A. Giovanidis 2021

08. Classification

Network Data Analysis - NDA'21
Anastasios Giovanidis

Sorbonne-LIP6

November 09, 2021

Bibliography

A. Giovanidis 2021

B.1 Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. "An introduction to statistical learning: with applications in R". Springer Texts in Statistics. ISBN 978-1-4614-7137-0
Chapter 2, Chapter 4
DOI 10.1007/978-1-4614-7138-7

B.2 Nesrine Ammar, Ludovic Noirie, Sébastien Tixeuil. "Amélioration de l'identification du type des objets connectés par classification supervisée", CORES 2019, Online: https://hal.archives-ouvertes.fr/hal-02126555

B.3 Giorgos Dimopoulos, Ilias Leontiadis, Pere Barlet-Ros, Konstantina Papagiannaki. "Measuring Video QoE from Encrypted Traffic", IMC '16 Proceedings of the 2016 Internet Measurement Conference Pages 513-526 .

Classification Setting

A. Giovanidis 2021

We have seen how to fit models to data when the response y_i to the input x_i is quantitative (e.g. "0.57", "24", "-24.3", etc.)

Question: How do we choose models and define their accuracy, when y_i's are qualitative?

Examples: ("Yes", "No"), ("Red", "Blue", "Green"), ("Malaria", "Yellow Fever", "Flu", "COVID") or more generally:

☞ ("Class 1", "Class 2", . . . , "Class M")

Application A: IoT Classifier

A. Giovanidis 2021

☞ Example application: Internet-of-Things (IoT) for home networks.
"Device identification assistant." from [B.2]

- Home devices can be controlled from distance. (Camera, Light, Sensor, Mobile, Switch, Alarm, Tablet, Speaker, TV.)
- For better quality-of-service these devices need to be identified **by type** from the network.
- Massive number of devices with heterogeneous functionality!

Use supervised learning to train an object classifier.

Input data:
(a) the data-flow information per device, i.e. traffic characteristics.
(b) a selected list of attributes (features).

A. IoT Features

A. Giovanidis 2021

☞ Once a device is connected, a MAC address is attributed.

Feature set to use for classification:

- Flow-based statistics:
  - Packet size (mean, max, min)
  - Mean inter-arrival packet time in a flow.
  - Flow-size measured in number of packets.
  - Protocol type: HTTP, HTTPS, SSDP, mDNS, TFTP, etc.
- Textual attributes (Bag-of-words): 0 or 1 per word per object?
  - Fabrication mark from MAC address.
  - Model and Type from HTTP.

A. IoT Implementation

A. Giovanidis 2021

- WiFi access connected to an Ethernet switch.
- A measurement computer is connected at the switch to trace traffic.
- The computer collects data from the new IoT device during 1 min.
- The computer contains the trained classifier, which decides the most relavant class the IoT device belongs to. The decision is probabilistic.

Types of classifier: K-Nearest Neighbours, Naive Bayes, Random Forest, Tree-based classifier, etc.

Application B: Classifying Video QoE

A. Giovanidis 2021

☞ How to detect video streaming QoE issues from encrypted traffic? (see [B.3])

- Use predictive models to detect different levels of QoE degradation, due to: stalling, average video quality, quality variations.

Labels:

- Stalling: (None, Mild, Severe)
- Video Quality: (Low, Medium, High)
- Quality Switch: use frequency and amplitude of switches.

Features:
(a) Chunk size percentiles, and average.
(b) Packet retransmissions, (c) Bandwidth-Delay Product (BDP),
(d) Bytes-In-Flight (BIF).

Training Accuracy

A. Giovanidis 2021

Suppose we have training observations:
$D_n = \{(x_1, y_1),\ (x_2, y_2), \dots, (x_n, y_n)\}$, with $y_1,\ \dots,\ y_n$ qualitative.

Consider a fitting model with an estimate $\hat{y}_i = \hat{f}(x_i)$.
We use the training error rate:

$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left(y_i \neq \hat{y}_i\right).$$

This is the fraction of incorrect classifications:

- $\hat{y}_i$ is the predicted class label for the i-th observation using $\hat{f}$.
- $\mathbf{1}\left(y_i \neq \hat{y}_i\right) = 0$ for correct classification, else 1.
- Similar to MSE_{train} in regression!

Test Accuracy

A. Giovanidis 2021

Most interested in the error rates of the classifier to test observations $(x_o, y_o) \notin D_n$, not used in training.

