Machine Learning

Rudolf Mayer

mayer@ifs.tuwien.ac.at

October 11th, 2017

Outline

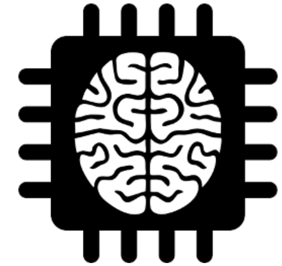
- Machine Learning: Intro
- Perceptron
- k-NN
- Performance evaluation (intro)
- Types of data & data preparation

Outline

- Machine Learning: Intro
- Perceptron
- k-NN
- Performance evaluation (intro)
- Types of data & data preparation

Defining the area

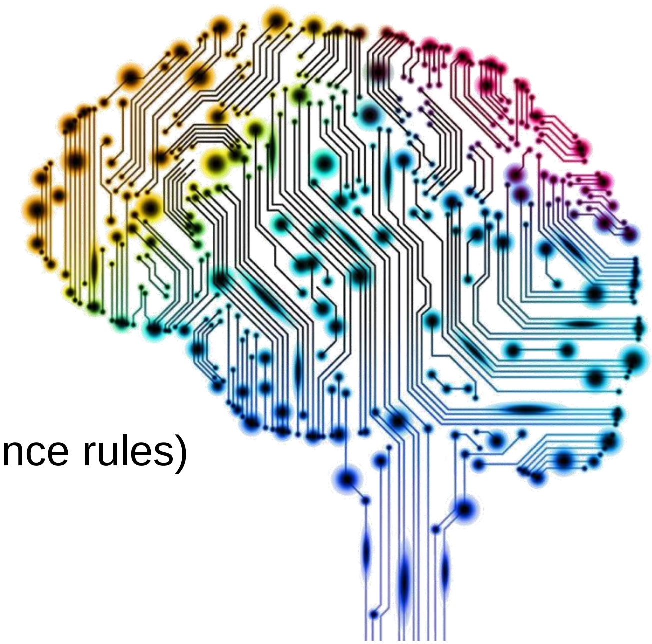
- Some related terms
  - Data Mining
  - Predictive analytics
  - Knowledge discovery
  - Data Science
  - Statistics
  - Cluster Analysis
  - Artificial Intelligence
  - Reasoning / deduction
  - Regression
  - Classification
  - Supervised/ unsupervised learning
  - ...



Artificial intelligence

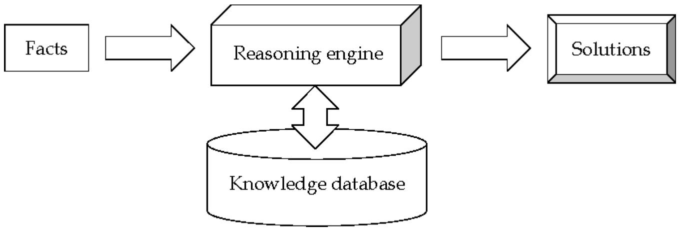
- Deals with intelligence of machines
  - *System that perceives its environment & takes actions that maximise its chances of success*

- Areas/problems of AI
  - Deduction, reasoning, problem solving
    - Often rule-based; manually created
  - Knowledge representation (reasoning)
    - Reasoning: generation of new knowledge (inference rules)
  - ***Machine learning***
  - Planning / scheduling
  - Natural language processing
  - …



Rule-based systems

- System that uses rules to make deductions or choices
  - E.g. domain-specific expert system



- Two components
  - Knowledge base: facts & rules (if → then style)
    - Knowledge representation (language)
    - Example rule: *IF (hot & smoke) THEN fire*
  - Inference engine: applies rules to deduce new facts
    - Forward chaining: assert new facts
    - Backward chaining: start with goal → determine which facts need to asserted
- Rules often manually specified (by expert)
  - Expensive, incomplete

Machine learning

- Studies computer algorithms that can ***learn*** from data and make ***predictions*** on data

- Automatic methods – no human assistance in learning (i.e. no specification of rules!)
  - Human assistance generally required for
    - Defining problem
    - Gathering and assessing data

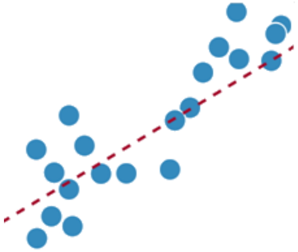
- Types of Machine Learning
  - Supervised
  - Unsupervised
  - Reinforcement

Machine learning – why?

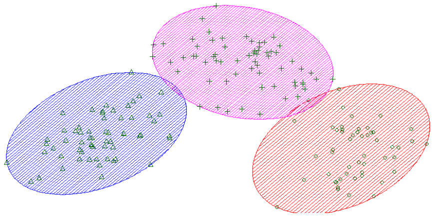
Source: Gartner (July 2016)

Defining the area

- Statistics
  - Collection, interpretation and presentation of data
  - Descriptive: summarise data (mean, standard deviation, probability density, correlation analysis)
  - Predictive modelling: e.g. regression

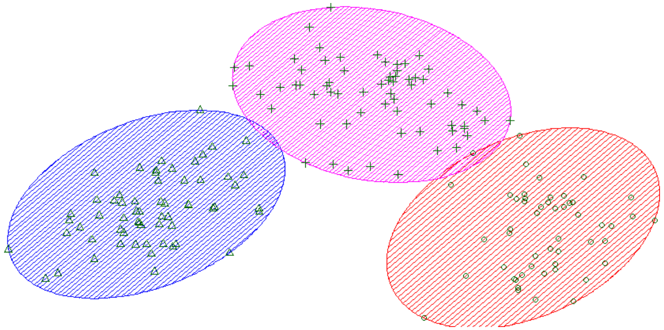


- Data Mining
  - Discovering patterns in large datasets
    - Cluster analysis, outlier detection, association rule mining



Unsupervised Learning

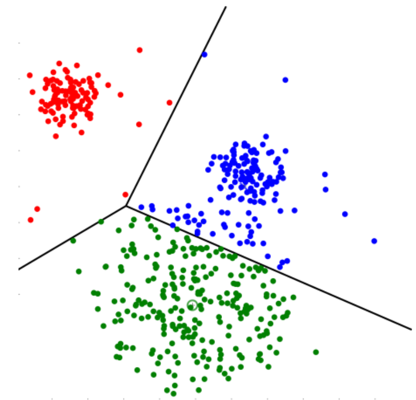
- Unsupervised learning
  - Data not ***labelled***
  - No information on which and how many classes or other structures, …
  - Goal:
    - Find structures (e.g. clustering)
    - Association rules learning
    - Find outliers (anomalies)



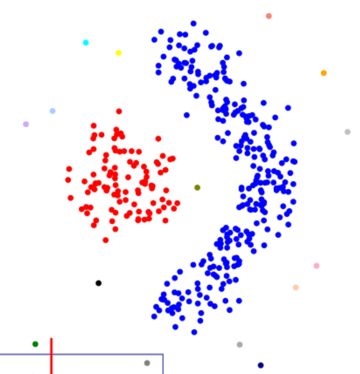
  - Most often associated with ***Data Mining***
- Covered in other lectures
  - Business Intelligence, Selbstorganisierende Systeme, ..

Cluster analysis

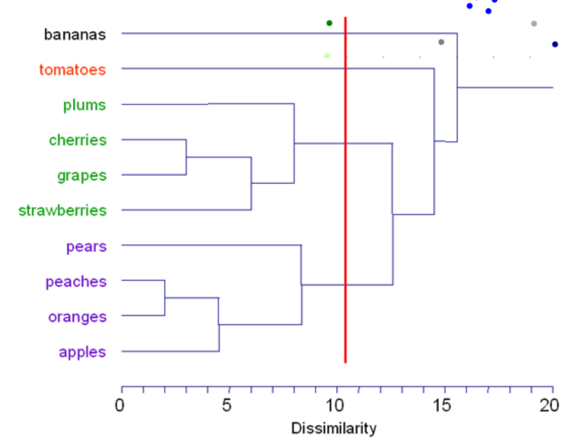
- Explorative method
- Find groups of similar objects
- Applications: e.g. market segmentation
- Several popular algorithms
  - K-means clustering (centroid based)



  - DBSCAN (density-based)



  - Linkage (hierarchical)
    - single, complete, average, Ward, ..



Cluster analysis

- Animal data set
  - Describes animal by some characteristics
  - Instances: cow, duck, cat, bee, sparrow, ...

- Characteristics
  - Size (tiny, small, medium, big)
  - Number of legs (2, 4, 6, 8)
  - Feathers (yes/no)
  - Eggs (yes/no)

	size	***legs***	***feathers***	***eggs***
duck	small	2	yes	yes
dog	medium	4	no	no
spider	tiny	8	no	yes
ladybird	tiny	6	no	yes
cow	large	4	no	no
bee	tiny	6	no	no
sparrow	small	2	yes	yes

Cluster analysis

- Animal data set
  - Describes animal by some characteristics

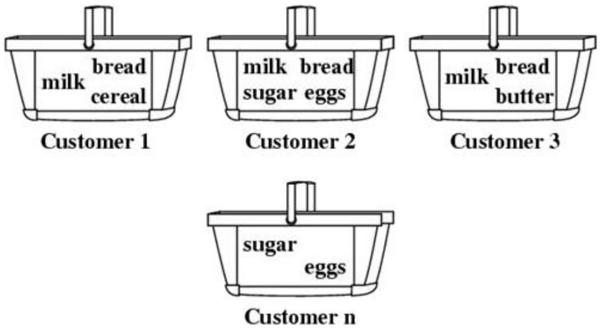
	size	*legs*	*feathers*	*eggs*
duck	small	2	yes	yes
dog	medium	4	no	no
spider	tiny	8	no	yes
ladybird	tiny	6	no	yes
cow	large	4	no	no
bee	tiny	6	no	no
sparrow	small	2	yes	yes

- Goal: find groups of related animals:
  - Mammals: cat, cow
  - Birds: duck, sparrow
  - Insects: bee, ladybird
  - Invertebrate: spider

Association rule mining

- Discover relations between variables
  - E.g. market basket analysis: which items are frequently bought together

  ➔ Useful for marketing
  - Identify rules that are
    - Frequent (“support”)
    - Reliable (“confidence”)



  - Anecdotal example: beer & diapers

  - Related: collaborative filtering / recommender systems

amazon NETFLIX

Supervised Learning

- Supervised learning
  - Data labelled with actual ***output*** variable
  - Goal: correctly label ***unknown*** data
  - Sometimes equivalently used with "machine learning"
    - In some definitions, machine learning means both unsupervised and supervised learning
    - In other definitions, unsupervised learning mostly equals/is a subset of data mining; and rather seen as part of statistics
- *Types of supervised learning?*
  - Regression & Classification

Regression vs. Classification

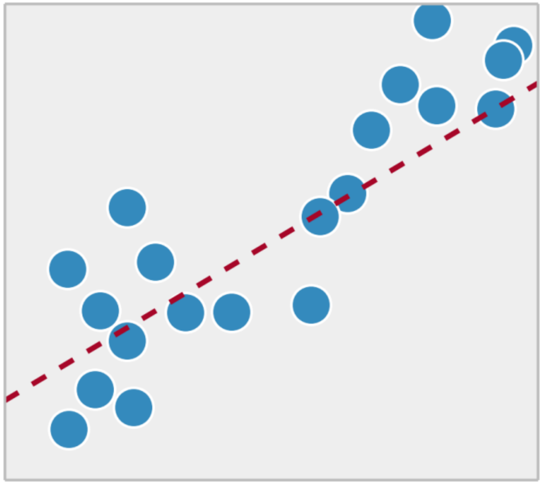
- Regression tries to predict a ***continuous*** variable

  – Examples?

Regression vs. Classification

- Regression tries to predict a ***continuous*** variable
  - *Income of a house hold* (depending on education, ...)
  - *Temperature* (depending on wind, humidity...)
  - *Housing price* (depending on e.g. size, age..)

  - Methods: e.g. *linear regression*
    - Mostly associated with statistics
  - Many classification methods can also be used for regression

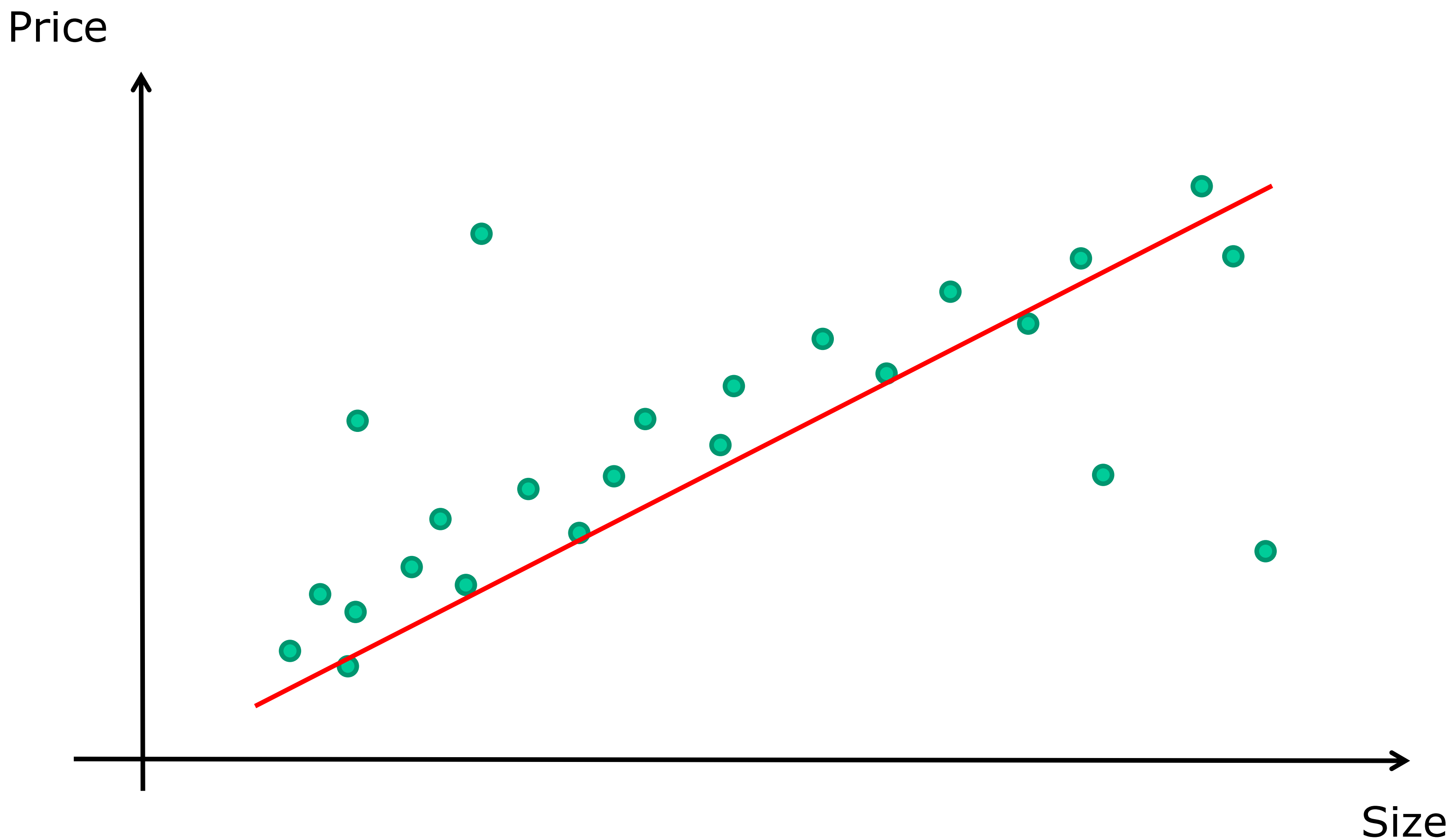


Regression example

- Data set about prices of houses (apartments)
- Input data e.g.
  - Size
  - Number of rooms
  - Size of garden/balcony/terrace
  - Year built / age of building
  - Location
- Goal: predict the expected price of the house in €
- Solution: multivariate regression
  - *If only one input: univariate regression*

