



ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Ανασκόπηση στοιχείων

Τζούβελη Παρασκευή

Τανυστές

- Τανυστές μηδενικής τάξης - Βαθμωτά (Scalars)
 - ένας μόνο αριθμός ($s \in \Re$)
- Τανυστές πρώτης τάξης - Διανύσματα (Vectors)
 - μονοδιάστατος πίνακας όπου $x_1 \dots x_n \in \Re$
- Τανυστές δεύτερης τάξης - Πίνακες (Matrices)
 - δισδιάστατοι πίνακες, με $a_{11} \dots a_{nn} \in \Re$ και
 -
- Τανυστές μεγαλύτερης από 2ης τάξης - Τανυστές (Tensors)
 - πολυδιάστατοι πίνακες π.χ. 3ης τάξης όπου $b_{111} \dots b_{kkmn} \in \Re$ και $B \in \Re^{kxmxn}$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$A \in \Re^{m \times n}, A = \begin{bmatrix} a_{11} & a_{12} & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{mn} \end{bmatrix}$$

Ανάστροφο διάνυσμα \mathbf{x}^T και πίνακας A^T

- Ορίζουμε τον ανάστροφο (transpose) του διανύσματος \mathbf{x} και συμβολίζουμε \mathbf{x}^T , ως τον πίνακα γραμμή Ν θέσεων με στοιχεία: $x^T = [x_1 \ x_2 \ \dots \ x_n]$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$(A^T)_{i,j} = (A)_{j,i}$$

- Ορίζουμε τον ανάστροφος πίνακα του πίνακα $A = (a_{ij}) \in M_{m \times n}$ και συμβολίζουμε A^T , ως τον πίνακα που προκύπτει από τον A με εναλλαγή μεταξύ γραμμών και στηλών του:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \\ A_{31} & A_{32} \end{bmatrix} A^T = \begin{bmatrix} A_{11} & A_{21} & A_{31} \\ A_{12} & A_{22} & A_{32} \end{bmatrix}$$

Κύριος διαγώνιος

Ανάστροφος πίνακας A^T

Παραδείγματα

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \dots & \dots & \dots & \dots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{bmatrix}_{m \times n}$$

$$C = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 5 & -6 \\ -4 & 3 & 0 \\ -1 & 8 & 9 \end{bmatrix}_{4 \times 3}$$

$$A^T = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,m} \\ a_{2,1} & a_{2,2} & \dots & a_{2,m} \\ \dots & \dots & \dots & \dots \\ a_{n,1} & a_{n,2} & \dots & a_{n,m} \end{bmatrix}_{n \times m}$$

Ιδιότητες

$$\begin{aligned} (A^T)^T &= A \\ (A + B)^T &= A^T + B^T \\ (AB)^T &= B^T A^T \end{aligned}$$

$$C^T = \begin{bmatrix} 1 & 2 & -4 & -1 \\ 0 & 5 & 3 & 8 \\ -1 & -6 & 0 & 9 \end{bmatrix}_{3 \times 4}$$

Μοναδιαίος Πίνακας Τύπου n , I_n

Ένας διαγώνιος πίνακας $I \in \mathbb{R}^{n \times n}$ ονομάζεται **μοναδιαίος πίνακας** αν και μόνο αν

$$(I_n)_{ij} = \delta_{ij} = \begin{cases} 1, & \text{για } i=j \\ 0, & \text{για } i \neq j \end{cases}$$

Ίχνος Τετραγωνικού Πίνακας

Έστω $A \in \mathbb{R}^{n \times n}$, τότε το **ίχνος του A** ορίζεται ως: $\text{tr}(A) = \text{trace}(A) : \sum_{i=1}^n a_{ii} = a_{11} + a_{22} + \dots + a_{nn}$

δηλαδή ίσο με το άθροισμα των στοιχείων της κύριας διαγωνίου.

Τετραγωνικοί Πίνακες

Ένας πίνακας A είναι **τετραγωνικό τύπου n** όταν έχει διάσταση $n \times n$ π.χ. $A = \begin{pmatrix} 5 & 2 \\ 4 & 6 \end{pmatrix}$ είναι τετραγωνικού τύπου 2.

Ένας **τετραγωνικός πίνακας** με γραμμικά εξαρτημένες στήλες (ή γραμμές) είναι γνωστός ως **μη αντιστρέψιμος** ή **ιδιόμορφος (singular)**.

→ Αν οι στήλες (ή γραμμές) είναι **γραμμικά εξαρτημένες**, τότε η ορίζουσα του είναι 0 και ο πίνακας δεν έχει αντίστροφο.

Ένας **τετραγωνικός πίνακας** με γραμμικά ανεξάρτητες στήλες (ή γραμμές) ονομάζεται **αντιστρέψιμος (non-singular)**.

→ Σε αυτή την περίπτωση, η ορίζουσα του πίνακα είναι διαφορετική από το μηδέν.

Διαγώνιοι Πίνακες

Ένας πίνακας $A \in \mathbb{R}^{n \times n}$ ονομάζεται **διαγώνιος πίνακας**, αν και μόνο αν $a_{ij} = 0$ όταν $i \neq j$ δηλαδή όλα τα στοιχεία **ξέω από την κύρια διαγώνιο** είναι 0

$$\text{π.χ. } A = \begin{pmatrix} 5 & 0 \\ 0 & -3 \end{pmatrix}$$

Κάτω Τριγωνικός Πίνακας

Ένας πίνακας $A \in \mathbb{R}^{n \times n}$ ονομάζεται **κάτω τριγωνικός** αν και μόνο αν $a_{ij} = 0$ όταν $i < j$, δηλαδή όλα τα στοιχεία πάνω από την κύρια διαγώνιο είναι 0

$$\begin{pmatrix} 3 & 0 & 0 \\ 4 & 6 & 0 \\ 5 & 7 & 8 \end{pmatrix}$$

Άνω Τριγωνικός Πίνακας

Ένας πίνακας $A \in \mathbb{R}^{n \times n}$ ονομάζεται **άνω τριγωνικός** αν και μόνο αν $a_{ij} = 0$ όταν $i > j$, δηλαδή όλα τα στοιχεία κάτω από την κύρια διαγώνιο είναι 0

$$\begin{pmatrix} 3 & 4 & 5 \\ 0 & 6 & 7 \\ 0 & 0 & 8 \end{pmatrix}$$

Αντίστροφος πίνακας

Ο πίνακας $A \in \mathbb{R}^{n \times n}$ είναι αντιστρέψιμος εάν υπάρχει ένας πίνακας $B \in \mathbb{R}^{n \times n}$ έτσι ώστε $B \cdot A = I$ και $A \cdot B = I$, όπου I είναι η ίδια μοναδιαίος πίνακας.

Υπάρχει το πολύ ένας τέτοιος B και λέγεται αντίστροφος του A και συμβολίζεται με A^{-1} : $A \cdot A^{-1} = A^{-1} \cdot A = I$

→ Οπότε, για να είναι ο πίνακας $A \in \mathbb{R}^{n \times n}$ αντιστρέψιμος θα πρέπει η ορίζουσά του είναι διάφορη του 0 : $\det A \neq 0$

Ιδιότητες

$$\begin{aligned} AA^{-1} &= I = A^{-1}A \\ (AB)^{-1} &= B^{-1}A^{-1} \\ (A+B)^{-1} &\neq A^{-1} + B^{-1} \end{aligned}$$

Αντίστροφος πίνακας

- Παράδειγμα 1.1 (Υπαρξή αντιστρόφου πίνακα 2×2) Θεωρούμε έναν πίνακα $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \in \mathbb{R}^{2 \times 2}$. Αν τον πολλαπλασιάσουμε με τον πίνακα $B = \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix} \in \mathbb{R}^{2 \times 2}$ τότε θα έχουμε $(a_{11}a_{22} - a_{12}a_{21})I$. Τότε ο αντίστροφος A^{-1} του πίνακα A θα είναι ο

$$A^{-1} = \frac{1}{(a_{11}a_{22} - a_{12}a_{21})} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix} \quad (1.1)$$

αν και μόνο αν $(a_{11}a_{22} - a_{12}a_{21}) \neq 0$ (όπου $(a_{11}a_{22} - a_{12}a_{21}$ είναι η ορίζουσά του πίνακα A). Μπορούμε να χρησιμοποιούμε την ορίζουσά του πίνακα για να αποφανθούμε αν αυτός είναι αντιστρέψιμος.

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad ad - bc \neq 0 \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Συμμετρικοί πίνακες

- Ένας τετραγωνικός πίνακας $S \in \mathbb{R}^{n \times n}$ ο οποίος είναι ίσος με τον ανάστροφό του, δηλαδή $S = S^T$, είναι ένας συμμετρικός πίνακας.

π.χ. εάν A είναι ένας πίνακας μέτρησης αποστάσεων, με $A_{i,j}$ που δίνει την απόσταση από το σημείο i στο σημείο j , τότε $A_{i,j} = A_{j,i}$ επειδή οι συναρτήσεις απόστασης είναι συμμετρικές.

$$\begin{pmatrix} 3 & 5 \\ 5 & -1 \end{pmatrix}$$

$$\begin{pmatrix} 7 & -5 & 2 \\ -5 & 3 & 4 \\ 2 & 4 & 8 \end{pmatrix}$$

$$\begin{pmatrix} 5 & 6 & -4 & -3 \\ 6 & 7 & 2 & 9 \\ -4 & 2 & 1 & 5 \\ -3 & 9 & 5 & 6 \end{pmatrix}$$

Αντισυμμετρικοί πίνακες

- Αν αντίθετα, ο τετραγωνικός πίνακας $S \in \mathbb{R}^{n \times n}$ ο οποίος είναι ίσος με τον αρνητικό του ανάστροφο, δηλαδή $S = -S^T$ τότε λέμε ότι ο S είναι ένας αντισυμμετρικός πίνακας.

Δηλαδή: $(A)_{ij} = -(A)_{ji}, \forall i, j$

Παραδείγματα

$$\begin{pmatrix} 0 & 3 \\ -3 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 5 & 2 \\ -5 & 0 & 4 \\ -2 & -4 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 6 & -4 & 3 \\ -6 & 0 & 2 & 9 \\ 4 & -2 & 0 & -5 \\ -3 & -9 & 5 & 0 \end{pmatrix}$$

Πράξεις πινάκων

- **Πρόσθεση πινάκων**

Αν οι πίνακες έχουν το ίδιο μέγεθος μπορούμε να προσθέτουμε τα αντίστοιχα στοιχεία τους:

$$C = A + B \text{ όπου } C_{i,j} = A_{i,j} + B_{i,j}$$

- **Πίνακας με βαθμωτό ή διάνυσμα**

Προσθέτουμε/πολλαπλασιάζουμε το βαθμωτό με κάθε στοιχείο του πίνακα:

$$D = a \cdot B + c \text{ όπου } D_{i,j} = a \cdot B_{i,j} + c$$

- **Πολλαπλασιασμός πινάκων**

Προϋπόθεση: Οι στήλες του Α ιστές με τις γραμμές του Β

$$C = A * B, C_{i,j} = \sum_k (A_{i,k} \cdot B_{k,j})$$

$$C(mxp) = A(mx n) * B(nxp)$$

- **Γινόμενο στοιχείο προς στοιχείο (Element-wise product ή Hadamard product)**

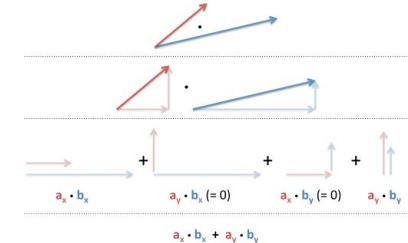
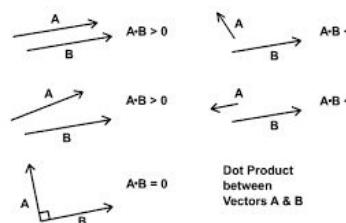
$$A \cdot B = a_{i,j} \cdot b_{i,j}$$

Εσωτερικό Γινόμενο Διανυσμάτων

Η ποσότητα $x^T y$ είναι το εσωτερικό γινόμενο των διανυσμάτων x και y του \mathbb{R}^n

Αν τα x, y είναι ορθογώνια, $x^T y = 0$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = [x_1 \ x_2 \ \dots \ x_n] * \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = x_1 * y_1 + x_2 * y_2 + \dots + x_n * y_n$$



Γραμμική Ανεξαρτησία Διανυσμάτων

Εάν ο τετριμένος συνδυασμός είναι ο μόνος που παράγει το μηδέν δηλαδή $c_1 u_1 + c_2 u_2 + \dots + c_n u_n = 0$ συμβαίνει μόνον όταν $c_1 = c_2 = \dots = c_n = 0$, τότε τα διανύσματα u_1, u_2, \dots, u_n είναι γραμμικώς ανεξάρτητα.

Πχ.

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -1 \end{bmatrix} = [1 \ 1] * \begin{bmatrix} 1 \\ -1 \end{bmatrix} = 0$$

Αν αυτό δεν συμβαίνει, τότε είναι γραμμικώς εξαρτημένα και κάποιο από αυτά είναι γραμμικός συνδυασμός των υπολοίπων.

$$\begin{aligned} c_1 - c_2 + 2c_3 &= 0 \\ c_2 + 3/2c_3 &= 0 \end{aligned}$$

Πχ.

$$u_1 = \begin{bmatrix} 1 \\ 1 \\ 3 \\ 3 \end{bmatrix}, u_2 = \begin{bmatrix} -1 \\ 1 \\ -1 \\ -3 \end{bmatrix}, u_3 = \begin{bmatrix} 2 \\ 5 \\ 9 \\ 6 \end{bmatrix} \rightarrow c_1 \begin{bmatrix} 1 \\ 1 \\ 3 \\ 3 \end{bmatrix} + c_2 \begin{bmatrix} -1 \\ 1 \\ -1 \\ -3 \end{bmatrix} + c_3 \begin{bmatrix} 2 \\ 5 \\ 9 \\ 6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 1 & -1 & 2 \\ 1 & 1 & 5 \\ 3 & -1 & 9 \\ 3 & -3 & 6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Γραμμική Ανεξαρτησία Διανυσμάτων

Άσκηση Εξετάστε αν τα διανύσματα $(1,1,0,0), (1,0,1,0), (0,0,1,1), (0,1,0,1)$ είναι γραμμικά ανεξάρτητα ή όχι και ελέγχετε εάν το διάνυσμα $(0,0,0,1)$ βρίσκεται στο χώρο που παράγουν.

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \xleftarrow{(-)} \rightarrow \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \xleftarrow{(-)} \rightarrow \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \xleftarrow{(-)} \rightarrow$$

$$\rightarrow \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} = U, \quad \text{όπως } r(A) = [\text{η διάλογος της-της διανύσματος}] \Rightarrow r(A) = 3 \rightarrow \text{Πλήθος βασικών μεταβλητών}$$

και $\dim N(A) = n - r(A) = 4 - 3 = 1 \neq 0 \rightarrow$ Πλήθος ελευθέρων μεταβλητών
όπως $\exists c \neq 0$ τ.ω. $A c = 0$ και τα διανύσματα ήνταν γρ. ανεξάρτητα

Γραμμική Ανεξαρτησία Διανυσμάτων

Άσκηση Εξετάστε αν τα διανύσματα $(1,1,0,0)$, $(1,0,1,0)$, $(0,0,1,1)$, $(0,1,0,1)$ είναι γραμμικά ανεξάρτητα ή όχι και ελέγχτε εάν το διάνυσμα $(0,0,0,1)$ βρίσκεται στο χώρο που παράγουν.

Το διάνυσμα $(0,0,0,1)$ βρίσκεται στο χώρο που παράγουν ανν Σ
 $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \in \mathbb{R}$ τ.ω. $\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \lambda_1 \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \lambda_2 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} + \lambda_3 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} + \lambda_4 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$

$$\Leftrightarrow \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Η τάσηταις γράφεται δίνει $0\lambda_1 + 0\lambda_2 + 0\lambda_3 + 0\lambda_4 = 1$, αρα το δύντηθε είναι ασύμβατο και υπομένως,

$(0,0,0,1) \notin \langle (1,1,0,0), (1,0,1,0), (0,0,1,1), (0,1,0,1) \rangle$

Αν δεν υπάρχει λύση, τότε λέμε ότι το σύστημα είναι ασύμβατο (inconsistent).

Γραμμική Ανεξαρτησία Διανυσμάτων

Ένα σύνολο n διανυσμάτων του \mathbb{R}^m είναι κατ' ανάγκη εξαρτημένο, όταν $n > m$

Κάθε διάνυσμα u του διανυσματικού χώρου V μπορεί να εκφραστεί ως γραμμικός συνδυασμός του w : για κάποιους συντελεστές c_i

$$u = c_1 w_1 + \dots + c_l w_l$$

Έστω διανύσματα που ξεκινούν από την αρχή ενός 3D χώρου:

2 εξαρτημένα διανύσματα \Rightarrow Περιέχονται στην ίδια ευθεία

3 εξαρτημένα διανύσματα \Rightarrow Περιέχονται στο ίδιο επίπεδο

Βάση και Διάσταση Διανυσματικού Χώρου

Βάση ενός διανυσματικού χώρου είναι ένα σύνολο διανυσμάτων που έχει ταυτόχρονα τις δύο ιδιότητες:

1. Είναι γραμμικώς ανεξάρτητο
2. Παράγει τον χώρο.

Οταν ένας τετραγωνικός πίνακας είναι αντιστρέψιμος (**non-singular-γραμμικά ανεξάρτητες στήλες (ή γραμμές)**), τότε οι στήλες του είναι ανεξάρτητες και αποτελούν μία βάση του.

Αν u_1, \dots, u_m και w_1, \dots, w_n είναι δύο βάσεις του ίδιου διανυσματικού χώρου V , τότε $m=n$

Το m (ή το n) εκφράζει τους "βαθμούς ελευθερίας" του χώρου και ονομάζεται **διάσταση του V**

$\rightarrow \dim(V) = [\text{μέγιστο πλήθος γραμμικώς ανεξάρτητων διανυσμάτων του } V]$

π.χ. $\dim(\mathbb{R}^2)=2$, $\dim(\mathbb{R}^3)=3$

Βάση και Διάσταση Διανυσματικού Χώρου

Να βρεθεί μία βάση και η διάσταση του χώρου

$$V = \{(x, y, z) \in \mathbb{R}^3 | x - y + 2z = 0\}$$

$$x - y + 2z = 0 \rightarrow x = y - 2z, y, z \in \mathbb{R}$$

Οδηγοί

$$V = \{(x, y, z) \in \mathbb{R}^3 | x = y - 2z\} = \{(y - 2z, y, z) | y, z \in \mathbb{R}\}$$

$$= \{(y, y, 0) + (-2z, 0, z) | y, z \in \mathbb{R}\}$$

$$= \{y(1, 1, 0) + z(-2, 0, 1) | y, z \in \mathbb{R}\}$$

$$= \langle (1, 1, 0), (-2, 0, 1) \rangle$$

$$\mathbf{e}_1, \mathbf{e}_2$$

Για να αποτελέσουν βάση του χώρου αυτά τα 2 διανύσματα θα πρέπει να είναι γραμμικώς ανεξάρτητα:

$$\begin{bmatrix} 1 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix} \xrightarrow{1st+2+2nd} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}$$

Ορθογώνιοι πίνακες

- Ένας τετραγωνικός πίνακας ονομάζεται **ορθογώνιος πίνακας** εάν και μόνο αν τα **η διανύσματα- στήλες** (ή η γραμμές) του αποτελούν ορθοκανονικό σύστημα του χώρου διάστασης n × n

Π.χ. (ταυτοτικός)

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

(μετάθεσης)

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

(περιστροφή)

$$\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

- Ισοδύναμα, ένας τετραγωνικός πίνακας A είναι **ορθογώνιος** αν η μετάθεσή του (ανάστροφος) είναι ίση με τον αντίστροφό του: $A^T = A^{-1}$

και ισχύει

$$A^T A = A A^T = I$$

Αντιστρέψιμος πίνακας: $(A^{-1})^T = (A^T)^{-1}$

Παράδειγμα

5) Αν $A \in \mathbb{R}^{n \times n}$ είναι αντιστρέψιμος, τότε: $(A^{-1})^T = (A^T)^{-1}$

π.χ. $A = \begin{bmatrix} 1 & -2 \\ -3 & 5 \end{bmatrix} \Rightarrow A^{-1} = \frac{1}{5-6} \begin{bmatrix} 5 & 2 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} -5 & -2 \\ -3 & -1 \end{bmatrix} \Rightarrow (A^{-1})^T = \begin{bmatrix} -5 & -3 \\ -2 & -1 \end{bmatrix}$

και $A^T = \begin{bmatrix} 1 & -3 \\ -2 & 5 \end{bmatrix} \Rightarrow (A^T)^{-1} = \frac{1}{-1-10} \begin{bmatrix} 5 & 3 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} -5 & -3 \\ -2 & -1 \end{bmatrix}$

Ορθογώνιοι πίνακες

- Η **ορίζουσα** οποιουδήποτε **ορθογώνιου πίνακα** είναι είτε +1 (κάνει μια καθαρή περιστροφή), ή -1 (είναι μια καθαρή αντανάκλαση, ή μια σύνθεση της αντανάκλασης και της περιστροφής).
- Εάν $A \in \mathbb{R}^{n \times n}$ είναι ορθογώνιος πίνακας, τότε $A^T A = I_n$

Έπειτα ότι:

$$\det(A^T A) = \det(I_n) = 1 ,$$

$$\det(A^T) \det(A) = 1 ,$$

$$\det(A) = \pm 1$$

Γραμμική απεικόνιση

Η **γραμμική απεικόνιση** (ή γραμμικός μετασχηματισμός) είναι μια μαθηματική συνάρτηση που μετασχηματίζει διανύσματα από έναν διανυσματικό χώρο σε έναν άλλον, με τρόπο που διατηρεί δύο βασικές ιδιότητες:

- Γραμμικότητα ως προς την πρόσθεση:** Αν έχουμε δύο διανύσματα v_1 και v_2 , τότε η γραμμική απεικόνιση T ικανοποιεί την εξίσωση: $T(v_1 + v_2) = T(v_1) + T(v_2)$
- Γραμμικότητα ως προς τον πολλαπλασιασμό με βαθμωτό αριθμό:** Αν έχουμε ένα διάνυσμα v και έναν αριθμό c , τότε ισχύει: $T(cv) = cT(v)$

Αυτές οι δύο ιδιότητες καθορίζουν τη **γραμμικότητα**.

Οι πίνακες (μήτρες) συχνά χρησιμοποιούνται για να περιγράψουν τέτοιους γραμμικούς μετασχηματισμούς.

- Αυτό ακριβώς κάνει η εξίσωση $\mathbf{Ax} = \mathbf{b}$, όπου η μήτρα A λειτουργεί ως η γραμμική απεικόνιση που μετασχηματίζει το διάνυσμα x στο αποτέλεσμα b .

Σύστημα Γραμμικών Εξισώσεων $\mathbf{Ax}=\mathbf{b}$

$\mathbf{A} \in \mathbb{R}^{m \times n}$, γνωστός πίνακας,

$\mathbf{b} \in \mathbb{R}^m$, διάνυσμα,

$\mathbf{x} \in \mathbb{R}^n$, διάνυσμα με τις άγνωστες μεταβλητές

Αναλύεται σε: $A_{1,:}x = b_1$

$$A_{2,:}x = b_2$$

.....

$$A_{m,:}x = b_m$$

$$A_{1,1}x_1 + A_{1,2}x_2 + \dots + A_{1,n}x_n = b_1$$

$$A_{2,1}x_1 + A_{2,2}x_2 + \dots + A_{2,n}x_n = b_2$$

.....

$$A_{m,1}x_1 + A_{m,2}x_2 + \dots + A_{m,n}x_n = b_m$$

Μπορεί να έχει καμία λύση, πολλές λύσεις ή ακριβώς μία λύση (πολ/σμό με ανάστροφο)

Πίνακες από διαφορετικές οπτικές γωνίες

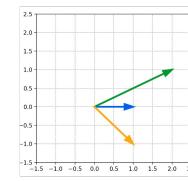
Υπάρχουν τρεις συμπληρωματικές προοπτικές για την προβολή πινάκων:

- **Προοπτική 1:** Ένας πίνακας ως πίνακας αριθμών
- **Προοπτική 2:** Ένας πίνακας ως λίστα διανυσμάτων (διανύσματα γραμμής και στήλης)
- **Προοπτική 3:** Ένας πίνακας ως συνάρτηση που αντιστοιχίζει διανύσματα από το ένα χώρο σε άλλο

Βλέποντας πίνακες μέσα από αυτές τις προοπτικές μπορούμε να αποκτήσουμε καλύτερη διαίσθηση για τους διανυσματικούς χώρους που προκαλούνται από τους πίνακες.

Διανύσματα στήλης

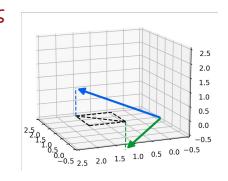
$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & -1 \end{bmatrix}$$



Προοπτική 2

Διανύσματα γραμμής

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & -1 \end{bmatrix}$$



Προοπτική 3

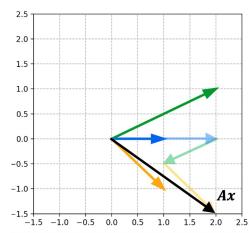
Προοπτική 3

Προοπτική 3: Κατανόηση του column space

Για έναν δεδομένο πίνακα $\mathbf{A} \in \mathbb{R}^{m \times n}$, μπορούμε να δούμε αυτόν τον πίνακα ως συνάρτηση που αντιστοιχίζει διανύσματα από το \mathbb{R}^n σε διανύσματα στο \mathbb{R}^m

- Ένα διάνυσμα $\mathbf{x} \in \mathbb{R}^n$ αντιστοιχίζεται στο διάνυσμα $\mathbf{b} \in \mathbb{R}^m$ μέσω $\mathbf{Ax}=\mathbf{b}$
 - Ορίσουμε μια συνάρτηση $T: \mathbb{R}^n \rightarrow \mathbb{R}^m$ ως: $T(\mathbf{x}):=\mathbf{Ax}$
 - Αποδεικνύεται ότι το column space είναι απλώς το εύρος (range) αυτής της συνάρτησης T
 - Το εύρος αφορά τη γραμμική απεικόνιση που πραγματοποιεί ο πίνακας \mathbf{A} , και ουσιαστικά είναι ο χώρος των εικόνων του \mathbf{A} στο σύνολο των πιθανών εξόδων του

$$\begin{array}{c|c|c} \mathbf{A} & \mathbf{x} & \mathbf{Ax} \\ \hline 1 & 2 & 1 \\ 0 & 1 & -1 \\ \hline 2 & -0.5 & 2 \\ 1 & 1 & -1.5 \end{array} = \begin{array}{c|c|c} 1 & 2 & 1 \\ 0 & 1 & -1 \\ \hline 2 & -0.5 & 2 \\ 1 & 1 & -1.5 \end{array} + \begin{array}{c|c|c} 0 & 1 & -1 \\ 1 & 1 & -1 \\ \hline 0 & 1 & -1 \end{array} = \begin{array}{c|c|c} 0 & 1 & -1 \\ 1 & 1 & -1 \\ \hline 0 & 1 & -1 \end{array} + \begin{array}{c|c|c} 1 & 2 & 1 \\ 0 & 1 & -1 \\ \hline 1 & 2 & 1 \end{array} = \begin{array}{c|c|c} 1 & 2 & 1 \\ 0 & 1 & -1 \\ \hline 2 & -0.5 & 2 \end{array}$$



Το column space είναι ολόκληρο το \mathbb{R}^2 αφού μπορούμε να σχηματίσουμε οποιοδήποτε δισδιάστατο διάνυσμα χρησιμοποιώντας έναν γραμμικό συνδυασμό αυτών των τριών διανυσμάτων.

<https://mbermste.github.io/posts/matrixspaces/>

Πολλαπλασιασμός πίνακα (\mathbf{A}) με διάνυσμα (\mathbf{x}): πράξη λήψης ενός γραμμικού συνδυασμού των στηλών του \mathbf{A} χρησιμοποιώντας τους συντελεστές του \mathbf{x} ως συντελεστές του γραμμικού συνδυασμού

Σύστημα Γραμμικών Εξισώσεων $\mathbf{Ax}=\mathbf{b}$

Σκέψη :

Οι στήλες του \mathbf{A} ορίζουν διαφορετικές κατευθύνσεις που μπορούμε να “κινηθούμε” στο χώρο από το αρχικό σημείο 0 για να προσεγγίσουμε το \mathbf{b} .

$$\mathbf{Ax} = \sum_i x_i \mathbf{A}_{:,i} \rightarrow \text{Γραμμικός συνδυασμός}$$

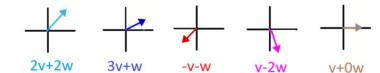
To
to
να
να
εύρος
σύνολο
ληφθούν
αρχικών διανυσμάτων.

(span)
όλων
με

ενός
των
γραμμικών
συνδυασμών.

v = $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ w = $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$
some combinations of v and w

είναι ιππορούν
JV



2v+2w
3v+w
-v-w
v-2w
v+0w

Σύστημα Γραμμικών Εξισώσεων $\mathbf{Ax} = \mathbf{b}$

Έχει λύση η $\mathbf{Ax} = \mathbf{b}$:

Θα ελέγχουμε αν το \mathbf{b} είναι στο εύρος των στηλών του \mathbf{A} , δηλαδή αν το \mathbf{b} εκφραστεί ως γραμμικός συνδυασμός των στηλών του \mathbf{A} .

Τρόποι Ελέγχου

1. Μέθοδος ελέγχου μέσω κατάταξης (rank)
2. Λύση μέσω απαλοιφής Gauss
3. Λύση μέσω Ορθογώνιας Προβολής (π.χ. Least Squares)
4. Επίλυση μέσω Singular Value Decomposition (SVD)

Για να μπορεί το σύστημα $\mathbf{Ax} = \mathbf{b}$ να έχει λύση για όλες τις τιμές του $\mathbf{b} \in \Re_m$, απαιτούμε ο χώρος των στηλών του \mathbf{A} να είναι ο \Re_m

Πρέπει ο \mathbf{A} να έχει ακριβώς m γραμμικές ανεξάρτητες στήλες, όχι τουλάχιστον m .

Σύστημα Γραμμικών Εξισώσεων

$$\mathbf{Ax} = \mathbf{b}$$

Προκειμένου ο πίνακας \mathbf{A} να έχει αντίστροφο, πρέπει επιπλέον να διασφαλίσουμε ότι η εξίσωση έχει το πολύ μία λύση για κάθε τιμή του \mathbf{b} . (Διαφορετικά, υπάρχουν περισσότεροι από έναν τρόποι παραμετροποίησης κάθε λύσης)

$$\begin{aligned} Ax &= b \\ A^{-1}Ax &= A^{-1}b \\ I_n x &= A^{-1}b \\ x &= A^{-1}b \end{aligned}$$

Για να έχει αντίστροφο ο πίνακας \mathbf{A} θα πρέπει:

- να είναι τετράγωνος, δηλαδή, απαιτούμε ότι $m = n$ και
- όλες οι στήλες πρέπει να είναι γραμμικώς ανεξάρτητες

Τριγωνική παραγοντοποίηση $A=LU$

Αν δεν απαιτούνται εναλλαγές γραμμών, ο αρχικός πίνακας μπορεί να γραφεί ως γινόμενο $A=LU$

$$\begin{aligned} L : &\text{ κάτω τριγωνικός} & L = \begin{bmatrix} 1 & 0 & \dots & 0 \\ \lambda_{2,1} & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots \\ \lambda_{n,1} & \lambda_{n,2} & \dots & 1 \end{bmatrix} & U = \begin{bmatrix} u_{1,1} & u_{1,2} & \dots & u_{1,n} \\ 0 & u_{2,2} & \dots & u_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & u_{m,n} \end{bmatrix} \\ U : &\text{ άνω τριγωνικός} \end{aligned}$$

- Εμφανίζεται μετά τη διαδοχική απαλοιφή και πριν την ανάδρομη αντικατάσταση.
- Τα διαγώνια στοιχεία του είναι οι οδηγοί

π.χ.

$$\begin{aligned} A &= \begin{bmatrix} 3 & 1 \\ 4 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{4}{3} & 1 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 0 & \frac{2}{3} \end{bmatrix} = LU \\ A &= \begin{bmatrix} -5 & 1 & -3 & 4 \\ 8 & -7 & 3 & 2 \\ -3 & -6 & -1 & -1 \\ 0 & 0 & 3 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{8}{5} & 1 & 0 & 0 \\ \frac{3}{5} & \frac{33}{27} & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} -5 & 1 & -3 & 4 \\ 0 & -\frac{27}{5} & -\frac{9}{5} & \frac{42}{5} \\ 0 & 0 & 3 & -\frac{41}{3} \\ 0 & 0 & 0 & \frac{68}{3} \end{bmatrix} = LU \\ A &= \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 7 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & 0 & 0 \end{bmatrix} = LU \end{aligned}$$

Εύρεση A^{-1} : Μέθοδος των Gauss-Jordan

Έστω η εξίσωση : $\mathbf{AA}^{-1}=\mathbf{I}$.

Εάν θεωρηθεί στήλη προς στήλη αυτή η εξίσωση προσδιορίζει τις στήλες του A^{-1}

$$Ax_1 = e_1, Ax_2 = e_2, Ax_3 = e_3$$

όπου

$$[e_1 \ e_2 \ e_3] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} [A \ e_1 \ e_2 \ e_3] &= [U \ L] \\ [U \ L] &= [I \ A^{-1}] \end{aligned}$$

Εύρεση A^{-1} : Μέθοδος των Gauss-Jordan

$$A = \begin{pmatrix} 9 & 3 & 4 \\ 4 & 3 & 4 \\ 1 & 1 & 1 \end{pmatrix} \quad (A \mid I_3) = \left(\begin{array}{ccc|ccc} 9 & 3 & 4 & 1 & 0 & 0 \\ 4 & 3 & 4 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{array} \right) = \left(\begin{array}{ccc|ccc} 9 & 3 & 4 & 1 & 0 & 0 \\ 0 & 5/3 & 20/9 & -4/9 & 1 & 0 \\ 0 & 2/3 & 5/9 & -1/9 & 0 & 1 \end{array} \right)$$

$$\begin{bmatrix} A & e_1 & e_2 & e_3 \\ U & L \end{bmatrix} = \begin{bmatrix} U & L \\ I & A^{-1} \end{bmatrix}$$

$$\begin{aligned} &= \begin{pmatrix} 9 & 0 & 0 & | & 9/5 & -9/5 & 0 \\ 0 & 5/3 & 20/9 & | & -4/9 & 1 & 0 \\ 0 & 0 & -1/3 & | & 1/15 & -2/5 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 9 & 0 & 0 & | & 9/5 & -9/5 & 0 \\ 0 & 5/3 & 0 & | & 0 & -5/3 & 20/3 \\ 0 & 0 & -1/3 & | & 1/15 & -2/5 & 1 \end{pmatrix} \end{aligned}$$

Διατρούμε όλα τα στοιχεία της i-γραμμής του πίνακα εκ δεξιών της διαικεκομένης γραμμής με το μη μηδενικό στοιχείο της i-γραμμής του διαγώνου πίνακα και παίρνουμε:

$$A^{-1} = \begin{pmatrix} 1/5 & -1/5 & 0 \\ 0 & -1 & 4 \\ -1/5 & 6/5 & -3 \end{pmatrix}$$

Παραγοντοποίηση $A=LDU$

- Ο πίνακας L παραμένει ίδιος όπως στην $A=LU$
- Ο πίνακας D είναι διαγώνιος και περιέχει την διαγώνιο του πίνακα U της παραγοντοποίησης LU (στοιχεία οδηγών):
- Ο νέος άνω τριγωνικός πίνακας U προκύπτει από τον πίνακα U της παραγοντοποίησης LU διαιρώντας κάθε στοιχείο του με το στοιχείο της διαγωνίου (δηλ. τον οδηγό) της ίδιας γραμμής

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 7 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1/2 \end{bmatrix} = LDU$$

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 7 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 0 & 3 & 6 \\ 0 & 0 & 4 \end{bmatrix} = LU$$

Νόρμες (Norms)

Η νόρμα είναι μια συνάρτηση που αναθέτει έναν θετικό αριθμό (ή μηδέν) σε έναν διανυσματικό χώρο ή έναν πίνακα, με στόχο να μετρήσει το "μέγεθος" ή το "μήκος" ενός διανύσματος ή πίνακα.

Ουσιαστικά, η νόρμα είναι ένας τρόπος μέτρησης της απόστασης από το μηδέν στο διάστημα που εξετάζουμε.

$$L^p = \|x\|_p = (\sum |x_i|^p)^{1/p} \quad p \in \mathbb{R}, p \geq 1$$

Η νόρμα είναι μία συνάρτηση f που ικανοποιεί τις ακόλουθες ιδιότητες:

- Μη αρνητικότητα:** Η νόρμα είναι πάντα θετική εκτός από την περίπτωση του μηδενικού διανύσματος, που έχει νόρμα μηδέν.
- Ανισότητα τριγώνου:** Η νόρμα του αθροίσματος δύο διανυσμάτων είναι μικρότερη ή ίση από το αθροίσμα των νόρμων τους (παρόμοια με την ανισότητα τριγώνου στη γεωμετρία).
- Ομογένεια (homogeneity):** Η νόρμα πολλαπλασιάζεται απόλυτα με το μέτρο του αριθμού a .

$$f(\mathbf{x}) = 0 \Rightarrow \mathbf{x} = \mathbf{0}$$

$$f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y}) \text{ (the triangle inequality)}$$

$$\forall \alpha \in \mathbb{R}, f(\alpha \mathbf{x}) = |\alpha| f(\mathbf{x})$$

L^2 νόρμα (Ευκλείδια νόρμα)

- Για $p=2 \rightarrow$ η ευκλείδεια απόσταση από την αρχή μέχρι το σημείο που προσδιορίζεται από το x .
- Δηλώνεται συχνά ως $\|\mathbf{x}\|$, με το δείκτη 2 να παραλείπεται.
- Συνήθως υπολογίζουμε το τετράγωνο της L^2 νόρμας, απλά ως $\mathbf{x}^T \mathbf{x}$
- Υπολογιστικά, συχνά, το τετράγωνο της L^2 νόρμα μπορεί να αυξάνεται πολύ αργά κοντά το σημείο που προσδιορίζεται από το x
→ απαγορευτικό υπολογιστικά

L¹ νόρμα

- Συχνά είναι σημαντικό να γίνεται διάκριση μεταξύ στοιχείων που είναι ακριβώς μηδενικά και αυτών που είναι μικρά αλλά μη μηδενικά.

$$\|x\|_1 = \sum_i |x_i|$$

- Κάθε φορά που ένα στοιχείο του x μετακινείται κατά ϵ μακριά από το 0, η L¹ αυξάνεται κατά ϵ
- Υποκαθιστά το πλήθος των μη μηδενικών τιμών.

Μερικές φορές μετράμε το μέγεθος του διανύσματος υπολογίζοντας τον αριθμό των μη μηδενικών στοιχείων (Λανθασμένη ορολογία το νόρμα L⁰).

L[∞] νόρμα (Max νόρμα)

Υπολογίζει την απόλυτη τιμή του μεγαλύτερου στοιχείου του διανύσματος x

$$\|x\|_\infty = \max_i |x_i|$$

Υπολογισμός μεγέθους πίνακα

Nόρμα Frobenius

$$\|A\|_F = (\sum_{i,j} |a_{ij}|^2)^{1/2}$$

Είναι το ανάλογο της L² ενός διανύσματος.

Εσωτερικό γινόμενο με χρήση νόρμας

$$x^T y = \|x\|_2 \|y\|_2 \cos\theta \quad , \text{όπου } \theta \text{ η γωνία μεταξύ } x \text{ και } y$$

Διαγώνιοι πίνακες

Ένας πίνακας D , είναι διαγώνιος εάν και μόνο εάν $D_{i,j} = 0$ για όλα τα $i \neq j$

$diag(v)$ τετραγωνικός διαγώνιος πίνακας,
το διάνυσμα v περιέχει τις τιμές της διαγωνίου

Ο πολύτιμος του με άλλο πίνακα είναι υπολογιστικά αποδοτικός.

$$diag(v)x = v \odot x$$

Αντίστροφος : \exists αν τα στοιχεία της διαγωνίου είναι μη μηδενικά

$$diag(v)^{-1} = diag([1/v_1, \dots, 1/v_n])$$

- Σε πολλές περιπτώσεις, αλγόριθμοι μηχανικής μάθησης περιορίζουν πίνακες να είναι διαγώνιοι, κερδίζοντας υπολογιστικό κόστος και χρόνο.
- Είναι δυνατή η κατασκευή διαγώνιου πίνακα

Ιδιοτιμές και ιδιοδιανύσματα

Επίλυση γραμμικών συστημάτων

Επινοήθηκαν για επίλυση διαφορικών εξισώσεων :

$$\begin{aligned} \text{Η λύση θα είναι της μορφής :} \quad & \frac{dy}{dt} = Ay \\ & y(t) = e^{\lambda t} x \end{aligned} \quad \Rightarrow \quad \begin{array}{l} \lambda e^{\lambda t} x = A e^{\lambda t} x \Rightarrow Ax = \lambda x \\ \text{Ιδιοτιμή} \quad \quad \quad \text{Ιδιοδιάνυσμα} \end{array}$$

- Ο αριθμός λ είναι ιδιοτιμή του A όταν και μόνον όταν ισχύει $\det(A - \lambda I) = 0$
- Αυτή είναι η χαρακτηριστική εξίσωση και σε κάθε λύση της λ αντιστοιχεί ένα ιδιοδιάνυσμα x :

$$(A - \lambda I)x = 0 \quad \text{ή} \quad Ax = \lambda x$$

Βήματα επίλυσης του προβλήματος ιδιοτιμών

- Υπολογισμός της ορίζουσας του
- Εύρεση των ρίζων αυτού του πολυωνύμου
- Για κάθε ιδιοτιμή, επίλυση του συστήματος

π.χ. Διαφορική εξίσωση $\begin{cases} \frac{dy_1}{dt} = 4y_1 - 5y_2 \\ \frac{dy_2}{dt} = 2y_1 - 3y_2 \end{cases}$

$$t = 0 : y_1 = 8, y_2 = 5$$

$$1., 2. |A - \lambda I| = \begin{bmatrix} 4 & -5 \\ 2 & -3 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 4-\lambda & -5 \\ 2 & -3-\lambda \end{bmatrix} = (4-\lambda)(-3-\lambda) + 10 = -(4-\lambda)(3+\lambda) + 10 = (\lambda-4)(\lambda+3) - 10 = \lambda^2 - 4\lambda + 3\lambda - 12 + 10 = \lambda^2 - \lambda - 2 \rightarrow \lambda_1 = -1, \lambda_2 = 2$$

$$3. (A - \lambda_1 I)x = \begin{bmatrix} 5 & -5 \\ 2 & -2 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow x_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$(A - \lambda_2 I)x = \begin{bmatrix} 2 & -5 \\ 2 & -5 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow x_2 = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$$

Ειδικές λύσεις: $u = e^{\lambda_1 t} x_1 = e^{-t} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

Γενική λύση: $u = c_1 e^{\lambda_1 t} x_1 + c_2 e^{\lambda_2 t} x_2$

$$u = e^{\lambda_2 t} x_2 = e^{2t} \begin{bmatrix} 5 \\ 2 \end{bmatrix} \quad \text{Χρήση αρχικών συνθηκών} \rightarrow 8 = c_1 + 5c_2 \quad u = 3e^{-t} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + e^{2t} \begin{bmatrix} 5 \\ 2 \end{bmatrix}$$

$$5 = c_1 + 2c_2$$

Ιδιοδιανύσματα (eigenvectors)

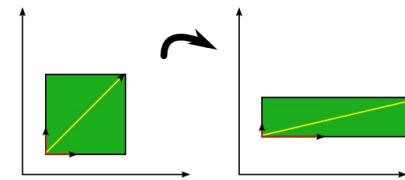
Ένα ιδιοδιανύσμα ενός γραμμικού μετασχηματισμού είναι ένα μη μηδενικό διάνυσμα που, όταν εφαρμοστεί πάνω του ο γραμμικός μετασχηματισμός (π.χ., μέσω ενός πίνακα A), δεν αλλάζει κατεύθυνση.

→ δηλαδή, η δράση του μετασχηματισμού στο ιδιοδιανύσμα οδηγεί απλά σε πολλαπλασιασμό του με έναν σταθερό αριθμό.

Για έναν πίνακα A και ένα ιδιοδιανύσμα v, ισχύει: $\mathbf{Av}=\lambda v$

όπου:

- A είναι ο πίνακας που εκφράζει τον γραμμικό μετασχηματισμό.
- v είναι το ιδιοδιανύσμα.
- λ είναι μια σταθερά, η **ιδιοτιμή** (eigenvalue), που δείχνει πόσο "κλιμακώνεται" το ιδιοδιανύσμα.



Τα **ιδιοδιανύσματα (κόκκινο)** δεν αλλάζουν κατεύθυνση όταν εφαρμόζεται σε αυτά ένας γραμμικός μετασχηματισμός (π.χ. κλιμάκωση).

- Μπορεί να αλλάξουν μέγεθος, αλλά η κατεύθυνση τους παραμένει ίδια.

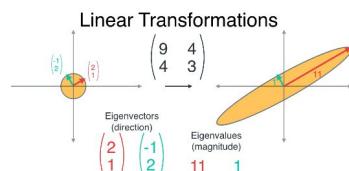
Άλλα διανύσματα (κίτρινο), που δεν είναι ιδιοδιανύσματα αλλάζουν.

Ιδιοτιμές (Eigenvalues)

Οι **ιδιοτιμές** είναι οι αντίστοιχοι κλιμακωτοί παράγοντες που συνδέονται με κάθε ιδιοδιανύσμα.

Δηλαδή, όταν εφαρμόζεται ο πίνακας A σε ένα ιδιοδιανύσμα v, το αποτέλεσμα είναι να "τεντώνεται" ή να "συρρικνώνεται" το v κατά έναν παράγοντα λ, αλλά η κατεύθυνσή του παραμένει η ίδια.

- Η ιδιοτιμή λ μετράει την **ποσότητα** κατά την οποία το αντίστοιχο ιδιοδιανύσμα v τεντώνεται ή συρρικνώνεται από τον γραμμικό μετασχηματισμό.
 - αν $\lambda > 1$, το διάνυσμα μεγαλώνει,
 - αν $0 < \lambda < 1$, συρρικνώνεται,
 - αν $\lambda < 0$, το διάνυσμα αντιστρέφει κατεύθυνση.



Εφαρμογές

- Μείωση διαστάσεων (PCA)**
Οι κύριες συνιστώσεις είναι τα ιδιοδιανύσματα ενός πίνακα συνδιακύμανσης, κι οι κύριες συνιστώσες.

- Δυναμικά Συστήματα:**
Τα ιδιοδιανύσματα και οι ιδιοτιμές βοηθούν στην ανάλυση της συμπεριφοράς ενός συστήματος, π.χ., αν ένα σύστημα θα συγκλίνει σε μια σταθερή κατάσταση.

- Γραμμικοί Μετασχηματισμοί**
Οι ιδιοτιμές χρησιμοποιούνται για την εύρεση λύσεων και την κατανόηση της δομής του πίνακα.

Ιδιοτιμές και ιδιοδιανύσματα

Εξίσωση-κλειδί: $Ax = \lambda x$

- Οι **ιδιοτιμές** παριστάνουν το **βηματισμό στο χρόνο** (magnitude)
- Τα **ιδιοδιανύσματα** παριστάνουν τις "**κανονικές καταστάσεις**" του συστήματος και επιδρούν ανεξάρτητα το ένα από το άλλο (direction)
- Μπορούμε να παρακολουθήσουμε τη **συμπεριφορά κάθε ιδιοδιανύσματος** και να συνδυάσουμε αυτές τις κανονικές καταστάσεις για να βρούμε **μια λύση**

→ Διαγωνοποίηση

Διαγωνοποίηση

$$A = V\Lambda V^{-1}$$

$$\begin{aligned}
 \frac{dy}{dt} = Ay & \Rightarrow \lambda e^{\lambda t} x = Ae^{\lambda t}x \Rightarrow Ax = \lambda x \\
 y(t) = e^{\lambda t}x & \Rightarrow A[x_1 \ x_2] = [\lambda_1 x_1 \ \lambda_2 x_2] \quad \text{Eigenvector Matrix} \\
 & \Rightarrow A[x_1 \ x_2] = [x_1 \ x_2] \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \quad \text{Eigenvalue Matrix} \\
 & \Rightarrow AV = V\Lambda \quad A^2 = ; \\
 & \Rightarrow AVV^{-1} = V\Lambda V^{-1} \quad A^2 = V\Lambda V^{-1}V\Lambda V^{-1} = V\Lambda I\Lambda V^{-1} = V\Lambda^2 V^{-1} \\
 & \Rightarrow AI = V\Lambda V^{-1} \quad A^n = ; \\
 & \Rightarrow \boxed{A = V\Lambda V^{-1}} \quad \text{Idios πίνακας ιδιοδιανυσμάτων όπως ο } A \\
 & \quad \quad \quad \text{Ο πίνακας ιδιοτυπών υψωμένος στην } n
 \end{aligned}$$

Πίνακες, ιδιοτιμές, ιδιοδιανύσματα

Συμμετρικός πίνακας $S = S^T$	→ πραγματικές ιδιοτιμές → ορθογώνια ιδιοδιανύσματα
Ανάστροφος πίνακας $A^T = -A$	→ φανταστικές ιδιοτιμές → ορθογώνια μιγαδικά ιδιοδιανύσματα
Ορθογώνιος πίνακας $Q^T Q = I$	→ για όλες τις ιδιοτιμές ισχύει: $ \lambda = 1$ → ορθογώνια μιγαδικά ιδιοδιανύσματα

Ένας πίνακας ονομάζεται **ιδιάζων (singular)** ανν οποιαδήποτε από τις ιδιοτιμές του είναι μηδέν.

Πίνακες, ιδιοτιμές, ιδιοδιανύσματα

Ορισμένος πίνακας

- Ένας συμμετρικός πχν πίνακας ονομάζεται **θετικά** (αρνητικά) ορισμένος, εάν για όλα τα μη μηδενικά διανύσματα $x \in \mathbb{R}^n$ το $Q(x) = x^T A x$ παίρνει μόνο θετικές τιμές (αρνητικές τιμές).

- Εάν ένας συμμετρικός πχν πίνακας παίρνει μόνο θετικές (αρνητικές) ονομάζεται **γνησίως-θετικός** (αρνητικός)

$$x^T A x = 0 \Rightarrow x = 0$$

- Εάν ένας συμμετρικός πχν πίνακας παίρνει θετικές (αρνητικές) ή μηδενικές ονομάζεται **ημι-θετικός** (αρνητικός)

$$\forall x, x = x^T A x \geq 0$$

- Εάν ένας συμμετρικός πχν δεν είναι ούτε θετικός ούτε αρνητικός τότε ο πίνακας είναι **αόριστος**.

Παραγοντοποίηση

$$A=LU \quad (\text{Απαλοιφή})$$

$$A=QR \quad (\text{Gram-Schmidt})$$

$$S=Q\Lambda Q^T \quad (\text{Symmetric: } [\sigma_1 \ \dots \ \sigma_n] \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \begin{bmatrix} \sigma_1^T \\ \vdots \\ \sigma_n^T \end{bmatrix})$$

$$A=X\Lambda X^{-1}$$

$$A=U\Sigma V^T \quad (\text{Ορθογώνιος } \times \text{ Διαγώνιος } \times \text{ Ορθογώνιος})$$

Trace Operator

Δίνει το άθροισμα όλων των διαγωνίων τιμών ενός πίνακα

$$Tr(\mathbf{A}) = \sum_i A_{i,i}$$

Frobenius νόρμα του \mathbf{A} : $\|\mathbf{A}\|_F = \sqrt{Tr(\mathbf{A}\mathbf{A}^T)}$

ΙΔΙΟΤΗΤΕΣ

$$Tr(A) = Tr(A^T)$$

$Tr(ABC) = Tr(CAB) = Tr(BCA)$, αν το επιπρέπουν οι διαστάσεις των πινάκων ή

$$Tr(\prod_{i=1}^n F^{(i)}) = Tr(F^{(n)} \prod_{i=1}^n F^{(i)})$$

Ένα βαθμωτό είναι το δικό του ίχνος: $a = Tr(a)$

Παράδειγμα

$$Tr \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} = a_{00} + a_{11} + a_{22}$$

Ορίζουσα (Determinant)

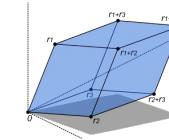
$\det(\mathbf{A})$: αντιστοιχεί πίνακα σε βαθμωτό

- Ισούται με το γινόμενο όλων των ιδιοτιμών του πίνακα

Γεωμετρική ερμηνεία

- Η απόλυτη τιμή της ορίζουσας δίνει την κλίμακα με την οποία το εμβαδόν ή ο όγκος (ή μιας μεγαλύτερης διάστασης αναλογία) πολλαπλασιάζεται με τον σχετικό γραμμικό μετασχηματισμό,
- ο το πρόσημό της δείχνει αν ο μετασχηματισμός διατηρεί τον προσανατολισμό.

Συνοπτικά, η ορίζουσα παρέχει γεωμετρική πληροφορία σχετικά με τη διάταξη και την κατεύθυνση των διανυσμάτων στο επίπεδο.



Ορίζουσα Πίνακα: Γεωμετρική ερμηνεία

Έστω ότι έχουμε 2 διανύσματα στο επίπεδο $v=[a \ c]$ και $w=[\ b \ d]$.

Ο πίνακας που σχηματίζεται από αυτά τα δύο διανύσματα είναι: $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$

Η ορίζουσα $\det(A)=ad-bc$, είναι επίσης η περιοχή του παραλληλογράμμου που σχηματίζεται από τα δύο διανύσματα.

- Αν η ορίζουσα είναι θετική, τότε τα δύο διανύσματα δείχνουν προς ίδια κατεύθυνση και η περιοχή του παραλληλογράμμου είναι θετική.
- Αν είναι αρνητική, τα δύο διανύσματα δείχνουν προς αντίθετες κατευθύνσεις και η περιοχή είναι αρνητική.
- Αν η ορίζουσα είναι μηδέν, τότε τα δύο διανύσματα είναι γραμμικά εξαρτημένα, και η περιοχή του παραλληλογράμμου είναι μηδενική.

π.χ. $A_{2 \times 2}, \det(A)=-2$: όταν εφαρμόζεται στην περιοχή ενός επιπέδου με πεπερασμένο

εμβαδόν, θα μετασχηματιστεί σε μια περιοχή με το διπλάσιο εμβαδόν, ενώ αντιστρέψει τον προσανατολισμό της.

$\det(A)=0$: ο χώρος συστέλλεται

$\det(A)=1$: ο μετασχηματισμός διατηρεί τον όγκο

Πιθανότητες

Βασικές Έννοιες Πιθανοτήτων

- Η πιθανότητα εκφράζει την αβεβαιότητα ενός γεγονότος.
- Χρησιμοποιείται για την **εκτίμηση** του πόσο πιθανό είναι να συμβεί ένα γεγονός.

Βασικές Έννοιες

- Τυχαίες μεταβλητές
- Κατανομές πιθανότητας
- Συναρτήσεις μάζας και πυκνότητας πιθανότητας

Τυχαίες Μεταβλητές

Μια τυχαία μεταβλητή είναι μια μεταβλητή που μπορεί να πάρει διαφορετικές τιμές σύμφωνα με κάποιες πιθανότητες.

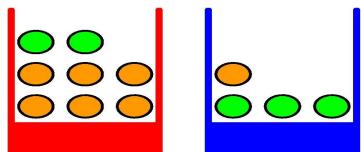
Χρησιμοποιούνται για την **περιγραφή** τυχαίων φαινομένων.

Οι τυχαίες μεταβλητές χωρίζονται σε δύο κατηγορίες:

- **Διακριτές Τυχαίες Μεταβλητές:** συγκεκριμένες και μετρήσιμες τιμές
 - ο αριθμός των παιδιών σε μια οικογένεια (0, 1, 2, ...).
 - οι ρίψεις ενός ζαριού (1, 2, 3, 4, 5, 6).
- **Συνεχείς Τυχαίες Μεταβλητές:** οποιαδήποτε τιμή σε ένα συνεχές διάστημα.
 - το ύψος ενός ατόμου (μπορεί να είναι 1.75m, 1.76m, 1.761m κ.ο.κ.).
 - η θερμοκρασία της ατμόσφαιρας.

Probability Theory - προσέγγιση

Apples and Oranges



Βασική έννοια στην αναγνώριση προτύπων:

Αβεβαιότητα

- θόρυβος στις μετρήσεις
- πεπερασμένο σύνολο δεδομένων

Θεωρία πιθανοτήτων

- ποσοτικοποίηση της αβεβαιότητας
- χειρισμός της αβεβαιότητας

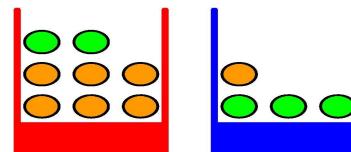
Θεωρία πιθανοτήτων + θεωρία αποφάσεων → Βέλτιστες Προβλέψεις

Πείραμα

1. Τυχαία επιλογή δοχείου
2. Τυχαία επιλογή φρούτου
3. Επαναποθέτηση φρούτου

Probability Theory - προσέγγιση

Apples and Oranges



→ Επανάληψη του πειράματος πολλές φορές

→ Έστω 40% επιλέχθηκε το red Box και 60% το blue Box

> Τυχαία μεταβλητή: $B = \{r, b\}$

> Τυχαία μεταβλητή: $F = \{a, o\}$

$$p(B = r) = 4/10$$

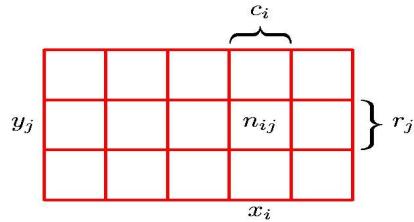
$$p(B = \beta) = 6/10$$

Πιθανά ερωτήματα:

- Ποια είναι η συνολική πιθανότητα να επιλέξουμε μήλο;
- Αν έχουμε επιλέξει ένα πορτοκάλι, ποια είναι η πιθανότητα να είναι από το μπλε δοχείο;

} Κανόνες αθροίσματος και γινομένου πιθανοτήτων

Probability Theory



Joint Probability
Από κοινού πιθανότητα

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$$

X = x_i , $i = \{1..M\}$ (**Boxes**)

Y = y_j , $j = \{1..L\}$ (**Fruits**)

N: συνολικές δοκιμές

n_{ij} = πλήθος δοκιμών όπου $X = x_i$ και $Y = y_j$

c_i = πλήθος δοκιμών όπου $X = x_i$ ανεξάρτητα του Y

r_j = πλήθος δοκιμών όπου $Y = y_j$ ανεξάρτητα του X

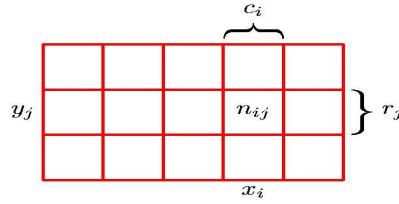
Marginal Probability
Οριακή Πιθανότητα

$$p(X = x_i) = \frac{c_i}{N}.$$

Conditional Probability
Υπό Συνθήκη Πιθανότητα

$$p(Y = y_j | X = x_i) = \frac{n_{ij}}{c_i}$$

Probability Theory



X = x_i , $i = \{1..M\}$ (**Boxes**)

Y = y_j , $j = \{1..L\}$ (**Fruits**)

N: συνολικές δοκιμές

n_{ij} = πλήθος δοκιμών όπου $X = x_i$ και $Y = y_j$

c_i = πλήθος δοκιμών όπου $X = x_i$ ανεξάρτητα του Y

r_j = πλήθος δοκιμών όπου $Y = y_j$ ανεξάρτητα του X

Sum Rule

$$\begin{aligned} p(X = x_i) &= \frac{c_i}{N} = \frac{1}{N} \sum_{j=1}^L n_{ij} \\ &= \sum_{j=1}^L p(X = x_i, Y = y_j) \end{aligned}$$

Product Rule

$$\begin{aligned} p(X = x_i, Y = y_j) &= \frac{n_{ij}}{N} = \frac{n_{ij}}{c_i} \cdot \frac{c_i}{N} \\ &= p(Y = y_j | X = x_i) p(X = x_i) \end{aligned}$$

The Rules of Probability

Sum Rule

$$p(X) = \sum_Y p(X, Y)$$

Product Rule

$$p(X, Y) = p(Y|X)p(X)$$

Bayes' Theorem

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$

X = x_i , $i = \{1..M\}$ (**Boxes**)
Y = y_j , $j = \{1..L\}$ (**Fruits**)

$$\text{όπου: } p(X) = \sum_Y p(X|Y)p(Y)$$

Δίνει έναν μαθηματικό τρόπο να ενημερώνουμε την πιθανότητα μιας υπόθεσης (ή γεγονότος) όταν λαμβάνουμε νέα δεδομένα.

- Έστω ότι έχουμε μια αρχική πίστη ή εκτίμηση για κάτι (**prior : P(X|Y)**).
- Όταν λάβουμε καινούργια πληροφορία (δεδομένα), το θεώρημα Bayes λέει **πώς να προσαρμόσεις** αυτή την πίστη / εκτίμηση και να βρεις την πιο ενημερωμένη πιθανότητα (**posterior : P(Y|X)**).
- Το posterior είναι η εκ των υστέρων πιθανότητα — δηλαδή η πιθανότητα για μια υπόθεση αφού έχουμε δει τα δεδομένα.

Το Θεώρημα Bayes μας μαθαίνει πώς να μαθαίνουμε, πώς να ενημερώνουμε τις πεποιθήσεις μας βάσει νέων στοιχείων.

Υπολογισμός posterior πιθανότητας

X = x_i , i={1..M} (Boxes)
Y = y_j , j={1..L} (Fruits)

Posterior: $P(\text{Hypothesis} | \text{Data})$ - Πώς το υπολογίζουμε; → Bayes' Theorem : $p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$
 $P(\text{Hypothesis} | \text{Data}) = P(\text{Data} | \text{Hypothesis}) \times P(\text{Hypothesis}) / P(\text{Data})$

$$\text{Posterior} = (\text{likelihood} \times \text{prior}) / \text{evidence}$$

prior = $P(\text{Hypothesis})$ → πιθανότητα πριν δούμε τα νέα δεδομένα

likelihood = $P(\text{Data} | \text{Hypothesis})$ → πιθανότητα να δούμε αυτά τα δεδομένα αν ισχύει η υπόθεση

evidence = $P(\text{Data})$ → συνολική πιθανότητα των δεδομένων

Συνήθως:

$$\text{Posterior} \propto \text{likelihood} \times \text{prior}$$

δηλαδή το posterior είναι ανάλογο του γινομένου του likelihood με το prior — γιατί το evidence

$P(\text{Data})$ είναι σταθερό για τα δεδομένα που έχουμε και απλώς κανονικοποιεί τις πιθανότητες.

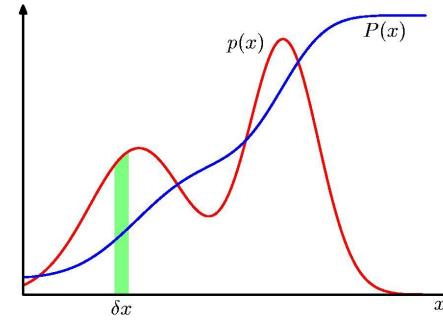
Probability Density Function

Έστω μια συνεχής τυχαία μεταβλητή x , τότε η συνάρτηση πυκνότητας πιθανότητας $p(x)$ ικανοποιεί:

$$p(x \in (a, b)) = \int_a^b p(x) dx$$

Οπου:

- $p(x)$ η PDF της μεταβλητής x
- $p(x \in (a, b))$ η πιθανότητα ότι η x παίρνει τιμές στο διάστημα $[a, b]$



$$P(z) = \int_{-\infty}^z p(x) dx$$

Ιδιότητες της $p(x)$

1. **Μη αρνητικότητα:** $p(x) \geq 0$
2. **Κανονικοποίηση (Normalization):** $\int_{-\infty}^{\infty} p(x) dx = 1$
3. **Μηδενική πιθανότητα σε σημείο:** $P(X=x)=0$ για κάθε x

Probability Density Function

Η Probability Density Function (PDF) είναι μια συνάρτηση που περιγράφει την κατανομή μιας συνεχούς τυχαίας μεταβλητής.

"Πόσο πιθανό είναι η τιμή να βρεθεί **κοντά** σε ένα συγκεκριμένο σημείο;" (όχι ακριβώς σε αυτό το σημείο)

Πυκνότητα ≠ Πιθανότητα

Για συνεχείς μεταβλητές (π.χ. ύψος, βάρος, χρόνος):

- Η πιθανότητα μια τιμή να είναι ακριβώς ίση με ένα νούμερο είναι πάντα μηδέν.
- Αυτό που έχει σημασία είναι η πιθανότητα η τιμή να βρίσκεται μέσα σε ένα διάστημα (π.χ. Ύψος ανθρώπου μεταξύ 1.70 και 1.80 μέτρων).

Expectations

Προσδοκία (ή Αναμενόμενη Τιμή ή Μαθηματική Ελπίδα)

Η μέση τιμή μιας συνάρτησης $f(x)$ κάτω από κατανομή πιθανότητας $p(x)$ όπου x είναι το τυχαίο μέγεθος (π.χ. θερμοκρασία) και η $f(x)$ είναι μια συνάρτηση αυτού του τυχαίου μεγέθους (π.χ. $f(x)=x^2$)

Απλούστερα: Η προσδοκία ενός τυχαίου μεγέθους είναι ένας τρόπος να περιγράψουμε το **μέσο αποτέλεσμα** που περιμένουμε μακροπρόθεσμα, αν επαναλάβουμε το πείραμα πολλές φορές.

$$\text{Διακριτή Κατανομή} \quad \mathbb{E}[f] = \sum_x p(x)f(x)$$

$$\text{Συνεχείς Μεταβλητές} \quad \mathbb{E}[f] = \int p(x)f(x) dx$$

$$\text{Δεσμευμένη Προσδοκία} \quad \mathbb{E}_x[f|y] = \sum_x p(x|y)f(x)$$

$$\mathbb{E}[f] \simeq \frac{1}{N} \sum_{n=1}^N f(x_n)$$

Conditional Expectation
(discrete)

Approximate Expectation
(discrete and continuous)

Variances and Covariances

$$\mathbb{E}[f] = \sum_x p(x)f(x)$$

Διακύμανση της $f(x)$: μέτρο μεταβλητότητας που υπάρχει στην $f(x)$ γύρω από τη μέση τιμή της $E[f(x)]$

$$\text{var}[f] = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])^2] = \mathbb{E}[f(x)^2] - \mathbb{E}[f(x)]^2$$

Συνδιακύμανση 2 τυχαίων μεταβλητών x, y : εκφράζει την έκταση που τα x και y μεταβάλλονται από κοινού

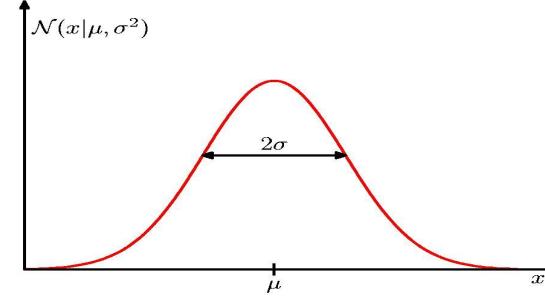
$$\begin{aligned} \text{cov}[x, y] &= \mathbb{E}_{x,y} [\{x - \mathbb{E}[x]\} \{y - \mathbb{E}[y]\}] \\ &= \mathbb{E}_{x,y}[xy] - \mathbb{E}[x]\mathbb{E}[y] \end{aligned}$$

$$\begin{aligned} \text{cov}[\mathbf{x}, \mathbf{y}] &= \mathbb{E}_{\mathbf{x}, \mathbf{y}} [\{\mathbf{x} - \mathbb{E}[\mathbf{x}]\}\{\mathbf{y}^T - \mathbb{E}[\mathbf{y}^T]\}] \\ &= \mathbb{E}_{\mathbf{x}, \mathbf{y}}[\mathbf{x}\mathbf{y}^T] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}^T] \end{aligned}$$

Gaussian Distribution

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

χ: μία μόνο πραγματική μεταβλητή
μ: μέσος, σ: τυπική απόκλιση

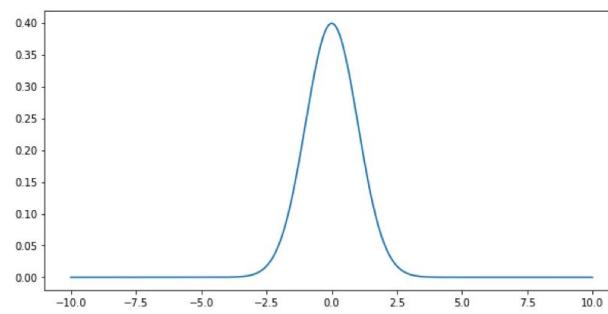


- Ικανοποιεί την ανισότητα $\mathcal{N}(x|\mu, \sigma^2) > 0$
- Κανονικοποιημένη $\int_{-\infty}^{\infty} \mathcal{N}(x|\mu, \sigma^2) dx = 1$

Normal Distribution

Density

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$



[1.5/2 Maximum Likelihood and Maximum a Posteriori](#)

Gaussian Mean and Variance

Μέση τιμή του x ή μέσος : $\mathbb{E}[x] = \int_{-\infty}^{\infty} \mathcal{N}(x|\mu, \sigma^2) x dx = \mu$

$$\mu = \mathbb{E}[x] \text{ hence } \hat{\mu} = \frac{1}{n} \sum_{i=1}^n x_i$$

Διακύμανση :
στην

$$\text{var}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 = \sigma^2$$

μέτρο μεταβλητότητας που υπάρχει

$$\sigma^2 = \mathbb{E}[(x-\mu)^2] \text{ hence } \hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu})^2$$

γύρω από τη μέση τιμή της $E[f(x)]$

Ροπή 2ης τάξης :

$$\mathbb{E}[x^2] = \int_{-\infty}^{\infty} \mathcal{N}(x|\mu, \sigma^2) x^2 dx = \mu^2 + \sigma^2$$

Συνάρτηση πιθανοφάνειας

Η συνάρτηση πιθανοφάνειας περιγράφει πόσο πιθανό είναι να παρατηρήσουμε τα δεδομένα μας, έχοντας ως δεδομένο συγκεκριμένες παραμέτρους του μοντέλου.

Συγκεκριμένα:

- Έστω ότι έχουμε ένα σετ παρατηρήσεων $\mathbf{x}=(x_1, x_2, \dots, x_n)$ και μια κατανομή πιθανότητας με παράμετρο θ .
- Η συνάρτηση πιθανοφάνειας $L(\theta)$ είναι η πιθανότητα να παρατηρήσουμε τα δεδομένα x , δεδομένης της παραμέτρου θ

Για διακριτές τυχαίες μεταβλητές, η πιθανοφάνεια $L(\theta|x)$ ορίζεται ως: $L(\theta|x)=P(x|\theta)$

Για συνεχείς τυχαίες μεταβλητές, η συνάρτηση πικνότητας πιθανότητας $f(x|\theta)$ χρησιμοποιείται αντί για την πιθανότητα P .

Πιθανότητα: τα δεδομένα είναι τυχαία και οι παράμετροι είναι σταθερές.

Πιθανοφάνεια: Θεωρούμε τα δεδομένα σταθερά και ψάχνουμε για τις παραμέτρους που κάνουν αυτά τα δεδομένα πιο πιθανά

Gaussian Parameter Estimation

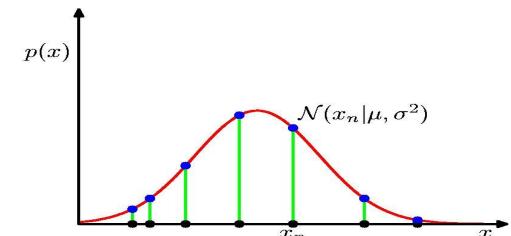
Likelihood function

$$p(\mathbf{x}|\mu, \sigma^2) = \prod_{n=1}^N \mathcal{N}(x_n|\mu, \sigma^2)$$

$$\mathbf{x}=(x_1, \dots, x_N)^T$$

(\mathbf{x} : i.i.d. Independent and identically distributed random variables)

Άγνωστα: μ, σ^2



Παράδειγμα i.i.d.: Ρίχνεις ένα κλασικό ζάρι 5 φορές:

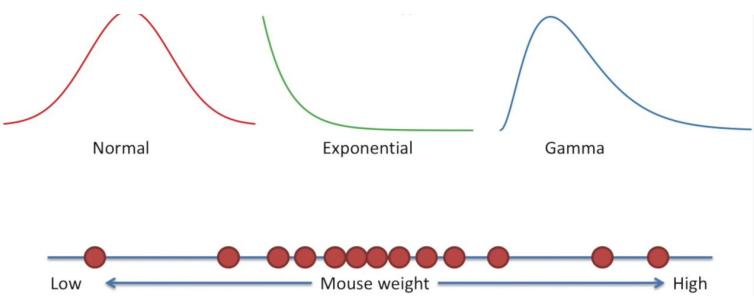
- Κάθε ρίψη δίνει μια τιμή από 1 έως 6
- Κάθε ρίψη δεν επηρεάζεται από τις προηγούμενες (ανεξάρτητη)
- Κάθε ρίψη έχει ίδια κατανομή (πιθανότητα 1/6 για κάθε πλευρά) (ομοιόμορφα κατανεμημένη)

Αρα οι μεταβλητές X1,X2,X3,X4,X5 είναι i.i.d.

Maximum (Log) Likelihood

Στόχος είναι να βρεθεί ο βέλτιστος τρόπος για να προσαρμοστεί μια κατανομή στα δεδομένα.

→ Ο λόγος που θέλουμε να προσαρμόσουμε μια κατανομή στα δεδομένα μας είναι ότι μπορεί να είναι πιο εύκολο να εργαστούμε με αυτήν και είναι επίσης πιο γενικό, ισχύει για κάθε πείραμα του ίδιου τύπου.

