



SDU Summer School

Deep Learning

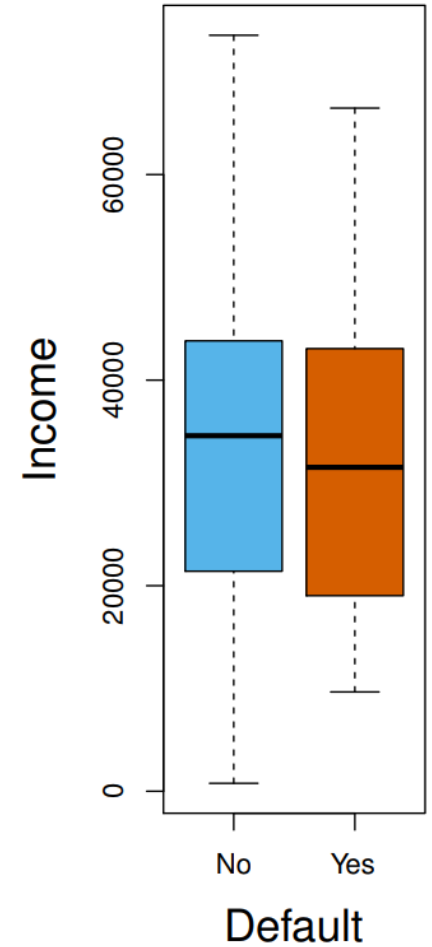
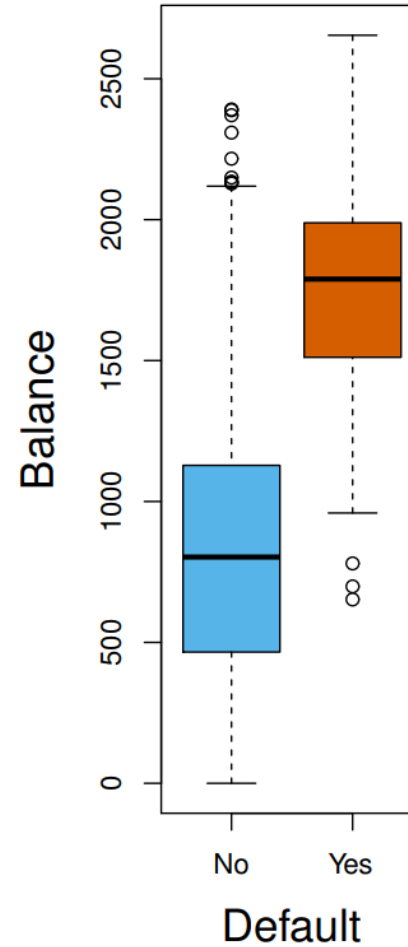
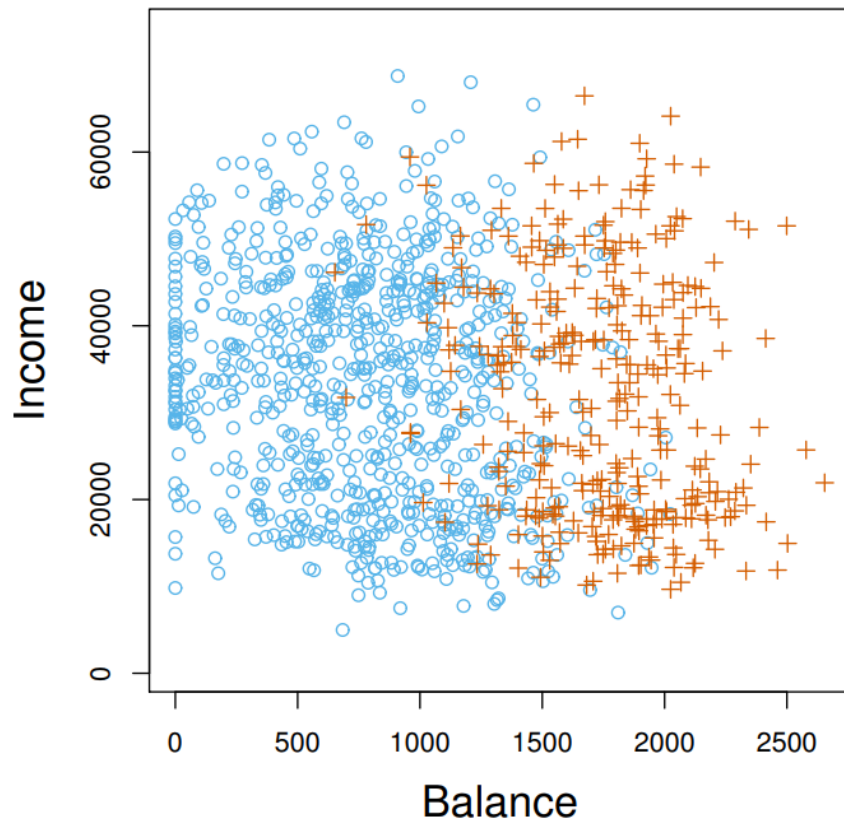
Summer 2018

Logistic Regression

Classification

- Qualitative variables take values in an unordered set C , such as:
 - eye color $\in \{\text{brown, blue, green}\}$
 - email $\in \{\text{spam, ham}\}$.
- Given a feature vector X and a qualitative response Y taking values in the set C , the classification task is to build a function $C(X)$ that takes as input the feature vector X and predicts its value for Y ; i.e. $C(X) \in C$.
- Often we are more interested in estimating the probabilities that X belongs to each category in C .
 - For example, it is more valuable to have an estimate of the probability that an insurance claim is fraudulent, than a classification fraudulent or not.

