NOTES 6

KNN and Logistic Regression

Acknowledgement: some of the contents are borrowed with or without modification from *An Introduction to Statistical Learning, with applications in R* (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R.Tibshirani.

KNN (K Nearest Neighbors)

① define our neighbour (shortest distance)

- Classification:
  - A qualitative response Y takes values in an unordered set C such as eye color∈ {brown, blue, green}
  - Given a feature vector X , predict the value for Y in the set C.
- Non-parametric
- Can be used when the response with multiple classes
- If the input vector X contains r attributes, $X_1, X_2 \ldots X_r$, then each observation lives in *r*-dimensional space.

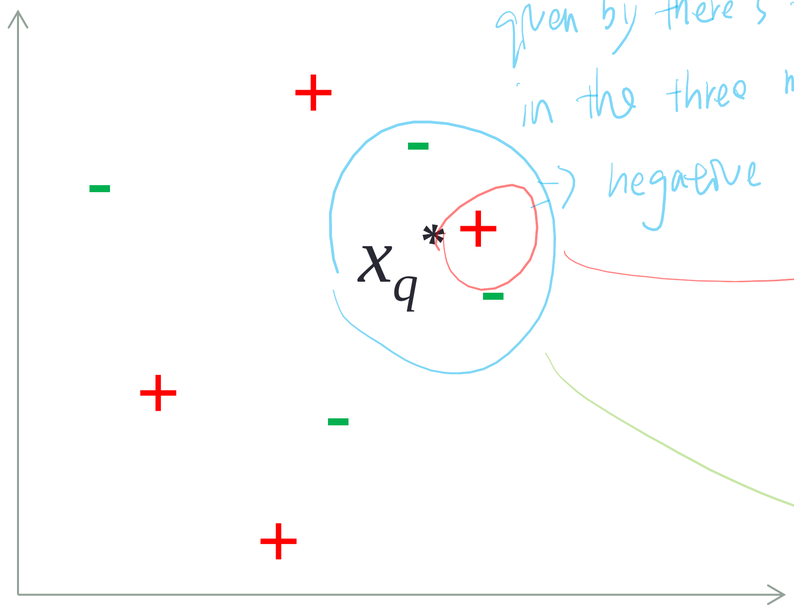
memorilize

- The Euclidean Distance between two observations $\mathbf{x}_i$ and $\mathbf{x}_j$ is:

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ir} - x_{jr})^2}$$

Example: nearest neighbor

- K is the number of neighbors that you want to take into the consideration.



given by there's two negative in the three most nearest → negative

1-Nearest neighbor?

+

3-Nearest neighbor?

−

- How to determine K
  - The one maximizing the accuracy using Cross-validation or LOOCV

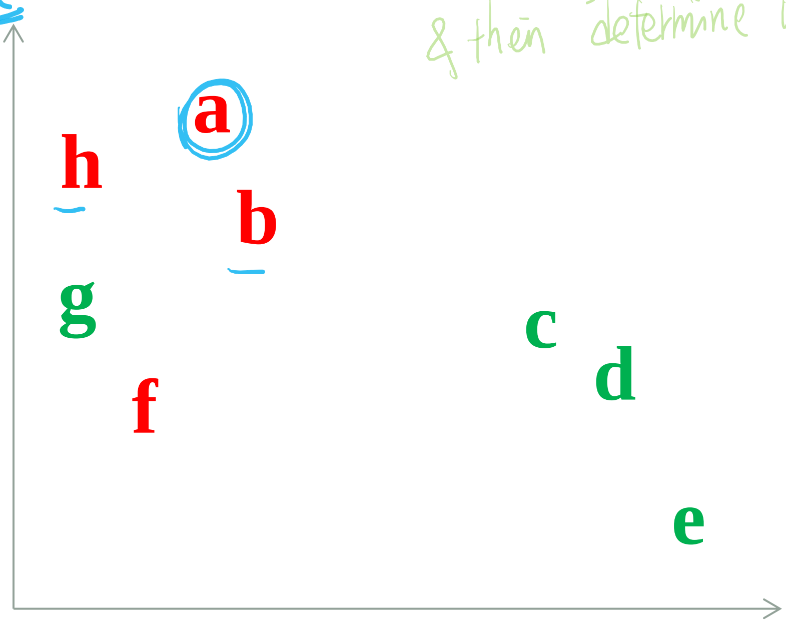
leave one out cross-validation

Example: nearest neighbor

We prefer to use LOOCV to determine when k=1 OR k=3 & then determine which one is better

① 1NN pick the nearest one

k=1? k=3?

