Lecture 3 – Classification, logistic regression

UPPSALA
UNIVERSITET

David Sumpter
Division of Systems and Control
Department of Information Technology
Uppsala University.

Summary of lecture 2 (I/III)

Regression is about learning a model that describes the relationship between an input variable $\mathbf{x}$ and a numerical output variable y

$$y = f(\mathbf{x}; \boldsymbol{\theta}) + \varepsilon.$$

Linear regression is regression with a linear model

$$y = \underbrace{\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p}_{f(\mathbf{x};\boldsymbol{\theta})} + \epsilon.$$

How to learn/train/estimate θ?

Use the **maximum likelihood** principle: assume $\varepsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$ iid
$\Rightarrow$ **least squares** & **normal equations**

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\mathsf{argmin}} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2, \qquad \widehat{\boldsymbol{\theta}} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y},$$

$$\mathbf{x}_i = \begin{bmatrix} x_{i1} \\ \vdots \\ x_{ip} \end{bmatrix}, \qquad \mathbf{X} = \begin{bmatrix} 1 & -\mathbf{x}_1^\mathsf{T}- \\ \vdots & \vdots \\ 1 & -\mathbf{x}_n^\mathsf{T}- \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}.$$

Summary of lecture 2 (II/III)

We can make **arbitrary nonlinear transformations** of the inputs, for example polynomials

$$y = \theta_0 + \theta_1 \underset{\substack{\| \\ x_1}}{v} + \theta_2 \underset{\substack{\| \\ x_2}}{v^2} + \theta_3 \underset{\substack{\| \\ x_3}}{v^3} + \cdots + \theta_p \underset{\substack{\| \\ x_p}}{v^p} + \varepsilon$$

($v =$ original input variable, x_i transformed input variables or features)

Categorical input variables are handled by creating dummy variables.

Summary of lecture 2 (III/III)

Overfitting may occur when the model is **too flexible!**

Can be handled using **regularization**, which amounts to adding a term $R(\boldsymbol{\theta})$ to the cost function which controls the model flexibility,

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \underbrace{V(\boldsymbol{\theta}, \mathbf{X}, \mathbf{y})}_{\text{data fit}} + \lambda \underbrace{R(\boldsymbol{\theta})}_{\text{penalty}}$$

Ridge regression

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmin}} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 + \gamma \|\boldsymbol{\theta}\|_2^2$$

LASSO

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmin}} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 + \gamma \|\boldsymbol{\theta}\|_1$$

Where are we in the course?

Have a look at the webpage and say something about mini-project and exam.

Outline – Lecture 3

Aim: To introduce the classification problem and derive a first useful classifier: logistic regression.

Outline:

1. Summary of Lecture 2
1. The classification problem
2. Logistic regression
3. Example of diagnostic tools for classification
4. Maximum likelihood estimate

Categorical outputs

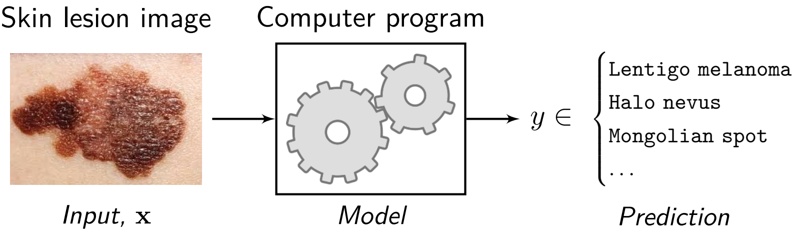
Many machine learning applications have **categorical outputs** y

- HiggsML: Separate $H \to \tau\tau$ decay from noise (see lecture 1):
  $y \in \{\texttt{signal}, \texttt{background}\}$
- Face verification:
  $y \in \{\texttt{match}, \texttt{no match}\}$
- Identify the spoken vowel from an audio signal:
  $y \in \{\texttt{A}, \texttt{E}, \texttt{I}, \texttt{O}, \texttt{U}, \texttt{Y}\}$
- Diagnosis system for leukemia:
  $y \in \{\texttt{ALL}, \texttt{AML}, \texttt{CLL}, \texttt{CML}, \texttt{no leukemia}\}$
- $\ldots$

The classification problem

Classification: learn a **model** which, for each input data point $\mathbf{x}$ can predict its **class** $y \in \{1, \ldots, M\}$.