Example: Credit Card Default



Can we use Linear Regression?

- Consider the a binary classification task

$$f(x) = \begin{cases} 0, & \text{is No} \\ 1, & \text{if Yes} \end{cases}$$

- Can we simply perform a linear regression of Y on X and classify as Yes if $\hat{Y} > 0.5$?

Can we use Linear Regression?

- Consider the a binary classification task

$$f(x) = \begin{cases} 0, & \text{is No} \\ 1, & \text{if Yes} \end{cases}$$

- Can we simply perform a linear regression of Y on X and classify as Yes if $\hat{Y} > 0.5$?
 - In this case of a binary outcome, linear regression does a good job as a classifier, and is equivalent to linear discriminant analysis
 - Since in the population $E(Y|X = x) = \Pr(Y = 1|X = x)$, we might think that regression is perfect for this task.

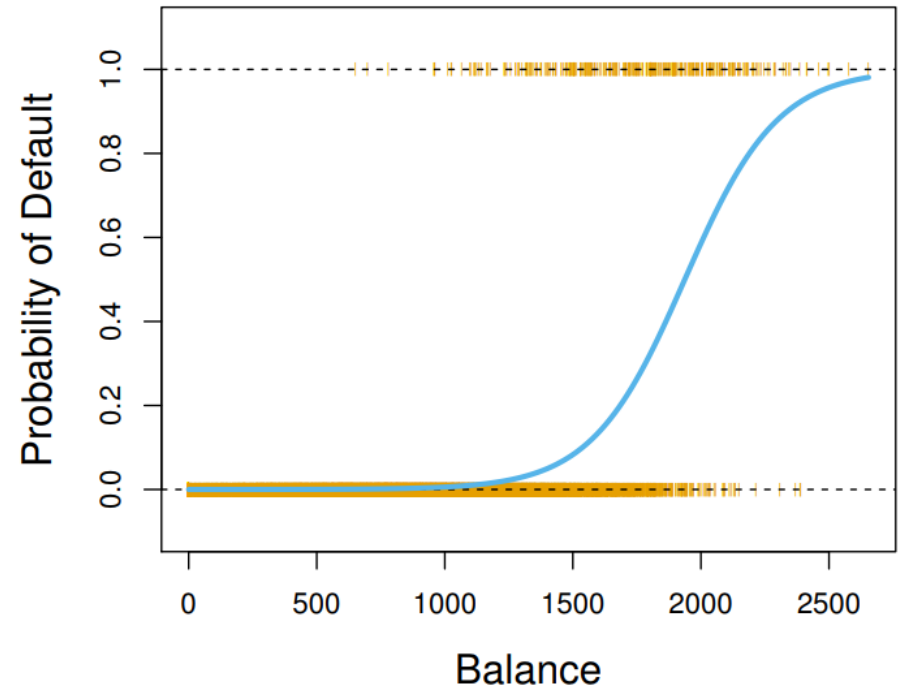
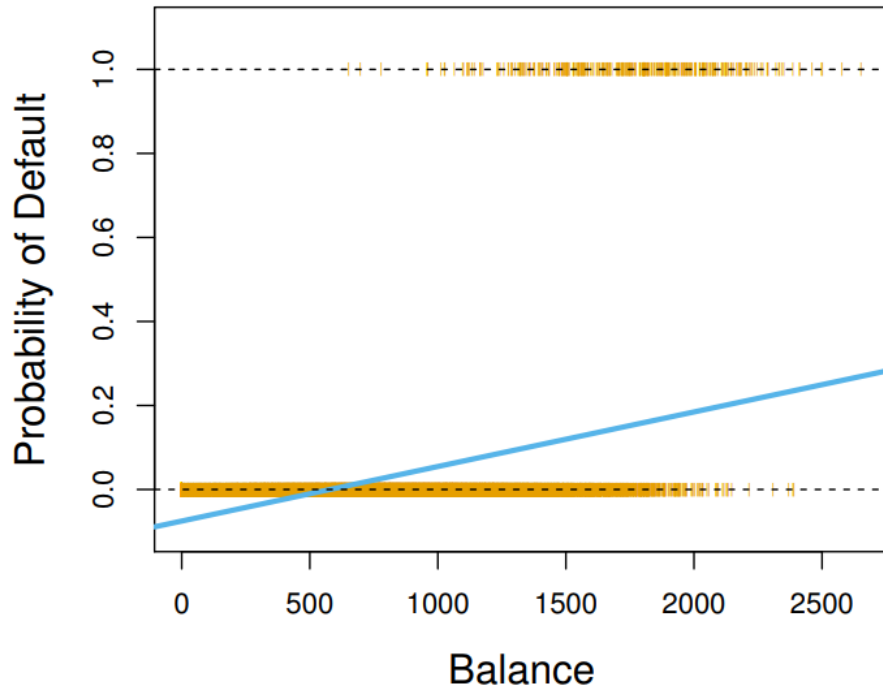
Can we use Linear Regression?