Again for an estimate $\hat{y}_o = \hat{f}(x_o)$ we use the test error rate:

$$Ave\left(\mathbf{1}\left(y_o \neq \hat{y}_o\right)\right).$$

☞ A good classifier is the one for which the test error is smallest !

Confusion Matrix

A. Giovanidis 2021

In abstract terms, the confusion matrix is as follows:

		Actual class	
		P	**N**
Predicted class	**P**	**TP**	FP
	N	FN	**TN**

where: P = Positive; N = Negative; TP = True Positive; FP = False Positive; TN = True Negative; FN = False Negative.

Figure: (source: wikipedia "Confusion matrix")

☞ Two types of errors (False Negative, and False Positive)

- FP: Incorrectly assign an individual of Class N to Class P.
- FN: Incorrectly assign an individual of Class P to Class N.

Definitions of performance

A. Giovanidis 2021

		True condition				
	Total population	Condition positive	Condition negative	Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) $= \frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	
Predicted condition	Predicted condition positive	**True positive**	**False positive**, Type I error	Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$	
	Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$	
		True positive rate (TPR), Recall, Sensitivity, probability of detection, Power $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$	F_1 score $= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
		False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$		

Figure: (source: wikipedia "Confusion matrix")

Precision and Recall

A. Giovanidis 2021

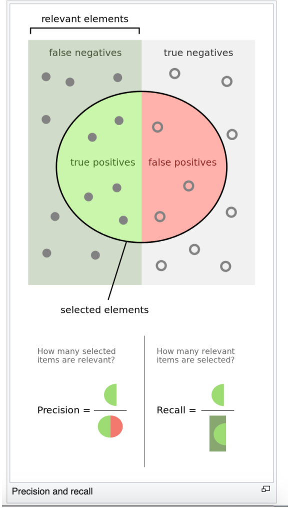

Figure: (source: wikipedia "Precision and recall")

Metrics

A. Giovanidis 2021

Accuracy (ACC)	$\frac{TP+TN}{TP+FP+TN+FN}$	
Precision Positive predictive value (PPV)	$\frac{TP}{TP+FP}$	
Recall (Sensitivity) True positive rate (TPR)	$\frac{TP}{TP+FN}$	**False negative rate** $FNR = 1 - TPR$
Specificity True negative rate (TNR)	$\frac{TN}{TN+FP}$	**False positive rate** $FPR = 1 - TNR$

ROC Curve

A. Giovanidis 2021

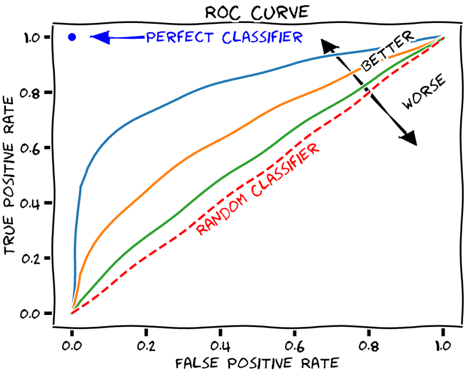

Figure: (source: wikipedia "Receiver operating characteristic")

A. Giovanidis 2021

Examples

A			B			C			C′		
TP=63	FP=28	91	TP=77	FP=77	154	TP=24	FP=88	112	TP=76	FP=12	88
FN=37	TN=72	109	FN=23	TN=23	46	FN=76	TN=12	88	FN=24	TN=88	112
100	100	200	100	100	200	100	100	200	100	100	200
TPR = 0.63			TPR = 0.77			TPR = 0.24			TPR = 0.76		
FPR = 0.28			FPR = 0.77			FPR = 0.88			FPR = 0.12		
PPV = 0.69			PPV = 0.50			PPV = 0.21			PPV = 0.86		
F1 = 0.66			F1 = 0.61			F1 = 0.23			F1 = 0.81		
ACC = 0.68			ACC = 0.50			ACC = 0.18			ACC = 0.82		

Figure: Four confusion matrices (source: wikipedia "Receiver operating characteristic")

A. Giovanidis 2021

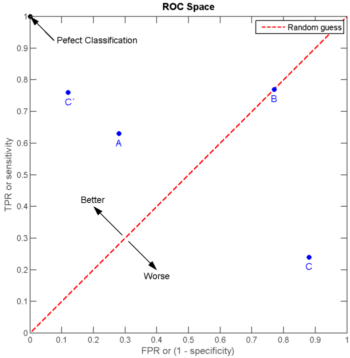

Figure: (source: wikipedia "Receiver operating characteristic")

Classifiers

A. Giovanidis 2021

We will further consider in this lecture the following classifiers:

- ✯ (Wise) Bayes classifier
- ✯ K-Nearest-Neighbours classifier (KNN)
- ✯ Naive Bayes classifier
- ✯ Logistic Regression (LR)

Bayes Classifier

A. Giovanidis 2021

Optimal Classifier: (If all misclassifications are equally important) Assign each observation to the most likely class, given its predictor values:

$$\max_{1 \leq j \leq M} \; Pr\left(Y = j \mid X = x_o\right)$$

- We consider *conditional probabilities* given the observed x_o.