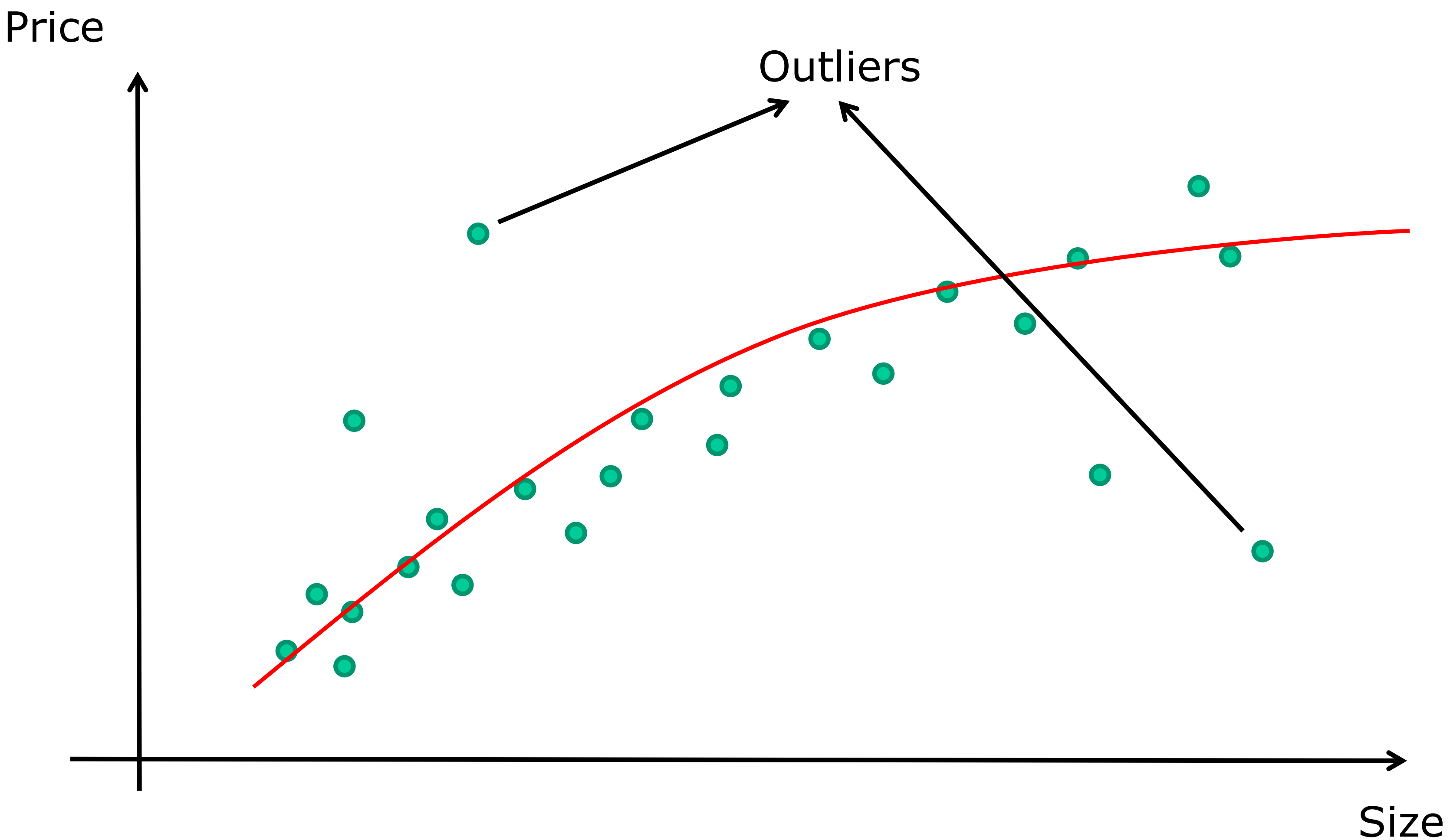
Regression example

Linear regression – find a “line” that describes the data



Regression example

Polynomial regression – find a "curve" that describes the data

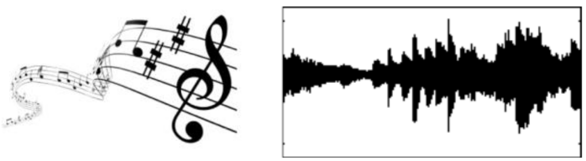


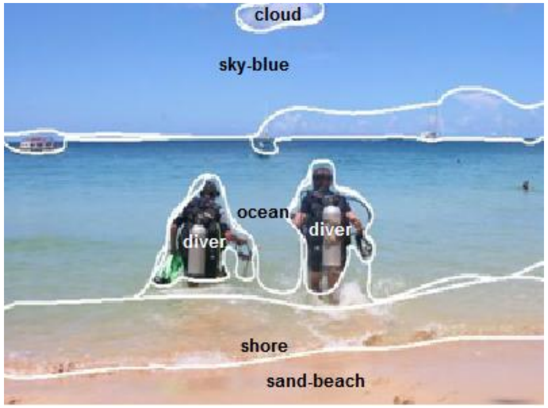
Regression vs. Classification

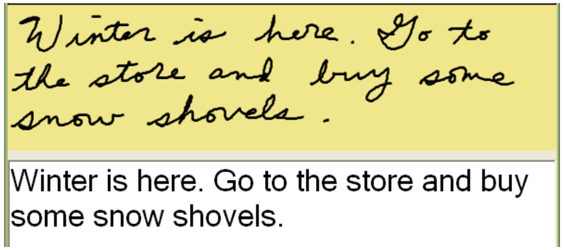
- If the output variable can take one of a predefined set of values: ***classification*** *(also: categorisation)*
  - Examples?

Regression vs. Classification

- If the output variable can take one of a predefined set of values: ***classification*** *(also: categorisation)*



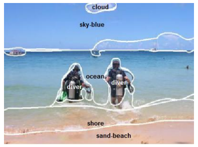




Regression vs. Classification

- If the output variable can take one of a predefined set of values: ***classification*** *(also: categorisation)*
  - ***Information***: SPAM filtering

  - ***Image***: classification of hand-written letters for OCR; automatic labelling of images



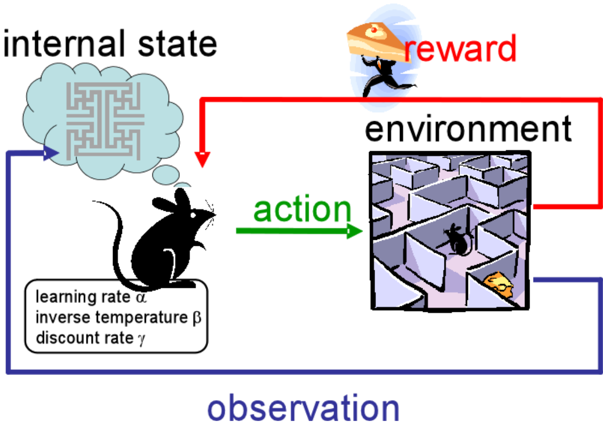
  - ***Music***: classification of music into genres

  - ***Medicine***: classification of whether a person has an illness, based e.g. on x-ray images, or other measurements

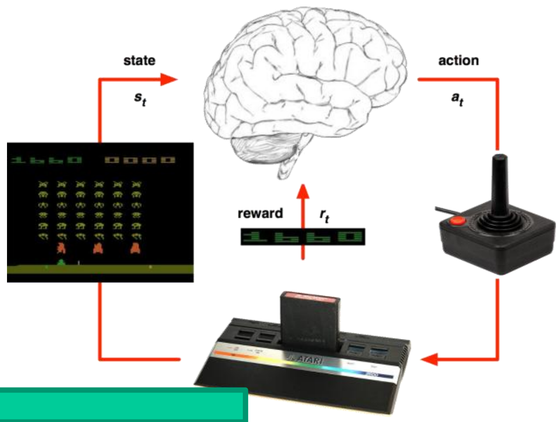
Reinforcement Learning

- Agent (e.g. computer program) takes actions in specified environment
  - – Not explicitly presenting input/output pair
  - – Reward (penalise) agent for actions

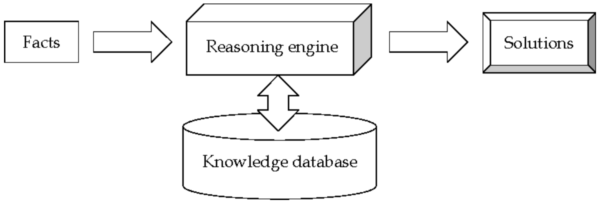
  ➔ Maximise cumulative reward



- Popular applications?

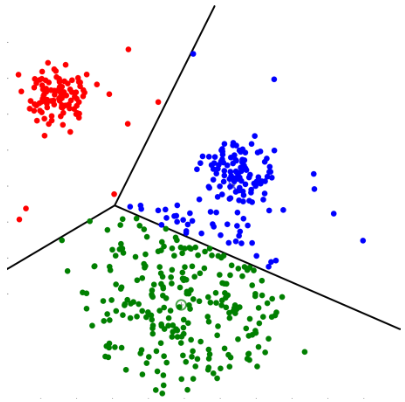


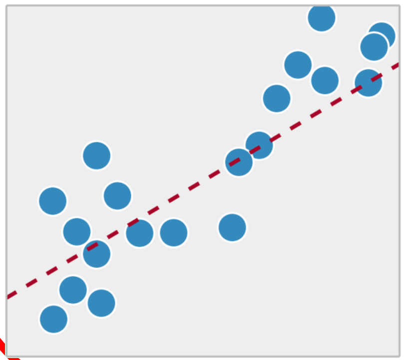
Rule-based systems

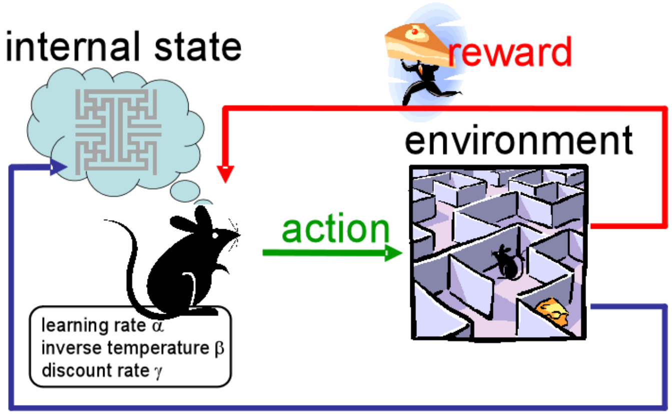


- *Rules often manually specified (by expert)*
- Alternative: learning rules
  - *Association rule mining – unsupervised*
  - Learning classifier systems
    - Can be supervised learning
    - Includes steps to discover new rules, selection of best rules, ...
- Related/similar approaches:
  - Ontologies (OWL: Web Ontology Language) & reasoning
  - *Bayesian Networks*

Scoping the lecture







Classification: Setting

- Example: data set describing characteristics of patients
  - Gender
  - Age
  - Smoker yes/no
  - Eye colour

- Want to predict whether a person has lung cancer
  - Available: some data with a label / annotation (cancer yes(no)

Classification: Setting

Attributes / Variables / Features

Output Variable

Training Samples

gender	age	smoker	eye colour
male	19	yes	green
female	44	yes	grey
male	49	yes	blue
male	12	no	brown
female	37	no	brown
female	60	no	brown
male	44	no	blue
female	27	yes	brown
female	51	yes	green
female	81	yes	grey
male	22	yes	brown
male	29	no	blue

lung cancer
No
yes
yes
no
no
yes
no
no
yes
no
no
no

Ground Truth / Gold standard

male	77	yes	grey
male	19	yes	green
female	44	no	grey

?
?
?

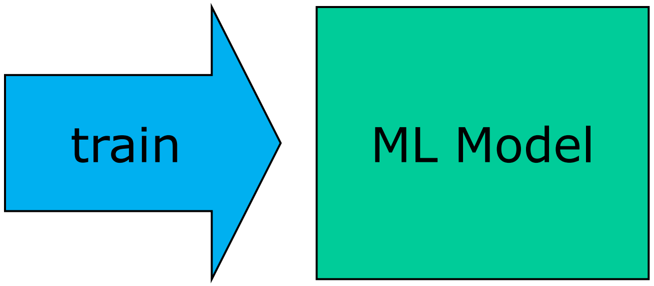
Prediction

Unlabelled samples

Classification: Setting

gender	age	smoker	eye colour
male	19	yes	green
female	44	yes	grey
male	49	yes	blue
male	12	no	brown
female	37	no	brown
female	60	no	brown
male	44	no	blue
female	27	yes	brown
female	51	yes	green
female	81	yes	grey
male	22	yes	brown
male	29	no	blue

lung cancer
No
yes
yes
no
no
yes
no
no
yes
no
no
no



male	77	yes	grey
male	19	yes	green
female	44	no	grey

?
?
?

Classification: Setting

gender	age	smoker	eye colour
male	19	yes	green
female	44	yes	grey
male	49	yes	blue
male	12	no	brown
female	37	no	brown
female	60	no	brown
male	44	no	blue
female	27	yes	brown
female	51	yes	green
female	81	yes	grey
male	22	yes	brown
male	29	no	blue

lung cancer
No
yes
yes
no
no
yes
no
no
yes
no
no
no

train

ML Model
(e.g. Neural Network)

Apply Model
➔ predict

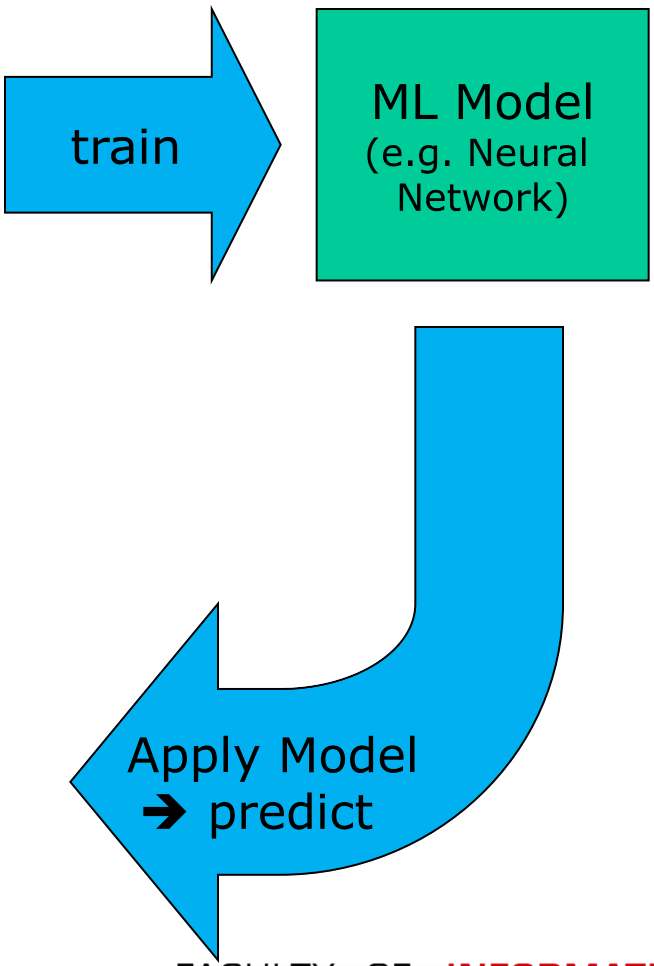
male	77	yes	grey
male	19	yes	green
female	44	no	grey

yes
no
yes

Regression: Setting

size	age	location	rooms	price
70	19	good	3	370000
60	44	good	2	165000
120	49	good	5	575000
90	12	bad	3	225000
80	37	bad	3	268000
45	60	bad	2	159000
52	44	bad	2	185000
85	27	good	4	472000
50	51	good	1	164000
35	81	good	1	155000
75	22	good	3	391000
95	29	bad	4	478000

35	77	good	4	274000
25	19	good	1	122000
85	44	bad	2	162000



Machine Learning: Setting

gender	age	smoker	eye colour	lung cancer
male	19	yes	green	No
female	44	yes	grey	yes
male	49	yes	blue	yes
male	12	no	brown	no
female	37	no	brown	no
female	60	no	brown	yes
male	44	no	blue	no
female	27	yes	brown	no
female	51	yes	green	yes
female	81	yes	grey	no
male	22	yes	brown	no
male	29	no	blue	no

male	77	yes	grey	?
male	19	yes	green	?
female	44	no	grey	?

