Linear Classification & Logistic Regression

Radoslav Neychev

PT, fall 2023

girafe ai

Recap

Lecture 2:
Linear Regression

- Linear Models overview
- Regression problem statement
- Linear Regression analytical solution
    - Instability
- Regularization
    - L2 aka Ridge
        - Analytical solution
    - L1 aka LASSO
        - Weights decay rule

Outline

- Linear classification
  - margin
  - loss functions
- Logistic regression
  - sigmoid derivation
  - Maximum Likelihood Estimation (MLE)
  - logistic loss
  - probability calibration
- Metrics in classification
  - Accuracy, Balanced accuracy
  - Precision, Recall, F-score
  - ROC curve, PR curve, AUC
  - Confusion matrix
- *Optional: Multiclass aggregation strategies*
  - *One vs Rest*
  - *One vs One*

Linear Classification

girafe
ai

01

Classification problem

$$X \in R^{n \times p}$$

$$Y \in C^n \qquad \text{e.g. } C = \{-1, 1\}$$

$$|C| < +\infty$$

$$c(X) = \hat{Y} \approx Y$$

Linear classifier

The most simple linear classifier

$$c(x) = \begin{cases} 1, & \text{if } f(x) \geq 0 \\ -1, & \text{if } f(x) < 0 \end{cases}$$

or equivalent

$$c(x) = \text{sign}(f(x)) = \text{sign}(x^T w)$$

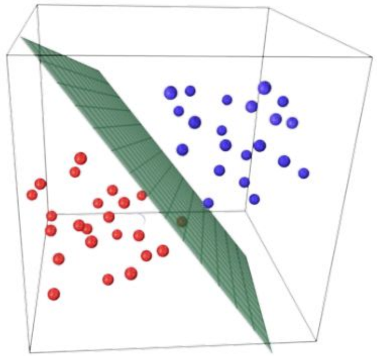
Geometrical interpretation:

hyperplane dividing space into two

subspaces

Why cutoff value is fixed?

(bias term is implied)



Margin

Let’s define linear model’s Margin as

$$M_i = y_i \cdot f(x_i) = y_i \cdot x_i^T w$$

main property:

negative margin reveals misclassification

$$M_i > 0 \Leftrightarrow y_i = c(x_i)$$

$$M_i \leq 0 \Leftrightarrow y_i \neq c(x_i)$$

Weights choice

Remembering old paradigm

Empirical risk = $\sum_{\text{by object}}$ Loss on object → $\min_{\text{model params}}$

Essential loss is misclassification

$$L_{\text{mis}}(y_i^t, y_i^p) = [y_i^t \neq y_i^p] = \\ = [M_i \leq 0]$$

Iverson bracket $[P] = \begin{cases} 1, & \text{if } P \text{ is true} \\ 0, & \text{otherwise} \end{cases}$

Disadvantages

- Not differentiable
- Overlooks confidence

Solution:

estimate it with a smooth function

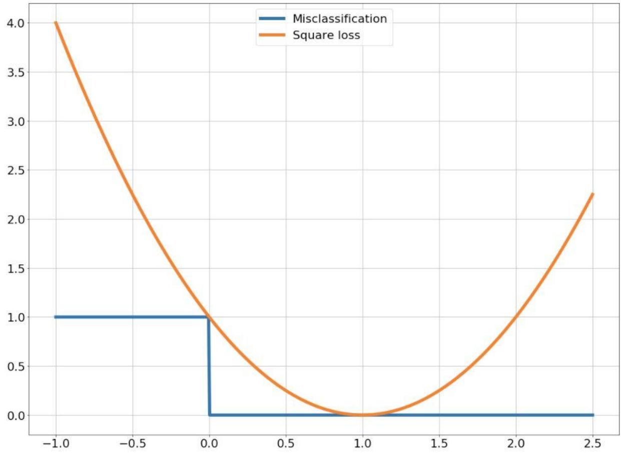
Square loss

Let's treat classification problem as regression problem:

$$Y \in \{-1, 1\} \mapsto Y \in R$$

thus we optimize MSE

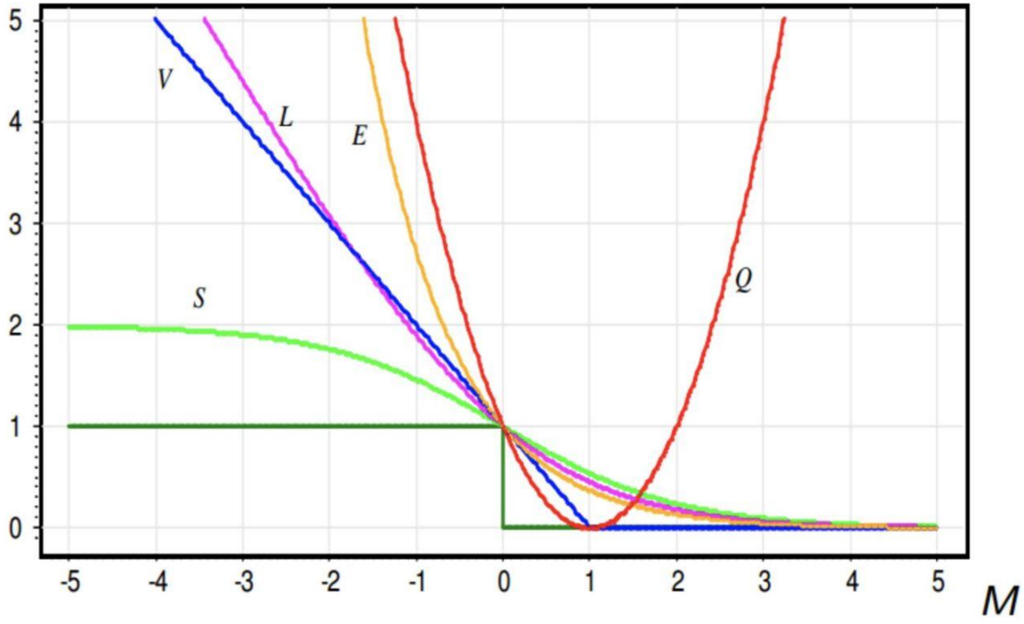
$$L_{\text{MSE}} = (y_i - x_i^T w)^2 = \frac{(y_i^2 - y_i \cdot x_i^T w)^2}{y_i^2} =$$
$$= (1 - y_i \cdot x_i^T w)^2 = (1 - M_i)^2$$



Advantage: already solved

Disadvantage: penalizes for high confidence

Other losses



$$Q(M) = (1 - M)^2$$
$$V(M) = (1 - M)_+$$
$$S(M) = 2(1 + e^M)^{-1}$$
$$L(M) = \log_2(1 + e^{-M})$$
$$E(M) = e^{-M}$$

[Loss functions for classification]