☞ In a two-class problem

$$Pr\left(Y = 1 \mid X = x_o\right) + Pr\left(Y = 2 \mid X = x_o\right) = 1:$$

$$\begin{array}{ll} \text{Class } 1, & \text{if } Pr\left(Y = 1 \mid X = x_o\right) > 0.5 \\ \text{Class } 2, & \text{if } Pr\left(Y = 2 \mid X = x_o\right) > 0.5 \end{array}$$

☞ Decision boundary $Pr\left(Y = 1 \mid X = x_o\right) = Pr\left(Y = 2 \mid X = x_o\right)$

Bayes example

A. Giovanidis 2021

orange region: $Pr(Y = \text{"orange"} \mid X) > 0.5$

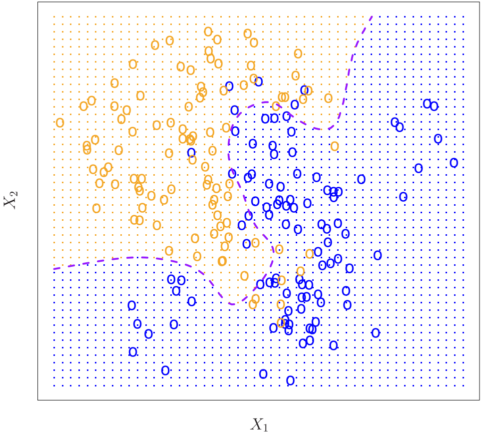

Figure: Bayes classifier : D_{100} data-set and 2 classes (blue, orange). [1]

[1] Source [B.1]

Bayes classifier cont'd

A. Giovanidis 2021

- Orange shaded region: $Pr(Y = \text{“}orange\text{”} \mid X) > 0.5$.
- Blue shaded region: $Pr(Y = \text{“}blue\text{”} \mid X) > 0.5$.
- The dashed line: Bayes decision boundary.
- Circles that fall in regions with different colour: misclassifications (False-Positives or False-Negatives)

☞ Bayes classifier produces lowest test error rate (irreducible) !

$$Test\ Error(x_o) = 1 - \max_j Pr\left(Y = j \mid X = x_o\right)$$

Drawback...

A. Giovanidis 2021

There is one problem however: For real data we do not know the conditional distribution $P(Y|X)$,

(unless we have generated data ourselves, in which case we know the joint distribution $P(X, Y)$).

Bayes classifier serves as an unreachable golden standard!

If we do not know exactly $P(Y|X)$ we can try to estimate it.

KNN classifier

A. Giovanidis 2021

How does the KNN classifier work?

- Choose a positive integer $K > 0$.
- Given a test observation $x_o \notin D_n$, the KNN classifier identifies the K points in the training data-set closest to x_o, it is the set $\mathcal{N}_K(x_o)$.
- The conditional probability for class j at x_o is estimated as:

$$Pr\left(Y = j \mid X = x_o\right) \quad = \quad \frac{1}{K} \sum_{i \in \mathcal{N}_K(x_o)} \mathbf{1}\left(y_i = j\right).$$

- Calculate the estimates for all classes $j = 1, \ldots, M$ and
- Finally, apply Bayes classification: classify x_o to the class with the largest estimated probability.

KNN example

A. Giovanidis 2021

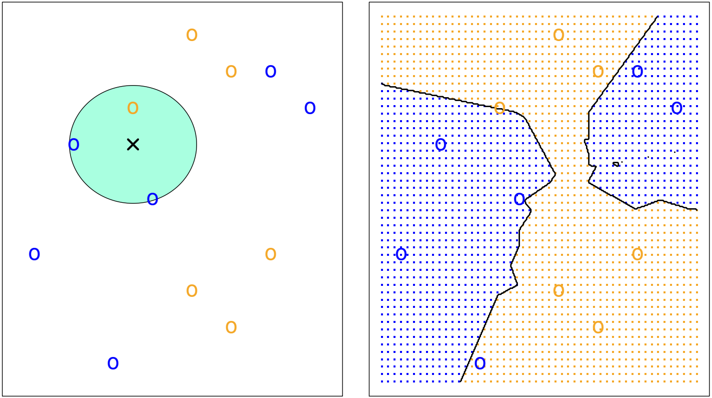

Figure: KNN classifier ($K = 3$) : D_{12} data-set and 2 classes. [2]

