Unit 3.
Parametric classification

Artificial Intelligence and Learning

Contents

3.1 Introduction. Cost function. Metrics in classification.

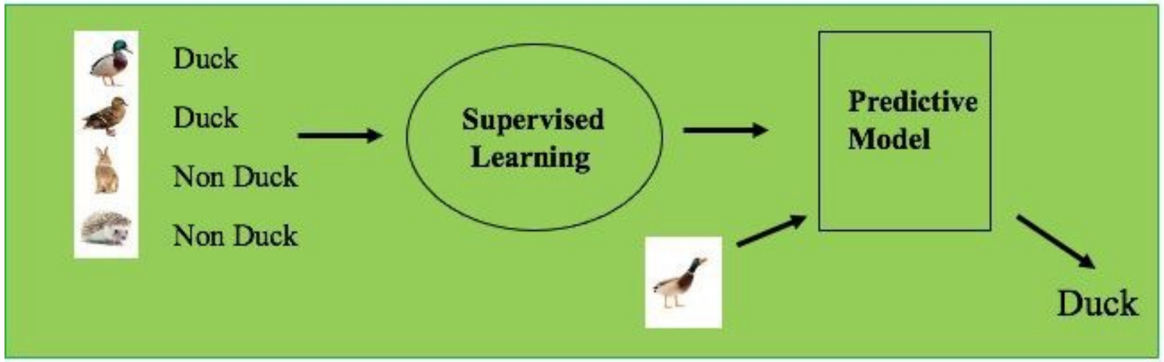
3.2 Logistic regression

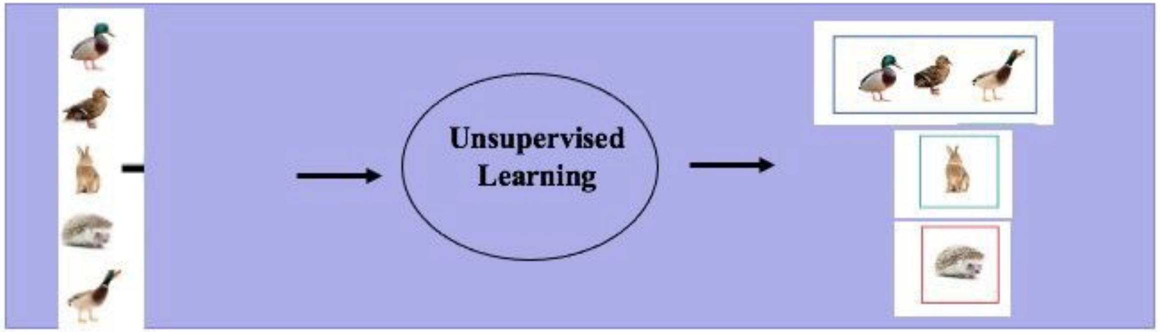
3.3 Logistic regression with regularisation

3.4 Biomedical examples and applications

3.1 Introduction

Supervised and Unsupervised Learning





3.1 Introduction

Supervised and Unsupervised Learning

- **Supervised Learning**
  - **Learn the relationship between feature vectors and the labels associated with each vector**
  - There are **two types** of supervised learning:
    - **Classification:** The set of labels is numerable
    - **Regression:** The set of labels is numerable
- **Unsupervised Learning**
  - Learn the "structure" of the data by grouping examples into consistent groups with similar characteristics
  - It only considers the characteristics vectors

3.1 Introduction

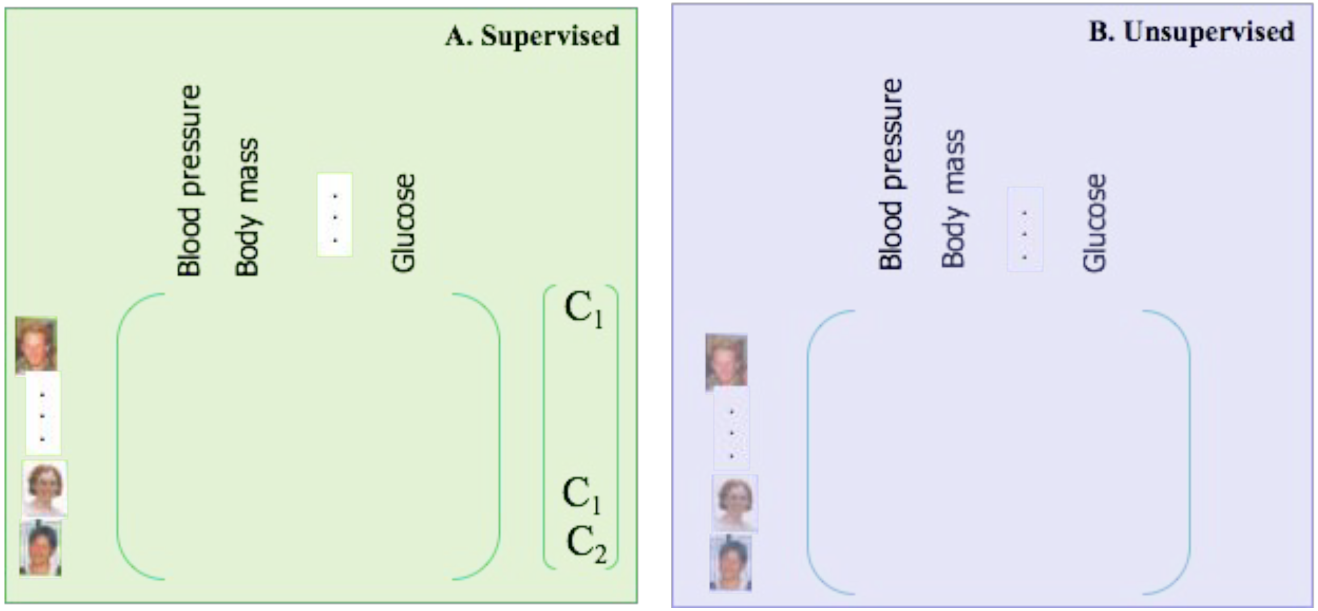
What is the main difference between a supervised and unsupervised learning scheme?

Database:
* 133,000 patients
* 72 variables or characteristics/attribute

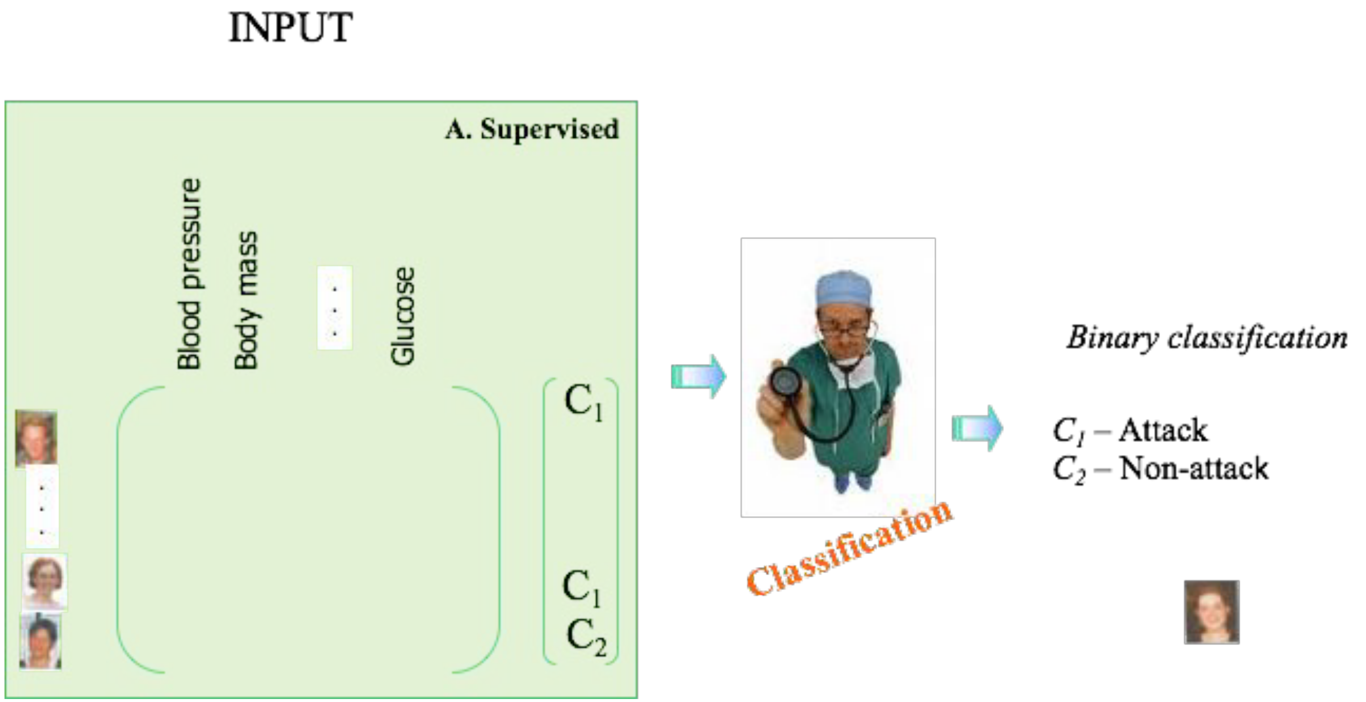
$$\begin{array}{l} \text{Blood pressure} \dashrightarrow \\ \text{Body mass} \dashrightarrow \\ \\ \text{Glucose} \dashrightarrow \end{array} \begin{bmatrix} x_1 \\ x_2 \\ . \\ x_{72} \end{bmatrix} \qquad \begin{array}{l} \text{Blood pressure} \dashrightarrow \\ \text{Body mass} \dashrightarrow \\ \\ \text{Glucose} \dashrightarrow \end{array} \begin{bmatrix} x_1 \\ x_2 \\ . \\ x_{72} \end{bmatrix}$$

