

Data Mining

Classification: Alternative Techniques

Lecture Notes for Chapter 5

Classification Problem

- Πρόβλημα μάθησης με επίβλεψη (**Supervised learning**)
- Δεδομένα του συνόλου εκπαίδευσης $X = \{(x_n, y_n)\}_{n=1}^N$ αποτελούμενα από ζεύγη σημείων σε ένα χώρο ($x_n \in R$) και ετικέτας ή πρόβλεψης y_n .
- Στόχος να φτιάξουμε μία συνάρτηση $f(x)$ που να προβλέπει την κατηγορία y των σημείων.

Rule-based Classifiers

- Classify records by using a collection of “**if...then...**” rules
- **Rule set** : $R = \{r_1 \cup r_2 \cup \dots \cup r_k\}$, όπου r_i ένας κανόνας ταξινόμησης
- **Rule** r_i : *if* (condition)_{*i*} **then** y_i
 - *Condition* είναι οι προϋποθέσεις ή το **αίτιο**
 - y είναι το αποτέλεσμα του κανόνα (**class label**)
- Condition = $(x_1 \text{ **op.** } U_1) \wedge (x_2 \text{ **op.** } U_2) \wedge \dots \wedge (x_k \text{ **op.** } U_k)$
 - όπου **op.** είναι λογικοί τελεστές (*logical operators*)

Application of Rule-Based Classifier

- Ένας κανόνας r **καλύπτει (covers)** μια εγγραφή x αν το αίτιο (συνθήκες) του r ταιριάζει με τα χαρακτηριστικά του.
- Τότε λέμε ότι ο κανόνας ενεργοποιείται ή πυροδοτείται (***fired***)

π.χ.

R1: (Give Birth = no) $\wedge$ (Can Fly = yes) $\rightarrow$ Birds

R2: (Give Birth = no) $\wedge$ (Live in Water = yes) $\rightarrow$ Fishes

R3: (Give Birth = yes) $\wedge$ (Blood Type = warm) $\rightarrow$ Mammals

R4: (Give Birth = no) $\wedge$ (Can Fly = no) $\rightarrow$ Reptiles

R5: (Live in Water = sometimes) $\rightarrow$ Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
hawk	warm	no	yes	no	?
grizzly bear	warm	yes	no	no	?

The rule R1 covers a hawk $\Rightarrow$ Bird

The rule R3 covers the grizzly bear $\Rightarrow$ Mammal

Rule Coverage and Accuracy

- **Coverage** of a rule r :

- Το ποσοστό των εγγραφών του συνόλου που καλύπτονται ή πυροδοτούνται από τον r .

- **Accuracy** of a rule:

- Το ποσοστό των εγγραφών που πυροδοτούν τον κανόνα r και είναι της ίδιας κατηγορίας με το $y(r)$.

<i>Tid</i>	Refund	Marital Status	Taxable Income	Class
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

(Status=Single) → No

Coverage = 40%, Accuracy = 50%

Rule Coverage and Accuracy

- **Coverage** of a rule r :

$$\text{Coverage}(r) = \frac{|A|}{|D|}$$

$|D|$: size of input dataset

$|A|$: # points cover r

- **Accuracy** of a rule:

$$\text{Accuracy}(r) = \frac{|A \cap y|}{|A|}$$

$|A \cap y|$: # points cover r
and satisfy class y

How does Rule-based Classifier Work?

Rule Set:

r1: (Give Birth = no) $\wedge$ (Can Fly = yes) $\rightarrow$ Birds

r2: (Give Birth = no) $\wedge$ (Live in Water = yes) $\rightarrow$ Fishes

r3: (Give Birth = yes) $\wedge$ (Blood Type = warm) $\rightarrow$ Mammals

r4: (Give Birth = no) $\wedge$ (Can Fly = no) $\rightarrow$ Reptiles

r5: (Live in Water = sometimes) $\rightarrow$ Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
lemur	warm	yes	no	no	?
turtle	cold	no	no	sometimes	?
dogfish shark	cold	yes	no	yes	?

A lemur triggers rule **r3**, so it is classified as a mammal

A turtle triggers both **r4** and **r5**

A dogfish shark triggers **none** of the rules

Characteristics of Rule-Based Classifier

- **Mutually exclusive rules (αλληλοαποκλειόμενοι)**
 - Classifier contains mutually exclusive rules if the rules are **independent of each other**
 - Δηλ. κάθε εγγραφή καλύπτεται από **το πολύ έναν** κανόνα.
- **Exhaustive rules (εξαντλητικοί)**
 - Ένα σύνολο κανόνων έχει εξαντλητική κάλυψη αν $\exists$ ένας κανόνας για κάθε δυνατό συνδυασμό χαρακτηριστικών.
 - Δηλ. κάθε εγγραφή καλύπτεται από **τουλάχιστον έναν** κανόνα.

Απλοποίηση κανόνων (Rule Simplification)

- Rules are no longer mutually exclusive
 - A record may trigger more than one rule
 - Solution?
 - ◆ Ordered rule set
 - ◆ Unordered rule set – use voting schemes
- Rules are no longer exhaustive
 - A record may not trigger any rules
 - Solution?
 - ◆ Use a default class

- **Ordered Rules:** Διατάσσουμε τους κανόνες που καλύπτουν μια εγγραφή με βάση ένα μέτρο αξιολόγησης (δίνοντας προτεραιότητα), π.χ.
 - Coverage
 - Accuracy
 - Total description length
 - ...
- **Απόφαση ταξινόμησης** για ένα άγνωστο σημείο:
 - Ταξινόμηση στη κατηγορία του κανόνα με το μεγαλύτερο σκορ μεταξύ αυτών που πυροδοτούνται. (ordered rules)
 - Ταξινόμηση στην πλειοψηφούσα κατηγορία μεταξύ των κανόνων που πυροδοτούνται ή χρήση κάποιου **voting scheme**.
 - Αν δεν υπάρχει κανόνας που ταιριάζει, τότε ταξινόμηση στην default class.

Κατασκευή κανόνων ταξινόμησης

- **Direct Methods** for Rule Extraction:
 - ◆ Extract rules directly from data
 - ◆ e.g.: RIPPER, CN2, Holte's 1R
- **Indirect Methods** for Rule Extraction:
 - ◆ Extract rules from other classification models (e.g. decision trees, neural networks, etc).
 - ◆ e.g: C4.5rules

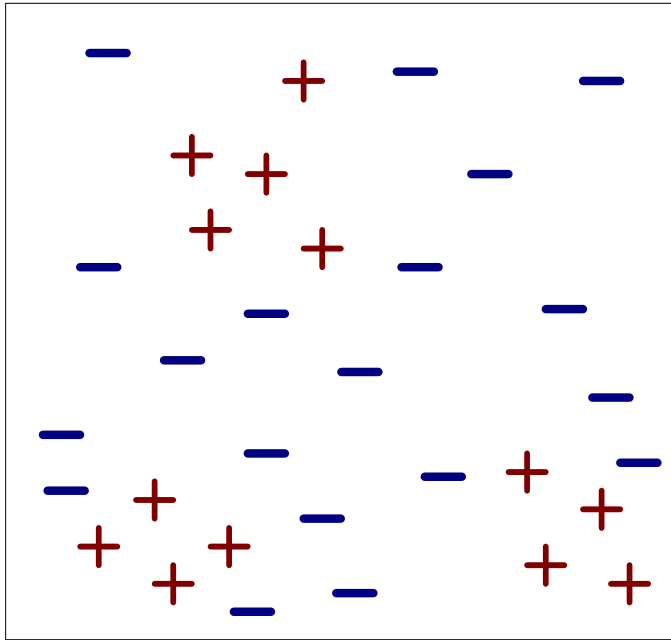
[1]. Direct Method: Sequential Covering

- Extract directly from data
- **Greedy** method: Ο αλγόριθμος παράγει κανόνες («λαίμαργα») για κάθε κατηγορία επαναληπτικά χρησιμοποιώντας την διαδικασία **Learn-One-Rule**
- Επιλογή κατηγορίας χρησιμοποιώντας κριτήρια όπως:
 - Ελάχιστη συχνότητα της κατηγορίας στο σύνολο εκπαίδευσης
 - Κόστος εσφαλμένης ταξινόμησης

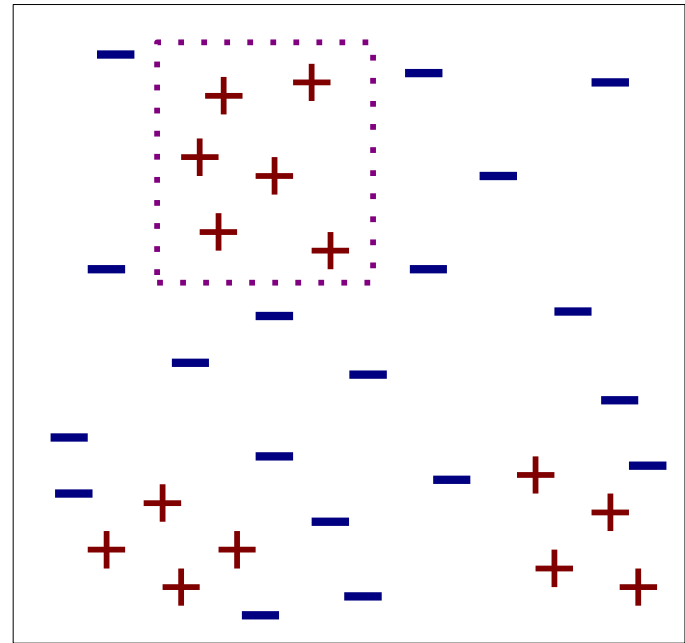
Learn-One-Rule procedure:

- Κατασκευή του βέλτιστου αριθμός κανόνων για την κατηγορία y που καλύπτει το τρέχον σύνολο εκπαίδευσης.
- Ένας κανόνας θεωρείται «αποδεκτός» αν καλύπτει τα περισσότερα δείγματα της ίδιας κατηγορίας (positive examples) του συνόλου εκπαίδευσης και όσο το δυνατόν λιγότερο από υπόλοιπα δείγματα που δεν ανήκουν στην ίδια κατηγορία (negative examples).

Example of Sequential Covering

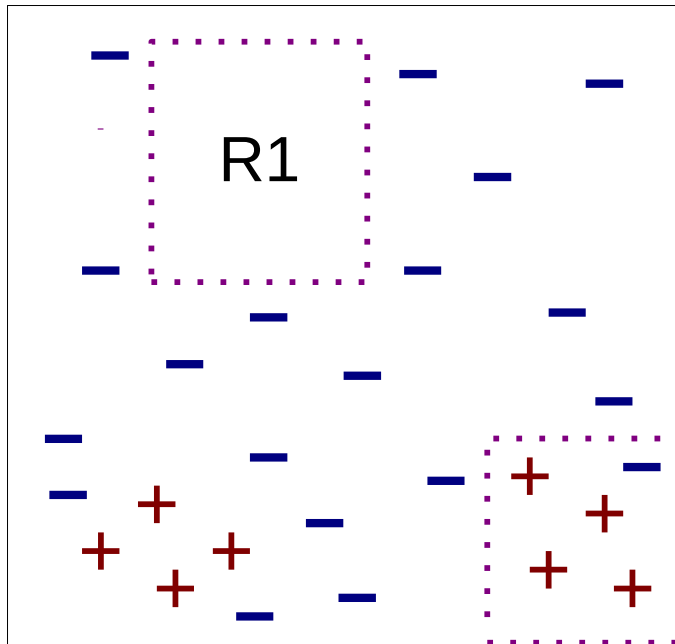


(i) Original Data

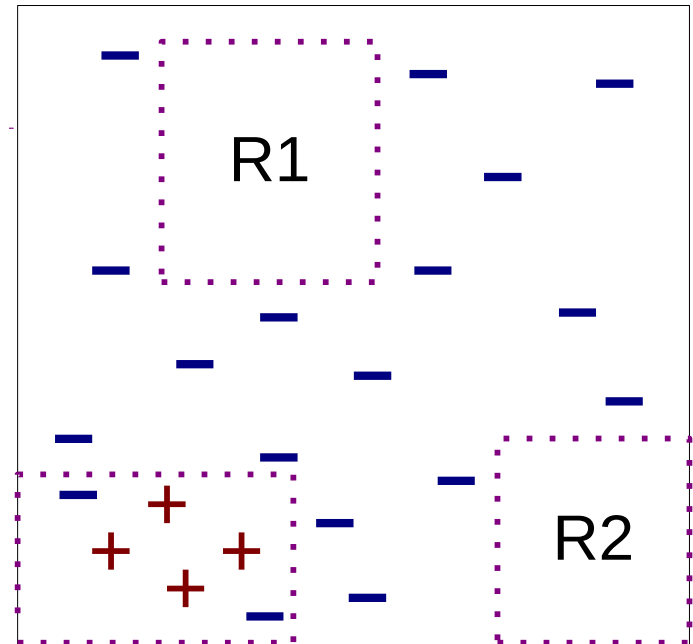


(ii) Step 1

Example of Sequential Covering...