[2] Source [B.1]

Optimal Choice of K

A. Giovanidis 2021

Despite its simplicity KNN can give classifiers surprisingly close to Bayes. Choice of K is important:

- If $K = 1$, very flexible decision boundary $\rightarrow$ Low Training Error ($= 0$) but! High Test Error.
- As K increases (less flexibility) Training Error increases but the Test Error may not !
- Find optimal K^* with minimum Test Error ($\bigcup$ shape)
- If $K = 100$ decision boundary close to linear.

Variance vs Bias Tradeoff
or
Flexibility vs Interpretability

A. Giovanidis 2021

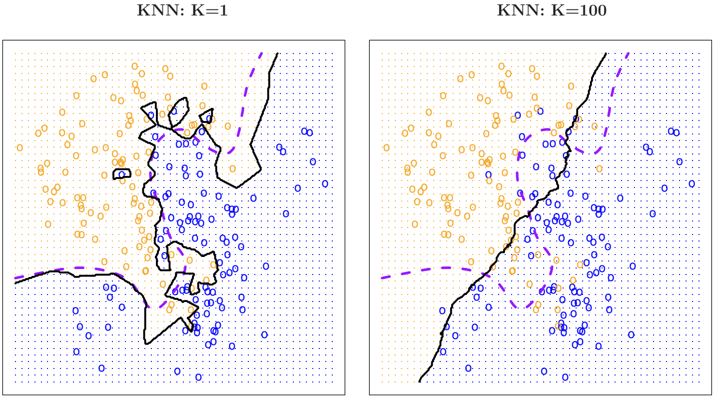

Figure: KNN with $K = 1$ (left) and $K = 100$ (right). [3]

[3] Source [B.1]

A. Giovanidis 2021

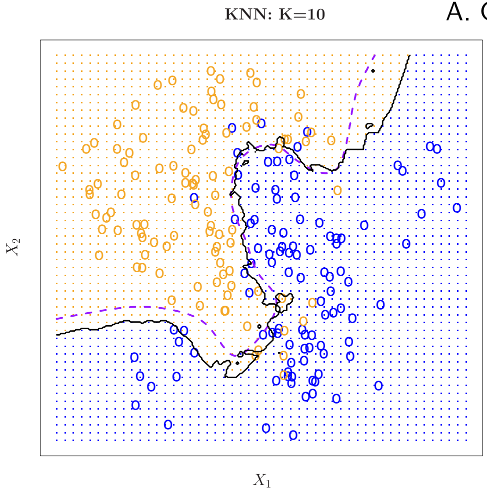

Figure: KNN with $K = 10$ close to Bayes optimal. [4]

[4] Source [B.1]

Variance vs Bias Tradeoff

A. Giovanidis 2021

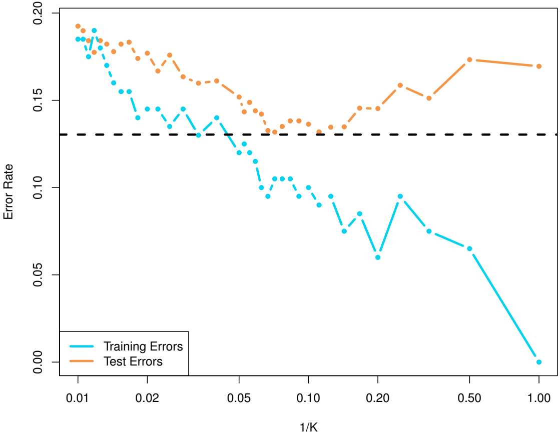

Figure: Training/Test Error Rate. [5]

[5] Source [B.1]

What if... Linear Regression?

A. Giovanidis 2021

Suppose we have again two classes: 'Class 1', 'Class 2'
and $K = 1$ feature.

- What if we used Linear Regression for the $P(Y|X)$?
- Let 'Class 1': $Y = 0$ and 'Class 2': $Y = 1$.
- We assume that the linear model describes the 0/1 data,

$$y_t = \beta_0 + \beta_1 x_t + \epsilon$$

and we look for the regression line

$$\mathbb{E}[Y \mid X] = \hat{\beta}_0 + \hat{\beta}_1 X$$

☞ Since $Y_t \in \{0, 1\}$ then $\mathbb{E}[Y_t \mid X_t] = Pr(Y_t = 1 | X_t) = \hat{\beta}_0 + \hat{\beta}_1 X_t$.