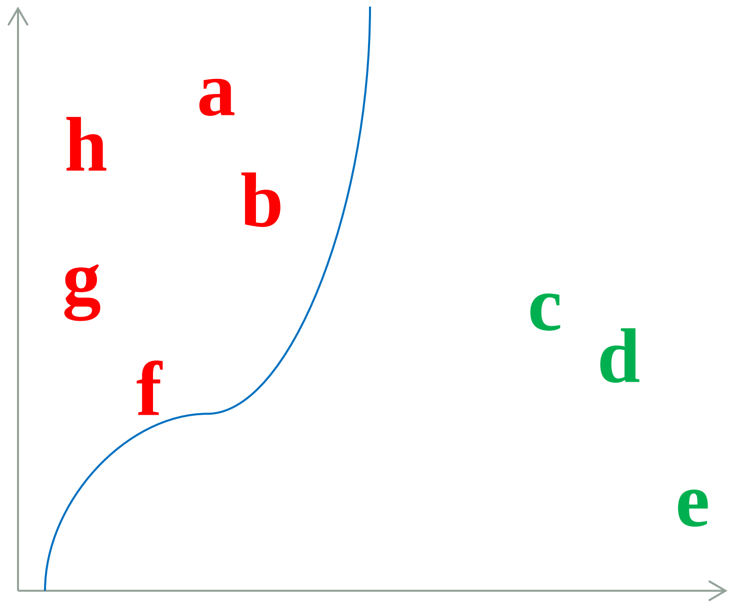


- For $k = 1, 2, \ldots, K$
  - $err(k) = 0$
  - For $i = 1, 2, \ldots, n$
    - $*$ Predict the class label $\widehat{y}_i$ for $\mathbf{x}_i$ using the remaining data points
    - $*$ $err(k) = err(k) + 1$ if $\widehat{y}_i \neq y_i$
- Output $k^* = \arg\min_{1 \leq k \leq K} err(k)$

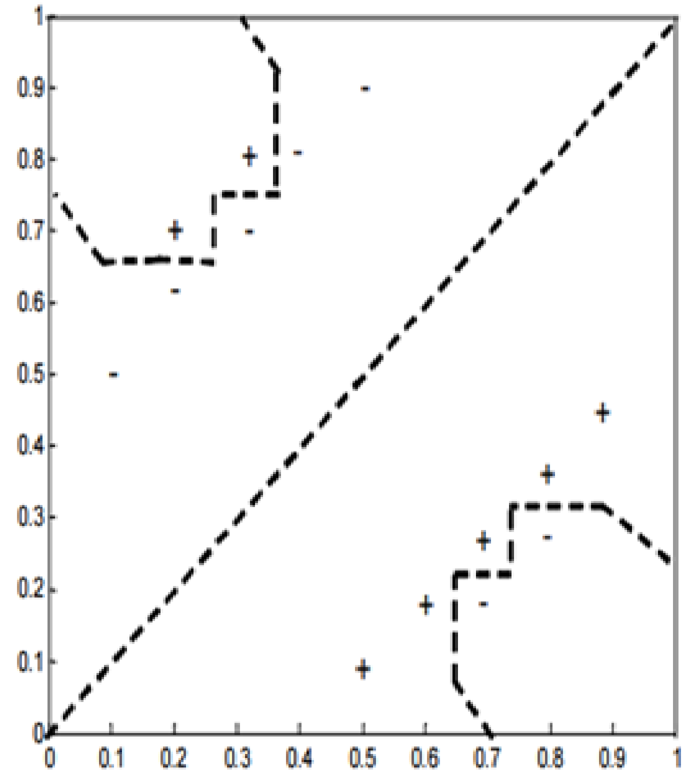
Use KNN to predict the missing values

Example: nearest neighbor

- Output when k=3 Decision boundary



- An example of decision boundary



Normalizing

- Some attributes may take larger values and others
- Normalize

should not take larger weight

$$newx_i = \frac{x_i - \min x_i}{\max x_i - \min x_i}$$

- All attributes on equal footing
- When use KNN?
  - Lots of training data
  - Less than 20 attributes per observations
  - Outperforms logistic regression when the decision boundary is highly non-linear

```
R Function: package (class)
knn(train, test, cl, k……)
cl: factor of true classifications of training set
```

Logistic Regression

not regression, but classification

- Logistic Regression:
  - A qualitative response Y
  - We are more interested in estimating the probabilities that Y belongs to each category, P(Y=Category 1|X)

Probability

- It is adopted when the predictions are desired to remain in the range [0,1], and still use a linear model.