Logistic Regression

girafe
ai

02

Intuition

I. Let's try to predict probability of an object to have positive class

$$p_+ = P(y = 1|x) \in [0, 1]$$

II. But all we can predict is a real number!

$$y = x^T w \in R$$

III. Time for some tricks

$$\frac{p_+}{1 - p_+} \in [0, +\infty)$$

$$\log \frac{p_+}{1 - p_+} \in R$$

Here is the match

IV. Reverse to closed form

$$\frac{p_+}{1 - p_+} = \exp(x^T w)$$

$$p_+ = \frac{1}{1 + exp(-x^T w)} = \sigma(x^T w)$$

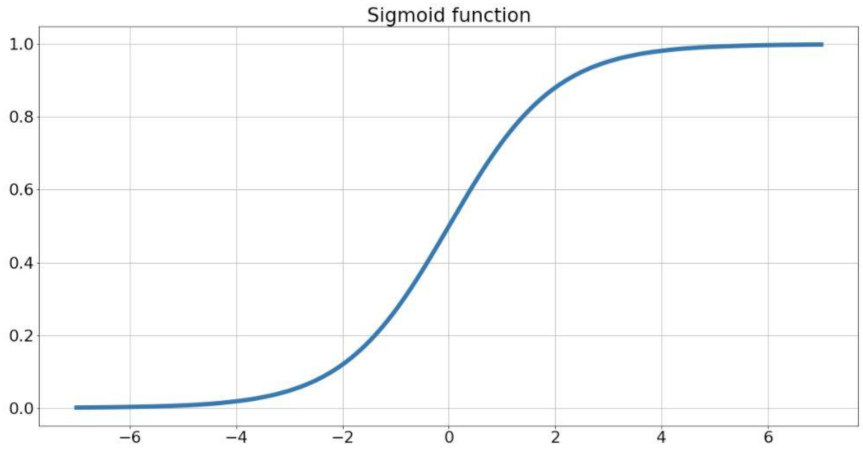
Sigmoid (aka logistic) function

$$\sigma(x) = \frac{1}{1 + exp(-x)}$$

Sigmoid is odd relative to (0, 0.5) point

Symmetric property:

$$1 - \sigma(x) = \sigma(-x)$$



Derivative: $\sigma'(x) = \sigma(x) \cdot (1 - \sigma(x))$

Maximum Likelihood Estimation

Just to remind

$$\log L(w|X,Y) = \log P(X,Y|w) = \log \prod_{i=1}^{n} P(x_i, y_i|w)$$

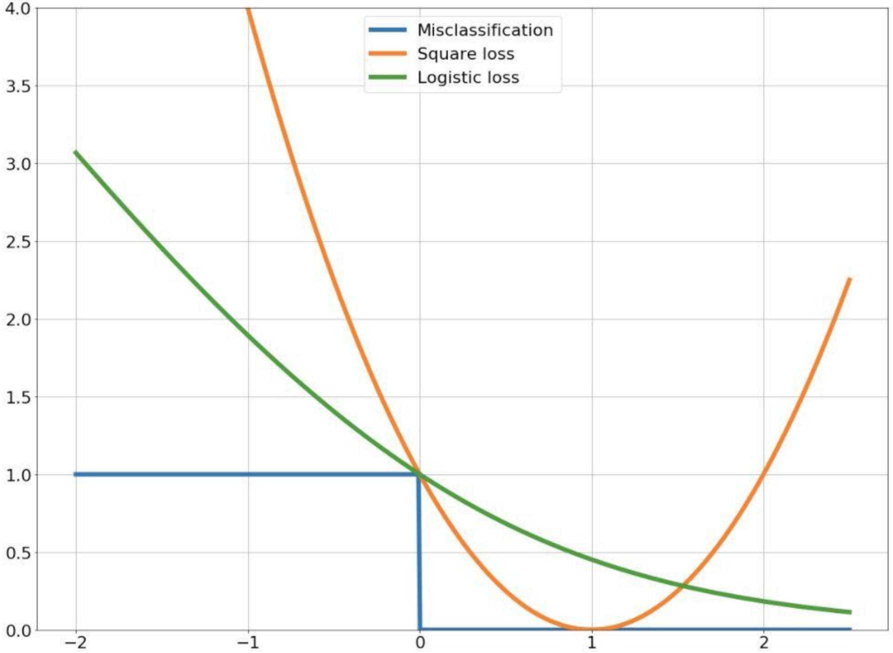
Calculating probabilities for objects

$$\begin{aligned} \text{if } y_i = 1: &\quad P(x_i, 1|w) = \sigma_w(x_i) = \sigma_w(M_i) \\ \text{if } y_i = -1: &\quad P(x_i, -1|w) = 1 - \sigma_w(x_i) = \sigma_w(-x_i) = \sigma_w(M_i) \end{aligned}$$

$$\log L(w|X,Y) = \sum_{i=1}^{n} \log \sigma_w(M_i) = \boxed{-\sum_{i=1}^{n} \log(1+\exp(-M_i)) \to \min_w}$$

Logistic loss

$$L_{Logistic} = \log(1 + \exp(-M_i))$$



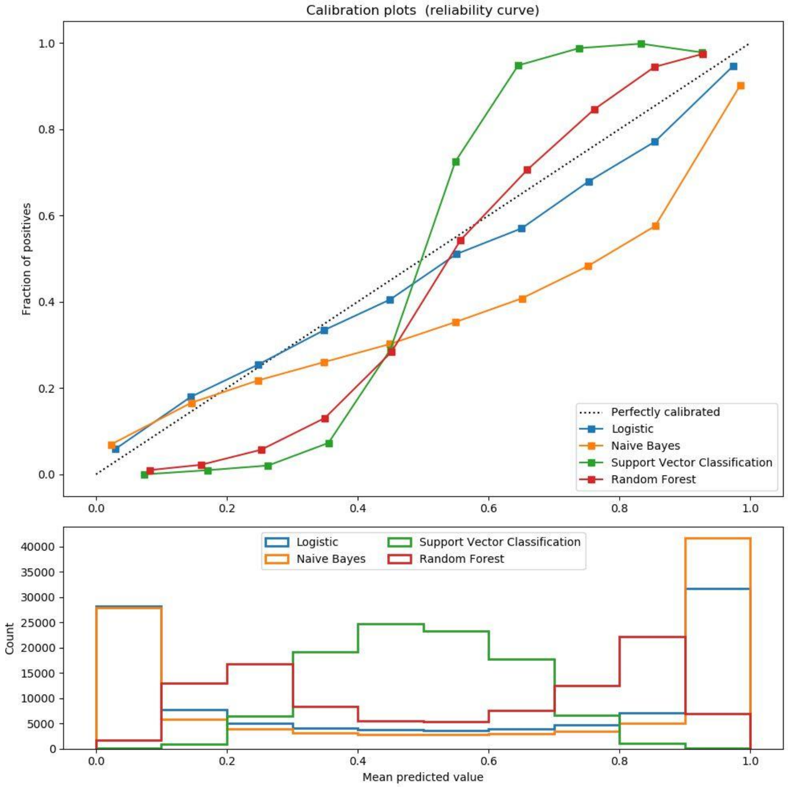
Probability calibration

By using Logistic Regression

we generate a Bernoulli distribution

in each point of space

[Calibration discussion]

