Summary

SUMMARY

(SKIPPED "COURSE INTRODUCTION" AND "INTRODUCTION TO DATA SCIENCE")

• DATA PREPROCESSING:

• DATA TYPES AND PROPERTIES:

• ATTRIBUTE TYPES:

PROPERTIES:

- NOMINAL: eg. ID NUMBERS, ZIP CODES —> DISTINCTNESS ~> =/≠
- ORDINAL: eg. RANKINGS —> DISTINCTNESS & ORDER ~> >/<
- INTERVAL: eg. CALENDAR DATES —> DISTINCTNESS, ORDER, ADDITION ~> +/-
- RATIO: eg. TEMPERATURE, LENGTH, —> DISTINCTNESS, ORDER, ADDITION, MULTIPLICATION ~> */÷

- DISCRETE VS CONTINOUS ATTRIBUTES

• DATA SET TYPES:

- RECORDS:
  - TABLES
  - DOCUMENT DATA
  - TRANSACTION DATA
- GRAPH: (STRUCTURED / UNSTRUCTURED TEXT)
  - WORLD WIDE WEB
  - MOLECULAR STRUCTURES
- ORDERED
  - SPATIAL DATA
  - MOLECULAR DATA
  - SEQUENTIAL DATA ~> eg. TRANSACTIONS

• DATA QUALITY:

- NOISE: MODIFICATION OF ORIGINAL DATA ~> eg. RECORDING OF A PERSON VOICE
- OUTLIERS: DATA OBJECT WITH CONSIDERABLY DIFFERENT CHARACTERISTICS IN DATA SET
  - NOISE THAT INTERFERES WITH THE ANALYSIS
  - GOAL OF ANALYSIS ~> eg. INTRUSION DETECTION
- MISSING VALUES:
- DUPLICATE DATA

DATA PREPARATION:

DATA PREPROCESSING:

- AGGREGATION: COMBINE 1 OR MORE ATTRIBUTES INTO A SINGLE ONE
  → DATA REDUCTION, CHANGE OF SCALE, LESS VARIABILITY
- DATA REDUCTION: REDUCED REPRESENTATION OF DATASET
  → SAMPLING, FEATURE SELECTION, DISCRETIZATION
  - SAMPLING:
    - SIMPLE RANDOM SAMPLING: ∀ ITEM → SAME p (PROBABILITY)
      - SAMPLING WITHOUT REPLACEMENT: ∀ CHOSEN ITEM → REMOVED FROM POPULATION
      - SAMPLING WITH REPLACEMENT: ∀ ITEM → CAN BE CHOSEN MORE THAN 1 TIME
    - STRATIFIED SAMPLING: SPLIT DATA IN PARTITIONS, THEN RANDOM SAMPLING
  - DIMENSIONALITY: ↑ DIMENSIONALITY → ↑ SPARSITY
- DIMENSIONALITY REDUCTION:
  - TECHNIQUES: PCA (PRINCIPAL COMPONENT ANALYSIS), SINGULAR VALUE DECOMPOSITION, OTHERS
    → FIND A PROJECTION THAT CAPTURES LARGEST AMOUNT OF VARIATION IN DATA
  - FEATURE SUBSET SELECTION
  - FEATURE CREATION: CREATE NEW ATTRIBUTES THAT CAN CAPTURE MORE IMPORTANT INFORMATIONS
    → FEATURE EXTRACTION, MAPPING TO NEW SPACE (EG. FOURIER TRANSFORM: $t \mapsto f$), FEATURE CONSTRUCTION
  - DISCRETIZATION:
    - UNSUPERVISED DISCRETIZATION: FIND BREAKS IN DATA VALUES
    - SUPERVISED DISCRETIZATION: USE CLASS LABELS TO FIND BREAKS
  - BINARIZATION: MAPPING ATTRIBUTE ↦ 1 OR MORE BINARY VARIABLES

- ATTRIBUTE TRANSFORMATION:
  - NORMALIZATION: VALUES ARE SCALED IN ORDER TO FALL IN A RANGE
    - MIN-MAX NORMALIZATION: $v' = \frac{v - min_A}{max_A - min_A}\left(max_{N,A} - min_{N,A}\right) + min_{N,A}$ (NUOVO)
    - Z-SCORE NORMALIZATION: $v' = \frac{v - \mu_A}{\sigma_A}$ (μ_A ~> MEAN VALUE, σ_A ~> STAND. DEV.)

STANDARD $Z \sim N(0,1)$

  - DECIMAL SCALING: $v' = \frac{1}{10^j} \cdot v$ / j : SMALLEST INT. SUCH THAT $max(|v'|) < 1$

- DATA PREPARATION FOR DOCUMENT DATA:
  - DOCUMENT REPRESENTED AS A SET OF FEATURES: SET OF WORDS, CHARACTERS, TERM, CONCEPTS
  - DOCUMENT PROCESSING:

    IT GENERATES A STRUCTURED DATA REPRESENTATION OF DOCUMENT DATA

    -> 5 STEPS: DOCUMENT SPLITTING, TOKENISATION, CASE NORMALIZATION, STOPWORD REMOVAL, STEMMING

    - DOCUMENT SPLITTING: DOCUMENT CAN BE SPLITTED IN PARAGRAPH, SEGMENTS, WORDS, SENTENCES, ECC.
    - TOKENISATION: BREAKING TEXT IN SENTENCES (TOKENS)
    - CASE NORMALIZATION: WORDS ⟼ UPPER/LOWER CASE CHAR
    - STEMMING: REDUCE A WORD INTO ITS ROOT FORM
      -> IMPLICATION OF PREFIXES, SUFFIXES, PLURALIZATION, ECC.
    - STOPWORD ELIMINATION: ELIMINATION OF ARTICLES, PREPOSITIONS, ECC.

  - WEIGHTED DOCUMENT REPRESENTATION:

    SD. N OF WORDS IN DOCUMENT

	learn	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

    - FEATURE VECTORS: DOCUMENT TEXT ⟼ FEATURES
    - WEIGHTED SCHEMES:
      - BINARY: ∀ WORD -> ∃ / ∄
      - FREQUENCY OCCURRENCE
    - TERM FREQUENCY INVERSE DOCUMENT FREQUENCY: $tf\text{-}idf(t) = f(t,d)\log\left(\frac{m}{f(t,D)}\right)$ (TERM t IN DOCUMENT d OF COLLECTION D; FREQUENCY; DOCUMENT; COLLECTION OF DOCUMENTS)
    - DOCUMENT-TERM MATRIX X: $X_{ij} = l_{ij} * g_j$ (LOCAL WEIGHT; GLOBAL WEIGHT)

• SIMILARITY AND DISSIMILARITY :

- SIMILARITY :

  MEASURE OF HOW ALIKE 2 DATA OBJECT ARE

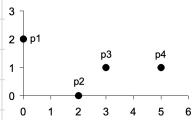
- DISSIMILARITY :

  MEASURE OF HOW DIFFERENT 2 DATA OBJECT ARE

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases}$	$s = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$
Ordinal	$d = \lvert x - y \rvert / (n-1)$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - d$
Interval or Ratio	$d = \lvert x - y \rvert$	$s = -d$, $s = \frac{1}{1+d}$, $s = e^{-d}$, $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

• EUCLIDEAN DISTANCE :

eg.



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Distance Matrix

$$d(x, y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$

→ n : NO OF DIMENSIONS , x_k, y_k ARE k^{th} ATTRIBUTES OF DATA OBJECTS x & y

• MINKOWSKI DISTANCE :

eg.

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

L_∞	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

Distance Matrix

IT'S A GENERALIZATION OF EUCLIDEAN DISTANCE → r = 2

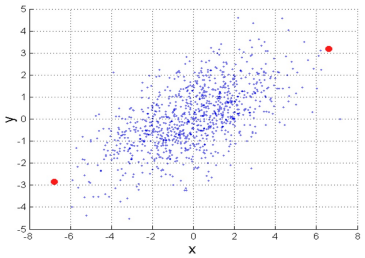
$$d(x, y) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{1/r}$$

→ r : PARAMETER, n : NO OF DIMENSIONS x_k, y_k ARE k^{th} ATTRIBUTES OF DATA OBJECTS x & y

- r = 1 : L_1 NORM, CITY BLOCK DISTANCE → eg. HAMMING DISTANCE / MANHATTAN
- r = 2 : L_2 NORM, EUCLIDEAN DISTANCE
- r = ∞ : L_∞ NORM → MAXIMUM DISTANCE BETWEEN ANY COMPONENT OF THE VECTORS

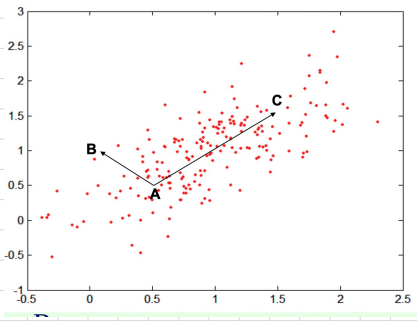
• MAHALANOBIS DISTANCE :

$$\text{mahalanobis}(x, y) = (x - y)^T \Sigma^{-1} (x - y)$$



Σ is the covariance matrix

For red points, the Euclidean distance is 14.7, Mahalanobis distance is 6.



Covariance Matrix:

$$\Sigma = \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{bmatrix}$$

A: (0.5, 0.5)

B: (0, 1)

C: (1.5, 1.5)

Mahal(A,B) = 5

Mahal(A,C) = 4

- # SIMILARITY BETWEEN BINARY VECTORS:

  - CONSIDER BINARY VECTORS p, q
  - COMPUTE SIMILARITY: $M_{00}, M_{01}, M_{10}, M_{11} \;/\; M_{xy} = \begin{cases} n^\circ \text{ OF ATTRIBUTES} \\ \text{WHERE } p = x,\ q = y \end{cases}$
  - SIMPLE MATCHING AND JACCARD COEFFICIENTS:
    - SIMPLE MATCHING: $SMC = \dfrac{n^\circ \text{ OF MATCHES}}{n^\circ \text{ OF ATTRIBUTES}} = \dfrac{M_{00} + M_{11}}{(M_{00} + M_{01} + M_{10} + M_{11})}$
    - $J = \dfrac{n^\circ \text{ OF 11 MATCHES}}{n^\circ \text{ OF NON-BOTH-ZERO ATTRIBUTES VALUE}} = \dfrac{M_{11}}{M_{01} + M_{10} + M_{11}} = SMC \Big|_{M_{00} = 0}$
  - # COSINE SIMILARITY:
    - LET d_1, d_2 BE 2 DOCUMENT VECTORS;
    - $\to \cos(d_1, d_2) = \dfrac{d_1 \cdot d_2}{\|d_1\| \cdot \|d_2\|}$ (DOT PRODUCT; $\|d_1\| = \sqrt{d_1^2}$)
  - IN GENERAL:

$$\text{SIMILARITY}(x, y) = \frac{\sum_{k=1}^{n} w_k \delta_k s_k(x, y)}{\sum_{k=1}^{n} w_k \delta_k}$$

w_k : WEIGHT

- $s_k(x, y)$: SIMILARITY FOR k^{TH} ATTRIBUTE
- $\delta_k = \begin{cases} 0, & \text{IF THE } k^{TH} \text{ IS ASYMMETRIC AND BOTH OBJECT HAVE VALUE } 0 \\ & \text{OR IF ONE OF THE OBJECTS HAS A MISSING VALUE FOR } k^{TH} \text{ ATTRIBUTE} \\ 1, & \text{OTHERWISE} \end{cases}$

ex.

$d_1 = 3205000200$
$d_2 = 1000000102$

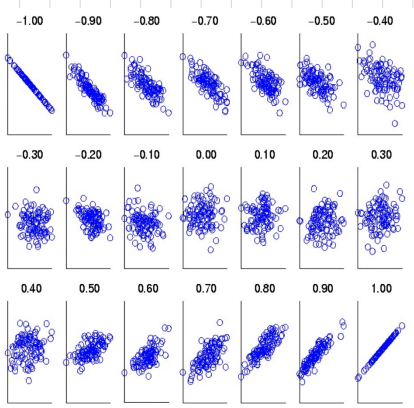
$d_1 \cdot d_2 = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5$
$\|d_1\| = (3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)^{0.5} = (42)^{0.5} = 6.481$
$\|d_2\| = (1*1+0*0+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2)^{0.5} = (6)^{0.5} = 2.245$

$\cos(d_1, d_2) = .3150$

- # CORRELATION:

$$\text{CORR}(x, y) = \frac{\sigma_{xy}}{\sigma_x \cdot \sigma_y}$$

- $\sigma_{xy} = \dfrac{1}{n-1} \sum_{k=1}^{n} (x_k - \bar{x})(y_k - \bar{y})$
- σ_x (ANALOG FOR σ_y) $= \sqrt{\dfrac{1}{n-1} \sum_{k=1}^{n} (x_k - \bar{x})^2}$



ASSOCIATIONS RULES:

- GOAL: EXTRACTION OF FREQUENT CORRELATION OR PATTERNS FROM A TRANSACTIONAL DB
- ASSOCIATION RULE MINING:
  - COLECTION OF TRANSACTION WHERE
  - ASSOCIATION RULE: $A, B \rightarrow C$
    - NOT CASUALITY

Tickets at a supermarket counter

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diapers, Milk
4	Beer, Bread, Diapers, Milk
5	Coke, Diapers, Milk
...	...