- Consider the a binary classification task

$$f(x) = \begin{cases} 0, & \text{is No} \\ 1, & \text{if Yes} \end{cases}$$

- Can we simply perform a linear regression of Y on X and classify as Yes if $\hat{Y} > 0.5$?
 - In this case of a binary outcome, linear regression does a good job as a classifier, and is equivalent to linear discriminant analysis
 - Since in the population $E(Y|X = x) = \Pr(Y = 1|X = x)$, we might think that regression is perfect for this task.
 - However: Linear regression might produce probabilities less than zero or bigger than one. Logistic regression is more appropriate.

Linear versus Logistic Regression



- The orange marks indicate the response Y , either 0 or 1. Linear regression does not estimate $\Pr(Y = 1|X)$ well. Logistic regression seems well suited to the task.

Logistic Regression

- Let's write $p(X) = \Pr(Y = 1|X)$ for short and consider using balance to predict default. Logistic regression uses the form

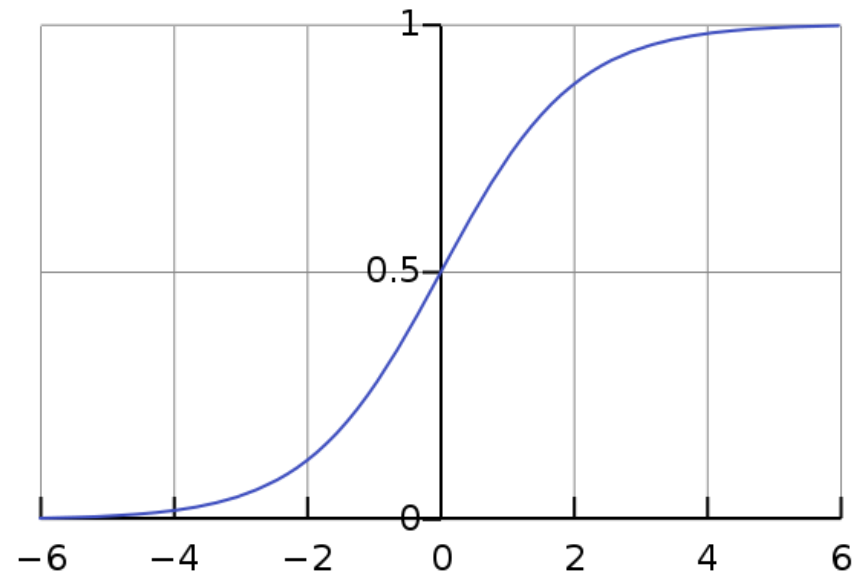
$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

- It is easy to see that no matter what values β_0 , β_1 or X take, $p(X)$ will have values between 0 and 1.

Different Point of View

- The Sigmoid Function:

$$S(x) = \frac{e^x}{e^x + 1}$$



Different Point of View

- That means, we can rewrite

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

as

$$p(X) = S(\beta_0 + \beta_1 X)$$

- That means, we perform a standard linear regression and afterwards transform the result by means of a Sigmoid function.
- So, what are we estimating?

Logistic Regression

- Remember the Formula

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

- A bit of rearrangement yields

$$\log\left(\frac{p(x)}{1 - p(x)}\right) = \beta_0 + \beta_1 X.$$

- This monotone transformation is called the log odds or logit transformation of $p(X)$.
- In other words, logistic regression assumes that the log odds is a linear function of X

Yet Another Way Looking at the Problem

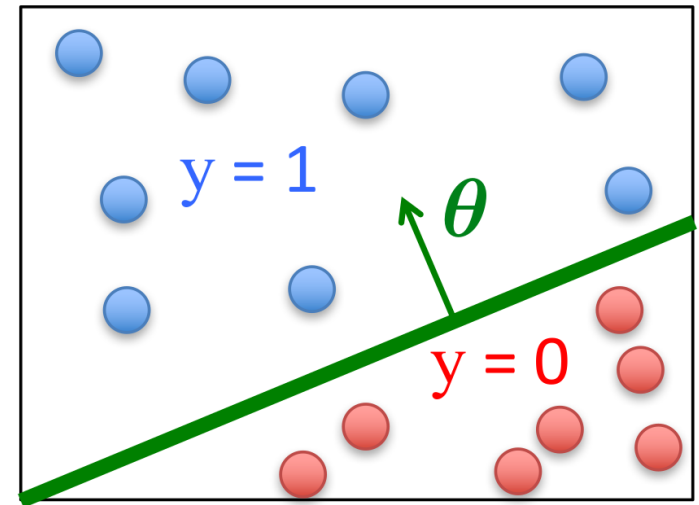
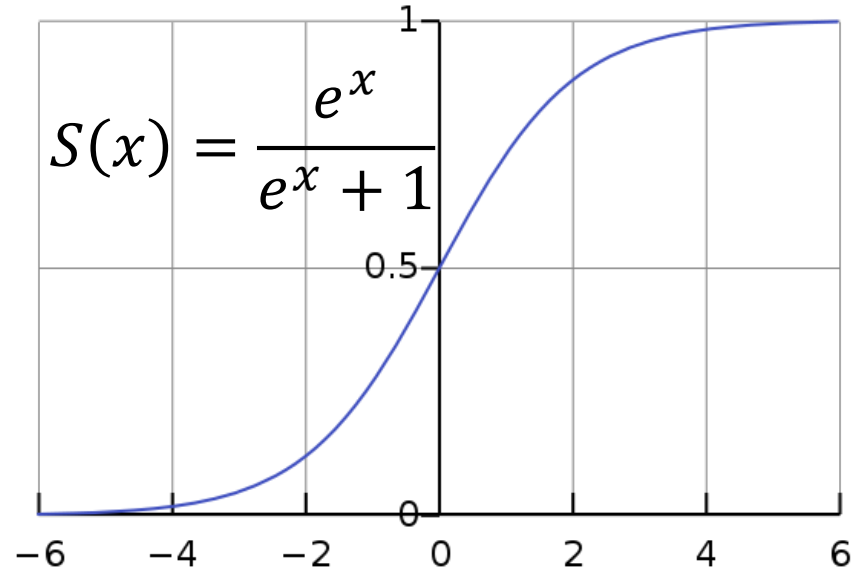
- What we are doing is, instead of predicting the class, we give the probability of the instance being in that class, i.e., learn

