CS 480-680

1/8/26

Lecture 2

Examples prediction

Nearest Neighbor (1NN)

L1 Posted
lecture notes posted

HW1 → TB Posted
Wed

TA OH Fri 9:30-10:20
10:30-11:20

Math Refresher

Lecture Notes I – Examples of Predictors. Nearest Neighbor and Kernel Predictors

Marina Meilă
mmp@uwaterloo.ca

With Thanks to Pascal Poupart & Gautam Kamath
Cheriton School of Computer Science
University of Waterloo

January 6, 2026

supervised learning

Prediction problems by the type of output

+ examples

The Nearest-Neighbor and kernel predictors

Some concepts in Classification

Reading HTF Ch.: 2.3.2 Nearest neighbor, 6.1–3. Kernel regression, 6.6.2 kernel classifiers,, Murphy Ch.: , Bach Ch.:

Hastie, Tibs.., F...

Prediction problems by the type of output

In supervised learning, the problem is *predicting* the value of an **output** (or **response** – typically in regression, or **label** – typically in classification) variable Y from the values of some observed variables called **inputs** (or **predictors, features, attributes**) $(X_1, X_2, \dots X_d) = X$. Typically we will consider that the input $X \in \mathbb{R}^d$. Prediction problems are classified by the type of response $Y \in \mathcal{Y}$:

- *regression*: $\boxed{Y \in \mathbb{R}}$
- *binary classification*: $Y \in \{-1, +1\}$
- *multiway classification*: $Y \in \{y_1, \dots y_m\}$ a finite set digits
- *ranking*: $Y \in \mathbb{S}_p$ the set of permutations of p objects
- *multilabel classification* $Y \subseteq \{y_1, \dots y_m\}$ a finite set (i.e. each X can have several labels)
- *structured prediction* $Y \in \Omega_V$ the state space of a graphical model over a set of [discrete] variables V

Example (**Regression.**)

- Y is the proportion of high-school students who go to college from a given school in given year. X are school attributes like class size, amount of funding, curriculum (note that they aren't all naturally real valued), median income per family, and other inputs like the state of the economy, etc. Note also that $Y \in [0, 1]$ here.
- $Y \geq 0$ is the income of a person, and X_j are attributes like education, age, years out of school, skills, past income, type of employment.

  Economic forecasts are another example of regression. Note that in this problem as well as in the previous, the Y in the previous period, if observed, could be used as a predictor variable for the next Y. This is typical of structured prediction problems.
- Weather prediction is typically a regression problem, as winds, rainfall, temperatures are continuous-valued variables.
- Predicting the box office totals of a movie. What should the inputs be?
- Predicting perovskite degradation. Perovskites are a type of crystal considered promising for the fabrication of solar cells. In standard use, such a material must have a life time Y of 30 years. How can one predict which material will last that long without waiting for 30 years?

  Y is time to degradation, X_j are material composition, experimental conditions, and measurements of initial values of physical parameters.

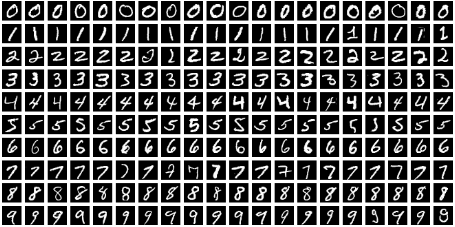
Example ((Anomaly) detection.)

This is a binary classification problem. $Y \in \{\mathrm{normal}, \mathrm{abnormal}\}$. For instance, Y could be “HIV positive” vs “HIV negative” (which could be abbreviated as “$+$”, “-”) and the inputs X are concentration of various reagents and lymph cells in the blood.
Anomaly detection is a problem also in artificial systems, as any device may be functioning normally or not. There are also more general detection problems, where the object detected is of scientific interest rather than an “alarm”: detecting Gamma-ray bursts in astronomy, detecting meteorites in Antarctica (a robot collects rocks lying on the ice and determines if the rock is terrestrial or meteorite). More recently, *Artificial Intelligence* tasks like detecting faces/cars/people in images or video streams have become possible.