- Association rule
  diapers ⇒ beer
  - 2% of transactions contains both items
  - 30% of transactions containing diapers also contains beer

- IT IS AN EXPLORATORY TECHNIQUE THAT CAN BE APPLIED TO ANY DATA TYPE
- DEFINITIONS:

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diapers, Milk
4	Beer, Bread, Diapers, Milk
5	Coke, Diapers, Milk
...	...

  - ITEMSET: SET INCLUDING 1 OR MORE ITEMSET
    ex. {BEER, DIAPERS}
  - K-ITEMSET: ITEMSET OF LENGTH K, K ITEMS CONTAINED
  - SUPPORT COUNT (#): ABSOLUTE # OF OCCURRENCE OF AN ITEMSET
    ex. $\#\{\text{BEER, DIAPERS}\} = 2$
  - SUPPORT: RELATIVE # OF TRANSACTIONS THAT CONTAINS AN ITEM
    ex. $\text{sup}(\{\text{BEER, DIAPERS}\}) = \frac{2}{5}$
  - FREQUENT ITEMSET: ITEM WHOSE $\text{sup}() > \text{min sup}$
  - GIVEN THE ASSOCIATION RULE $A \rightarrow B$:
    - SUPPORT $= \frac{\#\{A, B\}}{|T|}$ ⇝ CARDINALITY OF DB
    - CONFIDENCE $= \frac{\text{sup}(\{A, B\})}{\text{sup}(A)}$ ⇝ IT'S LIKE A CONDITIONAL PROBABILITY
      $A \rightarrow B$

ex.

- From itemset {Milk, Diapers} the following rules may be derived
- Rule: Milk ⇒ Diapers
  - support
    sup=#{Milk,Diapers}/#trans. =3/5=60%
  - confidence
    conf=#{Milk,Diapers}/#{Milk}=3/4=75%
- Rule: Diapers ⇒ Milk
  - same support
    s=60%
  - confidence
    conf=#{Milk,Diapers}/#{Diapers}=3/3 =100%

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diapers, Milk
4	Beer, Bread, Diapers, Milk
5	Coke, Diapers, Milk

→ ASSOCIATION RULES MINING IS THE EXTRACTION OF RULES $\begin{cases} \text{SUPPORT} \geq \text{min sup} \\ \text{CONFIDENCE} \geq \text{min conf} \end{cases}$

- BRUTE FORCE APPROACH:
  - ENUMERATE ALL POSSIBLE PERMUTATION
  - COMPUTE SUPPORT AND CONFIDENCE
  - PRUNE BASED ON minsup & minconf

  } -> GIVEN d ITEMS -> 2^d POSSIBLE ITEMSETS
  COMPUTATION VERY EXPENSIVE

2 - APRIORI ALGORITHM:

- APRIORI PRINCIPLE: "IF AN ITEMSET IS FREQUENT -> ALL OF HIS SUBSETS MUST BE FREQUENT"
  -> IF $A \subseteq B$ -> $sup(A) \geq sup(B)$
  -> IF ITEMSET S IS NOT FREQUENT -> ALL SUBSETS GENERATED FROM S CAN BE DISCARDED

- AGR 94 ALGORITHM:

PSEUDOCODE

C_k: Candidate itemset of size k
L_k : frequent itemset of size k

L_1 = {frequent items};
for (k = 1; L_k !=∅; k++) **do**
  begin
    C_{k+1} = candidates generated from L_k;
    for each transaction t in database do
      increment the count of all candidates in C_{k+1} that are contained in t
    L_{k+1} = candidates in C_{k+1} satisfying *minsup*
  end
return $\cup_k L_k$;

- ∀ ITERATION -> EXTRACT ITEMSETS OF LENGTH K
- 2 STEPS:

1. CANDIDATE GENERATION: (1ST ITER: K=1)
   1.1 GENERATE CANDIDATES OF LENGTH K+1 FROM FREQUENT ITEMSETS OF LENGTH K
   1.2 PRUNE LENGTH K+1 ITEMSETS / CONTAIN AT LEAST 1 K-ITEMSET THAT ARE NOT FREQUENT
   (APRIORI PRINCIPLE APPLIED)
2. FREQUENT ITEM GENERATION
   2.1 SCAN DB TO CON SUPPORT FOR K+1 CANDIDATES
   2.2 PRUNE CANDIDATES / $sup() \leq minsup$

ED. SLIDES 21 ÷ 35

- FACTORS AFFECTING PERFORMANCE:
  LOW MINIMUM SUPPORT THRESHOLD, DATA SET DIMENSIONS, SIZE OF DB, AVG TRANSACTION WIDTH
- IMPROVING APRIORI EFFICIENCY:
  - PARK 95: HASH BASED ITEMSET COUNTING, TRANSACTION REDUCTION
  - SAV 96: PARTITIONING
  - TOI 96: SAMPLING
  - MOTW 98: DYNAMIC ITEMSET COUNTING

FP-GROWTH ALGORITHM :

EXPLOIT A MAIN MEMORY COMPRESSED REPRESENTATION OF DB : THE FP-TREE

- FREQUENT PATTERN MINING BY FP-GROWTH
- ONLY 2 DB SCANS : COUNT ITEM SUPPORTS + BUILD FP-TREE

- **HAN 00 ALGORITHM :**

  - **FP-TREE CONSTRUCTION :** ~> SIMILAR TO HAMMING CODES TREE

    HEADER :
    {B} -> 8
    {A} -> 7
    ⋮

    1. COUNT ITEM SUPPORT AND PRUNE ITEMS WHOSE sup() < minsup
    2. BUILD HEADER TABLE, SORTING ITEMS BY DECREASING sup() ORDER
    3. CREATE FP-TREE :

       ∀ TRANSACTION t IN DB :

       3.1 ORDER TRANSACTION t ITEMS IN DECREASING sup() ORDER

       3.2 INSERT t IN FP-TREE : USE COMMON PATH FOR COMMON PREFIX, CREATE NEW BRANCH WHEN PATH BECOMES DIFFERENT

  - **FP-GROWTH ALGORITHM :**

    1. SCAN HEADER TABLE FROM LOWEST SUPPORT ITEM (IN HEADER TABLE)

       ∀ ITEM i IN HEADER TABLE, EXTRACT FREQUENT ITEMSETS INCLUDING i AND ITEMS PRECEDING IT

       2.1 BUILD CONDITIONAL PATTERN BASE FOR ITEM i -> i-CPB

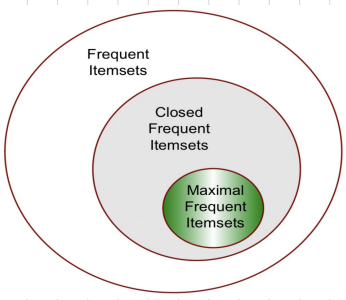
       - SELECT PREFIX-PATHS OF ITEM i FROM FP-TREE

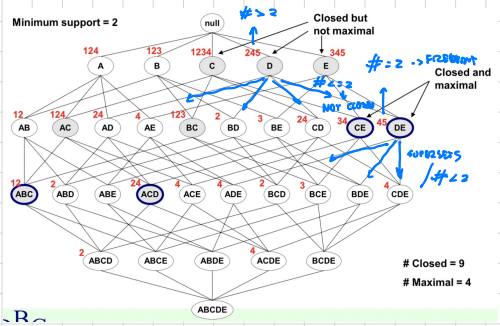
       2.2 RECURSIVE INVOCATION OF FP-GROWTH ON i-CPB

RD. SLIDES 46 ÷ 53 AND 56 ÷ 73

MAXIMAL FREQUENT ITEMSET : AN ITEM IS FREQUENT MAXIMAL IF NONE OF ITS SUPERSETS IS FREQUENT

CLOSED ITEMSET : ITEMSET IS CLOSED IF NONE OF ITS IMMEDIATE SUPERSETS HAS THE SAME SUPPORT AS THE ITEMSET





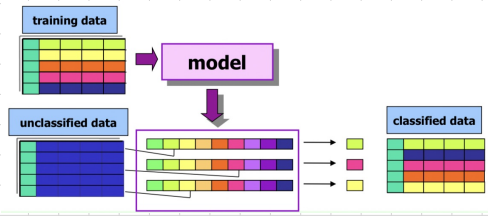
- EFFECT OF SUPPORT THRESHOLD :
  - TOO HIGH minsup : POSSIBLE LOSS FOR RARE BUT INTERESTING ITEMSETS
  - TOO LOW minsup : POSSIBILITY TO BECOME COMPUTATIONALLY EXPENSIVE, LARGE AMOUNT OF FREQUENT ITEMSETS
- CONFIDENCE MEASURE NOT ALWAYS RELIABLE -> CORRELATION IS BETTER

$$\text{CORRELATION} = \frac{P(A,B)}{P(A)\,P(B)} = \frac{\text{CONF}(A \to B)}{\text{sup}(B)} \to \begin{cases} \text{CORR.} = 1 \to \text{STAT. INDIP.} \\ \text{CORR.} > 1 \to \text{POSITIVE CORR.} \\ \text{CORR.} < 1 \to \text{NEGATIVE CORR.} \end{cases}$$

- CONSIDERING ITEMSETS WEIGHT
- CONSIDERING HIERARCHIES -> AGGREGATE DATA : TAXONOMY
  -> GENERALIZING DATA CAN BE MORE USEFUL / MEANINGFUL SOMETIMES
  L> OR COMBINE DETAILED AND GENERALIZED DATA

- **CLASSIFICATION FUNDAMENTALS:**

GOAL: PREDICTION OF A CLASS LABEL OF UNTRAINED DATA, GIVEN TRAINING DATA



- **DEFINITIONS:**
  - TRAINING SET: COLLECTION OF LABELED DATA OBJECTS, USED TO LEARN THE CLASSIFICATION MODEL
  - TEST SET: COLLECTION OF LABELED DATA OBJECTS TO VALIDATE THE CLASSIFICATION MODEL

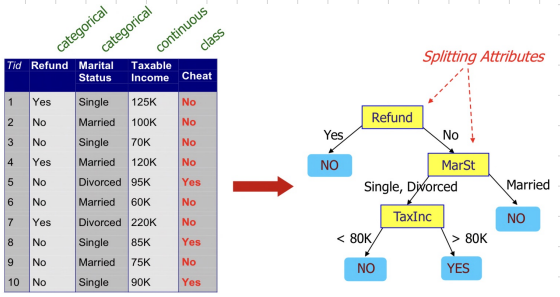
CLASSIFICATION TECHNIQUES:

- Decision trees
- Classification rules
- Association rules
- Neural Networks
- Naïve Bayes and Bayesian Networks
- k-Nearest Neighbours (k-NN)
- Support Vector Machines (SVM)
- ...

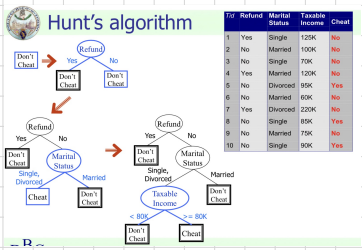
EVALUATION OF CLASSIFICATION TECHNIQUES:

- Accuracy
  - quality of the prediction
- Interpretability
  - model interpretability
  - model compactness
- Incrementality
  - model update in presence of newly labelled record
- Efficiency
  - model building time
  - classification time
- Scalability
  - training set size
  - attribute number
- Robustness
  - noise, missing data

- **DECISION TREES:**
  - MANY ALGORITHMS: HUNT'S ALGORITHM, CART, ID3, C4.5, C5.0 SLIQ, SPRINT



- **HUNT'S ALGORITHM (IN BRIEF):**
  - GIVEN D_t: SET OF TRAINING RECORDS THAT REACH NODE t
  - STEPS:
    - IF D_t CONTAINS RECORDS THAT BELONG TO MORE THAN 1 CLASS:
      - SELECT BEST ATTRIBUTE A ON WHICH TO SPLIT D_t AND LABEL NODE t AS A
      - SPLIT D_t IN SMALLER SUBSETS AND RECURSIVELY APPLY THIS PROCEDURE FOR EACH SUBSET
    - IF D_t CONTAINS RECORDS THAT BELONG TO SAME CLASS Y_t -> t IS A LEAF, LABELED Y_t
    - IF D_t IS AN EMPTY SET: t IS A LEAF NODE, LABELED AS THE MAJORITY CLASS Y_d



- DECISION TREE INDUCTION:

  IT ADOPTS A GREEDY STRATEGY -> BEST ATTRIBUTE IS SELECTED LOCALLY AT EACH STEP, IT'S NOT A GLOBAL OPTIMUM

  BINARY vs MULTIWAY SPLIT

  - ISSUES: STRUCTURES OF TEST CONDITIONS, BEST ATTRIBUTE SELECTION, STOPPING CONDITION
    - SPLITTING ON CONTINUOUS ATTRIBUTES:
      - DISCRETIZATION: STATIC (ONCE AT BEGINNING) OR DYNAMIC (DURING TREE INDUCTION)
      - BINARY DECISION: CONSIDER ALL POSSIBLE SPLITS AND FIND BEST CUT ($A < v$ OR $A \geq v$)

- **BEST ATTRIBUTE:**

  THE ONE WITH MOST HOMOGENEOUS CLASS DISTRIBUTION, LESS NODE IMPURITY

  - MEASURE OF NODE IMPURITY: GINI INDEX, ENTROPY, MISCLASSIFICATION ERROR

- **GINI INDEX:**

$$GINI(t) = 1 - \sum_j [p(j|t)]^2$$

$p(j|t)$ IS THE RELATIVE FREQUENCY OF CLASS j AT NODE t

LEAF FOUND

- $GINI(t) \in \left[0, 1 - \frac{1}{n_c}\right] \rightarrow \begin{cases} 0: \text{ALL RECORDS BELONG TO 1 CLASS} \\ 1 - \frac{1}{n_c}: \text{ALL RECORDS EQUALLY DISTRIBUTED} \end{cases}$ (n_c -> n° OF RECORDS)

ex.

