Lesson 09 Classification: k-NN

Lusine Zilfimian

April 20 (Monday), 2020

Contents

- Quiz

Contents

Contents

Contents

Contents

Last Lecture ReCap

- Why do we need the regularization?

Last Lecture ReCap

- Why do we need the regularization?
- What is the difference among Ridge, LASSO and EN?

Some examples of Distances

- The Euclidean distance

Some examples of Distances

- The Euclidean distance
- The Manhattan distance

Some examples of Distances

- The Euclidean distance
- The Manhattan distance
- The Minkowski distance (generalization)

Some examples of Distances

- Calculate the distance for these attributes

```
set.seed(2708)
x <- round(rnorm(3)); y <- round(rnorm(3))
(Points <- cbind(x,y))
```

```
##       x  y
## [1,]  0 -1
## [2,] -1  1
## [3,]  1  0
```

Distances: Properties

```r
stats::dist(Points, method = "euclidean", diag = T)
```

```
##          1        2        3
## 1 0.000000
## 2 2.236068 0.000000
## 3 1.414214 2.236068 0.000000
```

```r
stats::dist(Points, method = "manhattan", diag = T)
```

```
##   1 2 3
## 1 0
## 2 3 0
## 3 2 3 0
```

```r
stats::dist(Points, method = "minkowski", diag = T, p = 5)
```

Metrics

Measures that satisfy all three properties are known as metrics:

1. Positivity

Metrics

Measures that satisfy all three properties are known as metrics:

1. Positivity
2. Symmetry

Metrics

Measures that satisfy all three properties are known as metrics:

1. Positivity
2. Symmetry
3. Triangle Inequality

Metrics

Measures that satisfy all three properties are known as metrics:

1. Positivity
2. Symmetry
3. Triangle Inequality

- Non-metric Distances

Nearest neighbors

- If it walks like a duck, quacks like a duck, and looks like a duck, then it's probably a duck.

Nearest neighbors

- If it walks like a duck, quacks like a duck, and looks like a duck, then it's probably a duck.
- Given a test example, we compute its proximity to the rest of the data points in the training set.

Nearest neighbors

- If it walks like a duck, quacks like a duck, and looks like a duck, then it's probably a duck.
- Given a test example, we compute its proximity to the rest of the data points in the training set.
- The data point is classified based on the class labels of its neighbors.

More than one label

- Majority voting scheme

More than one label

- Majority voting scheme
- Averaging the target variable

Algorithm

- Computes the distance (or similarity) between each test point and all the training points

Algorithm

- Computes the distance (or similarity) between each test point and all the training points
- Determine its nearest-neighbor list (for chosen k)

Algorithm

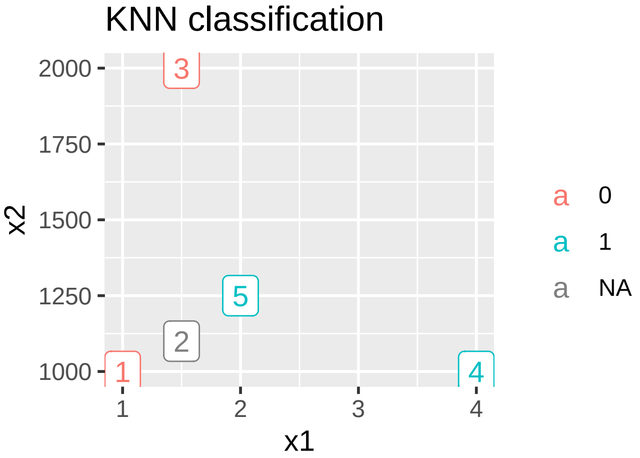
- Computes the distance (or similarity) between each test point and all the training points
- Determine its nearest-neighbor list (for chosen k)
- Test example is classified based on the majority class of its nearest neighbors

The problem with scales

```
x1 <- c(1, 1.5, 1.5, 4, 2); x2 <- c(1000, 1100, 2000, 1000, 12
y <- rbinom(n = 5, size = 1, prob = 0.6);
y.test = y; y.test[2] <- NA
(simpledata = data.frame(x1, x2, y))
```

```
##    x1   x2 y
## 1 1.0 1000 0
## 2 1.5 1100 0
## 3 1.5 2000 0
## 4 4.0 1000 1
## 5 2.0 1250 1
```

The problem with scales



The problem with scales

- Make the distance matrix and select distances for test point

```
(matd <- as.matrix(dist(simpledata[,1:2],
  method =  "euclidean")))
```

```
##           1         2         3         4        5
## 1    0.0000  100.0012 1000.0001    3.0000 250.0020
## 2  100.0012    0.0000  900.0000  100.0312 150.0008
## 3 1000.0001  900.0000    0.0000 1000.0031 750.0002
## 4    3.0000  100.0312 1000.0031    0.0000 250.0080
## 5  250.0020  150.0008  750.0002  250.0080   0.0000
```

```
sort(matd[,2])
```

```
##        2        1        4        5        3
##   0.0000 100.0012 100.0312 150.0008 900.0000
```

