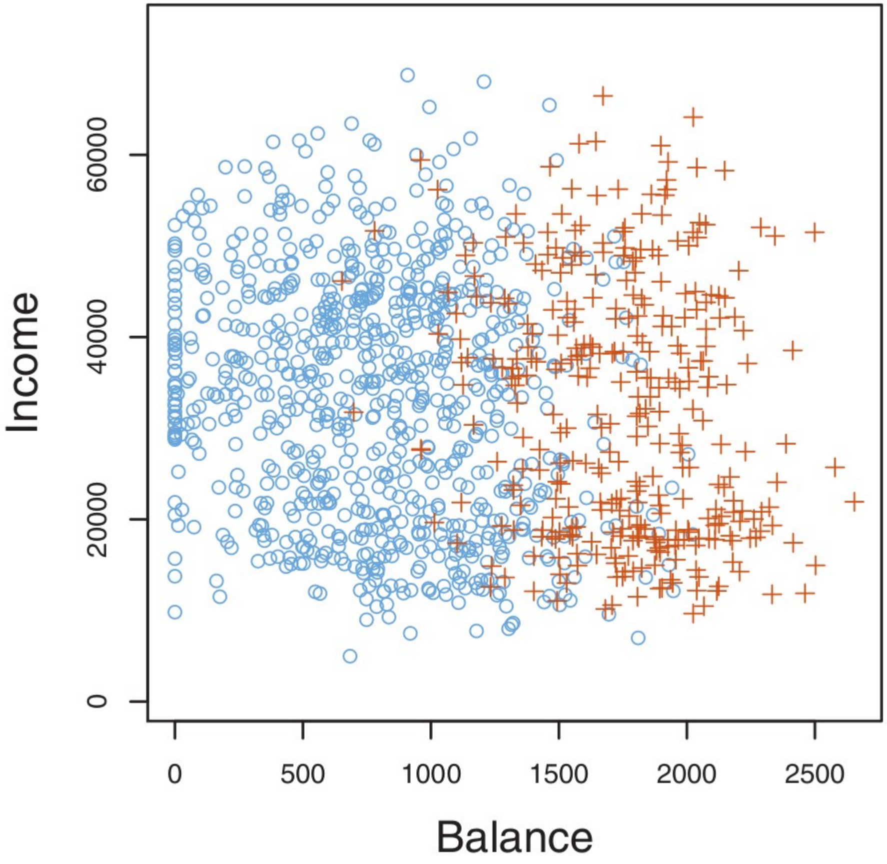
Introduction to Machine Learning

Lecture 3
Fundamentals II

Goker Erdogan
26 – 30 November 2018
Pontificia Universidad Javeriana

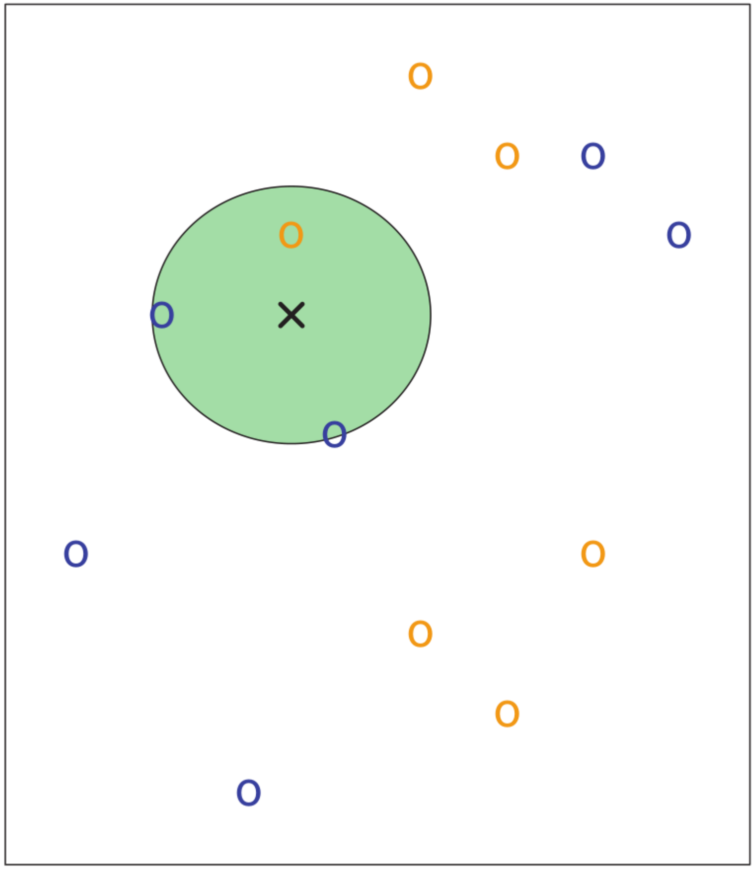
Setup

- Classification:
  - Given N samples (training set) of {x, c}, where c is one of K possible classes
  - Predict c for a new x
- Example: Predict if an individual will default on credit card payments[1]