C1	0
C2	6

- $\begin{cases} P(C1) = \frac{0}{6} = 0 \\ P(C2) = \frac{6}{6} = 1 \end{cases}$ -> $GINI = 1 - P(C1)^2 - P(C2)^2 = 0$

C1	1
C2	5

- $\begin{cases} P(C1) = \frac{1}{6} \\ P(C2) = \frac{5}{6} \end{cases}$ -> $GINI = 1 - \left(\frac{1}{6}\right)^2 - \left(\frac{5}{6}\right)^2 = 0.278$

SPLITTING BASED ON GINI:

GINI CAN BE USED AS A MEASURE FOR THE SPLITTING CRITERIA

- WHEN A NODE p IS SPLITTED IN K PARTITIONS, QUALITY OF THE SPLIT:

$$GINI_{SPLIT} = \sum_{i=1}^{K} \frac{n_i}{n} GINI(i)$$

n_i : n° OF RECORDS AT CHILD i

n : n° OF RECORDS AT NODE p

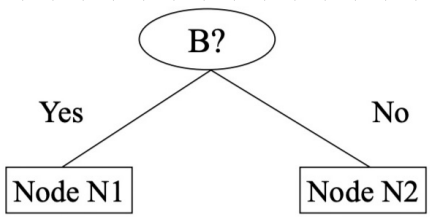
es.

- $GINI(N1) = 1 - \left(\frac{5}{7}\right)^2 - \left(\frac{2}{7}\right)^2 = 0.408$
- $GINI(N2) = 1 - \left(\frac{1}{5}\right)^2 - \left(\frac{4}{5}\right)^2 = 0.32$

$\rightarrow GINI(B\ SPLIT) = \frac{7}{12} 0.408 + \frac{5}{12} \cdot 0.32 = 0.371$

	Parent
C1	6
C2	6
Gini = 0.500	

	N1	N2
C1	5	1
C2	2	4
Gini=?		



- CATEGORICAL ATTRIBUTES:

  COMPUTE MULTI-WAY SPLITS

Multi-way split

	CarType		
	Family	Sports	Luxury
C1	1	2	1
C2	4	1	1
Gini	0.393		

Two-way split
(find best partition of values)

	CarType	
	{Sports, Luxury}	{Family}
C1	3	1
C2	2	4
Gini	0.400	

	CarType	
	{Sports}	{Family, Luxury}
C1	2	2
C2	1	5
Gini	0.419	

- CONTINOUS ATTRIBUTES:
  - BINARY DECISION ON 1 SPLITTING VALUE
  - $\forall$ POSSIBLE SPLITTING ON V: $A < V$ & $A > V$

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No
Taxable Income										
Sorted Values →	60	70	75	85	90	95	100	120	125	220
Split Positions →	55	65	72	80	87	92	97	110	122	172	230
	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >
Yes	0 3	0 3	0 3	0 3	1 2	2 1	3 0	3 0	3 0	3 0	3 0
No	0 7	1 6	2 5	3 4	3 4	3 4	3 4	4 3	5 2	6 1	7 0
Gini	0.420	0.400	0.375	0.343	0.417	0.400	0.300	0.343	0.375	0.400	0.420

- ENTROPY:

$$ENTROPY(t) = -\sum_j p(j|t) \log_2 p(j|t)$$

(t: NODE)

$ENTROPY(t) \in [0; \log_2 n_c]$ (0: LEAF FOUND)

- INFORMATION GAIN:

$$GAIN_{SPLIT} = ENTROPY(p) - \sum_{i=1}^{K} \frac{n_i}{n} ENTROPY(i)$$

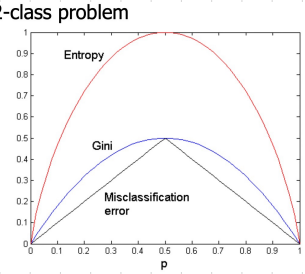
(p: PARENT NODE; K: PARTITIONS; n_i: RECORDS IN PARTITION i)

$\rightarrow$ S: TENDS TO PREFER SPLITS WITH LARGE NUMBER OF SMALL PARTITIONS $\rightarrow$ SOLVED BY GAIN RATIO

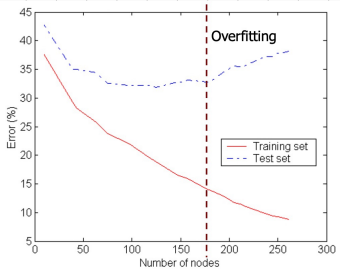
- GAIN RATIO:

$$GAIN\ RATIO_{SPLIT} = \frac{GAIN_{SPLIT}}{-\sum_{i=1}^{K} \frac{n_i}{n} \log_2 \frac{n_i}{n}}$$

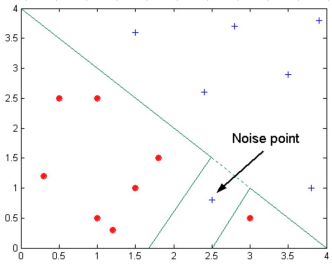
For a 2-class problem



- **STOPPING CRITERIA:**
  - ALL RECORDS BELONG TO THE SAME CLASS
  - ALL RECORDS HAVE SIMILAR ATTRIBUTE VALUES

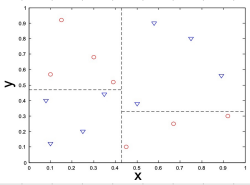


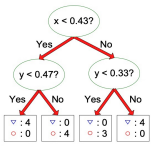
- **UNDERFITTING AND OVERFITTING:**
  - UNDERFITTING: WHEN MODEL IS TOO SIMPLE, BOTH TRAINING AND TEST ERRORS ARE LARGE
  - OVERFITTING. IT CAN BE CAUSED BY NOISE POINTS, DECISION BOUNDARY DISTORTED



→ OVERFITTING SOLUTION: PRE-PRUNING → EARLY STOPPING RULE
POST-PRUNING

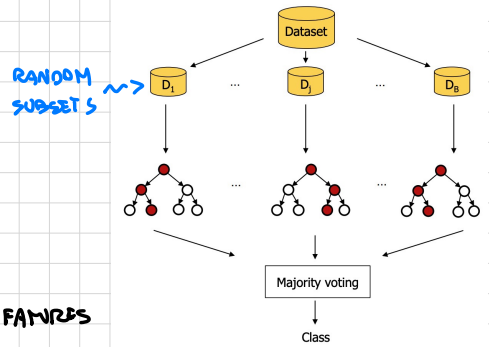
- DATA FRAGMENTATION: NUMBER OF INSTANCES AT LEAF NODE COULD BE TOO SMALL TO MAKE ANY STATISTICALLY SIGNIFICANT DECISION
- MISSING ATTRIBUTE VALUES
- DECISION BOUNDARY: PARALLEL TO AXES, ∵ TEST CONDITION INVOLVE SINGLE ATTRIBUTE AT A TIME





- **EVALUATION OF DECISION TREES:**
  - ACCURACY: COMPARABLE TO OTHER CLASSIFICATION TECHNIQUES FOR SIMPLE DATASETS
  - INTERPRETABILITY: ONLY FOR SMALL TREES, SINCE RELATIONS ARE INTERPRETABLE
  - INCREMENTALITY: NOT INCREMENTAL
  - ⊙ EFFICIENCY: FAST MODEL BUILDING, VERY FAST CLASSIFICATION
  - ⊙ SCALABILITY: SCALABLE BOTH IN TRAINING SET SIZE AND ATTRIBUTE NUMBER
  - ROBUSTNESS: DIFFICULT MANAGEMENT OF MISSING DATA

RANDOM FOREST :



- MULTIPLE BASE MODELS ARE COMBINED
  -> A NUMBER OF DECISION TREES BUILT AT SAME TIME
- CLASS ASSIGNED BY MAJORITY VOTING
- ∀ SUBSET -> TREE IS LEARNED ON A RANDOM SET OF FEATURES

- **BOOTSTRAP AGGREGATION :**
  GIVEN A TRAINING SET D OF n INSTANCES, IT SELECTS B TIMES A RANDOM SAMPLE (WITH REPLACEMENT) FROM D AND TRAIN TREES
  - FOR $b = 1, \dots, B$: DATASET SUBSET D_b GENERATED, CLASSIFICATION TREE ON D_b
- **FEATURE BAGGING :**
  ∀ CANDIDATE SPLITS, SELECT A RANDOM SUBSET OF THE FEATURES ~> IF p = n° FEATURES
  TYPICALLY $\sqrt{p}$,

- **N.B :** https://data36.com/random-forest-in-python/
  - DATASET : IF N TREES -> N DATASET SUBSETS (WITH REPLACEMENT)
  - ∀ TREE : 1 SUBSET + ALL FEATURES CONSIDERED
    ↳ ∀ SPLIT -> A RANDOM SUBSET OF FEATURE CHOSEN

- **EVALUATION :**

↑ • ACCURACY : HIGHER THAN DECISION TREES
  • INTERPRETABILITY : MODEL AND PREDICTION NOT INTERPRETABLE
  • INCREMENTALITY : NOT INCREMENTAL
  ⊙ EFFICIENCY : FAST MODEL BUILDING, VERY FAST CLASSIFICATION
  ⊙ SCALABILITY : SCALABLE BOTH IN TRAINING SET SIZE AND ATTRIBUTE NUMBER
  ⊙ ROBUSTNESS : ROBUST TO NOISE AND OUTLIERS

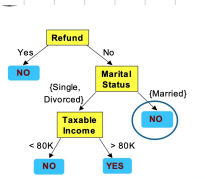
- # RULE - BASED CLASSIFICATION :

CLASSIFY RECORDS BY USING : (CONDITION) —> CLASS

ex.

- (Blood Type=Warm) ∧ (Lay Eggs=Yes) → Birds
- (Taxable Income < 50K) ∧ (Refund=Yes) → Cheat=No

- RULE r "COVERS" AN INSTANCE X : ATTRIBUTE OF INSTANCE SATISFY CONDITION OF THE RULE
- RULES CHARACTERISTICS .
  - MUTUALLY EXCLUSIVE : 2 RULE CONDITIONS CAN'T BE TRUE AT THE SAME TIME
    ∀ RECORD -> 1 RULE AT MOST EVERY RECORD IS COVERED BY AT MOST 1 RULE
  - EXHAUSTIVE : RULES ACCOUNT ∀ COMBINATIONS OF ATTRIBUTE VALUES
    ∀ RECORD -> 1 RULE AT LEAST ∀ RECORD -> COVERED BY AT LEAST 1 RULE



Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Initial Rule: (Refund=No) ∧ (Status=Married) → No
Simplified Rule: (Status=Married) → No

- DECISION TREES CAN BE TRANSFORMED INTO RULES
- SOMETIMES RULES CAN BE SIMPLIFIED
  - RULES ARE NO LONGER MUTUALLY EXCLUSIVE
    - SOL : ORDERED SET RULES / VOTING SCHEMES FOR UNORDERED RULE SET
      - OR RANK ORDERING ACCORDING TO PRIORITY
  - RULES NO LONGER EXHAUSTIVE
    - RECORD MAY NOT TRIGGER ANY RULES
      - SOL : USE A DEFAULT CLASS

- BUILDING :
  - DIRECT METHOD : RULES FROM DATA
  - INDIRECT METHOD : RULES FROM OTHER CLASSIFICATION MODEL

- EVALUATION :

- ACCURACY : HIGHER THAN DECISION TREES
- INTERPRETABILITY : MODEL AND PREDICTION INTERPRETABLE
- INCREMENTALITY : NOT INCREMENTAL
- EFFICIENCY : FAST MODEL BUILDING, VERY FAST CLASSIFICATION
- SCALABILITY : SCALABLE BOTH IN TRAINING SET SIZE AND ATTRIBUTE NUMBER
- ROBUSTNESS : ROBUST TO OUTLIERS

- ## ASSOCIATIVE CLASSIFIERS :

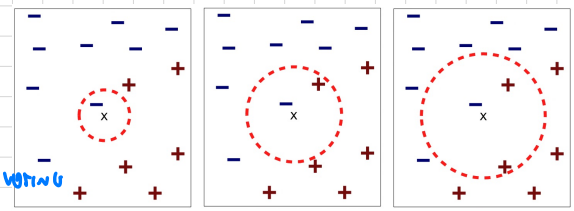
  (CONDITION) ⟶ Y

  - MODEL GENERATION :
    - RULE SELECTING AND SORTING : BASED ON SUPPORT, CONFIDENCE AND CORRELATION THRESHOLDS
    - RULE PRUNING : TRAINING SET COVERED BY SELECTING TOP MOST RULES ACCORDING TO PREVIOUS SORT
  - EVALUATION :
    - ↑ ACCURACY : HIGHER THAN DECISION TREES AND RULE-BASED CLASSIFIERS ⇝ CORRELATION IS CONSIDERED TOO
    - ⊙ INTERPRETABILITY : MODEL AND PREDICTION INTERPRETABLE
    - INCREMENTALITY : NOT INCREMENTAL
    - EFFICIENCY : RULE GENERATION MAY BE SLOW (↓ SUPPORT THRESHOLD), FAST CLASSIFICATION
    - SCALABILITY : SCALABLE BOTH IN TRAINING SET SIZE BUT REDUCED SCALABILITY IN ATTRIBUTE NUMBER
    - ⊙ ROBUSTNESS : ROBUST TO OUTLIERS, UNEFFECTED BY MISSING DATA