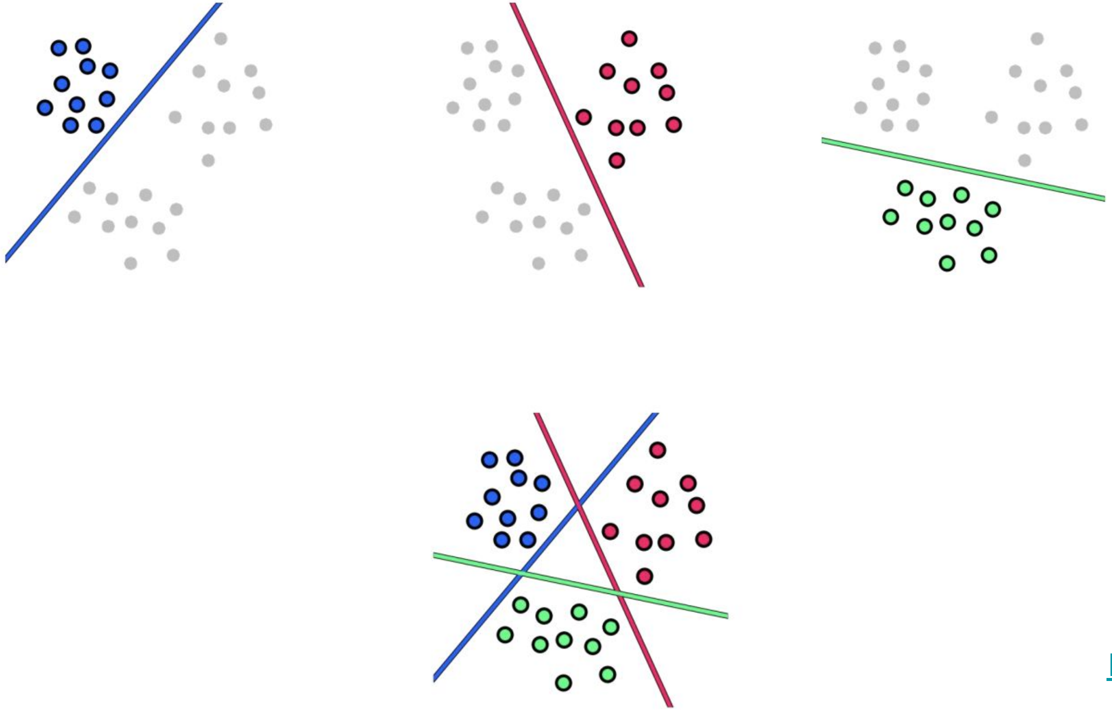


Multiclass aggregation strategies

girafe
ai

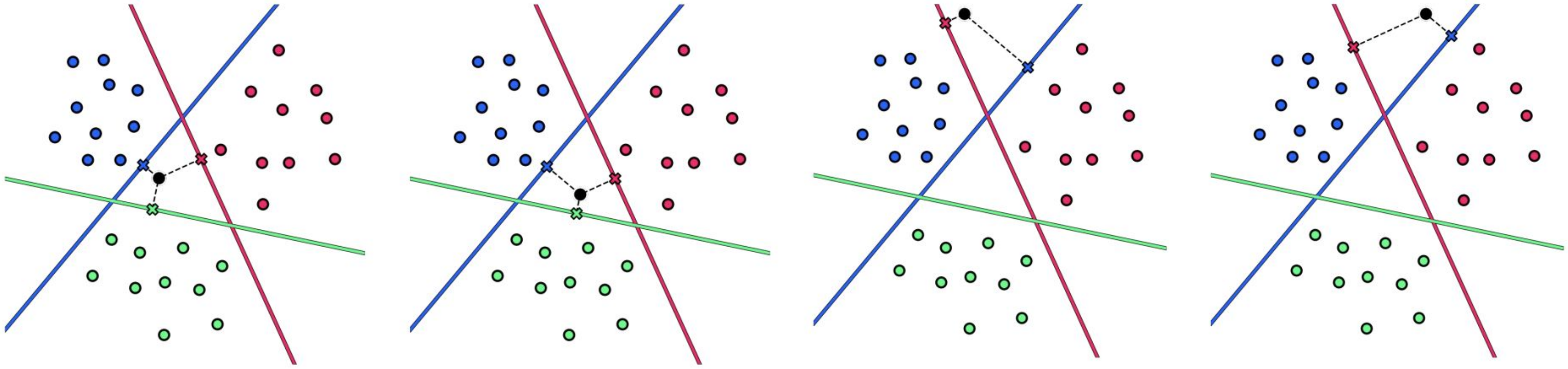
03

One vs Rest

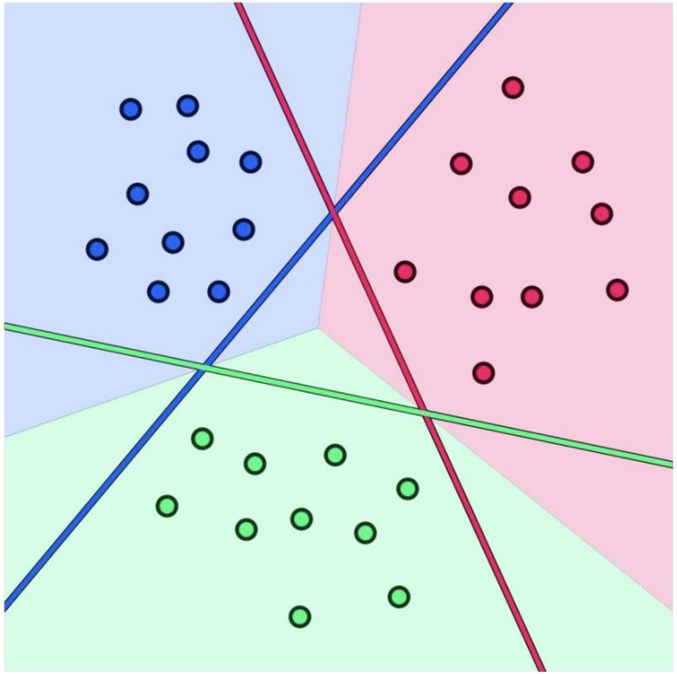


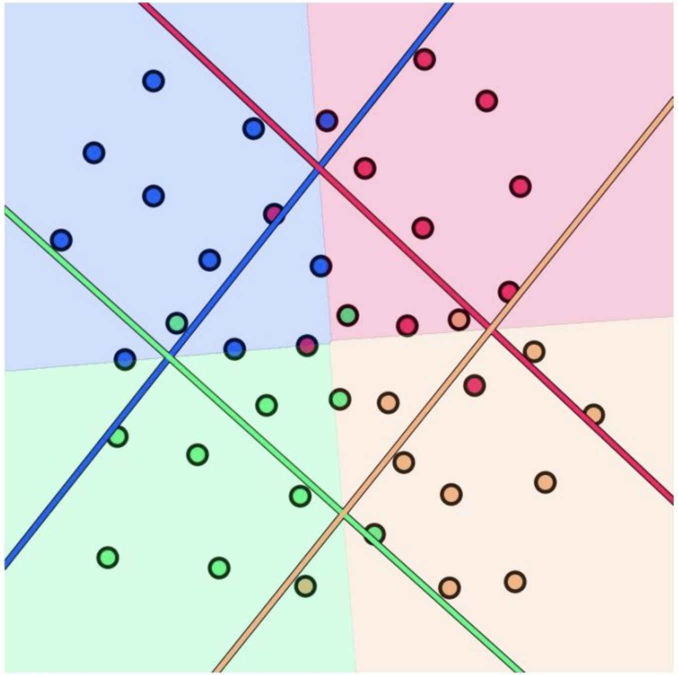
[Images source]

