The beauty of kNN

Readings for today

- Chapter 2: Statistical learning. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning: with applications in R (Vol. 6). New York: Springer.

- Chapter 4: Classification. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning: with applications in R (Vol. 6). New York: Springer.

Topics

1. kNN classification

2. kNN regression

kNN Classification

The fundamental classification problem

$$\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = f\left(\begin{pmatrix} x_{1,1} & \cdots & x_{1,p} \\ \vdots & & \vdots \\ x_{n,1} & \cdots & x_{n,p} \end{pmatrix}\right)$$

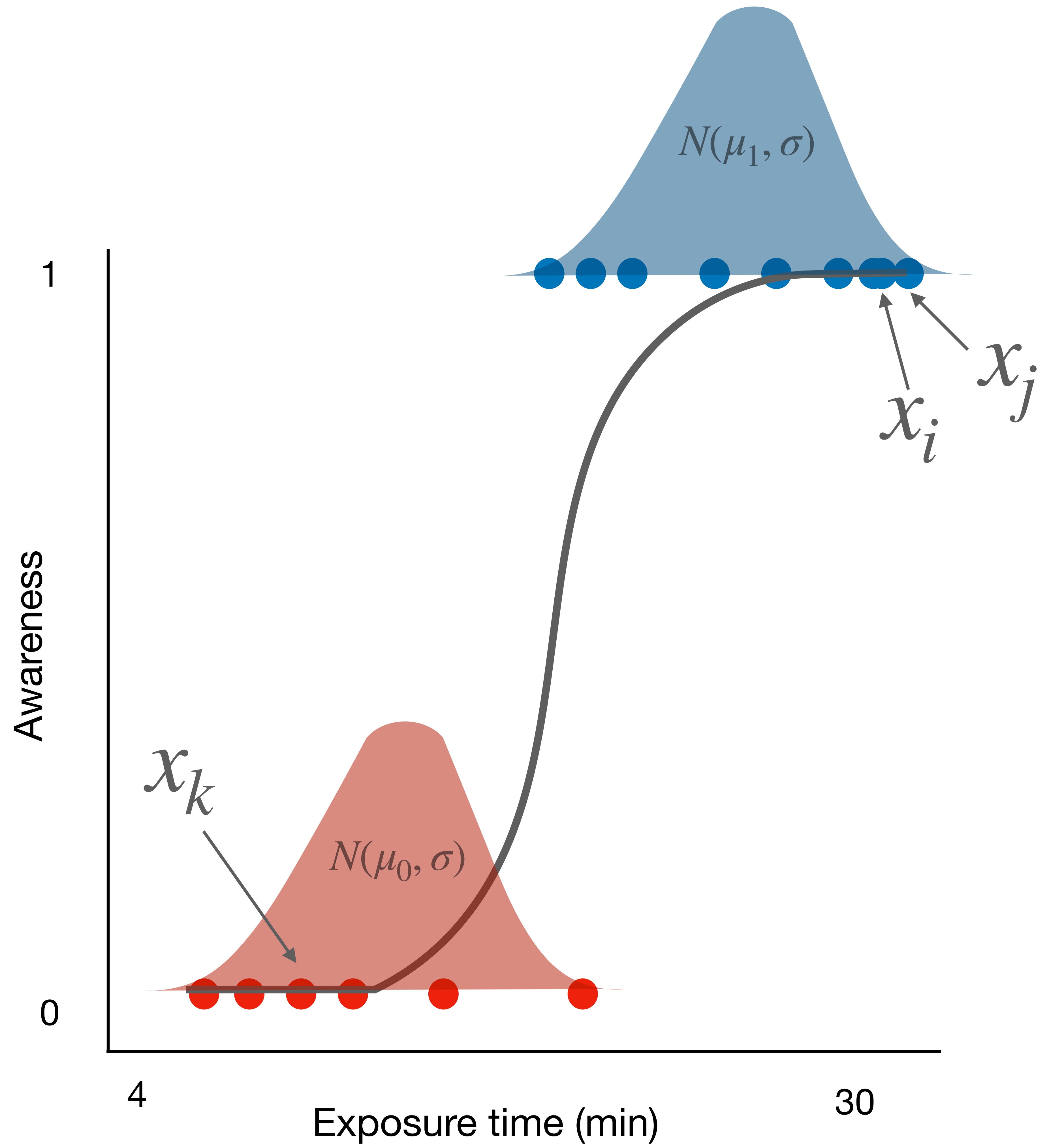
$$Y : \begin{cases} 1, & \text{if A} \\ 0, & \text{otherwise} \end{cases} \longrightarrow \text{categorical}$$

$$X \sim P(X \mid \beta) \longrightarrow \text{continuous}$$

$$P(Y = k \mid X = x_i) \leftarrow \text{Goal}$$

(James et al. 2013)