Standardization

```
sc <- as.data.frame(scale(simpledata[,1:2]))
d <- as.matrix(dist(sc))
sort(d[,2])
```

```
##         2         1         5         3         4
## 0.0000000 0.4881755 0.5558209 2.1392221 2.1452160
```

Characteristics and requirements

- Every neighbor should have the same impact on the classification

Characteristics and requirements

- Every neighbor should have the same impact on the classification
- Do not require model building (data-driven)

Characteristics and requirements

- Every neighbor should have the same impact on the classification
- Do not require model building (data-driven)
- Classifying a test example can be quite expensive

Characteristics and requirements

- Every neighbor should have the same impact on the classification
- Do not require model building (data-driven)
- Classifying a test example can be quite expensive
- Choice of k, distance metric can be difficult

Characteristics and requirements

- Every neighbor should have the same impact on the classification
- Do not require model building (data-driven)
- Classifying a test example can be quite expensive
- Choice of k, distance metric can be difficult
- Is non-parametric approach

Characteristics and requirements

- Every neighbor should have the same impact on the classification
- Do not require model building (data-driven)
- Classifying a test example can be quite expensive
- Choice of k, distance metric can be difficult
- Is non-parametric approach
- It is supervised learning algorithm

Characteristics and requirements

- Every neighbor should have the same impact on the classification
- Do not require model building (data-driven)
- Classifying a test example can be quite expensive
- Choice of k, distance metric can be difficult
- Is non-parametric approach
- It is supervised learning algorithm
- Can be applied to both classification and regression

K-NN for regression

- Parametric Regression assumes some forms

K-NN for regression

- Parametric Regression assumes some forms
- Non - parametric Regression does not have assumption

K-NN for regression

- Parametric Regression assumes some forms
- Non - parametric Regression does not have assumption
- K-NN can be used for both classification and regression problems.

K-NN for regression

- Parametric Regression assumes some forms
- Non - parametric Regression does not have assumption
- K-NN can be used for both classification and regression problems.
- Classifiers try to predict a category.

K-NN for regression

- Parametric Regression assumes some forms
- Non - parametric Regression does not have assumption
- K-NN can be used for both classification and regression problems.
- Classifiers try to predict a category.
- Regression try to predict a real number.

Algorithm of K-NN regression

1. The distance between the new point and each training point is calculated.

Algorithm of K-NN regression

1. The distance between the new point and each training point is calculated.
2. The closest k data points are selected (based on the distance).

Algorithm of K-NN regression

1. The distance between the new point and each training point is calculated.
2. The closest k data points are selected (based on the distance).
3. The **average** of these data points is the final prediction for the new point.

How to chose K?

- The optimal value for K will depend on the bias-variance tradeoff:

How to chose K?

- The optimal value for K will depend on the bias-variance tradeoff:
- A small value for K provides the most flexible fit: low bias but high variance (prediction in a given region is entirely dependent on just one observation).

How to chose K?

- The optimal value for K will depend on the bias-variance tradeoff:
- A small value for K provides the most flexible fit: low bias but high variance (prediction in a given region is entirely dependent on just one observation).
- A larger values of K provide a smoother and less variable fit (the prediction in a region is an average of several points, and so changing one observation has a smaller effect)

How to chose K?

- Cross-validation

How to chose K?

- Cross-validation
- Choose it based on the error calculation for our train and validation set

Example

Let's make predictions for a large number of possible values of range, for different values of k. Note that 11 is the total number of observations in this training dataset.

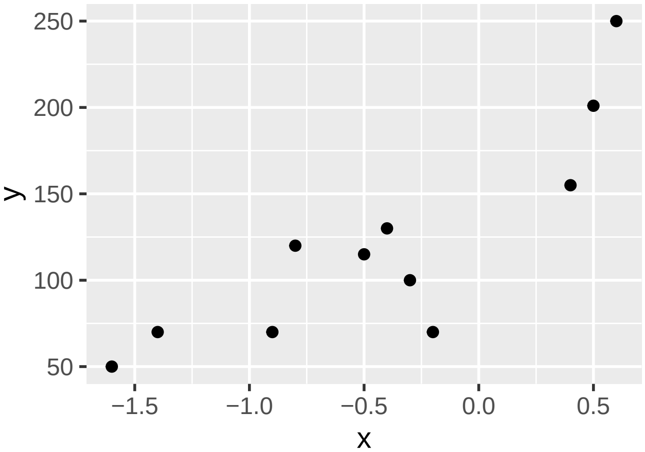
```
x <- c(-1.6, -1.4, -0.9, -0.8, -0.5, -0.4, -0.3, -0.2,
  0.4, 0.5, 0.6)
y <- c(50, 70, 70, 120, 115, 130, 100, 70, 155, 201, 250)
simpledata = data.frame(x,y)

testrange <- data.frame(test_range = seq(min(x), max(x), by =
length(testrange$test_range)
```

```
## [1] 23
```