One vs Rest: unclassified regions

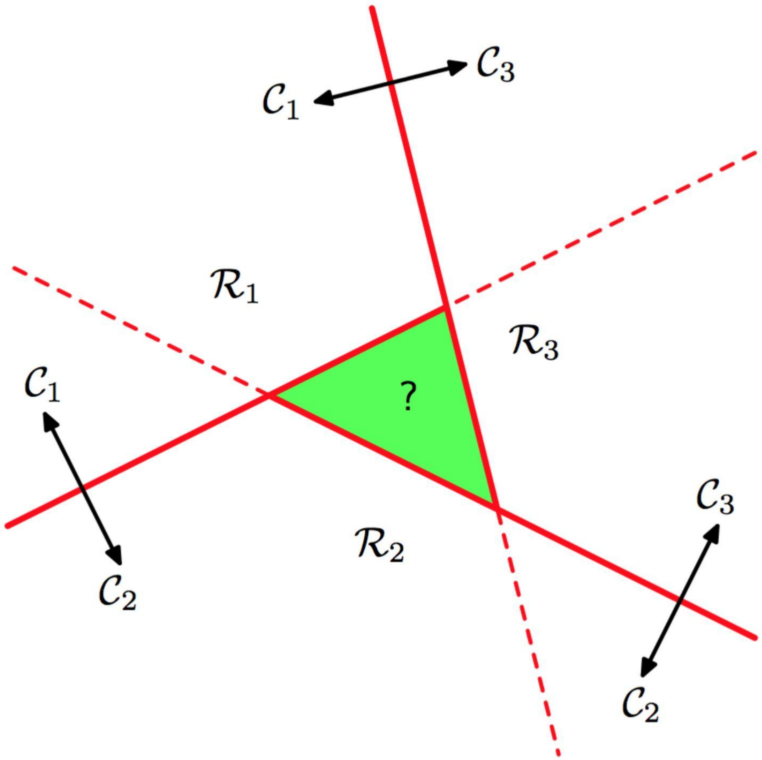


One vs Rest: final result

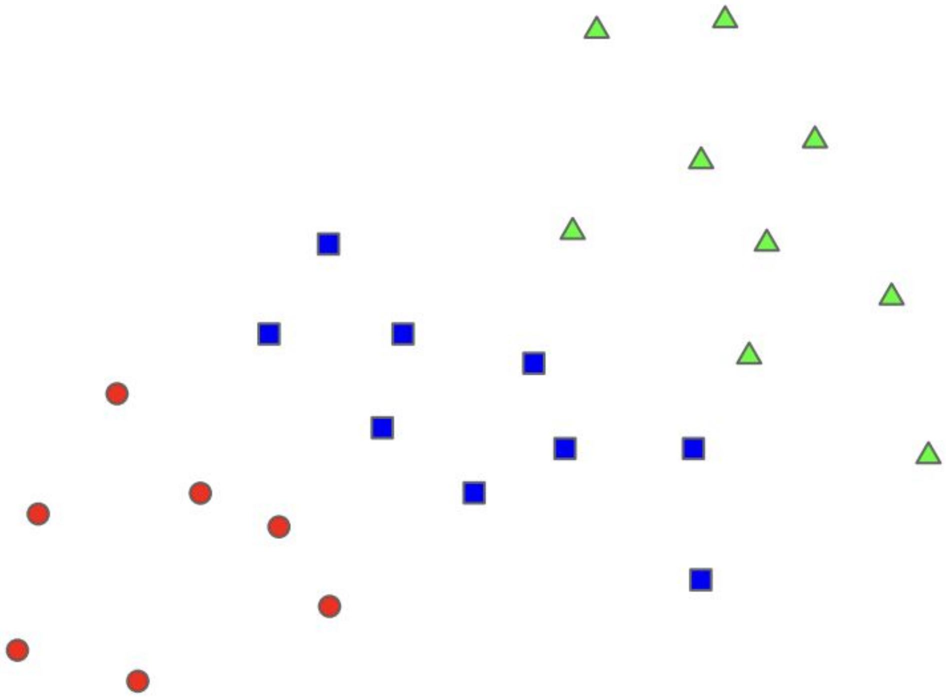




One vs One


$\mathcal{C}_1$
$\mathcal{C}_3$
$\mathcal{R}_1$
$\mathcal{R}_3$
$\mathcal{C}_1$
?
$\mathcal{C}_2$
$\mathcal{R}_2$
$\mathcal{C}_3$
$\mathcal{C}_2$

Failure case?



Summary

	One vs Rest	One vs One
#classifiers	k	k(k-1)/2
dataset for each	full	subsampled

Metrics in classification

girafe
ai

04

Metrics

- Accuracy
  - Balanced accuracy
- Precision
- Recall
- F-score
- ROC curve
  - ROC-AUC
- PR curve
  - PR-AUC
- Multiclass generalizations
- Confusion matrix

Accuracy

Number of right classifications

$$\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{n}[y_i^t = y_i^p]$$

target: 1 0 1 0 0 0 0 1 0 0

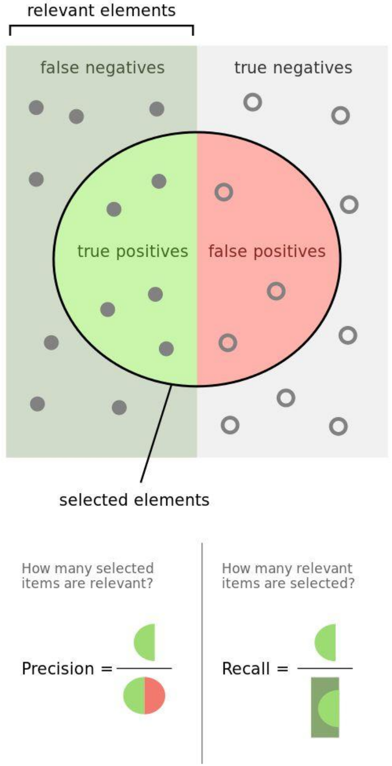
predicted: 0 0 1 0 0 0 0 1 1 0

accuracy = 8/10 = 0.8

$$\text{Balanced accuracy} = \frac{1}{C}\sum_{k=1}^{C}\frac{\sum_i [y_i^t = k \text{ and } y_i^t = y_i^p]}{\sum_i [y_i^t = k]}$$

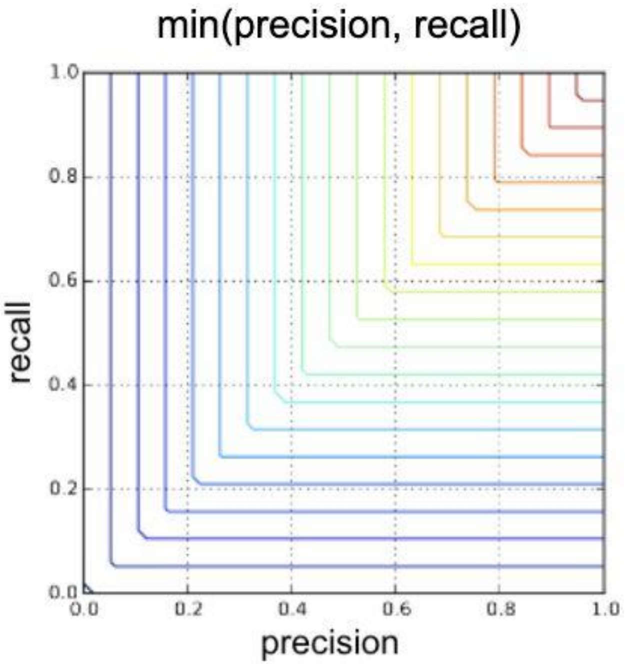
Precision and Recall

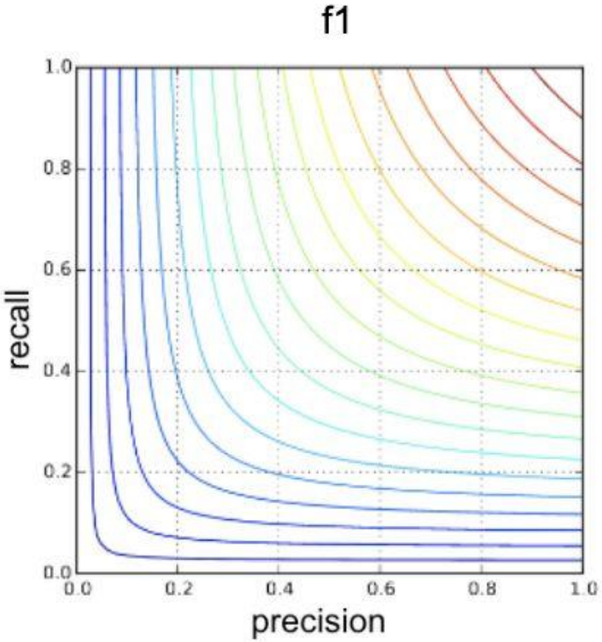
		True condition	
	Total population	Condition positive	Condition negative
Predicted condition	Predicted condition positive	**True positive**	**False positive**, Type I error
	Predicted condition negative	**False negative**, Type II error	**True negative**



$$\mathrm{Precision} = \frac{TP}{TP + FP} \qquad \mathrm{Recall} = \frac{TP}{TP + FN}$$