- *Where do we get the data from?*

Machine Learning: Setting

gender	age	smoker	eye colour
male	19	yes	green
female	44	yes	grey
male	49	yes	blue
male	12	no	brown
female	37	no	brown
female	60	no	brown
male	44	no	blue
female	27	yes	brown
female	51	yes	green
female	81	yes	grey
male	22	yes	brown
male	29	no	blue

lung cancer
No
yes
yes
no
no
yes
no
no
yes
no
no
no

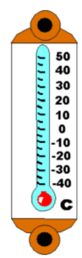
male	77	yes	grey
male	19	yes	green
female	44	no	grey

?
?
?

- Features:
  - Observation / measurement
  - *Extraction from media*
  - *(Transformation)*
- Labels (groundtruth):
  - Human experts
  - (Observation / measurement)

Machine Learning: Setting

Observation / Measurements

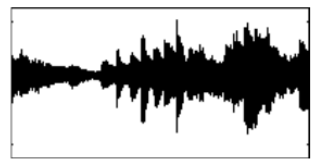


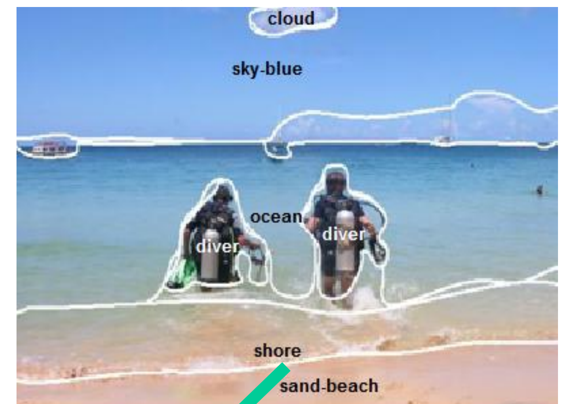
Annotations
(human experts)

Measurement/observation

Machine Learning: Setting

Feature Extraction





Human experts

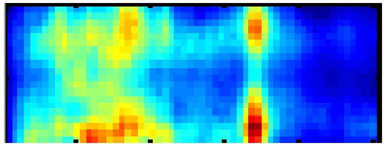
Machine Learning: Setting

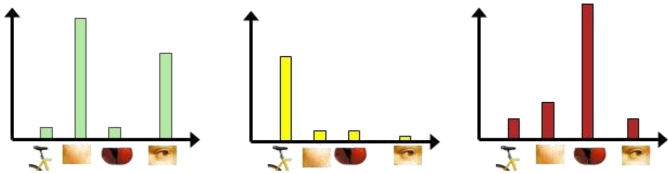
- Feature Extraction: description of complex content by derived values
  - Text: Bag of Words – counting term occurrences
  - Music: counting activity on various frequencies
  - Image: colour histograms, edge detection, “bag of visual words”, ...

- Important aspect of many data mining applications
  - Overview/intro in this course
  - *More details e.g. Information retrieval (188.412), ...*

Machine Learning: Setting

	Word 1	Word 2	Word 3	Word 4	Word 5	…	Word n	∑
Doc 1	1	0	0	2	3		0	6
Doc 2	2	0	0	0	2		2	6
Doc 3	1	3	0	0	1		5	10
Doc 4	2	0	0	2	0		2	6
Doc 5	5	4	0	0	1		0	10
…							0	
Doc m	1	2	1	3	0		0	7

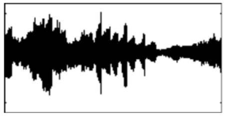


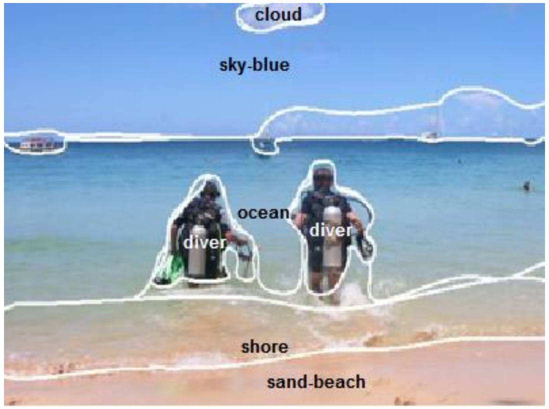


Text Feature Extraction

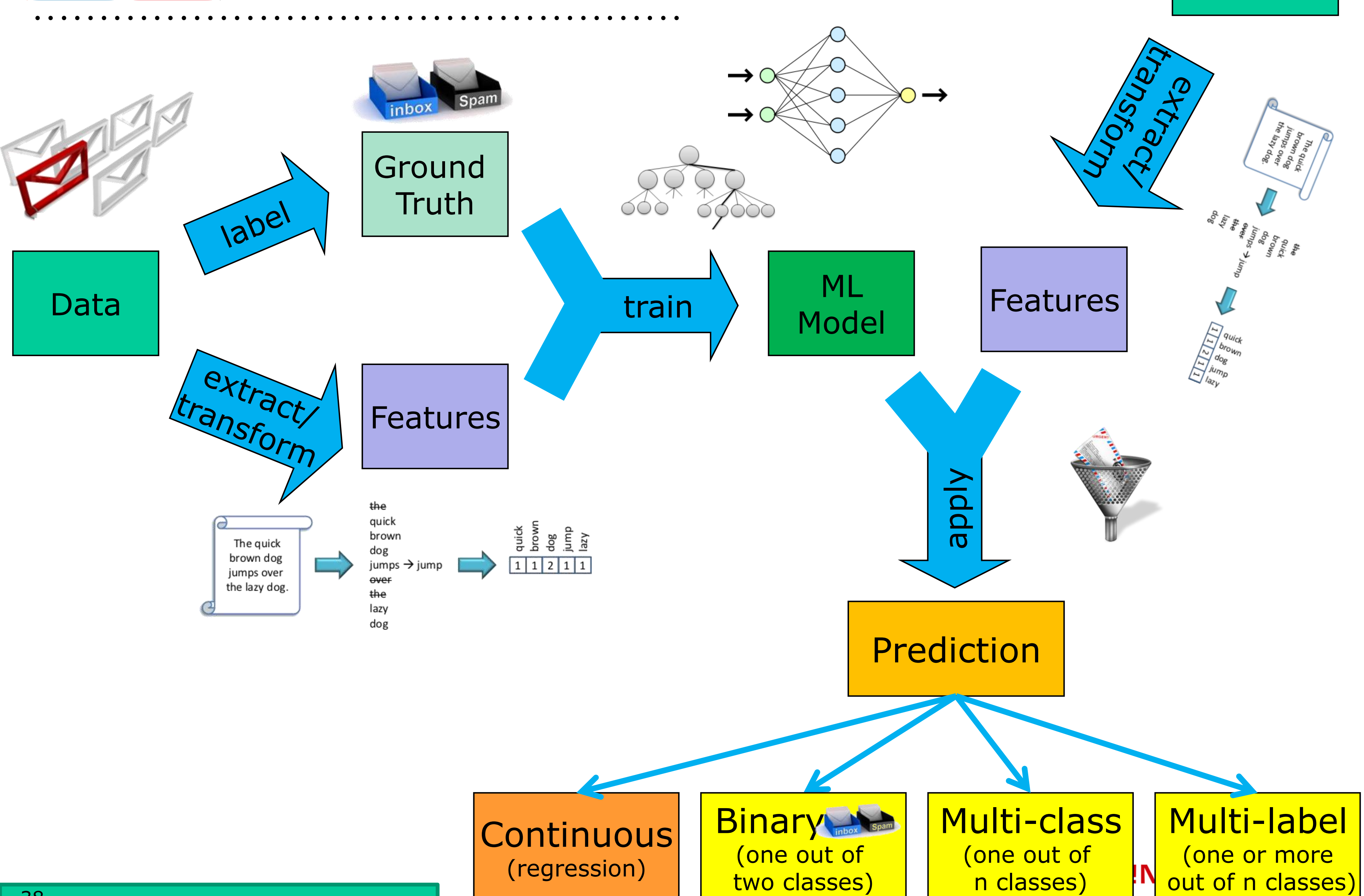
Music Feature Extraction

Image Feature Extraction





Machine Learning: steps



Machine Learning: Setting

- Feature Extraction: description of complex content by derived values
  - Text: Bag of Words – counting term occurrences
  - Music: counting activity on various frequencies
  - Image: colour histograms, edge detection, ...

- Important aspect of many ML applications
  - "Previews" in this lecture
  - More detailed, e.g. 188.412 "Information Retrieval"

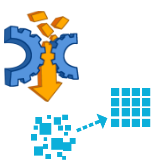
Active Learning

- Algorithm can query the user to obtain labels for (unlabelled) specific samples (data points)
  - Often “difficult” samples, where the algorithm “is not sure” on the output variable
- Motivation?
  - Labelling data is expensive...
  - Goal: obtain a good model with less effort for labelling data
  - E.g. by selecting those samples that are difficult to decide for the algorithm (least certainty)
- Semi-Supervised learning

General analytical process

Independent of exact learning (data mining) method:

1. Identify problem / question
2. Identify & capture the available data



3. Prepare your data: clean & transform
4. Analyse your data ➔ actual machine learning

5. Create report with results, visualisation, insights
6. Embed results into business / decision making

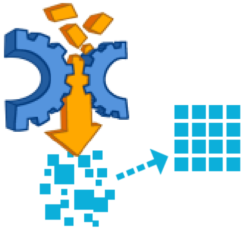
7. Plan for better data capturing in the future

General analytical process

1. Identify problem / question
   – What is the problem
   – Why does it need to be solved
   – How can it be solved **(as machine learning problem!)**

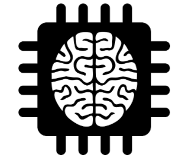
2. Identify & capture data
   – Select your data
   – Capture / extract your data

3. Clean & transform your data
   – Deal with missing values
   – Transform data: scale/normalise, attribute selection, ...



General analytical process

4. Analyse your data
   – Select fitting algorithms
   – Train and evaluate models, select best performing
   – Improve your results – parameter tuning, ...



5. Document & analyse your results
   – Problem definition, solutions, approach
   – Present & compare your results in graphs, tables, ...
   – Discoveries made during investigation
   – Methods that did / did not work
   – Limitations: when does the model not work? What questions can not be answered?

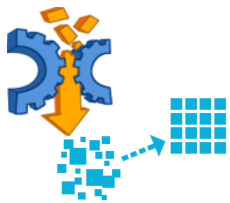
General analytical process

6. Embed results into business / decision making
   - Directly adjust your operation depending on the result
   - Use results for decision making
     - At various levels of the organisation

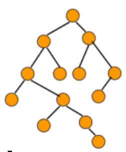
7. Plan for better data capturing in the future
   - Capture more data than currently available
   - Capture data in higher resolution (more detail, more frequent, ...)
   - Capture data in better quality

Scoping the lecture

Data types & feature extraction



Data preparation & transformation



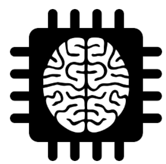
Decision Trees & Random Forests

Support Vector Machines

Neural Nets & Deep Learning

Ensemble learning

...



Evaluation

Model selection

Significance testing

Scoping the lecture

- Matching your expectations?

  - Anything not interesting at all?

  - Anything missing?

Outline

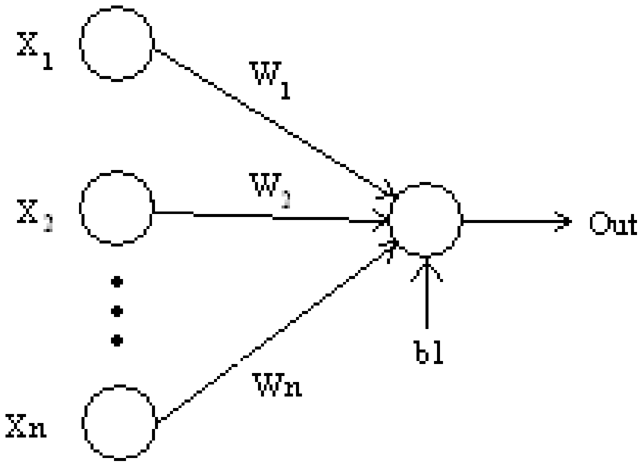
- Machine Learning: Intro
- Perceptron
- k-NN
- Performance evaluation (intro)
- Types of data & data preparation

Perceptron

- Proposed in 1957
- Artificial (neural) network
  - only one neuron; “single-layer perceptron”

- Input: continuous values X $(x_1, x_2, \ldots, x_n)$

- Output: activation a
  - Boolean value 0 / 1



Perceptron

- Linear combination of inputs
  - Using weights *W*

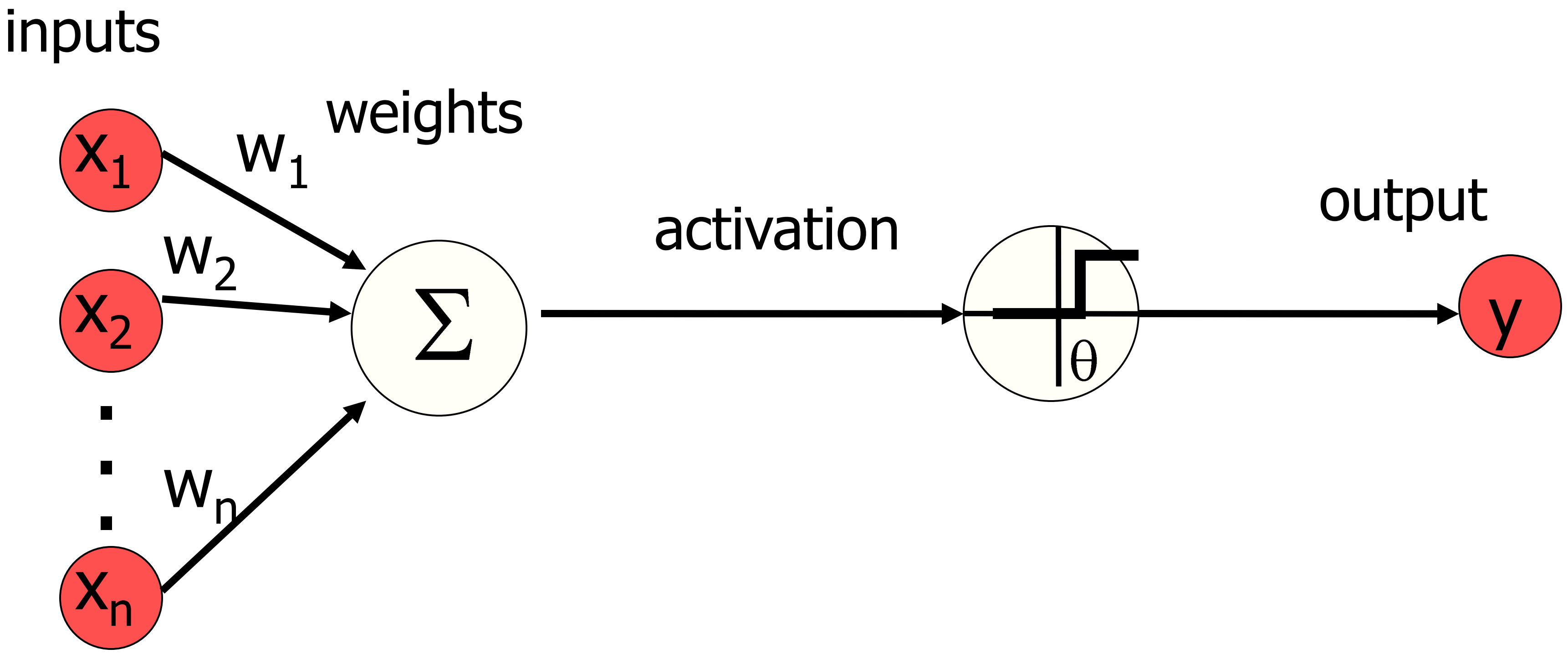
$$a = \sum_{i=1}^{n} w_i x_i$$

- Pass through threshold activation function with threshold θ

$$y = f(x) = \begin{cases} 1 \text{ if } a \geq \theta \\ 0 \text{ if } a < \theta \end{cases}$$