3.1 Introduction

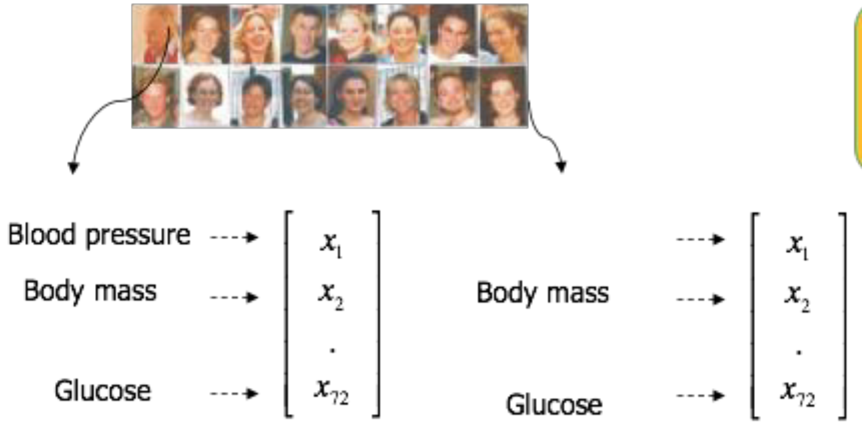
Database:
* 133,000 patients
* 72 variables or characteristics/attribute



3.1 Introduction

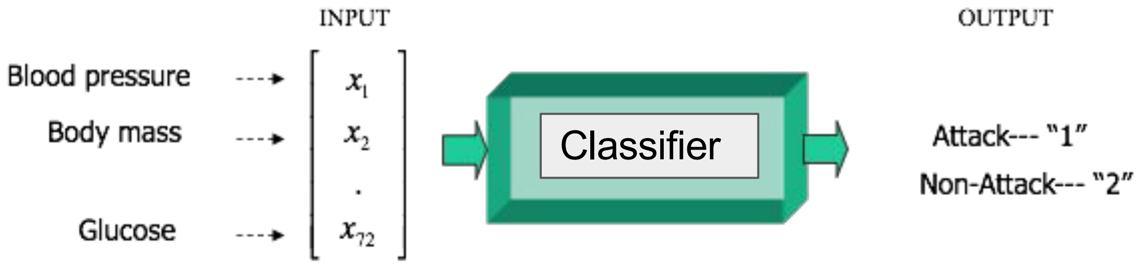


3.1 Introduction

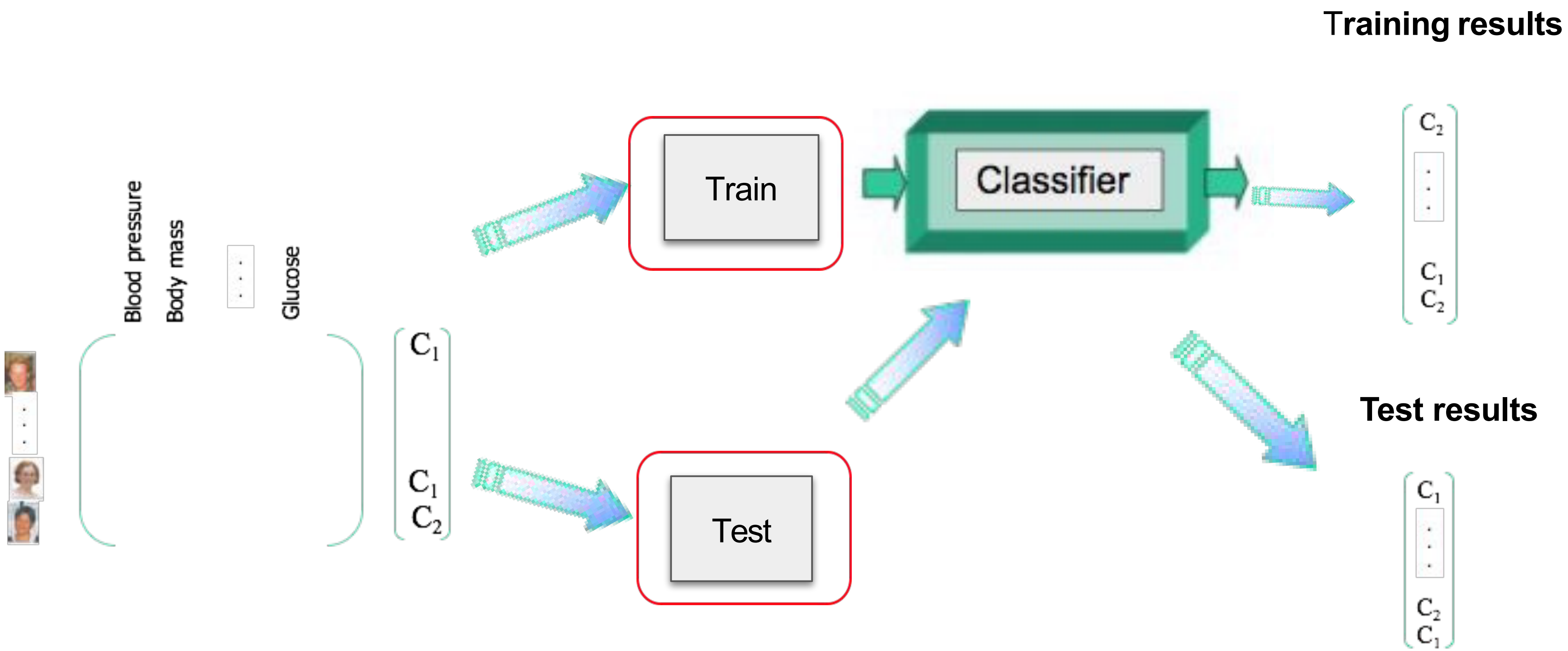


Database:
* 133,000 patients
* 72 variables or characteristics/attributes

What is the relationship between the 72 variables and the risk of attack?