Nearest neighbors



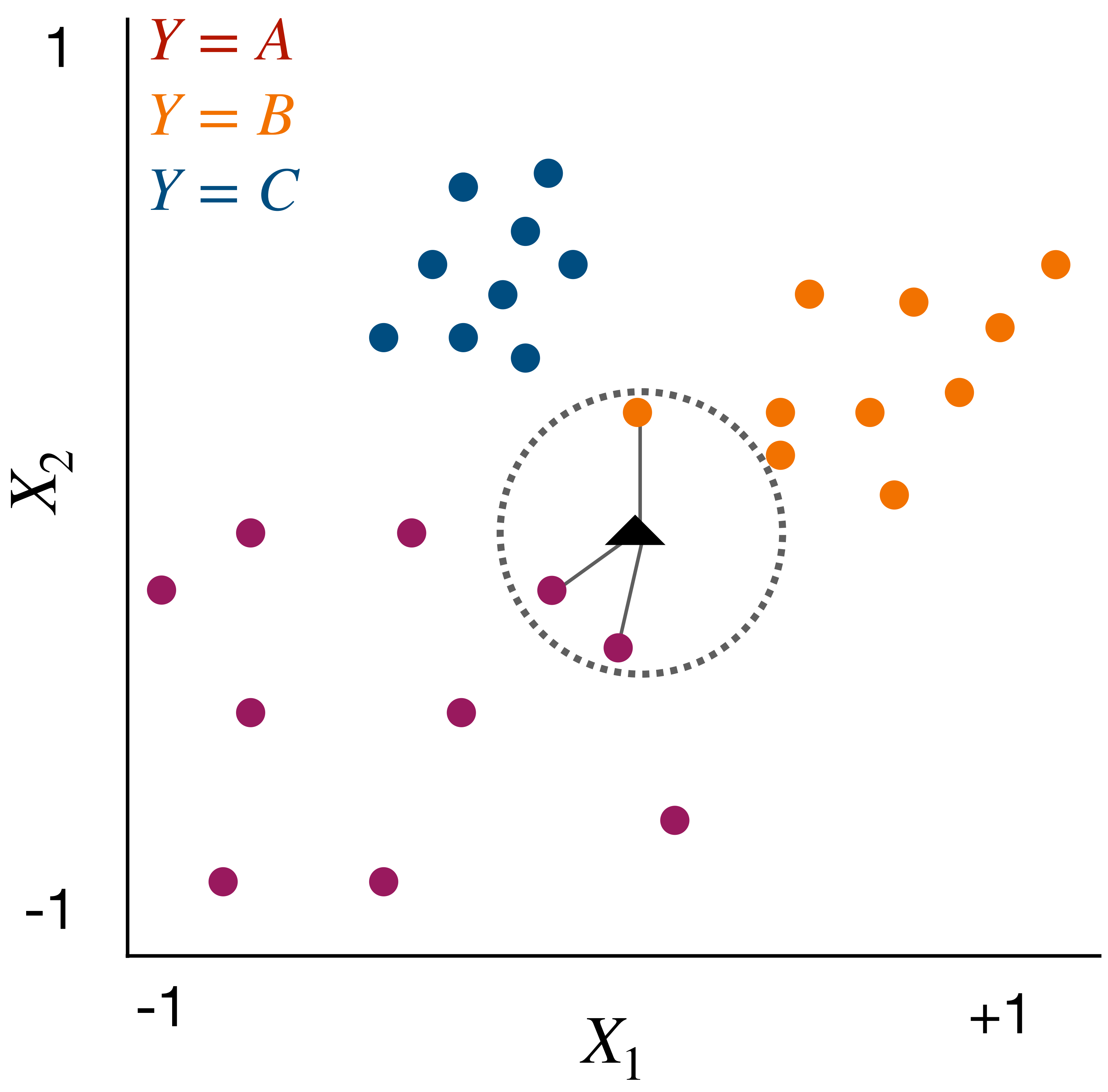
Interpretation

Observations in X that are closer together are likely to belong to the same group. No other assumptions required (i.e., non-parametric)

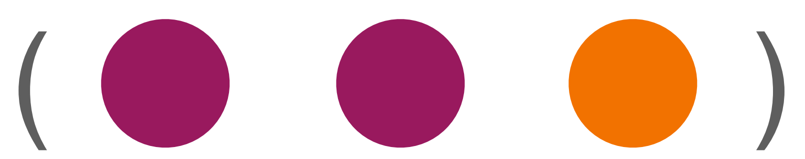
$\downarrow$ distance = $\uparrow$ likelihood

(James et al. 2013)

kNN classification



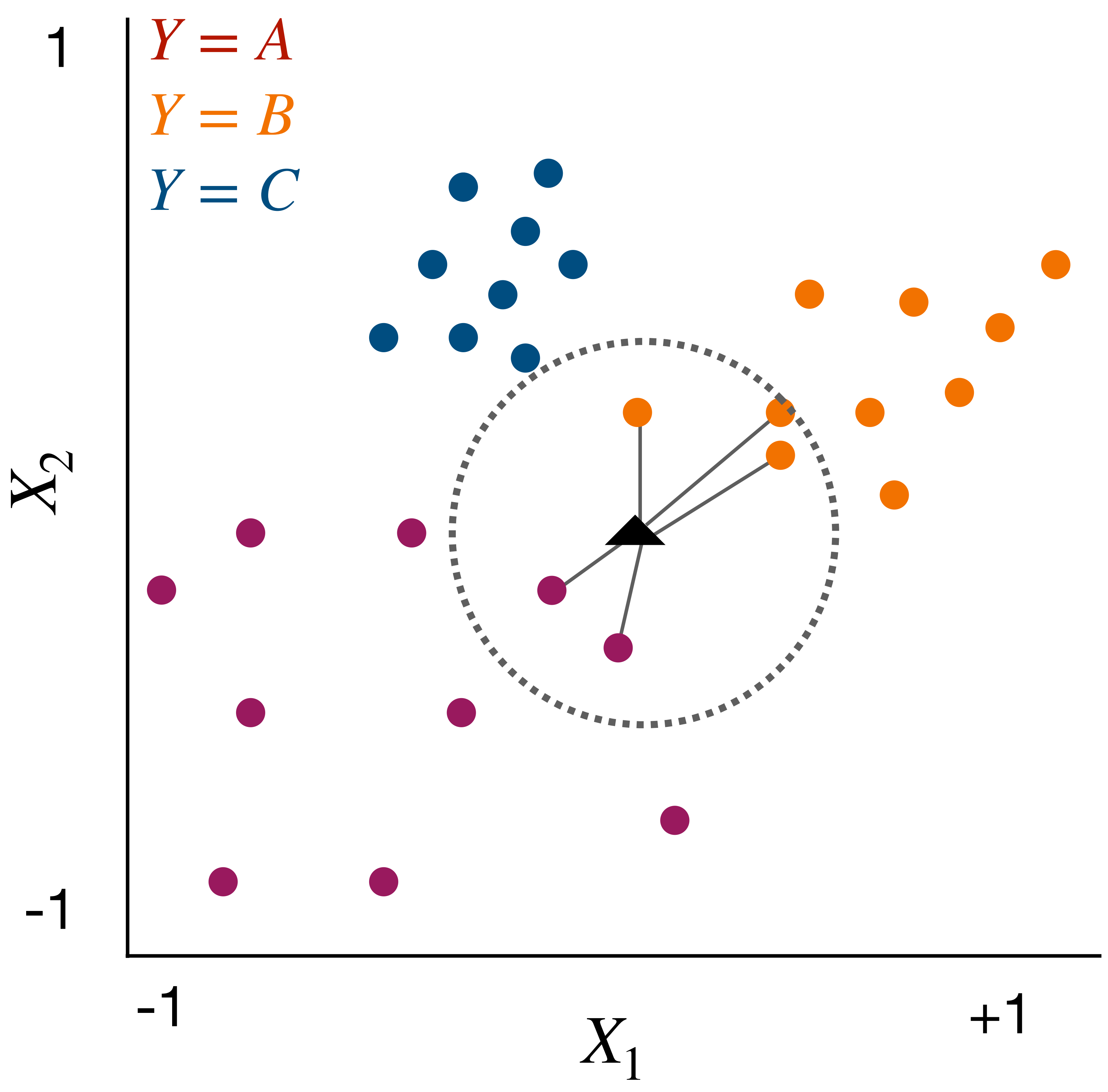
Euclidean distance: $d_{i,j} = \sqrt{\sum_{j=1}^{p} (x_{i,p} - x_{j,p})^2}$

k=3:  $\rightarrow A$

Decision: Categorize by popular vote.

(James et al. 2013)

kNN classification



Euclidean distance:

$$d_{i,j} = \sqrt{\sum_{j=1}^{p} (x_{i,p} - x_{j,p})^2}$$

k=3: (● ● ●) $\rightarrow A$

k=5: (● ● ● ● ●) $\rightarrow B$

Decision: Categorize by popular vote.

kNN classification algorithm

Step 1: Choose k.

Step 2: For every target observation x_i calculate all $d_{i,j}$.