K-nearest neighbor method

- A simple idea
- Given a test point x
  - Look at the training data and find K training points closest to it
  - Use these *K neighbors* to make a prediction



K-nearest neighbor example

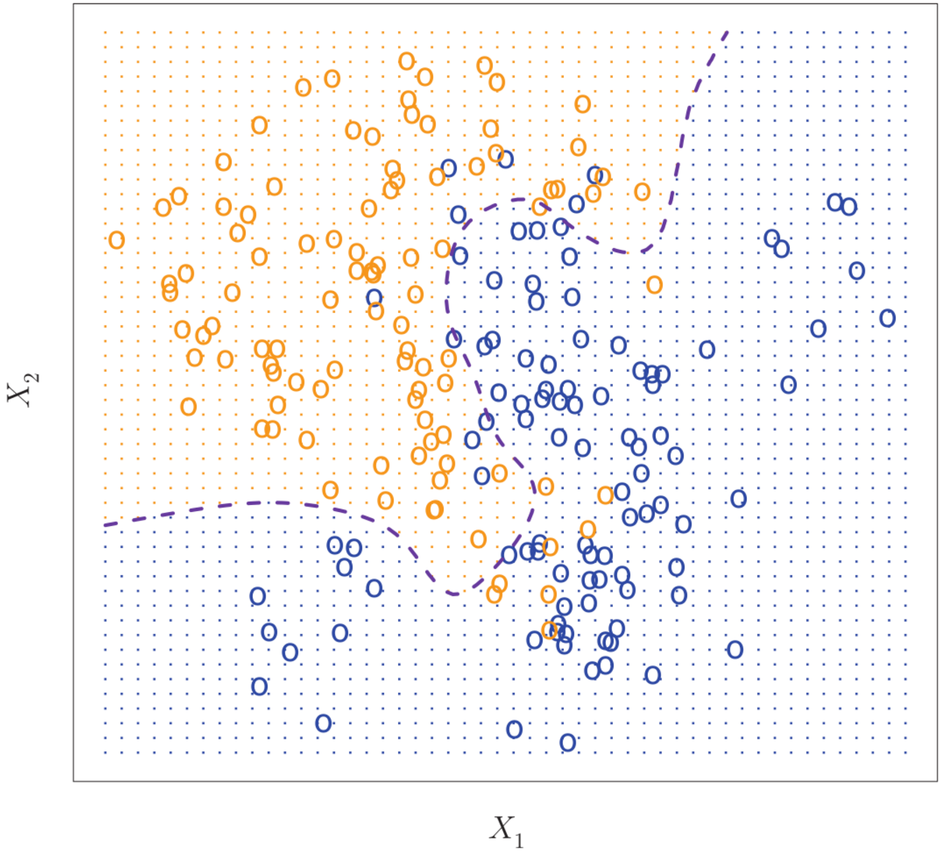
