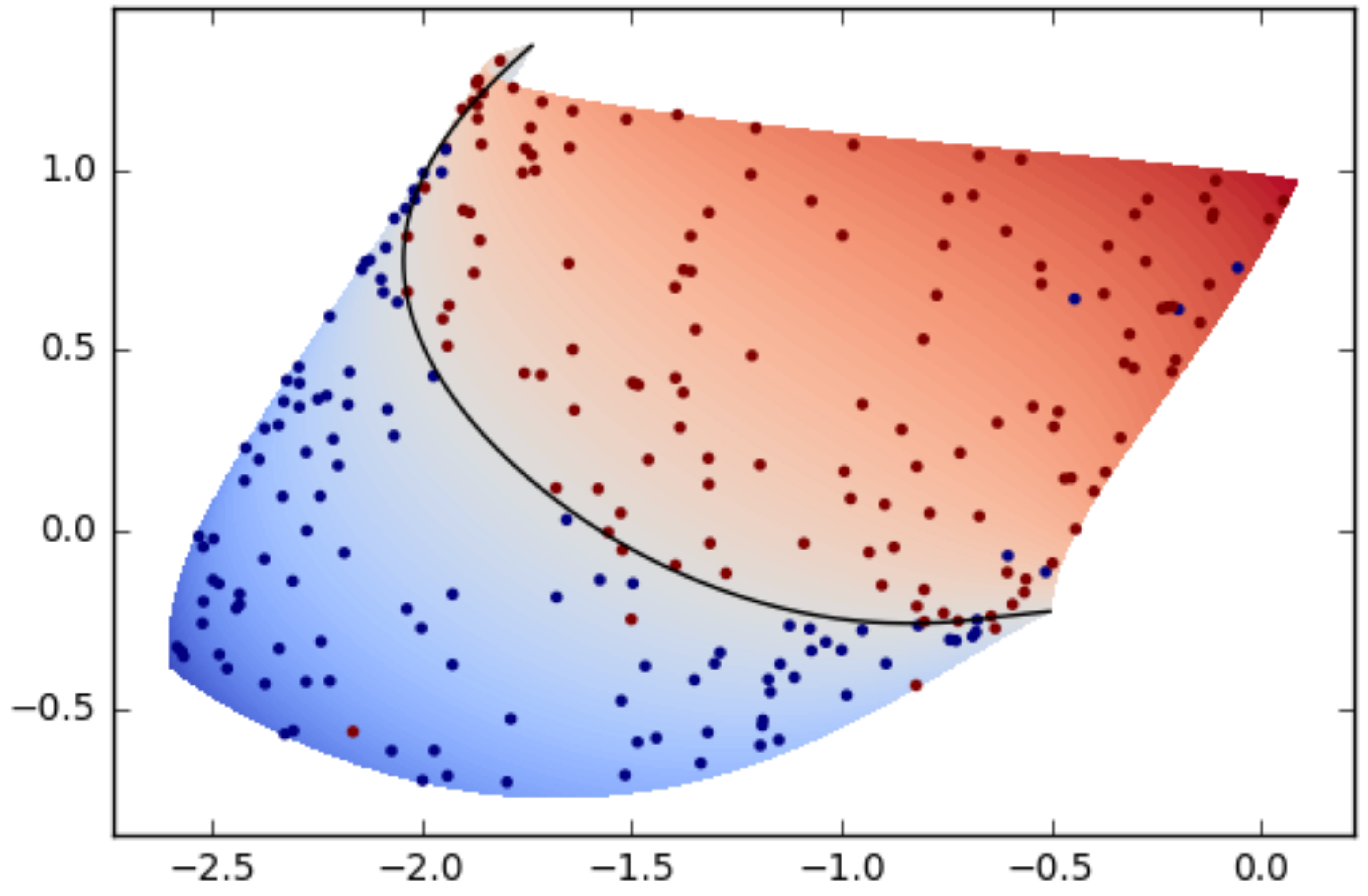
CS 273A: Machine Learning

Winter 2021

Lecture 2: Nearest Neighbors

Roy Fox
Department of Computer Science
Bren School of Information and Computer Sciences
University of California, Irvine

All slides in this course adapted from Alex Ihler & Sameer Singh



Logistics

assignment 1

- Assignment 1 is up on Canvas and gradescope
- Due: Thu, Jan 14 (PT)

lectures

- Lectures will be recorded and added to this playlist.

Today's lecture

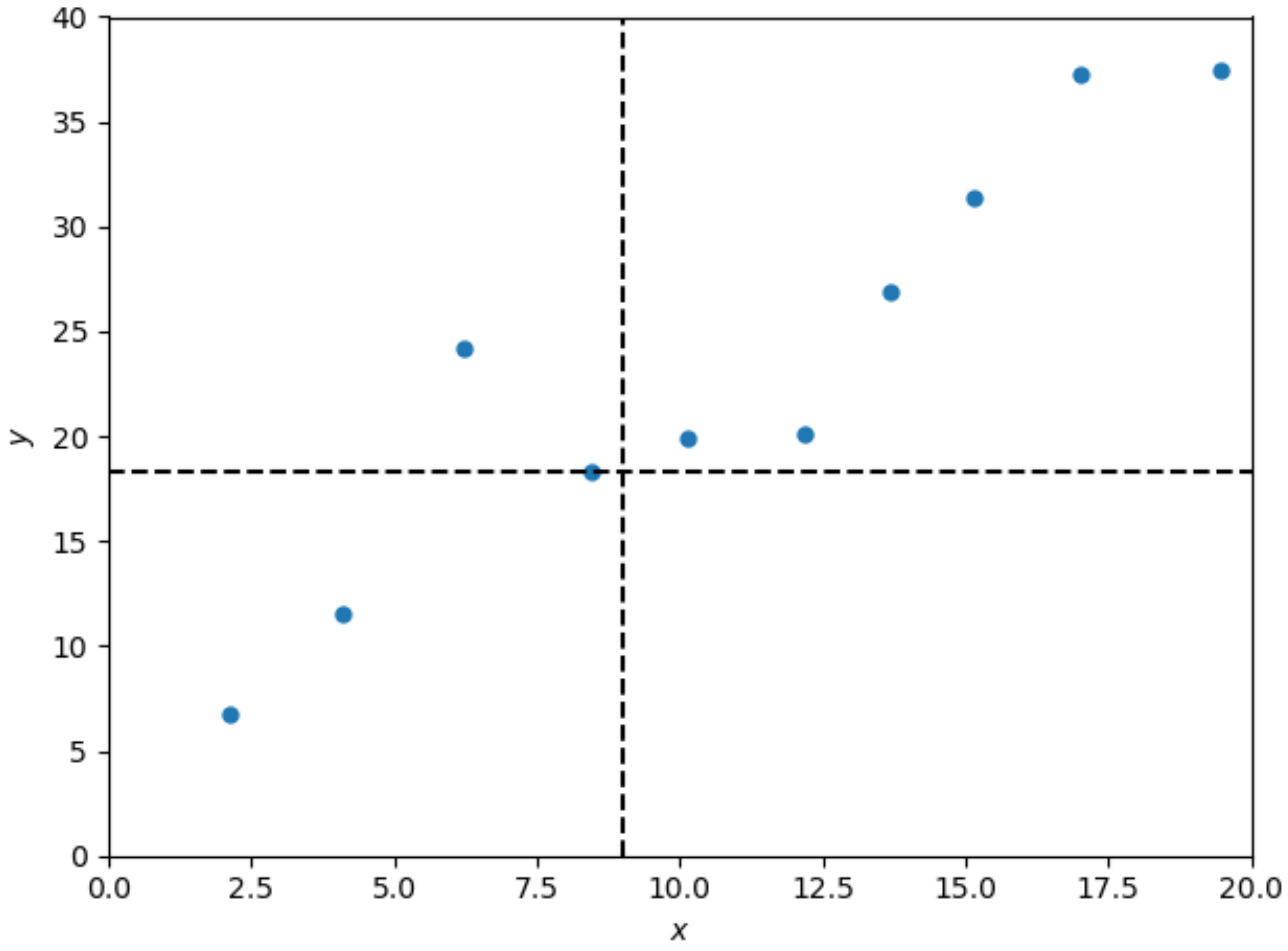
Nearest Neighbors

Overfitting and complexity

k-Nearest Neighbors

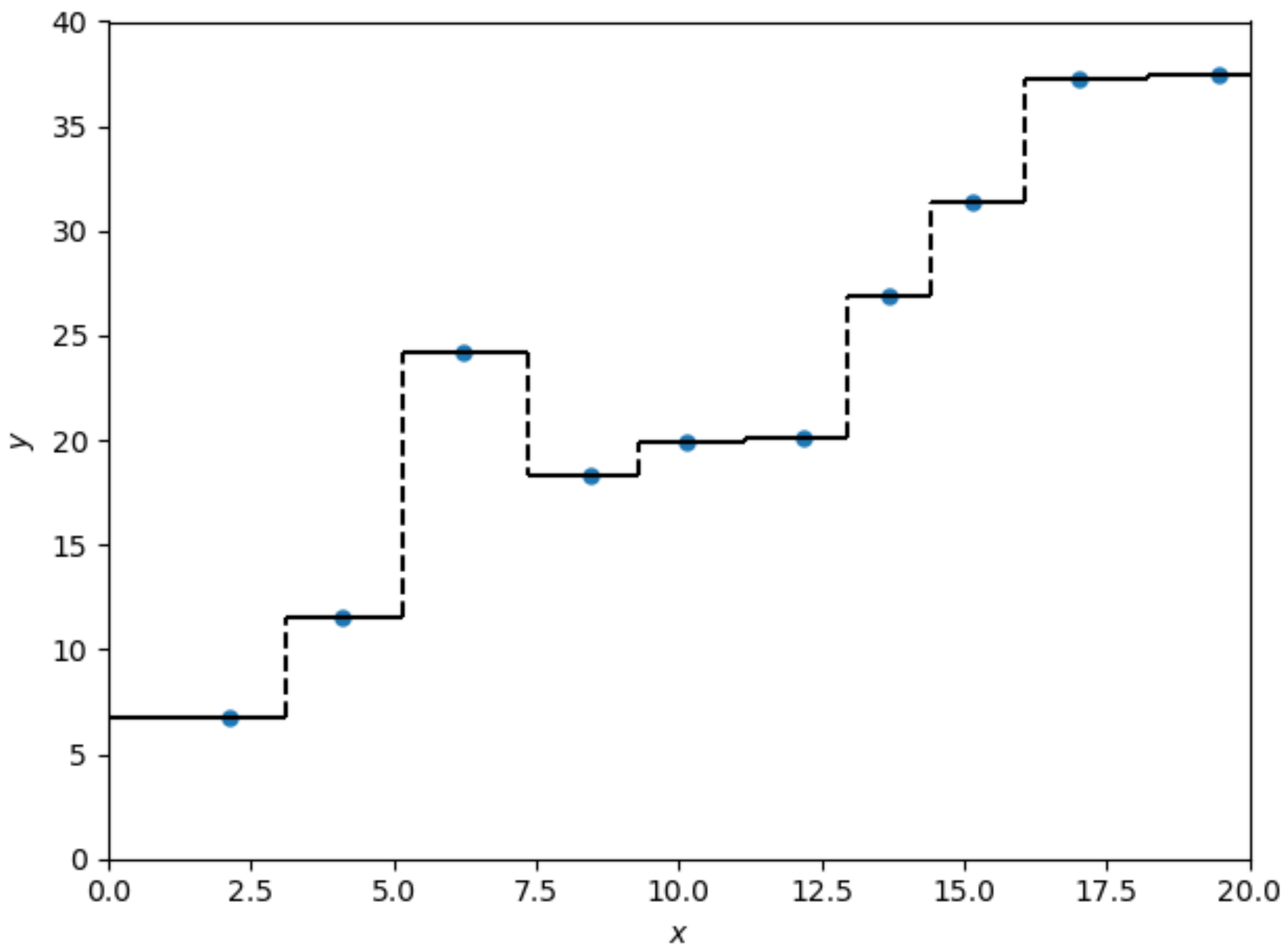
Bayes classifiers

Nearest-Neighbor regression



- $f(x) = y^{(i)}$, such that $x^{(i)} \in \mathscr{D}$ is the closest data point to x