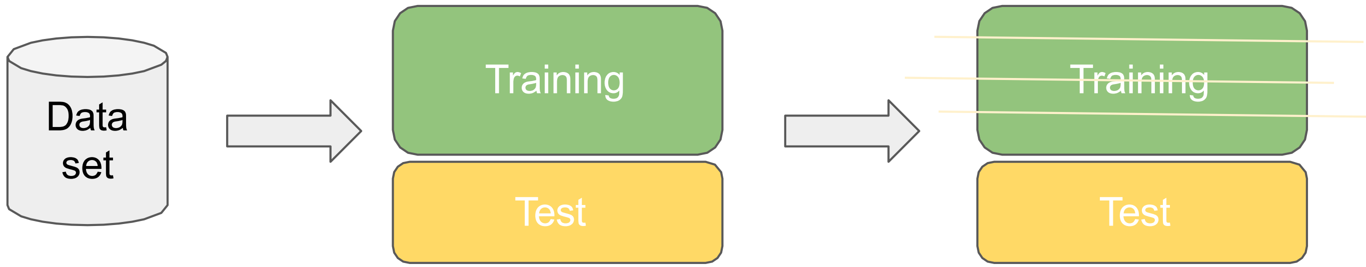


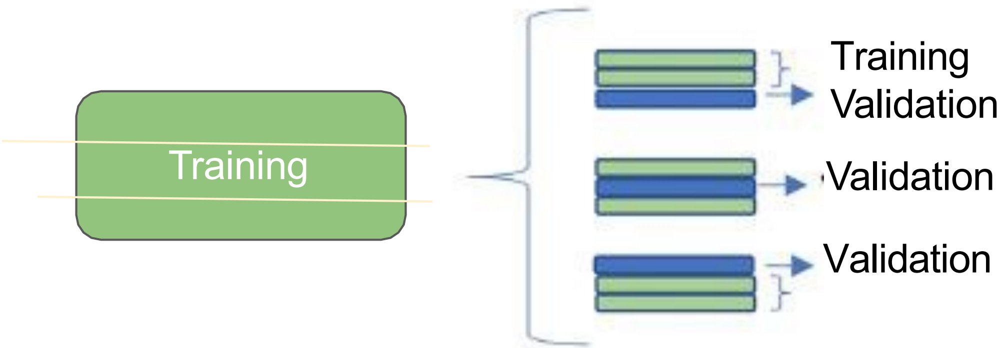
3.1 Introduction



3.1 Introduction

This is an example of 3-Fold Cross Validation



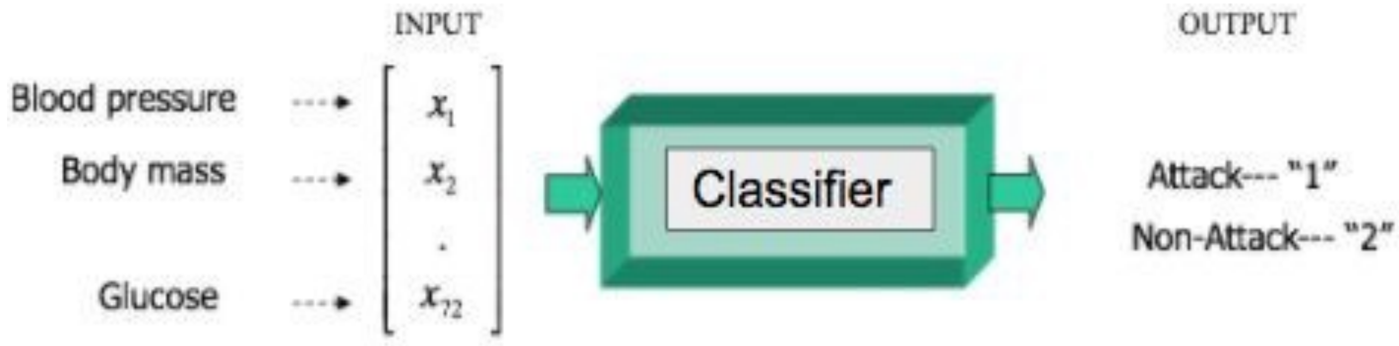


Training set ⇒ is used to determine the model parameters.
Validation set ⇒ is used to evaluate the generalizability of the design.

Choose the design model with the best performance on the validation set. The final model is evaluated with the test set.

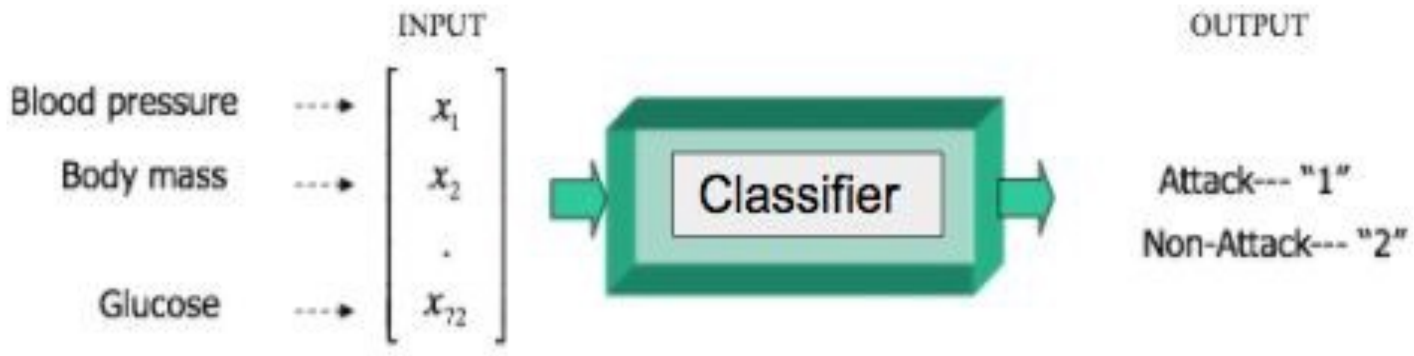
3.1 Introduction

Soft and hard output



As a result of the classification of a feature vector, it is assigned to the category "will suffer a heart attack" if the output is a "1", and to the category "will not suffer a heart attack" if the output is a "0".

3.1 Introduction



However, the classifier can provide **"soft" outputs** (in this case, **values in the range [0,1]**). In this case, it is necessary to apply a threshold to associate the characteristic vector to a class.

Example--> soft output: 0.8. If the threshold is 0.5, then 0.8 > 0.5 → Class 1 → Attack

3.1 Introduction

In classification tasks, there are as many classes (or categories) possible as there are different values of the target variable (output):

- **Binary classification**: the number of possible values of the target variable is just two.
- **Multi-class classification:** if you want to distinguish between more than two classes.

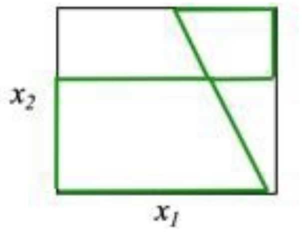
Example:

Output 1: A patient can be diabetic or hypertensive → Binary classification (2 classe)

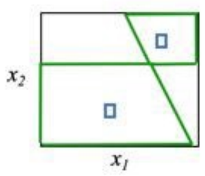
Output 2: A patient can be diabetic, hypertensive or diabetic and hypertensive → Multiclass classification (3 classes)

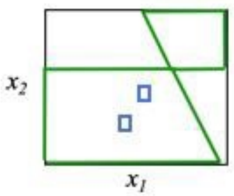
3.1 Introduction

Classification border. Linear and non-linear classifier



- After design the classifier, the input feature space is divided into different regions.
- Each region is associated with a class. The separation between two or more regions is called a **classification boundary**.



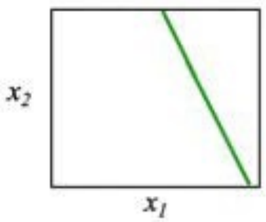


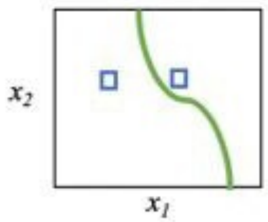
To classify a case, from a geometric point of view, it is enough to put the observation in the representation space and assign it as **hard output** the label of the region in which it is positioned.

In this example, the class associated with the two observations would be the same, since both are in the same region

3.1 Introduction

Classification border. Linear and non-linear classifier



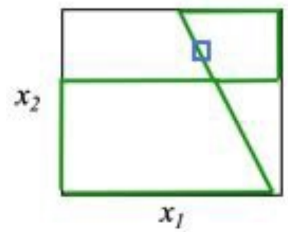


Linear classifier

$$w_0 + w_1x_1 + w_2x_2$$

Non-Linear classifier

$$w_0 + w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_2^2 + w_5x_1^3$$

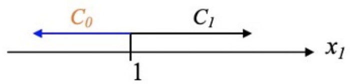


The EXAMPLES in the border, the classifier assigns the same probability of belonging to one or another class.

