Machine Learning: Business Applications

Francisco Rosales Marticorena, PhD.
frosales@esan.edu.pe

04.04.19 – 30.05.19

ESAN Graduate School of Business

Datos Generales del Curso

Asignatura:	Machine Learning: Aplicaciones en los Negocios
Área académica:	Programa de Especialización para Ejecutivos
Año y semestre:	2019 – I
Profesor:	Francisco Rosales Marticorena, PhD.
	Mail: frosales@esan.edu.pe
	Teléfono: (511) 317-7200 / 444340

Sumilla

- Este curso presenta métodos de machine learning con énfasis en problemas de clasificación y regresión de aprendizaje supervisado.
- El curso contiene sesiones de fundamentos matemáticos; sesiones de desarrollo metodológico y de aplicaciones.
- Se usará el software R para la resolución de casos de estudio.

Objetivos de la Asignatura

- Mejorar las capacidades cuantitativas de analistas y gerentes para interpretar los resultados de métodos que aprenden de los datos.
- Utilizar adecuadamente conceptos matemáticos básicos involucrados en los métodos de aprendizaje supervisado de Machine Learning.
- Utilizar el software R y sus librerías especializadas para es desarrollo de implementaciones propias o de terceros.

Programación de Contenidos

1. Estadística básica:
    - Sesión 1: Introducción
    - Sesión 2: Software R
    - Sesión 3: Regresión lineal
2. Métodos Lineales:
    - Sesión 4: Modelos de Clasificación
    - Sesión 5: Métodos de Resampleo
    - Sesión 6: Regularización
    - Sesión 7: Reducción dimensional
    - Sesión 8: Taller
3. Métodos No-Lineales:
    - Sesión 9: Splines
    - Sesión 10: GAMs
    - Sesión 11, 12: Árboles de decisión
    - Sesión 13, 14: Support Vector Machines
    - Sesión 15: Evaluación Final

Metodología

Las exposiciones del profesor se complementarán con actividades que harán los alumnos en el salón de clase, y fuera de él:

- Participar en clase.
- Leer la bibliografía indicada en el programa.
- Hacer las tareas.
- Rendir las evaluaciones programadas.

Evaluación

$$\text{Nota Final} = \text{seis tareas (60\%)} \quad + \quad \text{un examen final (40\%)}.$$

- Las tareas se podrán realizar de manera individual o en parejas.
- El examen final es individual.
- El examen final es obligatorio, y se rendirá el día 23.05.19.

Fuentes de Información

- [EH06] Everitt, B. and T. Hothorn. A handbook of statistical analyses using R. Chapman & Hall/CRC, 2006.
- [JO13] James, G. et. al. (2013). An Introduction to Statistical Learning with Applications in R. Springer Series in Statistics.
- [HT09] Hastie, T., R. Tibshirani and J. Friedman (2009). The Elements of Statistical Learning: Data Mining, Inference and Prediction. Springer Series in Statistics.
- [RS05] Ramsay, J.O. and B.W. Silverman (2006). Functional Data Analysis. Springer Series in Statistics.
- [RO03] Ruppert, D., M. Wand and R. Carrol (2003). Semiparametric Regression. Cambridge Series in Statistical and Probabilistic Mathematics.
- [W06] Wood, S. (2006). Generalized Additive Models: An Introduction with R. Chapman & Hall/CRC.

Docente

Educación:

- Doctor. Matemáticas y Ciencia Comp. Universidad de Göttingen.
- Magister. Matemáticas Ap. y Estadística. SUNY Stony Brook.
- Magister. Matemáticas. PUCP.
- Licenciado y Bachiller. Economía. UP.

Experiencia:

- 2019: Profesor Investigador. TI. U. ESAN.
- 2018: Gerente. Financial Services Office. EY Perú.
- 2017: Profesor investigador. Finanzas. U. del Pacífico.
- 2011–2016: Investigador asociado. IMS. U. Göttingen.
- 2005–2008: Científico Investigador. CGIAR. CIP.

Asistentes

- Expectativa: analista, gerente, etc.
- Sectores: banca, seguros, reguladores, etc.
- Lenguajes: Python, R, C++, Matlab, etc.

Materiales

Desde un celular:

Scan me

Desde un navegador:

`https://github.com/LFRM/Lectures`

Introduction

Overview

- Machine Learning: is a toolbox to understand data using statistics
- Objectives
  1. Prediction / to predict something
  2. Inference / to explain something
- Problems
  1. Supervised learning: “input $\Rightarrow$ output” structure.
  2. Unsupervised learning: only “input” structure.
- Methods
  1. Regression
  2. Classification
  3. Clustering

Basics

The general model follows:

$$Y = f(X) + \epsilon, \quad X = \{X_1, \dots, X_p\}$$

where

- Y is called response / dependent variable
- X are called features / independent variables / predictors
- f is a non-random function
- ϵ is a random error term, independent of X, and with zero mean.

