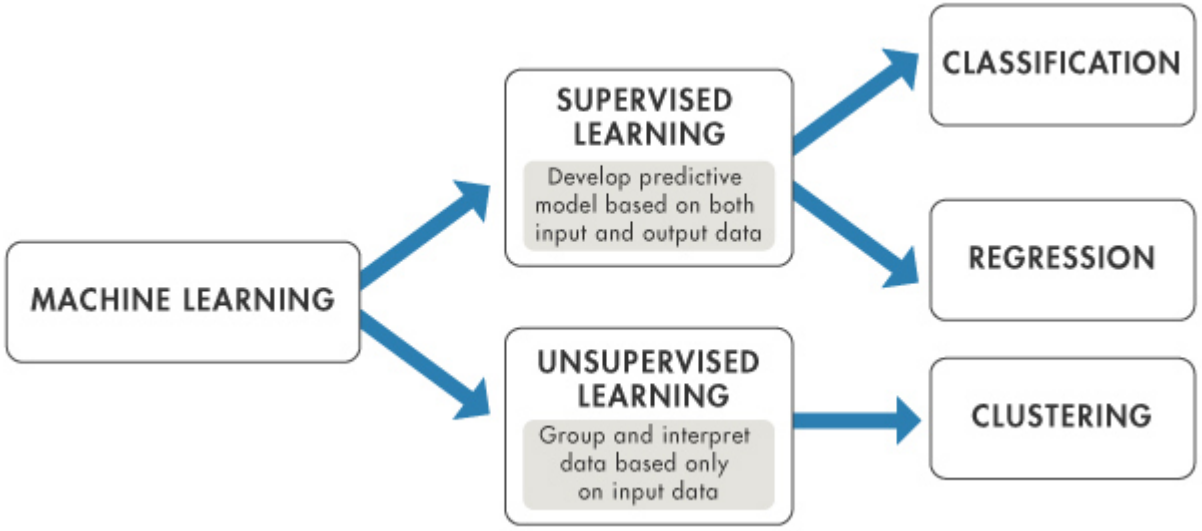
Κεφάλαιο 2

ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

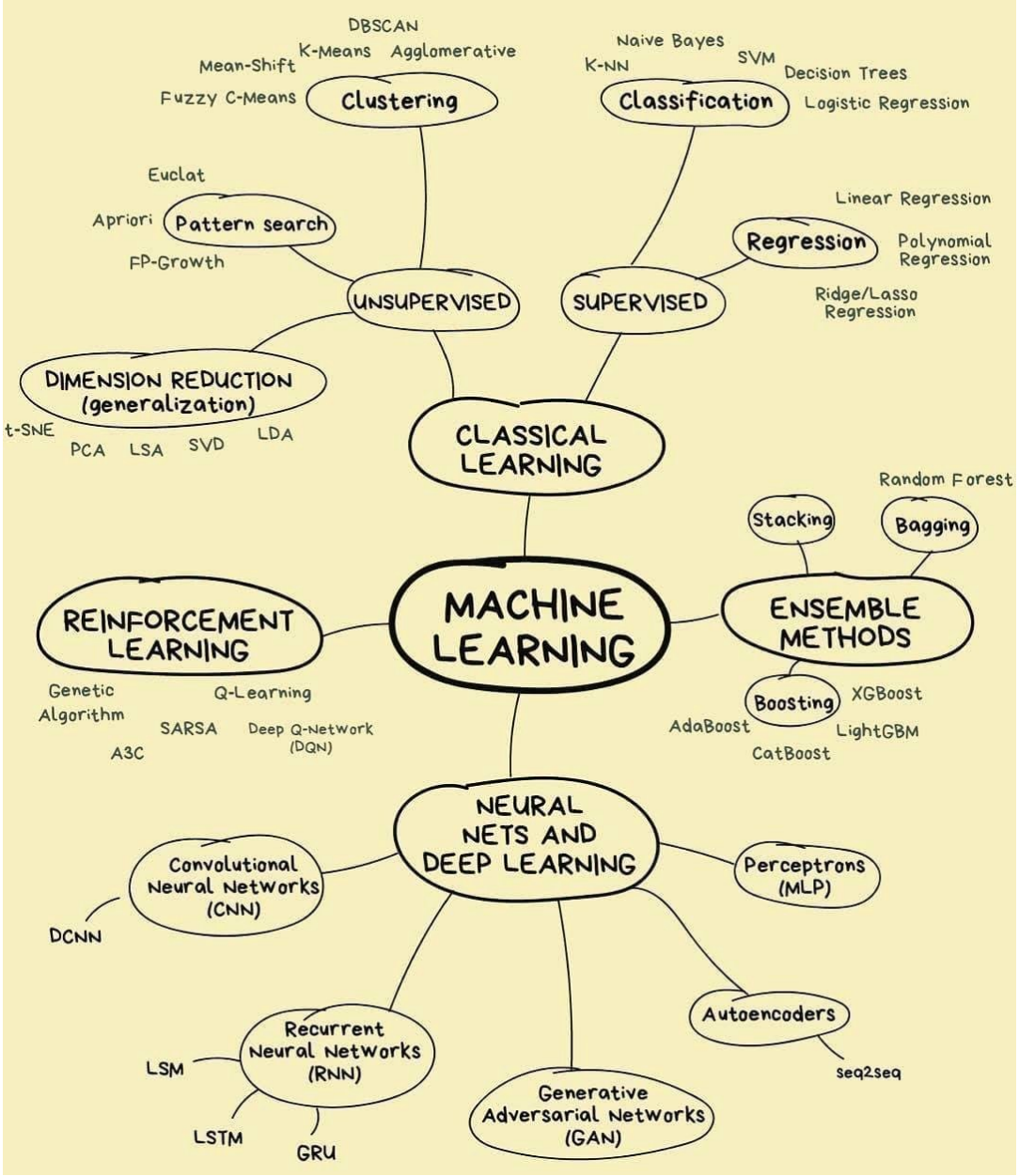


Μάθηση Συνάρτησης ή

Μάθηση με Επίβλεψη

Supervised learning

Κατηγορίες μεθόδων Μηχανικής Μάθησης



Επιβλεπόμενη Μάθηση - Ορολογία

- Δεδομένου ενός συνόλου N δεδομένων της μορφής

$$\{(x_1,y_1), \ldots ,(x_N,y_N)\}$$

  - όπου x_i είναι το διάνυσμα των τιμών (ανεξάρτητες μεταβλητές) του i-οστού παραδείγματος και y_i η ετικέτα του,
    - ✓ X_{i1} (Size), X_{i2} (Floor) και X_{i3} (Floor) οι ανεξάρτητες και Value η ετικέτα
  - χρησιμοποιείται ένας αλγόριθμος για να "μάθει" τη συνάρτηση απεικόνισης: $g: X \rightarrow Y$
  - Ο στόχος είναι η όσο το δυνατόν καλύτερη απεικόνιση έτσι ώστε όταν έχεις νέα δεδομένα εισόδου (x_i) να μπορείς να προβλέψεις με μεγάλη ακρίβεια τη μεταβλητή εξόδου (ετικέτα) Y
    - ✓ π.χ. το 10^o παράδειγμα στο διπλανό παράδειγμα

x_i	Size (sq meters) X_{i1}	Area X_{i2}	Floor X_{i3}	Value (K €) Y_i
1	100	East	2	200
2	120	East	4	250
3	80	East	3	150
4	90	Centre	5	250
5	100	Centre	1	280
6	120	Centre	2	360
7	110	West	3	100
8	130	West	5	120
9	90	West	1	90
10	150	West	4	?

- Η συνάρτηση την οποία θέλουμε να "μάθουμε" από ζευγάρια τιμών εισόδου και εξόδου, ονομάζεται ***συνάρτηση στόχος*** (*target function*).
  - Η συνάρτηση στόχος χρησιμοποιείται για την πρόβλεψη της τιμής μιας μεταβλητής (*εξαρτημένη μεταβλητή*) που ονομάζεται ***μεταβλητή στόχος*** με βάση τις τιμές ενός συνόλου από *ανεξάρτητες μεταβλητές* που ονομάζονται ***χαρακτηριστικά*** (*attributes* ή *features*).
  - Το σύνολο των διαφορετικών δυνατών τιμών εισόδου της συνάρτησης στόχος, δηλαδή το πεδίο ορισμού της, ονομάζεται **σύνολο των περιπτώσεων** ή **στιγμιότυπων** (*instances*), και συμβολίζεται με ***X***.

Ορολογία (συνεχ.)

- Ένα σύστημα επιβλεπόμενης μάθησης, κατά την εκπαίδευση του, έχει ως είσοδο ένα υποσύνολο των περιπτώσεων, για τις οποίες είναι γνωστή η τιμή της μεταβλητής στόχος
  - Αυτό ονομάζεται *σύνολο εκπαίδευσης* (*training set*) και συμβολίζεται με ***D***
  - Ένα στοιχείο του συνόλου εκπαίδευσης ονομάζεται *παράδειγμα* (*example*)
- Το σύστημα μάθησης για να προσεγγίσει όσο καλύτερα μπορεί τη συνάρτηση στόχο εξετάζει διάφορες εναλλακτικές συναρτήσεις που ονομάζονται *υποθέσεις* (*hypotheses*) και συμβολίζονται με ***h***
  - Το σύνολο όλων των δυνατών διαφορετικών υποθέσεων που το σύστημα μάθησης ενδέχεται να εξετάσει ονομάζεται *σύνολο των υποθέσεων* και συμβολίζεται με ***H***

Η υπόθεση της επαγωγικής μάθησης

- Η πιο απλή λύση στο πρόβλημα της μάθησης συνάρτησης είναι η αποστήθιση
  - Το σύστημα μάθησης απλά αποθηκεύει το σύνολο δεδομένων και μπορεί να δώσει την τιμή της *μεταβλητής στόχος* για μια νέα περίπτωση μόνο αν είναι ήδη αποθηκευμένη
  - Είναι ευνόητο ότι μια τέτοια προσέγγιση δεν περιέχει στοιχεία μάθησης και δεν είναι αποτελεσματική όταν το σύνολο των δεδομένων περιλαμβάνει ένα μικρό υποσύνολο του συνόλου των περιπτώσεων, όπως συμβαίνει στην πράξη
- Σε πολύ μεγάλα πεδία ορισμού είναι απαραίτητο να γίνει *γενίκευση* ώστε πολλές παρόμοιες περιπτώσεις να κωδικοποιηθούν πίσω από λίγες γενικευμένες περιπτώσεις
  - Δηλαδή απαιτείται μια διαδικασία *επαγωγής*
  - Το σύστημα μάθησης εξετάζοντας μόνο ένα μέρος του συνόλου των περιπτώσεων (το *σύνολο εκπαίδευσης*) καλείται να επάγει μια συνάρτηση που θα ισχύει για όλο το σύνολο
  - Αυτή η προσέγγιση στο πρόβλημα της μάθησης συνάρτησης καλείται *επαγωγική μάθηση* (*inductive learning*)
  - Η επαγωγική μάθηση στηρίζεται στην παρακάτω υπόθεση
- Υπόθεση της επαγωγικής μάθησης (inductive learning hypothesis)
  - *Κάθε υπόθεση που έχει βρεθεί να προσεγγίζει τη συνάρτηση στόχο καλά για ένα αρκετά μεγάλο σύνολο παραδειγμάτων εκπαίδευσης, θα προσεγγίζει το ίδιο καλά τη συνάρτηση στόχο και για άλλες περιπτώσεις που δεν έχει δει*