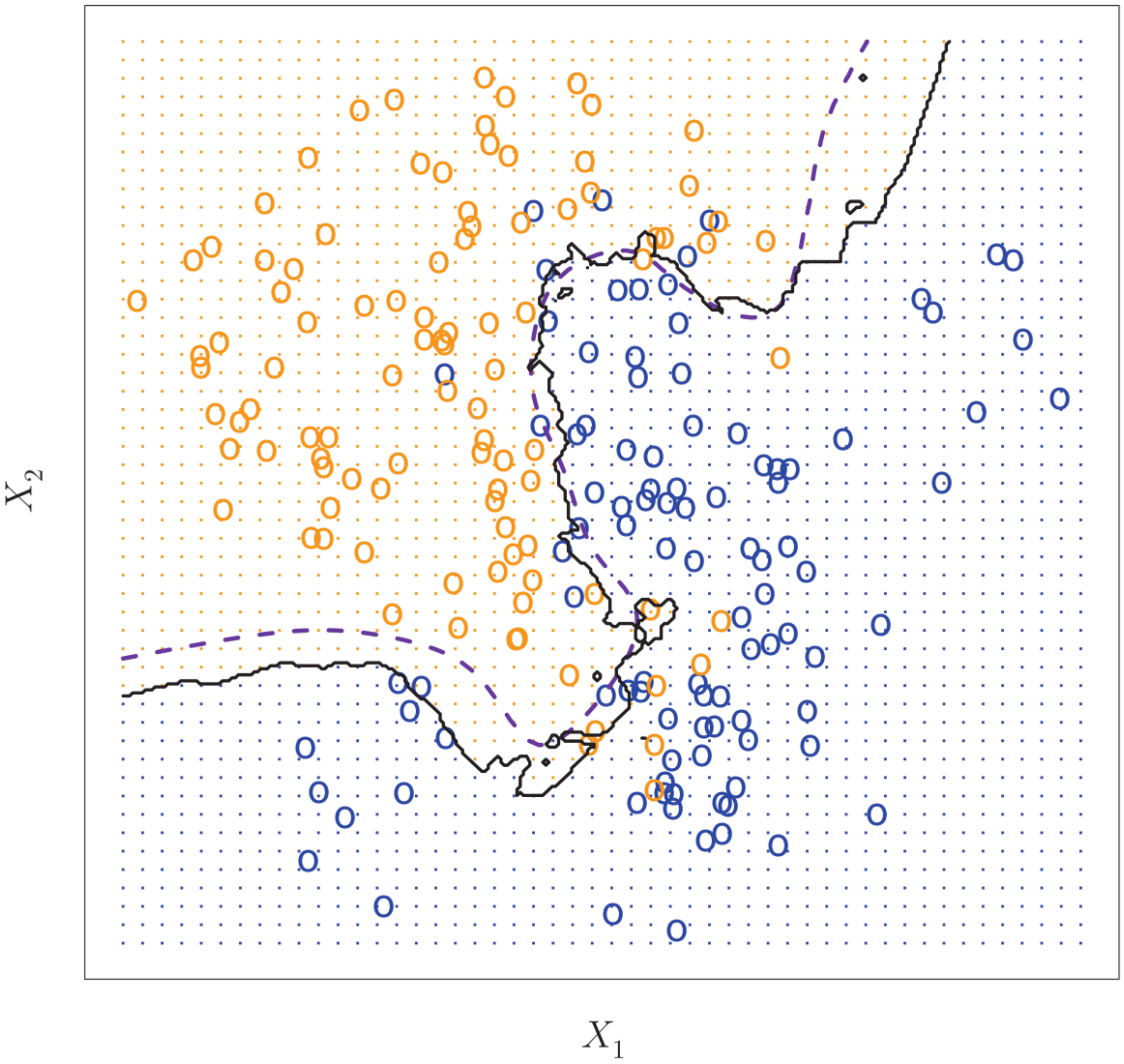
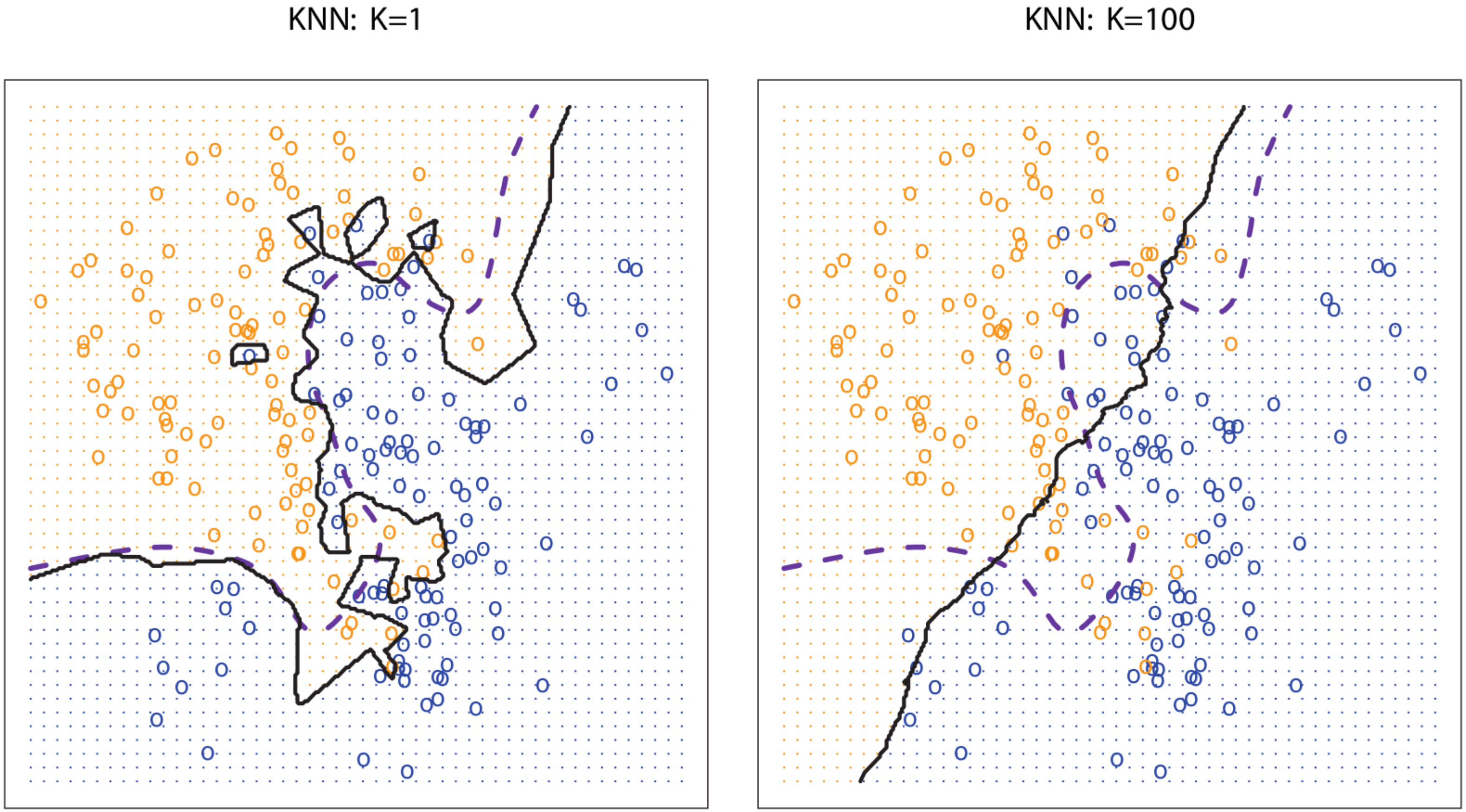
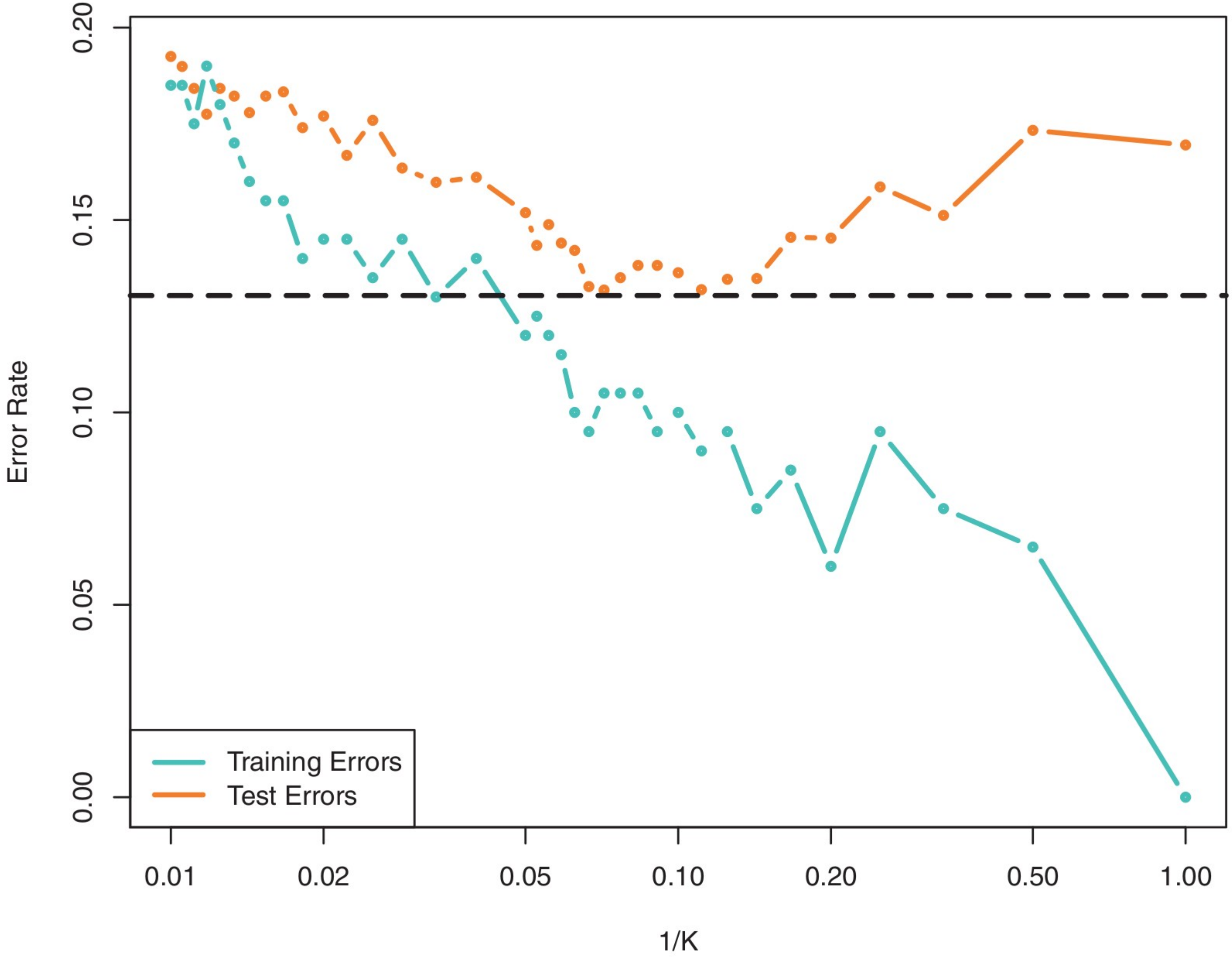