- ## K-NEAREST NEIGHBOR (KNN) :

  USE TRAINING RECORDS TO PREDICT THE CLASS LABEL OF UNSEEN LABELS

  - INSTANCE BASED CLASSIFIER
    - ROTE LEARNER : MEMORIZES ALL TRAINING DATA, CLASSIFY ONLY IF EXACT MATCH
    - NEAREST NEIGHBOR : USES K "CLOSEST" POINT
  - CLASSIFICATION :
    1. COMPUTE DISTANCE TO OTHER TRAINING RECORDS
    2. IDENTIFY K NEARBEST NEIGHBORS
    3. ASSIGN CLASS ACCORDING TO K NEAREST ⇝ MAJORITY VOTING



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

- K=1 : VORONOI DIAGRAMS
- DISTANCE :

  eg. EUCLIDEAN : $d(p,q) = \sqrt{\sum_i (p_i - q_i)^2}$

  ⇝ WEIGHT VOTE ∝ DISTANCE : $w = \frac{1}{d^2}$

- K
  - TOO SMALL: SENSITIVE TO NOISE
  - TOO LARGE: MAY INCLUDE POINTS FROM OTHER CLASSES

- SCALING ISSUE: ATTRIBUTE SHOULD BE NORMALIZED
  - e.g. HEIGHT ∈ [1.5 m ; 2 m], INCOME ∈ [10·10³ ; 10·10⁶]

- EVALUATION:
  - ACCURACY: COMPARABLE TO OTHER TECHNIQUES FOR SMALL DATASETS
  - INTERPRETABILITY: MODEL AND PREDICTION NOT INTERPRETABLE
  - INCREMENTALITY: INCREMENTAL, TRAINING SET MUST BE AVAILABLE
  - EFFICIENCY: (ALMOST) NO MODEL BUILDING, SLOWER CLASSIFICATION (COMPUTE d(·))
  - SCALABILITY: USUALLY SCALABLE IN TRAINING SET SIZE
  - ROBUSTNESS: DEPENDS ON DISTANCE COMPUTATION

BAYESIAN CLASSIFICATION:

- BAYES THEOREM:

$$P(C|X) = \frac{P(X|C)\,P(C)}{P(X)} \qquad P(X) \Rightarrow = 1$$

C: ANY CLASS LABEL

X = ⟨x₁, …, xₙ⟩ RECORD TO BE CLASSIFIED

- BAYESIAN CLASSIFICATION:
  1. ∀ CLASS → COMPUTE $P(C|X)$: p THAT X BELONGS TO C
  2. ASSIGN X TO CLASS WITH MAX $P(C|X)$

- SUPPOSE ⟨x₁, …, xₖ⟩ STAT. INDIP.: $P(x_1, \dots, x_k \mid C) = P(x_1|C) \cdot \dots \cdot P(x_n|C)$

20.

$P(c|\vec{x}) = P(\vec{x}|c)\,P(c)$

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

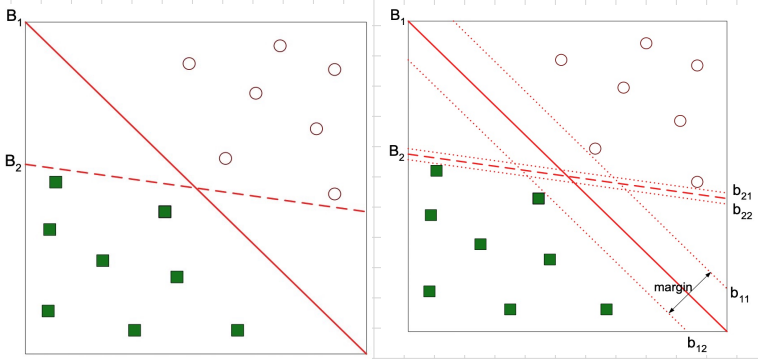
outlook	
P(sunny|p) = 2/9	P(sunny|n) = 3/5
P(overcast|p) = 4/9	P(overcast|n) = 0
P(rain|p) = 3/9	P(rain|n) = 2/5
temperature	
P(hot|p) = 2/9	P(hot|n) = 2/5
P(mild|p) = 4/9	P(mild|n) = 2/5
P(cool|p) = 3/9	P(cool|n) = 1/5
humidity	
P(high|p) = 3/9	P(high|n) = 4/5
P(normal|p) = 6/9	P(normal|n) = 2/5
windy	
P(true|p) = 3/9	P(true|n) = 3/5
P(false|p) = 6/9	P(false|n) = 2/5

P(p) = 9/14
P(n) = 5/14

- Data to be labeled
  X = <rain, hot, high, false>
- For class p
  $P(X|p)\cdot P(p) =$
  $= P(rain|p)\cdot P(hot|p)\cdot P(high|p)\cdot P(false|p)\cdot P(p)$
  $= 3/9\cdot 2/9\cdot 3/9\cdot 6/9\cdot 9/14 = 0.010582$
- For class n
  $P(X|n)\cdot P(n) =$
  $= P(rain|n)\cdot P(hot|n)\cdot P(high|n)\cdot P(false|n)\cdot P(n)$
  $= 2/5\cdot 2/5\cdot 4/5\cdot 2/5\cdot 5/14 = 0.018286$

- EVALUATION :
  - ACCURACY : SIMILAR / LOWER TO DECISION TREES
  - INTERPRETABILITY : MODEL AND PREDICTION NOT INTERPRETABLE
  - INCREMENTALITY : FULLY INCREMENTAL
  - EFFICENCY : FAST MODEL BUILDING, VERY FAST CLASSIFICATION
  - SCALABILITY : SCALABLE IN TRAINING SET SIZE AND ATTRIBUTE NUMBERS
  - ROBUSTNESS : AFFECTED BY ATTRIBUTE CORRELATION

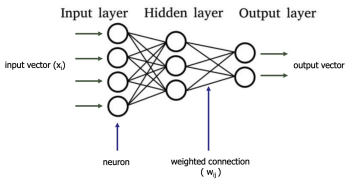
- SUPPORT VECTOR MACHINES :
  - FIND A LINEAR DECISION BOUNDARY THAT WILL SEPARATE THE DATA
  - B_1 OR B_2 ?
    → THE ONE WHICH MAXIMIZES MARGINS

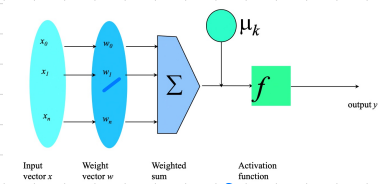


- BOUNDARY NOT LINEAR ? → TRANSFORM DATA INTO OTHERS DIMENSIONS

- EVALUATION :
  - ACCURACY : AMONG BEST PERFORMERS
  - INTERPRETABILITY : MODEL AND PREDICTION NOT INTERPRETABLE
  - INCREMENTALITY : NOT INCREMENTAL
  - EFFICENCY : MODEL BUILDING REQUIRES SIGNIFICANT PARAMETER TUNING, VERY FAST CLASSIFICATION
  - SCALABILITY : MEDIUM SCALABLE IN BOTH TRAINING SET SIZE AND ATTRIBUTE NUMBERS
  - ROBUSTNESS : ROBUST TO ATTRIBUTE CORRELATION

ARTIFICIAL NEURAL NETWORKS:



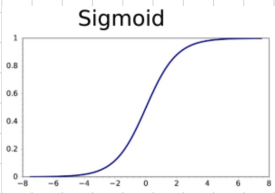


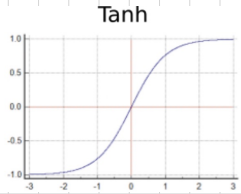
- ACTIVATION FUNCTIONS:

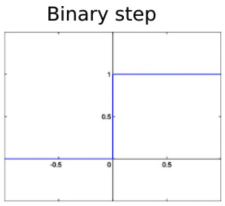
  SIMULATES ACTIVATION ON INPUT STIMULI

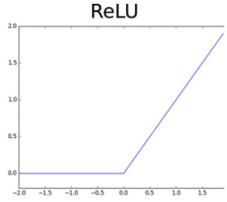
  - NEURON OUTPUT IN FIXED RANGE,
  - REGULATE IF NEXT NEURON IS ACTIVATED OR NOT

$$\text{SOFTMAX}(z_j) = \frac{e^{z_j}}{\sum_{i=0}^{N-1} e^{z_i}}$$









SOFTMAX

- USED IN OUTPUT LAYER
- GENERATE A FINAL PMF

- BUILDING A FFNN: ⇝ FEED FORWARD NN
  - $\forall$ NODE : $\{w_1, \dots, w_m\}$ (WEIGHTS) AND OFFSET VALUES
  - ALGORITHM:
    - INITIALLY: RANDOM VALUES $\forall w_i$
    - INSTANCES PROCESSING:
      - $\forall$ NEURON : COMPUTE RESULT (·) $\{w_1, \dots, w_m\}$, OFFSET, ACTIVATION FUNCTION
      - FORWARD PROPAGATION UNTIL OUTPUT COMPUTED
      - COMPARE OUTPUT WITH TRUE VALUES
      - BACK PROPAGATION

      -> IT STOPS $\Leftrightarrow$ (ACCURACY > $\epsilon_{\text{TRESHOLD}}$), (% ERROR ON PARTITION > ϵ), MAX # EPOCH

- EVALUATION:

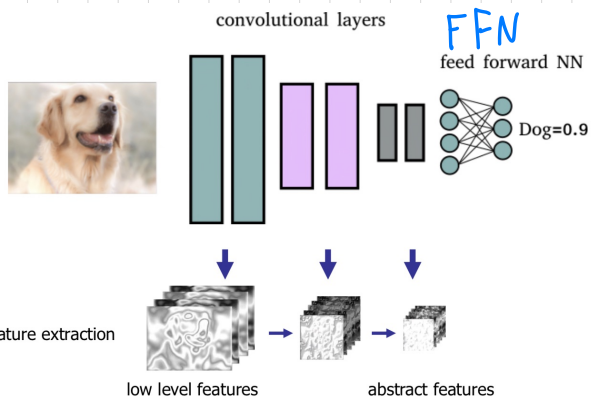
  ① ACCURACY: AMONG BEST PERFORMERS

  - INTERPRETABILITY: MODEL AND PREDICTION NOT INTERPRETABLE (BLACK BOX)
  - INCREMENTALITY: NOT INCREMENTAL
  - EFFICIENCY: MODEL BUILDING REQUIRES SIGNIFICANT PARAMETER TUNING, VERY FAST CLASSIFICATION
  - SCALABILITY: MEDIUM SCALABLE IN BOTH TRAINING SET SIZE AND ATTRIBUTE NUMBER

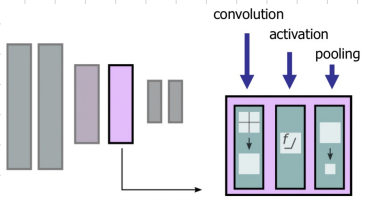
  ② ROBUSTNESS: ROBUST TO NOISE AND OUTLIERS, REQUIRE LARGE TRAINING SET

- **CNN** : ~> CONVOLUTIONAL NN

AUTOMATIC EXTRACTION OF FEATURES FROM IMAGES AND PERFORM CLASSIFICATION



- **CONVOLUTIONAL LAYER** :
  - CONVOLUTION : FEATURES EXTRACTION WITH SLIDING FILTERS
  - ACTIVATION : APPLY ACTIVATION FUNCTIONS TO INPUT TENSOR
  - POOLING : TENSOR DOWNSAMPLING



- **TENSORS** :

N-DIMENSIONAL VECTOR

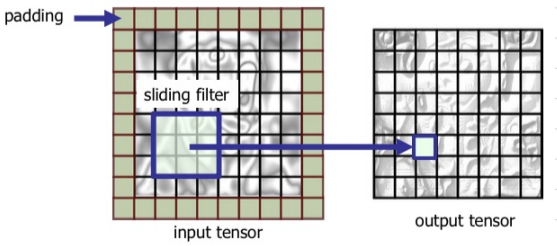
-> THEY FLOW THROUGH CNN LAYERS

DEPTH : 1 FOR GRAYSCALE, 3 FOR RGB

EG. IMAGES : RANK-3 TENSOR WITH $[d, h, w]$ (d: DIMENSIONS)

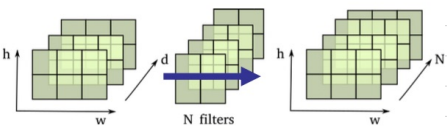
- **CONVOLUTION** :

INPUT IMAGE/TENSOR ⟼ OUTPUT TENSOR WITH EXTRACTED FEATURES



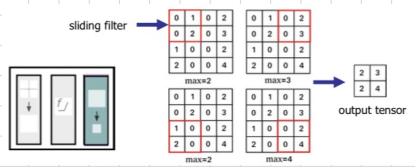
- SLIDING FILTER CONTAIN THE TRAINABLE WEIGHTS OF NN
- EACH CONVOLUTIONAL LAYER -> HUNDREDS OF FILTER