LSST

Example (**Multiway classification.)**

Handwritten digit classification: $Y \in \{0, 1, \dots 9\}$ and X=black/white 64× 64 image of the digit. For example, ZIP codes are being now read automatically off envelopes.
OCR (Optical character recognition). The task is to recognized printed characters automatically. X is again a B/W digital image, $Y \in \{a - z, A - Z, 0 - 9, "." , "," , \dots\}$, or another character set (e.g. Chinese).



Example (**Diagnosis)**

Diagnosis is multiway classification + anomaly detection. $Y = 0$ means "normal/healthy", while $Y \in \{1, 2, \dots\}$ enumerates failure modes/diseases.

Example (**Structured prediction.**)

Structure = graphical model

Speech recognition. X is a segment of digitally recorded speech, Y is the word corresponding to it. Note that it is not trivial to *segment speech*, i.e to separate the speech segment that corresponds to a given word. These segments have different lengths too (and the length varies even when the same word is spoken).
The classification problem is to associate to each segment X of speech the corresponding word. But one notices that the words are not indepedent of other neighboring words. In fact, people speak in sentences, so it is natural to recognize each word in dependence from the others. Thus, one imposes a graphical model structure on the words corresponding to an utterance $X^1, X^2, \ldots X^m$. For instance, the labels $Y^{1:m}$ could form a *chain* $Y^1 - Y^2 - \ldots Y^m$. Other more complex graphical models structures can be used too.

Example (**Large Language Models (LLM)**)

LLMs are machines for structured prediction, where the label space $\mathcal{Y}$ coincides with the input space $\mathcal{X}$, e.g., both the input and the output are sequences of English words (called **tokens**). The output tokens $y^1, y^2, \ldots y^t \ldots$ depend on the previous output tokens (the context), as well as on the entire sequence of input tokens.

Part-of-speech tagging

$Y^1 — Y^2 — \quad — Y^5$

Noun Verb A Adj N

Mary had a little lamb ← speech

$X^1 — X^2 — X^3 — X^4 — X^5$

Supervised Learning

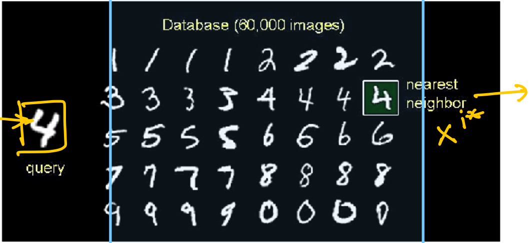
- Nearest neighbour: predictor

Training set

$\mathcal{D} = \{(x^i, y^i), i=1:n\}$

new x

$y = ?$



x^{i*}

output $y^{i*} = \hat{y}$

x^{i*} = most similar to x

$x^{1:n}$

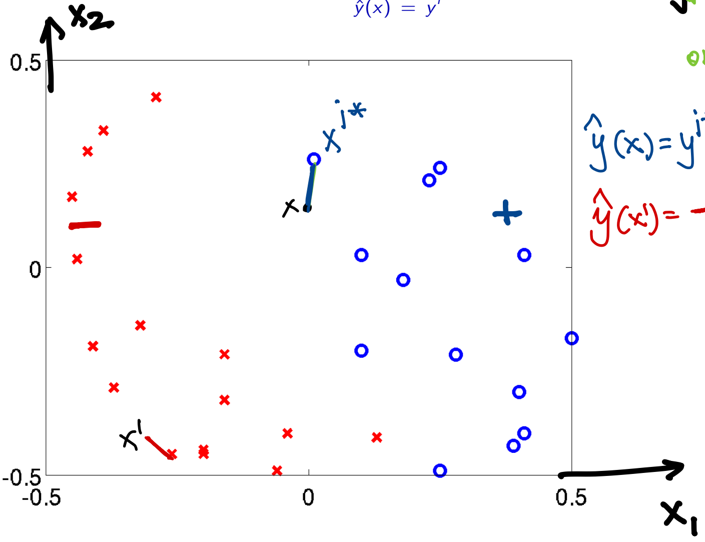
The Nearest-Neighbor predictor

$x \in \mathbb{R}^2$

x_j^i ← example index; ← dimension or feature

- **1-Nearest Neighbor** The label of a point x is assigned as follows:
  1. find the example x^i that is nearest to x in $\mathcal{D}$ (in Euclidean distance)
  2. assign x the label y^i, i.e.