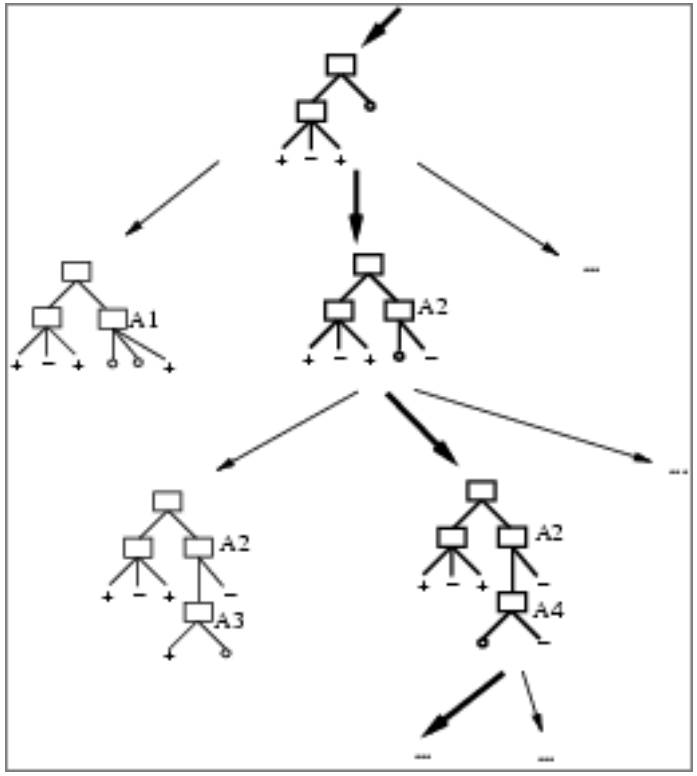
Είδη Προβλημάτων και Κυριότερες Τεχνικές

- Διακρίνονται δυο είδη προβλημάτων (*learning tasks*):
  - Τα προβλήματα *παλινδρόμησης* ή *παρεμβολής* (*regression*), που αφορούν στη δημιουργία μοντέλων πρόβλεψης συνεχών αριθμητικών τιμών, όπως για παράδειγμα:
    - ✓ η τιμή της θερμοκρασίας σε πρόγνωση καιρού ή της τιμής μιας μετοχής
    - ✓ Κάθε υπόθεση $h \in H$, αντιστοιχεί σε μια πραγματική συνάρτηση $h: X \rightarrow R$
  - Τα προβλήματα *ταξινόμησης* ή *κατηγοριοποίησης* (*classification*), που αφορούν στη δημιουργία μοντέλων πρόβλεψης διακριτών τάξεων (κλάσεων/κατηγοριών), *βαθμωτών* (*ordinal*) ή *ονομαστικών* (*nominal*)* ή τιμών, όπως για παράδειγμα η ομάδα αίματος ή η πιστοληπτική ικανότητα ενός πελάτη τράπεζας
    - ✓ Κάθε υπόθεση $h \in H$, αντιστοιχεί σε μία διακριτή συνάρτηση $h: X \rightarrow \{c_1, c_2, ..., c_n\}$ (προβλήματα ταξινόμησης n κλάσεων)
    - ✓ Στην περίπτωση δυαδικής (binary) ταξινόμησης, η διακριτή συνάρτηση εκφυλίζεται σε $h: X \rightarrow \{0, 1\}$

* With or without ordering (more information here)

Η μάθηση με επίβλεψη ως πρόβλημα αναζήτησης

- Μια από τις περισσότερο συνηθισμένες προσεγγίσεις του προβλήματος της μηχανικής μάθησης με επίβλεψη είναι αυτή της αντιμετώπισής του ως προβλήματος αναζήτησης (*search problem*) κατά την οποία:
  - "Η μηχανική μάθηση μπορεί να θεωρηθεί ως αναζήτηση σε ένα χώρο πιθανών υποθέσεων, ώστε να βρεθεί εκείνη που ταιριάζει καλύτερα στα υπό εξέταση δεδομένα και στην πιθανώς προϋπάρχουσα γνώση"
  - Για παράδειγμα στα Δένδρα Αναζήτησης:
    - ✓ Ο χώρος υποθέσεων στον οποίο κάνει αναζήτηση ο αλγόριθμος ID3 απαρτίζεται από όλα τα πιθανά δένδρα αποφάσεων.
    - ✓ Απαιτείται ο ορισμός κάποιου μηχανισμού ο οποίος θα καθοδηγήσει την αναζήτηση προς το κατ' εκτίμηση καλύτερο δένδρο (περιγραφή) μέσα στο σύνολο των δυνατών δένδρων.
    - ✓ Η αναζήτηση ξεκινά με ένα άδειο δένδρο, ενώ στη συνέχεια ο αλγόριθμος το επεκτείνει προοδευτικά με στόχο να βρει ένα δένδρο που ταξινομεί σωστά τα δεδομένα εκπαίδευσης.
    - ✓ Η στρατηγική αναζήτησης είναι αναρρίχηση λόφων (hill climbing) γιατί σε κάθε κύκλο λειτουργίας επεκτείνει το τρέχον δένδρο με τον τοπικά καλύτερο τρόπο και συνεχίζει χωρίς δυνατότητα οπισθοδρόμησης.
    - ✓ Αυτό τον κάνει αφενός εξαιρετικά αποδοτικό, αφετέρου ισχυρά εξαρτώμενο από το μηχανισμό διαχωρισμού που θα επιλεγεί.



Επαγωγική Μεροληψία (Inductive Bias) (1/2)

- Είναι το φαινόμενο παραγωγής μεροληπτικών (*biased*) αποτελεσμάτων, που οφείλεται στις υποθέσεις/επιλογές που είμαστε υποχρεωμένοι να κάνουμε
  - είτε για να αναπαραστήσουμε το χώρο των υποθέσεων (π.χ. δένδρα, νευρωνικό δίκτυο, κτλ.) ή
  - για να ορίσουμε το μηχανισμό αναζήτησης στο χώρο των υποθέσεων
- Ο χώρος των υποθέσεων, χωρίς περιορισμούς, είναι άπειρος. Συνεπώς:
  - Επιλέγοντας το είδος γνώσης δηλ. την αναπαράσταση (π.χ. δένδρα, πιθανότητες), περιορίζουμε αυτόν τον χώρο και εισάγουμε ένα πρώτο επίπεδο μεροληψίας
    - ✓ Ακόμη κι έτσι όμως, ο χώρος αναζήτησης πιθανώς να εξακολουθεί να παραμένει μεγάλος για να συντελεστεί σε αυτόν πλήρης αναζήτηση
  - Έτσι εισάγεται ένα δεύτερο επίπεδο μεροληψίας, αυτό του αλγόριθμου αναζήτησης
    - ✓ π.χ. ο αλγόριθμος ID3 δεν επιτελεί πλήρη αναζήτηση στο χώρο των πιθανών δένδρων αλλά ευρετική και μάλιστα χωρίς οπισθοδρόμηση
- Ωστόσο, χωρίς αυτές τις μεροληπτικές επιλογές, ένας αλγόριθμος μάθησης δεν θα ήταν καλύτερος από έναν αλγόριθμο τυχαίας επιλογής

Επαγωγική Μεροληψία (Inductive Bias)… (2/2)

- Τα δεδομένα εκπαίδευσης που χρησιμοποιούνται είναι πεπερασμένα, οπότε δεν αντικατοπτρίζουν με ακρίβεια την πραγματικότητα
  - Η διαδικασία της επιλογής τους αλλά και η υπόθεση που κάνουμε ότι αυτά τα δεδομένα θα έχουν την ίδια κατανομή με το σύνολο των (νέων, μελλοντικών) περιπτώσεων εισάγει ακόμη ένα επίπεδο μεροληψίας
  - Οπότε είμαστε υποχρεωμένοι να κάνουμε την υπόθεση (*υπόθεση επαγωγικής μάθησης*) ότι το μοντέλο που παράγεται από ένα περιορισμένο πλήθος δεδομένων (εκπαίδευσης) θα περιγράφει εξίσου καλά και τις νέες, άγνωστες περιπτώσεις

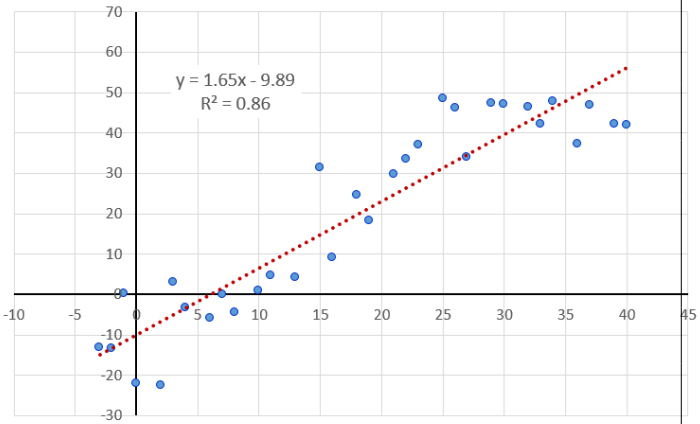
- Συνεπώς, δεν υπάρχει αμερόληπτο σύστημα μηχανικής μάθησης
  - Κάθε σύστημα (αλγόριθμος) μάθησης έχει κάποια συγκεκριμένη μεροληψία σε συγκεκριμένα στοιχεία του (αναπαράσταση, αλγόριθμο, δεδομένα) και αυτό είναι ένα θεμελιώδες, απαραίτητο χαρακτηριστικό
    - ✓ Mitchell, T. M. (1980), *The need for biases in learning generalizations*, CBM-TR 5-110, New Brunswick, New Jersey, USA: Rutgers University

Κυριότερες τεχνικές επιβλεπόμενης μάθησης

- Γραμμική Παλινδρόμηση ή Παρεμβολή – Λογιστική Παλινδρόμηση
- Μάθηση Εννοιών
- Δένδρα Απόφασης
- Μάθηση Κανόνων
- Μάθηση κατά περίπτωση
- Μάθηση κατά Bayes
- Μηχανές Υποστήριξης Διανυσμάτων-SVMs
- Νευρωνικά Δίκτυα
- Γενετικοί Αλγόριθμοι

Παλινδρόμηση ή Παρεμβολή (Regression)