ex) Classifying skin lesions



Skin lesion image

Computer program

$y \in$

`Lentigo melanoma`

`Halo nevus`

`Mongolian spot`

$\ldots$

Input, $\mathbf{x}$

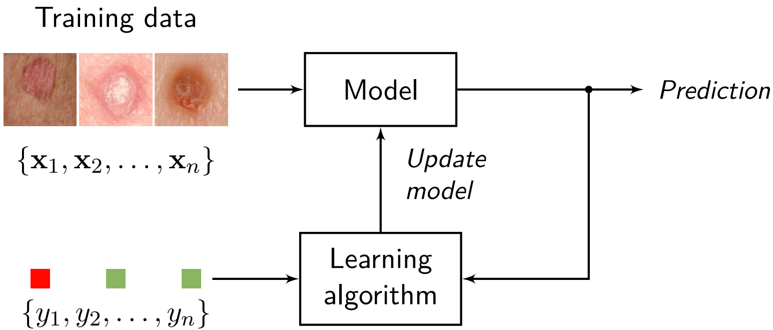
Model

Prediction

Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau and Sebastian Thrun. **Dermatologist-level classification of skin cancer with deep neural networks**. *Nature*, 542:115–118, 2017.

Training a classifier

Supervised learning: The model is **learned** by adapting it to labeled **training data** $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$.


Training data
$\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$
🟥 🟩 🟩
$\{y_1, y_2, \dots, y_n\}$
Model
Prediction
Update model
Learning algorithm

Classification

A **classification model** can be specified in terms of the **conditional class probabilities**

$$p(y = m \,|\, \mathbf{x}) \quad \text{for} \quad m = 1, \dots, M,$$

encoding the probability for class m given (i.e. conditioned on) that we know the input $\mathbf{x}$.

More specifically (for a binary classification task, $M = 2$, $y = 1$ or $y = -1$) we learn a model $f(\mathbf{x})$ that describes the conditional class probabilities $p(y \,|\, \mathbf{x})$ (= a **classifier**),

$$f(\mathbf{x}) = p(y = 1 \,|\, \mathbf{x}).$$

Classification – the multi-class case

We return a vector valued classifier $\boldsymbol{f}(\mathbf{x})$, where

$$\begin{bmatrix} p(y = 1 \,|\, \mathbf{x}) \\ p(y = 2 \,|\, \mathbf{x}) \\ \vdots \\ p(y = M \,|\, \mathbf{x}) \end{bmatrix} \quad \text{is modeled by} \quad \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_M(\mathbf{x}) \end{bmatrix} = \boldsymbol{f}(\mathbf{x}).$$

Each element $f_m(\mathbf{x})$ corresponds to the conditional class probability $p(y = m \,|\, \mathbf{x})$.

Why not linear regression?

Can we use linear regression for classification problems?

ex) Classifying e-mails as spam. Code the output as

$$y = \begin{cases} -1 & \text{if } \texttt{ham} \text{ (= good email)}, \\ 1 & \text{if } \texttt{spam}, \end{cases}$$

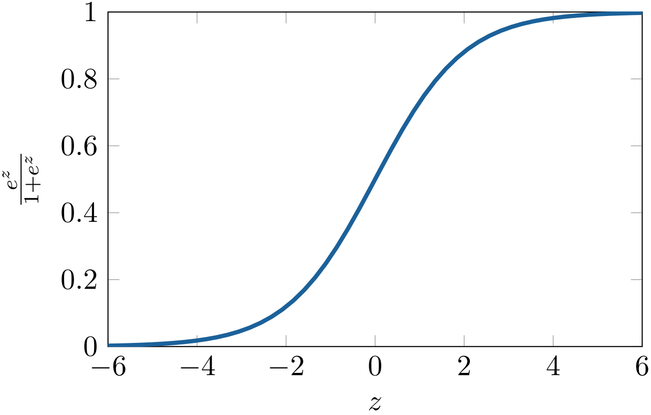
and learn a linear regression model. Classify as spam if $\hat{y} > 0$.

Why is this not a good idea?

- ▼ Sensitive to unequally sized classes in the training data.
- ▼ Difficult to generalize to $M > 2$ classes.
- ▼ Doesn't correspond nicely to probabilities.