In this course: we find f to predict / explain.

Basics

Usual Steps:

1. We observe predictors X and response Y.
2. We characterize their relationship

$$Y = f(X) + \epsilon. \tag{1}$$

3. We estimate $\hat{f}$ “somehow”.
4. We use $\hat{f}$ in X to make prediction $\hat{Y}$

$$\hat{Y} = \hat{f}(X). \tag{2}$$

Focus depends on Goals:

- Prediction: we care mostly about $\hat{Y}$.
- Inference: we care mostly about $\hat{f}$.

Estimation Error

Proposition 1

$$\mathbb{E}[(Y-\hat{Y})^2] = \underbrace{\mathbb{E}[(f(X)-\hat{f}(X))^2]}_{\textit{Reducible}} + \underbrace{\mathrm{Var}[\epsilon]}_{\textit{Irreducible}} \quad . \tag{3}$$

Proof. Trivial. Direct substitution from (1) and (2). ■

Interpretation of proposition 1:

- The magnitude of the estimation error has a reducible and an irreducible component.
- We cannot reduce the estimation error below Var[ϵ].

Estimation of f: Parametric vs. Non-Parametric

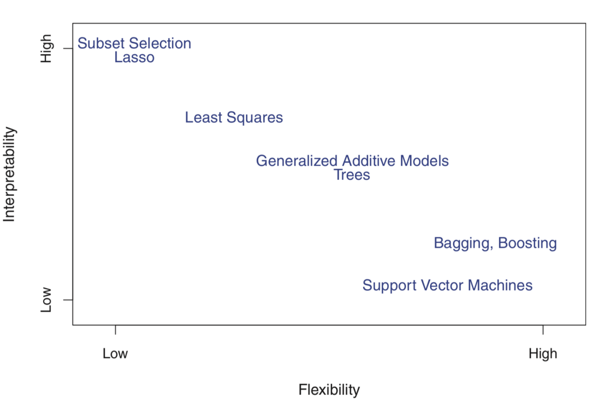
Parametric Methods:

- Impose rigid structure on f, e.g. f is linear.
- Trade-off: easy to interpret vs. bad accuracy.

Non-Parametric Methods:

- Impose flexible structure on f, e.g. f is a piecewise polynomial.
- Trade-off: difficult to interpret vs. good accuracy.

Estimation of f: Parametric vs. Non-Parametric



Source: [JO13]

Estimation of f: Regression vs. Classification

- Discrete response $\Rightarrow$ classification / cont. response $\Rightarrow$ regression.
- Specific methods for regression or classification.
- Some methods deal with both, e.g. K-nearest neighbors, boosting.

Estimation of f: Assessing Model Accuracy

Definition 1.1 (Mean Squared Error)

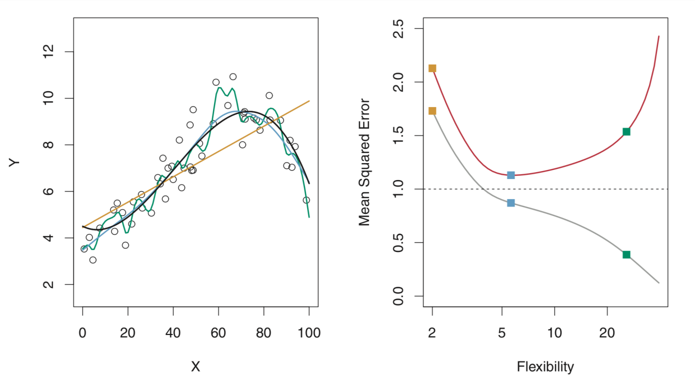
The Mean Squared Error (MSE) is defined as

$$\begin{aligned} MSE &:= Ave\{(y_i - \hat{f}(x_i))^2\} \\ &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2. \end{aligned} \tag{4}$$

We call it "train" MSE if we compute it with training data (X, Y) and "test" MSE if we compute it with "test" data (X_0, Y_0).

- Train MSE can be reduced arbitrarily (overfitting).
- Test MSE cannot be reduced arbitrarily.
- Test MSE is used for model selection.

Estimation of f: Assessing Model Accuracy



Left: data (circles), true function (black), linear fit (orange), spline fit 1 (blue), spline fit 2 (green). Right: train MSE (gray), test MSE (red).

Source: [JO13]

Estimation of f: Bias - Variance Trade-off

Proposition 2

Given x_0, the expected test MSE can be decomposed into the sum of three quantities: the variance of $\hat{f}(x_0)$, the squared bias of $\hat{f}(x_0)$ and the variance of the error term.

