Lecture 3 – Classification, logistic regression

UPPSALA
UNIVERSITET

Thomas Schön
Division of Systems and Control
Department of Information Technology
Uppsala University.

Email: thomas.schon@it.uu.se

Summary of lecture 2 (I/III)

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

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

Linear regression is regression with a linear model

$$y = \underbrace{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p}_{f(\mathbf{x};\boldsymbol{\beta})} + \epsilon.$$

How to learn/train/estimate β?

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

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

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

Summary of lecture 2 (II/III)

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

$$y = \beta_0 + \beta_1 \underset{\substack{\| \\ x_1}}{v} + \beta_2 \underset{\substack{\| \\ x_2}}{v^2} + \beta_3 \underset{\substack{\| \\ x_3}}{v^3} + \cdots + \beta_p \underset{\substack{\| \\ x_p}}{v^p} + \varepsilon$$

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

Qualitative 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 $R(\boldsymbol{\beta})$ to the cost function which controls the model flexibility,

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

Ridge regression

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

LASSO

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

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. Bayes' classifier — *the optimal classifier w.r.t. minimizing misclassification error*
4. Diagnostic tools for classification (via example)

Qualitative outputs

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

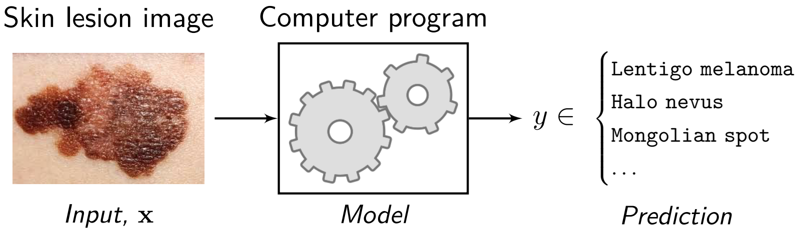
- HiggsML: Separate $H \to \tau\tau$ decay from noise (see lecture 1): $y \in \{\texttt{signal}, \texttt{background}\}$
- Face verification: $y \in \{\texttt{match}, \texttt{no match}\}$
- Identify the spoken vowel from an audio signal: $y \in \{\texttt{A}, \texttt{E}, \texttt{I}, \texttt{O}, \texttt{U},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

UPPSALA UNIVERSITET

The classification problem

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

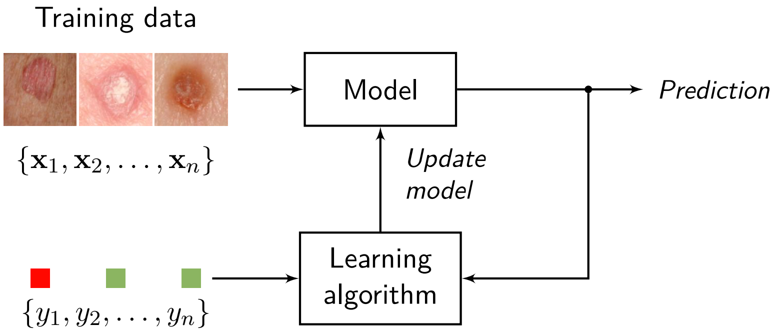
ex) Classifying skin lesions



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** $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$.



Classification

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

$$\Pr(y = k \mid \mathbf{x}) \quad \text{for} \quad k = 1, \dots, K.$$

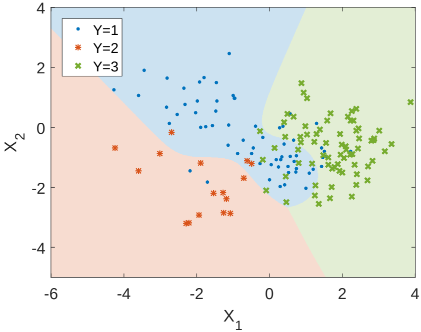
A **prediction model** for classification (= a **classifier**) is a function $\hat{g}$ that maps an input to a class

$$\hat{y} = \hat{g}(\mathbf{x}) \quad \text{where} \quad \hat{y} \in \{1, \dots, K\}.$$

Decision boundaries

The prediction model $\hat{y} = \hat{g}(\mathbf{x})$ is such that $\hat{y} \in \{1, \ldots, K\}$.

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



Why not linear regression?

Can we use linear regression for classification problems?