(iii) Step 2



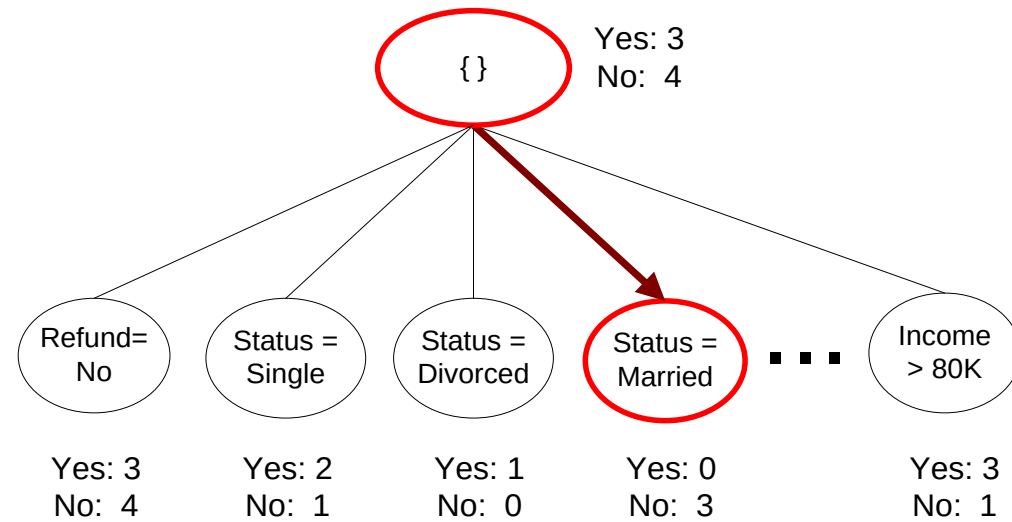
(iv) Step 3

Rule Growing Strategy

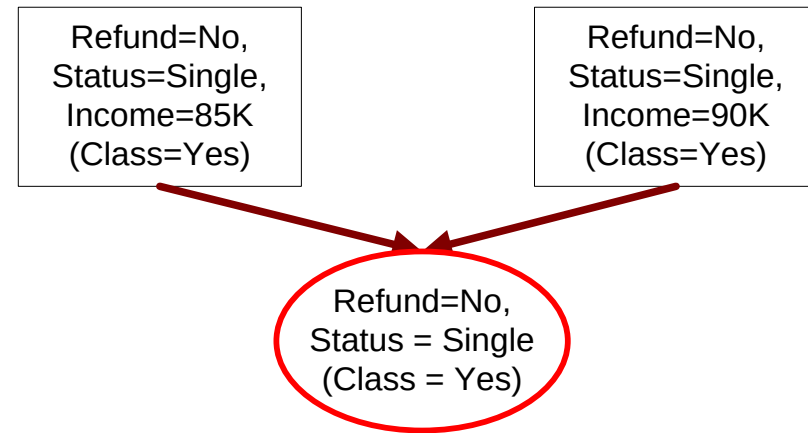
- Δημιουργούμε αρχικά έναν κανόνα και σταδιακά τον αυξάνουμε έως ότου ισχύει κάποιο κριτήριο.
- 2 στρατηγικές
 - **General-to-specific** (top-down)
 - ◆ Αρχικά κανόνας με κενό condition $r: \{ \} \rightarrow y$ (general rule)
 - ◆ Επαναληπτικά προσθέτουμε τιμές χαρακτηριστικών μέχρι να μην βελτιώνεται ποιοτικά ο κανόνας (δηλ. πιο specific)
 - **Specific-to-general** (bottom-up)
 - ◆ Αρχικά επιλέγουμε τυχαία ένα «θετικό» δείγμα φτιάχνοντας έναν ειδικό κανόνα που το καλύπτει.
 - ◆ Επαναληπτικά διαγράφουμε ένα χαρακτηριστικό για να καλύπτει περισσότερα «θετικά» δείγματα (δηλ. πιο γενικός). Τερματισμός όταν καλύπτει και «αρνητικά» δείγματα

Rule Growing Strategy (συν.)

- παράδειγματα



(a) General-to-specific



(b) Specific-to-general

Rule evaluation metrics (Μέτρα αξιολόγησης κανόνων)

- Metrics:

- Accuracy = $\frac{n_c}{n}$

Ποσοστό σωστά ταξινομημένων δειγμάτων από τον κανόνα. **Μειονέκτημα:** δεν λαμβάνει υπόψιν του την κάλυψη.

- Laplace = $\frac{n_c + 1}{n + k}$

Η Laplace είναι η M-estimate για $p=1/k$

Αν δεν καλύπτει κανένα δείγμα, τότε

$$\text{Laplace} = 1/k$$

$$\text{M-estimate} = p$$

Αν μεγάλη κάλυψη ($n_c \gg$), τότε

$$\text{Laplace} = \text{M-estimate} \approx n_c / n$$

- M-estimate = $\frac{n_c + kp}{n + k}$

n : Number of instances of covered by rule

n_c : Number of positive instances

k : Number of classes

p : Prior probability of positive class

— FOIL κέρδος πληροφορίας (information gain)

- Έστω f_0^+ , f_0^- οι συχνότητες κάλυψης των θετικών και αρνητικών δειγμάτων αντίστοιχα
- Μεταβάλλοντας το αίτιο του κανόνα αλλοιώνονται οι συχνότητες, έστω f_1^+ , f_1^-
- Τότε, FOIL information gain = $f_1^+ \log_2 \frac{\left(\frac{f_1^+}{f_1^+ + f_1^-} \right)}{\left(\frac{f_0^+}{f_0^+ + f_0^-} \right)}$
- Προτιμούμε κανόνες με μεγάλο *support* αλλά και *accuracy*

Ο Αλγόριθμος κατασκευής κανόνων RIPPER

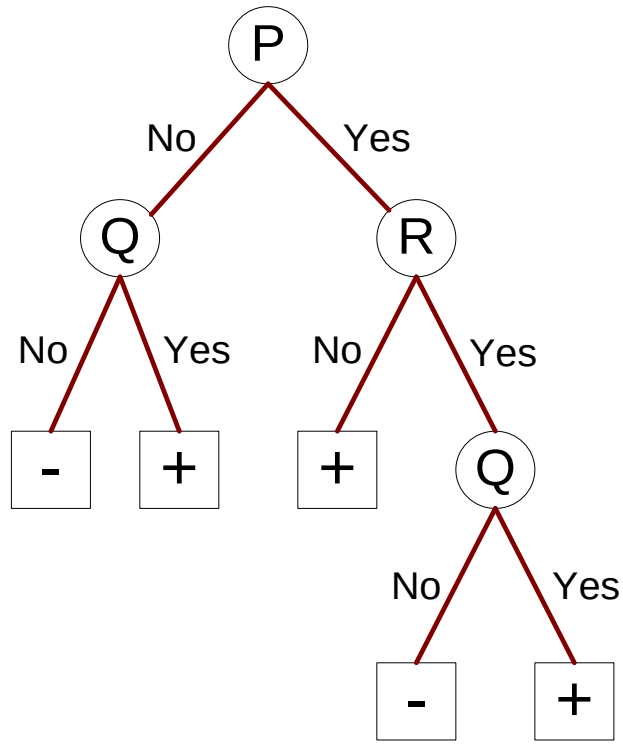
- Γραμμικός ως προς τον αριθμό των δειγμάτων εκπαίδευσης.
- Αρχικά διατάσσουμε τις κατηγορίες κατά αύξουσα ανάλογα με την συχνότητά τους στο σύνολο εκπαίδευσης. *** **γιατί ;;;**
- Για κάθε μία από τις υπάρχουσες κατηγορίες:
 - Θεωρούμε τα δείγματα της κατηγορίας «θετικά» και των υπολοίπων «αρνητικά».
 - Εφαρμόζουμε την μέθοδο *sequential covering* για να κατασκευάσουμε κανόνες που χωρίζουν θετικά από αρνητικά παραδείγματα.

Ο Αλγόριθμος RIPPER (συν.)

- Rule Growing:

- Ο RIPPER εφαρμόζει την στρατηγική general-to-specific για την δημιουργία ενός κανόνα, ενώ ως μέτρο αξιολόγησης το FOIL information gain.
- Stop when rule no longer covers negative examples
- Υπάρχει η δυνατότητα για pruning του κανόνα διαγράφοντας χαρακτηριστικά.
- Αφού παραχθεί ο κανόνας, τότε διαγράφουμε όλα τα δείγματα (θετικά ή αρνητικά) που καλύπτονται από τον κανόνα και τοποθετείται στην λίστα των εξαγόμενων κανόνων.

[2]. Indirect Methods κατασκευής κανόνων

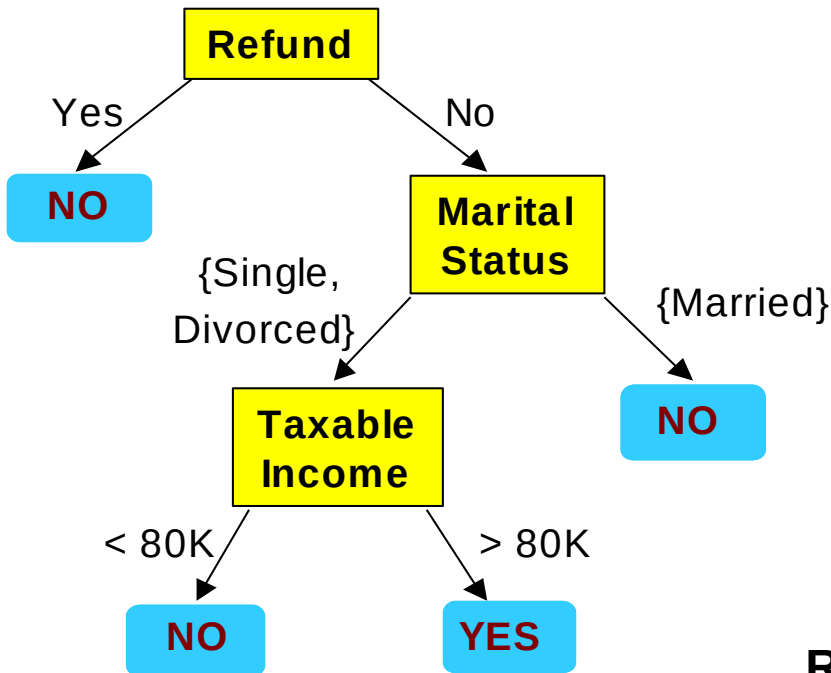


Rule Set

r1: (P=No,Q=No) ==> -
r2: (P=No,Q=Yes) ==> +
r3: (P=Yes,R=No) ==> +
r4: (P=Yes,R=Yes,Q=No) ==> -
r5: (P=Yes,R=Yes,Q=Yes) ==> +

- Κατασκευή κανόνων από δέντρα απόφασης (decision trees)
 - Κάθε μονοπάτι του δέντρου από τη ρίζα έως τα φύλλα εκφράζει ένα κανόνα ταξινόμησης
 - Οι συνθήκες είναι πάνω στους κόμβους (χαρακτηριστικά) και ακμές (τιμές), ενώ η κατηγορία βρίσκεται στο φύλλο του δέντρου.