Proof. From proposition 1 we known that

$$\begin{aligned}\mathbb{E}[(y_0 - \hat{f}(x_0))^2] &= \mathbb{E}[(f(x_0) - \hat{f}(x_0))^2] + \text{Var}[\epsilon] \\ &= \mathbb{E}[(\{f(x_0) - \mathbb{E}[\hat{f}(x_0)]\} \\ &\quad -\{\hat{f}(x_0) - \mathbb{E}[\hat{f}(x_0)]\})^2] + \text{Var}[\epsilon],\end{aligned}$$

Thus,

$$\begin{aligned}\mathbb{E}[(y_0 - \hat{f}(x_0))^2] &= \underbrace{\mathbb{E}[(f(x_0) - \mathbb{E}[\hat{f}(x_0)])^2]}_{\text{Squared Bias of } \hat{f}} + \underbrace{\mathbb{E}[(\hat{f}(x_0) - \mathbb{E}[\hat{f}(x_0)])^2]}_{\text{Variance of } \hat{f}} \\ &+ \underbrace{\text{Var}[\epsilon]}_{\text{Variance of error}}\end{aligned}$$

Estimation of f: Bias - Variance Trade-off

Interpretation of proposition 2:

- The reducible error is a mixture of bias and variance.
- To minimize the expected test MSE we need a method that minimizes both Bias and Variance.

Methods:

- More flexible methods: $\downarrow$ bias and $\uparrow$ variance.
- Less flexible methods: $\uparrow$ bias and $\downarrow$ variance.
- The adequacy of the method depends on the data.

Estimation of f: Classification Setting

Definition 1.2 (Error Rate)

The error rate (ER) is the proportion of mistakes that the classifier makes in a given data set

$$\begin{aligned} ER: &= Ave\{\mathcal{I}_{\hat{y}_i \neq y_i}\} \\ &= \frac{1}{n}\sum_{i=1}^{n} \mathcal{I}_{\hat{y}_i \neq y_i}, \end{aligned} \tag{5}$$

Definition 1.3 (Bayes Classifier)

The Bayes classifier minimizes (5), by selecting the value of j such that

$$max_j \mathbb{P}[Y = j | X = x_0], \tag{6}$$

for each x in the data set.

Estimation of f: Classification Setting

Note that the Bayes classifier:

- Is theoretical, i.e. the conditional probability in (32) is unknown.
- Provides a lower bound on ER,

$$\text{ER} \geq 1 - \mathbb{E}\left[\max_j \mathbb{P}[Y = j | X = x_0]\right]$$

Estimation of f: Classification Setting

Definition 1.4 (K-Nearest Neighbors)

K-Nearest Neighbors (KNN) is a classification method that requires a positive integer K. For a test observation x_0, it identifies K points near x_0, called $\mathcal{N}_0$, and estimates the conditional probability for class j as:

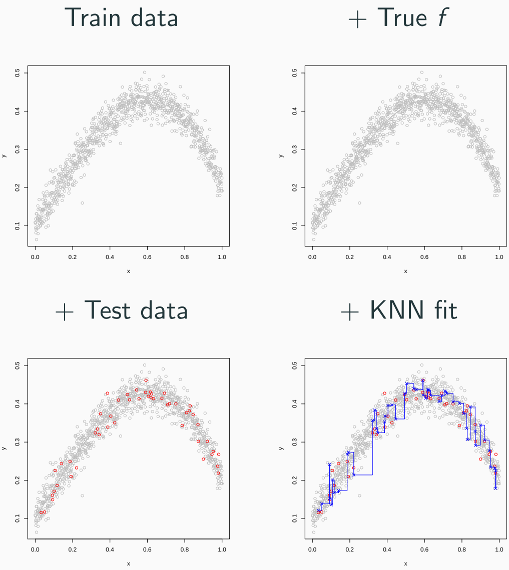
$$\hat{\mathbb{P}}[Y = j | X = x_0] = \frac{1}{K} \sum_{i \in \mathcal{N}_0} \mathcal{I}_{y_i = j},$$

i.e. as the fraction of the points in $\mathcal{N}_0$ whose response values equal j.

Note that:

- K has a strong effect on the classification obtained by KNN.
- Small K: $\downarrow$ bias and $\uparrow$ variance.
- Large K: $\uparrow$ bias and $\downarrow$ variance.

Estimation of f: Classification Setting



Train data

+ True f

+ Test data

+ KNN fit

R Software

The R Project for Statistical Computing

- This is an extremely short introduction to R.
- Click here to download the software.
- For more information look at my lecture notes in Applied Statistics.