$$p(y|x, \theta)$$

- That means, we learn the parameters of a Bernoulli distribution depending on the given input
- We have to choose the θ in such a way, that it maximizes the agreement with our observations

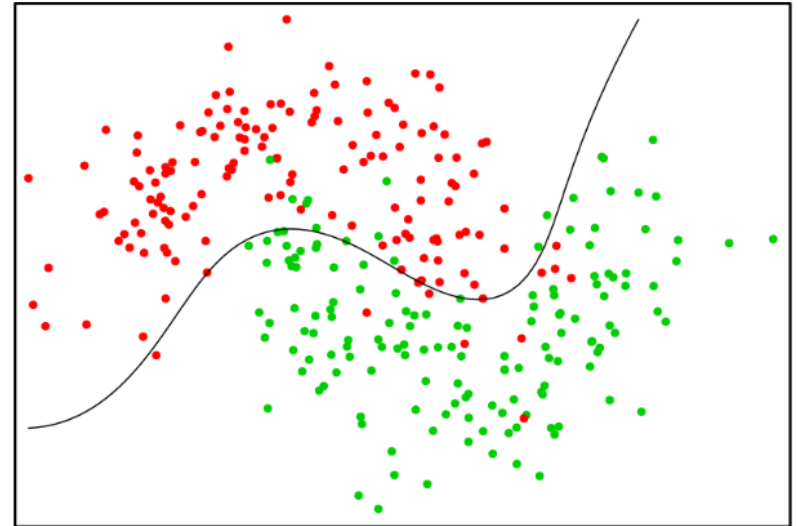
Learning Goal

- Our Model:
$$h_{\theta}(\mathbf{x}) = p(y|\mathbf{x}, \theta) = S(\theta^T \mathbf{x})$$
- $\theta^T \mathbf{x}$ should be large negative for negative instances
- $\theta^T \mathbf{x}$ should be large positive for positive instances
- Select a threshold t and
 - Predict $y = 1$ if $p(y|\mathbf{x}, \theta) \geq t$
 - Predict $y = 0$ if $p(y|\mathbf{x}, \theta) < t$



Non-Linear Decision Boundary

$$\mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1 x_2 \\ x_1^2 \\ x_2^2 \\ x_1^2 x_2 \\ x_1 x_2^2 \\ \vdots \end{bmatrix}$$



Logistic Regression Objective Function

- We cannot just use the least squared approach as with linear regression

$$J(\theta) = \frac{1}{n} \sum_i^n (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

- Since the logistic regression model with the sigmoid function results in a non-convex optimization problem.
- And due to other problem would not lead to the optimal parameter set

Maximum Likelihood Estimation

- The Maximum Likelihood Estimation (MLE) is a method of estimating the parameters of a model. This estimation method is one of the most widely used.
- The method of maximum likelihood selects the set of values of the model parameters that maximizes the likelihood function. Intuitively, this maximizes the "agreement" of the selected model with the observed data.
- The Maximum-likelihood Estimation gives an unified approach to estimation.

What is a Maximum Likelihood Estimator?

- The likelihood that one datapoint was generated by any distribution function is given by

$$l(\mathbf{x}, \boldsymbol{\theta}) = p(y|\mathbf{x}; \boldsymbol{\theta})$$

- For the entire dataset, we have the likelihood

$$L(X, \boldsymbol{\theta}) = \prod_{i=1}^n p(y^{(i)}|\mathbf{x}^{(i)}; \boldsymbol{\theta})$$

What is a Maximum Likelihood Estimator?

- Goal of the MLE is to find θ in such a way that $L(X, \theta)$ is maximized

$$\theta_{\text{MLE}} = \operatorname{argmax}_{\theta} L(X, \theta)$$

- Very often (for simplicity), the log likelihood is used

$$\theta_{\text{MLE}} = \operatorname{argmax}_{\theta} \log \prod_{i=1}^n p(y^{(i)} | \mathbf{x}^{(i)}; \theta)$$

$$= \operatorname{argmax}_{\theta} \sum_{i=1}^n \log p(y^{(i)} | \mathbf{x}^{(i)}; \theta)$$

- This doesn't change the maximum of the function
- Eases many calculations

Example: MLE for a Gaussian

- Target: Estimate μ and σ for a Gaussian. The likelihood is

$$L(X|\theta) = \prod_{i=1}^n \frac{1}{\sigma\sqrt{2\pi}} \exp \left[-\frac{(x - \mu)^2}{2\sigma^2} \right]$$

- Taking the log results in an easier version:

$$LL(X|\theta) = -\frac{1}{2}n \log 2\pi - n \log \sigma - \sum_{i=1}^n \frac{(x - \mu)^2}{2\sigma^2}$$