- Ένας από τους πιο παλιούς χώρους υποθέσεων που έχει μελετηθεί εκτενώς στη στατιστική και γενικότερα στα μαθηματικά είναι οι γραμμικές συναρτήσεις συνεχών μεταβλητών
  - ✓ Belongs to both statistics and machine learning. Is an attractive model because the representation is so simple.
  - ✓ Assumes a linear relationship between the input variables (x) and the single output variable (y)
- Είναι η διαδικασία προσδιορισμού της σχέσης μιας συνεχούς μεταβλητής y, (*εξαρτημένη μεταβλητή* ή *έξοδος*), με μια ή περισσότερες άλλες μεταβλητές $x_1, x_2, \dots, x_n$ (*ανεξάρτητες μεταβλητές* ή *είσοδοι*)
  - Σκοπός: η πρόβλεψη της τιμής της εξόδου για νέες περιπτώσεις εισόδου
  - Ευρέως χρησιμοποιούμενη στατιστική τεχνική μοντελοποίησης
- Απλή ή Μονοπαραμετρική Γραμμική Παρεμβολή (*linear regression*)
  - Στη Γραμμική παρεμβολή αναμενόμενη τιμή της εξόδου μοντελοποιείται με μία γραμμική συνάρτηση f ή σταθμισμένο άθροισμα (*weighted sum*) των παραμέτρων εισόδου.
    - ✓ Η συνάρτηση *f* είναι η ζητούμενη υπόθεση *h*
  - Στην απλή παρεμβολή, η εξαρτημένη μεταβλητή εξαρτάται από μια μόνο ανεξάρτητη μεταβλητή και μπορεί να παρασταθεί με μια εξίσωση ευθείας γραμμής της μορφής $y = f_{\boldsymbol{w}}(x) = w_0 + w_1 \cdot x$
    - ✓ στην οποία πρέπει να υπολογιστούν οι παράμετροι w_0 και w_1 (ονομάζονται και βάρη - weights) από τα δεδομένα (παραδείγματα) εκπαίδευσης
    - ✓ Υπάρχουν πολλές επιλογές και κάθε επιλογή δίνει διαφορετική ευθεία
    - ✓ Ζητούμενο: η ευθεία να είναι όσο το δυνατόν πιο κοντά στα δεδομένα



  - Κάτι τέτοιο μπορεί να υπολογιστεί ελαχιστοποιώντας το συνολικό τετραγωνικό σφάλμα στα δεδομένα εκπαίδευσης

Linear Regression for Machine Learning

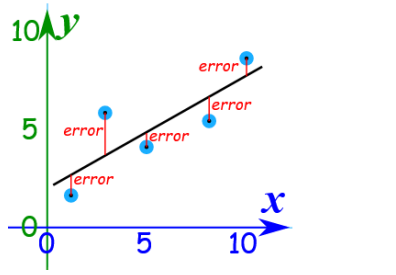
Isn’t Linear Regression from Statistics?

- In applied machine learning we borrow, reuse and steal algorithms from many different fields, including statistics and use them towards these ends.
  - As such, linear regression was developed in the field of statistics and is studied as a model for understanding the relationship between input and output numerical variables but has been borrowed by machine learning.
  - It is both a statistical algorithm and a machine learning algorithm

Μέθοδος των ελαχίστων τετραγώνων (least squares)

❖ [1]The method of least squares is a parameter estimation method in regression analysis based on minimizing the sum of the squares of the residuals

$$S = \sum_{i=1}^{N} (y_i - f_{\boldsymbol{w}}(x_i))^2 = \sum_{i=1}^{N} (y_i - (w_0 + w_1 \cdot x_i))^2$$



The straight line minimizes the sum of squared errors

❑ όπου residual είναι η ποσότητα $\left(y_i - f_{\boldsymbol{w}}(x_i)\right)$ και είναι το σφάλμα μεταξύ της γνωστής τιμής y_i και της υπολογιζόμενης τιμής $f_{\boldsymbol{w}}(x_i)$ με τις τρέχουσες τιμές των βαρών.

✓ Ο δείκτης *i* διατρέχει όλα τα δεδομένα εκπαίδευσης.

❖ Το παραπάνω άθροισμα, ως συνάρτηση των βαρών, ελαχιστοποιείται στα w_0 και w_1 στα οποία μηδενίζονται οι πρώτες μερικές παράγωγοι της συνάρτησης ως προς w_0 και w_1, αντίστοιχα*.

❑ Προκύπτουν έτσι οι ακόλουθες τιμές:

$$w_1 = \frac{N \cdot \sum(x_i \cdot y_i) - (\sum x_i) \cdot (\sum y_i)}{N \cdot (\sum x_i^2) - (\sum x_i)^2} \qquad w_0 = \frac{(\sum y_i - w_1 \cdot (\sum x_i))}{N}$$

❖ Η διαδικασία αυτή είναι γνωστή και ως μέθοδος των ελαχίστων τετραγώνων (*least squares*) μια από τις πιο διαδεδομένες μεθόδους υπολογισμού των συντελεστών.

❑ Για τα δεδομένα του προηγούμενου σχήματος δίνει w_1=1.65 και w_0=9.89 (δηλ. *y=1.65x-9.89*)

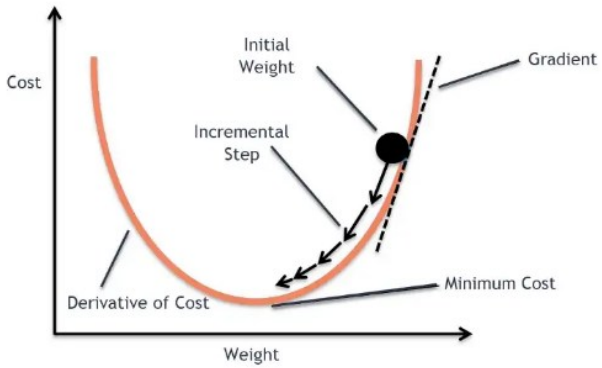
❑ Η συνάρτηση $y = w_0 + w_1 \cdot x$ λέγεται ευθεία ελαχίστων τετραγώνων ή ευθεία παλινδρόμησης (παρεμβολής)

*Finding Maxima and Minima using Derivatives

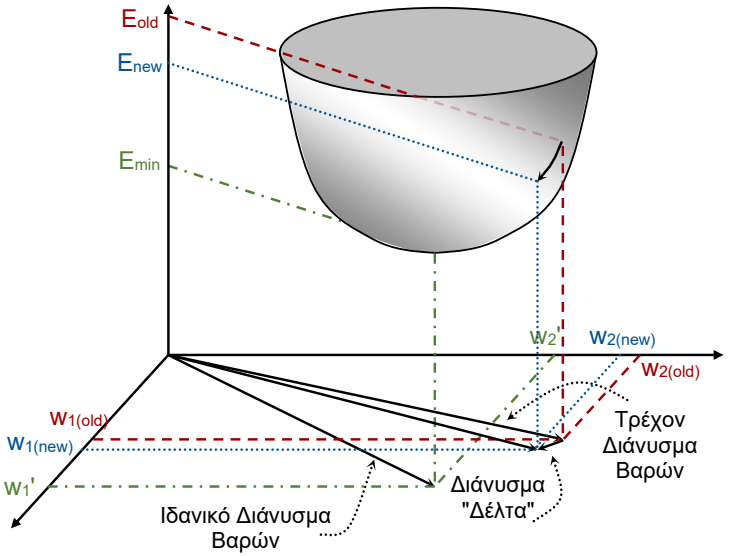
[1] https://www.mathsisfun.com/data/least-squares-regression.html

Συνέχεια με …(Μηχανική Μάθηση)

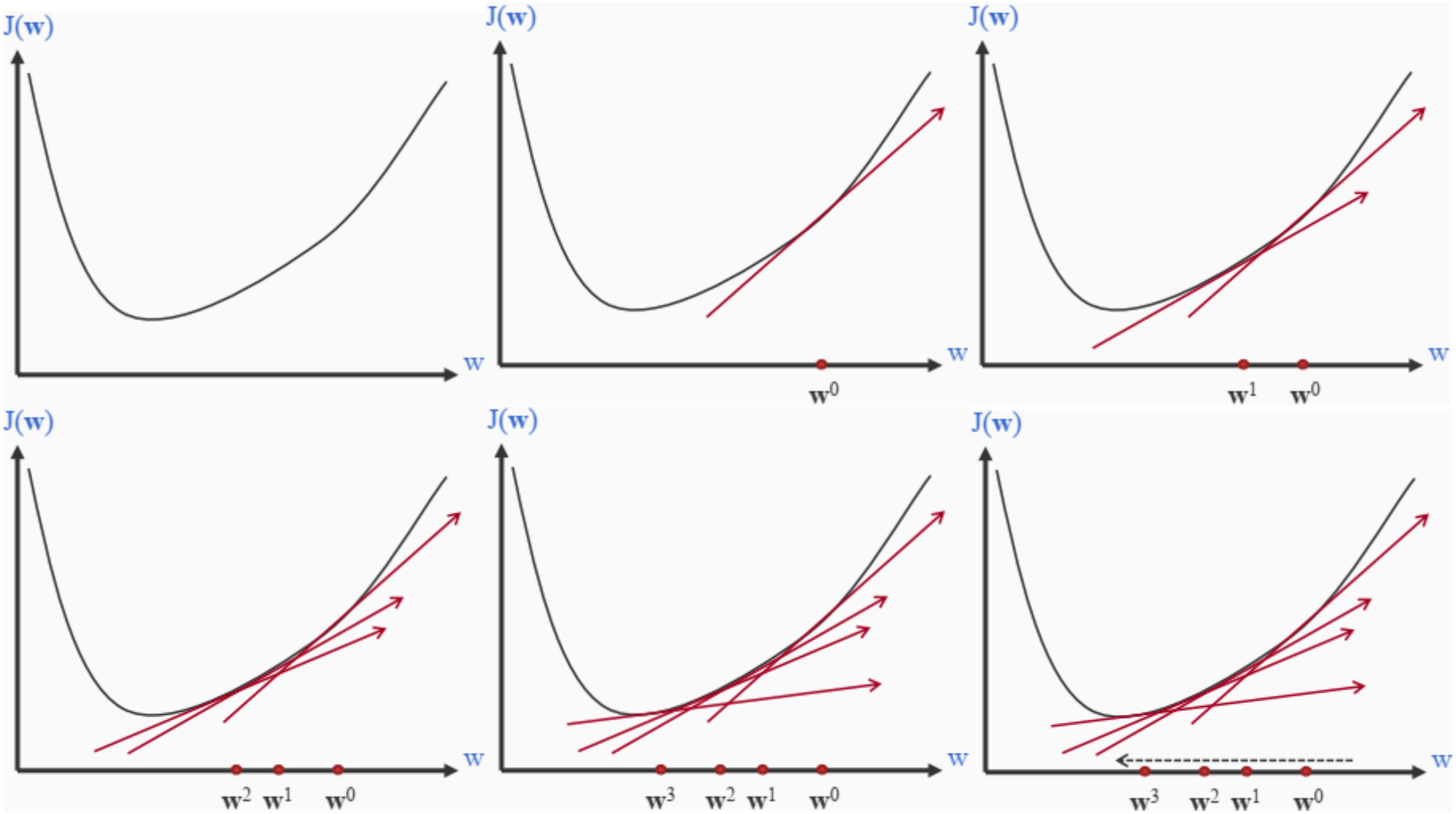
- Στο παραπάνω πρόβλημα, η συνάρτηση S είναι μια κυρτή επιφάνεια στον τρισδιάστατο χώρο (w_0, w_1, $(S(_w)$) που έχει ένα και μοναδικό ελάχιστο
  - Ο υπολογισμός των w_0 και w_1 από τις παραπάνω εξισώσεις είναι η προφανής οδός προσδιορισμού της ζητούμενης ευθείας



- Στις περιπτώσεις που δεν είναι τόσο απλές, μια γενική μεθοδολογία αναζήτησης του ελαχίστου είναι η χρήση ενός optimizer και μιας Loss function
- Ένας optimizer είναι ο GD (*Gradient Decent)* που μπορεί να αναζητήσει το ελάχιστο ακολουθώντας την αρνητική κλίση της εφαπτομένης στην παραπάνω επιφάνεια
  - Ο κανόνας της διαβαθμισμένης καθόδου (*gradient descent rule*) ή *κανόνας Δέλτα*, ακολουθεί την αρνητική κλίση της επιφάνειας σφάλματος, με κατεύθυνση προς μικρότερες τιμές της
  - Ταυτόχρονα, εξασφαλίζει και τον βέλτιστο τρόπο μετακίνησης του διανύσματος των βαρών