For now, lets run R online:

Scan me

Basic Commands

- The call for the `funcname` function is `funcname(arg1,arg2)`, where `arg1` and `arg2` are args. of the function.
  1. What does `c()` evaluated in the arguments 1 and 2 does?
  2. Does the order of the arguments matter?
- We can store values using “<-” or “=”.

  ```
  > x <- c(1,2)
  > y = c(4,-1)
  > x + y
  [1]  5 1
  ```

  Let `z1 = 1000` and `z2 = rep(1000, 3)`. Compute

  1. `x + z1`
  2. `x + z2`

Basic Commands

- To ask for help on function `funcname` use `?funcname`
- To list all the objects in the workspace use `ls()`.
- To delete object `obj` from the workspace use `rm(obj)`.
- To delete every object use `rm(list = ls())`

Basic Commands

- To create a matrix use `matrix`

```
> x <- matrix(data = 1:4, nrow = 2, ncol = 2)
> x
          [,1]      [,2]
[1,]        1         3
[2,]        2         4
```

- Compute

1. `z <- x ^ 2`
2. `r <- matrix(1, 2, 4)`
3. `q <- z %*% r`

Basic Commands

- To create a realization of a normal random variable use `rnorm`. To specigy the mean, use `mean` and, to specify the standard deviation, use `sd`

  ```
  > x <- rnorm(10000)
  > y <- rnorm(10000, mean = 50, sd = 0.1)
  ```

- To compute the correlation between two r.v.s use `cor`.

  ```
  > z <- x + y
  > cor(x, z)
  [1]  0.9951
  ```

- To reproduce the exact same random number use `set.seed()` with an arbitrar integer argument, e.g. `set.seed(123)`

  ```
  > set.seed(123)
  > rnorm(5)
  [1]  -0.56047565 -0.23017749  1.55870831  ...
  ```

Graphics

- There are various functions for plotting. See ?plot

```
> x = rnorm(100) + 1:100
> y = rnorm(100) + seq(-1, -100, length = 100)
> plot(x, y)
> plot(x, y, xlab = "this is my x-axis",
  ylab = "this is my y lab", main = "Plot x vs y")
```

- To save the output use pdf(), or jpeg()

```
> pdf("myfgure.pdf")
> plot(x, y, color = 2, lwd = 3)
> dev.off()
null device
          1
```

Graphics

Let $f : \mathbb{R}^2 \to \mathbb{R}$,

- To plot the contour of f use `contour` or `image`.

```
> x = 1:10
> y = x
> f = outer( x, y, function (x, y) cos(y) / (1 + x ^ 2) )
> contour(x, y, f)
> contour(x, y, f, nlevels = 45)
> fa = ( f - t(f) ) / 2
> contour( x, y, fa, nlevels = 15)
> image(x, y, fa)
```

- To plot f in three dimensions use `persp`.

```
> persp(x, y, fa)
> persp(x, y, fa, theta = 30, phi = 20)
> persp(x, y, fa, theta = 30, phi = 40)
```

Indexing Matrix Data

Extracting part of a data set can be done in different ways:

```
> A = t(matrix(1:16, 4, 4))
> A
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
[4,]   13   14   15   16
```

Compute

1. `A[2,3]`
2. `A[1,]`
3. `A[1:3, c(2,4)]`
4. `A[,-4]`
5. `dim(A)`

Loading Data

- To import data use read.table() or read.csv(); and to visualize the imported data use fix().

```
> Auto = read.csv("Auto.csv", header = T,  sep=" ")
> fix(Auto)
> dim(Auto)
[1] 392 9
```

- To list the variable names in the data set use names().

```
> names(Auto)
[1] "mpg"     "cylinders"     "displacement" "horsepower"
[5] "weight"  "acceleration"  "year"         "origin"
[9] "name"
```

- To export data use write.table()

Additional Graphical and Numerical Summaries

- To access a variable `cylinders` in data frame `Auto`, we use

```
> Auto$cylinders
```

- To avoid using the dollar symbol, we can simply attach the data, so that all the variables in the data frame are added to the workspace.

```
> attach(Auto)
> plot(cylinders, mpg)
```

- Plot the following:
  1. mpg vs. cylinders using red circles
  2. mpg vs. cylinders using a red circles, with axis labels "mpg" and "cylinders" resp.

Additional Graphical and Numerical Summaries

- To plot a histogram use `hist()`

  ```
  > hist(mpg)
  > hist(mpg, col = 2)
  ```

- To create a scatterplot matrix use `pairs()`

  ```
  > pairs(Auto)
  > pairs(~ mpg + displacement + horsepower
    + weight + acceleration, Auto)
  ```

- To print a summary of a given variable or data frame, use `summary()`

  ```
  > summary(mpg)
  > summary(Auto)
  ```

Linear Regression

Motivation

What?

Supervised learning method for continuous response.

Why?

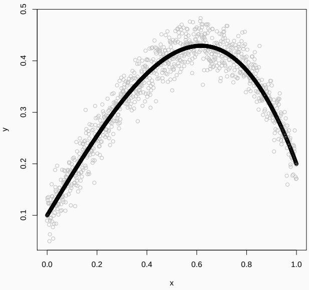
- Well document starting point.
- Recall: not flexible, tipically ↑ bias and ↓ variance.
- Fancy approaches are extensions/generalizations of linear regression.