FIGURE 2.13. *A simulated data set consisting of* 100 *observations in each of two groups, indicated in blue and in orange. The purple dashed line represents the Bayes decision boundary. The orange background grid indicates the region in which a test observation will be assigned to the orange class, and the blue background grid indicates the region in which a test observation will be assigned to the blue class.*

KNN: K=10




KNN: K=1
KNN: K=100


Error Rate
0.00
0.05
0.10
0.15
0.20
1/K
0.01
0.02
0.05
0.10
0.20
0.50
1.00
Training Errors
Test Errors

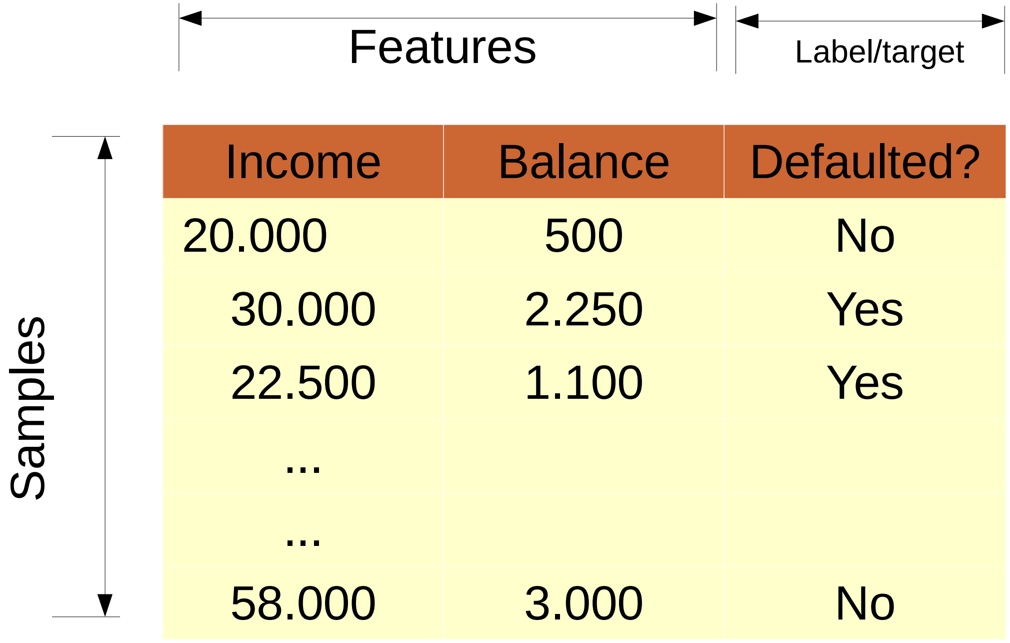
K-nearest neighbor contd.

- Note that we haven't learned any function x → c
  - We used data directly to make predictions
  - We have not assumed any form for the relation x → c
  - K-nearest neighbor is a *non-parametric* model
    - Compare this to the polynomial curve fitting example

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

- Non-parametric methods are appealing
  - Because they do not make assumptions that might later turn out to be wrong
  - In the limit (as N → ∞), K-nearest neighbor's error rate < 2*minimum possible error rate
- However,
  - You need to store the training set
  - Parametric methods need far less data to perform well

What happens in higher dimensions?