F-score motivation





F-score

Harmonic mean of precision and recall

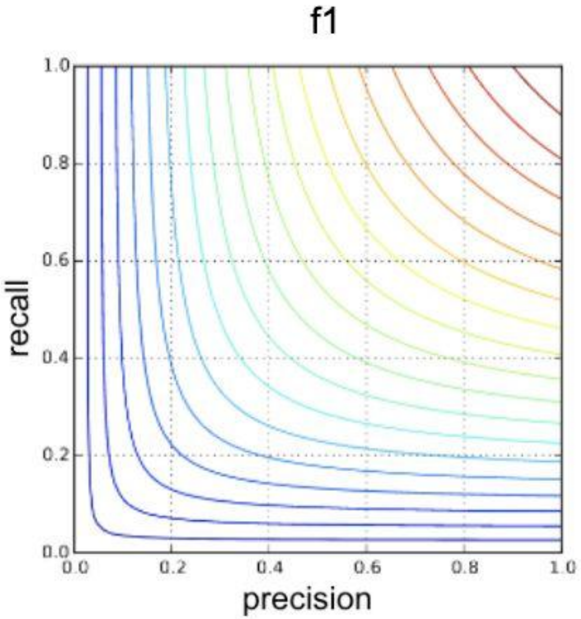
Closer to smaller one

$$F_1 = \frac{2}{\mathrm{precision}^{-1} + \mathrm{recall}^{-1}} = 2\,\frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

Generalization to different ratio between

Precision and Recall

$$F_\beta = (1 + \beta^2)\,\frac{\mathrm{precision} \cdot \mathrm{recall}}{\beta^2\,\mathrm{precision} + \mathrm{recall}}$$

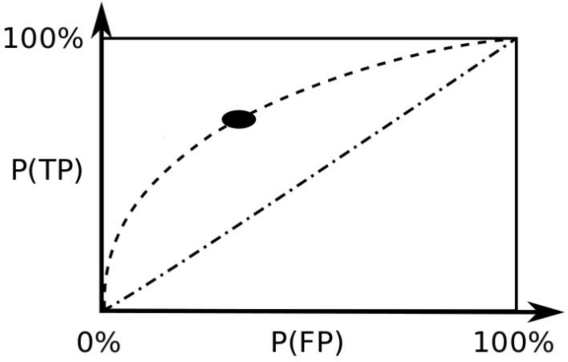


Receiver Operating Characteristic (ROC)

		True condition	
	Total population	Condition positive	Condition negative
Predicted condition	Predicted condition positive	**True positive**	**False positive**, Type I error
	Predicted condition negative	**False negative**, Type II error	**True negative**

$$\mathrm{FPR} = \frac{FP}{FP + TN}$$

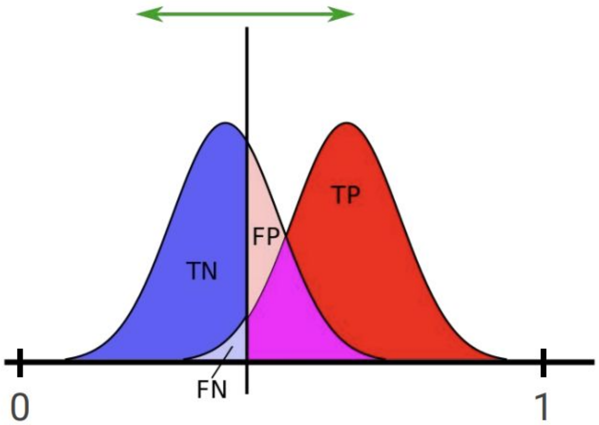
$$\mathrm{TPR} = \frac{TP}{TP + FN} (= \mathrm{Recall})$$

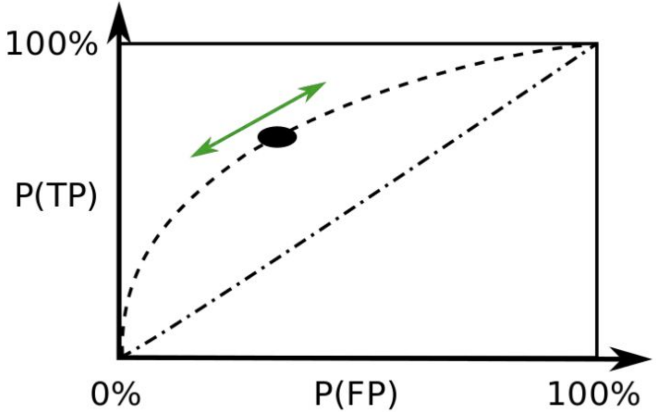


Receiver Operating Characteristic (ROC)

Classifier needs to predict probabilities

Objects get sorted by positive probability





Line is plotted as threshold moves

Receiver Operating Characteristic (ROC)

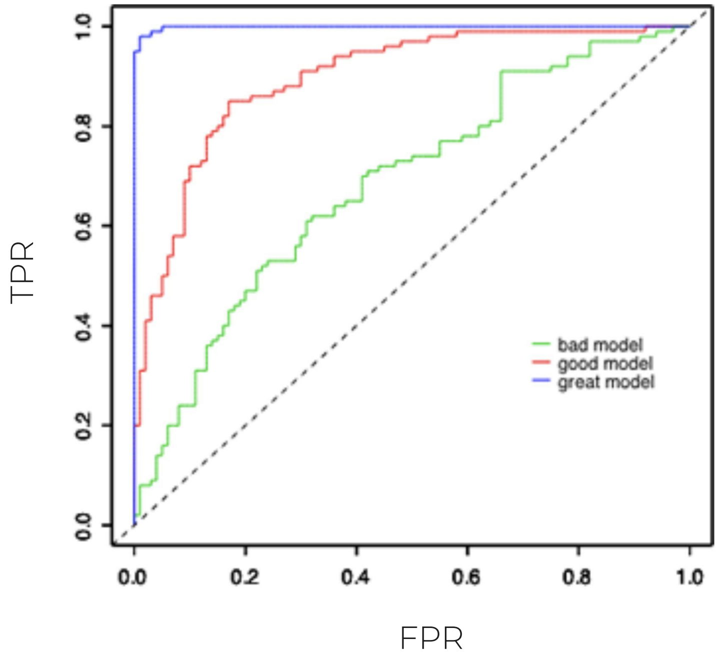
Baseline is random predictions

Always above diagonal (for reasonable classifier)

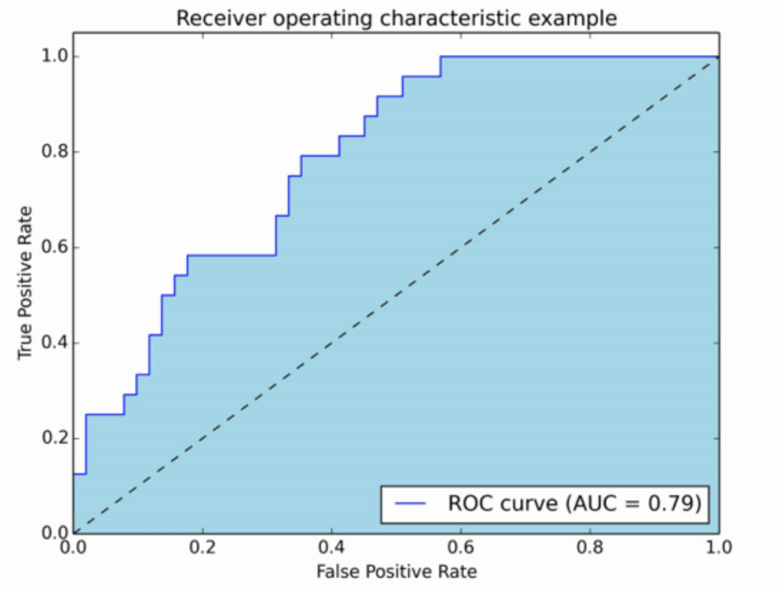
If below - change sign of predictions

Strictly higher curve means better classifier

Number of steps (thresholds) not bigger than dataset



ROC Area Under Curve (ROC-AUC)