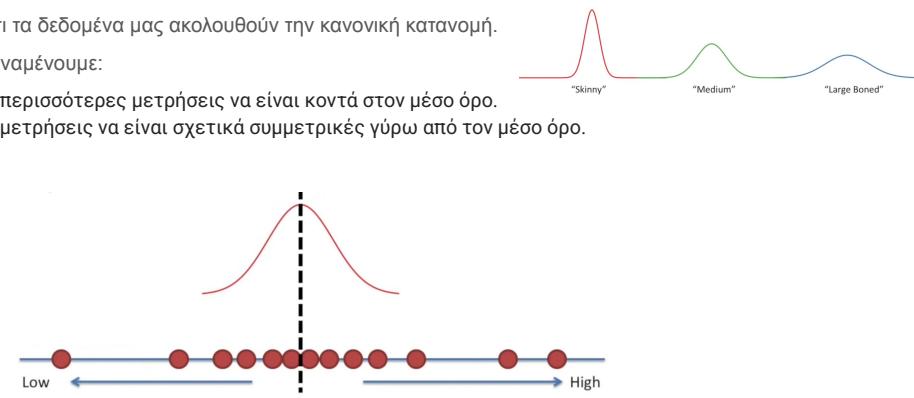


Maximum (Log) Likelihood

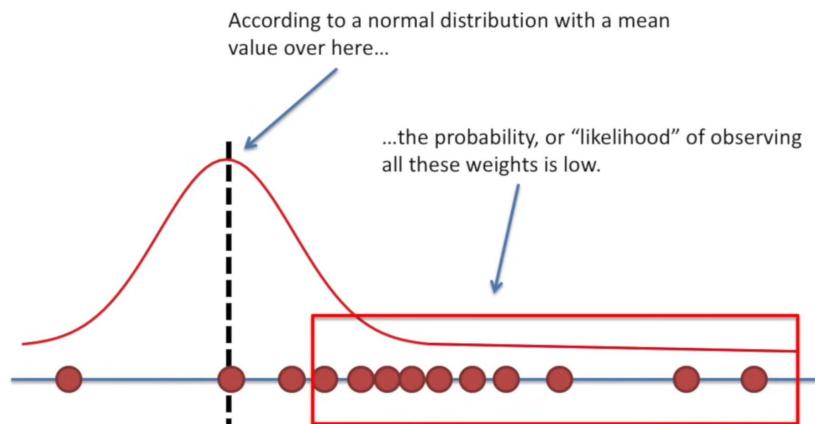
Έστω ότι τα δεδομένα μας ακολουθούν την κανονική κατανομή.

Οπότε αναμένουμε:

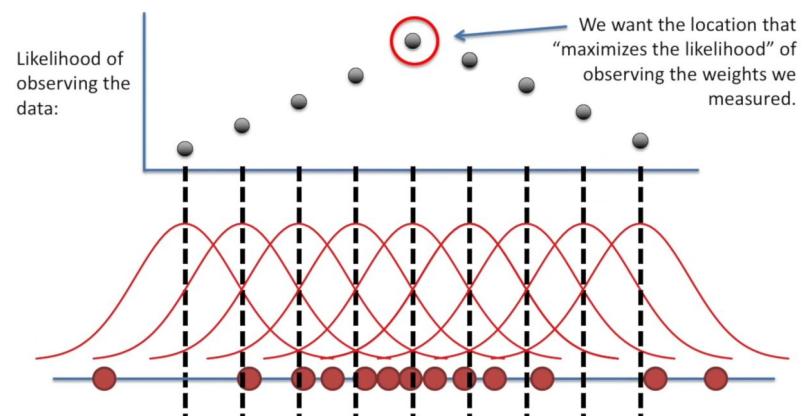
- οι περισσότερες μετρήσεις να είναι κοντά στον μέσο όρο.
- οι μετρήσεις να είναι σχετικά συμμετρικές γύρω από τον μέσο όρο.



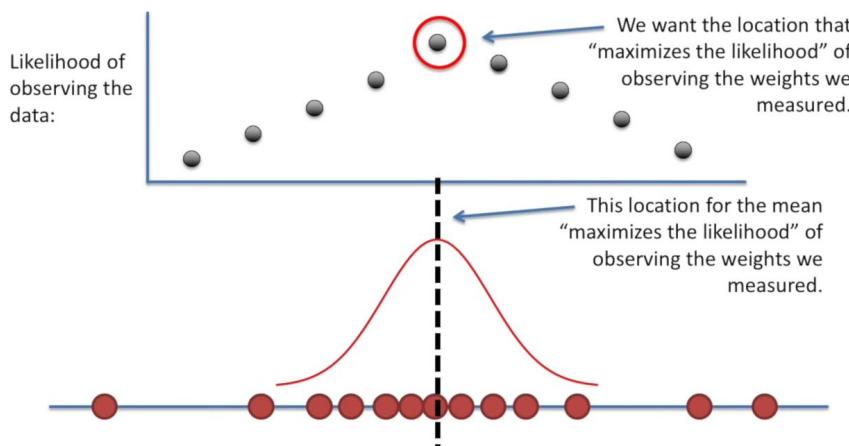
Maximum (Log) Likelihood



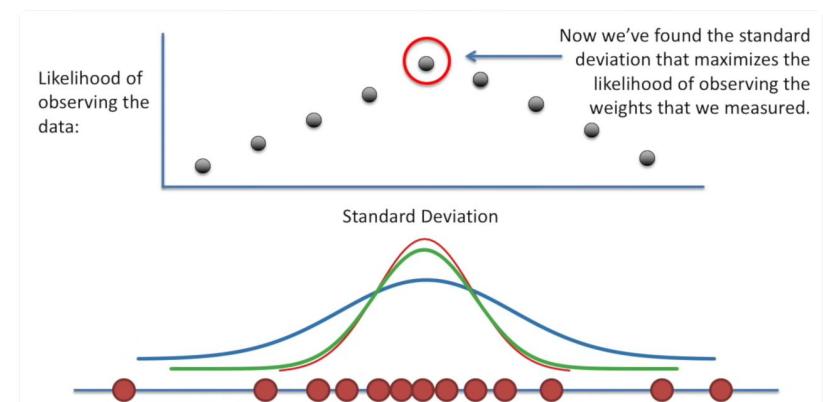
Maximum (Log) Likelihood



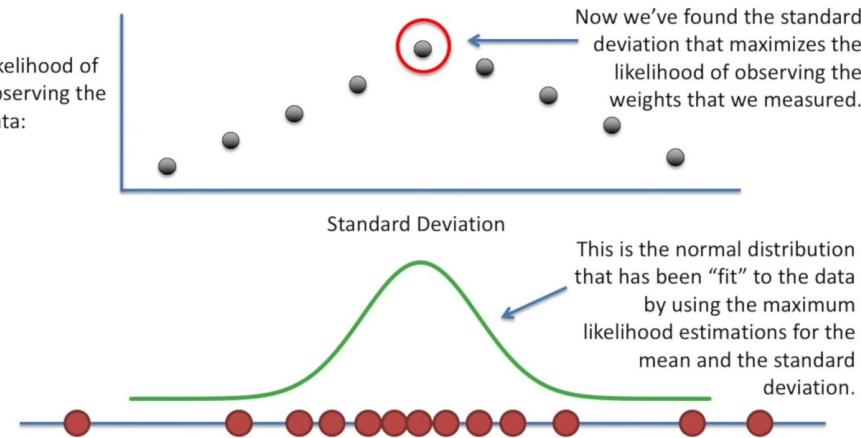
Maximum (Log) Likelihood



Maximum (Log) Likelihood



Maximum (Log) Likelihood



Likelihood

- Observe some data $X = \{x_1, \dots, x_n\}$
 - Assume that the data is drawn from a Gaussian
- $$p(X; \mu, \sigma^2) = \prod_{i=1}^n p(x_i; \mu, \sigma^2) = \prod_{i=1}^n \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right)$$
- **Fitting parameters is maximizing** $p(X; \mu, \sigma^2)$ wrt. μ, σ^2 (maximize likelihood that data was generated by model)
 - **Practical simplification**
- $$\underset{\mu, \sigma^2}{\text{maximize}} p(X; \mu, \sigma^2) \iff \underset{\mu, \sigma^2}{\text{minimize}} -\log p(X; \mu, \sigma^2)$$

Maximum Likelihood

- Estimate parameters by finding ones that explain the data

$$\underset{\mu, \sigma^2}{\text{minimize}} -\log p(X; \mu, \sigma^2)$$

- **Decompose likelihood**

$$-\log p(X; \mu, \sigma^2) = \sum_{i=1}^n \left(\frac{1}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2} (x_i - \mu)^2 \right) = \frac{n}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2$$

Minimize $\mu = \frac{1}{n} \sum_{i=1}^n x_i$

$$p(X; \mu, \sigma^2) = \prod_{i=1}^n p(x_i; \mu, \sigma^2) = \prod_{i=1}^n \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right)$$

Maximum Likelihood

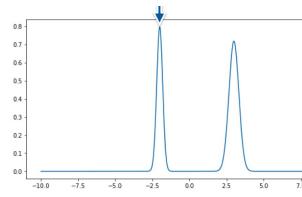
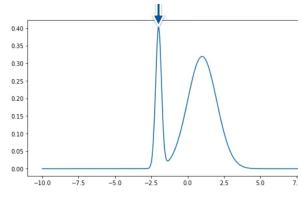
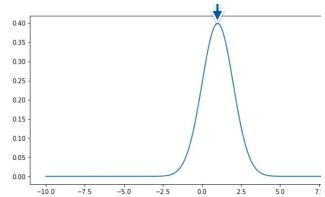
- Estimating the variance

$$\frac{n}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2$$

- Take derivatives with respect to it

$$\begin{aligned} \partial_{\sigma^2} [\cdot] &= \frac{n}{2\sigma^2} - \frac{1}{2\sigma^4} \sum_{i=1}^n (x_i - \mu)^2 = 0 \\ \implies \sigma^2 &= \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2 \end{aligned}$$

Maximum Likelihood Estimation



Κάτι μένει εκτός... χρησιμοποιώντας το MLE

Βιβλιογραφία

Strang Gilbert

→ Linear Algebra and Learning from Data (math.mit.edu/learningfromdata)

→ Introduction to Linear Algebra - Fifth Edition

→ [MIT 18.065 Matrix Methods in Data Analysis, Signal Processing, and Machine Learning](#)

Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong

→ [Mathematics for Machine Learning - Book](#)

→ [Mathematics for Machine Learning - github](#)

Ian Goodfellow, Yoshua Bengio, Aaron Courville

→ Deep Learning, [Chapter 2: Linear Algebra](#)

C. Bishop

→ [Pattern Recognition and Machine Learning.. Chapter 1](#)

[2.3. Linear Algebra — Dive into Deep Learning 0.16.1 documentation](#)

Παλινδρόμηση (Regression)

Μηχανική Μάθηση

ΔΠΜΣ Επιστήμης Δεδομένων & Μηχανικής Μάθησης

Γιώργος Αλεξανδρίδης – gealexan@mail.ntua.gr

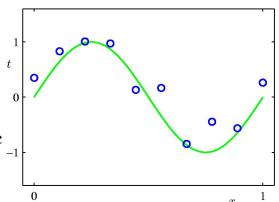
Εισαγωγικές Έννοιες

Ταξινόμηση και Παλινδρόμηση

- Δύο κύριες κατηγορίες της επιβλεπόμενης μάθησης (supervised learning)
 - Γνωστή και ως **μάθηση με παραδείγματα** (learning by examples)
 - Το σύστημα καλείται να μάθει την περιγραφή του μοντέλου από ένα **επιγεγραμμένο** (labelled) σύνολο δεδομένων, το οποίο αποτελείται από στυγμότυπα για τα οποία γνωρίζουμε την επιθυμητή έξοδο
- Ταξινόμηση** (Classification)
 - Η επιθυμητή έξοδος εντάσσεται σε μια ή περισσότερες διακριτές μεταξύ τους κατηγορίες
- Παλινδρόμηση** (Regression)
 - Η επιθυμητή έξοδος έχει ένα **συνεχές** πεδίο τιμών
 - λχ δεδομένης της οιμερινής ισοτιμίας δολαρίου και ευρώ, που θα είναι η αυριανή ισοτιμία;

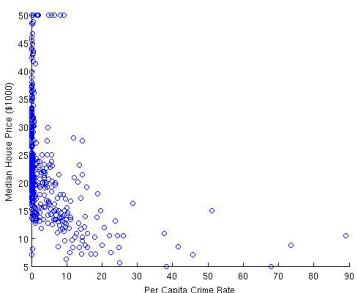
Παράδειγμα παλινδρόμησης

- Δεδομένα **μιας διάστασης**, τα οποία προέρχονται **ομοιόμορφα** από το πεδίο τιμών $x \in \mathbb{R}$
- Στην ετικέτα y μπορεί να έχει εμφιλοχωρήσει **θόρυβος**, δηλαδή $t(x) = f(x) + \epsilon$
- Στο διπλανό σχήμα, η πράσινη καμπύλη είναι αυτή που περιγράφει τη σχέση (συνάρτηση) ειωδόου-εξόδου
- Στόχος της παλινδρόμησης είναι να προσδιοριστεί αυτή η σχέση (συνάρτηση) από τις διαθέσιμες παρατηρήσεις (μπλε κουκίδες)
- Ερωτήματα**
 - Πως μπορούμε να παραμετροποιήσουμε το εν λόγω μοντέλο;
 - Ποια συγάρτηση οφάλματος μπορούμε να χρησιμοποιήσουμε για να ζειλογράψουμε την προσαρμογή του μοντέλου στα δεδομένα
 - Πως μπορούμε να γενικεύσουμε (generalization) σε νέα δεδομένα;



Παράδειγμα: Τιμές ακινήτων στη Βοστώνη

- Θέλουμε να εκτιμήσουμε το μέσο κόστος απόκτησης κατοικίας στη Βοστώνη συναρτήσει διαφόρων χαρακτηριστικών
- Επιλέγουμε ως πρώτο χαρακτηριστικό την εγκληματικότητα
- Αποτελεί καλό χαρακτηριστικό για το σκοπό μας;



5

Γραμμική Παλινδρόμηση

Linear Regression

6

Τυπικός Οριομός για μονοδιάστατα δεδομένα

- Έστω ότι έχουμε συλλογή δεδομένων \mathcal{D} που αποτελείται από N ζεύγη $\{(x^{(1)}, t^{(1)}), \dots (x^{(N)}, t^{(N)})\}$, όπου
 - $x \in \mathbb{R}$ το πεδίο οριομού της εισόδου
 - Στο προηγούμενο παράδειγμα, ο δείκτης εγκληματικότητας
 - $t \in \mathbb{R}$ το πεδίο τιμών των επικετών (συνεχής τιμές)
 - Στο προηγούμενο παράδειγμα, η τιμή του ακινήτου
- Μοντελοποιύμε τη σχέση εισόδου-εξόδου υπό τη μορφή *πρωτοβάθμιας (γραμμικής) εξίσωσης*

$$y = w_0 + w_1 x$$

- Χωρίζουμε τη συλλογή δεδομένων μας σε δύο μη επικαλυπτόμενα σύνολα
 - Δεδομένα εκπαίδευσης:** Χρησιμοποιούνται για την κατασκευή της υπόθεσης
 - Απλοδή της συνάρτησης που απεικονίζει τα δεδομένα εισόδου x στην έξοδο y
 - Δεδομένα ελέγχου:** Επαληθεύουν την υπόθεση

7

Θόρυβος

- Το γραμμικό μοντέλο είναι αρκετά απλό και μπορεί να μην προσαρμόζεται στα δεδομένα
 - Αυτή η έλλειψη προσαρμογής μπορεί να μοντελοποιηθεί ως **θόρυβος (noise)**
- Πηγές θορύβου**
 - Θόρυβος εισόδου:** Ανακριβείς στις ιδιότητες/χαρακτηριστικά των δεδομένων
 - η θόρυβος στην καταγραφή των δεικτών εγκληματικότητας
 - Θόρυβος εξόδου:** Λανθασμένη επισημείωση των δεδομένων
 - η θόρυβος στην καταγραφή της ακριβούς τιμής των ακινήτων
 - Λανθάνουσες μεταβλητές (latent variables):** Επιπλέον χαρακτηριστικά που δεν έχουν ληφθεί υπόψη επηρεάζουν τη σχέση εισόδου-εξόδου
 - Ποιο άλλο μέγεθος θα μπορούσε να επηρέάσει τις τιμές των ακινήτων;
 - Χωρητικότητα (capacity) μοντέλου:** το μοντέλο μπορεί να είναι πολύ απλό για να περιγράψει τη σχέση εισόδου-εξόδου

8

Εύρεση Παραμέτρων: 1ος Τρόπος

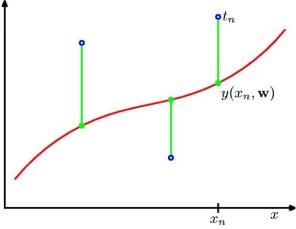
- Μέθοδος συνήθων ελαχίστων τετραγώνων
(ordinary least squares)

- Ελαχιστοποιεί το άθροισμα των τετραγώνων της διαφοράς (υπολογίου) μεταξύ ομποίου και ευθείας

- Οδηγεί σε λύση κλειστής μορφής
 $w = (X^T X)^{-1} X^T t$

- Προσέγγιση αμερόληπτη και ουνεπής αν

- Τα οφάλματα έχουν πεπερασμένη διακύμανση
 - Αυτό συνήθως λογώνει στην πράξη
- Ασυνχέπτωτα με το μοντέλο
 - Αυτό συνήθως δεν λογώνει στην πράξη, καθώς μπορεί να υπάρχουν λανθάνουσες συμμεταβλητές (latent covariates) που σχετίζονται με την παρατηρούμενη μεταβλητή και την έξοδο
 - η ίδια παράγοντας που επηρεάζουν τις τιμές των ακινήτων πλην της εγκληματικότητας



9

Εύρεση παραμέτρων: 2ος Τρόπος

- Κατάβαση κλήσης (gradient descend)

- Αρχικοποίηση w σε τυχαίες τιμές
 - Ενημέρωση w σε κατεύθυνση αντίθετη της κλήσης $w \leftarrow w - \eta \frac{\partial L(y, t)}{\partial w}$
 - η ρυθμός μάθησης

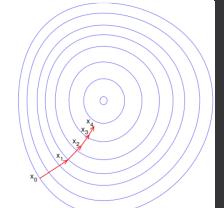
$$L(y^{(n)}, t^{(n)}) = (t^{(n)} - y^{(n)})^2$$

- Στοχαστική κατάβαση κλήσης (stochastic gradient descend – SGD)

- Κατά την t -οστιή επανάληψη του αλγορίθμου εκπαίδευσης εξετάζουμε το n -οστό παράδειγμα εκπαίδευσης
 - $w^{(t)} \leftarrow w^{(t-1)} + 2\eta(t^{(n)} - y^{(n)})x^{(n)}$
 - Σφάλμα $\epsilon = t^{(n)} - y^{(n)}$
 - Όσο το οφάλμα προσεγγίζει το 0, η ενημέρωση περιορίζεται (το w σταματά να μεταβάλλεται)
 - Που αλληλ αναλογία βλέπετε;

- Ενημέρωση κατά δέσμη (batch update)

- N μέγεθος δέσμης (ομάδας παραδειγμάτων)
 - $w^{(t)} \leftarrow w^{(t-1)} + 2\eta \frac{1}{N} \sum_{n=1}^N (t^{(n)} - y^{(n)})x^{(n)}$



10

Πολυδιάστατα δεδομένα

- Ένας τρόπος επέκτασης του μοντέλου είναι να λάβουμε υπόψη μας τον γραμμικό συνδυασμό και άλλων διαστάσεων της εισόδου (χαρακτηριστικών)

$$y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$$

- Στο παραδείγμα των τιμών ακινήτων στη Βοστώνη, μπορούμε να λάβουμε υπόψη μας την έκταση του ακινήτου, τον αριθμό των δωματίων, ...

- Στη γενική περίπτωση, τα δεδομένα εισόδου έχουν d διαστάσεις.

$$\mathbf{x} = (x_1, \dots, x_d)$$

- Μοντέλο: $y(\mathbf{x}) = w_0 + \sum_{j=1}^d w_j x_j = \mathbf{w}^T \mathbf{x}$

11

Πολυωνυμική Παλινδρόμηση

Polynomial Regression

12

Πολυωνυμικά μοντέλα

- Πιο σύνθετα μοντέλα, μοντελοποιούν και μη-γραμμικούς ουνδυασμούς μεταξύ των χαρακτηριστικών της εισόδου

- Πολυωνυμικό μοντέλο M -οσής τάξης

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^M w_j \mathbf{x}^j$$

- Γραμμική παλινδρόμηση ειδική περίπτωση πολυωνυμικής παλινδρόμησης για $M = 1$

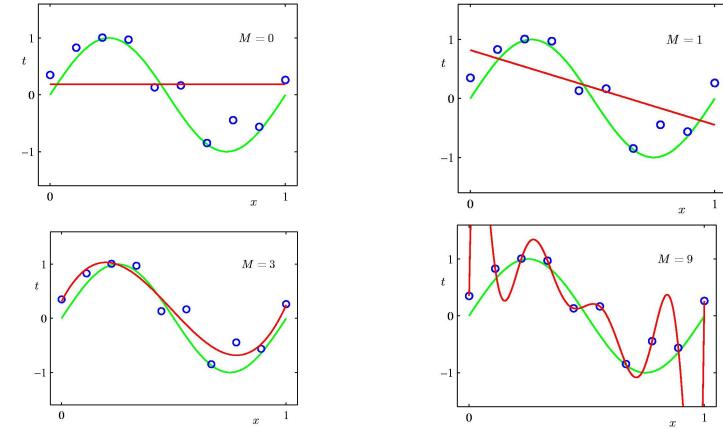
- Γιατί μας ενδιαφέρουν:

- Θεώρημα προσέγγισης Weierstrass:** Κάθε συνεχής συνάρτηση f πολ λαμβάνει πραγματικές τιμές εντός διαστήματος $[a, b]$ μπορεί να προσέγγισεται υπό τη μορφή πολυωνύμων για οποιοδήποτε επιθυμητό μέγεθος ακριβείας $\varepsilon > 0$
- K. Weierstrass (1885). Über die analytische Darstellbarkeit sogenannter willkürlicher Functionen einer reellen Veränderlichen. Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin, 1885 (II).

- Ποιες άλλες θεωρήματα/τεχνικές προσέγγισης γνωρίζετε;

13

Ποιο μοντέλο προσαρμόζεται καλύτερα στα δεδομένα;



14

Τρόπος Εργασίας

1. Αφελής εξαντλητική αναζήτηση (Naïve Exhaustive Search)

- Ξεκινάντας από τη γραμμική παλινδρόμηση, ανζίνονται συνεχώς την τάξη του μοντέλου μέχρι το σφάλμα στα δεδομένα επαλήθευσης (validation data) να σταματήσει να μειώνεται.

2. Περιορισμός της χωρητικότητας του μοντέλου

- Ξεκινάμε από ένα μοντέλο **πολύ μεγάλης** χωρητικότητας και μέσω των δεδομένων εκπαίδευσης και επαλήθευσης προσπαθούμε να **περιορίσουμε** τον χώρο υποθέσεων του
- Στην προηγούμενη διαφάνεια, το πολυωνυμικό μοντέλο 9th τάξης έκανε πιο περιλογες υποθέσεις για τη σχέση εισόδου-εξόδου από την πραγματικότητα, με αποτέλεσμα να **υπερπροσαρμόζεται** (*overfit*) στα δεδομένα εκπαίδευσης
- Οι συντελεστές του πολυωνύμου λάμβαναν πολύ μεγάλες, κατ' απόλυτη τιμή, τιμές
- Ένας τρόπος αποφυγής της υπερπροσαρμογής είναι να κάνεις το μοντέλο να ψάχνει για λύσεις που έχουν μικρά, κατ' απόλυτη τιμή, βάρη **Ομαλοποίηση** (*Regularization*)

15

Αμφικλινής παλινδρόμηση (Ridge Regression)

- Εναλλακτικές ονομασίες:** **Φθορά βαρών** (*weight decay*) και **ομαλοποίηση Tikhonov** (*Tikhonov regularization*)
- Προσθήκη δρού ομαλοποίησης στη συνάρτηση σφάλματος L , ο οποίος είναι ανάλογος του τετραγώνου του διανύσματος των βαρών $\Omega(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|_2^2 = \frac{1}{2} \mathbf{w}^T \mathbf{w}$
 - Νόρμα Frobenius ή Ευκλείδεια νόρμα
- $\tilde{L}(\mathbf{w}; \mathbf{X}, \mathbf{y}) = L(\mathbf{w}; \mathbf{X}, \mathbf{y}) + \alpha \frac{1}{2} \mathbf{w}^T \mathbf{w}$, όπου \tilde{L} ομαλοποιημένη συνάρτηση σφάλματος
- Ενημέρωση βαρών μέσω στοχαστικής κατάβασης κλίσης (SGD)
 - Υπολογισμός κλίσης: $\nabla_{\mathbf{w}} \tilde{L}(\mathbf{w}; \mathbf{X}, \mathbf{y}) = \nabla_{\mathbf{w}} L(\mathbf{w}; \mathbf{X}, \mathbf{y}) + \alpha \mathbf{w}$
 - Ενημέρωση βαρών
 - $\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \eta (\nabla_{\mathbf{w}^{(t)}} \tilde{L}(\mathbf{w}^{(t)}; \mathbf{X}, \mathbf{y}) + \alpha \mathbf{w}^{(t)}) \Rightarrow \mathbf{w}^{(t+1)} \leftarrow (1 - \eta \alpha) \mathbf{w}^{(t)} - \eta \nabla_{\mathbf{w}^{(t)}} \tilde{L}(\mathbf{w}^{(t)}; \mathbf{X}, \mathbf{y})$
 - Το αποτέλεσμα είναι να **μειώνεται** («φθείρεται») το εύρος του διανύσματος των βαρών κατά παράγοντα $(1 - \eta \alpha)$ σε κάθε βήμα.
 - α βαθμός ομαλοποίησης (υπερ-παραμέτρος του μοντέλου)

Αμφικλινής παλινδρόμηση : Επίδραση στη διαδικασία μάθησης (1/3)

- Έστω w^* το διάνυσμα βαρών που ελαχιστοποιεί την L
- Αντικατάσταση της L από την τετραγωνική της προσέγγιση \hat{L} γύρω από το w^*
 - Ειδικά για προβλήματα γραμμικής παλινδρόμησης όπου η αντικεμενική συνάρτηση υπολογίζεται διαφορές τετραγώνων (π.χ. MSE), η προσέγγιση είναι τέλεια
 - $\hat{L}(w) = L(w^*) + \frac{1}{2}(w - w^*)^T H(w - w^*)$
 - Δεν υπάρχει πρωτοβάθμιος όρος μιας εξ ορισμού είναι ελάχιστο και άρα η κλίση γύρω από το $w - w^*$ είναι (οχεδόν) μηδενική
 - H : Εστιανός Πίνακας (Hessian Matrix) όλων των $\frac{\partial^2 L}{\partial w_i \partial w_j}$
 - Επειδή βρισκόμαστε γύρω από ελάχιστο, ο H είναι θετικά ημι-καθοριομένος

Αμφικλινής παλινδρόμηση : Επίδραση στη διαδικασία μάθησης (2/3)

- Η \hat{L} ελαχιστοποιείται στα σημεία όπου η κλίση της $\nabla_w \hat{L}(w; X, y) = H(w - w^*)$ γίνεται ίση με το 0
- Προσθέτοντας τον όρο πιονής $\alpha \frac{1}{2} w^T w$ το τοπικό ελάχιστο αλλάζει και γίνεται πλέον \tilde{w}
 - $\nabla_w (\hat{L}(\tilde{w}) + \frac{\alpha}{2} \tilde{w}^T \tilde{w}) = 0 \Rightarrow H(\tilde{w} - w^*) + \alpha \tilde{w} = 0 \Rightarrow \tilde{w} = (H + \alpha I)^{-1} H w^*$
- Όταν $\alpha \rightarrow 0$, το \tilde{w} προσεγγίζει το w^*
- Όταν $\alpha \neq 0$, χρησιμοποιούμε αποσύνθεση ιδιοτυπών (eigen decomposition) $H = Q \Lambda Q^T$
 - Εφικτό μιας και H ουμετρικός με πραγματικές τιμές
 - Λ διαγώνιος πίνακας ιδιοτυπών, Q ορθο-κανονικός πίνακας ιδιοδιανυσμάτων
 - $\tilde{w} = Q(\Lambda + \alpha I)^{-1} \Lambda Q^T w^*$

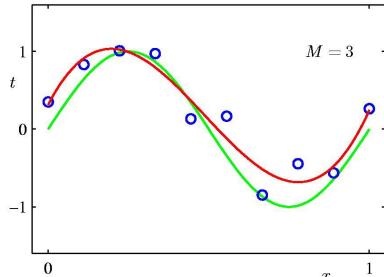
Αμφικλινής παλινδρόμηση : Επίδραση στη διαδικασία μάθησης (3/3)

- Ουσιαστικά το w^* αναπροσαρμόζεται στην κατεύθυνση των αξόνων που ορίζονται από τα ιδιοδιανυσμάτα του H κατά ένα παράγοντα $\frac{\lambda_i}{\lambda_i + \alpha}$
 - Για μεγάλες ιδιοτυπές ($\lambda_i \gg \alpha$), η επίδραση της ομαλοποίησης είναι πολύ μικρή
 - Για μικρές ιδιοτυπές ($\lambda_i \ll \alpha$), η επίδραση της ομαλοποίησης είναι πολύ μεγάλη
 - Συρρικνώνει την επίδραση των αντιστοιχών ιδιοδιανυσμάτων στο 0

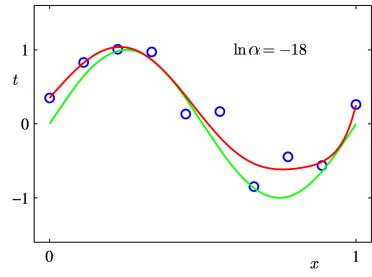
Αμφικλινής Γραμμική Παλινδρόμηση

- Συνάρτηση σφάλματος: τετραγωνικό οφάλμα
 - $L(w) = (Xw - y)^T (Xw - y)$
 - Βέλτιστο διάνυσμα βαρών: $w^* = (X^T X)^{-1} X^T y$
- Προσθήκη όρους ομαλοποίησης αμφικλινούς παλινδρόμησης
 - $L(w) = (Xw - y)^T (Xw - y) + \frac{\alpha}{2} w^T w$
 - Βέλτιστο διάνυσμα βαρών: $\tilde{w} = (X^T X + \alpha I)^{-1} X^T y$
- Ο όρος $X^T X$ είναι αναλόγος του πίνακα συνδιασποράς των χαρακτηριστικών της εισόδου
 - Διαγώνιες τιμές: Αντιστοιχούν στη διακύμανση των χαρακτηριστικών της εισόδου
 - Η αμφικλινής παλινδρόμηση αναγκάζει τον αλγόριθμο μάθησης να θεωρήσει ότι η εισόδος παρουσιάζει μεγαλύτερη διακύμανση και συνεπώς να μικρύνει τα βάρη εκείνα που εμφανίζουν μικρότερη συνδιασπορά.

Παράδειγμα αμφικλινούς παλινδρόμησης



Πολυωνυμικό μοντέλο 3^{ης} τάξης



Πολυωνυμικό μοντέλο 9^{ης} τάξης με χρήση αμφικλινούς παλινδρόμησης

Παλινδρόμηση Lasso (Lasso Regression)

- Προσθήκη όρου ομαλοποίησης στη συνάρτηση οφάλματος L , ο οποίος είναι ανάλογος της **απόλυτης τιμής** του διανύσματος των βαρών $\Omega(\mathbf{w}) = \|\mathbf{w}\|_1$
- Ομαλοποιημένη αντικειμενική συνάρτηση
 - $\tilde{L}(\mathbf{w}; \mathbf{X}, \mathbf{y}) = L(\mathbf{w}; \mathbf{X}, \mathbf{y}) + \alpha \|\mathbf{w}\|_1$
- Υπολογισμός κλίσης: $\nabla_{\mathbf{w}} \tilde{L}(\mathbf{w}; \mathbf{X}, \mathbf{y}) = \nabla_{\mathbf{w}} L(\mathbf{w}; \mathbf{X}, \mathbf{y}) + \alpha \text{sgn}(\mathbf{w})$
- Διαφορά** ως προς αμφικλινή παλινδρόμηση
 - Η συνειφορά της ομαλοπίδησης στην κλίση **δεν** είναι πλέον **ανάλογη** του εύρους w_i της κάθε παραμέτρου, αλλά του προσήμου της
 - Για να προχωρήσουμε, κάνουμε την **επιπλέον παραδοχή** ότι ο εσωτερικός πίνακας είναι **διαγώνιος** με $H_{i,i} > 0$
 - Έχουν αφαιρεθεί οι συσχετίσεις μεταξύ των χαρακτηριστικών της εισόδου λ.-χ. με την προεπεξεργασία μέσω PCA

21

Χαρακτηριστικά Παλινδρόμησης Lasso

- Τετραγωνική προσέγγιση \hat{L} της L
 - $\hat{L}(\mathbf{w}) = L(\mathbf{w}^*) + \sum_i \left[\frac{1}{2} H_{i,i} (\mathbf{w} - \mathbf{w}^*)^2 + \alpha |w_i| \right]$
- Τοπικό ελάχιστο: $\tilde{w}_i = \text{sgn}(w_i^*) \max \left(|w_i^*| - \frac{\alpha}{H_{i,i}}, 0 \right)$
 - Όταν $|w_i^*| \leq \frac{\alpha}{H_{i,i}}$, τότε το \tilde{w}_i γίνεται 0
 - Όταν $|w_i^*| > \frac{\alpha}{H_{i,i}}$, τότε το \tilde{w}_i «σύρεται» προς το 0 κατά έναν όρο $\frac{\alpha}{H_{i,i}}$
- Η παλινδρόμηση Lasso οδηγεί σε πιο **αραιές** (*sparse*) αναπαραστάσεις σε σύγκριση με την αμφικλινή παλινδρόμηση
 - Υπό την έννοια ότι **περισσότερες παράμετροι** έχουν **μηδενικές τιμές**
 - Χρησιμοποιείται ως μηχανισμός **επιλογής χαρακτηριστικών** (*feature selection*)

Λογιστική Παλινδρόμηση

Logistic Regression

24

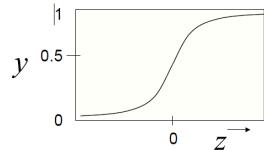
Λογιστική Παλινδρόμηση

- Χρήση σε προβλήματα τα οποία έχουν **διακριτή** έξοδο, ωστόσο εμάς μας ενδιαφέρει να την **εκφράσουμε** υπό το πρώτομα **συνεχούς τιμής**
 - Πχ για συγκεκριμένες τιμές δεικτών στις εξετάσεις αίματος ασθενούς, ποια η πιθανότητα να εμφανίσεται στεφανιατά νόος;
- Επέκταση γραμμικής παλινδρόμησης μέσω της προσθήκης **σιγμοειδούς συνάρτησης**

$$y = \sigma(\mathbf{w}^T \mathbf{x} + w_0)$$

- Σιγμοειδής συνάρτηση

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



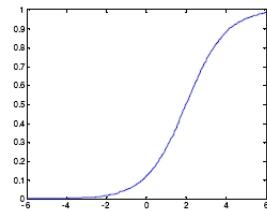
- Έξοδος αποτελεί **ομαλή** συνάρτηση εισόδου και βαρών
 - Λαμβάνει τιμές στο [0,1]
- Τι μας θυμίζει;

25

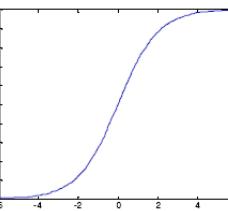
Μορφή λογιστικής παλινδρόμησης

- Η μεταβολή των βαρών w αλλάζει τη μορφή της συνάρτησης
$$y = \sigma(w_1 x + w_0)$$

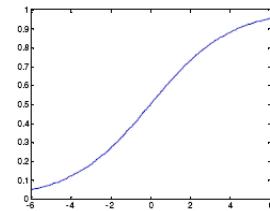
$$w_0 = 0, w_1 = 1$$



$$w_0 = 0, w_1 = 0.5$$



$$w_0 = -2, w_1 = 1$$

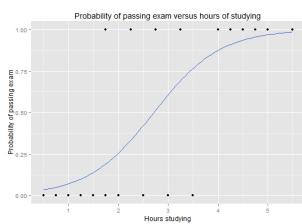


Παράδειγμα

- Γνωρίζοντας τις ώρες που αφέρωσε στη μελέτη ένας σπουδαστής/στρια, θα περάσει το διαγώνισμα;
- Δεδομένα εκπαίδευσης

Hours	0.50	0.75	1.00	1.25	1.50	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
Pass	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1

- Πρόβλημα λογιστικής παλινδρόμησης
 - Εύρεση βαρών w (πχ μέσω κατάβασης κλίσης) που θα μας επιτρέπουν να κάνουμε προβλέψεις



Ώρες Μελέτης	Πιθανότητα επιτυχίας στο διαγώνισμα
1	0.07
2	0.26
3	0.61
4	0.87
5	0.97

27

Βιβλιογραφία

- Christopher M. Bishop "Pattern Recognition and Machine Learning" – Springer (<https://link.springer.com/book/9780387310732>)
 - Γραμμικά Μοντέλα για Παλινδρόμηση (§3)
- Ian Goodfellow, Yoshua Bengio, Aaron Courville "Deep Learning" – MIT Press (<https://www.deeplearningbook.org/>)
 - Ομαλοποίηση μέσω της προσθήκης όρων ποντής (§7.1)

26

Εισαγωγή στην Ταξινόμηση

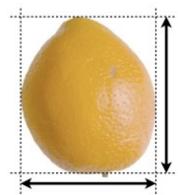
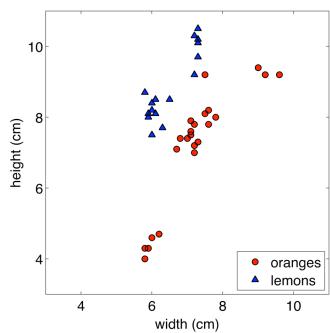
Μηχανική Μάθηση

ΔΠΜΣ Επιστήμης Δεδομένων & Μηχανικής Μάθησης

Γιώργος Αλεξανδρίδης – gealexan@mail.ntua.gr

1

Πρόβλημα ταξινόμησης: Πορτοκάλια ή Λεμόνια;



3

Κατηγορίες ταξινομητών

1. Διαιμεριστικοί (divisive)

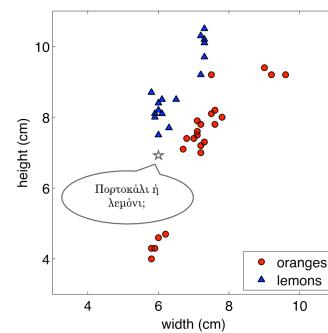
- Προσπαθούν να τεμαχίσουν το χώρο των δεδομένων σε μη-επικαλυπτόμενες υπό-περιοχές εντός των οποίων βρίσκονται δεδομένα μιας κλάσης
- Ταξινομητές πλησιέστερων γενιών, δέντρα αποφάσεων, (πολυεπίπεδα) νευρωνικά δίκτυα πρόσθιας τροφοδότησης, μηχανές διανυσμάτων υποστήριξης, ...

2. Παραγωγικοί (generative)

- Στατιστική θεώρηση των δεδομένων
- Προσπαθούν να μάθουν την υποκείμενη κατανομή (underlying distribution) που παράγει τα δεδομένα
- Αφελείς μπεζιάνοι ταξινομητές, γκαουσιανά μοντέλα μιξης, κρυφά μαρκοβιανά μοντέλα, ...

2

Ταξινόμηση και Επαγωγή (Induction)



4

Εκμάθηση Μέσω Παραδειγμάτων

Instance-based Learning

5

Εκμάθηση μέσω Παραδειγμάτων

- **Μη-παραμετρικά μοντέλα (non-parametric models)**
- Πρόκειται για απλά μοντέλα τα οποία χρησιμοποιούνται για την προσέγγιση προβλημάτων συνεχών τιμών (παλινδρόμηση) ή διακριτών τιμών (ταξινόμηση)
 - Ταξινομητές *k*-πλησιέστερων γειτόνων (*k*-nearest neighbors classifier – *kNN*), δίκτυα ακτινικών συναρτήσεων βάσης (*radial basis function* – *RBF*), ..
- Η διαδικασία της μάθησης είναι ισοδύναμη με την αποθήκευση των **δεδομένων** που χρησιμοποιούνται για την **εκπαίδευση** του μοντέλου (*training data*)
- Τα νέα στιγμότυπα ταξινομούνται χρησιμοποιώντας «օμοειδή» στιγμότυπα από το σύνολο εκπαίδευσης
 - Τα νέα στιγμότυπα αναφέρονται επίσης και ως **στιγμότυπα δοκιμής (test instances)** ή **δεδομένα δοκιμής (test data)**
- Κωδικοποιούν λογικές **υποκείμενες υποθέσεις (underlying assumptions)**
 - Οι κλάσεις (έξοδος) μεταβάλλονται «**ομαλά (smooth)**» συναρτήσεις της εισόδου
 - Τα δεδομένα καταλαμβάνουν έναν υπο-χώρο του αρχικού χώρου μεγάλων διαστάσεων

Ταξινομητής Πλησιέστερου Γείτονα

- Έστω ότι το σύνολο T των δειγμάτων δεδομένων εκπαίδευσης ανήκουν στον Ευκλείδειο χώρο ($x \in \mathbb{R}^d$)

Δευτουργία

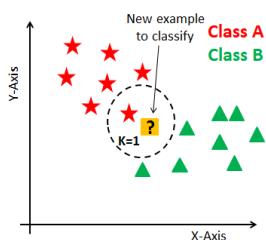
- Η επικέτα που λαμβάνει κάθε νέο δείγμα είναι η επικέτα του πλησιέστερου δείγματος δεδομένων εκπαίδευσης
- Συνήθως χρησιμοποιείται η Ευκλείδεια απόσταση

$$\|x^{(a)} - x^{(b)}\|_2 = \sqrt{\sum_{j=1}^d (x_j^{(a)} - x_j^{(b)})^2}$$

Αλγόριθμος

1. Βρες το παράδειγμα $(x^*; t^*)$ από τα δεδομένα εκπαίδευσης το οποίο είναι «εγγύτερων στο νέο στιγμότυπο x
$$x^* = \underset{x^{(i)} \in T}{\operatorname{argmin}} \operatorname{distance}(x^{(i)}, x)$$
1. Ανάθεσε στο x την επικέτα t^* : $(x; t^*)$

- Στην πραγματικότητα, δεν χρειάζεται να υπολογίσουμε την ρίζα
 - Γιατί:



7

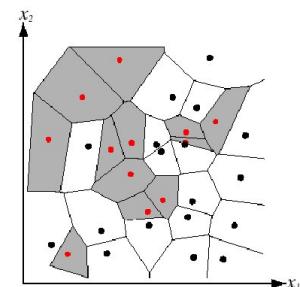
Όρια απόφασης

- Ο ταξινομητής πλησιέστερου γείτονα δεν υπολογίζει άμεσα όρια απόφασης μεταξύ των κλάσεων

- Ωστόσο, αυτά προκύπτουν έμμεσα

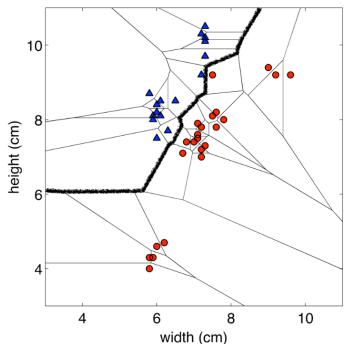
Όρια απόφασης: Διαγράμμα Voronoi

- Ο χώρος των δεδομένων χωρίζεται σε διαφορετικές περιοχές ανάλογα με την επικέτα (κλάση) τους
- Τα ευθύγραμμα τιμήματα αποτελούν τις μεσοκαθέτους μεταξύ δύο «εγγενετικών» στιγμοτύπων των δεδομένων εκπαίδευσης



8

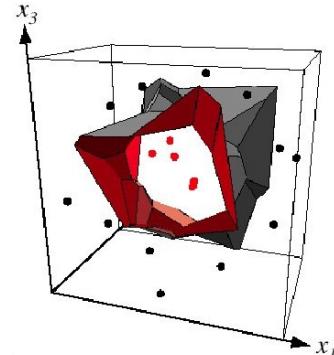
Πορτοκάλια ή Λεμόνια: Όριο απόφασης



Σύνθετο όριο απόφασης (όχι ευθύγραμμο τμήμα)

9

Όριο απόφασης στις 3 διαστάσεις



10

k -πλησιέστεροι γείτονες

- Ο ταξινομητής πλησιέστερου γείτονα είναι «εναίσθητος» στην ύπαρξη θορίδων στα δεδομένα

Δύση

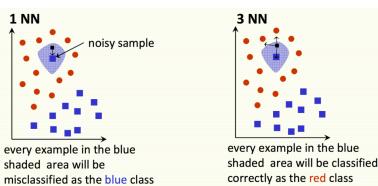
- «Ομαλοποίηση» της ταξινόμησης μέσω «ηγετοφορίας» των k πλησιέστερων γειτόνων

Αλγόριθμος

- Βρες τα k πλησιέστερα παράδειγμα ($x^{(i)}; t^{(i)}$) από τα δεδομένα εκπαίδευσης τα οποία είναι «εγγύτερα» στο νέο στιγμιότυπο x

- Ανάθεσε στο x την ετικέτα y :

$$y = \underset{t^{(x)}}{\operatorname{argmax}} \sum_{r=1}^k \delta(t^{(x)}, t^{(r)})$$



11

Προσδιορισμός k

- Πως βρίσκουμε την **κατάλληλη** τιμή για το k :
 - Όσο μεγαλύτερο είναι, τόσο βελτώνεται η απόδοση του ταξινομητή
 - Ωστόσο αν είναι αρκετά μεγάλο, τότε το εύρος της «γειτονιάς» μεγαλώνει, συμπεριλαμβανοντας δείγματα τα οποία μπορεί να θρίσκονται πολύ μακριά από το νέο στιγμιότυπο
- Εύρεση k μέσω τεχνικών **διασταυρούμενης επικύρωσης** (cross-validation)
 - Θα μιλήσουμε σε επόμενες διαλέξεις και στο εργαστήριο για αυτές
- Εμπειρικός κανόνας
 - $k < \sqrt{N}$, όπου N το πλήθος των παραδειγμάτων εκπαίδευσης

12

k -πλησιέστεροι γείτονες: ζητήματα πολυπλοκότητας

• Υψηλή πολυπλοκότητα κατά τη διαδικασία ελέγχου

- Για να δρούμε έναν πλησιέστερο γείτονα πρέπει να υπολογίσουμε την απόσταση από όλα τα δεδομένα εκπαιδευσης

• Δύσεις

- Χρησιμοποίηση υποσύνδου των διαστάσεων, χρήση αποδοτικών δομών δεδομένων (πχ kd-trees), υπολογισμός προσεγγιστικής απόστασης, αφαίρεση πλεονασματικών δεδομένων, ...

• Υψηλές απατήσεις αποθήκευσης

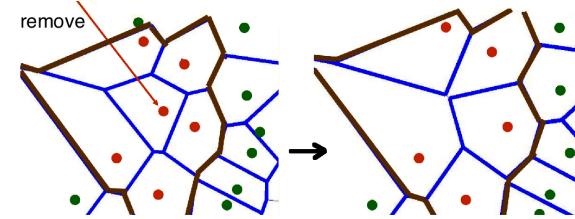
- Πρέπει να αποθηκεύσουμε στη μνήμη όλα τα δεδομένα εκπαιδευσης
- Δύσεις
 - Αφαίρεση πλεονασματικών δεδομένων

• Δεδομένα πολλών διαστάσεων («Κατάρα διαστατικότητας»)

- Το πλήθος των απαιτούμενων δεδομένων εκπαίδευσης αυξάνει εκθετικά όσο αυξάνουν οι διαστάσεις
- Επίσης αυξάνει και το υπολογιστικό κόστος
- Δύσεις
 - Εφαρμογή τεχνικών μείωσης διαστάσεων και επιλογής χαρακτηριστικών

13

Αφαίρεση πλεονασματικών δεδομένων



Αν όλοι οι γείτονες έχουν την ίδια κλάση, μπορούμε να αφαίρεσουμε το δείγμα δεδομένων

Εφαρμογή Ταξινόμησης Πλησιέστερων Γειτόνων

• Σε ποιο μέρος τραβήχτηκε η ακόλουθη φωτογραφία;

- James Hays, Alexei A. Efros. im2gps: estimating geographic information from a single image. CVPR'08. Project page: <http://graphics.cs.cmu.edu/projects/im2gps/>



15

Που τραβήχθηκε η ακόλουθη φωτογραφία;

• Δεδομένα εκπαίδευσης

- 6 εκ. εικόνες από το Flickr που περιέχουν μεταδεδομένα τοποθεσίας
- Αρκετά πικνή δειγματοληψία σε όλη την υφήλιο

- Αναπαράσταση κάθε φωτογραφίας με συγκεκριμένα, περιγραφικά χαρακτηριστικά

- Πρόβλεψη τοποθεσίας για νέες φωτογραφίες μέσω ταξινόμησης k -πλησιέστερων γειτόνων

- Οι ερευνητές θρήκων ότι το βέλτιστο k ήταν 120



16

k-πλησιέστεροι γείτονες: Συμπεράσματα

- Σχηματίζουν περίπλοκα όρια απόφασης, τα οποία προσαρμόζονται στην πυκνότητα των δεδομένων εκπαίδευσης
- Σε περιπτώσεις που τα δεδομένα εκπαίδευσης είναι πολλά, η μέθοδος των *k*-πλησιέστερων γειτόνων λειτουργεί ικανοποιητικά
- **Ζητήματα**
 1. Ευαισθησία στο θόρυβο
 2. Ευαισθησία στο εύρος τιμών των χαρακτηριστικών των δεδομένων
 3. Η έννοια της απόστασης μεταξύ στιγμιοτύπων δεδομένων χάνει τη σημασία της όσο οι διαστάσεις μεγαλώνουν
 4. Γραμμική υπολογιστική πολυπλοκότητα συναρμήσει του πλήθους των δεδομένων εκπαίδευσης (ψευδο-πολυωνυμικός αλγόριθμος)

17

Αφελής Μπεϋζιανός Ταξινομητής

Naïve Bayesian Classifier

18

Αφελείς Μπεϋζιανοί Ταξινομητές

• Naïve Bayesian Classifiers (NBC)

- Οικογένεια πιθανοτικών ταξινομητών οι οποίοι βασίζονται στο **Θεώρημα του Bayes**
- Υποθέτουν **ισχυρή ανεξαρτησία** μεταξύ των χαρακτηριστικών των δεδομένων
- Εξου και ο χαρακτηριστικός **αφελείς** (naïve)
- **Thomas Bayes** (1701-1761)
 - Βρετανός στατιστικός, φιλόσοφος και τερωμένος της Πρεσβυτεριανής Εκκλησίας
- **Θεώρημα Bayes ή Κανόνας Bayes**
 - Έστω A_1, A_2, \dots, A_n **αριθμίσιας αποκλειόμενα** (ανεξάρτητα) ενδεχόμενα που καθορίζουν διεγματικό χώρο S . Έστω B ένα οποιοδήποτε ενδεχόμενο του χώρου, τέτοιο ώστε $P(B) > 0$. Τότε



$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_{j=1}^n P(B|A_j)P(A_j)}$$

19

Θεώρημα Bayes: Απλή μορφή

- $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$, όπου A, B ενδεχόμενα με $P(B) > 0$
- **Το Θεώρημα του Bayes**
 - επιπρέπει την **ενημέρωση** της πιθανότητας εμφάνισης ενός ενδεχομένου, **ενσωματώνοντας νέα ιλληροφορία**
 - Ενσωματώνει τις εκ των προτέρων πιθανότητες (prior probabilities) για να δημιουργήσει εκ των νοιτέρων πιθανότητες (posterior probabilities)
- $P(A|B)$: εκ των νοιτέρων πιθανότητα εμφάνισης ενδεχομένου A , δεδομένου ότι το ενδεχόμενο B έχει συμβεί
- $P(A)$: εκ των προτέρων πιθανότητα εμφάνισης ενδεχομένου A , πριν την πραγματοποίηση νέας παρατήρησης
- $P(B|A)$: πιθανοφάνεια (likelihood) ενδεχομένου A
 - πιθανότητα να συμβεί το B ενώ το A έχει ήδη συμβεί
- $P(B)$: πιθανότητα εμφάνισης ενδεχομένου B
 - Καλείται και «απόδειξη» (evidence)

20

Θεώρημα Bayes: Παράδειγμα

Έστω ότι έχουμε κατάστημα ηλεκτρολογικού υλικού και προμηθεύμαστε λαμπτήρες από τρεις κατασκευαστές: τον **A**, τον **B** και τον **C**. Πιο συγκεκριμένα ο **A** μας προμηθεύει το 80% των λαμπτήρων που πουλάμε, ο **B** το 15% και ο **C** το υπόλοιπο 5%. Επίσης, οι κατασκευαστές μας έχουν ενημέρωσει ο μεν **A** ότι το 4% των λαμπτήρων του είναι **ελλαττωματικό**, ο **B** το 6% και ο **C** το 9%. Δεδομένου ότι ένας πελάτης μας επιστρέφει **έναν λαμπτήρα πίσω** ως **ελλαττωματικό**, **πουα** είναι η **πιθανότητα** να έχει κατασκευαστεί από τον **A**:

Δύση

$$P(A) = 0,8, P(B) = 0,15, P(C) = 0,05$$

$$P(E|A) = 0,04, P(E|B) = 0,06, P(E|C) = 0,09$$

Εφαρμογή Θεωρήματος Bayes

$$P(A|E) = \frac{P(E|A)P(A)}{P(E|A)P(A) + P(E|B)P(B) + P(E|C)P(C)} = \frac{0,04 \cdot 0,8}{0,04 \cdot 0,8 + 0,06 \cdot 0,15 + 0,09 \cdot 0,05} \approx 0,7033$$

Άρα η πιθανότητα ο ελλαττωματικός λαμπτήρας να έχει κατασκευαστεί από τον **A** είναι **περίπου 70,33%**

21

Αφελής Μπεϋζιανός Ταξινομητής: Μοντέλο

• Αφελής υπόθεση (naïve assumption)

- Όλα τα x χαρακτηριστικά του νέου δείγματος δεδομένων x είναι **ανεξάρτητα** μεταξύ τους
- $x = \{x_1, x_2, \dots, x_n\}$
- $P(x_i | x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n, C_k) = P(x_i | C_k)$, C_k : Κλάση δείγματος
- $P(x | C_k) = \prod_{i=1}^n P(x_i | C_k)$

• Εφαρμογή Θεωρήματος Bayes

- $P(C_k | x) = \frac{P(C_k)P(x | C_k)}{P(x)}$ $\propto P(C_k)P(x | C_k) \Rightarrow P(C_k | x) \propto P(C_k) \prod_{i=1}^n P(x_i | C_k)$
- $P(x)$: Πιθανότητα εμφάνισης συγκεκριμένου δείγματος δεδομένων, σταθερά «Απόδειξη» (evidence)

• Ταξινόμηση νέου δείγματος δεδομένων x

- $\hat{y} = \operatorname{argmax}_{k \in 1, \dots, K} P(C_k) \prod_{i=1}^n P(x_i | C_k)$

22

Αφελής Μπεϋζιανός Ταξινομητής: Εκπαίδευση Μοντέλου

- Ταξινομητής: $\hat{y} = \operatorname{argmax}_{k \in 1, \dots, K} P(C_k) \prod_{i=1}^n P(x_i | C_k)$

• Δεδομένα εκπαίδευσης $\{X, y\}$

- m στιγμάτων (instances) n χαρακτηριστικών το κάθε ένα, μαζί με την αντίστοιχη ετικέτα τους

• Παράμετροι μοντέλου

1. $P(C_k)$: Υπολογισμός κατανομής κλάσεων στο σύνολο δεδομένων εκπαίδευσης
2. $P(x_i | C_k)$: Υπολογισμός πιθανότητας εμφάνισης κάθε χαρακτηριστικού σε κάθε κλάση, στο σύνολο δεδομένων εκπαίδευσης

- Με την προσθήκη νέων δεδομένων εκπαίδευσης, οι τιμές των παραμέτρων του μοντέλου **ενημερώνονται**

23

Αφελής Μπεϋζιανός Ταξινομητής: Παράδειγμα

Παρατηρήσεις βροχόπτωσης

Θερμοκρασία	Υγρασία	Βροχή
Κρύο	Υψηλή	Ναι
Κρύο	Χαμηλή	Όχι
Μέση	Χαμηλή	Ναι
Μέση	Μέτρια	Όχι
Ζέστη	Μέτρια	Όχι
Ζέστη	Υψηλή	Όχι

Αν οήμερα η **Θερμοκρασία** είναι **Μέση** και η **Υγρασία** **Υψηλή**, θα βρέξει ή όχι;

Δύση

- $P(B) = \frac{2}{6}, P(OB) = \frac{4}{6}$
- $P(M|B) = \frac{1}{2}, P(M|OB) = \frac{1}{4}$
- $P(Y|B) = \frac{1}{2}, P(Y|OB) = \frac{1}{4}$
- $P(B|M, Y) \propto P(B) P(M|B)P(Y|B) = \frac{2}{6} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{12}$
- $P(OB|M, Y) \propto P(OB)$
 $P(M|OB)P(Y|OB) = \frac{4}{6} \cdot \frac{1}{4} \cdot \frac{1}{4} = \frac{1}{24}$
- Επειδή $P(B|M, Y) > P(OB|M, Y)$ ο αφελής μπεϋζιανός ταξινομητής προβλέπει ότι **Θα βρέξει** οήμερα

24

Συμπεράσματα

• Περιορισμοί

- Ελλιπή δεδομένα εκπαίδευσης
 - Αν η τιμή κάπου ου χαρακτηριστικού δεν υπάρχει καθόλου στα δεδομένα εκπαίδευσης, τότε δεν μπορεί να προσδιοριστεί σε που κλάση ανήκει το δείγμα.
- Συνεχή χαρακτηριστικά
 - Πρέπει να θρεψεί *υποκείμενη κατανομή* τους (πχ κανονική, Laplace, Poisson)
- Μη-ανεξαρτησία μεταξύ των χαρακτηριστικών των δεδομένων
 - Εφαρμογή τεχνικών απεικόνισης των χαρακτηριστικών σε νέο χώρο, όπου η ανεξαρτησία τους είναι διασφαλισμένη ως ένα βαθμό (π.χ. ανάλυση κυρίων συνιστώσων)

• Εφαρμογές

- Φίλτρα ενοχλητικής αλληλογραφίας
- Εκτιμήσεις ρίσκου για χορήγηση δανείων/έκδοση πιστωτικών καρτών
- ...

25

Βιβλιογραφία

1. Ταξινομητές Πλησιέστερων Γειτόνων (Εκμάθηση μέσω Παραδειγμάτων)
 - M. Kubat – Εισαγωγή στη Μηχανική Μάθηση
 - Κεφάλαιο 3
2. Αφελείς Μπεῦζιανοί Ταξινομητές
 - M. Kubat – Εισαγωγή στη Μηχανική Μάθηση
 - Κεφάλαιο 2 – Ενότητες 2.1-2.4

26

Μηχανική Μάθηση

1^ο εξάμηνο | ακαδημαϊκό έτος 2023-2024

Θάνος Βουλόδημος
Επ. Καθηγητής ΣΗΜΜΥ ΕΜΠ

Νευρωνικά Δίκτυα – Γραμμικά Μοντέλα – Perceptron – Multi Layer Perceptron



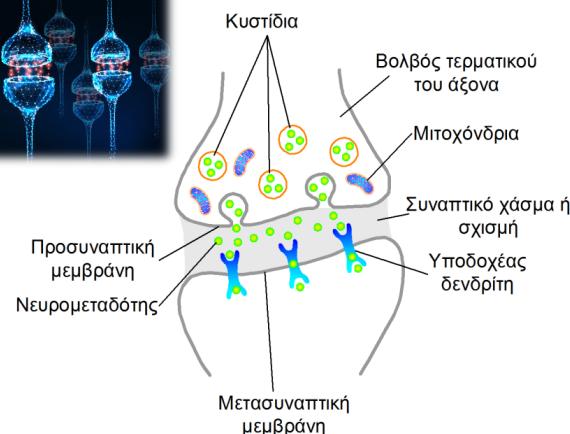
Συνάψεις

Είδη συνάψεων

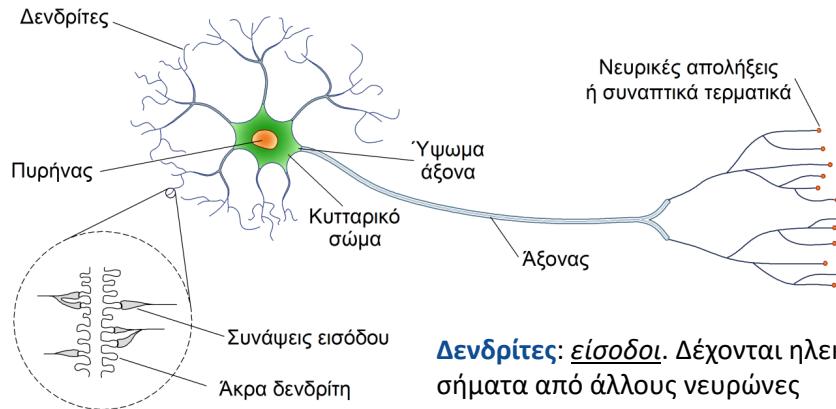


ενισχυτικές
(excitatory)

ανασταλτικές
(inhibitory)



Φούσκες με ιόντα (Na^+ , K^+). Το πλάτος της σύναψης, η απόστασή της από τον δενδρίτη και η πυκνότητα του ηλεκτροχημικού υλικού επηρεάζουν την ευκολία με την οποία η ηλεκτρική δραστηριότητα μεταδίδεται από τον άξονα στο δενδρίτη. Το ποσοστό της ηλεκτρικής δραστηριότητας που μεταδίδεται τελικά στο δενδρίτη λέγεται **συναπτικό βάρος**.



Δενδρίτες: είσοδοι. Δέχονται ηλεκτρικά σήματα από άλλους νευρώνες

Άξονας: έξοδος. Μήκος από μερικά χλιοστά έως >1m. Στέλνει ηλεκτρικούς παλμούς σταθερού πλάτους αλλά μεταβλητής συχνότητας.

Συνάψεις: σημεία ένωσης μεταξύ διακλαδώσεων του άξονα ενός νευρώνα και των δενδριτών από άλλους νευρώνες.

Μηχανική Μάθηση | ΣΗΜΜΥ ΕΜΠ | 7^ο εξάμηνο 2023-2024

2

Λειτουργία βιολογικού νευρώνα

- Συχνότητα παλμών στον άξονα (έξοδο) = ανάλογη της συνολικής διέγερσης
- Συνολική διέγερση = άθροισμα των διεγέρσεων σε όλους τους δενδρίτες

Όμως:

Συχνότητα παλμών $< 1 / (t_p + t_r)$

- Πλαστικότητα:** οι νευρώνες έχουν ρυθμιζόμενες (μεταβαλλόμενες) συνάψεις
- Πολύ μεγάλο πλήθος** νευρώνων: 100 δισεκατομμύρια κατά μέσο όρο στον ανθρώπινο εγκέφαλο
 - παραλληλισμός** της επεξεργασίας
 - κατανομή** της πληροφορίας



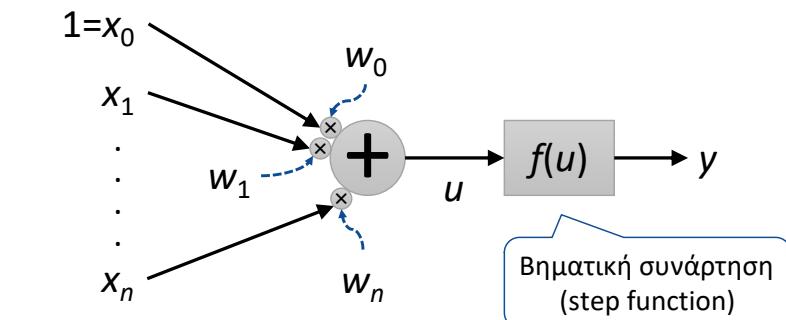
Threshold Logic Unit (Warren McCulloch, Walter Pitts, 1943)

- Μοντελοποίηση νευρώνα όπως περίπου τα τρανζίστορ (on/off):
 - $y = 0 \rightarrow$ ο νευρώνας είναι αδρανής
 - $y = 1 \rightarrow$ μέγιστη συχνότητα παλμών
- Είσοδοι: x_1, x_2, \dots, x_n
- Συναπτικά βάρη: w_1, w_2, \dots, w_n
- Πόλωση: w_0
- Συνολική διέγερση: $u = w_1x_1 + \dots + w_nx_n + w_0$

Δύο δυνατές καταστάσεις εξόδου

Βηματική συνάρτηση (step function)

$$y = f(u) = \begin{cases} 0 & \text{αν } u < 0 \\ 1 & \text{αν } u > 0 \end{cases}$$



Το μοντέλο Perceptron είναι ουσιαστικά ένας νευρώνας McCulloch-Pitts

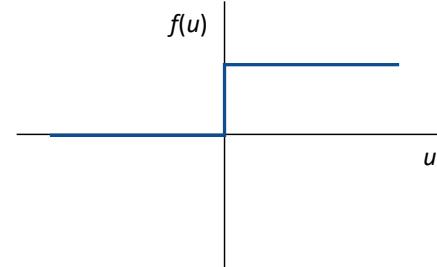
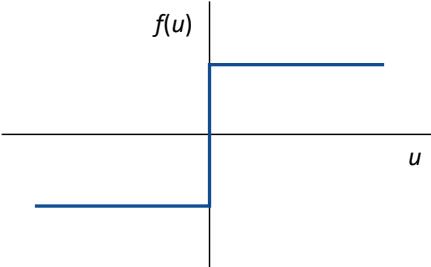


Συναρτήσεις ενεργοποίησης

Βηματική (διπολική – sgn) -1/1 Βηματική (κλασική) 0/1

$$f(u) = \begin{cases} -1, & \text{αν } u < 0 \\ 1, & \text{αν } u > 0 \end{cases}$$

$$f(u) = \begin{cases} 0, & \text{αν } u < 0 \\ 1, & \text{αν } u > 0 \end{cases}$$

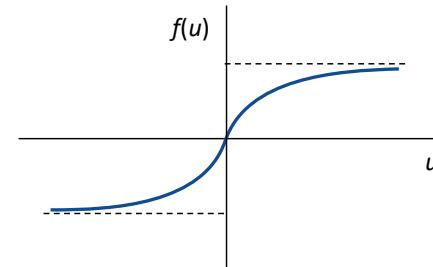


Συναρτήσεις ενεργοποίησης (2)

Υπερβολική εφαπτομένη

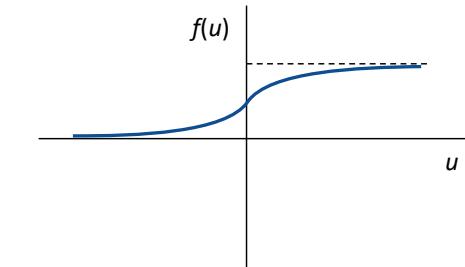
$$f(u) = \tanh(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$$

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



Σιγμοειδής

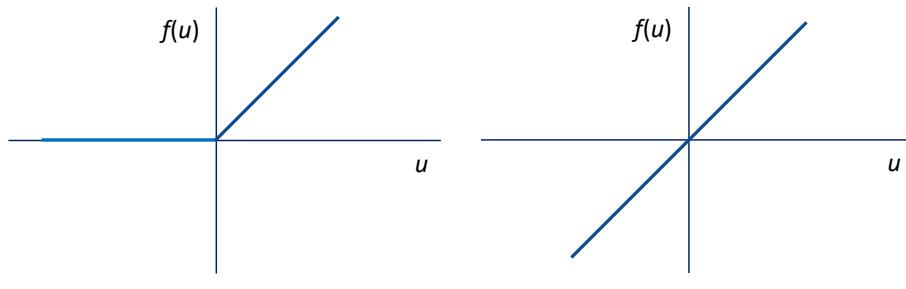
$$f(u) = \frac{1}{1 + e^{-u}}$$





Ράμπα (ReLU = Rectifier Linear Unit)

$$f(u) = \begin{cases} 0, & \text{αν } u < 0 \\ u, & \text{αν } u > 0 \end{cases}$$



Γραμμική

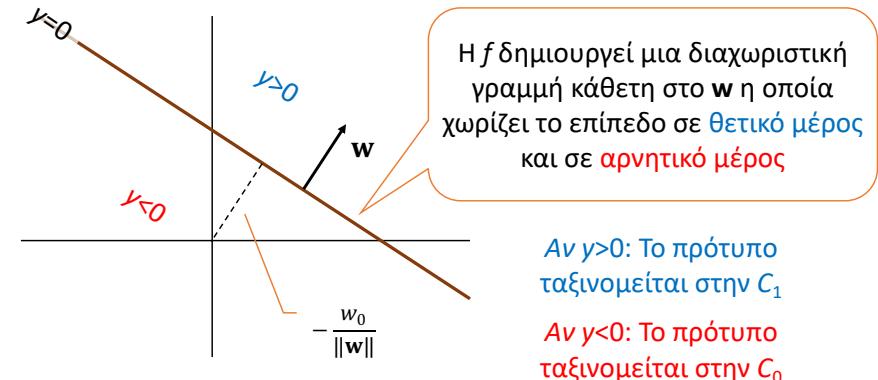
$$f(u) = u$$



Ειδική περίπτωση όπου η $y=f(\mathbf{x};\mathbf{w})$ είναι της μορφής :

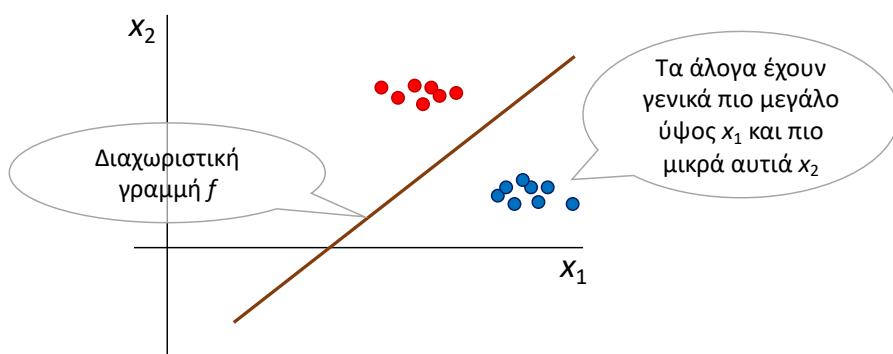
$$y=\mathbf{w}^T \mathbf{x} + w_0 \quad (=w_1 x_1 + w_2 x_2 + w_0)$$

\mathbf{w} : διάνυσμα βαρών (weight vector), w_0 : πόλωση (bias)



Χρήση Συνάρτησης Διαχωρισμού

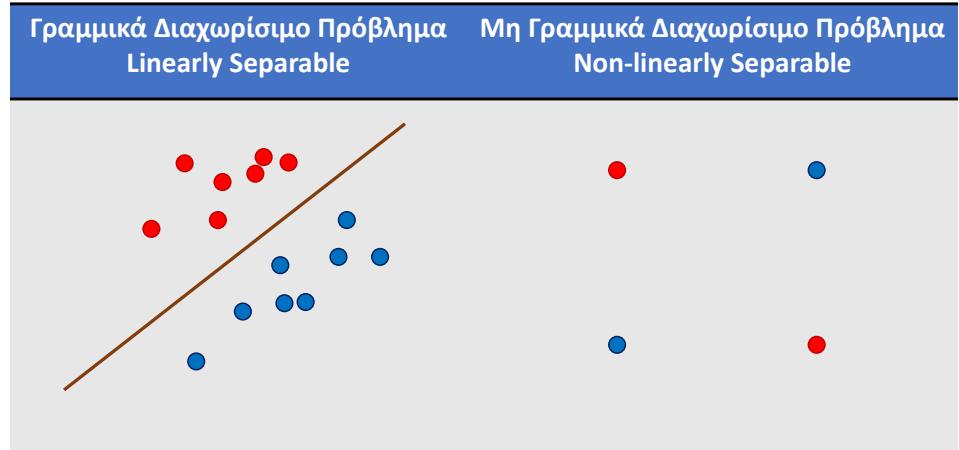
C_0 : Γαϊδούρια ● C_1 : Άλογα ●
 x_1 = ύψος ζώου, x_2 = μήκος αυτιών



Γραμμική Διαχωρισμότητα

- Linear Separability: Υπάρχει γραμμική επιφάνεια (πχ. ευθεία 2-d, επίπεδο 3-d) που χωρίζει τις δύο κλάσεις;

Γραμμικά Διαχωρίσιμο Πρόβλημα Linearly Separable



Μη Γραμμικά Διαχωρίσιμο Πρόβλημα Non-linearly Separable



- Πολλαπλές Κλάσεις:** Δίνεται ένα πρότυπο εισόδου $\mathbf{x} \in \mathbb{R}^n$
 - Ζητείται να ταξινομηθεί σε μια από L κλάσεις C_0, \dots, C_{L-1}
 - Επικέτες: $\mathbf{t} = [1, 0, \dots, 0]$ για C_0 , $\mathbf{t} = [0, 1, \dots, 0]$ για $C_1, \dots, \mathbf{t} = [0, 0, \dots, 1]$ για C_{L-1}

- Παράδειγμα.** Αναγνώριση χειρόγραφων ψηφίων
 - $\mathbf{x} = \begin{bmatrix} 2 \\ 8 \end{bmatrix}$ $\mathbf{t} = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0]$ (κλάση C_2)
 - Δίνονται εικόνες υπό μορφή διανυσμάτων από pixels
 - $\mathbf{x} = \begin{bmatrix} 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0 \end{bmatrix}$ (κλάση C_8)



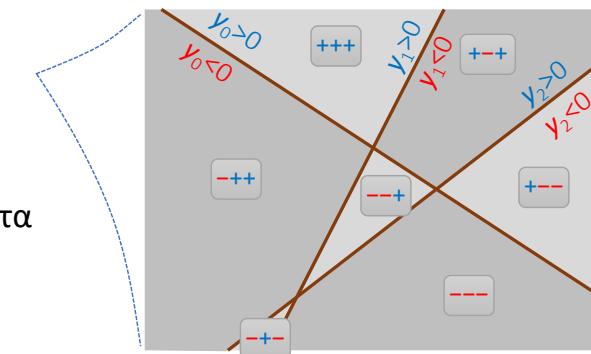
- Προσέγγιση **Ένας ενάντια σε όλους (One vs All ή One vs Rest):** Για κάθε κλάση C_i δημιουργούμε και μια διαφορετική συνάρτηση διαχωρισμού $y_i = \mathbf{w}_i^T \mathbf{x} + w_{i,0}$, ώστε να διακρίνει τα πρότυπα της κλάσης αυτής από όλες τις άλλες κλάσεις.