$$\hat{y}(x) = y^i$$



$\hat{y}(x) = y^{j*} = +$

$\hat{y}(x') = -$

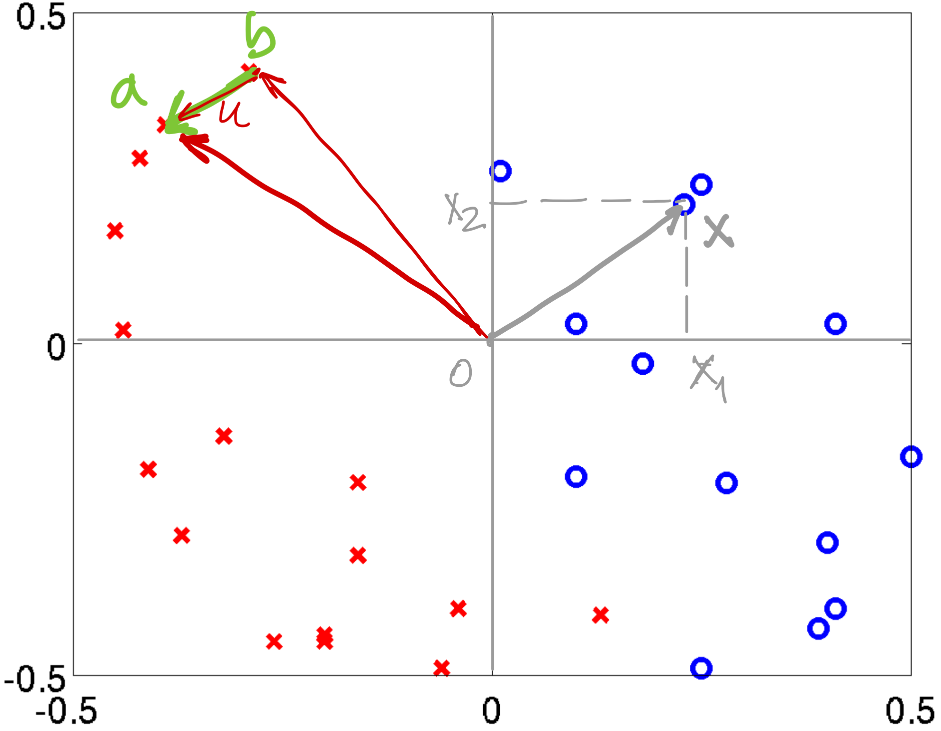
$i = 1, 2, \ldots n$

$j \in \{1, 2\}$

$$\|u\|_2 = \sqrt{u_1^2 + u_2^2}$$

$$a = b + u$$

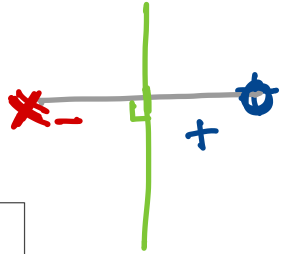
$u = a - b$



Analysis of NN

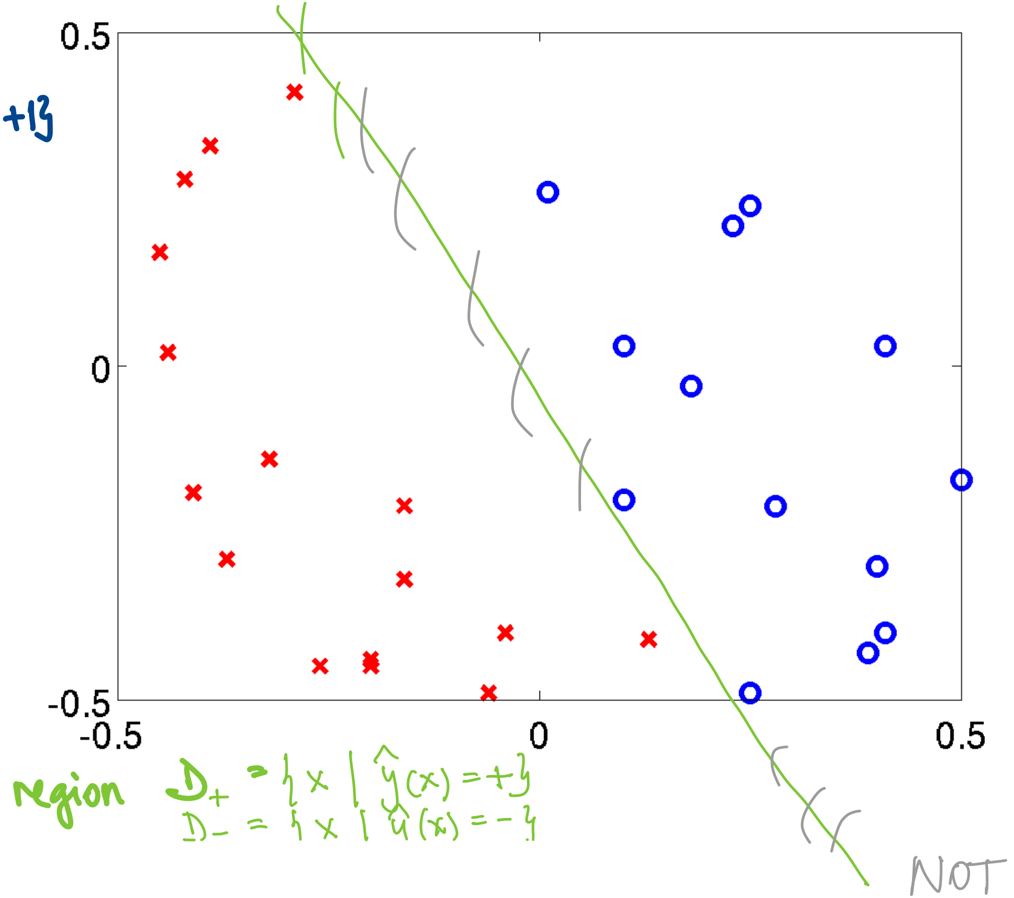
$x \longrightarrow f(x) = \hat{y}$

$f : \mathbb{R}^d \to \{-1, +1\}$



Mediatrix