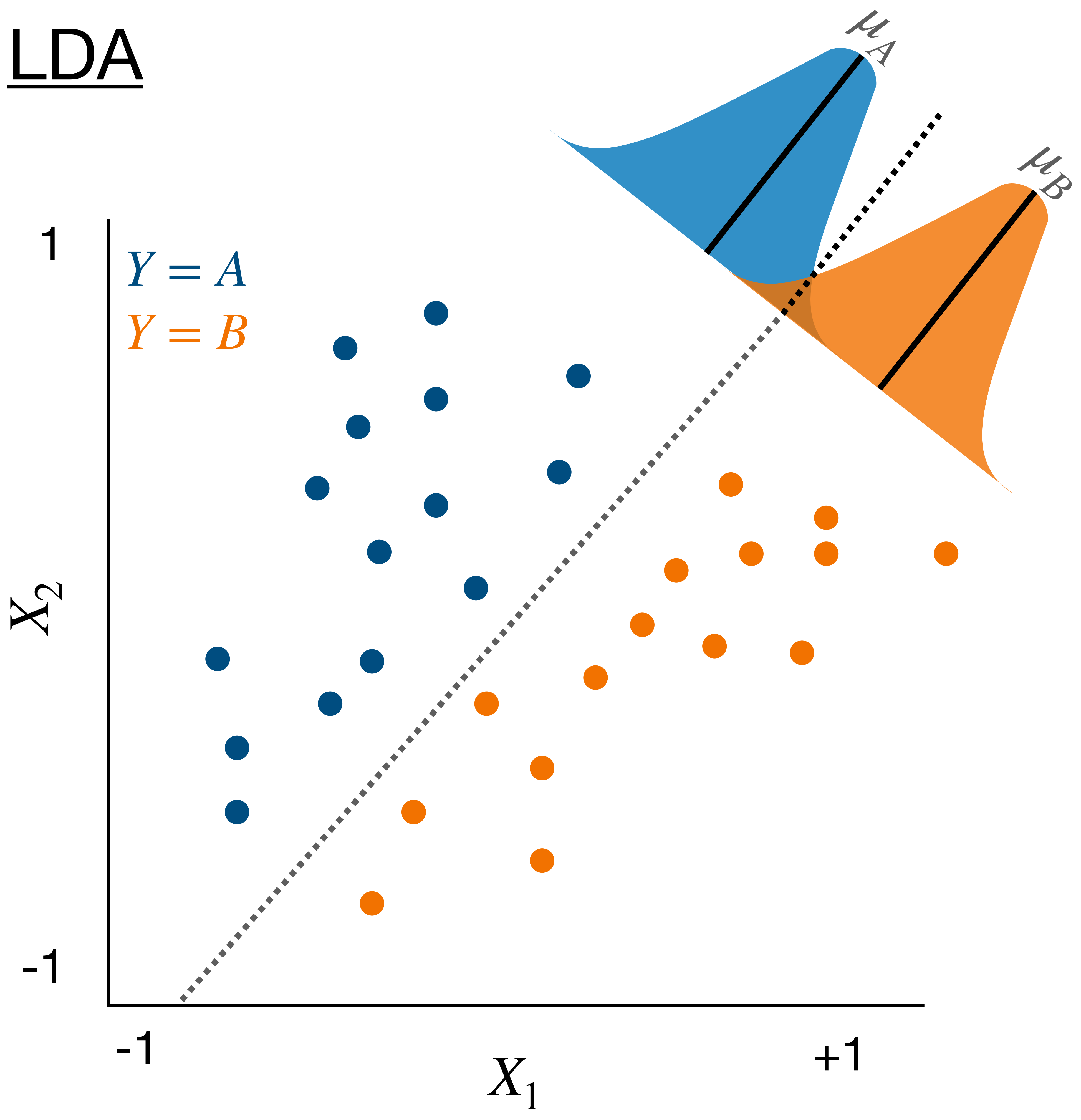
Step 3: Sort all $d_{i,j}$'s and select the k smallest to x_i

Step 4: Categorize based on median class in Step 3.

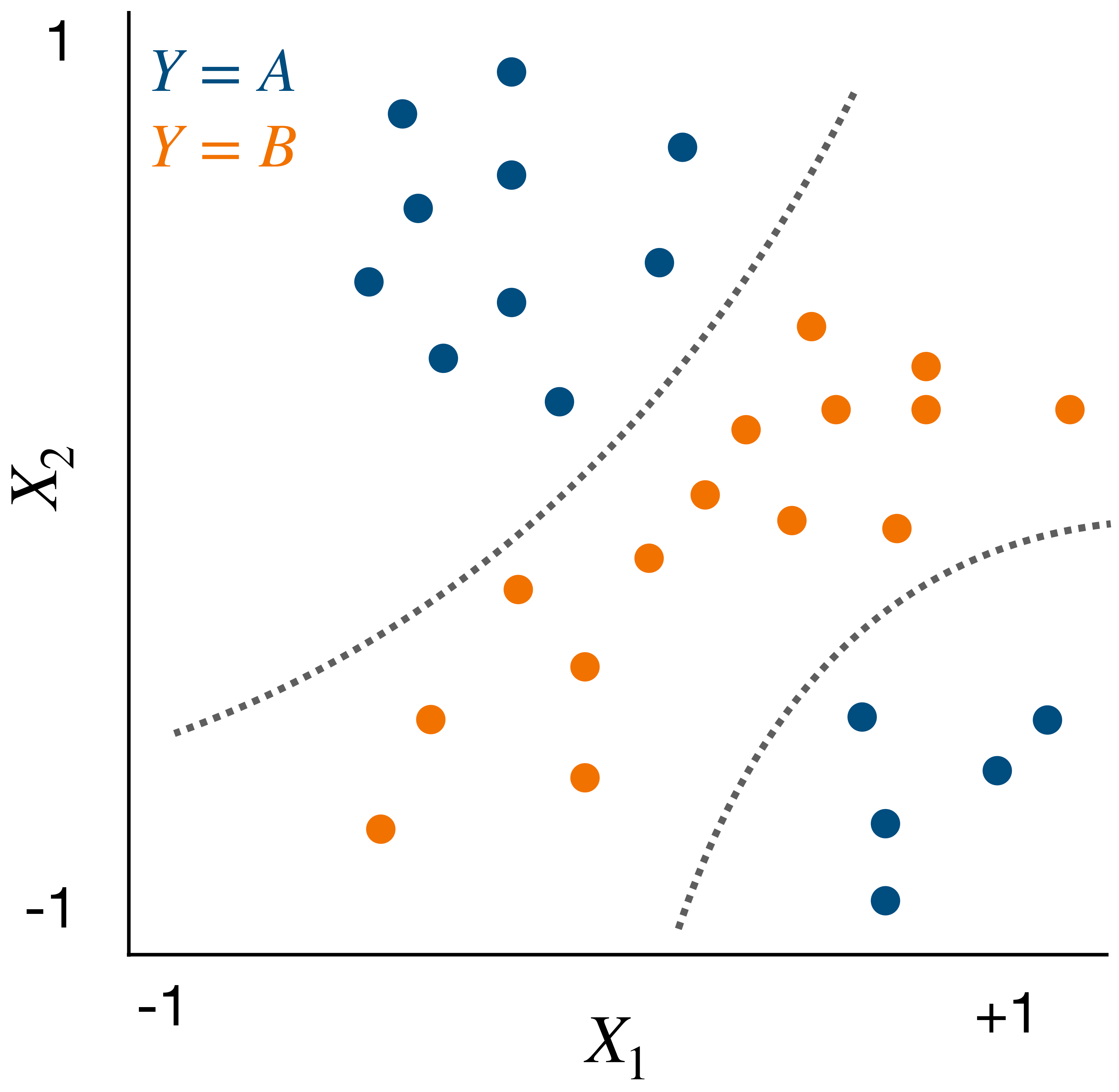
(James et al. 2013)