Logistic function (aka sigmoid function)

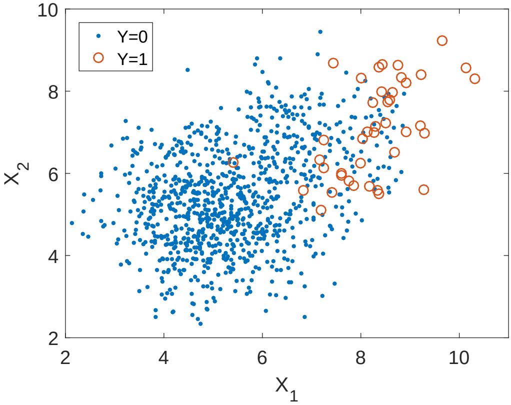
The function $f : \mathbb{R} \mapsto [0, 1]$ defined as $f(z) = \frac{e^z}{1+e^z}$ is known as the **logistic function**.



ex) Logistic regression

Consider a (made-up) problem where we want to build a classifier for whether a person has a certain disease $(y = 1)$ or not $(y = -1)$ based on two biological indicators x_1 and x_2.

The training data consists of $n = 1,000$ labeled samples (right).



ex) Logistic regression

A logistic regression model

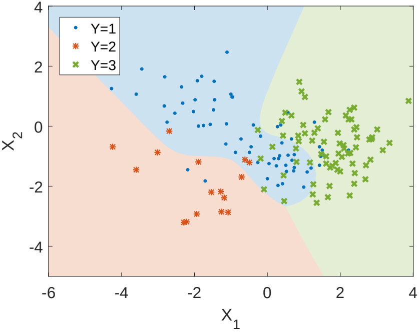
$$p(y = 1 \mid \mathbf{x}) = \frac{e^{\theta_0 + \theta_1 x_1 + \theta_2 x_2}}{1 + e^{\theta_0 + \theta_1 x_1 + \theta_2 x_2}}$$

is learned using maximum likelihood (we will come back to how we do this later in the lecture).

The parameters are found to be: i.e., $\widehat{\beta} = (-17.6, 1.81, 0.277)^{\mathsf{T}}$.

Decision boundaries

The input space can be segmented into M regions, separated by so-called **decision boundaries**.



Finding the decision boundary

The **decision boundary** is found by solving the equation

$$f(x) = 1 - f(x)$$

For logistic regression this corresponds to

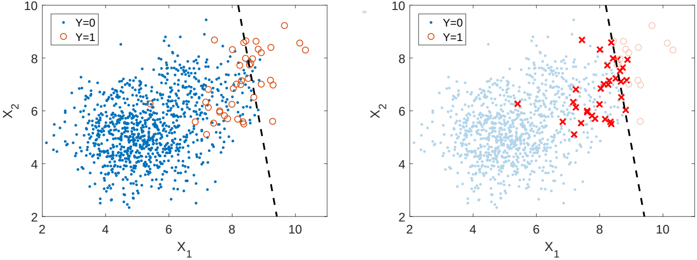
$$\frac{e^{\boldsymbol{\theta}^\mathsf{T}\mathbf{x}}}{1 + e^{\boldsymbol{\theta}^\mathsf{T}\mathbf{x}}} = \frac{1}{1 + e^{\boldsymbol{\theta}^\mathsf{T}\mathbf{x}}}$$

which we can write as $e^{\boldsymbol{\theta}^\mathsf{T}\mathbf{x}} = 1$. Hence, we have the following expression for the decision boundary

$$\boldsymbol{\theta}^\mathsf{T}\mathbf{x} = 0.$$

Linear expression for the decision boundary for logistic regression.

ex) Logistic regression: training error



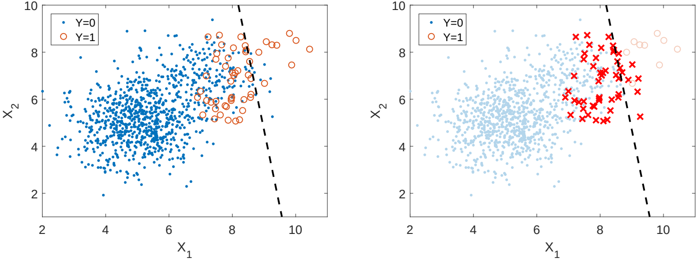
In right hand figure: cross is a mistake and circle or dot is correct. Dashed line is decision boundary.