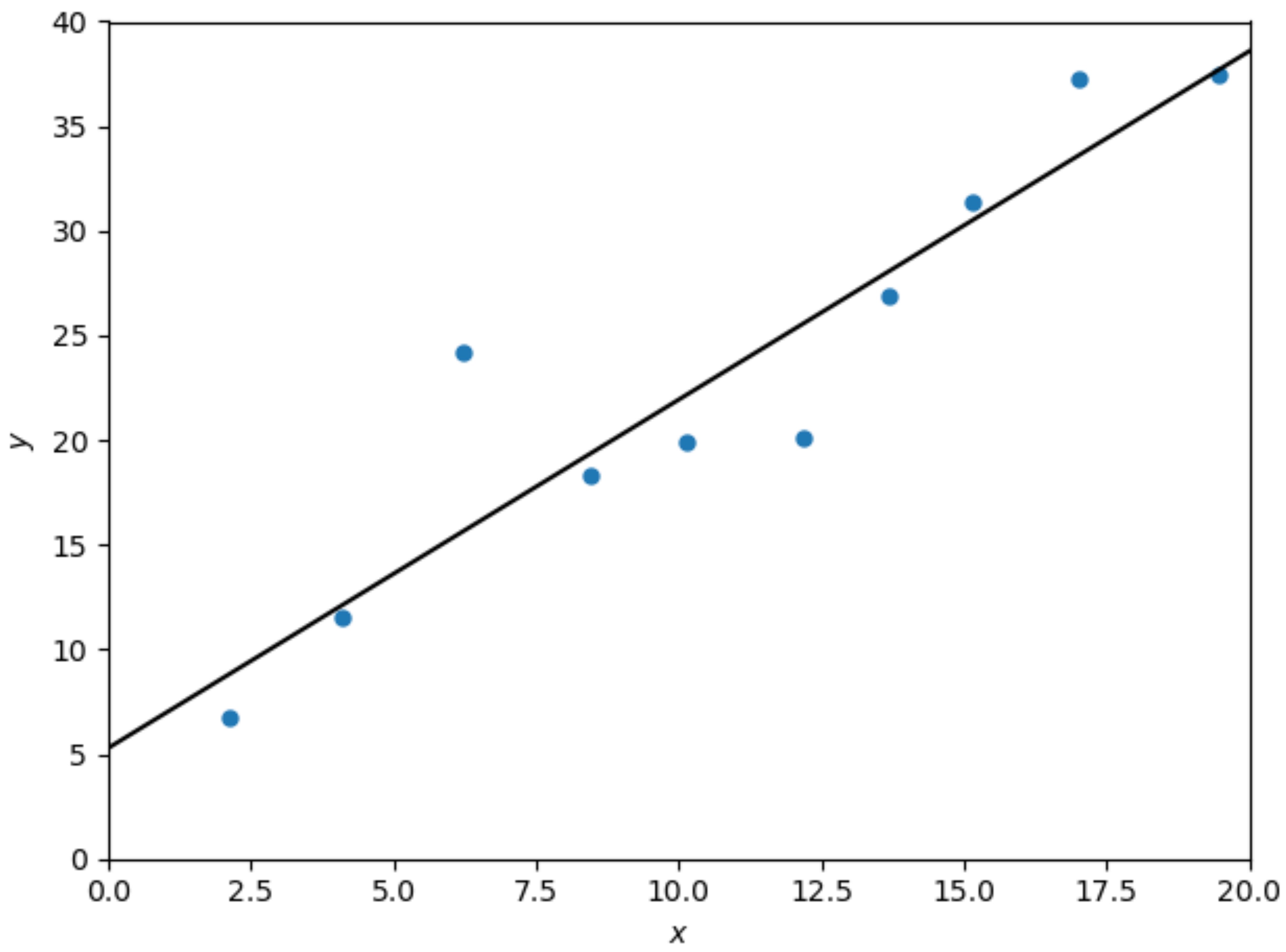
Nearest-Neighbor regression



- Decision function $f : x \mapsto y$ is piecewise constant (for 1D x)

- Data induces f implicitly; f is never stored explicitly, but can be computed

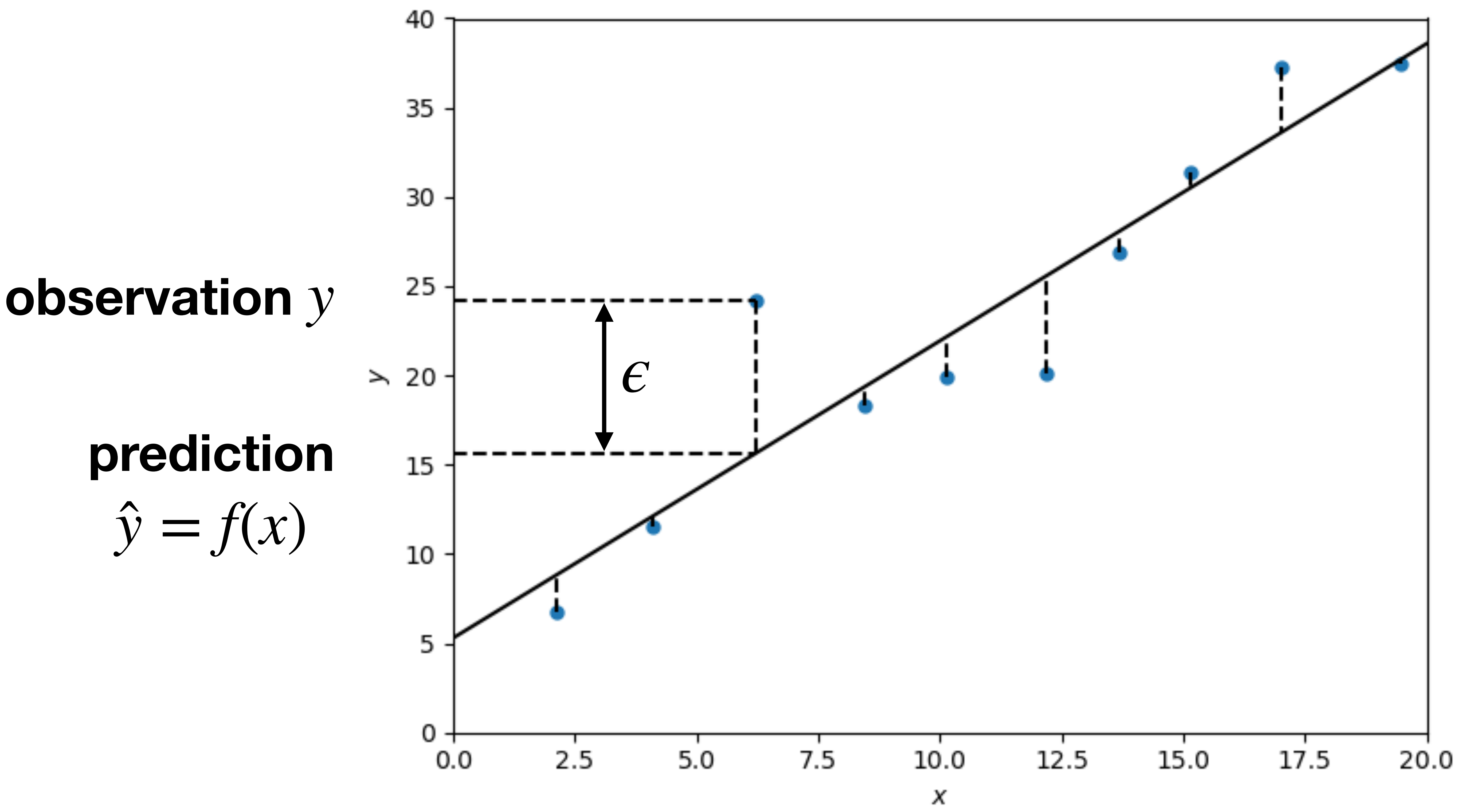
Alternative: linear regression



- Decision function $f : x \mapsto y$ is linear, $f(x) = \theta_0 + \theta_1 x$

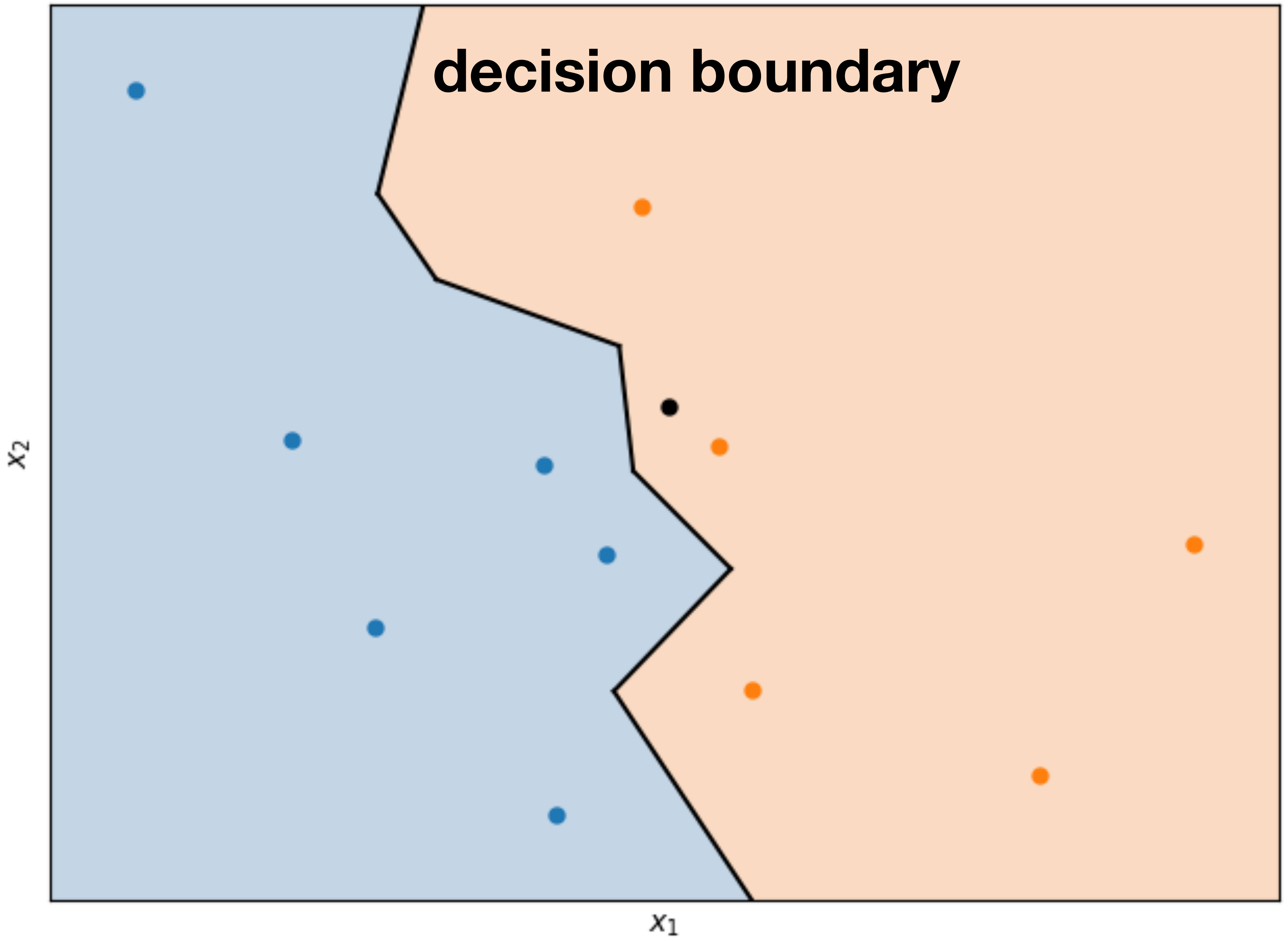
- f is stored by its parameters $\theta = \begin{bmatrix} \theta_0 & \theta_1 \end{bmatrix}$