3.1 Introduction

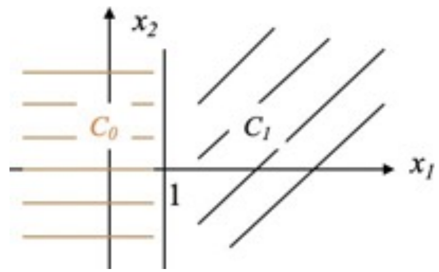
Curse of Dimensionality

$$x_1 \underset{D_0}{\overset{D_1}{\gtrless}} 1$$



$$w_0 + w_1 x_1 = 0 \Rightarrow \quad -1 + 1x_1 = 0$$

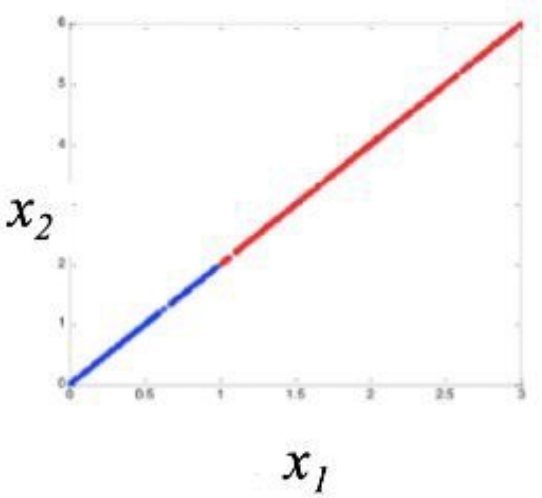
Is it convenient to use x2?



3.1 Introduction

Curse of Dimensionality

Is it convenient to use x2?



There is no interest in incorporating variables:

- Irrelevant
- Redundant

They make it difficult to design the classifier

3.1 Introduction. Figure of merit

Classification

Confusion matrix

		Real Class	
		Patients without heart attack	**Patients with heart attack**
Predicted class	**Patients without heart attack**	12 (TN)	3 (FN)
	Patients with heart attack	7 (FP)	5 (TP)

3.1 Introduction. Figure of merit

Accuracy. The proportion of correctly classified cases.

Sensitivity. The probability of correctly classifying a diseased individual.

Specificity. The probability of correctly classifying healthy individuals.

Precision. The percentage of positive predictions that were correct.

F1 score. The average between precision and recall (sensitivity)

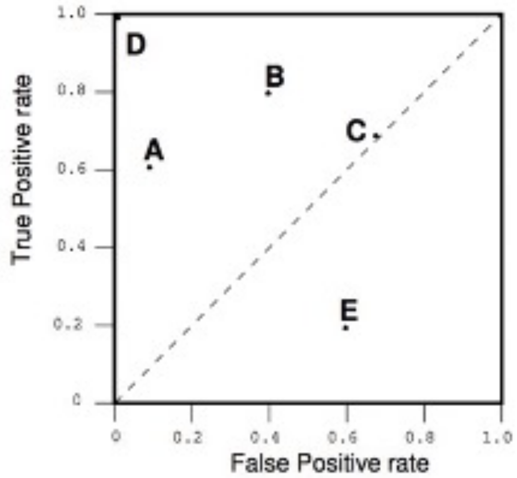
		Predicted Class: Positive	Predicted Class: Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	Negative Predictive Value $\frac{TN}{(TN+FN)}$	Accuracy $\frac{TP+TN}{(TP+TN+FP+FN)}$

3.1 Introduction. Figure of merit

ROC curve (receiver operating characteristic)

It is a graphical representation of the sensitivity (also called true positive rate) versus the one minus the specificity (also called false positive rate) for a binary classifier system as the discrimination threshold is varied.

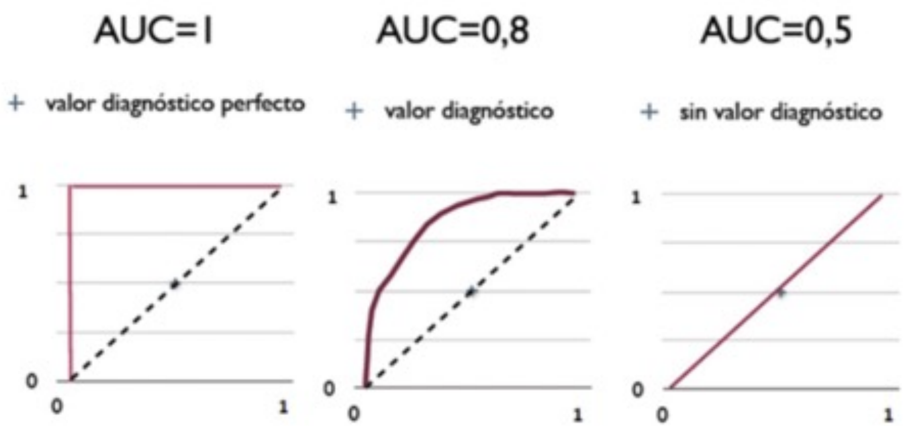
The false positive rate is plotted on the X-axis and the false positive rate on the Y-axis, and for each classifier is represented by a point on the ROC curve, given by the pair FP, TP.



- **Point D**, represents the perfect classification
- The diagonal presents the strategy of randomly guessing a class. A totally random classification would yield a point along the diagonal line

3.1 Introduction. Figure of merit

Area under the curve (AUC). It provides a measure of the aggregate (integral calculation) of performance at all possible classification thresholds. It is the two-dimensional area under the ROC curve. The AUC reflects how good the classifier is at discriminating, for example, patients with and without the disease over the full range of possible thresholds.



Contents

3.1 Introduction. Cost function. Metrics in classification.

3.2 Logistic regression

3.3 Logistic regression with regularisation

3.4 Biomedical examples and applications

3.2 Logistic regression

Sometimes the output (the dependent variable) is not a continuous variable.

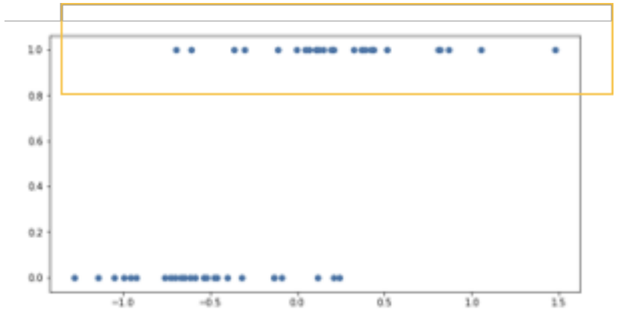
Examples:

- Knowing whether a patient is going to get cancer or not
- Knowing if a tumor exists or not
- Knowing whether or not a patient will suffer from diabetes, ...

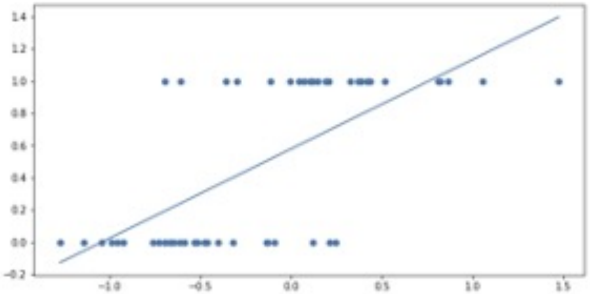
$$\left\{ \underline{\mathrm{x}}^{(\mathrm{k})}, y^{(\mathrm{k})} \right\}_{\mathrm{k}=1}^{\mathrm{K}}$$

In the above examples, the variable y can be represented as a binary variable with values {0,1}. Therefore, we could approach the problem as a supervised learning (binary classification).

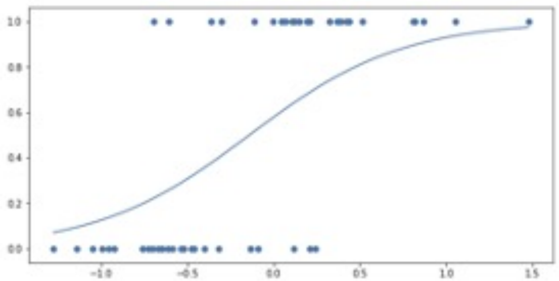
3.2 Logistic regression



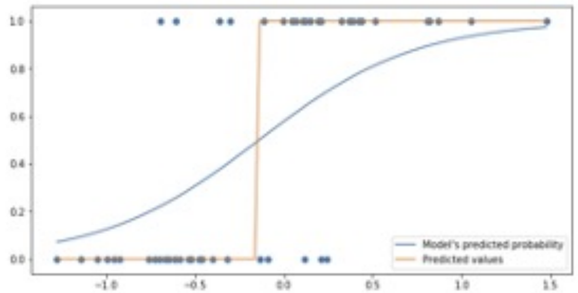
Input data



Linear regression



Logistic regression



Predicted values, predicted class

3.2 Logistic regression

- It is a (linear) method of classification.
- This is because the dependent variable to be predicted is discrete. We will assume that it only takes two values (0,1).
- If the y variable is binary, we can assume that the probability of the y variable being 1 is different for each observation:

$$y^{(k)} = \begin{cases} 1 & P(y^{(k)} = 1) = p^{(k)} \\ 0 & P(y^{(k)} = 0) = 1 - p^{(k)} \end{cases}$$

Generally speaking, the variable y can be separated into two components: g and u, so

$$y = g(w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_N x_N) + u = g(\mathbf{w}^T \mathbf{x}) + u$$

g is an invertible and differentiable function, generally non-linear

3.2 Logistic regression

Generally speaking, the variable y can be separated into two components: g and u, so

$$y = g(w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_N x_N) + u = g(\mathbf{w}^T \mathbf{x}) + u$$

g is an **invertible and differentiable** function, generally **non-linear**

In our case, it is important that the result of the g function is a number in the interval [0.1], so that the result can be interpreted as a probability.