Decision boundaries

LDA

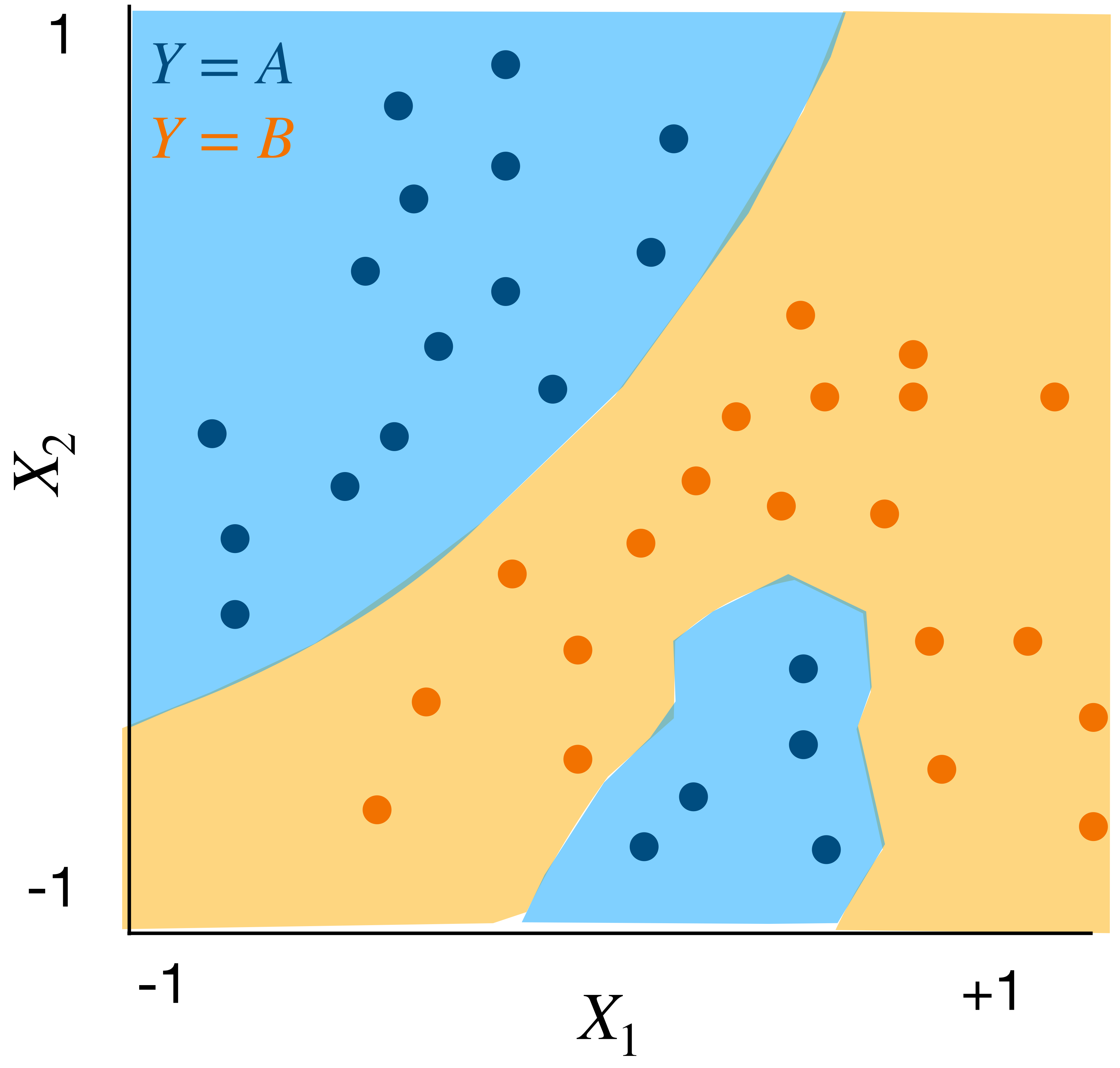


kNN



(James et al. 2013)