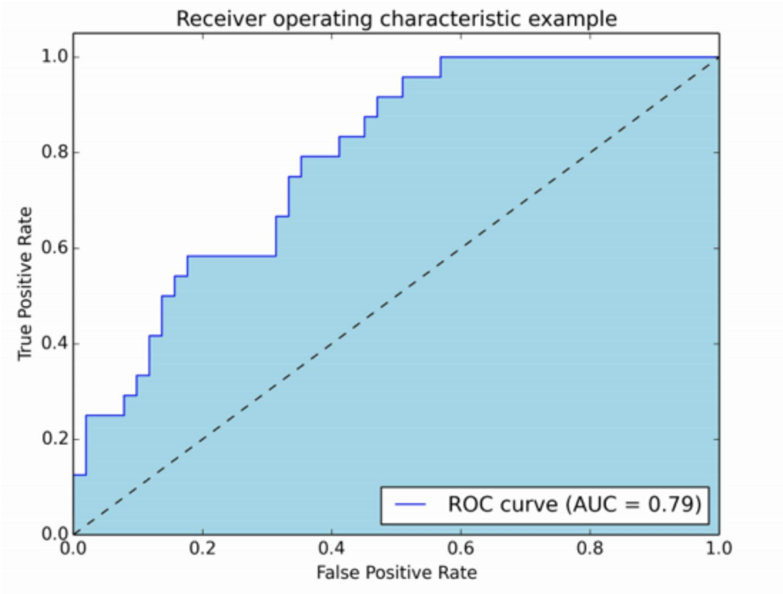


Effectively lays in (0.5, 1)

Bigger ROC-AUC doesn’t imply higher curve everywhere

[More explanations with pictures]

ROC-AUC properties



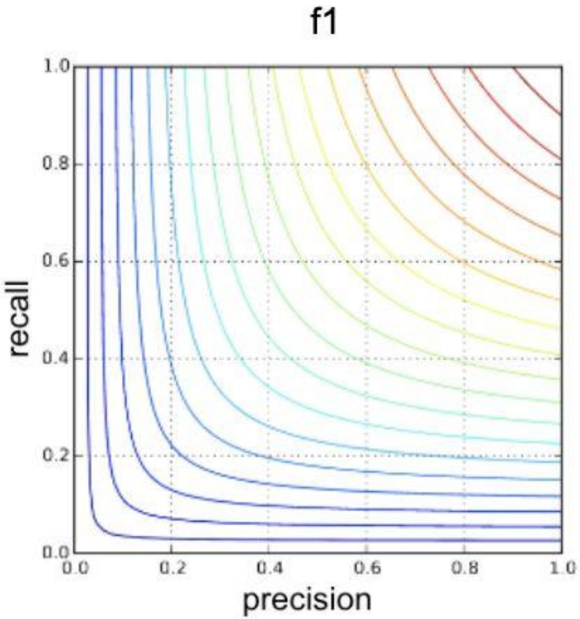
Scale-invariant. It measures how well predictions are ranked, rather than their absolute values.

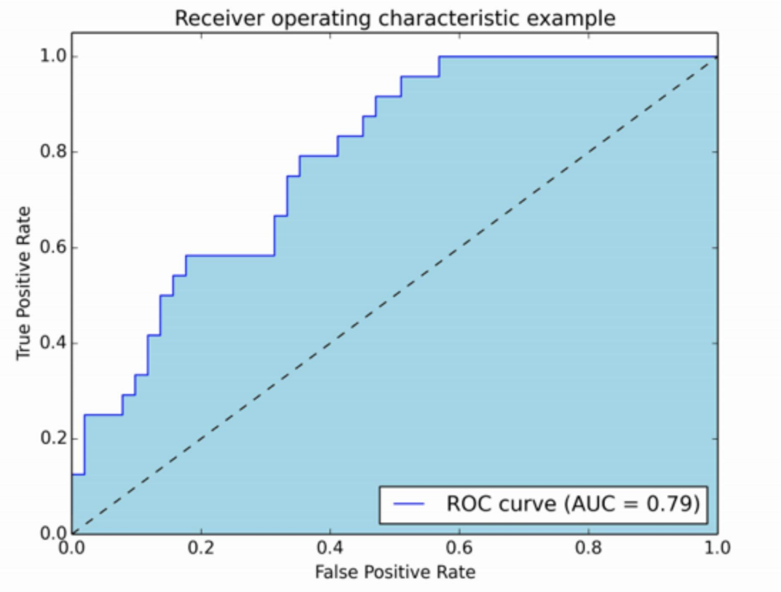
Classification-threshold-invariant. It measures the quality of the model's predictions irrespective of what classification threshold is chosen.

Source

F-score vs ROC-AUC

Which is better to tune?





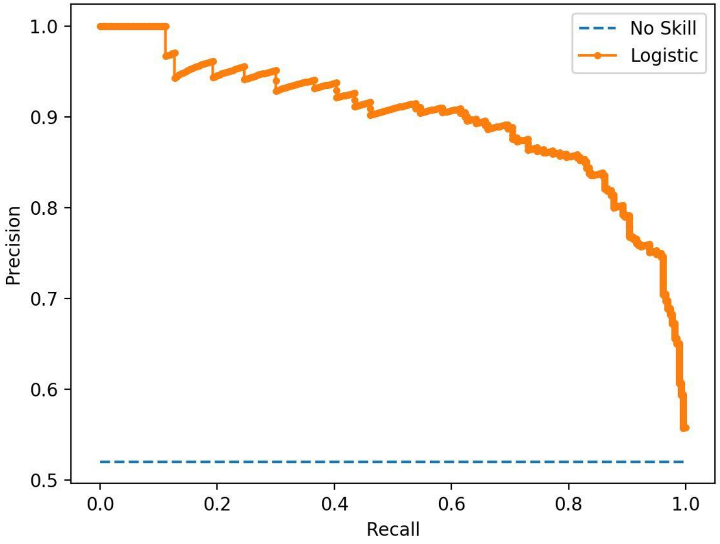
Precision-Recall Curve

AUC is in (0, 1)

Source of AP metric

(important for next semester)

[Nice article]