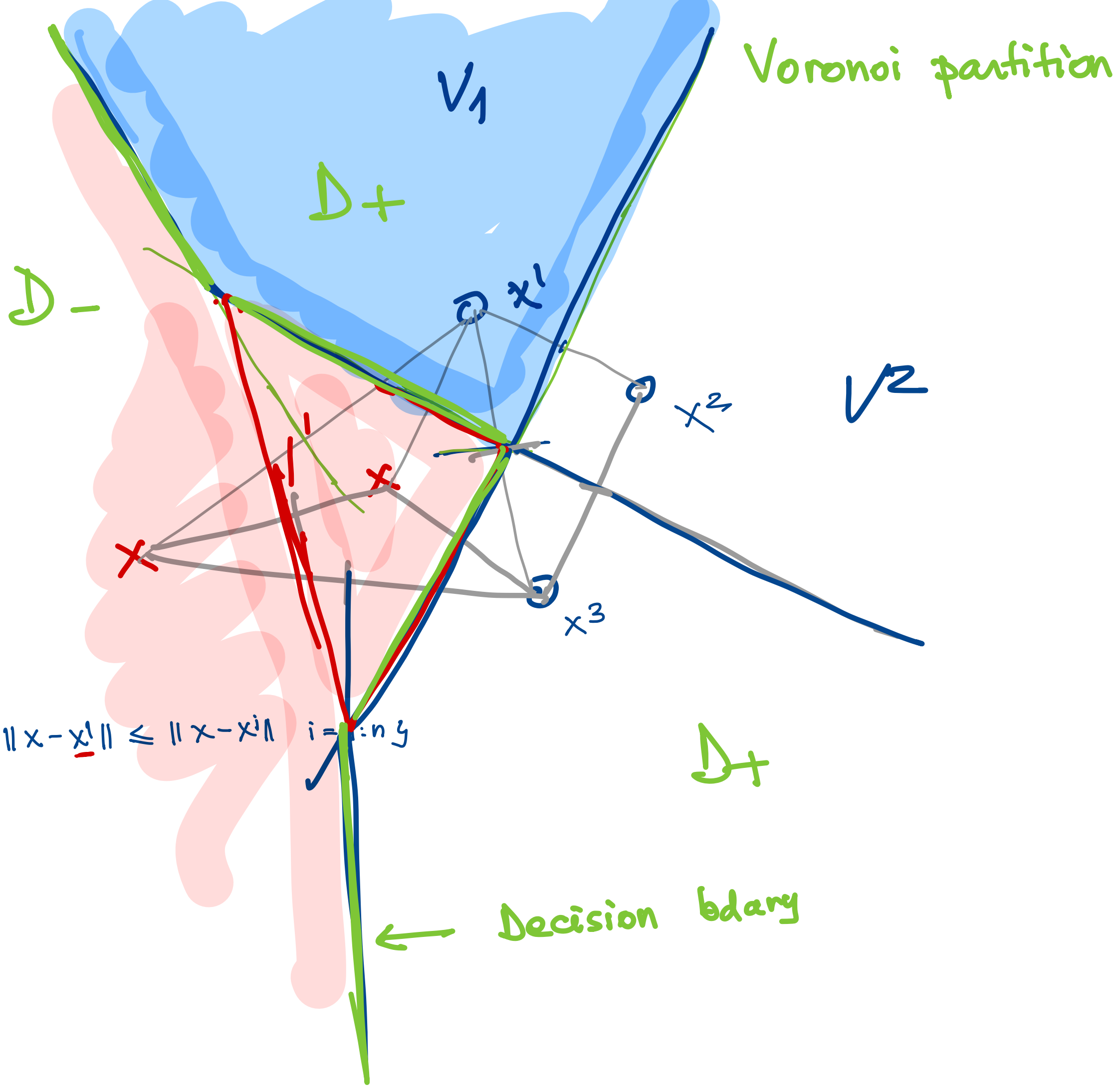
= decision boundary



Decision region $D_+ = \{x \mid \hat{y}(x) = +\}$

$D_- = \{x \mid \hat{y}(x) = -\}$

NOT NN



$$V^1 = \{ x \in \mathbb{R}^d \mid \|x - x^1\| \leq \|x - x^i\| \quad i = 1{:}n \}$$

The Nearest-Neighbor predictor

- **1-Nearest Neighbor** The label of a point x is assigned as follows:
  1. find the example x^i that is nearest to x in $\mathcal{D}$ (in Euclidean distance)
  2. assign x the label y^i, i.e.

$$\hat{y}(x) = y^i$$

- **K-Nearest Neighbor** (with $K = 3, 5$ or larger)
  1. find the K nearest neighbors of x in $\mathcal{D}$: $x^{i_1, \dots i_K}$
  2.
     - for classification $f(x) =$ the most frequent label among the K neighbors (well suited for multiclass)
     - for regression $f(x) = \frac{1}{K}\sum_{i \text{ neighbor of } x} y^i$ = mean of neighbors' labels