Intuitive results:



Intuition: The gradient is the direction of steepest increase in the function. To get to the minimum, go in the opposite direction

Optimizers

- Are algorithms or methods used to minimize an error function (loss function)or to maximize the efficiency of a model tweaking learnable parameters e.g. weights.
  - ✓ After we pass input, we calculate the error and update the weights accordingly.
  - ✓ algorithm based on a convex function and tweaks its parameters iteratively to minimize a given function to its local minimum
  - ✓ PyTorch itself has 13 optimizers, making it challenging and overwhelming to pick the right one for the problem
- The five most popular optimizers
  - Gradient Descent. Algorithm based on a convex function
    - ✓ Stochastic Gradient Descent(SGD) — calculates gradient for each random sample
    - ✓ Mini-Batch Gradient Descent — computes gradient over randomly sampled batch
    - ✓ Batch Gradient Descent — computes gradients for the entire dataset
  - Adam optimizer. Is the extended version of SGD.
    - ✓ Adam is proposed as the most efficient stochastic optimization which only requires first-order gradients where memory requirement is too little
    - ✓ https://optimization.cbe.cornell.edu/index.php?title=Adam
  - Exponential Moving Average
  - Momentum
  - AdaGrad
  - MadGrad optimizer: A novel optimization method in the family of AdaGrad adaptive gradient methods.
    - ✓ MADGRAD shows excellent performance on deep learning optimization problems.
  - RMSP (Root Mean Square Propagation)

https://medium.com/nerd-for-tech/optimizers-in-machine-learning-f1a9c549f8b4

The Most Common Machine Learning Loss Functions

❖ Loss Functions for Regression

❑ Mean Absolute Error

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

❑ Mean Square Error / Quadratic Loss

- ✓ Τετραγωνική συνάρτηση απώλειας (squared or quadratic or L2 loss function)
- ✓ $Loss(f_{\boldsymbol{w}}) = \sum_{i=1}^{N} L_2(y_i, f_{\boldsymbol{w}}(x_i)) = \sum_{i=1}^{N} (y_i - f_{\boldsymbol{w}}(x_i))^2 = \sum_{i=1}^{N} (y_i - (w_0 + w_1 \cdot x_i))^2$

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (\hat{Y}_i - Y_i)^2$$

❑ Huber Loss / Smooth Mean Absolute Error

- ✓ The Huber loss function is defined as the combination of MSE and MAE loss functions

❑ Log-Cosh Loss

- ✓ The log-cosh loss function is defined as the logarithm of the hyperbolic cosine of the prediction error

$$L(y, y^p) = \sum_{i=1}^{n} \log(\cosh(y_i^p - y_i))$$

❑ Quantile Loss

- ✓ A quantile is a value below which a fraction of samples in a group falls.

❖ Loss Functions for Classification

❑ Binary Cross-Entropy Loss / Log Loss

- ✓ This is the most common loss function used in classification problems. It decreases as the predicted probability converges to the actual label. It measures the performance of a classification model whose predicted output is a probability value between 0 and 1.

$$L = -\frac{1}{m} \sum_{i=1}^{m} y_i \cdot \log(\hat{y}_i)$$

❑ Hinge Loss

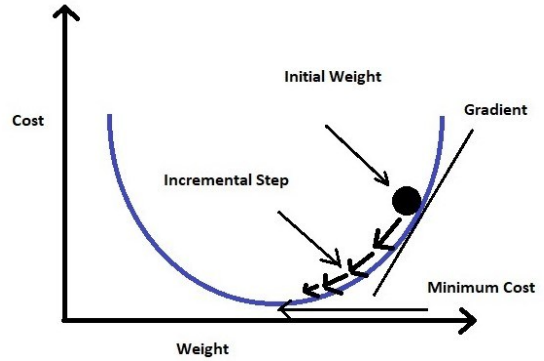
- ✓ The second most common loss function used for classification problems and an alternative to the cross-entropy loss function. Primarily developed for SVM model evaluation.

$$L = \max(0, 1 - y * f(x))$$

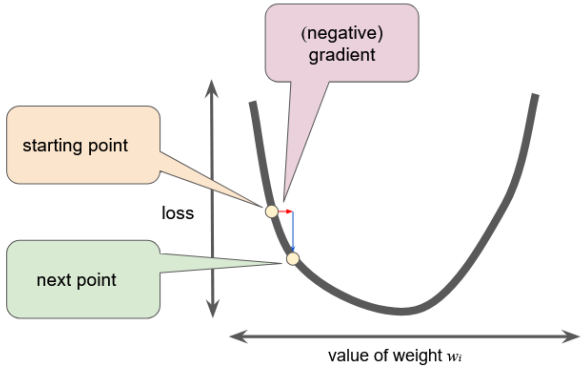
- ✓ A Unified View of Loss Functions in Supervised Learning - The 7 Most Common Machine Learning Loss Functions
- ✓ https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23

Gradient Descent Optimization (Reducing Loss)

- Gradient descent aims at finding the optimal weights which minimize the loss function of our model
  - It is an iterative method that finds the minimum of a function by figuring out the slope at a random point and then moving in the opposite direction



- The resulting plot of loss vs. w_1 will always be convex (bowl-shaped)
- Convex problems have only one minimum; where the slope is exactly 0
  - That minimum is where the loss function converges
    - ✓ The first stage in gradient descent is to pick a starting value (point) for w_1
    - ✓ The algorithm then calculates the gradient of the loss curve at the starting point
    - ✓ The gradient of the loss is equal to the derivative (slope) of the curve, and tells us which way is "warmer" or "colder."
    - ✓ When there are multiple weights, the gradient is a vector of partial derivatives with respect to the weights.



  - A gradient is a vector, so it has both a direction and a magnitude
    - ✓ The gradient always points in the direction of steepest increase in the loss function.
    - ✓ The algorithm takes a step in the direction of the negative gradient in order to reduce loss as quickly as possible.
    - ✓ To determine the next point along the loss function curve, the algorithm adds some fraction of the gradient's magnitude to the starting point
  - The gradient descent then repeats this process, edging ever closer to the minimum

---

1. Gradient
2. slope or gradient of a line is a number that describes both the direction and the steepness of the line
3. Reducing Loss: Gradient Descent

Linear Regression for Machine Learning

Simple Linear Regression

- With simple linear regression when we have a single input, we can use statistics to estimate the coefficients.
- This requires that you calculate statistical properties from the data such as means, standard deviations, correlations and covariance. All of the data must be available to traverse and calculate statistics.
- This is fun as an exercise in excel, but not really useful in practice.

Ordinary Least Squares

- When we have more than one input we can use *Ordinary Least Squares* or *Linear least squares* to estimate the values of the coefficients.
- The *Ordinary Least Squares* procedure seeks to minimize the sum of the squared residuals. This means that given a regression line through the data we calculate the distance from each data point to the regression line, square it, and sum all of the squared errors together. This is the quantity that *ordinary least squares* seeks to minimize.
- This approach treats the data as a matrix and uses linear algebra operations to estimate the optimal values for the coefficients. It means that all of the data must be available and you must have enough memory to fit the data and perform matrix operations.
- It is unusual to implement the Ordinary Least Squares procedure yourself unless as an exercise in linear algebra. It is more likely that you will call a procedure in a linear algebra library. This procedure is very fast to calculate.

Gradient Descent

- When there are one or more inputs you can use a process of optimizing the values of the coefficients *by iteratively minimizing the error of the model on your training data*.
- This operation is called *Gradient Descent* and works by starting with random values for each coefficient.
- The sum of the squared errors is calculated for each pair of input and output values. A learning rate is used as a scale factor and the coefficients are updated in the direction towards minimizing the error. The process is repeated until a minimum sum squared error is achieved or no further improvement is possible.
- When using this method, you must select a learning rate (alpha) parameter that determines the size of the improvement step to take on each iteration of the procedure.
- Gradient descent is often taught using a linear regression model because it is relatively straightforward to understand. In practice, it is useful when you have a very large dataset either in the number of rows or the number of columns that may not fit into memory.

Regularization

- There are extensions of the training of the linear model called regularization methods. These seek to both minimize the sum of the squared error of the model on the training data but also to reduce the complexity of the model (like the number or absolute size of the sum of all coefficients in the model).

https://machinelearningmastery.com/linear-regression-for-machine-learning/

Στοχαστική διαβαθμισμένη κάθοδος (Stochastic Gradient Decent)

- Γίνεται διόρθωση βαρών μετά από χρήση <u>ενός μόνο παραδείγματος</u> εκπαίδευσης
  - Ξεκινώντας από ένα τυχαίο διάνυσμα βαρών, η διόρθωση σε αυτά εξαιτίας ενός μόνο παραδείγματος εκπαίδευσης θα είναι:

$$\Delta w_i = -a \cdot \frac{\partial}{\partial w_i} Loss(\boldsymbol{w})$$

  - όπου ο συντελεστής *a* ονομάζεται ρυθμός μάθησης (*learning rate*) και ελέγχει το ρυθμό διόρθωσης.

$$\Delta w_0 = a \cdot \left(y - f_{\boldsymbol{w}}(x)\right) \quad \kappa\alpha\iota \quad \Delta w_1 = a \cdot (y - f_{\boldsymbol{w}}(x)) \cdot x$$

- Άρα, ανάλογα με το αν η $f_{\boldsymbol{w}}(x)$ υπερβαίνει ή υπολείπεται του y, τα βάρη τροποποιούνται έτσι ώστε αυτή η διαφορά να περιοριστεί.
  - Η διαδικασία εφαρμόζεται για κάθε δεδομένο εκπαίδευσης και όταν αυτά εξαντληθούν, <u>ο κύκλος επαναλαμβάνεται</u> μέχρι να επιτευχθεί η σύγκλιση του διανύσματος βαρών στο διάνυσμα που αντιστοιχεί στο ελάχιστο της Loss(w). Πρακτικά αυτό σημαίνει ότι θα πάψει να μειώνεται η Loss(w).
- Η προσέγγιση αυτή είναι κατά βάση γρήγορη, αν και τυχόν θόρυβος στα δεδομένα μπορεί προς στιγμή να μεταβάλει τα βάρη προς λάθος κατεύθυνση.
  - Η διαδρομή προς το ελάχιστο εξαρτάται από τη σειρά χρήσης των δεδομένων εκπαίδευσης και γι' αυτό το λόγο ενδείκνυται η χρήση των δεδομένων εκπαίδευσης με τυχαία σειρά ώστε να διερευνηθούν εναλλακτικά μονοπάτια προς το ελάχιστο
  - Σχετικά μεγάλη τιμή του ρυθμού μάθησης a δεν εγγυάται σύγκλιση στο ελάχιστο
    - ✓ Αντιμετωπίζεται με τη χρήση *φθίνοντος ρυθμού μάθησης* (*decaying learning rate*) που ξεκινά με μεγάλες τιμές a στο αρχικό στάδιο (a_0) τις οποίες μειώνει με ρυθμό dRate (decay rate) σε κάθε κύκλο εκπαίδευσης (epochNo) και καταλήγει με μικρές στο τελικό σύμφωνα με τη σχέση: $a = \frac{1}{1+dRate \cdot epochNo} \cdot a_0$

Διαβαθμισμένη κάθοδος δέσμης (batch gradient decent)

- Ένας άλλος τρόπος χρήσης του συνόλου των N δεδομένων εκπαίδευσης είναι να υπολογιστεί η συνολική διόρθωση στα βάρη με βάση το συνολικό σφάλμα $\sum(y_i - f_{\boldsymbol{w}}(x_i))$.
- Οι προηγούμενες σχέσεις διόρθωσης των βαρών τώρα γίνονται:

$$\Delta w_0 = a \cdot \sum\left(y_i - f_{\boldsymbol{w}}(x_i)\right) \quad \kappa\alpha\iota \quad \Delta w_1 = a \cdot \sum(y_i - f_{\boldsymbol{w}}(x_i)) \cdot {x_i}^2$$

- Εφόσον ο ρυθμός μάθησης είναι επαρκώς μικρός, η κάθοδος στο μοναδικό ελάχιστο είναι εξασφαλισμένη αν και μπορεί να πάρει χρόνο (δηλ. να απαιτηθούν πολλοί κύκλοι).
  - Όπως και στην στοχαστική διαβαθμισμένη κάθοδο, έτσι και εδώ, η χρήση φθίνοντος ρυθμού μάθησης μπορεί να περιορίσει τον απαιτούμενο χρόνο.