Training error:

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(\hat{y}_i \neq y_i) = 3.3\%$$

ex) Logistic regression: test error

To further test the classifier we evaluate it on ***previously unseen*** **test data**: $\{(x'_i, y'_i)\}_{i=1}^{n_t}$



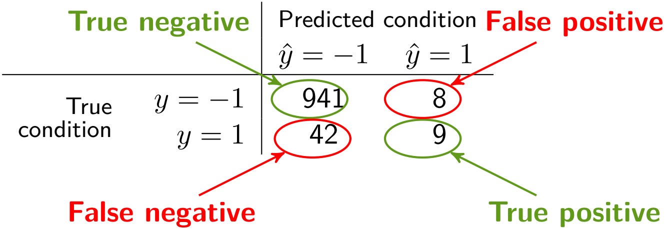
(Estimated) error on the test data:

$$\frac{1}{n_t}\sum_{i=1}^{n_t}\mathbb{I}(\hat{y}'_i \neq y'_i) = 5.0\%$$

Error on the test data for the naive classifier $\hat{y} \equiv 0$: 5.1%.

ex) Logistic regression: confusion matrix

Instead of just looking at the misclassification error it is better to compute the so-called **confusion matrix**.



True negative

False positive

		Predicted condition	
		$\hat{y} = -1$	$\hat{y} = 1$
True condition	$y = -1$	941	8
	$y = 1$	42	9

False negative

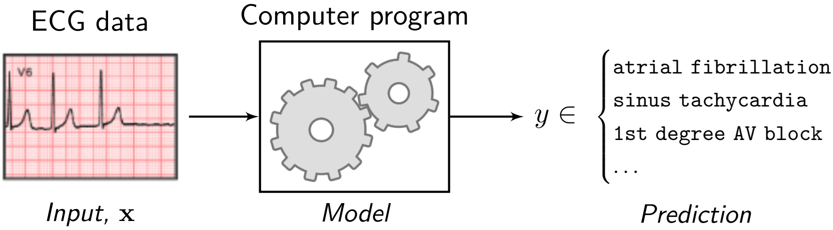
True positive

> Out of 51 patients affected by the disease, only 9 are correctly classified!

The **True Positive Rate (TPR)** is just $9/51 \approx 17.7\%$

A classification problem from our research

Aim: Automatic classification of Electrocardiography (ECG) data.



ECG data

Computer program

$$y \in \begin{cases} \texttt{atrial fibrillation} \\ \texttt{sinus tachycardia} \\ \texttt{1st degree AV block} \\ \dots \end{cases}$$

Input, $\mathbf{x}$

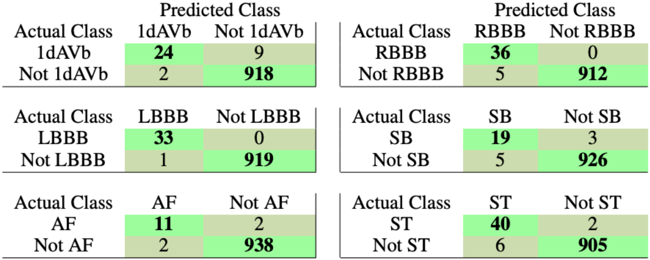
Model

Prediction

We are now at human level (medical doctors) performance.

Antonio H. Ribeiro, Manoel Horta, Gabriela Paixao, Derick Oliveira, Paulo R. Gomes, Jessica A. Canazart, Milton Pifano, Wagner Meira Jr., Thomas B. Schön and Antonio Luiz Ribeiro. **Automatic diagnosis of short-duration 12-lead ECG using a deep convolutional network**. In *Nature Communications*, 2020. To appear.