Measuring error



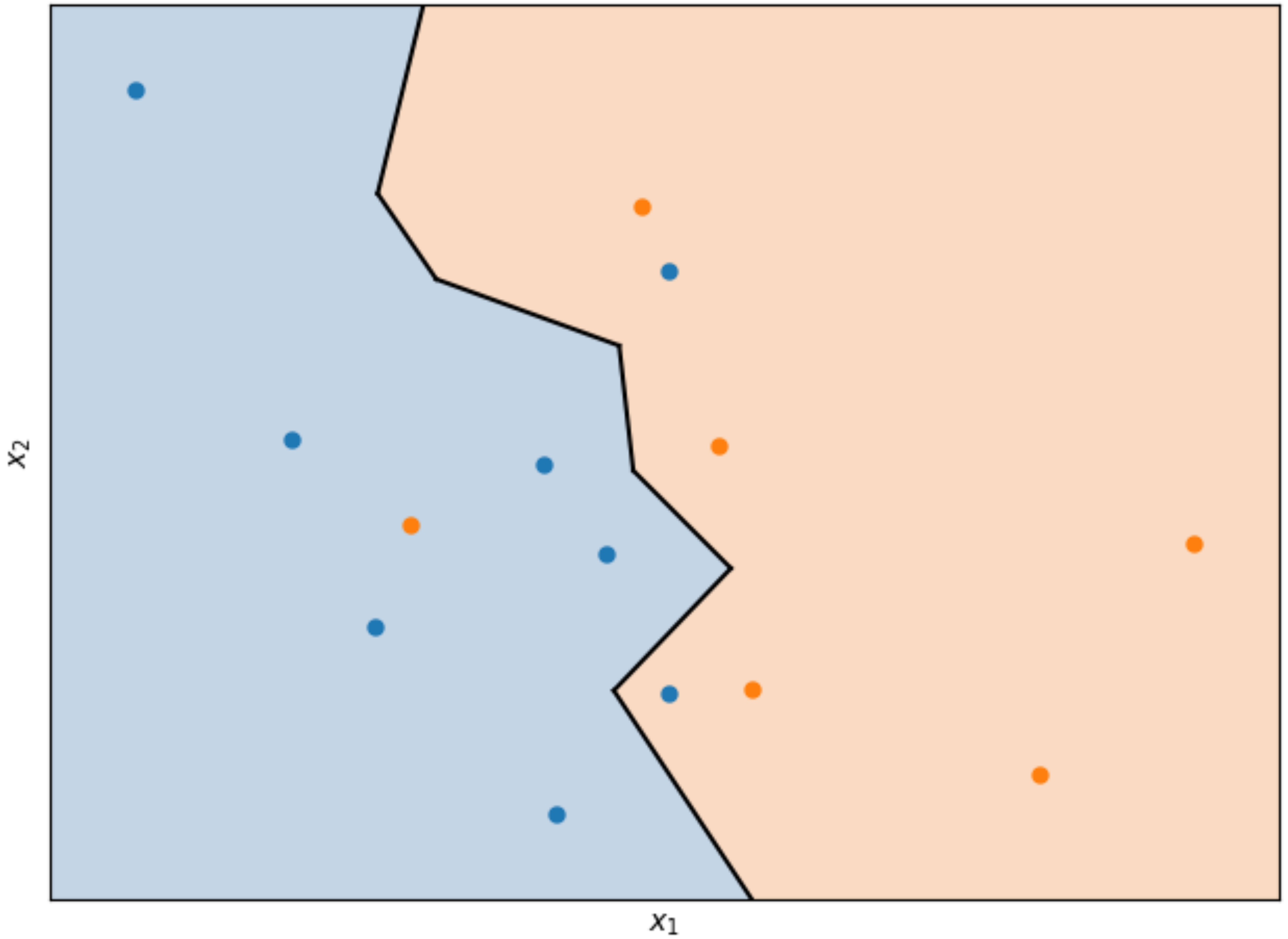
- Error / residual: $\epsilon = y - \hat{y}$
- Mean square error (MSE): $\dfrac{1}{m}\sum_i (\epsilon^{(i)})^2 = \dfrac{1}{m}\sum_i (y^{(i)} - \hat{y}^{(i)})^2$

Classification



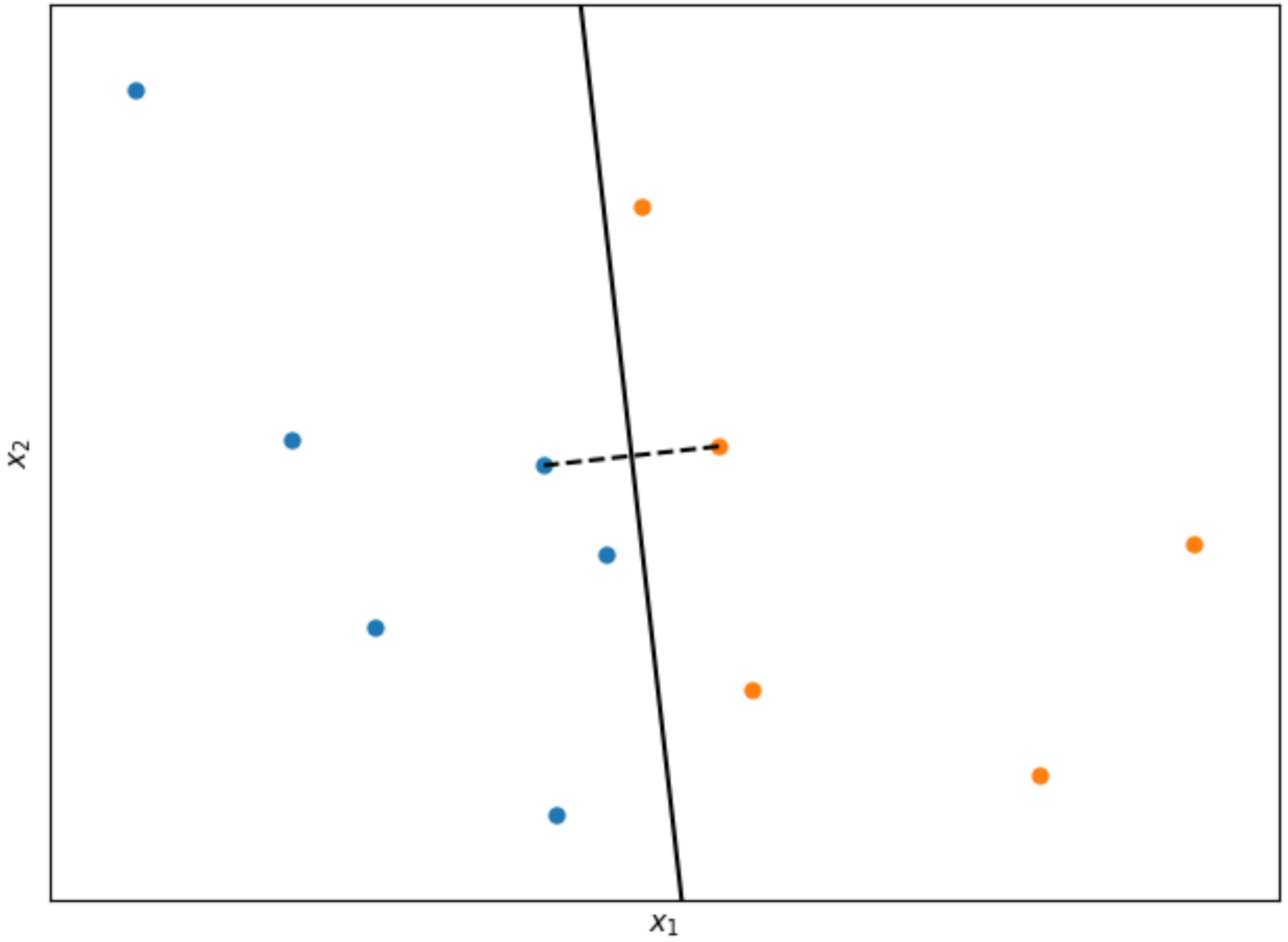
- Using colors as our "third dimension", we can visualize in 2D
- Particularly clear for classification, where y is discrete

Measuring error



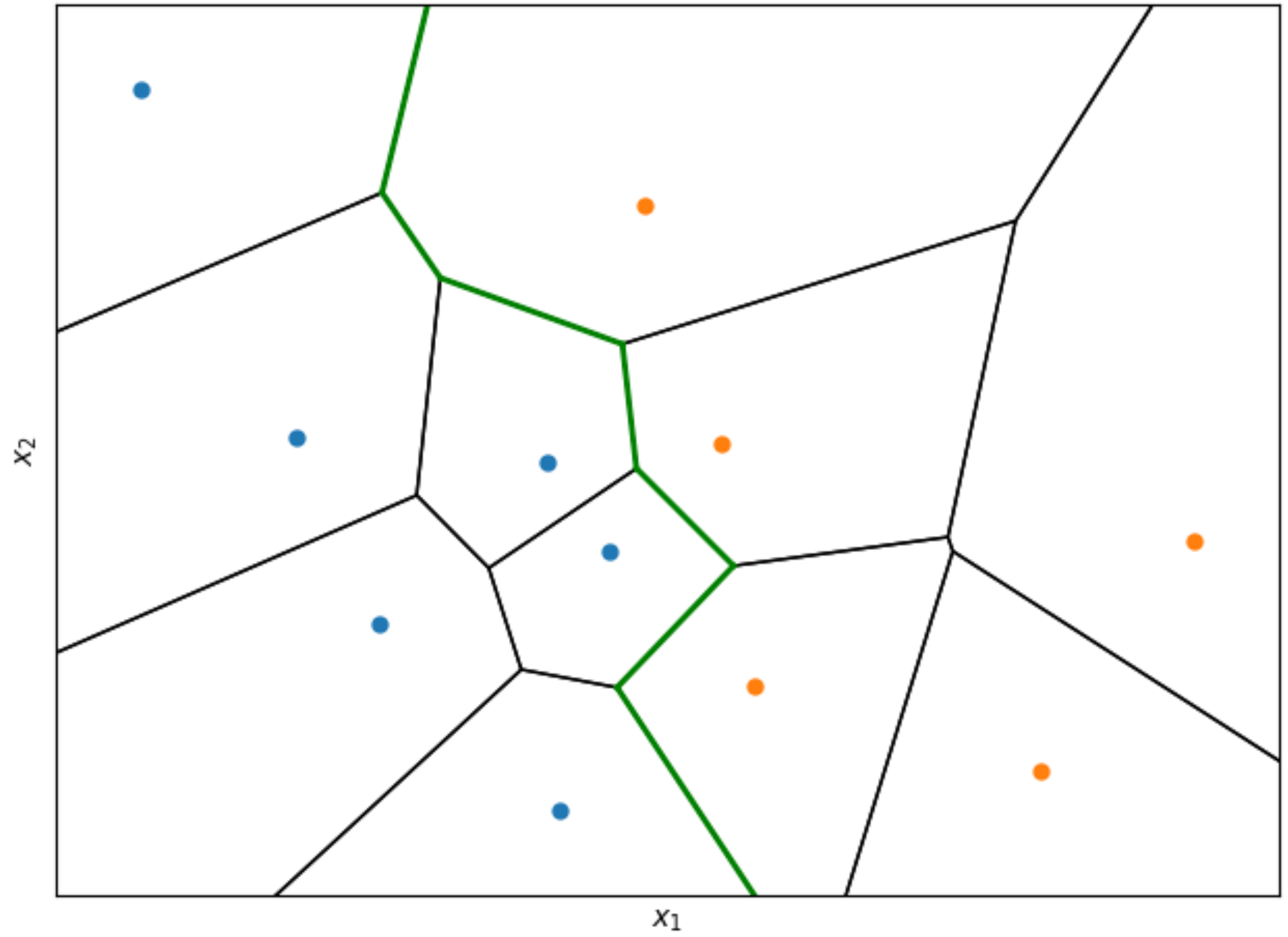
- Error rate: $\frac{1}{m}\sum_i \delta[y^{(i)} \neq \hat{y}^{(i)}]$