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

$$y = \begin{cases} 0 & \text{if } \texttt{ham} \text{ (= good email)}, \\ 1 & \text{if } \texttt{spam}, \end{cases}$$

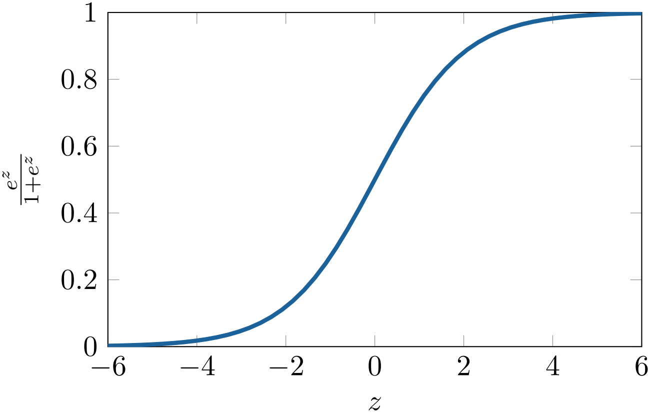
and learn a linear regression model. Classify as spam if $\hat{y} > 0.5$.

Why is this not a good idea?

- $\hat{y}$ can be viewed as an estimate of $\Pr(y = \texttt{spam} \mid \mathbf{x})$. However, there is no guarantee that $\hat{y} \in [0, 1]$, making it hard to interpret as a probability.
- Sensitive to unequally sized classes in the training data.
- Difficult to generalize to $K > 2$ classes.

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**.



Finding the decision boundary

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

$$\Pr(y = 1 \,|\, \mathbf{x}) = \Pr(y = 0 \,|\, \mathbf{x}).$$

For logistic regression this corresponds to

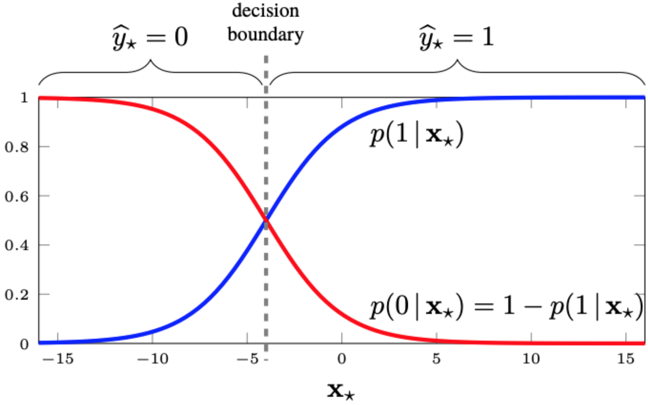
$$\frac{e^{\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x}}}{1 + e^{\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x}}} = \frac{1}{1 + e^{\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x}}}$$

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

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

Linear expression for the decision boundary!

Decision boundary – logistic regression



To turn the modeled probabilities $\Pr(y = 1 \mid \mathbf{x}_\star)$ into actual class predictions (i.e. $\hat{y}_\star$ is either 0 or 1), the class which is modeled to have the highest probability is taken as the prediction.

Bayes' classifier

The **optimal classifier**—in terms of minimizing the **misclassification test error**—is the one which assigns each prediction to the most likely class, given its input value.

That is, the predicted output for input $\mathbf{x}$ is the class k for which

$$\Pr(y = k \mid \mathbf{x})$$

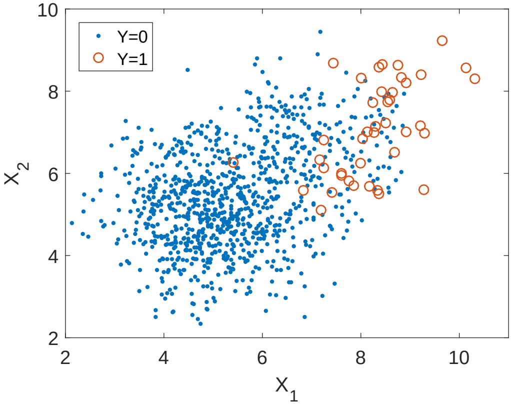
is largest. This is referred to as **Bayes' classifier**.

In practice we do not know the conditional probability of the class y given the input $\mathbf{x}$. Practical classification methods typically try to ***approximate*** Bayes' classifier as well as possible.

ex) Logistic regression

Consider a *toyish* problem where we want to build a classifier for whether a person has a certain disease ($y = 1$) or not ($y = 0$) based on two biological indicators x_1 and x_2.