To achieve this, an alternative is to use the **logistic function**, so that

$$\boxed{\hat{y}^{(k)} = p^{(k)} = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}^{(k)}}}} \rightarrow w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_N x_N$$

3.2 Logistic regression

$$\hat{y}^{(k)} = p^{(k)} = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}^{(k)}}}$$

$$g(\mathbf{w}^T \mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}}} \qquad g(z) = \frac{1}{1 + e^{-z}} \qquad g'(z) = g(z)(1 - g(z))$$

How can we find the coefficients?

$$\left. \begin{aligned} P(y = 1 | \mathbf{x}, \mathbf{w}) &= g(\mathbf{x}, \mathbf{w}) \\ P(y = 0 | \mathbf{x}, \mathbf{w}) &= 1 - g(\mathbf{x}, \mathbf{w}) \end{aligned} \right\} \qquad P(y | \mathbf{x}, \mathbf{w}) = (g(\mathbf{x}, \mathbf{w}))^y (1 - g(\mathbf{x}, \mathbf{w}))^{1-y}$$

Assuming that the k training examples were generated independently, we can write the likelihood (likelihood) of the parameters as.:

$$\boxed{\begin{aligned} L(\mathbf{w}) &= p(Y | X; \mathbf{w}) \\ &= \prod_{i=1}^{n} p(y^{(i)} | x^{(i)}; \mathbf{w}) \\ &= \prod_{i=1}^{n} g(\mathbf{x}^{(i)}, \mathbf{w})^y (1 - g(\mathbf{x}^{(i)}, \mathbf{w}))^{1-y} \end{aligned}}$$

3.2 Logistic regression

Función de coste

$$
\begin{aligned}
L(\mathbf{w}) &= p(Y|X;\mathbf{w}) \\
&= \prod_{i=1}^{n} p(y^{(i)}|x^{(i)};\mathbf{w}) \\
&= \prod_{i=1}^{n} g(\mathbf{x}^{(i)},\mathbf{w})^{y}(1-g(\mathbf{x}^{(i)},\mathbf{w}))^{1-y}
\end{aligned}
$$

It is easier to calculate the logarithm

Maximize probability

$$
\boxed{
\begin{aligned}
l(\mathbf{w}) &= log L(\mathbf{w}) \\
&= \sum_{i=1}^{n} y^{(i)} log\ g(\mathbf{x}^{(i)},\mathbf{w}) + (1-y^{(i)})\ log(1-g(\mathbf{x}^{(i)},\mathbf{w}))
\end{aligned}
}
$$

3.2 Logistic regression

Cost function

$$\boxed{\begin{aligned} l(\mathbf{w}) &= logL(\mathbf{w}) \\ &= \sum_{i=1}^{n} y^{(i)} log\ g(\mathbf{x}^{(i)}, \mathbf{w}) + (1 - y^{(i)})\ log(1 - g(\mathbf{x}^{(i)}, \mathbf{w})) \end{aligned}}$$

How can we find the coefficients?

Derivatives and gradient descent

$$\mathbf{w} := \mathbf{w} + \alpha \nabla_{\mathbf{w}} l(\mathbf{w})$$

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}_j} l(\mathbf{w}) &= \left(\mathbf{y} \frac{1}{g(\mathbf{w}^T\mathbf{x})} - (1 - \mathbf{y}) \frac{1}{1 - g(\mathbf{w}^T\mathbf{x})} \right) \frac{\partial}{\partial \mathbf{w}_j} g(\mathbf{w}^T\mathbf{x}) \\ &= (\mathbf{y} - g(\mathbf{w}^T\mathbf{x}))\mathbf{x}_j \end{aligned}$$

$$\boxed{w_j := w_j + \alpha(y^{(i)} - g(\mathbf{w}, x^{(i)}))x_j^{(i)}}$$

α: learning rate

0

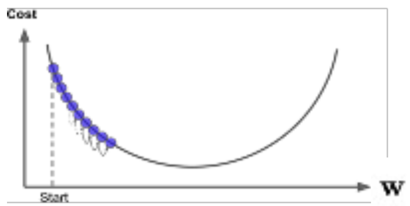
3.2 Logistic regression

Gradient descent

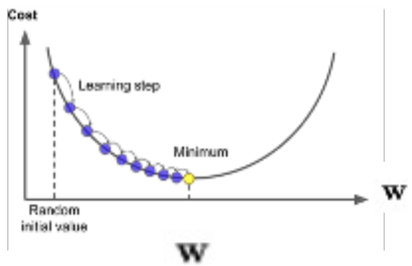
The general idea of gradient descent is to iteratively adjust parameters to minimize a cost function.

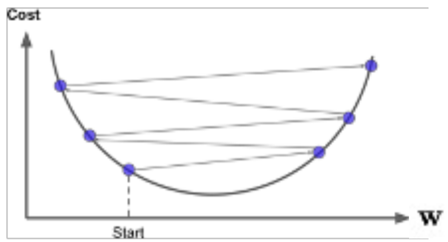
The local gradient of the cost function with respect to the parameter vector w (weights/coefficients) is calculated, and goes in the direction of the downward gradient. Once the gradient is zero, a minimum has been reached!

Concretely, one starts with a random initialization, and then gradually improves, each step trying to decrease the cost function (e.g., the MSE), until the algorithm converges to a minimum.



α small values





α high values

2.4 Logistic Regression

$$\hat{y}^{(k)} = p^{(k)} = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}^{(k)}}}$$

This relationship is also written as: $$logit(p^{(k)}) = log\left(\frac{p^{(k)}}{1 - p^{(k)}}\right) = w_o + w_1 x_1 + w_2 x_2 + \ldots + w_d x_d$$

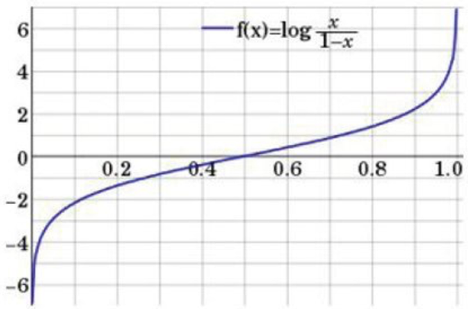
So the logit function is equal to the classic linear regressor. It is modeled as a linear combination of the independent variables x_i

The coefficients w_i are the parameters that are learned to build the model from the training cases

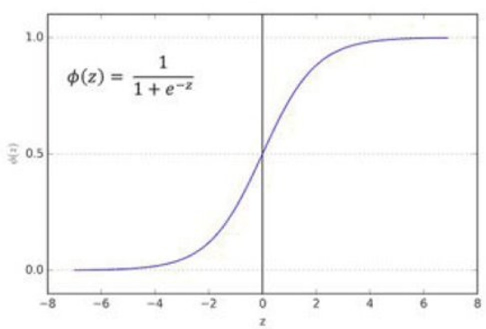
2.4 Logistic Regression

The logistic regression model consists of looking for a set of parameters to generate a boundary that allows the separation of classes.

In addition, the probability of each case to each class is obtained.