Decision boundary is piecewise linear



- For every two data points $x^{(i)}, x^{(j)}$ of different classes $y^{(i)} \neq y^{(j)}$
  - The hyperplane orthogonal to their midpoint is where $d(x, x^{(i)}) = d(x, x^{(j)})$
  - The decision boundary consists of some of these hyperplanes

Voronoi tessellation



- Each data point has a region in which it is the nearest neighbor
  - This region is a polygon
- The decision boundary consists of the edges that cross classes

Today's lecture

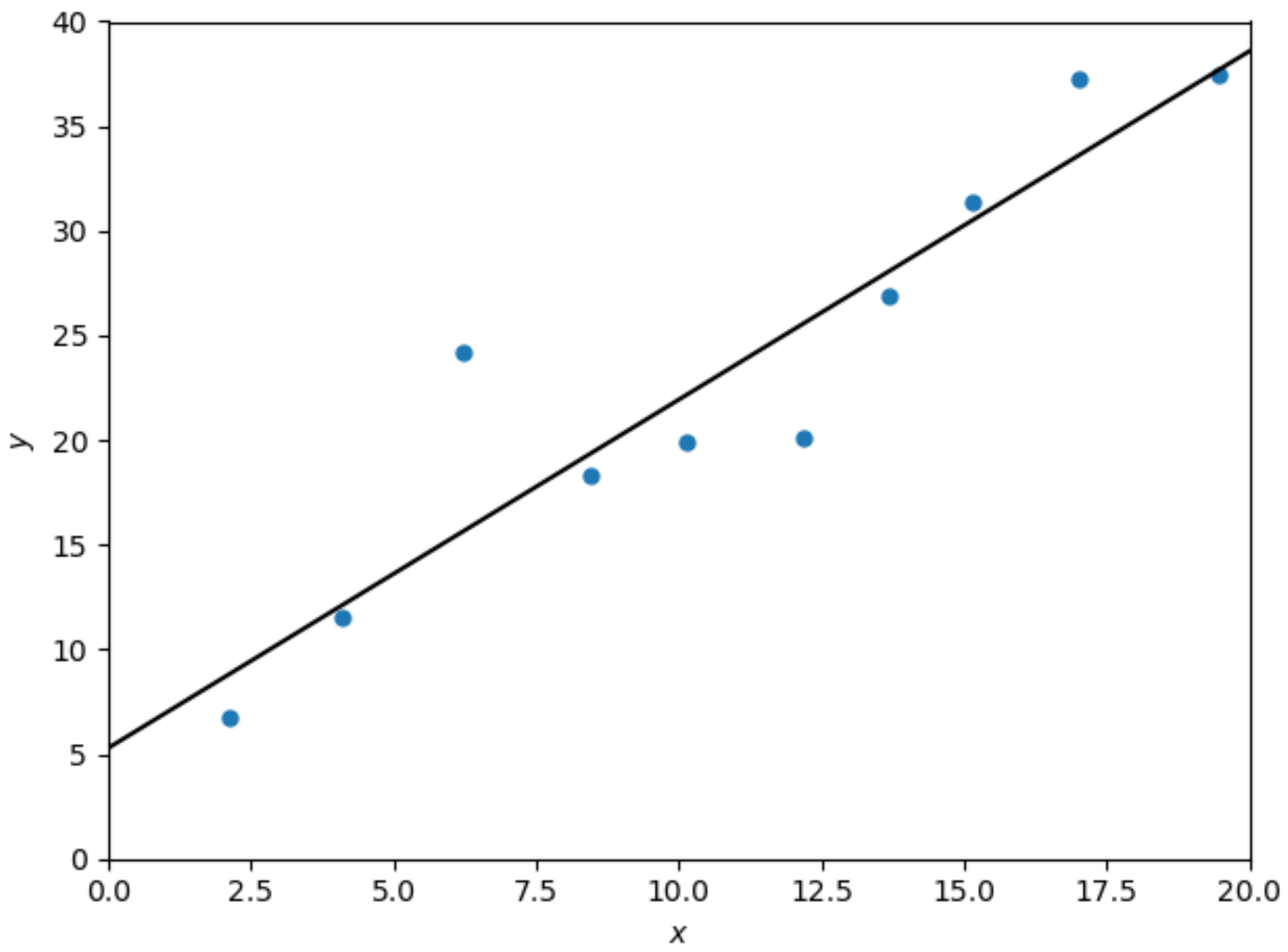
Nearest Neighbors

Overfitting and complexity

k-Nearest Neighbors

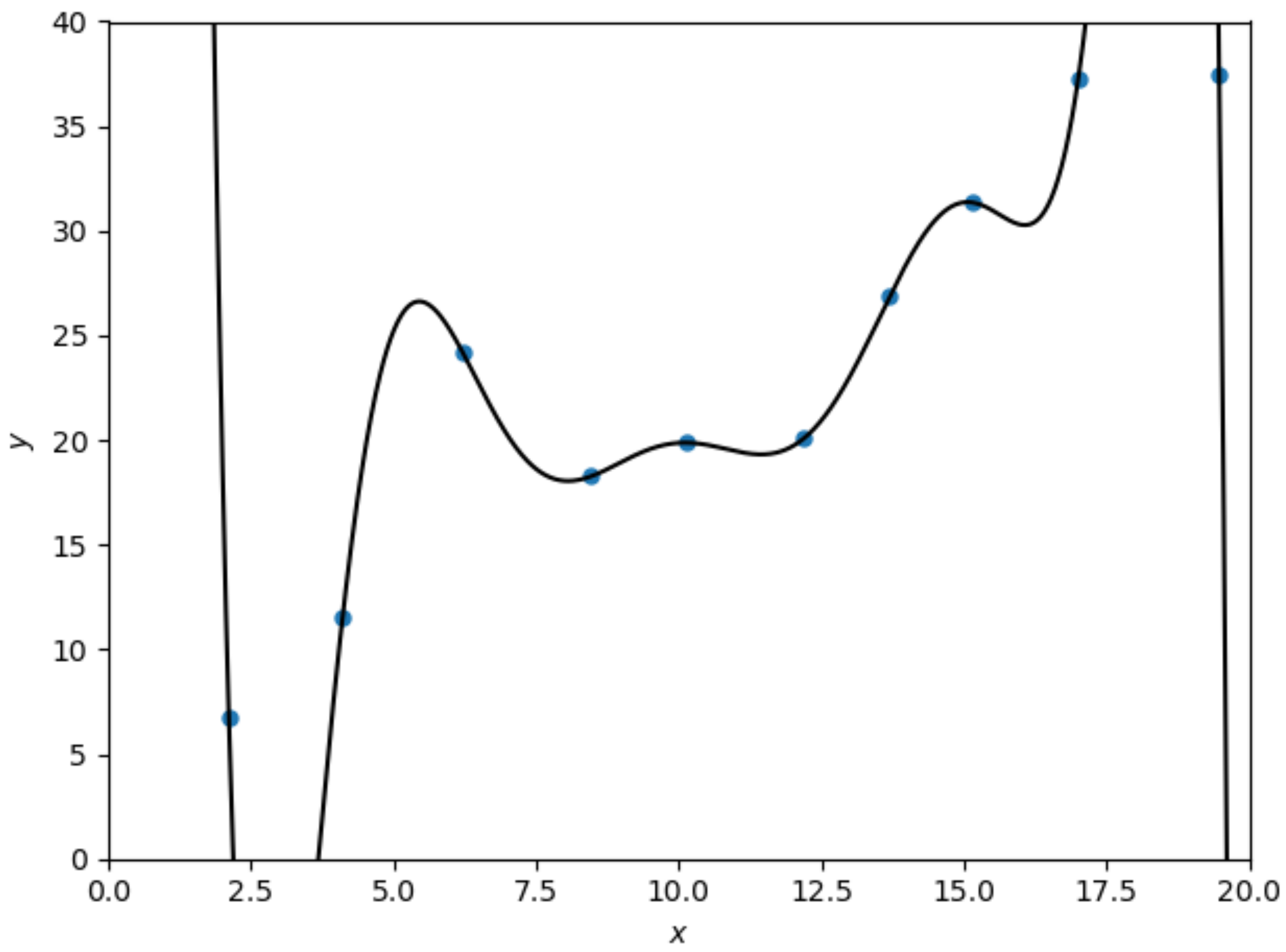
Bayes classifiers

Overfitting and complexity



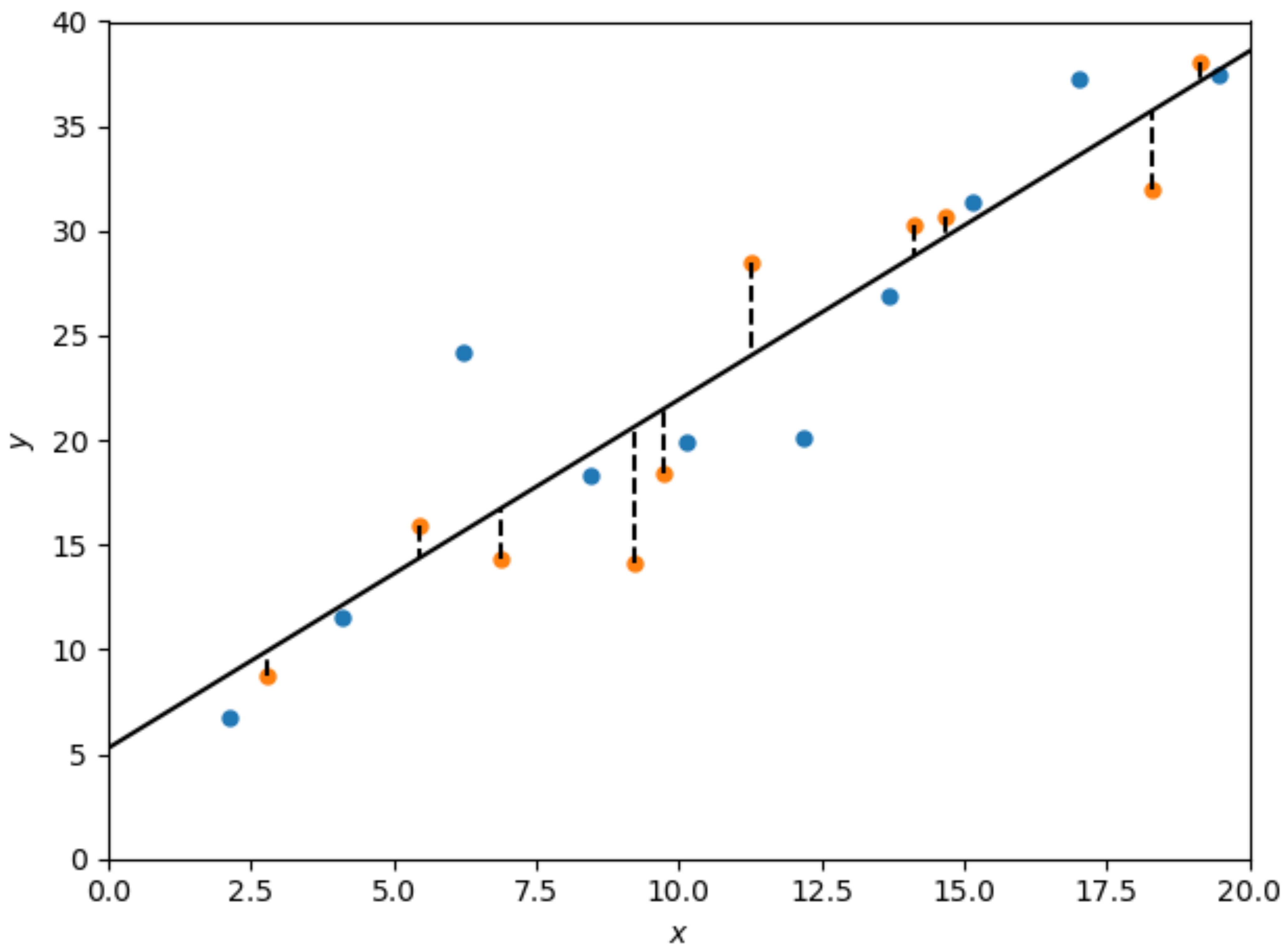
- Simple linear model
- Fits the training data, but with errors
- Interpolation seems reasonable