(Heaviside step function)

Perceptron

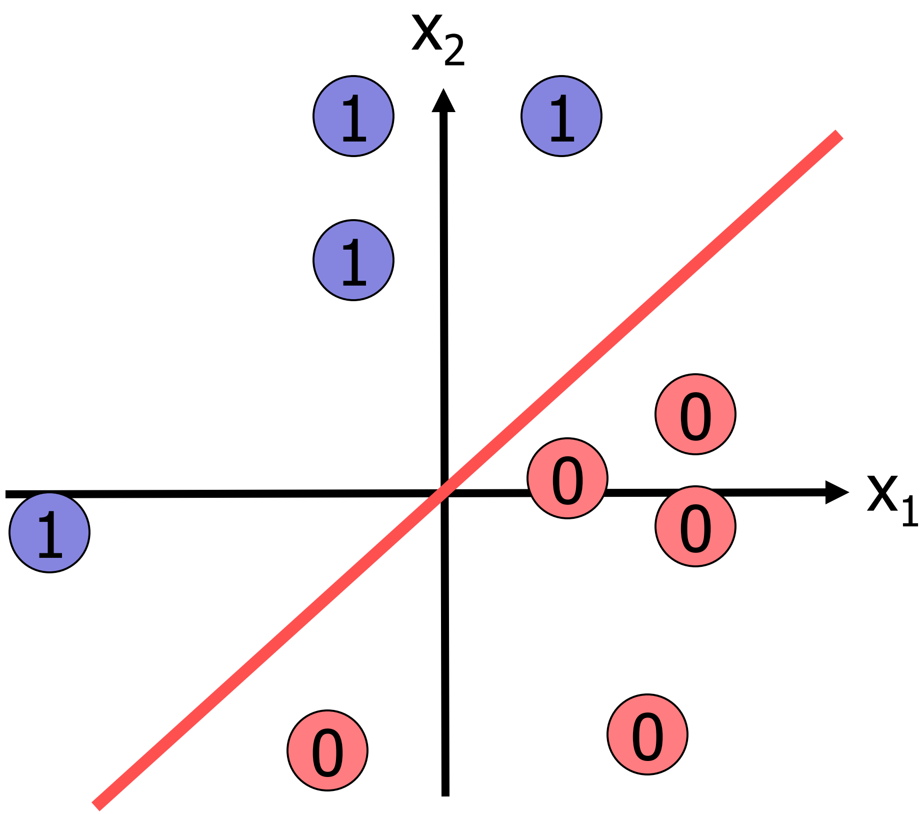


inputs
weights
x_1
w_1
w_2
x_2
w_n
x_n
Σ
activation
θ
output
y

Perceptron: visual example

- Two-dimensional data set
  - Variables x_1 & x_2
    - E.g. position on a map
- Weight vector also has two components
  - w_1 & w_2
- Data set can be easily visualised in Cartesian coordinate system
- Two classes: ***0*** and ***1***
  - E.g. **poor** and **rich** countries

Perceptron visualised



Decision line

$w_1\ x_1 + w_2\, x_2 = \theta$

Perceptron: learning

- Training the model: learning the weights from labelled samples (label: **t**)

  - Initialise weights

  - Repeat
    - Present training sample **x**
    - Predict sample label: **y** = f(x)
    - Prediction correct? Compare **t** and **y**
      – if **y ≠ t** ➔ Compute new weights **w'** as **w'**=**w** + α (t-y) **x**

  - Until prediction correct (y=t) for all samples

Perceptron: learning

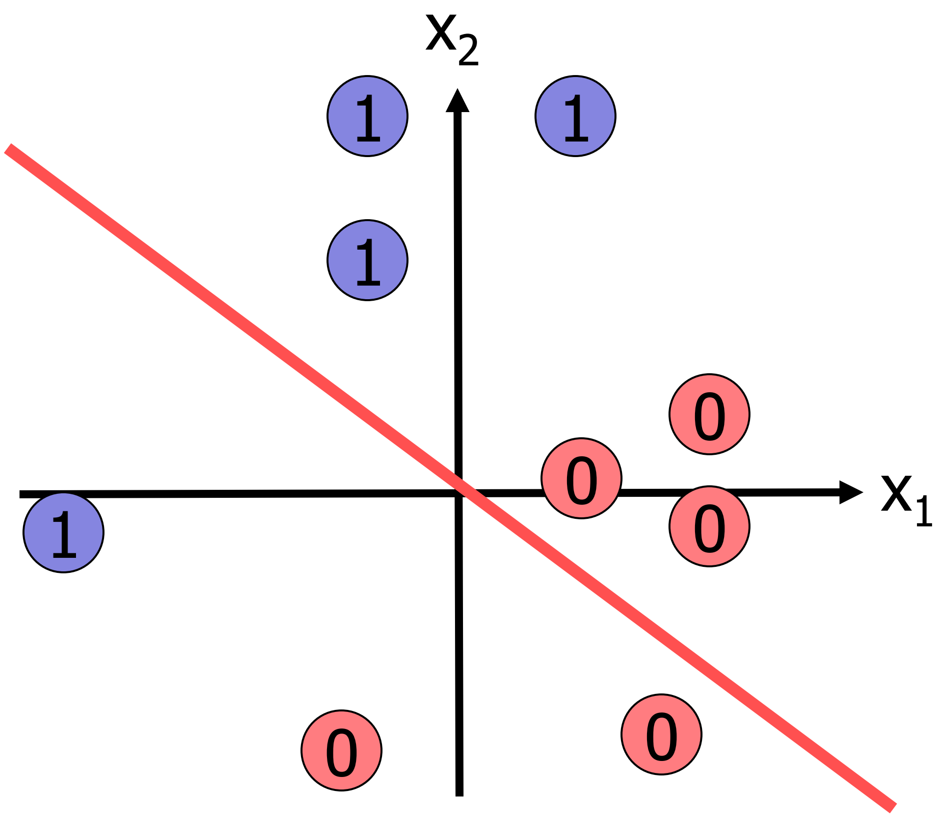
- Parameter α: *learning rate*
  - determines the magnitude of weight updates

- If output is correct (t=y)
  - weights **not** changed

- If output is incorrect (t ≠ y)
  - weights changed such that the output of the for the new weights **w’** is ***closer*** to the input x_i

Perceptron training

- t=0: random weight



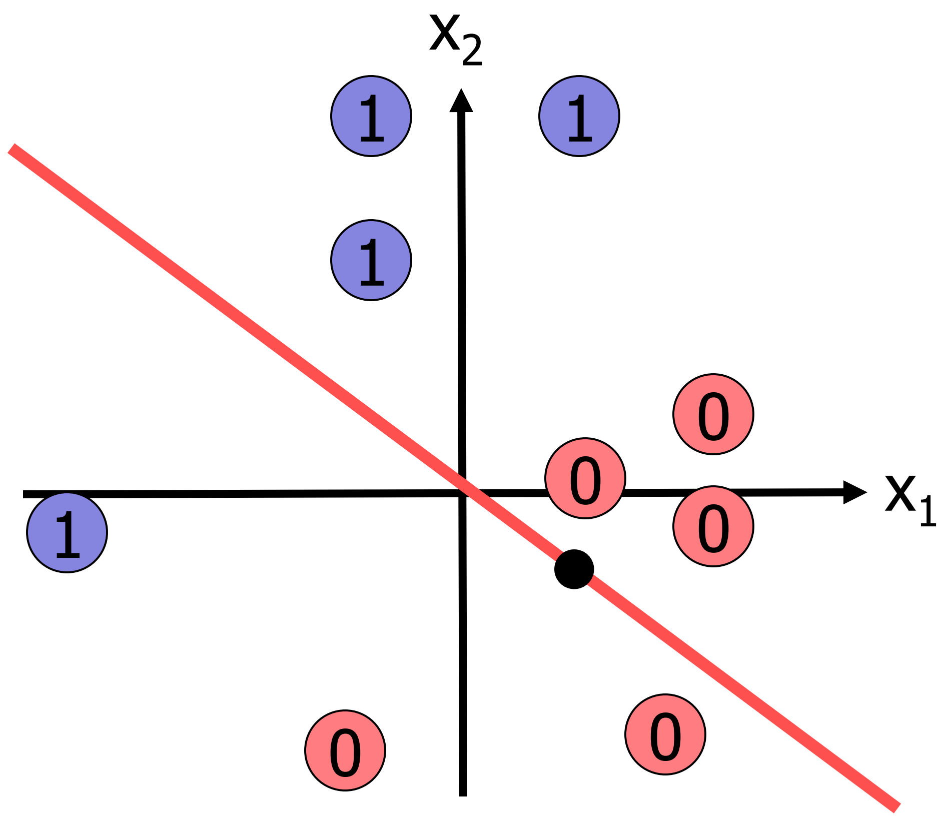
Decision line

$w_1\, x_1 + w_2\, x_2 = \theta$

(cf. y = ax + b)

Perceptron training

- t=0: random weight



Decision line

$w_1 x_1 + w_2 x_2 = \theta$

Initial weight vector:

$W_1=1, W_2=1$

Point on decision line

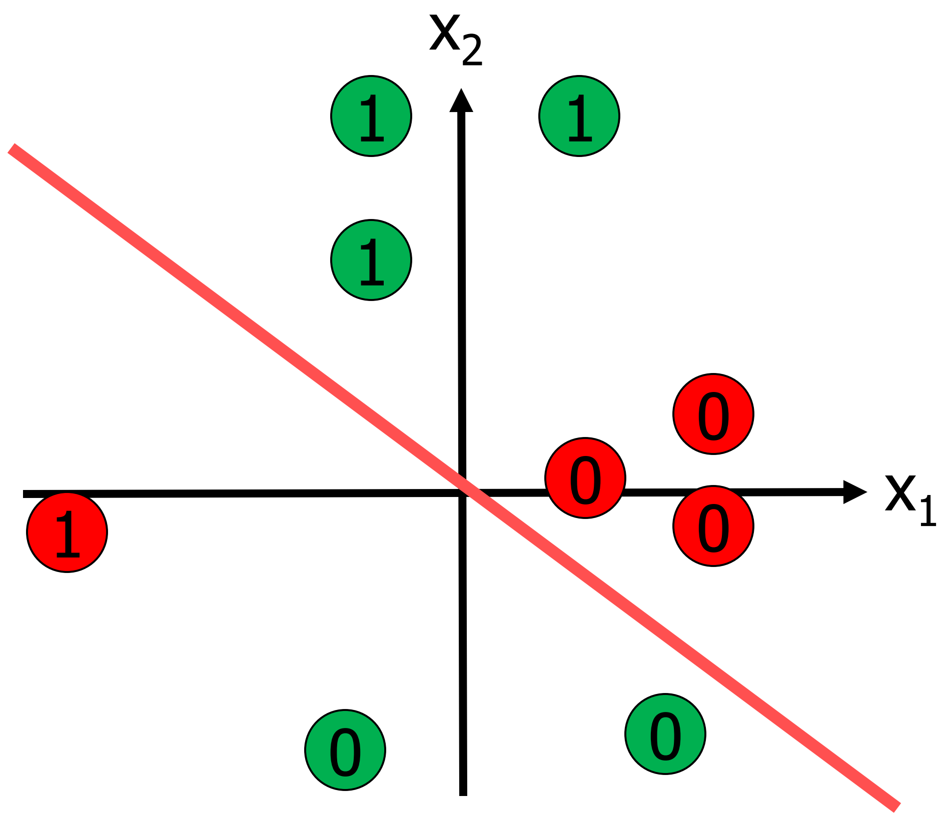
(1,-1)

➔ 1*1 +1*(-1) =

= 1 - 1 = 0 = θ

Perceptron training

- t=1: compute y = f(x), compare to *t*

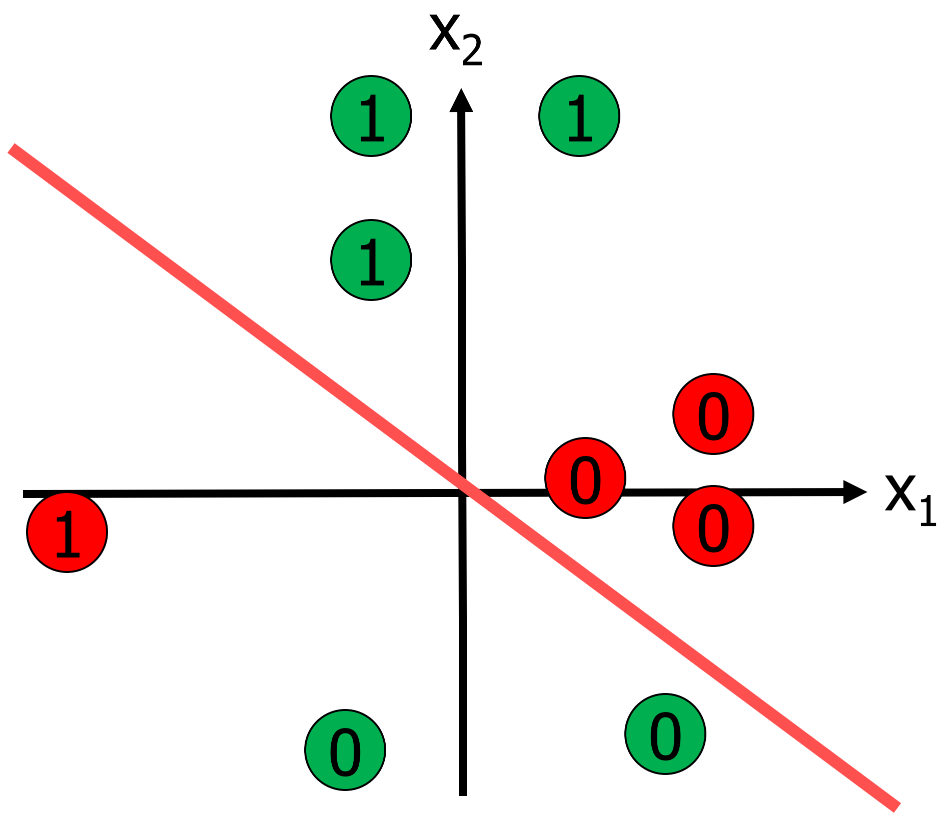


Decision line
$w_1 x_1 + w_2 x_2 = \theta$

(green: correctly , red: wrongly classified)

Perceptron training

- t=1: not all f(x) = t ➔ adapt weights (decision line)