An Illustrated Example

- Default.csv predict the value of default

- We are interested in predicting whether an individual will default on his or her credit card payment, on the basis of income, monthly credit card balance and student status.

- Response: default with two categories (Yes or No)

- We can use logistical regression to estimate
  - Pr(default = Yes|X) & Pr(default = No|X)
  - If Pr(default = Yes|X)>threshold, e.g. 0.5, classify as “Yes”

Data Structure

```
> head(Default)
   default student   balance    income
 1      No      No  729.5265 44361.625
 2      No     Yes  817.1804 12106.135
 3      No      No 1073.5492 31767.139
 4      No      No  529.2506 35704.494
 5      No      No  785.6559 38463.496
 6      No     Yes  919.5885  7491.559
> dim(Default)
[1] 10000     4
> str(Default)
'data.frame':   10000 obs. of  4 variables:
 $ default: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1
1 ...
 $ student: Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 1 2 1
1 ...
 $ balance: num  730 817 1074 529 786 ...
 $ income : num  44362 12106 31767 35704 38463 ...
```

Why Not Linear Regression?

- When the response with more than three categories
  - The coding suggests an order, which is in fact not at all

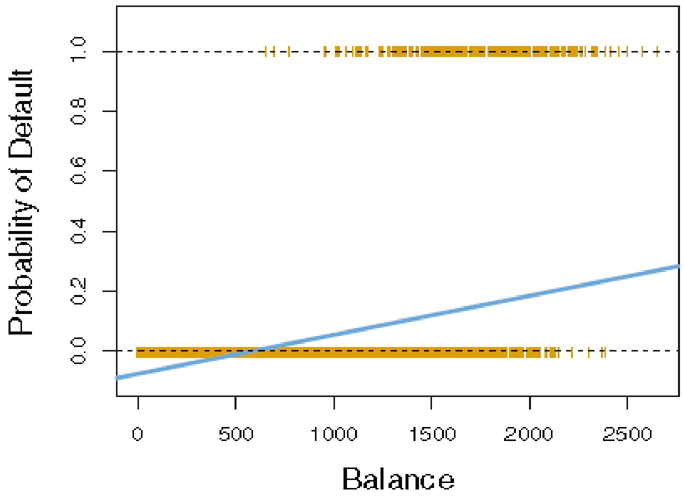
$$y = \begin{cases} 1 & if\ red \\ 2 & if\ green \\ 3 & if\ blue \end{cases}$$

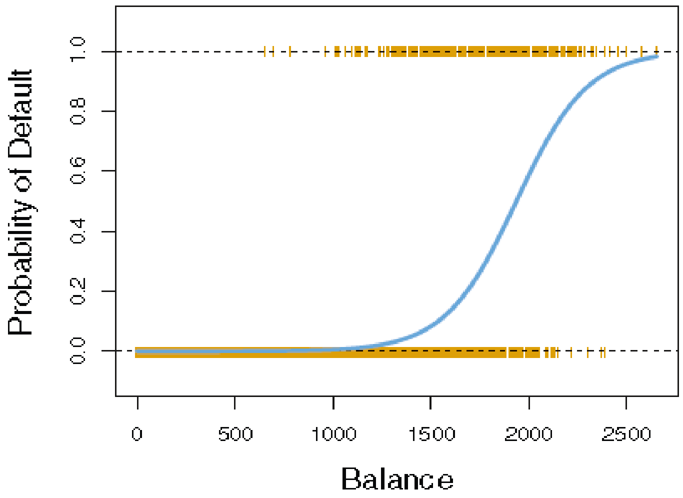
  - Different coding could lead to very different model.

$$y = \begin{cases} 1 & if\ green \\ 2 & if\ blue \\ 3 & if\ red \end{cases}$$

Why Not Linear Regression?

- Binary response:
  - We indeed can code default as 0/1, and perform a linear regression between Y and X
  - However, a linear regression is a line. can't make sure the outcome is between 0 & 1
  - If a straight line is fit to a binary response, it can always produce probabilities less than zero or bigger than one.
  - Logistic regression is preferred.





Logistic Regression

$$default = \begin{cases} 0 & if\ No \\ 1 & if\ Yes \end{cases}$$

- Let's use p(X) = Pr(Y = 1|X) for short and predict default=Yes using X.