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



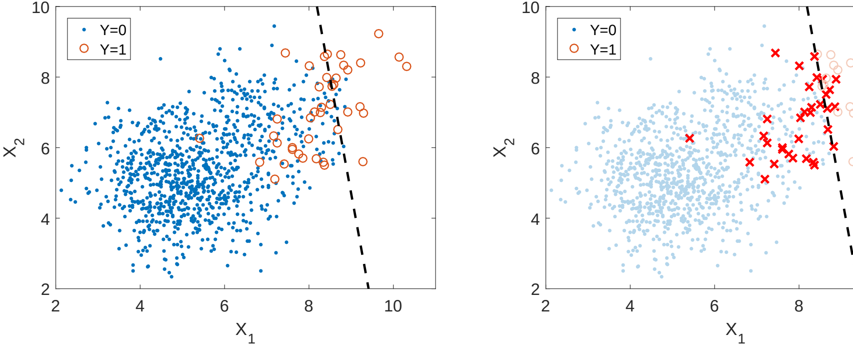
ex) Logistic regression

A logistic regression model

$$\Pr(y = 1 \,|\, \mathbf{x}) = \frac{e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2}}{1 + e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2}}$$

is learned using maximum likelihood.

ex) Logistic regression: training error



Training error:

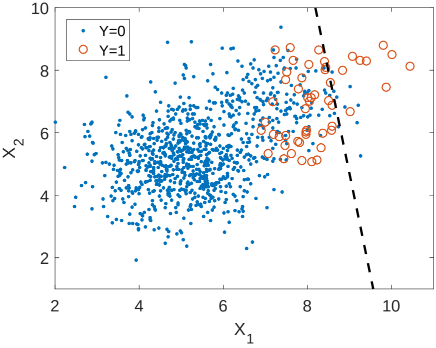
$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(\hat{g}(x_i) \neq y_i) = 3.3\%$$

UPPSALA
UNIVERSITET

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) test error:

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

The naive classifier $\hat{g}(x) \equiv 0$ attains a test error of 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	
		$\hat{y} = 0$	$\hat{y} = 1$
True condition	$y = 0$	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 $9/51 \approx 17.7\%$

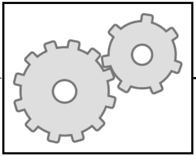
UPPSALA UNIVERSITET

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 *ML4H: Machine Learning for Health, workshop at NeurIPS*, Montréal, Canada, December 2018.

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**

Conservative predictions

The Bayes classifier minimizes the total misclassification error!

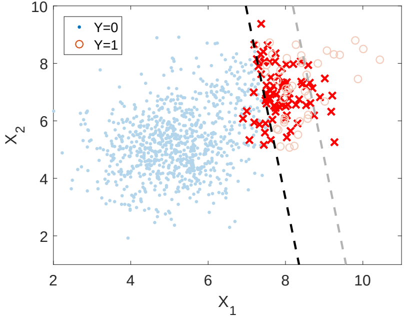
What if false negatives are "worse" than false positives?

Idea: Modify the prediction model:

$$\hat{g}(\mathbf{x}) = \begin{cases} 1 & \text{if } \Pr(y = 1 \,|\, \mathbf{x}) > r, \\ 0 & \text{otherwise,} \end{cases}$$

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

ex) Logistic regression, cont'd



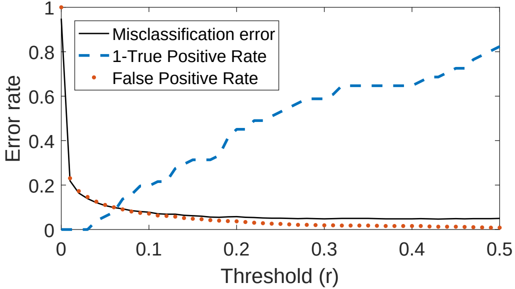
	$\hat{y}=0$	$\hat{y}=1$
$y=0$	881	68
$y=1$	10	41

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%

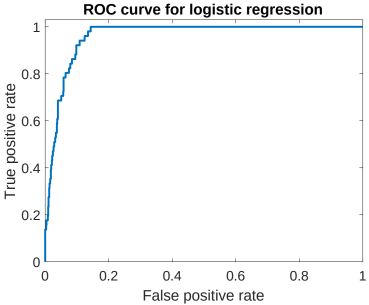
ex) Logistic regression: error rates



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.

A few concepts to summarize lecture 3

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

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

Bayes classifier: The optimal classifier w.r.t. minimizing misclassification error.

Logistic regression: Models the Bayes classifier using a linear regression model for the log-odds.

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