Confusion matrices for ECG classification



	Predicted Class	
Actual Class	1dAVb	Not 1dAVb
1dAVb	**24**	9
Not 1dAVb	2	**918**

	Predicted Class	
Actual Class	RBBB	Not RBBB
RBBB	**36**	0
Not RBBB	5	**912**

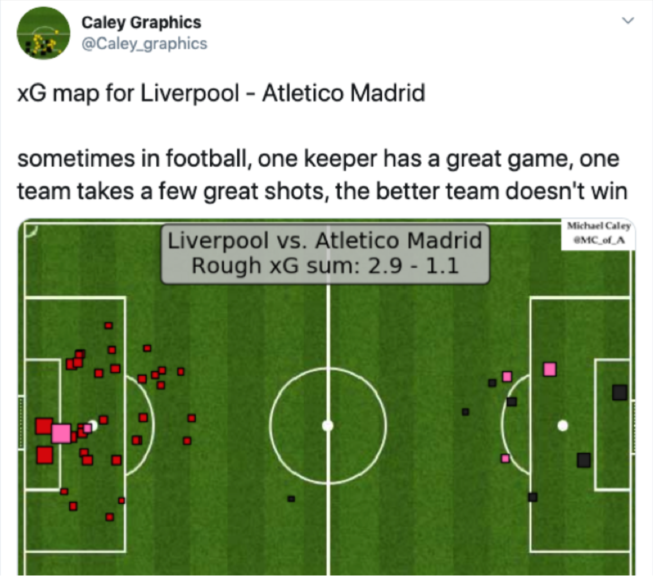
Actual Class	LBBB	Not LBBB
LBBB	**33**	0
Not LBBB	1	**919**

Actual Class	SB	Not SB
SB	**19**	3
Not SB	5	**926**

Actual Class	AF	Not AF
AF	**11**	2
Not AF	2	**938**

Actual Class	ST	Not ST
ST	**40**	2
Not ST	6	**905**

Expected Goals in Football

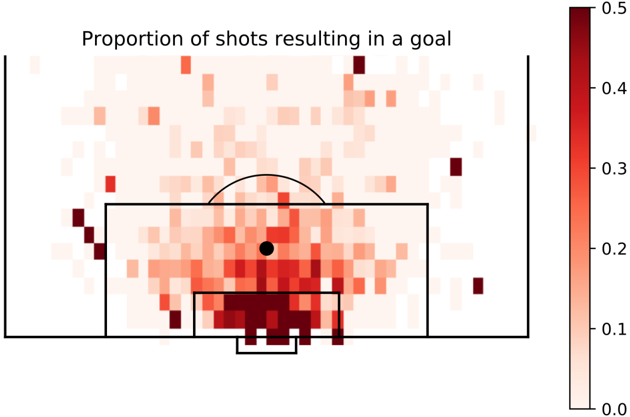


Caley Graphics
@Caley_graphics

xG map for Liverpool - Atletico Madrid

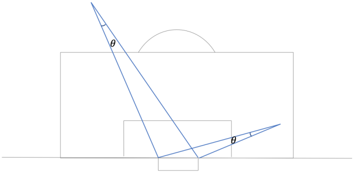
sometimes in football, one keeper has a great game, one team takes a few great shots, the better team doesn't win

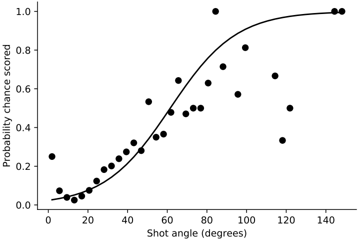
Expected Goals in Football



Expected Goals in Football

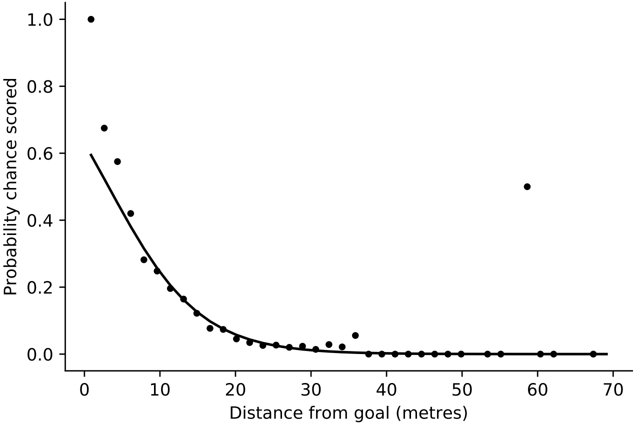
Goal angle is a good predictor of success:





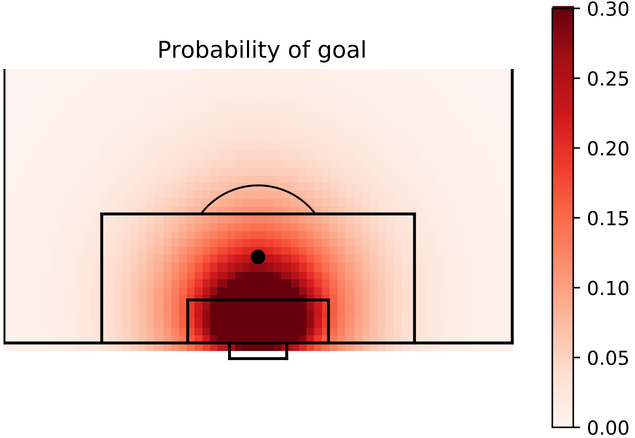
Expected Goals in Football

Distance is also a good predictor of success:



Expected Goals in Football

Distance and goal angle (along with non-linear combinations) combined:



Conservative predictions

Common to start thinking about a classifier minimizing the total misclassification error.