Defining territories via brute search



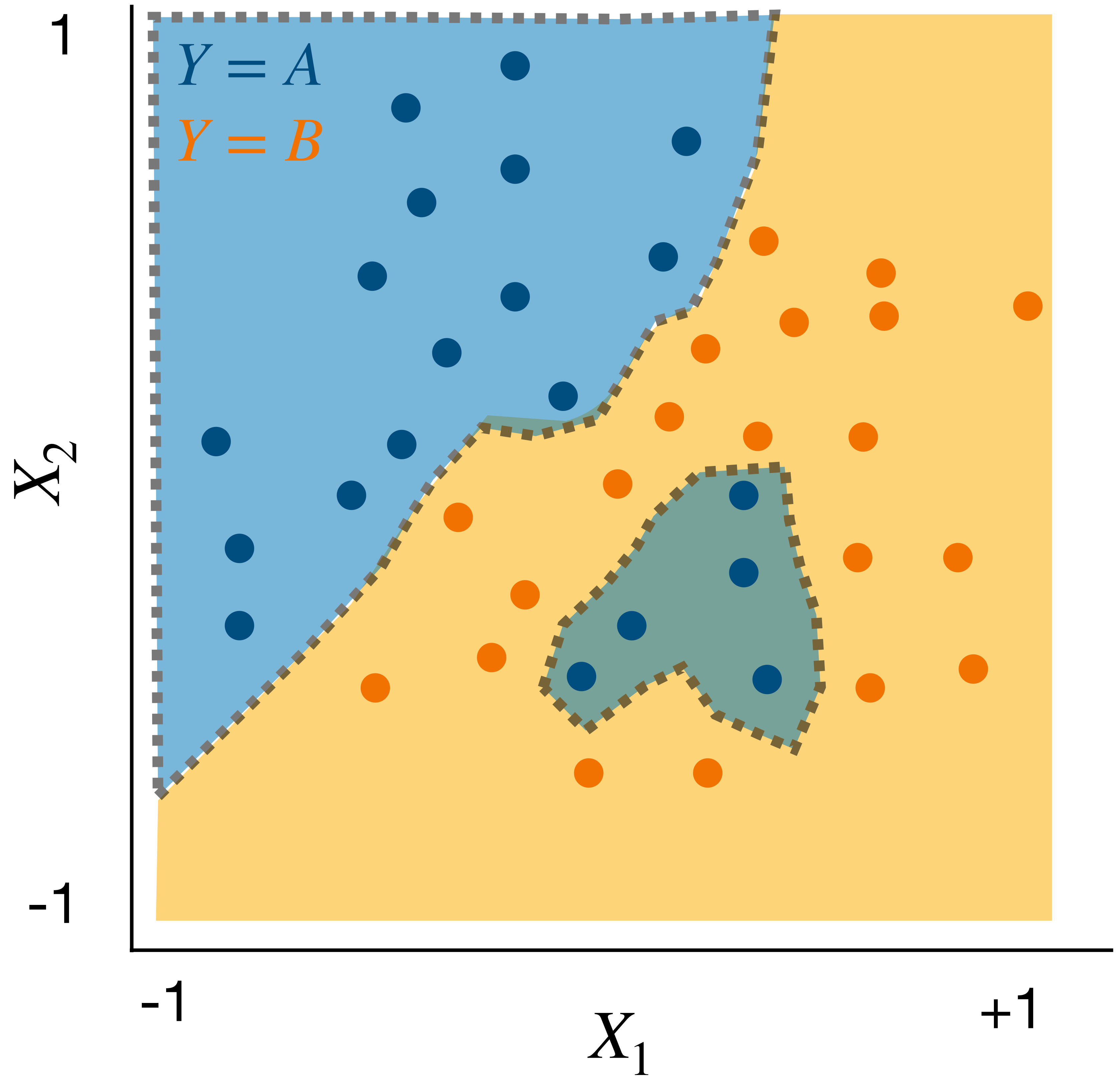
Territories

Iteratively search through all possible values of X (i.e. $[x_{min}, x_{max}]$) and use kNN to classify any possible state of X.

Decision Boundaries: Positions in X where the vote is an exact tie.

(James et al. 2013)

Bias-variance tradeoff



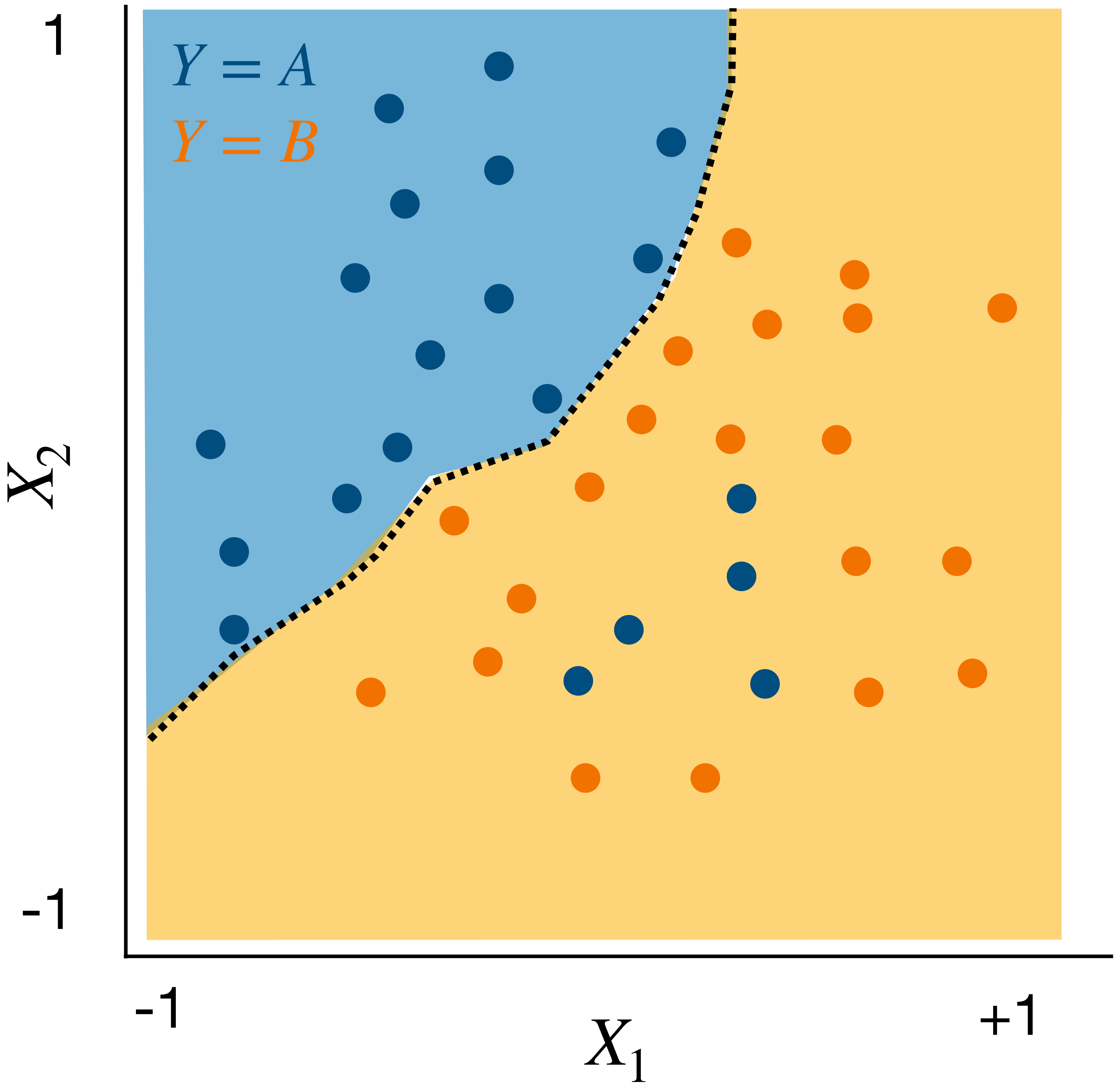
$\uparrow k = \downarrow$ variance

<u>k=1</u>

- $\uparrow$ flexibility
- “islands” of group clusters

(James et al. 2013)

Bias-variance tradeoff



$\uparrow k = \downarrow$ variance

k=1

- $\uparrow$ flexibility
- “islands” of group clusters

k=25

- clear segmentation
- higher error rate

(James et al. 2013)

Prediction with kNN classifiers

Full dataset

$$\begin{pmatrix} y_1 \\ \cdot \\ y_m \\ y_{m+1} \\ \vdots \\ y_n \end{pmatrix} = f(\begin{pmatrix} x_{1,1} & \dots & x_{1,p} \\ \cdot & & \cdot \\ x_{m,1} & \dots & x_{m,p} \\ x_{m+1,1} & \dots & x_{m+1,p} \\ \vdots & & \vdots \\ x_{n,1} & \dots & x_{n,p} \end{pmatrix})$$

Test set

$$\begin{pmatrix} \hat{y}_1 \\ \cdot \\ \hat{y}_m \end{pmatrix} = \hat{f}_{train}(\begin{pmatrix} x_{1,1} & \dots & x_{1,p} \\ \cdot & & \cdot \\ x_{m,1} & \dots & x_{m,p} \end{pmatrix})$$

