Machine Learning: Business Applications

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

04.04.19 – 23.05.19

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: Sofware 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

Nota Final $=$ seis tareas (60%) $+$ 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. ESAN.
- 2018: Gerente de Servicios Financieros. EY Perú.
- 2017–2018: Profesor investigador. Finanzas. UP.
- 2011–2016: Investigador asociado. IMS. U. Göttingen.
- 2005–2008: Científico. CGIAR. CIP.

Asistentes

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

Materiales

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

Introduction

Overview

- What is it? a toolbox to understand data using statistics
- Understand?
  1. Prediction / to predict something
  2. Inference / to explain something
- Type of problems?
  1. Supervised learning: "input $\Rightarrow$ output" structure.
  2. Unsupervised learning: only "input" structure.
- Type of methods?
  1. Regression
  2. Classification
  3. Clustering

Example 1.1 (Direct Marketing Campaign)

Consider a client who wants to implement a direct marketing campaign, e.g. email, phone, and for that needs you to tell him/her who to focus the campaign on, i.e. a list of who to write/call.

Note that:

- We mostly care about the list.
- Idea: scan the population and mark each person as "Yes" or "No".

Examples

Example 1.2 (Advertising)

Consider a client that sells a some product and to do so spends money on adds in TV, radio and Youtube. Your client thinks investing in Youtube advertising makes no sense. He/she asks you to verify if this is the case. You have monthly data on how much is sold on that product and how much was spent for that period in each media type. Typical questions:

1. *Which media contributes more to sells?*
2. *Which media generates the biggest boost in sells?*
3. *How much increase in sales is related to an increase in TV advertising?*

Basics

The general model follows:

$$\boldsymbol{Y} = f(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \ldots, \boldsymbol{x}_p) + \epsilon,$$

where

- $\boldsymbol{Y}$ is called response / dependent variable
- $\boldsymbol{x}_i$ are called features / independent variables / predictors
- f is a non-random function
- ϵ is a random error term, s.t. $\mathbb{E}[\epsilon] = 0$.

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

Basics

Usual Steps:

1. We observe predictors $\boldsymbol{X}$ and response $\boldsymbol{Y}$.
2. We characterize their relationship

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

3. We estimate $\hat{f}$ "somehow".
4. We use $\hat{f}$ in $\boldsymbol{X}$ to predict $\hat{\boldsymbol{Y}}$

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

Focus depends on Goals:

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

Estimation Error

Proposition 1

The magnitude of the total estimation error is bounded below by the variance of ϵ.

Proof. Consider (1) and (2). The total error variance decomposes as

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

Thus $\mathbb{E}[(\boldsymbol{Y} - \hat{\boldsymbol{Y}})^2] \geq \mathrm{Var}[\epsilon]$ is a lower bound by construction. ∎

Estimation of f: Parametric vs. Non-Parametric

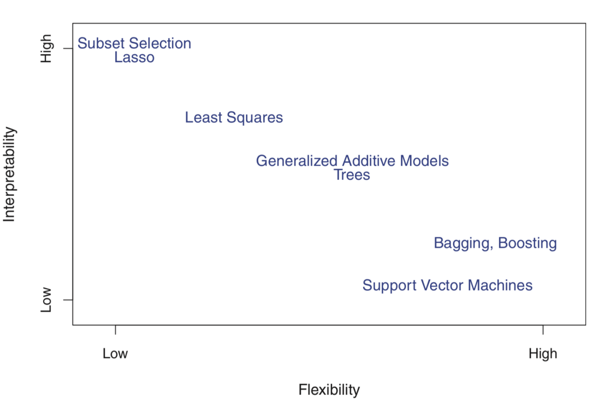
Parametric Methods:

- First: impose rigid structure on f, e.g. f is linear.
- Second: estimate $\hat{f}$.
- Trade-off: p is small, thus easy to interpret (good), but poor accuracy (bad).

Non-Parametric Methods:

- First: impose flexible structure on f, e.g. f has a piecewise polynomial representation
- Second: estimate $\hat{f}$.
- Trade-off: p is large, thus difficult to interpret (bad), better accuracy (good).

Estimation of f: Parametric vs. Non-Parametric



Source: [JO13]

Estimation of f: Regression vs. Classification

- If the response is discrete then its a classification problem, and if it is continuous it is a regression problem.
- There are different methods depending on whether we face a regression or a classification problem.
- Some methods deal with both, e.g. K-nearest neighbors, boosting.