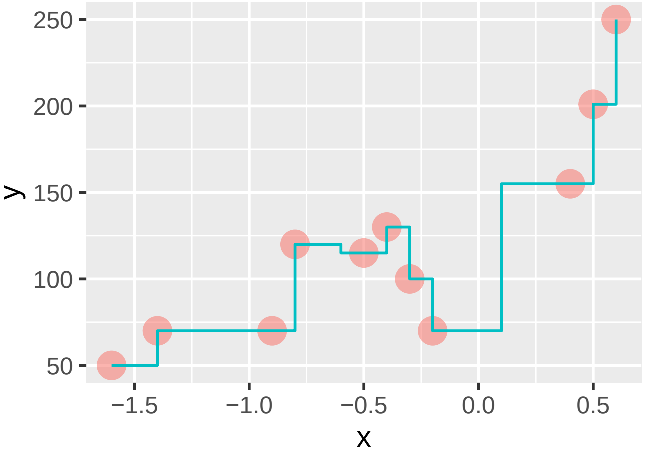
Example

```
ggplot(data = simpledata, aes(x = x, y = y)) +
  geom_point()
```

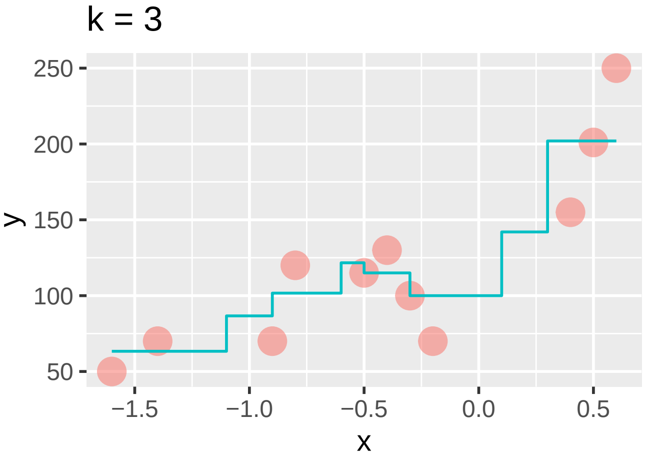


Example

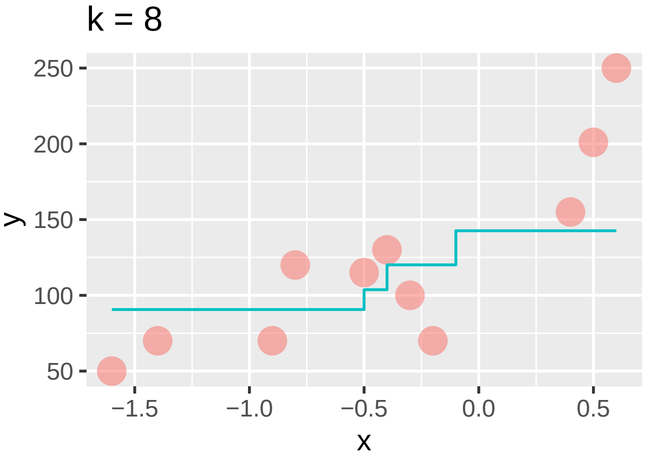
- K = 1 is clearly overfitting, as k = 1 is a very complex, highly variable model.