From Decision Trees To Rules



Classification Rules

(Refund=Yes) ==> No

(Refund=No, Marital Status={Single, Divorced}, Taxable Income<80K) ==> No

(Refund=No, Marital Status={Single, Divorced}, Taxable Income>80K) ==> Yes

(Refund=No, Marital Status={Married}) ==> No

Rules are mutually exclusive and exhaustive

Rule set contains as much information as the tree

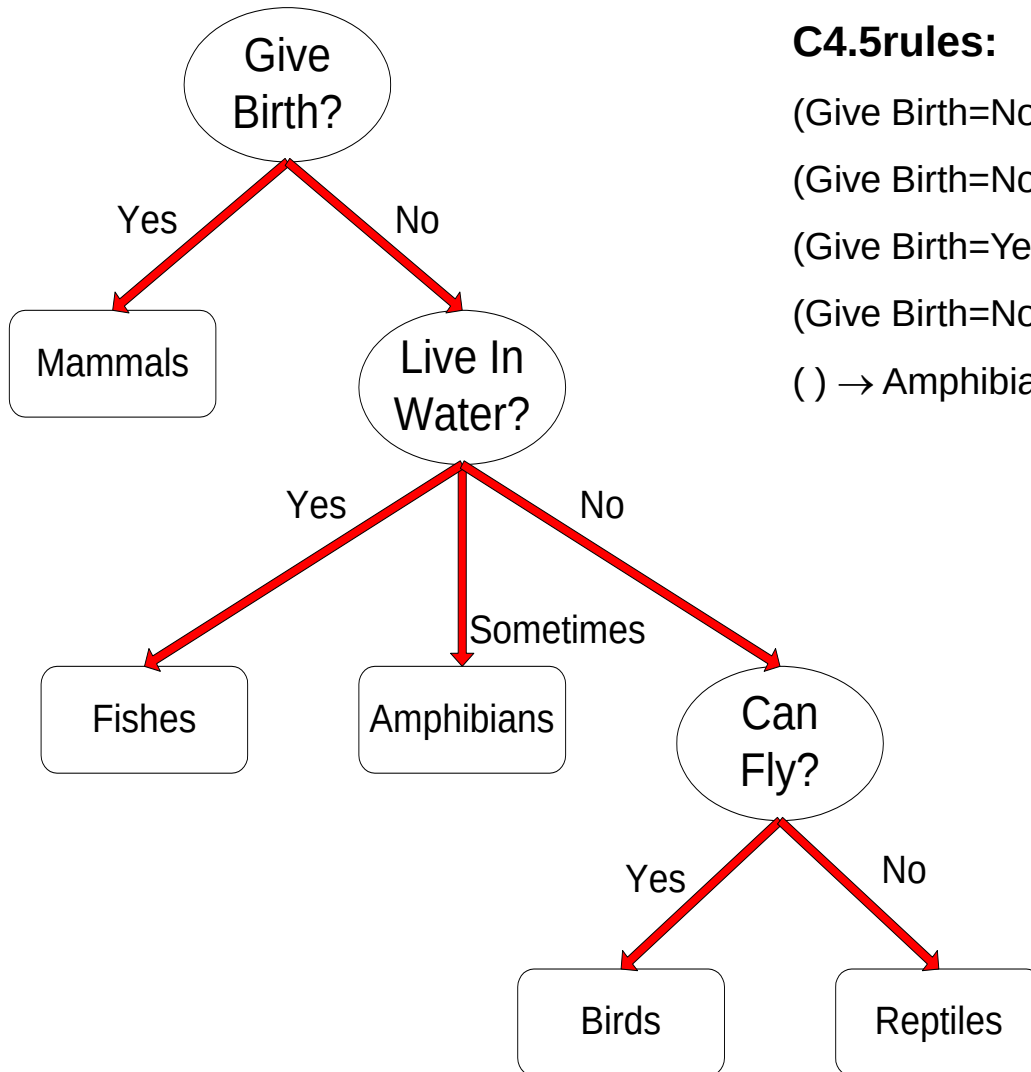
Indirect Method: C4.5rules

- Συχνά οι κανόνες που παράγονται είναι πολύπλοκοι χωρίς ερμηνευτική ή γενικευτική ικανότητα.
- Απαιτείται διαδικασία απλοποίησης (simplification)
- Pruning:
 - Δοκιμάζουμε να βγάζουμε χαρακτηριστικά από τις συνθήκες του κανόνα,
 - Ελέγχουμε αν βελτιώνεται η απόδοση του κανόνα, π.χ. ελαττώνεται το σφάλμα ταξινόμησης
 - Κρατάμε την μορφή του κανόνα με την βέλτιστη απόδοση.
 - Παράλληλα, μετά το pruning, ελέγχουμε για τυχόν αντίγραφα μεταξύ των κανόνων του συνόλου.

Example

Name	Give Birth	Lay Eggs	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	no	yes	mammals
python	no	yes	no	no	no	reptiles
salmon	no	yes	no	yes	no	fishes
whale	yes	no	no	yes	no	mammals
frog	no	yes	no	sometimes	yes	amphibians
komodo	no	yes	no	no	yes	reptiles
bat	yes	no	yes	no	yes	mammals
pigeon	no	yes	yes	no	yes	birds
cat	yes	no	no	no	yes	mammals
leopard shark	yes	no	no	yes	no	fishes
turtle	no	yes	no	sometimes	yes	reptiles
penguin	no	yes	no	sometimes	yes	birds
porcupine	yes	no	no	no	yes	mammals
eel	no	yes	no	yes	no	fishes
salamander	no	yes	no	sometimes	yes	amphibians
gila monster	no	yes	no	no	yes	reptiles
platypus	no	yes	no	no	yes	mammals
owl	no	yes	yes	no	yes	birds
dolphin	yes	no	no	yes	no	mammals
eagle	no	yes	yes	no	yes	birds

C4.5 versus C4.5rules versus RIPPER



C4.5rules:

(Give Birth=No, Can Fly=Yes) → Birds

(Give Birth=No, Live in Water=Yes) → Fishes

(Give Birth=Yes) → Mammals

(Give Birth=No, Can Fly=No, Live in Water=No) → Reptiles

() → Amphibians

RIPPER:

(Live in Water=Yes) → Fishes

(Have Legs=No) → Reptiles

(Give Birth=No, Can Fly=No, Live In Water=No) → Reptiles

(Can Fly=Yes, Give Birth=No) → Birds

() → Mammals

Advantages of Rule-Based Classifiers

- Προσφέρουν ερμηνευτική ικανότητα.
- Ευκολία στην κατασκευή.
- Γρήγορη απόφαση σε άγνωστα δείγματα.
- Απόδοση συγκρίσιμη με αυτή των δέντρων αποφάσεων.
- Επιτρέπουν την ανισορροπία κατηγορίας μεταξύ των δειγμάτων

Nearest Neighbor Classifiers (Ταξινομητές του κοντινότερου γείτονα)

- Η απλούστερη μέθοδος ταξινόμησης.

- Έστω σύνολο εκπαίδευσης

$$X = \left\{ (x_n, y_n), x_n \in R^d, y_n \in \{1, 2, \dots, K\} \right\}_{n=1}^N$$

- **Κανόνας ταξινόμησης:**

«Ένα άγνωστης κατηγορίας σημείο x^* ταξινομείται στην κατηγορία του κοντινότερου γείτονα»

Nearest Neighbor Classifier – 1NN

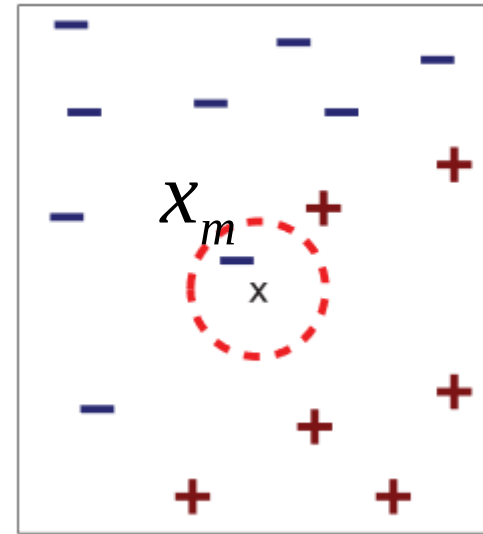
Διαδικασία ταξινόμησης

1. Βρίσκουμε τον κοντινότερο γείτονα (ή περισσότερο όμοιο)