- p(X) should be between 0 and 1 for all values of *X*

- Logistic function

$$1 < \quad p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}. \quad < 0$$

- *e* ≈ 2.71828 is a mathematical constant

Logistic Regression, Odds, Logit

- Given $p(X) = \frac{e^{\cdots}}{1+e^{\cdots}}$

- We can obtain: $$\boxed{\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1*X_1+\ldots+\beta_P*X_P}}$$

- $\frac{p(X)}{1-p(X)}$ is called the odds, which is between 0 and ∞

- A bit of rearrangement gives:

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

  - Looks familiar? It is why we say it is a linear model
  - The left-hand side is called the *log-odds* or *logit*

Model Estimation

- Logistic regression are usually fit using maximum likelihood methods.

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'}))$$

generalized linear model

- This can be done using the **glm** in R

```
>glm(default~balance+income+student,family="binomial",data=Default)

Call:  glm(formula = default ~ balance + income + student, family = "binomial",
data = Default)

Coefficients:
(Intercept)      balance        income    studentYes
 -1.087e+01    5.737e-03     3.033e-06    -6.468e-01

Degrees of Freedom: 9999 Total (i.e. Null);  9996 Residual
Null Deviance:      2921
Residual Deviance: 1572         AIC: 1580
```

not changed

similar to MSE/RSE

The larger the difference the better to model

	Coefficient	Std. error	Z-statistic	P-value
Intercept	−10.8690	0.4923	−22.08	<0.0001
balance	0.0057	0.0002	24.74	<0.0001
income	0.0030	0.0082	0.37	0.7115
student[Yes]	−0.6468	0.2362	−2.74	0.0062

significant

- Note that the variable income is in thousands of dollars.

Coefficient Interpretation

- The estimated value of β1=0.0057.

Balance ↑ 1 ⇒ logit ↑ 0.005

- The coefficient $\beta 1$ tells us when increasing *Balance* by one unit, the log odds of default (versus non-default) is expected to increase by 0.0057, with all other predictors held fixed.

odds ↑ by 1.0058

- We can also calculate that Exp(β1)=1.0058. The exponentiated coefficient is called Odds Ratio.
- It means that: holding income and student at a fixed value, for a one-unit increase in Balance, the odds of default =Yes (versus not default) increase by a factor of 1.0058; Or we expect to see about 0.0058 or 0.58% increase in the odds of default.

Make a Prediction

- A student with a credit card balance of \$1,500 and an income of \$40 K has an estimated probability of default of:

$$\hat{p}(X) = \frac{e^{-10.869+0.00574\times 1{,}500+0.003\times 40-0.6468\times 1}}{1+e^{-10.869+0.00574\times 1{,}500+0.003\times 40-0.6468\times 1}} = 0.058.$$

- A decision can be made given this probability
- When the response has more than 2 classes, logistic regression can be generalized to:

$$\Pr(Y=k|X) = \frac{e^{\beta_{0k}+\beta_{1k}X_1+\ldots+\beta_{pk}X_p}}{\sum_{\ell=1}^{K} e^{\beta_{0\ell}+\beta_{1\ell}X_1+\ldots+\beta_{p\ell}X_p}}$$

- But logistic regression is more popular with 2 classes

Confusion Matrix

```
> table(test.truevalue,glm.pred1)
                    glm.pred1
test.truevalue      No   Yes
             No   1927     9
             Yes    46    18
```

N P

predict value

true value

T

F

55 errors

more important

accuracy for Yes

No

$$\frac{1927+18}{1927+18+46+9}$$

$$\frac{18}{18+46}$$

$$\frac{1927}{1927+9}$$

higher recall

classify people who are likely to default

Classification Evaluation

- Two types errors
  - Type I: False Positive
  - Type II: False Negative

- Confusion Matrix
  - The diagonal elements of the confusion matrix indicate correct predictions----Accuracy!

accuracy = $\frac{TN + TP}{TN + TP + FN + FP}$

		Predicted class		
		− or Null	+ or Non-null	Total
True class	− or Null	True Neg. (TN)	False Pos. (FP)	N
	+ or Non-null	False Neg. (FN)	True Pos. (TP)	P
	Total	N*	P*	

should be negative, but we predict if as positive.

should be positive, but we predict it as negative

recall: ~~[illegible]~~ $\frac{TP}{TP+FN}$

Name	Definition	Synonyms
False Pos. rate	FP/N	Type I error, 1−Specificity
True Pos. rate	TP/P	1−Type II error, power, sensitivity, recall
Pos. Pred. value	TP/P*	Precision, 1−false discovery proportion
Neg. Pred. value	TN/N*	

$\frac{TP}{TP+FP}$

performance is ~~poor~~ given that recall & precision