Overfitting and complexity



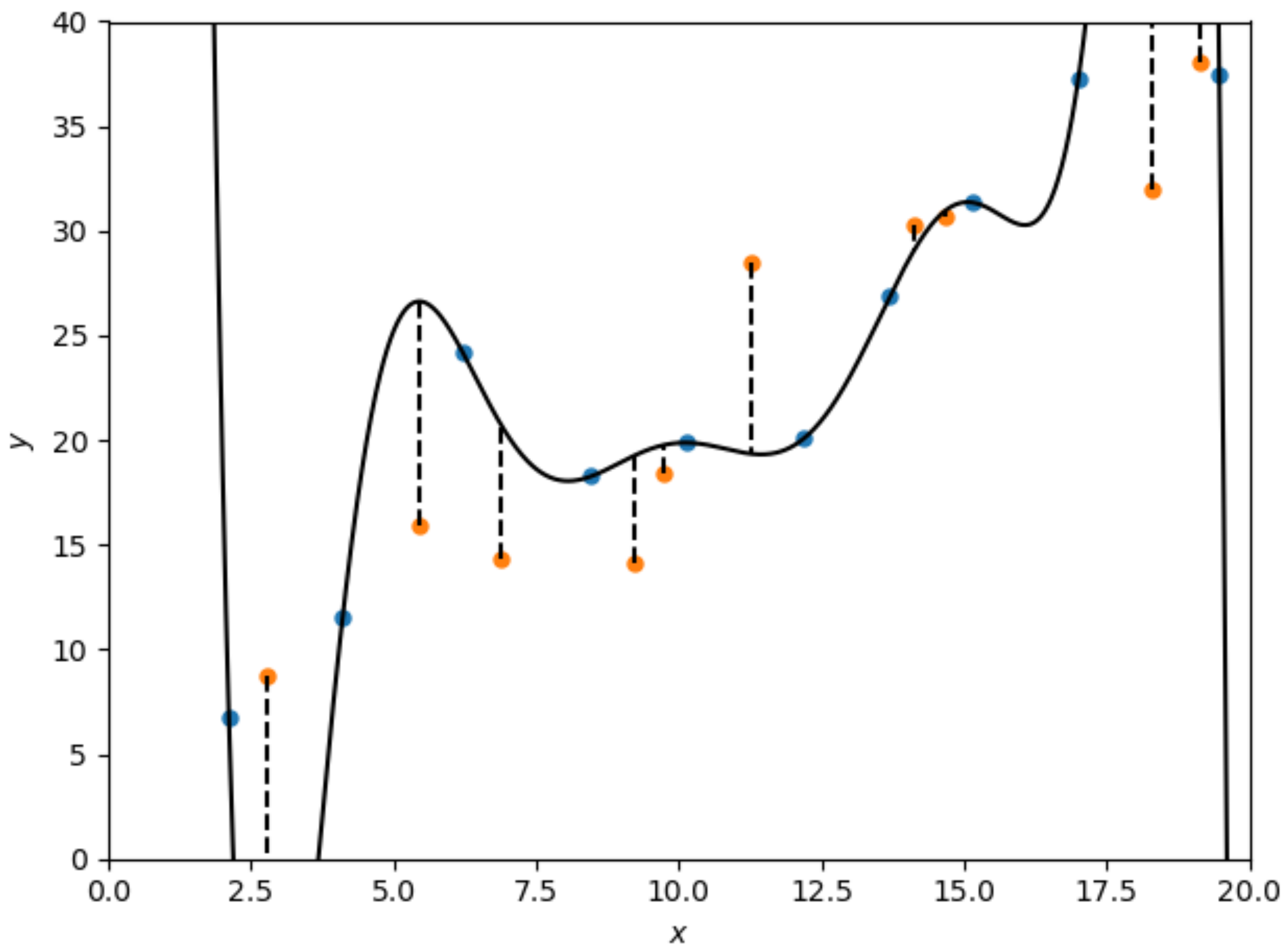
- High-order polynomial model
- Fits the training data perfectly
- Interpolation? more like confabulation, amirite?

Overfitting and complexity



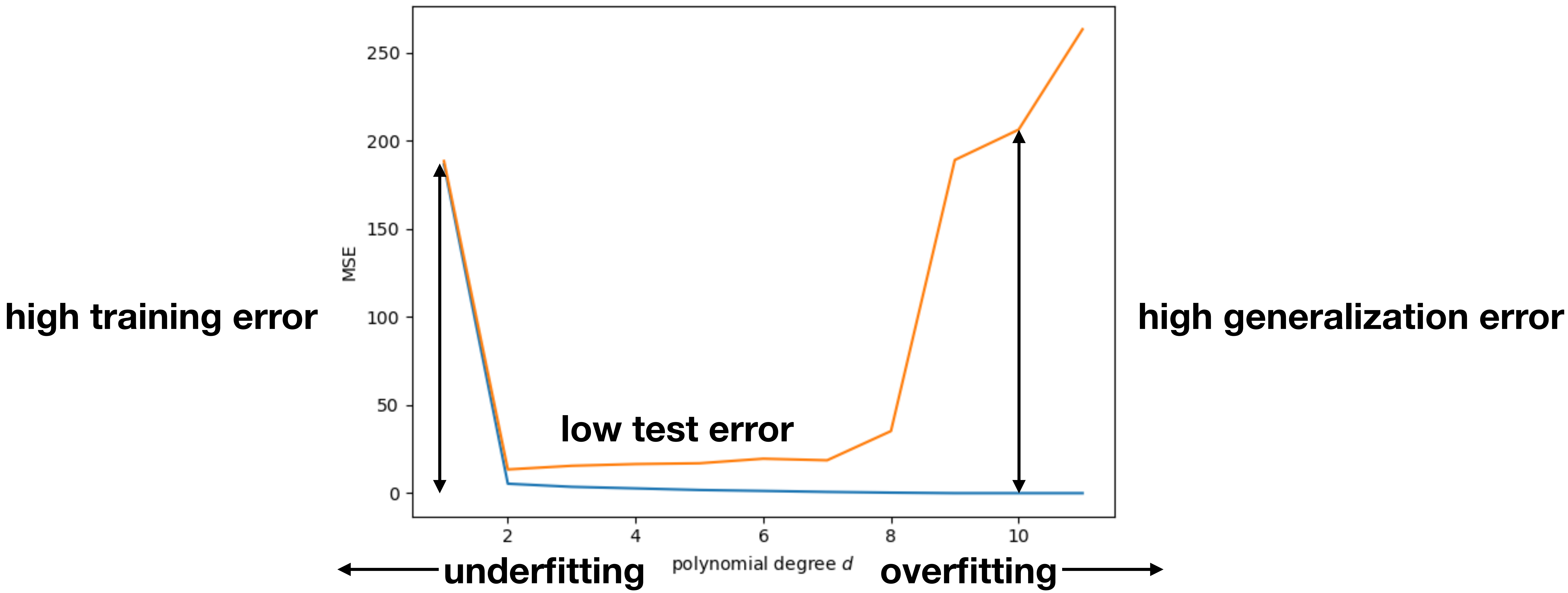
- New test data will also have prediction errors
- Good generalization = test errors will be similar to training errors

Overfitting and complexity



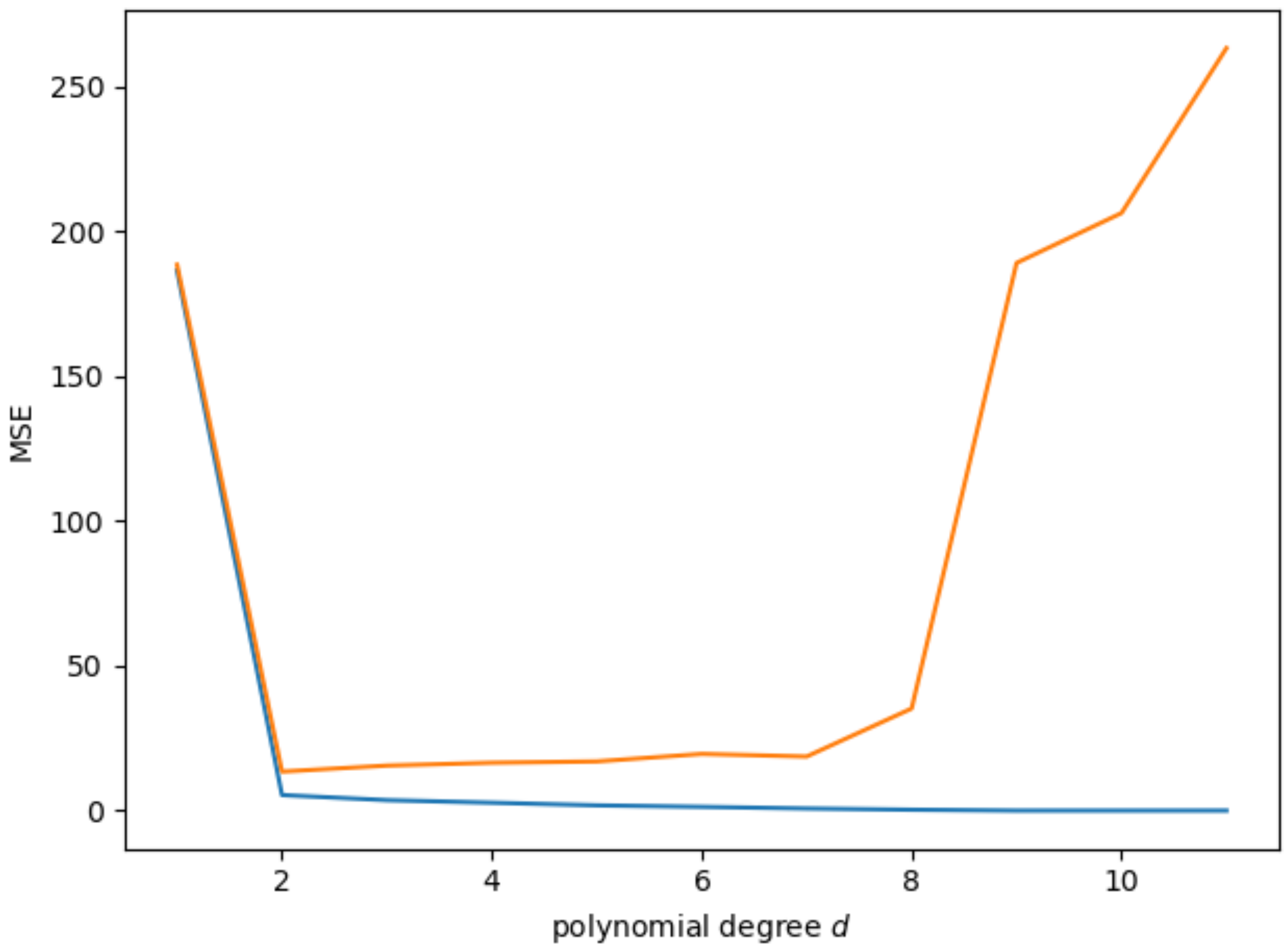
- A complex model may fit the training data well → low training error
- But it may generalize poorly to test data → high test error
- This is called overfitting the training data