Disadvantages

- computation $O(nd)$
- "nonparametric" - hard to analyze

+ can approximate any f arbitrarily well

- choosing K ← how to handle outliers
- not differentiable

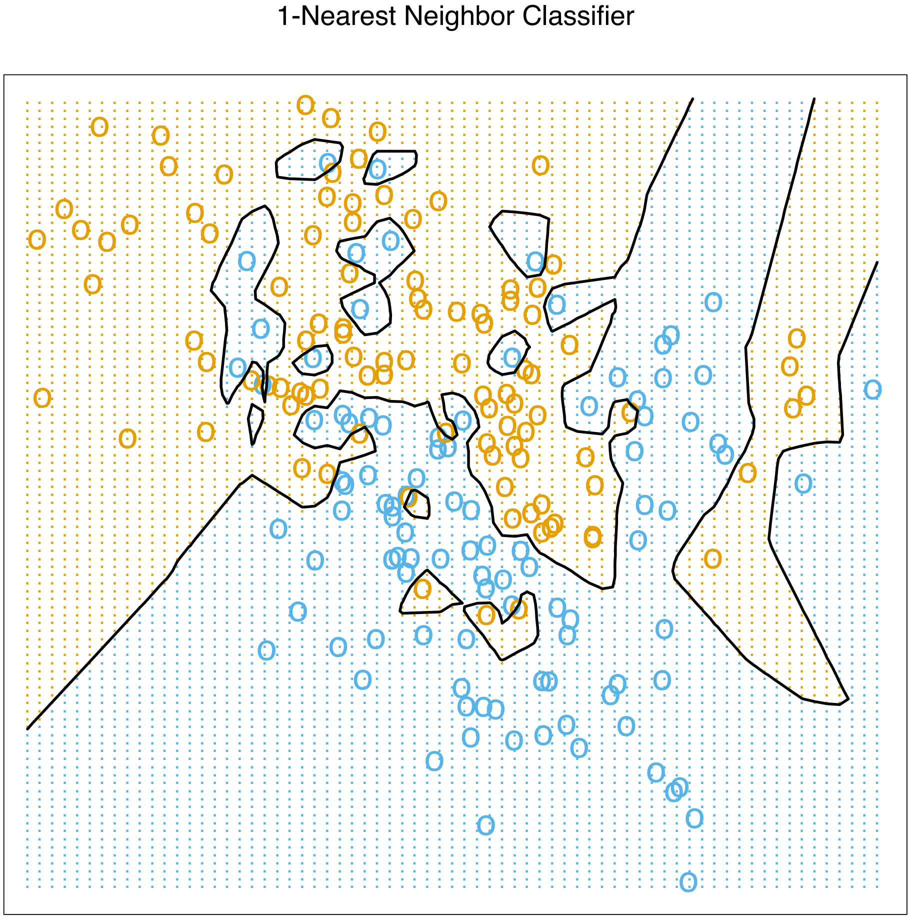


FIGURE 2.3. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (BLUE $= 0$, ORANGE $= 1$)*, and then predicted by* 1*-nearest-neighbor classification.*

The Nearest-Neighbor predictor

- **1-Nearest Neighbor** The label of a point x is assigned as follows:
  1. find the example x^i that is nearest to x in $\mathcal{D}$ (in Euclidean distance)
  2. assign x the label y^i, i.e.

$$\hat{y}(x) = y^i$$

- **K-Nearest Neighbor** (with $K = 3, 5$ or larger)
  1. find the K nearest neighbors of x in $\mathcal{D}$: $x^{i_1, \dots i_K}$
  2. - for classification $f(x) =$ the most frequent label among the K neighbors (well suited for multiclass)
     - for regression $f(x) = \frac{1}{K} \sum_{i \text{ neighbor of } x} y^i$ = mean of neighbors' labels

- No parameters to estimate!
- No training!
- But all data must be stored (also called memory-based learning)