$[d, h, w] \longmapsto [N, h, w]$

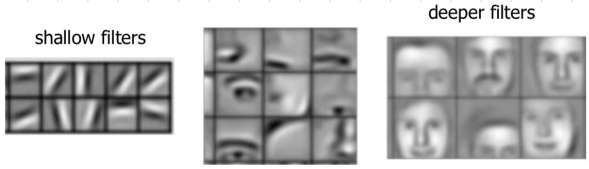


- **POOLING** :
  - SLIDING FILTERS : REPLACES TENSOR WITH SUMMARY STATISTICS OF NEARBY OUTPUTS

EG. MAXPOOL : COMPUTE MAX AS STATISTIC (MOST COMMON)

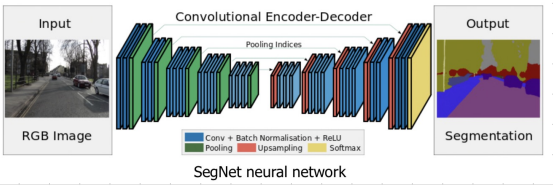


- DURING TRAINING, EACH FILTER LEARN TO RECOGNISE A PARTICULAR PATTERN



- SHALLOW LAYERS AND DEEPER LAYER

- **SEMANTIC SEGMENTATION CNN's :**
  - ∀ PIXEL OF IMAGE → CLASS
  - ENCODER NETWORK + DECODER NETWORK



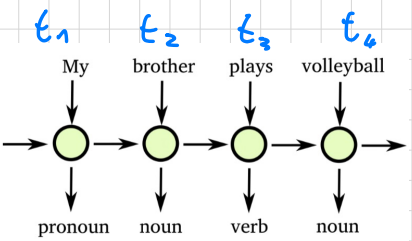
SegNet neural network

- **RECURRENT NN :**

SEQUENTIAL DATA PROCESSING $X(t)$

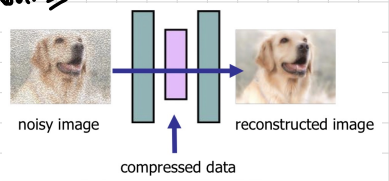
→ THEY KEEP A STATE AT TIME t

- INPUT : $x(t)$, PREVIOUS STATE $s(t-1)$ → OUTPUT $Y(t)$
- ERROR $Y(t) - X(t)$ BACKPROPAGATED ∀ STEP
  - ↳ PROBLEM : VANISHING GRADIENT → ∀ STEP → LESS INPUT TO CHANGE NEXT WEIGHT
  - ↳ SOL : LSTM (LONG SHORT TERM MEMORIES)



- **AUTOENCODERS :**

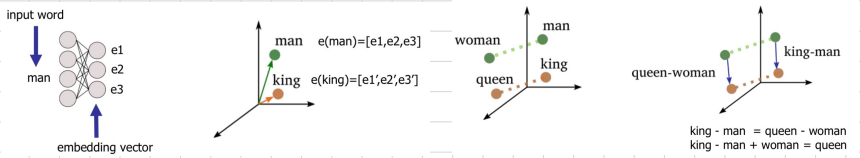
COMPRESS INPUT AND THEN RECONSTRUCT INPUT



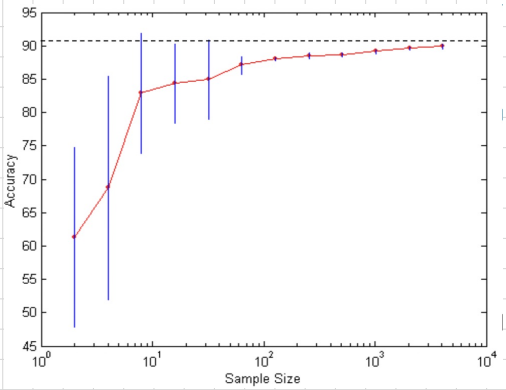
- **WORD EMBEDDINGS :**

WORDS ⟷ VECTORS

→ MATH OPERATIONS CAN BE PERFORMED



- **MODEL EVALUATION:**



  - **LEARNING CURVE:**
    - ACCURACY = f (SAMPLE SIZE)
    - IT NEEDS LARGE SAMPLES, OTHERWISE: BIAS IN ESTIMATE, VARIANCE OF ESTIMATES

- **METHODS OF ESTIMATION:** HOLDOUT, CROSS VALIDATION
  - HOLD OUT: 80% FOR TRAINING, 20% FOR TESTING
  - CROSS VALIDATION: PARTITION DATA IN K DISJOINT SUBSETS
    - K-FOLD: $\forall$ FOLD → TRAIN ON K-1 PARTITIONS, TEST ON REMAINING ONE

- **MODEL VALIDATION:**
  - TRAINING 60%, VALIDATION 20%, TEST 20%

- **METRICS FOR MODEL EVALUATION:** ~> ex. FOR BINARY CLASSIFIER

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

  - **ACCURACY** $= \dfrac{TP + TN}{TP + TN + FP + FN}$

  - NOT ALWAYS RELIABLE:

ex. $\begin{cases} C:0 \rightarrow \#: 9900 \\ C:1 \rightarrow \#: 100 \end{cases}$ MODEL: $\forall \mapsto C:0$ → ACC $= \dfrac{9900}{10000} = 99\%$ (?)

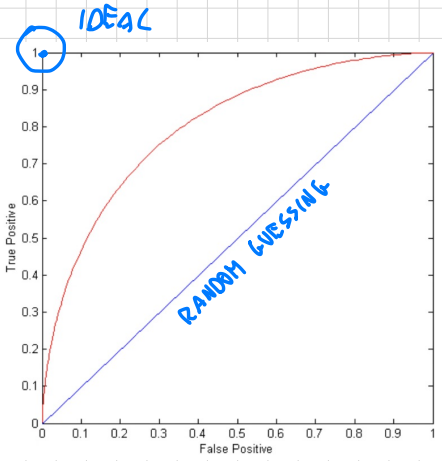
→ INAPPROPRIATE FOR UNBALANCED LABEL DISTRIBUTION

  - **RECALL** $(c = YES) = \dfrac{TP}{TP + FN}$

  - **PRECISION** $(c = YES) = \dfrac{TP}{TP + FP}$

  - **F1-SCORE** $(c) = 2 \cdot \dfrac{\text{PRECISION} \cdot \text{RECALL}}{\text{PRECISION} + \text{RECALL}} = 2 \cdot \dfrac{P \cdot R}{P + R}$

- **ROC** : ~> RECEIVER OPERATING CHARACTERISTICS



- GIVEN :
  - $TPR = \frac{TP}{TP + FN}$, $FPR = \frac{FP}{FP + TN}$

TPR : TRUE POSITIVE RATE

FPR : FALSE POSITIVE RATE

-> ROC : $TPR = f(FPR)$

- $(0, 1)$; IDEAL
- $(0, 0)$: DECLARE EVERYTHING TO NEGATIVE CLASS
- $(1, 1)$; DECLARE EVERYTHING TO POSITIVE CLASS

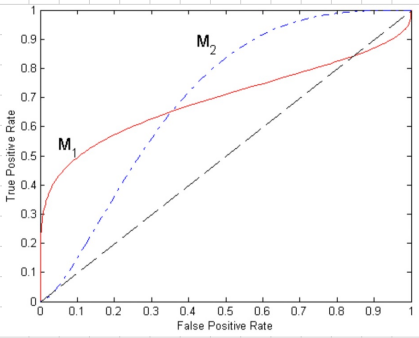
-> DIAGONAL LINE : RANDOM GUESSING

-> BELOW DIAG. LINE; PREDICTION IS OPPOSITE OF THE TRUE CLASS

- BUILDING : USE CLASSIFIER THAT PRODUCES $P(+|A)$, $\forall$ TEST INSTANCE

  SORT INSTANCES DESC

  APPLY THRESHOLD $\forall$ VALUE IN $P(+|A)$



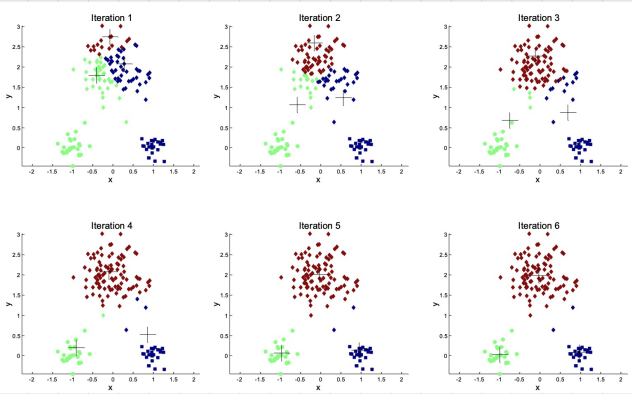
- M1 : BETTER FOR SMALL FPR
- M2 : BETTER FOR LARGE FPR

- # CLUSTERING FUNDAMENTALS:

  - PARTITIONAL: DATA ⟶ NON OVERLAPPING SET
  - HIERARCHICAL CLUSTERING: DATA ⟶ NESTED CLUSTERS

- CATEGORIES:
  - WELL SEPARATED
  - CENTER-BASED
  - CONTIGUOUS CLUSTERS
  - DENSITY BASED
  - SHARED PROPERTY

- ## K-MEANS CLUSTERING:



  - K CLUSTERS, ∀ CLUSTER → CENTROID

  S: CLUSTERS WITH DIFFERENT SHAPES, DENSITIES, NON GLOBULAR SHAPES

  - ALGORITHM:

    1. SELECT K
    2. WHILE (CENTROIDS DON'T CHANGE):
       - 2.1 FORM K CLUSTER COMPUTING DISTANCE TO CLOSEST CENTROID
       - 2.2 RECOMPUTE CENTROID

  - CLUSTERS EVALUATION:

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} DIST^2(m_i, x)$$

    ↳ SUM OF SQUARED ERRORS

  - PRE PROCESSING: NORMALIZE DATA, ELIMINATE OUTLIERS
  - POST PROCESSING: DELETE SMALL CLUSTERS, SPLIT CLUSTERS / HIGH SSE, MERGE CLUSTERS / LOW SSE

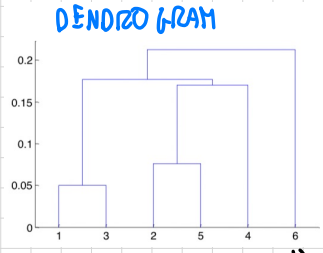
- ## BISECTING K-MEANS:

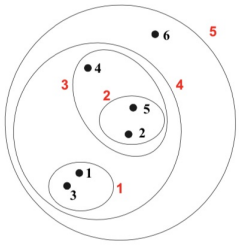
K-MEANS, BUT 1 CLUSTER IN LIST IS BISECTED THEN SELECT 2 CLUSTERS WITH LOWER SSE AND ADD TO LIST

1: Initialize the list of clusters to contain the cluster containing all points.
2: **repeat**
3: Select a cluster from the list of clusters
4: **for** $i = 1$ to *number_of_iterations* **do**
5: Bisect the selected cluster using basic K-means
6: **end for**
7: Add the two clusters from the bisection with the lowest SSE to the list of clusters.
8: **until** Until the list of clusters contains K clusters

- **HIERARCHICAL CLUSTERING:**

SET OF NESTED CLUSTERS ORGANIZED AS HIERARCHICAL TREE





V: DO NOT HAVE TO ASSUME A N° OF CLUSTER -> JUST NEED TO "CUT" AT A LEVEL

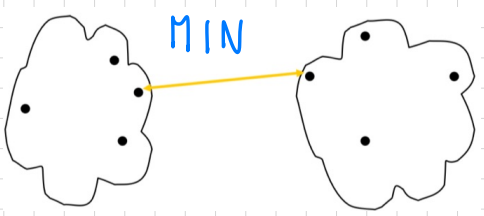
- TYPES:
  - AGGLOMERATIVE: START AS ALL-INCLUSIVE CLUSTER —> JOIN 2 CLOSEST CLUSTER UNTIL 1/K REMAIN (∀ STEP)
  - DIVISIVE: START AS ALL-INCLUSIVE CLUSTER —> SPLIT CLUSTER UNTIL 1/K REMAINING (∀ STEP)

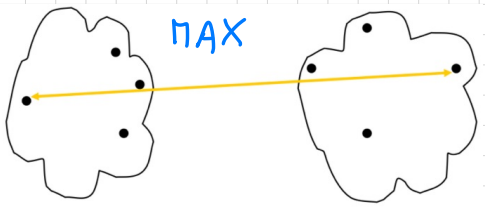
-> USES OF A SIMILARITY OR DISTANCE MATRIX

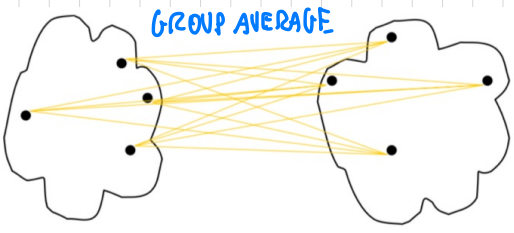
- **ALGORITHM** (MOST POPULAR):

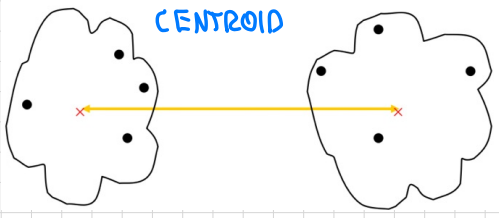
1. COMPUTE PROXIMITY MATRIX
2. LET EACH DATA POINT BE A CLUSTER
3. while (UNTIL 1 CLUSTER REMAIN):
   - 3.1 MERGE THE 2 CLOSEST CLUSTERS
   - 3.2 UPDATE PROXIMITY MATRIX -> DISTINGUISH ALGORITHM DIFFERENCES (DIFFERENCE IN DEFINING CLUSTERS DISTANCE)

- HOW TO DEFINE CLUSTERS SIMILARITY ?