Παράδειγμα.

3 κλάσεις: C_0, C_1, C_2

Πρόβλημα:

Δημιουργούνται τμήματα χωρίς σαφή «νικητή»



Γραμμική Ταξινόμηση Δύο Κλάσεων

Ελάχιστα Τετράγωνα, Perceptron



Λύση Ελαχίστων Τετραγώνων

Ας υποθέσουμε ότι οι στόχοι είναι $t_i = -1, 1$.

Μπορούμε να δούμε το πρόβλημα σαν ένα σύστημα εξισώσεων :

$$\mathbf{w}^T \mathbf{x}_1 + w_0 = t_1$$

...

$$\mathbf{w}^T \mathbf{x}_P + w_0 = t_P$$

ή πιο απλά

$$\tilde{\mathbf{w}}^T \tilde{\mathbf{x}}_1 = t_1$$

...

$$\tilde{\mathbf{w}}^T \tilde{\mathbf{x}}_P = t_P$$

όπου $\tilde{\mathbf{w}}^T = [\mathbf{w}^T, w_0]$ και $\tilde{\mathbf{x}}_i = \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix}$



Υπό μορφή Πίνακα:

$$\tilde{\mathbf{X}}\tilde{\mathbf{w}} = \mathbf{t}$$

$$\text{όπου } \tilde{\mathbf{X}} = \begin{bmatrix} \tilde{\mathbf{x}}_1^T \\ \vdots \\ \tilde{\mathbf{x}}_P^T \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} t_1 \\ \vdots \\ t_P \end{bmatrix}.$$

Ωστόσο, υπάρχει το εξής πρόβλημα: το πλήθος των **εξισώσεων** P (πλήθος γραμμών του πίνακα $\tilde{\mathbf{X}}$) είναι συνήθως **μεγαλύτερο** από τη διάσταση του $\tilde{\mathbf{w}}$ (πλήθος στηλών του πίνακα $\tilde{\mathbf{X}}$).

Αν η διάσταση του \mathbf{w} είναι n (ισούται με τη διάσταση των προτύπων), τότε η διάσταση του $\tilde{\mathbf{w}}$ είναι $(n+1)$.

Συνήθως $P > n + 1$.



Στην περίπτωση αυτή, το σύστημα $\tilde{\mathbf{x}}_i^T \tilde{\mathbf{w}} = t_i$ έχει περισσότερες εξισώσεις από αγνώστους (**υπερορισμένο**) και άρα δεν έχει λύση.

Προφανώς, όποιο διάνυσμα $\tilde{\mathbf{w}}$ και αν επιλέξουμε, θα έχουμε ένα σφάλμα ε_i για κάθε πρότυπο $\tilde{\mathbf{x}}_i$:

$$\tilde{\mathbf{x}}_i^T \tilde{\mathbf{w}} = t_i + \varepsilon_i$$

$$\varepsilon_i = \tilde{\mathbf{x}}_i^T \tilde{\mathbf{w}} - t_i$$

- **Λύση ελαχίστων τετραγώνων:** Θα προσπαθήσουμε να βρούμε εκείνο το $\tilde{\mathbf{w}}$ που θα **ελαχιστοποιεί** το άθροισμα των τετραγώνων των ε_i , δηλαδή θα προσπαθήσουμε να ελαχιστοποιήσουμε το κόστος:

$$J_{LS} = \sum_{i=1}^P \varepsilon_i^2 = \sum_{i=1}^P (\tilde{\mathbf{x}}_i^T \tilde{\mathbf{w}} - t_i)^2 = \|\tilde{\mathbf{X}}\tilde{\mathbf{w}} - \mathbf{t}\|^2$$



- Η λύση είναι γνωστή από τα μαθηματικά:

$$\tilde{\mathbf{w}} = (\tilde{\mathbf{X}}^T \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^T \mathbf{t} \quad (\mathbf{Xw} = \mathbf{t}, \mathbf{X}^{-1} \mathbf{Xw} = \mathbf{Iw} = \mathbf{X}^{-1} \mathbf{t}, \mathbf{w} = \mathbf{X}^{-1} \mathbf{t})$$

- Ο πίνακας $(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^T$ λέγεται και ψευδο-αντίστροφος (pseudo-inverse) του $\tilde{\mathbf{X}}$ και συμβολίζεται με $\tilde{\mathbf{X}}^+$.

Ο $\tilde{\mathbf{X}}^+$ λέγεται ψευδο-αντίστροφος διότι :

$$\tilde{\mathbf{X}}^+ \tilde{\mathbf{X}} = (\tilde{\mathbf{X}}^T \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^T \tilde{\mathbf{X}} = \mathbf{I}$$

ακόμη και αν ο $\tilde{\mathbf{X}}$ δεν είναι τετραγωνικός πίνακας.

Βέβαια

$$\tilde{\mathbf{X}} \tilde{\mathbf{X}}^+ \neq \mathbf{I}$$



- Αν το πρόβλημα **είναι γραμμικά** διαχωρίσιμο:
 - Η μέθοδος των ελαχίστων τετραγώνων δίνει συχνά τη διαχωριστική γραμμή.
 - Κάποιες φορές όμως μπορεί να μην τη βρει.
- Αν το πρόβλημα **δεν είναι γραμμικά** διαχωρίσιμο:
 - Η μέθοδος θα δώσει κάποια ευθεία με καλά αποτελέσματα (συνήθως).



To Μοντέλο Perceptron

- Το μοντέλο Perceptron είναι παρόμοιο με το γραμμικό μοντέλο που χρησιμοποιήθηκε στα ελάχιστα τετράγωνα:

$$y_i = \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}_i$$

Γραμμικό μοντέλο ελαχίστων τετραγώνων

$$y_i = f(\tilde{\mathbf{w}}^T \tilde{\mathbf{x}}_i)$$

Μοντέλο Perceptron

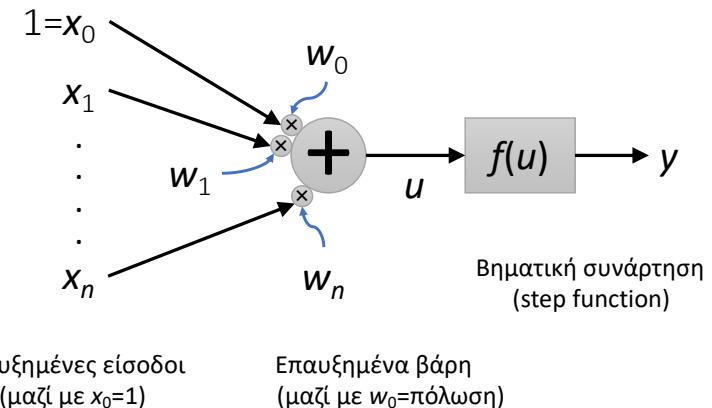
- Η διαφορά είναι στη συνάρτηση f που παίρνει τιμές -1 ή 1:

$$f(u) = \begin{cases} -1, & \text{αν } u < 0 \\ 1, & \text{αν } u > 0 \end{cases}$$

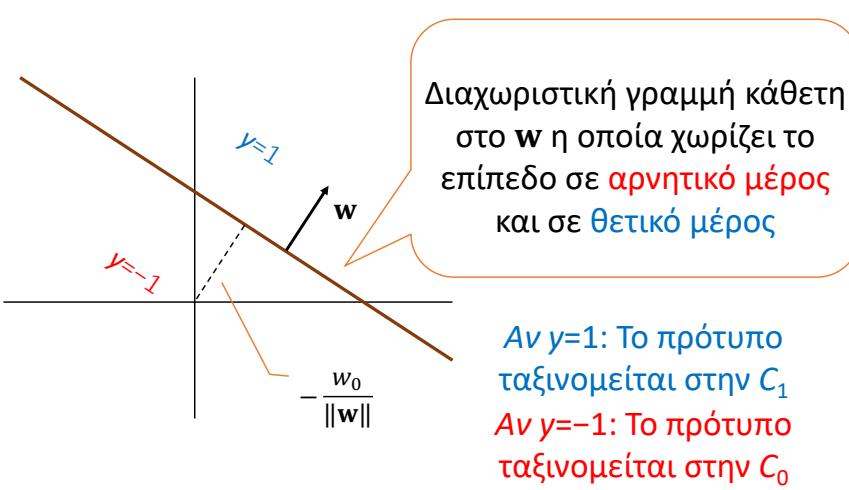
- Η έξοδος του μοντέλου Perceptron είναι $-1/1$, ενώ στο γραμμικό μοντέλο των ελαχίστων τετραγώνων η έξοδος ήταν πραγματικός αριθμός από $-\infty$ έως $+\infty$.



Σχεδιάγραμμα Perceptron



Δυνατότητες Perceptron



Επιλογή Βηματικής Συνάρτησης

Βηματική -1/1

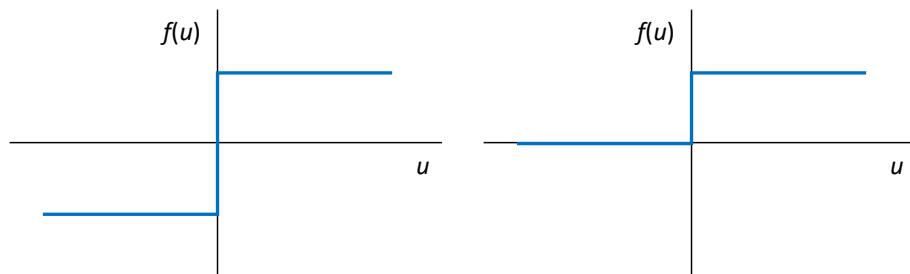
Αν οι στόχοι είναι $t=-1/1$

$$f(u) = \begin{cases} -1, & \text{αν } u < 0 \\ 1, & \text{αν } u > 0 \end{cases}$$

Βηματική 0/1

Αν οι στόχοι είναι $t=0/1$

$$f(u) = \begin{cases} 0, & \text{αν } u < 0 \\ 1, & \text{αν } u > 0 \end{cases}$$





Επαυξημένα Πρότυπα Εισόδου	\tilde{x}_1	\tilde{x}_2	...	\tilde{x}_P
Στόχοι	t_1	t_2	...	t_P
Έξοδοι	y_1	y_2	...	y_P

Εποχή = μια κυκλική επανάληψη όλων των προτύπων.

Αρχικά τυχαίο \tilde{w}_0 , $k=0$

Για κάθε εποχή = 1: Maxepochs

Για κάθε πρότυπο $p = 1:P$

$$k = k+1$$

$$y_p = f(\tilde{w}_k^T \tilde{x}_p)$$

$$\tilde{w}_{k+1} = \tilde{w}_k + \beta(t_p - y_p)\tilde{x}_p$$

- Διόρθωση μόνο αν στόχος $t_p \neq$ έξοδος y_p



- Ο αλγόριθμος τερματίζεται όταν δεν γίνεται πλέον καμία διόρθωση σε κανένα πρότυπο. Αυτό σημαίνει ότι **όλοι** οι στόχοι είναι **ίσοι** με **όλες** τις εξόδους:

$$\begin{array}{ll} t_1 = y_1 \\ \text{ΚΑΙ} & t_2 = y_2 \\ \text{ΚΑΙ} & \dots \\ \text{ΚΑΙ} & t_P = y_P \end{array}$$

- **Πρόβλημα:** Αν τα δεδομένα δεν είναι γραμμικά διαχωρίσιμα ο κανόνας Perceptron **δεν συγκλίνει** ποτέ. Αναγκαστικός ο πρόωρος τερματισμός.
- **Θεώρημα:** Αν τα δεδομένα είναι γραμμικά διαχωρίσιμα τότε ο κανόνας Perceptron **συγκλίνει** σε πεπερασμένο (αλλά άγνωστο) αριθμό επαναλήψεων.



Το Βήμα Εκπαίδευσης β

- Η παράμετρος β λέγεται **βήμα εκπαίδευσης** και είναι (συνήθως μικρή) θετική σταθερά.
- Η διόρθωση των βαρών είναι ανάλογη του β .
- Μεγάλο $\beta \rightarrow$ κίνδυνος ταλάντωσης.
- Μικρό $\beta \rightarrow$ αργή σύγκλιση.



Κανόνας ADALINE

Adaptive linear element

- Δεν χρησιμοποιείται η μη-γραμμική συνάρτηση ενεργοποίησης
- Η έξοδος παίρνει συνεχείς τιμές
- Οι στόχοι μπορούν να παίρνουν συνεχείς τιμές

$$\mathbf{X} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(P)}]^T \quad \mathbf{d} = [d^{(1)}, d^{(2)}, \dots, d^{(P)}]^T$$

$$\boxed{\mathbf{X}\mathbf{w} = \mathbf{d}}$$

Επίλυση συστήματος P εξισώσεων με $n+1$ αγνώστους



Adaptive linear element

- Αν $P > n+1$ (συνήθης περίπτωση) το σύστημα μπορεί να μην έχει λύση
- Στην περίπτωση αυτή αναζητούμε προσεγγιστική λύση, χρησιμοποιώντας κριτήριο απόστασης από όλα τα πρότυπα
- Ελάχιστο τετραγωνικό σφάλμα

$$J = \sum_{i=1}^P (d^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)})^2$$

Ο αλγόριθμος τερματίζει αν το σφάλμα γίνει μικρότερο από ε



- **BHMA 1:**
 1. Αρχικοποίησε το διάνυσμα βαρών τυχαίες τιμές
 2. Δώσε μία μικρή θετική τιμή στο βήμα εκπαίδευσης
 3. Όρισε ένα όριο ε για το σφάλμα εκπαίδευσης
 4. Όρισε το μέγιστο αριθμό εποχών

- **BHMA 2:** Επανάλαβε έως τον μέγιστο αριθμό εποχών
Για κάθε ένα από τα P πρότυπα εκτέλεσε

1. Υπολόγισε την έξοδο του perceptron
2. Αν υπάρχει σφάλμα προσάρμοσε τα βάρη ως:

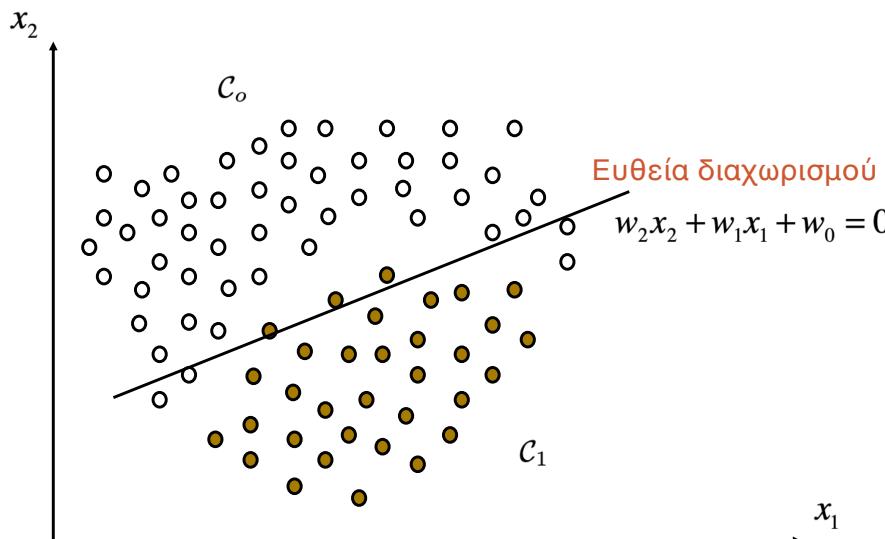
$$\mathbf{w}(k) = \mathbf{w}(k-1) + \beta(d - y)\mathbf{x}$$

Επίστρεψε τα βάρη αν ισχύει ότι: $\sum_{i=1}^P (d^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)})^2 < \varepsilon$

- **BHMA 3:** Επίστρεψε ΣΦΑΛΜΑ



Ταξινόμηση με ADALINE

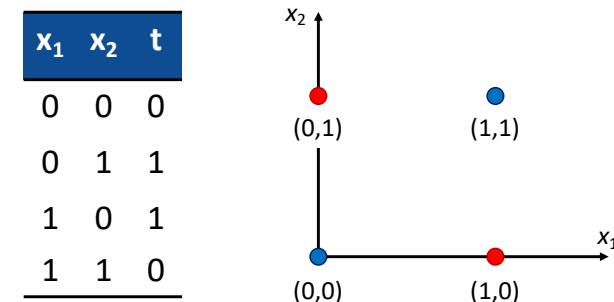


Παρατηρήσεις

- Αν το πρόβλημα είναι γραμμικά διαχωρίσιμο, το perceptron βρίσκει πάντα λύση, σε αντίθεση με το adaline
- Αν το πρόβλημα δεν είναι γραμμικά διαχωρίσιμο, το perceptron δεν συγκλίνει, ενώ το adaline μπορεί να συγκλίνει επιτρέποντας λανθασμένες ταξινομήσεις
- Το adaline συγκλίνει σε περιπτώσεις μη-γραμμικά διαχωρίσιμων προβλημάτων μόνο όταν το πρόβλημα είναι σχεδόν γραμμικά διαχωρίσιμο (κάτω από το σφάλμα ε)

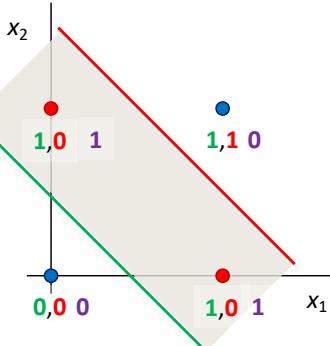
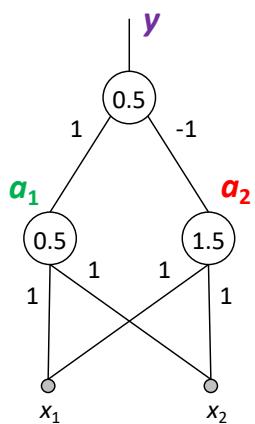


- Το μοντέλο Perceptron είναι ουσιαστικά ένας νευρώνας
- Έχει περιορισμένες δυνατότητες: Δημιουργεί μόνο γραμμικές διαχωριστικές επιφάνειες (πχ. ευθείες σε 2-D)
- Αδυναμία επίλυσης μη-γραμμικά διαχωρίσιμων προβλημάτων. Παράδειγμα, το πρόβλημα XOR:



Multi Layer Perceptron

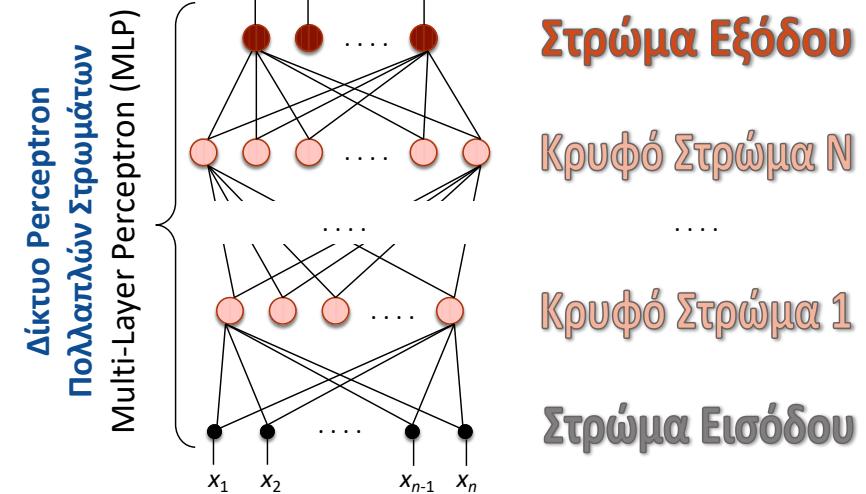
Λύση του XOR με 2 στρώματα νευρώνων

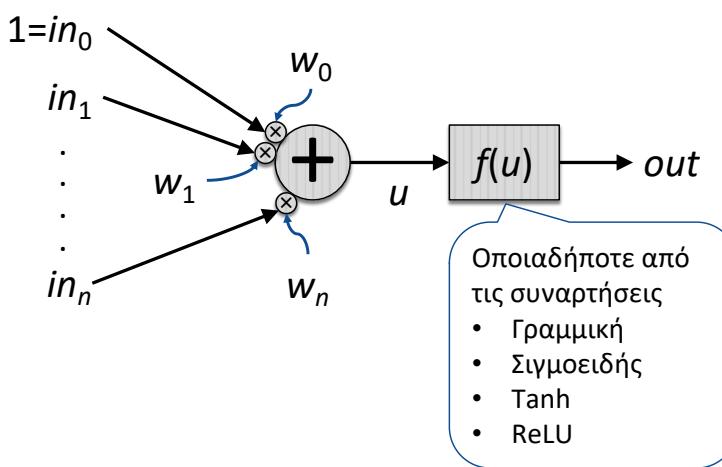


* Έστω για απλούστευση ότι η συνάρτηση ενεργοποίησης των κρυφών νευρώνων είναι η βηματική συνάρτηση 0/1

Multi-Layer Perceptron (MLP):

Πολλαπλοί νευρώνες σε στρώματα





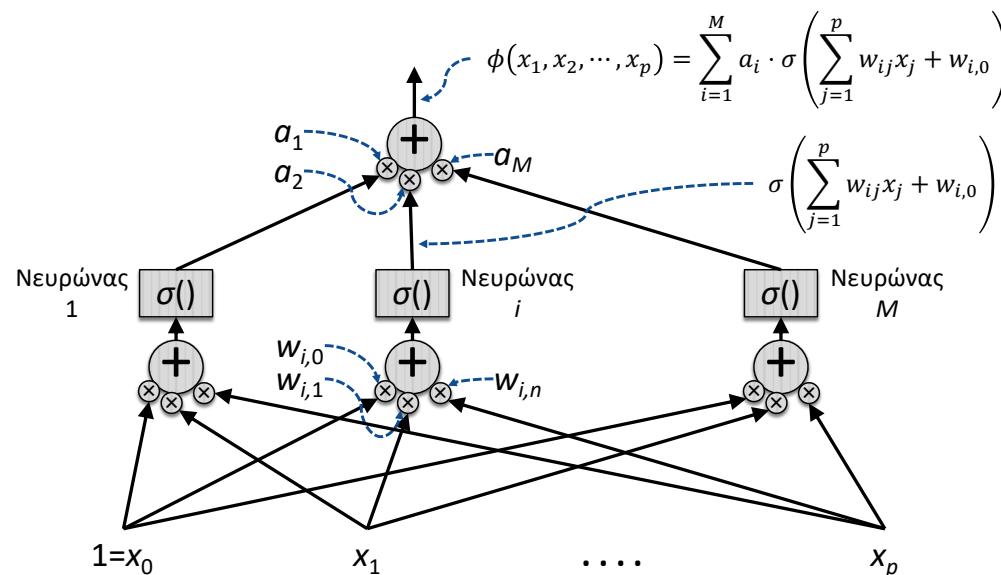
Θεώρημα: Έστω $\sigma(\cdot)$ = η σιγμοειδής συνάρτηση και $g(x_1, x_2, \dots, x_p)$ οποιαδήποτε συνεχής συνάρτηση με p μεταβλητές $0 \leq x_i \leq 1$. Τότε υπάρχει ακέραιος M και κάποιες τιμές των παραμέτρων a_i, w_{ij} , έτσι ώστε η συνάρτηση

$$\phi(x_1, x_2, \dots, x_p) = \sum_{i=1}^M a_i \cdot \sigma\left(\sum_{j=1}^p w_{ij}x_j + w_{i,0}\right)$$

προσεγγίζει την $g(x_1, x_2, \dots, x_p)$ με σφάλμα μικρότερο του ε για όλες τις τιμές $0 \leq x_1, x_2, \dots, x_p \leq 1$ και για οποιοδήποτε $\varepsilon > 0$.



Ερμηνεία Θεωρήματος



Ερμηνεία Θεωρήματος (2)

- Ιδιότητα **Καθολικού Προσεγγιστή** (*Universal Approximator*)
- Με απλά λόγια: Ένα δίκτυο δύο στρωμάτων μπορεί να προσεγγίσει όσο καλά επιθυμούμε οποιαδήποτε συνεχή συνάρτηση, αρκεί
 - Να έχουμε αρκετούς κρυφούς νευρώνες M
 - Οι νευρώνες του κρυφού στρώματος να έχουν τη σιγμοειδή συνάρτηση ενεργοποίησης
 - Ο νευρώνας εξόδου να έχει τη γραμμική συνάρτηση ενεργοποίησης



- Όσο πιο πολύπλοκη είναι η συνάρτηση που επιθυμούμε να προσεγγίσουμε τόσο περισσότερους κρυφούς νευρώνες θέλουμε
- Δύο στρώματα αρκούν (θεωρητικά)
- Universal Approximator: ισχύει και για συναρτήσεις πολλών εξόδων
- Γενικού σκοπού: Κατάλληλα και για ταξινόμηση και για παλινδρόμηση.



Αλγόριθμος

Είσοδοι: $a_1(0) = x_1, \dots, a_{N(0)}(0) = x_{N(0)}$

$0 =$ στρώμα εισόδου, $L =$ στρώμα εξόδου

Έξοδοι: $y_1 = a_1(L) = \dots, y_{N(L)} = a_{N(L)}(L)$

Για κάθε στρώμα $l = 1, \dots, L$ {

Για κάθε νευρώνα $i = 1, \dots, N(l)$ {

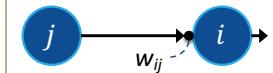
$$a_i(l) = \sigma \left(\sum_{j=1}^{N(l-1)} w_{ij}(l) a_j(l-1) + w_{i0}(l) \right)$$

}

}

Συμβολισμοί

- $N(l)$: πλήθος νευρώνων στο στρώμα l
- $a_i(l)$: έξοδος του νευρώνα i στο στρώμα l
- x_i : είσοδοι του δικτύου (στρώμα 0)
- y_i : έξοδος του δικτύου (στρώμα L)
- w_{ij} : συναπτικό βάρος από τον νευρώνα j στον i



Κατάβαση δυναμικού (gradient descent) – αλγόριθμος back-propagation



Δύο βασικά προβλήματα Βελτιστοποίησης

• Ελαχιστοποίηση

- Δίνεται: μια συνάρτηση $y = f(\mathbf{x})$ όπου $\mathbf{x} \in \mathbb{R}^n$, $y \in \mathbb{R}$
- Ζητείται: να βρεθεί το σημείο \mathbf{x}_* ώστε το $y_* = f(\mathbf{x}_*)$ να είναι το ελάχιστο δυνατό, δηλαδή, $y_* = f(\mathbf{x}_*) \leq f(\mathbf{x})$, $\forall \mathbf{x}$.

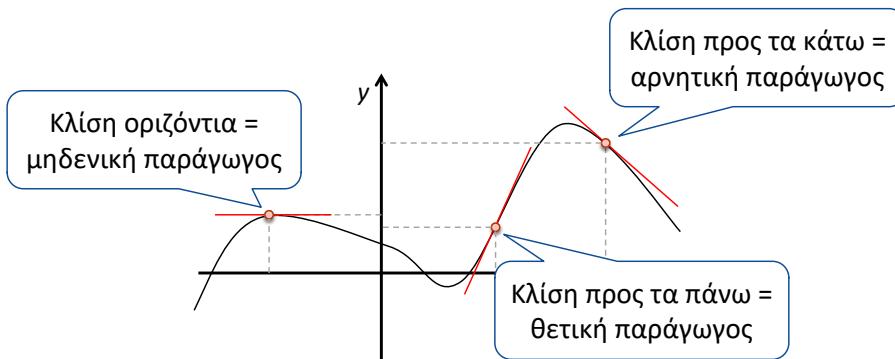
• Μεγιστοποίηση

- Δίνεται: μια συνάρτηση $y = f(\mathbf{x})$ όπου $\mathbf{x} \in \mathbb{R}^n$, $y \in \mathbb{R}$
- Ζητείται: να βρεθεί το σημείο \mathbf{x}_* ώστε το $y_* = f(\mathbf{x}_*)$ να είναι το μέγιστο δυνατό, δηλαδή, $y_* = f(\mathbf{x}_*) \geq f(\mathbf{x})$, $\forall \mathbf{x}$.

• Δεν υπάρχει πάντα λύση στα προβλήματα αυτά.



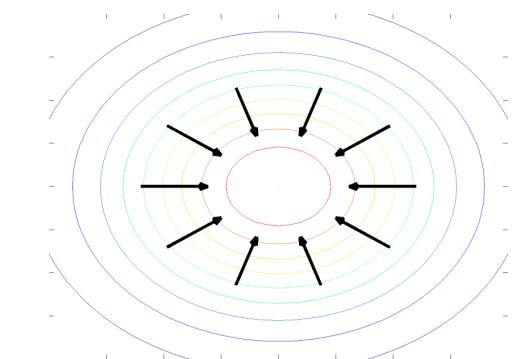
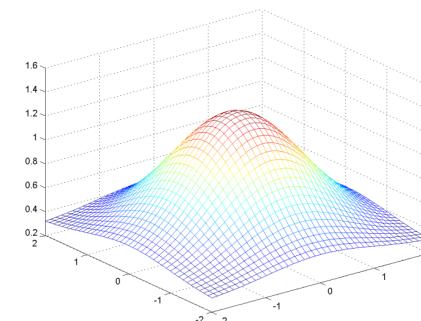
- Παράγωγος βαθμωτής συνάρτησης f με βαθμωτή μεταβλητή x , δηλ. $y = f(x)$, $x, y \in \mathbb{R}$:
- $\frac{df}{dx} = \lim_{\delta \rightarrow 0} \frac{f(x+\delta)-f(x)}{\delta}$



- Παράγωγος βαθμωτής συνάρτησης με διανυσματική μεταβλητή, δηλ. $y = f(\mathbf{x})$, $y \in \mathbb{R}, \mathbf{x} \in \mathbb{R}^n$:

- $\nabla_{\mathbf{x}} f = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \dots & \frac{\partial f}{\partial x_n} \end{bmatrix}^T$
- Συχνά γράφεται και $\frac{\partial f}{\partial \mathbf{x}}$.

Καλείται **κλίση** (gradient)



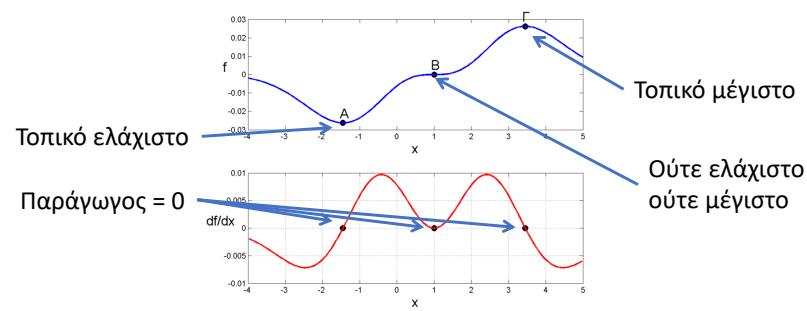
Ελαχιστοποίηση

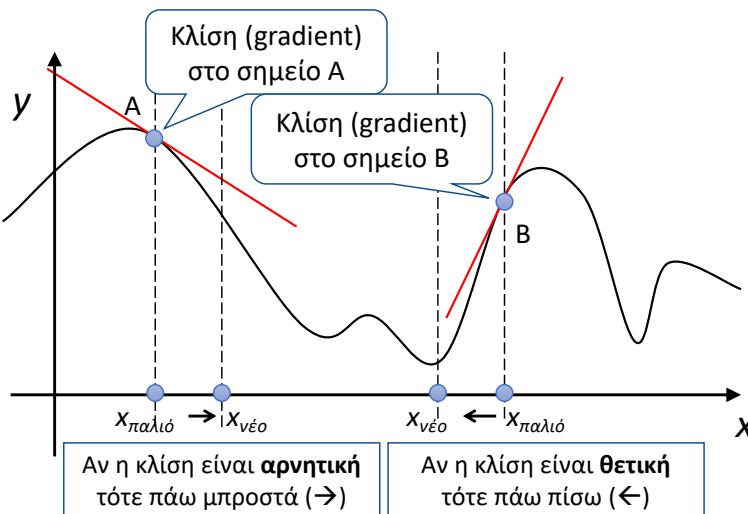
- Εντελώς παρόμοια με την μεγιστοποίηση
- Εμφανίζεται σε προβλήματα όπου ψάχνουμε το μικρότερο κόστος (πχ. στη μηχανική μάθηση, στα οικονομικά, κα.) ή την ελάχιστη ενέργεια (πχ. στη φυσική, κα.).
- Όσο πιο πολύπλοκη είναι η συνάρτηση $f(\cdot)$ τόσο πιο δύσκολο να βρεθεί η λύση με αναλυτικό τρόπο.



Συνθήκες ελαχίστου

- Συνθήκη Karush-Kuhn-Tucker (ΚΚΤ): Αν το σημείο \mathbf{x}_* δίνει το ελάχιστο $f(\mathbf{x}_*)$ τότε
- $\nabla_{\mathbf{x}} f(\mathbf{x}_*) = \mathbf{0}$
- Το αντίστροφο δεν ισχύει. Δηλαδή αν ικανοποιείται η συνθήκη δεν σημαίνει ότι το σημείο είναι οπωσδήποτε ελάχιστο.
- Παράδειγμα: Βαθμωτή συνάρτηση βαθμωτής μεταβλητής $y = f(x)$





- Κινούμαστε αντίθετα απ' ότι λέει η παράγωγος: αν η παράγωγος είναι θετική τότε μειώνω το x , αν η παράγωγος είναι αρνητική τότε αυξάνω το x .
- Κάνω μικρά βήματα χρησιμοποιώντας το $\beta = \text{βήμα εκπαίδευσης}$ (= μικρή θετική τιμή)

$$x(t+1) = x(t) - \beta \frac{df}{dx}$$

Αν το x είναι βαθμωτό

ή

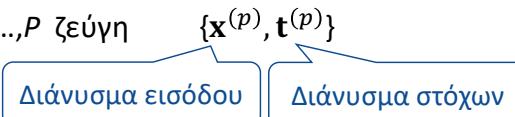
$$\mathbf{x}(t+1) = \mathbf{x}(t) - \beta \nabla_{\mathbf{x}} f$$

Αν το \mathbf{x} είναι διάνυσμα



Αλγόριθμος Back-Propagation (BP)

- Εκπαίδευση MLP με L στρώματα (Paul Werbos, 1974)
- Μάθηση με επίβλεψη
- $p = 1, \dots, P$ ζεύγη $\{\mathbf{x}^{(p)}, \mathbf{t}^{(p)}\}$



- Διάνυσμα εισόδου: $\mathbf{x}^{(p)} = [x_1^{(p)} \dots x_n^{(p)}]^T$
- Διάνυσμα στόχων: $\mathbf{t}^{(p)} = [t_1^{(p)} \dots t_m^{(p)}]^T$
- Διάνυσμα εξόδου: $\mathbf{y}^{(p)} = [y_1^{(p)} \dots y_m^{(p)}]^T$



Αλγόριθμος BP (2)

- Κριτήριο μάθησης: ελάχιστα τετράγωνα.
- Ελαχιστοποίηση του μέσου τετραγωνικού σφάλματος για όλα τα πρότυπα (P το πλήθος)

$$J_{MLP} = \frac{1}{P} \sum_{p=1}^P \|\mathbf{t}^{(p)} - \mathbf{y}^{(p)}\|^2 = \frac{1}{P} \sum_{p=1}^P \sum_{i=1}^m [t_i^{(p)} - y_i^{(p)}]^2$$

- Μέθοδος βελτιστοποίησης: Κατάβαση Δυναμικού

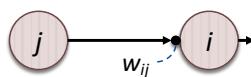
$$w_{ij}(k+1) = w_{ij}(k) - \beta \frac{\partial J_{MLP}}{\partial w_{ij}}$$



Αλγόριθμος BP (3)

- Έχει υπολογιστεί η παράγωγος:

$$\frac{\partial J_{MLP}}{\partial w_{ij}} = -\delta_i a_j$$



Αν ο νευρώνας i είναι στο τελευταίο στρώμα L

$$\delta_i(L) = (t_i - y_i)f'(u_i)$$

Αν ο νευρώνας i είναι σε οποιοδήποτε εσωτερικό στρώμα $l < L$

$$\delta_i(l) = f'(u_i) \sum_{j=1}^{N(l+1)} w_{ji} \delta_j(l+1)$$

$f'(u_i)$ = παράγωγος της συνάρτησης ενεργοποίησης f

- f = σιγμοειδής: $f'(u_i) = a_i[1 - a_i]$
- f = $\tanh()$: $f'(u_i) = 1 - a_i^2$
- f = γραμμική: $f'(u_i) = 1$



Αλγόριθμος BP (4)

- Ένας **κανόνας εκπαίδευσης** άσχετα από το στρώμα $w_{ij}(k+1) = w_{ij}(k) + \beta \delta_i a_j$
- Μόνη διαφορά ο τρόπος υπολογισμού του σφάλματος δ_i ανάλογα με το στρώμα.

A. Στο εσωτερικό στρώμα:

Το σφάλμα δ_i αποτελείται από το γινόμενο δύο όρων

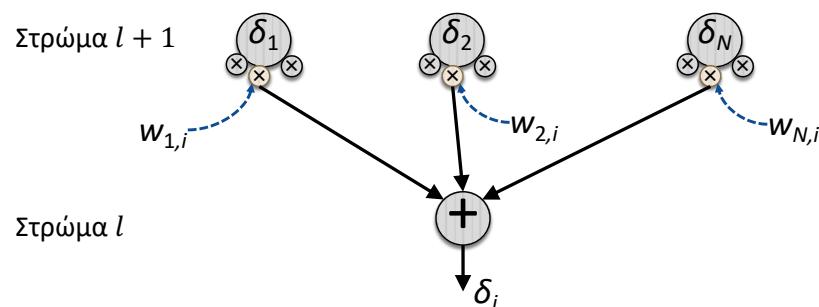
- Το σφάλμα $(t_i - y_i)$ και
- Την παράγωγο $f'(u_i)$ της συνάρτησης ενεργοποίησης f των νευρώνων του στρώματος L .



Αλγόριθμος BP (5)

B. Σε οποιοδήποτε εσωτερικό στρώμα $l < L$:

- Υπολογισμός $\delta_i(l)$ → προώθηση προς τα πίσω (backward propagation). Υπολογίζω το $\delta_i(l)$ του στρώματος l χρησιμοποιώντας τα $\delta_j(l+1)$ του πιο πάνω στρώματος.



Αλγόριθμος BP (6)

Αρχικοποίησε τα βάρη $w_{ij}(l)$ σε μικρές τυχαίες τιμές

Επανάλαβε {

Για κάθε πρότυπο $p = 1, \dots, P$ { /* Εκπαίδευση */

/* Φάση ανάκλησης = Forward phase */

Υπολόγισε τις εξόδους

/* Φάση υπολογισμού δ = Backward phase */

Υπολόγισε τα σφάλματα δ

/* Φάση ενημέρωσης βαρών = Update phase */

Ανανέωσε τα βάρη

}

} Μέχρι J_{MLP} μικρότερο από κάποιο κατώφλι MINJ ή να έχουμε φτάσει στο μέγιστο αριθμό εποχών



Stochastic gradient descent (SGD)

- Υποθέστε ότι η συνάρτηση που ελαχιστοποιούμε είναι **άθροισμα N όρων**, για παράδειγμα:
$$f(\mathbf{w}) = \sum_{n=1}^N E_n(\mathbf{w})$$
- Η παράγωγος είναι
$$\nabla_{\mathbf{w}} f = \sum_{n=1}^N \nabla_{\mathbf{w}} E_n$$
- Αν εφαρμόζουμε κλασική κατάβαση δυναμικού πρέπει να αθροίσουμε όλους τους όρους $\sum_{n=1}^N \nabla_{\mathbf{w}} E_n$, για $n = 1, \dots, N$ και μετά να ενημερώσουμε τα βάρη \mathbf{w} .
- Μια άλλη προσέγγιση: Επιλέγουμε δειγματοληπτικά (τυχαία) και δημιουργούμε υποσύνολα του συνόλου δεδομένων, που ονομάζονται **batches**. Στη συνέχεια ανανέωνουμε τα βάρη με βάση μόνο τα δεδομένα που ανήκουν στο batch.
- Η προσέγγιση αυτή συγκλίνει πιο γρήγορα και λέγεται **Στοχαστική Κατάβαση Δυναμικού (Stochastic Gradient Descent - SGD)**.



Stochastic gradient descent (SGD)

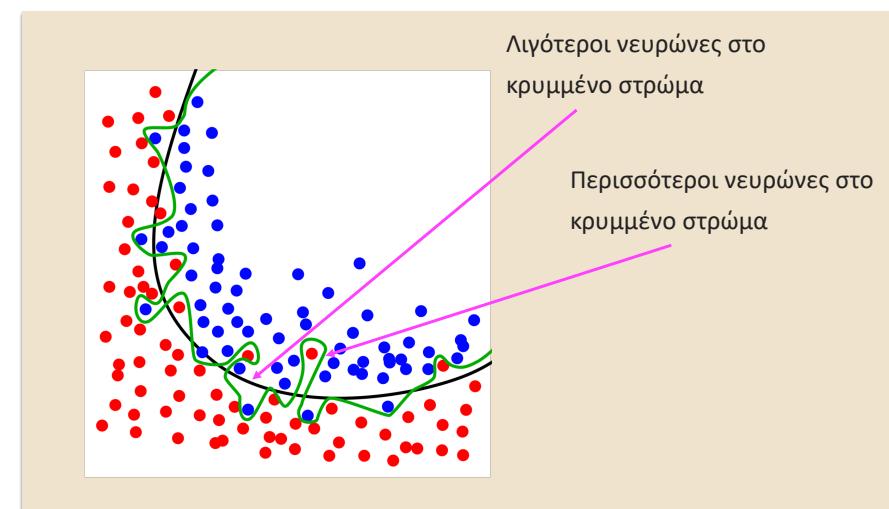
- Στην ακραία περίπτωση που το μέγεθος του batch $B=1$, έχουμε ανανέωση των βαρών μετά από την παρουσίαση ενός μόνο προτύπου.
- Η ακραία περίπτωση $B=N$ αντιστοιχεί πρακτικά στην κλασική κατάβαση δυναμικού
- Για πιο λόγο μπορεί να είναι προτιμότερη η SGD έναντι της GD;
 - Σε μεγάλα datasets, είναι υπολογιστικά ακριβό να γίνεται ανανέωση των βαρών μετά από την παρουσίαση του συνόλου των δεδομένων
 - Επιπλέον, η δειγματοληψία δεδομένων εισάγει κατά έναν τρόπο «θόρυβο», ο οποίος αποτρέπει την εύρεση «ρηγών» τοπικών ελαχίστων, γεγονός που είναι καλό για την βελτιστοποίηση μη-κυρτών συναρτήσεων

Επίδοση αλγορίθμου back-propagation

- Ο αλγόριθμος back-propagation δεν βρίσκει πάντα λύση (παρότι με βάση το θεώρημα προσέγγισης πάντα υπάρχει)
 - Δεν προσεγγίζει πάντα τη συνάρτηση εισόδου-εξόδου με σφάλμα μικρότερο του ϵ
 - Όσο πιο σύνθετη η δομή του δικτύου, τόσο πιο πολλές οι πιθανότητες να βρεθεί λύση (σε κάποιο αριθμό εποχών)
 - Δεν είναι όμως απαραίτητο ότι η λύση αυτή θα είναι πρακτικά καλή (πρόβλημα *overfitting*)
- Ο αλγόριθμος back-propagation δεν βρίσκει πάντα τη λύση που δίνει το μικρότερο τετραγωνικό σφάλμα
 - Εξαρτάται από πολλούς παράγοντες όπως η αρχικοποίηση (πρόβλημα τοπικών ελαχίστων)

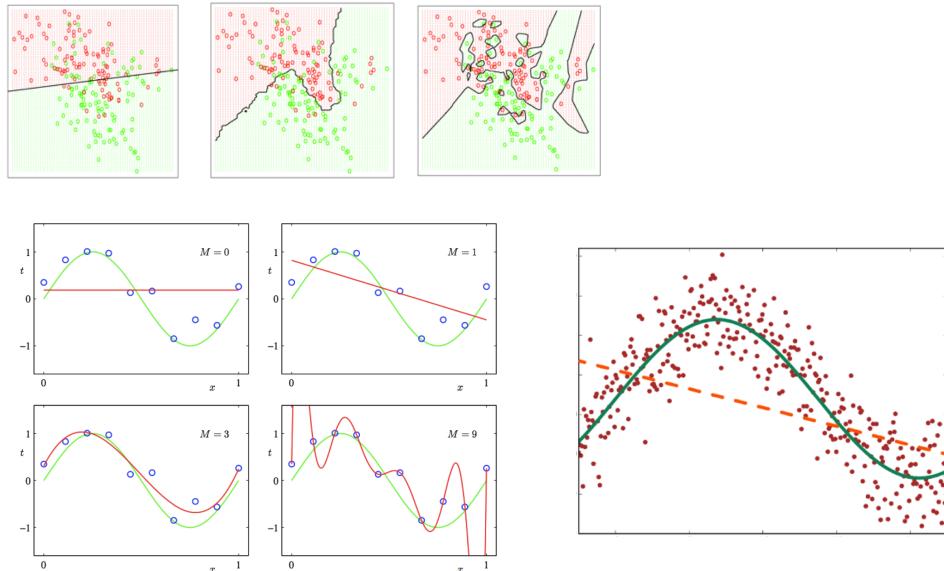


Πρόβλημα υπερπροσαρμογής (overfitting)



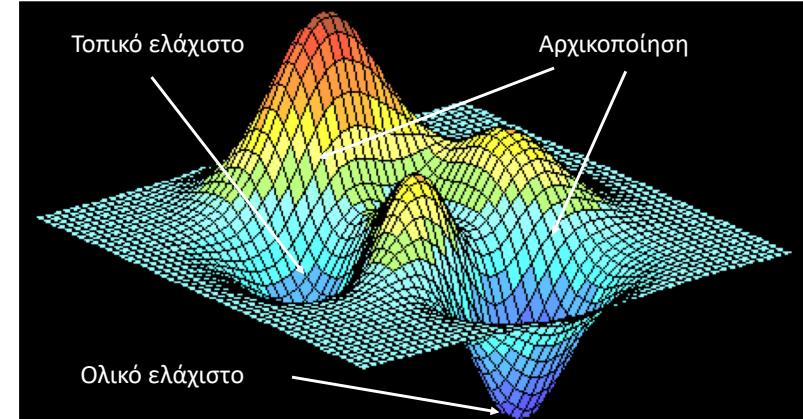


Πρόβλημα υπερπροσαρμογής (overfitting)



Πρόβλημα τοπικών ελαχίστων

Καμπύλη σφάλματος ως προς τα βάρη του δίκτυου



Επιτάχυνση σύγκλισης με χρήση Ορμής (momentum)

- Χρήση ορμής (momentum):** Κρατάμε την προηγούμενη διόρθωση $\Delta w_{ij}(k) = w_{ij}(k) - w_{ij}(k-1)$ και την προσθέτουμε στην τωρινή:

$$w_{ij}(k+1) = w_{ij}(k) + \beta \delta_i \alpha_j + \mu \cdot \Delta w_{ij}(k)$$
- Ο συντελεστής μ πρέπει να είναι μικρότερος από το 1 αλλά κοντά στο 1 (π.χ. 0.95). Όσο πιο κοντά, τόσο πιο γρήγορα τρέχει, αλλά αν είναι πολύ κοντά μπορεί να οδηγήσει σε απόκλιση.
- Αποτέλεσμα: Σε περιοχές όπου το J μειώνεται αργά, η ορμή επιταχύνει τον αλγόριθμο. Σε περιοχές όπου το w ταλαντώνεται επιβάλλει εξομάλυνση (είτε τείνει να κινείται προς μια κατεύθυνση είτε μειώνει τις ταλαντώσεις).



Πρακτικά θέματα εκπαίδευσης

- Βήμα εκπαίδευσης β:**

- Η βέλτιστη τιμή εξαρτάται από το πρόβλημα και το δίκτυο
- Μικρό, αλλά πόσο μικρό; (0.1, 0.01, 0.001, ...);
- Ιδέα-1: διαφορετικά β_ℓ για διαφορετικά στρώματα ℓ
- Ιδέα-2: διαφορετικά β_{ij} για διαφορετικά βάρη w_{ij}
- Ιδέα-3: το β μεταβάλλεται από εποχή σε εποχή. Πχ.

$$\beta_{ij}(k) = \beta_{ij}(k-1) + \eta \frac{\partial J}{\partial w_{ij}} \Big|_k \cdot \frac{\partial J}{\partial w_{ij}} \Big|_{k-1}$$



- Κριτήρια τερματισμού:
 - Το σφάλμα $J(k)$ σε κάποια εποχή k είναι κάτω από κάποιο όριο ε
 - Το σφάλμα σε δύο διαδοχικές εποχές $k, k - 1$, δεν βελτιώνεται σημαντικά: $|J(k) - J(k - 1)| < \varepsilon$
 - Τα βάρη σε μια εποχή k δέχονται αμελητέα διόρθωση: $\sum_{i,j} (\Delta w_{ij})^2 < \varepsilon$
 - Φτάνουμε στο μέγιστο πλήθος εποχών



• Αρχικοποίηση Βαρών:

- Πρέπει να αποφεύγεται οι αρχικές τιμές των βαρών να είναι ίδιες μεταξύ τους.
- Τα βάρη δεν πρέπει να είναι πολύ μεγάλα για να μην επέρχεται κορεσμός της παραγώγου. Πχ. για τη σιγμοειδή συνάρτηση $\sigma(u)$ η παράγωγος είναι

$$\sigma'(u) = \sigma(u) \cdot (1 - \sigma(u))$$



- Μεγάλο $w \rightarrow$ Μεγάλο $u = w^T x \rightarrow$ Μικρή παράγωγος \rightarrow Μικρή διόρθωση βαρών
- Καλό είναι να αρχικοποιούνται με μικρές τυχαίες τιμές
- Καλή ιδέα οι πολλαπλές εκπαιδεύσεις από διαφορετικές αρχικές τιμές, ώστε να επιβεβαιωθεί ότι η επίδοση του ΝΔ είναι ανεξάρτητη των αρχικών τιμών. Επίσης, έτσι αυξάνεται η πιθανότητα αποφυγής παγίδευσης σε τοπικά ελάχιστα.



Βιβλιογραφία

- [1] Κ. Διαμαντάρας, Δ. Μπότσης, Μηχανική Μάθηση, Εκδόσεις Κλειδάριθμος, 2019.
- [2] S. Theodoridis, Machine Learning: A Bayesian and Optimization Perspective, 2nd Edition, Academic Press, 2020.
- [3] Shai Ben-David and Shai Shalev-Shwartz, Understanding Machine Learning, Cambridge University Press
- Διαφάνειες των συγγραφέων για το σύγγραμμα [1].

Μηχανική Μάθηση

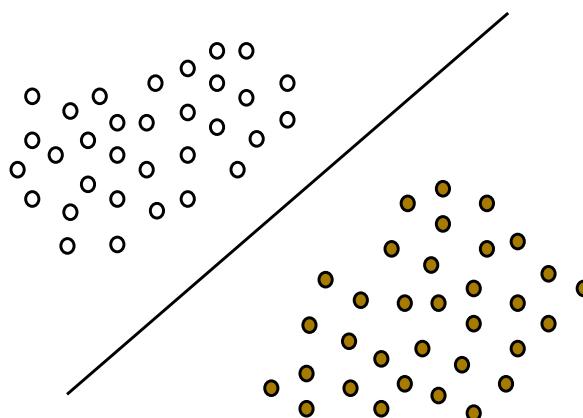
1^ο εξάμηνο | ακαδημαϊκό έτος 2023-2024

Θάνος Βουλόδημος
Επ. Καθηγητής ΣΗΜΜΥ ΕΜΠ

Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines)



Το πρόβλημα της ταξινόμησης



Προβλήματα στην ταξινόμηση με νευρωνικά δίκτυα

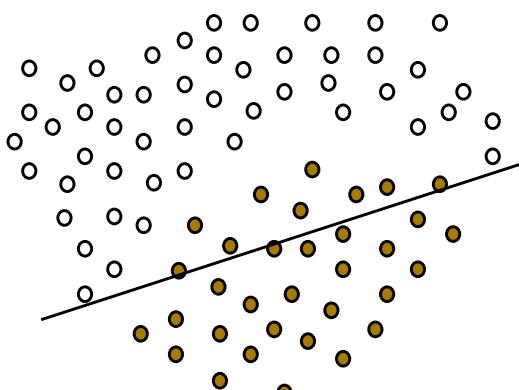
- Η ταξινόμηση με perceptrons δουλεύει μόνο με γραμμικά διαχωρίσιμες κλάσεις
- Η ταξινόμηση με δίκτυα MLP υποφέρει από βραδεία εκπαίδευση

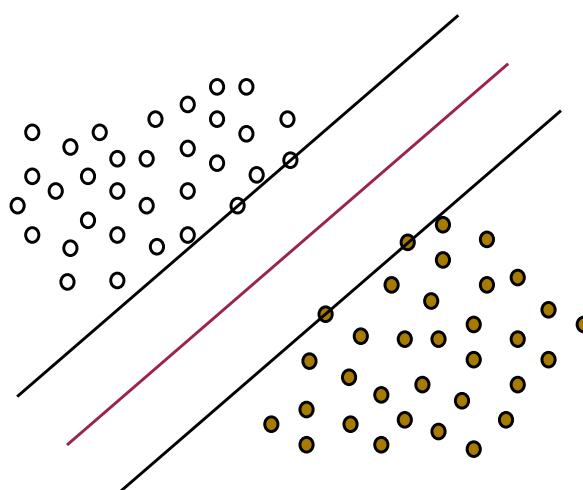
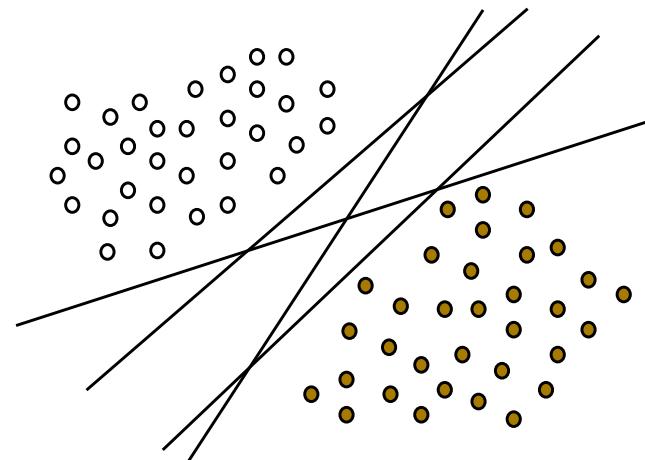
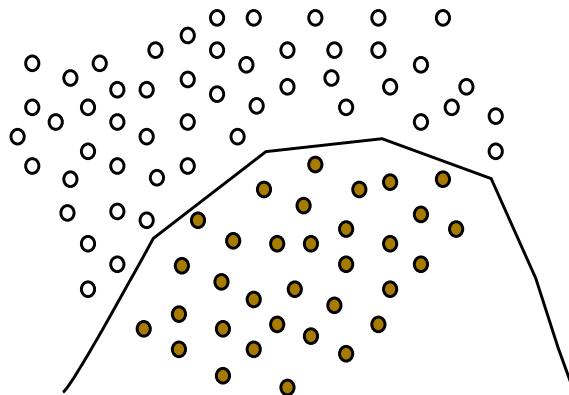
Ιδέα

- Αν επικεντρωθούμε στο πρόβλημα της ταξινόμησης μπορούμε να πετύχουμε καλύτερους χρόνους εκπαίδευσης και καλύτερες ιδιότητες γενίκευσης

Μηχανική Μάθηση | ΔΠΜΣ ΕΔΕΜΜ | 1^ο εξάμηνο 2023-2024

Μη γραμμικά διαχωρίσιμες κλάσεις





Διατύπωση προβλήματος

Δίνεται ένα σύνολο ζευγών $(\mathbf{x}_1, d_1), \dots, (\mathbf{x}_p, d_p)$
με $d_i = -1$ αν $\mathbf{x}_i \in \mathcal{C}_o$ και $d_i = 1$ αν $\mathbf{x}_i \in \mathcal{C}_1$

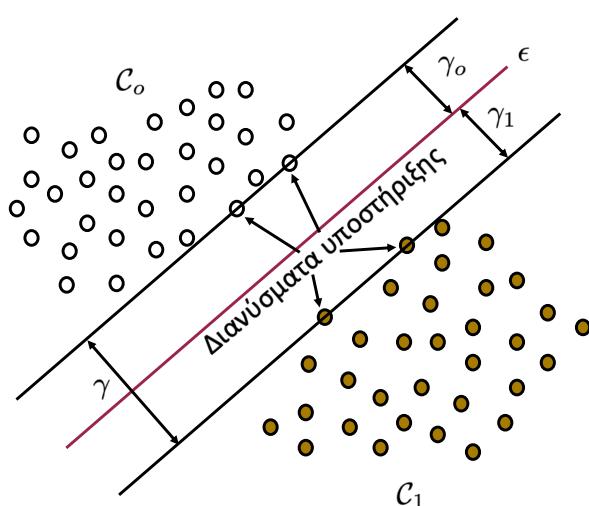
Ζητάμε την εύρεση των βαρών \mathbf{w} και του κατωφλίου w_o , έτσι ώστε:
 $\mathbf{w}^\top \mathbf{x}_i + w_o \geq 0$ αν $d_i = 1$ ($\mathbf{x}_i \in \mathcal{C}_o$)
 $\mathbf{w}^\top \mathbf{x}_i + w_o < 0$ αν $d_i = -1$ ($\mathbf{x}_i \in \mathcal{C}_1$)

Υπόθεση

Υπάρχει τέτοια ευθεία (οι κλάσεις είναι γραμμικά διαχωρίσιμες)

Απαίτηση

Η ευθεία που θα κατασκευαστεί πρέπει να έχει όσο το δυνατόν
μεγαλύτερο περιιθώριο ταξινόμησης



$$\gamma_o = \min_{\mathbf{x} \in \mathcal{C}_o} d(\mathbf{x}, \epsilon)$$

$$\gamma_1 = \min_{\mathbf{x} \in \mathcal{C}_1} d(\mathbf{x}, \epsilon)$$

$$\gamma = \gamma_o + \gamma_1$$

Κανονικό υπερεπίπεδο

$$\gamma_o = \gamma_1$$

$$\begin{aligned} \mathbf{w}^T \mathbf{x}_i + w_o &\geq 1 \text{ αν } \mathbf{x}_i \in \mathcal{C}_o \\ \mathbf{w}^T \mathbf{x}_i + w_o &\leq -1 \text{ αν } \mathbf{x}_i \in \mathcal{C}_1 \end{aligned}$$

- Παρατηρώ τι συμβαίνει με την κλιμάκωση του \mathbf{w} . Για παράδειγμα,
- Αν

$$\begin{aligned} \mathbf{w}^T \mathbf{x}_k + b &\leq -10 & \text{όταν } t_k = -1 \\ \mathbf{w}^T \mathbf{x}_k + b &\geq 10 & \text{όταν } t_k = 1 \end{aligned}$$

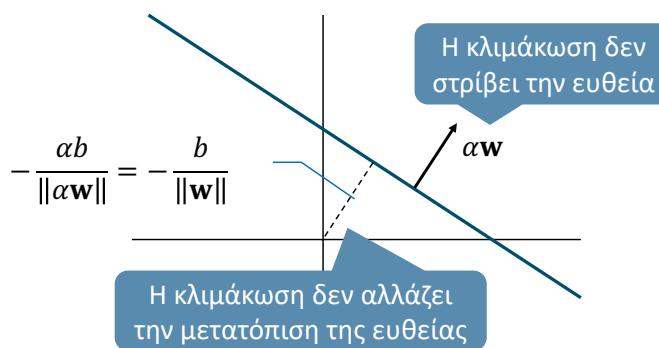
• Τότε

$$\begin{aligned} 2\mathbf{w}^T \mathbf{x}_k + 2b &\leq -20 & \text{όταν } t_k = -1 \\ 2\mathbf{w}^T \mathbf{x}_k + 2b &\geq 20 & \text{όταν } t_k = 1 \end{aligned}$$

$$\begin{aligned} 3\mathbf{w}^T \mathbf{x}_k + 3b &\leq -30 & \text{όταν } t_k = -1, \text{ κλπ} \\ 3\mathbf{w}^T \mathbf{x}_k + 3b &\geq 30 & \text{όταν } t_k = 1 \end{aligned}$$

Επίδραση της κλιμάκωσης (2)

- Στην ουσία όμως η κλιμάκωση αφήνει την ευθεία στην ίδια θέση



Υπολογισμός περιθωρίου ταξινόμησης



Η συνάρτηση $g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_o$ ένα μέτρο της απόστασης του \mathbf{x} από το βέλτιστο υπερεπίπεδο (όπου \mathbf{w} και w_o τα βέλτιστα βάρη).

Υπολογίζουμε το \mathbf{x} ως $\mathbf{x} = \mathbf{x}_p + r \frac{\mathbf{w}}{\|\mathbf{w}\|}$, όπου r η απόσταση του \mathbf{x} από το βέλτιστο υπερεπίπεδο

$$\text{Συνεπώς } g(\mathbf{x}) = \mathbf{w}^T \left(\mathbf{x} = \mathbf{x}_p + r \frac{\mathbf{w}}{\|\mathbf{w}\|} \right) + w_o$$

$$\Rightarrow g(\mathbf{x}) = \mathbf{w}^T \mathbf{x}_p + w_o + r \frac{\mathbf{w}^T \mathbf{w}}{\|\mathbf{w}\|}$$

$$\Rightarrow g(\mathbf{x}) = r \frac{\|\mathbf{w}\|^2}{\|\mathbf{w}\|} \Rightarrow r = \frac{g(\mathbf{x})}{\|\mathbf{w}\|}$$

Άρα αφού για τα διανύσματα υποστήριξης έχουμε $g(x) = 1$ ($\mathbf{x}_i \in \mathcal{C}_o$) και $g(x) = -1$ ($\mathbf{x}_i \in \mathcal{C}_1$)

Τελικά: $\boxed{\gamma = \frac{2}{\|\mathbf{w}\|}}$



Ορισμός προβλήματος βελτιστοποίησης

Υπολόγισε το ελάχιστο της συνάρτησης:

$$\mathcal{J}(\mathbf{w}, w_o) = \frac{1}{2} \|\mathbf{w}\|^2$$

υπό τους περιορισμούς των P ανισοτήτων:

$$d_i(\mathbf{w}^\top \mathbf{x}_i + w_o) \geq 1, \quad i = 1, \dots, P$$

Παρατηρήσεις

- Η συνάρτηση κόστους είναι κυρτή
- Οι περιορισμοί είναι γραμμικοί

Καλούματε να επιλύσουμε ένα πρόβλημα τετραγωνικού προγραμματισμού

Ορίζουμε τη συνάρτηση κόστους:

$$\mathcal{L}(\mathbf{w}, w_o, \lambda_1, \dots, \lambda_p) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^P \lambda_i [d_i(\mathbf{w}^\top \mathbf{x}_i + w_o) - 1]$$

$$\text{με } \lambda_i \geq 0, \quad i = 1, \dots, P$$

Η συνάρτηση αυτή πρέπει να ελαχιστοποιηθεί ως προς τα \mathbf{w} , w_o και να μεγιστοποιηθεί ως προς τα λ_i

Συνθήκες Karush-Kuhn-Tucker (για το βέλτιστο σημείο)

$$\frac{\partial \mathcal{L}}{\partial w_o} = 0 \quad \frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 \quad \lambda_i [d_i(\mathbf{w}^\top \mathbf{x}_i + w_o) - 1] \geq 0, \quad i = 1, \dots, P$$



Από τις συνθήκες KKT έχουμε:

$$\frac{\partial \mathcal{L}}{\partial w_o} = 0 \longrightarrow \sum_{i=1}^P \lambda_i d_i = 0$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 \longrightarrow \mathbf{w} = \sum_{i=1}^P \lambda_i d_i \mathbf{x}_i$$

Συνεπώς η βέλτιστη διαχωριστική επιφάνεια δίνεται από τη σχέση:

$$g^*(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + w_o = \sum_{i=1}^P \lambda_i d_i \mathbf{x}_i^\top \mathbf{x} + w_o$$

Για τα διανύσματα υποστήριξης ισχύει ότι:

$$d_i(\mathbf{w}^\top \mathbf{x}_i + w_o) = 1 \longrightarrow w_o = \frac{1}{d_i} - \mathbf{w}^\top \mathbf{x}_i$$

Για λόγους αριθμητικής ευστάθειας, χρησιμοποιούμε τη σχέση:

$$w_o = \frac{1}{|I_{sv}|} \sum_{i \in I_{sv}} \left(\frac{1}{d_i} - \mathbf{w}^\top \mathbf{x}_i \right)$$

όπου:

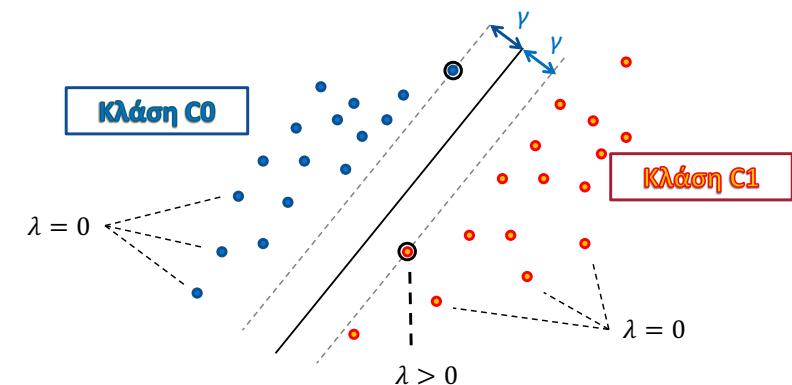
$$I_{sv} = \{i : \mathbf{x}_i \text{ διάνυσμα υποστήριξης}\}$$

Παρατήρηση

Οι μόνοι πολλαπλασιαστές λ_i που μπορούν να είναι θετικοί είναι αυτοί που αντιστοιχούν σε κάποιο διάνυσμα υποστήριξης \mathbf{x}_i .

Για τους υπόλοιπους ισχύει $\lambda_i = 0$.

- Αν το \mathbf{x}_k είναι διάνυσμα υποστήριξης τότε $\lambda_k > 0$:
 $d_k(\mathbf{w}^T \mathbf{x}_k + w_0) = 1 \Leftrightarrow \lambda_k > 0$
- Αλλιώς $\lambda_k = 0$:
 $d_k(\mathbf{w}^T \mathbf{x}_k + w_0) > 1 \Leftrightarrow \lambda_k = 0$
- Συνεπώς το βέλτιστο \mathbf{w} είναι γραμμικός συνδυασμός των διανυσμάτων υποστήριξης και μόνο



Δυϊκό πρόβλημα (1)

Από τα παραπάνω έχουμε:

$$\frac{1}{2} \|\mathbf{w}\|^2 = \frac{1}{2} \mathbf{w}^T \mathbf{w} = \frac{1}{2} \sum_{i=1}^P \sum_{j=1}^P \lambda_i \lambda_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\begin{aligned} \sum_{i=1}^P \lambda_i [d_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1] &= \sum_{i=1}^P \lambda_i d_i \sum_{j=1}^P \lambda_j d_j \mathbf{x}_j^T \mathbf{x}_i + w_0 \sum_{i=1}^P \lambda_i d_i - \sum_{i=1}^P \lambda_i \\ &= \sum_{i=1}^P \sum_{j=1}^P \lambda_i \lambda_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^P \lambda_i \end{aligned}$$

Επομένως:

$$\mathcal{L}(\lambda_1, \dots, \lambda_P) = \sum_{i=1}^P \lambda_i - \frac{1}{2} \sum_{i=1}^P \sum_{j=1}^P \lambda_i \lambda_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j$$

Δυϊκό πρόβλημα (2)

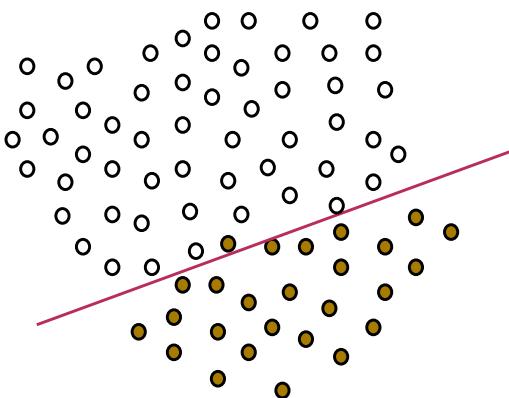
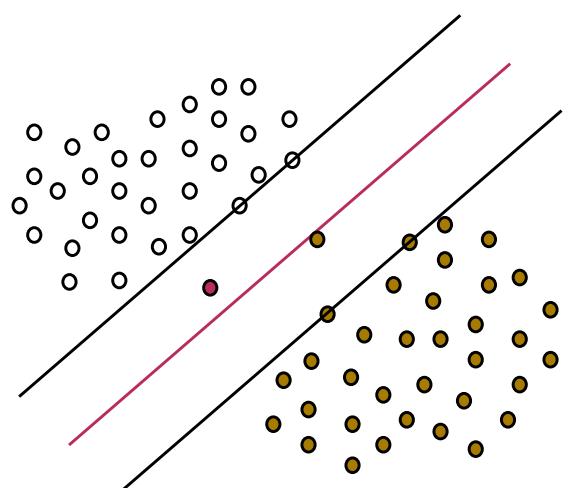
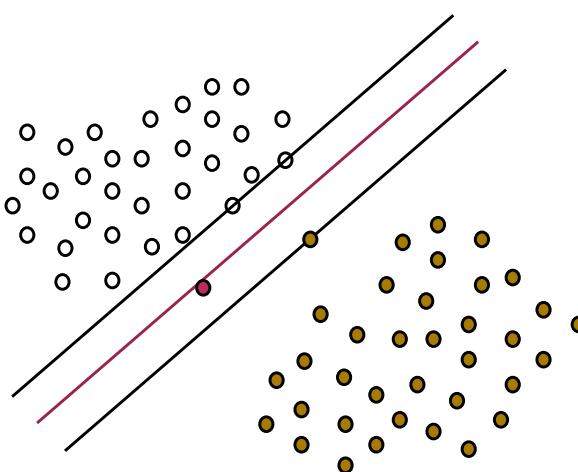
Ορισμός δυϊκού προβλήματος βελτιστοποίησης

Υπολόγισε το ελάχιστο της συνάρτησης:

$$\mathcal{L}^d(\lambda_1, \dots, \lambda_P) = \frac{1}{2} \sum_{i=1}^P \sum_{j=1}^P \lambda_i \lambda_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^P \lambda_i$$

ως προς τα $\lambda_1, \dots, \lambda_P$, υπό τους περιορισμούς

$$\sum_{i=1}^P \lambda_i d_i = 0 \quad \lambda_i \geq 0, i = 1, \dots, P$$



Μεταβλητές χαλαρότητας

Ορίζουμε ένα σύνολο $\{\xi_i\}_{i=1}^N$ από θετικές τιμές και τις εισάγουμε στην εξίσωση της βέλτιστης ευθείας διαχωρισμού ως εξής:

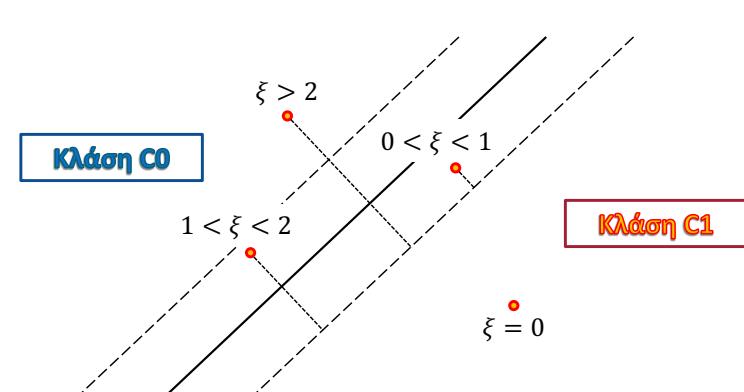
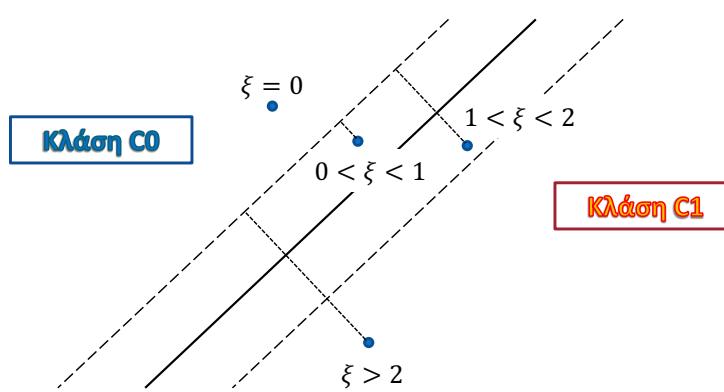
$$d_i(\mathbf{w}^\top \mathbf{x}_i + w_0) \geq 1 - \xi_i, i = 1, \dots, P$$

με $\xi_i \geq 0, i = 1, \dots, P$

Παρατηρούμε ότι:

Αν $\xi_i \leq 1$ δεν υπάρχει λάθος ταξινόμηση

Αν $\xi_i > 1$ υπάρχει λάθος ταξινόμηση
και το πρότυπο \mathbf{x}_i ταξινομείται σε λάθος κλάση



Ορισμός προβλήματος βελτιστοποίησης

Υπολόγισε το ελάχιστο της συνάρτησης:

$$\mathcal{J}(\mathbf{w}, w_o) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^P \xi_i$$

υπό τους περιορισμούς των P ανισοτήτων:

$$d_i(\mathbf{w}^\top \mathbf{x}_i + w_o) \geq 1 - \xi_i, i = 1, \dots, P$$

όπου η παράμετρος C επιλέγεται από το χρήστη και είναι το βάρος του κόστους των λάνθασμένων ταξινομήσεων

Αν $C = 0$ τότε αγνοούμε τελείως τις παραμέτρους χαλαρότητας, επομένως δεν μας ενδιαφέρει αν έχουμε λανθασμένες ταξινομήσεις
Αν $C \rightarrow \infty$ τότε δίνουμε έμφαση στη σωστή ταξινόμηση των προτύπων

Ορισμός δυϊκού προβλήματος βελτιστοποίησης

Υπολόγισε το ελάχιστο της συνάρτησης:

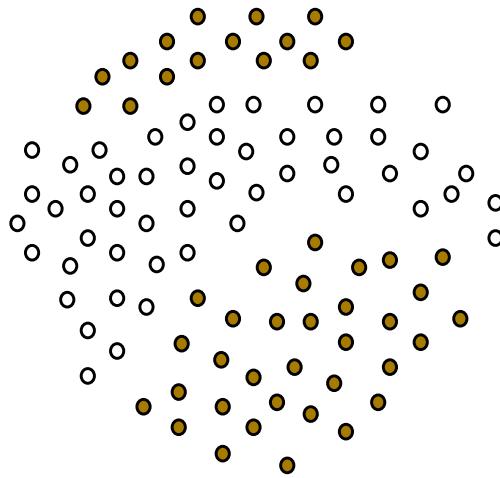
$$\mathcal{L}_{ns}^d(\lambda_1, \dots, \lambda_P) = \frac{1}{2} \sum_{i=1}^P \sum_{j=1}^P \lambda_i \lambda_j d_i d_j \mathbf{x}_i^\top \mathbf{x}_j - \sum_{i=1}^P \lambda_i$$

ως προς τα $\lambda_1, \dots, \lambda_P$, υπό τους περιορισμούς

$$\sum_{i=1}^P \lambda_i d_i = 0 \quad 0 \leq \lambda_i \leq C, i = 1, \dots, P$$

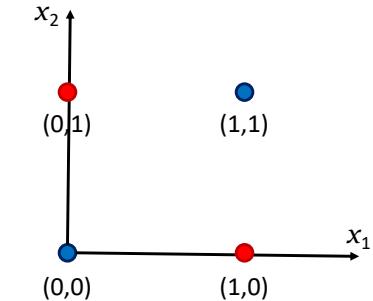
Παρατήρηση

Παρατηρούμε ότι τα ξ_i εμφανίζονται μόνο στο δεύτερο περιορισμό



- Το πρόβλημα XOR ξανά ...

x_1	0	0	1	1
x_2	0	1	0	1
t	-1	1	1	-1



- Ξέρουμε ότι το πρόβλημα δεν είναι γραμμικά διαχωρίσιμο

Παράδειγμα

- Ας θεωρήσουμε τον μετασχηματισμό

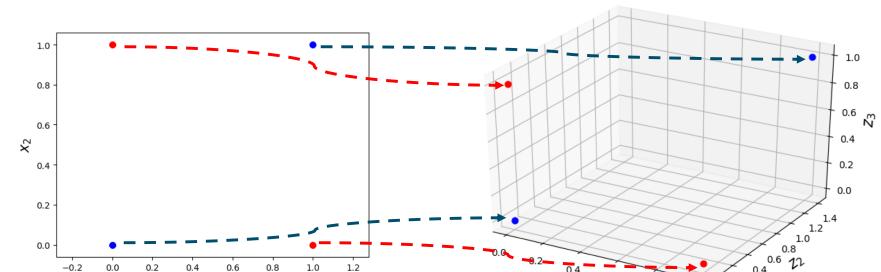
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mapsto \mathbf{z} = \Phi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}$$

x_1	0	0	1	1
x_2	0	1	0	1
t	-1	1	1	-1



z_1	0	0	1	1
z_2	0	0	0	1.4142
z_3	0	1	0	1
t	-1	1	1	-1

Μετασχηματισμός



$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

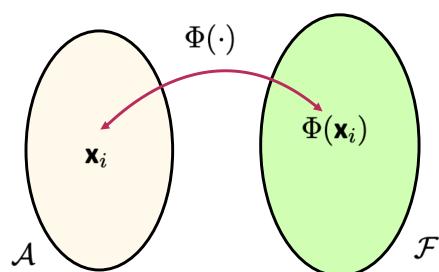
$$\mathbf{z} = \Phi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}$$



- Το μετασχηματισμένο πρόβλημα τώρα είναι γραμμικά διαχωρίσιμο!!
- Δοκιμάστε πχ. $\mathbf{w} = [1, -\sqrt{2}, 1], b = -0.5$

w_1	1
w_2	1.4142
w_3	1

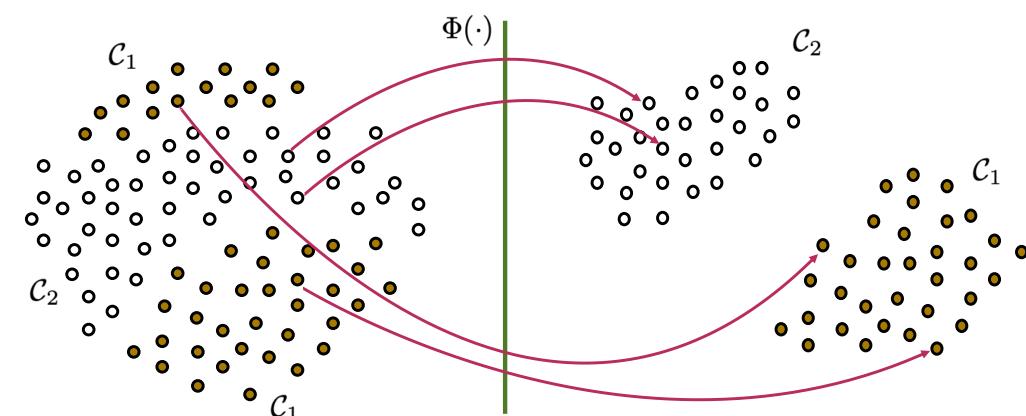
z_1	0	0	1	1
z_2	0	0	0	1.4142
z_3	0	1	0	1
$\mathbf{w}^T \mathbf{z}$	0	1	1	0
$\mathbf{w}^T \mathbf{z} + b$	-0.5	0.5	0.5	-0.5



\mathcal{A} : χώρος εισόδου
 \mathcal{F} : χώρος χαρακτηριστικών
 $\Phi(\cdot)$: μη-γραμμική συνάρτηση απεικόνισης

Θεώρημα Cover

Κάθε πολυδιάστατος χώρος με μη γραμμικά διαχωρίσιμα πρότυπα, μπορεί να μετασχηματιστεί σε ένα νέο χώρο στον οποίο τα πρότυπα είναι γραμμικά διαχωρίσιμα με υψηλή πιθανότητα, αρκεί ο μετασχηματισμός να είναι μη γραμμικός και ο νέος αυτός χώρος να έχει την απαραίτητη διάσταση



Βέλτιστη διαχωριστική επιφάνεια:

$$g^*(\mathbf{x}) = \mathbf{w}^\top \Phi(\mathbf{x}) + w_o = \sum_{i=1}^P \lambda_i d_i \Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}) + w_o$$

Κατώφλι:

$$w_o = \frac{1}{|I_{sv}|} \sum_{i \in I_{sv}} \left(\frac{1}{d_i} - \mathbf{w}^\top \Phi(\mathbf{x}_i) \right)$$

Συνάρτηση κόστους του δυϊκού προβλήματος:

$$\mathcal{L}(\lambda_1, \dots, \lambda_P) = \sum_{i=1}^P \lambda_i - \frac{1}{2} \sum_{i=1}^P \sum_{j=1}^P \lambda_i \lambda_j d_i d_j \Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}_j)$$



Παρατήρηση

Παρατηρούμε ότι σε όλες τις εξισώσεις που χρησιμοποιούμε εμφανίζονται γινόμενα της μορφής $\Phi(\mathbf{x})^\top \Phi(\mathbf{y})$.