$$x_m : \min_{n=1,\dots,N} \{d(x^*, x_n)\}$$

2. Η κατηγορία του x^* είναι η ίδια με αυτή του κοντινότερου γείτονα

$$y^* = y_m$$

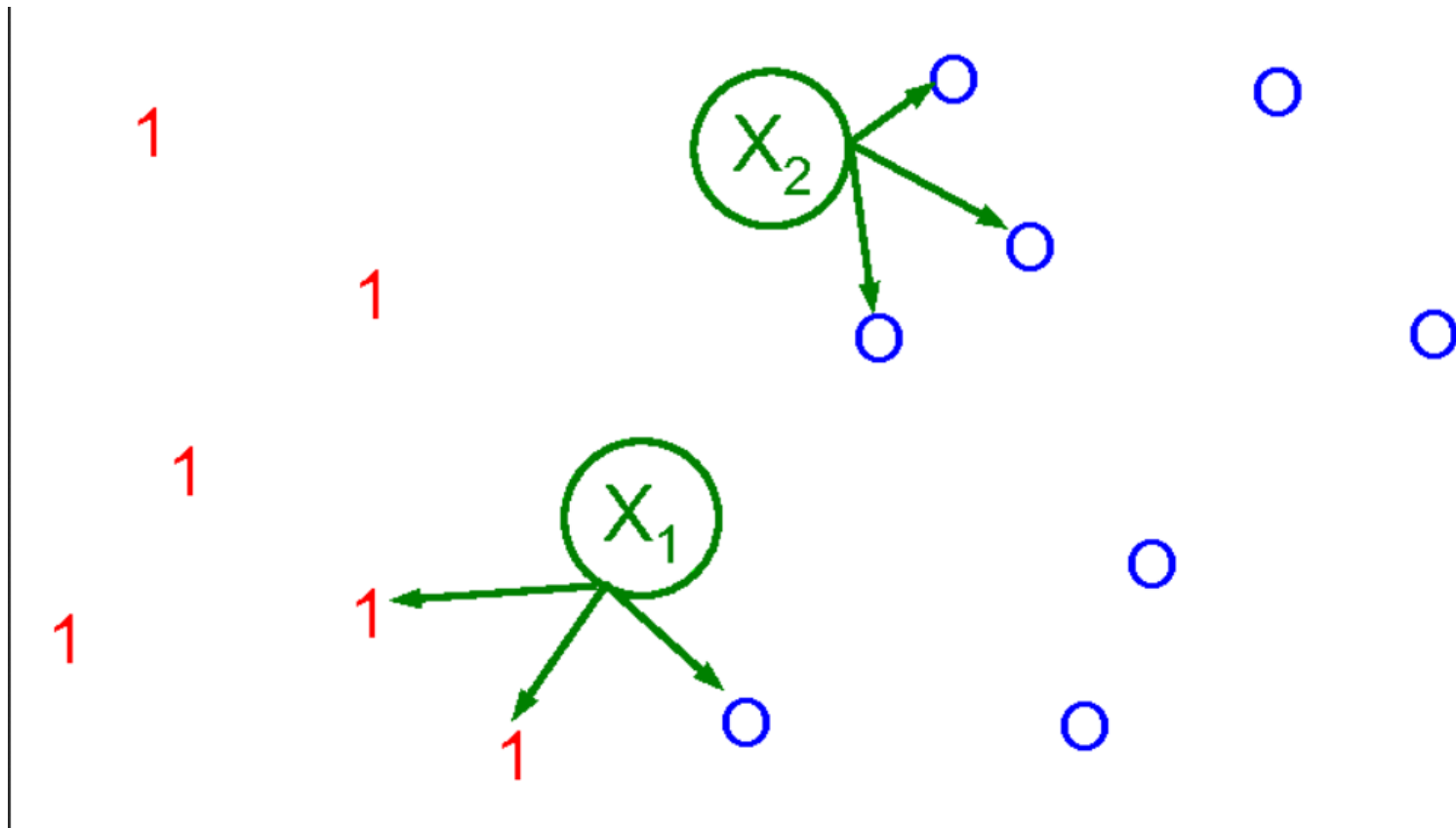


Επέκταση k -NN

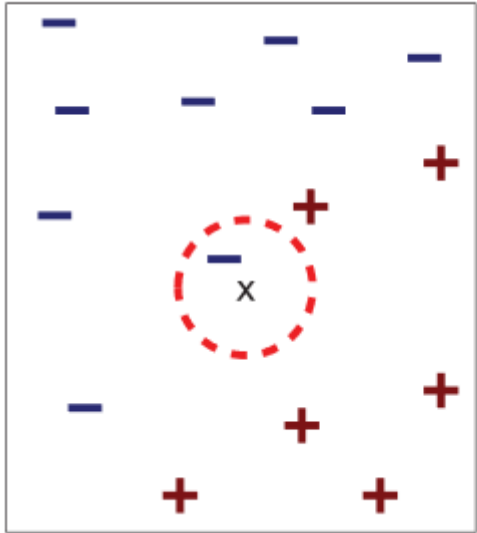
- Βρίσκουμε τους k κοντινότερους γείτονες
- Η κατηγορία του άγνωστου σημείου x^* είναι η **πλειοψηφούσα** μεταξύ των k γειτόνων.
- Παρατήρηση: Συνήθως το k επιλέγεται *περιττός* αριθμός ώστε να μην υπάρξει πρόβλημα ισοβαθμίας.
- Ερώτημα: Ποιο είναι το **κατάλληλο** k ;

Ταξινομητής k -NN

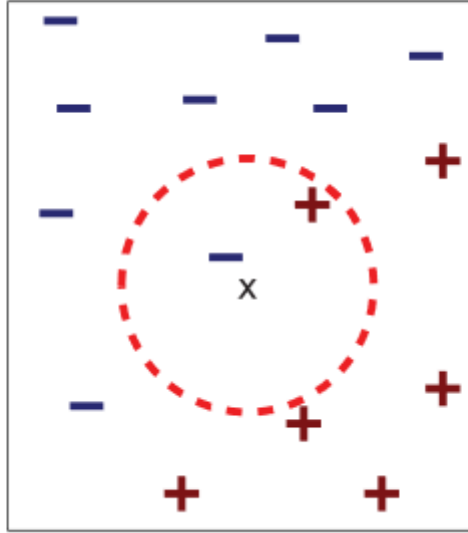
$k=3$ γείτονες



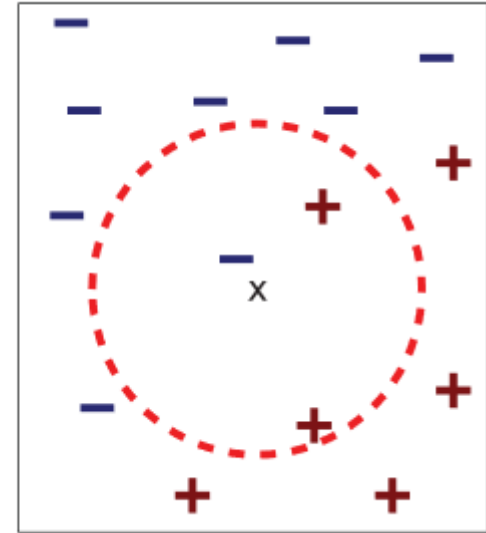
Ταξινομητής k -NN



$k=1$



$k=2$



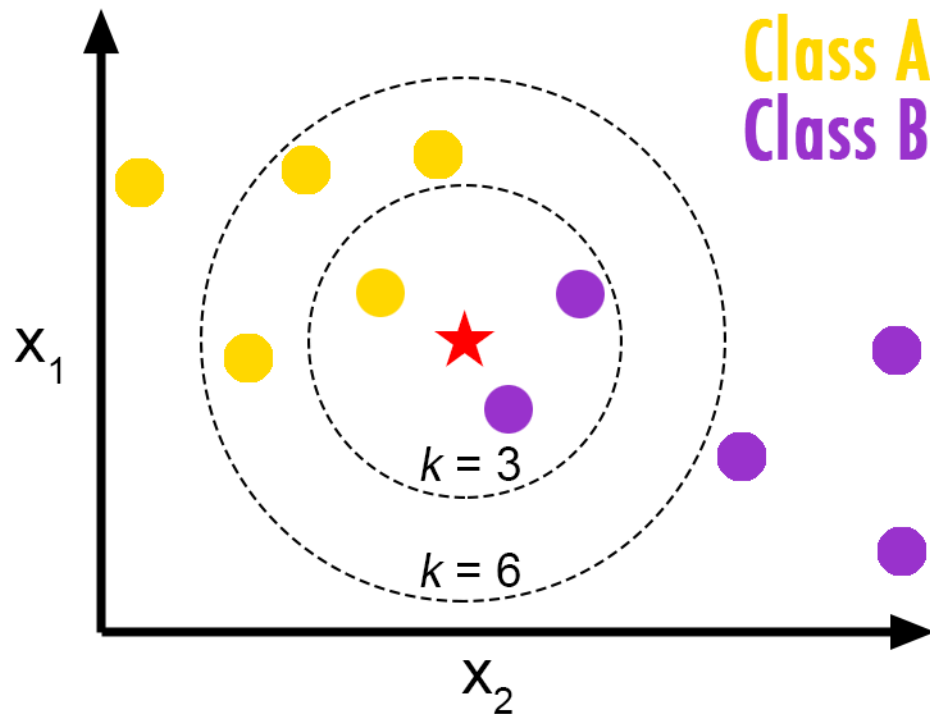
$k=3$

k -NN Classifier

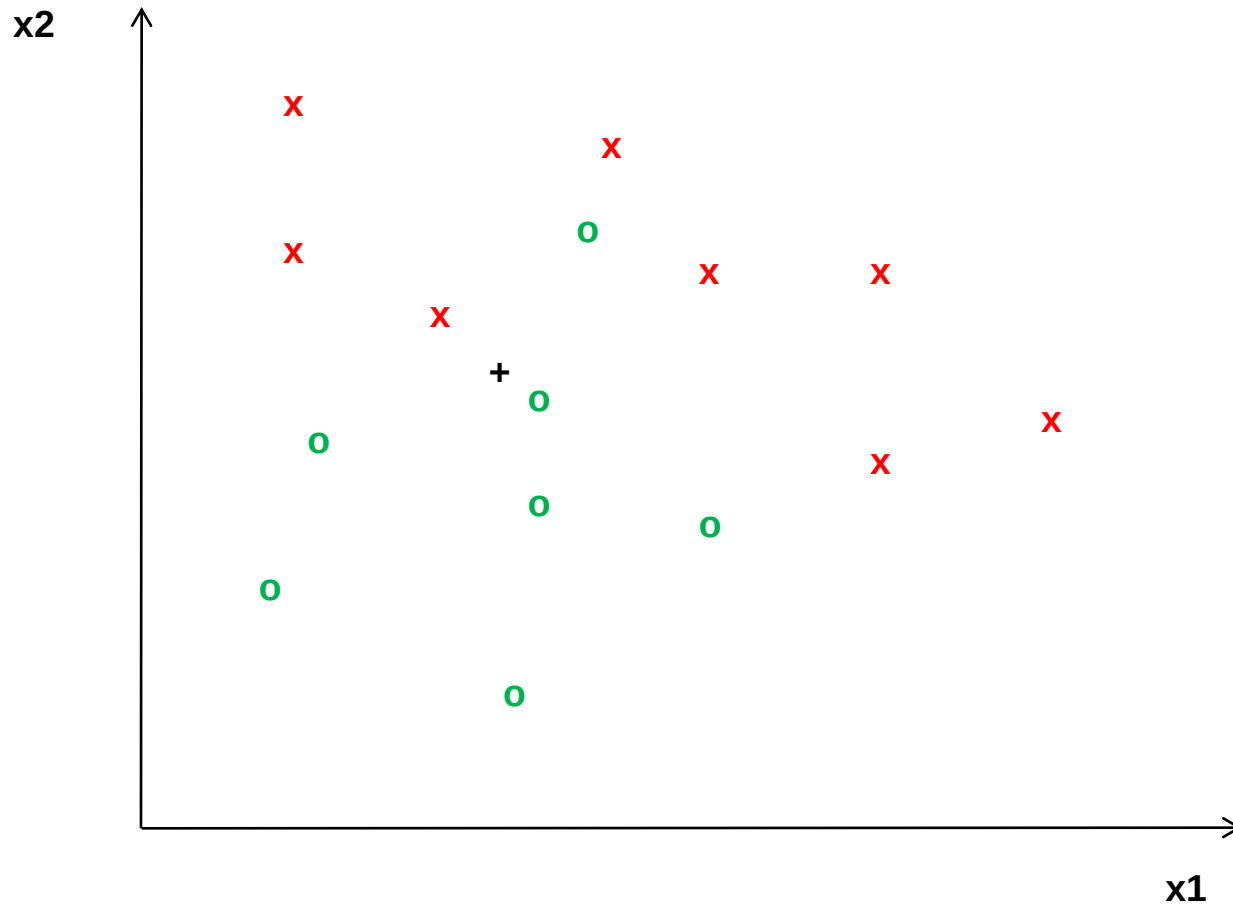
Πλεονεκτήματα	Μειονεκτήματα
Ευκολία στη χρήση.	Πλήθος δεδομένων (N) πρέπει να είναι αρκετά μεγάλο (υψηλό κόστος)
Μη - παραμετρική μέθοδος. Απουσία «εκπαίδευσης».	Απαιτείται μεγάλος αποθηκευτικός χώρος
Κατάλληλη για πολύπλοκα δεδομένα.	Πολυπλοκότητα για <i>sort</i> ως προς την απόσταση
Δεν χάνεται πληροφορία! (όλη είναι διαθέσιμη)	Εμφανίζει τοπικότητα στη λύση και όχι καθολικότητα
Ποικίλες εφαρμογές: ανάκτηση πληροφορίας – εικόνας, σήματος, video.	Βέλτιστη τιμή του k ;