[2] Διαφορικός λογισμός

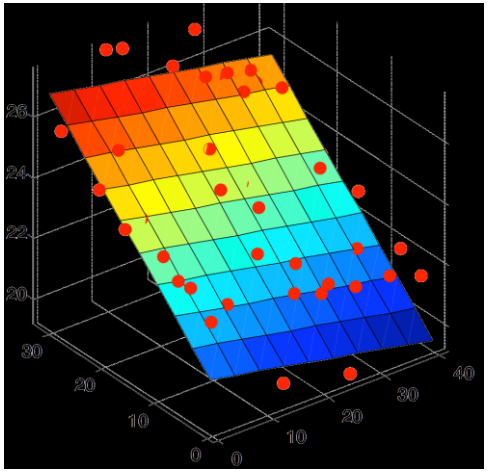
SGDClassifier

- This estimator implements regularized linear models with stochastic gradient descent (SGD) learning: the gradient of the loss is estimated each sample at a time and the model is updated along the way with a decreasing strength schedule (aka learning rate).
- SGD allows minibatch (online/out-of-core) learning via the partial_fit method. For best results using the default learning rate schedule, the data should have zero mean and unit variance.
- The regularizer is a penalty added to the loss function that shrinks model parameters towards the zero vector using either the squared euclidean norm L2 or the absolute norm L1 or a combination of both (Elastic Net).
- class sklearn.linear_model.SGDClassifier(loss='hinge', *, penalty='l2', alpha=0.0001, l1_ratio=0.15, fit_intercept=True, max_iter=1000, tol=0.001, shuffle=True, verbose=0, epsilon=0.1, n_jobs=None, random_state=None, learning_rate='optimal', eta0=0.0, power_t=0.5, early_stopping=False, validation_fraction=0.1, n_iter_no_change=5, class_weight=None, warm_start=False, average=False)

Πολλαπλή Παρεμβολή

❖ Στην πολλαπλή ή πολυπαραμετρική γραμμική παρεμβολή η εξαρτημένη μεταβλητή εξαρτάται από περισσότερες ανεξάρτητες μεταβλητές και η σχέση τους περιγράφεται από μια εξίσωση της μορφής:

$$y = f_{\boldsymbol{w}}(\boldsymbol{x}) = w_0 \cdot x_0 + w_1 \cdot x_1 + \cdots + w_n \cdot x_n = \sum_i w_i \cdot x_i = \boldsymbol{w} \cdot \boldsymbol{x}$$



- ❑ όπου η τελική έκφραση είναι το εσωτερικό γινόμενο του διανύσματος βαρών με το διάνυσμα εισόδου, (για ευκολία θεωρήθηκε ότι ισχύει πάντα x_0=1).
  - ✓ Ο δείκτης i τώρα διατρέχει τις n παραμέτρους εισόδου.
- ❑ Η εξίσωση της ευθείας τώρα επεκτείνεται σε επίπεδο (όταν έχουμε 2 ανεξάρτητες μεταβλητές) ή υπερεπίπεδο (για περισσότερες μεταβλητές).
- ❑ Εφαρμόζοντας την παραπάνω σχέση σε πολλά παραδείγματα εκπαίδευσης προκύπτει ότι το διάνυσμα βαρών w* που ελαχιστοποιεί την *τετραγωνική συνάρτηση απώλειας* (squared or quadratic loss function) είναι:

$$\boldsymbol{w}^* = argmin_{\boldsymbol{w}} \sum_j L_2\left(y_j, \boldsymbol{w} \cdot \boldsymbol{x_j}\right)$$ όπου ο δείκτης j διατρέχει τα δεδομένα εκπαίδευσης

- ❑ Η εύρεση των συνιστωσών w_i του w* μπορεί να γίνει και εδώ με επικλινή κάθοδο:

$$\nabla Loss(\boldsymbol{w})^3 = \left[\frac{\partial L_2}{\partial w_0}, \quad \frac{\partial L_2}{\partial w_1}, \quad \ldots, \quad \frac{\partial L_2}{\partial w_n}\right], \text{ όπου } L_2(\boldsymbol{w}) = \sum_j \left(y_j - \boldsymbol{w} \cdot \boldsymbol{x_j}\right)^2$$

- ❑ Από τις μερικές παραγώγους ως προς w_i της L_2 προκύπτει ότι η σχέση διόρθωσης του βάρους w_i είναι:

$$\Delta w_i = a \cdot \sum_j \left(y_j - \boldsymbol{w} \cdot \boldsymbol{x_j}\right) \cdot x_{j,i}$$ όπου υιοθετήθηκε επικλινής κάθοδος δέσμης πάνω στα δεδομένα εκπαίδευσης

[3] Ανάδελτα

συνέχ.....

- Η παραπάνω μεθοδολογία υπολογισμού των βαρών χρησιμοποιώντας διαβαθμισμένη κάθοδο (batch gradient decent) για τη μείωση του τετραγωνικού σφάλματος εξόδου, είναι γνωστή και ως *Gradient Descent Least Mean Squares* (LMS) και προτάθηκε από τους Widrow και Hoff τη δεκαετία του '60.
- Επιπλέον, συναντάται και στα απλά νευρωνικά δίκτυα τύπου perceptron ως κανόνας εκπαίδευσης Delta

Python (sklearn) class:

- sklearn.linear_model.SGDRegressor(loss=['squared_loss', 'huber', 'Log Loss'], learning_rate=['constant', 'adaptive'], eta0=0.01, tol=1e-3)
  - ✓ SGD stands for Stochastic Gradient Descent
  - ✓ Iterative approach
- sklearn.linear_model.LinearRegression()
  - ✓ Analytical approach (Statistics)
  - ✓ Ordinary Least Squares

What's the Difference Between LMS and Gradient Descent Adaptation?

- Gradient descent (GD) just refers to the method used to hunt for the minimum-cost solution; it doesn't force the use of any particular cost function.
- Stochastic Gradient Descent (SGD) would refer to techniques that don't directly compute the gradient, instead using an approximation to the gradient that has similar stochastic properties (e.g. the same expected value).
  - ✓ The approximation refers to a batch version of the update rule, rather than a full data-pass update
  - ✓ Semi-batch update variants of the GD algorithm also exist
  - ✓ GD and SGD methods are widely used in ML algorithms, such as (Deep) Neural Networks
- LMS is a specialization of gradient descent that uses a mean-squared error cost function and an approximation for the gradient at each time step:
- LMS: GD, SGD or both?

Ομαλοποίηση (regularization)

- Ένα πρόβλημα στην πολυπαραμετρική γραμμική παρεμβολή είναι το να φαίνεται σημαντική κάποια διάσταση (μεταβλητή) που πιθανώς να μην σχετίζεται στην πραγματικότητα με την έξοδο.
  - Αυτό οδηγεί σε κατάσταση *υπερπροσαρμογής* (*overfitting*) όπου η υπό διαμόρφωση συνάρτηση f αδυνατεί να μοντελοποιήσει επαρκώς τη γενική σχέση μεταξύ εισόδου/εξόδου.
- Για τον περιορισμό της υπερπροσαρμογής, υιοθετείται στη συνάρτηση κόστους μια επιπλέον παράμετρος που "τιμωρεί" κατά κάποιο τρόπο τα μεγάλα βάρη:

$$Cost(f) = Loss(f) + \lambda \cdot Complexity(f)$$

  - όπου λ είναι μια σταθερά που εξασφαλίζει παρόμοια τάξη μεγέθους στους δύο όρους, ώστε να μην επισκιάζει συστηματικά ο μεγάλος τον μικρό και πρακτικά να τον καταργεί.
  - Η χρήση αυτού του επιπλέον όρου ονομάζεται *ομαλοποίηση* (*regularization*)
- Ο παράγοντας ομαλοποίησης $Complexity(f_w)$ θα πρέπει να έχει τέτοια μορφή που να επιβαρύνει τη συνάρτηση κόστους όταν τα βάρη παίρνουν μεγάλες τιμές
  - γιατί τότε η αντίστοιχη διάσταση στα δεδομένα του προβλήματος θα συμβάλει στη αύξηση του *κόστους*
- Η μορφή αυτού του παράγοντα εξαρτάται από το χώρο υποθέσεων και μια καλή (και απλή) τέτοια έκφραση για πολυωνυμικές συναρτήσεις f είναι η $\Sigma|w_i|^n$.
  - Για n=1 προκύπτει η ομαλοποίηση L_1 (*L_1 norm*[4] regularization ή *lasso*)[5] ενώ
  - για n=2 η ομαλοποίηση L_2 (*L_2 norm* regularization ή *ridge*).

[4] Vector Norms in Machine Learning
[5] Intuitions on L1 and L2 Regularisation

- Η L_1 ομαλοποίηση έχει καλύτερη ικανότητα στο να μηδενίζει κάποια βάρη και έτσι να αφήνει πρακτικά εκτός προβλήματος κάποιες διαστάσεις στα δεδομένα εισόδου, οδηγώντας σε απλούστερα μοντέλα.
  - Τα απλούστερα μοντέλα είναι λιγότερο ευαίσθητα στις μεταβολές της εισόδου καθώς αυτές δεν πολλαπλασιάζονται με μεγάλα βάρη (άρα έχουν χαμηλή *διακύμανση*) και γενικά αποδίδουν καλύτερα.
- Elastic Net (L1+L2 norm regularization)
- Παράδειγμα: αν η *συνάρτηση απώλειας* (*loss*) είναι το Άθροισμα των τετραγώνων των αποκλίσεων (residual sum of squares - RSS) που ορίζεται ως[6]

$$\text{RSS ή SSE} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} e_i^2$$

- Με την ομαλοποίηση μετασχηματίζεται στην: $Cost(f) = RSS + \lambda \cdot \sum |w_i|^n$
- Οπότε τώρα
  - RSS is modified by adding a shrinkage quantity
  - the coefficients are estimated by minimizing this function
  - λ is the tuning parameter that decides how much we want to penalize the flexibility of our model
  - Selecting (with Cross validation?) a good value of λ is critical
  - Για n=1 προκύπτει η ομαλοποίηση L_1 (*L_1 norm* ή *lasso*) ενώ για n=2 η ομαλοποίηση L_2 (*L_2 norm* ή *ridge*).
  - Η διαφορά τους είναι στο πόσο "τιμωρούν" τα μεγάλα βάρη
  - Python command (sklearn): sklearn.linear_model.Lasso(alpha=1.0)

[6] Περισσότερα για μετρικές στο κεφάλαιο 11: Αξιολόγηση Μοντέλων

- ✓ Alpha is λ

- [Regularization in Machine Learning]

Παράδειγμα

- Έστω ότι θέλουμε να προβλέψουμε την αξία ενός ακινήτου με βάση την επιφάνειά του και τον όροφο στον οποίο βρίσκεται.
- Θεωρώντας μια γραμμική εξάρτηση της αξίας του ακινήτου σε σχέση με το εμβαδό του και τον όροφο στον οποίο βρίσκεται, το ζητούμενο μοντέλο θα είναι της μορφής:

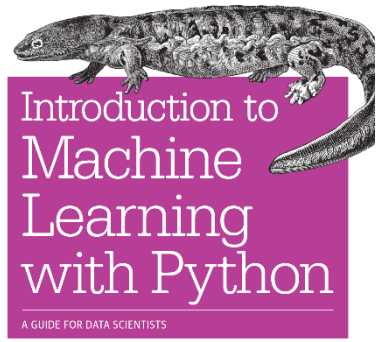
$$y=w_o+w_1\cdot x_1+w_2\cdot x_2$$

  - δηλαδή έχουμε πολυπαραμετρική γραμμική παρεμβολή στην οποία x_1 είναι το εμβαδό και x_2 ο όροφος.