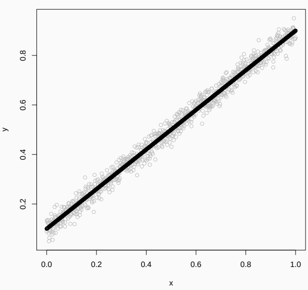
Simple Linear Regression

Model:

$$Y = f(X) + \epsilon, \quad f(X) = \beta_0 + \beta_1 X, \tag{7}$$

where ϵ is a centered random noise, uncorrelated to X.


Example 1


Example 2

Estimation

- Idea: estimate $f(X)$ via

$$\hat{f}(X) = \hat{\beta}_0 + \hat{\beta}_1 X, \tag{8}$$

e.g. identifying $\hat{\beta}_0$, $\hat{\beta}_1$ that fulfills the least squares criteria.

- Least Squares Criteria:

$$\begin{aligned} (\hat{\beta}_0, \hat{\beta}_1) &= \text{argmin}_{\hat{\beta}_0, \hat{\beta}_1} \text{RSS} \qquad (9) \\ \text{RSS} &:= \sum_{i=1}^{n} e_i^2, \quad e_i := y_i - \hat{y}_i, \qquad (10) \end{aligned}$$

where RSS denotes the "Residual Sum of Squares" (RSS) of the model.

Estimation

- FOC:

$$\frac{\partial RSS}{\partial \hat{\beta}_0} = \hat{\beta}_0 + \hat{\beta}_1 \bar{x} - \bar{y} = 0, \tag{11}$$

$$\frac{\partial RSS}{\partial \hat{\beta}_1} = \hat{\beta}_0 \sum_{i=1}^{n} x_i + \hat{\beta}_1 \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i y_i = 0. \tag{12}$$

  with solution

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}, \tag{13}$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}. \tag{14}$$

- RSS is convex, thus the SOC is fullfilled.

Example: Sample Mean Estimator

Recall:

- Let $Z \sim (\mu, \sigma)$, where the mean μ and s.d. σ are unknown.
- Sample mean estimator: Collect n random samples of Z, and estimate μ by $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} z_i$. Repeat the previous m times with different samples of the same size: $\{\hat{\mu}^{(1)}, \hat{\mu}^{(2)}, \ldots, \hat{\mu}^{(m)}\}$.
- Unbiasedness: We say $\hat{\mu}$ is an unbiased estimator of μ, since

$$\frac{1}{m}\sum_{k=1}^{m} \hat{\mu}^{(k)} \to \mu \quad \text{as} \quad k \to \infty.$$

- Variance: defined as

$$\text{SE}(\hat{\mu})^2 = \frac{\sigma^2}{n},$$

measures the precision of the sample mean estimator.

Bias in Simple Linear Regression

- β_0 and β_1 are r.v.s, and given some data we estimate them by $\hat{\beta}_0$ and $\hat{\beta}_1$, but the estimators vary according to the random sample.
- Unbiasedness: if we consider m random samples, we obtain $\hat{\beta}_0^{(k)}$ and $\hat{\beta}_1^{(k)}$, for $k = 1, \ldots, m$. It can be shown that

  $$\frac{1}{m}\sum_{k=1}^{m}\hat{\beta}_0^{(k)} \to \beta_0 \quad \text{and} \quad \frac{1}{m}\sum_{k=1}^{m}\hat{\beta}_1^{(k)} \to \beta_1 \quad \text{as} \quad k \to \infty,$$

  thus $\hat{f}(X)$ is an unbiased estimator of $f(X)$, that is

  $$\frac{1}{m}\sum_{k=1}^{m}\hat{f}(X)^{(k)} \to f(X) \quad \text{as} \quad k \to \infty,$$

  where $f(X)$ is the "population line" and $\hat{f}(X)^{(k)}$ is a "sample estimate of the line" corresponding to the least squares criteria.

Variance

- Variance: The precision of the estimators follow:

$$
\begin{aligned}
\mathrm{SE}[\hat{\beta}_0]^2 &= \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \right], && (15) \\
\mathrm{SE}[\hat{\beta}_1]^2 &= \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}, && (16)
\end{aligned}
$$

  where we have used $\mathrm{Var}[\epsilon_i] = \sigma^2$ and $\mathrm{Cov}(\epsilon_i, \epsilon_j) = 0$, for $i \neq j$.