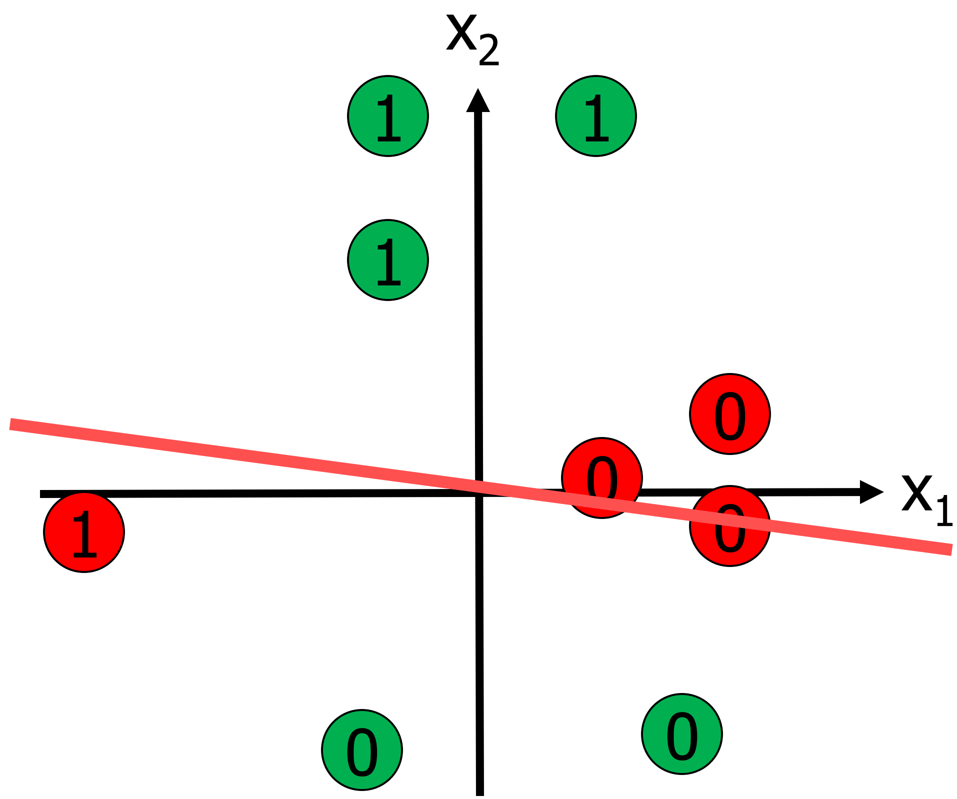
Decision line

$w_1 x_1 + w_2 x_2 = \theta$

(green: correctly , red: wrongly classified)

Perceptron training

- t=1: not all f(x) = t ➔ adapt weights (decision line)



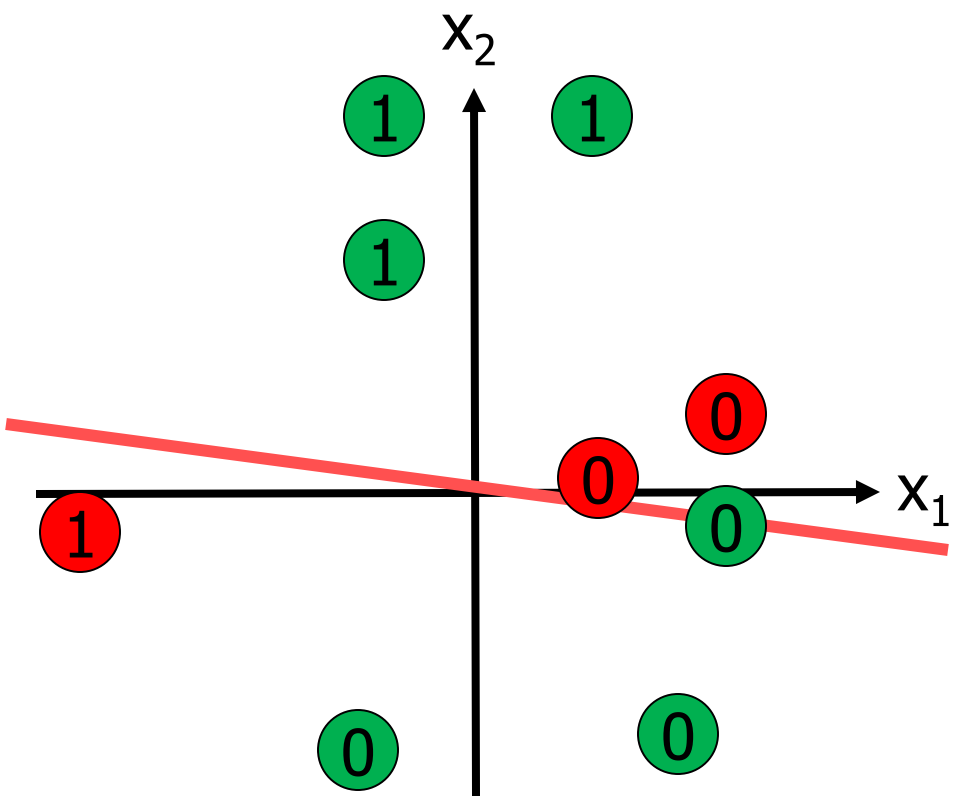
Decision line

$w_1 x_1 + w_2 x_2 = \theta$

(green: correctly , red: wrongly classified)

Perceptron training

- t=2: not all f(x) = t ➔ adapt weights (decision line)



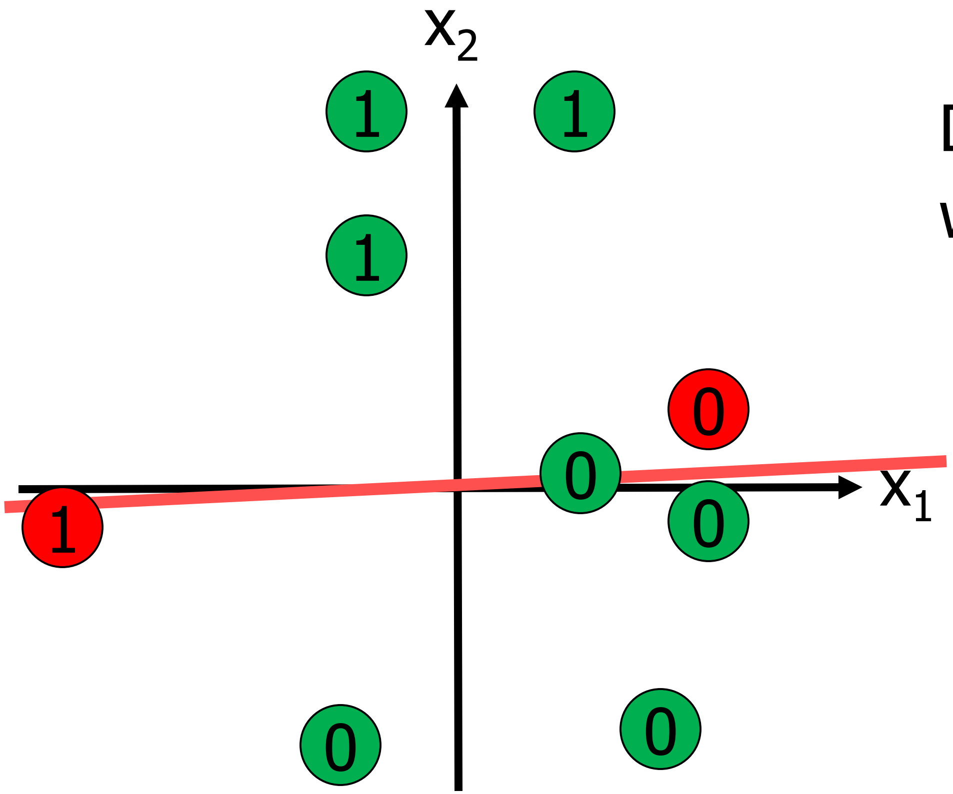
Decision line

$w_1 x_1 + w_2 x_2 = \theta$

(green: correctly , red: wrongly classified)

Perceptron training

- t=3: not all f(x) = t ➔ adapt weights (decision line)



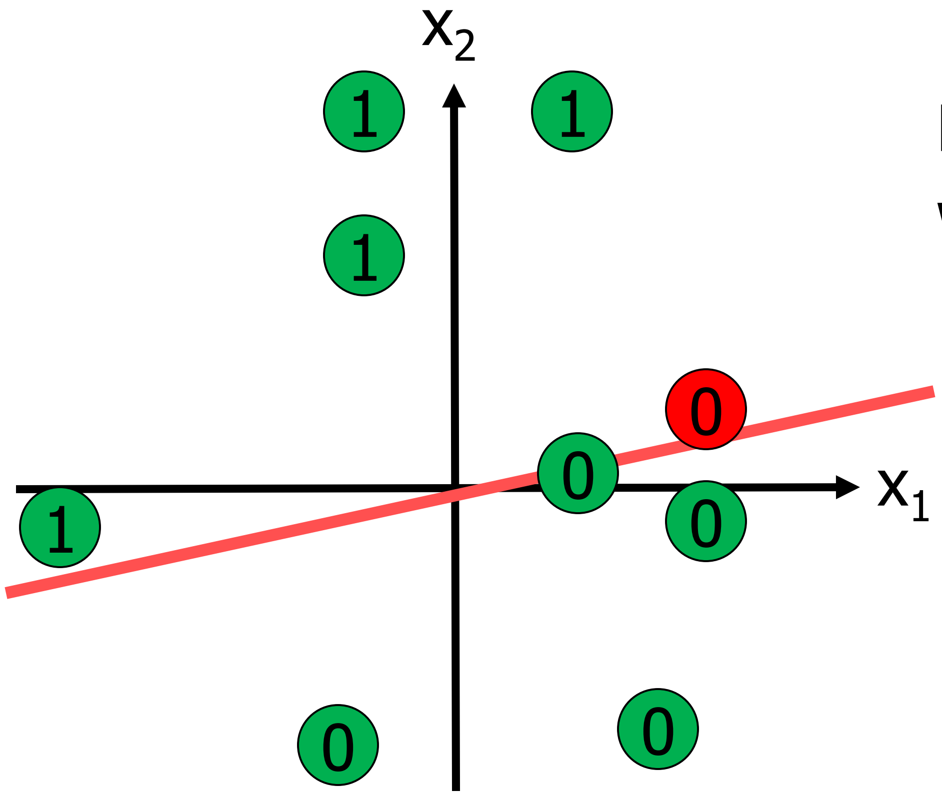
Decision line

$w_1\, x_1 + w_2\, x_2 = \theta$

(green: correctly , red: wrongly classified)

Perceptron training

- t=4: not all f(x) = t ➔ adapt weights (decision line)



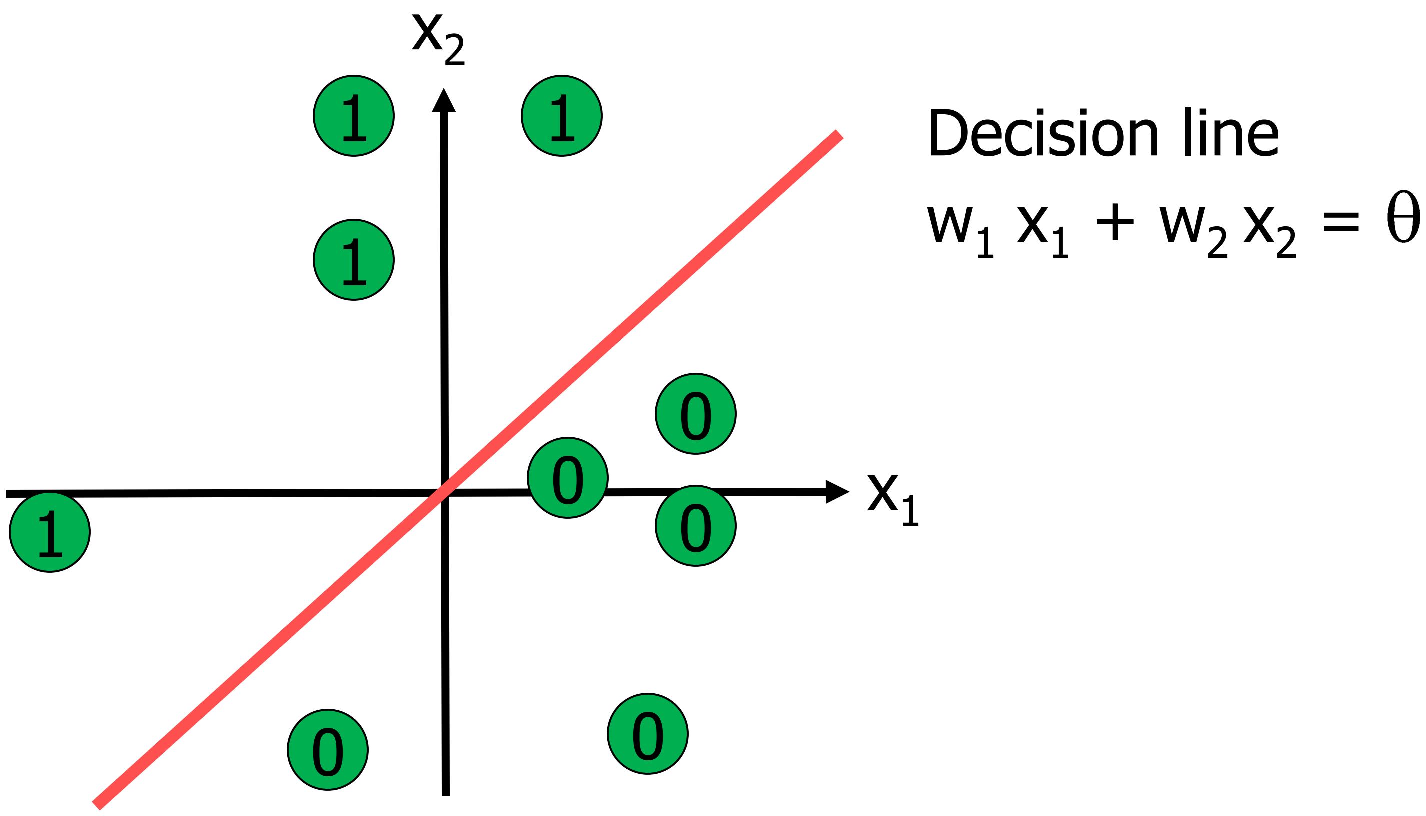
Decision line

$w_1\, x_1 + w_2\, x_2 = \theta$

(green: correctly , red: wrongly classified)