How overfitting affects prediction error



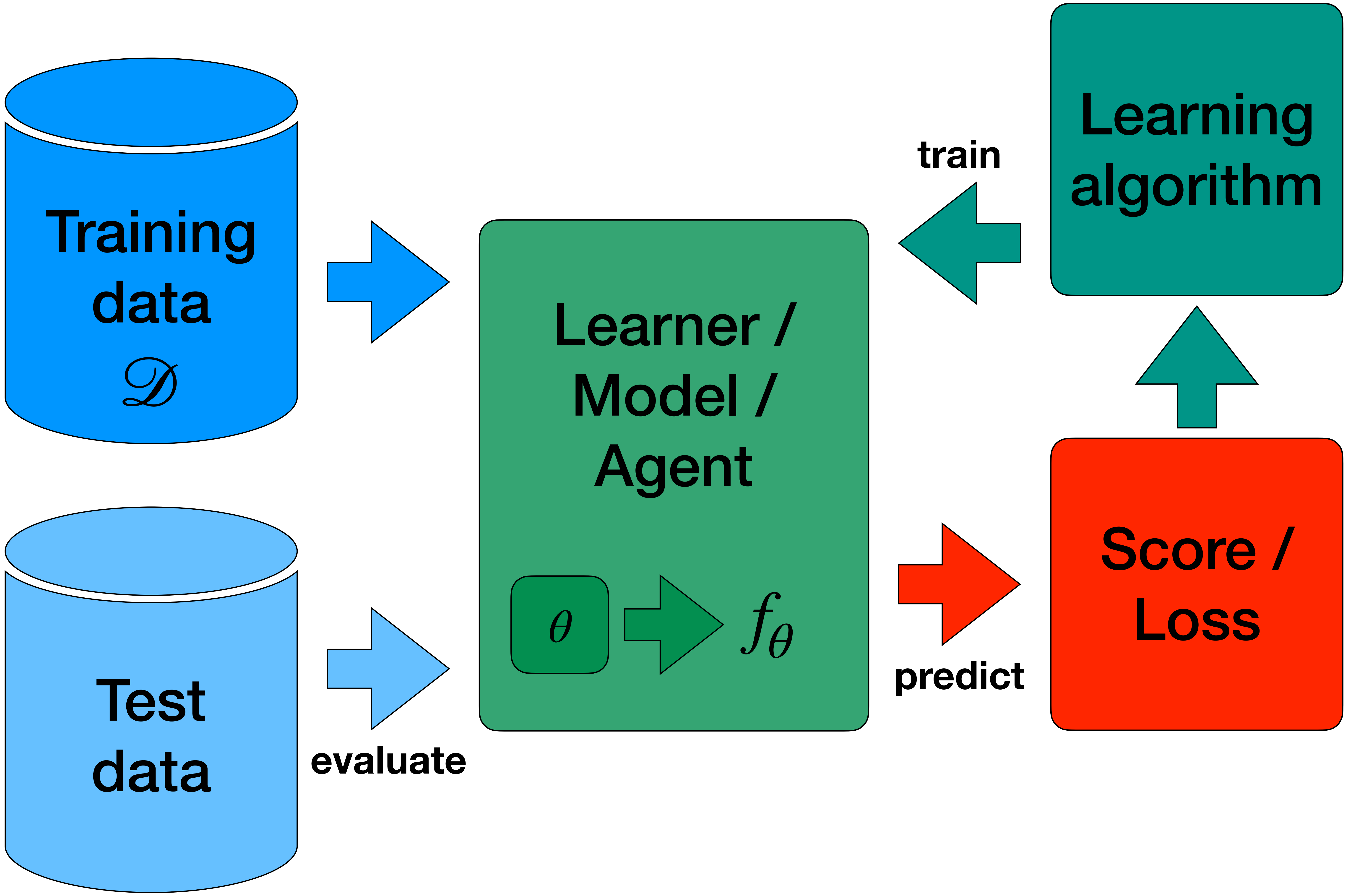
- Low model complexity → underfitting
  - High test error = high training error + low generalization error
- High model complexity → overfitting
  - High test error = low training error + high generalization error

Validation

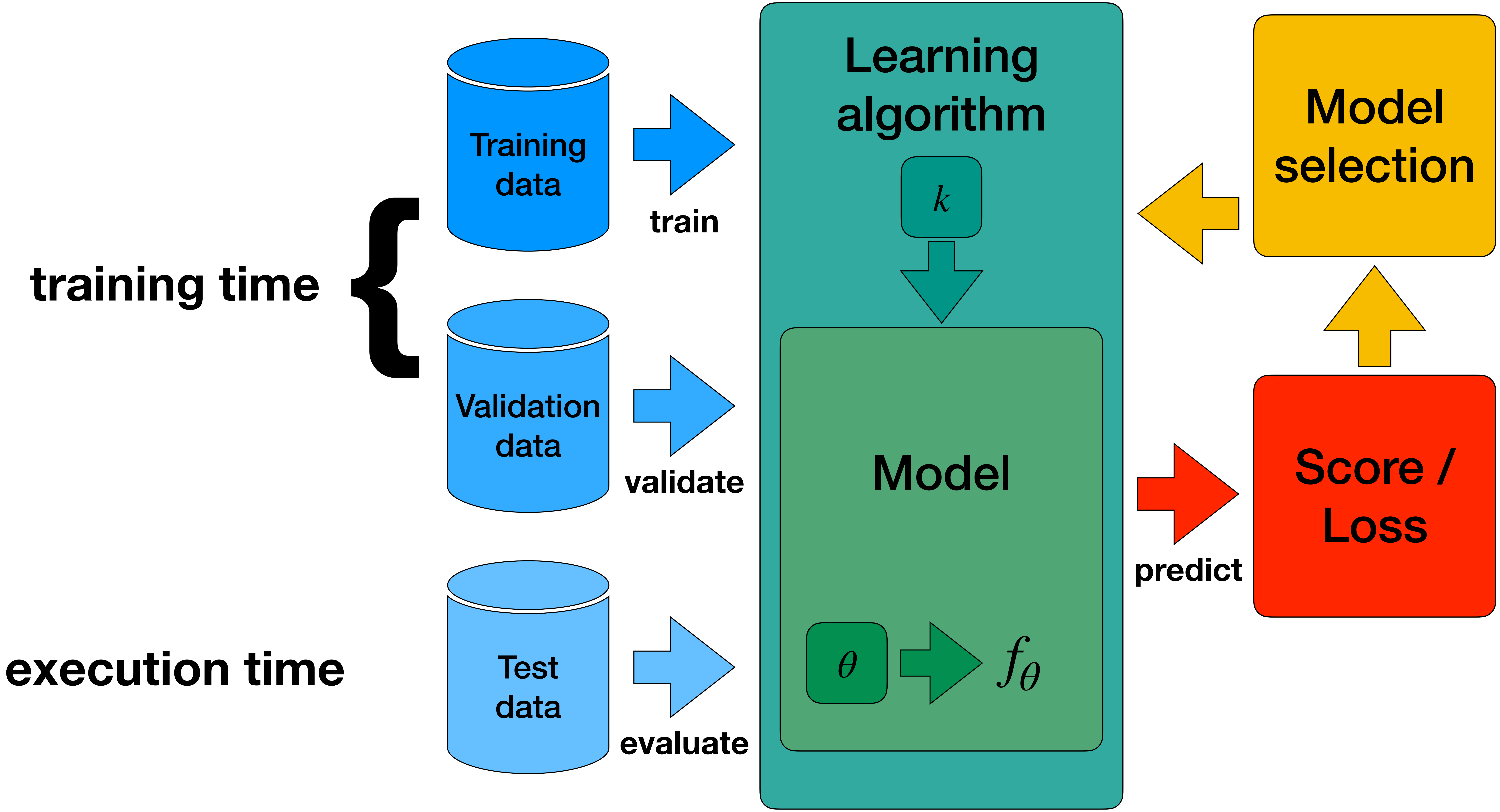


- How can we choose the model complexity? with learning!
    - Model selection = choose our model class
    - Score function: low test error = training error + generalization error

Model learning



Model selection



Recap: overfitting and complexity

- Test error = training error + generalization error
- Model complexity may lead to overfitting
  - Fit the training data very well, but generalize poorly
- Model simplicity may lead to underfitting
  - Do as poorly on the test data as on the training data

Today's lecture

Nearest Neighbors

Overfitting and complexity

k-Nearest Neighbors

Bayes classifiers

k-Nearest Neighbor (kNN)

- Find the k nearest neighbors to x in the dataset
  - Given x, rank the data points by their distance from x, $d(x, x^{(j)})$
    - Usually, Euclidean distance $d(x, x^{(j)}) = \sqrt{\frac{1}{n}\sum_i (x_i - x_i^{(j)})^2}$
  - Select the k data points which are have smallest distance to x
- What is the prediction?
  - Regression: average $y^{(j)}$ for the k closest training examples
  - Classification: take a majority vote among $y^{(j)}$ for the k closest training examples
    - No ties in 2-class problems when k is odd