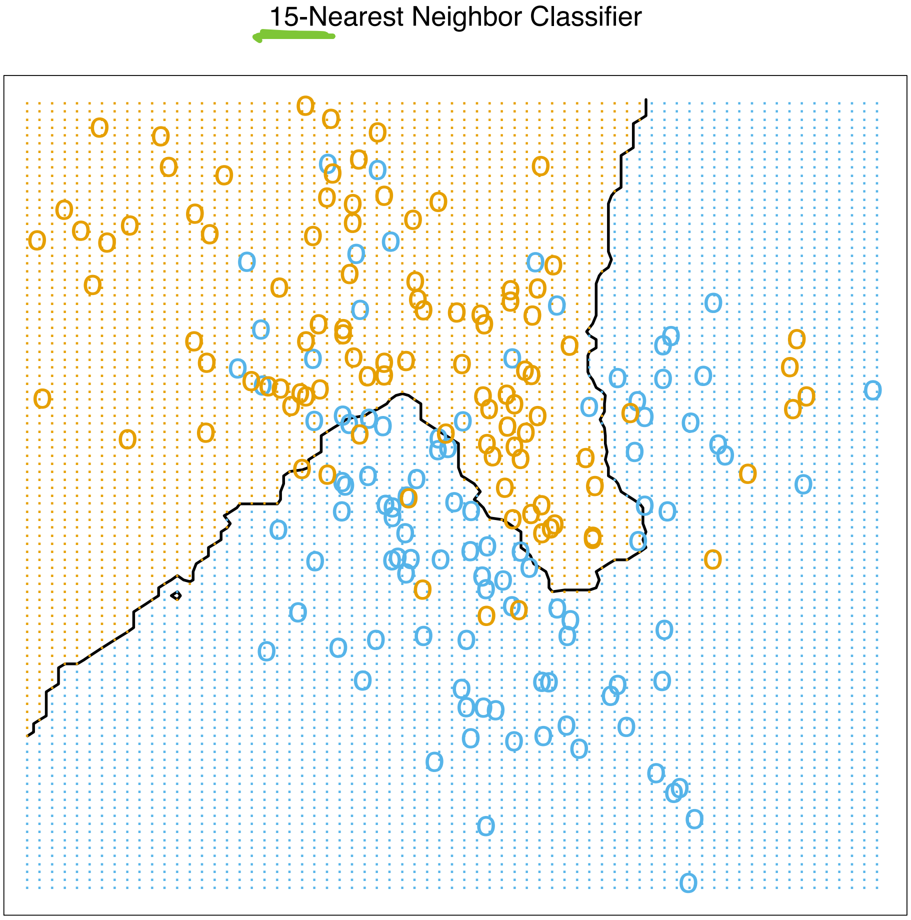


FIGURE 2.2. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (BLUE $= 0$, ORANGE $= 1$) *and then fit by* 15*-nearest-neighbor averaging as in (2.8). The predicted class is hence chosen by majority vote amongst the* 15*-nearest neighbors.*