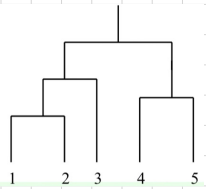


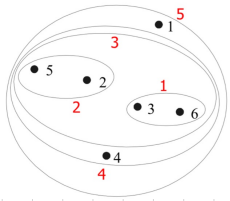


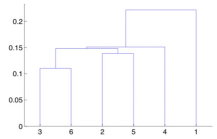


- **MIN**:

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00

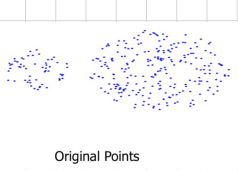


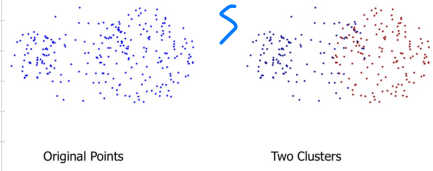




V: NON-ELLIPTICAL SHAPE HANDLING

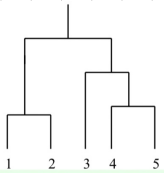
S: SENSITIVE TO NOISE AND OUTLIERS

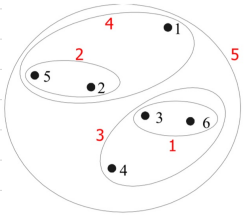


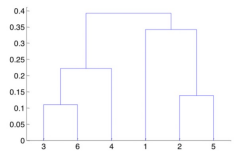


- **MAX**:

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



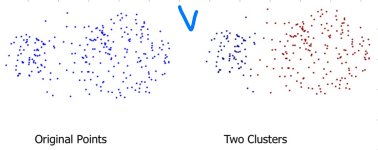




V: LESS SUSCEPTIBLE TO NOISE AND OUTLIERS

S: TENDS TO BREAK LARGE CLUSTERS

BIASED TOWARDS GLOBULAR CLUSTERS

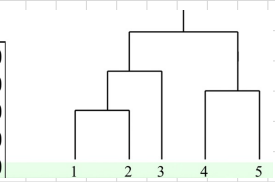


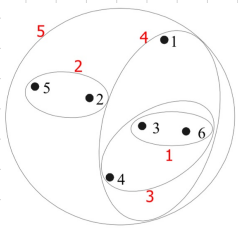
- **GROUP AVERAGE**:

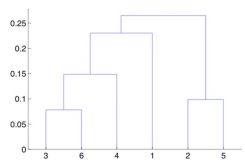
$$\text{proximity}(C_i, C_j) = \frac{\sum_{p \in C_i} \text{proximity}(p_i, p_j)}{|C_i| \cdot |C_j|}$$

(C_i: cluster, $p \in C_i$)

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00







V: LESS SUSCEPTIBLE TO NOISE AND OUTLIERS

S: BIASED TOWARDS GLOBULAR CLUSTERS

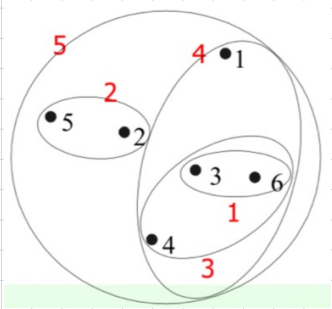
- **WARD'S METHOD**:

- SIMILARITY ↔ INCREASE OF MSE WHEN 2 CLUSTERS ARE MERGED

→ HIERARCHICAL ANALOGUE OF K-MEANS

V: LESS SUSCEPTIBLE TO NOISE AND OUTLIERS

S: BIASED TOWARDS GLOBULAR CLUSTERS



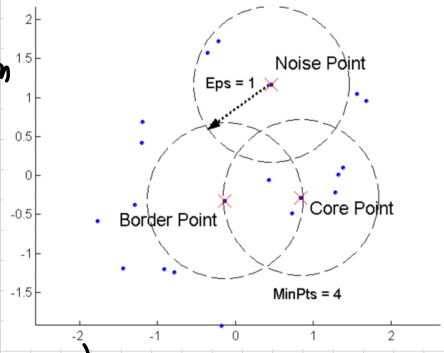
- COMPLEXITY:
  - $O(n^2)$ IN SPACE: USE OF A MATRIX $n \times n$
  - $O(n^3)$ IN TIME: N STEPS / $\forall$ STEP $\rightarrow N^2$ MATRIX TO BE UPDATED
    $\rightarrow$ CAN BE OPTIMIZED TO $O(N^2 \log(N))$

- **DBSCAN**:

  IT IS A DENSITY BASED ALGORITHM / DENSITY: n° POINTS WITHIN A SPECIFIC RADIUS $\sim$ EPS

  C_{EPS}: CIRCLE OF RADIUS EPS

  $\rightarrow$ TRESHOLD FOR MINIMUM POINTS

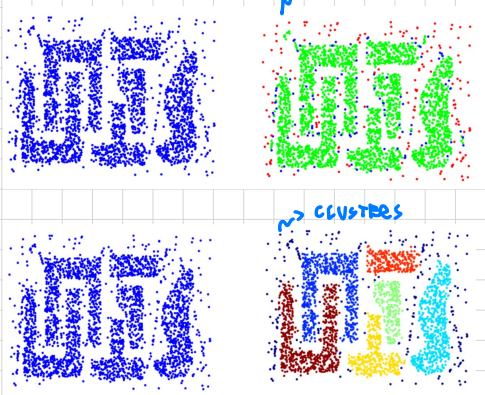


  - CORE POINT $\iff$ #POINTS > MINPTS IN C_{EPS}
  - BORDER POINT $\iff$ (# POINTS $\in C_{EPS} <$ MIN PTS) AND ($\in$ NEIGHBORHOOD OF A CORE POINT)
  - NOISE POINT $\iff$ (! CORE POINT) AND (! BORDER POINT)

CORE, BORDER, NOISE POINTS

- **ALGORITHM**:

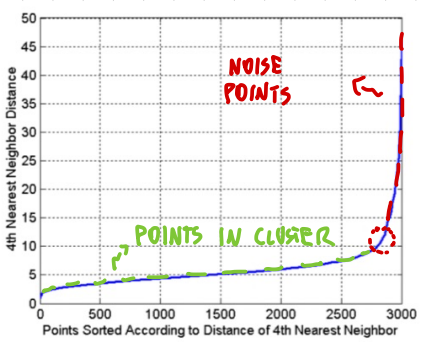
```
current_cluster_label ← 1
for all core points do
  if the core point has no cluster label then
    current_cluster_label ← current_cluster_label + 1
    Label the current core point with cluster label current_cluster_label
  end if
  for all points in the Eps-neighborhood, except i^th the point itself do
    if the point does not have a cluster label then
      Label the point with cluster label current_cluster_label
    end if
  end for
end for
```



V: NOISE RESISTANT

CAN HANDLE DIFFERENT CLUSTER SHAPES

- **DETERMINING EPS AND MIN PTS:**
  - $\forall$ POINT IN CLUSTER $\rightarrow$ K^{TH} NEAREST NEIGHBOUR $\approx$ SAME DISTANCE
    $\rightarrow$ NOISE POINTS HAVE K^{TH} NEAREST NEIGHBOUR AT DISTANCE $\gg 0$



• CLUSTER VALIDITY :

- DETERMIN CLUSTERING TENDENCY OF A SET OF DATA -> ∃ NON RANDOM DATA ?
- COMPARISON OF RESULT WITH WELL KNOWN RESULTS
- EVALUATING CLUSTER ANALYSIS RESULTS
- COMPARING RESULTS OF 2 DIFFERENT SET OF CLUSTERS ANALYSIS
- DETERMING "CORRECT" n° OF CLUSTERS

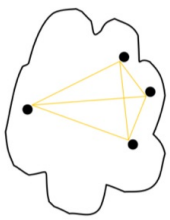
• MEASURE OF CLUSTER VALIDITY :

- EXTERNAL INDEX : MEASURE WHICH CLUSTER LABEL MATCH EXTERNAL SUPPLIED LABLES
  -> ex. ENTROPY, PURITY
- INTERNAL INDEX : MEASURE OF A CLUSTER GOODNESS ⊄ EXTERNAL INFO
- RELATIVE INDEX : COMPARE 2 DIFFERENT CLUSTERS / CLUSTERING
  -> ex. SSE, ENTROPY

• INTERNAL MEASURE :

• CLUSTER COHESION :

MEASURE OF HOW CLOSELY RELATED ARE OBJECT IN CLUSTER

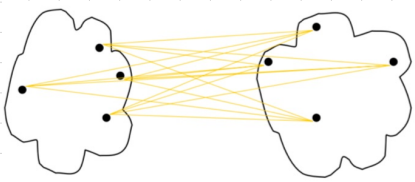


$$WSS = \sum_{i}^{\forall \text{ CLUSTER}} \sum_{x \in C_i} (x - m_i)^2$$

m_i -> CENTROID

• CLUSTER SEPARATION :

MEASURE OF HOW WELL DISTINCTED CLUSTERS ARE



$$BSS = \sum_{i} |C_i| (m - m_i)^2$$

• SILHOUETTE : HOW WELL OBJECT LIES IN ITS CLUSTER, (-) COHESION, SEPARATION

——> COMPUTABLE FOR INDIVIDUAL POINTS, INDIVIDUAL C, ALL CLUSTERING

- $\in [-1, 1]$
- BETTER IF ↦ 1

$$s(i) = \frac{b(i) - a(i)}{\max\{b(i), a(i)\}}$$

OBJECT i

$a(i)$: AVG DISSIMILARITY OF i WITH ALL OTHER OBJECTS IN C OF POINT i (-> CHECK DISTANCE MATRIX)

$b(i)$: $\min\{$ AVG DISSIMILARITY OF i TO ANY POINT NOT IN C OF i, BUT IN OTHER C $\}$

- AVG $(s(i))$ ON ALL p IN CLUSTER : HOW WELL RIGHT DATA ARE IN C
- AVG $(s(i))$ ON ALL DATASET : CORRECTNESS ON THE CLUSTERING

- **EXTERNAL MEASURES:**
  - **ENTROPY AND PURITY**

entropy For each cluster, the class distribution of the data is calculated first, i.e., for cluster j we compute p_{ij}, the 'probability' that a member of cluster j belongs to class i as follows: $p_{ij} = m_{ij}/m_j$, where m_j is the number of values in cluster j and m_{ij} is the number of values of class i in cluster j. Then using this class distribution, the entropy of each cluster j is calculated using the standard formula $e_j = \sum_{i=1}^{L} p_{ij} \log_2 p_{ij}$, where the L is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e., $e = \sum_{i=1}^{K} \frac{m_i}{m} e_j$, where m_j is the size of cluster j, K is the number of clusters, and m is the total number of data points.

purity Using the terminology derived for entropy, the purity of cluster j, is given by $purity_j = \max p_{ij}$ and the overall purity of a clustering by $purity = \sum_{i=1}^{K} \frac{m_i}{m} purity_j$.

Table 5.9. K-means Clustering Results for LA Document Data Set

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Entropy	Purity
1	3	5	40	506	96	27	1.2270	0.7474
2	4	7	280	29	39	2	1.1472	0.7756
3	1	1	1	7	4	671	0.1813	0.9796
4	10	162	3	119	73	2	1.7487	0.4390
5	331	22	5	70	13	23	1.3976	0.7134
6	5	358	12	212	48	13	1.5523	0.5525
Total	354	555	341	943	273	738	1.1450	0.7203

- **RAND INDEX:**
  - MAIN IDEA: 2 OBJECT IN SAME CLUSTER $\Leftrightarrow$ SAME CLASS

$$\text{RAND INDEX} = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}}$$

PAIRS OF 2 POINTS: (p_1, p_2) $\rightsquigarrow$ CONSIDER DIFFERENT COMBINATION

- f_{00}: # PAIRS / CLASS IS $\neq$, C_i IS $\neq$
- f_{01}: # PAIRS / CLASS IS $\neq$, C_i IS $=$
- f_{10}: # PAIRS / CLASS IS $=$, C_i IS $\neq$
- f_{11}: # PAIRS / CLASS IS $=$, C_i IS $=$

- **REGRESSION ANALYSIS:**

DEF.: REGRESSION . PREDICTION OF A VALUE $\in \mathbb{R}$ ↔ CLASSIFICATION: PREDICTION OF A FINITE CLASS

- REGRESSION FINDS A MODEL THAT ALLOWS PREDICTING THE TARGET VARIABLE OF NEW OBJECT USING $Y = f(x_1, \dots, x_m)$

- **SIMPLE LINEAR REGRESSION:**

$$Y = \beta_0 + \beta_1 \cdot X$$

- X: INDEPENDENT VARIABLE
- Y: ESTIMATED / PREDICTED VARIABLE
- β_1: ESTIMATION OF REGRESSION SLOPE ($Y = mx + q$)
- β_0: REGRESSION INTERCEPT ($Y = mx + q$)
  - ↳ ESTIM. Y WHEN $X = 0$

- HOW TO FIND β_0 AND β_1?