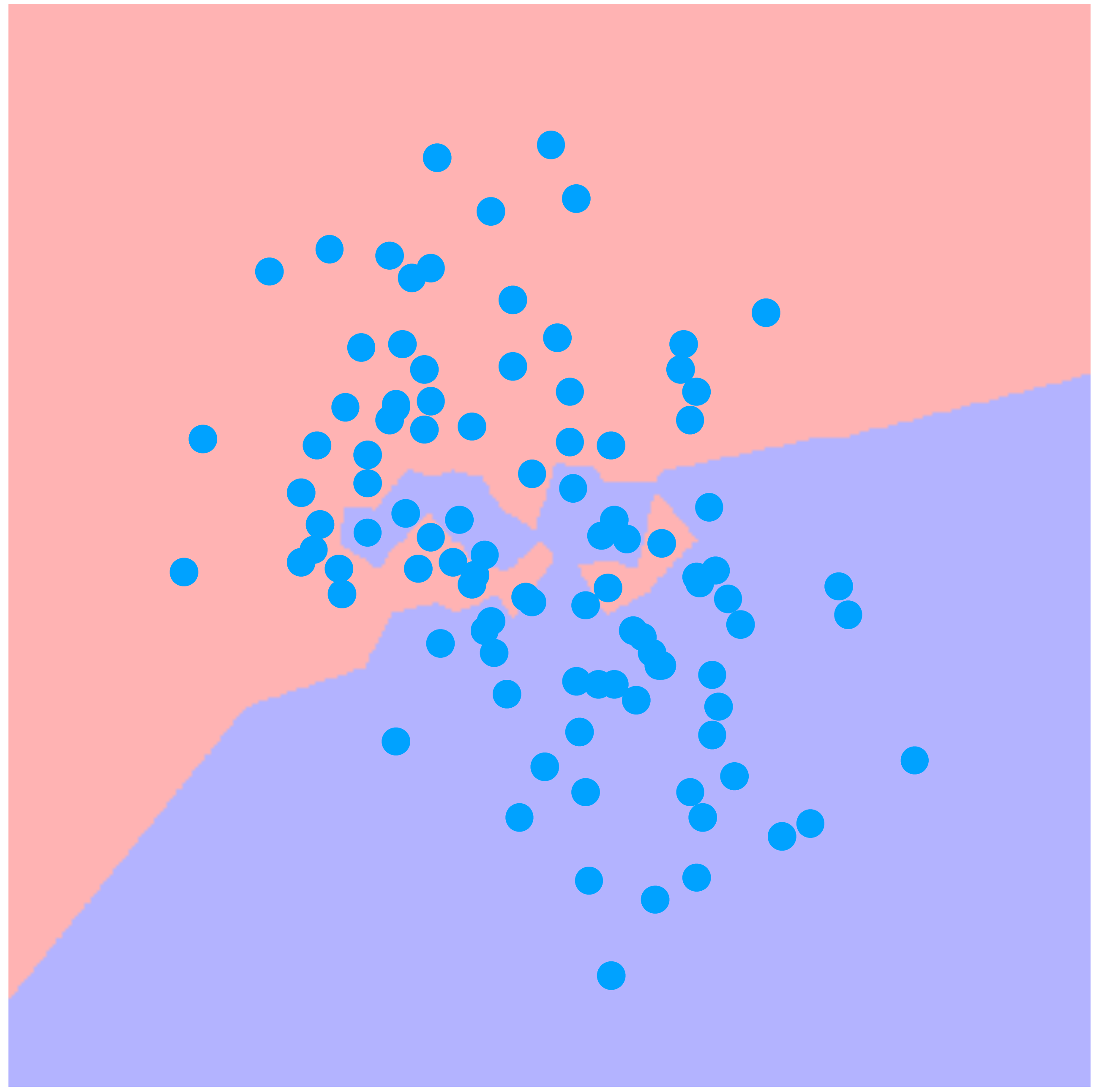
kNN decision boundary

- For classification, the decision boundary is piecewise linear

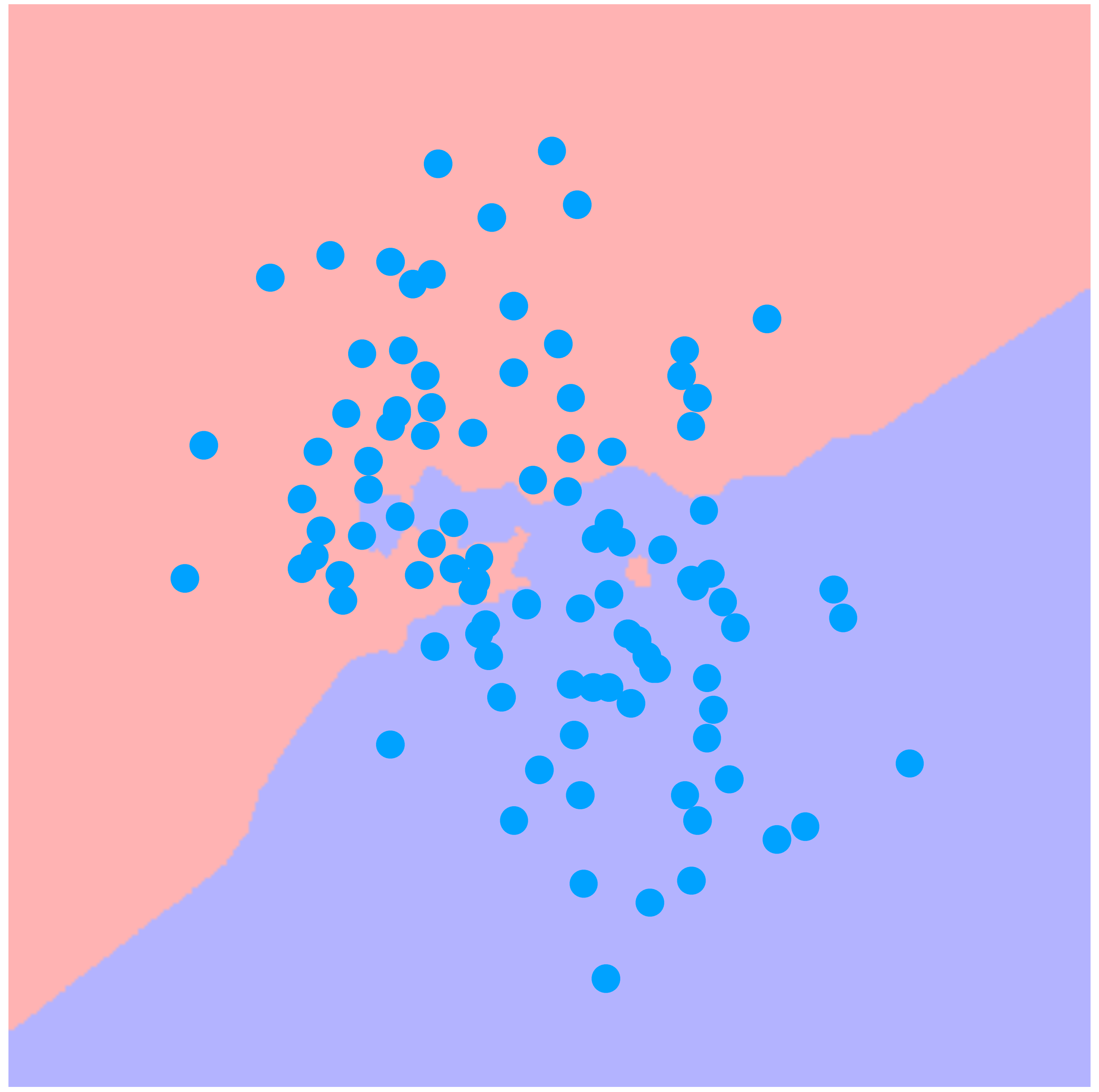
- Increasing k "simplifies" the decision boundary