Η συνάρτηση $\Phi(\cdot)$ δεν εμφανίζεται ποτέ μόνη της.

Ορισμός

Ορίζουμε τη συνάρτηση $k(x, y) = \Phi(\mathbf{x})^\top \Phi(\mathbf{y})$, την οποία θα ονομάζουμε συνάρτηση πυρήνα.

Χρησιμοποιώντας τη συνάρτηση πυρήνα κάνουμε οικονομία πράξεων ειδικά όταν η διάσταση του $\Phi(\mathbf{x})$ είναι πολύ μεγαλύτερη από τη διάσταση του \mathbf{x} (όπως συνήθως συμβαίνει).

$$\text{Έστω } \mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^\top$$

$$\Phi(\mathbf{x}) = \begin{bmatrix} x_1^2 & \sqrt{2}x_1x_2 & x_2^2 \end{bmatrix}^\top$$

$$\text{Για } \mathbf{x} = \begin{bmatrix} 1 & 2 \end{bmatrix}^\top$$

$$\Phi([1 \ 2]^\top) = \begin{bmatrix} 1 & 2\sqrt{2} & 4 \end{bmatrix}^\top$$

$$\begin{aligned} k(x, y) &= \Phi(\mathbf{x})^\top \Phi(\mathbf{y}) = (x_1^2 y_1^2 + 2x_1 y_1 y_2 + x_2^2 y_2^2) \\ &= (x_1 y_1 + x_2 y_2)^2 = (\mathbf{x}^\top \mathbf{y})^2 \end{aligned}$$



To Kernel trick

- Επιλέγουμε μια συνάρτηση πυρήνα $K(\mathbf{a}, \mathbf{b})$ που να παράγεται από κάποια συνάρτηση Φ ως $K(\mathbf{a}, \mathbf{b}) = \Phi(\mathbf{a})^\top \Phi(\mathbf{b})$
- Δεν είναι ανάγκη να ξέρουμε την Φ αρκεί να ξέρουμε μαθηματικά ότι υπάρχει
- Δεν είναι όλες οι συναρτήσεις $K(\cdot, \cdot)$ συναρτήσεις πυρήνα. Πρέπει να ικανοποιούν τις προϋποθέσεις του Θεωρήματος Mercer
- Χρησιμοποιούμε την K για να υπολογίσουμε το \mathbf{Q} στο δυϊκό πρόβλημα.
- Λύνουμε το δυϊκό πρόβλημα.
- Επειδή η Φ μετασχηματίζει τα δεδομένα σε χώρο μεγάλων διαστάσεων ελπίζουμε ότι το πρόβλημα εκεί θα λύνεται γραμμικά. Όμως η συνάρτηση διαχωρισμού στον αρχικό χώρο \mathbf{x} δεν θα είναι γραμμική.



Επιλογή συναρτήσεων πυρήνα



Θεώρημα Mercer

Έστω $k(\mathbf{x}, \mathbf{y})$ ένας συνεχής συμμετρικός πυρήνας, με $\mathbf{a} \leq \mathbf{x}, \mathbf{y} \leq \mathbf{b}$.

Ο πυρήνας $k(\mathbf{x}, \mathbf{y})$ μπορεί να γραφεί ως:

$$k(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{\infty} \alpha_i \Phi_i(\mathbf{x}) \Phi_i(\mathbf{y})$$

με $\alpha_i > 0, \forall i$, αν και μόνο αν:

$$\int_b^a \int_b^a k(\mathbf{x}, \mathbf{y}) \psi(\mathbf{x}) \psi(\mathbf{y}) dx dy \geq 0$$

$$\text{για κάθε } \psi(\cdot) \text{ για την οποία } \int_b^a \psi^2(\mathbf{x}) dx \leq \infty$$



Γκαουσιανή RBF:

$$e^{-\|\mathbf{x}-\mathbf{y}\|^2/(2\sigma^2)}$$

Πολυωνυμική:

$$[\mathbf{x}^\top \mathbf{y} + \theta]^p$$

Σιγμοειδής:

$$\tanh(\alpha \mathbf{x}^\top \mathbf{y} + \theta)$$

Αντίστροφη πολυτετραγωνική:

$$\frac{1}{\sqrt{\|\mathbf{x} - \mathbf{y}\|^2 + c^2}}$$

Υπολόγισε το μέγιστο της συνάρτησης:

$$\mathcal{L}_{SVM}(\lambda_1, \dots, \lambda_P) = \sum_{i=1}^P \lambda_i - \frac{1}{2} \sum_{i=1}^P \sum_{j=1}^P \lambda_i q_{ij} \lambda_j$$

υπό τους περιορισμούς:

$$0 \leq \lambda_i \leq C \quad \sum_{i=1}^P \lambda_i d_i = 0$$

όπου: $q_{ij} = d_i d_j k(\mathbf{x}_i, \mathbf{x}_j)$

Παρατήρηση

Το πλήθος των στοιχείων του πίνακα $\mathbf{Q} = [q_{ij}]$ είναι P^2 , συνεπώς είναι αρκετά πολύπλοκη η επίλυση του προβλήματος.



Μέθοδος τεμαχισμού

Η συνάρτηση κόστους δεν αλλάζει αν αφαιρέσουμε τις γραμμές και τις στήλες του \mathbf{Q} που αντιστοιχούν σε μηδενικές τιμές του λ_i .

Διαλέγουμε σε κάθε βήμα την επίλυση του προβλήματος για το τμήμα του \mathbf{Q} που αντιστοιχεί στα μη μηδενικά λ_i από το προηγούμενο πρόβλημα και επιπλέον στα K χειρότερα λ_i (που παραβιάζουν περισσότερο τις συνθήκες KKT).

Μέθοδος Osuna

Αν επιλύσουμε ένα μικρότερο πρόβλημα, επιλέγοντας μερικές μόνο γραμμές του \mathbf{Q} έτσι ώστε να περιέχεται τουλάχιστον ένα λ_i που παραβιάζει τις συνθήκες KKT τότε η συνάρτηση κόστους μειώνεται και όλοι οι περιορισμοί συνεχίζουν να ικανοποιούνται.

Επιλύσουμε το πρόβλημα προσθέτοντας μία μεταβλητή λ_i που παραβιάζει τις συνθήκες και αφαιρώντας μία μεταβλητή για την οποία $\lambda_i = 0$ ή $\lambda_j = C$

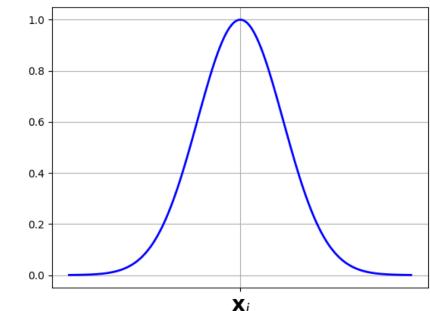
(Support Vector Machines – SVM)

- Μηχανές μάθησης που υλοποιούν την παραπάνω θεωρία με πυρήνα ή χωρίς
- Απαιτούν σαν είσοδο τα διανύσματα \mathbf{x}_k μαζί με τους στόχους t_k άρα ανήκουν στην κατηγορία των μηχανών μάθησης με επίβλεψη
- Βασίζονται στην λύση προβλήματος τετραγωνικού προγραμματισμού. Το πρόβλημα έχει μελετηθεί εκτενώς στα μαθηματικά. Υπάρχουν διάφορες υλοποιήσεις.



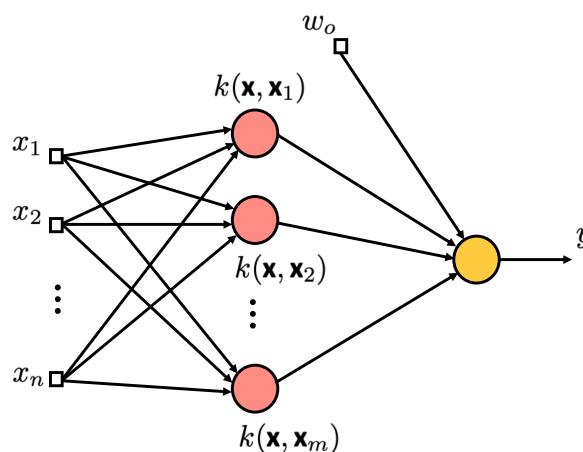
- Ο Γκαουσσιανός πυρήνας

$$K(\mathbf{x}_i, \mathbf{x}) = \exp\left\{-\frac{\|\mathbf{x}_i - \mathbf{x}\|^2}{2\sigma^2}\right\}$$



παράγεται από μια κρυφή συνάρτηση Φ απεύρων διαστάσεων.

- Είναι γι' αυτό το λόγο ίσως ο πιο δημοφιλής πυρήνας.



Ταξινόμηση σε περισσότερες από δύο κλάσεις



- Τα μέχρι τώρα αφορούν σε προβλήματα δυαδικής ταξινόμησης (binary classification)
- Ωστόσο πολλές πρακτικές εφαρμογές περιλαμβάνουν περισσότερες από δύο κλάσεις (multi-class classification)
- Επομένως, είναι εμφανής η ανάγκη γενίκευσης του προβλήματος δυαδικής ταξινόμησης σε ταξινόμηση με περισσότερες κλάσεις
- Διάφορες προσεγγίσεις – δύο απλές είναι οι:
 - One-against-all (ή one-against-the-rest)
 - One-against-one

One-against-all



- Κατασκευάζει δυαδικά μοντέλα SVM με μια κατηγορία ως θετική και τις υπόλοιπες ως αρνητικές
- Π.χ. για 4 κλάσεις, θα δημιουργηθούν τα ακόλουθα 4 SVM:

$y_i = 1$	$y_i = -1$	Συνάρτηση Απόφασης
Κλάση 1	Κλάσεις 2,3,4	$f^1(x) = (w^1)^T x + b^1$
Κλάση 2	Κλάσεις 1,3,4	$f^2(x) = (w^2)^T x + b^2$
Κλάση 3	Κλάσεις 1,2,4	$f^3(x) = (w^3)^T x + b^3$
Κλάση 4	Κλάσεις 1,2,3	$f^4(x) = (w^4)^T x + b^4$

- Για οποιοδήποτε δεδομένο που ανήκει στην κλάση i , περιμένουμε ότι:

$$f^i(x) \geq 1 \text{ και } f^j(x) \leq -1, i \neq j$$

- Επομένως, ο κανόνας απόφασης είναι:

$$\text{Αναμενόμενη κλάση} = \arg \max_{i=1,\dots,4} f^i(x)$$



- Εδώ κατασκευάζονται συνολικά $\binom{k}{2} = \frac{k(k-1)}{2}$ SVMs, για τη δυαδική ταξινόμηση όλων των κλάσεων ανά δύο
- Κάθε δυαδικός ταξινομητής εκπαιδεύεται με δεδομένα 2 κλάσεων
- Κάθε νέα παρατήρηση x προς ταξινόμηση δοκιμάζεται σε όλους τους ταξινομητές
- Αν το πρόβλημα των κλάσεων i και j δείξει ότι η x παρατήρηση θα πρέπει να είναι στην i , η κλάση i παίρνει μία ψήφο
- Στο τέλος, η παρατήρηση x αντιστοχίζεται στην κλάση που έχει λάβει τις περισσότερες ψήφους
- Π.χ.

Κλάσεις	Νικητής
1 2	1
1 3	1
1 4	1
2 3	2
2 4	4
3 4	3

Νικήτρια η κλάση 1				
Κλάση	1	2	3	4
Πλήθος ψήφων	3	1	1	1

- Εδώ οι τιμές των στόχων ανήκουν σε συνεχές σύνολο τιμών
- Έστω σύνολο προτύπων x_i και στόχων $t_i, i = 1, \dots, N$.
- Συνάρτηση σφάλματος με ανοχή ε :

$$l_\varepsilon(t, g) = \begin{cases} 0, & \text{αν } |t - g| \leq \varepsilon \\ |t - g| - \varepsilon, & \text{αν } |t - g| > \varepsilon \end{cases}$$

- Η συνάρτηση τιμωρεί τη διαφορά μεταξύ του στόχου και της εκτιμώμενης τιμής g μόνο αν η απόλυτη διαφορά είναι μεγαλύτερη από μια θετική σταθερά ε (ανοχή στο σφάλμα)
- Για γραμμικά επιλύσιμο πρόβλημα, ξεκινάμε θεωρώντας ότι υπάρχει (w, w_0) , ώστε $L_\varepsilon(w, w_0) = \sum_{i=1}^N l_\varepsilon(t_i, g(x_i; w, w_0)) = 0$. Επιλέγεται η λύση με τη μικρότερη τιμή $\|w\|^2$.
- Πρόβλημα παλινδρόμησης με ανοχή ε :

$$\min \frac{1}{2} \|w\|^2$$

$$t_i - w^T x_i - w_0 \leq \varepsilon \quad t_i - w^T x_i - w_0 \geq -\varepsilon$$



Βιβλιογραφία

- [1] K. Διαμαντάρας, Δ. Μπότσης, Μηχανική Μάθηση, Εκδόσεις Κλειδάριθμος, 2019.
- [2] S. Theodoridis, Machine Learning: A Bayesian and Optimization Perspective, 2nd Edition, Academic Press, 2020.
- [3] Shai Ben-David and Shai Shalev-Shwartz, Understanding Machine Learning, Cambridge University Press
- Διαφάνειες των συγγραφέων για το σύγγραμμα [1].

ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ



Δέντρα αποφάσεων

Γιώργος Στάμου

Καθηγητής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών ΕΜΠ
Διευθυντής Εργαστηρίου Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης - AILS Lab

ΔΕΝΤΡΑ ΑΠΟΦΑΣΗΣ - ΠΡΟΒΛΗΜΑ ΤΑΞΙΝΟΜΗΣΗΣ



Πρόβλημα

X : σύνολο στιγμιοτύπων

Y : σύνολο ετικετών

$f: X \rightarrow Y$: ιδιαίτερος ταξινομητής

$H = \{h | h: X \rightarrow Y\}$: σύνολο ταξινομητών (υποθέσεις)

Στήλες χαρακτηριστικών: x_i

Στήλη απόφασης: y_i

Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	FALSE	No
Sunny	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Rainy	Mild	High	FALSE	Yes
Rainy	Cool	Normal	FALSE	Yes
Rainy	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Sunny	Mild	High	FALSE	No
Sunny	Cool	Normal	TRUE	Yes
Rainy	Mild	Normal	FALSE	Yes
Sunny	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Mild	High	TRUE	No

(x_i, y_i)

Είσοδος

Σύνολο δεδομένων $\{\langle x_i, y_i \rangle\}_{i=1}^n = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$

Έξοδος

Υπόθεση $h \in H$ που προσεγγίζει το f

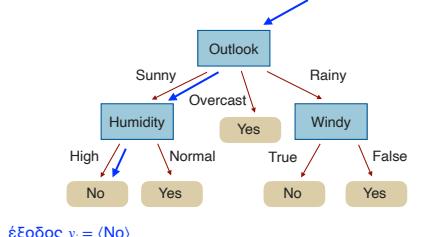
2

ΔΕΝΤΡΑ ΑΠΟΦΑΣΗΣ - ΟΡΙΣΜΟΣ



Δέντρο απόφασης

Είσοδος $x_i = (\text{Sunny}, \text{Hot}, \text{High}, \text{True})$



Σε κάθε εσωτερικό κόμβο ελέγχεται η τιμή του χαρακτηριστικού x_i

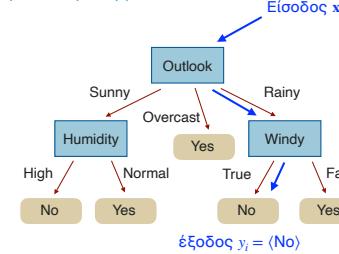
Σε κάθε διακλάδωση επιλέγεται μία τιμή του χαρακτηριστικού x_i

Σε κάθε φύλλο αποδίδεται μία ετικέτα γ στο στοιχείο

ΔΕΝΤΡΑ ΑΠΟΦΑΣΗΣ - ΟΡΙΣΜΟΣ

Δέντρο απόφασης

Είσοδος $x_i = (\text{Rainy}, \text{Mild}, \text{Normal}, \text{True})$



Σε κάθε εσωτερικό κόμβο ελέγχεται η τιμή του χαρακτηριστικού x_i

Σε κάθε διακλάδωση επιλέγεται μία τιμή του χαρακτηριστικού x_i

Σε κάθε φύλλο αποδίδεται μία ετικέτα γ στο στοιχείο

Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	FALSE	No
Sunny	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Rainy	Mild	High	FALSE	Yes
Rainy	Cool	Normal	FALSE	Yes
Rainy	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Sunny	Mild	High	FALSE	No
Sunny	Cool	Normal	TRUE	Yes
Rainy	Mild	Normal	FALSE	Yes
Sunny	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Mild	High	TRUE	No

3

4

ΔΕΝΤΡΑ ΑΠΟΦΑΣΗΣ - ΚΑΜΠΥΛΕΣ ΔΙΑΧΩΡΙΣΜΟΥ

Δέντρο απόφασης

Καμπύλη διαχωρισμού

▶ Η καμπύλη διαχωρισμού χωρίζει το χώρο χαρακτηριστικών σε (υπερ-)ικύβους παράλληλους των αξόνων

▶ Κάθε (υπερ-)κυβική επιφάνεια αντιστοιχίζεται σε μία ετικέτα - ή (στη γενική περίπτωση) σε μία κατανομή πιθανοτήτων πάνω στις ετικέτες

5

ΔΕΝΤΡΑ ΑΠΟΦΑΣΗΣ - ΕΚΦΡΑΣΤΙΚΟΤΗΤΑ

Ικανότητα αναπαράστασης

Δέντρο απόφασης για XOR

IF (Outlook IS Overcast) v ((Outlook IS Sunny) \wedge Humidity IS Normal) v ((Outlook IS Rainy) \wedge Windy IS False)
THEN (PlayTennis IS YES)

▶ Ένα δέντρο απόφασης αντιστοιχεί σε μία κανονική διαζευκτική μορφή (disjunctive normal form - DNF) μίας λογικής έκφρασης

▶ Τα δέντρα απόφασης μπορούν να αναπαραστήσουν οποιαδήποτε λογική συνάρτηση

6

ΔΕΝΤΡΑ ΑΠΟΦΑΣΗΣ - ΕΚΦΡΑΣΤΙΚΟΤΗΤΑ

Μοναδικότητα

Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	FALSE	No
Sunny	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Rainy	Mild	High	FALSE	Yes
Rainy	Cool	Normal	FALSE	Yes
Rainy	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Sunny	Mild	High	FALSE	No
Sunny	Cool	Normal	TRUE	Yes
Rainy	Mild	Normal	FALSE	Yes
Sunny	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Mild	High	TRUE	No

▶ Διαφορετικά δέντρα απόφασης μπορούν να είναι ισοδύναμα για το συγκεκριμένο σύνολο δεδομένων (να ταξινομούν στην ίδια κλάση στηγμότυπο)

▶ Πόσα διαφορετικά δέντρα απόφασης αναπαριστούν μία συγκεκριμένη λογική έκφραση;

7

ΔΕΝΤΡΑ ΑΠΟΦΑΣΗΣ - ΕΚΦΡΑΣΤΙΚΟΤΗΤΑ

Λογική συνάρτηση $f(A, B, C) = (A \wedge B) \vee (\neg A \wedge C)$

Δέντρο απόφασης 1

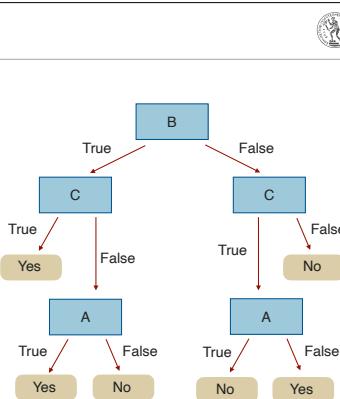
Δέντρο απόφασης 2

8

ΔΕΝΤΡΑ ΑΠΟΦΑΣΗΣ - ΠΡΩΤΑ ΣΥΜΠΕΡΑΣΜΑΤΑ

Παρατηρήσεις

- Τα δέντρα απόφασης μπορούν να γίνουν πολύ μεγάλα σε μέγεθος (να έχουν εκθετικά πολλούς κόμβους σε σχέση με τα χαρακτηριστικά των στιγμιοτύπων)
- Τα δέντρα απόφασης είναι κατανοητά από τους ανθρώπους (interpretable), όταν είναι μικρά σε μέγεθος (ταξινομητές με χρήση υπερκύβων)
- Η απλούστερη συνεπής επερήφανη είναι η βέλτιστη (Ockham's Razor) It is vain to do more what can be done with less... Entities should not be multiplied beyond necessity (William of Ocham - 1324) - Είναι οπαντικό να κατασκευάζουμε και να χρησιμοποιούμε απλά δέντρα απόφασης
- Το πρόβλημα εύρεσης του βέλτιστου δέντρου απόφασης είναι δυσεπίλυτο (Laurent Hyafil, Ronald L. Rivest, Constructing optimal binary decision trees is NP-complete, Information Processing Letters, Volume 5, Issue 1, May 1976, Pages 15-17) (Hancock T. R., Jiang T., Li M., and Tromp J., Lower bounds on learning decision lists and trees, Information and Computation 126(2):114–122, 1996)
- Ζητήματα που θα μας απασχολήσουν: αξιολόγηση, επιλογή χαρακτηριστικών, κλάδεμα, αλγόριθμοι εκπαίδευσης



Rokach, L. and Maimon, O.Z. (2008) Data Mining with Decision Trees: Theory and Applications. World Scientific Publishing Co., Inc., Singapore

9

ΕΚΜΑΘΗΣΗ ΔΕΝΤΡΩΝ ΑΠΟΦΑΣΗΣ - ΑΞΙΟΛΟΓΗΣΗ

Δεδομένα

X : σύνολο στιγμιοτύπων

Y : σύνολο ετικετών

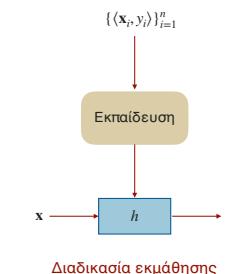
$x \sim \mathcal{D}(X)$

$f: X \rightarrow Y$: ιδανικός ταξινομητής

Σύνολο δεδομένων $\mathbb{D} = \{\langle x_i, y_i \rangle\}_{i=1}^n = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$

$x_i \sim \mathcal{D}(X)$

$H = \{h | h: X \rightarrow Y\}$: σύνολο δέντρων απόφασης $X \rightarrow Y$



10

Πρόβλημα

$h \leftarrow \text{DecisionTree.train}(\{\langle x_i, y_i \rangle\}_{i=1}^n)$

$\text{minimal}(h)$

- Με την διαδικασία εκμάθησης κατασκευάζεται ένα δέντρο απόφασης που προσεγγίζει τον ιδανικό ταξινομητή

- Παρότι είναι υπολογιστικά δύσκολο, θα πρέπει το δέντρο απόφασης να είναι όσο πο κοντά γίνεται στο ελάχιστο

ΕΚΜΑΘΗΣΗ ΔΕΝΤΡΩΝ ΑΠΟΦΑΣΗΣ - ΣΦΆΛΜΑ ΕΚΜΑΘΗΣΗΣ (ΑΚΡΙΒΕΙΑ)

Δεδομένα

X : σύνολο στιγμιοτύπων

Y : σύνολο ετικετών

$x \sim \mathcal{D}(X)$

$f: X \rightarrow Y$: ιδανικός ταξινομητής

Σύνολο δεδομένων $\mathbb{D} = \{\langle x_i, y_i \rangle\}_{i=1}^n = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$

$x_i \sim \mathcal{D}(X)$

$H = \{h | h: X \rightarrow Y\}$: σύνολο δέντρων απόφασης $X \rightarrow Y$

Πρόβλημα

$h \leftarrow \text{DecisionTree.train}(\{\langle x_i, y_i \rangle\}_{i=1}^n)$

Δέντρο απόφασης $h \approx f$

$\text{minimal}(h)$

$$\Sigma\text{φάλμα (error)} = L_f(h) = \frac{| \{x \in X : h(x) \neq f(x) \} |}{| X |}$$

Άγνωστο το f

Άγνωστα τα f, \mathcal{D}

$$L_{(\mathcal{D}, f)}(h) = \Pr_{x \sim \mathcal{D}(X)}(h(x) \neq f(x))$$

Η πιθανότητα να επιλέξω ένα τυχαίο δείγμα $x \in X$ για το οποίο $h(x) \neq f(x)$

Εμπειρικό σφάλμα - σφάλμα εκμάθησης
(empirical error - training error)

$$L_{\mathbb{D}}(h) = \frac{| \{i \in \mathbb{N}_n : h(x_i) \neq y_i \} |}{| n |}$$

ΕΠΙΠΛΕΟΝ ΜΕΤΡΑ ΑΞΙΟΛΟΓΗΣΗΣ ΤΑΞΙΝΟΜΗΤΩΝ

Δεδομένα

X : σύνολο στιγμιοτύπων

Y : σύνολο ετικετών

$x \sim \mathcal{D}(X)$

$f: X \rightarrow Y$: ιδανικός ταξινομητής

Σύνολο δεδομένων $\mathbb{D} = \{\langle x_i, y_i \rangle\}_{i=1}^n = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$

$x_i \sim \mathcal{D}(X)$

$H = \{h | h: X \rightarrow Y\}$: σύνολο δέντρων απόφασης $X \rightarrow Y$

Πρόβλημα

$h \leftarrow \text{DecisionTree.train}(\{\langle x_i, y_i \rangle\}_{i=1}^n)$

Δέντρο απόφασης $h \approx f$

$\text{minimal}(h)$

Μέθοδος απόκρυψης (hold-out)

Διαχωρισμός του \mathbb{D} σε δύο υποσύνολα δεδομένων:

$\mathbb{D}_L \subset \mathbb{D}$: δεδομένα εκμάθησης

$\mathbb{D}_T \subset \mathbb{D}$: δεδομένα ελέγχου

Συνήθως: $\mathbb{D}_L \cup \mathbb{D}_T = \mathbb{D}$ και $\mathbb{D}_L > \mathbb{D}_T$

$$L_{\mathbb{D}_L}(h) = \frac{| \{i \in \mathbb{N}_k : h(x_i) \neq y_i \} |}{| k |}$$

$$L_{\mathbb{D}_T}(h) = \frac{| \{i \in \mathbb{N}_m : h(x_i) \neq y_i \} |}{| m |}$$

Σφάλμα εκμάθησης (training error)

Σφάλμα ελέγχου (testing error)

11



12



Επιπλέον μέτρα αξιολόγησης

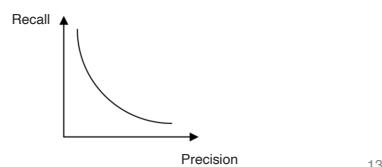
Συνήθως οι κλάσεις $\{+1, -1\}$ είναι σημαντικά διαφορετικές

Δηλαδή είναι διαφορετικό αν $h(x) \neq y$ ($y = +1$) ή αν $h(x) \neq y$ ($y = -1$)

$$\text{positive} = |y_i|, y_i = +1 \quad \text{negative} = |y_i|, y_i = -1 \quad \text{true-positive} = |h(x_i) = +1|, y_i = +1 \quad \text{true-negative} = |h(x_i) = -1|, y_i = -1 \\ \text{false-positive} = |h(x_i) = +1|, y_i = -1 \quad \text{false-negative} = |h(x_i) = -1|, y_i = +1$$

$$\text{Sensitivity} = \frac{\text{true-positive}}{\text{positive}} \quad (\text{λέγεται και Recall}) \quad \text{Specificity} = \frac{\text{true-negative}}{\text{negative}}$$

$$\text{Accuracy} = \text{Sensitivity} \frac{\text{positive}}{\text{positive} + \text{negative}} + \text{Specificity} \frac{\text{negative}}{\text{positive} + \text{negative}}$$



13

DesicionTree . train(\mathbb{D})

1. Επίλεξε ένα χαρακτηριστικό εισόδου a που παίρνει διαφορετικές τιμές στο \mathbb{D}
2. Φτιάξε ένα νέο κόμβο A και όρισε το a ως χαρακτηριστικό απόφασης
3. Για κάθε διαφορετική τιμή του a :

 - 3.1 Φτιάξε ένα νέο κόμβο ως παιδί του τρέχοντος κόμβου
 - 3.2 Όρισε το \mathbb{D}' ως το υποσύνολο του \mathbb{D} με τα στοιχεία που έχουν τη τιμή αυτή για το a
 - 3.3 Αν όλα τα $y \in \mathbb{D}'$ έχουν την ίδια ετικέτα, τότε κάνε τον A φύλλο με τιμή την ετικέτα αυτή
Αλλιώς DesicionTree . train(\mathbb{D}')

14



Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	FALSE	No
Sunny	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Rainy	Mild	High	FALSE	Yes
Rainy	Cool	Normal	FALSE	Yes
Rainy	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Sunny	Mild	High	FALSE	No
Sunny	Cool	Normal	TRUE	Yes
Rainy	Mild	Normal	FALSE	Yes
Sunny	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Mild	High	TRUE	No

DesicionTree . train(\mathbb{D})

1. Επίλεξε ένα χαρακτηριστικό εισόδου a που παίρνει διαφορετικές τιμές στο \mathbb{D}
2. Φτιάξε ένα νέο κόμβο A και όρισε το a ως χαρακτηριστικό απόφασης
3. Για κάθε διαφορετική τιμή του a :

 - 3.1 Φτιάξε ένα νέο κόμβο ως παιδί του τρέχοντος κόμβου
 - 3.2 Όρισε το \mathbb{D}' ως το υποσύνολο του \mathbb{D} με τα στοιχεία που έχουν τη τιμή αυτή για το a
 - 3.3 Αν όλα τα $y \in \mathbb{D}'$ έχουν την ίδια ετικέτα, τότε κάνε τον A φύλλο με τιμή την ετικέτα αυτή
Αλλιώς DesicionTree . train(\mathbb{D}')

15



Outlook	Temperature	Humidity	Windy	PlayTennis
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes

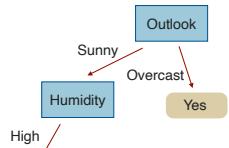
DesicionTree . train(\mathbb{D})

1. Επίλεξε ένα χαρακτηριστικό εισόδου a που παίρνει διαφορετικές τιμές στο \mathbb{D}
2. Φτιάξε ένα νέο κόμβο A και όρισε το a ως χαρακτηριστικό απόφασης
3. Για κάθε διαφορετική τιμή του a :

 - 3.1 Φτιάξε ένα νέο κόμβο ως παιδί του τρέχοντος κόμβου
 - 3.2 Όρισε το \mathbb{D}' ως το υποσύνολο του \mathbb{D} με τα στοιχεία που έχουν τη τιμή αυτή για το a
 - 3.3 Αν όλα τα $y \in \mathbb{D}'$ έχουν την ίδια ετικέτα, τότε κάνε τον A φύλλο με τιμή την ετικέτα αυτή
Αλλιώς DesicionTree . train(\mathbb{D}')

16

ΕΚΜΑΘΗΣΗ ΔΕΝΤΡΩΝ ΑΠΟΦΑΣΗΣ - ΑΦΕΛΗΣ ΑΛΓΟΡΙΘΜΟΣ



Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	FALSE	No
Sunny	Hot	High	TRUE	No
Sunny	Mild	High	FALSE	No
Sunny	Cool	Normal	TRUE	Yes
Sunny	Mild	Normal	TRUE	Yes

DesicionTree . train(\mathbb{D})

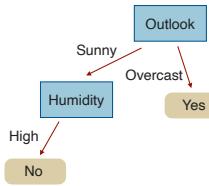
1. Επίλεξε ένα χαρακτηριστικό εισόδου a που παίρνει διαφορετικές τιμές στο \mathbb{D}
2. Φτιάξε ένα νέο κόμβο A και όρισε το a ως χαρακτηριστικό απόφασης
3. Για κάθε διαφορετική τιμή του a :

 - 3.1 Φτιάξε ένα νέο κόμβο ως παιδί του τρέχοντος κόμβου
 - 3.2 Όρισε το D' ως το υποσύνολο του \mathbb{D} με τα στοιχεία που έχουν τη τιμή αυτή για το a
 - 3.3 Αν όλα τα $y \in D'$ έχουν την ίδια ετικέτα, τότε κάνε τον A φύλλο με τιμή την ετικέτα αυτή

Αλλιώς DecisionTree . train(\mathbb{D}')

17

ΕΚΜΑΘΗΣΗ ΔΕΝΤΡΩΝ ΑΠΟΦΑΣΗΣ - ΑΦΕΛΗΣ ΑΛΓΟΡΙΘΜΟΣ



Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	FALSE	No
Sunny	Hot	High	TRUE	No
Sunny	Mild	High	FALSE	No

DesicionTree . train(\mathbb{D})

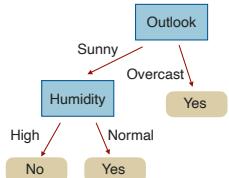
1. Επίλεξε ένα χαρακτηριστικό εισόδου a που παίρνει διαφορετικές τιμές στο \mathbb{D}
2. Φτιάξε ένα νέο κόμβο A και όρισε το a ως χαρακτηριστικό απόφασης
3. Για κάθε διαφορετική τιμή του a :

 - 3.1 Φτιάξε ένα νέο κόμβο ως παιδί του τρέχοντος κόμβου
 - 3.2 Όρισε το D' ως το υποσύνολο του \mathbb{D} με τα στοιχεία που έχουν τη τιμή αυτή για το a
 - 3.3 Αν όλα τα $y \in D'$ έχουν την ίδια ετικέτα, τότε κάνε τον A φύλλο με τιμή την ετικέτα αυτή

Αλλιώς DecisionTree . train(\mathbb{D}')

18

ΕΚΜΑΘΗΣΗ ΔΕΝΤΡΩΝ ΑΠΟΦΑΣΗΣ - ΑΦΕΛΗΣ ΑΛΓΟΡΙΘΜΟΣ



Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Cool	Normal	TRUE	Yes
Sunny	Mild	Normal	TRUE	Yes

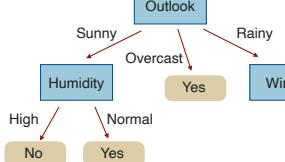
DesicionTree . train(\mathbb{D})

1. Επίλεξε ένα χαρακτηριστικό εισόδου a που παίρνει διαφορετικές τιμές στο \mathbb{D}
2. Φτιάξε ένα νέο κόμβο A και όρισε το a ως χαρακτηριστικό απόφασης
3. Για κάθε διαφορετική τιμή του a :

 - 3.1 Φτιάξε ένα νέο κόμβο ως παιδί του τρέχοντος κόμβου
 - 3.2 Όρισε το D' ως το υποσύνολο του \mathbb{D} με τα στοιχεία που έχουν τη τιμή αυτή για το a
 - 3.3 Αν όλα τα $y \in D'$ έχουν την ίδια ετικέτα, τότε κάνε τον A φύλλο με τιμή την ετικέτα αυτή

Αλλιώς DecisionTree . train(\mathbb{D}')

ΕΚΜΑΘΗΣΗ ΔΕΝΤΡΩΝ ΑΠΟΦΑΣΗΣ - ΑΦΕΛΗΣ ΑΛΓΟΡΙΘΜΟΣ



Outlook	Temperature	Humidity	Windy	PlayTennis
Rainy	Mild	High	FALSE	Yes
Rainy	Cool	Normal	FALSE	Yes
Rainy	Cool	Normal	TRUE	No
Rainy	Mild	Normal	FALSE	Yes
Rainy	Mild	High	TRUE	No

DesicionTree . train(\mathbb{D})

1. Επίλεξε ένα χαρακτηριστικό εισόδου a που παίρνει διαφορετικές τιμές στο \mathbb{D}
2. Φτιάξε ένα νέο κόμβο A και όρισε το a ως χαρακτηριστικό απόφασης
3. Για κάθε διαφορετική τιμή του a :

 - 3.1 Φτιάξε ένα νέο κόμβο ως παιδί του τρέχοντος κόμβου
 - 3.2 Όρισε το D' ως το υποσύνολο του \mathbb{D} με τα στοιχεία που έχουν τη τιμή αυτή για το a
 - 3.3 Αν όλα τα $y \in D'$ έχουν την ίδια ετικέτα, τότε κάνε τον A φύλλο με τιμή την ετικέτα αυτή

Αλλιώς DecisionTree . train(\mathbb{D}')

20

ΕΚΜΑΘΗΣΗ ΔΕΝΤΡΩΝ ΑΠΟΦΑΣΗΣ - ΑΦΕΛΗΣ ΑΛΓΟΡΙΘΜΟΣ

Outlook	Temperature	Humidity	Windy	PlayTennis
Rainy	Cool	Normal	TRUE	No
Rainy	Mild	High	TRUE	No

DesicionTree . train(D)

- Επίλεξε ένα χαρακτηριστικό εισόδου a που παίρνει διαφορετικές τιμές στο D
- Φτιάξε ένα νέο κόμβο A και όρισε το a ως χαρακτηριστικό απόφασης
- Για κάθε διαφορετική τιμή του a :
 - Φτιάξε ένα νέο κόμβο a παιδί του τρέχοντος κόμβου
 - Όρισε το D' ως το υποσύνολο του D με τα στοιχεία που έχουν τη τιμή αυτή για το a
 - Αν όλα τα $y \in D'$ έχουν την ίδια ετικέτα, τότε κάνε τον A φύλλο με τιμή την ετικέτα αυτή

Αλλιώς DecisionTree . train(D')

ΕΚΜΑΘΗΣΗ ΔΕΝΤΡΩΝ ΑΠΟΦΑΣΗΣ - ΑΦΕΛΗΣ ΑΛΓΟΡΙΘΜΟΣ

Outlook	Temperature	Humidity	Windy	PlayTennis
Rainy	Mild	High	FALSE	Yes
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	FALSE	Yes

DesicionTree . train(D)

- Επίλεξε ένα χαρακτηριστικό εισόδου a που παίρνει διαφορετικές τιμές στο D
- Φτιάξε ένα νέο κόμβο A και όρισε το a ως χαρακτηριστικό απόφασης
- Για κάθε διαφορετική τιμή του a :
 - Φτιάξε ένα νέο κόμβο a παιδί του τρέχοντος κόμβου
 - Όρισε το D' ως το υποσύνολο του D με τα στοιχεία που έχουν τη τιμή αυτή για το a
 - Αν όλα τα $y \in D'$ έχουν την ίδια ετικέτα, τότε κάνε τον A φύλλο με τιμή την ετικέτα αυτή

Αλλιώς DecisionTree . train(D')

ΕΚΜΑΘΗΣΗ ΔΕΝΤΡΩΝ ΑΠΟΦΑΣΗΣ - ΑΦΕΛΗΣ ΑΛΓΟΡΙΘΜΟΣ

DesicionTree . train(D)

- Επίλεξε ένα χαρακτηριστικό εισόδου a που παίρνει διαφορετικές τιμές στο D
- Φτιάξε ένα νέο κόμβο A και όρισε το a ως χαρακτηριστικό απόφασης
- Για κάθε διαφορετική τιμή του a :
 - Φτιάξε ένα νέο κόμβο a παιδί του τρέχοντος κόμβου
 - Όρισε το D' ως το υποσύνολο του D με τα στοιχεία που έχουν τη τιμή αυτή για το a
 - Αν όλα τα $y \in D'$ έχουν την ίδια ετικέτα, τότε κάνε τον A φύλλο με τιμή την ετικέτα αυτή

Αλλιώς DecisionTree . train(D')

► Η επιλογή του χαρακτηριστικού εισόδου γίνεται με στόχο την αύξηση της πιθανότητας να οδηγηθούμε σε μικρότερο δέντρο

► Προφανώς, δεν μπορούμε να διασφαλίσουμε την κατασκευή του ελάχιστου δέντρου (υπενθυμίζουμε ότι το πρόβλημα είναι NP-complete)

ΔΕΝΤΡΑ ΑΠΟΦΑΣΗΣ - ΕΠΙΛΟΓΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΟΥ ΑΠΟΦΑΣΗΣ

Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	FALSE	No
Sunny	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Rainy	Mild	High	FALSE	Yes
Rainy	Cool	Normal	FALSE	Yes
Rainy	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Sunny	Mild	High	FALSE	No
Sunny	Cool	Normal	TRUE	Yes
Rainy	Mild	Normal	FALSE	Yes
Sunny	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Mild	High	TRUE	No

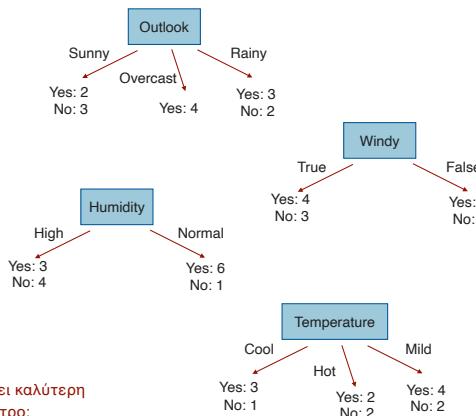
► Ποια από τις επιλογές χαρακτηριστικού έχει καλύτερη πιθανότητα να οδηγήσει σε μικρότερο δέντρο;

ΔΕΝΤΡΑ ΑΠΟΦΑΣΗΣ - ΕΠΙΛΟΓΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΟΥ ΑΠΟΦΑΣΗΣ



Μέθοδος

- Τυχαία: Επίλεξε ένα χαρακτηριστικό χωρίς κάποιο συγκεκριμένο κριτήριο
- Λιγότερες τιμές: Επίλεξε το χαρακτηριστικό με τη μικρότερη πληθικότητα του πεδίου τιμών
- Περισσότερες τιμές: Επίλεξε το χαρακτηριστικό με τη μεγαλύτερη πληθικότητα του πεδίου τιμών
- Μεγαλύτερο όφελος: Επίλεξε το χαρακτηριστικό με το μεγαλύτερο κέρδος πληροφορίας (information gain)



Ποια από τις επιλογές χαρακτηριστικού έχει καλύτερη πιθανότητα να οδηγήσει σε μικρότερο δέντρο;

25

ΔΕΝΤΡΑ ΑΠΟΦΑΣΗΣ - ΕΠΙΛΟΓΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΟΥ ΑΠΟΦΑΣΗΣ



Μέτρα μη-καθαρότητας (impurity measures)

Έστω μία τυχαία μεταβλητή x με k διακριτές τιμές, με κατανομή πιθανότητας $P = (p_1, p_2, \dots, p_k)$

Μέτρο μη-καθαρότητας της μεταβλητής x είναι μία συνάρτηση $\phi : [0,1]^k \rightarrow R$ που ικανοποιεί τις συνθήκες:

$$\phi(P) \geq 0$$

$$\phi(P) = 0 \text{ αν υπάρχει } i \in \mathbb{N}_k \text{ τέτοιο ώστε } p_i = 1$$

$$\phi(P) = 1 \text{ αν για κάθε } i \in \mathbb{N}_k \text{ ισχύει ότι } p_i = \frac{1}{k}$$

ϕ συμμετρική στα p_1, p_2, \dots, p_k

ϕ ομαλή

26

ΔΕΝΤΡΑ ΑΠΟΦΑΣΗΣ - ΕΠΙΛΟΓΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΟΥ ΑΠΟΦΑΣΗΣ



Gini index

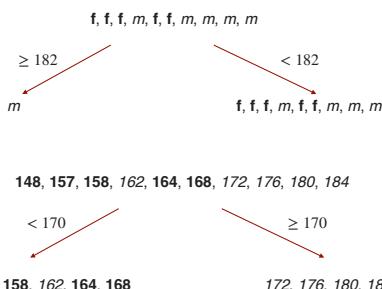
- Ταξινόμιω σε male, female ανάλογα με το ύψος
- Έστω ότι γνωρίζω την κατανομή (m,f) σε ένα σύνολο
- Επιλέγω κάποιο στοιχείο και το ταξινόμιω με βάση την κατανομή
- Ποια είναι η πιθανότητα να το ταξινομήσω λάθος;

148, 157, 158, 162, 164, 168, 172, 176, 180, 184

184

148, 157, 158, 162, 164, 168, 172, 176, 180, 184

148
157, 158, 162, 164, 168, 172, 176, 180, 184



27

ΔΕΝΤΡΑ ΑΠΟΦΑΣΗΣ - ΕΠΙΛΟΓΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΟΥ ΑΠΟΦΑΣΗΣ - GINI INDEX



Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	FALSE	No
Sunny	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Rainy	Mild	High	FALSE	Yes
Rainy	Cool	Normal	FALSE	Yes
Rainy	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Sunny	Mild	High	FALSE	No
Sunny	Cool	Normal	TRUE	Yes
Rainy	Mild	Normal	FALSE	Yes
Sunny	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Mild	High	TRUE	No

- Πόσο συχνά ένα τυχαίο δείγμα που επιλέγεται ταξινομείται λανθασμένα, αν του αποδοθεί μία τυχαία ετικέτα;

$$\text{gini}(\text{Root}) = 1 - \left(\frac{|\text{PlayTennis} = \text{Yes}|}{|\text{Root}|} \right)^2 - \left(\frac{|\text{PlayTennis} = \text{No}|}{|\text{Root}|} \right)^2$$

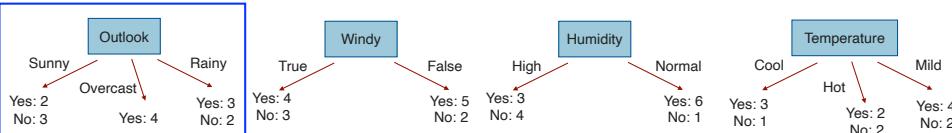
$$\text{gini}(\text{D}) = \sum_{i \in \text{labels}(\text{D})} p_i(1 - p_i) = 1 - \sum_{i \in \text{labels}(\text{D})} p_i^2$$

$$= 1 - \left(\frac{9}{14} \right)^2 - \left(\frac{5}{14} \right)^2$$

28



ΔΕΝΤΡΑ ΑΠΟΦΑΣΗΣ - ΕΠΙΛΟΓΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΟΥ ΑΠΟΦΑΣΗΣ - GINI INDEX



Information Gain: $ig(\text{Outlook}) = \text{gini}(\text{Root}) - \text{gini}(\text{Outlook}) = \text{gini}(\text{Root}) - \sum_{v \in \text{values}(\text{Outlook})} \frac{ \text{Outlook} = v }{ \text{Root} } \text{gini}(\text{Outlook} = v)$																																																																												
$\text{gini}(\text{Root}) = 1 - \left(\frac{9}{14} \right)^2 - \left(\frac{5}{14} \right)^2$																																																																												
$\text{gini}(\text{Outlook}) = \frac{5}{14} \text{gini}(\text{Outlook} = \text{Sunny}) + \frac{4}{14} \text{gini}(\text{Outlook} = \text{Overcast}) + \frac{5}{14} \text{gini}(\text{Outlook} = \text{Rainy})$																																																																												
$\text{gini}(\text{Outlook} = \text{Sunny}) = 1 - \left(\frac{3}{5} \right)^2 - \left(\frac{2}{5} \right)^2$																																																																												
$\text{gini}(\text{Outlook} = \text{Overcast}) = 1 - \left(\frac{4}{4} \right)^2 - \left(\frac{0}{4} \right)^2$																																																																												
$\text{gini}(\text{Outlook} = \text{Rainy}) = 1 - \left(\frac{3}{5} \right)^2 - \left(\frac{2}{5} \right)^2$																																																																												
<table border="1"> <thead> <tr> <th>Outlook</th> <th>Temperature</th> <th>Humidity</th> <th>Windy</th> <th>PlayTennis</th> </tr> </thead> <tbody> <tr> <td>Sunny</td> <td>Hot</td> <td>High</td> <td>FALSE</td> <td>No</td> </tr> <tr> <td>Sunny</td> <td>Hot</td> <td>High</td> <td>TRUE</td> <td>No</td> </tr> <tr> <td>Overcast</td> <td>Hot</td> <td>High</td> <td>FALSE</td> <td>Yes</td> </tr> <tr> <td>Rainy</td> <td>Mild</td> <td>High</td> <td>FALSE</td> <td>Yes</td> </tr> <tr> <td>Rainy</td> <td>Cool</td> <td>Normal</td> <td>FALSE</td> <td>Yes</td> </tr> <tr> <td>Rainy</td> <td>Cool</td> <td>Normal</td> <td>TRUE</td> <td>No</td> </tr> <tr> <td>Overcast</td> <td>Cool</td> <td>Normal</td> <td>TRUE</td> <td>Yes</td> </tr> <tr> <td>Sunny</td> <td>Mild</td> <td>High</td> <td>FALSE</td> <td>No</td> </tr> <tr> <td>Sunny</td> <td>Cool</td> <td>Normal</td> <td>TRUE</td> <td>Yes</td> </tr> <tr> <td>Rainy</td> <td>Mild</td> <td>Normal</td> <td>FALSE</td> <td>Yes</td> </tr> <tr> <td>Sunny</td> <td>Mild</td> <td>Normal</td> <td>TRUE</td> <td>Yes</td> </tr> <tr> <td>Overcast</td> <td>Mild</td> <td>High</td> <td>TRUE</td> <td>Yes</td> </tr> <tr> <td>Overcast</td> <td>Hot</td> <td>Normal</td> <td>FALSE</td> <td>Yes</td> </tr> <tr> <td>Rainy</td> <td>Mild</td> <td>High</td> <td>TRUE</td> <td>No</td> </tr> </tbody> </table>		Outlook	Temperature	Humidity	Windy	PlayTennis	Sunny	Hot	High	FALSE	No	Sunny	Hot	High	TRUE	No	Overcast	Hot	High	FALSE	Yes	Rainy	Mild	High	FALSE	Yes	Rainy	Cool	Normal	FALSE	Yes	Rainy	Cool	Normal	TRUE	No	Overcast	Cool	Normal	TRUE	Yes	Sunny	Mild	High	FALSE	No	Sunny	Cool	Normal	TRUE	Yes	Rainy	Mild	Normal	FALSE	Yes	Sunny	Mild	Normal	TRUE	Yes	Overcast	Mild	High	TRUE	Yes	Overcast	Hot	Normal	FALSE	Yes	Rainy	Mild	High	TRUE	No
Outlook	Temperature	Humidity	Windy	PlayTennis																																																																								
Sunny	Hot	High	FALSE	No																																																																								
Sunny	Hot	High	TRUE	No																																																																								
Overcast	Hot	High	FALSE	Yes																																																																								
Rainy	Mild	High	FALSE	Yes																																																																								
Rainy	Cool	Normal	FALSE	Yes																																																																								
Rainy	Cool	Normal	TRUE	No																																																																								
Overcast	Cool	Normal	TRUE	Yes																																																																								
Sunny	Mild	High	FALSE	No																																																																								
Sunny	Cool	Normal	TRUE	Yes																																																																								
Rainy	Mild	Normal	FALSE	Yes																																																																								
Sunny	Mild	Normal	TRUE	Yes																																																																								
Overcast	Mild	High	TRUE	Yes																																																																								
Overcast	Hot	Normal	FALSE	Yes																																																																								
Rainy	Mild	High	TRUE	No																																																																								

Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	FALSE	No
Sunny	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Rainy	Mild	High	FALSE	Yes
Rainy	Cool	Normal	FALSE	Yes
Rainy	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Sunny	Mild	High	FALSE	No
Sunny	Cool	Normal	TRUE	Yes
Rainy	Mild	Normal	FALSE	Yes
Sunny	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Mild	High	TRUE	No

ΕΚΜΑΘΗΣΗ ΔΕΝΤΡΩΝ ΑΠΟΦΑΣΗΣ - ΑΛΓΟΡΙΘΜΟΣ CART

DesicionTree . CART(D)

1. Επίλεξε το χαρακτηριστικό εισόδου a με το μεγαλύτερο κέρδος πληροφορίας στο D με βάση το gini
 2. Φτιάξε ένα νέο κόμβο A και όρισε το a ως χαρακτηριστικό απόφασης
 3. Για κάθε διαφορετική τιμή του a :

3.1 Φτιάξε ένα νέο κόμβο ως παιδί του τρέχοντος κόμβου

3.2 Όρισε το D' ως το υποσύνολο του D με τα στοιχεία που έχουν τη τιμή αυτή για το x .

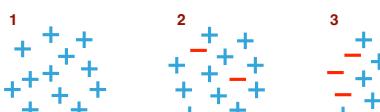
3.3 Αν όλα τα $y \in D'$ έχουν την ίδια ετικέτα, τότε κάνε τον A φύλλο με τιμή την ετικέτα αυτή.

Αλλιώς DecisionTree . CART(▷

Leo Breiman, Jerome Friedman, Charles J. Stone, R.A. Olshen, Classification and Regression Trees
Chapman and Hall/CRC (1984)

30

ΔΕΝΤΡΑ ΑΠΟΦΑΣΗΣ - ΕΠΙΛΟΓΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΟΥ ΑΠΟΦΑΣΗΣ - ΕΝΤΡΟΠΙΑ



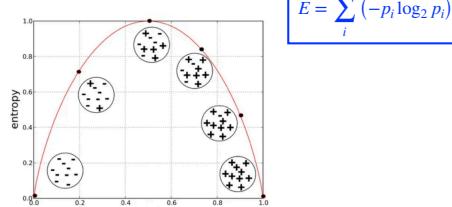
Πόση πληροφορία λείπει

- ▶ Ποια από τις κατανομές έχει μεγαλύτερο κέρδος πληροφορίας;

$$E_1 = -1 \log_2 1 - 0 \log_2 0 =$$

$$E_2 = -0.133 \log_2 0.133 - 0.87 \log_2 0.87 \simeq 0.565$$

$$E_2 \equiv -0.5 \log_2 0.5 = 0.5 \log_2 0.5 \equiv 1$$

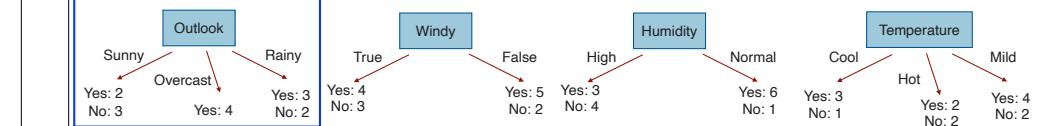


Επιλέγουμε το χαρακτηριστικό που γνωρίζοντας την τιμή του πετυχαίνουμε τη μεναδύτεσσον μείωση της εντοπίσας

Provost, Foster; Fawcett, Tom. Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking

3

ΔΕΝΤΡΑ ΑΠΟΦΑΣΗΣ - ΕΠΙΛΟΓΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΟΥ ΑΠΟΦΑΣΗΣ - ΕΝΤΡΟΠΙΑ



$$\text{Information Gain: } \text{ig(Outlook)} = E(\text{Root}) - E(\text{Outlook}) = E(\text{Root}) - \sum_{v \in \text{values(Outlook)}} \frac{|\text{Outlook} = v|}{|\text{Root}|} E(\text{Outlook} = v)$$

$$E(\text{Outlook}) = \frac{5}{14}E(\text{Outlook} = \text{Sunny}) + \frac{4}{14}E(\text{Outlook} = \text{Overcast})$$

$$E(\text{Outlook} = \text{Sunny}) = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2$$

$$E(\text{Outlook} = \text{Overcast}) = -\frac{4}{4} \log_2 \frac{4}{4} - \frac{0}{4} \log_2$$

Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	FALSE	No
Sunny	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Rainy	Mild	High	FALSE	Yes
Rainy	Cool	Normal	FALSE	Yes
Rainy	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Sunny	Mild	High	FALSE	No
Sunny	Cool	Normal	TRUE	Yes
Rainy	Mild	Normal	FALSE	Yes
Sunny	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Mild	High	TRUE	No

32

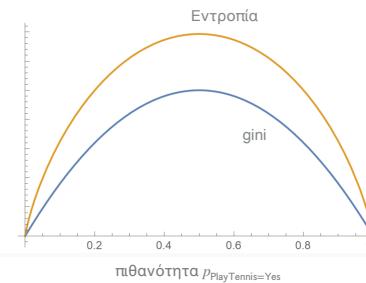


DesicionTree . ID3(ℳ)

1. Επίλεξε το χαρακτηριστικό εισόδου a με το μεγαλύτερο κέρδος πληροφορίας στο \mathbb{D} με βάση την εντροπία
 2. Φτιάξε ένα νέο κόμβο A και όρισε το a ως χαρακτηριστικό απόφασης
 3. Για κάθε διαφορετική τιμή του a :
 - 3.1 Φτιάξε ένα νέο κόμβο ως παιδί του τρέχοντος κόμβου
 - 3.2 Όρισε το \mathbb{D}' ως το υποσύνολο του \mathbb{D} με τα στοιχεία που έχουν τη τιμή αυτή για το a
 - 3.3 Αν όλα τα $y \in \mathbb{D}'$ έχουν την ίδια ετικέτα, τότε κάνε τον A φύλλο με τιμή την ετικέτα αυτή
- Αλλιώς DecisionTree . ID3(ℳ')

Quinlan, J. R. 1986. Induction of Decision Trees. *Mach. Learn.* 1, 1 (Mar. 1986), 81–106

33



- ▶ Παρόμοια αποτελέσματα στην πράξη
- ▶ διαφωνούν σε ελάχιστες περιπτώσεις στην επιλογή χαρακτηριστικού
- ▶ Δυσκολότερος ο υπολογισμός για την εντροπία

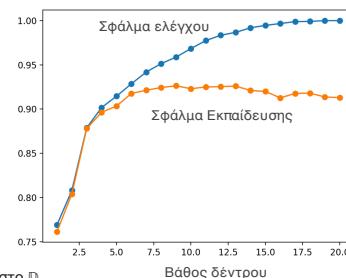
Laura Elena Raileanu and Kilian Stoffel, Theoretical comparison between the Gini Index and Information Gain criteria, *Annals of Mathematics and Artificial Intelligence* 41: 77–93, 2004

34



Υπερπροσαρμογή (overfitting)

- ▶ Τα δένδρα απόφασης μπορούν να ταξινομήσουν όλα τα δεδομένα εκμάθησης χωρίς σφάλμα
- ▶ στην περίπτωση αυτή θα καταλήξουμε με μεγάλα δένδρα απόφασης που θα έχουν δυσκολία γενίκευσης (σφάλμα στα δεδομένα ελέγχου)
- ▶ Απαιτείται συστηματική απλοποίηση του δένδρου



DesicionTree . train(ℳ)

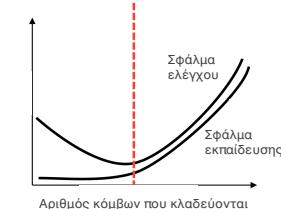
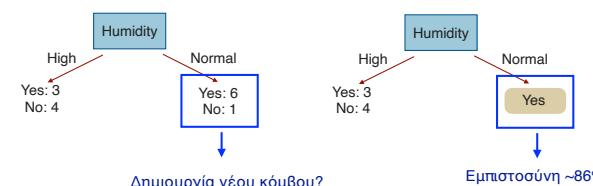
1. Επίλεξε ένα χαρακτηριστικό εισόδου a που παίρνει διαφορετικές τιμές στο \mathbb{D}
 2. Φτιάξε ένα νέο κόμβο A και όρισε το a ως χαρακτηριστικό απόφασης
 3. Για κάθε διαφορετική τιμή του a :
 - 3.1 Φτιάξε ένα νέο κόμβο ως παιδί του τρέχοντος κόμβου
 - 3.2 Όρισε το \mathbb{D}' ως το υποσύνολο του \mathbb{D} με τα στοιχεία που έχουν τη τιμή αυτή για το a
 - 3.3 Αν όλα τα $y \in \mathbb{D}'$ έχουν την ίδια ετικέτα τότε κάνε τον A φύλλο με τιμή την ετικέτα αυτή
- Αλλιώς DecisionTree . train(ℳ')

35



DesicionTree . train(ℳ)

1. Επίλεξε ένα χαρακτηριστικό εισόδου a που παίρνει διαφορετικές τιμές στο \mathbb{D}
 2. Φτιάξε ένα νέο κόμβο A και όρισε το a ως χαρακτηριστικό απόφασης
 3. Για κάθε διαφορετική τιμή του a :
 - 3.1 Φτιάξε ένα νέο κόμβο ως παιδί του τρέχοντος κόμβου
 - 3.2 Όρισε το \mathbb{D}' ως το υποσύνολο του \mathbb{D} με τα στοιχεία που έχουν τη τιμή αυτή για το a
 - 3.3 Αν όλα τα $y \in \mathbb{D}'$ έχουν την ίδια ετικέτα τότε κάνε τον A φύλλο με τιμή την ετικέτα αυτή
- Αλλιώς DecisionTree . train(ℳ')



36



Reduced error pruning (REP)

- ▶ Ορίζουμε ένα pruning set (μέρος του dataset)
- ▶ Κατασκευάζουμε το δέντρο απόφασης
- ▶ Εξετάζουμε όλους τους εσωτερικούς κόμβους από τα φύλλα προς τη ρίζα
- ▶ Ελέγχουμε τη μετατροπή του κόμβου σε φύλλο με επικέτα την πιο πιθανή κλάση
- ▶ Αν επηρεάζεται η επίδοση του ταξινομητή στο pruning set συνεχίζουμε
- ▶ Άλλιώς, μετατρέπουμε τον κόμβο σε φύλλο
- ▶ Συνεχίζουμε μέχρι να μην υπάρχει κόμβος που η μετατροπή του να μην επηρεάζει την επίδοση του δέντρου στο pruning set

Quinlan, J. R. 1987. Simplifying Decision Trees. International Journal of Human-Computer Studies, Volume 21, Issue 2, August 1999, Pages 497-510

37

Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	FALSE	No
Sunny	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Rainy	Mild	High	FALSE	Yes
Rainy	Cool	Normal	FALSE	Yes
Rainy	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Sunny	Mild	High	FALSE	No
Sunny	Cool	Normal	TRUE	Yes
Rainy	Mild	Normal	FALSE	Yes
Sunny	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Mild	High	TRUE	No

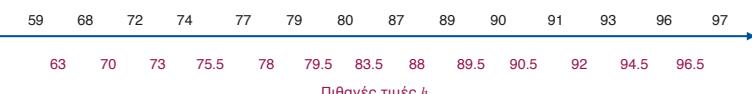
Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	90	FALSE	No
Sunny	Hot	87	TRUE	No
Overcast	Hot	93	FALSE	Yes
Rainy	Mild	89	FALSE	Yes
Rainy	Cool	79	FALSE	Yes
Rainy	Cool	59	TRUE	No
Overcast	Cool	77	TRUE	Yes
Sunny	Mild	91	FALSE	No
Sunny	Cool	68	TRUE	Yes
Rainy	Mild	80	FALSE	Yes
Sunny	Mild	72	TRUE	Yes
Overcast	Mild	96	TRUE	Yes
Overcast	Hot	74	FALSE	Yes
Rainy	Mild	97	TRUE	No



38

Επιλογή τιμής h_0

Ταξινομημένες τιμές humidity στο σύνολο δεδομένων



$$\text{Information Gain: } ig(h_0) = E(\text{Root}) - \frac{| \text{Humidity} \geq h_0 |}{| \text{Root} |} E(\text{Humidity} \geq h_0) - \frac{| \text{Humidity} < h_0 |}{| \text{Root} |} E(\text{Humidity} < h_0)$$

Βέλτιστη τιμή $h_0 = 83.5$ ($ig(83.5) = 0.94$)

39

Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	FALSE	No
Sunny	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Rainy	Mild	High	FALSE	Yes
Rainy	Cool	Normal	FALSE	Yes
Rainy	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Sunny	Mild	High	FALSE	No
Sunny	Cool	Normal	TRUE	Yes
Rainy	Mild	Normal	FALSE	Yes
Sunny	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Mild	High	TRUE	No

 $x_3^{\text{Outlook}} = \text{NULL}$

Διαχείριση άγνωστων τιμών κατά τη μάθηση

- ▶ Συμπληρώνεις με την πιθανότερη τιμή του χαρακτηριστικού $x_3^{\text{Outlook}} = \text{Sunny}$
- ▶ Συμπληρώνεις με την πιθανότερη τιμή του χαρακτηριστικού για τη συγκεκριμένη ετικέτα εξόδου $x_3^{\text{Outlook}} = \text{Overcast}$
- ▶ Προσθέτεις νέα στιγμιότυπα με όλες τις τιμές των χαρακτηριστικών
- ▶ Αγνοείς το συγκεκριμένο στιγμιότυπο όποτε εμπλέκεται στην επιλογή χαρακτηριστικού
- ▶ Κατασκευάζει δέντρα απόφασης για την πρόβλεψη της τιμής των άγνωστων τιμών

40

ΔΕΝΤΡΑ ΑΠΟΦΑΣΗΣ - ΑΓΝΩΣΤΕΣ ΤΙΜΕΣ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ

▶ Κατασκευάζεις δέντρα απόφασης για την πρόβλεψη της τιμής των άγνωστων τιμών

Δέντρο απόφασης h

Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	FALSE	No
Sunny	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Rainy	Mild	High	FALSE	Yes
	Cool	Normal	FALSE	Yes
Rainy	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Sunny	Mild	High	FALSE	No
Sunny	Cool	Normal	TRUE	Yes
Rainy	Mild	Normal	FALSE	Yes
Sunny	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Mild	High	TRUE	No

Temperature	Humidity	Windy	PlayTennis	Outlook
Hot	High	FALSE	No	Sunny
Hot	High	TRUE	No	Sunny
Hot	High	FALSE	Yes	Overcast
Mild	High	FALSE	Yes	Rainy
Cool	Normal	TRUE	No	Rainy
Cool	Normal	TRUE	Yes	Overcast
Mild	High	FALSE	No	Sunny
Cool	Normal	TRUE	Yes	Sunny
Mild	Normal	FALSE	Yes	Rainy
Mild	Normal	TRUE	Yes	Sunny
Mild	High	TRUE	Yes	Overcast
Hot	Normal	FALSE	Yes	Overcast
Mild	High	TRUE	No	Rainy

$h \rightarrow \text{Outlook} = ?$

41

ΔΕΝΤΡΑ ΑΠΟΦΑΣΗΣ - ΑΓΝΩΣΤΕΣ ΤΙΜΕΣ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ

Διαχείριση άγνωστων τιμών κατά την πρόβλεψη

▶ Ελέγχεις όλες τις διακλαδώσεις, δίνεις στην έξοδο την πιθανότερη τιμή φύλλου

Είσοδος $x_i = (_, \text{Hot}, \text{Normal}, \text{False})$

```

graph TD
    Outlook[Outlook] -- Sunny --> Humidity1[Humidity]
    Outlook -- Overcast --> Humidity2[Humidity]
    Outlook -- Rainy --> Humidity3[Humidity]
    Humidity1 -- High --> Leaf1[Yes]
    Humidity1 -- Normal --> Leaf2[No]
    Humidity2 -- True --> Leaf3[Yes]
    Humidity2 -- False --> Leaf4[No]
    
```

έξοδος $y = \langle \text{Yes} \rangle$

42

ΕΚΜΑΘΗΣΗ ΔΕΝΤΡΩΝ ΑΠΟΦΑΣΗΣ - ΑΛΓΟΡΙΘΜΟΣ C4.5

Βελτιώσεις στον αλγόριθμο ID3

- Χειρισμός ποιοτικών (κατηγορικών), διακριτών ποσοτικών και συνεχών ποσοτικών (αριθμητικών) χαρακτηριστικών
- Βρίσκεις τη βέλτιστη τιμή, με βάση την εντροπία
- Χειρισμός αγνώστων τιμών
- αγνοείς τις τιμές των χαρακτηριστικών κατά τη μάθηση
- εξετάζεις όλες τις διακλαδώσεις κατά την πρόβλεψη
- Κλάδεμα για ομαλοποίηση
- (Διαχείριση χαρακτηριστικών με διαφορετικά κόστη)
- Επιπλέον βελτιώσεις στον αλγόριθμο C5 (βελτιστοποίησεις σε ταχύτητα, μνήμη, μικρότερα δέντρα)

Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993

Quinlan, J. R. Improved use of continuous attributes in C4.5. Journal of Artificial Intelligence Research, 4:77-90, 1996