- Note that σ is not known, and it is estimated by

$$\hat{\sigma} = \sqrt{\frac{RSS}{n-2}}, \qquad (17)$$

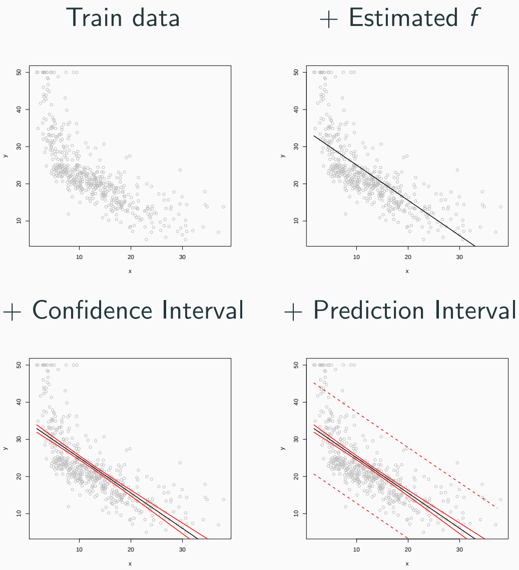
  sometimes called the "residual sum of errors" (RSE).

Confidence Bands

- Is a range of values that contain the true unknown value of the parameter with certain probability.
- Example: a 95% confidence band says that with 95% probability, the band contains the true value of the parameter
- In linear regression, if we add the assumption that ϵ is normally distributed, the bands for $\hat{\beta}_i$ look approx. like this

$$[\hat{\beta}_i - 2 \times \text{ SE}(\hat{\beta}_i), \hat{\beta}_i + 2 \times \text{SE}(\hat{\beta}_i)], \quad i = 0, 1. \tag{18}$$

Confidence Bands: Boston Example



Train data

+ Estimated f

+ Confidence Interval

+ Prediction Interval

Hypothesis testing

- Consider the hypothesis "there is no relation between X and Y".

$$\begin{aligned} H_0 &: \beta_1 = 0 \\ H_a &: \beta_1 \neq 0 \end{aligned}$$

- We test H_0 by computing the quantity

$$t = \frac{\hat{\beta}_1 - 0}{\mathrm{SE}[\hat{\beta}_1]}, \tag{19}$$

which is distributed t with $n-2$ degrees of freedom (t_{n-2} in short), and is called t-statistic.

- We reject H_0 if the t-statistic is "large" $\Leftrightarrow$ if the p-value is "small", where "small" means, e.g., less than < 0.05.

Hypothesis Testing

Remark 3.1

To see that $t \sim t_{n-2}$ in (19), recall that a r.v. $X = \frac{Z}{\sqrt{V/\nu}}$ is distributed t_ν if $Z \sim \mathcal{N}(0,1)$, and $V \sim \chi^2_\nu$, where χ^2_ν denotes the chi-squared distribution with ν degrees of freedom. The denominator of (19) reads

$$SE[\hat{\beta}_1] = \sqrt{\frac{RSS/\sigma^2}{(n-2)}}\sqrt{\frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}, \tag{20}$$

where $\frac{RSS}{\sigma^2} \sim \chi^2_{n-2}$, while the numerator follows

$$(\hat{\beta}_1 - 0) \Bigg/ \sqrt{\frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2}} \sim \mathcal{N}\left(0,1\right). \tag{21}$$

Writing (21) over (20) we obtain the desired expression.

Model Accuracy

- Residual Standard Error (RSE).

$$\text{RSE} := \sqrt{\frac{\text{RSS}}{n-2}} = \sqrt{\frac{\sum_{i=1}^{n} e_i^2}{n-2}} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-2}}.$$

  Problem: it is not clear what is a “good” RSE.

- R-Squared (R^2)

$$\begin{aligned} R^2 &= \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}, \qquad (22) \\ \text{TSS} &= \sum_{i=1}^{n}(y_i - \bar{y})^2. \end{aligned}$$

  Interpretation: It is between 0 and 1. It indicates the proportion of the variability of Y that can be explained by X. Further, it can be shown that $R^2 = \text{Cor(X,Y)}^2$, by plugging-in (13) and (14) in (22).

Multiple Linear Regression

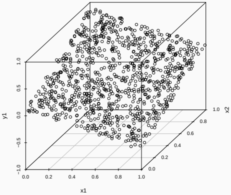
Model:

$$Y = f(X) + \epsilon, \quad f(X) = \beta_0 + \beta_1 X + \cdots + \beta_p X, \tag{23}$$

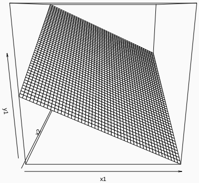
where ϵ is a centered random noise, uncorrelated to X.