Perceptron training

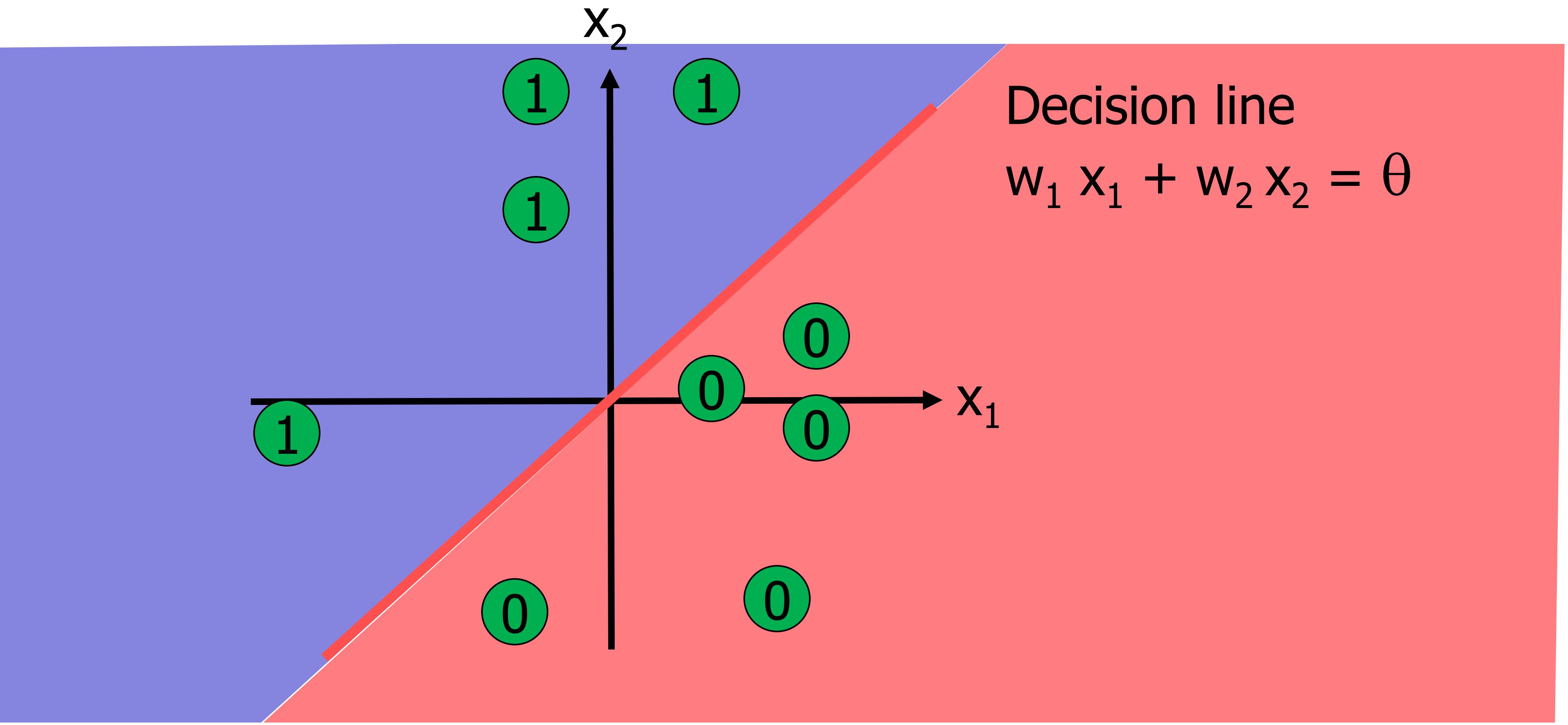
- t=5: all f(x) = t ➔ final state



(green: correctly , red: wrongly classified)

Perceptron training

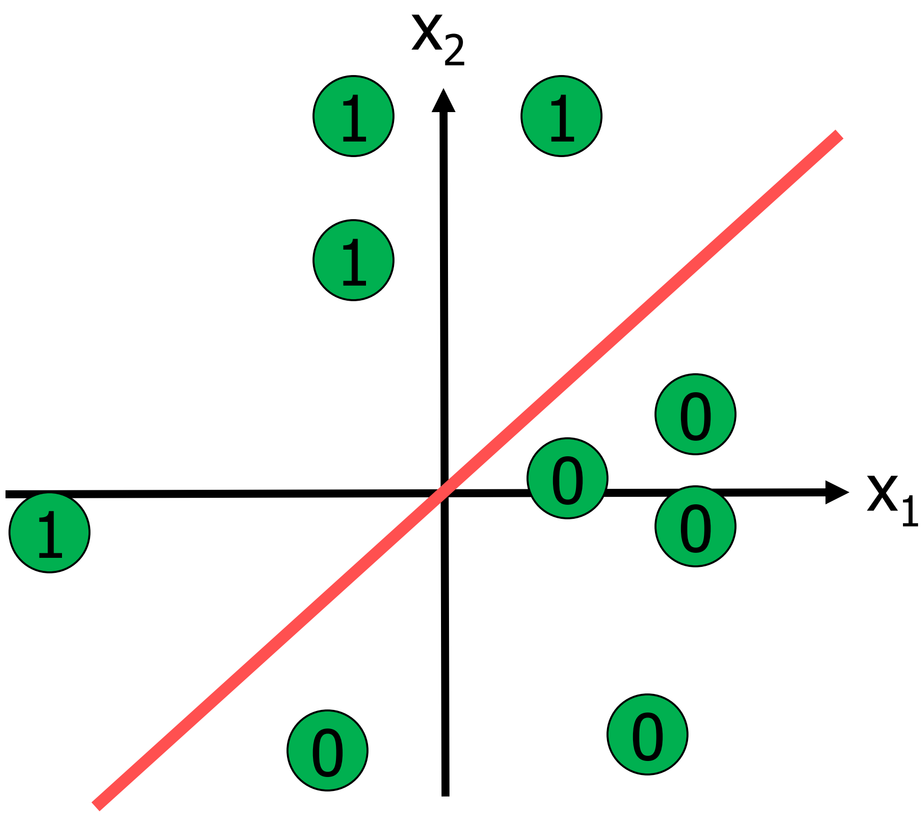
- t=5: all f(x) = t ➔ final state; classification in **rich**/**poor**



(green: correctly , red: wrongly classified)

Perceptron training

- t=5: all f(x) = t ➔ final state



Decision line

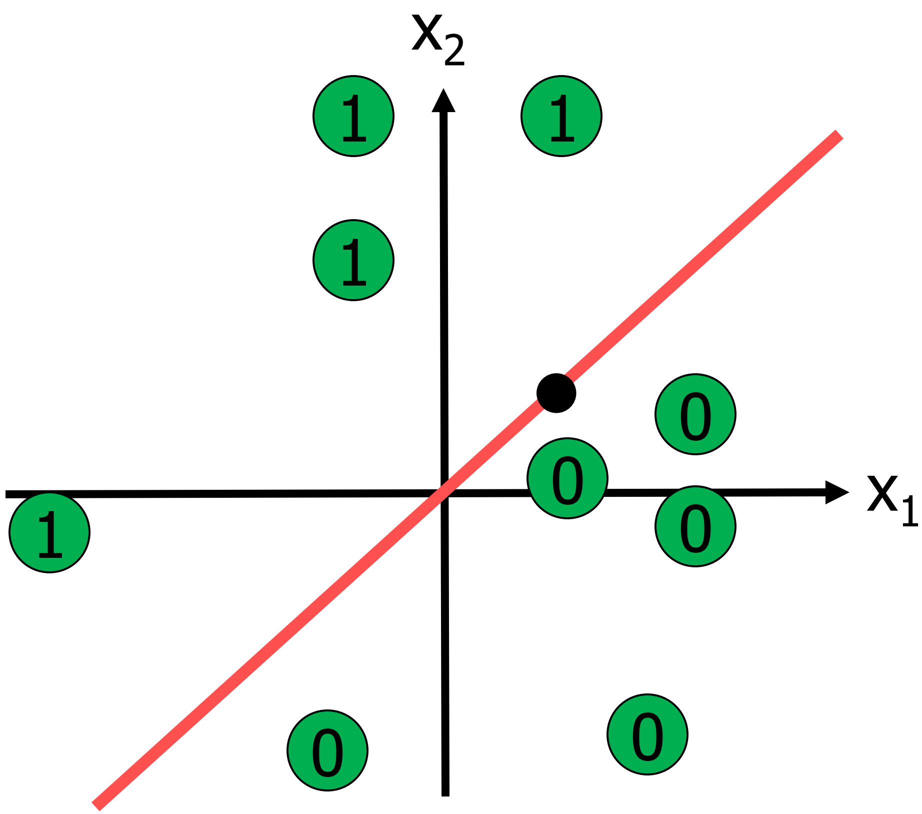
$w_1 x_1 + w_2 x_2 = \theta$

Weight vector ?

(green: correctly , red: wrongly classified)

Perceptron training

- t=5: all f(x) = t ➔ final state



Decision line

$w_1\ x_1 + w_2\ x_2 = \theta$

Weight vector ?

$W_1 = -1,\ W_2 = 1$

Point on decision line

(1,1)

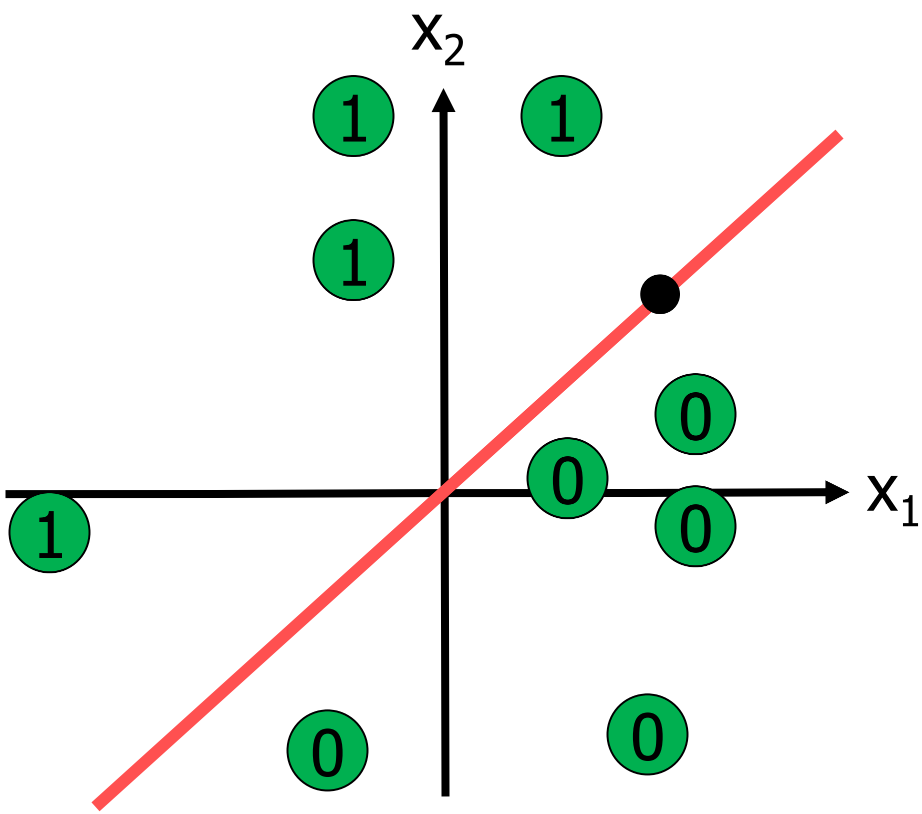
➔ (-1)*1 + 1*1 =

= -1 + 1 = 0 = θ

(green: correctly , red: wrongly classified)

Perceptron training

- t=5: all f(x) = t ➔ final state



Decision line

$w_1 x_1 + w_2 x_2 = \theta$

Weight vector ?

$w_1 = -1$, $w_2 = 1$

Point on decision line

(2,2)

➔ (-1)*2 + 1*2 =

= -2 + 2 = 0 = θ

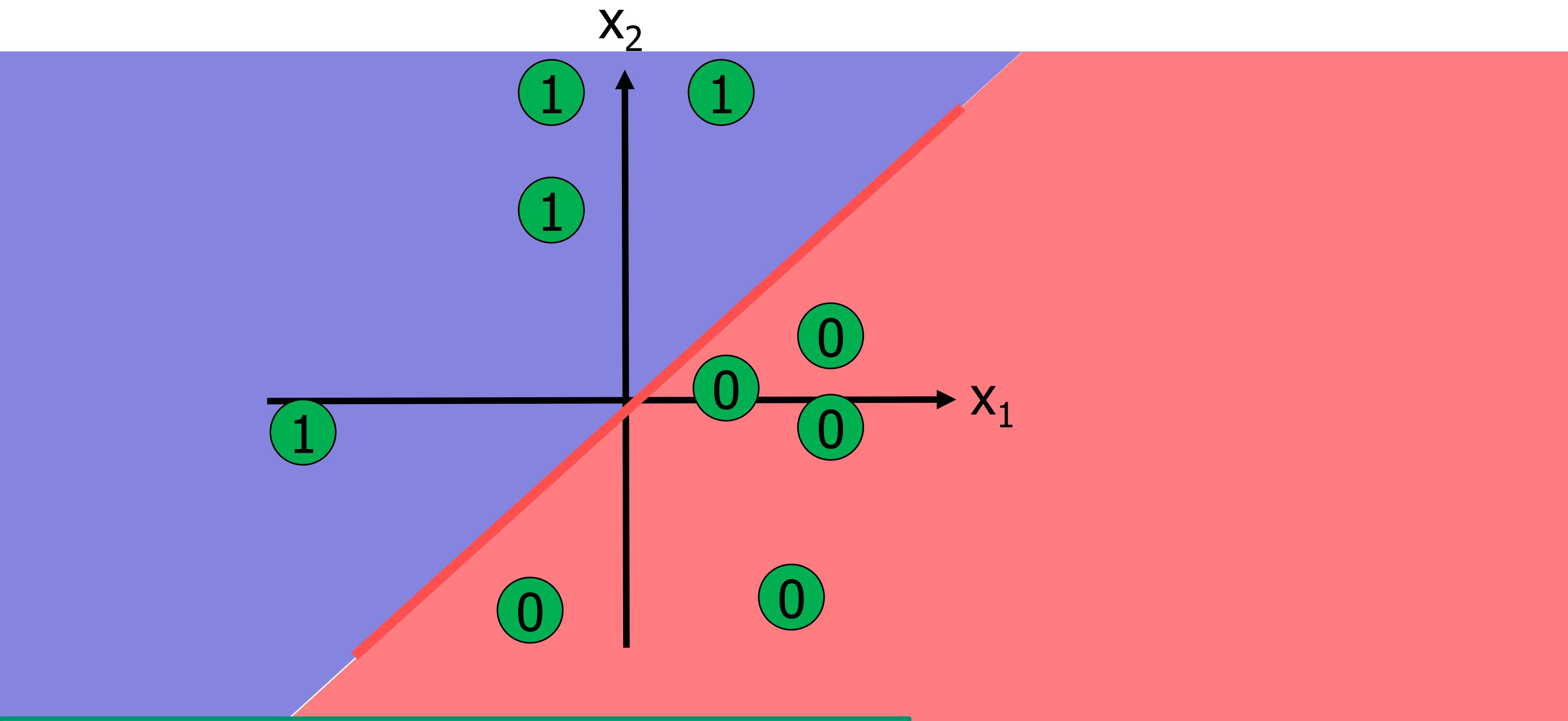
(green: correctly , red: wrongly classified)

Perceptron: properties

- Separates linearly
  - Converges to a stable state when data is *linear separable*

Perceptron: properties

- Separates linearly
  – Converges to a stable state when data is *linear separable*



Perceptron: properties

- Separates linearly
  - Converges to a stable state when data is *linear separable*

- Can predict binary decisions (true/false)
  - Can be extended for multi-class problems e.g. with different activation functions
  - *Training rule similar to other online learning algorithms, e.g. Self-Organising Maps*

- *More details next lecture*

Perceptron demos

- http://ditam.github.io/demos/perceptron/perceptronDemo.html
- https://www.cs.utexas.edu/~teammco/misc/perceptron/