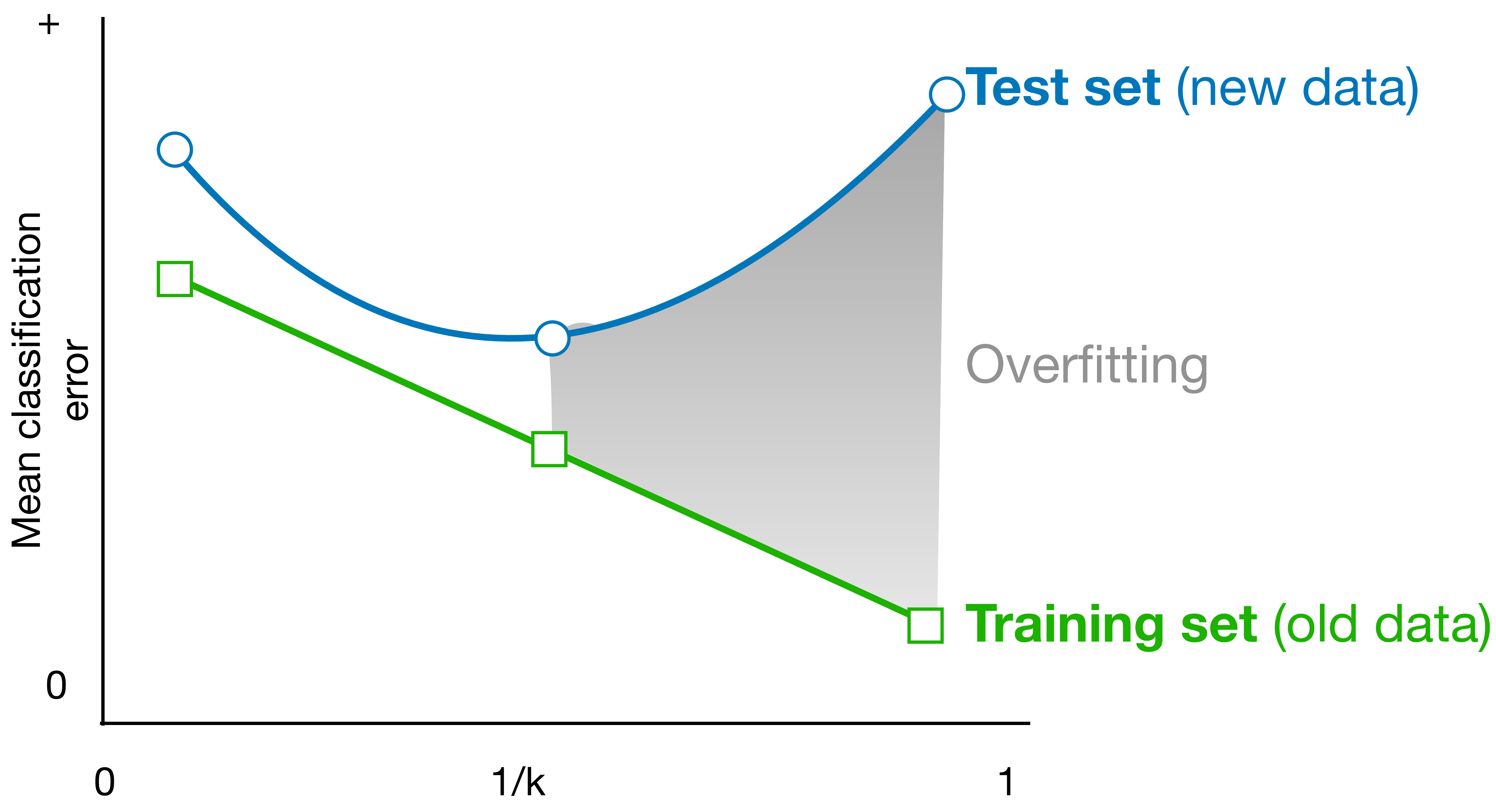
Training set

$$\begin{pmatrix} \hat{y}_{m+1} \\ \vdots \\ \hat{y}_n \end{pmatrix} = \hat{f}_{train}(\begin{pmatrix} x_{m+1,1} & \dots & x_{m+1,p} \\ \vdots & & \vdots \\ x_{n,1} & \dots & x_{n,p} \end{pmatrix})$$

Prediction: $\hat{y}_i^{test} = \hat{f}(X_i^{test}, [X^{train}, Y^{train}])$

(James et al. 2013)

Bias-variance tradeoff



$$\uparrow k = \downarrow \text{variance}$$

(James et al. 2013)

kNN Regression

Classification vs. regression with kNN

1. The classification problem: $\hat{y}_i = \hat{f}(x_i) = P(Y = k \,|\, X = x_i) = \frac{1}{m} \sum_{l \in m} I(y_l = k)$

$$I(y_i = k) = \begin{cases} 1, & \text{in k} \\ 0, & \text{otherwise} \end{cases}$$

2. The regression problem: $\hat{y}_i = \hat{f}(x_i) = P(Y = k \,|\, X = x_i) = \frac{1}{m} \sum_{l \in m} y_l$

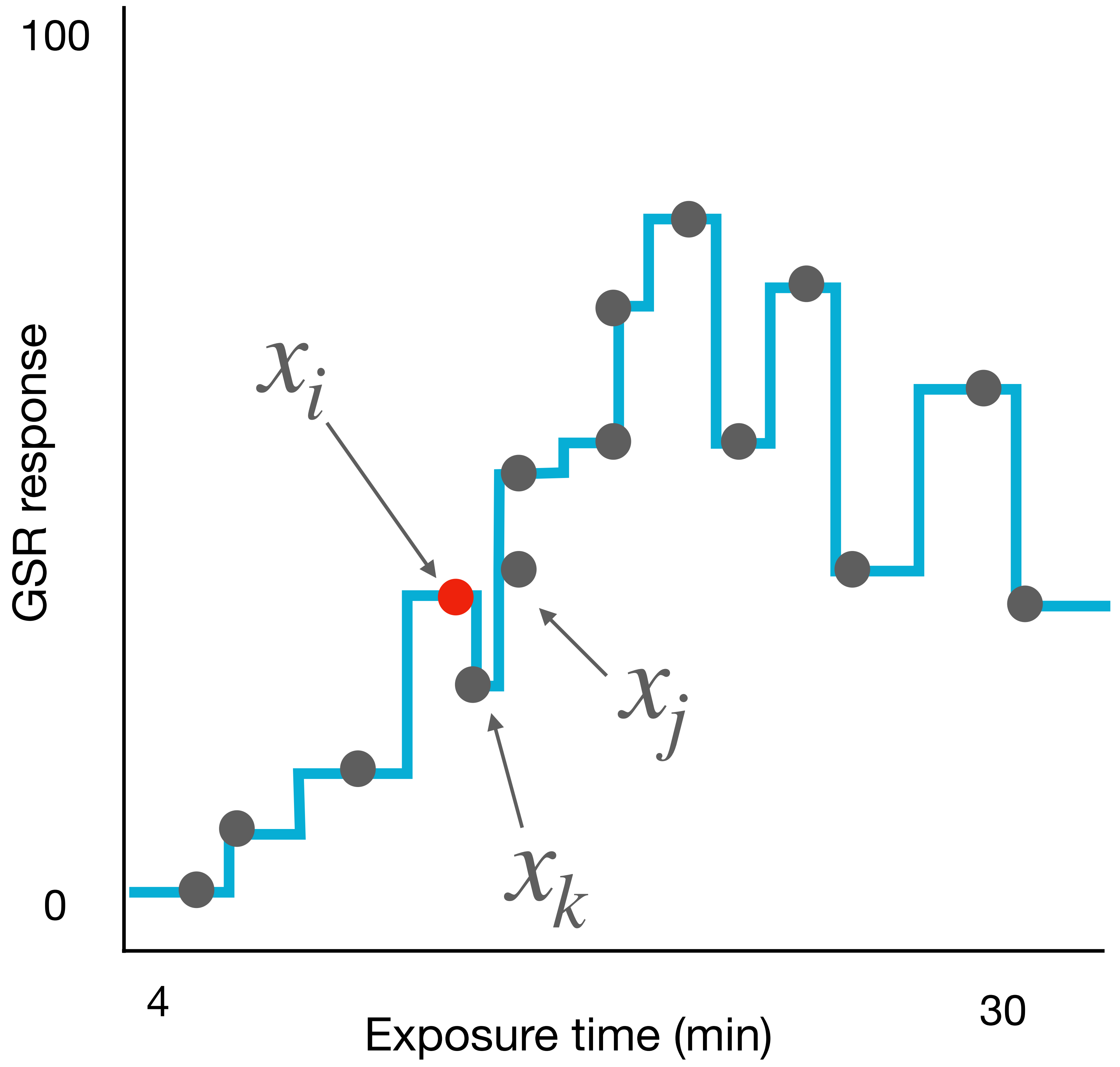
Regression with kNN is a classification problem where every value of y is its own unique category.