Función logit



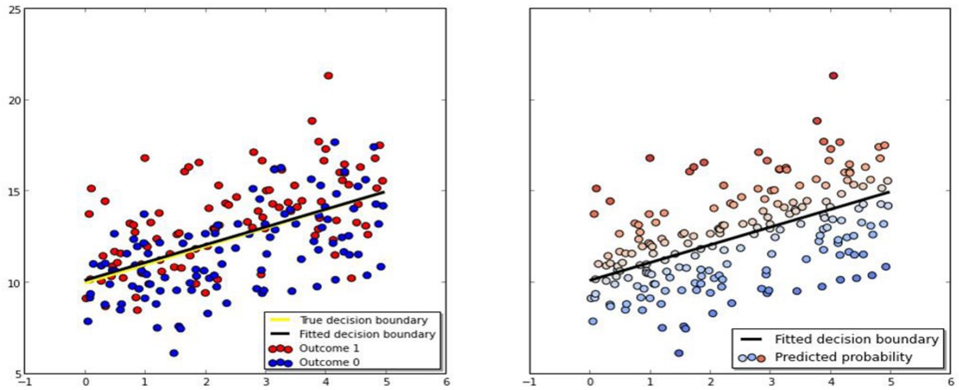
Función logística

To obtain a probability, which must necessarily be in the interval [0 1].

2.4 Logistic Regression

Probability of belonging to a class. The points near the border take on very low saturation colors, implying that the predicted probability of belonging to that class is approximately 0.5

The logistic regression predicts the probability of class membership ("**soft" output**).

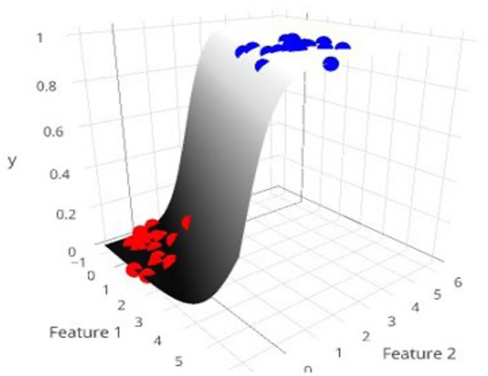


2.4 Logistic Regression

There is a set of training cases characterized by two variables (Features 1 and 2, x-axis), and the associated class (y-axis).

The logistic regression learns the parameters that best fit these cases (red and blue points).

For a new case belonging to the test set, the logistic regression model predicts the probability of belonging to a class.



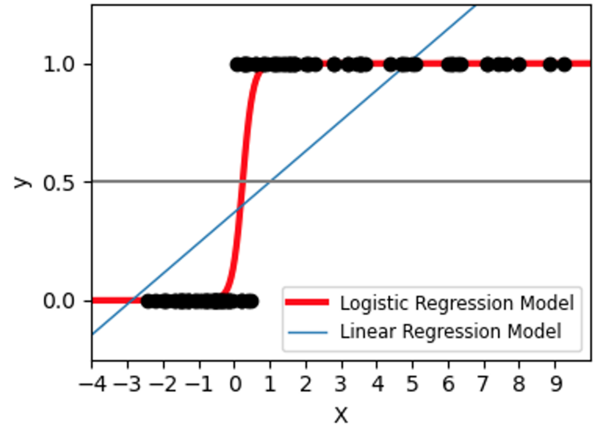
It has been widely used in the health field as it is possible to interpret the importance of the parameters.

2.4 Logistic Regression

```
from sklearn.linear_model import LogisticRegression
from sklearn import metrics

# Separate in train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
model= LogisticRegression()

# Fit/Build the model using training data
model.fit(X_train, y_train)

# Predict results on the test set
y_pred = logreg.predict(X_test)

#  Classification problem →
from sklearn.metrics import confusion_matrix
confusion_matrix = confusion_matrix(y_test, y_pred)
print(confusion_matrix)
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred)
```

2.4 Logistic Regression

```
print(__doc__)


# Code source: Gael Varoquaux
# License: BSD 3 clause

import numpy as np
import matplotlib.pyplot as plt

from sklearn import linear_model
from scipy.special import expit

# General a toy dataset:s it's just a straight line with some Gaussian noise:
xmin, xmax = -5, 5
n_samples = 100
np.random.seed(0)
X = np.random.normal(size=n_samples)
y = (X > 0).astype(np.float)
X[X > 0] *= 4
X += .3 * np.random.normal(size=n_samples)

X = X[:, np.newaxis]

# Fit the classifier
clf = linear_model.LogisticRegression(C=1e5)
clf.fit(X, y)

# and plot the result
plt.figure(1, figsize=(4, 3))
plt.clf()
plt.scatter(X.ravel(), y, color='black', zorder=20)
X_test = np.linspace(-5, 10, 300)

loss = expit(X_test * clf.coef_ + clf.intercept_).ravel()
plt.plot(X_test, loss, color='red', linewidth=3)

ols = linear_model.LinearRegression()
ols.fit(X, y)
plt.plot(X_test, ols.coef_ * X_test + ols.intercept_, linewidth=1)
plt.axhline(.5, color='.5')

plt.ylabel('y')
plt.xlabel('X')
plt.xticks(range(-5, 10))
plt.yticks([0, 0.5, 1])
plt.ylim(-.25, 1.25)
plt.xlim(-4, 10)
plt.legend(('Logistic Regression Model', 'Linear Regression Model'),
           loc="lower right", fontsize='small')
plt.tight_layout()
plt.show()
```



Contents

3.1 Introduction. Cost function. Metrics in classification.

3.2 Logistic regression

3.3 Logistic regression with regularization

3.4 Biomedical examples and applications

3.3 Logistic regression with regularization

Problem: Overfitting of the training data, especially when the data are high dimensional and the training data are sparse.

Solution: Reduce the overfitting by using regularization, i.e. penalize large values of w, using the penalized log likelihood function.

- The regularization is performed by adding to the function to be optimized during learning a penalty term (**regularization**) dependent on the model parameters and which allows to find a balance between a solution that does not overfit the design cases and provides the lowest possible error.

- The influence of the regularization term is controlled by weighting the penalty term by an adjustable multiplicative parameter (**λ, non-negative**) which is called the **regularization parameter.**

3.3 Logistic regression with regularization

The **regularization** term imposes some constraint on the solution (i.e., on the values of w), usually smoothness (models robust to noise in the data are of interest).

Intuitively:

- *If the value of λ is high*, the learning algorithm (algorithm used to find the w parameters defining the boundary) will pay more attention to constructing a smooth boundary than to minimizing the difference between the model output and the desired value.

- If most parameters are zero or near zero, the boundary will be smooth.

3.3 Logistic regression with regularization

Ridge

$$l(\mathbf{w}) = logL(\mathbf{w}) + \lambda||\mathbf{w}||^2$$
$$\sum_{i=1}^{n} y^{(i)} log\ g(\mathbf{x}^{(i)}, \mathbf{w}) + (1 - y^{(i)}) log(1 - g(\mathbf{x}^{(i)}, \mathbf{w})) + \frac{\lambda}{2}||\mathbf{w}||^2$$

$$w_j := w_j + \alpha[y^{(i)} - g(\mathbf{x}^{(i)}, \mathbf{w}))x_j^{(i)} + \lambda \mathbf{w}]$$

LASSO

$$l(\mathbf{w}) = logL(\mathbf{w}) + \lambda||\mathbf{w}||^1$$
$$\sum_{i=1}^{n} y^{(i)} log\ g(\mathbf{x}^{(i)}, \mathbf{w}) + (1 - y^{(i)}) log(1 - g(\mathbf{x}^{(i)}, \mathbf{w})) + \lambda||\mathbf{w}||^1$$

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Contents

3.1 Introduction. Cost function. Metrics in classification.

3.2 Logistic regression

3.3 Logistic regression with regularisation

3.4 Biomedical examples and applications

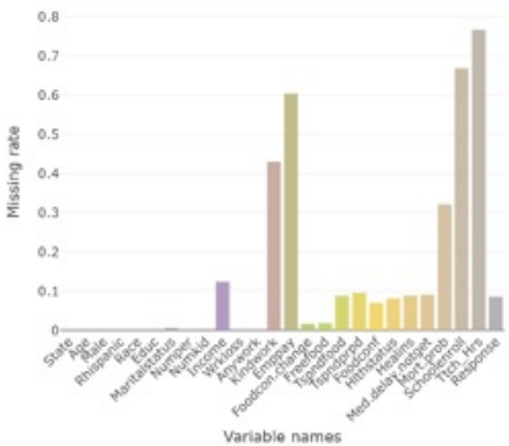
3.4 Biomedical examples and applications

Impact of COVID-19 on Mental Health: A Longitudinal Study Using Penalized Logistic Regression

Jingyu Cui[1], Jingwei Lu[1], Yijia Weng[1], Grace Y.Yi[1,2], and Wenqing He[1]

[1]*Department of Statistical and Actuarial Sciences, Western University, London, Ontario, Canada*
[2]*Department of Computer Science, Western University, London, Ontario, Canada*

Preprocessing



3.1 Multiple Imputation by Chained Equations (MICE)

The presence of missing values in the dataset brings in a challenge for data analysis and model fitting. Leaving out the observations with missing features would not be the best strategy, and it would eliminate potential valuable information from the dataset or even yield biased results. A useful approach to handle missing observations in a complex dataset is *multiple imputation by chained equations* (MICE), which invokes fully conditional specification (FCS) under the assumption of the missing at random (MAR) mechanism. Each incomplete variable is imputed by its own imputation model which generates plausible values to replace the missing ones. MICE can be used for various types of variables with missing values, such as binary, continuous, nominal, and ordinal data. Technical details can be found in van Buuren et al. (2015).