Outline

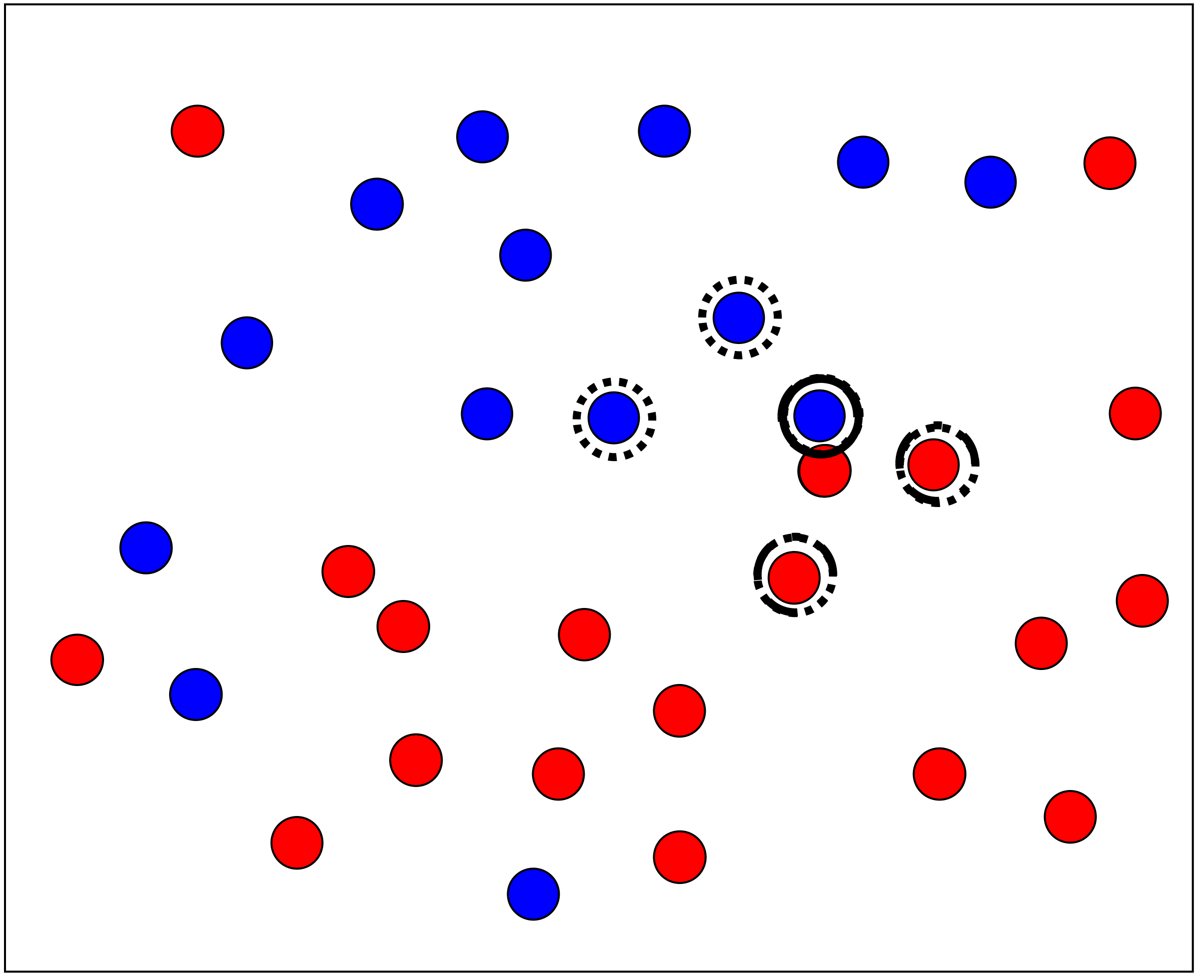
- Machine Learning: Intro
- Perceptron
- k-NN
- Performance evaluation (intro)
- Types of data & data preparation

k-nearest neighbour

- Simple algorithm, but well known
- Classify inputs based on *k* closest training examples
  - *k* to be chosen, high influence
  - Definition of “closest” – distance function

k-nn: Example

- 2-dimensional data, two classes (red & blue)



k = 1
k = 3
k = 5

k-nn – Properties

- Very sensitive to local noise
  - Little abstraction / generalisation
- Larger values of *k* reduce the effect of noise
  - makes boundaries between classes less distinct

- How to determine *k* ?
  - Good values for *k* vary (a lot!) with the data
- *k*=1 is called “nearest neighbour algorithm”

k-nn – Properties

- *What is the maximum value for k?*

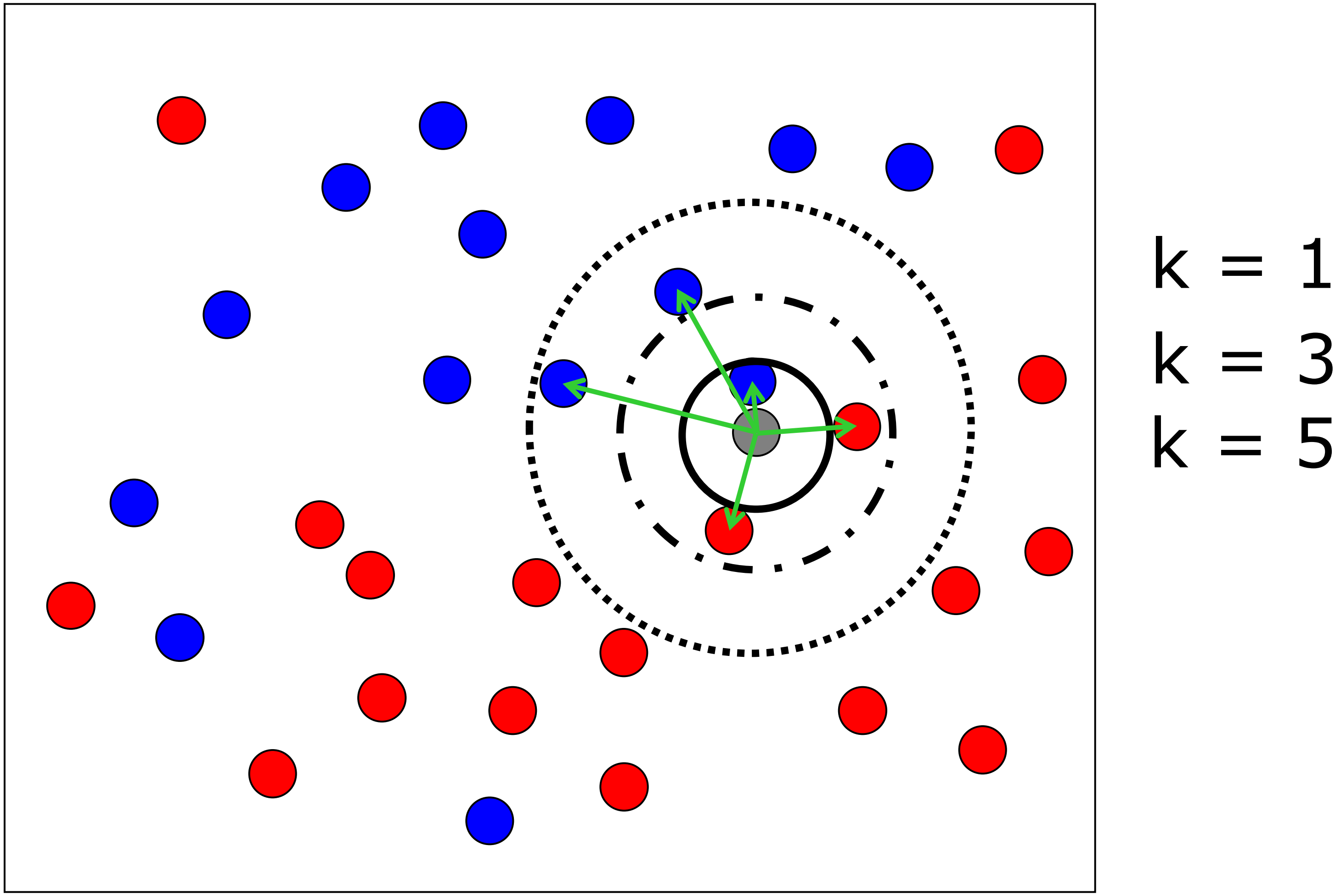
k-nn – Properties

- *What is the maximum value for k?*

- *What happens if k = n (number of samples)*

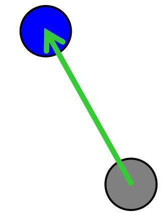
k-nn: distance function

- In this example: Euclidean distance

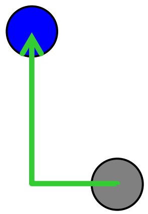


k-nn: distance function

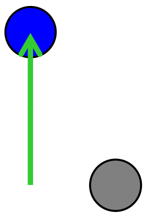
- Previous example: Euclidean distance



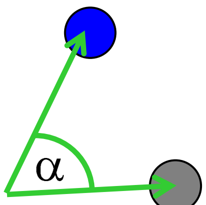
- Any other distance measure for numeric variables can be employed
  - Manhattan/City Block/L1



  - Ln
  - Linfinity



  - Cosine



k-NN vs. Perceptron

- Major differences between k-NN and perceptron algorithms?

- ***k-NN is a “Lazy learner”***
  - no model built beforehand
    ➔ computation is done at classification step
  - Opposite is called “eager learning”

k-NN vs. Perceptron

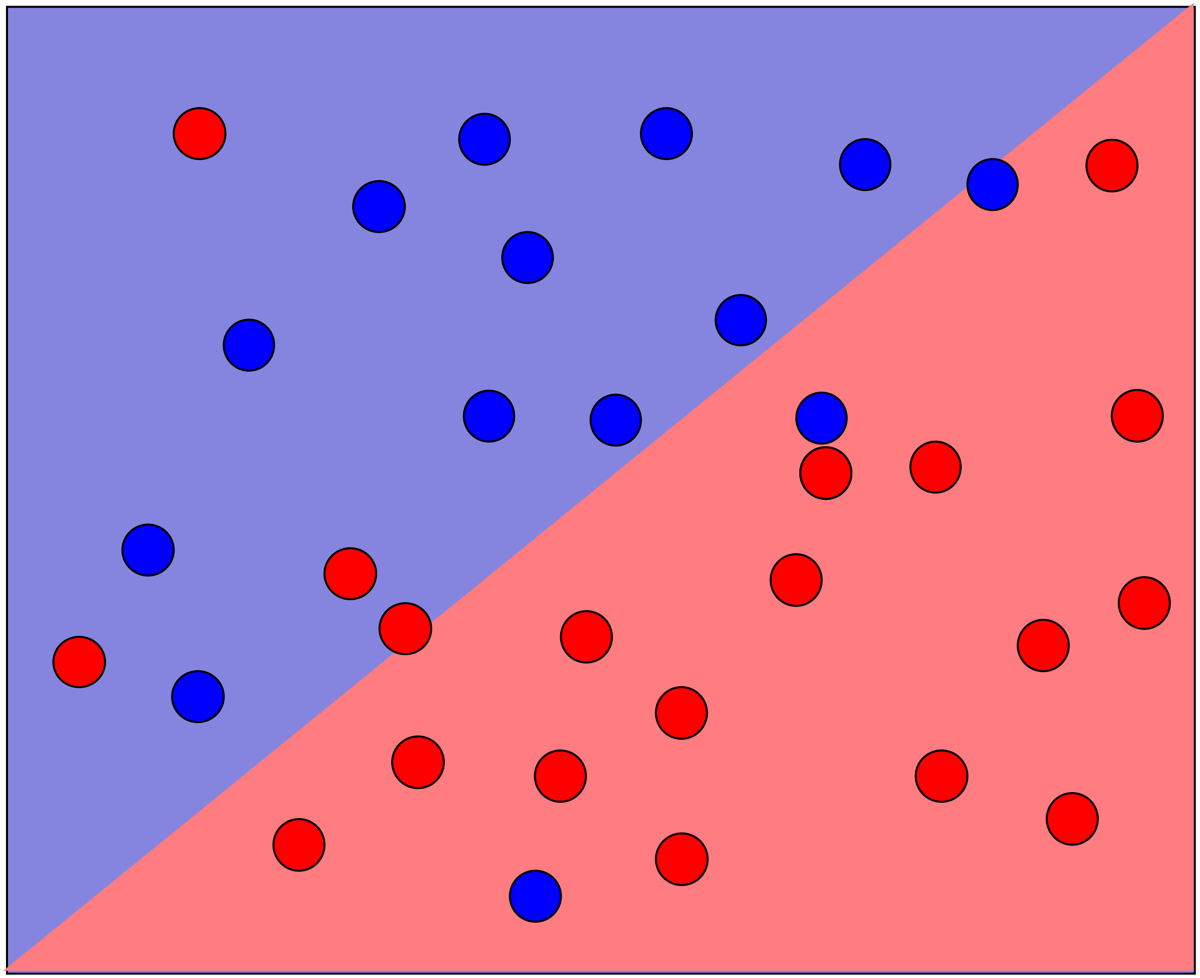
- Major differences between k-NN and perceptron algorithms?

- ***k-NN is a "Lazy learner"***
  - no model built beforehand
    ➔ computation is done at classification step
  - Opposite is called "eager learning"