Example: MLE for a Gaussian

- Derivate for μ :

$$\frac{\partial LL}{\partial \mu} = \frac{1}{\sigma^2} \sum_{i=1}^n (x_i - \mu)$$

- And for σ^2 :

$$\frac{\partial LL}{\partial \sigma^2} = \frac{1}{2\sigma^2} \left[\frac{1}{\sigma^2} \sum_{i=1}^n (x_i - \mu)^2 - n \right]$$

- These derivates are 0 if

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i \quad \text{and} \quad \sigma = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2$$

Back to Logistic Regression

- Our MLE looks as follows:

$$\begin{aligned}\boldsymbol{\theta}_{\text{MLE}} &= \operatorname{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^n \log p(y^{(i)} | \mathbf{x}^{(i)}; \boldsymbol{\theta}) = \\ &= \operatorname{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^n \left[y^{(i)} \log p(y^{(i)} = 1 | \mathbf{x}^{(i)}; \boldsymbol{\theta}) + \right. \\ &\quad \left. + (1 - y^{(i)}) \log (1 - p(y^{(i)} = 1 | \mathbf{x}^{(i)}; \boldsymbol{\theta})) \right] = \\ &\operatorname{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^n \left[y^{(i)} \log h_{\boldsymbol{\theta}}(\mathbf{x}) + (1 - y^{(i)}) \log (1 - h_{\boldsymbol{\theta}}(\mathbf{x})) \right]\end{aligned}$$

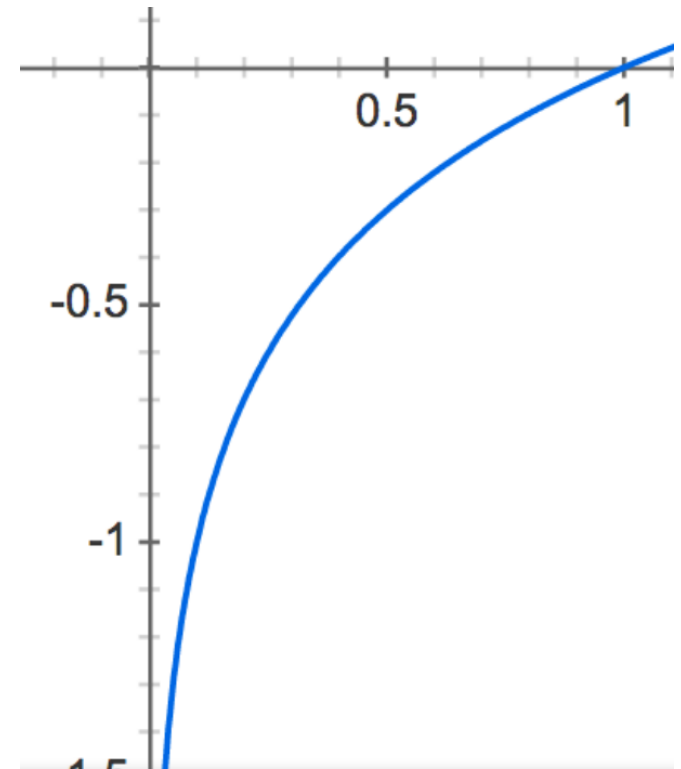
Intuition Behind the Objective

$$\operatorname{argmax}_{\theta} \sum_{i=1}^n [y^{(i)} \log h_{\theta}(\mathbf{x}) + (1 - y^{(i)}) \log(1 - h_{\theta}(\mathbf{x}))]$$

- Likelihood for a single Instance:

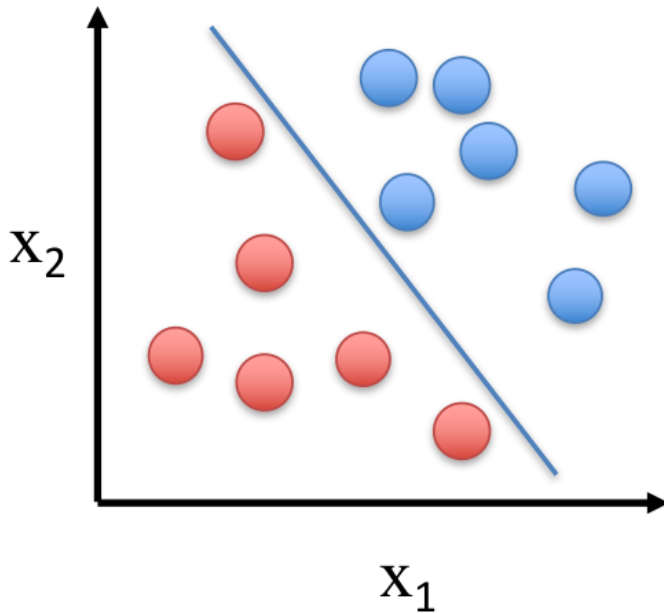
$$l(\mathbf{x}, \theta) = \begin{cases} -\log(h_{\theta}(\mathbf{x})) & \text{for } y = 1 \\ -\log(1 - h_{\theta}(\mathbf{x})) & \text{for } y = 0 \end{cases}$$

- Effect: Extremely negative with wrong classifications
- Trained with gradient descent methods