→ ONE METHOD IS MINIMIZING THE RESIDUAL SUM OF SQUARES (RSS):

- GIVEN Y: ACTUAL VALUES, $\hat{Y}$: PREDICTED VALUES:

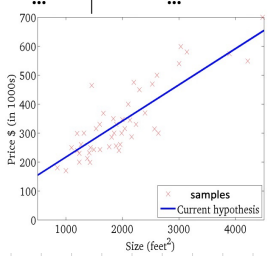
$$\min(RSS) = \min\left(\sum_i (Y_i - \hat{Y}_i)^2\right) = \min\left[\sum_i \left(Y_i - (\beta_0 + \beta_1 x)\right)^2\right]$$

- ESTIMATION OF β_0 AND β_1:

$$\cdot\ Y = \beta_0 + \beta_1 x \longrightarrow \begin{cases} \beta_1 = \dfrac{\sum_i (x_i - \bar{x})(Y_i - \bar{Y})}{\sum_i (x_i - \bar{x})^2} \\ \beta_0 = \bar{Y} - \beta_1 x \end{cases} \quad \Big/ \quad \bar{Y} = \frac{1}{n}\sum_i Y_i, \quad \bar{x} = \frac{1}{n}\sum_i x_i$$

ex. ESTIM. HOME $ = f(\text{SIZE IN } \text{feet}^2)$

Size in feet²	Price ($) in 1000's
2104	460
1416	232
1534	315
852	178
...	...



- x: SIZE IN feet^2, Y: PRICE IN $ IN 1000's

→ $Y = \beta_0 + \beta_1 x$

- β_0 MEANING: PRICE WHEN $X = 0$

→ β_0 IS THE PORTION OF HOUSE PRICE WHICH IS INDEPENDENT FROM SQUARE FEET

- **MULTIPLE LINEAR REGRESSION:**

n-DIMENSIONAL PLANE EQUATION

$$Y = f(\vec{X}) = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \dots + \beta_n \cdot X_n + \varepsilon$$

RESIDUAL ε: PORTION OF Y WHICH IS NOT EXPLAINED BY THE MODEL

β_i: AVG EFFECT ON Y OF A UNIT INCREASE IN X_i, HOLDING FIXED ALL OTHER PREDICTORS

- CORRELATION AMONG X_i CAN CAUSE PROBLEMS
  - ↑ σ^2 OF COEFFICIENT
  - COMPLEX INTERPRETATION
    - L> WHEN X_i CHANGES -> EVERYTHING CHANGES
  - L> CAUSALITY SHOULD BE AVOIDED AMONG X_i's
- IF $\#X_i >> 0$ -> POSSIBILITY OF PROVIDING REDUNDANT INFORMATION
  - -> FEATURE X_i SELECTION AND REMOVAL (BASED ON CORRELATION)
  - -> FEATURE SELECTION BASED ON CORRELATION TEST

- **POLYNOMIAL REGRESSION:**

IT IS USED WHEN THE RELATION $Y = f(\vec{X})$ IS NON-LINEAR:

$$Y = \beta_0 + \beta_1 \vec{X} + \beta_2 \vec{X}^2 + \dots + \beta_K \vec{X}^K + \varepsilon$$

- IT CONSISTS OF:
  - COMPUTING NEW FEATURES AS A POWER FUNCTION OF INPUT FEATURES
  - APPLY LINEAR REGRESSION ON THESE NEW FEATURES

ex. $n=2$ -> $\vec{X} = (X_1, X_2)$:

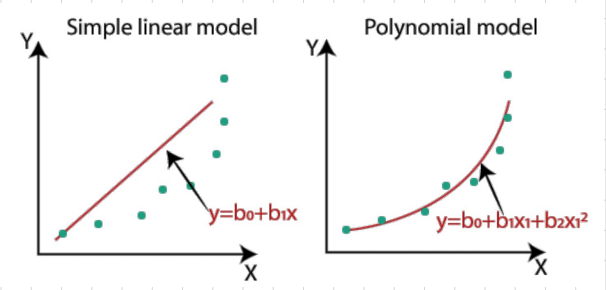
$$Y = \beta + \beta_1 \vec{X} + \beta_2 \vec{X}^2 + \varepsilon \longrightarrow Y = \beta_0 + \underbrace{\beta_1 X_1 + \beta_2 X_2}_{\beta_1 \vec{X}} + \underbrace{\beta_{11} X_1^2 + \beta_{22} X_2^2 + \beta_{12} X_1 X_2}_{\beta_2 \vec{X}^2} + \varepsilon$$

- THE MODEL CAN BE WRITTEN AS: $Y = X \cdot \beta + \varepsilon$

$X = [X^1, \dots, X^K]^T$

$\beta = [\beta_1, \dots, \beta_K]^T$

ex.



- CONSIDERATIONS :
  - KEEP THE ORDER OF THE POLYNOMIAL AS LOW AS POSSIBLE
    ~> USUALLY MAX ORDER = 2
  - HIGHER ORDER CAN LEAD TO OVERFITTING
  - TRY MODEL FITTING FOR INCREASING ORDER UNTIL MORE ORDERS IS NON SIGNIFICANT

V : POLYNOMIAL FIT WIDE RANGE OF CURVATURE

USUALLY MAY ARE THE BEST APPROXIMATION

S : TOO SENSITIVE TO OUTLIERS

HIGH DEGREE ——> POSSIBLY DATA OVERFITTING

↳ AVOID DATA OVERFITTING :
- USE A LARGE TRAINING DATA
- USE LOWER MODEL COMPLEXITY
- USE REGULARIZATION TECHNIQUES

- **REGULARIZATION TECHNIQUES** -

THEY PERFORM VARIABLE SELECTION AND REGULARIZATION TO ENHANCE ACCURACY AND INTERPRETABILITY

-> REDUCTION IN MODEL COMPLEXITY AND OVERFITTING WHEN: (# VARIABLES > # OBSERVATIONS TO BE DESCRIBED) OR (LEARNED MODEL IS POORLY GENERABLE)

-> THEY PERFORM CHANGES ON β_i

- NORMAL LINEAR REGRESSION : $RSS = \sum_i (Y_i - \hat{Y}_i)^2 = \sum_i \left(Y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij}\right)^2$

- **RIDGE REGRESSION** : $RSS + \lambda \sum_{j=1}^{p} \beta_j^2$

IT ADDS $L2$ (~> NORM 2) AS PENALTY : $\sum \beta_j^2$

-> IT'S LIKE RSS, UNDER THE CONDITION $\sum_{j=1}^{p} \beta_j^2 < C$, FOR $C > 0$

- Ridge tends to lower uniformly all the coefficients
  - Coefficients already close to 0 have little effect on the sum of **squares** (if $x \approx 0$, $x^2 < x$)

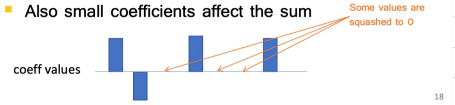
L FACT ABSOLUTE SHRINKAGE AND SELECTION OPERATOR

- **LASSO REGRESSION** : $RSS + \lambda \sum_{j=1}^{p} |\beta_j|$

IT ADDS $L1$ AS PENALTY : $\sum |\beta_j|$

-> IT APPLIES THE CONSTRAINT $\sum_{j=1}^{p} |\beta_j| < C$, FOR $C > 0$

- Lasso tends to assign the value 0 to **some** coefficients (feature selection)
  - Also small coefficients affect the sum



λ : AMOUNT OF SHRINKAGE

- **RIDGE VS LASSO** :

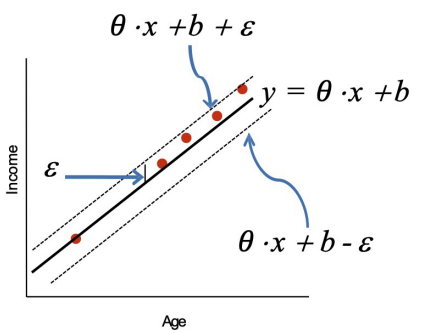
- LASSO IDENTIFIES A SIMPLER MODEL : LESS $\beta_j \neq 0$ COEFFICIENT

- { RIDGE : SHRINK SIZE OF SOME β_j
  { LASSO : SETS SOME $\beta_j := 0$

- **SUPPORT VECTOR REGRESSION:**

DEFINE HOW MUCH ERROR IS ACCEPTABLE IN MODEL

-> IT FINDS AN APPROPRIATE HYPER PLANE TO FIT DATA



- TRAINING PROBLEM IS A CONVEX OPTIMIZATION PROBLEM:

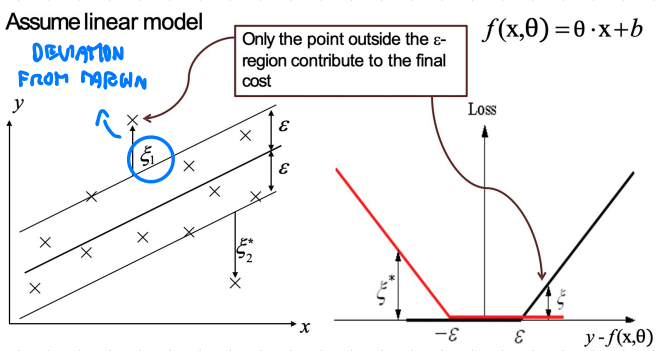
$$\min \left\{ \frac{1}{2} \|\theta\|^2 \right\}, \quad \text{CONSTRAINTS: } \begin{cases} y^i - \theta x^i - b \leq \varepsilon \\ \theta x^i + b - y^i \leq \varepsilon \end{cases}$$

- **SOFT MARGIN:**

SOMETIMES, GIVEN AN ε

-> PROBLEM NOT FEASIBLE

-> REFORMULATE PROBLEM CONSIDERING ERRORS / DON'T SATISFY ε



-> OPTIMIZATION PROBLEM:

$$\min \left\{ \frac{1}{2} \|\theta\|^2 \right\} + C \sum_{i=1}^{m} (\xi_i + \xi_j^*)$$

$$\text{CONSTRAINTS: } \begin{cases} y^i - \theta x^i - b \leq \varepsilon + \xi_i \\ \theta x^i + b - y^i \leq \varepsilon + \xi_i^* \end{cases}, \quad \xi_i, \xi_i^* > 0$$

-> WE MINIMIZE DEVIATION ξ_i FROM THE MARGIN

- **NON-LINEAR CASE:**

- USE A MAPPING FOR AN HIGHER ORDER DIMENSIONAL SPACE:

KERNEL TRANSFORMATION (POLYNOMIAL, GAUSSIAN RADIAL, ...)

$$x_i \longmapsto \phi(x_i)$$

$$\min \frac{1}{2} \|\theta\|^2 + C \sum_{i=1}^{m} (\xi_i + \xi_i^*)$$

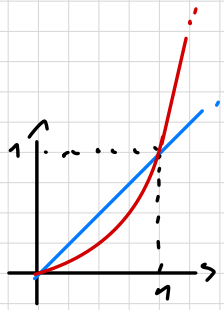
$$s.t. \quad y^i - \theta \cdot \phi(x^i) - b \leq \varepsilon + \xi_i;$$
$$\theta \cdot \phi(x^i) + b - y^i \leq \varepsilon + \xi_i$$
$$\xi_i, \xi_i^* \geq 0, i = 1, \ldots, m$$

- **EVALUATING REGRESSION:**

  - **MEAN ABSOLUTE ERROR:** $MAE = \frac{1}{n} \sum_i |Y_i - \hat{Y}_i|$

  - **MEAN SQUARED ERROR:** $MSE = \frac{1}{n} \sum_i (Y_i - \hat{Y}_i)^2$

    ↳ IT PENALIZES LESS ERRORS CLOSER TO 0



→ MAE, MSE > 0 , ↓ MAE, MSE → BETTER MODEL

  - **RESIDUAL STANDARD ERROR:** $RSE = \sqrt{\frac{1}{n-2} \underbrace{\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2}_{RSS}}$ / n: # SAMPLES

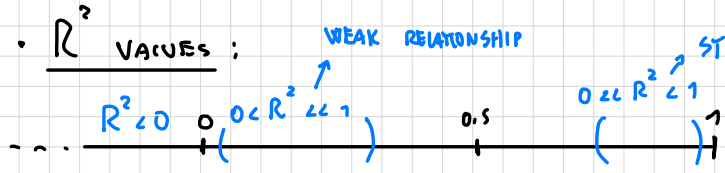
  - **R^2 - SQUARED:**

    IT MEASURES THE GOODNESS OF FIT OF A MODEL

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \underbrace{\left[\frac{MSE}{\sigma^2}\right]}_{FVU} \quad / \quad TSS = \sigma^2 = \sum_i (Y_i - \bar{Y}_i)^2$$

    - FVU: RATIO BETWEEN UNEXPLAINED VARIANCE AND TOTAL VARIANCE

  - **R^2 VALUES:**



    - $R^2 = 1$: PERFECT LINEAR RELATIONSHIP BETWEEN X AND Y
    - $R^2 = 0$: NO LINEAR RELATIONSHIP BETWEEN X AND Y
    - $R^2 < 0$: MODEL IS PREDICTING WORSE THAN $\hat{Y} = \bar{Y}$

  - **R^2 ADJUSTED:**

    IF MULTIPLE LINEAR REGRESSION → ↑ # X_i → ↓ R^2 NEVER (REMAIN CONSTANT OR INCREASES ONLY)