Wrong Shape ! less than 0, more than 1

A. Giovanidis 2021

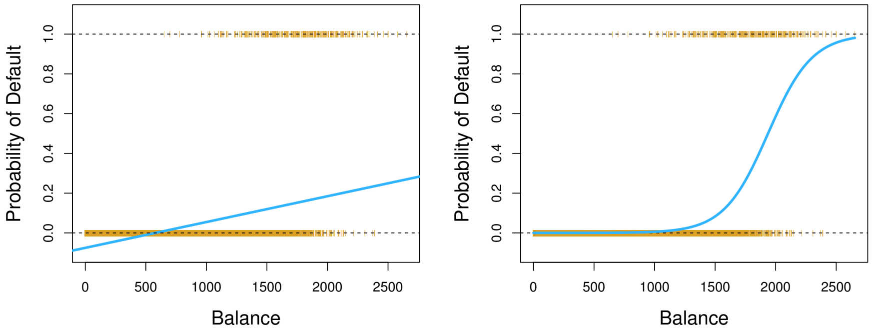

Figure: $Pr\left(Y=1|X\right)$. Linear vs Sigmoidal fit. [6]

[6] Source [B.1]

Logistic Regression

A. Giovanidis 2021

Suppose for the two-class problem $Pr\left(Y=1|X\right)$ follows the logistic function.

$$p(X) := Pr\left(Y=1|X\right) = \frac{e^{\beta_o+\beta_1 X}}{1+e^{\beta_o+\beta_1 X}}$$

- For $X \to -\infty$: $p(X) \to 0$
- For $X \to +\infty$: $p(X) \to 1$
- It is an S-shaped curve.

☞ We need to fit $\beta_o,\ \beta_1$ in the non-linear logistic function.

Logistic fit

A. Giovanidis 2021

We consider a Training data-set D_n with $Y_n = (0, 0, 1, \ldots, 0, 1)$.

- We don't want to use *MSE* fit $\rightarrow$ complicated expressions.
- Better use: log-likelihood function.

What is the likelihood $g(D_n)$ of the data-sample?

$$g(D_n) = \prod_{t:y_t=1} p(x_t) \prod_{t':y_{t'}=0} (1 - p(x_{t'}))$$

because we assumed that for any X

$$Y = \begin{cases} 1, & p(X) \\ 0, & 1 - p(X) \end{cases}$$

and for all $x_t \in D_n$ we know what is the y_t answer.

Log-likelihood maximization

A. Giovanidis 2021

The log-likelihood function, is then equal to

$$
\begin{aligned}
\ell(\beta_0, \beta_1; D_n) &= \log(g(D_n)) \\
&= \sum_{t:y_t=1} \log p(x_t) + \sum_{t':y_{t'}=0} \log\left(1 - p(x_{t'})\right) \\
&= \sum_{t=1}^{n} \left\{ y_t \log p(x_t) + (1 - y_t) \log\left(1 - p(x_t)\right) \right\} \\
&\overset{p(X)=\frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}}{=} \sum_{i=1}^{n} \left\{ y_i \left(\beta_0 + \beta_1 x_i\right) - \log\left(1 + e^{\beta_0+\beta_1 X}\right) \right\}
\end{aligned}
$$

☞ We want to $\max_{\beta_0,\beta_1} \ell(\beta_0, \beta_1;\ D_n)$.

Newton's method

A. Giovanidis 2021

We follow standard process:

- $\nabla \ell(\beta_0, \beta_1; D_n) = \begin{bmatrix} \frac{\partial \ell}{\partial \beta_o} \\ \frac{\partial \ell}{\partial \beta_1} \end{bmatrix}$
- $\nabla^2 \ell(\beta_0, \beta_1; D_n) = \begin{bmatrix} \frac{\partial^2 \ell}{\partial {\beta_0}^2} & \frac{\partial^2 \ell}{\partial \beta_o \partial \beta_1} \\ \frac{\partial^2 \ell}{\partial \beta_1 \partial \beta_0} & \frac{\partial^2 \ell}{\partial {\beta_1}^2} \end{bmatrix} < 0$ negative-definite
- Hence the log-likelihood logistic function is strictly concave.

$$\begin{bmatrix} \beta_0^{(k+1)} \\ \beta_1^{(k+1)} \end{bmatrix} = \begin{bmatrix} \beta_0^{(k)} \\ \beta_1^{(k)} \end{bmatrix} - \left(\nabla^2 \ell(\beta_0, \beta_1; D_n)\right)^{-1} \cdot \nabla \ell(\beta_0, \beta_1; D_n)$$

"What are the odds?"