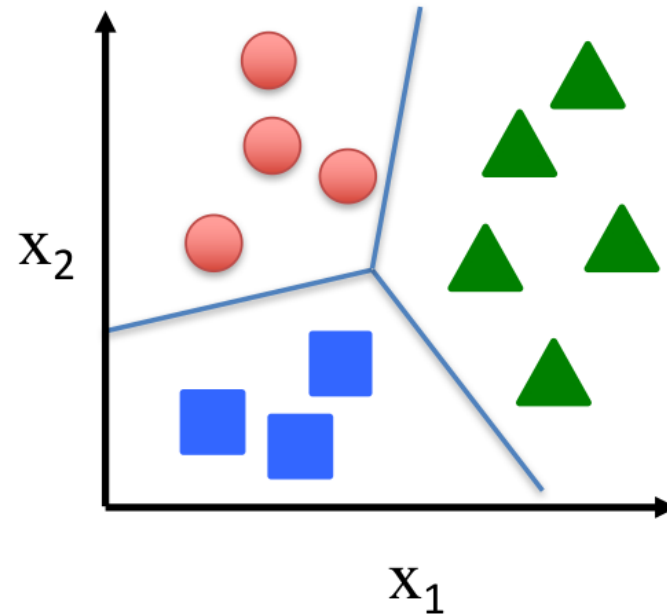


Multi-Class Classification

Binary classification:



Multi-class classification:




Disease diagnosis: healthy / cold / flu / pneumonia

Object classification: desk / chair / monitor / bookcase

Multi-Class Classification

- For 2 classes:

$$h_{\theta}(\mathbf{x}) = \frac{1}{1 + \exp(-\theta^T \mathbf{x})} = \frac{\exp(\theta^T \mathbf{x})}{\boxed{1} + \boxed{\exp(\theta^T \mathbf{x})}}$$



weight assigned to $y = 0$ weight assigned to $y = 1$

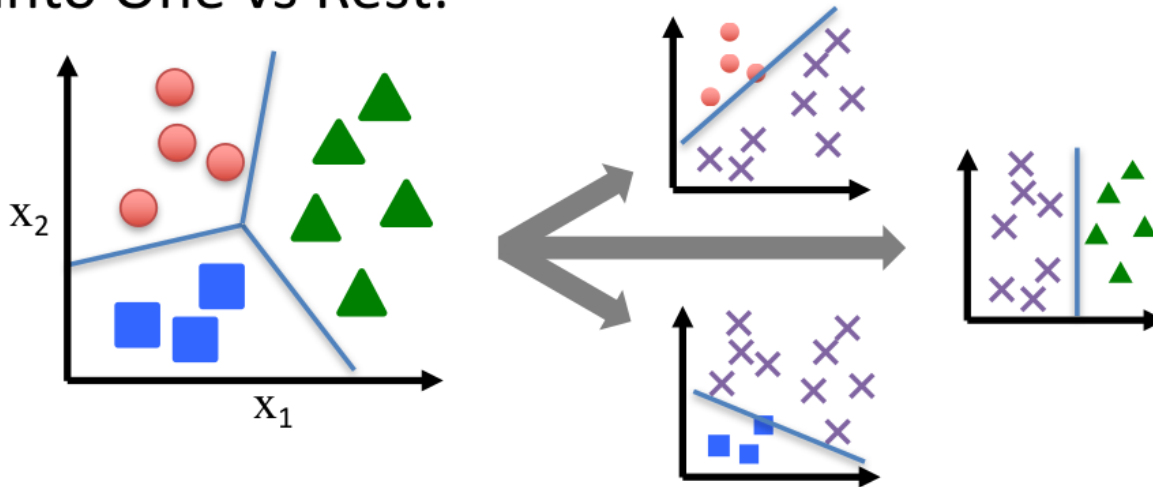
- For C classes $\{1, \dots, C\}$:

$$p(y = c \mid \mathbf{x}; \theta_1, \dots, \theta_C) = \frac{\exp(\theta_c^T \mathbf{x})}{\sum_{c=1}^C \exp(\theta_c^T \mathbf{x})}$$

– Called the **softmax** function

Multi-Class Classification

Split into One vs Rest:



- Train a logistic regression classifier for each class i to predict the probability that $y = i$ with

$$h_c(\mathbf{x}) = \frac{\exp(\boldsymbol{\theta}_c^T \mathbf{x})}{\sum_{c=1}^C \exp(\boldsymbol{\theta}_c^T \mathbf{x})}$$



Classifier Evaluation

Evaluate the Classifier Performance

- For classification, we need different measures than MSE
- Most simple measure: **Accuracy / Error Rate**
 - Accuracy: Percentage of correct classifications
 - Error Rate: Percentage of incorrect classifications
- Higher accuracy does not necessarily imply better performance on target task!
 - Implicit assumption: the class distribution among examples is relative balanced
 - Biased in favor of the majority class!
 - Should be used with caution!