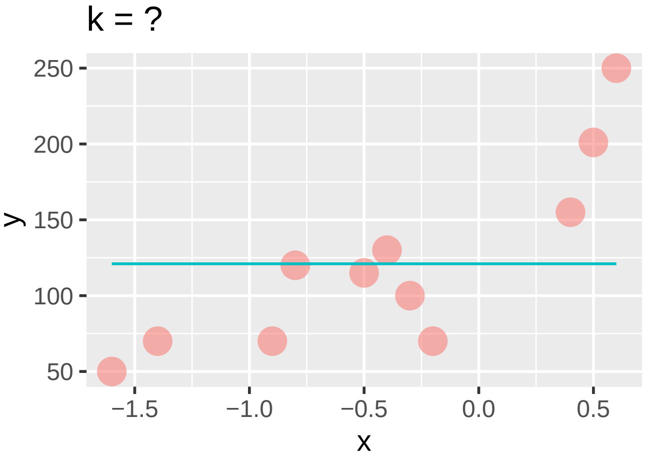
Example



Example

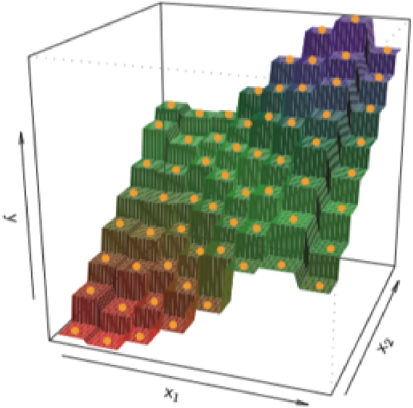


Example



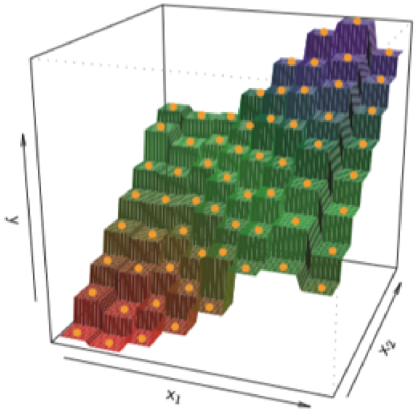
3-D case

- For two predictors predictors with $K = 1$ we have 3D step-wise a stepwise constant graph.



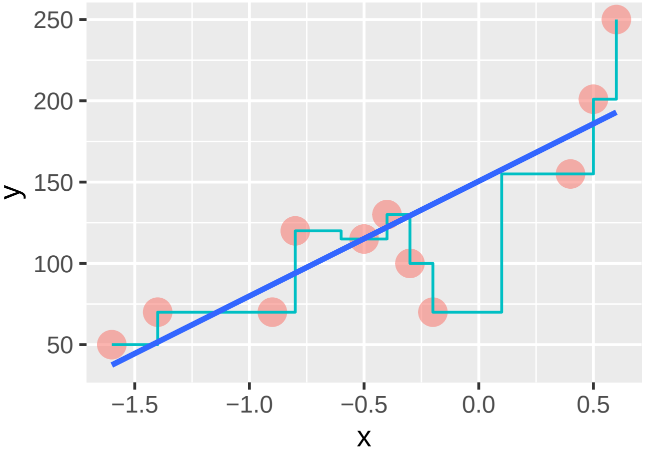
3-D case

- For two predictors predictors with $K = 1$ we have 3D step-wise a stepwise constant graph.

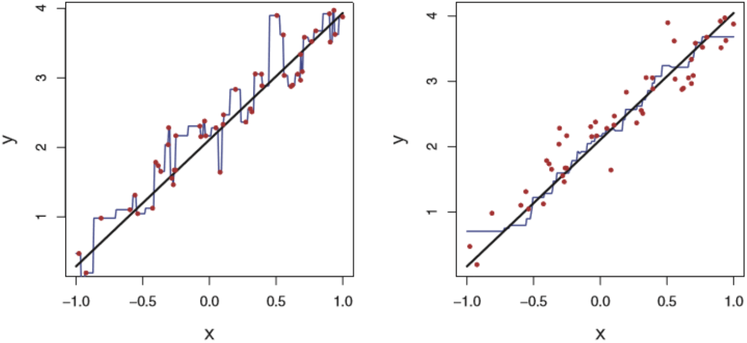


- Graph is from Introduction to Statistical Learning

LR vs KNN



LR vs KNN



- Graph is from Introduction to Statistical Learning

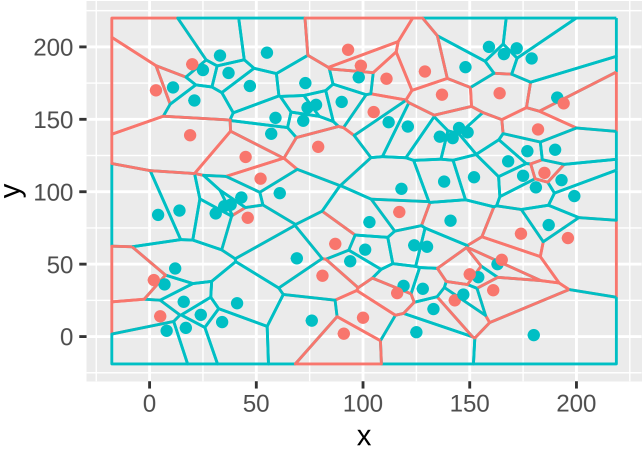
LR vs KNN

- The parametric approach will outperform the nonparametric approach if the parametric form that has been selected is close to the true form of f.

LR vs KNN

- The parametric approach will outperform the nonparametric approach if the parametric form that has been selected is close to the true form of f.
- If the true relationship is linear, it is hard for a non-parametric approach to compete with linear regression:

Voronoi



Ideas for project

- Decision boundaries for K-NN

Ideas for project

- Decision boundaries for K-NN
- Comparison of LR and K-NN