Παρατηρήσεις

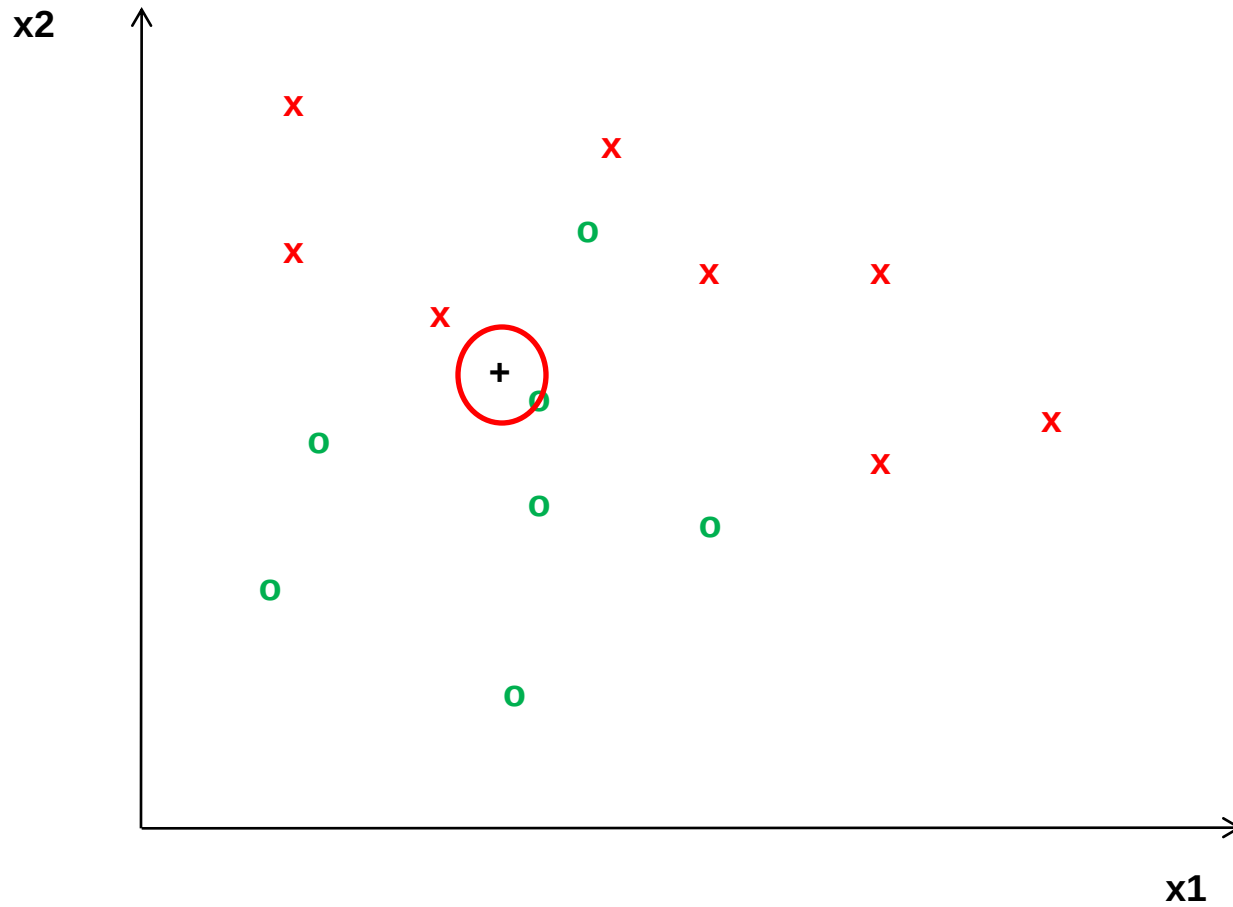
- Κατασκευή σφαιρικών επιφανειών απόφασης