(James et al. 2013)

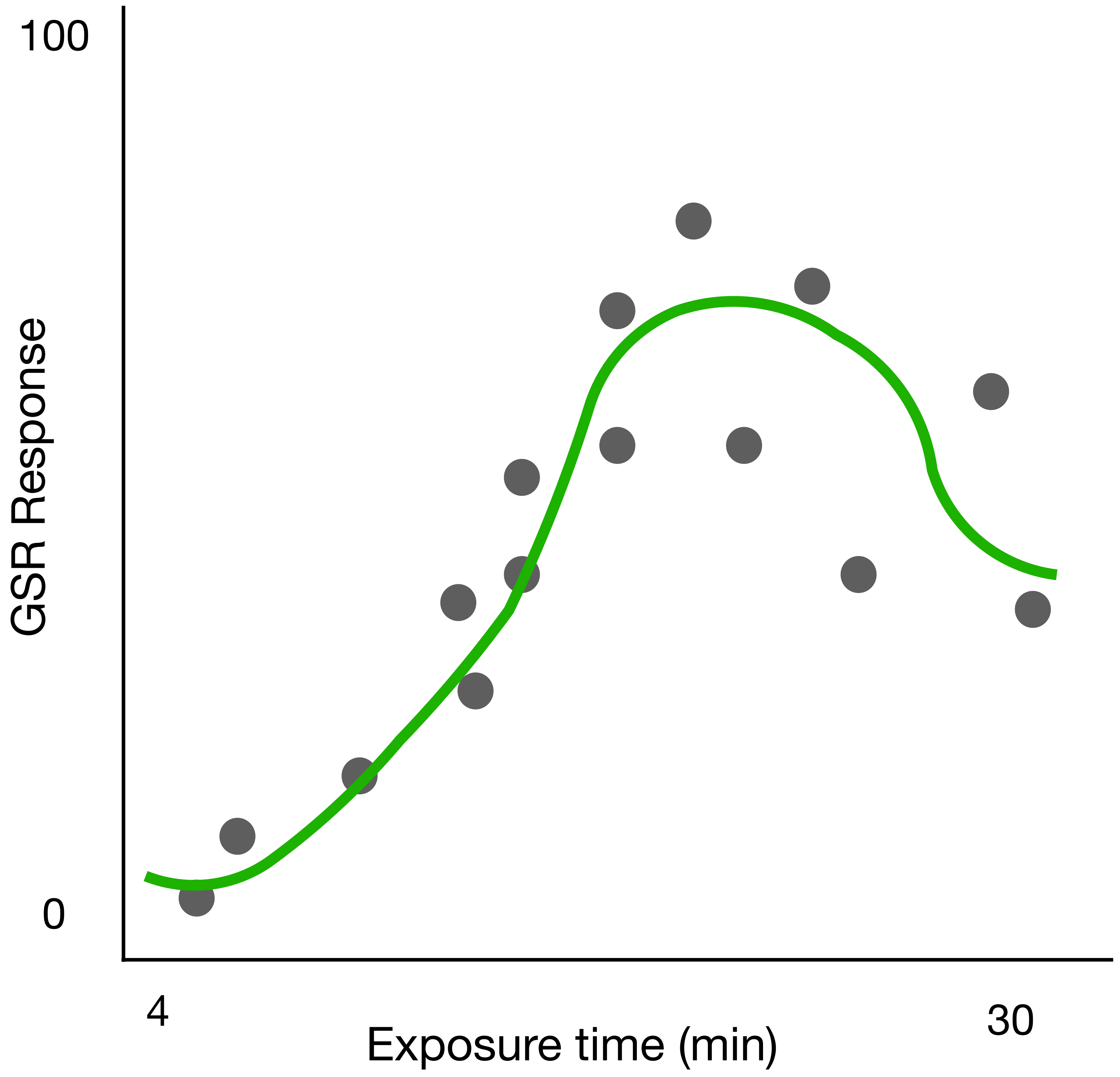
kNN regression

$\uparrow k = \downarrow$ variance

k=1

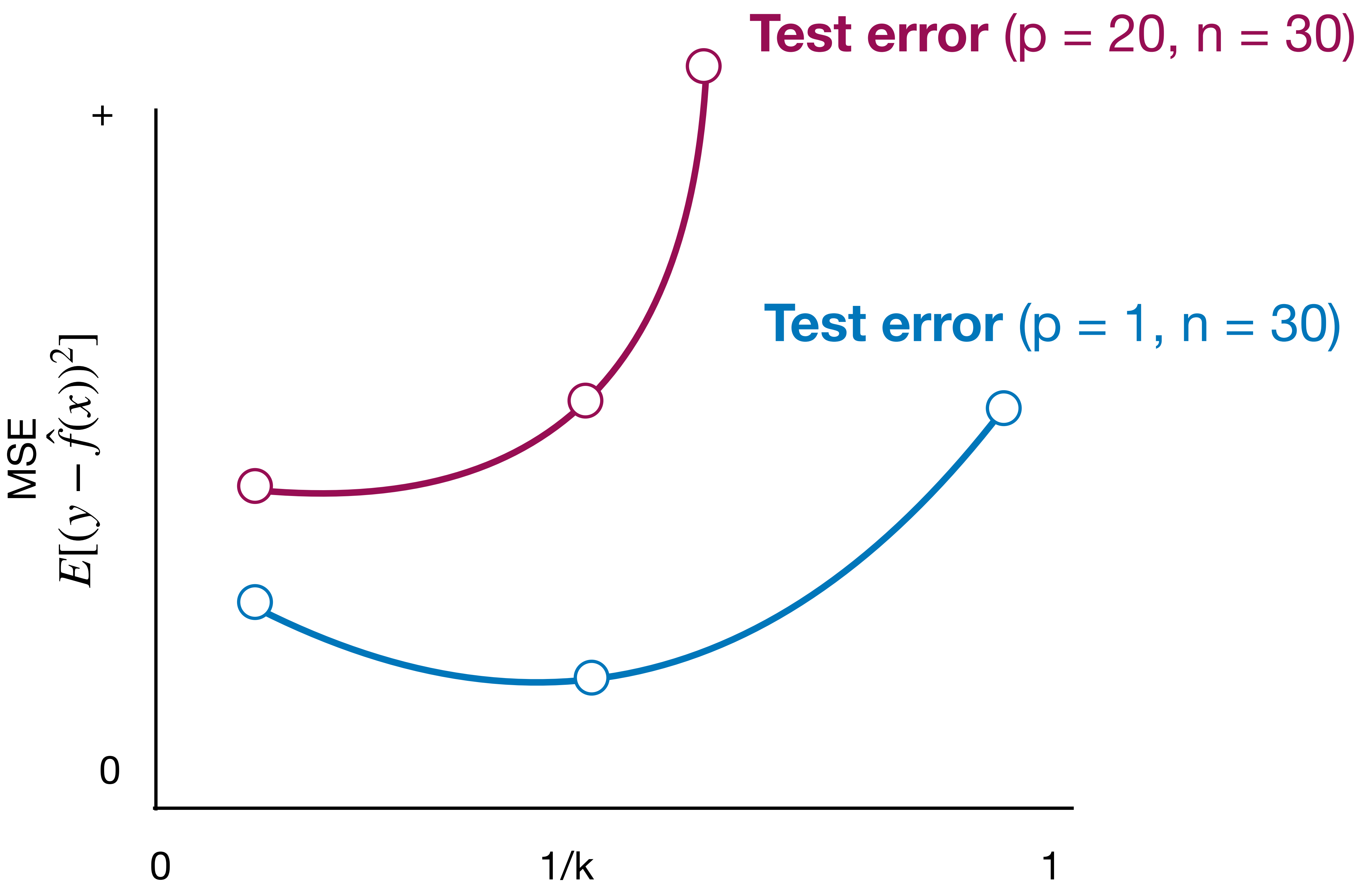


k=25



(James et al. 2013)

Curse of dimensionality

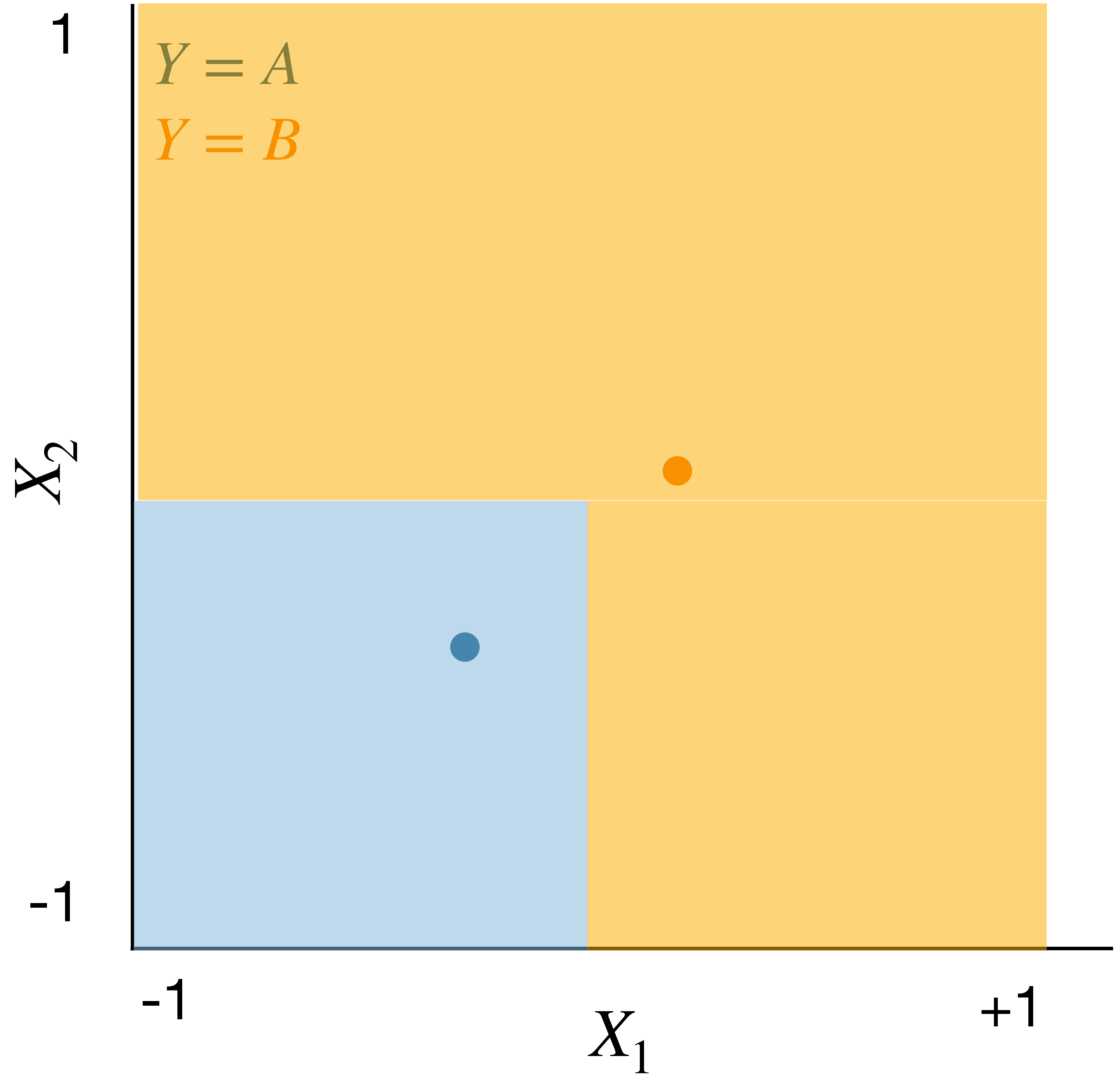


Problem:

As $n \to p$, there are not enough neighbors to query and the distance become too sparse.

(James et al. 2013)

Curse of dimensionality



kNN fails when $\frac{p}{n}$ gets too high

Problem:

As $n \rightarrow p$, there are not enough neighbors to query and the distance become too sparse.

(James et al. 2013)

Take home message

- kNN offers a simple, non-parametric way to ask classification and regression questions in the prediction context.