43

ΜΕΘΟΔΟΙ ENSEMBLE

Θεώρημα Condorcet's jury (προσαρμοσμένο)

Έστω $A = \{c_1, c_2, \dots, c_k\}$ ένα σύνολο από δυαδικούς ταξινομητές $p(c_i)$ η πιθανότητα να δώσει ο ταξινομητής $c_i, i \in \mathbb{N}_k$ σωστή πρόβλεψη M ο ταξινομητής που δίνει την πρόβλεψη της πλειοψηφίας των c_i

$\text{Av } p(c_i) = p > 0.5$ για κάθε $c_i, i \in \mathbb{N}_k$ τότε $p(M) > p$

$\text{Av } |A| \rightarrow \infty$ τότε $p(M) \rightarrow 1$

Nicolas de Condorcet (1743–1794). Application of Analysis to the Probability of Majority Decisions (Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix), 1785

44



Μπορούν όλα τα σύνολα ταξινομητών να οδηγήσουν σε καλούς πλειοψηφικούς ταξινομητές

► Κριτήρια

- ▶ **Ποικιλία απόψεων** — Κάθε ταξινομητής-μέλος θα πρέπει να έχει ιδιωτική πληροφορία, ακόμα κι αν αυτή είναι απλώς μια επικεντρική ερμηνεία των γνωστών γεγονότων
- ▶ **Ανεξαρτησία** — Οι απόψεις των ταξινομητών-μελών δεν καθορίζονται από τις απόψεις των άλλων
- ▶ **Διασπορά** — Τα μέλη εξειδικεύονται και εξάγουν συμπεράσματα με βάση μία τοπική γνώση
- ▶ **Σύνθεση απόψεων** — Πρέπει να καθοριστεί ένας έξυπνος μηχανισμός που συνδυάζει τις ατομικές κρίσεις και συνθέτει τη συλλογική απόφαση

45



Βασική ιδέα

- ▶ **Παρατήρηση:** Οι αλγόριθμοι εκμάθησης δέντρων αποφάσεων μπορούν να κατασκευάσουν δέντρα με ελαφρά διαφορετική δομή αλλά σημαντικά διαφορετικές προβλέψεις, ακόμα και για μικρές διαφορές στο σύνολο δεδομένων
- ▶ **Τεχνική:** Χρησιμοποιώντας **τυχήματα** του συνόλου δεδομένων, κατασκεύασε πολλά διαφορετικά δέντρα αποφάσεων και **συνδύασε** τις προβλέψεις τους

Τμηματοποίηση συνόλου δεδομένων

- ▶ **Bagging (Bootstrap aggregating):** Επιλέξει k διαφορετικά υποσύνολα του συνόλου δεδομένων
- ▶ **Feature Bagging (random suspace method):** Επιλέξει k διαφορετικά υποσύνολα του συνόλου χαρακτηριστικών και με βάση αυτά κατασκεύασε τα αντίστοιχα σύνολα δεδομένων (διαφορετικού χώρου εισόδου)

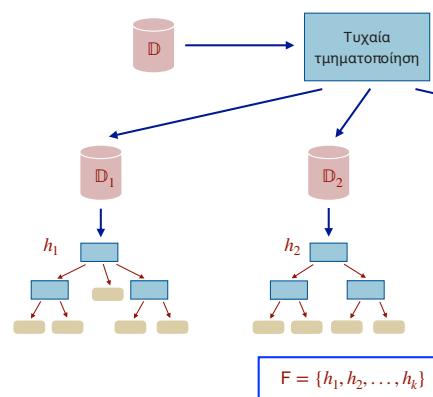
Εκμάθηση μοντέλου και πρόβλεψη

- ▶ Κατασκεύασε ένα δέντρο αποφάσεων για κάθε ένα από τα k διαφορετικά σύνολα δεδομένων
- ▶ Συνέννασε τα αποτελέσματα των k δέντρων αποφάσεων και δώσε στην έξοδο την πρόβλεψη ταξινόμησης της πλειοψηφίας

46



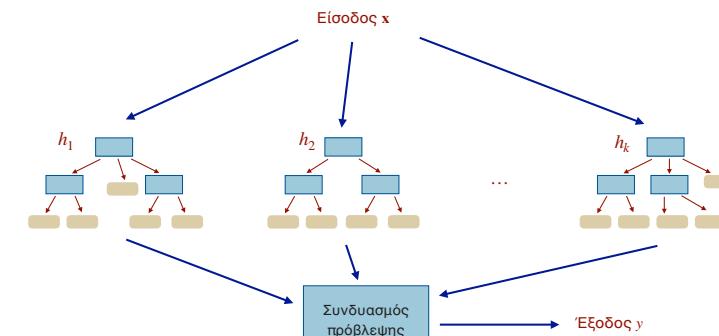
Εκμάθηση



47



Ταξινόμηση



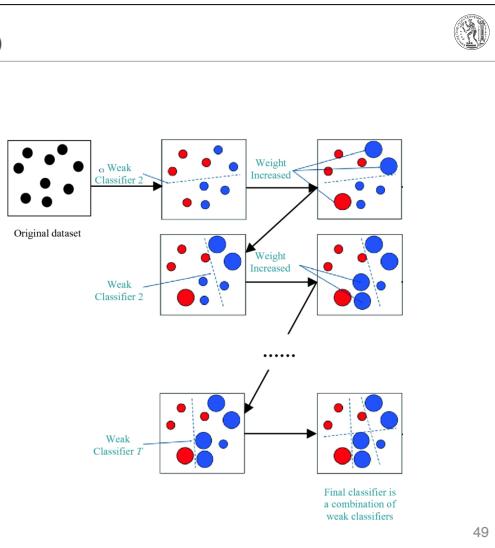
Breiman, L. Random Forests. *Machine Learning* **45**, 5–32 (2001)

48

ΜΕΘΟΔΟΙ ENSEMBLE - ΕΝΙΣΧΥΣΗ (BOOSTING)

Βασική ιδέα

- Παρατίρηση:** Όσο βελτιώνεται η απόδοση των δέντρων απόφασης σε μία περιοχή του πεδίου ορισμού, τόσο χειροτερεύει σε άλλες
- Τεχνική:** Εκπαιδεύοντας ταξινομητές που έχουν καλύτερη απόδοση σε περιοχές που άλλοι έχουν χειρότερη, μπορούν οι ταξινομητές να εξειδικεύονται και έτσι ο συνδυασμός τους να αποδίδει καλύτερα



49

ΜΕΘΟΔΟΙ ENSEMBLE - ΕΝΙΣΧΥΣΗ (BOOSTING)

Βασική ιδέα

Σύνολο δεδομένων $\mathbb{D} = \{\langle \mathbf{x}_i, y_i \rangle\}_{i=1}^n = \{\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle, \dots, \langle x_n, y_n \rangle\}$

$H = \{h | h : X \rightarrow Y\}$: σύνολο δέντρων απόφασης $X \rightarrow Y$

$h \in H \leftarrow \text{DecisionTree.train}(\{\langle \mathbf{x}_i, y_i \rangle\}_{i=1}^n)$

$\text{minimal}(h)$

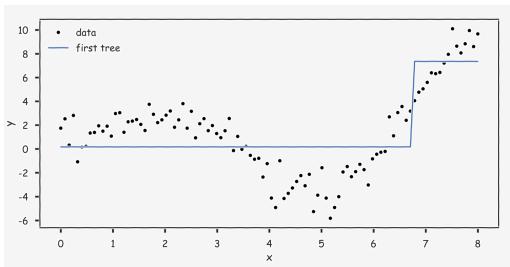
$$h = \sum_{h_j \in H} \lambda_j h_j \leftarrow \text{DecisionTree.train}(\{\langle \mathbf{x}_i, y_i \rangle\}_{i=1}^n)$$

GradientBoostTree . train(\mathbb{D})

- $h_0 = \text{DecisionTree . train}(\mathbb{D})$, όπου $h_0 \in H$, $\mathbb{D} = \{\langle \mathbf{x}_i, y_i \rangle\}_{i=1}^n$
 - $h = h_0$
 - $\mathbb{D}' = \{\langle \mathbf{x}_i, r_i \rangle\}_{i=1}^n$, $r_i = y_i - h(\mathbf{x}_i)$
 - $h_k = \text{DecisionTree . train}(\mathbb{D}')$, όπου $h_k \in H$, $\mathbb{D}' = \{\langle \mathbf{x}_i, r_i \rangle\}_{i=1}^n$
 - $h = h + \lambda h_k$
5. Επανέλαβε τα βήματα 3 έως 5, έως ότου φτάσεις το επιθυμητό αφάλιμα

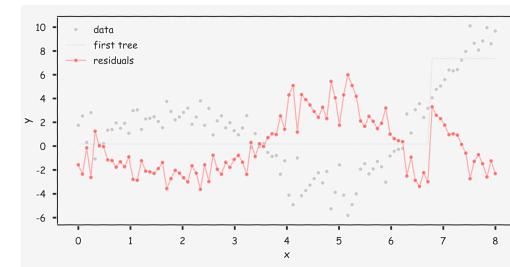
50

ΜΕΘΟΔΟΙ ENSEMBLE - ΕΝΙΣΧΥΣΗ (BOOSTING)



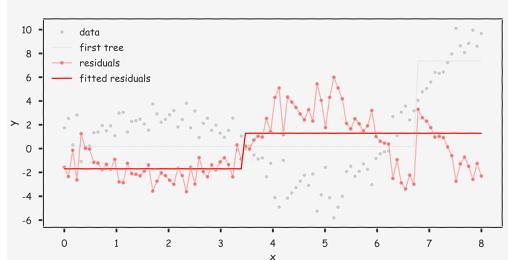
51

ΜΕΘΟΔΟΙ ENSEMBLE - ΕΝΙΣΧΥΣΗ (BOOSTING)



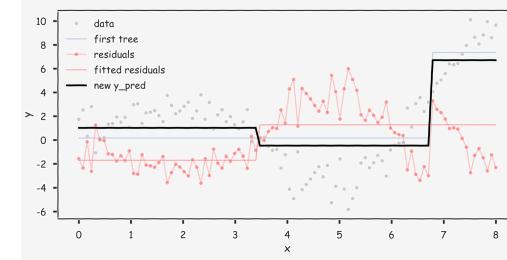
52

ΜΕΘΟΔΟΙ ENSEMBLE - ΕΝΙΣΧΥΣΗ (BOOSTING)



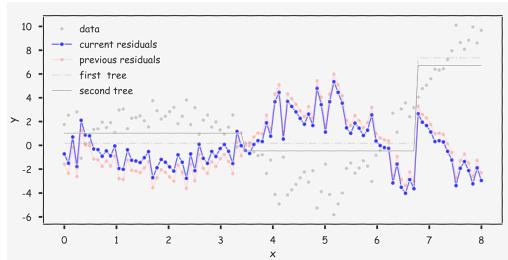
53

ΜΕΘΟΔΟΙ ENSEMBLE - ΕΝΙΣΧΥΣΗ (BOOSTING)



54

ΜΕΘΟΔΟΙ ENSEMBLE - ΕΝΙΣΧΥΣΗ (BOOSTING)



55

Βελτιστοποίηση (Optimization)

Μηχανική Μάθησης

ΔΠΜΣ Επιστήμης Δεδομένων & Μηχανικής Μάθησης

Γιώργος Αλεξανδρίδης – gealexan@mail.ntua.gr

Εισαγωγικές Έννοιες

Το ζητούμενο της επιβλεπόμενης μάθησης

- Εύρεση εκείνων των τιμών των παραμέτρων θ του συστήματος μάθησης, για τις οποίες η αντικειμενική συνάρτηση (objective function) ή συνάρτηση απώλειας (loss function) $J(\theta)$ γίνεται βέλτιστη

Παραδείγματα

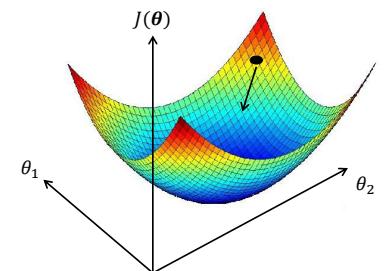
1. Γραμμική Παλινδρόμηση
$$J(\theta) = \sum_{i=1}^N \|x_i\theta - y\|^2 \text{ και } \min_{\theta} J(\theta)$$
 2. Εκτίμηση Μέγιστης Πιθανοφάνειας
$$J(\theta) = \sum_{i=1}^N \log p_{\theta}(x_i) \text{ και } \max_{\theta} J(\theta)$$
 3. Μηχανικές Διανυσμάτων Υποστήμρυξης
$$J(\theta, \xi_i) = \|\theta\|^2 + C \sum_{i=1}^N \xi_i \text{ και } \min_{\theta} J(\theta)$$

subject to $\xi_i \leq 1 - y_i x_i^T \theta, \xi_i \geq 0$
- Χωρίς βλάβη της γενικότητας, θα υποθέσουμε ότι το ζητούμενο είναι να ελαχιστοποιήσουμε την αντικειμενική συνάρτηση

3

Συναρτήσεις απώλειας

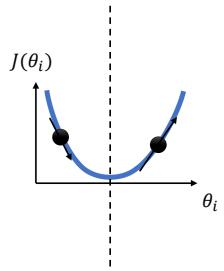
- Υποθέτουμε ότι το μοντέλο μας έχει δύο παραμέτρους (δεξιά οχήμα)
$$\theta^* \leftarrow \arg \min_{\theta} J(\theta)$$
- Βέλτιστη τιμή παραμέτρων θ^*
- Απλός αλγόριθμος προσδιορισμού θ^*
 1. Ξεκίνα με τυχαία αρχική ανάθεση θ
 2. Βρες κατεύθυνσην η όπου η $J(\theta)$ μειώνεται
 3. $\theta \leftarrow \theta + \eta \nu$
 4. Επανέλαβε τα βήματα 2 ως 4 μέχρι τη σύγκλιση στο θ^*
- Το η είναι μια μικρή σταθερά που καλείται **ρυθμός μάθησης** (learning rate) ή **βήμα** (step size)



4

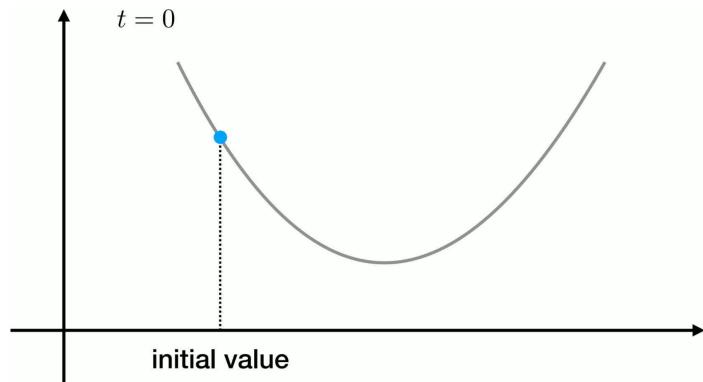
Κατάβαση Κλίσης (Gradient Descent)

- Προς τα πιο κατεύθυνση μειώνεται η αντικειμενική συνάρτηση;
- Υπολογισμός της κλίσης $\frac{\partial J(\theta)}{\partial \theta_i}$ της $J(\theta)$ ως προς παράμετρο θ_i
- Κατεύθυνση μείωσης κλίσης $J(\theta_i)$** (δεξιά σχήμα)
 - Αν η κλίση της J ως προς θ_i είναι **αρνητική** (αριστερό υποτμήμα), θα πρέπει να κινηθούμε προς τα δεξιά.
 - Αν η κλίση της J ως προς θ_i είναι **θετική** (δεξιό υποτμήμα), θα πρέπει να κινηθούμε προς τα αριστερά.
- Σε κάθε περίπτωση και για κάθε παράμετρο κινούμαστε σε κατεύθυνση αντίθετη της κλίσης της αντικειμενικής συνάρτησης ως προς τη συγκεκριμένη παράμετρο $v_i = -\frac{\partial J(\theta)}{\partial \theta_i}$



5

Παράδειγμα Κατάβασης Κλίσης



6

Κατάβαση Κλίσης

Gradient Descent

7

Τεχνικές Κατάβασης Κλίσης

- Διαφέρουν ως προς την ποσότητα των δεδομένων εκπαίδευσης που χρησιμοποιούνται σε κάθε ενημέρωση
 - Κατάβαση κλίσης με **εντατικά δέσμη** δεδομένων (*batch gradient descent*)
 - Στοχαστική** κατάβαση κλίσης (*stochastic gradient descent - SGD*)
 - Κατάβαση κλίσης με **μικρό-δέσμες** δεδομένων (*mini-batch gradient descent*)

8

Ενιαία δέσμη δεδομένων

- Υπολογίζεται την κλίση σε όλα τα N στυγμάτυπα του συνόλου δεδομένων εκπαίδευσης

$$\theta \leftarrow \theta - \eta \nabla_{\theta} \sum_{i=1}^N J(\theta; x_i, y_i)$$

Πλεονεκτήματα

- Εγγυημένη σύγκληση στο ολικό ελάχιστο για κυρτές (convex) αντικειμενικές συναρτήσεις και σε τοπικό ελάχιστο για μη-κυρτές (non-convex)

Μειονεκτήματα

- Πολύ αργή σύγκλιση
- Μη-υπολογίσιμη (intractable) για πολύ μεγάλα σύνολα δεδομένων
 - Που δεν χωράνε, δηλαδή, στη μνήμη του υπολογιστή
- Δεν υποτηρίζεται την online μάθηση

9

Στοχαστική κατάβαση κλίσης

- Υπολογίζεται την κλίση σε κάθε στυγμάτυπο x_i του συνόλου δεδομένων εκπαίδευσης

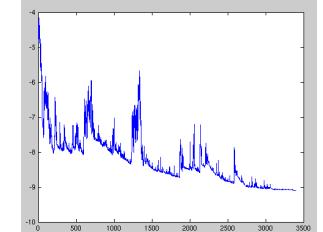
$$\theta \leftarrow \theta - \eta \nabla_{\theta} J(\theta; x_i, y_i)$$

Πλεονεκτήματα

- Πολύ πιο γρήγορη από την κατάβαση κλίσης ενιαίας δέσμης
- Υποτηρίζεται την online μάθηση

Μειονεκτήματα

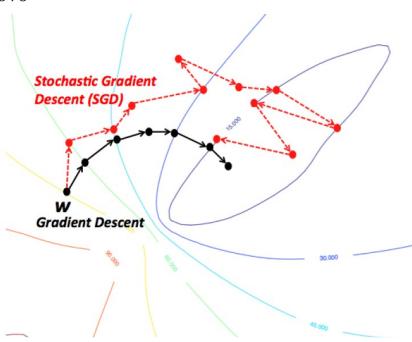
- Μπορεί να εμφανίζεται μεγάλη απόκλιση στις ενημερώσεις των παραμέτρων



Πηγή: https://en.wikipedia.org/wiki/Stochastic_gradient_descent#/media/File:Stogra.png

Σύγκριση των δύο μεθόδων

- Η στοχαστική κατάβαση κλίσης εμφανίζει παρόμοια συμπεριφορά σύγκλισης με τη κατάβαση κλίσης ενιαίας δέσμης αν ο ρυθμός μάθησης μειώνεται σταδιακά με το χρόνο



11

Κατάβαση κλίσης σε μικρό-δέσμες δεδομένων

- Υπολογίζεται την κλίση σε όλα τα $p \ll N$ στυγμάτυπα του συνόλου δεδομένων εκπαίδευσης

$$\theta \leftarrow \theta - \eta \nabla_{\theta} \sum_{i=1}^p J(\theta; x_i, y_i)$$

Πλεονεκτήματα

- Μειώνει τις αποκλίσεις μεταξύ των ενημερώσεων
- Μπορεί να υπολογιστεί γρήγορα με πράξεις πολλαπλασιασμού πυνάκων

Μειονεκτήματα

- Το μέγεθος δέσμης είναι υπερ-παράμετρος της διαδικασίας εκπαίδευσης
 - Αρα πρέπει να δρεθεί η βέλτιστη τιμή της
 - Συνήθως χρησιμοποιούνται δυνάμεις του 2 μεταξύ 8 και 256.

- Συνηθέστερα χρησιμοποιούμενη τεχνική
 - Αναφέρεται και ως στοχαστική κατάβαση κλίσης παρότι χρησιμοποιούνται μικροδέσμες

10

12

Σύγκριση τεχνικών

Τεχνική	Ακρίβεια	Ταχύτητα Ενημέρωσης	Χρήση Μνήμης	Online Μάθηση
Ενιαία δέσμη	Καλή	Χαμηλή	Υψηλή	Όχι
Στοχαστική	Καλή*	Υψηλή	Χαμηλή	Ναι
Μικρο-δέσμες	Καλή	Μεσαία	Μεσαία	Ναι

* Υπό την προϋπόθεση ότι μειώνεται σταδιακά ο ρυθμός μάθησης

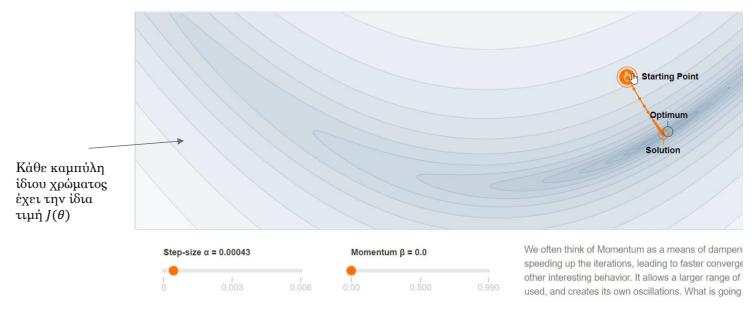
Προκλήσεις

- Αποφυγή τοπικών ελαχίστων
- Επιλογή ρυθμού μάθησης
- Καθορισμός διαδικασίας μεταβολής (μείωσης) του ρυθμού μάθησης

13

14

Απεικόνιση κατάβασης κλίσης



15

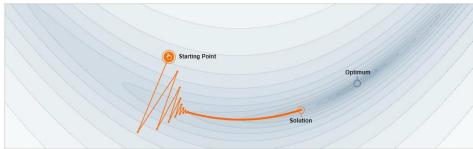
Επιφάνειες Συναρτήσεων Απώλειας

Loss surfaces

16

Κίνηση αντίθετα από την κλιση

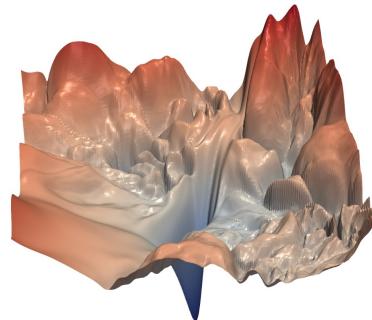
- Η ενημέρωση των παραμέτρων του μοντέλου σε κατεύθυνση αντίθετη από την κλιση της συνάρτησης απόλεις δεν εγγυάται πάντα το καλύτερο αποτέλεσμα
 - Εδικά αν η συνάρτηση απόλεις είναι μη-κυρτή
- Υπενθύμιση**
 - Κυρτές συναρτήσεις είναι δεσ οι είναι διπλά-παραγωγισμες στο πεδίο ορισμού τους και ταυτόχρονα η δεύτερη παράγωγος τους είναι παντού στο πεδίο ορισμού τους μη-αρνητική



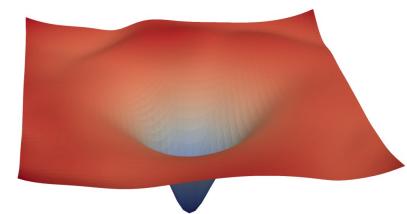
17

Συναρτήσεις απώλειας νευρωνικών δικτύων

ResNet-56 χωρίς skip connections



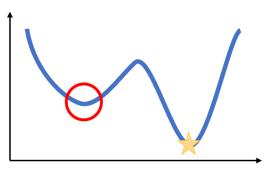
ResNet-56 με skip connections



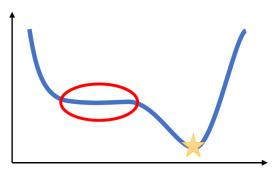
Πηγή: <https://arxiv.org/abs/1712.09913>

18

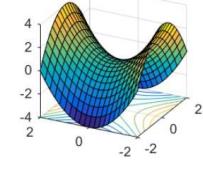
Ειδικές περιπτώσεις μη-κυρτότητας



Τοπικό ελάχιστο



«Οροπέδιο»



Σαγματικό Σημείο

19

Ειδικές περιπτώσεις μη-κυρτότητας

Τοπικά ελάχιστα

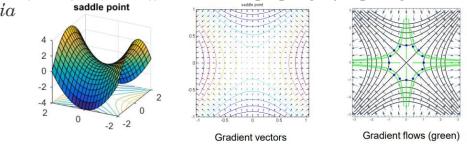
- Παρότι φαίνονται να είναι σοβαρό πρόβλημα, στην πράξη η επίδρασή τους μετριάζεται όσο ανάδονον οι παράμετροι του μοντέλου
- Σε μεγάλα μοντέλα, παρότι υπάρχουν, εμπειρικά έχει φανεί πως δεν είναι πολύ χειρότερα από τα ολικά ελάχιστα

Οροπέδια

- Δεν πρέπει να επλέγουμε εξ' αρχής πολύ μικρούς μικρούς μάθησης για να μην «κολλάμε» σε οροπέδια

Σαγματικά σημεία

- Πολλά μικρές κλισεις στα σαγματικά σημεία
- Στην πράξη έχει παρατηρηθεί ότι τα περισσότερα «κρίσιμα» σημεία σε συναρτήσεις οφάλματος νευρωνικών δικτύων είναι σαγματικά σημεία



20

Επιτάχυνση κλίσης

21

Μέθοδος Newton

- Ανάπτυγμα Taylor συνάρτησης απώλειας γύρω από σημείο θ_0

$$J(\theta) \approx J(\theta_0) + \nabla_{\theta} J(\theta_0)(\theta - \theta_0) + \frac{1}{2}(\theta - \theta_0)^T \nabla_{\theta}^2 J(\theta_0)(\theta - \theta_0)$$

- Αν το μοντέλο έχει n παραμέτρους τότε

• Κλίση $\nabla_{\theta} J(\theta_0) = \left[\frac{\partial J(\theta_0)}{\partial \theta_1}, \frac{\partial J(\theta_0)}{\partial \theta_2}, \dots, \frac{\partial J(\theta_0)}{\partial \theta_n} \right]$ έχει n παραμέτρους

• Εστιανός Πίνακας $\nabla_{\theta}^2 J(\theta_0) = \begin{bmatrix} \frac{\partial^2 J(\theta_0)}{\partial \theta_1 \partial \theta_1} & \dots & \frac{\partial^2 J(\theta_0)}{\partial \theta_1 \partial \theta_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 J(\theta_0)}{\partial \theta_n \partial \theta_1} & \dots & \frac{\partial^2 J(\theta_0)}{\partial \theta_n \partial \theta_n} \end{bmatrix}$ έχει n^2 παραμέτρους

- Κλίση γύρω από το τοπικό ελάχιστο θ^* μηδενική, οπότε

$$\theta^* \leftarrow \theta_0 - (\nabla_{\theta}^2 J(\theta_0))^{-1} \nabla_{\theta} J(\theta_0)$$

- Όρος $(\nabla_{\theta}^2 J(\theta_0))^{-1} \nabla_{\theta} J(\theta_0)$ έχει πολυπλοκότητα $\mathcal{O}(n^3)$ που τον καθιστά απαγορευτικό για μεγάλα μοντέλα

- Για αυτό το λόγο αποφεύγουμε μεθόδους που απαιτούν τον υπολογισμό δευτέρων παραγώγων και προσπαθούμε να «επιταχύνουμε» την κατάβαση κλίσης

Ορμή (Momentum)

Κεντρική Ιδέα

- Αν διαδρχικές κλίσεις «θείχνονται» προς την *ιδία* κατεύθυνση, θα πρέπει να κινηθούμε προς τα εκεί πιο γρήγορα

Όρος ορμής

- Προσθήκη ενός κλάσματος γ (συνήθως 0,9) του διανύσματος του προηγούμενου βήματος στο τωρινό

$$v_t \leftarrow \gamma v_{t-1} + \eta \nabla_{\theta} J(\theta) \text{ και } \theta \leftarrow \theta - v_t$$

- Μειώνει το είρος της ενημέρωσης για τις παραμέτρους εκείνες που αλλάζουν κατεύθυνση στην κλίση
- Αυξάνει το είρος της ενημέρωσης για τις παραμέτρους εκείνες που δεν αλλάζουν κατεύθυνση στην κλίση



SGD χωρίς ορμή



SGD με ορμή

23

Επιταχυνόμενη Κλίση Nesterov

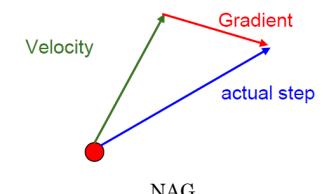
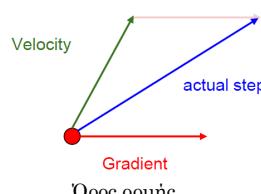
- Η προσθήκη όρου ορμής επιταχύνει την κατάβαση στα τωφλά

- Πρώτα υπολογίζει την κλίση και στη συνέχεια πραγματοποιεί ένα μεγάλο άλμα

- Η **επιταχυνόμενη κλίση Nesterov** (*Nesterov Accelerated Gradient – NAG*)

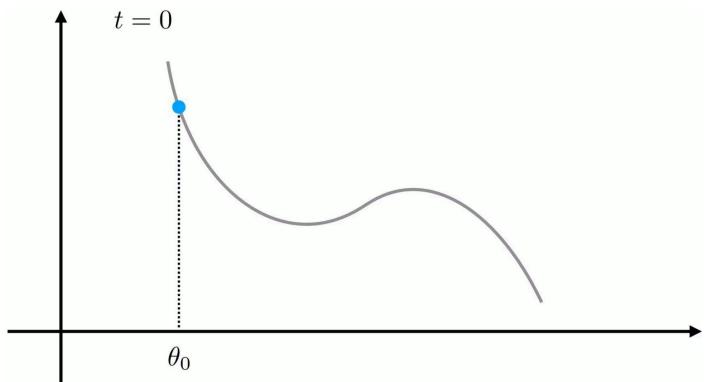
- Πρώτα πραγματοποιεί το άλμα στην κατεύθυνση της προηγούμενης κλίσης $\theta - \gamma v_{t-1}$
- Κατόπιν «μετρά» που έχει καταλήξει και κάνει διόρθωση των παραμέτρων

$$v_t \leftarrow \gamma v_{t-1} + \eta \nabla_{\theta} J(\theta - \gamma v_{t-1}) \text{ και } \theta \leftarrow \theta - v_t$$



24

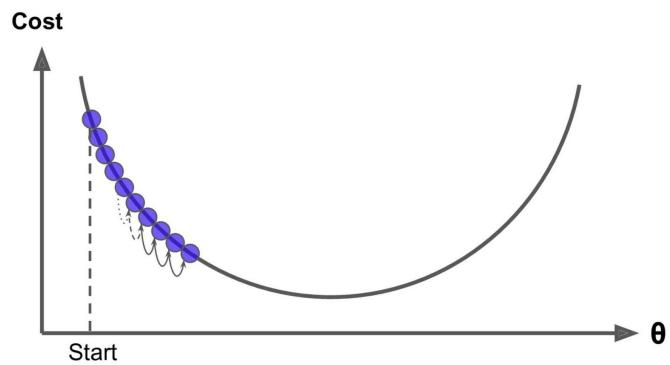
Κατάβαση κλίσης με ορμή



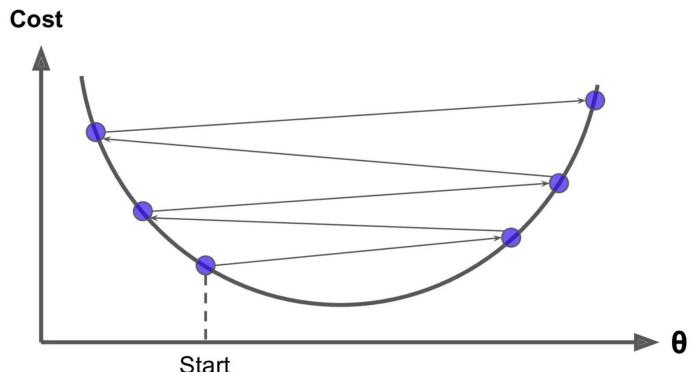
Επίδραση Ρυθμού
Μάθησης

26

Χαμηλός ρυθμός μάθησης



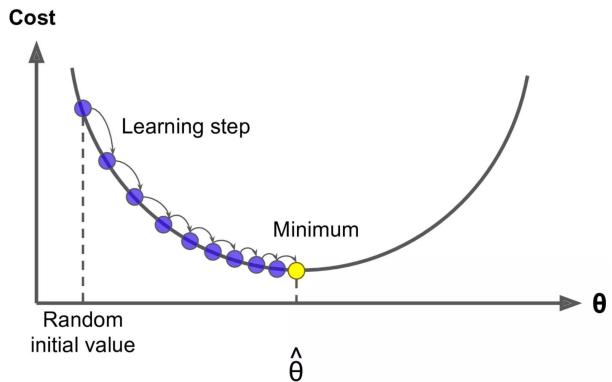
Υψηλός ρυθμός μάθησης



27

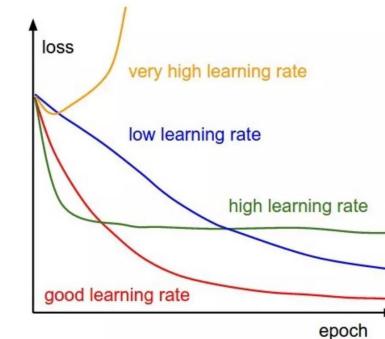
28

Κανονικός ρυθμός μάθησης



29

Επίδραση ρυθμού μάθησης



30

Adagrad

- Προσαρμόζεται το ρυθμό μάθησης στις παραμέτρους του μοντέλου
 - Μεγάλες ενημερώσεις για μη-συχνές παραμέτρους, μικρές ενημερώσεις για συχνές παραμέτρους
- Κανόνας ενημέρωσης SGD: $\theta_{t+1} = \theta_t - \eta \cdot g_t$ και $g_t = \nabla_{\theta_t} J(\theta_t)$
- Ο Adagrad διαιρεί το ρυθμό μάθησης με την τετραγωνική ρίζα του αθροίσματος των τετραγώνων των προηγούμενων κλίσεων
$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t$$

 - $G_t \in \mathbb{R}^{d \times d}$ διαγώνιος πίνακας με το i-oστό του στοιχείο να είναι ίσο με το άθροισμα των τετραγώνων των κλίσεων του θ_t μέχρι τη χρονική στιγμή t
 - ϵ όρος ομαλοποίησης για να αποφευχθεί η διαιρέση με το μηδέν
 - \odot πολλαπλασιασμός Hadamard

31

Adagrad

- Πλεονεκτήματα**
 - Κατάλληλος για χρήση όταν τα δεδομένα είναι αραιά
 - Βελτιώνει ομαντικά την ενραστία του SGD
 - Δεν απαιτεί τη «χειροκίνητη» ρύθμιση του ρυθμού μάθησης
- Μειονεκτήματα**
 - Συσσωρεύει τετραγωνικές κλίσεις στον παρονομαστή και έτοι προκαλεί μεγάλη μείωση του ρυθμού μάθησης

32

Adadelta

- Αντιμετωπίζει το μειονέκτημα του Adagrad μέσω του περιορισμού του πλήθους των παρελθοντικών κλίσεων που λαμβάνονται υπόψη, σε παράθυρο σταθερού μεγέθους

Ενημέρωση SGD

$$\theta_{t+1} = \theta_t + \Delta\theta_t \text{ και } \Delta\theta_t = -\eta \cdot g_t$$

- Ορίζει έναν τρέχοντα μέσο όρο του τετραγώνου των κλίσεων $\mathbb{E}[g^2]_t$ τη χρονική στιγμή t

$$\mathbb{E}[g^2]_t = \gamma \mathbb{E}[g^2]_{t-1} + (1-\gamma) g_t^2$$

- γ : όρος αντίστοιχου του παράγοντα ορμής, συνήθως λαμβάνει την τιμή 0,9

Ενημέρωση Adadelta

33

RMSProp

- Προτάθηκε από τον Geoff Hinton την ίδια περίοδο με τον Adadelta και έχει παρόμοια φιλοσοφία
- Επίσης διαιρεί το ρυθμό μάθησης με τον τρέχοντα μέσο όρο του τετραγώνου των κλίσεων

$$\begin{aligned}\mathbb{E}[g^2]_t &= \gamma \mathbb{E}[g^2]_{t-1} + (1-\gamma) g_t^2 \\ \theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{\mathbb{E}[g^2]_t + \epsilon}} g_t\end{aligned}$$

- Ο όρος φθοράς γ συνήθως τίθεται στο 0,9 ενώ ο ρυθμός μάθησης η στο 0,001

34

Επίδραση Ρυθμού Μάθησης και Ορμής

35

Adaptive Moment Estimation (Adam)

- Όπως οι Adadelta και RMSprop, ο Adam υπολογίζει τον κινούμενο μέσο όρο των προηγούμενων τετραγωνικών κλίσεων v_t
- Όπως οι βελτιστοποιητές με παράγοντα ορμής, υπολογίζει τον κινούμενο μέσο όρο των προηγούμενων κλίσεων m_t

Ενημέρωση Adam

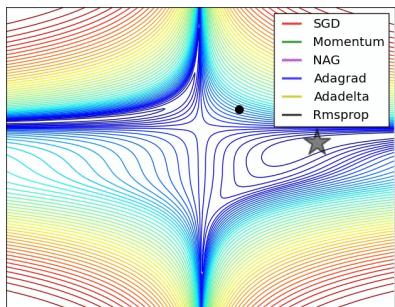
$$\begin{aligned}m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^2\end{aligned}$$

- m_t : μέση τιμή κλίσεων (διάνυσμα που αρχικοποιείται στο 0)
- v_t : (μη-κεντραρισμένη) διασπορά κλίσεων (διάνυσμα που αρχικοποιείται στο 0)
- β_1, β_2 : ρυθμός φθοράς
- Αφαίρεση μεροληψίας (bias) προς το 0 από m_t, v_t
 $\hat{m}_t = \frac{m_t}{1-\beta_1^t}$ και $\hat{v}_t = \frac{v_t}{1-\beta_2^t}$
- Κανόνας Ενημέρωσης Adam: $\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t$

36

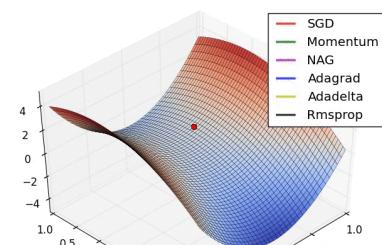
Οπτικοποίηση των αλγορίθμων

Επιφάνειες συναρτήσεων απώλειας



Πηγή: <https://imgur.com/a/visualizing-optimization-algos-Hqolp>

Σαγματικό ομρείο



Καταλληλότητα

- Μέθοδοι του προσαρμοστικού ρυθμού μάθησης (Adagrad, Adadelta, RMSProp, Adam) είναι καταλληλότεροι για προβλήματα με αραιά χαρακτηριστικά
- *Adagrad, Adadelta, RMSProp, Adam* έχουν ικανοποιητική απόδοση σε παρόμοιες συνθήκες
- Οι δημιουργοί του *Adam* υχυρίζονται ότι το βήμα διόρθωσης της μεροληψίας των καθιστά ελαφρώς καλύτερο από τον *RMSProp*

37

38

Βιβλιογραφία

- Quian, N. (1999) - “On the momentum term in gradient descent learning algorithms” – *Neural networks : the official journal of the International Neural Network Society*, 12(1):145-151
- Nesterov, Y. (1983) – “A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$ ” - *Doklady ANSSSR (translated as Soviet.Math.Docl.)*, 269:543-547
- Duchi, J. (2011) – “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization” - *Journal of Machine Learning Research*, 12:2121-2159.
- Zeiler, M. D. (2012) – “ADADELTA: An Adaptive Learning Rate Method” *arXiv preprint arXiv:1212.5701*.
- Kingma, D. P. (2015) – “Adam: a Method for Stochastic Optimization.” *International Conference on Learning Representations*, pages 1-13.



Machine Learning

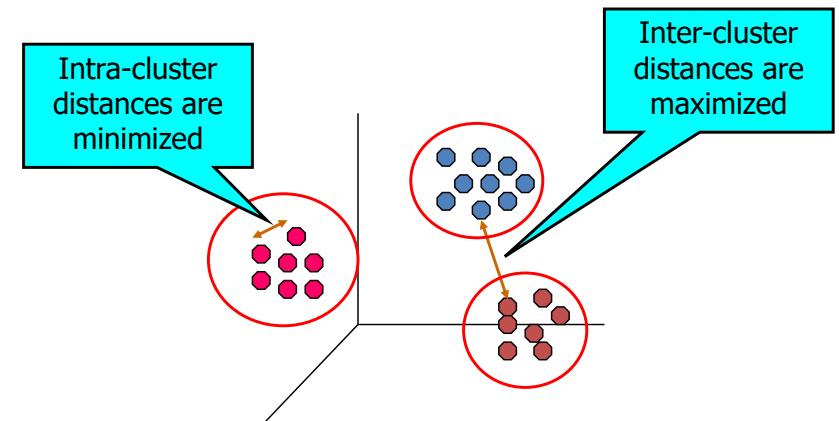
Unsupervised Learning (Clustering)

Athanasiос (Thanos) Voulodimos
Assistant Professor

Artificial Intelligence and Learning Systems Laboratory
School of Electrical and Computer Engineering
National Technical University of Athens

What is a Clustering?

- In general a **grouping** of objects such that the objects in a **group (cluster)** are similar (or related) to one another and different from (or unrelated to) the objects in other groups



Applications of Cluster Analysis

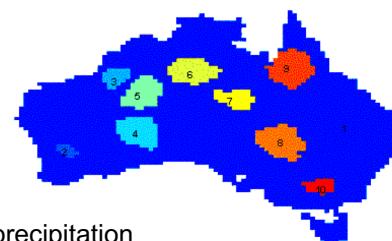
• Understanding

- Group related documents for browsing, group genes and proteins that have similar functionality, or group stocks with similar price fluctuations

	Discovered Clusters	Industry Group
1	Applied-Mail-DOWN,Bay-Network-Down,3-COM-DOWN, Cabletron-Sys-DOWN,CISCO-DOWN,HP-DOWN, DSC-Comm-DOWN,INTEL-DOWN,LSI-Logic-DOWN, Micron-Tech-DOWN,Texas-Inst-DOWN,Tellabs-Inc-Down, Natl-Semiconductor-DOWN,Oracle-DOWN,SGI-DOWN, Sun-DOWN	Technology1-DOWN
2	Apple-Comp-DOWN,Autodesk-DOWN,DEC-DOWN, ADV-Micro-Device-DOWN,Andrew-Corp-DOWN, Computer-Assoc-DOWN,Circuit-City-DOWN, Compaq-DOWN,EMC-Corp-DOWN,Gen-Inst-DOWN, Motorola-DOWN,Microsoft-DOWN,Scientific-Atti-DOWN	Technology2-DOWN
3	Fannie-Mae-DOWN,Fed-Home-Loan-DOWN, MBNA-Corp-DOWN,Morgan-Stanley-DOWN	Financial-DOWN
4	Baker-Hughes-UP,Dresser-Inds-UP,Halliburton-HLD-UP, Louisiana-Land-UP,Phillips-Petro-UP,Uncocal-UP, Schlumberger-UP	Oil-UP

• Summarization

- Reduce the size of large data sets



Clustering precipitation
in Australia

Early applications of cluster analysis

• John Snow, London 1854

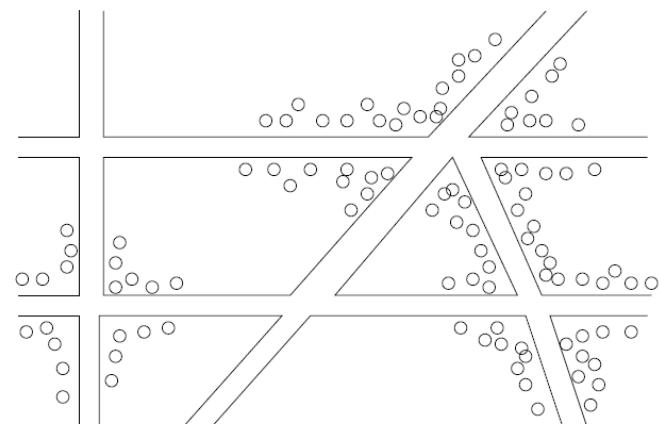
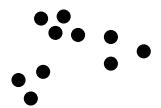
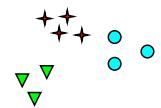


Figure 1.1: Plotting cholera cases on a map of London

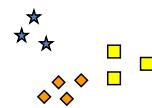
Notion of a Cluster can be Ambiguous



How many clusters?

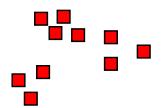


Six Clusters

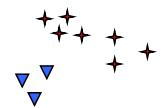
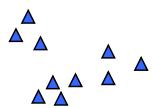


Types of Clusterings

- A **clustering** is a set of **clusters**
- Important distinction between **hierarchical** and **partitional** sets of clusters
- **Partitional Clustering**
 - A division data objects into subsets (**clusters**) such that each data object is in exactly one subset
- **Hierarchical clustering**
 - A set of nested clusters organized as a hierarchical tree

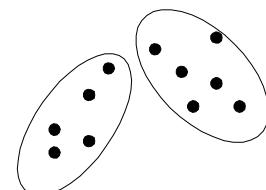
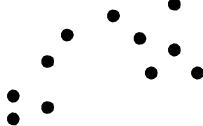


Two Clusters

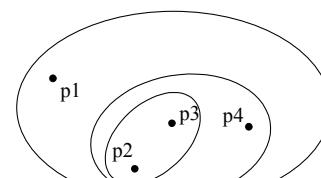
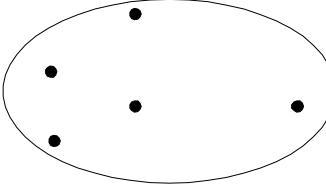
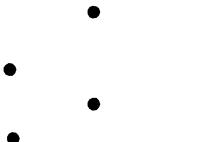


Four Clusters

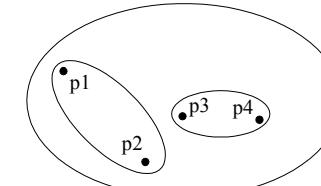
Partitional Clustering



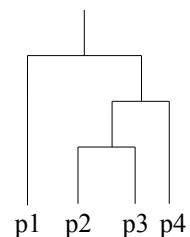
Hierarchical Clustering



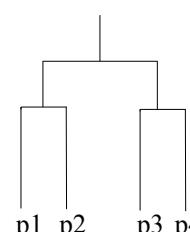
Traditional Hierarchical Clustering



Non-traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Dendrogram

Original Points

A Partitional Clustering

Other types of clustering

- Exclusive (or non-overlapping) versus non-exclusive (or overlapping)
 - In non-exclusive clusterings, points may belong to multiple clusters.
 - Points that belong to multiple classes, or 'border' points
- Fuzzy (or soft) versus non-fuzzy (or hard)
 - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
 - Weights usually must sum to 1 (often interpreted as probabilities)
- Partial versus complete
 - In some cases, we only want to cluster some of the data

Types of Clusters: Objective Function

- Clustering as an optimization problem
 - Finds clusters that minimize or maximize an objective function.
 - Enumerate all possible ways of dividing the points into clusters and evaluate the 'goodness' of each potential set of clusters by using the given objective function. (NP Hard)
 - Can have global or local objectives.
 - Hierarchical clustering algorithms typically have local objectives
 - Partitional algorithms typically have global objectives
 - A variation of the global objective function approach is to fit the data to a parameterized model.
 - The parameters for the model are determined from the data, and they determine the clustering
 - E.g., Mixture models assume that the data is a 'mixture' of a number of statistical distributions.

Clustering Algorithms

- K-means and its variants
- Hierarchical clustering
- DBSCAN

K-MEANS

K-means Clustering

- Partitional clustering approach
- Each cluster is associated with a **centroid** (center point)
- Each point is assigned to the cluster with the **closest** centroid
- Number of clusters, **K**, must be specified
- The objective is to **minimize the sum of distances** of the points to their respective **centroid**

K-means Clustering

- **Problem:** Given a set **X** of **n** points in a **d**-dimensional space and an integer **K** group the points into **K** clusters $C = \{C_1, C_2, \dots, C_k\}$ such that

$$Cost(C) = \sum_{i=1}^k \sum_{x \in C_i} dist(x, c_i)$$

is **minimized**, where c_i is the **centroid** of the points in cluster C_i

K-means Clustering

- Most common definition is with euclidean distance, minimizing the **Sum of Squares Error (SSE)** function
 - Sometimes K-means is defined like that

- **Problem:** Given a set **X** of **n** points in a **d**-dimensional space and an integer **K** group the points into **K** clusters $C = \{C_1, C_2, \dots, C_k\}$ such that

$$Cost(C) = \sum_{i=1}^k \sum_{x \in C_i} (x - c_i)^2$$

is **minimized**, where c_i is the **mean** of the points in cluster C_i

Sum of Squares Error (SSE)

Complexity of the k-means problem

- **NP-hard** if the dimensionality of the data is at least 2 (**d>=2**)
 - Finding the best solution in polynomial time is infeasible
- For **d=1** the problem is solvable in polynomial time
 - A simple iterative algorithm works quite well in practice

K-means Algorithm

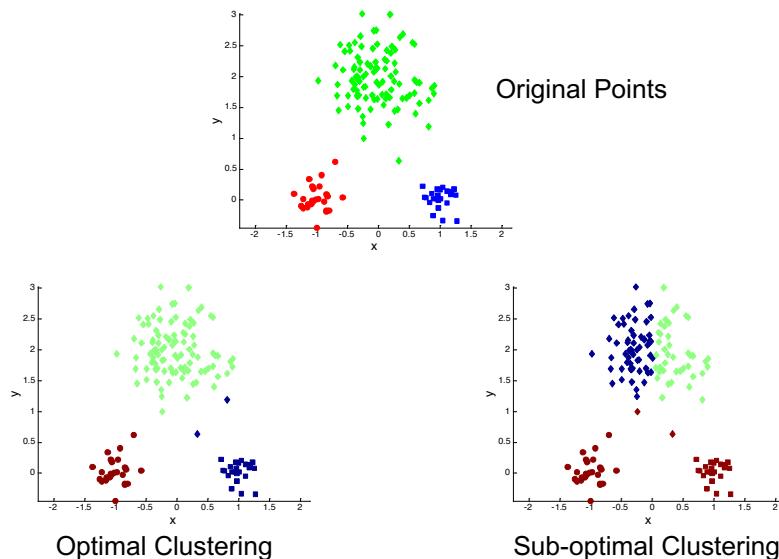
- Also known as **Lloyd's algorithm**.
- K-means is sometimes synonymous with this algorithm

```
1: Select  $K$  points as the initial centroids.  
2: repeat  
3:   Form  $K$  clusters by assigning all points to the closest centroid.  
4:   Recompute the centroid of each cluster.  
5: until The centroids don't change
```

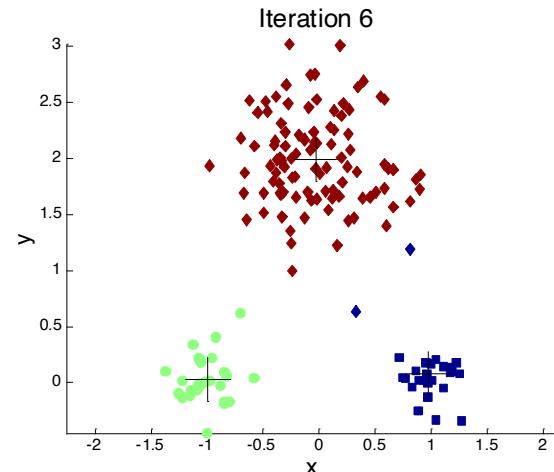
K-means Algorithm – Initialization

- Initial centroids are often chosen **randomly**.
 - Clusters produced vary from one run to another.

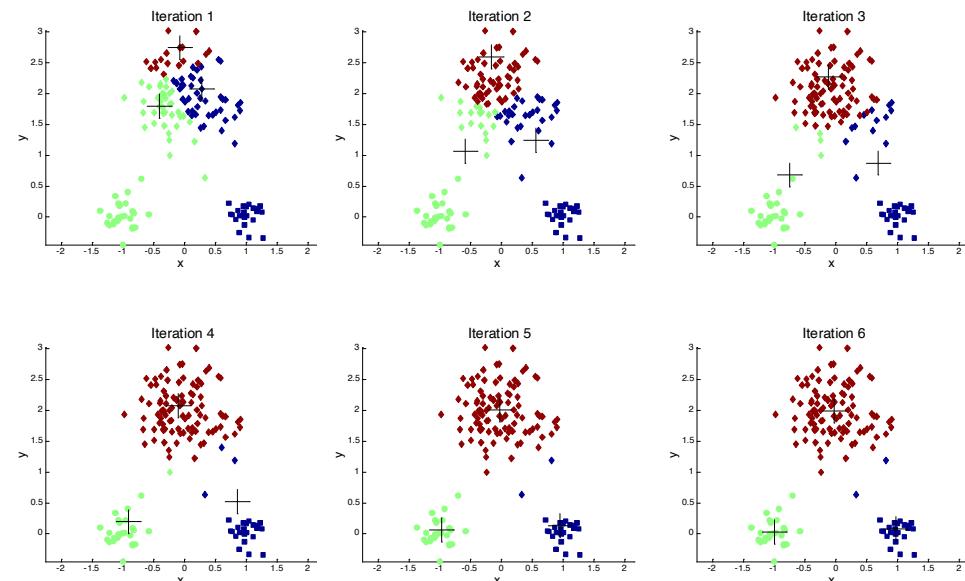
Two different K-means Clusterings



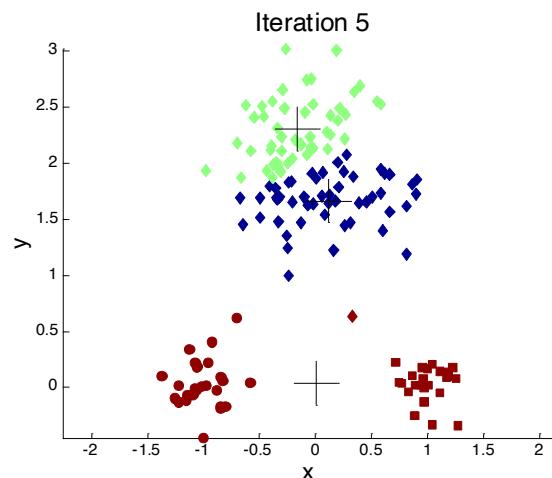
Importance of Choosing Initial Centroids



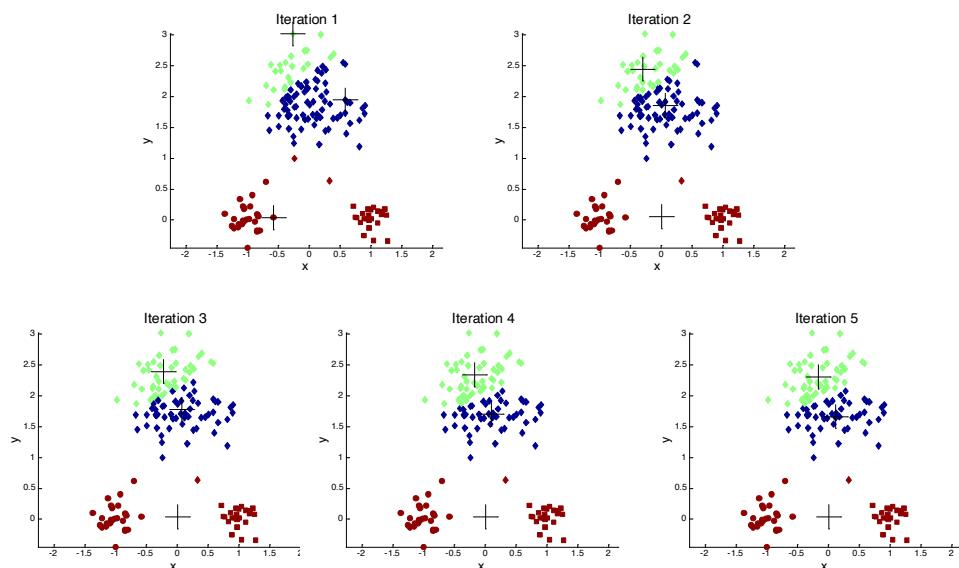
Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids ...



Dealing with Initialization

- Do **multiple runs** and select the clustering with the smallest error
- Select original set of points by methods other than random. E.g., pick the most distant (from each other) points as cluster centers (**K-means++** algorithm)

K-means Algorithm – Centroids

- The **centroid** depends on the distance function
 - The **minimizer** for the distance function
- ‘**Closeness**’ is measured by Euclidean distance (SSE), cosine similarity, correlation, etc.
- **Centroid:**
 - The **mean** of the points in the cluster for SSE, and cosine similarity
 - The **median** for Manhattan distance.
- Finding the centroid is not always easy
 - It can be an NP-hard problem for some distance functions
 - E.g., median from multiple dimensions

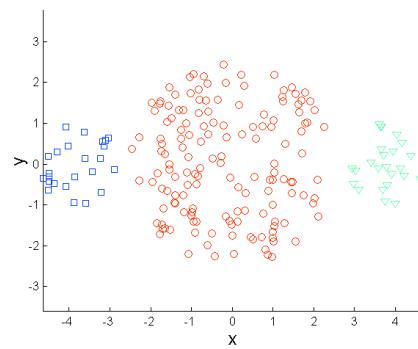
K-means Algorithm – Convergence

- K-means will **converge** for common similarity measures mentioned above.
 - Most of the convergence happens in the first few iterations.
 - Often the stopping condition is changed to ‘Until relatively few points change clusters’
- Complexity is $O(n * K * I * d)$
 - n = number of points, K = number of clusters, I = number of iterations, d = dimensionality
- In general a fast and efficient algorithm

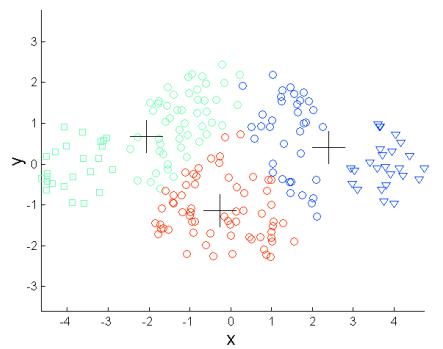
Limitations of K-means

- K-means has problems when clusters are of different
 - Sizes
 - Densities
 - **Non-globular** shapes
- K-means has problems when the data contains outliers.

Limitations of K-means: Differing Sizes

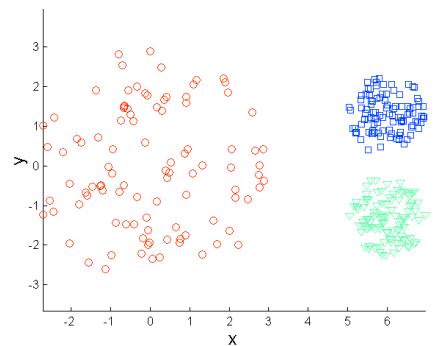


Original Points

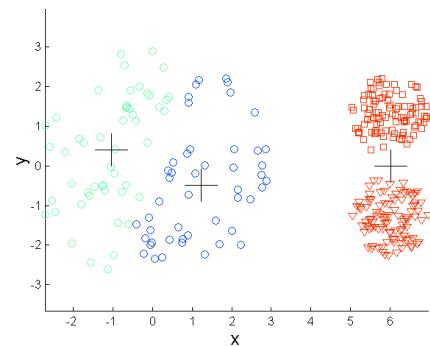


K-means (3 Clusters)

Limitations of K-means: Differing Density

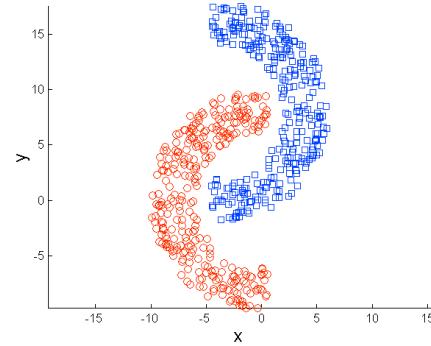


Original Points

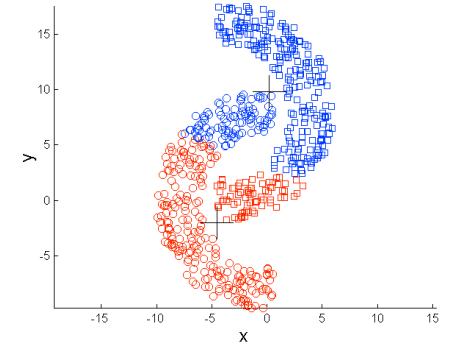


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes

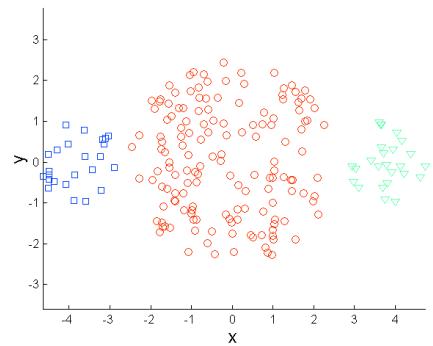


Original Points

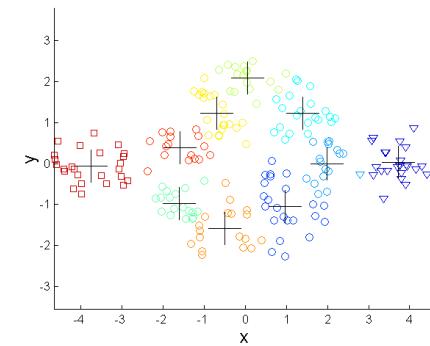


K-means (2 Clusters)

Overcoming K-means Limitations



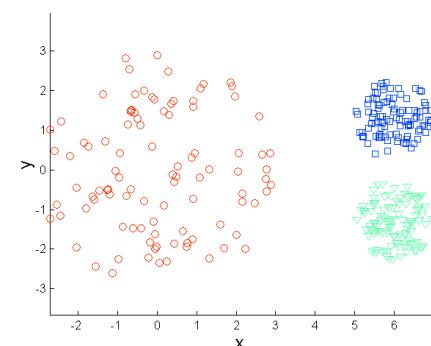
Original Points



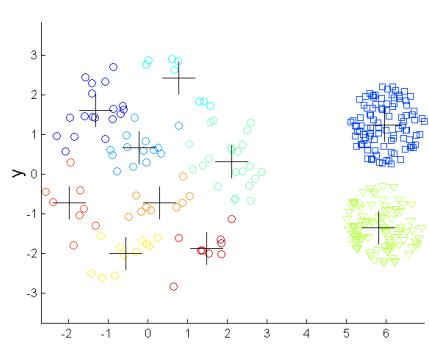
K-means Clusters

One solution is to use many clusters.
Find parts of clusters, but need to put together.

Overcoming K-means Limitations

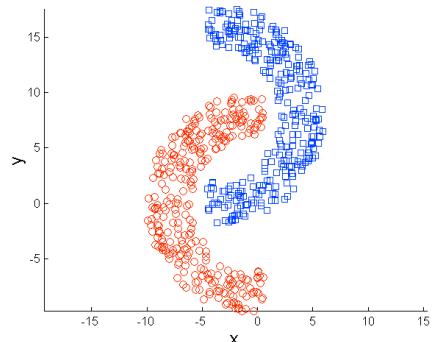


Original Points

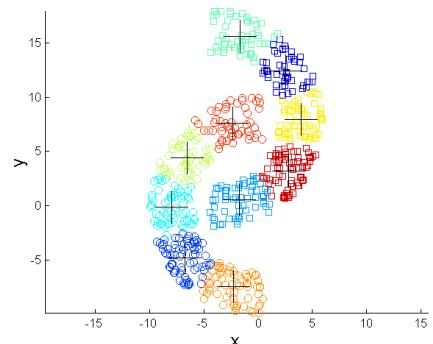


K-means Clusters

Overcoming K-means Limitations



Original Points



K-means Clusters

Variations

- **K-medoids:** Similar problem definition as in K-means, but the centroid of the cluster is defined to be one of the points in the cluster (the **medoid**).
- **K-centers:** Similar problem definition as in K-means, but the goal now is to minimize the maximum **diameter** of the clusters (diameter of a cluster is maximum distance between any two points in the cluster).

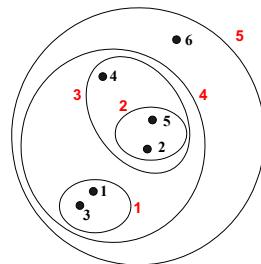
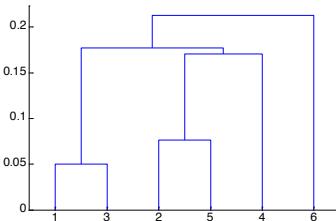
HIERARCHICAL CLUSTERING

Hierarchical Clustering

- Two main types of hierarchical clustering
 - **Agglomerative:**
 - Start with the points as individual clusters
 - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
 - **Divisive:**
 - Start with one, all-inclusive cluster
 - At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a **similarity** or **distance matrix**
 - Merge or split one cluster at a time

Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
 - A tree like diagram that records the sequences of merges or splits



Strengths of Hierarchical Clustering

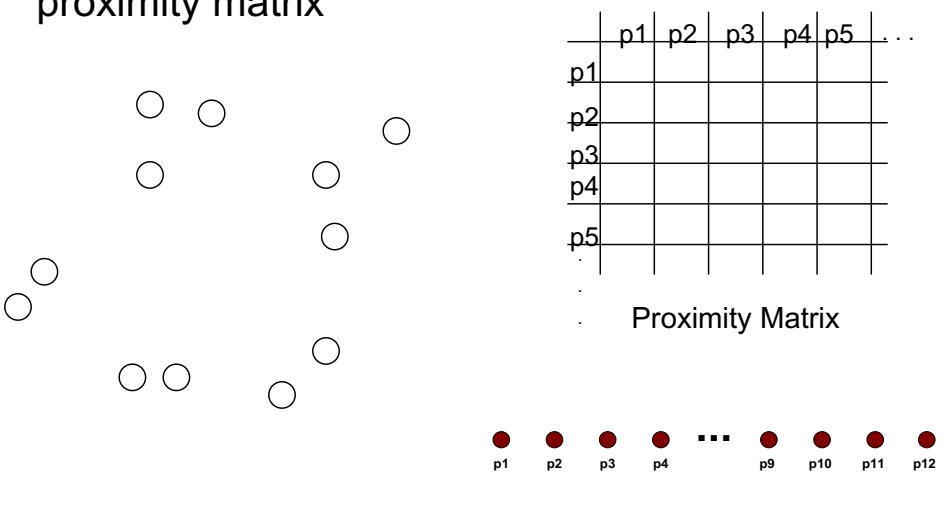
- Do not have to assume any particular number of clusters
 - Any desired number of clusters can be obtained by ‘cutting’ the dendrogram at the proper level
- They may correspond to meaningful taxonomies
 - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, ...)

Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique
- Basic algorithm is straightforward
 1. Compute the **proximity matrix**
 2. Let each data point be a cluster
 3. **Repeat**
 4. Merge the two closest clusters
 5. Update the proximity matrix
 6. **Until** only a single cluster remains
- Key operation is the computation of the **proximity of two clusters**
 - Different approaches to defining the distance between clusters distinguish the different algorithms

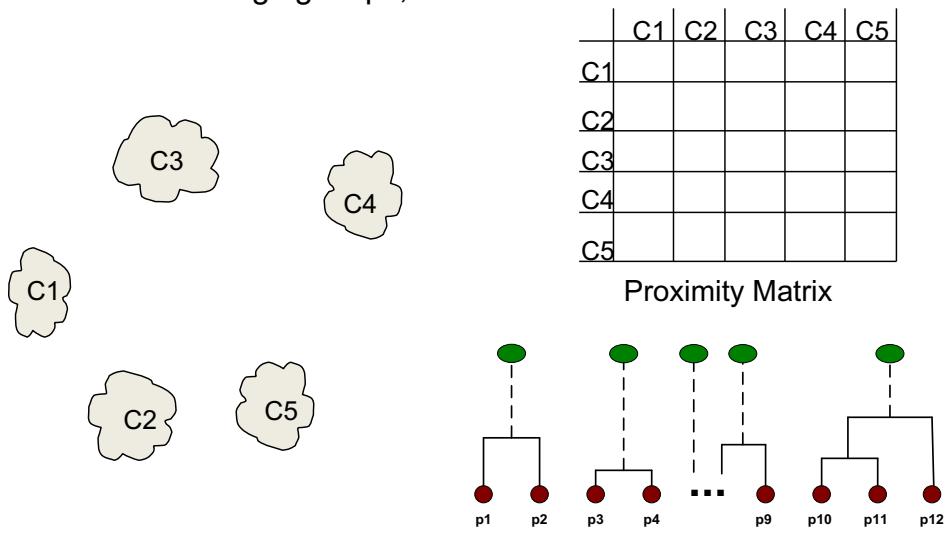
Starting Situation

- Start with clusters of individual points and a proximity matrix



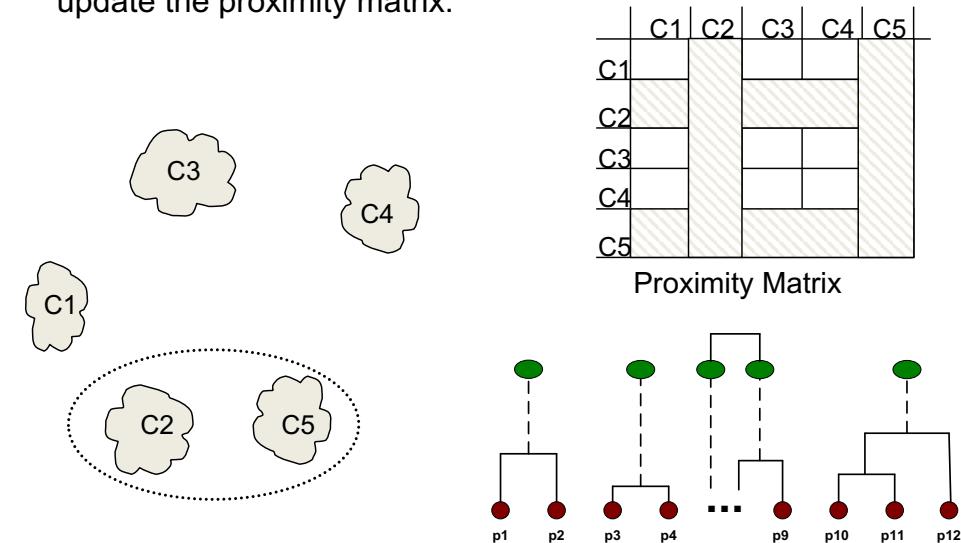
Intermediate Situation

- After some merging steps, we have some clusters



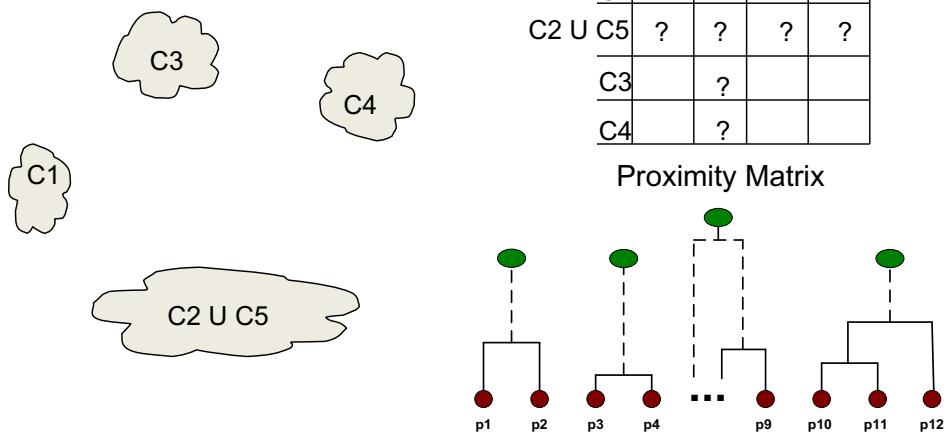
Intermediate Situation

- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.

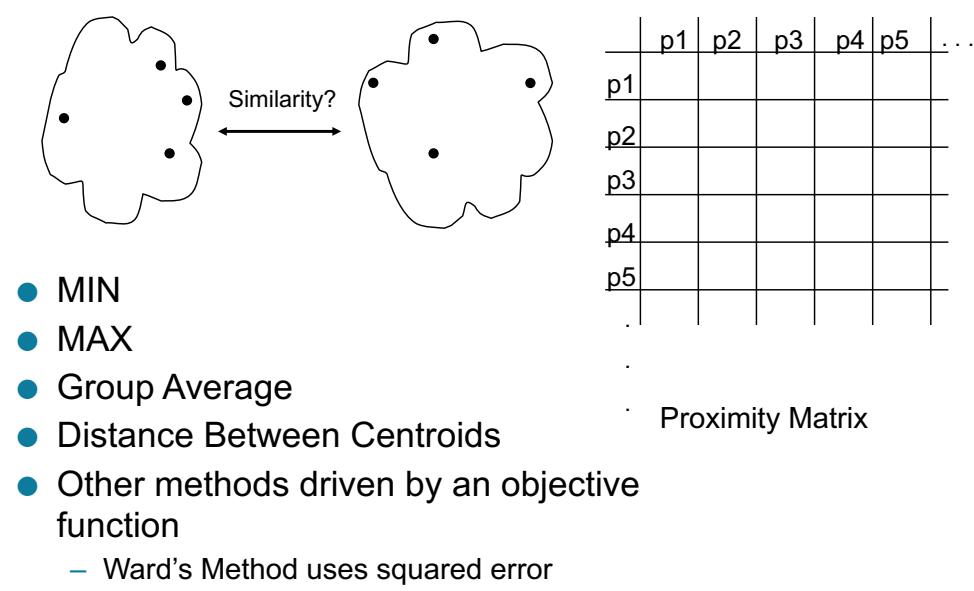


After Merging

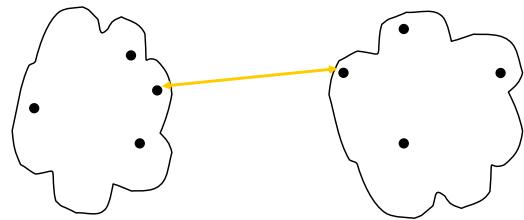
- The question is “How do we update the proximity matrix?”



