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

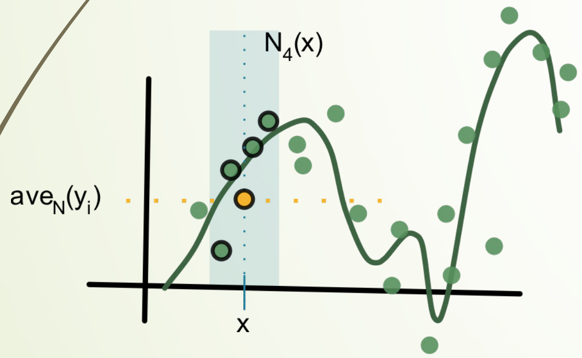
- If you **don't have too many samples**, you should prefer a **simpler model**
- If you have **many samples**, you can afford a **more complex model**
- We'll need **some sort of mechanism** to tell which regime we're in
- Examples of different model families
  - Useful tutorial: pyGAM
- **Reading**: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

Algorithms: K-Nearest Neighbors (KNN) Regression

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

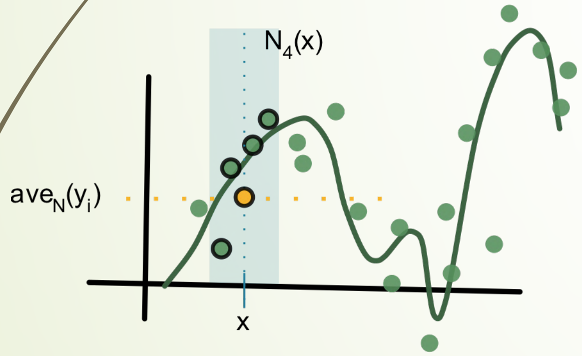
Algorithms: K-Nearest Neighbors (KNN) Regression

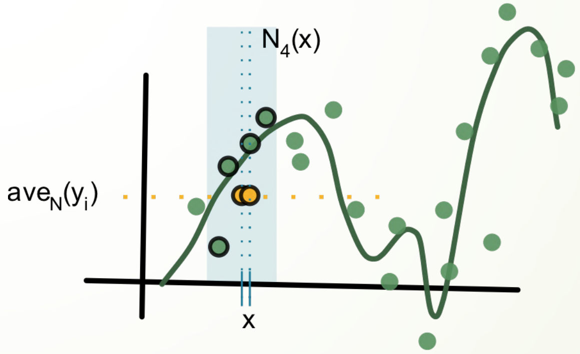
- K fixed and given
- **Samples:** $(x_i, \mathrm{y_i})_{\{i=1\}}^{N}$
- **Estimate data generating function:** $\hat{f}(x) = \frac{1}{K}\sum_{i \in N_K(x)} y_i$
- N_K: K **nearest neighbors of** x within the **training set**