Confusion matrix, two classes only

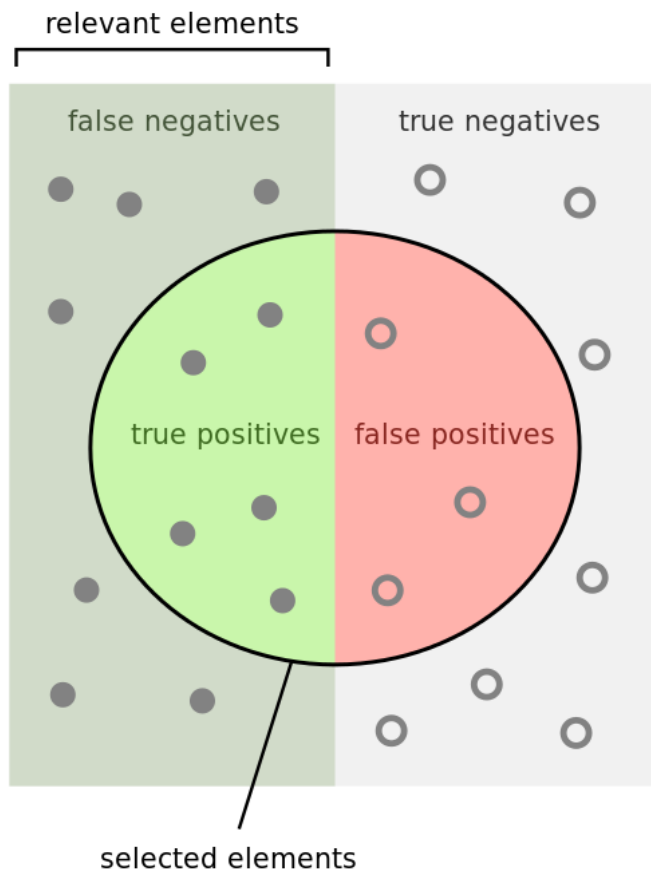
Performance measures calculated from the confusion matrix entries:

- ◆ Accuracy = $(a + d)/(a + b + c + d) = (TN + TP)/total$
- ◆ **True positive rate**, recall, sensitivity = $d/(c + d) = TP/actual\ positive$
- ◆ Specificity, true negative rate = $a/(a + b) = TN/actual\ negative$
- ◆ Precision, predicted positive value = $d/(b + d) = TP/predicted\ positive$
- ◆ **False positive rate**, false alarm = $b/(a + b) = FP/actual\ negative = 1 - specificity$
- ◆ False negative rate = $c/(c + d) = FN/actual\ positive$

		predicted	
		negative	positive
actual examples	negative	a TN - True Negative correct rejections	b FP - False Positive false alarms type I error
	positive	c FN - False Negative misses, type II error overlooked danger	d TP - True Positive hits


F-Measure

- Harmonic mean between Precision (Prec) and Recall (Rec):




$$F = 2 \cdot \frac{\text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}}$$

How many selected items are relevant?

Precision = 

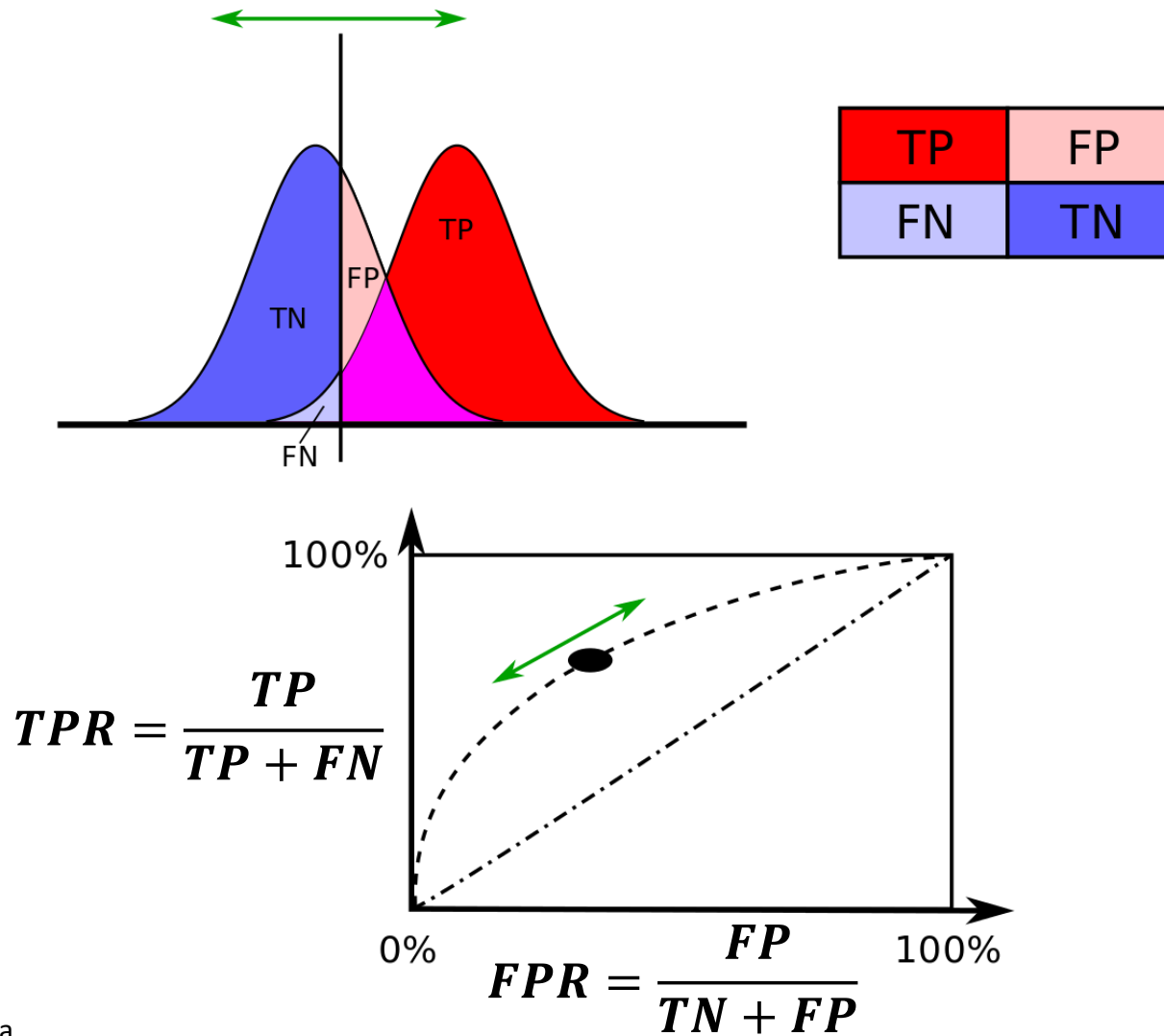
How many relevant items are selected?