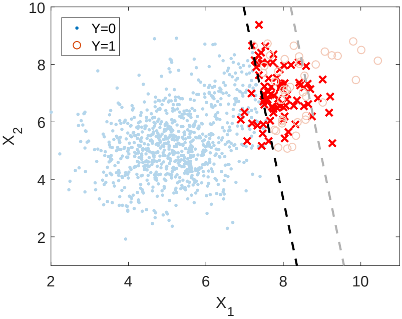
What if false negatives are "worse" than false positives?

Idea: Modify the prediction model

$$\hat{y}(\mathbf{x}_\star) = \begin{cases} 1 & \text{if } f(\mathbf{x}_\star) > r, \\ -1 & \text{otherwise,} \end{cases}$$

where $0 \leq r \leq 1$ is a user chosen threshold.

ex) Logistic regression, cont'd



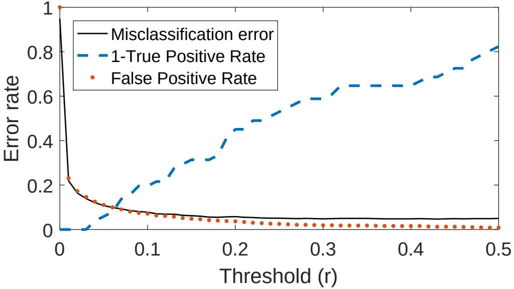
	$\hat{y} = -1$	$\hat{y} = 1$
$y = -1$	881	68
$y = 1$	10	41

Table: Confusion matrix ($r = 0.2$)

If we set the threshold at $r = 0.2$,

- ▲ The ***true positive rate*** is increased to $41/51 = 80.4\%$
- ▼ However, the ***misclassification error*** is increased to 7.8%

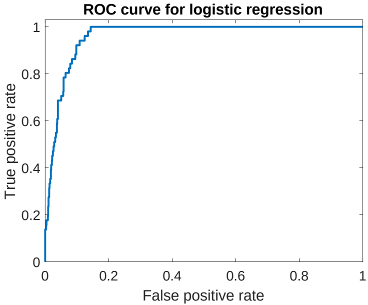
ex) Logistic regression: error rates



As we increase the threshold r from 0 to 0.5:

- ▲ The ***misclassification error*** decreases
- ▲ The number of ***non-diseased persons incorrectly classified as diseased*** **(False Positive Rate)** decreases.
- ▼ The number of ***diseased persons incorrectly classified as non-diseased*** **(1 − True Positive Rate)** increases!

ex) Logistic regression: ROC and AUC



- **ROC[1] curve**: plot of TPR vs. FPR.
- **Area Under Curve (AUC):** condensed performance measure for the classifier, taking all possible thresholds into account.
- AUC $\in [0, 1]$ where AUC $= 0.5$ corresponds to a random guess. [***ex)*** AUC $= 0.96$]

[1] For *Receiver Operating Characteristics*, which is a historical name.

Maximum likelihood estimation of parameters

Lets look in a bit more detail at the logistic function (for one input but result generalisable).

$$p = p(y = 1 \,|\, \mathbf{x}, \theta) = \frac{e^{\theta_0 + \theta_1 x_1}}{1 + e^{\theta_0 + \theta_1 x_1}}$$

We can rewrite

$$\begin{aligned}
\left(1 + e^{\theta_0 + \theta_1 x_1}\right) p &= e^{\theta_0 + \theta_1 x_1} \\
e^{\theta_0 + \theta_1 x_1} p &= 1 - p \\
\theta_0 + \theta_1 x_1 &= \log(\frac{1 - p}{p})
\end{aligned}$$

which is a linear relationship!
But we can't put observations in to equation (i.e. set $p_i = y$).
Because get $\log(0) = -\infty$ and $\log(\infty) = \infty$