Multiclass metrics

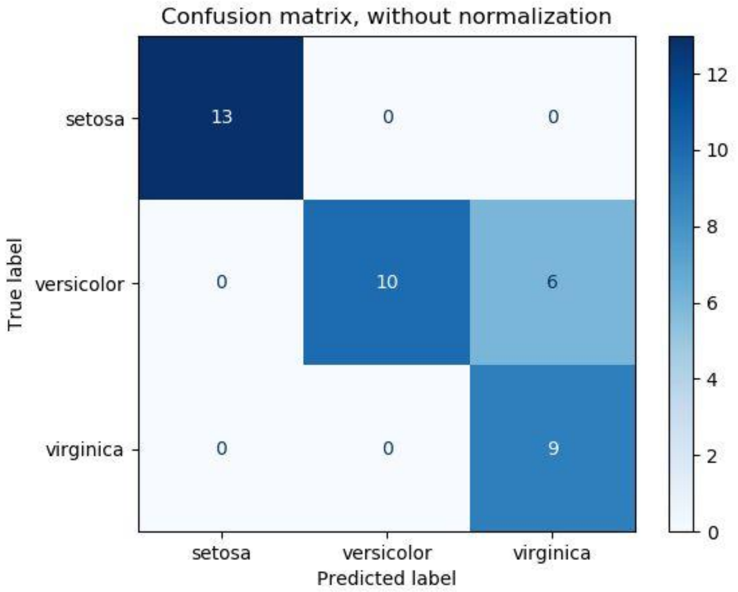
As with linear models we need some magic to measure multiclass problems

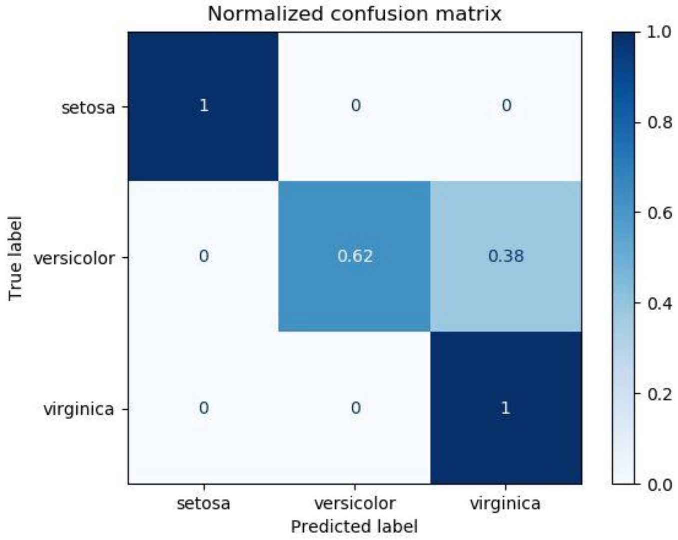
Basically it's mean of one or another kind

Detailed info here and here

average	Precision	Recall	F_beta
"micro"	$P(y, \hat{y})$	$R(y, \hat{y})$	$F_\beta(y, \hat{y})$
"samples"	$\frac{1}{\|S\|} \sum_{s \in S} P(y_s, \hat{y}_s)$	$\frac{1}{\|S\|} \sum_{s \in S} R(y_s, \hat{y}_s)$	$\frac{1}{\|S\|} \sum_{s \in S} F_\beta(y_s, \hat{y}_s)$
"macro"	$\frac{1}{\|L\|} \sum_{l \in L} P(y_l, \hat{y}_l)$	$\frac{1}{\|L\|} \sum_{l \in L} R(y_l, \hat{y}_l)$	$\frac{1}{\|L\|} \sum_{l \in L} F_\beta(y_l, \hat{y}_l)$
"weighted"	$\frac{1}{\sum_{l \in L} \|\hat{y}_l\|} \sum_{l \in L} \|\hat{y}_l\| P(y_l, \hat{y}_l)$	$\frac{1}{\sum_{l \in L} \|\hat{y}_l\|} \sum_{l \in L} \|\hat{y}_l\| R(y_l, \hat{y}_l)$	$\frac{1}{\sum_{l \in L} \|\hat{y}_l\|} \sum_{l \in L} \|\hat{y}_l\| F_\beta(y_l, \hat{y}_l)$

Confusion matrix





Revise

- Linear classification
  - margin
  - loss functions
- Logistic regression
  - sigmoid derivation
  - Maximum Likelihood Estimation
  - Logistic loss
  - probability calibration
- Multiclass aggregation strategies
  - One vs Rest
  - One vs One
- Metrics in classification
  - Accuracy, Balanced accuracy
  - Precision, Recall, F-score
  - ROC curve, PR curve, AUC
  - Confusion matrix

Model validation and evaluation

girafe
ai

04

Supervised learning problem statement

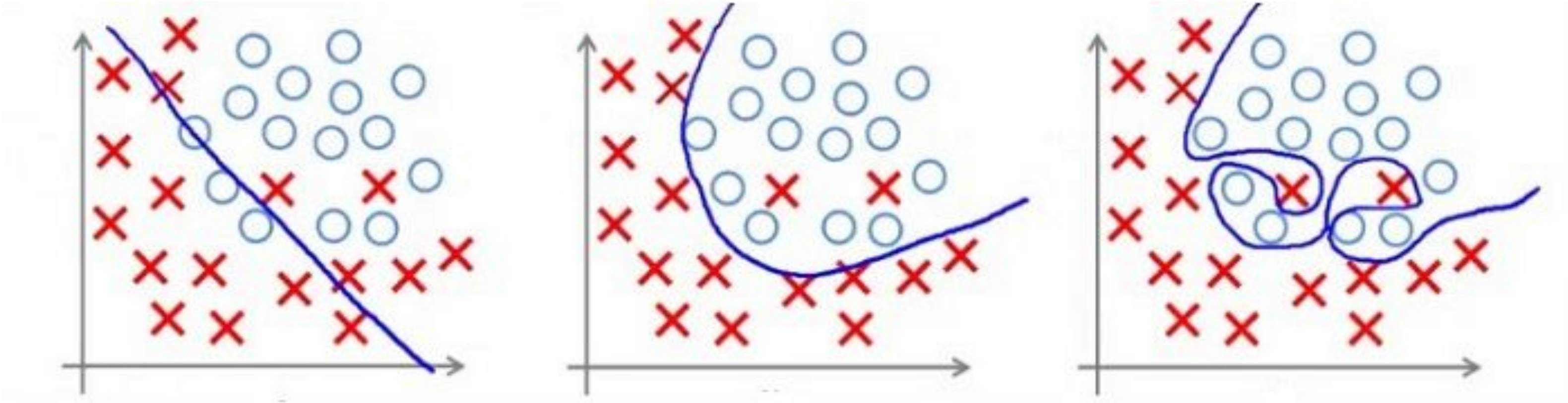
Let's denote:

- Training set $\mathcal{L} = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$, where
  - $(\mathbf{x} \in \mathbb{R}^p,\ y \in \mathbb{R}\ )$ for regression
  - $\mathbf{x}_i \in \mathbb{R}^p$, $y_i \in \{+1, -1\}$ for binary classification