Recall = 

Roc Curve

- Receiver Operating Characteristic
 - The ROC curve was first used during World War II for the analysis of radar signals before it was employed in signal detection theory ... for these purposes they measured the ability of a radar receiver operator to make these important distinctions, which was called the Receiver Operating Characteristic.”
- ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.
- For our Regression, we simply vary the threshold t and
 - Predict $y = 1$ if $p(y|\mathbf{x}, \boldsymbol{\theta}) \geq t$
 - Predict $y = 0$ if $p(y|\mathbf{x}, \boldsymbol{\theta}) < t$

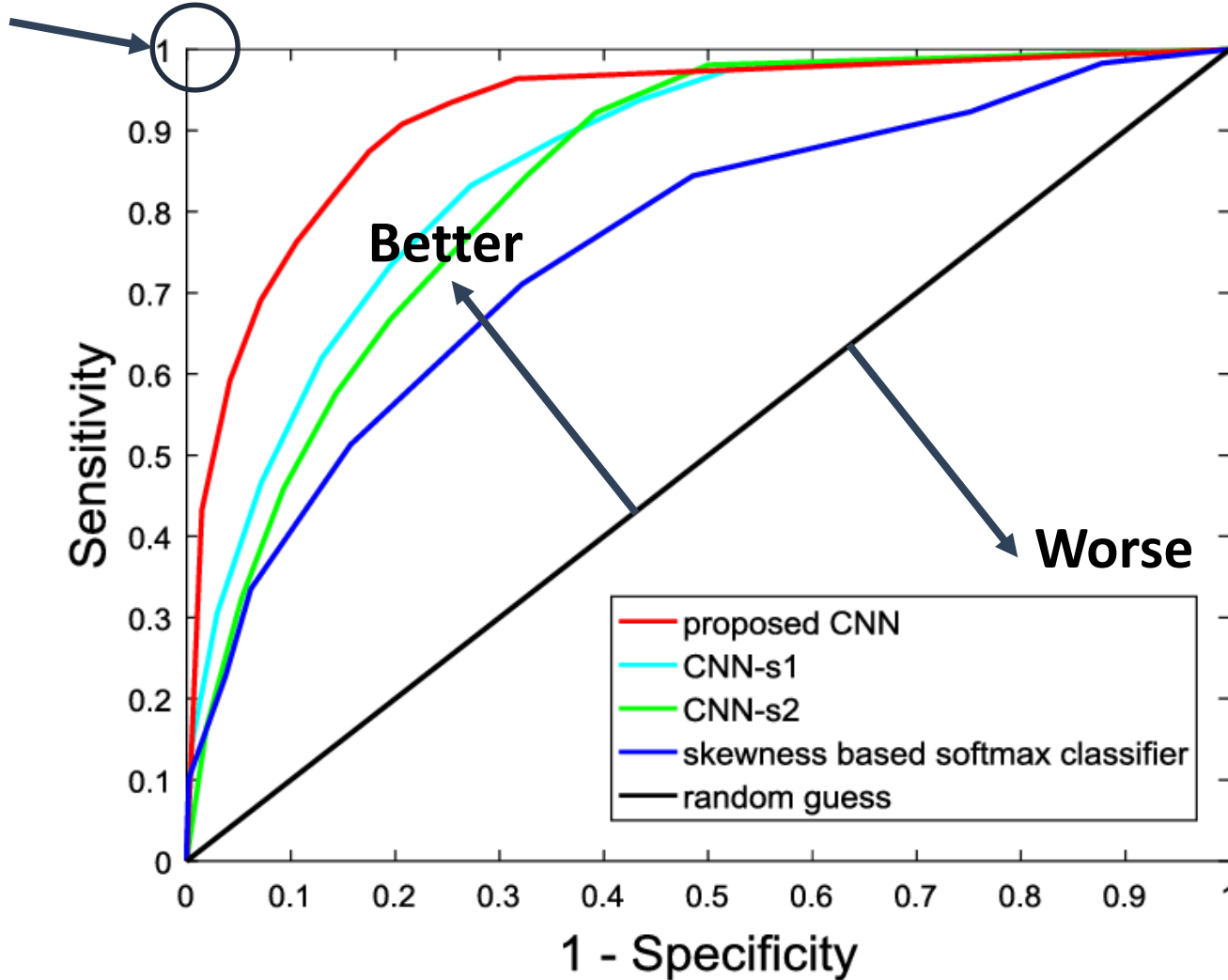
ROC Curve for Changing Threshold



➤ Source: wikipedia

Interpreting a ROC Curve

Perfect Classifier



AUC

- To summarize a ROC Curve, often the AUC is used
- AUC: Area under the curve
 - Perfect Classifier: $AUC = 1$
 - Random Classifier: $AUC = 0.5$