Multiple Linear Regression

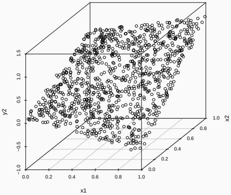
Data 1



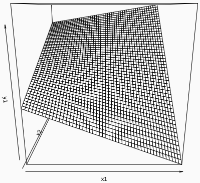
True Function 1



Data 2



True Function 2



Hypothesis Testing

Is there is at least one X_j that explains Y?

$$\begin{aligned} H_0 &: \quad \beta_1 = \beta_2 = \cdots = \beta_p \\ H_a &: \quad \text{at least one } \beta_j \text{ is non-zero} \end{aligned}$$

Compute the F-statistic

$$F = \frac{(TSS - RSS)/p}{RSS/(n-p-1)}$$

Recall that the r.v. $X = \frac{U_1/d_1}{U_2/d_2}$ is dist. F_{d_1,d_2}, for $U_1 \sim \chi^2_{d_1}$, $U_2 \sim \chi^2_{d_2}$. If the p-value is small (e.g. < 0.05) we reject H_0. This means that there is at least one X_j that explains Y.

Variable Selection

How to select a good subset of X_j's out of all predictors?
Idea: try all possible models. Problem: need to evaluate 2^p models. If $p = 30$, this is more than a million models.

- Forward selection: Start with a model that only uses the intercept to predict Y ("the best model with zero variables") and add one variable to the model at a time. To select "the best model with one variable", evaluate all of them and select the one with the smallest RSS. Repeat the idea to select "the best model with $2, 3, \ldots k$ variables". Repeat until some stopping criteria is reached.
- Backward selection: Start with a model that has all variables. Remove one variable at a time selecting the one that has the largest p-value. Re-run the model and repeat the process until some stopping criteria is reached.

Note that backward selection cannot be done if $p > n$. The stopping criteria can be related to some target p-value on the models in the model.

Model Accuracy

Advantages and disadvantages:

- RSS: non scaled.
- R-squared: scaled in $[0, 1]$. In fact it can be shown that $R^2 = \text{Cor}(Y, \hat{Y})^2$. Problem: increases wrt p.

Predictors

We can compute $\hat{y}_i = \hat{f}(x_i)$, as a predictor of

$$y_i = f(x_i) + \epsilon, \quad f(x_i) = \beta_0 + \sum_{i=1}^{p} \beta_i x_i$$

- Confidence intervals: compares $\hat{f}(X)$ with $f(X)$, i.e. includes only reducible errors.
- Prediction intervals: compares $\hat{f}(X)$ with $f(X) + \epsilon$, i.e. includes reducible and irreducible errors.

Other Considerations

- Qualitative predictors. An m-level predictor that induces m sub-models requires $m - 1$ indicator variables.
- Interaction between predictors. If of the form $x_i \times x_j$, we have no longer a linear model.

Potential Problems

- The true relation $f(X)$ is non-linear. See pattern in error (e_i) plots, e.g. not centered.
- Serial correlation between error terms. See persistence in e_i plots.
- Non-constant variance of errors. See $\hat{y}_i$ vs e_i plot.
- Outliers. See $\hat{Y}$ vs Y plot.

Potential Problems

- High leverage points. For simple linear regression, check

$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{j=1}^{n}(x_j - \bar{x})^2}, \quad h_i \in [0, 1]. \tag{24}$$

  called "leverage statistic". $h_i \approx 1$, means high leverage at i.

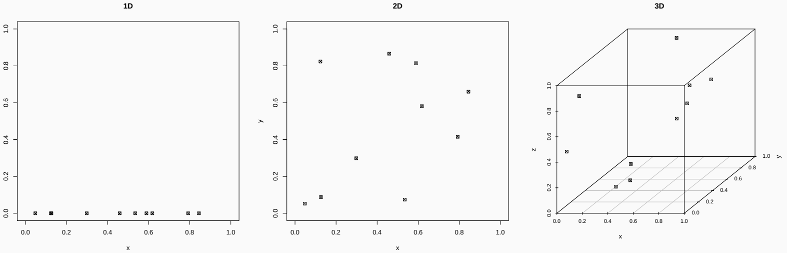
- Collinearity. Check the variance inflation factor

$$\mathrm{VIF}(\hat{\beta}_j) = \frac{1}{1 - R^2_{X_j|X_{-j}}}, \quad \mathrm{VIF}(\hat{\beta}_j) \in [1, \infty]. \tag{25}$$

  where $R^2_{X_j|X_{-j}}$ is the R-squared of the regression of X_j against all other predictors except X_j.

Comparison with KNN

- If p is small and non-linear relation, KNN is better.
- If p is large, KNN has problems due to the “curse of dimensionality”.
- The plot below shows 10 realizations of $x, y, z \sim \mathcal{U}(0, 1)$ plotted the line, de square and the cube. See how the separation of the points increase with the dimensionality.



Homework

Due 4.18.19 - 19:30.