  - Majority voting means less emphasis on individual points

$k = 1$



$k = 3$



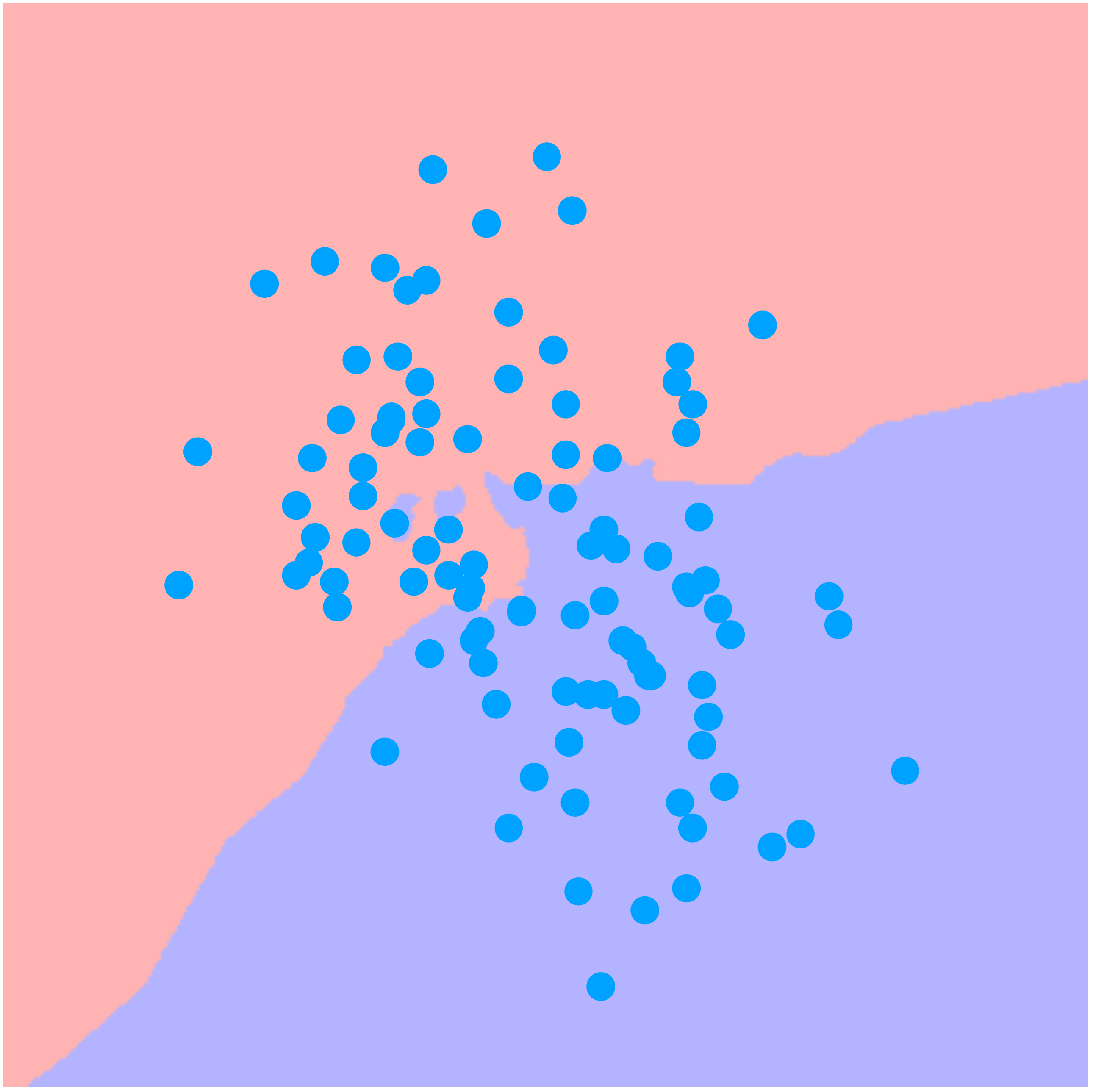
kNN decision boundary

- For classification, the decision boundary is piecewise linear

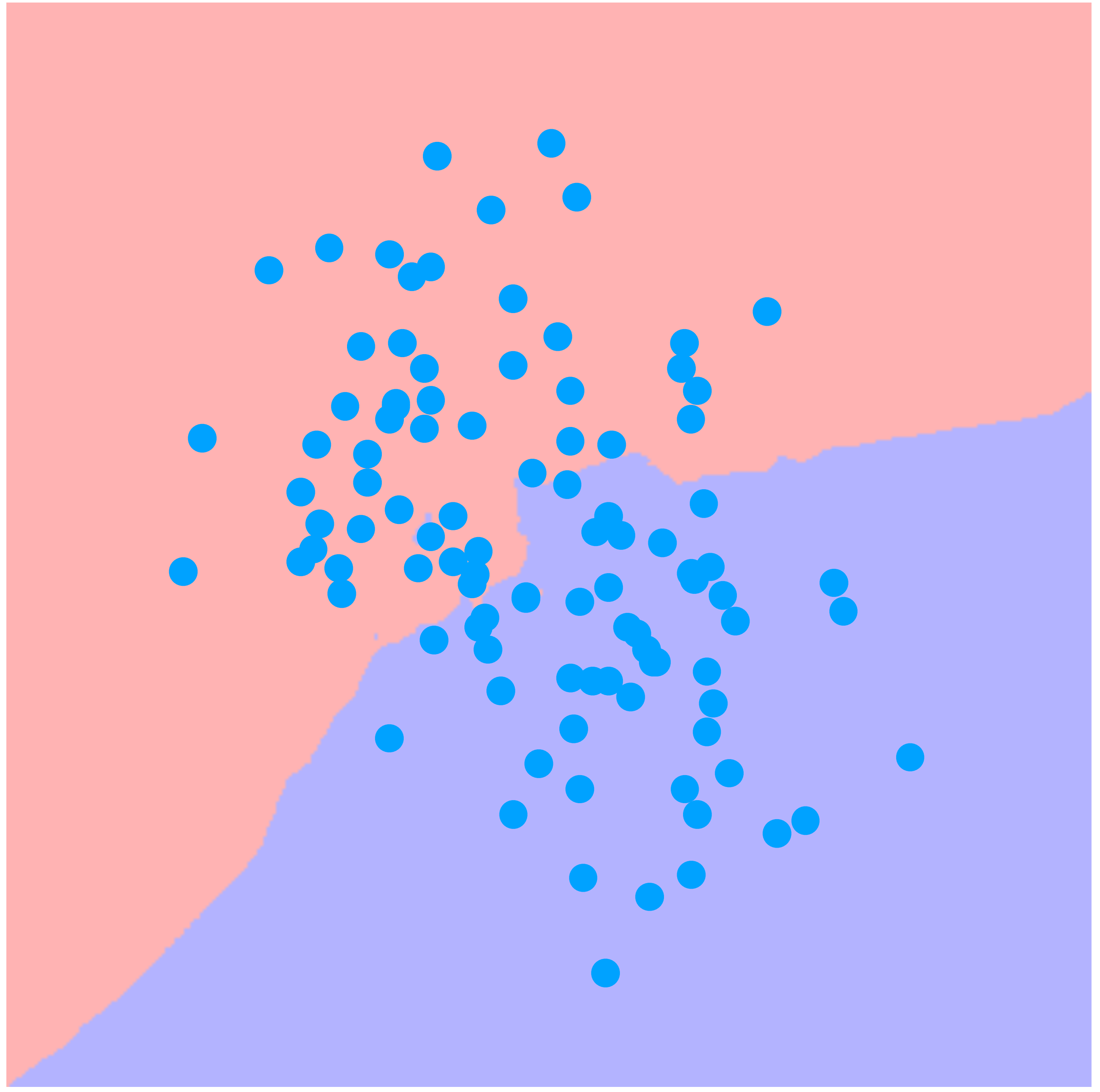
- Increasing k “simplifies” the decision boundary

  - Majority voting means less emphasis on individual points

$k = 5$



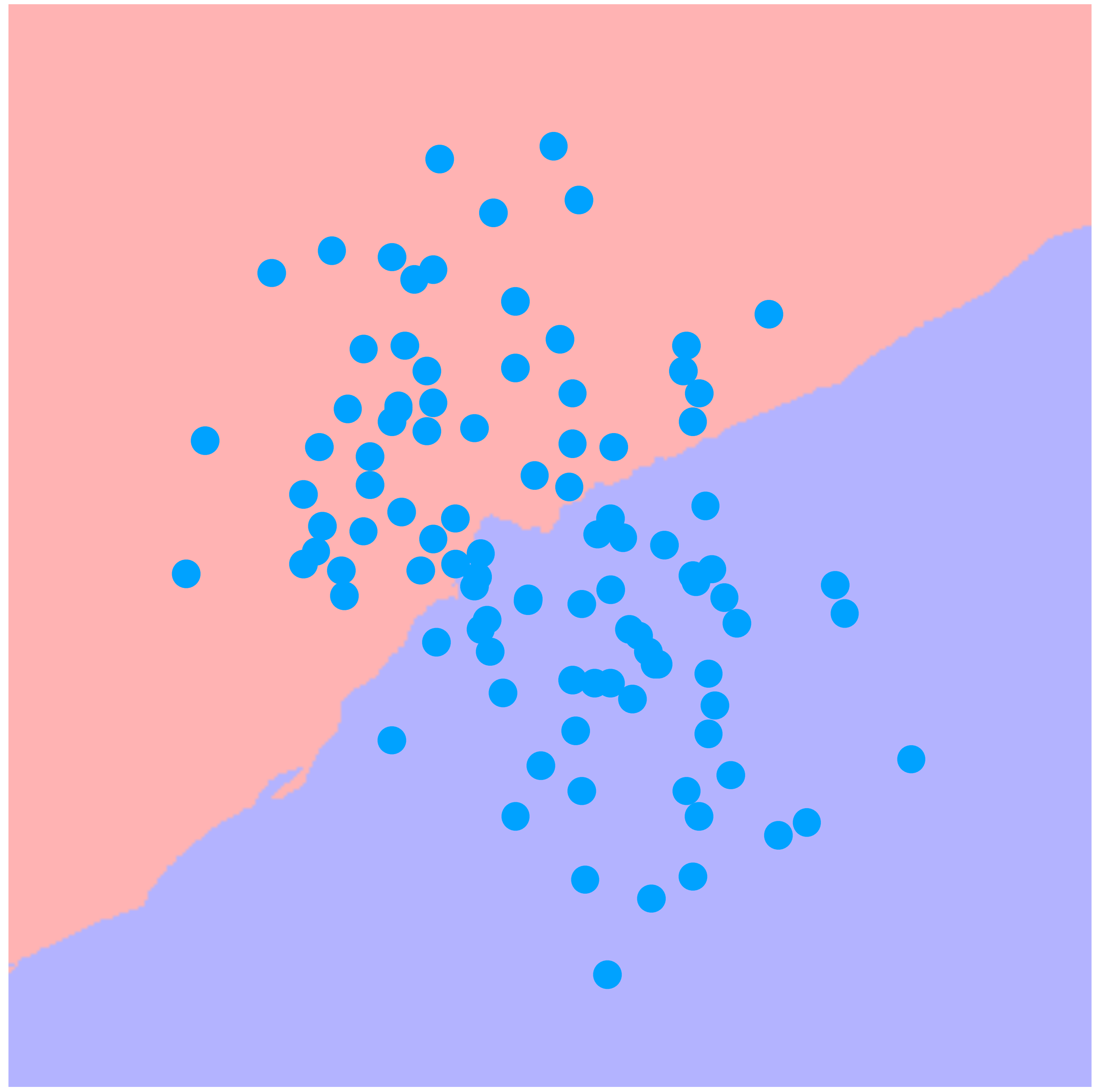
$k = 7$



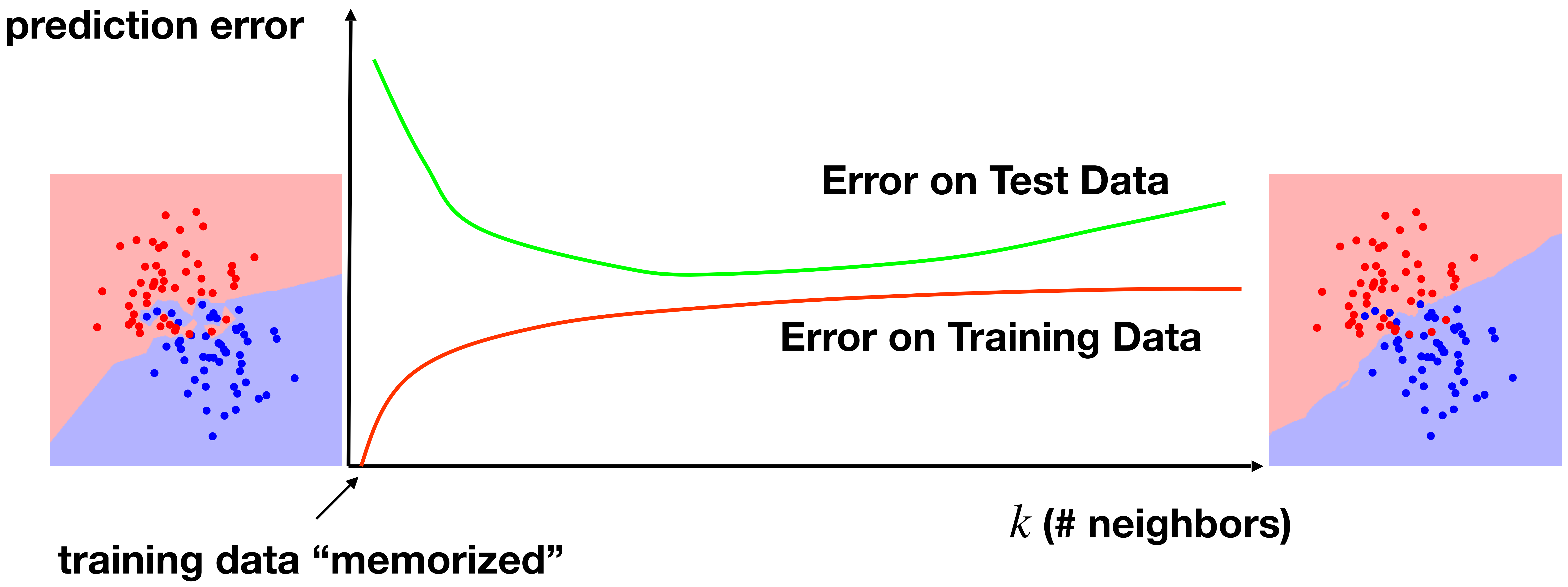
kNN decision boundary

- For classification, the decision boundary is piecewise linear
- Increasing k “simplifies” the decision boundary
  - Majority voting means less emphasis on individual points

$k = 25$



Error rates and k



- A complex model fits training data but generalizes poorly
- $k = 1$: perfect memorization of examples = complex
- $k = m$: predict majority class over entire dataset = simple
- We can select k with validation

kNN classifier: further considerations

- Decision boundary smoothness
  - Increases with k, as we average over more neighbors
  - Decreases with training size m, as more points support the boundary
  - Generally, optimal k should increase with m
- Extensions of k-Nearest Neighbors
  - Do features have the same scale? importance?
    - Weighted distance: $d(x, x') = \sqrt{\sum_i w_i (x_i - x'_i)^2}$
    - Non-Euclidean distances may be more appropriate for type of data
  - Fast search techniques (indexing) to find k closest points in high-dimensional space
  - Weighted average / voting based on distance: $\hat{y} = \sum_j w(d(x, x^{(j)}))y^{(j)}$

Recap: k-Nearest Neighbors

- Piecewise linear decision boundary
  - Just for analysis — the algorithm doesn't compute the boundary
- With $k > 1$:
  - Regression → (weighted) average
  - Classification → (weighted) vote
- Overfitting and complexity:
  - Model "complexity" goes down as k grows
  - Use validation data to estimate test error rates and select k

Today's lecture

Nearest Neighbors

Overfitting and complexity

k-Nearest Neighbors

Bayes classifiers

A basic classifier

- Training data: $\mathcal{D} = \{x^{(i)}, y^{(i)}\}_i$, classifier: $f(x; \mathcal{D})$
  - If x has discrete features → $f(x; \mathcal{D})$ is a contingency table
- Example: credit rating prediction (bad/good)
  - x = income (low / med / high)

Features	# bad	# good
x = 0	42	15
x = 1	338	287
x = 2	3	5

- How can we make the highest proportion of correct predictions?
  - Predict the more likely outcome for each possible observation

A basic classifier

- Training data: $\mathscr{D} = \{x^{(i)}, y^{(i)}\}_i$, classifier: $f(x; \mathscr{D})$
  - If x has discrete features → $f(x; \mathscr{D})$ is a contingency table
- Example: credit rating prediction (bad/good)
  - x = income (low / med / high)

Features	# bad	# good
x = 0	.7368	.2632
x = 1	.5408	.4592
x = 2	.3750	.6250

- How can we make the highest proportion of correct predictions?
  - Predict the more likely outcome for each possible observation
- Normalize counts into probabilities: $p(y = \text{good} \,|\, x = c)$
  - How does this scale to multiple features?

Logistics

assignment 1

- Assignment 1 is up on Canvas and gradescope
- Due: Thu, Jan 14 (PT)

lectures

- Lectures will be recorded and added to this playlist.