A. Giovanidis 2021

One can see the logistic expression of the predictions from a different point-of-view:

$$q(x_t) := \frac{p(x_t)}{1 - p(x_t)} \quad = \quad e^{(\beta_0 + \beta_1 x_t)}.$$

☞ odds function: often used in... Horse-racing!

"What are the odds ?"

- If $q(x_t) = 1/4$, then $p(x_t) = P(Y_t = 1 \mid x_t) = 0.2$
- If $q(x_t') = 9/1$, then $p(x_t') = P(Y_t = 1 \mid x_t') = 0.9$.

The logits (or log-odds)

A. Giovanidis 2021

$$Q(x_t) := \log\left(\frac{p(x_t)}{1-p(x_t)}\right) \quad = \quad \beta_0 + \beta_1 x_t.$$

Here we come back to the expression for the Linear Regression!

Separating hyperplane: For $p = 0.5$, we get the "linear" boundary

$$0 = \beta_0 + \beta_1 x_{t,1} \quad (+\beta_2 x_{t,2} + \ldots + \beta_K x_{t,K}), \qquad \textit{for } K \geq 1.$$

e.g. for $K = 1$, it is a point $x_{bound} = -\beta_0/\beta_1$. (left: 1, right: 0)

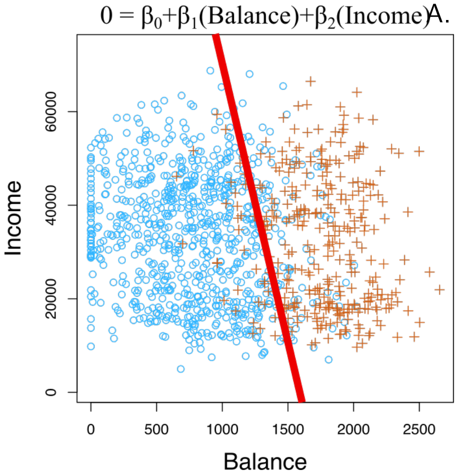

Figure: The boundary separates "blue" from "orange". [7]

[7] Source [B.1]

Test Data (Logistic)

A. Giovanidis 2021

If we have test input data $x_o \notin D_n$, how do we choose its Class?
Say $x_o = (x_{o,1}, x_{o,2}, \dots, x_{o,K})$.

Use the fitted values of $\beta_0, \beta_1, \dots, \beta_K$

- Either calculate $p(x_o) = \frac{e^{\beta_0+\beta_1 x_{o,1}+\dots+\beta_K x_{o,K}}}{1+e^{\beta_0+\beta_1 x_{o,1}+\dots+\beta_K x_{o,K}}}$ and check if $>, =, < 0.5$,
- or check the position of x_o related to the boundary: $\beta_0 + \beta_1 x_{o,1} + \beta_2 x_{o,2} + \dots + \beta_K x_{o,K} >, =, < 0$.

e.g. $\beta_0 + \beta_1 x_{o,1} + \beta_2 x_{o,2} + \dots + \beta_K x_{o,K} > 0 \Rightarrow p(x_o) > 0.5$

☞ We need not always use the value of 0.5 for the boundary...

Multiple Logistic Regression

A. Giovanidis 2021

We have implied that the Logistic Regression is generalised to higher than 1 dimension:

$$\log\left(\frac{p(X)}{1-p(X)}\right) \quad = \quad \beta_0 + \beta_1 X_1 + \ldots + \beta_K X_K,$$

where $X = (X_1, \ldots, X_K)$ are K predictors.

Equivalently,

$$p(X) \quad = \quad \frac{e^{\beta_0 + \beta_1 X_1 + \ldots + \beta_K X_K}}{1 + e^{\beta_0 + \beta_1 X_1 + \ldots + \beta_K X_K}}.$$

☞ $\beta_0, \ldots, \beta_K$ are estimated by the maximum likelihood method.

Example

A. Giovanidis 2021

Using the data set Default we want to decide, whether an individual is likely to default on its bank account, or not. (balance =credit card debt)

$X = (\text{balance}, \ \text{income}, \ \text{student[Yes]})$, so $K = 3$.
$Y = \text{default[Yes]}$.