Estimation of f: Assessing Model Accuracy

Definition 1.1 (Mean Squared Error)

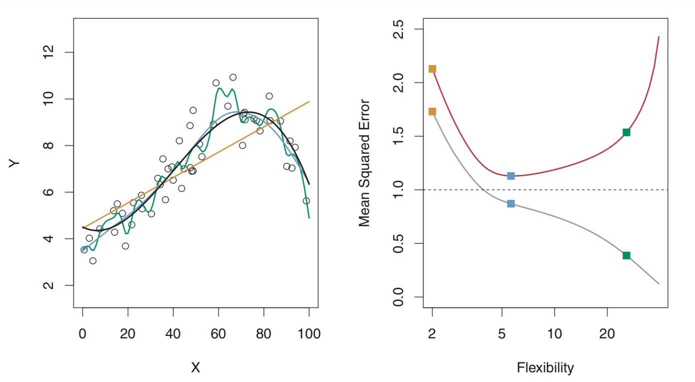
The Mean Squared Error (MSE) is defined as

$$\begin{aligned} MSE &:= Ave\{(\boldsymbol{Y} - \hat{f}(\boldsymbol{X}))^2\} \\ &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_{i,1}, \ldots, x_{i,p}))^2. \end{aligned} \tag{4}$$

We call it "train" MSE if we compute it with training data $(\boldsymbol{X}, \boldsymbol{Y})$ and "test" MSE if we compute it with "test" data $(\boldsymbol{X}_0, \boldsymbol{Y}_0)$.

- Train MSE can be reduce arbitrarily (overfitting).
- Test MSE is used for model selection.

Estimation of f: Assessing Model Accuracy



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

Source: [JO13]

Proposition 2

Given $\boldsymbol{X}_0$, the expected test MSE can always be decomposed into the sum of three quantities: the variance of $\hat{f}(\boldsymbol{X}_0)$, the squared bias of $\hat{f}(\boldsymbol{X}_0)$ and the variance of the error term.

Proof. From proposition 1 we known that

$$\begin{aligned}
\mathbb{E}[(\boldsymbol{Y}_0 - \hat{f}(\boldsymbol{X}_0))^2] &= \mathbb{E}[(f(\boldsymbol{X}_0) - \hat{f}(\boldsymbol{X}_0))^2] + \mathrm{Var}[\epsilon] \\
&= \mathbb{E}[(\{f(\boldsymbol{X}_0) - \mathbb{E}[\hat{f}(\boldsymbol{X}_0)]\} \\
&\quad -\{\hat{f}(\boldsymbol{X}_0) - \mathbb{E}[\hat{f}(\boldsymbol{X}_0)]\})^2] + \mathrm{Var}[\epsilon],
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathbb{E}[(\boldsymbol{Y}_0 - \hat{f}(\boldsymbol{X}_0))^2] &= \underbrace{\mathbb{E}[(f(\boldsymbol{X}_0) - \mathbb{E}[\hat{f}(\boldsymbol{X}_0)])^2]}_{\text{Squared Bias}} + \underbrace{\mathbb{E}[(\hat{f}(\boldsymbol{X}_0) - \mathbb{E}[\hat{f}(\boldsymbol{X}_0)])^2]}_{\text{Variance of } \hat{f}} \\
&\quad + \underbrace{\mathrm{Var}[\epsilon]}_{\text{Irreducible error}}
\end{aligned}$$

Estimation of f: Bias - Variance Trade-off

Proposition 2 says that:

- 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.
- Again, it is not possible to obtain an error below the irreducible error.

To reduce test MSE:

- More flexible methods tend to reduce bias and increase variance.
- Less flexible methods tend to increase bias and reduce variance.
- The adequacy of the method depends on the data.

Estimation of f: Classification Setting

Definition 1.2 (Training Error Rate)

The training error rate (TER) is the proportion of mistakes that the classifier makes in the training set

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

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_0 in the test data set.

Estimation of f: Classification Setting

Note that the Bayes classifier:

- Is theoretical, i.e. the conditional probability in (6) is unknown.
- Provides a lower bound on TER,

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

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.

Regarding KNN:

- K has a strong effect on the classification obtained by KNN.
- K small implies low bias and high variance.
- K large implies high bias and low variance.

R Software

The R Project for Statistical Computing

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

Click here to download.

Basic Commands

- The call for a function called `funcname`, which has two arguments, `arg1` and `arg2`, is of the form `funcname(arg1,arg2)`.
  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     2
[2,]    3     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`. For the mean, use `mean` and, for the variance, `var`

```
> x <- rnorm(100)
> y <- rnorm(100, mean = 50, sd = 0.1)
```

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

```
> z <- x + y
> cor(x, z)
[1]  0.94
```

- 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, add = T)
> 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.table("Auto.data", header = T,
  na.string = "?")
> fix(Auto)
> Auto = read.csv("Auto.csv", header = T,
  na.string = "?")
> fix(Auto)
```

Loading Data

- To delete rows with missing values use `na.omit()`.

```
> dim(Auto)
[1] 397 9
> Auto = na.omit(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, mgp)
  ```

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

Additional Graphical and Numerical Summaries

- To plot a histogram use `hist()`

```
> hist(mgp)
> hist(mgp, col = 2)
> hist(mgp, col = 2, breaks = 15)
```

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

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

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

```
> summary(mgp)
> summary(Auto)
```

Homework

Exercises 2, 3, 4, 6, 7, 9, 10 from [JO13].