3.4 Biomedical examples and applications

Methods

4 Model Building and Inference

In this section, we employ the Lasso method to logistic regression to analyze the mental health data which contain a binary response and 25 predictors. For $i = 1, ..., n$, let Y_i represent the binary response with value 1 indicating that the mental health problem occurs for subject i and 0 otherwise. Let X_{ij} denote the jth covariate for subject i, where $j = 1, ..., p$, and p is the number of predictors. Write $X_i = (X_{i1}, X_{i2}, ..., X_{ip})^{\mathrm{T}}$ and let $\pi_i = P(Y_i = 1|X_i)$. Consider the logistic regression model

$$\text{logit } \pi_i = \beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j, \tag{1}$$

where $\beta = (\beta_0, \beta_1, ..., \beta_p)^{\mathrm{T}}$ denotes the vector of regression parameters.

The odds of the occurrence of mental health problems is defined by the ratio of the probability of having mental health problem happening to that of not having mental health issues i.e., $\frac{\pi_i}{1-\pi_i}$. The log-likelihood function for β is given by

$$\begin{aligned} l(\beta) &= \sum_{i=1}^{n} [Y_i \log \pi_i + (1 - Y_i)\log(1 - \pi_i)] \\ &= \sum_{i=1}^{n} [Y_i X_i^{\mathrm{T}}\beta - \log(1 + \exp(X_i^{\mathrm{T}}\beta))] . \end{aligned} \tag{2}$$

Since our objective is to select a subset of the predictors highly related to the dichotomous response, the Lasso method is used to do variable selection. The Lasso estimates are the values that maximize the penalized log-likelihood function, obtained by adding an L_1 penalty

$$l_\lambda^{lasso}(\beta) = \sum_{i=1}^{n} [Y_i X_i^{\mathrm{T}}\beta - \log(1 + \exp(X_i^{\mathrm{T}}\beta))] + \lambda\sum_{j=1}^{p} |\beta_j|, \tag{3}$$

where λ is the tuning parameter that controls the complexity of the model; variable selection is realized by tuning the value of λ.

A proper value of the tuning parameter λ is data-driven and can be chosen by K-fold cross-validation, with K being user specified. In our analysis below, K is chosen as 10. We use the "one-standard-error" rule (Hastie et al., 2009, p. 60) to pick the most parsimonious model within one standard error of the minimum cross-validation misclassification rate. This rule was also used by other authors, such as Krstajic et al. (2014).

3.4 Biomedical examples and applications

Results

any week. The *full model* includes all the 25 predictors in the original data, and the *reduced model* contains 11 predictors: *Age*, *Male*, *MS*, *Numkid*, *Wrkloss*, *Anywork*, *Foodconf*, *Hlthstatus*, *Healins*, *Med.delay.notget*, and *Mort.prob*. We expect the predictors in the *final model* to form a set in-between the sets of the predictors for the *reduced model* and the *full model*. Now, the problem is how to find the *final model* using the *reduced* and *full models*. To tackle this, we carry out the following steps.

In Step 1, we fit logistic regression with predictors in the *full model* and in the *reduced model*, respectively, to each of the five surrogate datasets for each of the 12 weeks. In Step 2, the estimates and standard errors of the model coefficients for a given week are obtained using the algorithm described by Allison (2000). To be specific, let M be the number of surrogate datasets for the original incomplete data, which is 5 in our analysis. Let β_j be the jth component of the model parameter vector β. For $k = 1, ..., M$, let $\hat{\beta}_j^{(k)}$ denote the estimate of the model parameter β_j obtained from fitting the kth surrogate dataset in a week and let $S_j^{(k)}$ be its associated standard error. Define

$$\hat{\beta}_j = \frac{1}{M}\sum_{k=1}^{M}\hat{\beta}_j^{(k)} \tag{4}$$

and

$$se(\hat{\beta}_j) = \sqrt{\frac{1}{M}\sum_{k=1}^{M}\left\{S_j^{(k)}\right\}^2 + \left(1+\frac{1}{M}\right)\left(\frac{1}{M-1}\right)\sum_{k=1}^{M}\{\hat{\beta}_j^{(k)} - \hat{\beta}_j\}^2}. \tag{5}$$

$$\begin{aligned}\text{logit}\ \pi = {} & \beta_0 + \beta_1 \times State.mild + \beta_2 \times State.moderate.daily + \beta_3 \times State.serious \\ & + \beta_4 \times Age + \beta_5 \times Male + \beta_6 \times Rhispanic + \beta_7 \times Race2 + \beta_8 \times Race3 \\ & + \beta_9 \times Race4 + \beta_{10} \times MS2 + \beta_{11} \times MS3 + \beta_{12} \times MS4 + \beta_{13} \times MS5 \\ & + \beta_{14} \times Nsmper + \beta_{15} \times Numkid + \beta_{16} \times Wrkloss + \beta_{17} \times Anywork \\ & + \beta_{18} \times Foodconf2 + \beta_{19} \times Foodconf3 + \beta_{20} \times Foddconf4 \\ & + \beta_{21} \times Hlthstatus2 + \beta_{22} \times Hlthstatus3 + \beta_{23} \times Hlthstatus4 \\ & + \beta_{24} \times Hlthstatus5 + \beta_{25} \times Healins + \beta_{26} \times Med.delay.notget \\ & + \beta_{27} \times Mort.prob + \beta_{28} \times Schoolenroll,\end{aligned} \tag{6}$$

3.4 Biomedical examples and applications

Results

Table 8: Coefficient estimates, standard errors and p-values from the *final model* for the 12 weeks