How to Define Inter-Cluster Similarity



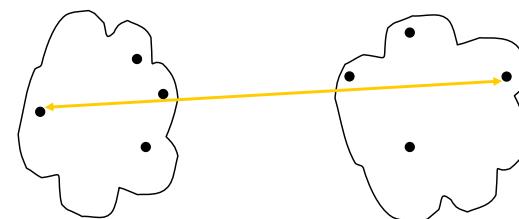
How to Define Inter-Cluster Similarity



- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.

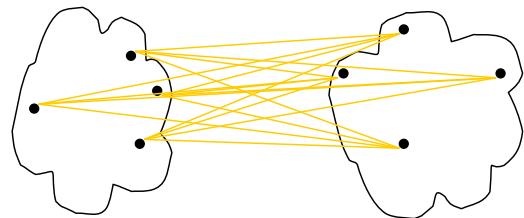
How to Define Inter-Cluster Similarity



- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.

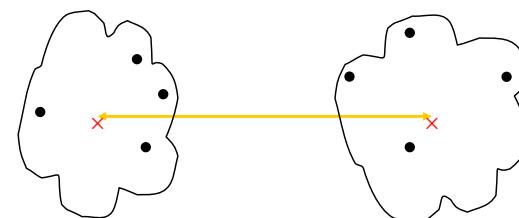
How to Define Inter-Cluster Similarity



- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.

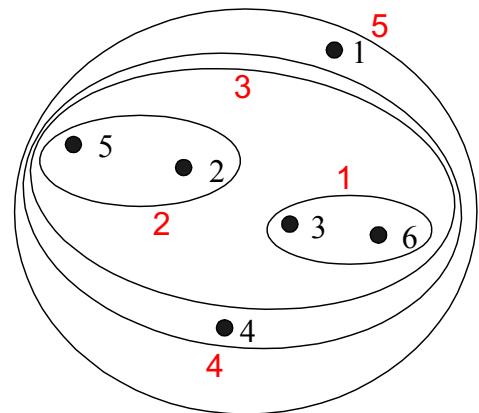
How to Define Inter-Cluster Similarity



- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

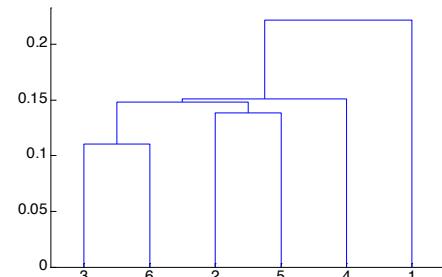
	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.

Hierarchical Clustering: MIN

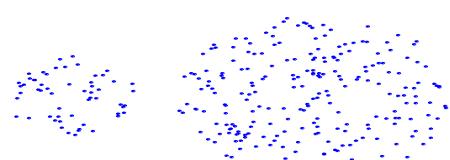


Dendrogram

	1	2	3	4	5	6
1	0	.24	.22	.37	.34	.23
2	.24	0	.15	.20	.14	.25
3	.22	.15	0	.15	.28	.11
4	.37	.20	.15	0	.29	.22
5	.34	.14	.28	.29	0	.39
6	.23	.25	.11	.22	.39	0

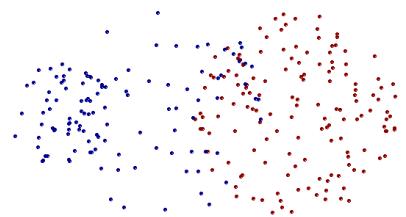
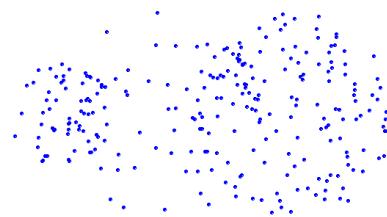


Strength of MIN



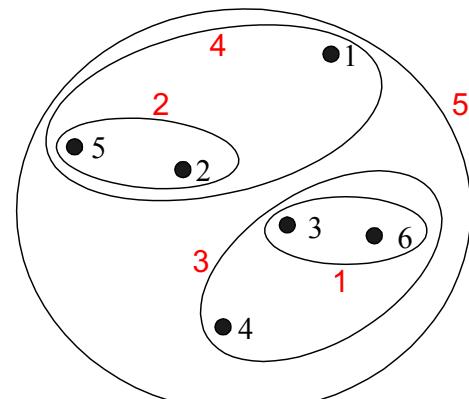
- Can handle non-elliptical shapes

Limitations of MIN



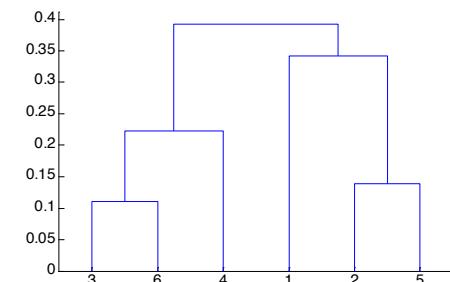
- Sensitive to noise and outliers

Hierarchical Clustering: MAX

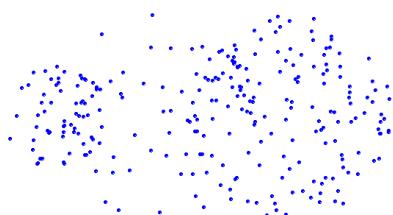


Dendrogram

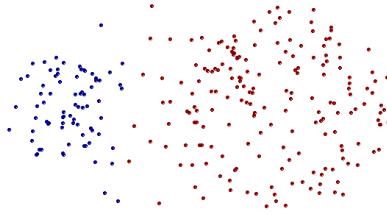
	1	2	3	4	5	6
1	0	.24	.22	.37	.34	.23
2	.24	0	.15	.20	.14	.25
3	.22	.15	0	.15	.28	.11
4	.37	.20	.15	0	.29	.22
5	.34	.14	.28	.29	0	.39
6	.23	.25	.11	.22	.39	0



Strength of MAX



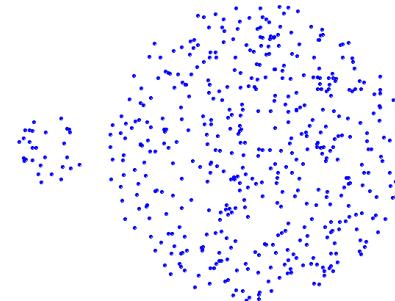
Original Points



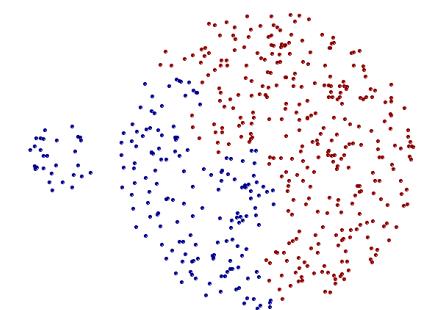
Two Clusters

- Less susceptible to noise and outliers

Limitations of MAX



Original Points



Two Clusters

- Tends to break large clusters
- Biased towards globular clusters

Cluster Similarity: Group Average

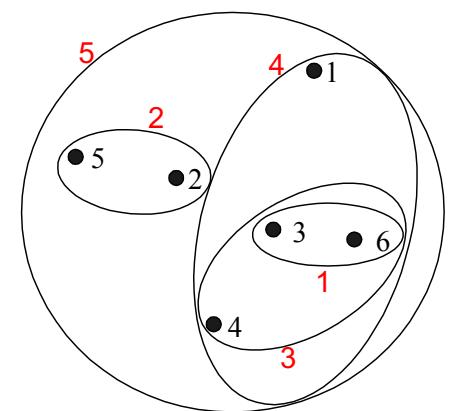
- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

- Need to use average connectivity for scalability since total proximity favors large clusters

	1	2	3	4	5	6
1	0	.24	.22	.37	.34	.23
2	.24	0	.15	.20	.14	.25
3	.22	.15	0	.15	.28	.11
4	.37	.20	.15	0	.29	.22
5	.34	.14	.28	.29	0	.39
6	.23	.25	.11	.22	.39	0

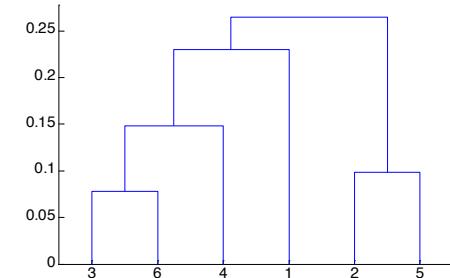
Hierarchical Clustering: Group Average



Nested Clusters

Dendrogram

	1	2	3	4	5	6
1	0	.24	.22	.37	.34	.23
2	.24	0	.15	.20	.14	.25
3	.22	.15	0	.15	.28	.11
4	.37	.20	.15	0	.29	.22
5	.34	.14	.28	.29	0	.39
6	.23	.25	.11	.22	.39	0



Hierarchical Clustering: Problems and Limitations

- Computational complexity in time and space
- Once a decision is made to combine two clusters, it cannot be undone
- No objective function is directly minimized
- Different schemes have problems with one or more of the following:
 - Sensitivity to noise and outliers
 - Difficulty handling different sized clusters and convex shapes
 - Breaking large clusters

DBSCAN

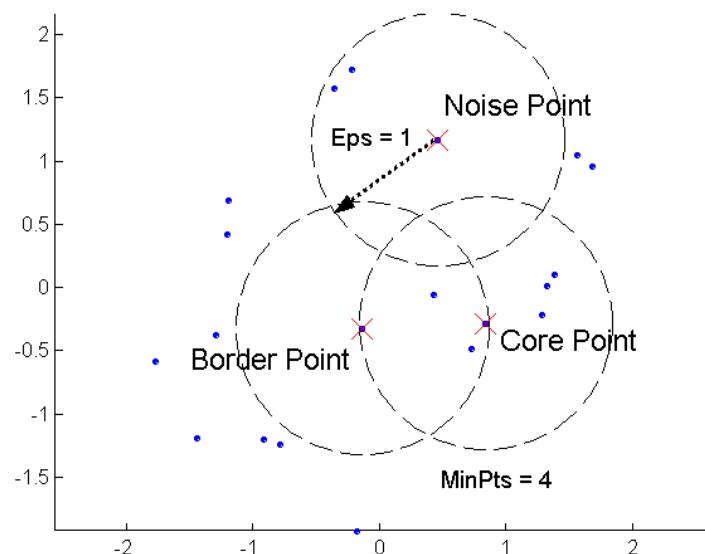
DBSCAN: Density-Based Clustering

- DBSCAN is a Density-Based Clustering algorithm
- Reminder: In density based clustering we partition points into dense regions separated by not-so-dense regions.
- Important Questions:
 - How do we measure density?
 - What is a dense region?
- DBSCAN:
 - Density at point p : number of points within a circle of radius Eps
 - Dense Region: A circle of radius Eps that contains at least $MinPts$ points

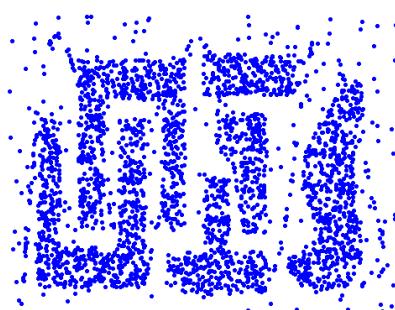
DBSCAN

- Characterization of points
 - A point is a **core point** if it has more than a specified number of points ($MinPts$) within Eps
 - These points belong in a **dense region** and are at the **interior** of a cluster
 - A **border point** has fewer than $MinPts$ within Eps , but is in the neighborhood of a **core** point.
 - A **noise point** is any point that is not a core point or a border point.

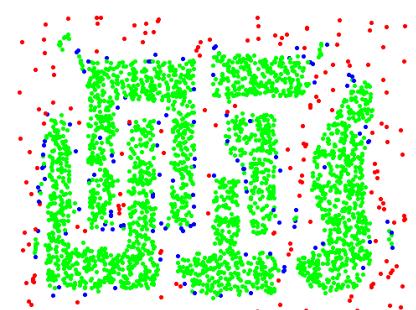
DBSCAN: Core, Border, and Noise Points



DBSCAN: Core, Border and Noise Points



Original Points



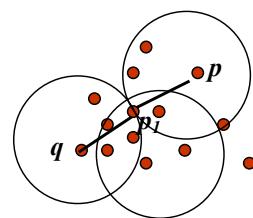
Point types: **core**,
border and **noise**

Eps = 10, MinPts = 4

Density-Connected points

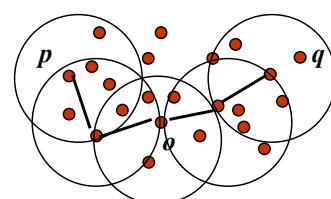
• Density edge

- We place an **edge** between two core points **q** and **p** if they are within distance **Eps**.



• Density-connected

- A point **p** is **density-connected** to a point **q** if there is a **path of edges** from **p** to **q**

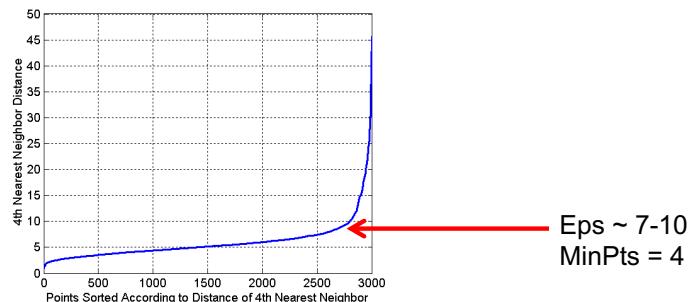


DBSCAN Algorithm

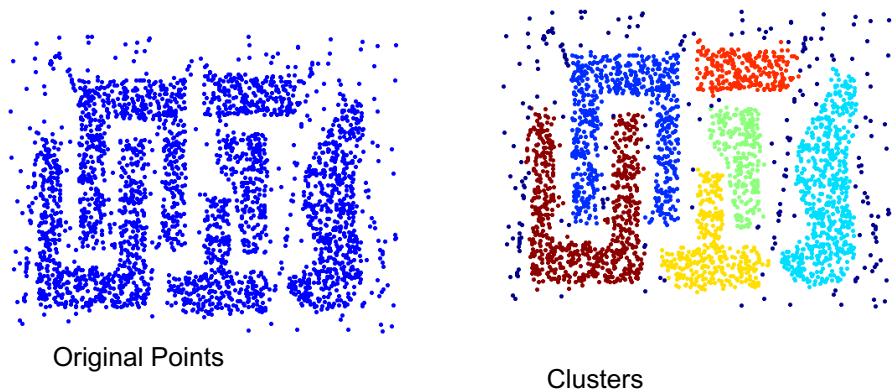
- Label points as **core**, **border** and **noise**
- Eliminate **noise** points
- For every **core** point **p** that has not been assigned to a cluster
 - Create a new cluster with the point **p** and all the points that are **density-connected** to **p**.
- Assign **border** points to the cluster of the closest core point.

DBSCAN: Determining Eps and MinPts

- Idea is that for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance
- Noise points have the k^{th} nearest neighbor at farther distance
- So, plot sorted distance of every point to its k^{th} nearest neighbor
- Find the distance d where there is a “knee” in the curve
 - $\text{Eps} = d$, $\text{MinPts} = k$

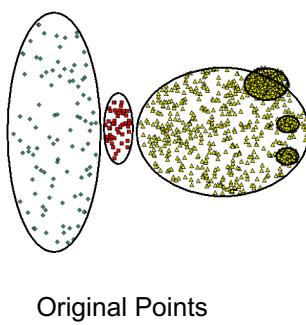


When DBSCAN Works Well

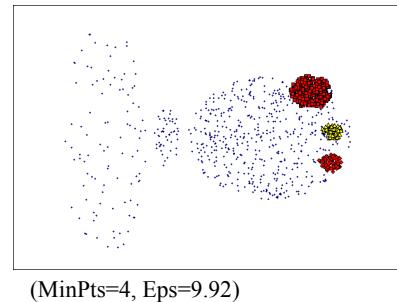


- Resistant to Noise
- Can handle clusters of different shapes and sizes

When DBSCAN Does NOT Work Well



- Varying densities
- High-dimensional data



DBSCAN: Sensitive to Parameters

Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

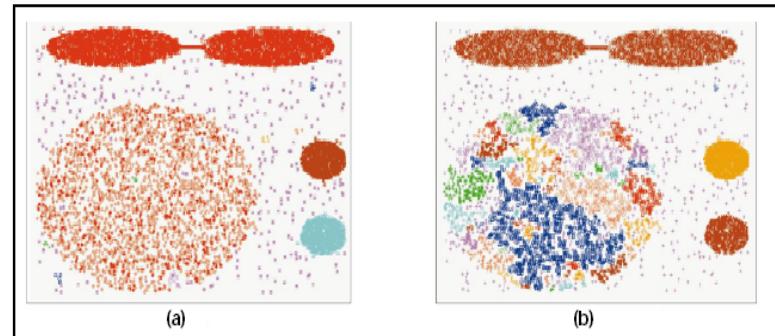
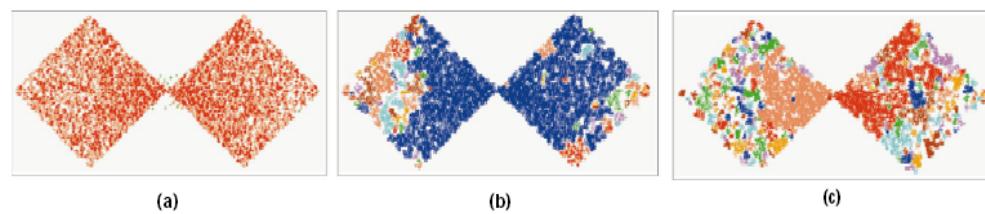


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.





Μηχανική μάθηση



Ενισχυτική μάθηση



- Ένα σύστημα (πράκτορας) αλληλοεπιδρά με το περιβάλλον εκτελώντας ενέργειες (**actions**) και λαμβάνοντας ανταμοιβές (**rewards**).



- Η ανταμοιβή μπορεί να μην είναι άμεσα διαθέσιμη και να προκύπτει στο τέλος κάποιας ακολουθίας ενέργειών, πχ. στο τέλος μιας παρτίδας παιχνιδιού (νίκη, ισοπαλία, ήττα).
- Το σύστημα μπορεί να είναι δυναμικό.
- Στόχοι μάθησης:
 - Να εκτιμηθεί η βέλτιστη πολιτική ενέργειών ώστε να μεγιστοποιηθεί η ανταμοιβή
 - Να εκτιμηθεί η κατανομή πιθανότητας των ανταμοιβών
 - Αν το σύστημα είναι δυναμικό να αποτιμηθεί η κατάστασή του ή να εκτιμηθεί η πιθανότητα μετάβασης στην επόμενη κατάσταση



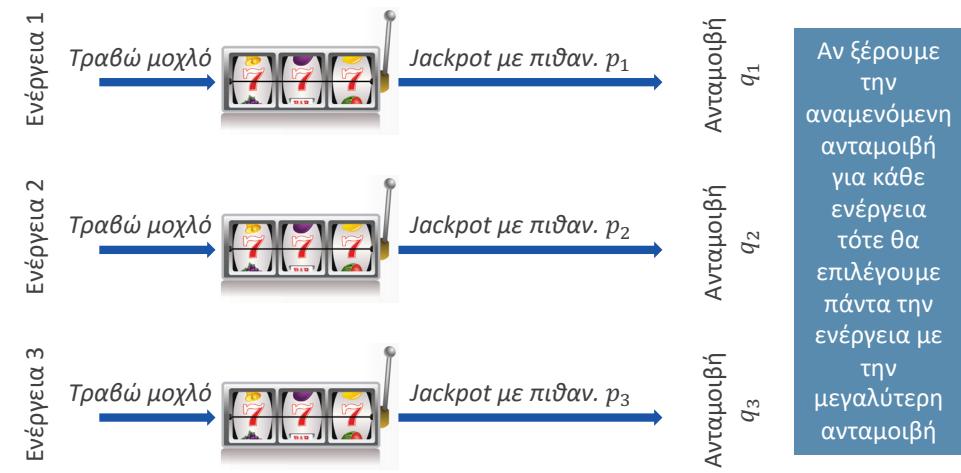
Περιγραφή προβλήματος μονόχειρων ληστών

- Νέα κλάση μεθόδων μάθησης διαφορετική από μάθηση με επίβλεψη ή τη μάθηση χωρίς επίβλεψη.
- Κεντρικές έννοιες:
 - Εξερεύνηση (Exploration):** δοκιμή πολλών διαφορετικών ενέργειών ώστε να καταγραφεί η αντίδραση του περιβάλλοντος
 - Εκμετάλλευση (Exploitation):** χρήση των εκτιμήσεών μας έτσι ώστε να επιλέξουμε τις καλύτερες ενέργειες
 - Δίλημμα εξερεύνησης/εκμετάλλευσης
- Εφαρμογές:**
 - Ρομποτική
 - Λήψη αποφάσεων
 - Παιχνίδια

- Το απλούστερο πρόβλημα ενισχυτικής μάθησης.
 - Στατικό περιβάλλον χωρίς μνήμη
 - Ορισμός: Κάθε χρονική στιγμή t , σε μια χρονική ακολουθία $t = 1, \dots, T$, πρέπει να επιλέξουμε 1 ανάμεσα από k πιθανές **ενέργειες**. Έστω $A(t)$ η ενέργεια που επιλέγουμε τη στιγμή t .
 - Κάθε ενέργεια a δίνει μια ανταμοιβή q που εξαρτάται μόνο από το a (και όχι από το t). Έστι έχουμε μια ακολουθία ανταμοιβών $R(t)$:
$$A(t) = a \Rightarrow R(t) = q$$
 - Η ανταμοιβή q είναι μια τυχαία μεταβλητή με άγνωστη κατανομή.
 - Ζητάμε την **αναμενόμενη ανταμοιβή** με δεδομένο το a :
- $$\mu(a) = E\{R(t)|A(t) = a\} = \int_q P(q|a)dq$$
- Στόχος είναι να επιλέξουμε μια ακολουθία ενέργειών ώστε να μεγιστοποιηθεί η συνολική αναμενόμενη ανταμοιβή για τη χρονική περίοδο T .



- Ιατρικές εφαρμογές:** Έστω ότι έχουμε k θεραπείες για την ίδια ασθένεια.
 - Θέλουμε να βρούμε την καλύτερη θεραπεία για το μέσο πληθυσμό
 - Δεν θέλουμε να δώσουμε κακές ή όχι βέλτιστες θεραπείες σε πολλούς ασθενείς
- Συστήματα συστάσεων:** Θέλουμε να συστήσουμε k προϊόντα σε χρήστες.
 - Θέλουμε να βρούμε την καλύτερη σύσταση για το μέσο πληθυσμό
 - Δεν θέλουμε να κάνουμε πολλές κακές συστάσεις στους χρήστες
- Δρομολόγηση (Δίκτυα υπολογιστών):** Έχουμε k δυνατές διαδρομές για ένα μήνυμα
 - Θέλουμε να βρούμε την καλύτερη διαδρομή κατά μέσο όρο
 - Δεν θέλουμε να κάνουμε πολλές κακές δοκιμές



Εξερεύνηση εναντίον εκμετάλλευσης

- Εξερεύνηση:** Καθώς αρχικά δεν ξέρουμε την κατανομή πιθανότητας για την ανταμοιβή καμίας ενέργειας, πρέπει να την εκτιμήσουμε κάνοντας δοκιμές
- Εκμετάλλευση:** Αφού εκτιμήσουμε τις πιθανότητες ανταμοιβών εκμεταλλεύόμαστε τη γνώση αυτή για να κάνουμε την καλύτερη επιλογή ενέργειας.
- Η Εξερεύνηση είναι απαραίτητη ώστε να συλλεχθούν στατιστικά για τις ανταμοιβές των ενεργειών και να εκτιμηθεί η ενέργεια με την μεγαλύτερη ανταμοιβή. Από την άλλη μεριά, κατά τη διάρκεια της Εξερεύνησης μπορεί να δοκιμάζουμε μη βέλτιστες ενέργειες.
- Το δίλημμα Εξερεύνηση/Εκμετάλλευση:** χωρίς εξερεύνηση κάνουμε ενέργειες στα τυφλά. Εφαρμόζοντας πολλή Εξερεύνηση κάνουμε πολλές κακές ή μη βέλτιστες ενέργειες.



Απλοϊκή προσέγγιση

- Εξερευνούμε κάθε ενέργεια a ένα συγκεκριμένο πλήθος φορών έτσι ώστε να εκτιμήσουμε την αναμενόμενη ανταμοιβή $\hat{m}(a)$. Κατόπιν, θα επιλέγουμε συνεχώς την ενέργεια με την μεγαλύτερη αναμενόμενη ανταμοιβή:
 - Διάλεξε μια ενέργεια a επανειλημμένα N φορές
 - Σύλλεξε στατιστικά και εκτίμησε την κατανομή $R(i)$ της ανταμοιβής γι' αυτή την ενέργεια
 - Εκτίμησε την αναμενόμενη ανταμοιβή για την ενέργεια a :

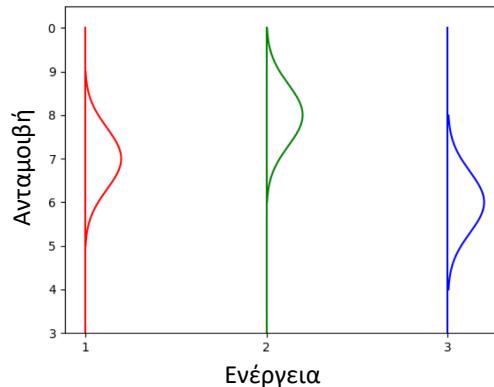
$$\hat{m}(a) = \frac{\text{Άθροισμα ανταμοιβών } R(i) \text{ όταν επιλέγουμε } a}{\text{Πλήθος φορών που επιλέξαμε } a}$$
 - Αφού υπολογίσουμε όλα τα $\hat{m}(1), \dots, \hat{m}(k)$, επιλέγουμε πάντα την ενέργεια a^* με τη μεγαλύτερη αναμενόμενη ανταμοιβή $\hat{m}(a^*)$.
- Σημαντική παρατήρηση:** έχουμε υποθέσει ότι η στατιστική συμπεριφορά των ανταμοιβών δεν εξαρτάται από το χρόνο t αλλά μόνο από την ενέργεια a .



Παράδειγμα: 3 bandits

- Έστω ότι έχουμε 3 δυνατές ενέργειες $a = 0$, $a = 1$, $a = 2$, με ανταμοιβές που ακολουθούν την Γκαουσσιανή κατανομή

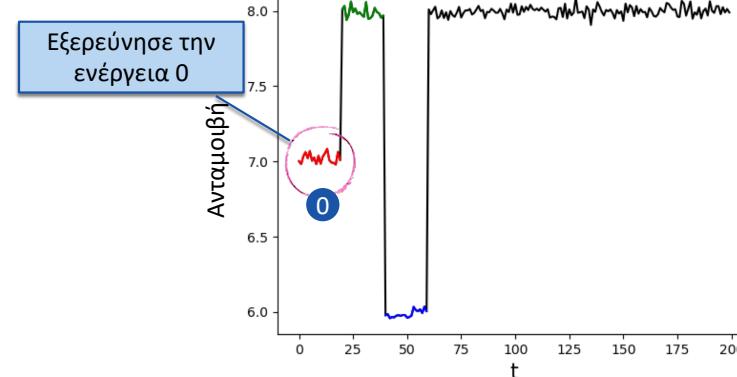
- Ανταμοιβή για $a = 0$:
 $q(0) \sim N(\mu = 7, \sigma^2 = 1)$
- Ανταμοιβή για $a = 1$:
 $q(1) \sim N(\mu = 8, \sigma^2 = 1)$
- Ανταμοιβή για $a = 2$:
 $q(2) \sim N(\mu = 6, \sigma^2 = 1)$



Παράδειγμα: 3 bandits

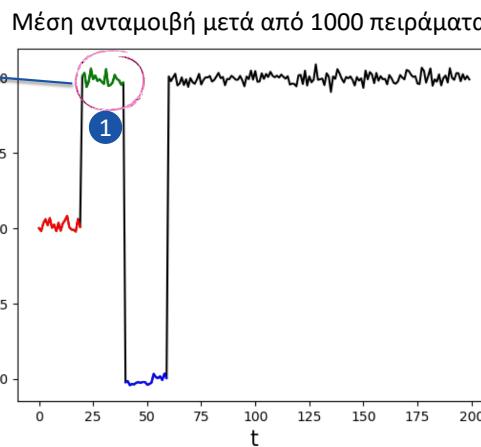
- Απλοϊκή προσέγγιση: Εξερεύνησε κάθε ενέργεια για 20 χρονικές στιγμές κάθε μια

Μέση ανταμοιβή μετά από 1000 πειράματα



Παράδειγμα: 3 bandits

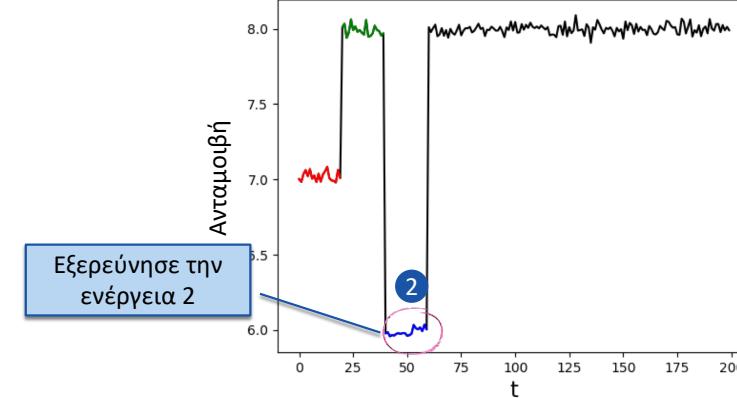
- Απλοϊκή προσέγγιση: Εξερεύνησε κάθε ενέργεια για 20 χρονικές στιγμές κάθε μια



Παράδειγμα: 3 bandits

- Απλοϊκή προσέγγιση: Εξερεύνησε κάθε ενέργεια για 20 χρονικές στιγμές κάθε μια

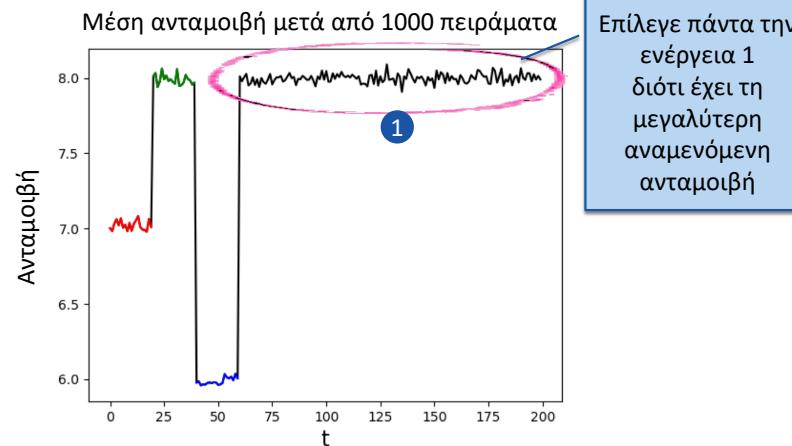
Μέση ανταμοιβή μετά από 1000 πειράματα





Παράδειγμα: 3 bandits

- Απλοϊκή προσέγγιση: Εξερεύνησε κάθε ενέργεια για 20 χρονικές στιγμές κάθε μια



Άπληστος αλγόριθμος

- Εναλλακτικά μπορούμε να εκμεταλλευόμαστε αμέσως τη γνώση που έχουμε συλλέξει μέχρι στιγμής.
- Αυτό σημαίνει ότι έχουμε μια εκτίμηση $\hat{\mu}_t(a)$ της αναμενόμενης ανταμοιβής για κάθε ενέργεια a τη στιγμή t με βάση τις μέχρι τώρα παρατηρήσεις μας.

$$\hat{\mu}_t(a) = \frac{\text{Άθροισμα ανταμοιβών } R(i) \text{ όταν επιλέγω } a \text{ πριν τη στιγμή } t}{\text{Πλήθος φορών που επέλεξα } a \text{ πριν τη στιγμή } t}$$

$$= \frac{\sum_{i=1, A(i)=a}^{t-1} R(i)}{n_{t-1}(a)}$$

- Σε κάθε χρονική t στιγμή επιλέγω την ενέργεια με τη μέγιστη εκτιμώμενη ανταμοιβή $\hat{\mu}_t(a)$.
- Αφού συλλέξουμε την ανταμοιβή $R(t)$, ενημερώνουμε την εκτίμησή μας $\hat{\mu}_{t+1}(a)$ για την επιλεγμένη ενέργεια $A(t) = a$:

$$\hat{\mu}_{t+1}(a) = \frac{\sum_{i=1}^t R(i)}{n_t(a)}$$

- Η μέθοδος καλείται **άπληστη** διότι κάθε στιγμή επιλέγουμε την δράση με την μέγιστη εκτιμώμενη ανταμοιβή αμέσως μόλις ενημερωθούν οι εκτιμήσεις μας.

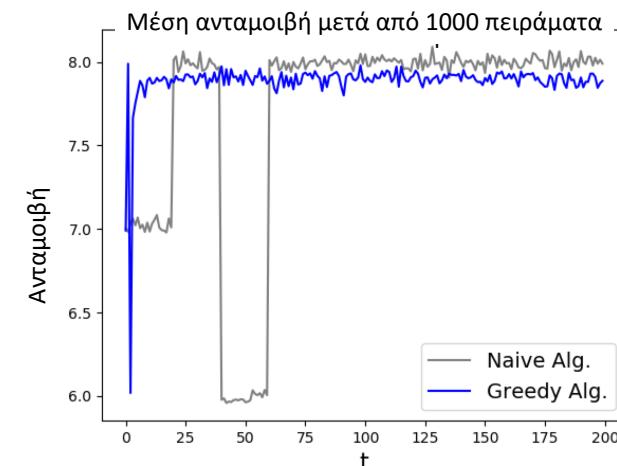


Επαναληπτική μέθοδος

- Ξεκίνα με κάποια αρχική εκτίμηση $\hat{\mu}(a)$ και θέσε $n(a) = 0 =$ μετρητής φορών που επιλέγω την ενέργεια a .
 - Για κάθε χρονική στιγμή t :
 - Επίλεξε την ενέργεια $A(t) = a^*$ με το μέγιστο $\hat{\mu}(a)$:
- $$a^* = \arg \max_a \hat{\mu}(a)$$
- (σε περίπτωση ισοπαλίας επέλεξε τυχαία)
- Κατάγραψε την ανταμοιβή $R(t)$
 - Ενημέρωσε τον μετρητή:
- $$n(a^*) \leftarrow n(a^*) + 1$$
- Ενημέρωσε την αναμενόμενη ανταμοιβή για την ενέργεια a^* :
- $$\hat{\mu}(a^*) \leftarrow \frac{n(a^*) - 1}{n(a^*)} \hat{\mu}(a^*) + \frac{1}{n(a^*)} R(t)$$
- $$= \hat{\mu}(a^*) + \frac{1}{n(a^*)} [R(t) - \hat{\mu}(a^*)]$$



Απλοϊκός εναντίον Άπληστου αλγόριθμου





- Μειονέκτημα** του άπληστου αλγορίθμου: Επιλέγοντας πάντα την ενέργεια με τη μεγαλύτερη εκτιμώμενη ανταμοιβή $\hat{\mu}(a)$ μπορεί να αγνοούμε ενέργειες που δεν έχουμε δει μέχρι στιγμής → **ελλιπής εξερεύνηση!**
- Μια άλλη εναλλακτική μέθοδος είναι με πιθανότητα $(1 - \varepsilon)$ να επιλέγουμε ενέργειες με την άπληστη μέθοδο, ενώ με πιθανότητα ε να επιλέγουμε ενέργειες τυχαία ώστε να δίνουμε στο σύστημα την δυνατότητα να εξερευνάει κι άλλες ενέργειες. Αυτή η μέθοδος είναι γνωστή ως **ε-άπληστος αλγόριθμος**.
- Προφανώς ο άπληστος αλγόριθμος είναι ειδική περίπτωση του ε -άπληστου αλγορίθμου για $\varepsilon = 0$.



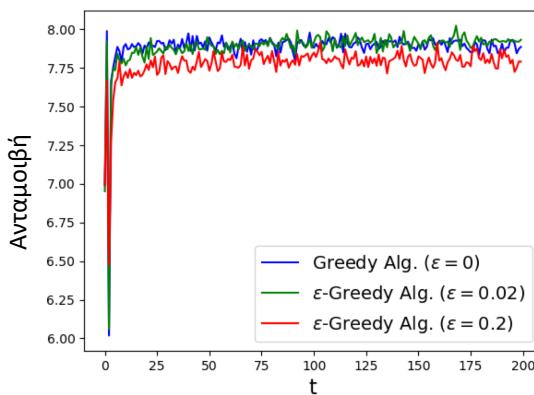
- Ξεκίνα με $\hat{\mu}(a) = 0$ και $n(a) = 0$
 - Για κάθε χρονική στιγμή t :
 - Με πιθανότητα $1 - \varepsilon$ επίλεξε την ενέργεια $A(t) = a = \arg \max_{a'} \hat{\mu}(a')$
 - Με πιθανότητα ε επίλεξε την ενέργεια $A(t) = a = \text{τυχαία}$
 - Σύλλεξε την ανταμοιβή $R(t)$
 - Ενημέρωσε τον μετρητή:
$$n(a) \leftarrow n(a) + 1$$
 - Ενημέρωσε την αναμενόμενη ανταμοιβή για την ενέργεια a :
- $$\hat{\mu}(a) \leftarrow \frac{n(a^*) - 1}{n(a^*)} \hat{\mu}(a) + \frac{1}{n(a)} R(t) = \hat{\mu}(a) + \frac{1}{n(a)} [R(t) - \hat{\mu}(a)]$$



Παράδειγμα: 3 μονόχειρες ληστές

Επίδραση του ε στη μάθηση:

- $\varepsilon = 0$ (άπληστος)
Όχι καλό αποτέλεσμα
- Μεγαλύτερο ε →*
Περισσότερη εξερεύνηση
→ βελτίωση επίδοσης
καθώς το t μεγαλώνει.
- Πολύ μεγάλο ε →*
Πολλή εξερεύνηση
→ δεν βελτιώνει
την επίδοση.



Δυσαρέσκεια (Regret)

- Η **Δυσαρέσκεια (Regret)** είναι ένα κριτήριο βελτιστοποίησης για την επιλογή της καλύτερης ενέργειας.
 - Ορίζεται ως το συνολικό άθροισμα των διαφορών μεταξύ της ανταμοιβής $R(a^*)$ από τη βέλτιστη ενέργεια a^* και της ανταμοιβής $R(A(t))$ από την ενέργεια $A(t)$ που επιλέχτηκε:
- $$r_T = E\{\sum_{t=1}^T R(a^*) - R(A(t))\} = \sum_{t=1}^T E\{R(a^*) - R(A(t))\} = \sum_{t=1}^T \mu(a^*) - \mu(A(t))$$

- Έχοντας K δυνατές ενέργειες $a \in \{1, \dots, K\}$ η παραπάνω έκφραση γράφεται

$$r_T = \sum_{a=1}^K (\mu(a^*) - \mu(a)) n(a)$$

όπου $n(a)$ είναι ο γνωστός μετρητής φορών που επιλέξαμε την ενέργεια a .



- Ιδέα:** Τη στιγμή t επίλεξε την ενέργεια a με βάση το συνδυασμό της έως τώρα εκτιμώμενης ανταμοιβής $\hat{\mu}_t(a)$ και της **αβεβαιότητας** της εκτίμησής μας η οποία προκύπτει από το πόσες φορές έχουμε εκτελέσει αυτή την ενέργεια στο παρελθόν.
- Αν η ενέργεια a δεν έχει δοκιμαστεί πολλές φορές μέχρι τώρα, τότε **δεν είμαστε πολύ βέβαιοι** για την εκτίμηση $\hat{\mu}_t(a)$. Στην περίπτωση αυτή δίνουμε μεγάλο “**bonus**” στην ενέργεια αυτή αυξάνοντας την πιθανότητα να την επιλέξουμε. Έτσι, διευκολύνουμε την εξερεύνηση.
- Το bonus είναι αντιστρόφως ανάλογο του μετρητή $n_t(a)$: όσες πιο πολλές φορές έχουμε εκτελέσει την ενέργεια a τόσο λιγότερο αβέβαιοι είμαστε για την εκτίμησή μας.
- Αν μια ενέργεια έχει δοκιμαστεί πολλές φορές και έχει μικρή εκτιμώμενη ανταμοιβή $\hat{\mu}_t(a)$ τότε δεν θα επιλεγεί.
- Το bonus ενθαρρύνει την εξερεύνηση αλλά δεν πρέπει να είναι τόσο μεγάλο ώστε να επιλέγουμε ενέργειες που δεν έχουν καμία ελπίδα να είναι βέλτιστες.



- Παράδειγμα:** Η ενέργεια a_1 δοκιμάστηκε 100 φορές και έχει εκτιμώμενη ανταμοιβή $\hat{\mu}_t(a_1) = 10$. Λόγω μικρής αβεβαιότητας έχει μικρό bonus: $B_1 = 1$
 - Η ενέργεια a_2 δοκιμάστηκε μόνο 5 φορές και έχει εκτιμώμενη ανταμοιβή $\hat{\mu}_t(a_2) = 9$. Λόγω μεγάλης αβεβαιότητας έχει μεγάλο bonus: $B_2 = 3$.
 - Η πολιτική του αλγορίθμου είναι να επιλέξουμε την ενέργεια με το μεγαλύτερο άθροισμα $\hat{\mu}_t + B$.
- $$\hat{\mu}(a_1) + B_1 = 11, \quad \hat{\mu}(a_2) + B_2 = 12$$
- Έτσι, αν και η ενέργεια a_2 έχει μικρότερη αναμενόμενη ανταμοιβή την επιλέγουμε λόγω μεγαλύτερης αβεβαιότητας.
 - Αν ωστόσο, η αναμενόμενη ανταμοιβή της a_2 ήταν πολύ μικρή, πχ $\hat{\mu}_t(a_2) = 1$, τότε ακόμη και με το bonus δεν θα την επιλέγαμε διότι είναι πολύ χειρότερη από την a_1 . Έτσι δεν σπαταλάμε εξερεύνηση σε ενέργειες που δεν έχουν καμία ελπίδα.



Αλγόριθμος UCB1

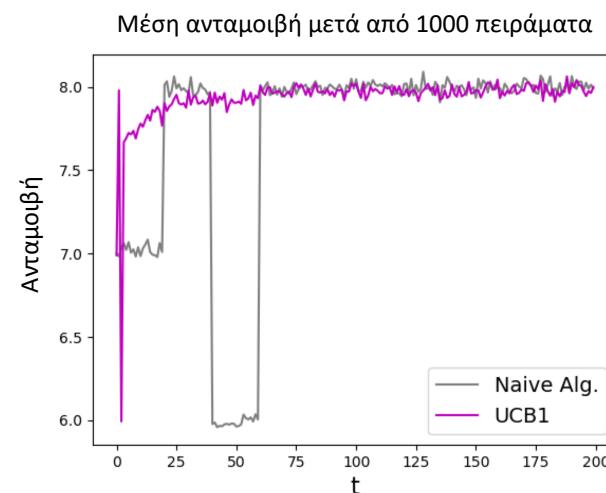
- Αρχικοποίηση:** Δοκίμασε κάθε ενέργεια από μία φορά. Θέσε αρχικό $\hat{\mu}_1(a)=$ ανταμοιβή για την ενέργεια a .
- Για $t = 1 \dots T$

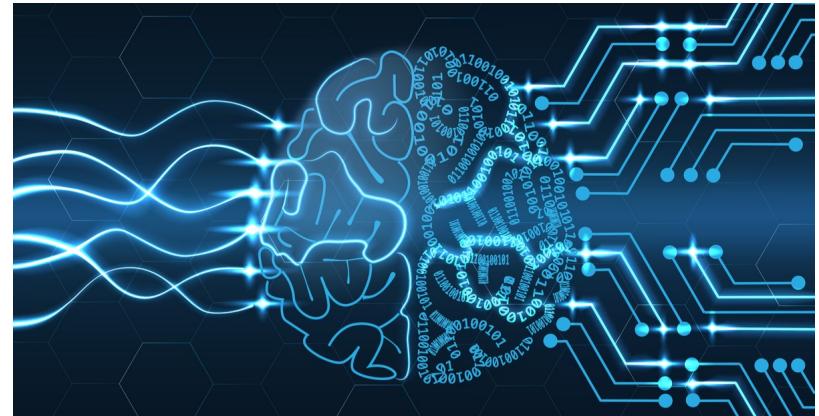
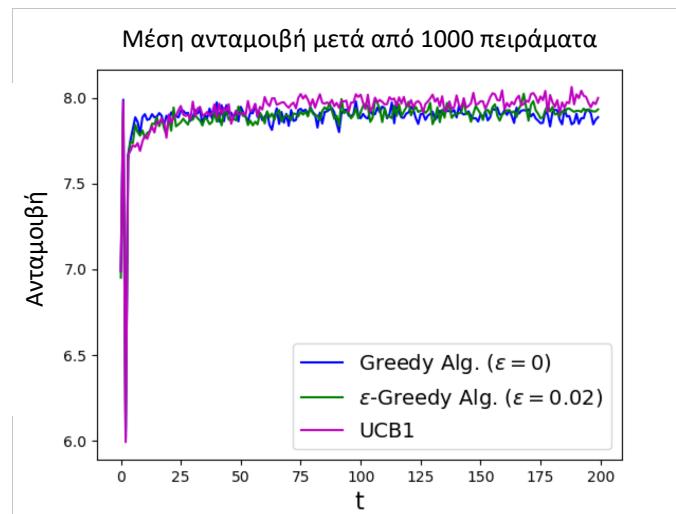
Επίλεξε την ενέργεια a με το μέγιστο άθροισμα

$$\hat{\mu}_t(a) + \sqrt{\frac{2 \ln t}{n_t(a)}} \quad \text{Bonus}$$

όπου $\hat{\mu}_t(a)$ είναι η εκτιμώμενη ανταμοιβή μέχρι τη στιγμή t
 $n_t(a)$ είναι ο μετρητής των φορών που επιλέξαμε την ενέργεια a μέχρι τη στιγμή t

P. Auer, N. Cesa-Bianchi, P. Fischer, “Finite-time Analysis of the Multiarmed Bandit Problem”, Machine Learning, 47, 235–256, 2002





Markov Decision Processes (MDP)

Αλληλεπίδραση του πράκτορα με το περιβάλλον

Βασικές Υποθέσεις:

- Ένας πράκτορας παίρνει αποφάσεις επιλέγοντας ανάμεσα σε K δυνατές ενέργειες
- Ο πράκτορας αλληλεπιδρά με το περιβάλλον σε διακριτές χρονικές στιγμές t .
- Το περιβάλλον βρίσκεται πάντα σε μια από N δυνατές καταστάσεις.
- Ανάλογα με την ενέργεια $A(t)$ και την κατάσταση $S(t)$ τη στιγμή t ο πράκτορας λαμβάνει μια ανταμοιβή $R(t)$ και το περιβάλλον μεταβαίνει στην κατάσταση $S(t + 1)$.
- Ορίζουμε την πιθανότητα το σύστημα να βρίσκεται στην κατάσταση s' δίνοντας ανταμοιβή r τη στιγμή $t + 1$ δεδομένου ότι τη στιγμή t ήταν στην κατάσταση s και η επιλεγμένη ενέργεια ήταν η a :
- $p(s', r | s, a) = \Pr(S(t + 1) = s', R(t + 1) = r | S(t) = s, A(t) = a)$



Παράδειγμα: Ρομπότ ανακύκλωσης

- Ένα ρομπότ συλλέγει άδεια κουτάκια αναψυκτικών. Οι δυνατές ενέργειες του ρομπότ είναι:

$$\mathcal{A} = \{"\text{search for cans}", "\text{wait}", "\text{recharge}"\}$$
 - Η μπαταρία του ρομπότ είναι στην κατάσταση "low" ή "high".
 - Η ανταμοιβή του ρομπότ για την ενέργεια "search" είναι υψηλή, πχ. $r_{\text{search}} = 10$ αφού ο σκοπός του είναι η αναζήτηση άδειων κουτιών.
 - Η ανταμοιβή του ρομπότ για την ενέργεια "wait", είναι μικρότερη αλλά όχι μηδέν, πχ. $r_{\text{wait}} = 2$
- Ο λόγος είναι ότι μετά από κάποιο χρόνο αναζήτησης το ρομπότ μπορεί να έχει μαζέψει τα περισσότερα κουτιά και είναι καλύτερο να περιμένει για να μαζευτούν και άλλα αντί να συνεχίζει την αναζήτηση.



Παράδειγμα: Ρομπότ ανακύκλωσης

- Πίνακας πιθανοτήτων μετάβασης ("Όχι "recharge" όταν κατάσταση="High")

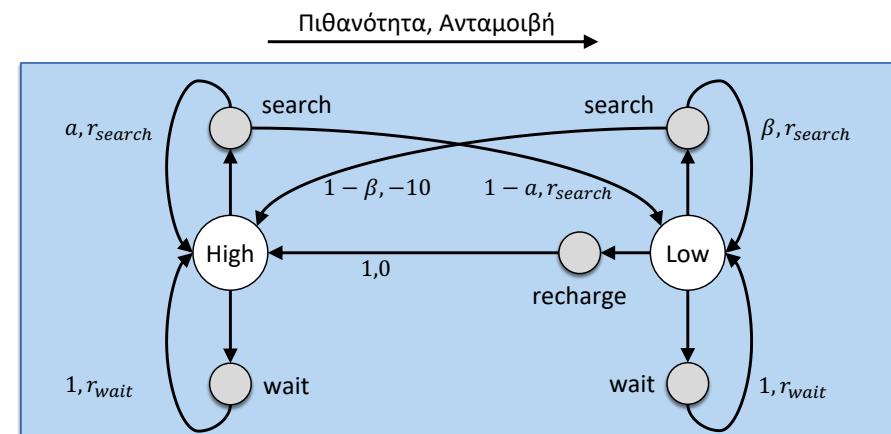
$S(t)$	$A(t)$	$S(t + 1)$	$p(s' s, a)$	$R(t)$
High	Search	High	α	r_{search}
High	Search	Low	$1 - \alpha$	r_{search}
Low	Search	High	$1 - \beta$	-10
Low	Search	Low	β	r_{search}
High	Wait	High	1	r_{wait}
High	Wait	Low	0	r_{wait}
Low	Wait	High	0	r_{wait}
Low	Wait	Low	1	r_{wait}
Low	Recharge	High	1	0
Low	Recharge	Low	0	0

Πρέπει να σώσουμε το ρομπότ και να το φορτίσουμε

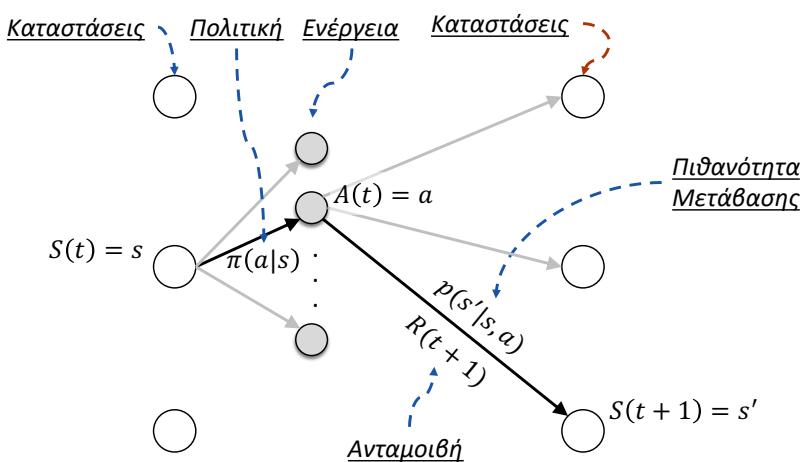


Παράδειγμα: Ρομπότ ανακύκλωσης

- Γράφος μετάβασης. = Κόμβος κατάστασης = Κόμβος ενέργειας



Γράφος μετάβασης



Πολιτική, επεισοδιακές και συνεχείς εργασίες

- Ο αλγόριθμος επιλογής ενέργειών με βάση την τρέχουσα κατάσταση του συστήματος και την προηγούμενη εμπειρία καλείται **πολιτική (policy)**
- Με δεδομένη την πολιτική επιλέγουμε μια σειρά ενέργειών $A(t)$ και συλλέγουμε μια σειρά ανταμοιβών $R(t)$, για $t = 1, \dots, T$.
- Σε μερικές περιπτώσεις το T είναι πεπερασμένος αριθμός. Για παράδειγμα, όταν ένα ρομπότ βγαίνει από ένα λαβύρινθο το κάνει σε πεπερασμένο πλήθος βημάτων. Παρομοίως όταν παίζουμε ένα παιχνίδι αυτό τελειώνει σε πεπερασμένο πλήθος κινήσεων. Τέτοιες περιπτώσεις καλούνται **επεισοδιακές εργασίες** ή **εργασίες πεπερασμένου ορίζοντα**.
- Σε άλλες περιπτώσεις, ωστόσο, είναι ορθό να υποθέσουμε ότι $T = \infty$. Για παράδειγμα ένα βιομηχανικό ρομπότ μπορεί να εκτελεί την ίδια εργασία αδιάλειπτα, ή ένας ελεγκτής πρέπει να εφαρμόζει συνεχώς έλεγχο σε μια συσκευή ή ένα σύστημα. Τέτοιες περιπτώσεις καλούνται **συνεχείς εργασίες** ή **εργασίες άπειρου ορίζοντα**.



- Ένα μέτρο της επίδοσης μιας πολιτικής κατά το χρόνο t είναι το άθροισμα των ανταμοιβών που ελήφθησαν μετά το t χρησιμοποιώντας αυτή την πολιτική.
 - Ωστόσο, δεδομένου ότι το T μπορεί να είναι άπειρο, το άθροισμα μπορεί να αποκλίνει. Μια λύση στο πρόβλημα είναι να εισαχθεί ένας θετικός συντελεστής λήθης (ή έκπτωσης) $\gamma < 1$ που μειώνει εκθετικά τη σημασία των ανταμοιβών $R(t+k)$ καθώς $k \rightarrow \infty$
- $$G(t) = R(t+1) + \gamma R(t+2) + \gamma^2 R(t+3) + \gamma^3 R(t+4) + \dots$$
- $$= \sum_{k=0}^{\infty} \gamma^k R(t+k+1)$$
- Αυτό το κριτήριο ονομάζεται **κέρδος** κατά το χρόνο t . Εάν οι ανταμοιβές οριοθετούνται μεταξύ πεπερασμένων ορίων, το κέρδος θα συγκλίνει πάντα σε μια πεπερασμένη τιμή.
 - Προφανώς το κέρδος υπακούει στον αναδρομικό τύπο
- $$G(t) = R(t+1) + \gamma G(t+1).$$



- Η πολιτική (*policy*) π καθορίζεται πλήρως από την πιθανότητα ανάληψης της ενέργειας a με δεδομένη μια κατάσταση s :
$$\pi(a, s) = p(A(t) = a | S(t) = s)$$
- Ορίζουμε ως **αξία (value)** μιας κατάστασης s (στο πλαίσιο μιας πολιτικής π) το αναμενόμενο κέρδος αν ξεκινήσουμε από την κατάσταση s και ακολουθήσουμε την π :
$$v_\pi(s) = E_\pi\{G(t) | S(t) = s\}$$
- Ο συμβολισμός $E_\pi\{\cdot\}$ δηλώνει την αναμενόμενη τιμή μιας τυχαίας μεταβλητής, δεδομένου ότι ο πράκτορας ακολουθεί την πολιτική π σε κάθε βήμα στιγμή t .
- Σημειώστε ότι η συνάρτηση αξίας $v_\pi(s)$ δεν εξαρτάται από το t . Υποθέτουμε ότι το μοντέλο είναι στατικό, δηλαδή τα στατιστικά στοιχεία των ανταμοιβών καθώς και οι πιθανότητες μετάβασης από κατάσταση σε κατάσταση δεν αλλάζουν με το χρόνο.



Εξίσωση Bellman για την συνάρτηση v

- Γενική αναδρομική σχέση:
$$v_\pi(s) = \sum_a \pi(a, s) \sum_{r, s'} p(r, s' | s, a) [r + \gamma v_\pi(s')]$$
- Στην ειδική περίπτωση που η ανταμοιβή r είναι ντετερμινιστική συνάρτηση του ζεύγους s, a , τότε η σχέση απλοποιείται:
$$v_\pi(s) = \sum_a \pi(a, s) \left[r + \gamma \sum_{s'} p(s' | s, a) v_\pi(s') \right]$$
- Αν επί πλέον η επόμενη κατάσταση s' είναι επίσης ντετερμινιστική συνάρτηση του ζεύγους s, a , τότε :
$$v_\pi(s) = \sum_a \pi(a, s) [r + \gamma v_\pi(s')]$$
- Τέλος, αν η πολιτική μας είναι ντετερμινιστική επιλογή μιας ενέργειας a ως συνάρτηση της κατάστασης s τότε:
$$v_\pi(s) = r + \gamma v_\pi(s')$$



Βασικό πρόβλημα

- Θέλουμε να βρούμε τη βέλτιστη πολιτική π που μεγιστοποιεί την αναμενόμενο κέρδος ξεκινώντας από οποιαδήποτε αρχική κατάσταση s .
- Με άλλα λόγια, η συνάρτηση ποιότητας που πρέπει να μεγιστοποιηθεί είναι:
$$J(s) = E_\pi\{G(0) | S(0) = s\} = v_\pi(s)$$
- Η βέλτιστη πολιτική για την κατάσταση s είναι:
$$\pi^*(s) = \arg \max_\pi v_\pi(s)$$
- Και η βέλτιστη αξία της κατάστασης s είναι:
$$v^*(s) = \max_\pi v_\pi(s) = v_{\pi^*}(s)$$



- Για την επίλυση του προβλήματος της βέλτιστης πολιτικής, είναι χρήσιμο να ορίσουμε τη συνάρτηση

$$q_\pi(s, a) = E_\pi\{G(t) \mid S(t) = s, A(t) = a\}$$

δηλαδή, το αναμενόμενο κέρδος αν ξεκινήσουμε από την κατάσταση s και ακολουθήσουμε τις ενέργειες a που υποδεικνύει η πολιτική π .

- Η $q_\pi(\cdot)$ είναι γνωστή ως **συνάρτηση-Q (Q-function)**
- Αποδεικνύεται ότι υπάρχουν οι εξής σχέσεις μεταξύ q_π και v_π

$$q_\pi(s, a) = \sum_{s', r} p(s', r|s, a)[r + \gamma v_\pi(s')]$$

$$v_\pi(s) = \sum_a \pi(a|s) q_\pi(s, a)$$

- Δεδομένης της σχέσης μεταξύ $v_\pi(s)$ και $q_\pi(s, a)$: $v_\pi(s) = \sum_a \pi(a|s) q_\pi(s, a)$ μπορούμε να καταλήξουμε σε ένα σημαντικό συμπέρασμα όσον αφορά τη βέλτιστη πολιτική:
- Αν υποθέσουμε ότι $q_{\pi^*}(s, a)$ είναι η συνάρτηση Q υπό τη βέλτιστη πολιτική π^* , τότε η συνάρτηση $v_\pi(s)$ μεγιστοποιείται αν

$$\pi^*(a|s) = \begin{cases} 1 & \text{αν } a = \arg \max_{a'} (q_{\pi^*}(s, a')) \\ 0 & \text{Διαφορετικά} \end{cases}$$

- Με άλλα λόγια, η βέλτιστη πολιτική για μια δεδομένη κατάσταση s είναι να επιλέγουμε πάντα την ενέργεια a που δίνει το μέγιστο $q_{\pi^*}(s, a)$. Επομένως:

$$v_{\pi^*}(s) = \max_{a'} q_{\pi^*}(s, a')$$



Αναδρομή υπό βέλτιστη πολιτική

- Εάν η πολιτική είναι βέλτιστη, τότε

$$q_{\pi^*}(s, a) = r(s, a) + \gamma \sum_{s'} p(s'|s, a) v_{\pi^*}(s')$$

$$q_{\pi^*}(s, a) = r(s, a) + \gamma \sum_{s'} p(s'|s, a) \max_{a'} q_{\pi^*}(s', a')$$

- Ο τύπος ονομάζεται **εξίσωση βελτιστοποίησης Bellman**.
- Είναι ένας αναδρομικός τύπος για το q .
- Σημειώστε ότι ο όρος $r(s, a)$ δεν εξαρτάται από την πολιτική π και μπορεί να υπολογιστεί εκ των προτέρων για κάθε ζεύγος κατάστασης s και ενέργειας a .



Dynamic Programming



- Πρόβλημα: Δεδομένης της πολιτικής $\pi(a|s)$ να υπολογιστούν οι αξίες των καταστάσεων $v_\pi(s)$.

$$\text{Θυμόμαστε τη σχέση: } v_\pi(s) = \sum_a \pi(a|s) q_\pi(s, a)$$

• Και επίσης

$$\begin{aligned} q_\pi(s, a) &= E\{R(t+1)| S(t) = s, A(t) = a\} + \gamma \sum_{s'} p(s'|s, a) v_\pi(s') \\ &= \sum_{s'} [\sum_r p(s', r|s, a) r + \gamma \sum_r p(s', r|s, a) v_\pi(s')] \\ &= \sum_{s', r} p(s', r|s, a) [r + \gamma v_\pi(s')] \end{aligned}$$

• Οπότε

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s', r} p(s', r|s, a) [r + \gamma v_\pi(s')]$$



- Ας υποθέσουμε ότι γνωρίζουμε την από κοινού πιθανότητα ανταμοιβής $R(t) = r$ και επόμενης κατάστασης $S(t+1) = s'$ με δεδομένη την προηγούμενη κατάσταση $S(t) = s$ και ενέργεια $A(t) = a$:
$$p(s', r|s, a)$$
- Από αυτό μπορούμε να βρούμε:
 - Την πιθανότητα της επόμενης κατάστασης $S(t+1) = s'$, με δεδομένη προηγούμενη κατάσταση $S(t) = s$ και την ενέργεια $A(t) = a$:
$$p(s'|s, a) = \int_r p(s', r|s, a) dr$$
 - Την πιθανότητα ανταμοιβής $R(t) = r$ με δεδομένη την προηγούμενη κατάσταση $S(t) = s$, ενέργεια $A(t) = a$ και επόμενη κατάσταση $S(t+1) = s'$:
$$p(r|s', s, a) = \frac{p(s', r|s, a)}{p(s'|s, a)}$$



Επαναληπτική αξιολόγηση πολιτικής

- Επομένως πρέπει να λύσουμε το παρακάτω σύστημα γραμμικών εξισώσεων με $K = |\mathcal{S}|$ αγνώστους $v_\pi(1), \dots, v_\pi(K)$:

$$v_\pi(s) = \sum_a \pi(a|s)' \sum_{s', r} p(s', r|s, a) [r + \gamma v_\pi(s')]$$

- Μια προφανής προσέγγιση είναι η επίλυση του παραπάνω γραμμικού συστήματος.

- Μια εναλλακτική προσέγγιση είναι η **επαναληπτική μέθοδος**:

- Ξεκινάμε με μια αρχική εκτίμηση των τιμών κατάστασης $v_0(1), \dots, v_0(K)$

- Ενημερώνουμε με χρήση του ακόλουθου κανόνα:

$$v_{k+1}(s) = \sum_a \pi(a|s) \sum_{s', r} p(s', r|s, a) [r + \gamma v_k(s')]$$



Παράδειγμα: Μετακίνηση σε πλέγμα

Θεωρήστε το πρόβλημα μετακίνησης σε έναν 2-Διάστατο κόσμο που αποτελείται από 16 τετράγωνα όπως φαίνεται στο διπλανό σχήμα.

Το πράσινο τετράγωνο είναι τερματικό.

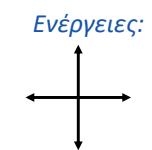
Σε κάθε θέση υπάρχουν 4 δυνατές ενέργειες:

- Κίνηση προς τα πάνω
- Κίνηση προς τα κάτω
- Κίνηση προς τα δεξιά
- Κίνηση προς τα αριστερά

Ο πράκτορας πρέπει:

- Να αποφύγει το κόκκινο τετράγωνο
- Να καταλήξει στο πράσινο τετράγωνο

Καταστάσεις:			
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15





Παράδειγμα: Μετακίνηση σε πλέγμα

Κάθε μετακίνηση έχει ανταμοιβή $r = -1$ (διότι σπαταλήθηκε ενέργεια)

Μετακίνηση εκτός ταμπλό επαναφέρει τον πράκτορα στην αρχική του θέση. Πχ, η ανταμοιβή r και η επόμενη κατάσταση s' αν βρισκόμαστε στη θέση 8 και κινηθούμε δεξιά είναι:

$$r(s = 8, a = \text{right}) = -1$$

$$s'(s = 8, a = \text{right}) = 9$$

Επίσης

$$r(s = 1, a = \text{up}) = -1$$

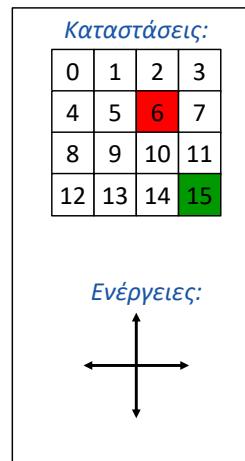
$$s'(s = 1, a = \text{up}) = 1$$

Το κόκκινο τετράγωνο είναι παγίδα.

Μετακίνηση από αυτό έχει ανταμοιβή $r = -10$
 $r(s = 6, a = \text{left}) = -10$

Μετακίνηση στο πράσινο τετράγωνο δίνει $r = 0$.

Καθώς το τετράγωνο αυτό είναι τερματικό κάθε μετακίνηση από αυτό επαναφέρει τον πράκτορα σε αυτό.



Παράδειγμα: Μετακίνηση σε πλέγμα

Αποτιμούμε την πολιτική όπου όλες οι ενέργειες είναι ισοπίθανες για κάθε s :

$$\pi(\text{up}|s) = \pi(\text{down}|s) = \pi(\text{left}|s) = \pi(\text{right}|s) = \frac{1}{4}.$$

Στην περίπτωση αυτή μια συγκεκριμένη ενέργεια a

Δίνει μια συγκεκριμένη ανταμοιβή r

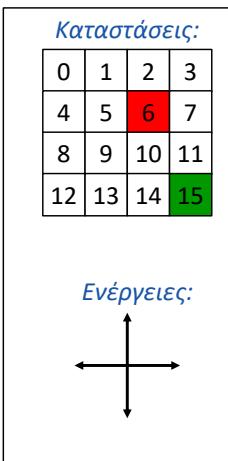
Και φέρνει τον πράκτορα σε μια νέα κατάσταση s' .

Αυτό απλοποιεί την φόρμουλα:

$$v_{k+1}(s) = \sum_a \pi(a|s)[r + \gamma v_k(s')]$$

Αποτελέσματα επαναληπτικής αποτίμησης: ($\gamma = 0.9$)

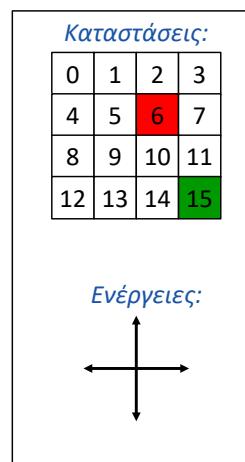
- Επανάληψη 1: $V =$
- | | | | |
|------|------|-------|------|
| -1.0 | -1.0 | -1.0 | -1.0 |
| -1.0 | -1.0 | -10.0 | -1.0 |
| -1.0 | -1.0 | -1.0 | -0.8 |
| -1.0 | -1.0 | -0.8 | 0.0 |



Παράδειγμα: Μετακίνηση σε πλέγμα

- Επανάληψη 2: $V =$

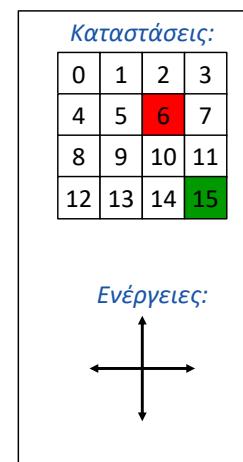
-1.9	-1.9	-3.9	-1.9
-1.9	-3.9	-10.9	-3.9
-1.9	-1.9	-3.8	-1.4
-1.9	-1.8	-1.4	0.0



Παράδειγμα: Μετακίνηση σε πλέγμα

- Επανάληψη 3: $V =$

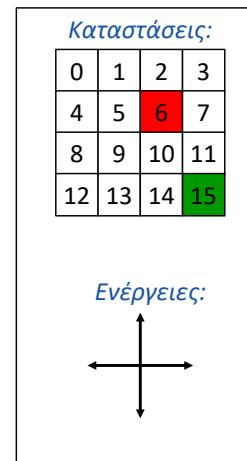
-2.7	-3.6	-5.2	-3.6
-3.2	-4.7	-13.5	-5.1
-2.7	-3.6	-4.5	-2.8
-2.7	-2.6	-2.3	0.0





- Επανάληψη 10: $V =$

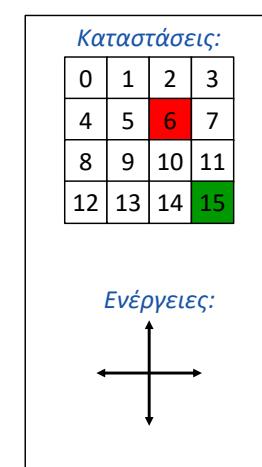
-8.3	-9.7	-12.1	-10.4
-8.4	-10.8	-19.4	-11.3
-7.6	-8.4	-9.6	-6.7
-7.0	-6.9	-5.4	0.0



- Σταθερή μετά από την επανάληψη 50.

- Επανάληψη ∞ : $V =$

-12.8	-14.0	-16.3	-14.5
-12.7	-15.0	-23.2	-14.9
-11.6	-12.1	-12.7	-9.0
-10.7	-10.1	-7.6	0.0



Επιλογή πολιτικής

- Η αξιολόγηση των καταστάσεων μας επιτρέπει να εκτιμήσουμε τις βέλτιστες δράσεις σε μια δεδομένη κατάσταση.
- Ας επικεντρωθούμε τώρα σε ντετερμινιστικές πολιτικές όπου το π δεν είναι πλέον μια πιθανότητα, αλλά μια συνάρτηση που αντιστοιχεί καταστάσεις σε ενέργειες: $a = \pi(s)$
- **Θεώρημα βελτίωσης πολιτικής:**

Έστω π και π' είναι δύο ντετερμινιστικές πολιτικές, έτσι ώστε για όλες τις καταστάσεις $s \in \mathcal{S}$ έχουμε:

$$\pi_\pi(s, \pi'(s)) \geq \nu_\pi(s)$$

Τότε οι τιμές των καταστάσεων υπό την πολιτική π' είναι μεγαλύτερες ή ίσες από τις αξίες των καταστάσεων υπό την π :

$$\nu_{\pi'}(s) \geq \nu_\pi(s), \text{ για κάθε } s \in \mathcal{S}$$



Άπληστη επιλογή πολιτικής

- Σε κάθε κατάσταση επιλέγουμε την ενέργεια που οδηγεί στην κατάσταση με την μέγιστη τιμή:
$$\pi(s) = \arg \max_a \nu(s'(a, s))$$
- Αν δύο ή περισσότερες ενέργειες οδηγούν σε καταστάσεις με ίσες τιμές επιλέγουμε τυχαία.
- **Παράδειγμα: Μετακίνηση σε πλέγμα**
- Επανάληψη 1: $V =$

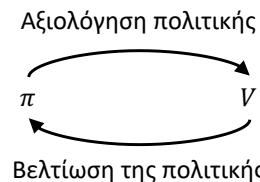
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-10.0	-1.0
-1.0	-1.0	-1.0	-0.8
-1.0	-1.0	-0.8	0.0

Άπληστη Πολιτική

↑	↑	↔	↓
↑	↔	↔	↓
↑	↔	↑	↓
↓	→	→	↓



- Εκτέλεση βρόχου που εναλλάσσεται ανάμεσα σε:
 - Αξιολόγηση πολιτικής
 - Βελτίωση πολιτικής



- Επαναλαμβάνουμε το βρόχο αξιολόγησης για L βήματα πριν εφαρμόσουμε τη βελτίωση της πολιτικής.
- Στην ειδική περίπτωση όπου $L = 1$, η διαδικασία ονομάζεται "Επανάληψη τιμής".



- Μετά από δύο επαναλήψεις:

$$\pi^{(0)} \rightarrow V^{(0)} \rightarrow \pi^{(1)} \rightarrow V^{(1)} \rightarrow \pi^{(2)}$$

- $V^{(2)} =$

-4.1	-4.7	-5.2	-1.9
-3.4	-2.7	-10.9	-1.0
-2.7	-1.9	-1.0	0.0
-1.9	-1.0	0.0	0.0

Πολιτική $\pi^{(2)}$:

↓	↓	→	↓
↖	↓	↖	↓
↖	↖	↖	↓
→	→	→	█

- Η πολιτική $\pi^{(2)}$ είναι η βέλτιστη: Ξεκινώντας από κάθε κατάσταση, οι ενέργειες παίρνουν το συντομότερο μονοπάτι προς το πράσινο τετράγωνο, αποφεύγοντας το κόκκινο τετράγωνο.



Δυναμικός Προγραμματισμός: πλεονεκτήματα και μειονεκτήματα

- Ισχυρή προσέγγιση.
- Πάντα οδηγεί σε βελτίωση της πολιτικής.
- Ωστόσο, για να εφαρμόσουμε τη μέθοδο δυναμικού προγραμματισμού χρειαζόμαστε εκτιμήσεις όλων των πιθανοτήτων

$$p(s', r | s, a)$$

- Αυτό δεν είναι πάντα εύκολο ή απλό να γίνει.
- Σε πολλά προβλήματα ο αριθμός των καταστάσεων σε συνδυασμό με τις πιθανές ενέργειες είναι πολύ μεγάλος.



Η προσέγγιση Monte Carlo

- Γενική έννοια του πειράματος Monte Carlo:
- Εκτέλεση πολλών προσομοιώσεων ενός πειράματος. Προφανώς πρέπει να υπάρχει κάποια τυχαιότητα στα πειράματα.
- Συλλογή στατιστικών στοιχείων μετά την εκτέλεση N πειράματα και λήψη απόφασης ή εκτίμησης με βάση τα στατιστικά. Όσο μεγαλύτερη είναι η τιμή του N τόσο πιο αξιόπιστες είναι οι εκτιμήσεις.
- **Απλό παράδειγμα:** Θέλουμε να ελέγχουμε αν ένα ζάρι είναι «πειραγμένο». Δηλαδή θέλουμε να ελέγχουμε την υπόθεση $P(\text{Dice} = 1) = \dots = P(\text{Dice} = 6)$.
- Ρίχνουμε το ζάρι N φορές και μετράμε πόσες φορές το αποτέλεσμα ήταν $1, 2, \dots, 6$. Οι εκτιμήσεις των πιθανοτήτων είναι $\hat{P}(\text{Dice} = i) = \frac{N_{\text{Dice}=i}}{N}$. Ελέγχουμε αν είναι ίσες (ή περίπου ίσες).