Features

Label/target

Samples

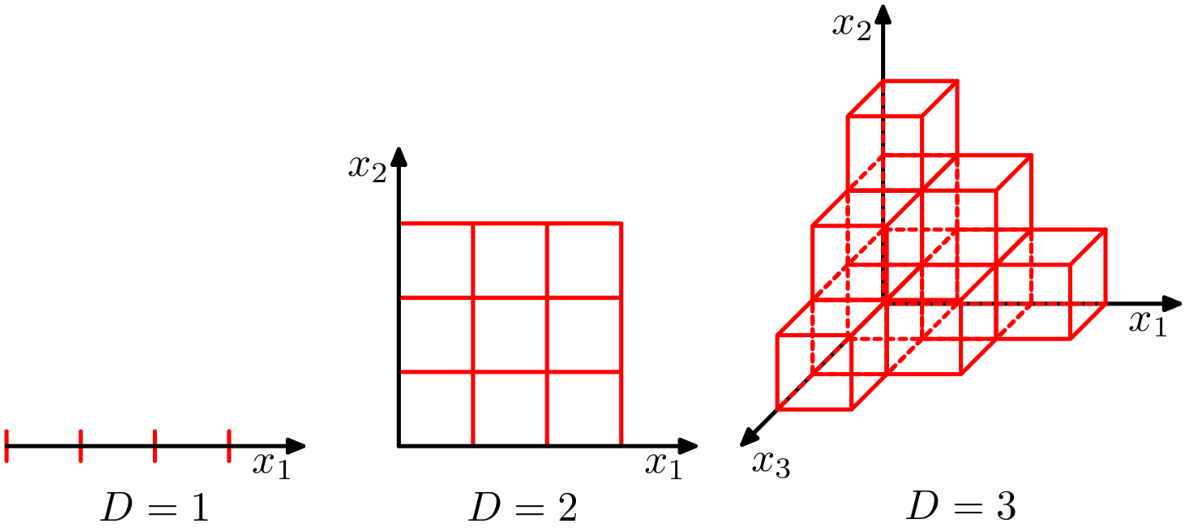
Income	Balance	Defaulted?
20.000	500	No
30.000	2.250	Yes
22.500	1.100	Yes
...		
...		
58.000	3.000	No

- What happens if there are 5 features/10 features/1000 features?
- Imagine a face classification problem
  - Each training sample is an image of size 100x100
  - We have 10.000 features!

K-nearest neighbor in higher dimensions

- Imagine you have 1000 training samples uniformly distributed in $[-1, 1]^D$
  - For D=1, there are 500 samples in [0, 1]
  - For D=2, there are 250 samples in $[0, 1]^2$
  - For D=4, there are 62 samples in $[0, 1]^3$
- As D increases, you need exponentially more samples to have the same density of training samples

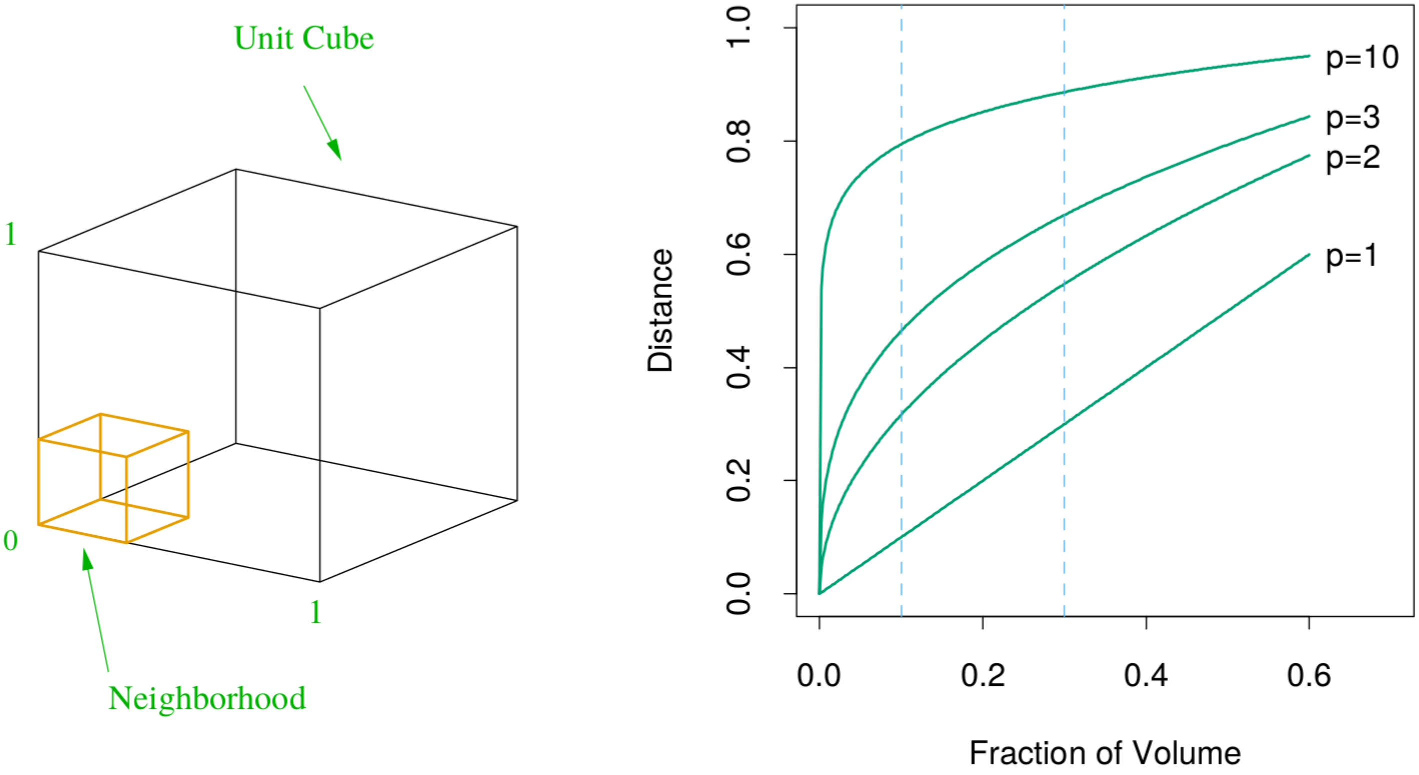
Figure 1.21 Illustration of the curse of dimensionality, showing how the number of regions of a regular grid grows exponentially with the dimensionality D of the space. For clarity, only a subset of the cubical regions are shown for $D = 3$.