UPPSALA UNIVERSITET

Maximum likelihood estimation of parameters

So instead we use method from lecture 1. The likelihood (probability of the data given the model).

$$l(\theta) = \prod_{i=1}^{n} p(\hat{y}_i = y_i \,|\, \mathbf{x}_i, \theta)$$

By taking the logarithm we get:

$$J(\theta) = \sum_{i=1}^{n} \log \left(p(\hat{y}_i = y_i \,|\, \mathbf{x}_i, \theta) \right)$$

Maximum likelihood estimation of parameters

For example, imagine 4 people asked about if they like gerkins $y = (1, -1, 1, -1)$ and they have have ages $x = (45, 23, 65, 12)$. We try θ_0=0 and θ_1=0.01. Then

$$\begin{aligned} l(\theta) &= \left(\frac{e^{0.45}}{1+e^{0.45}}\right)\left(\frac{1}{1+e^{0.23}}\right)\left(\frac{e^{0.65}}{1+e^{0.65}}\right)\left(\frac{1}{1+e^{0.12}}\right) \\ &= 0.61 \cdot 0.44 \cdot 0.65 \cdot 0.47 = 0.081 \end{aligned}$$

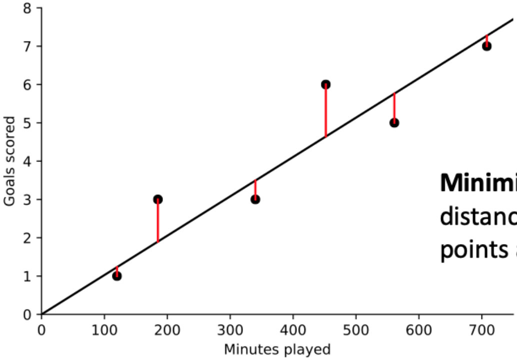
and

$$J(\theta) = \log(0.61) + \log(0.44) + \log(0.65) + \log(0.47) = -2.501$$

To find $\hat{\theta_0}$ and $\hat{\theta_1}$ we need to have a way of maximising $J(\theta)$.

Maximum likelihood estimation of parameters

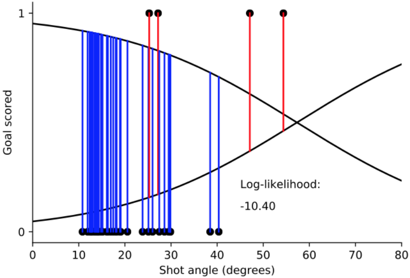
Linear regression



Minimize sum of distance between points and line.

Maximum likelihood estimation of parameters

Logistic regression



Maximize product of distances between outcome and prediction.

Equivalently **maximize sum** of log of distance between points and line. (Loglikelihood)

Maximum likelihood estimation of parameters

In general, when our $y_i \in -1, 1$, then we want to find:

$$J(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} -\ln \frac{e^{y_i(\theta_0+\theta_1 x_i)}}{1+e^{y_i(\theta_0+\theta_1 x_i)}} = \frac{1}{n}\sum_{i=1}^{n} \underbrace{\ln\left(1+e^{y_i(\theta_0+\theta_1 x_i)}\right)}_{\text{Logistic loss } L(x_i, y_i, \boldsymbol{\theta})}$$

where $J(\boldsymbol{\theta})$ is known as the cost function. This is for $k = 1$ input parameter, but same applies for $k > 1$.

We cannot find the maximum directly, but we can find the gradient

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = -\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{1+e^{y_i \boldsymbol{\theta}^T \mathbf{x}_i}}\right) y_i \mathbf{x}_i$$

which describes the direction in which $J(\boldsymbol{\theta})$ increases.
See text book chapter 5.2 for details.

A few concepts to summarize lecture 3

Classification: The problem of modeling a relationship between an (arbitrary) input $\mathbf{x}$ and a categorical output y.

Decision boundaries: Points in the input space where the classifier $\hat{y}(\mathbf{x})$ changes value.

Logistic regression: A model that uses a logistic function to model a binary output variable.

Confusion matrix: Table with predicted vs true class labels (for binary classification ⇒ number of true negatives, true positives, false negatives, and false positives).

Maximum likelihood: Find the probability of the model given the data. Can be found by gradient descen).