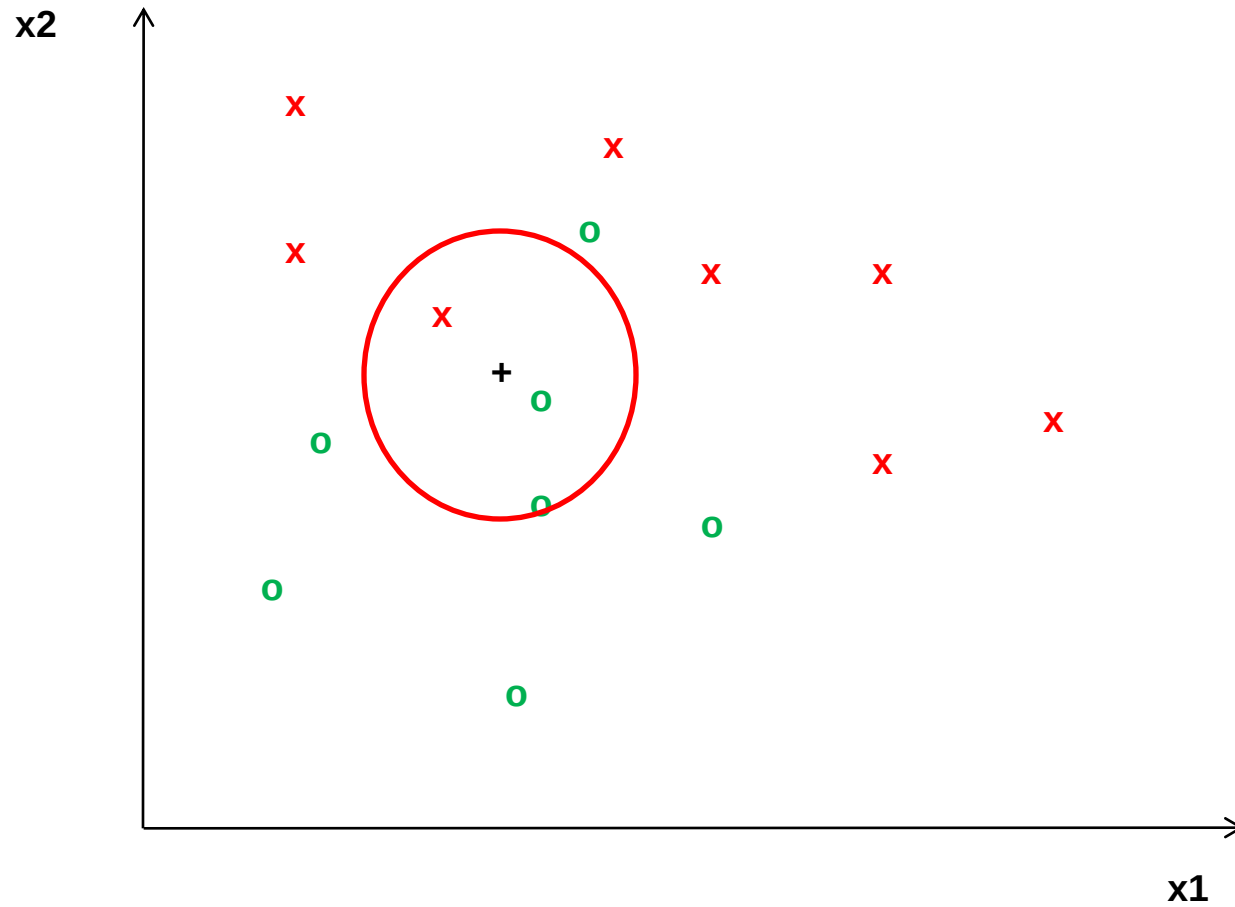
1-nearest neighbor



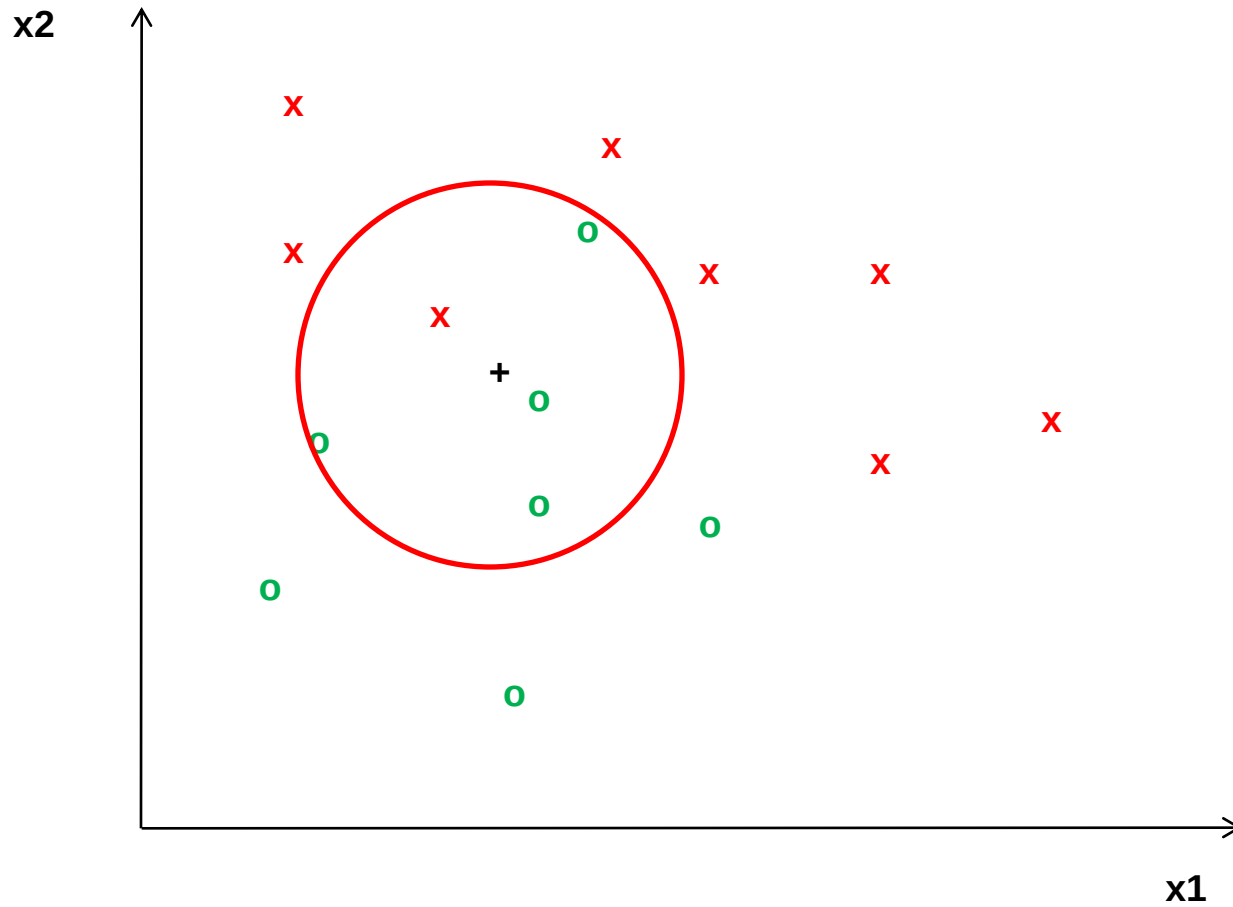
1-nearest neighbor



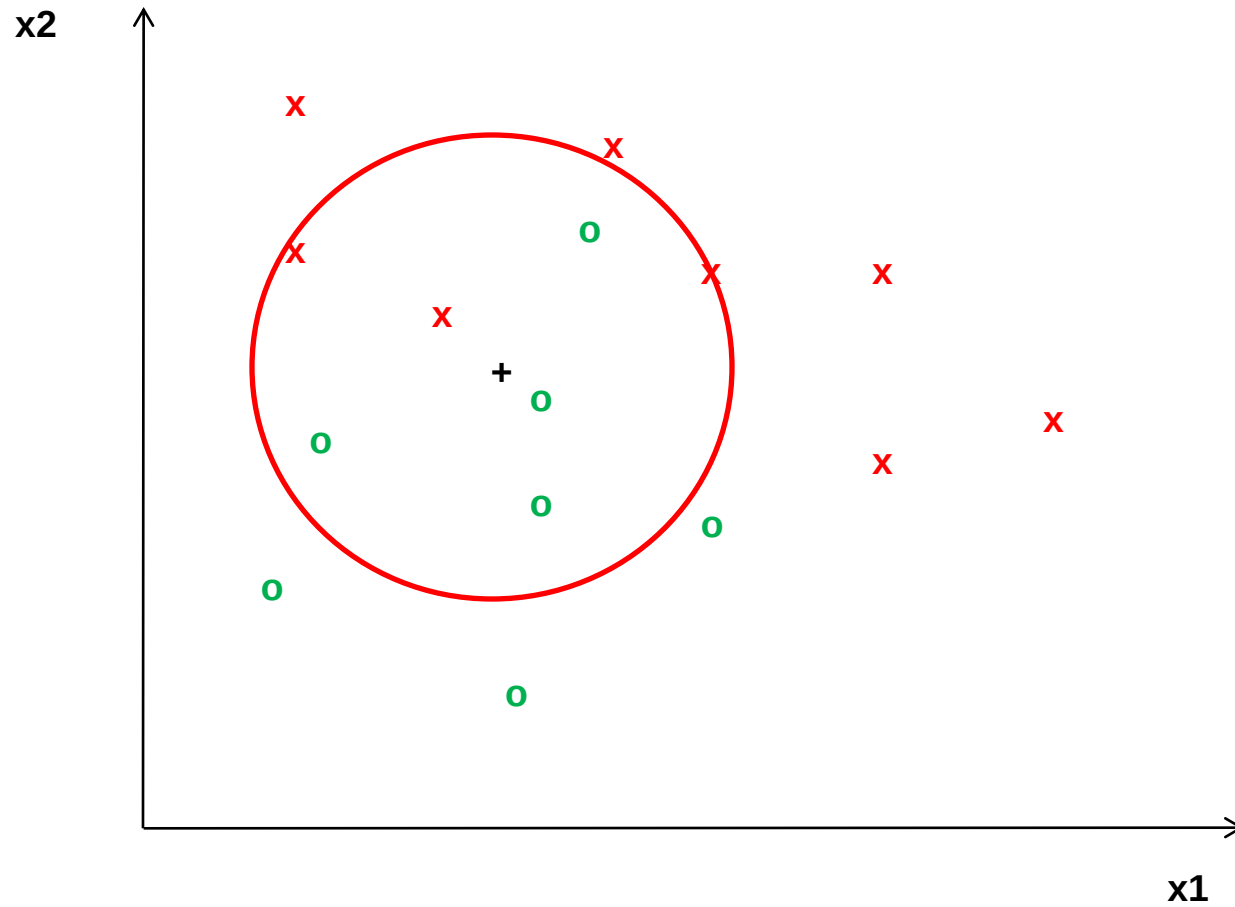
3-nearest neighbor



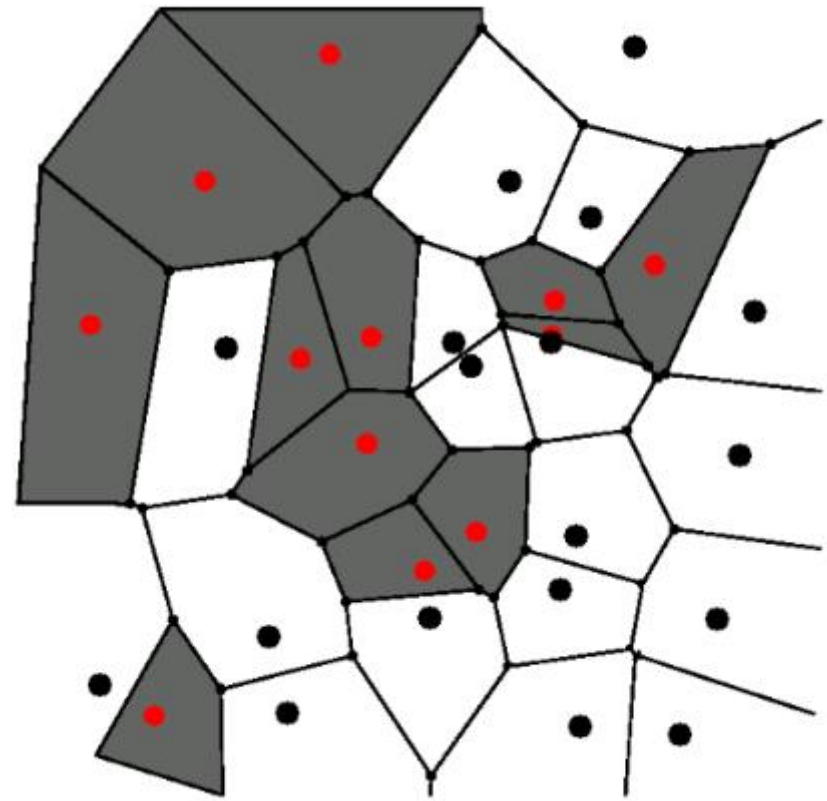
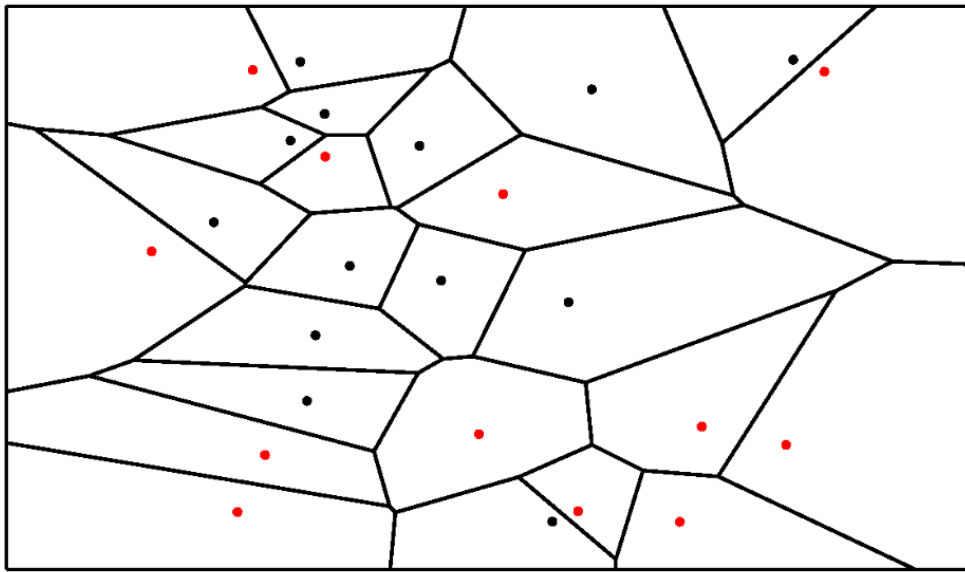
5-nearest neighbor



7-nearest neighbor

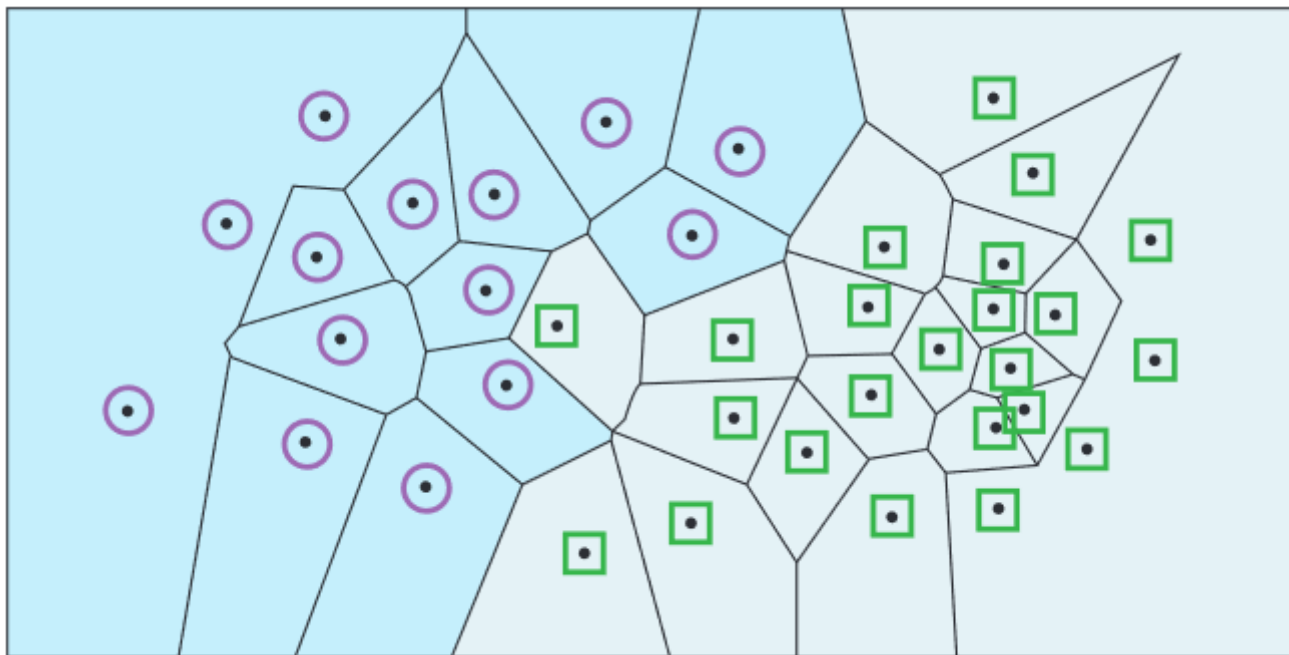


- Γεωμετρικές επιφάνειες απόφασης :
Διαγράμματα **Voronoi**



Ταξινομητής k -NN

- Επιφάνειες απόφασης από Voronoi διαγράμματα

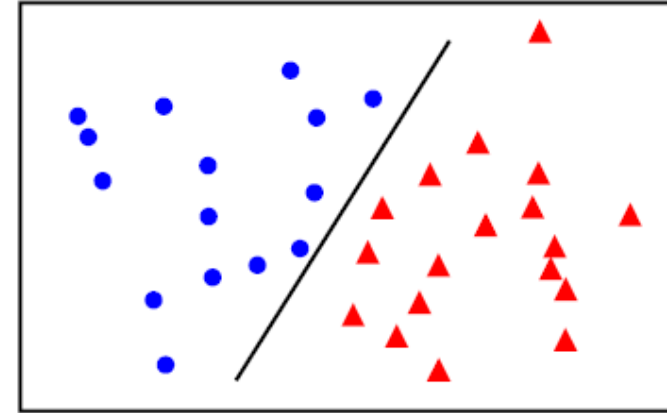
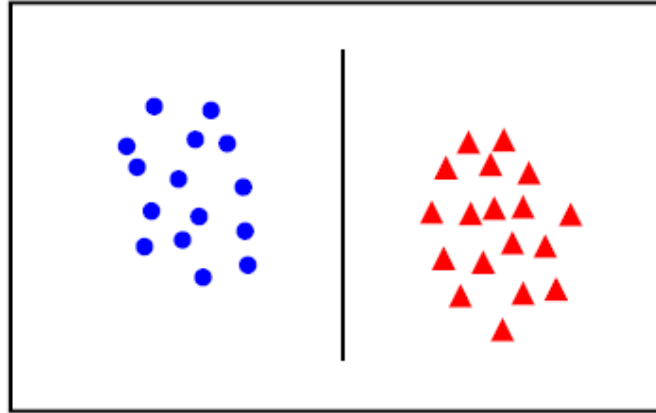


Linear classifiers (Γραμμικοί ταξινομητές)

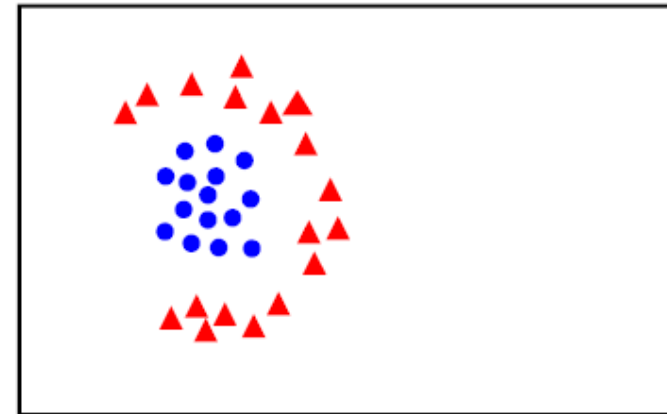
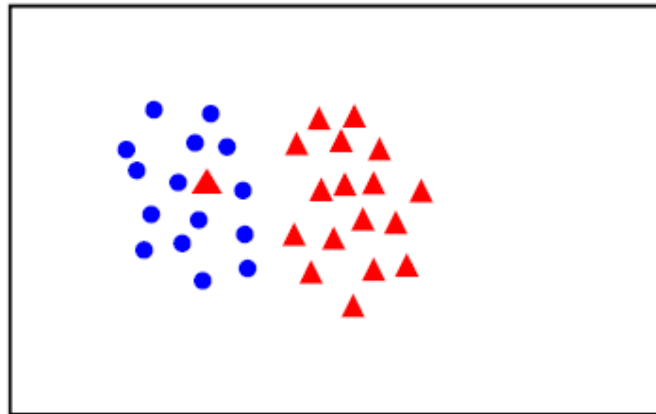
- Έστω πρόβλημα ταξινόμησης σε 2 κατηγορίες ($K=2$), ετικέτες Ω_1 και Ω_2 (**binary classification**)
- Κατασκευή συνάρτησης $f(x;w)$ για να αποφασίσουμε την κατηγορία ενός δείγματος x
- w είναι οι παράμετροι της μεθόδου-συνάρτησης
 - $if \quad f(x_n;w) \geq 0 \quad then \quad x_n \in \Omega_1$
- Κανόνας απόφασης: *else*
 - $if \quad f(x_n;w) < 0 \quad then \quad x_n \in \Omega_2$