[2]

K-nearest neighbor in higher dimensions

- Another way to look at it
  - Assume you want to pick a neighborhood such that you cover 10% of all training data

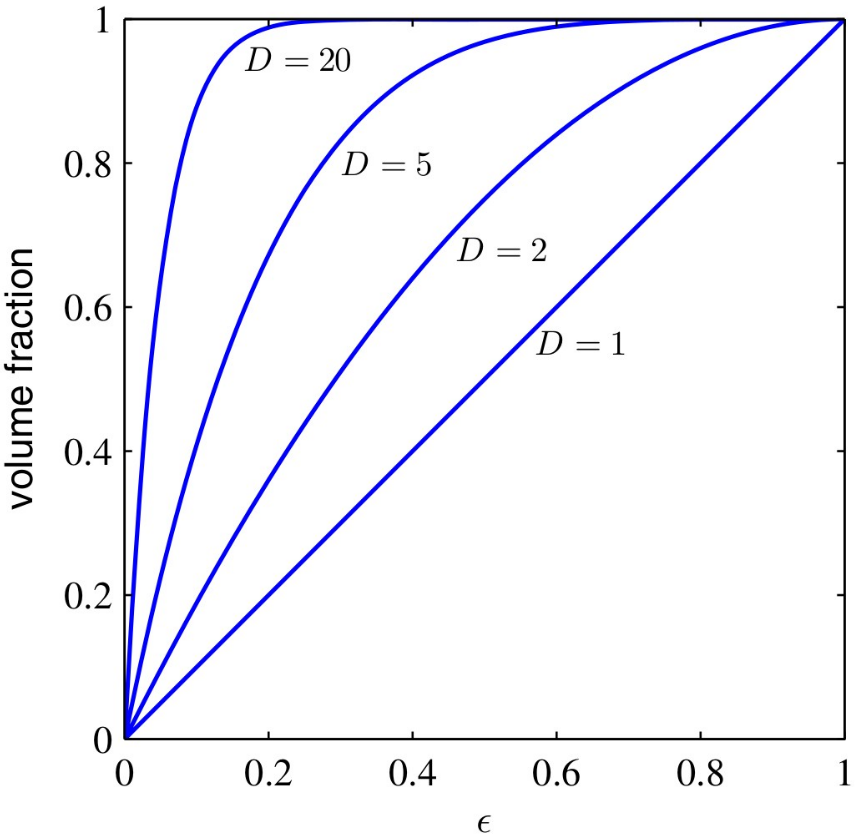


[3]

Curse of dimensionality

- This problem is known as *curse of dimensionality*
  - It is impossible to get enough samples to cover the input space in high dimensions
  - Neighborhoods are no longer *local*
  - Intuitions from lower dimensions rarely hold in higher dimensions

Figure 1.22 Plot of the fraction of the volume of a sphere lying in the range $r = 1-\epsilon$ to $r = 1$ for various values of the dimensionality D.



[2]

What can we do about it?