Algorithms: K-Nearest Neighbors (KNN) Regression

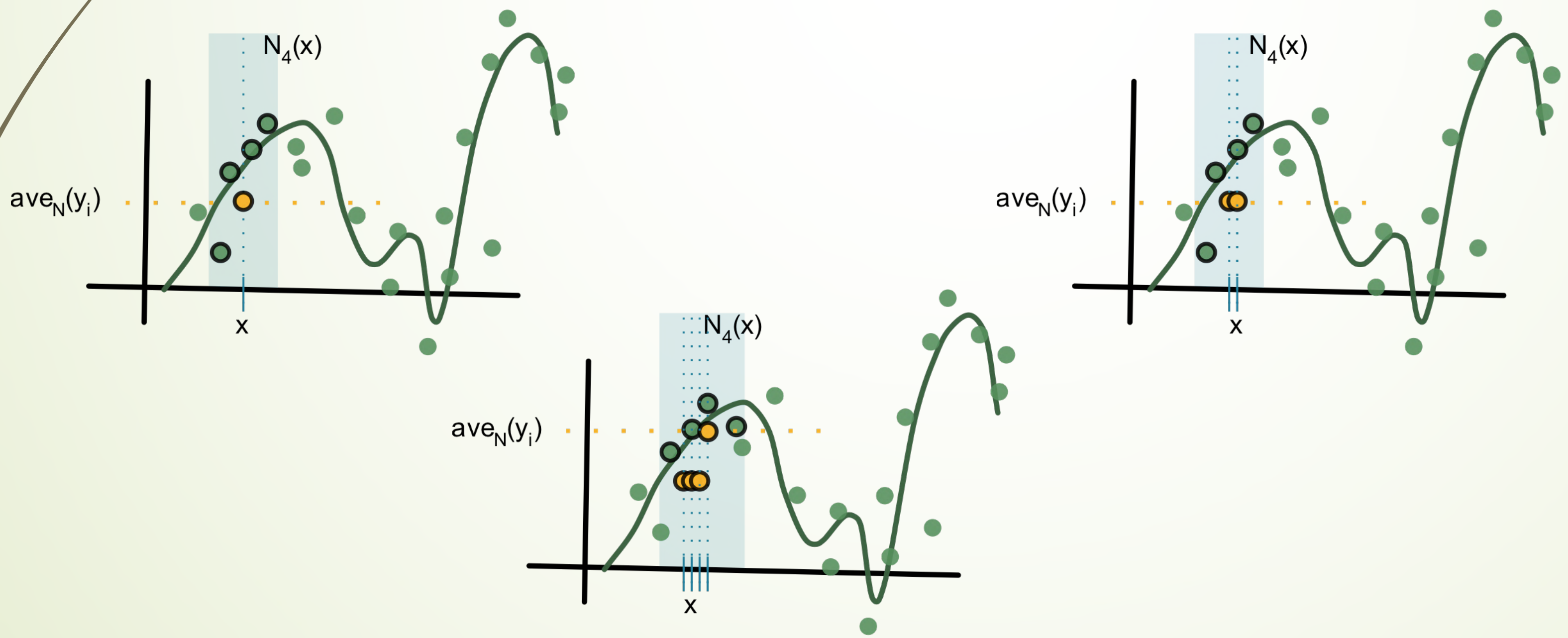
- K fixed and given
- **Samples:** $(x_i, \mathrm{y_i})_{\{i=1\}}^{N}$
- **Estimate data generating function:** $\hat{f}(x) = \frac{1}{K}\sum_{i \in N_K(x)} y_i$
- N_K: K **nearest neighbors of** x within the **training set**





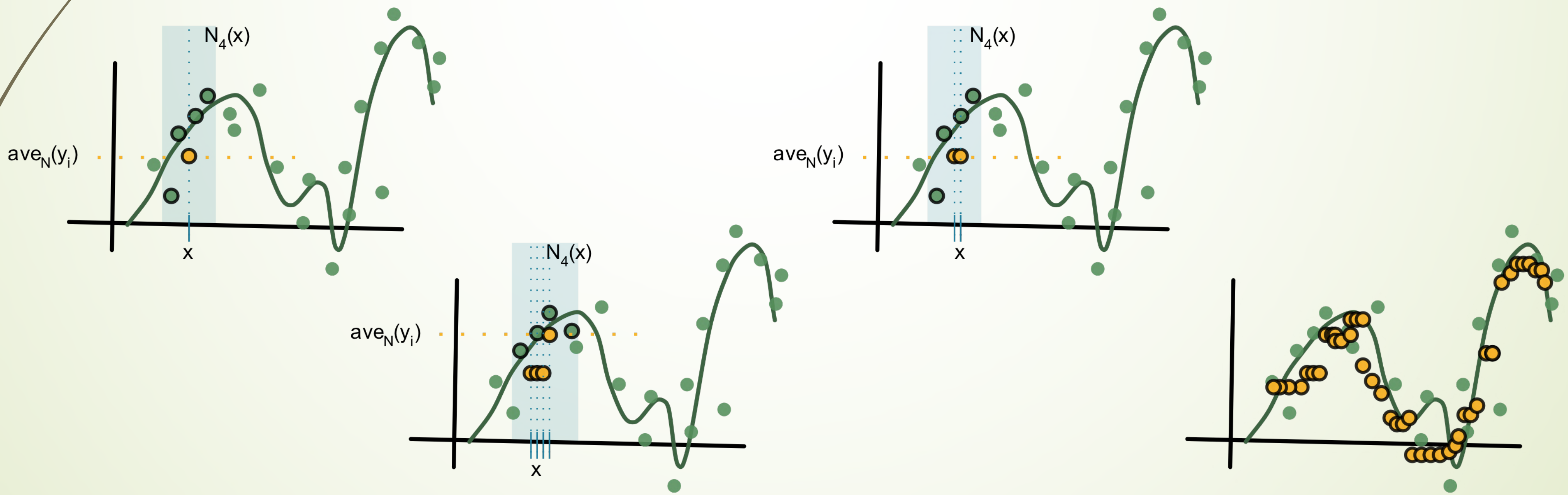
Algorithms: K-Nearest Neighbors (KNN) Regression

- K fixed and given
- **Samples:** $(x_i, y_i)_{\{i=1\}}^{N}$
- **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**



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

Impact of $\boldsymbol{K}$: 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