1. Chapter 2: Exercises 2, 4, 7, 9 from [JO13].
2. Chapter 3: Exercises 3, 4, 9, 14 from [JO13].

Classification Models

Introduction

- Setup: Qualitative response, i.e. Y takes a specific set of categories

$$Y \in \{a, b, c\}.$$

- Cannot use Linear regression.
  - Coding: need to assign numerical values to the categories, e.g. $a = 1, b = 2, c = 3$, which can be arbitrary. In linear regression different coding for identical X can generate different predictions.
  - Interpretability: in linear regression we estimate $\hat{f} : \mathbb{R}^p \to \mathbb{R}$, meaning that we are mapping onto the whole real line, and not only onto $\{1, 2, 3\}$. This means we can predict 1.5, which is meaningless.
  - In this section assume Y coded as 0/1, unless otherwise stated.

Simple Logistic Regression

- Simple logistic regression model:

$$\mathbb{P}(Y = 1|X) = p(X), \quad p(X) := \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}, \tag{26}$$

  where $p : \mathbb{R} \to [0, 1]$ is the logistic function.

- In this context, the odds are usually reported

$$\frac{p(X)}{1 - p(X)} = e^{\beta_0+\beta_1 X} \quad \to \quad \log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X. \tag{27}$$

- Interpretation of coefficients: Note that β_1 is the marginal contribution of X to the log-odds (not the response).

Estimation & Prediction

- The estimation is done via maximization of a likelihood function

$$\begin{aligned}(\hat{\beta}_0, \hat{\beta}_1) &= \text{argmax}_{\beta_0,\beta_1}\ell(\beta_0, \beta_1) && (28)\\ \ell(\beta_0, \beta_1) &:= \Pi_{i:y_i=1}(p(x_i))\Pi_{j:y_j=0}(1-p(x_j)), && (29)\end{aligned}$$

  which is a convex function, thus only the FOC needs to be verified.

- Once the parameters are estimated, the prediction is computed as

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+\hat{\beta}_0+\hat{\beta}_1 X},$$

  and the classification follows a rule of the form:

$$\hat{y}_i = \begin{cases} 1 & \text{if} \quad \hat{p}(X) > 0.5 \\ 0 & \text{if} \quad \hat{p}(X) \leq 0.5 \end{cases}$$

Multiple Logistic Regression

- Multiple logistic regression model:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p, \quad (30)$$

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_p X_p}} \quad (31)$$

  where $p : \mathbb{R} \to [0,1]$ is the logistic function.

- Estimation and prediction: analog to the simple logistic model case.
- Logistic regression for > 2 response classes are not used often. For these case we consider linear discriminant analysis.

Bayesian Classifier

Recall: Bayes Theorem

$$p_k(X) = \frac{\pi_k f_k(x)}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(x)}, \tag{32}$$

where:

- $\pi_k = \mathbb{P}(Y = k)$: prior probability that a randomly chosen observation comes from the k-th class.
- $f_k(x) = \mathbb{P}(X = x | Y = k)$: density function of X for an observation that comes from the k-th class.
- $p_k(X) = \mathbb{P}(Y = k | X = x)$: posterior probability that a randomly chosen observation comes from the k-th class

Idea: π_k is easy, $f_k(x)$ is difficult. With reasonable $\hat{f}(x)$, can approximate the Bayes classifier (classifier with the smallest error rate).

Bayesian Classifier (p=1)

- Let $f_k(x) \sim \mathcal{N}(\mu_k, \sigma)$, thus:

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu_k)^2}{2\sigma^2}\right\} \tag{33}$$

- Pluggin (33) in (32), we obtain

$$p_k(X) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu_k)^2}{2\sigma^2}\right\}}{\sum_{\ell=1}^{K} \pi_\ell \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu_\ell)^2}{2\sigma^2}\right\}}, \tag{34}$$

Bayesian Classifier ($p = 1$)

- Decision boundary between classes i and j:

$$0 = \log(\pi_i) - \log(\pi_j) - \frac{(\mu_i^2 - \mu_j^2)}{2\sigma^2} + \frac{\mu_i - \mu_j}{\sigma^2} x. \tag{35}$$

- Example: let $K = 2$, $\pi_1 = \pi_2$, and $\sigma = 1$. Note that at the boundary $p_1(X) = p_2(X)$, thus

$$x = \frac{\mu_1 + \mu_2}{2},$$

  is the boundary separating the two categories in variable X.

Linear Discriminant Analysis ($p = 1$)

- In practice, need to estimate σ, $\mu_1, \ldots, \mu_k$, and $\pi_1, \ldots, \pi_k$.

$$\hat{\pi}_k = \frac{n_k}{n} \tag{36}$$

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i, \tag{37}$$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \mu_k)^2 \tag{38}$$

- LDA uses estimators of these quantities and plug them in (35) to select the boundaries by pairs.

Linear Discriminant Analysis ($p > 1$)

x