- Η αξιολόγηση της πολιτικής μπορεί να πραγματοποιηθεί με τη μέθοδο Monte Carlo, εάν το σύστημα έχει πεπερασμένο ορίζοντα (δηλαδή το πείραμα δεν τρέχει για πάντα).
- Το βασικό πείραμα Μόντε Κάρλο (MC) καλείται **Επεισόδιο**:

 - Έναρξη από τυχαία αρχική κατάσταση
 - Επιλογή ενεργειών σύμφωνα με την πολιτική
 - Μετάβαση στην επόμενη κατάσταση σύμφωνα με την πιθανότητα μετάβασης
 - Επαναλάβετε τα βήματα 2-3 μέχρι να φτάσετε σε κατάσταση τερματισμού

Τυπική εφαρμογή είναι τα παιχνίδια (πχ. black-jack, Τάβλι, Κλπ).



Είσοδος: π : πολιτική που πρέπει να αξιολογηθεί
Αρχικοποίηση:

- \hat{v}_π : τυχαίες αρχικές αξίες καταστάσεων
- $Return(s)$: κενή λίστα κερδών για κάθε κατάσταση $s \in \mathcal{S}$
- Βρόχος:

Δημιουργία επεισοδίου με χρήση πολιτικής π

Για κάθε κατάσταση s που εμφανίζεται στο επεισόδιο:

$G \leftarrow$ κέρδος που συμβαίνει μετά την πρώτη εμφάνιση του s

Προσθήκη του G στη λίστα $Return(s)$

$\hat{v}_\pi(s) \leftarrow Average(Return(s))$



Βελτίωση πολιτικής Monte Carlo

- Στη βελτίωση της πολιτικής, αντί να υπολογιστεί η αξία των κατάστασεων, είναι πιο βολικό να εκτιμηθεί η συνάρτηση Q , δεδομένου ότι η πολιτική θα βελτιωθεί με την επιλογή του μέγιστου $q_\pi(s, a)$.
- Αρχικοποίηση: $\pi(s)$: Αυθαίρετη; $\hat{q}(s, a)$: αυθαίρετη για κάθε ζεύγος $s \in \mathcal{S}$ and $a \in \mathcal{A}$; $Return(s, a)$: κενή λίστα κερδών για κάθε ζεύγος $s \in \mathcal{S}$ and $a \in \mathcal{A}$
- Βρόχος:
Δημιουργία επεισοδίου με τυχαία αρχική κατάσταση s_0 και ενέργεια a_0 με βάση την πολιτική π
Για κάθε ζεύγος (s, a) που εμφανίζεται στο επεισόδιο:
 $G \leftarrow$ κέρδος που συμβαίνει μετά την πρώτη εμφάνιση του ζεύγους s, a
Προσθήκη G στη λίστα $Return(s, a)$
 $\hat{q}_\pi(s, a) \leftarrow Average(Return(s, a))$
Για κάθε s που εμφανίζεται στο επεισόδιο:
 $\pi^{new}(s) = \arg \max_a \hat{q}_\pi(s, a)$



Προσέγγιση χρονικής διαφοράς

- Ένα πρόβλημα με τις απλές μεθόδους Monte Carlo είναι ότι πρέπει να φτάσουμε στο τέλος του επεισοδίου πριν ενημερώσουμε την αξία των καταστάσεων.
- Θυμόμαστε ότι
 $v(s) = E_\pi\{R(t+1) + \gamma v(S(t+1)) | S(t) = s\}$
- Μπορούμε να επιταχύνουμε τη διαδικασία αξιολόγησης κάθε φορά που είμαστε σε κατάσταση s , εκτελούμε ενέργεια a , παρατηρήστε μια ανταμοιβή r και προχωράμε σε μια μεταγενέστερη κατάσταση s' , με την άμεση ενημέρωση του $v(s)$ μέσω της ακόλουθης προσέγγισης:
 $\hat{v}(s) \leftarrow Average(R(t+1) + \gamma v(S(t+1)) | S(t) = s)$
- Αυτή η μέθοδος ονομάζεται μέθοδος TD(0), δεδομένου ότι κοιτάμε μόνο ένα βήμα μπροστά στην κατάσταση $S(t+1)$.



- Η μέθοδος TD(0) για την αξιολόγηση της πολιτικής θα είναι η ακόλουθη:
- Για κάθε επεισόδιο:
 - Έναρξη σε τυχαία κατάσταση s_0
 - Εκτελούμε ενέργεια a καταγράφουμε την ανταμοιβή r και προχωράμε σε κατάσταση s'
 - Υπολογισμός μέσου όρου του όρου $r + \gamma v(s')$:
 - Αύξηση μετρητή $n(s) \leftarrow n(s) + 1$
 - Ενημέρωση: $v(s) \leftarrow \frac{n(s)-1}{n(s)} v(s) + \frac{1}{n(s)} (r + \gamma v(s'))$
ή ισοδύναμα, $v(s) \leftarrow v(s) + \beta(r + \gamma v(s') - v(s))$ με $\beta = 1/n(s)$.
 - Θέτουμε $s \leftarrow s'$ και επαναλαμβάνουμε τα βήματα 2-3 μέχρι να φτάσουμε σε τερματική κατάσταση.
- Επαναλάβετε για όσο το δυνατόν περισσότερα επεισόδια.



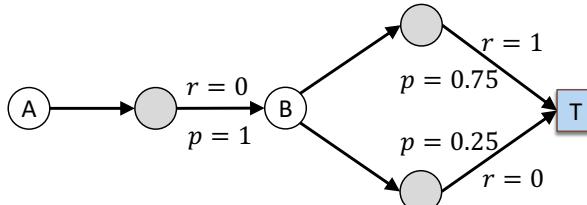
- Είσοδος: π την πολιτική που πρέπει να αξιολογηθεί
 - Αρχικοποίηση: $\hat{v}_\pi(s) = 0$, για κάθε $s \in S$
 - Για κάθε επεισόδιο:
- Θέσε $s = s_0$
 Για κάθε βήμα του επεισοδίου:
 Εκτέλεση ενέργειας $a = \pi(s)$ Temporal Difference
 Παρατηρούμε την ανταμοιβή r και την επόμενη κατάσταση s'

$$\hat{v}_\pi(s) \leftarrow \hat{v}_\pi(s) + \beta(r + \gamma \hat{v}_\pi(s') - \hat{v}_\pi(s))$$



Monte Carlo vs. TD(0)

- Εξετάστε το ακόλουθο απλό διάγραμμα ενεργειών καταστάσεων



- Έστω ότι παρατηρούμε τα ακόλουθα 8 επεισόδια:

1) $A, r = 0, B, r = 0$	2) $B, r = 1$
3) $B, r = 1$	4) $B, r = 1$
5) $B, r = 1$	6) $B, r = 1$
7) $B, r = 0$	8) $B, r = 1$



Monte Carlo vs. TD(0)

- Εάν χρησιμοποιήσουμε TD(0) για να εκτιμήσουμε τις αξίες των καταστάσεων θα βρούμε

$$\hat{v}(B) = 0.75$$
 διότι 75% του χρόνου παίρνουμε ανταμοιβή $r = 1$.

$$\hat{v}_{TD}(B) = \text{Average}(r + \hat{v}_{TD}(T)) = \text{Average}(r + 0) = 0.75$$
- Στη συνέχεια, αυτή η τιμή θα μεταδοθεί στην κατάσταση A, επειδή

$$\hat{v}_{TD}(A) = \text{Average}(r + \hat{v}_{TD}(B)) = \text{Average}(0 + \hat{v}_{TD}(B)) = 0.75$$
 (υποθέσαμε $\gamma = 1$).
- Αυτό είναι λογικό δεδομένου ότι η κατάσταση A οδηγεί πάντα στην κατάσταση B χωρίς πρόσθετη ανταμοιβή. Ως εκ τούτου, οι δύο τιμές θα πρέπει να είναι ίσες: $v(A) = v(B)$.



- Αν χρησιμοποιήσουμε την προσέγγιση Monte Carlo, τότε
 $\hat{v}_{MC}(A) = 0$
 επειδή έχουμε μόνο ένα επεισόδιο που εμπλέκει την κατάσταση A και το οποίο δίνει κέρδος $G = 0 + 0 = 0$.
- Η εκτίμηση $v(B)$ θα είναι ακριβής
 $\hat{v}_{MC}(B) = 0.75$
 διότι 6 από τα 8 επεισόδια που εμπλέκουν το B δίνουν κέρδος $G = 1$ ενώ 2 από τα 8 δίνουν κέρδος $G = 0$.
- Η μέθοδος MC είναι πιο πιστή στα δεδομένα εκπαίδευσης και έχει φτωχότερη επίδοση στα δεδομένα ελέγχου.



- Παιχνίδι "Go". Πρόγραμμα "AlphaGo" της Deep Mind: μέθοδος Monte Carlo με βαθύ συνελικτικό δίκτυο για τη μοντελοποίηση της αξίας καταστάσεων. Νίκησε τον Fan Hui 5/0 και τον Lee Sedol 4/1. Η νεότερη έκδοση "AlphaGo Zero" νίκησε το "AlphaGo" 100/0 και το "AlphaGo Master" 89/11.
- Σύσταση περιεχομένου Web: Ποια σελίδα να συστήσουμε μεταξύ n διαφορετικών σελίδων? → Πρόβλημα πολλαπλών μονόχειρων ληστών. Ανταμοιβή = Click-through rate = [αριθμός κλικ στη σελίδα]/[αριθμός επισκέψεων]
- Βελτιστοποίηση ελεγκτών μνήμης (Βελτιστοποίηση DRAM)
- Παίζουμε βίντεο-παιχνιδιών σε επίπεδο αντίστοιχο ή καλύτερο του ανθρώπου.



- Ρομποτική
- (Έλεγχος βάσισης τετράποδου) Policy Gradient Reinforcement Learning for Fast Quadrupedal Locomotion by Nate Kohl and Peter Stone
- (Πιάσιμο μπάλας από τετράποδο) Learning Ball Acquisition on a Physical Robot by Peggy Fidelman and Peter Stone
- (Air Hockey) Learning from Observation Using Primitives, and particularly the movie of a humanoid robot playing air hockey. An example paper.
- (Active Sensing) Active Sensing Using Reinforcement Learning by Cody Kwok and Dieter Fox.



- Έλεγχος
- (Έλεγχος ελικοπτέρων) Inverted autonomous helicopter flight via reinforcement learning, by Andrew Y. Ng, Adam Coates, Mark Diel, Varun Ganapathi, Jamie Schulte, Ben Tse, Eric Berger and Eric Liang. In International Symposium on Experimental Robotics, 2004.
- Autonomous helicopter control using Reinforcement Learning Policy Search Methods, by J.A. Bagnell and J. Schneider. In Proceedings of the International Conference on Robotics and Automation, 2001.



- Επιχειρησιακή Έρευνα
- (*Τιμολόγηση*) [Opportunities and Challenges in Using Online Preference Data for Vehicle Pricing: A Case Study at General Motors](#) by P. Rusmevichtientong, J. A. Salisbury, L. T. Truss, B. Van Roy, and P. W. Glynn.
- (*Δρομολόγηση οχημάτων*) [Scaling Average-reward Reinforcement Learning for Product Delivery](#) by S. Proper and P. Tadepalli.
- (*Στοχευμένο μάρκετινγκ*) [Cross Channel Optimized Marketing by Reinforcement Learning](#), by Naoki Abe, Naval Verma, Chid Apte and Robert Schrok, Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 2004.



- Παιχνίδια
- (*Τάθλι*) [Temporal difference learning and TD-Gammon](#) by Gerald Tesauro, Communications of the ACM, 38(3), March 1995.
- (*Πασιέντζα*) [Solitaire: Man Versus Machine](#), by X. Yan, P. Diaconis, P. Rusmevichtientong, and B. Van Roy, to appear in Advances in Neural Information Processing Systems 17, MIT Press, 2005.
- (*Σκάκι*) [The KnightCap program](#), which went from a rating of 1600 to a rating of 2100 by altering its heuristic evaluation function using TD-lambda. [pdf](#)
- (*Ντάμα*) [Temporal Difference Learning Applied to a High-Performance Game-Playing Program](#) by Jonathan Schaeffer, Markian Hlynka, and Vili Jussila, International Joint Conference on Artificial Intelligence (IJCAI), pp. 529-534, 2001.



- Human Computer Interaction
- (*Συστήματα Προφορικού Διαλόγου*) [Optimizing Dialogue Management with Reinforcement Learning: Experiments with the NJFun System](#). S. Singh, D. Litman, M. Kearns and M. Walker. In Journal of Artificial Intelligence Research (JAIR), Volume 16, pages 105-133, 2002
- (*Software Agent in MOOs*) [Cobot in LambdaMOO: An Adaptive Social Statistics Agent](#). C. Isbell, M. Kearns, S. Singh, C. Shelton, P. Stone and D. Korman.



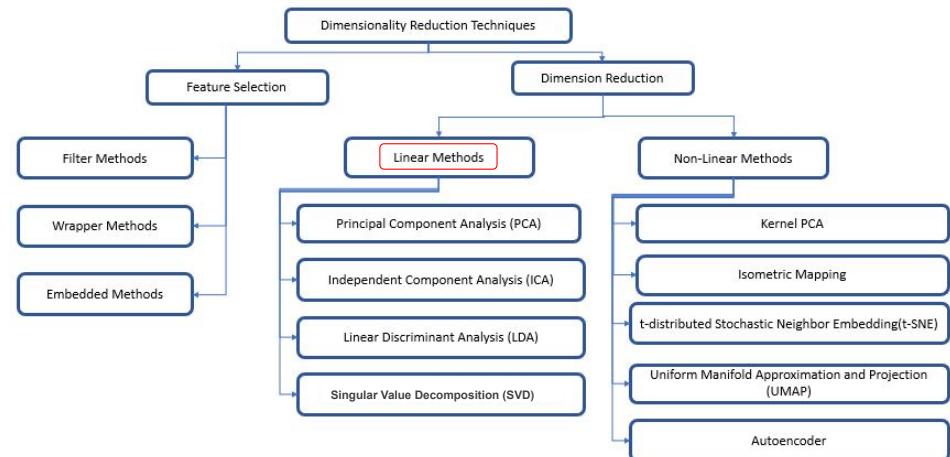
- Οικονομικά
- (*Trading*) Learning to Trade via Direct Reinforcement. John Moody and Matthew Saffell, IEEE Transactions on Neural Networks, Vol 12, No 4, July 2001.
- Σύνθετες προσομοιώσεις
- (*Robot Soccer*) [Scaling Reinforcement Learning toward RoboCup Soccer](#), by Peter Stone and Richard S. Sutton, Proceedings of the Eighteenth International Conference on Machine Learning, pp. 537–544, Morgan Kaufmann, San Francisco, CA, 2001.



ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Dimensionality Reduction

Τζούβελη Παρασκευή



Dimensionality Reduction

Taking a picture



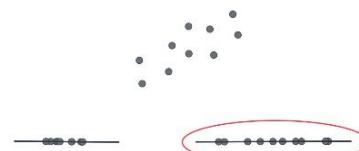
Dimensionality Reduction



Taking a picture



Dimensionality Reduction



Ποια είναι η ιδανική γραμμή για προβολή των δεδομένων, όπου αυτά θα είναι όσο πιο διακριτά γίνεται;

[10.6 Key Steps of PCA in Practice pg 336, MATHEMATICS FOR MACHINE LEARNING](#)

Principal Component Analysis (PCA)

Στόχος: Εύρεση προβολών των σημείων δεδομένων x_n όπου:

- διατηρούν τη μέγιστη ομοιότητα με τα αρχικά δεδομένα και ελαχιστοποιούν το σφάλμα ανακατασκευής
- πετυχαίνουν χαμηλότερη διάστασης, μειώνοντας την πολυπλοκότητα και διατηρώντας τη δομή των δεδομένων.

π.χ. Housing Dataset

Datasetⁿ features

Size
Number of rooms
Number of bathrooms
Schools around
Crime rate

Διάνυσμα 5 χαρακτηριστικών

Μείωση διαστάσεων

Διάνυσμα 2 χαρακτηριστικών

Size
Number of rooms
Number of bathrooms

Schools around
Crime rate

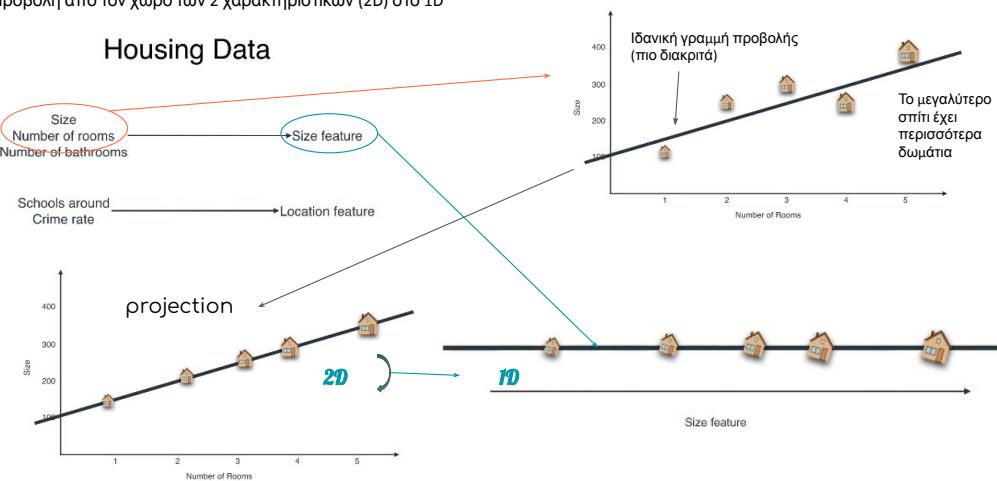
→ Size feature

→ Location feature

Principal Component Analysis (PCA)

Προβολή από τον χώρο των 2 χαρακτηριστικών (2D) στο 1D

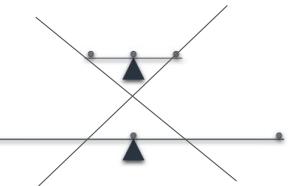
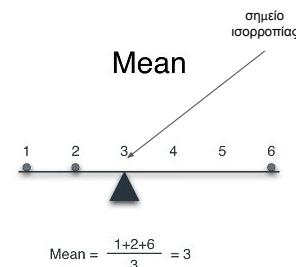
Housing Data



Principal Component Analysis (PCA): μετρικές

σημείο ισορροπίας

Mean

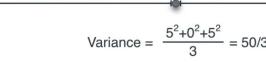
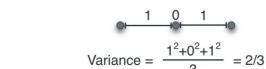


1διο mean value, → δεν είναι βολική μετρική

$$\sigma^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{N}$$

where:
 x_i = Each value in the data set
 \bar{x} = Mean of all values in the data set
 N = Number of values in the data set

Variance



Υπολογισμός: παίρνω την απόσταση κάθε σημείου από το κέντρο.

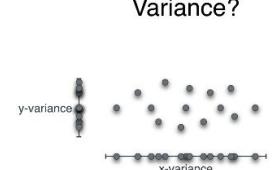
$$\text{Variance} = \frac{2^2 + 1^2 + 3^2}{3} = 14/3$$

Principal Component Analysis (PCA): μετρικές

Εξετάζουμε τη διακύμανση στο χώρο
→ εξετάζουμε και τις δύο διαστάσεις

$$\text{var}[f] = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])^2] = \mathbb{E}[f(x)^2] - \mathbb{E}[f(x)]^2$$

Variance?



$$\text{Var}(X) = \mathbb{E}[(X - \mu)^2]$$

where:

- X : Random variable.
- $\mu = \mathbb{E}[X]$: Mean (expected value) of X .
- \mathbb{E} : Expectation operator.

Alternatively, it can be expressed as:

$$\text{Var}(X) = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$$

$$\text{x-variance} = \frac{2^2 + 0^2 + 2^2}{3} = 8/3$$

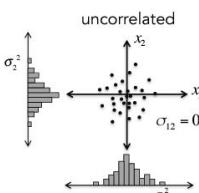
$$\text{y-variance} = \frac{1^2 + 0^2 + 1^2}{3} = 2/3$$

Variance?



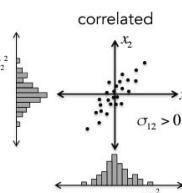
Principal Component Analysis (PCA): μετρικές

Αν, όμως, έχω δύο διαφορετικά σύνολα μπορεί να έχουν την ίδια διακύμανση στις προβολές x_1 και $x_2 \rightarrow$ μη διαχωρίσιμα
Άρα χρειάζεται άλλη μετρική για να περιγράψουμε την μεταξύ τους σχέση.



covariance

$$\sigma_{12} = \frac{1}{m} \sum_{j=1}^m (x_1^{(j)} - \mu_1)(x_2^{(j)} - \mu_2)$$



correlation

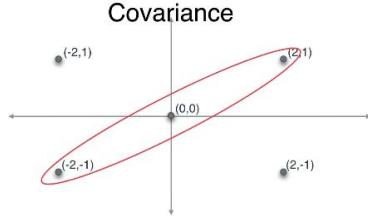
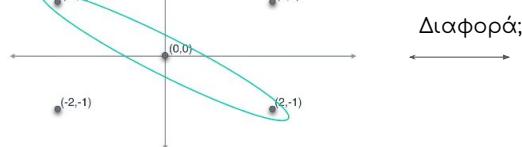
$$\rho = \frac{\sigma_{12}}{\sqrt{\sigma_1^2 \sigma_2^2}}$$

Principal Component Analysis (PCA): μετρικές

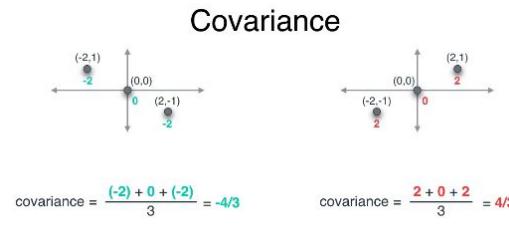
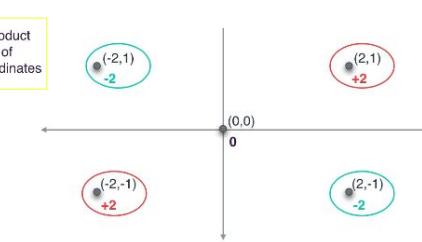


$$\begin{aligned}\text{cov}[x, y] &= \mathbb{E}_{x,y} [\{x - \mathbb{E}[x]\} \{y - \mathbb{E}[y]\}] \\ &= \mathbb{E}_{x,y}[xy] - \mathbb{E}[x]\mathbb{E}[y] \\ \text{cov}[\mathbf{x}, \mathbf{y}] &= \mathbb{E}_{\mathbf{x}, \mathbf{y}} [\{\mathbf{x} - \mathbb{E}[\mathbf{x}]\} \{\mathbf{y}^T - \mathbb{E}[\mathbf{y}^T]\}] \\ &= \mathbb{E}_{\mathbf{x}, \mathbf{y}}[\mathbf{xy}^T] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}^T]\end{aligned}$$

Covariance



Principal Component Analysis (PCA): μετρικές



$\text{Cov}(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)]$
where:

- $\mu_X = \mathbb{E}[X]$: Mean of X .
- $\mu_Y = \mathbb{E}[Y]$: Mean of Y .
- \mathbb{E} : Expectation operator.

 Alternatively, it can be expressed as:
 $\text{Cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$
 If X and Y are independent, then $\text{Cov}(X, Y) = 0$.

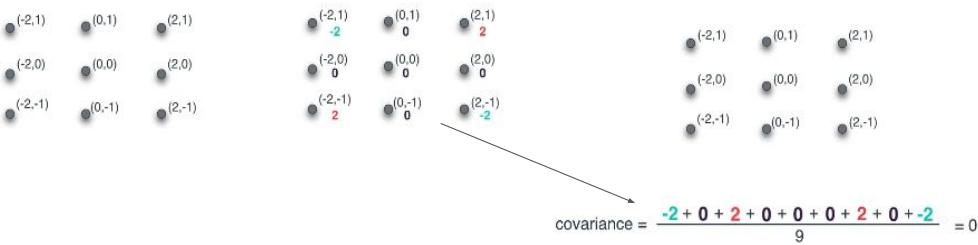
$$\text{Cov}(X, Y) = \frac{\sum(X_i - \bar{X})(Y_j - \bar{Y})}{n}$$

Principal Component Analysis (PCA): μετρικές

Ας υπολογίσουμε το covariance των σημείων:

Είναι 0 και αυτό φαίνεται λογικό αφού δεν φαίνεται να υπάρχει κάποια θετική ή αρνητική συνδιακύμανση

Covariance



Principal Component Analysis (PCA): μετρικές

Covariance



negative covariance

covariance zero (or very small)

positive covariance

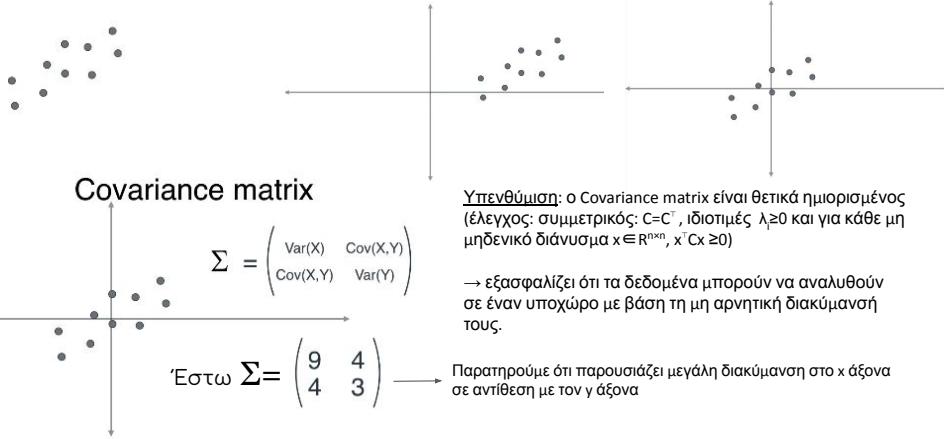
Αν το x αυξάνεται, τότε το y μειώνεται

To x ανεξάρτητο του y

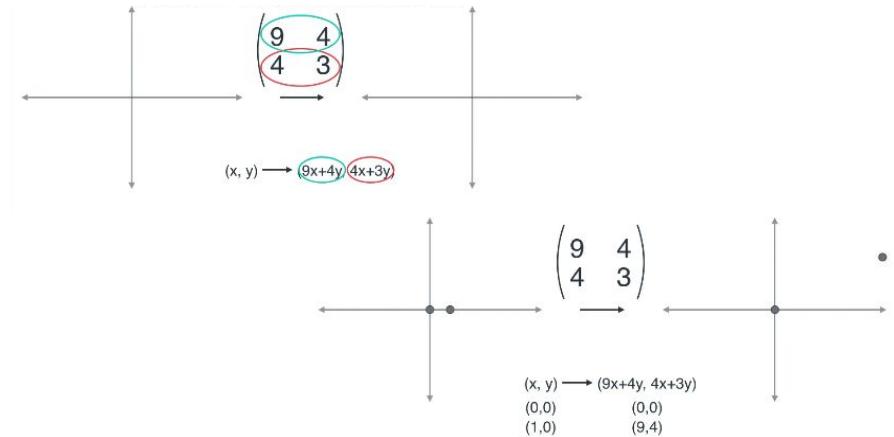
Ta x και y αυξομειώνονται μαζί

Principal Component Analysis (PCA): Covariance Matrix

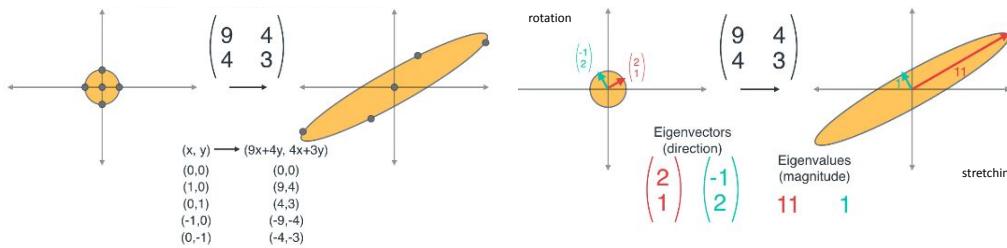
Πώς θα βρούμε την τέλεια προβολή; Θα ορίσουμε σύστημα αξόνων → θα βρούμε το σημείο όπου τα δεδομένα ισορροπούν



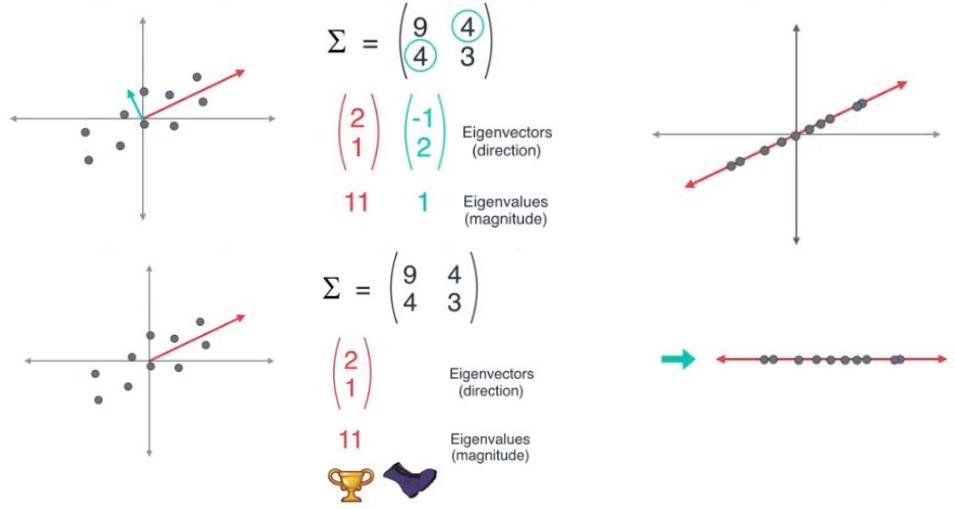
Principal Component Analysis (PCA): Linear Transformation



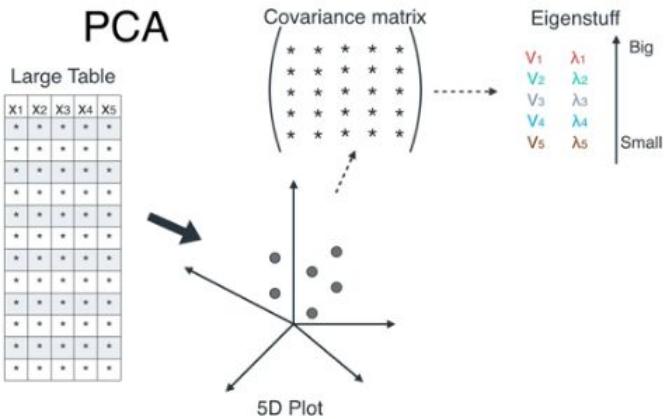
Principal Component Analysis (PCA): Linear Transformation



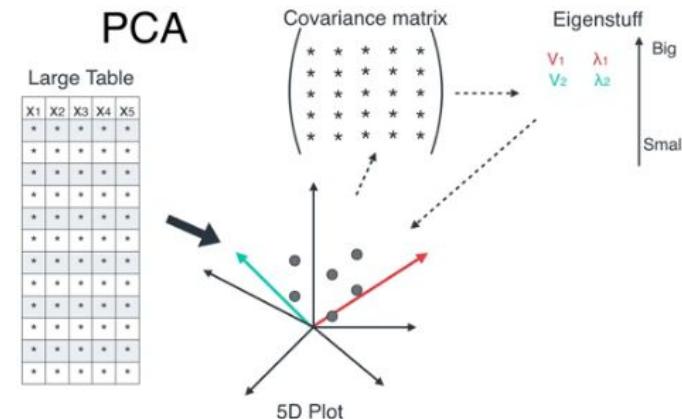
Principal Component Analysis (PCA)



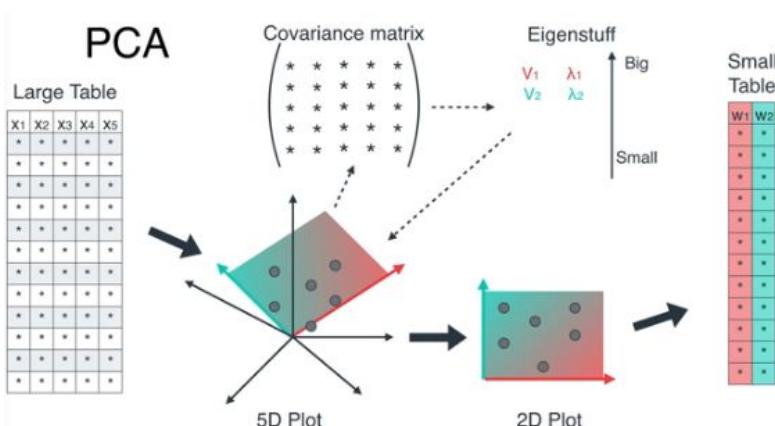
Principal Component Analysis (PCA): 5D features → 2D features



Principal Component Analysis (PCA): 5D features → 2D features



Principal Component Analysis (PCA) 5D features → 2D features

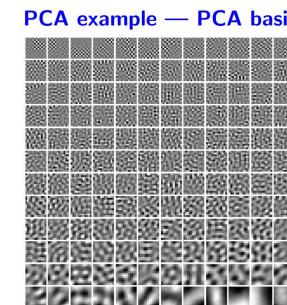


Εφαρμογές PCA : Συμπίεση εικόνας

PCA example — 90% compressed



PCA example — original



PCA example — PCA basis



PCA example — 50% compressed



Implementing a Principal Component Analysis (PCA)

Principal Component Analysis (PCA) : Projection onto a subspace

Πρόβλημα: χωρητικότητα δεδομένων

Λύση: Συμπίεσης δεδομένων με τρόπο που απαιτεί λιγότερη μνήμη, αλλά χωρίς να χάνει πολύ σε ακρίβεια.

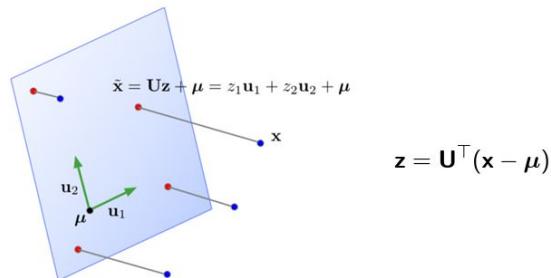
Στόχος: αναπαράσταση σε μικρότερες διαστάσεις, χωρίς να χάνουμε πολύ σε ακρίβεια

Set-up: given a dataset $\mathcal{D} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\} \subset \mathbb{R}^D$

Set μ to the mean of the data, $\mu = \frac{1}{N} \sum_{i=1}^N \mathbf{x}^{(i)}$

Goal: find a K -dimensional subspace $\mathcal{S} \subset \mathbb{R}^D$ such that $\mathbf{x}^{(n)} - \mu$ is "well-represented" by its projection onto \mathcal{S}

Recall: The **projection** of a point \mathbf{x} onto \mathcal{S} is the point in \mathcal{S} closest to \mathbf{x} .



In machine learning, $\tilde{\mathbf{x}}$ is also called the **reconstruction** of \mathbf{x} .

\mathbf{z} is its **representation**, or **code**.

Principal Component Analysis (PCA) : Projection onto a subspace

Let $\{\mathbf{u}_k\}_{k=1}^K$ be an orthonormal basis of the subspace \mathcal{S}

Approximate each data point \mathbf{x} as:

$$\begin{aligned}\tilde{\mathbf{x}} &= \mu + \text{Proj}_{\mathcal{S}}(\mathbf{x} - \mu) \\ &= \mu + \sum_{k=1}^K z_k \mathbf{u}_k\end{aligned}$$

From linear algebra: $z_k = \mathbf{u}_k^T(\mathbf{x} - \mu)$

Let \mathbf{U} be a matrix with columns $\{\mathbf{u}_k\}_{k=1}^K$ then $\mathbf{z} = \mathbf{U}^T(\mathbf{x} - \mu)$

Also: $\tilde{\mathbf{x}} = \mu + \mathbf{U}\mathbf{z}$

Principal Component Analysis (PCA) : Projection onto a Subspace

Principal Component Analysis (PCA) : Learning a Subspace

How to choose a good subspace \mathcal{S} ?

- Need to choose $D \times K$ matrix \mathbf{U} with orthonormal columns.

Two criteria:

- Minimize the **reconstruction error**

$$\min \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}^{(i)} - \tilde{\mathbf{x}}^{(i)}\|^2$$

- Maximize the variance of the code vectors

$$\begin{aligned}\max_j \sum_i \text{Var}(z_j) &= \frac{1}{N} \sum_j \sum_i (z_j^{(i)} - \bar{z}_j)^2 \\ &= \frac{1}{N} \sum_i \|\mathbf{z}^{(i)} - \bar{\mathbf{z}}\|^2 \\ &= \frac{1}{N} \sum_i \|\mathbf{z}^{(i)}\|^2\end{aligned}$$

Exercise: show $\bar{\mathbf{z}} = 0$

- Note: here, $\bar{\mathbf{z}}$ denotes the mean, not a derivative.

Principal Component Analysis (PCA) : Learning a Subspace

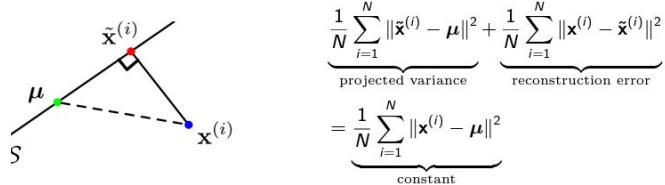
These two criteria are equivalent! I.e., we'll show

$$\frac{1}{N} \sum_{i=1}^N \|\mathbf{x}^{(i)} - \tilde{\mathbf{x}}^{(i)}\|^2 = \text{const} - \frac{1}{N} \sum_i \|\mathbf{z}^{(i)}\|^2$$

Observation: by unitarity,

$$\|\tilde{\mathbf{x}}^{(i)} - \boldsymbol{\mu}\| = \|\mathbf{U}\mathbf{z}^{(i)}\| = \|\mathbf{z}^{(i)}\|$$

By the Pythagorean Theorem,



Principal Component Analysis (PCA)

Choosing a subspace to maximize the projected variance, or minimize the reconstruction error, is called **principal component analysis (PCA)**.

Recall:

- **Spectral Decomposition:** a symmetric matrix \mathbf{A} has a full set of eigenvectors, which can be chosen to be orthogonal. This gives a decomposition

$$\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top,$$

where \mathbf{Q} is orthogonal and Λ is diagonal. The columns of \mathbf{Q} are eigenvectors, and the diagonal entries λ_j of Λ are the corresponding eigenvalues.

- I.e., symmetric matrices are diagonal in some basis.
- A symmetric matrix \mathbf{A} is positive semidefinite iff each $\lambda_j \geq 0$.

Principal Component Analysis (PCA)

Consider the **empirical covariance matrix**:

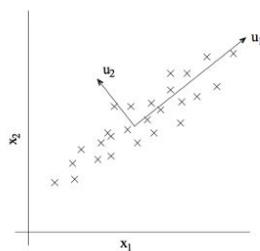
$$\boldsymbol{\Sigma} = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}^{(i)} - \boldsymbol{\mu})(\mathbf{x}^{(i)} - \boldsymbol{\mu})^\top$$

Recall: Covariance matrices are symmetric and positive semidefinite.

The optimal PCA subspace is spanned by the top K eigenvectors of $\boldsymbol{\Sigma}$.

- More precisely, choose the first K of any orthonormal eigenbasis for $\boldsymbol{\Sigma}$.
- The general case is tricky, but we'll show this for $K = 1$.

These eigenvectors are called **principal components**, analogous to the principal axes of an ellipse.



Principal Component Analysis (PCA)

For $K = 1$, we are fitting a unit vector \mathbf{u} , and the code is a scalar $z = \mathbf{u}^\top(\mathbf{x} - \boldsymbol{\mu})$.

$$\begin{aligned} \frac{1}{N} \sum_i [z^{(i)}]^2 &= \frac{1}{N} \sum_i (\mathbf{u}^\top(\mathbf{x}^{(i)} - \boldsymbol{\mu}))^2 \\ &= \frac{1}{N} \sum_{i=1}^N \mathbf{u}^\top(\mathbf{x}^{(i)} - \boldsymbol{\mu})(\mathbf{x}^{(i)} - \boldsymbol{\mu})^\top \mathbf{u} \\ &= \mathbf{u}^\top \left[\frac{1}{N} \sum_{i=1}^N (\mathbf{x}^{(i)} - \boldsymbol{\mu})(\mathbf{x}^{(i)} - \boldsymbol{\mu})^\top \right] \mathbf{u} \\ &= \mathbf{u}^\top \boldsymbol{\Sigma} \mathbf{u} \\ &= \mathbf{u}^\top \mathbf{Q}\Lambda\mathbf{Q}^\top \mathbf{u} \\ &= \mathbf{a}^\top \boldsymbol{\Lambda} \mathbf{a} \\ &= \sum_{j=1}^D \lambda_j a_j^2 \end{aligned}$$

Spectral Decomposition
for $\mathbf{a} = \mathbf{Q}^\top \mathbf{u}$

Principal Component Analysis (PCA)

Maximize $\mathbf{a}^\top \Lambda \mathbf{a} = \sum_{j=1}^D \lambda_j a_j^2$ for $\mathbf{a} = \mathbf{Q}^\top \mathbf{u}$.

- This is a change-of-basis to the eigenbasis of Σ .

Assume the λ_i are in sorted order. For simplicity, assume they are all distinct.

Observation: since \mathbf{u} is a unit vector, then by unitarity, \mathbf{a} is also a unit vector. I.e., $\sum_j a_j^2 = 1$.

By inspection, set $a_1 = \pm 1$ and $a_j = 0$ for $j \neq 1$.

Hence, $\mathbf{u} = \mathbf{Q}\mathbf{a} = \mathbf{q}_1$ (the top eigenvector).

A similar argument shows that the k th principal component is the k th eigenvector of Σ . If you're interested, look up the [Courant-Fischer Theorem](#).

Principal Component Analysis (PCA)

Interesting fact: the dimensions of \mathbf{z} are decorrelated. For now, let Cov denote the empirical covariance.

$$\begin{aligned}\text{Cov}(\mathbf{z}) &= \text{Cov}(\mathbf{U}^\top(\mathbf{x} - \boldsymbol{\mu})) \\ &= \mathbf{U}^\top \text{Cov}(\mathbf{x}) \mathbf{U} \\ &= \mathbf{U}^\top \Sigma \mathbf{U} \\ &= \mathbf{U}^\top \mathbf{Q} \Lambda \mathbf{Q}^\top \mathbf{U} \\ &= (\mathbf{I} \ \mathbf{0}) \Lambda \begin{pmatrix} \mathbf{I} \\ \mathbf{0} \end{pmatrix} \quad \text{by orthogonality} \\ &= \text{top left } K \times K \text{ block of } \Lambda\end{aligned}$$

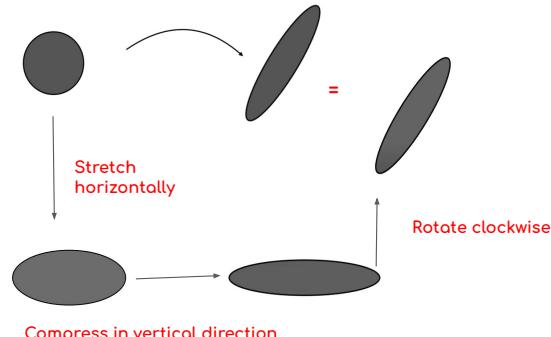
If the covariance matrix is diagonal, this means the features are uncorrelated.

This is why PCA was originally invented (in 1901!).

Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

Μετασχηματισμοί: μόνο ως προς οριζόντια και κάθετα κατεύθυνση, όχι υπό άλλη γωνία

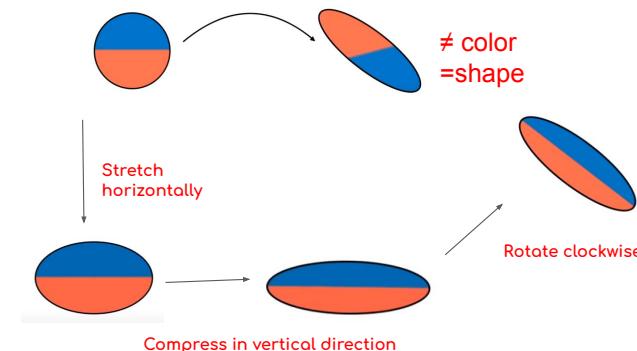
Puzzle (easy)



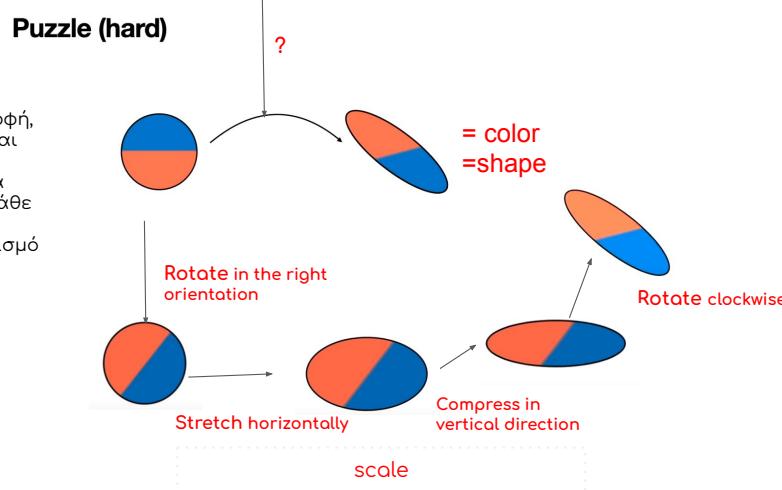
[4.5 SingularValueDecomposition pg 119](#)
MATHEMATICS FOR MACHINE LEARNING

Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

Puzzle (hard)

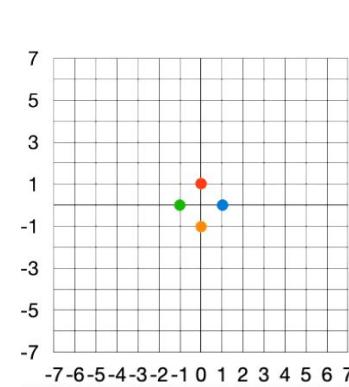


Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

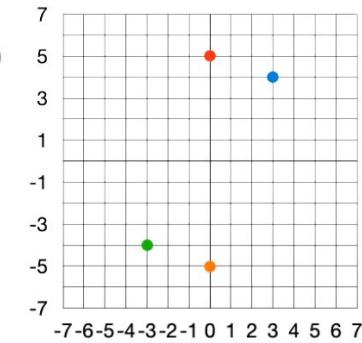


Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

Γραμμικός μετασχηματισμός

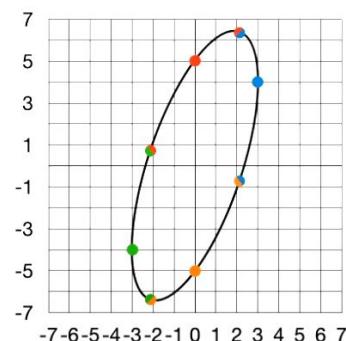
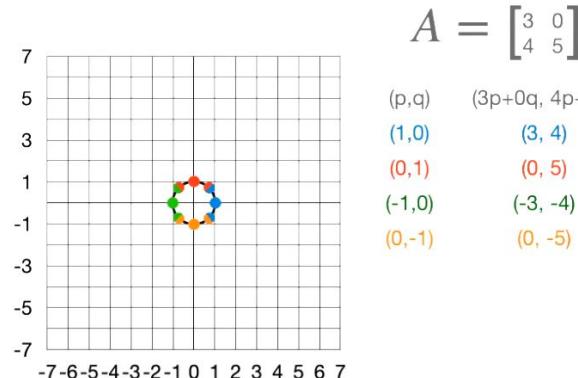


$$A = \begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix}$$



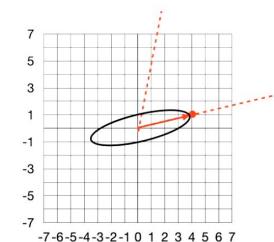
Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

Γραμμικός μετασχηματισμός

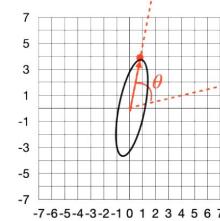


Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

Πίνακας περιστροφής



$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

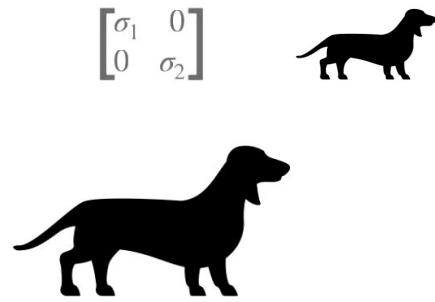
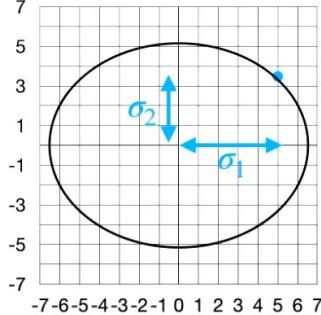


$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$



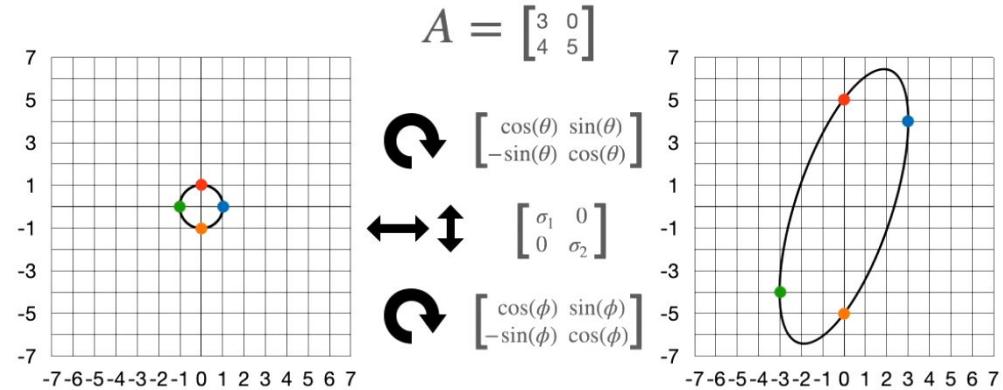
Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

Πίνακας κλιμάκωσης



$$\begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}$$

Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)



Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

WolframAlpha computational intelligence

singular value decomposition [[3,0],[4,5]]

Input: singular value decomposition $\begin{pmatrix} 3 & 0 \\ 4 & 5 \end{pmatrix}$

Result: $M = U\Sigma V^\dagger$

where

$$M = \begin{pmatrix} 3 & 0 \\ 4 & 5 \end{pmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{bmatrix}$$

Code:

```
from numpy.linalg import svd
A = np.array([[3,0],[4,5]])
svd(A)
```

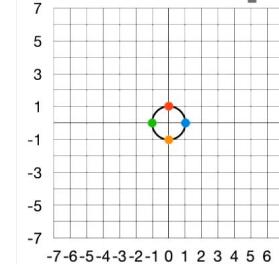
Output:

```
(array([[-0.31622777, -0.9486833 ],
       [-0.9486833 ,  0.31622777]]),
 array([6.70820393, 2.23606798]),
 array([[ -0.70710678, -0.70710678],
       [-0.70710678,  0.70710678]]))
```

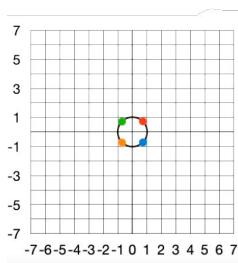
Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

$$A = U\Sigma V^\dagger$$

$$\begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix} = \begin{bmatrix} 0.316 & -0.949 \\ 0.949 & 0.316 \end{bmatrix} \begin{bmatrix} 6.708 & 0 \\ 0 & 2.236 \end{bmatrix} \begin{bmatrix} 0.7071 & 0.7071 \\ -0.7071 & 0.7071 \end{bmatrix}$$

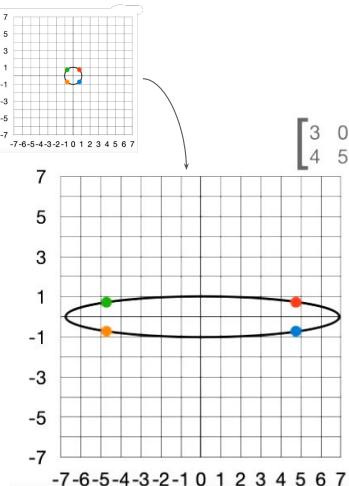


$$\begin{bmatrix} 1/\sqrt{10} & -3/\sqrt{10} \\ 3/\sqrt{10} & 1/\sqrt{10} \end{bmatrix} \begin{bmatrix} 3\sqrt{5} & 0 \\ 0 & \sqrt{5} \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$



Rotation of $\theta = -\frac{\pi}{4} = -45^\circ$

Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)



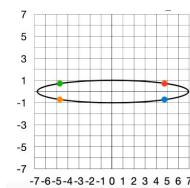
$$A = U\Sigma V^\dagger$$

$$\begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix} = \begin{bmatrix} 0.316 & -0.949 \\ 0.949 & 0.316 \end{bmatrix} \begin{bmatrix} 6.708 & 0 \\ 0 & 2.236 \end{bmatrix} \begin{bmatrix} 0.7071 & 0.7071 \\ -0.7071 & 0.7071 \end{bmatrix}$$

$$\begin{bmatrix} 1/\sqrt{10} & -3/\sqrt{10} \\ 3/\sqrt{10} & 1/\sqrt{10} \end{bmatrix} \begin{bmatrix} 3\sqrt{5} & 0 \\ 0 & \sqrt{5} \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

Horizontal scaling by $3\sqrt{5}$

Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

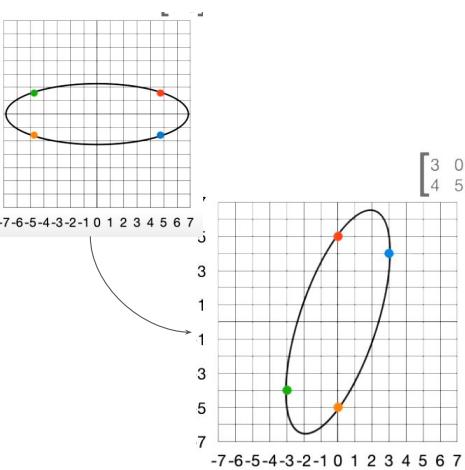


$$\begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix} = \begin{bmatrix} 0.316 & -0.949 \\ 0.949 & 0.316 \end{bmatrix} \begin{bmatrix} 6.708 & 0 \\ 0 & 2.236 \end{bmatrix} \begin{bmatrix} 0.7071 & 0.7071 \\ -0.7071 & 0.7071 \end{bmatrix}$$

$$\begin{bmatrix} 1/\sqrt{10} & -3/\sqrt{10} \\ 3/\sqrt{10} & 1/\sqrt{10} \end{bmatrix} \begin{bmatrix} 3\sqrt{5} & 0 \\ 0 & \sqrt{5} \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

Vertical scaling by $\sqrt{5}$

Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

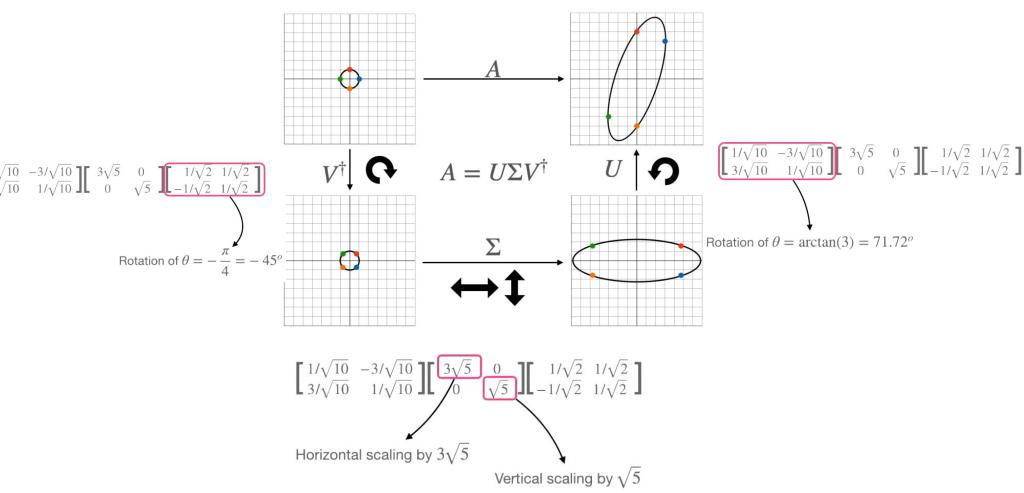


$$\begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix} = \begin{bmatrix} 0.316 & -0.949 \\ 0.949 & 0.316 \end{bmatrix} \begin{bmatrix} 6.708 & 0 \\ 0 & 2.236 \end{bmatrix} \begin{bmatrix} 0.7071 & 0.7071 \\ -0.7071 & 0.7071 \end{bmatrix}$$

$$\begin{bmatrix} 1/\sqrt{10} & -3/\sqrt{10} \\ 3/\sqrt{10} & 1/\sqrt{10} \end{bmatrix} \begin{bmatrix} 3\sqrt{5} & 0 \\ 0 & \sqrt{5} \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

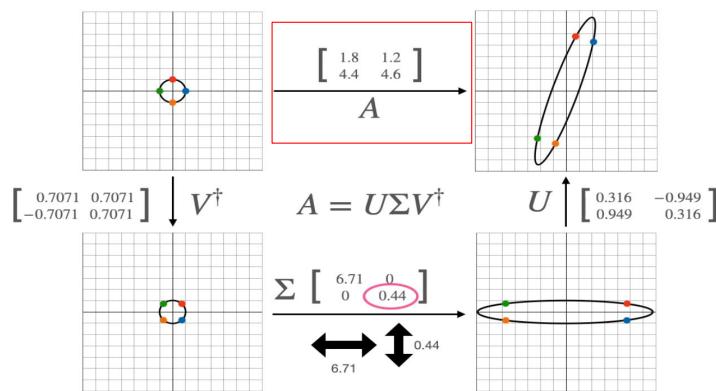
Rotation of $\theta = \arctan(3) = 71.72^\circ$

Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

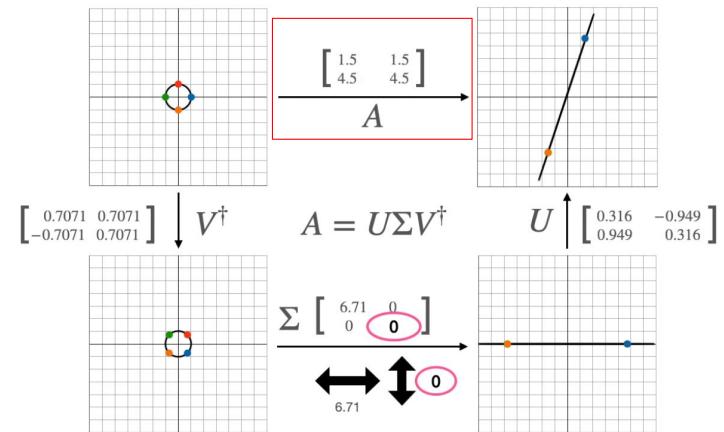


Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

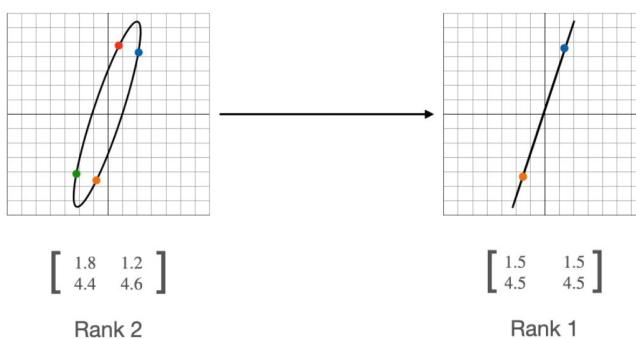
Μείωση διαστάσεων



Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)



Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)



Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

Πίνακας με rank=1

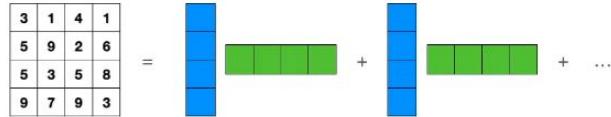
	1	2	3	4
1	1	2	3	4
-1	-1	-2	-3	-4
2	2	4	6	8
10	10	20	30	40

1	2	3	4
-1	-2	-3	-4
2	4	6	8
10	20	30	40

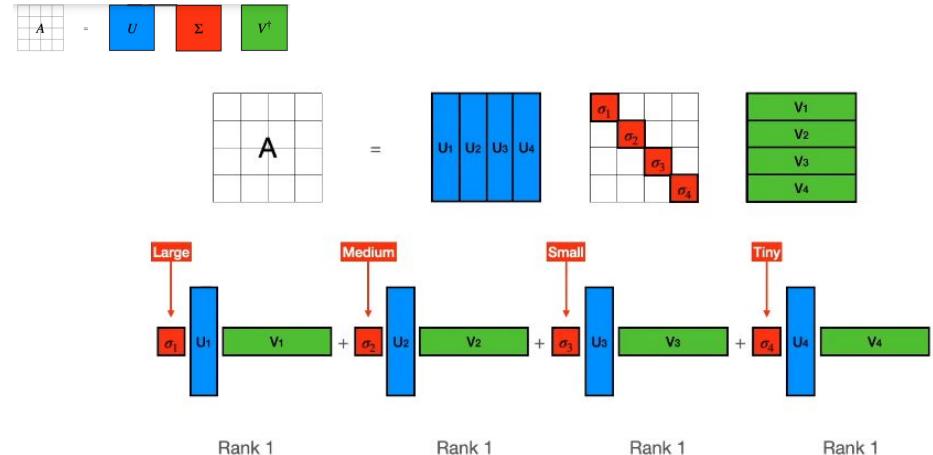
1
-1
2
3
4

Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

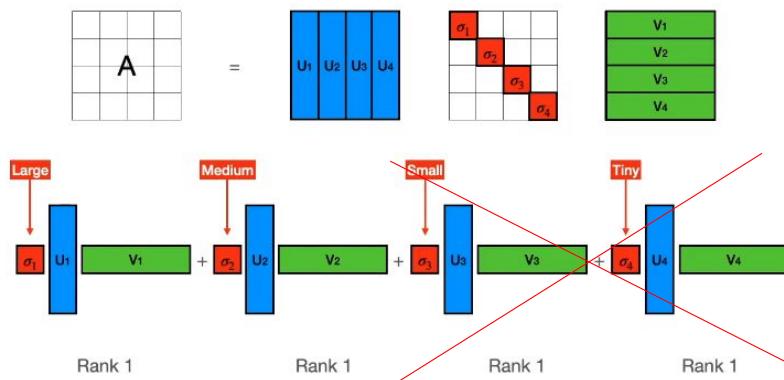
Πίνακας με rank>1 : προσέγγιση όπως λειτουργεί για rank=1



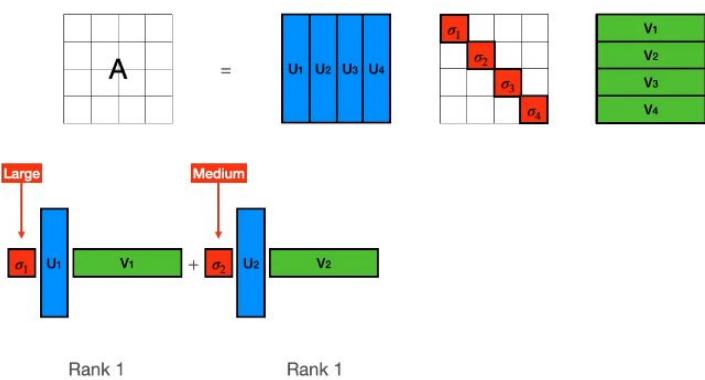
Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)



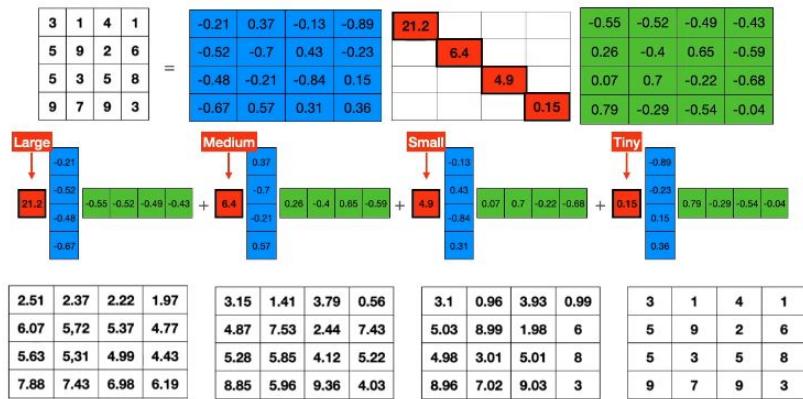
Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)



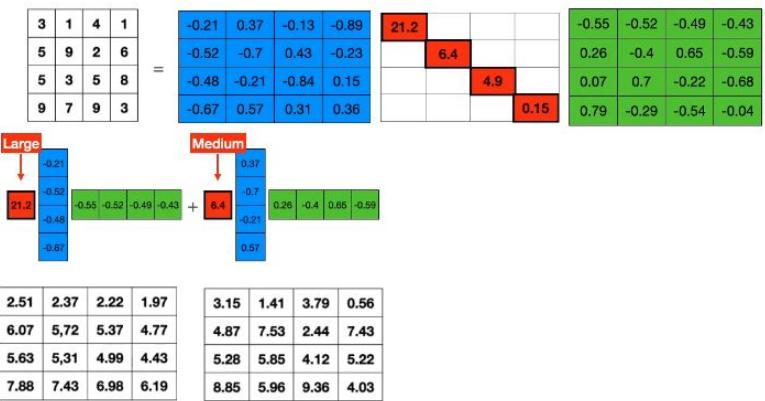
Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)



Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

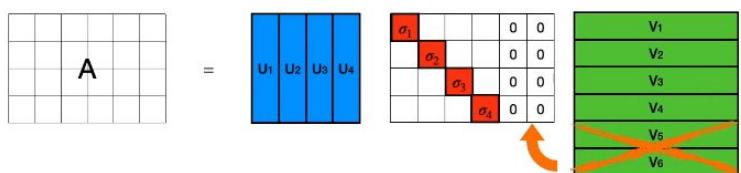


Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)



Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

Μη τετραγωνικός πίνακας;



Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

Ανάλυση διανυσμάτων πάνω σε ορθογώνιους άξονες

- Αν ένας πίνακας δεν είναι τετραγωνικός, η ανάλυση ιδιοτιμών δεν ορίζεται.
 - Αντ' αυτού, χρησιμοποιούμε την ανάλυση σε Ιδιάζουσες τιμές
- Κάθε πίνακας A ($m \times n$) μπορεί να γραφεί ως $A = U\Sigma V^T$

Σημαντική εφαρμογή: Μπορούμε να γενικεύσουμε, εν μέρει, την αντιστροφή ενός μη τετραγωνικού πίνακα

Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

$$A = U\Sigma V^T$$

Όπου U : Ορθογώνιος πίνακας $m \times m$ → Περιστροφή (Rotation)

Σ : Διαγώνιος πίνακας διάστασης $m \times n$ → Έκταση (Stretch)
(όχι απαραίτητα τετραγωνικός)

V : Ορθογώνιος πίνακας $n \times n$ → Περιστροφή (Rotation)

$$A = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix}$$

Left singular vector singular values Right singular vector

wikipedia:SV

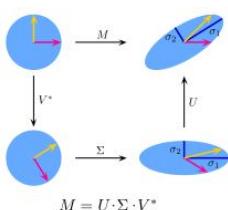
Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

$$A = U\Sigma V^T$$

$$A^T A = (V\Sigma^T U^T)U\Sigma V^T = V\Sigma^T(U^T U)\Sigma V^T = V(\Sigma^T \Sigma)V^T$$

$$AA^T = U\Sigma V^T(V\Sigma^T U^T) = U\Sigma(V^T V)\Sigma^T U^T = U(\Sigma \Sigma^T)U^T$$

$$\begin{aligned} M &= U \Sigma V^T \\ M_{m \times n} &= U_{m \times m} \Sigma_{m \times n} V^*_{n \times n} \\ U \cdot U^* &= I_m \\ V \cdot V^* &= I_n \end{aligned}$$



wikipedia:SV

<http://stefansavvy.com/blog/singular-value-decomposition-all-posts/>

Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

Τι είναι τα U, Σ, V^T

Ιδιος πίνακας

$$A = V\Lambda V^{-1}$$

Διαφορετικοί Πίνακες

$$A = U\Sigma V^T$$

Βήμα προς βήμα με python

Deep Learning Book Series · 2.8 Singular Value Decomposition

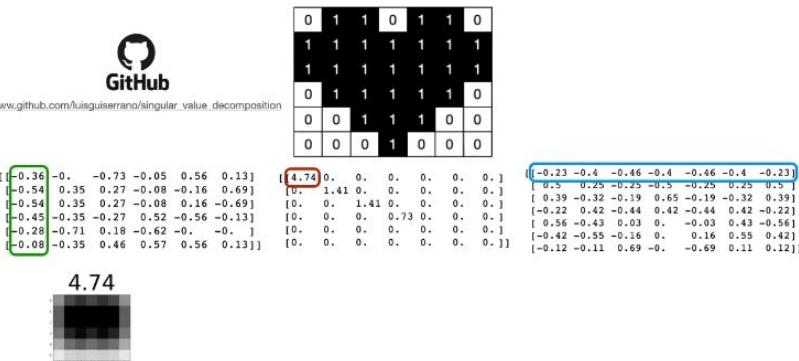
A : πίνακας που μπορούμε να τον “δούμε” ως γραμμικό μετασχηματισμό που μπορεί να αναλυθεί σε 3 υπό- μετασχηματισμούς (αντιστοιχούν σε 3 πίνακες) :

1. Περιστροφή,

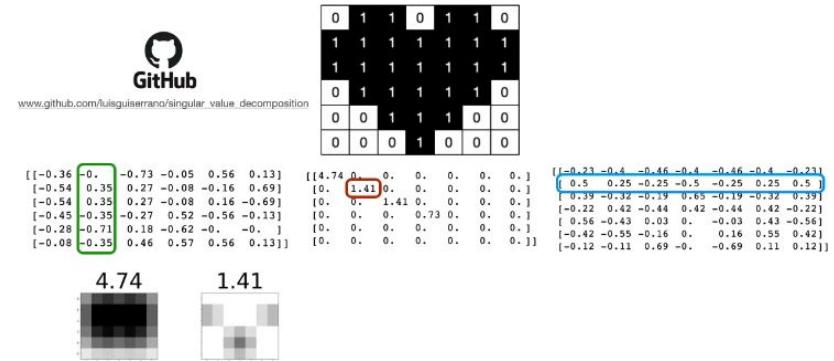
2. Έκταση,

3. Περιστροφή.