- Χρησιμοποιώντας τα δεδομένα που έχει ο Πίνακας και τις τεχνικές που παρουσιάστηκαν στις προηγούμενες ενότητες, θα υπολογιστούν οι καλύτερες τιμές των βαρών w_i που ελαχιστοποιούν τη συνάρτηση απώλειας.
- Έχοντας υπολογίσει την παραπάνω εξίσωση θα μπορούμε να κάνουμε την επιθυμητή πρόβλεψη, δηλαδή δοθέντων των τιμών x_1 και x_2 ενός νέου ακινήτου ($\boldsymbol{x}\in\Re^2$) θα υπολογίζεται η αξία του y ($y\in\Re$).

No	x_1 Εμβαδό (m^2)	x_2 Όροφος	y Αξία (x1000 €)
1	100	2	170
2	120	2	160
3	80	2	105
4	90	3	170
5	100	4	200
6	120	3	210
7	110	1	140
8	130	1	180
9	110	3	225
10	150	1	170

Representation input data



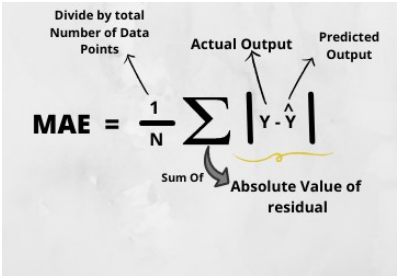
- For both supervised and unsupervised learning tasks, it is important to have a representation of your input data that a computer can understand.
  - Often it is helpful to think of your data as a table.
- Each data point that you want to reason about (each email, each customer, each transaction) is a row, and each property that describes that data point (say, the age of a customer or the amount or location of a transaction) is a column.
  - You might describe users by their age, their gender, when they created an account, and how often they have bought from your online shop.
  - You might describe the image of a tumor by the grayscale values of each pixel, or maybe by using the size, shape, and color of the tumor.
- Each entity or row here is known as a sample (or data point) in machine learning, while the columns—the properties that describe these entities—are called features.
- The topic of building a good representation of our data, is called feature extraction or feature engineering.
- No machine learning algorithm will be able to make a prediction on data for which it has no information.

Preparing Data for Linear Regression

- There is a lot of literature on how your data must be structured to make best use of the model. In practice, you can uses these rules more as rules of thumb
  - **Linear Assumption**. Linear regression assumes that the relationship between your input and output is linear. It does not support anything else. This may be obvious, but it is good to remember when you have a lot of attributes.
    - ✓ You may need to transform data to make the relationship linear (e.g. log transform for an exponential relationship).
  - **Remove Noise**. Linear regression assumes that your input and output variables are not noisy. Consider using data cleaning operations that let you better expose and clarify the signal in your data.
    - ✓ This is most important for the output variable and you want to remove outliers in the output variable (y) if possible.
  - **Remove Collinearity**. Linear regression will over-fit your data when you have highly correlated input variables. Consider calculating pairwise correlations for your input data and removing the most correlated.
  - **Gaussian Distributions**. Linear regression will make more reliable predictions if your input and output variables have a Gaussian distribution.
    - ✓ You may get some benefit using transforms (e.g. log or Box-Cox) on you variables to make their distribution more Gaussian looking
  - **Rescale Inputs**: Linear regression will often make more reliable predictions if you rescale input variables using (standardization or) normalization
    - ✓ Standardization (or Z-score normalization)

Evaluation of Regression Models

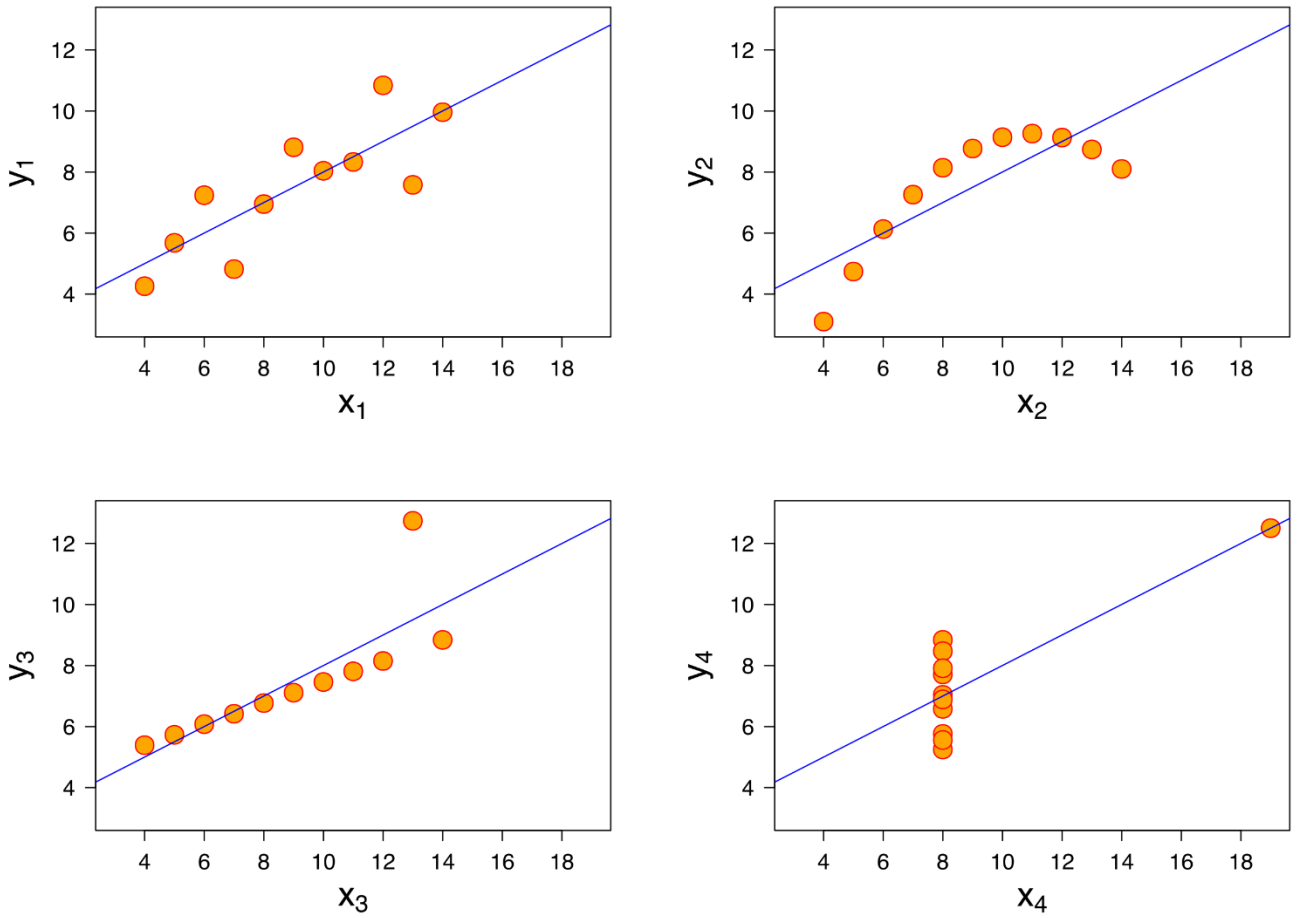
- Performance metrics are vital for supervised machine learning models
  - To be sure that your model is doing well in its predictions, you need to evaluate the model
    - ✓ i.e.to identify how well the model performs on new data
  - There are many evaluation metrics. All of these are loss functions, because we want to minimize them
  - Examples:
    - ✓ Mean Absolute Error (MAE) $\mathrm{MAE} = \frac{1}{\mathrm{n}}\sum_{t=1}^{\mathrm{n}}\left|\mathrm{Y_t} - \widehat{\mathrm{Y}}_\mathrm{t}\right| = \frac{1}{\mathrm{n}}\sum_{\mathrm{t}=1}^{\mathrm{n}}|\mathrm{e_t}|$



    - ✓ Residual Sum of Squares (RSS) ή (sum of squares error - SSE) $\mathrm{RSS\ ή\ SSE} = \sum_{\mathrm{t}=1}^{\mathrm{n}}\left(\mathrm{Y_i} - \widehat{\mathrm{Y}}_\mathrm{i}\right)^2 = \sum_{\mathrm{t}=1}^{\mathrm{n}}\mathrm{e_t^2}$
    - ✓ Mean Squared Error (MSE) $\mathrm{MSE} = \frac{1}{\mathrm{n}}\mathrm{RSS} = \frac{1}{n}\cdot\sum_{t=1}^{n} e_t^2$
    - ✓ Huber Loss: Υβριδική μετρική η οποία συνδυάζει τα πλεονεκτήματα του MAE και του MSE.
    - ✓ Root Mean Squared Error(RMSE)
    - ✓ Root Mean Squared Log Error(RMSLE)
    - ✓ R Squared (R2)

    - ✓ Know The Best Evaluation Metrics for Your Regression Model

Παραδείγματα αποτελεσματικότητας παρεμβολής

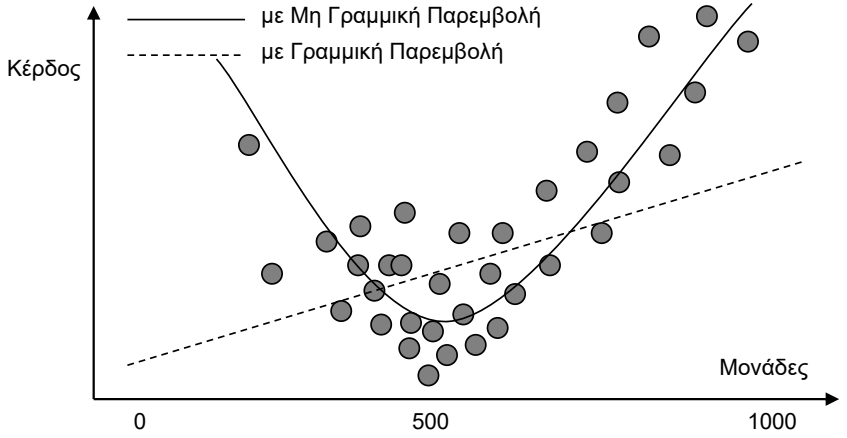


What's the Difference Between a Metric and a Loss Function?