Πρόβλημα δυαδικής ταξινόμησης (binary classification)

Γραμμική
περίπτωση



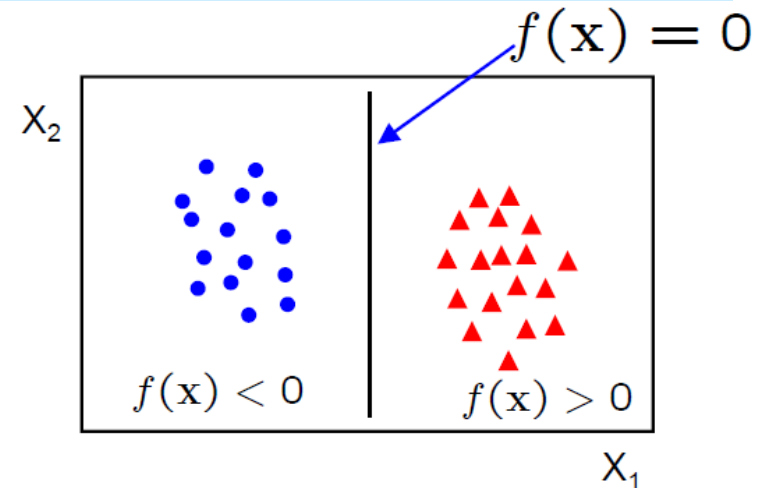
Μη γραμμική
περίπτωση



Το γραμμικό μοντέλο (Linear model)

- Υποθέτουμε ότι

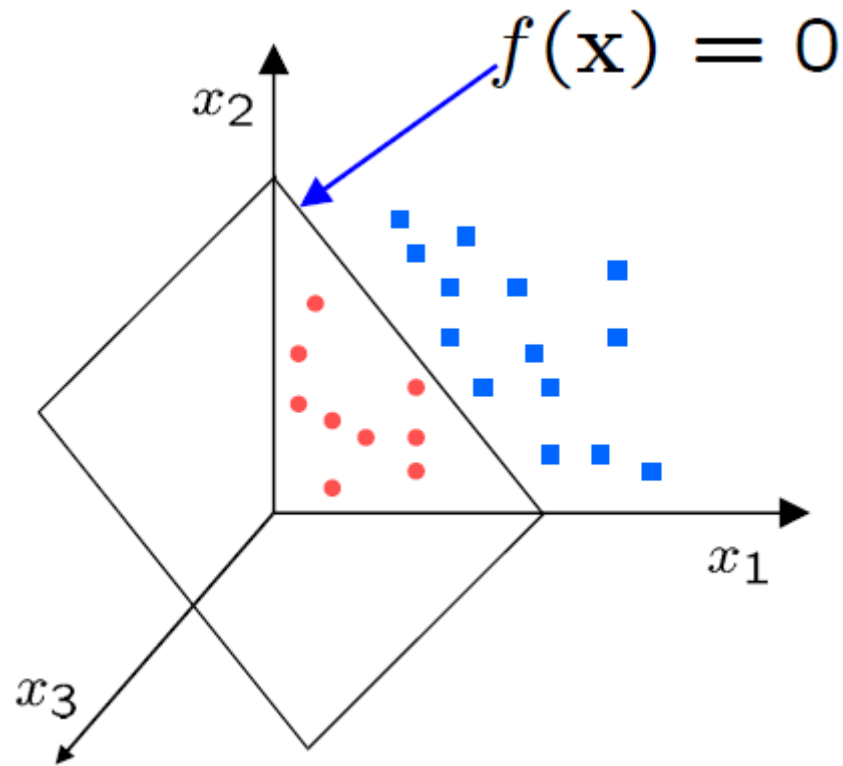
$$f(x, w) = w_0 + \sum_{j=1}^d w_j x_j = w_0 + w^T x$$



όπου $w = (w_1, \dots, w_d)$ το διάνυσμα των βαρών
 w_0 πόλωση (bias) ο σταθερός όρος

- Ορίζει μία γραμμική διαχωριστική επιφάνεια μεταξύ των 2 κατηγοριών

- Η διαχωριστική επιφάνεια σε 3D δεδομένα



- Στη περίπτωση όπου οι δυαδικές κατηγορίες έχουν τιμές $y_n \in \{+1, -1\}$, τότε:

$$\text{if } y_n = +1 \quad \Rightarrow \quad w^T x_n + w_0 \geq 0$$

else

$$\text{if } y_n = -1 \quad \Rightarrow \quad w^T x_n + w_0 < 0$$

- Εναλλακτικά:

$$\boxed{\forall x_n \in X \quad y_n (w^T x_n + w_0) \geq 0}$$

Γεωμετρική ερμηνεία του γραμμικού μοντέλου

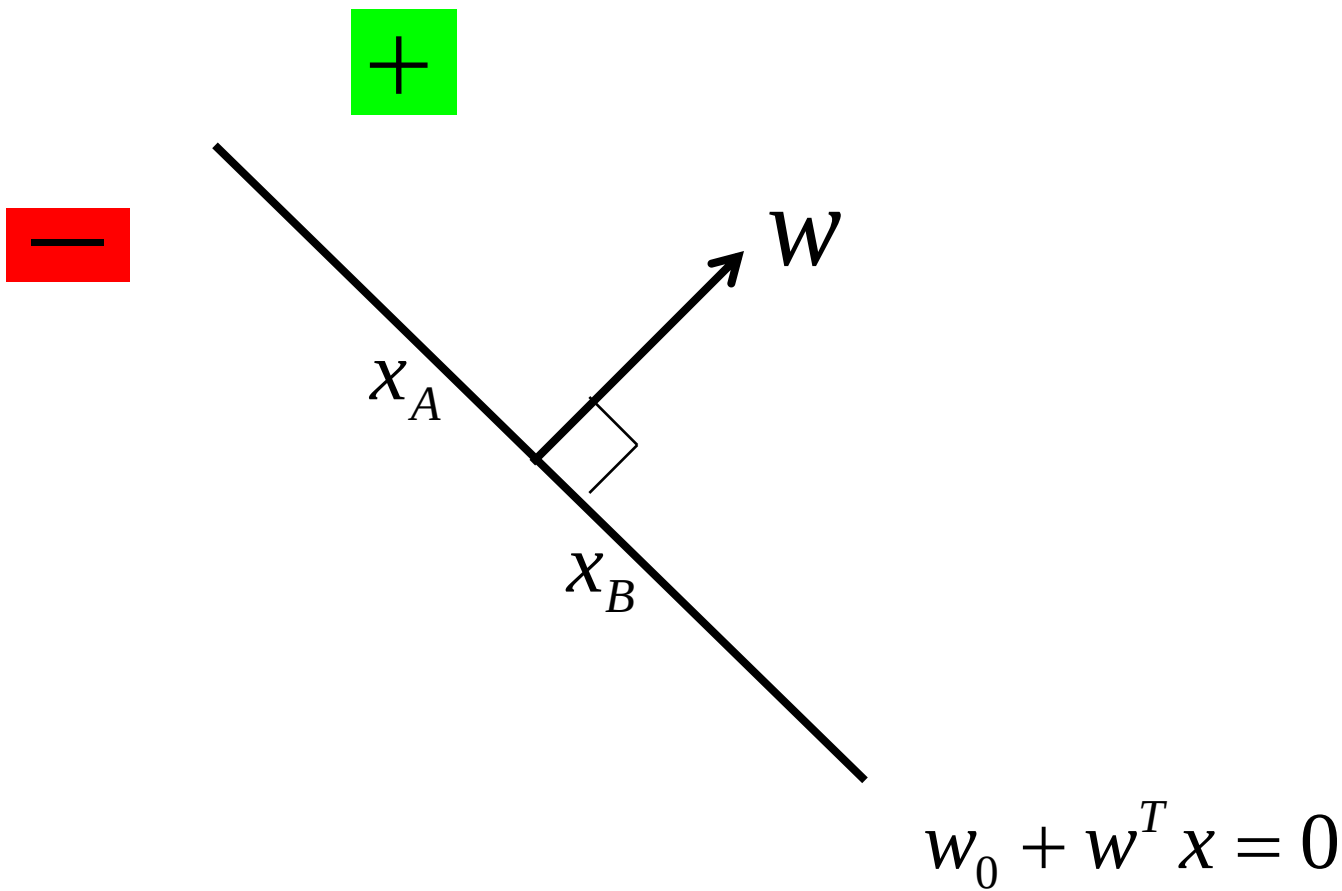
- Έστω 2 σημεία $\{x_A, x_B\}$ πάνω στην διαχωριστική επιφάνεια. Τότε ισχύει:

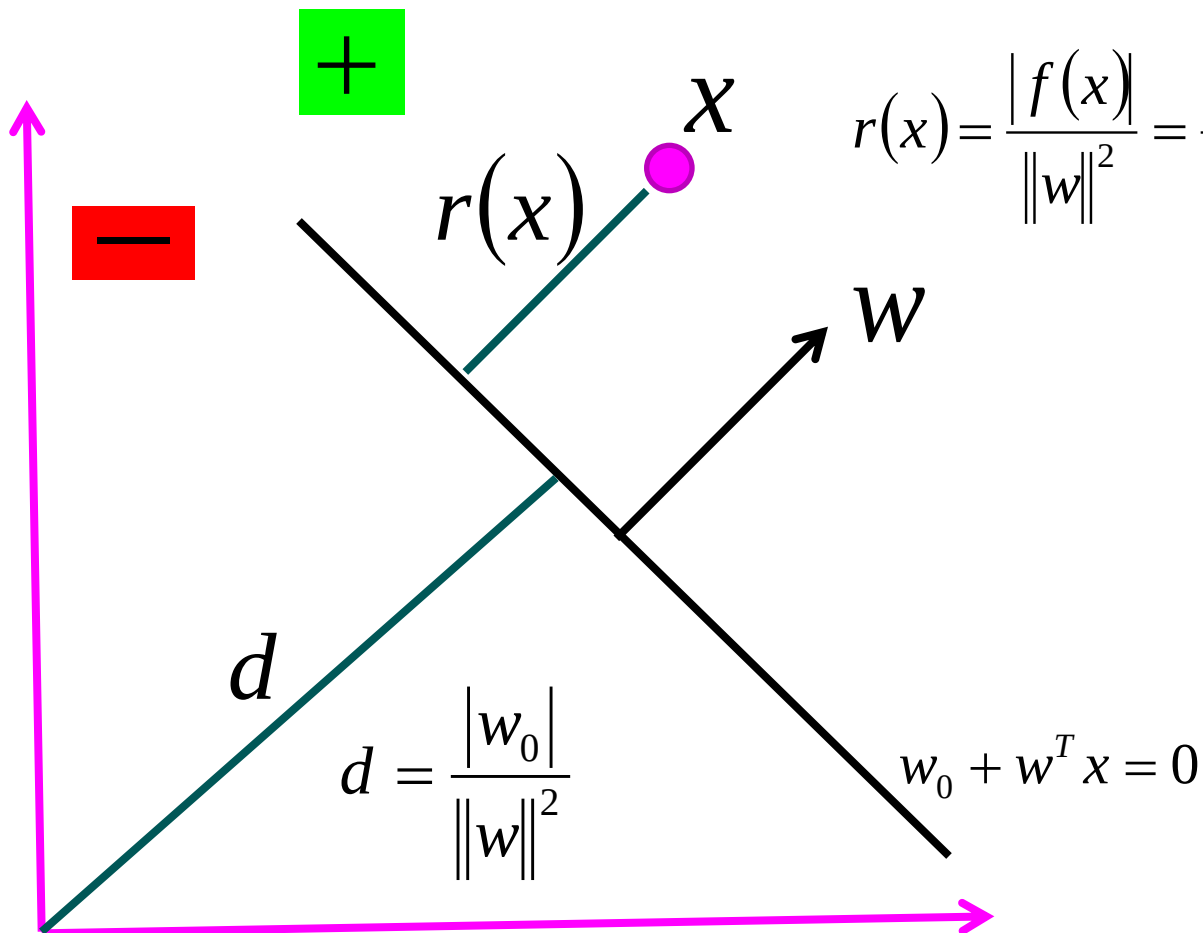
$$f(x_A) = w_0 + w^T x_A = 0$$

$$f(x_B) = w_0 + w^T x_B = 0$$

- Αφαιρώντας κατά μέλη παίρνουμε:

$$w^T (x_A - x_B) = 0$$





- Η παράμετρος w (γραμμικοί συντελεστές) είναι η παράμετρος που ορίζει τον **προσανατολισμό** της διαχωριστικής επιφάνειας,

