear regression, logistic regression, decision trees

- Go to this page