- First consider only balance, $K = 1$.

	Coefficient	Std. error	Z-statistic	P-value
Intercept	−10.6513	0.3612	−29.5	<0.0001
balance	0.0055	0.0002	24.9	<0.0001

☞ 1-unit increase in (credit-card) balance is associated to $\beta_1 = 0.0055$ units increase in log-odds of default.

A. Giovanidis 2021

Example (predictions)

$\mathrm{default[Yes]}$ probability for an individual with $\mathrm{balance} = 1000$ EUR

$$\hat{p}(\mathrm{balance} = 1000) \quad = \quad \frac{e^{-10.6513+0.0055x1000}}{1 + e^{-10.6513+0.0055x1000}} = 0.00576$$

- Now consider binary $\mathrm{student[Yes]}$, $K = 1$.

	Coefficient	Std. error	Z-statistic	P-value
`Intercept`	−3.5041	0.0707	−49.55	<0.0001
`student[Yes]`	0.4049	0.1150	3.52	0.0004

$$\hat{p}(\mathrm{student[Yes]} = 1) = 0.0431 \quad > \quad \hat{p}(\mathrm{student[Yes]} = 0) = 0.0292$$

Conclusion 1: Students are more likely to default.

Example (multiple)

A. Giovanidis 2021

- Now consider the entire X vector, $K = 3$.

	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

Paradox: Conclusion 2: Students are less likely to default !!!!
($\beta_{\mathrm{student[Yes]}} < 0$)

Why? The $\mathrm{student[Yes]}$ and $\mathrm{balance}$ predictors are correlated.

A. Giovanidis 2021

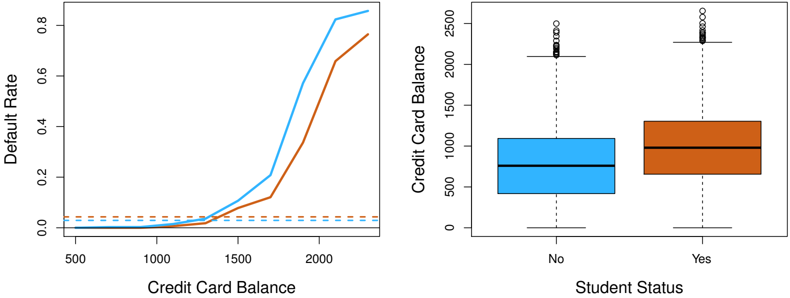

Figure: Students tend to have higher debts in the US/GB/D. [8]

Conclusion 1: For the same credit-card balance a student is less likely to default.

[8] Source [B.1]

Logistic Regression for > 2 Classes

A. Giovanidis 2021

We can easily generalise to M classes:

$$
\begin{aligned}
\log \frac{Pr(Class = 1|X = x)}{Pr(Class = M|X = x)} &= \beta_{1,0} + \beta_1^T x \\
&\dots \\
\log \frac{Pr(Class = M-1|X = x)}{Pr(Class = M|X = x)} &= \beta_{M-1,0} + \beta_{M-1}^T x \\
Pr(Class = M|X = x) &= \frac{1}{1 + \sum_{m=1}^{M-1} \exp(\beta_{m,0} + \beta_m^T x)}
\end{aligned}
$$

• We need $M-1$ log-odds. • The probabilities sum-up to 1.
• The choice of denominator class is arbitrary. • Max likelihood.

☞ For multiple classes, discriminant analysis is more popular...

Naive Bayes

A. Giovanidis 2021

☞ The Naive Bayes classifier:

- Assumes that the K features are independent.
- Uses a simple MAP or ML estimator

$$\begin{aligned} P(Y \mid \mathcal{D}_n) &\propto P(\mathcal{D}_n \mid Y)P(Y) \quad \textbf{[MAP]} \\ P(Y \mid \mathcal{D}_n) &\propto P(\mathcal{D}_n \mid Y) \qquad \textbf{[ML]} \end{aligned}$$

where Y is the class label.
We choose MAP or ML, depending on the prior information over the class distribution Y.