Model $f(\mathbf{x})$ predicts some value for every object

Loss function $Q(\mathbf{x}, y, f)$ that should be minimized

Overfitting vs. underfitting



Under-fitting

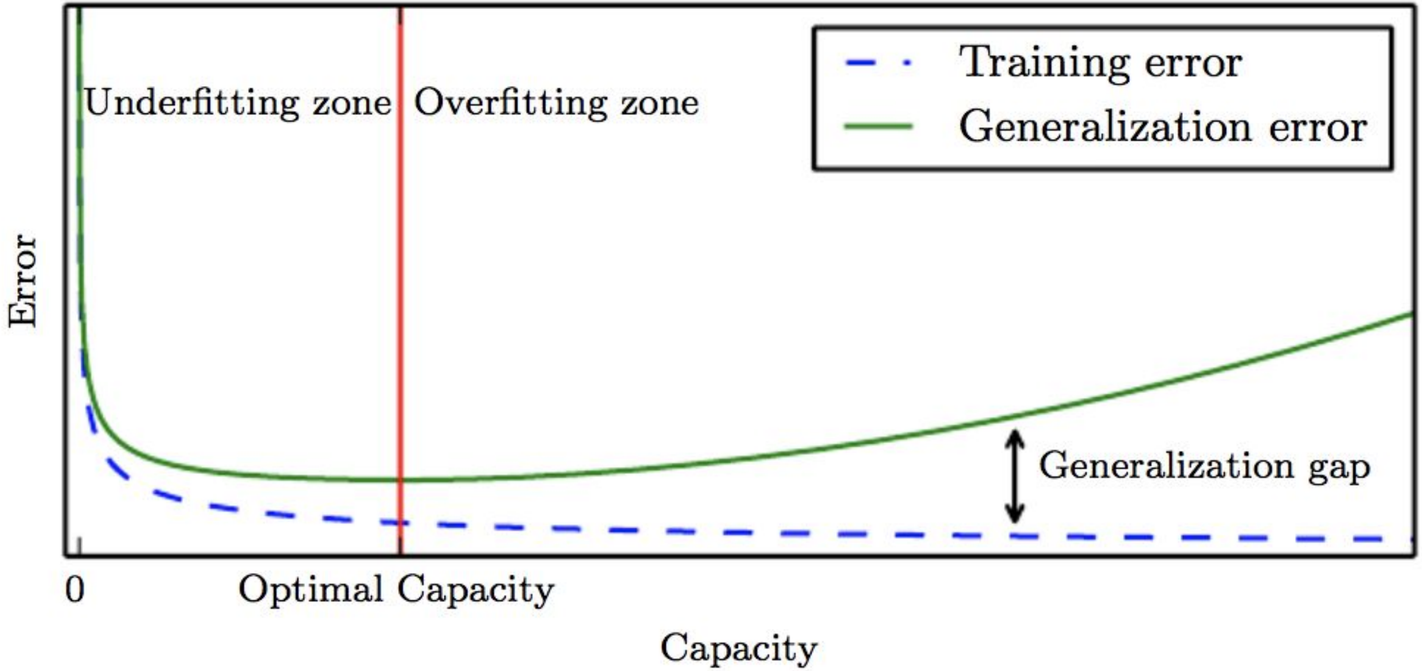
(too simple to explain the variance)

Appropriate-fitting

Over-fitting

(forcefitting -- too good to be true)

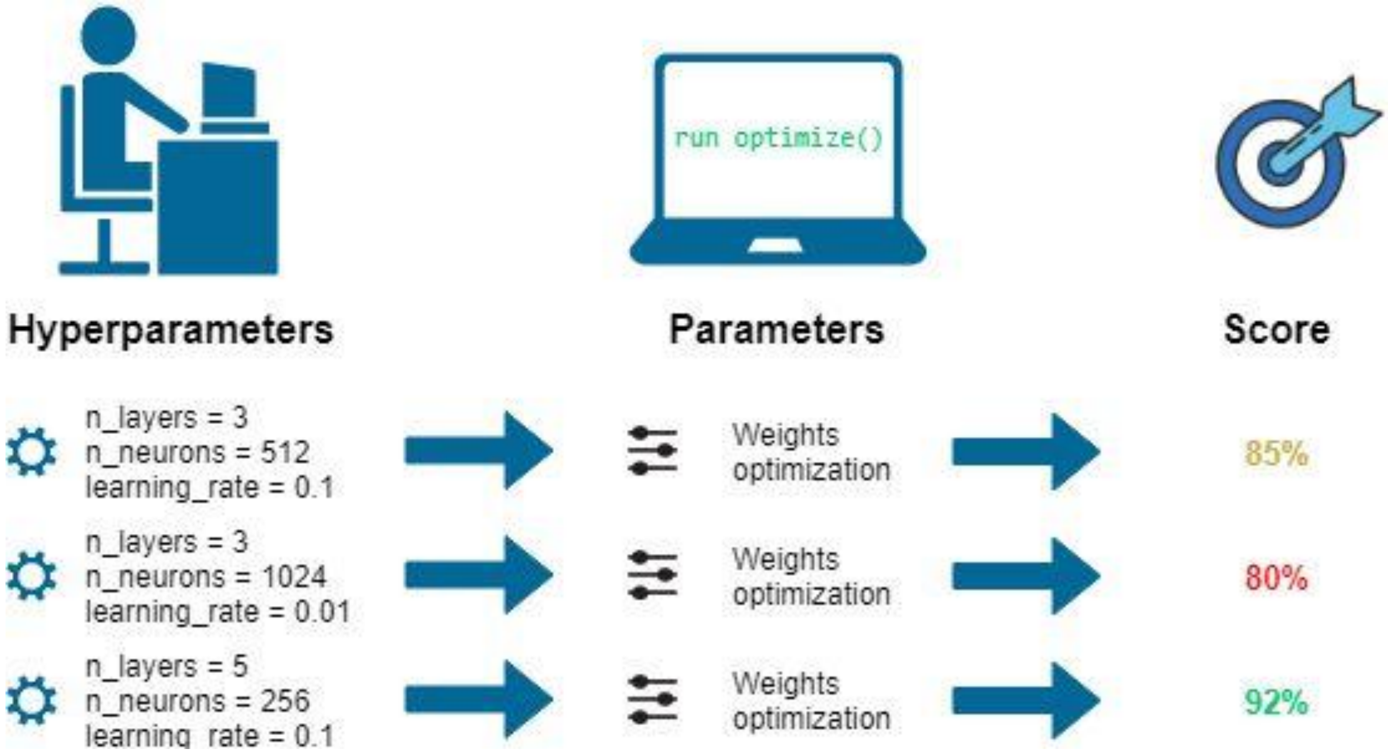
Overfitting vs. underfitting



Evaluating the quality

Is it good enough?

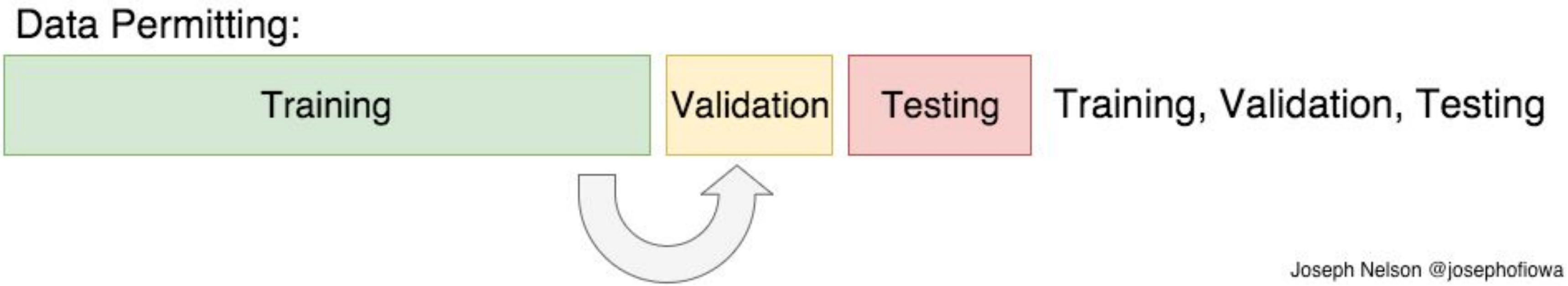
Parameters and hyperparameters



Parameters and hyperparameters: differences

	Defined by	Depend on the training data	Order of optimization methods	Required for	Affect the complexity of the model
Parameters	during the training	yes	first (gradient)	predictions	no
Hyperparameters	before the start of training	no	zero (manual, Bayesian)	training	yes

Dataset splits



Stages of model training

Split	training	validation	test
Used for	parameters optimization	hyperparameters selection	quality measurement
Overfitting level	high	average	low

Next time

- Support Vector Machines
- Principal Component Analysis
- Linear Discriminant Analysis

Thanks for attention!

Questions?

girafe ai