    → USING THE ADJUSTED $\bar{R}^2$: ↑ $\bar{R}^2$ ⟺ NEW X_i ADDED IMPROVE THE MODEL

$$\bar{R}^2 = 1 - \frac{n-1}{n-p-1} \underbrace{\left(1 - R^2\right)}_{\frac{RSS}{TSS}}$$

n: # SAMPLES
p: # X_i

- **TIME SERIES ANALYSIS :**

TIME SERIES : SEQUENTIAL SET OF DATA POINTS MEASURED OVER SUCCESSIVE TIMES → $S = f(t)$

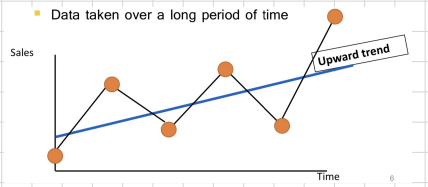
- **COMPONENTS :**

  - **TREND COMPONENT :**

    LONG-TERM MOVEMENT, INCREASING / DECREASING OVER t

    UPWARD OR DOWNWARD MOVEMENT

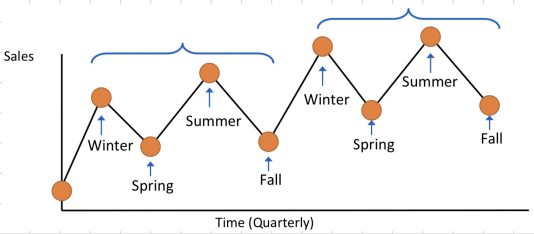
    CAN BE LINEAR OR NON-LINEAR



  - **SEASONAL COMPONENT :**

    REGULAR PERIODIC FLUCTUATIONS

    REGULAR WAVE-LIKE PATTERN

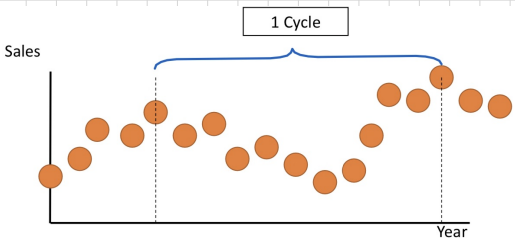


  - **CYCLICAL COMPONENT :**

    REPEATING SWINGS OR MOVEMENTS

    REGULARLY OCCUR BUT MAY VARY IN LENGTH

    MEASURED PEAK TO PEAK

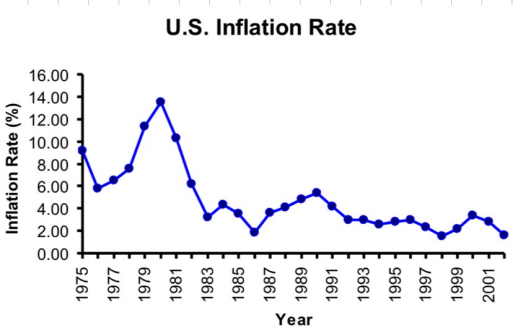


  - **IRREGULAR / RANDOM COMPONENTS :**

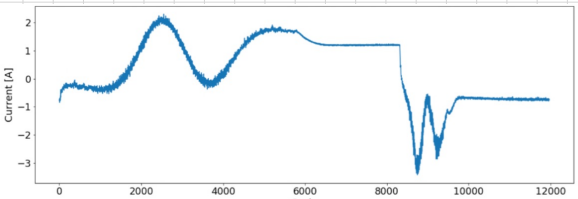
    CAUSED BY UNPREDICTABLE INFLUENCES

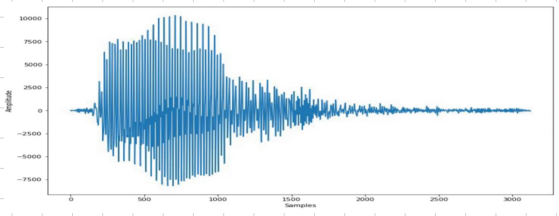
    → THIS USUALLY REPRESENTS NOISE IN TIME SERIES

DISCRETE TIME SERIES

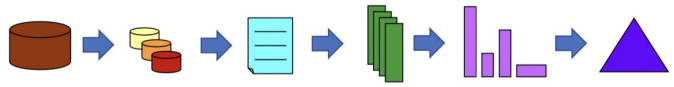


CONTINOUS TIME SERIES





- KDD . KNOWLEDGE DISCOVERY IN DB



PRE-PROCESSING
- TIME SERIES ALIGNMENT TECHNIQUES MIGHT BE REQUIRED $S_1, \dots, S_n = f(t)$
- IN CASE OF MULTIVARIATE PROBLEMS → CORRELATED TIME SERIES SHOULD BE REMOVED

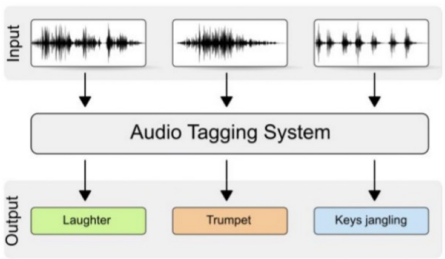
TRANSFORMATION
- FEATURE ENGINEERING, FEATURE EMBEDDING

- ANALYTICS TASKS:

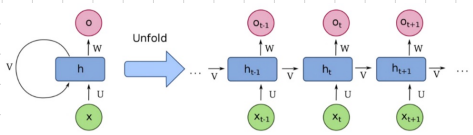
CLASSIFICATION AND FORECASTING

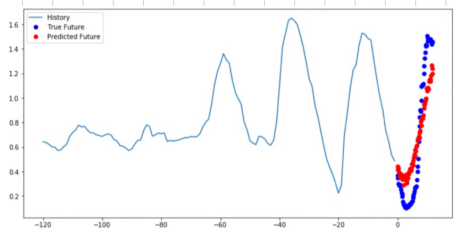
- ARTIFICIAL NEURAL NETWORKS CAN BE USED
  - CONVOLUTIONAL
    - CNN: CLASSIFICATION TASK
    - RNN: FORECASTING TASK → RE CURRENT

  - CNN:
    - VGGISH IS A CNN TO EXTRACT FEATURES FROM AUDIO SIGNALS
    - log MEL SPECTOGRAM AUDIO ⟶ AUDIO TAG



  - RNN:
    CONNECTION BETWEEN NODES FORM A DIRECTED GRAPH ALONG A TEMPORAL SEQUENCE
    → RNN CAN USE THEIR INTERNAL STATE MEMORY TO PROCESS SEQUENCES OF INPUT





V: POWERFUL ALGORITHMS TO TRAIN ACCURATE MODELS ABLE TO DEAL WITH DIFFERENT COMPLEX ANALYTICS TASKS

S: HUGE AMOUNT OF DATA REQUIRED
TRAINING IS AN HEAVY TASK COMPUTATIONALLY, IN TIME AND HARDWARE
FEATURE LEARNING STEP HIDDEN IN THE NETWORK

- **FEATURE ENGINEERING :**

FEATURE COMPUTATION OVER A TIME SERIES :

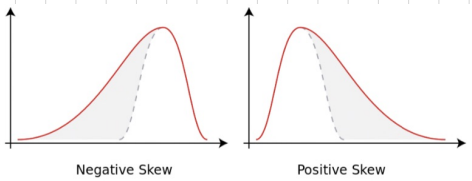
- BASICS STATISTICS (MIN, MAX, ecc.), INDICES (KURTOSIS, SKEWNESS), , SUMMARIZATION (PERCENTILE, CDF + PERCENTILE, DERIVATE + CDF + PERCENTILE), LINEAR REGRESSION

- **SKEWNESS :**

MEASURE OF ASYMMETRY OF PROB. DISTRIBUTION OF A REAL VALUED RV ABOUT ITS MEAN

$$\gamma_1 = E\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] = \frac{M_3}{\sigma^3} = \frac{K_3}{K_2^{3/2}}$$

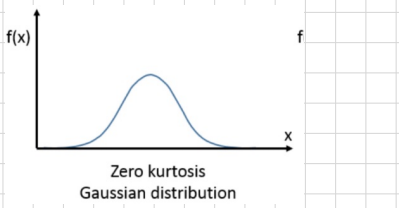
K_t : t-TH CUMULANTS



- **KURTOSIS :**

MEASURE OF "TAILEDNESS" OF PROB. DISTRIBUTION OF A REAL VALUED RV

$$KURT[X] = E\left[\left(\frac{X-\mu}{\sigma}\right)^4\right] = \frac{M_4}{\sigma^4}$$

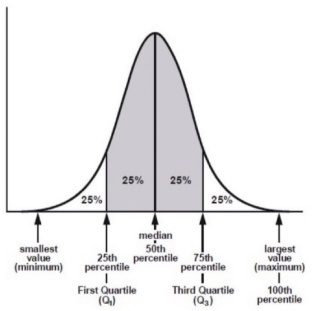


- **SUMMARIZATION :**

- **PERCENTILE**: VALUE BELOW WHICH A GIVEN % OF OBSERVATIONS IN A GROUP OF OBSERVATIONS FALL

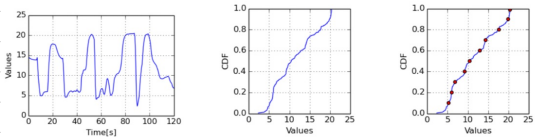
-> OUTLIERS CAN BE REMOVED, REMOVING LAST PERCENTILES OF THE DISTRIBUTION

- PERCENTILES ARE FEATURE DESCRIBING THE TIME SERIES

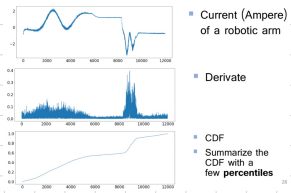


- **CDF :** $F_X(x) = P(X \leq x)$

$$\rightarrow P(a < X < b) = F_X(b) - F_X(a)$$



- **LINEAR REGRESSION :**



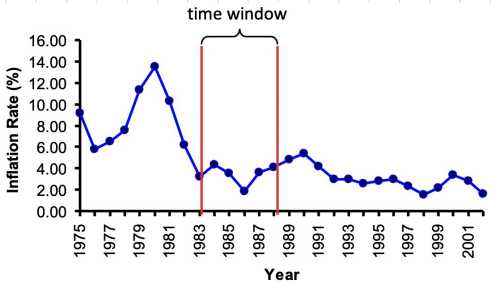
- FEATURE ENGINEERING CAN ALSO BE COMPUTED IN LOCAL PARTS OF ENTIRE TIME SERIES, ex. TIME WINDOW

- **TIME WINDOW:**

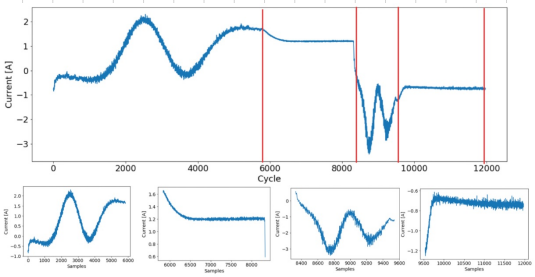
  TIME SERIES PORTION ON WHICH COMPUTE FEATURES

  - **WINDOW LENGTH:** SIZE IN t UNITS
    - DOMAIN-DRIVEN / DATA-DRIVEN

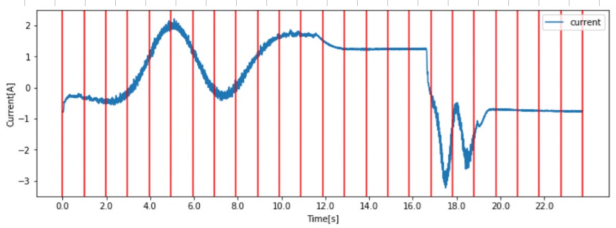
  - **WINDOW SHIFT:** POSITION IN RESPECT TO CONSECUTIVE WINDOWS
    - WINDOWS CAN BE NON-OVERLAPPED / OVERLAPPED



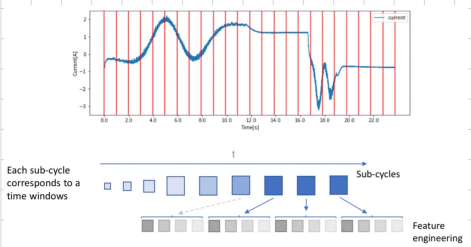
DOMAIN-DRIVEN



DATA-DRIVEN

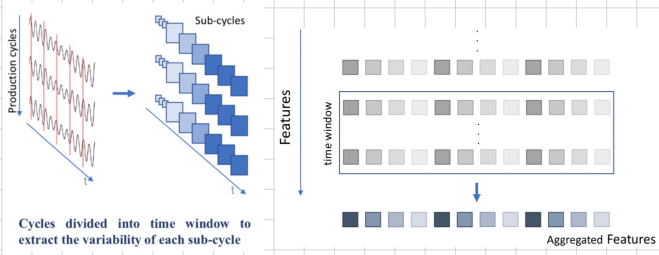


- TIME SERIES TREND CAN BE CAPTURED THROUGH THE FEATURES EXTRACTED FROM EACH SUB-CYCLE OF EACH TIME SERIES



- TO DEAL WITH LONG HORIZON PREDICTION
  → AGGREGATION OVER A TIME WINDOW

- IN SOME CASES MORE FEATURES CAN BE CORRELATED
  → REMOVE CORRELATED FEATURES



- **TIME SERIES IN FREQUENCY DOMAIN:**