και

- Η παράμετρος w_0 (bias) είναι η **παράμετρος** που ορίζει την **θέση** της διαχωριστικής επιφάνειας

Πρόβλημα: εύρεση καταλληλότητας του γραμμικού μοντέλου

- Σύνολο εκπαίδευσης $X = \{(x_n, y_n)\}_{n=1}^N$
όπου $x_n \in R^d$ και $y_n \in \{+1, -1\}$
- **Εκτίμηση** των παραμέτρων $\{w, w_0\}$ του γραμμικού μοντέλου, έτσι ώστε:

$$\forall n \quad y_n (w^T x_n + w_0) \geq 0$$

[1]. Ο αλγόριθμος Perceptron (Rosenblatt 1958)

To Perceptron criterion:

- Αν το πρότυπο x_n ταξινομείται **σωστά**
(δηλ. $y_n(w^T x_n + w_0) \geq 0$) τότε καμία ενέργεια.
- Αν το πρότυπο ταξινομείται **λάθος** ($y_n(w^T x_n + w_0) < 0$)
τότε η ποσότητα $-y_n(w^T x_n + w_0)$ εκφράζει την
συνεισφορά της λάθος εκτίμησης.

To Perceptron criterion:

Συνάρτηση καταλληλότητας (**ελαχιστοποίησης**):

$$E(w) = - \sum_{x_n \in \mathbf{M}} y_n (w^T x_n + w_0)$$

όπου $\mathbf{M} = \{x_n : y_n (w^T x_n + w_0) < 0\}$

το σύνολο των **λάθος ταξινομημένων** προτύπων

Ο αλγόριθμος Perceptron

- Αρχικοποίηση των παραμέτρων $\{w, w_0\}$
- Επαναληπτικά
 1. Τυχαία επιλογή ενός σημείου x_n του συνόλου εκπαίδευσης
 2. If $y_n (w^T x_n + w_0) \geq 0$ (**correct** classification) go to 1
 3. If $y_n (w^T x_n + w_0) < 0$ (**wrong** classification) then
update weights
$$w^{(new)} = w^{(old)} + \eta y_n x_n$$
$$w_0^{(new)} = w_0^{(old)} + \eta y_n$$
- Μέχρι σύγκλιση ή μέγιστο αριθμό επαναλήψεων

Ο αλγόριθμος Perceptron (συν.)

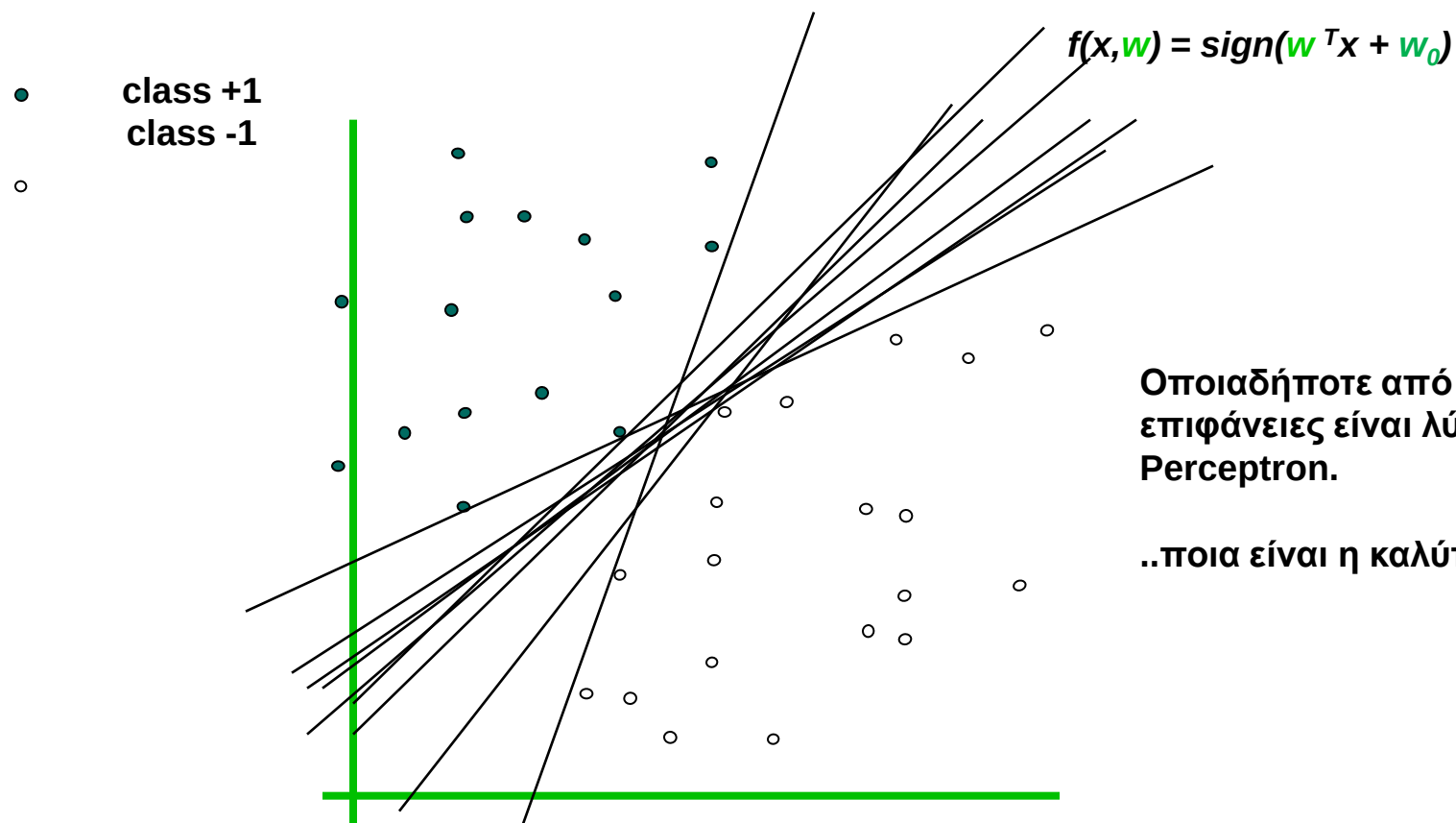
Παρατήρηση: Σε κάθε βήμα επανάληψης το σφάλμα ενός λάθος ταξινομημένου σημείου μειώνεται. Γιατί;;;

$$\begin{aligned} -y_n \left(\mathbf{w}^{(new)} \right)^T \tilde{\mathbf{x}}_n &= -y_n \left(\mathbf{w}^{(old)} + \eta y_n \tilde{\mathbf{x}}_n \right)^T = \\ &= -y_n \left(\mathbf{w}^{(old)} \right)^T \tilde{\mathbf{x}}_n - \eta y_n \left(y_n \tilde{\mathbf{x}}_n \right)^T \tilde{\mathbf{x}}_n = \\ &= -y_n \left(\mathbf{w}^{(old)} \right)^T \tilde{\mathbf{x}}_n - \underbrace{\eta y_n^2 \tilde{\mathbf{x}}_n^T \tilde{\mathbf{x}}_n}_{>0} < -y_n \left(\mathbf{w}^{(old)} \right)^T \tilde{\mathbf{x}}_n \end{aligned}$$

Παρόλ' αυτά **δεν** εξασφαλίζεται η μείωση του συνολικού σφάλματος σε κάθε επανάληψη!

Perceptron convergence theorem (Rosenblatt 1962)

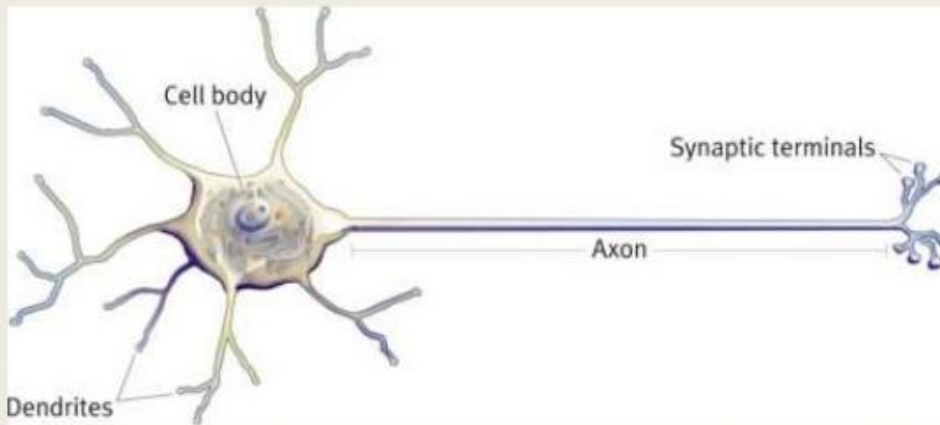
«Αν υπάρχει μοναδική λύση, τότε με βεβαιότητα ο αλγόριθμος Perceptron θα βρει την λύση. Αν δεν υπάρχει γραμμική διαχωρισιμότητα, τότε δεν είναι βέβαιο ότι ο αλγόριθμος θα βρει την βέλτιστη λύση»



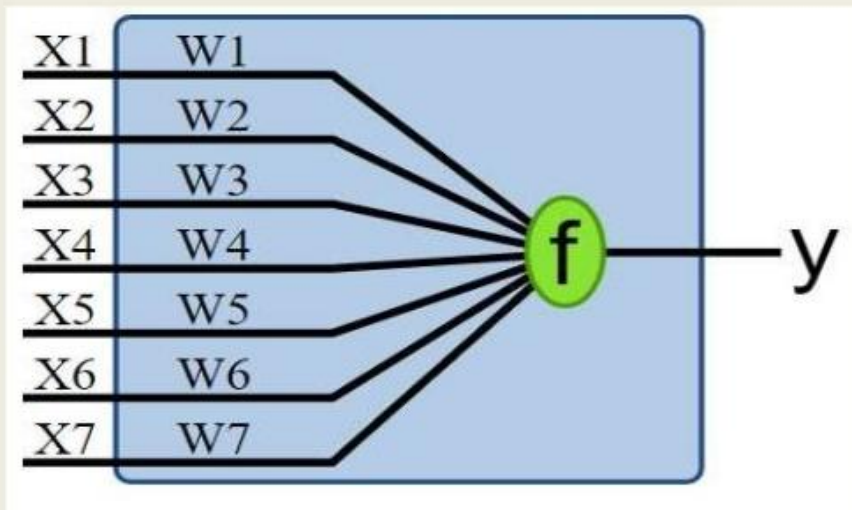
Οποιαδήποτε από τις
επιφάνειες είναι λύση του
Perceptron.

..ποια είναι η καλύτερη?

1957: Rosenblatt's "Perceptron"



http://bio3520.nicerweb.com/Locked/chap/ch03/3_11-neuron.jpg



<https://upload.wikimedia.org/wikipedia/commons/3/31/Perceptron.svg>



"The embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence."

- Office of Naval Research

<https://en.wikipedia.org/wiki/Perceptron>