- Loss functions and metrics have different purposes
  - *Metrics* evaluate the performance of the final model and compare the performance of different models
  - *Loss functions* are used during the model-building phase as an *optimizer* for the model under creation
    - ✓ Loss functions guide the model on how to minimize error
    - ✓ Are calculated for each individual observation (each sample)
    - ✓ It is also sometimes called an *error function*
  - The function that averages the values of all loss functions (the entire training dataset) is called the *Cost Function*
  - The optimization strategies aim at minimizing the cost function
- if you are using Python, you can use
  - sklearn.metrics or tensorflow.keras.metrics for metrics and
  - sklearn.losses or tensorflow.keras.losses for loss functions
- Alternatively, you can also define your own metrics and loss functions using mathematical expressions or functions
- So we need two separate model scoring functions for evaluation and optimization… and possibly a third one for statistical testing
  - Statistical testing examines if the model is good enough for us to use. In other words, does the model pass our rigorous hypothesis testing criteria?

✓ Understanding Loss Functions to Maximize Machine Learning Model Performance (Updated 2023)

Μη Γραμμικά Μοντέλα – Πολυωνυμική Παρεμβολή



- Στα μη γραμμικά μοντέλα παρεμβολής (βλ. παράδειγμα στο Σχήμα) η αναμενόμενη τιμή εξόδου συνδέεται με τις παραμέτρους εισόδου με πιο πολύπλοκο τρόπο καθώς υπεισέρχονται εκθετικές, λογαριθμικές, κτλ. εκφράσεις.
  - όπως για παράδειγμα: $y_i = \beta_0 x_{1j}^{\beta 1} \ldots j = 1,2,\ldots,m$
  - Στο Σχήμα, η μοντελοποίηση των δεδομένων με γραμμική παρεμβολή δίνει μεγάλο σφάλμα (διακεκομμένη γραμμή) ενώ η χρήση μη γραμμικών τεχνικών μοντελοποιεί καλύτερα τα δεδομένα (συνεχής γραμμή).
- Στην *πολυωνυμική* παρεμβολή, που είναι μια περίπτωση μη γραμμικής παλινδρόμησης, η σχέση μεταξύ της εξαρτημένης μεταβλητής και των ανεξάρτητων μεταβλητών περιγράφεται με ένα πολυώνυμο n-οστού βαθμού της μορφής:

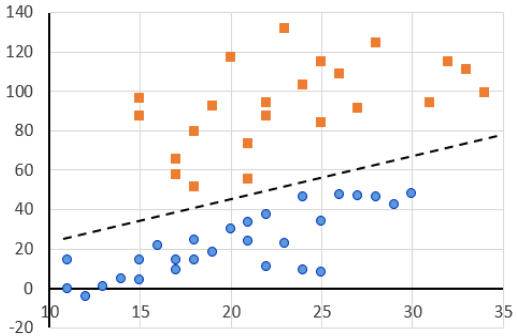
$$y = f(x) = w_0 + w_1 x + w_2 x^2 + \cdots + w_n x^n$$

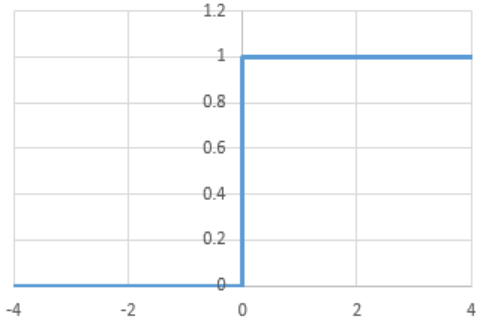
  - Η προσέγγιση άγνωστων συναρτήσεων με πολυώνυμο (πολυωνυμική παρεμβολή) είναι πάντα δυνατή και με όση ακρίβεια απαιτείται.
  - Επιπλέον πλεονέκτημα είναι ότι η παράγωγος και το ολοκλήρωμά τους υπολογίζονται εύκολα και είναι επίσης πολυώνυμα.
- Η πολυωνυμική παρεμβολή είναι ένα καλά μελετημένο μαθηματικό πρόβλημα και στη βιβλιογραφία υπάρχουν διάφορες μέθοδοι υπολογισμού πολυωνύμων παρεμβολής, όπως για παράδειγμα η παρεμβολή Newton, η παρεμβολή Lagrange και η μέθοδος ελαχίστων τετραγώνων.

- Σε κάποιες περιπτώσεις, τα μη γραμμικά μοντέλα μπορούν να μετατραπούν σε γραμμικά με κατάλληλο μετασχηματισμό των μεταβλητών, ώστε τελικά να επιλυθούν με τη μέθοδο των ελαχίστων τετραγώνων.
- Οι μη γραμμικές τεχνικές δεν περιορίζονται μόνο στην πολυωνυμική παρεμβολή αλλά υπάρχουν και άλλες πολύ αξιόλογες τεχνικές, όπως οι μηχανές διανυσμάτων υποστήριξης (Support Vector Machines - SVMs) και τα τεχνητά νευρωνικά δίκτυα (ΤΝΔ)

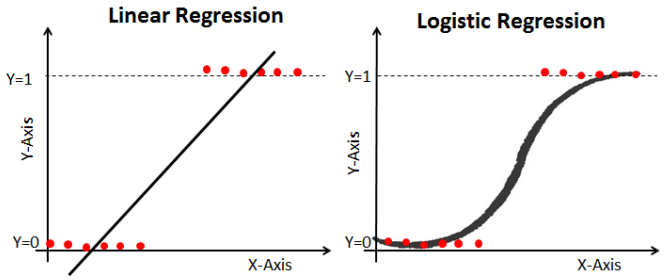
Γραμμικοί Ταξινομητές (Classification)

- Τα γραμμικά μοντέλα παρεμβολής που εξετάστηκαν προηγούμενα μπορούν να χρησιμοποιηθούν και σε προβλήματα ταξινόμησης, δηλαδή η έξοδος να είναι διακριτή και όχι συνεχής.

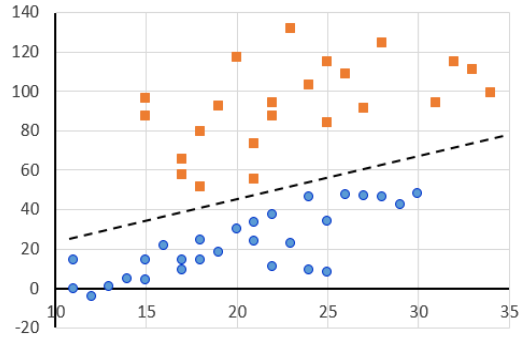




- Το ζητούμενο εδώ είναι ο προσδιορισμός της κλάσης
  - Δηλαδή όταν δοθούν ως είσοδος στο διάγραμμα δύο συντεταγμένες, η απάντηση είναι για παράδειγμα 0 ή 1, όπου κάθε τιμή αντιστοιχεί σε μία κλάση.



Χρήση συνάρτησης κατωφλίου



- Η ευθεία-σύνορο των δύο κλάσεων στο διάγραμμα, είναι της μορφής:

$$x_2 = w_1 \cdot x_1 + w_0 \quad \Rightarrow \quad w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 = 0 \quad \Rightarrow \quad \mathbf{w} \cdot \mathbf{x} = 0$$

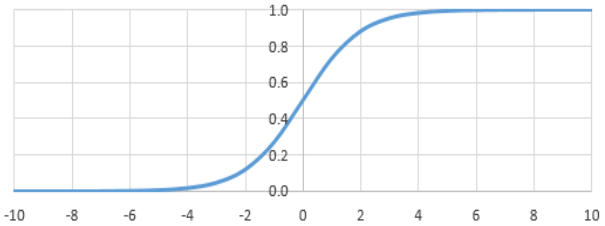
- Η ανισότητα $\mathbf{w} \cdot \mathbf{x} \geq 0$ ορίζει το γεωμετρικό τόπο των σημείων που βρίσκονται πάνω ή κάτω από την ευθεία $\mathbf{w} \cdot \mathbf{x} = 0$ (κλάση "κύκλος ή "τετράγωνο").
- Για να έχουμε ως αποτέλεσμα 0 (τετράγωνο) ή 1 (κύκλος), αρκεί να περάσει η σχέση $\mathbf{w} \cdot \mathbf{x}$ μέσα από μια συνάρτηση κατωφλίου $f(\mathbf{w} \cdot \mathbf{x})$ με κατώφλι στην τιμή 0.
  - Για $\mathbf{w} \cdot \mathbf{x} < 0$ η f θα δίνει πάντα έξοδο 0 ενώ για $\mathbf{w} \cdot \mathbf{x} \geq 0$ θα δίνει πάντα 1.
  - Το πρόβλημα όμως είναι ότι τα βάρη w_i δεν είναι δυνατό να υπολογιστούν με τους τρόπους που αναφέρθηκαν στη γραμμική παρεμβολή, γιατί στη συνάρτηση f (κατωφλίου) η πρώτη παράγωγος είναι μηδέν παντού εκτός από το κατώφλι, ενώ στο κατώφλι δεν ορίζεται.
- Ένας απλός κανόνας διόρθωσης των βαρών, που για γραμμικώς διαχωρίσιμα προβλήματα συγκλίνει σε λύση (δηλ. βρίσκει τα w_i), έχει τη μορφή:

$$\Delta w_i = a \cdot \left(y - f(\mathbf{w} \cdot \mathbf{x})\right) \cdot x_i \qquad \text{όπου } a = \rho υθμός\ μάθησης$$

  - Πρακτικά είναι η σχέση διόρθωσης της πολυπαραμετρικής γραμμικής παρεμβολής.
  - Η διαδικασία εφαρμόζεται επιλέγοντας τυχαία παραδείγματα από τα δεδομένα εκπαίδευσης και επαναλαμβάνοντας μέχρι η απόδοση του μοντέλου (της f) να φτάσει το 100%, το οποίο είναι εφικτό διότι οι κλάσεις είναι γραμμικώς διαχωρίσιμες.
- Στην περίπτωση μη γραμμικώς διαχωρίσιμων κλάσεων προφανώς δεν υπάρχει τέτοια ευθεία/σύνορο

Λογιστική παρεμβολή (logistic or logit regression)

- Η χρήση της συνάρτησης κατωφλίου για τη δημιουργία ενός ταξινομητή με βάση ένα γραμμικό σύνορο δεν είναι ιδιαίτερα αποδοτική, εξαιτίας των προβλημάτων με την παράγωγο που αναφέρθηκαν
  - Επιπλέον η μέθοδος αυτή προβλέπει πάντα 0 ή 1 κάτι που ίσως δεν είναι επιθυμητό για περιπτώσεις κοντά στο σύνορο των κλάσεων.
- Αυτά τα προβλήματα μπορεί να αντιμετωπιστούν με μια άλλη συνάρτηση, λιγότερο απότομη από τη συνάρτηση κατωφλίου, όπως η λογιστική (logistic or sigmoid) συνάρτηση που εκφράζεται με τη σχέση:



$$f(u) = \frac{1}{1+e^{-u}} = \frac{e^{u}}{e^{u}+1}$$

- Αντικαθιστώντας τη συνάρτηση κατωφλίου με τη λογιστική συνάρτηση, προκύπτει ότι:

$$f(\boldsymbol{w}\cdot\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{w}\cdot\boldsymbol{x}}}$$

- Επομένως η έξοδος της f δεν είναι τώρα 0 ή 1 αλλά και οι ενδιάμεσες τιμές.
  - Καθώς μάλιστα οι τιμές είναι μεταξύ 0 και 1, μπορεί να θεωρηθεί η έξοδος της f ως η πιθανότητα της περίπτωσης x να ανήκει σε μια κλάση
  - Καθώς τώρα η f είναι παραγωγίσιμη μπορεί να χρησιμοποιηθεί η τεχνική της επικλινούς καθόδου (Gradient Descent), με συνάρτηση απώλειας βασισμένη στην L2
- Η διαδικασία υπολογισμού του διανύσματος των βαρών w, έτσι ώστε το μοντέλο που προκύπτει να ελαχιστοποιεί τη συνάρτηση απώλειας, ονομάζεται *λογιστική παρεμβολή* (*logistic or logit regression*).
  - Σε γραμμικώς διαχωρίσιμα προβλήματα η παραπάνω διαδικασία συγκλίνει σε λύση ενώ σε μη γραμμικώς διαχωρίσιμα προβλήματα η μέθοδος συγκλίνει σε λύση ελάχιστου αλλά όχι μηδενικού σφάλματος.