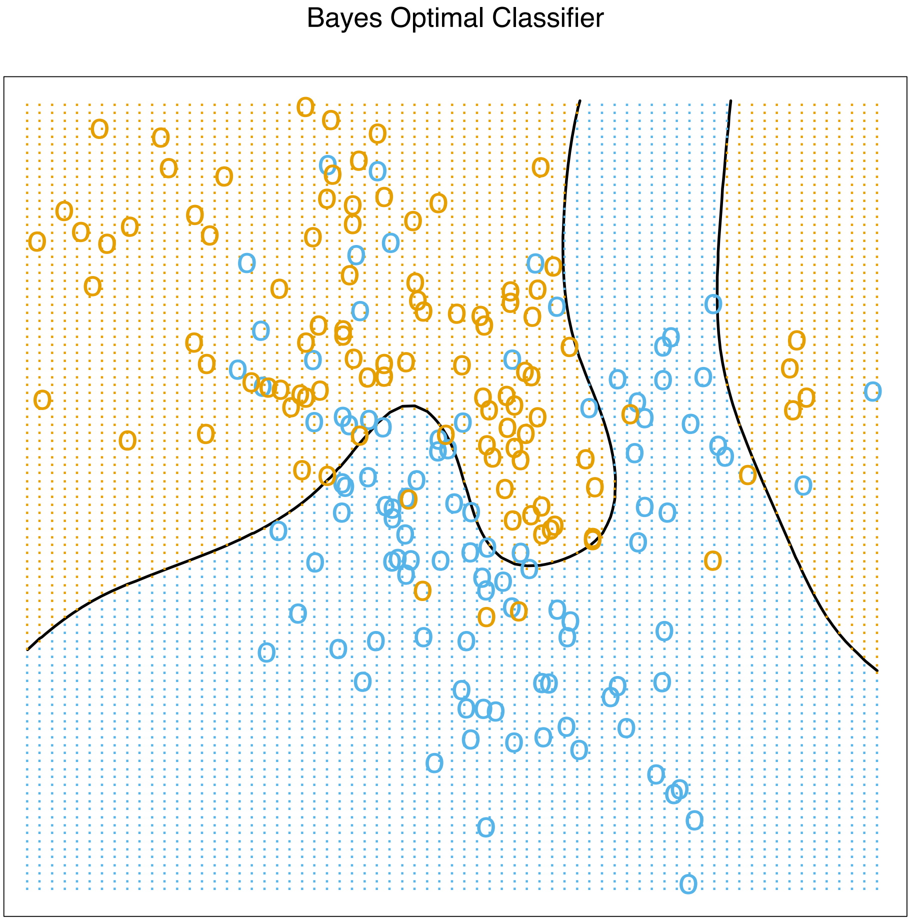


FIGURE 2.5. *The optimal Bayes decision boundary for the simulation example of Figures 2.1, 2.2 and 2.3. Since the generating density is known for each class, this boundary can be calculated exactly (Exercise 2.2).*

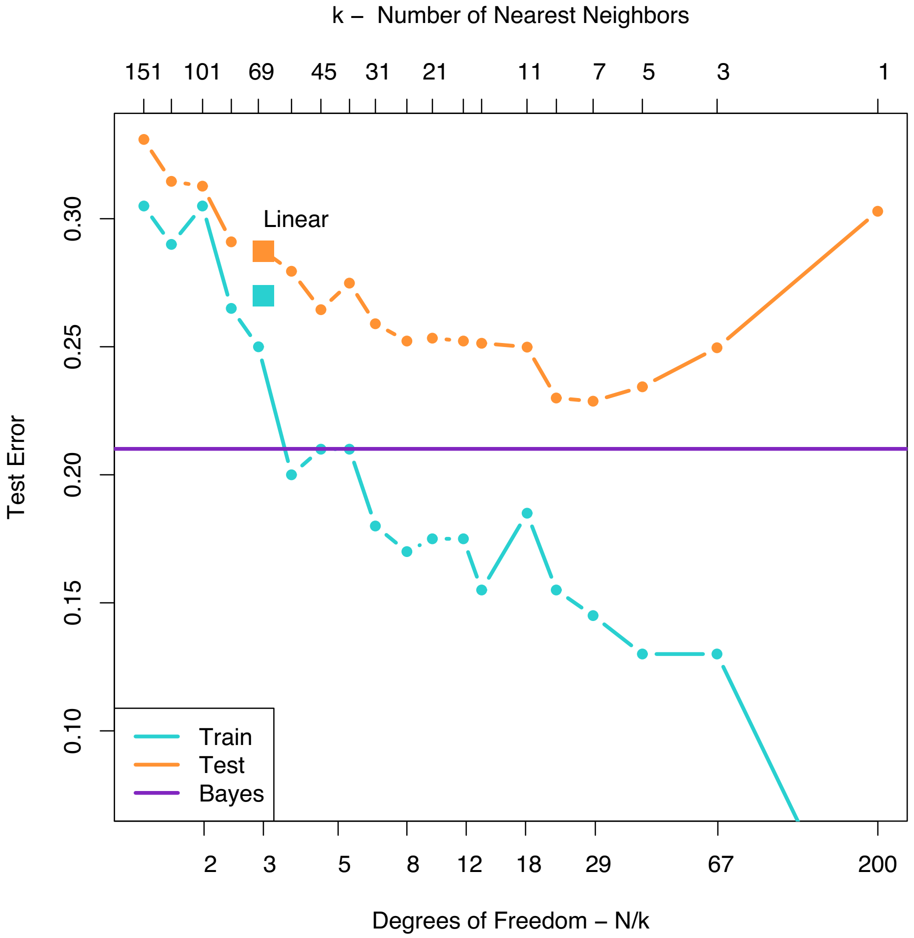


FIGURE 2.4. *Misclassification curves for the simulation example used in Figures 2.1, 2.2 and 2.3. A single training sample of size* 200 *was used, and a test sample of size* 10,000. *The orange curves are test and the blue are training error for k-nearest-neighbor classification. The results for linear regression are the bigger orange and blue squares at three degrees of freedom. The purple line is the optimal Bayes error rate.*