- k-NN can learn complex (almost arbitrary) **decision boundaries**

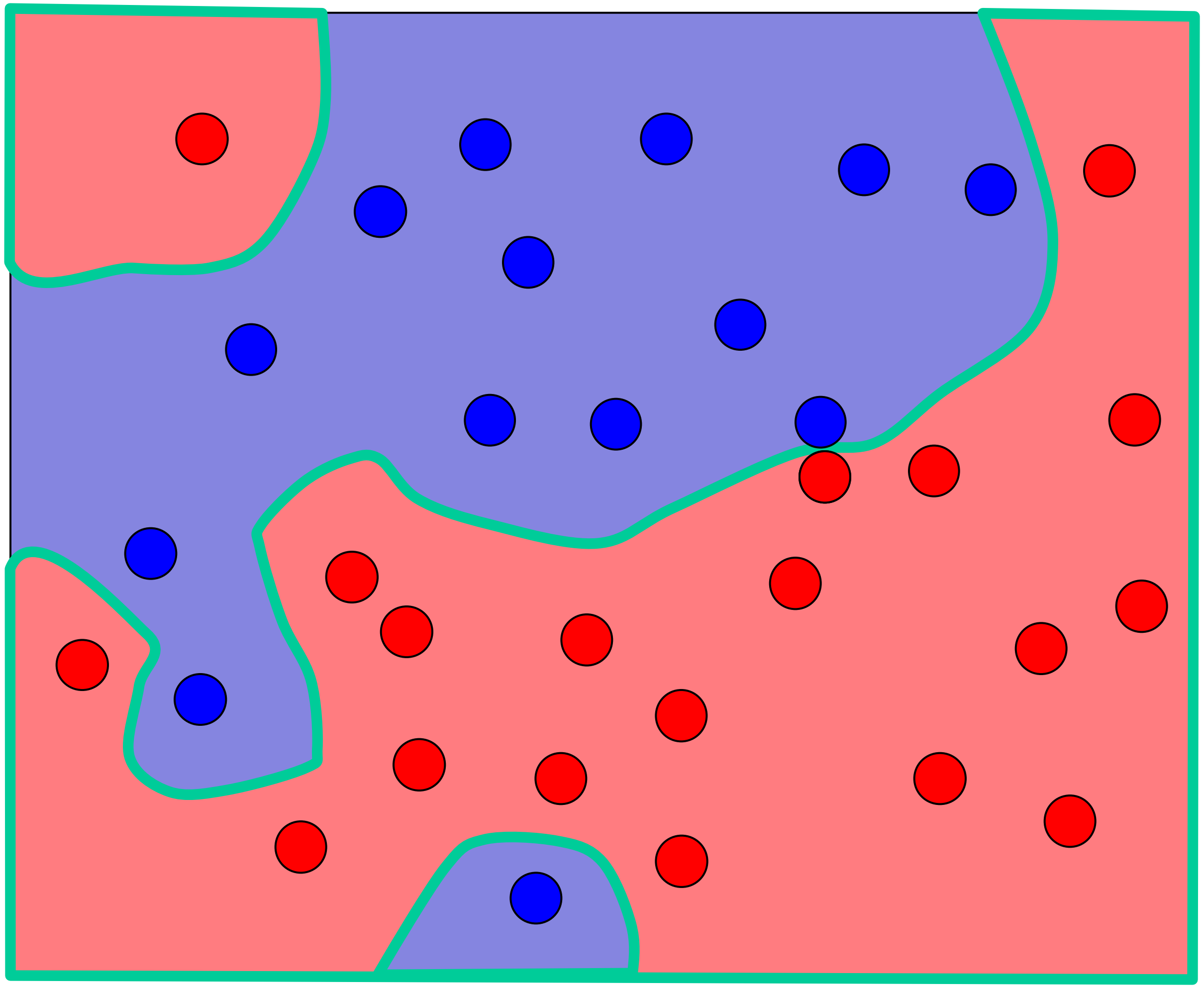
Perceptron: decision boundary

- *Only linear separation!*



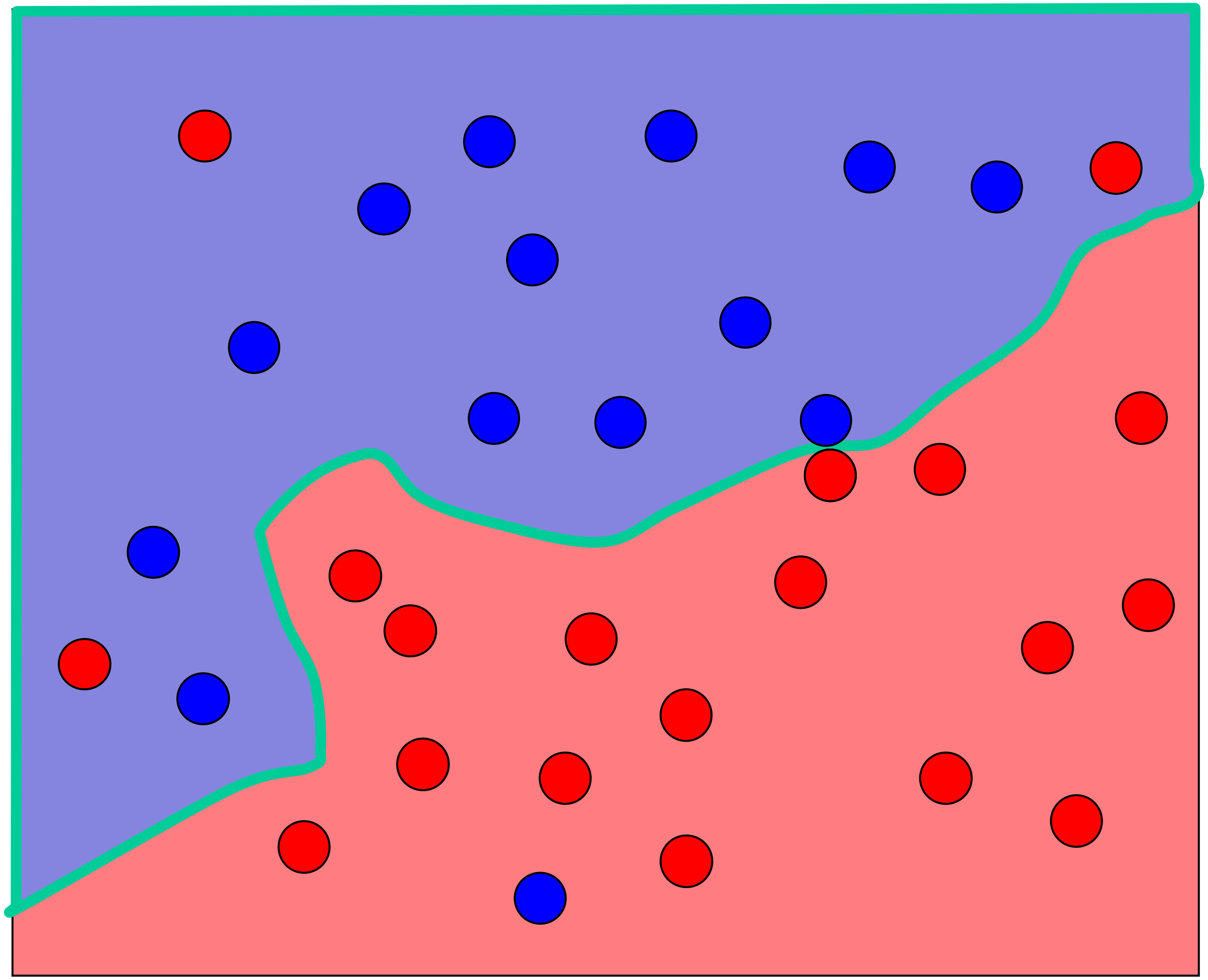
k-nn: decision boundary

- *(Almost) Arbitrary boundary*



k = 1

k-nn: decision boundary



k = 3

k-nn – Properties

- Easy to understand and interpret
- Easy to implement
- No training time needed…

- Memory requirements depend on training data
- Becomes computationally expensive with many items to classify
  - Linear search: O (Nd), N = # samples, d = dimension
  - Several optimisations proposed

- *More details next lecture*

Outline

- Machine Learning: Intro
- Perceptron
- k-NN
- Performance evaluation (intro)
- Types of data & data preparation

Evaluation

- Need to measure performance of an algorithm
  - To select the best algorithm
  - To estimate usefulness of models
  - ...

- Approach:
  - Train model on (labelled) training data
  - Test on (labelled) test data

- Measure performance
  - Several different measures …

Evaluation

- Binary classification, classes true/false
- 4 possible outcomes prediction / groundtruth

		Actual value (groundtruth)	
		true	false
Prediction (test outcome)	true	True positive (TP)	False positive (FP, Type I error)
	false	False negative (FN, Type II error)	True negative (TN)

Evaluation measures

	true	false
true	**True positive (TP)**	**False positive (FP, Type I error)**
false	**False negative (FN, Type II error)**	**True negative (TN)**

- Accuracy: # correctly predicted samples

$$\frac{TP + TN}{TP + FP + TN + FN} = \frac{TP + TN}{\#\ samples}$$

- Precision $\frac{TP}{TP + FP}$ ➔ *How to optimise?*

- Recall $\frac{TP}{TP + FN}$ ➔ *How to optimise?*

Evaluation measures

	true	false
true	True positive (TP)	False positive (FP, Type I error)
false	False negative (FN, Type II error)	True negative (TN)

- F-Measure: trade-off between precision and recall
- F1 Measure: equal weights, *harmonic mean*

$$\frac{2 * (precision * recall)}{(precision + recall)}$$

- Generally: $\frac{(1 + \beta^2) * (precision * recall)}{(\beta^2 * precision + recall)}$

Evaluation

- Where to do evaluation on?

- The samples used for training?
  - *Why not?*

  - Would not tell us how good the model works for unknown data
    - Which is however why we train a model in first place ...

  - Will often achieve better (or perfect) results on training data
    - *Why?*

Training & Test set

- Split data into training and test sets
  - E.g. ~80% training, 20% test, 66% - 33%
  - Linear, random, …

- Results can vary a lot according to how the split is done
  - ➔ Cross validation *(discussed later)*

Evaluation: outlook

- Micro vs. macro averaging
  - Confusion matrix
  - Cost functions
- ROC curves
- Cross-validation & Bootstrapping
- Significance testing

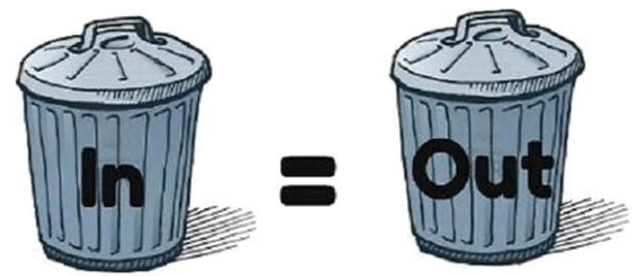
- Evaluation measures for regression

Outline

- Machine Learning: Intro
- Perceptron
- k-NN
- Performance evaluation (intro)
- Types of data & data preparation

Data Preparation

- Vital step for machine learning (supervised and unsupervised)
- ML algorithm will ***always*** give you a model
  - Quality of that model depends highly on the quality of the input data
- “Garbage in” ➔ “Garbage out”



- One major goal of data preparation:
  - Eliminate “wrong influence” of variables

Data Preparation

- Example data set from earlier (lung cancer)

gender	age	height	smoker	eye colour
male	19	170	yes	green
female	44	162	yes	grey
male	49	185	yes	blue
male	12	178	no	brown
female	37	165	no	brown
female	60	157	no	brown
male	44	19ß	no	blue
female	27	178	yes	brown
female	51	162	yes	green
female	81	168	yes	grey
male	22	184	yes	brown
male	29	176	no	blue

- Potential issues ?
  - Missing values
  - Quantitative (continuous) data with different scales
  - Categorical data

Categorical variables

- Non-numeric variables with a finite number of levels
  - *Also called **nominal data***
- E.g. eye colour with values “green", “grey”, “blue”, “brown“

- Some Machine Learning algorithms can *only handle numeric variables* !
  - *Which ones?*
- One solution: 1-to-N coding
  - *Also called **“one hot” (en)coding***

1-to-N Coding (one hot encoding)

colour
brown
blue
green
grey
brown
green
blue

green	*blue*	*brown*	*grey*
0	0	1	0
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
1	0	0	0
0	1	0	0

Categorical data: animals data set

- Animal data set
  - Describes animal by some characteristics
  - Instances: cow, duck, bee, …
- Variables
  - Size (tiny, small, medium, big)
  - Number of legs (2, 4, 6, 8)
  - Feathers (yes/no)
  - Eggs (yes/no)

	size	***legs***	***feathers***	***eggs***
duck	small	2	yes	yes
dog	medium	4	no	no
spider	tiny	8	no	yes
ladybird	tiny	6	no	yes
cow	large	4	no	no
bee	tiny	6	no	no
sparrow	small	2	yes	yes

1-to-N Coding animal data set

- Replace categorical attribute “size” with four binary attributes

	size	*legs*
duck	small	2
dog	medium	4
spider	tiny	8
ladybird	tiny	6
cow	large	4
bee	tiny	6
sparrow	small	2

tiny	*small*	*med*	*large*	*legs*
0	1	0	0	2
0	0	1	0	0
1	0	0	0	8
1	0	0	0	6
0	0	0	1	4
1	0	0	0	6
0	1	0	0	2

- *N.b.: could also encode to numerical value*
  - *E.g.: tiny=1, small=2, med=3, large=4 (if sizes are equally distant)*

Categorical data: another example

- *Any other pre-processing needed?*

- Variable "legs"
  - If considered categorical: defined order ➔ ordinal data
  - Can compute similarity: 2 closer to 4 than to 6

	size	***legs***
duck	small	2
dog	medium	4
spider	tiny	8
ladybird	tiny	6
cow	large	4
bee	tiny	6
sparrow	small	2

  - Numerical value, can compute distance directly

  - *Does the number of legs denote similarity?*

1-N coding animal data set

- *Does number of legs denote similarity?*
  - – i.e., is an animal with 2 legs more similar to one with 4, or with 6?
  - – And, is one with 4 equally similar to the one with 2 and 6?
    - Dog to monkey vs. dog to spider
  - – One with 6 equally similar to one with 4 and 8?
    - Bee to spider vs. bee to cow

1-to-N Coding animal data set

	size	*legs*
duck	small	2
dog	medium	4
spider	tiny	8
ladybird	tiny	6
cow	large	4
bee	tiny	6
sparrow	small	2

2 legs	*4 legs*	*6 legs*	*8 legs*
1	0	0	0
0	1	0	0
0	0	0	1
0	0	1	0
0	1	0	0
0	0	1	0
1	0	0	0

k-nn: distance function

- *What to do with categorical data?*

a) 1-n coding – then apply any distance function mentioned earlier

b) Definition of custom distance functions that applies to categorical data

k-nn: distance function

- Definition of custom distance functions
  - E.g. adapt hamming distance

    Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different

    ➔ count number of different nominal values

	size	*legs*	*feathers*	*eggs*	*distance*
horse	large	4	no	no	
duck	**small**	**2**	**yes**	**yes**	4
dog	**medium**	4	no	no	1
spider	**tiny**	**8**	no	**yes**	3
ladybird	**tiny**	**6**	no	**yes**	3
cow	large	4	no	no	
bee	**tiny**	**6**	no	no	2
sparrow	**small**	**2**	**yes**	**yes**	4

k-nn: distance function

- Definition of custom distance functions
  - E.g. adapt hamming distance

    Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different

    ➔ count number of different nominal values

  - Define custom distance for each attribute, aggregate e.g. via sum

Scaling / Normalisation / Standardisation

- Different variables may exhibit significantly different value ranges
  - E.g. a length variable measured in cm, inch, or meters
  - Different types of measurements: length, speed, temperature, …
  - Different types of measuring devices capturing different value ranges
  - …
  - *Why is this a (potential) problem?*

Scaling / Normalisation / Standardisation

- Some ML algorithms rely on *measuring the (numeric) distance* between samples

- ➔ There should be no impact by the value range

  – higher values in one attribute / variable would have unproportional effect on measure distance
  ➔ would dominate distance metric

  ➔ might thus dominate learning

Scaling / Normalisation / Standardisation

- Remove effects of different value ranges in each attribute/variable

- Common method in statistics and machine learning
  - With many different notations
  - Scaling, Normalisation, Standardisation, ...
  - Often used interchangely, different for each field ...

Scaling / Normalisation / Standardisation

- Z-score standardisation / normalization

- Each variable *x* is converted to have a mean of 0 and a standard deviation of 1
  - subtracting the mean
  - dividing by standard deviation

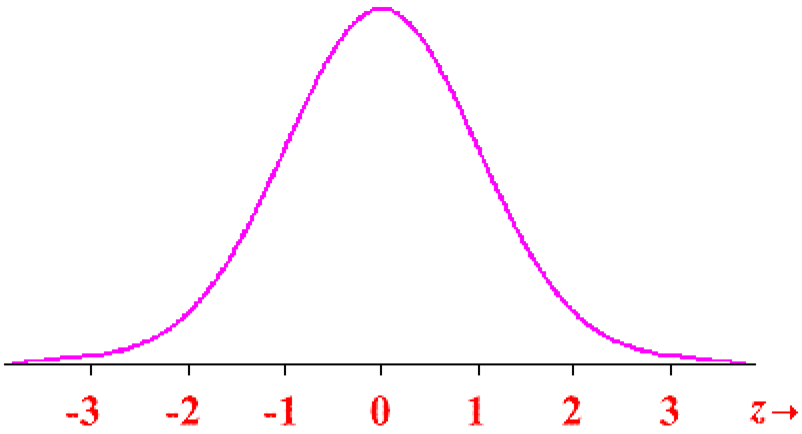
$$z_i = \frac{x_i - \mu}{\sigma}$$

Scaling / Normalisation / Standardisation

- Z-score standardisation / normalization

$$z_i = \frac{x_i - \mu}{\sigma}$$

- *What's the new value range after z-score?*



Scaling / Normalisation / Standardisation

- Min-Max scaling
  - Scale all variables to the same (fixed) range
  - Often between 0 and 1
  - Subtract minimum value for each variable
  - Divide by value range of each variable

$$z_i = \frac{x_i - \min(X)}{\max(X) - \min(X)}$$

  - *Multiply by new range (if different than 0..1)*

Scaling

- *Is scaling an issue for*
  - *k-nn?*
  - *Perceptron?*

Scaling / Normalisation / Standardisation

- Especially algorithms relying on distances
  - k-Nearest Neighbours
  - …

- Scaling not needed for algorithms that don't use distances, e.g.
  - Naïve Bayes
  - Decision trees
  - …

Scaling / Normalisation / Standardisation

- Especially algorithms relying on distances
  - k-Nearest Neighbours
  - ...

- *Caveat*
  - Many implementations already do this pre-processing implicitly (e.g. WEKA)
  - Check default settings carefully

Missing values

- For some samples, not all attribute values are known
- Some ML algorithms can handle missing values
- Solutions for other algorithms
  - Deletion of sample
    - bad when only few labelled samples
  - Imputation

Data Imputation

- Substitution of a missing value

- Different methods
  - Mean value of the attribute (computed from other samples)
  - Random selection of value from another sample
  - Regression – using other attributes to predict
  - Clustering – values of cluster centroid
  - Nearest Neighbour – value of closest sample

- *Use in first exercise !*

Data Imputation

- *When is imputation useful?*

- Most useful when the number of labelled samples (w/o missing values) is small
  - *Relatively easy to identify*

- When samples with missing values contain important information
  - *Difficult to identify*

Questions?

Upcoming topics:

- Decision trees
- Random Forests
- Support Vector Machines