- Curse of dimensionality is usually not a problem because[2]
  1) Real data usually lies in a lower dimensional manifold in high dimensional input space
  2) Real data is smooth: small changes in input lead to small changes in output

- Parametric methods suffer less from the curse of dimensionality
  - Need less data
  - Outperforms non-parametric methods if assumptions are right
    - Non-parametric methods tend to have higher bias/variance

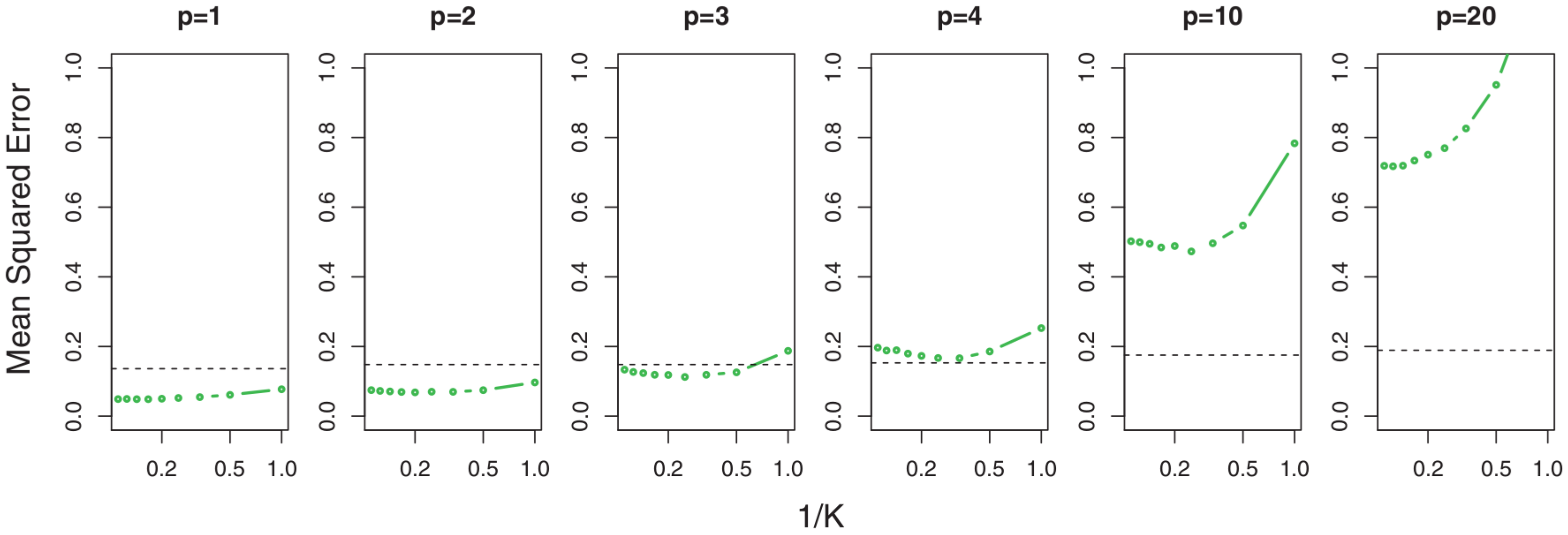


FIGURE 3.20. *Test MSE for linear regression (black dashed lines) and KNN (green curves) as the number of variables p increases. The true function is non–linear in the first variable, as in the lower panel in Figure 3.19, and does not depend on the additional variables. The performance of linear regression deteriorates slowly in the presence of these additional noise variables, whereas KNN's performance degrades much more quickly as p increases.*

[4]

No free lunch theorem

- There is no universally best learning method[5]
  - This is known as the *no free lunch theorem*
  - A set of assumptions that works well in one domain may work poorly in another
  - In other words, if a method performs well on a certain class of problems, it won’t perform well on others
- Therefore, it is crucial to understand the problem domain and pick models that make the appropriate assumptions

Summary

- K-nearest neighbor
  - Find the K nearest neighbors in training data to make a prediction
- Parametric vs non-parametric methods
  - Whether or not a method assumes a parametric form for x → c
  - Parametric methods usually need less data
- Curse of dimensionality
  - As the number of features increases, you need more and more data
- No free lunch theorem
  - There is no universally best method

References

[1] James, Witten, Hastie, and Tibshirani. An Introduction to Statistical Learning with Applications in R. Section 4.1.

[2] Bishop C. Pattern Recognition and Machine Learning. Section 1.4

[3] Hastie, Tibshirani, and Friedman. The Elements of Statistical Learning. Section 2.5

[4] Hastie, Tibshirani, and Friedman. The Elements of Statistical Learning. Section 3.5

[5] Murphy K. Machine Learning: A Probabilistic Perspective. Section 1.4.9