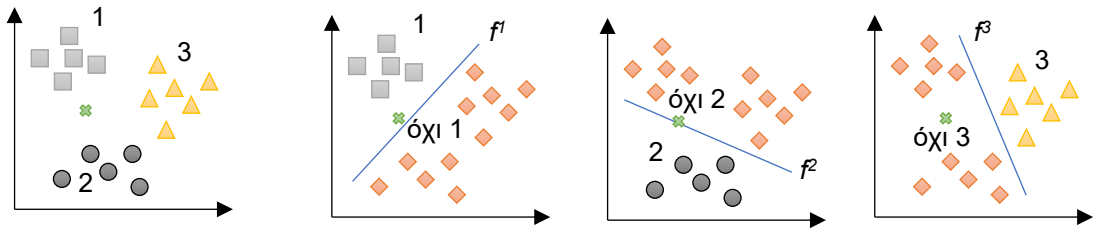
- Η λογιστική παρεμβολή είναι μία από τις πιο δημοφιλείς τεχνικές ταξινόμησης και χρησιμοποιείται ευρέως σε τομείς όπως ιατρική, οικονομία, μάρκετινγκ, κτλ.
- Python command (sklearn):
  - sklearn.linear_model.LogisticRegression(solver=['liblinear', 'newton-cg'], tol=1e-4)

- Why are we calling a classification model the Logistic 'Regression'?
  - The reason behind this is that just like Linear Regression, logistic regression starts from a linear equation.
  - However, this equation consists of log-odds which is further passed through a sigmoid function which squeezes the output of the linear equation to a probability between 0 and 1.
  - And, we can decide a decision boundary and use this probability to conduct classification task.
  - Logistic regression is majorly used for binary classification tasks; however, it can be used for multiclass classification

    ✓ The math behind Logistic Regression

Ταξινόμηση πολλαπλών κλάσεων

- Αν και έως τώρα έγινε αναφορά μόνο σε προβλήματα ταξινόμησης δύο κλάσεων (binary classification), οι μεθοδολογίες μπορούν να χρησιμοποιηθούν και σε ταξινόμηση σε περισσότερες από δύο κλάσεις.
- Σε τέτοιες περιπτώσεις ακολουθείται συνήθως η τεχνική του ενός εναντίον όλων (one-vs-all ή one-vs-rest) σύμφωνα με την οποία ένα πρόβλημα ταξινόμησης n κλάσεων μετατρέπεται σε n προβλήματα ταξινόμησης δύο κλάσεων.
- Έχοντας λύσει (π.χ. με λογιστική παρεμβολή) τα επιμέρους προβλήματα δυαδικής ταξινόμησης, για μια νέα περίπτωση ***x*** του αρχικού προβλήματος υπολογίζεται η τιμή της λογιστικής συνάρτησης f_i σε κάθε μία από τις *n* εκδοχές της και νικήτρια είναι η κλάση με τη μεγαλύτερη τιμή.
- Για παράδειγμα, για το σημείο *x* στο παρακάτω σχήμα, θεωρώντας ότι τα επιμέρους προβλήματα δυαδικής ταξινόμησης δίνουν:
  - f1(x)=0.7 f2(x)=0.5 f3(x)=0
  - η νέα περίπτωση x θα ανήκε στην κλάση "1".



Μετρικές Αξιολόγησης Ταξινόμησης

- Έστω ένας δυαδικός ταξινομητής με αποτελέσματα που χαρακτηρίζονται ως θετικά (positive - p) ή αρνητικά (negative – n).
- Έχουμε 4 περιπτώσεις αποτελεσμάτων που μπορούν να περιγραφούν σε έναν 2x2 πίνακα ενδεχομένων (*contingency matrix*) ή πίνακα σύγχυσης (confusion matrix)
  - Είναι ένας τρόπος παρουσίασης <u>των επιδόσεων ανά κλάση</u> ενός ταξινομητή

		Προβλεπόμενη Κλάση (Predicted class)	
		Class= Positive	*Class= Negative*
Πραγματική Κλάση (Actual Class)	*Class= Positive*	TP (True Positive)	FN (False Negative)
	Class=Negative	FP (False Positive)	TN (True Negative)

- Για παράδειγμα σε ένα διαγνωστικό τεστ για την εξακρίβωση μιας πάθησης
  - FP: το τέστ είναι θετικό αλλά στην πραγματικότητα ο ασθενής δεν έχει την πάθηση
  - FN: Το τέστ είναι αρνητικό, αλλά ο ασθενής έχει την πάθηση
- Από έναν πίνακα ενδεχομένων μπορούν να παραχθούν αρκετά μέτρα εκτίμησης

Μέτρα Εκτίμησης

- *Accuracy* (*Ακρίβεια*): το ποσοστό των παραδειγμάτων που ταξινομούνται σωστά:

$$\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$$

  - Ευκολονόητη μετρική που όμως σε πολλές περιπτώσεις δίνει παραπλανητική πληροφόρηση και για αυτό είναι χρήσιμη μόνο σε συνδυασμό με τις άλλες μετρικές

- *Precision* (P) (*Ευστοχία*) ή positive predictive value (PPV):
  - Ποσοστό παραδειγμάτων που ταξινομούνται ως θετικά και είναι σωστά (θετικά): $\mathrm{P}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$
  - Σημαντική μετρική όταν θέλουμε να είμαστε πολύ σίγουροι για την πρόβλεψη μας

- True Positive Rate (TPR) ή Sensitivity (ευαισθησία): $\mathrm{TPR}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$
  - Το ποσοστό των θετικών παραδειγμάτων που βρήκε ο ταξινομητής
  - Ισοδύναμο με την ανάκληση (Recall - r)
  - χρήσιμη μετρική όταν θέλουμε ο ταξινομητής μας να "πιάνει" όσο το δυνατόν περισσότερα θετικά παραδείγματα ακόμη και αν δεν είναι πολύ σίγουρος.
    - ✓ Για παράδειγμα σε ένα σύστημα πρόβλεψης μια ασθένειας

- Επειδή τα κριτήρια *Precision* και *Recall* δεν αρκούν από μόνα τους για να περιγράψουν την συνολική επίδοση του ταξινομητή συνήθως συνδυάζονται στο κριτήριο *F-measure*
  - ✓ (ή αλλιώς *F1-score*) sklearn.metrics.f1_score
  - Είναι ο αρμονικός μέσος (harmonic average) της ακρίβειας (precision) και της ανάκλησης (recall)
    - ✓ Δηλ. το πηλίκο του γεωμετρικού μέσου προς το αλγεβρικό μέσο όρο των δύο κριτηρίων

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

- Καλύτερη τιμή1 (ιδανικό precision και recall) και χειρότερη το 0
  - ✓ Ονομάζεται και *balanced F-score* ή F_1 measure, γιατί τα recall και precision συμμετέχουν ισότιμα

Example: Regression

Data set: Relative CPU Performance Data

Weka file name: cpu.arff

Creators: Phillip Ein-Dor and Jacob Feldmesser

Number of Instances: 209

Missing Attribute Values: None

Attributes

1. MYCT: machine cycle time in nanoseconds (numeric)
2. MMIN: minimum main memory in kilobytes (numeric)
3. MMAX: maximum main memory in kilobytes (numeric)
4. CACH: cache memory in kilobytes (numeric)
5. CHMIN: minimum channels in units (numeric)
6. numeric PRP: published relative performance (numeric)
7. **Class**: ERP: estimated relative performance from the original article (numeric)

@data
125,256,6000,256,16,128,198
29,8000,32000,32,8,32,269
29,8000,32000,32,8,32,220
29,8000,32000,32,8,32,172
29,8000,16000,32,8,16,132
26,8000,32000,64,8,32,318
23,16000,32000,64,16,32,367
23,16000,32000,64,16,32,489
……
…….

Weka Algorithm: **LinearRegression**

Total Number of Instances 209

Evaluation Metrics

Correlation coefficient	0.9012
Mean absolute error	41.0886
Root mean squared error	69.556
Relative absolute error	42.6943 %
Root relative squared error	43.2421 %

❖ Το ίδιο παράδειγμα με Random Forest:

❑ Evaluation Metrics

Correlation coefficient	0.9532
Mean absolute error	25.6115
Root mean squared error	51.4883
Relative absolute error	26.6123 %
Root relative squared error	32.0096 %

Example: Classification

Data set: Breast cancer Data

Weka file name: cpu.arff

Creators: Matjaz Zwitter & Milan Soklic (physicians), Institute of Oncology

University Medical Center, Ljubljana, Slovenia

Number of Instances: 286

Number of Attributes: 9 + the class attribute

Missing Attribute Values: None

Attributes

1. age {'10-19','20-29','30-39','40-49','50-59','60-69','70-79','80-89','90-99'}
2. menopause {'lt40','ge40','premeno'}
3. tumor-size {'0-4','5-9','10-14','15-19','20-24','25-29','30-34','35-39','40-44','45-49','50-54','55-59'}
4. inv-nodes {'0-2','3-5','6-8','9-11','12-14','15-17','18-20','21-23','24-26','27-29','30-32','33-35','36-39'}
5. node-caps {'yes','no'}
6. deg-malig {'1','2','3'}
7. breast {'left','right'}
8. breast-quad {'left_up','left_low','right_up','right_low','central'}
9. 'irradiat' {'yes','no'}
10. 'Class' {'no-recurrence-events','recurrence-events'}

@data

'40-49','premeno','15-19','0-2','yes','3','right','left_up','no','recurrence-events'
'50-59','ge40','15-19','0-2','no','1','right','central','no','no-recurrence-events'
'50-59','ge40','35-39','0-2','no','2','left','left_low','no','recurrence-events'
'40-49','premeno','35-39','0-2','yes','3','right','left_low','yes','no-recurrence-events'
'40-49','premeno','30-34','3-5','yes','2','left','right_up','no','recurrence-events'
'50-59','premeno','25-29','3-5','no','2','right','left_up','yes','no-recurrence-events'
……..
……..

Weka Algorithm: **Logistic**

Total Number of Instances 286

Evaluation Metrics

Correctly Classified Instances	197	68.8811 %
Incorrectly Classified Instances	89	31.1189 %
Precision	0,752	
Recall	0,831	
F-Measure	0,790	

Questions?