	Week 1			Week 2			Week 3			Week 4		
	Estimate	s.e.	p-value	Estimate	s.e.	p-value	Estimate	s.e.	p-value	Estimate	s.e.	p-value
(Intercept)	0.0745	0.0937	0.4269	0.0062	0.1303	0.9621	0.0719	0.0775	0.3534	0.1840	0.0894	0.0395
State.mild	-0.0948	0.0317	0.0028	-0.0853	0.0431	0.0476	-0.1068	0.0230	0.0000	-0.0992	0.0275	0.0003
State.moderate.daily	-0.0189	0.0455	0.6776	0.0568	0.0640	0.3751	-0.0877	0.0345	0.0109	-0.0395	0.0414	0.3401
State.serious	0.1880	0.0743	0.0114	0.1347	0.1017	0.1851	0.0106	0.0578	0.8539	-0.0384	0.0684	0.5739
Age	-0.0285	0.0010	0.0000	-0.0302	0.0014	0.0000	-0.0292	0.0008	0.0000	-0.0306	0.0009	0.0000
Male	-0.3032	0.0248	0.0000	-0.2164	0.0324	0.0000	-0.2841	0.0190	0.0000	-0.2605	0.0209	0.0000
Rhispanic	-0.1477	0.0473	0.0018	-0.1400	0.0576	0.0151	-0.1313	0.0324	0.0001	-0.1244	0.0368	0.0007
Race2	-0.4496	0.0388	0.0000	-0.3598	0.0576	0.0000	-0.4741	0.0308	0.0000	-0.5026	0.0391	0.0000
Race3	-0.2135	0.0633	0.0007	-0.2789	0.0798	0.0005	-0.2252	0.0453	0.0000	-0.3064	0.0520	0.0000
Race4	-0.0738	0.0492	0.1339	-0.0971	0.0676	0.1510	-0.0446	0.0436	0.3059	0.0648	0.0461	0.1600
MS2	0.1847	0.0618	0.0028	0.1846	0.0798	0.0206	0.1889	0.0535	0.0004	0.2839	0.0505	0.0000
MS3	0.2488	0.0321	0.0000	0.3095	0.0437	0.0000	0.2421	0.0272	0.0000	0.2452	0.0290	0.0000
MS4	0.2766	0.0727	0.0001	0.2879	0.1085	0.0080	0.2217	0.0592	0.0002	0.2455	0.0669	0.0002
MS5	0.1494	0.0311	0.0000	0.1280	0.0431	0.0030	0.1504	0.0260	0.0000	0.1973	0.0283	0.0000
Numper	-0.0258	0.0118	0.0292	0.0109	0.0156	0.4841	-0.0289	0.0083	0.0005	-0.0353	0.0090	0.0001
Numkid	-0.0982	0.0212	0.0000	-0.1429	0.0235	0.0000	-0.0851	0.0149	0.0000	-0.0852	0.0177	0.0000
Wrkloss	0.3284	0.0257	0.0000	0.3515	0.0341	0.0000	0.3771	0.0199	0.0000	0.3827	0.0211	0.0000
Anywork	-0.1002	0.0251	0.0001	-0.1333	0.0326	0.0000	-0.1570	0.0191	0.0000	-0.1274	0.0226	0.0000
Foodconf2	-0.6510	0.0423	0.0000	-0.5290	0.0594	0.0000	-0.6267	0.0323	0.0000	-0.6042	0.0447	0.0000
Foodconf3	-0.8663	0.0467	0.0000	-0.7455	0.0654	0.0000	-0.7910	0.0363	0.0000	-0.8647	0.0429	0.0000
Foodconf4	-1.2961	0.0485	0.0000	-1.3443	0.0702	0.0000	-1.3368	0.0368	0.0000	-1.4043	0.0452	0.0000
Hlthstatus2	0.2763	0.0359	0.0000	0.3281	0.0495	0.0000	0.3267	0.0275	0.0000	0.3026	0.0360	0.0000
Hlthstatus3	0.6821	0.0353	0.0000	0.7788	0.0497	0.0000	0.7618	0.0287	0.0000	0.7668	0.0322	0.0000
Hlthstatus4	1.3065	0.0416	0.0000	1.3194	0.0567	0.0000	1.3237	0.0330	0.0000	1.3837	0.0371	0.0000
Hlthstatus5	1.8343	0.0642	0.0000	1.9825	0.0907	0.0000	1.9328	0.0501	0.0000	2.0758	0.0554	0.0000
Realize	-0.0917	0.0412	0.0261	-0.1518	0.0580	0.0089	-0.1083	0.0376	0.0040	-0.1398	0.0426	0.0010
Med.delay.notget	0.6220	0.0230	0.0000	0.6846	0.0310	0.0000	0.6780	0.0189	0.0000	0.6701	0.0209	0.0000
Mort.prob	0.3047	0.0329	0.0000	0.1917	0.0429	0.0000	0.2413	0.0368	0.0000	0.2021	0.0289	0.0000
Schoolenroll	0.1038	0.0445	0.0197	0.0710	0.0575	0.2169	0.1259	0.0298	0.0000	0.1188	0.0414	0.0041

	Week 5			Week 6			Week 7			Week 8		
	Estimate	s.e.	p-value	Estimate	s.e.	p-value	Estimate	s.e.	p-value	Estimate	s.e.	p-value
(Intercept)	0.2264	0.0871	0.0093	0.2276	0.0994	0.0221	0.1793	0.0945	0.0578	0.2682	0.0780	0.0006
State.mild	-0.1829	0.0265	0.0000	-0.1780	0.0299	0.0000	-0.1617	0.0302	0.0000	-0.1389	0.0257	0.0000
State.moderate.daily	-0.0764	0.0421	0.0696	-0.0742	0.0426	0.0814	-0.0750	0.0432	0.0821	-0.0106	0.0361	0.7695
State.serious	-0.1611	0.0668	0.0159	0.0196	0.0738	0.7903	-0.1327	0.0751	0.0772	-0.0587	0.0649	0.3655
Age	-0.0319	0.0009	0.0000	-0.0332	0.0010	0.0000	-0.0297	0.0010	0.0000	-0.0304	0.0008	0.0000
Male	-0.2594	0.0237	0.0000	-0.2381	0.0230	0.0000	-0.1947	0.0244	0.0000	-0.2169	0.0195	0.0000
Rhispanic	-0.1896	0.0339	0.0000	-0.1964	0.0389	0.0000	-0.2460	0.0426	0.0000	-0.1740	0.0352	0.0000
Race2	-0.4204	0.0372	0.0000	-0.4429	0.0408	0.0000	-0.4059	0.0418	0.0000	-0.4090	0.0322	0.0000
Race3	-0.3617	0.0488	0.0000	-0.2974	0.0601	0.0000	-0.2757	0.0548	0.0000	-0.2746	0.0463	0.0000
Race4	-0.0788	0.0502	0.1163	-0.0035	0.0509	0.9445	-0.0900	0.0532	0.0907	-0.0157	0.0392	0.6888
MS2	0.2048	0.0526	0.0001	0.2008	0.0549	0.0003	0.1811	0.0579	0.0017	0.2141	0.0488	0.0000
MS3	0.2415	0.0284	0.0000	0.2726	0.0321	0.0000	0.2730	0.0321	0.0000	0.2101	0.0270	0.0000
MS4	0.1995	0.0651	0.0022	0.1742	0.0707	0.0137	0.3480	0.0747	0.0000	0.3125	0.0602	0.0000
MS5	0.2604	0.0271	0.0000	0.2559	0.0322	0.0000	0.2408	0.0323	0.0000	0.1582	0.0255	0.0000
Numper	-0.0357	0.0093	0.0001	-0.0284	0.0101	0.0050	-0.0136	0.0101	0.1772	-0.0393	0.0089	0.0000
Numkid	-0.0990	0.0168	0.0000	-0.1263	0.0175	0.0000	-0.1252	0.0207	0.0000	-0.1109	0.0140	0.0000
Wrkloss	0.3453	0.0214	0.0000	0.3415	0.0232	0.0000	0.3733	0.0247	0.0000	0.3333	0.0219	0.0000
Anywork	-0.1221	0.0223	0.0000	-0.1679	0.0259	0.0000	-0.1666	0.0262	0.0000	-0.1137	0.0200	0.0000
Foodconf2	-0.6544	0.0372	0.0000	-0.6494	0.0429	0.0000	-0.5702	0.0489	0.0000	-0.6591	0.0348	0.0000
Foodconf3	-0.9093	0.0404	0.0000	-0.8432	0.0479	0.0000	-0.8403	0.0506	0.0000	-0.8926	0.0375	0.0000
Foodconf4	-1.3966	0.0416	0.0000	-1.3154	0.0462	0.0000	-1.3294	0.0522	0.0000	-1.4082	0.0383	0.0000
Hlthstatus2	0.4160	0.0329	0.0000	0.3687	0.0359	0.0000	0.2648	0.0372	0.0000	0.3355	0.0334	0.0000
Hlthstatus3	0.8466	0.0325	0.0000	0.8493	0.0364	0.0000	0.7424	0.0372	0.0000	0.7957	0.0319	0.0000
Hlthstatus4	1.4557	0.0368	0.0000	1.4368	0.0426	0.0000	1.3498	0.0419	0.0000	1.4181	0.0348	0.0000
Hlthstatus5	2.1120	0.0602	0.0000	2.0741	0.0623	0.0000	2.0221	0.0643	0.0000	2.1172	0.0540	0.0000
Realize	-0.0575	0.0345	0.0958	-0.0571	0.0422	0.1764	-0.1245	0.0436	0.0043	-0.1013	0.0353	0.0041
Med.delay.notget	0.6661	0.0203	0.0000	0.7256	0.0220	0.0000	0.6495	0.0250	0.0000	0.6580	0.0187	0.0000
Mort.prob	0.2421	0.0323	0.0000	0.2322	0.0343	0.0000	0.2397	0.0389	0.0000	0.2095	0.0281	0.0000
Schoolenroll	0.1261	0.0325	0.0001	0.1857	0.0418	0.0000	0.1213	0.0595	0.0415	0.1358	0.0232	0.0000