A. Giovanidis 2021

Naive Bayes with discrete features

✐ Let us classify texts (e.g. books, sentences) in one of two classes:

1. History
2. Science

To do so, we will use some features from the available data (texts). These are a certain bag-of-words: {'king', 'food', 'equals', 'proof'}

	Bag-Of-Words				Label	
	1:'king'	2:'food'	3:'equals'	4:'proof'	History	Science
Text 1	No	Yes	Yes	Yes	No	Yes
Text 2	No	No	Yes	No	No	Yes
Text 3	Yes	Yes	No	Yes	Yes	No
...	...	...	...	...	...	...
Text n	Yes	No	Yes	Yes	No	Yes

Naive Bayes with discrete features (II)

A. Giovanidis 2021

✐ If X contains K binary state features, with $X_{t,k} \in \{0,1\}$, then

$$X_t = (X_{t,1}, \ldots, X_{t,K}), \quad t = 1, \ldots, n.$$

$X_{t,k}$ says whether feature k appears or not in the t-th data sample of $\mathcal{D}_n$.

Also, Y is the label of each text. Then, let

$$Y_t = \begin{cases} 0 & \text{if 'History'} \\ 1 & \text{if 'Science'} \end{cases}$$

ML estimators

$$p_1 = p_{Sc} = P(Y=1) = \frac{1}{n}\sum_{t=1}^{n} Y_t, \qquad p_0 = p_{Hi} = P(Y=0) = \frac{1}{n}\sum_{t=1}^{n}(1 - Y_t)$$

$$p_{1,k} = p_{Sc,k} = P(X_k = 1 \mid Y = 1) = \frac{\sum_{t=1}^{n} Y_t \cdot X_{t,k}}{\sum_{t=1}^{n} Y_t}$$

Naive Bayes with discrete features (III)

A. Giovanidis 2021

☞ How does Naive Bayes work in 2 classes ('History'-'Science'), for a Test sample (X_o, y_o)?

- We make use of the estimated Likelihood!
- Suppose the distribution for each feature k per class j is $\mathrm{Bernoulli}(p_{j,k})$ and independent of other features. Then for ML posteriors:

$$P(Y=0 \mid X_o) \propto P(X_o \mid Y=0) = \prod_{k=1}^{K} p_{0,k}^{X_{o,k}}(1-p_{0,k})^{1-X_{o,k}},$$

$$P(Y=1 \mid X_o) \propto P(X_o \mid Y=1) = \prod_{k=1}^{K} p_{1,k}^{X_{o,k}}(1-p_{1,k})^{1-X_{o,k}}$$

For MAP posteriors we need also the Prior distribution over classes, i.e. $p_0 = P(Y=0)$ and $p_1 = P(Y=1)$,

$$P(Y=j \mid \mathcal{D}_n) = P(\mathcal{D}_n \mid Y=j) \cdot P(Y=j), \quad j=0,1.$$

Naive Bayes with continuous features

A. Giovanidis 2021

✐ Suppose that X contains K continuous state features.

- Prior distribution over classes, is assumed uniform, i.e. $P(Y=0)=P(Y=1)=0.5$.
- Suppose the distribution for each feature k per class j is Gaussian $\mathcal{N}(\mu_{j,k},\sigma^2_{j,k})$.

ML estimates for mean and variance

$$\overline{X}_{1,k}=\frac{1}{n}\sum_{t\in\mathcal{D}_n,Y_t=1}X_{t,k},\qquad \overline{X}_{0,k}=\frac{1}{n}\sum_{t\in\mathcal{D}_n,Y_t=0}X_{t,k}$$

$$\overline{S}^2_{1,k}=\frac{1}{n}\sum_{t\in\mathcal{D}_n,Y_t=1}(X_{t,k}-\overline{X}_{1,k})^2,\qquad \overline{S}^2_{0,k}=\frac{1}{n}\sum_{t\in\mathcal{D}_n,Y_t=0}(X_{t,k}-\overline{X}_{0,k})^2.$$

Naive Bayes with continuous features (II)

A. Giovanidis 2021

Given a Test sample (x_o, y_o), the estimated class is the one which maximizes the ML (or MAP) estimator, i.e. the maximum between

$$P(Y = 0 \mid \mathcal{D}_n) = \prod_{k=1}^{K} \frac{1}{(2\pi \overline{S}_{0,k}^2)^{1/2}} \exp\left(-\frac{(x_{o,k} - \overline{X}_{0,k})^2}{2\overline{S}_{0,k}^2}\right) \quad \textit{for} \quad \textit{Class } 0$$

$$P(Y = 1 \mid \mathcal{D}_n) = \prod_{k=1}^{K} \frac{1}{(2\pi \overline{S}_{1,k}^2)^{1/2}} \exp\left(-\frac{(x_{o,k} - \overline{X}_{1,k})^2}{2\overline{S}_{1,k}^2}\right) \quad \textit{for} \quad \textit{Class } 1$$

and similarly as in the discrete case for MAP estimators.

A. Giovanidis 2021

END