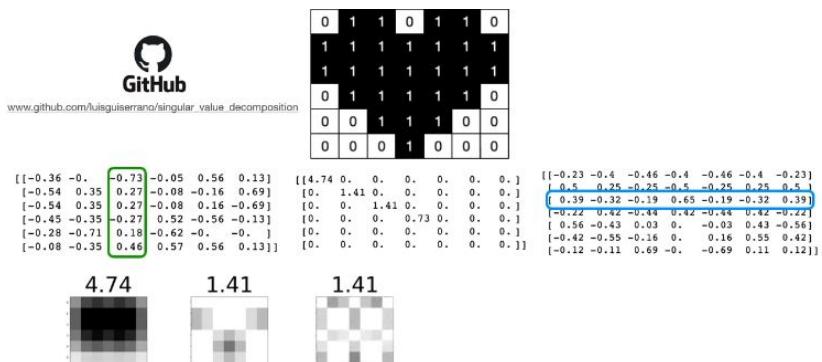
Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)



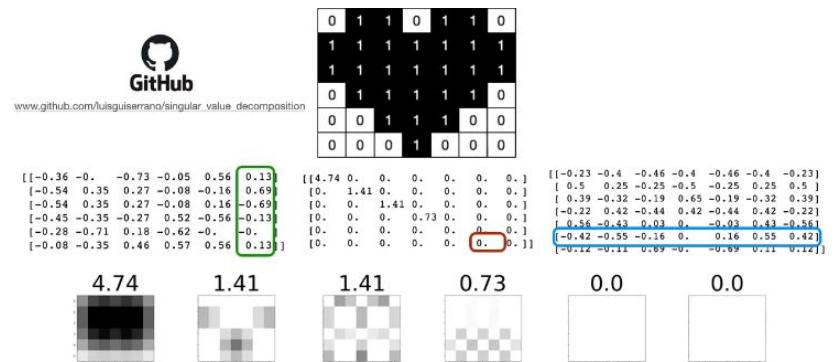
Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)



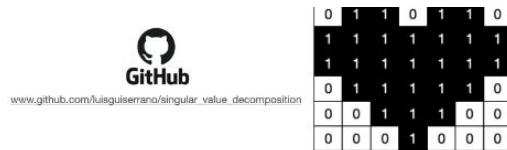
Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)



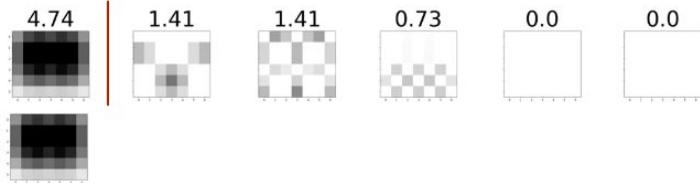
Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)



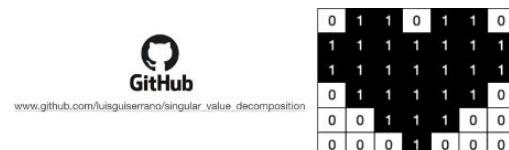
Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)



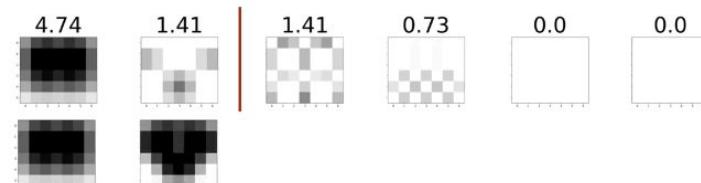
```
[[-0.36 -0. -0.73 -0.05 0.56 0.13] [[4.74 0. 0. 0. 0. 0. 0.] [[-0.23 -0.4 -0.46 -0.4 -0.46 -0.4 -0.23]
[-0.54 0.35 0.27 -0.08 -0.16 0.69] [0. 1.41 0. 0. 0. 0. 0.] [-0.5 0.25 -0.25 -0.5 -0.25 0.25 0.5 ]
[-0.54 0.35 0.27 -0.08 0.16 -0.69] [0. 0. 1.41 0. 0. 0. 0.] [-0.39 -0.32 -0.19 0.65 -0.19 -0.32 0.39]
[-0.45 -0.35 -0.27 0.52 -0.56 -0.13] [0. 0. 0. 0. 0.73 0. 0. 0.] [-0.22 0.42 -0.44 0.42 -0.44 0.42 -0.22]
[-0.28 -0.71 0.18 -0.62 -0. -0. ] [0. 0. 0. 0. 0. 0. 0.] [0.56 -0.43 0.03 0. -0.03 0.43 -0.56]
[-0.08 -0.35 0.46 0.46 0.57 0.56 0.13]] [0. 0. 0. 0. 0. 0. 0.] [-0.42 -0.55 -0.16 0. 0.16 0.55 0.42]
[-0.12 -0.11 0.69 -0. -0.69 0.11 0.12]]]
```



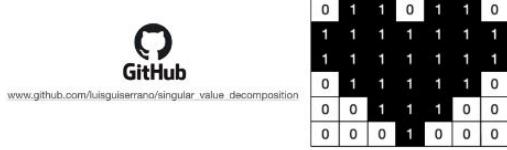
Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)



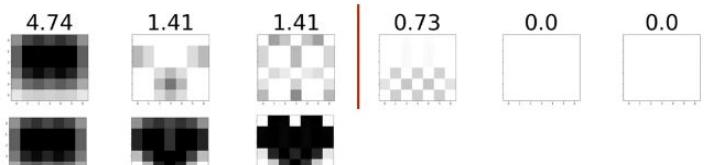
```
[[-0.36 -0. -0.73 -0.05 0.56 0.13] [[4.74 0. 0. 0. 0. 0. 0.] [[-0.23 -0.4 -0.46 -0.4 -0.46 -0.4 -0.23]
[-0.54 0.35 0.27 -0.08 -0.16 0.69] [0. 1.41 0. 0. 0. 0. 0.] [-0.5 0.25 -0.25 -0.5 -0.25 0.25 0.5 ]
[-0.54 0.35 0.27 -0.08 0.16 -0.69] [0. 0. 1.41 0. 0. 0. 0.] [-0.39 -0.32 -0.19 0.65 -0.19 -0.32 0.39]
[-0.45 -0.35 -0.27 0.52 -0.56 -0.13] [0. 0. 0. 0. 0.73 0. 0. 0.] [-0.22 0.42 -0.44 0.42 -0.44 0.42 -0.22]
[-0.28 -0.71 0.18 -0.62 -0. -0. ] [0. 0. 0. 0. 0. 0. 0.] [0.56 -0.43 0.03 0. -0.03 0.43 -0.56]
[-0.08 -0.35 0.46 0.46 0.57 0.56 0.13]] [0. 0. 0. 0. 0. 0. 0.] [-0.42 -0.55 -0.16 0. 0.16 0.55 0.42]
[-0.12 -0.11 0.69 -0. -0.69 0.11 0.12]]]
```



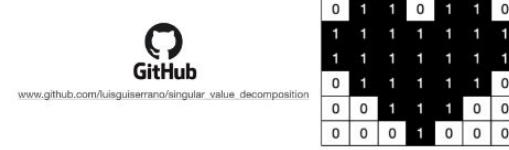
Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)



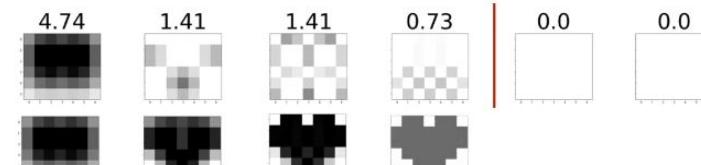
```
[[-0.36 -0. -0.73 -0.05 0.56 0.13] [[4.74 0. 0. 0. 0. 0. 0.] [[-0.23 -0.4 -0.46 -0.4 -0.46 -0.4 -0.23]
[-0.54 0.35 0.27 -0.08 -0.16 0.69] [0. 1.41 0. 0. 0. 0. 0.] [-0.5 0.25 -0.25 -0.5 -0.25 0.25 0.5 ]
[-0.54 0.35 0.27 -0.08 0.16 -0.69] [0. 0. 1.41 0. 0. 0. 0.] [-0.39 -0.32 -0.19 0.65 -0.19 -0.32 0.39]
[-0.45 -0.35 -0.27 0.52 -0.56 -0.13] [0. 0. 0. 0. 0.73 0. 0. 0.] [-0.22 0.42 -0.44 0.42 -0.44 0.42 -0.22]
[-0.28 -0.71 0.18 -0.62 -0. -0. ] [0. 0. 0. 0. 0. 0. 0.] [0.56 -0.43 0.03 0. -0.03 0.43 -0.56]
[-0.08 -0.35 0.46 0.46 0.57 0.56 0.13]] [0. 0. 0. 0. 0. 0. 0.] [-0.42 -0.55 -0.16 0. 0.16 0.55 0.42]
[-0.12 -0.11 0.69 -0. -0.69 0.11 0.12]]]
```



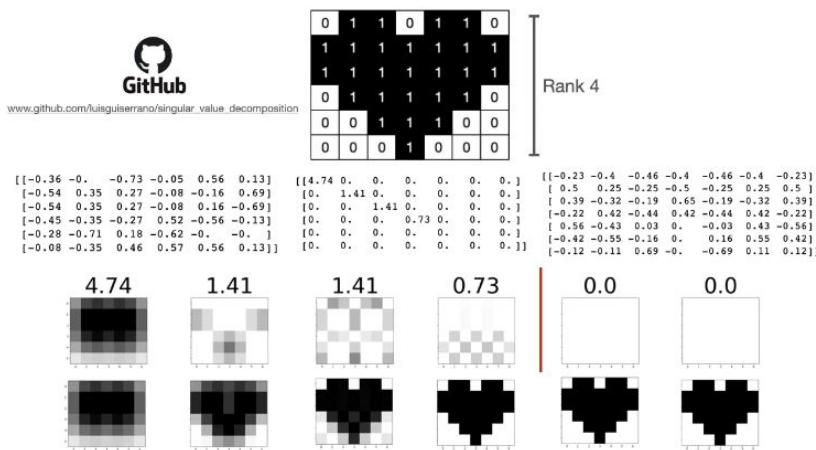
Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)



```
[[-0.36 -0. -0.73 -0.05 0.56 0.13] [[4.74 0. 0. 0. 0. 0. 0.] [[-0.23 -0.4 -0.46 -0.4 -0.46 -0.4 -0.23]
[-0.54 0.35 0.27 -0.08 -0.16 0.69] [0. 1.41 0. 0. 0. 0. 0.] [-0.5 0.25 -0.25 -0.5 -0.25 0.25 0.5 ]
[-0.54 0.35 0.27 -0.08 0.16 -0.69] [0. 0. 1.41 0. 0. 0. 0.] [-0.39 -0.32 -0.19 0.65 -0.19 -0.32 0.39]
[-0.45 -0.35 -0.27 0.52 -0.56 -0.13] [0. 0. 0. 0. 0.73 0. 0. 0.] [-0.22 0.42 -0.44 0.42 -0.44 0.42 -0.22]
[-0.28 -0.71 0.18 -0.62 -0. -0. ] [0. 0. 0. 0. 0. 0. 0.] [0.56 -0.43 0.03 0. -0.03 0.43 -0.56]
[-0.08 -0.35 0.46 0.46 0.57 0.56 0.13]] [0. 0. 0. 0. 0. 0. 0.] [-0.42 -0.55 -0.16 0. 0.16 0.55 0.42]
[-0.12 -0.11 0.69 -0. -0.69 0.11 0.12]]]
```



Ανάλυση σε Ιδιάζουσες Τιμές (Singular Value Decomposition)

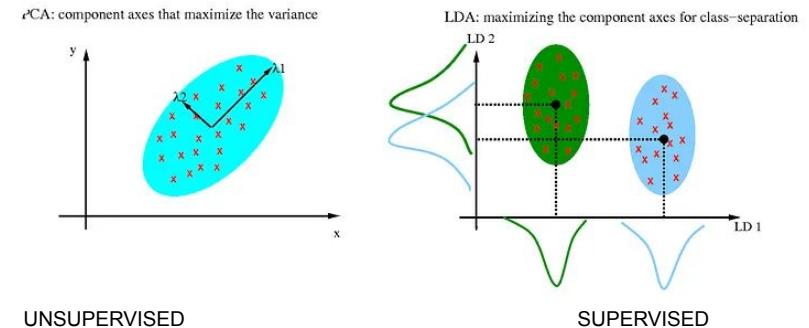


PCA vs LDA (Linear Discriminant Analysis)

To PCA εστιάζει κυρίως στη μεγαλύτερη απόκλιση μεταξύ όλων των μεταβλητών.

→ To LDA μας ενδιαφέρει να μεγιστοποιήσουμε τη διαχωριστικότητα μεταξύ όλων των γνωστών κατηγοριών.

→ To LDA προβάλλει τα δεδομένα με τρόπο που μεγιστοποιεί τον διαχωρισμό δύο κατηγοριών.



LDA (Linear Discriminant Analysis)

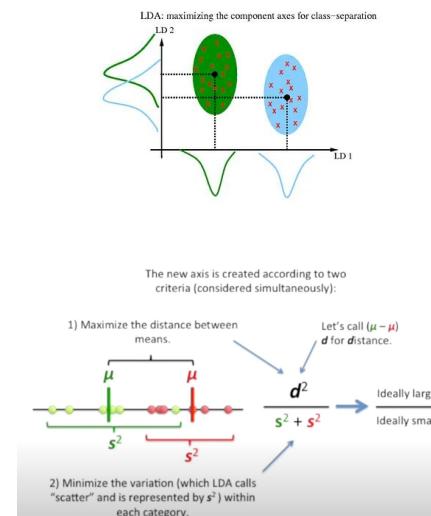
Δύο κριτήρια:

- Μεγιστοποιήστε την απόσταση μεταξύ των μέσων των κατηγοριών
- Ελαχιστοποιήστε τη διασπορά σε κάθε κατηγορία

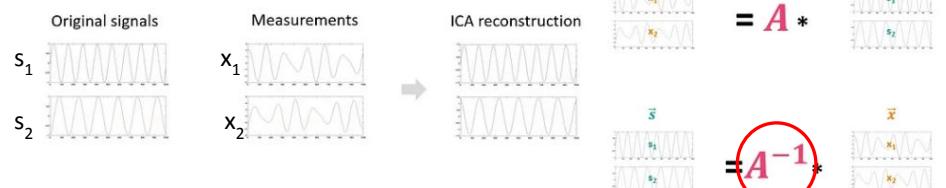
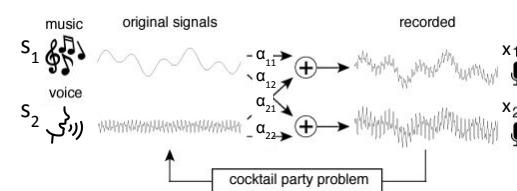
Βήμα 1: Υπολογίζεται ο μέσος όρος και η τυπική απόκλιση κάθε χαρακτηριστικού εντός της κλάσης

Βήμα 2: Υπολογίζονται οι πίνακες διασποράς εντός και μεταξύ των κλάσεων και χρησιμοποιούνται για τον υπολογισμό των ιδιοιανυσμάτων και των ιδιοτιμών.

Βήμα 3: Κατασκευάζεται χώρος χαμηλότερων διαστάσεων που μεγιστοποιεί τη διακύμανση μεταξύ κλάσεων και ελαχιστοποιεί τη διακύμανση εντός κλάσης



Independent Component Analysis (ICA)



$$x_1(t) = a_{11}s_1 + a_{12}s_2$$

$$x_2(t) = a_{21}s_1 + a_{22}s_2$$

Assume s_1, s_2 : statistically independent

ICA: Στατιστική παρουσίαση

Θεωρήστε ότι δύο ανεξάρτητα στοιχεία (πηγές) περιγράφονται από τις ομοιόμορφες κατανομές:

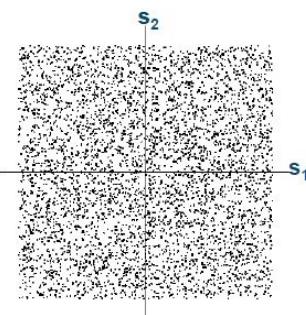
$$p(s_i) = \begin{cases} \frac{1}{2\sqrt{3}} & \text{if } |s_i| \leq \sqrt{3} \\ 0 & \text{otherwise} \end{cases}$$

Αυτή η ομοιόμορφη κατανομή έχει μηδενικό μέσο όρο και η διακύμανση είναι ίση με 1.

Ας υποθέσουμε ότι αυτές οι δύο μεμονωμένες πηγές αναμειγνύονται από τον ακόλουθο πίνακα:

$$A_0 = \begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}$$

Joint distribution s_i



Οι x μπορούν να υπολογιστούν χρησιμοποιώντας το μοντέλο ICA: $x = As$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \Rightarrow \begin{cases} x_1 = 2s_1 + 3s_2 \\ x_2 = 2s_1 + s_2 \end{cases}$$

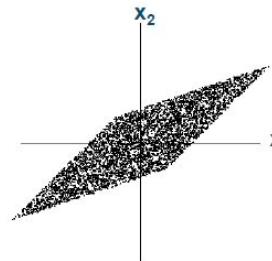
ICA: Στατιστική παρουσίαση

Σημειώστε ότι οι τυχαίες μεταβλητές x_1, x_2 δεν είναι πλέον ανεξάρτητες.

- Ένας εύκολος τρόπος για να το δείτε αυτό είναι να εξετάσετε, εάν είναι δυνατόν να προβλεφθεί η τιμή πχ. της x_2 από την τιμή της x_1
- Παρατηρήστε τον πίνακα A

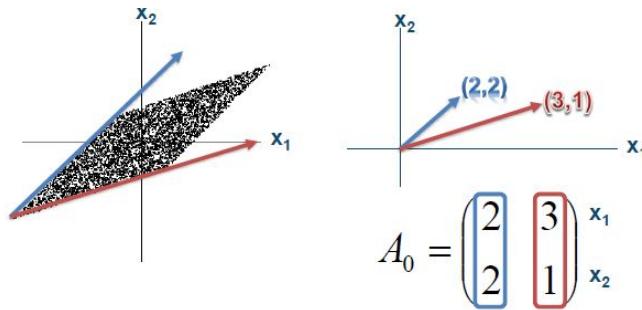
$$A_0 = \begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}$$

Joint distribution of mixture x_1, x_2



ICA: Στατιστική παρουσίαση

Οι ακμές του παραληλογράμμου είναι οι κατευθύνσεις των στηλών του A_0 .



Αυτό σημαίνει ότι θα μπορούσαμε, κατ' αρχήν, να εκτιμήσουμε το μοντέλο ICA υπολογίζοντας πρώτα την από κοινού κατανομή (joint distribution) των x_1, x_2 , και στη συνέχεια να εντοπίσουμε τις ακμές.

Άρα, το πρόβλημα φαίνεται να έχει λύση.

ICA - Blind Source Separation (BSS)

Στόχος: Προσπαθούμε να εξάγουμε τα ανεξάρτητα συστατικά από τα αναμειγμένα δεδομένα, χωρίς να γνωρίζουμε εκ των προτέρων το πώς έγινε η ανάμειξη των σημάτων.

Σήμα πηγής: Το αυθεντικό, ανεξάρτητο σήμα (π.χ. μια φωνή)

Αναμειγμένα σήμα: Η συνισταμένη δύο ή περισσότερων πηγών που έχουν αναμειχθεί.

Ανάμειξη Σημάτων - ICA Υποθέσεις

Όταν αναμειγνύονται δύο ή περισσότερα σήματα (π.χ. φωνές), παρατηρούνται τρία βασικά αποτελέσματα:

1. **Ανεξαρτησία:** Τα σήματα πηγής είναι στατιστικά ανεξάρτητα, αλλά τα αναμειγμένα σήματα δεν είναι, διότι κάθε ανάμειξη περιέχει και τα δύο σήματα.
2. **Κανονικότητα:** Τα σήματα πηγής έχουν κατανομές που δεν είναι Γκαουσιανές, ενώ τα αναμειγμένα σήματα τείνουν να έχουν Γκαουσιανές κατανομές.
3. **Πολυπλοκότητα:** Η πολυπλοκότητα των αναμειγμένων σημάτων είναι μεγαλύτερη ή ίση με την πολυπλοκότητα των πιο απλών σημάτων πηγής (ο άγνωστος πίνακας ανάμειξης είναι τετράγωνος).

Independent Component Analysis (ICA)

Για να ανακτήσουμε τις αρχικές πηγές από τα αναμεμειγμένα σήματα, χρησιμοποιούμε τις παρακάτω τρεις στρατηγικές:

- Ανεξαρτησία:** Εφόσον τα σήματα πηγής είναι ανεξάρτητα, ενώ τα αναμεμειγμένα δεν είναι, $p(x,y) = p(x)p(y)$ ανεξάρτητα σήματα από τα αναμεμειγμένα σήματα μας επιπρέπει να ανακτήσουμε τις πηγές.
→ Αυτό το κάνει το ICA μέσω της μεγιστοποίησης της στατιστικής ανεξαρτησίας των εξαγόμενων σημάτων.
- Κανονικότητα:** Εάν τα σήματα πηγής έχουν μη-Γκαουσιανές κατανομές και τα ανα⁺ γκαουσιανές κατανομές, τότε η εξαγωγή σημάτων με μη-Γκαουσιανές κατανομές από τα αναμεμειγμένα σήματα μας επιπρέπει να ανακτήσουμε τις πηγές. 
- Πολυπλοκότητα:** Εάν τα σήματα πηγής είναι πιο απλά (με χαμηλή πολυπλοκότητα) σε σχέση με τα αναμεμειγμένα σήματα, τότε η εξαγωγή των πιο απλών συστατικών θα μας δώσει τα αρχικά σήματα.

Η στρατηγική είναι να αναζητήσουμε την ιδιότητα (ανεξαρτησία, μη-Γκαουσιανές κατανομές, ή απλότητα) που

Independent Component Analysis (ICA): Παραδοχές

Αριθμός Πηγών: Πρέπει τουλάχιστον ο αριθμός αναμεμειγμένων σημάτων να είναι ίσος με τον αριθμό των πηγών.

Αναμεμειγμένα Σήματα: Αν ο αριθμός των πηγών είναι μεγαλύτερος από τον αριθμό των αναμεμειγμένων σημάτων, δεν μπορεί εύκολα να ανακτηθεί όλος ο αριθμός των πηγών.

Σύγκριση Στρατηγικών για το BSS

- Υπάρχουν διάφορες μέθοδοι για την εφαρμογή του BSS, η κάθε μία με τις δικές της παραδοχές (minimization of mutual information, non-Gaussianity maximization)
- Η μέθοδος που επιλέγεται εξαρτάται από την φύση των δεδομένων και τις υποθέσεις που κάνουμε για τα σήματα και τη διαδικασία ανάμειξης.

ICA - Minimization of mutual information

- Βασίζεται στη θεωρία της πληροφορίας.
- Είναι μέτρο της στατιστικής εξάρτησης μεταξύ συνιστωσών ενός τυχαίου διανύσματος

Διαφορική Εντροπία (H): Η διαφορική εντροπία H ενός τυχαίου διανύσματος y με πυκνότητα πιθανότητας $p(y)$ ορίζεται ως:

$$H(y) = - \int p(y) \log p(y)$$

Αρνητική Εντροπία (Negentropy - J): Μια κανονικοποιημένη έκδοση της εντροπίας, ορίζεται ως αρνητική εντροπία J :

$$J(y) = H(y_{gauss}) - H(y)$$

όπου το y_{gauss} είναι ένα Γκαουσιανό τυχαίο διάνυσμα με τον ίδιο πίνακα συσχέτιση όπως το y .

Η αρνητική εντροπία είναι πάντα θετική, και μηδέν μόνο για Gaussian random vectors

Αμοιβαία Πληροφορία (I): Η αμοιβαία πληροφορία μεταξύ m τυχαίων μεταβλητών y_i , όπου $i=1,...,m$ δίνεται από:

$$I(y_1, y_2, \dots, y_m) = \sum_{i=1}^m H(y_i) - H(y)$$

ICA - Minimization of mutual information

Χρησιμοποιώντας την έννοια της διαφορικής εντροπίας, η αμοιβαία πληροφορία (mutual information) I μεταξύ m τυχαίων μεταβλητών δίνεται από:

$$I(y_1, y_2, \dots, y_m) = \sum_{i=1}^m H(y_i) - H(y)$$

- Η αμοιβαία πληροφορία είναι το φυσικό μέτρο της εξάρτησης μεταξύ τυχαίων μεταβλητών.
- Η τιμή της είναι πάντα μη αρνητική και μηδενική εάν και μόνο εάν οι μεταβλητές εξαρτώνται στατιστικά.

Όταν το αρχικό τυχαίο διάνυσμα x υφίσταται έναν αντιστρέψιμο γραμμικό μετασχηματισμό $y = Wx$ ($x=As$), η αμοιβαία πληροφορία για το y ως προς το x είναι :

$$I(y_1, \dots, y_m) = \sum_i H(y_i) - H(x) - \log |\det W|$$

ICA - Minimization of mutual information

Υποθέστε ότι το y_i περιορίζεται να είναι ασυχέτιστο και με μοναδιαία διακύμανση, τότε θα ισχύει:

$$E\{yy^T\} = \mathbf{W}E\{\mathbf{xx}^T\}\mathbf{W}^T = \mathbf{I}$$

Η εφαρμογή της ορίζουσας σε όλες τις πλευρές της εξίσωσης έχουμε:

$$I(y_1, \dots, y_m) = \sum_i H(y_i) - H(\mathbf{x}) - \log |\det \mathbf{W}| \quad \rightarrow \quad \det I = 1 = \det(W E\{\mathbf{xx}^T\} W^T) = (\det W)(\det E\{\mathbf{xx}^T\})(\det W^T)$$

- Το $\det \mathbf{W}$ πρέπει να είναι σταθερό αφού $\det E\{\mathbf{xx}^T\}$ δεν εξαρτάται από το \mathbf{W} .
- Για y_i μοναδιαίας διακύμανσης, η εντροπία και η αρνητική εντροπία διαφέρουν μόνο κατά σταθερά και πρόσημο.

Επομένως, η θεμελιώδης σχέση ανάμεσα στην εντροπία και την αρνητική εντροπία είναι:

$$I(y_1, \dots, y_n) = C - \sum_i J(y_i)$$

όπου C είναι μια σταθερά που δεν εξαρτάται από το \mathbf{W} .

⇒ Βρίσκοντας έτσι έναν αντιστρέψιμο μετασχηματισμό \mathbf{W} που ελαχιστοποιεί την αμοιβαία πληροφορία είναι περίπου ισοδύναμο με την εύρεση κατευθύνσεων στις οποίες η αρνητική εντροπία (μια έννοια που σχετίζεται σε μη γκαουσιανή κατανομή) μεγιστοποιείται.

Maximum Likelihood Estimation

Με τη χρήση της πυκνότητας πιθανότητας ενός γραμμικού μετασχηματισμού υπολογίζεται η πυκνότητα p_x του αναμεμειγμένου δεδομένου $x = As$ όπου $W = A^{-1}$, και f_i δηλώνουν τις πυκνότητες των ανεξάρτητων συστατικών s_i :

$$f_x(x) = |\det W| f_s(s) = |\det W| \prod_{i=1}^n f_i(s_i)$$

Η πυκνότητα p_x μπορεί επίσης να εκφραστεί ως

$$f_x(x) = |\det W| \prod_{i=1}^n f_i(w_i^T x)$$

αν του $W = (w_1, w_2 \dots w_n)^T$, δηλαδή,

$$L = \sum_{t=1}^T \sum_{i=1}^n \log f_i(w_i^T x(t)) + T \log |\det W|$$

-

Πρόβλημα: Οι συναρτήσεις πυκνότητας f_i πρέπει να εκτιμηθούν σωστά, διαφορετικά η Maximum Likelihood Estimation θα δώσει λάθος αποτέλεσμα.

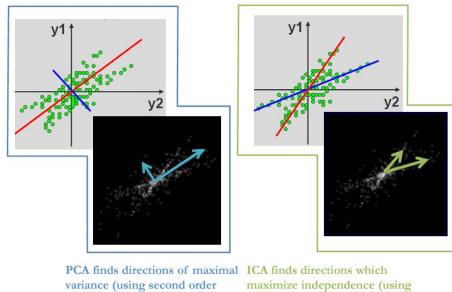
Independent Component Analysis(ICA) vs PCA

Ομοιότητες:

- Εξαγωγή χαρακτηριστικών
- Μείωση διαστάσεων.

Σύγκριση:

- Το PCA βρίσκει κατευθύνσεις μέγιστης διακύμανσης.
- Το ICA βρίσκει κατευθύνσεις μέγιστης ανεξαρτησίας.

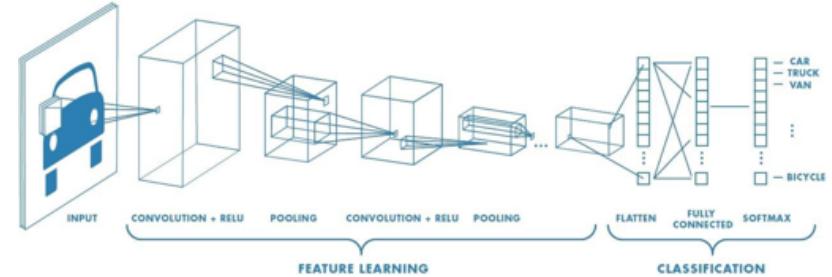




Convolutional Neural Network

Convolutional Neural Networks

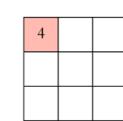
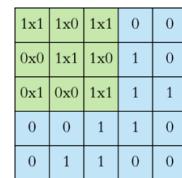
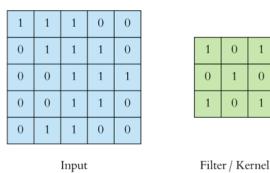
Νευρωνικά Δίκτυα και Βαθιά Μάθηση



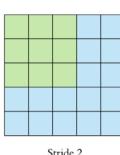
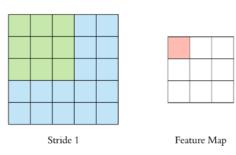
[Understanding of Convolutional Neural Network \(CNN\) – Deep Learning](#), Prabhu in Towards Data Science

Convolutional Neural Network

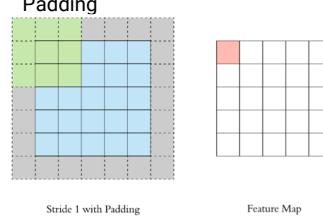
Convolution Layer



Stride



Padding



Convolutional Neural Network

Οπότε, αν στην είσοδο του Convolution Layer έχω τένσορες διάστασης $W_1 \times H_1 \times D_1$ απαιτείται ο ορισμός 4 υπερπαραμέτρων:

1. το πλήθος των φίλτρων (K)
2. το μέγεθος των φίλτρων (F)
3. το stride (S)
4. το padding (P)

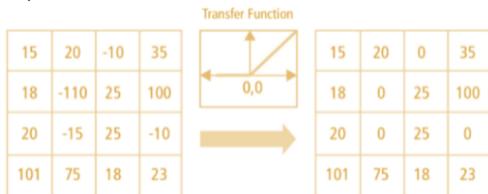
Στην έξοδο του Convolution Layer θα πάρω τένσορες μεγέθους $W_2 \times H_2 \times D_2$ όπου:

1. $W_2 = \frac{(W_1 - F + 2*P)}{S} + 1$
2. $H_2 = \frac{(H_1 - P + 2*P)}{S} + 1$
3. $D_2 = K$

Convolutional Neural Network

ReLU: Rectified Linear Unit for a non-linear operation

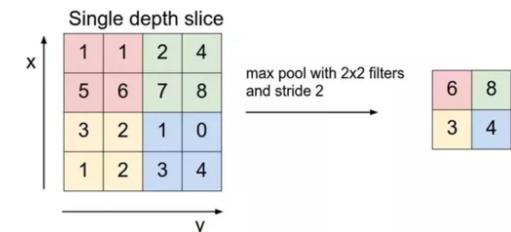
$$f(x) = \max(0, x).$$



Convolutional Neural Network

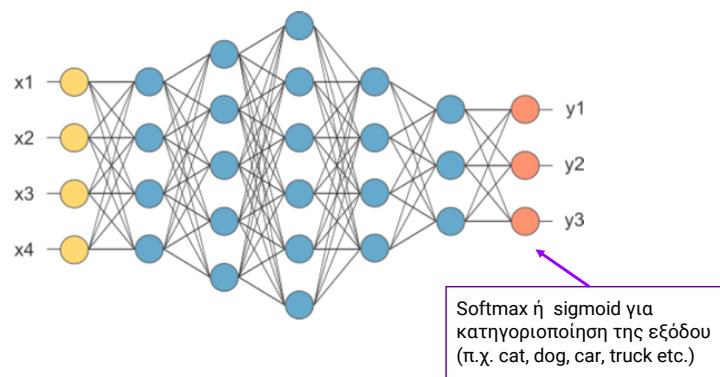
Pooling Layer

- Max Pooling
- Average Pooling
- Sum Pooling



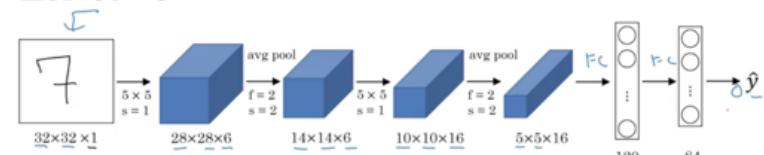
Convolutional Neural Network

Fully Connected Layer



CNN Architectures: LeNet

LeNet - 5



$$W \times H \rightarrow 32 \times 32 \text{ (Width x Height)}$$

$$F(w \times h) \rightarrow 5 \times 5 \text{ (Filter)}$$

$$S = 1 \text{ (Stride)}$$

$$P = 0 \text{ (Pooling)}$$

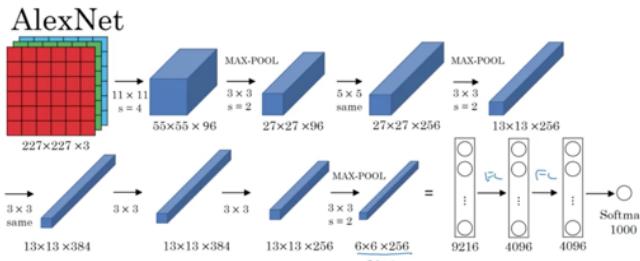
$$\left(\frac{W - Fw + 2P}{Sw} \right) + I \Rightarrow \left(\frac{32 - 5 + 0}{I} \right) + I = > 27 + I = > 28$$

$$\left(\frac{H - Fh + 2P}{Sh} \right) + I \Rightarrow \left(\frac{32 - 5 + 0}{I} \right) + I = > 27 + I = > 28$$

$$Output Volume = 28 \times 28$$

CNN Architectures: AlexNet

- Activation function
 - ReLU (όχι Sigmoid ή Tanh)
 - 5 x ταχύτητα,
 - ίδια ακρίβεια
- OverFitting
 - Dropout
 - Διπλασιασμός χρόνου εκπαίδευσης
- Περισσότερα δεδομένα και μεγαλύτερο μοντέλο
 - 7 hidden layers, 650K units και 60M parameters.



[Krizhevsky et al., 2012. ImageNet classification with deep convolutional neural networks]

Andrew Ng

CNN Architectures: Inception (GoogLeNet)

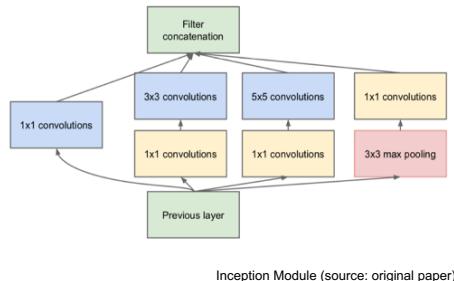
[C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition \(CVPR\), Boston, MA, 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594.](#)

- Διαδοχικά αλλά και παράλληλα CNN (error rate 6.7%)
- Πολλαπλοί πυρήνες διαφορετικών μεγεθών εφαρμόζονται στο ίδιο επίπεδο με σκοπό τη ανίχνευση συγκεκριμένων χαρακτηριστικών περιοχής
 - ◆ Μεγάλοι πυρήνες → καθολικά χαρακτηριστικά που κατανέμονται σε μεγάλη περιοχή της εικόνας,
 - ◆ Μικροί πυρήνες → ανίχνευση συγκεκριμένων χαρακτηριστικών περιοχής που κατανέμονται σε ολόκληρο το πλαίσιο εικόνας.

CNN Architectures: Inception (GoogLeNet)

Mονάδα Inception :

Καταγράφει προεξέχοντα χαρακτηριστικά (salient features) σε διαφορετικά επίπεδα.



- 4 παράλληλες λειτουργίες
 - ◆ 1x1 conv layer, μείωση βάθους
 - ◆ 3x3 conv layer, Κατανεμημένα χαρακτηριστικά (distributed features)
 - ◆ 5x5 conv layer, Γενικά χαρακτηριστικά (global features)
 - ◆ max pooling, Χαμηλού επιπέδου χαρακτηριστικά (low level features)

- Φίλτρο συνένωσης

π.χ. εάν οι εικόνες στο σύνολο δεδομένων έχουν πολλά καθολικά χαρακτηριστικά και ελάχιστα χαρακτηριστικά χαμηλού επιπέδου, τότε το εκπαιδευμένο δίκτυο Inception θα έχει πολύ μικρά βάρη που αντιστοιχούν στον πυρήνα 3x3 σε σύγκριση με τον πυρήνα 5x5.

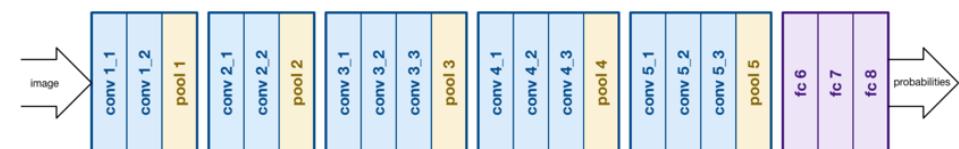
CNN Architectures : VGG

[Simonyan, K. and Zisserman, A. \(2015\) Very Deep Convolutional Networks for Large-Scale Image Recognition. 3rd International Conference on Learning Representations \(ICLR2015\).](#)

13 συνελικτικά και 3 πλήρως συνδεδεμένα επίπεδα, ReLU, φίλτρα μικρότερου μεγέθους (2×2 και 3×3) από το AlexNet , 138M παραμέτρους, 500MB.

Ανέδειξαν τη σημασία του βάθους του δικτύου ως σημαντικής παραμέτρου για την αποτελεσματικότητά του.

- Μείωση του αριθμού των παραμέτρων στα επίπεδα CONV
- Βελτίωση του χρόνου εκπαίδευσης
- Σχεδίασαν επίσης βαθύτερες παραλλαγές, VGG-16, VGG-19.



CNN Architectures : VGG-16

Η ιδέα πίσω από την ύπαρξη πυρήνων σταθερού μεγέθους είναι ότι όλοι οι conv πυρήνες μεταβλητού μεγέθους που χρησιμοποιούνται στο Alexnet (11×11 , 5×5 , 3×3) μπορούν να αναπαραχθούν χρησιμοποιώντας πολλαπλούς πυρήνες 3×3 ως δομικά στοιχεία.

π.χ. Έστω επίπεδο εισόδου μεγέθους $5 \times 5 \times 1$

Περίπτωση 1: 1ο conv επίπεδο: ένας πυρήνας 5×5 και βήμα 1 → Έξοδος: χάρτης χαρακτηριστικών 1×1

Πλήθος μεταβλητών $5 \times 5 \times 1 = 25$ ($(m \times n + 1) \times k$, k: πλήθος πυρήνων)

Περίπτωση 2: 1ο conv επίπεδο: δύο πυρήνες 3×3 και βήμα 1 → Έξοδος: χάρτης χαρακτηριστικών 1×1 .

Πλήθος μεταβλητών $3 \times 3 \times 2 = 18$ → Μείωση 28%

Αντίστοιχα αν αντί για χρήση πυρήνων 7×7 (11×11) εφαρμόσουμε 3 (5) 3×3 πυρήνες → μείωση αριθμού εκπαίδευμένων μεταβλητών κατά 44,9% (62,8%)

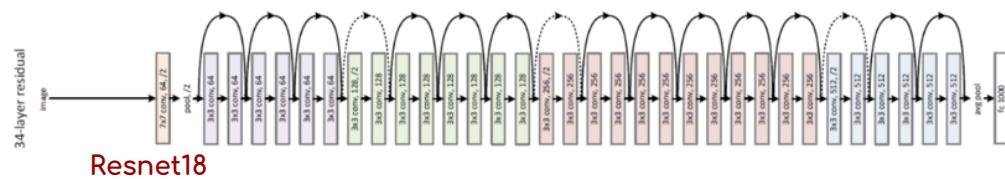
★ Ταχύτερη εκμάθηση



Αποφυγή overfitting

CNN Architectures: ResNet (MSRA)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun Deep Residual Learning for Image Recognition, CVPR 2015



Resnet18

- ◆ 152 επίπεδα, 11M παράμετροι, πυρήνες, 3×3 (όπως το VGGNet), 2 pooling επίπεδα

Σύνδεση ταυτότητας (Identity connection) ανά δύο επιπέδων CONV, διάσταση εισόδου ίδια με της εξόδου

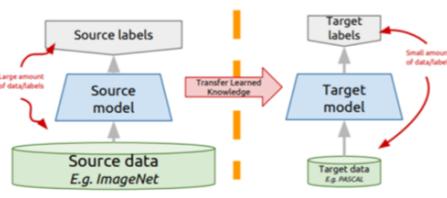
Σύνδεση προβολής (Projection connection) όπου οι διαστάσεις εισόδου διαφέρουν με της εξόδου.

Υπάρχουν πολλές εκδόσεις αρχιτεκτονικών ResNetXX όπου το «XX» υποδηλώνει τον αριθμό των επιπέδων (ResNet50, ResNet101)

Transfer Learning για Deep Learning

Ορισμός

Με δεδομένη μια εργασία Transfer Learning που ορίζεται από $\langle D_s, T_s, D_t, T_t, f_T(\cdot) \rangle$, η "μεταφορά μάθησης" στοχεύει στη μάθηση της μη γραμμικής συνάρτησης f_T που αντικατοπτρίζει ένα βαθύ νευρωνικό δίκτυο.



Στρατηγικές Deep Transfer Learning

- Προεκπαίδευμένα μοντέλα για εξαγωγή χαρακτηριστικών

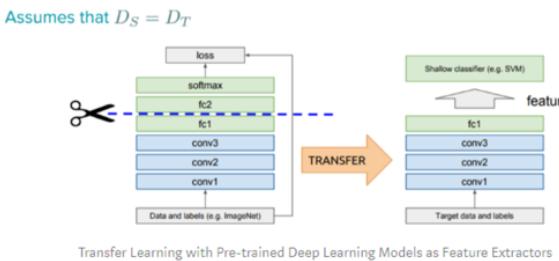
Off-the-shelf Pre-trained Models as fixed Feature Extractors

- Ακριβής προσαρμογή προεκπαίδευμένων μοντέλων

Fine Tuning Off-the-shelf Pre-trained Models

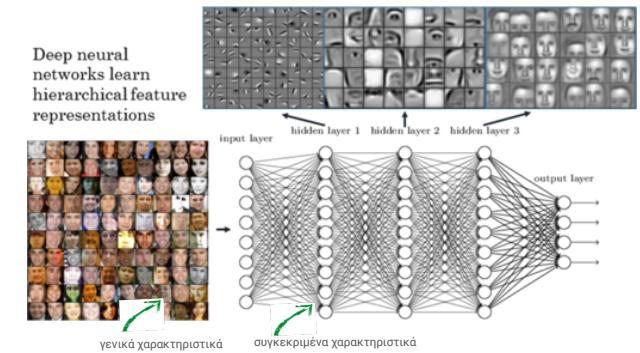
Προεκπαίδευμένα μοντέλα για εξαγωγή χαρακτηριστικών

- Η έξοδος μετά από κάποιο επίπεδο ενός δικτύου βαθιάς μάθησης, που εκπαιδεύτηκε σε διαφορετική εργασία ($T_s \neq T_t$), χρησιμοποιείται ως γενικευμένος ανιχνευτής χαρακτηριστικών.
- Εκπαίδευση νέου μοντέλου (π.χ. SVM) με μεταφορά αυτών των χαρακτηριστικών.



Ακριβής προσαρμογή προεκπαίδευμένων μοντέλων

Δεν αντικαθιστούμε απλώς το τελικό επίπεδο (για ταξινόμηση / παλινδρόμηση), αλλά επανεκπαίδευουμε επιλεκτικά ορισμένα από τα προηγούμενα επίπεδα.

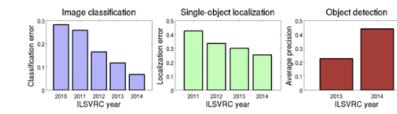


Πρακτικές συμβουλές

- Περιορισμοί από προκατασκευασμένα μοντέλα.
 - ◆ Η χρήση ενός προκαθορισμένου δίκτυου, ενδέχεται να είναι δεσμευτική ως προς την αρχιτεκτονική που μπορείτε να χρησιμοποιήσετε για το νέο σύνολο δεδομένων.
 - π.χ. δεν μπορείτε να αφαιρέσετε αυθαίρετα Conv επίπεδα από το προκαθορισμένο δίκτυο.
 - ◆ Συνήθως χρησιμοποιούμε μικρότερο learning rate για τα ρυθμισμένα βάρη ConvNet, σε σύγκριση με τα (τυχαία αρχικοποιημένα) βάρη που θα χρησιμοποιούσαμε για το νέο γραμμικό ταξινομητή που υπολογίζει τα βάρη ταξινόμησης του νέου συνόλου δεδομένων μας.
 - Αυτό συμβαίνει επειδή περιμένουμε ότι τα ρυθμισμένα βάρη ConvNet είναι σχετικά καλά, επομένων δεν θέλουμε να τα παραμορφώσουμε πολύ γρήγορα και πάρα πολύ.

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

- ILSVRC: ετήσιος διαγωνισμός που χρησιμοποιεί υποσύνολα από το σύνολο δεδομένων ImageNet για ανάπτυξη και συγκριτική αξιολόγηση αλγορίθμων τελευταίας τεχνολογίας.
- ImageNet: πολύ μεγάλη συλλογή χαρακτηρισμένων (Amazon Mechanical Turk Worker) φωτογραφιών για την ανάπτυξη αλγορίθμων όρασης υπολογιστή.
- Οι εργασίες του ILSVRC οδήγησαν σε σημαντικές αρχιτεκτονικές μοντέλων και τεχνικές σύνδεσης της όρασης υπολογιστή και της βαθιάς μάθησης



Κατηγορίες

Image classification

Πρόβλεψη των κατηγοριών των αντικειμένων που υπάρχουν στην εικόνα

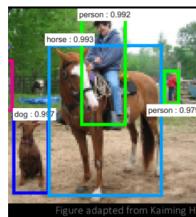


Figure adapted from Kaiming He

Single-object localization

Image classification + σχεδιασμός bounding box

Object detection

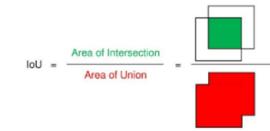
Image classification + σχεδιασμός bounding box γύρω από κάθε αντικείμενο.

Object Segmentation

Ανίχνευση όλων των αντικειμένων της εικόνας σε pixel level



Εισαγωγικές Έννοιες Ανίχνευσης Αντικειμένων



1. Μετρική της ομοιότητας μεταξύ δύο αντικειμένων

Σκοπός : σύγκριση και αξιολόγηση ομοιότητας αντικειμένων (π.χ. Ανιχνευμένο αντικείμενο με το αληθινό αντικείμενο (ground truth))

Ευρέως χρησιμοποιούμενο μέγεθος: Intersection over Union - IoU

Το IoU ορίζεται ως το εμβαδόν της τομής των δύο πλαισίων προς το εμβαδόν της ένωσής τους.

2. Πλαισιο Οριοθέτησης (Bounding Box)



- Ως πλαισιο οριοθέτησης ενός αντικειμένου σε μία εικόνα ορίζεται το μικρότερο δυνατό ορθογώνιο τμήμα της εικόνας στο εσωτερικό του οποίου βρίσκεται ολόκληρο το αντικείμενο.
- Για την περιγραφή ενός πλαισίου οριοθέτησης είναι απαραίτητες 4 τιμές. π.χ.
 - οι συντεταγμένες της κάτω αριστερής και της πάνω δεξιάς γωνίας του
 - οι συντεταγμένες της πάνω αριστερής γωνίας, το πλάτος w και το ύψος h του πλαισίου
 - οι συντεταγμένες του κέντρου του πλαισίου, το πλάτος w και το ύψος h

3. Περιοχή Ενδιαφέροντος (Region of Interest - ROI)

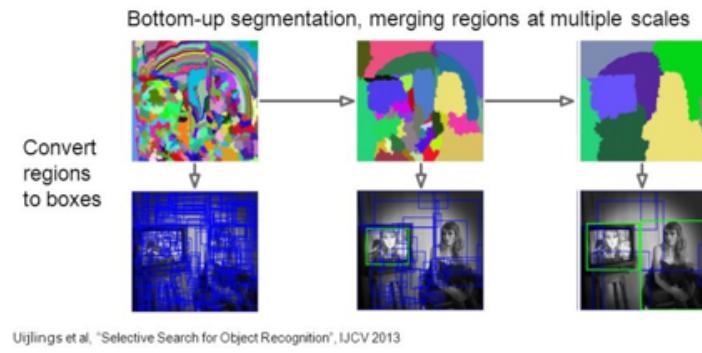
Ορίζεται μία ορθογώνια περιοχή της εικόνας εισόδου η οποία θεωρητικά είναι πιθανό να περιέχει ένα αντικείμενο.

Οι περιοχές αυτές μπορούν να υπολογιστούν:

- με χρήση κάποιου εξωτερικού αλγορίθμου όπως το Selective Search ή το Edge Box detection,
- με χρήση ενός Δικτύου Προτάσεων Περιοχών (Region Proposal Network - RPN).

Region-Based CNN: περιοχή ενδιαφέροντος (RoI)

Region Proposals: Selective Search



Object Detection: απλή προσέγγιση με CNN

1. Χωρίζουμε την εικόνα σε περιοχές και τροφοδοτούμε την κάθε περιοχή ως ξεχωριστή εικόνα στο CNN το οποίο τις ταξινομεί σε διάφορες τάξεις.

2. Αφού χωρίσουμε κάθε περιοχή στην αντίστοιχη κλάση, μπορούμε να συνδυάσουμε όλες αυτές τις περιοχές για να πάρουμε την αρχική εικόνα με τα αντικείμενα που εντοπίστηκαν.

(-) : Τα αντικείμενα μπορεί να έχουν διαφορετικά aspect ratios, χωρικές θέσεις και να έχουν υποστεί διάφορους μετασχηματισμούς

(-) : Χρειάζεται πολύ μεγάλος αριθμός περιοχών, μεγάλη υπολογιστική ισχύς

Λύση: Region-based CNN

4. Καταστολή μη μεγίστων (Non-Maximum Suppression)

Πρόβλημα: ύπαρξη πολλών προβλέψεων με μικρές διαφορές οι οποίες αντιστοιχούν στο ίδιο αντικείμενο.



Λύση: Καταστολή μη μεγίστων (Non-Maximum Suppression - NMS)

- Απληστος (greedy) αλγόριθμος που συγχωνεύει αυτά τα αλληλοεπικαλυπτόμενα πλαίσια οριοθέτησης:
 - Ταξινομεί όλα τα πλαίσια οριοθέτησης σε αύξουσα σειρά ως προς την πιθανότητά τους να αντιστοιχούν σε κάποιο αντικείμενο.
 - Επιλέγει το πλαίσιο οριοθέτησης με τη μεγαλύτερη πιθανότητα και, συγκρίνοντάς το με κάθε ένα από τα Bounding Box με μικρότερη πιθανότητα, απορρίπτει όσα έχουν επικάλυψη IoU μικρότερη από μία προκαθορισμένη τιμή.

(Η τιμή αυτή αποτελεί μία από τις υπερπαραμέτρους του συστήματος) και επαναλαμβάνει τα βήματα όσες φορές είναι απαραίτητο.

Πρόβλημα: Ανίχνευσης Αντικειμένων

Σύνθεση δύο διαφορετικών προβλημάτων:

- ένα πρόβλημα ταξινόμησης και
- ένα πρόβλημα παλινδρόμησης, γνωστό και ως bounding box regression.

Με δεδομένη μία εικόνα εισόδου, πρέπει να προβλεφθεί η τοποθεσία και η έκταση των αντικειμένων της εικόνας που ανήκουν σε ένα σύνολο προκαθορισμένων κλάσεων, και να αποδοθεί η σωστή κλάση στο κάθε αντικείμενων.

- Η τοποθεσία των αντικειμένων συνήθως εκφράζεται ως το ελάχιστο πλαίσιο οριοθέτησης που περικλείει εξ ολοκλήρου το αντικείμενο.

Κατηγορίες μοντέλων ανίχνευσης αντικειμένων

Χωρίζονται σε δύο κατηγορίες ως προς τη δομή τους:

- Τα μοντέλα ενός σταδίου (one-step models) χρησιμοποιούν :
 - ένα feed forward CNN για να προσδιορίσουν την τοποθεσία των αντικειμένων ενδιαφέροντος.
 - απλούστερα και ταχύτερα, αφού δεν παρέχουν region proposals
 - η απόδοσή τους είναι μειωμένη, κυρίως όταν απαιτείται και κατάτμηση της εικόνας

π.χ. YOLO, Multibox, AttentionNet, G-CNN

Κατηγορίες μοντέλων ανίχνευση αντικειμένων

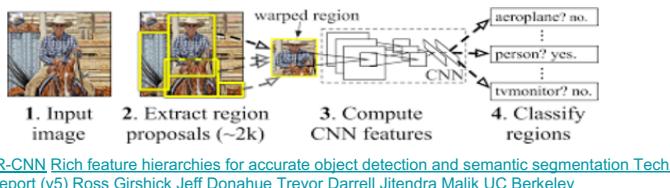
Τα μοντέλα δύο σταδίων (two-step models ή region-based models), χρησιμοποιούν

1. Έναν αλγόριθμο (π.χ. Selective Search) ή ένα μοντέλο (π.χ. region-based CNN) που δέχεται ως είσοδο την εικόνα και προτείνει διαφορετικές πιθανές περιοχές ενδιαφέροντος
2. Έναν feature extractor π.χ. CNN ώστε να υπολογιστεί ο χάρτης χαρακτηριστικών κάθε περιοχής ενδιαφέροντος ο οποίος δίνεται σε ένα πλήρως συνδεδεμένο υπεύθυνο για την ταξινόμηση.

π.χ. R-CNN, Fast R-CNN, FPN, Faster R-CNN

- έχουν αρκετές διαφορές αλλά περίπου κοινή δομή

Region-Based Convolutional Neural Network: [R-CNN](#)

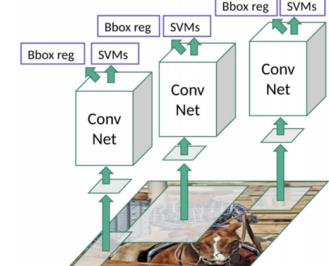


1. Χρησιμοποιεί τον αλγόριθμο Selective Search και παράγει 2000 προτάσεις περιοχών ανά εικόνα.
2. Η κάθε περιοχή, μετά από προσαρμογή του μεγέθους της, δίνεται ως είσοδος σε ένα προεκπαίδευμένο CNN
3. Η έξοδος από το ConvNet είναι ένα διάνυσμα 4096 χαρακτηριστικών
4. Εκπαιδεύουμε το τελευταίο επίπεδο του δικτύου **της κάθε περιοχής** ένα ταξινομητή με βάση τον αριθμό των κατηγοριών που πρέπει να εντοπιστούν

Region-Based Convolutional Neural Network: R-CNN

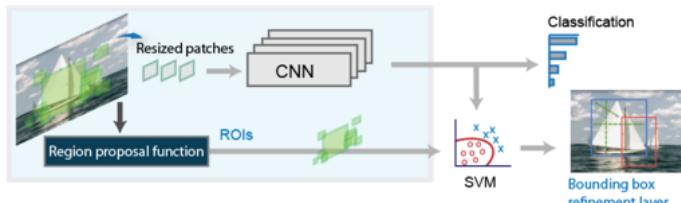
5. Αφού αποκτήσουμε τις περιοχές, εκπαιδεύουμε ένα δυαδικό SVM ανά περιοχή για να ταξινομήσουμε αντικείμενα και φόντο.

6. Τέλος, εκπαιδεύουμε ένα μοντέλο γραμμικής παλινδρόμησης για τη δημιουργία αυστηρότερων bounding boxes για κάθε αναγνωρισμένο αντικείμενο στην εικόνα.



R-CNN Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5) Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik UC Berkeley

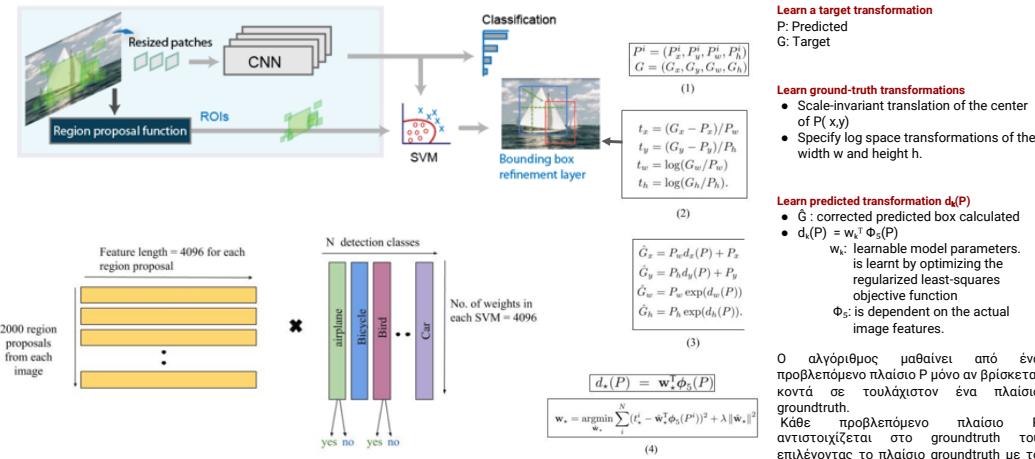
Region-Based Convolutional Neural Network: R-CNN



Μειονεκτήματα

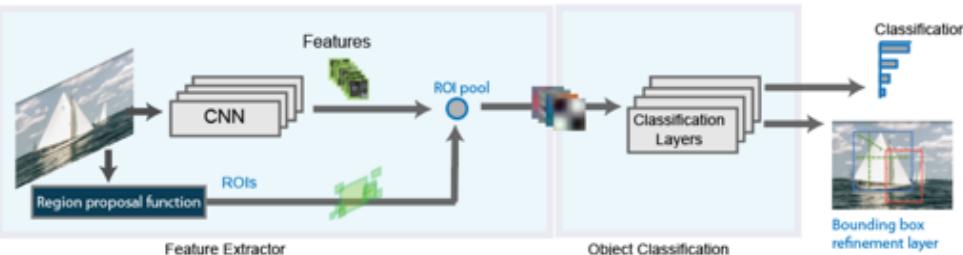
- Οι 2000 προτάσεις περιοχών ανά εικόνα → πολύ μεγάλο χρόνο εκπαίδευσης
- Ο χρόνος του testing είναι απαγορευτικά μεγάλος → μη χρήση του μοντέλου για εφαρμογές πραγματικού χρόνου.
- Προκαθορισμένη συμπεριφορά του αλγορίθμου Selective Search → η αναγνώριση δε βελτιώνεται μέσω εκπαίδευσης.

Region-Based Convolutional Neural Network: R-CNN



Ο αλγόριθμος μαθαίνει από ένα προβλεπόμενο πλαίσιο P μόνο τα βρίσκεται κοντά σε τουλάχιστον ένα πλαίσιο groundtruth. Κάθε προβλεπόμενο πλαίσιο P αντιστοιχίζεται στο groundtruth του επιλεγόντας το πλαίσιο groundtruth με το οποίο έχει μέγιστη επικάλυψη (πότε την προϋπόθεση ότι έχει επικάλυψη IoU > 0,5).

Region-Based CNN: Fast R-CNN

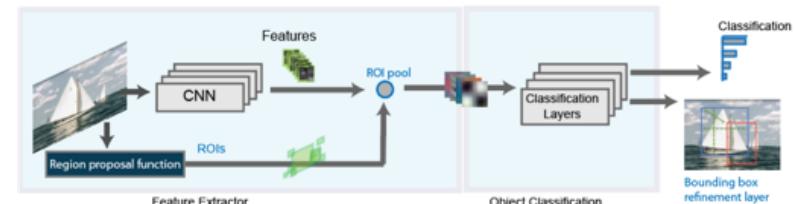


- Βελτιώνει σημαντικά την ταχύτητα του μοντέλου αλλάζοντας απλώς τη σειρά των επιπέδων του.
- Αντί να δίνεται ως είσοδος στο CNN κάθε μία από τις περιοχές ενδιαφέροντος, το CNN εξάγει τα χαρακτηριστικά ολόκληρης της εικόνας σε μορφή χάρτη χαρακτηριστικών, και στη συνέχεια για κάθε περιοχή απομονώνεται το αντίστοιχο τμήμα.

Με αυτό τον τρόπο, η εξαγωγή των χαρακτηριστικών γίνεται μόνο μία φορά αντί για 2000, γεγονός που είναι προφανές ότι βελτιώνει κατά πολύ την ταχύτητα της εκπαίδευσης και της πρόβλεψης.

Region-Based CNN: Fast R-CNN

1. Επεξεργάζεται ολόκληρη την εικόνα,
2. Ενώ ο R-CNN detector κατηγοριοποιεί κάθε περιοχή, ο Fast R-CNN συγκεντρώνει τα features maps από το CNN που αντιστοιχούν σε κάθε προτεινόμενη περιοχή (region proposal),
3. Κάθε περιοχή περνά από ένα fully connected network και ένα softmax layer δίνει τις κατηγορίες εξόδου.
4. Μαζί με το στρώμα softmax, χρησιμοποιείται επίσης ένα linear regression layer, παράλληλα για την παραγωγή των συντεταγμένων του bounding box για προβλεπόμενες κατηγορίες.



[Fast R-CNN Ross Girshick, ICCV 2015](#)

Region-Based CNN: Faster R-CNN

[Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun](#)

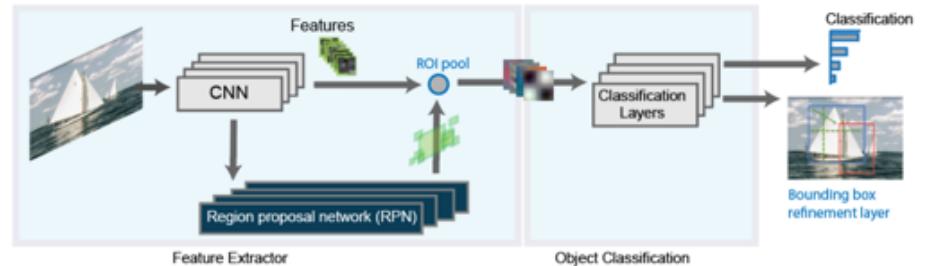
- Πρότειναν την αντικατάστασή του Selective Search, από τον δίκτυο τους αλγόριθμο:

Δίκτυο Πρότασης Περιοχών

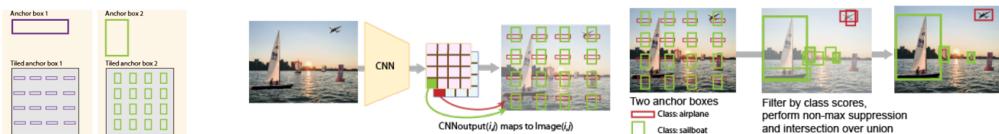
- Συγκεκριμένα, αντιλήφθηκαν ότι ο χάρτης χαρακτηριστικών που παράγεται από το συνελικτικό τμήμα του Fast R-CNN μπορεί να χρησιμοποιηθεί αποτελεσματικά και για το πρόβλημα της πρότασης περιοχών, αντικαθιστώντας τις αργές μεθόδους όπως η Selective Search με ένα εκπαιδεύσιμο νευρωνικό δίκτυο.

Region-Based CNN: Faster R-CNN

- To Faster R-CNN προσθέτει ένα region proposal network (RPN) για να δημιουργήσει region proposals απευθείας μέσω δικτύου
- To RPN χρησιμοποιεί Anchor Boxes για το Object Detection



Region-Based CNN: Faster R-CNN → Anchor Boxes



Tα anchor boxes είναι ένα σύνολο από προκαθορισμένα bounding boxes με συγκεκριμένο πλάτος και ύψος:

- ορίζονται για να καταγράφουν την κλίμακα και το λόγο διαστάσεων συγκεκριμένων κατηγοριών αντικειμένων που θέλουμε να εντοπίσουμε, (υπορούμε να έχουμε anchor boxes διαφορετικών μεγεθών)
- επιλέγονται συνήθως με βάση τα μεγέθη αντικειμένων στα training datasets,

Κατά τη διάρκεια της ανίχνευσης:

- τα predefined anchor boxes διατρέχουν την εικόνα,
- το δίκτυο προβλέπει την πιθανότητα και άλλα χαρακτηριστικά (background, IoU, offsets) για κάθε anchor box καθώς διατρέχει την εικόνα,
- επιστρέφεται ένα μοναδικό σύνολο προβλέψεων για κάθε καθορισμένο bounding box

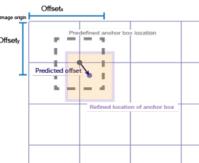
Το τελικό feature map αντιπροσωπεύει ανιχνεύσεις αντικειμένων για κάθε κατηγορία

Η χρήση anchor boxes επιτρέπει σε ένα δίκτυο να ανιχνεύει πολλά αντικείμενα, αντικείμενα διαφορετικών κλιμάκων και αλληλεπικαλυπτόμενα αντικείμενα.

[Anchor Boxes for Object Detection - MATLAB & Simulink](#)

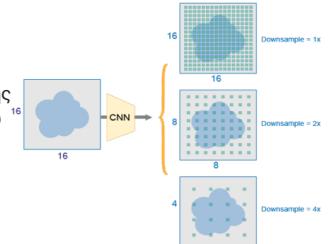
Σφάλματα εντοπισμού και βελτίωση

Σφάλματα εντοπισμού



Βελτίωση

- Η απόσταση μεταξύ των Anchor Boxes είναι συνάρτηση του ποσού της δειγματοληψίας που υπάρχει στο CNN (max Pooling 2d Layer και του stride του Conv Layer)
- Τα feature maps που παράγουν τα αρχικά επίπεδα του CNN έχουν υψηλότερη χωρική ανάλυση, αλλά μπορεί να εξαγάγουν λιγότερες σημασιολογικές πληροφορίες σε σύγκριση με τα επίπεδα που βρίσκονται πιο κάτω στο δίκτυο



Δημιουργία ανιχνευτών αντικειμένων

- Αφαιρούνται τα Tiled Anchor Boxes που ανήκουν στην κατηγορία φόντου και τα υπόλοιπα φιλτράρονται από τη βαθμολογία εμπιστοσύνης τους.
- Τα Anchor Boxes με τη μεγαλύτερη βαθμολογία εμπιστοσύνης επιλέγονται χρησιμοποιώντας non-max suppression (NMS).

Algorithm 1 Non-Max Suppression

```

1: procedure NMS( $B, c$ )
2:    $B_{nms} \leftarrow \emptyset$  Initialize empty set
3:   for  $b_i \in B$  do Iterate over all the boxes
4:      $discard \leftarrow \text{False}$  Take bi variable and set it as false. This variable indicates whether bi should be kept or discarded
5:     for  $b_j \in B$  do Start another loop to compare with bi
6:       if same( $b_i, b_j$ )  $> \lambda_{nms}$  then If both boxes having same IOU
7:         if score( $c, b_j$ )  $>$  score( $c, b_i$ ) then
8:            $discard \leftarrow \text{True}$  Compare the scores. If score of bj is less than that of bi, bj should be discarded, so set the flag to True.
9:         if not  $discard$  then
10:            $B_{nms} \leftarrow B_{nms} \cup b_i$  Once bi is compared with all other boxes and still the discarded flag is False, then bi should be considered. So add it to the final list.
11:   return  $B_{nms}$  Do the same procedure for remaining boxes and returns the final list

```

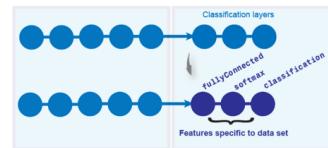


Region-Based CNN

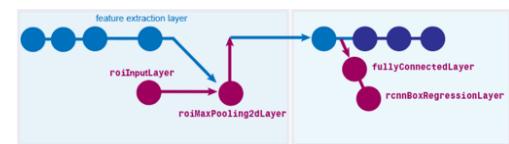
Transfer Learning model

'alexnet','vgg16','vgg19','resnet50','resnet101','inceptionv3','googlenet','inceptionresnetv2','squeezenet'

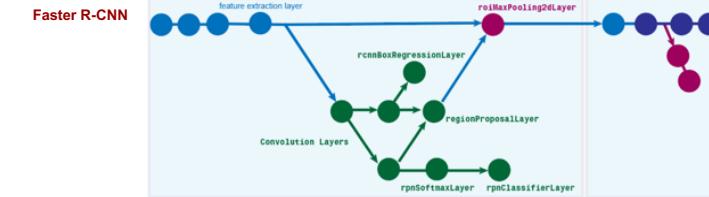
R-CNN



Fast R-CNN



Faster R-CNN



YOLO- You Only Look Once

You only look once (YOLO) at an image to predict what objects are present and where they are present.

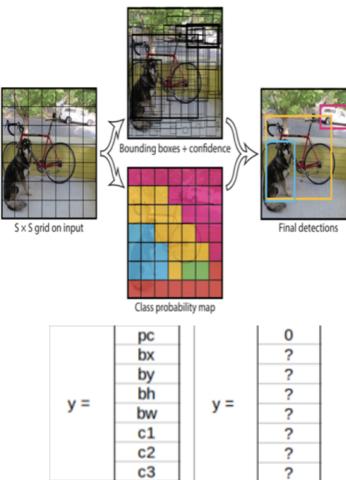


- Χρησιμοποιεί ένα απλό CNN και βλέπει ολόκληρη την εικόνα κατά τη διάρκεια του training και του validation, όποτε κωδικοποιεί σιωπηρά πληροφορίες για τις κλάσεις καθώς και τις εμφανίσεις τους, σε αντίθεση με τις τεχνικές sliding window ή region-based (κάνοντας έτσι λιγότερο από το ήμισυ του αριθμού των αιφαλμάτων σε σύγκριση με το Fast R-CNN).
- Το YOLO χρησιμοποιεί features από ολόκληρη την εικόνα για να προβλέψει κάθε bounding box
- Προβλέπει επίσης όλα τα bounding box σε όλες τις κλάσεις για μια εικόνα ταυτόχρονα με τις αντίστοιχες πιθανότητες
- Αντιμετωπίζει την ανίχνευση ως πρόβλημα παλινδρόμησης
- Εξαιρετικά γρήγορος και ακριβής αλγόριθμος

[You Only Look Once: Unified, Real-Time Object Detection](#) Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi

Λειτουργία YOLO- You Only Look Once

- Παίρνει μια εικόνα και τη χωρίζει σε πλέγμα SxS.
- Κάθε κελί πλέγματος προβλέπει μόνο ένα αντικείμενο.
- Η ταξινόμηση εικόνας και ο εντοπισμός εφαρμόζονται σε κάθε κελί του πλέγματος.
- Εάν το κέντρο ενός αντικειμένου πέσει σε ένα κελί πλέγματος, αυτό το κελί πλέγματος είναι υπεύθυνο για την ανίχνευση αυτού του αντικειμένου.
- Κάθε ένα από τα κελιά πλέγματος προβλέπει bounding boxes B με βαθμολογίες εμπιστοσύνης για αυτά τα bounding boxes
 - Οι βαθμολογίες εμπιστοσύνης αντικατοπτρίζουν το πόσο σίγουρο είναι το μοντέλο ότι το πλέγμα περιέχει ένα αντικείμενο και πόσο ακριβές πιστεύει ότι το πλαίσιο είναι αυτό που προβλέπει. Εάν δεν υπάρχουν αντικείμενα, τότε οι βαθμολογίες εμπιστοσύνης θα είναι μηδέν.
- Bounding box όταν ένα αντικείμενο υπάρχει στο κελί πλέγματος
- Πιθανότητα για class C



Region-Based CNN

[Object Detection and Tracking in 2020 | by Borjan Georgievski](#)

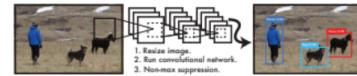
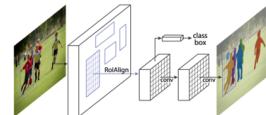
[TensorFlow Hub Object Detection Colab](#)

[Mask R-CNN](#)

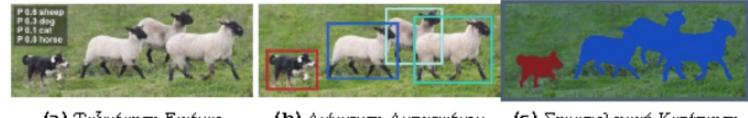
[YOLO original paper](#)

[YOLOv2 YOLO9000](#)

[DarkNet implementation](#)



Πρόβλημα Σημασιολογικής Κατάτμησης



→ Ταξινόμηση σε επίπεδο εικονοστοιχείου

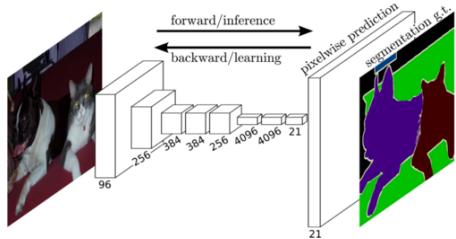
- ◆ Ανάθεση κλάσης σε κάθε εικονοστοιχείο της εικόνας
- ◆ Μέσω χωρικής ανάλυσης ενός εικονοστοιχείου.

→ Αφελής πρώτη προσέγγιση: Υλοποίηση ενός μοντέλου με διαδοχικά συνελικτικά επίπεδα, του οποίου η έξοδος θα είχε την ίδια διάσταση με την είσοδο.

- ◆ το κάθε εικονοστοιχείο της εξόδου θα αποτελούσε την πρόβλεψη για την κλάση του αντίστοιχου εικονοστοιχείου της αρχικής εικόνας.
- ◆ απαγορευτική υπολογιστική πολυπλοκότητα.

Πρόβλημα Σημασιολογικής Κατάτμησης

Συνηθισμένες προσεγγίσεις: encoder-decoder.



1. Η διάσταση της εικόνας μειώνεται αρχικά (encoder), παράγοντας χαμηλότερης ανάλυσης χάρτες χαρακτηριστικών οι οποίοι έχουν πολύ καλά αποτελέσματα για την ταξινόμηση μεταξύ των κλάσεων,
2. Στη συνέχεια αυξάνεται και πάλι (decoder), μέχρι να προκύψει ο τελικός χάρτης κατάτμησης.

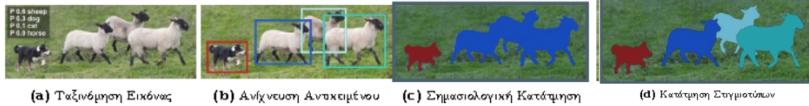
Πρόβλημα Σημασιολογικής Κατάτμησης

Πλήρως Συνελικτικό Δίκτυο (Fully Convolutional Network - FCN)

• CNN : Fully Connected Layers → Fully Convolutional Network

- **Κωδικοποιητής:** Εξαγάγει τα χαρακτηριστικά και είναι εκπαιδευμένος με βάση το πρόβλημα της ταξινόμησης.
- **Αποκωδικοποιητής:** Προβάλλει το χάρτη χαρακτηριστικών χαμηλής ανάλυσης που προέκυψε από τον κωδικοποιητή στην αρχική εικόνα.
 - Αποτελείται από μία σειρά συνελίξεις (backwards convolutions ή deconvolutions)
 - πραγματοποιούν αύξηση της χωρικής ανάλυσης με χρήση διγραμμικής παρεμβολής (bilinear interpolation).
 - Κάνει χρήση παρακαμπτήριων συνδέσεων (skip connections), που εκμεταλλεύονται τις παρόμιες διαστάσεις των εκατέρωθεν επιπέδων του FCN και συνδέουν σειριακά τους χάρτες ενεργοποίησης του κωδικοποιητή με την αντίστοιχη δομή που προκύπτει μετά από κάθε αποσυνέλιξη.

Πρόβλημα Κατάτμησης Στιγμιοτύπων (Instance Segmentation)



Κατάτμηση Στιγμιοτύπων = Ανίχνευσης Αντικειμένων + Σημασιολογική Κατάτμηση

Στοχεύει στον εντοπισμό των διαφορετικών αντικειμένων σε μία εικόνα όχι με χρήση πλαισίων οριοθέτησης αλλά με ακρίβεια εικονοστοιχείου.

→ Κάθε εικονοστοιχείο ταξινομείται σε μία κλάση, όπως στη Σημασιολογική Κατάτμηση, αλλά τα διαφορετικά αντικείμενα θα έχουν άλλη μάσκα, ακόμα κι αν ανήκουν στην ίδια κλάση.

[B. Hariharan, P. Arbelaez, R. B. Girshick, and J. Malik. Simultaneous Detection and Segmentation. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, Computer Vision – ECCV 2014, volume 8695 of Lecture Notes in Computer Science. Springer, Cham, 2014.](#)

Πρόβλημα Κατάτμησης Στιγμιοτύπων

Mask R-CNN

- Faster R-CNN με δίκτυο RPN, για την πρόταση των υποψηφίων περιοχών,
- Τμήμα για τον υπολογισμό των μασκών, αντίστοιχο με ένα Πλήρως Συνελικτικό Δίκτυο (FCN).

