Τεχνητή Νοημοσύνη

11η διάλεξη (2025-26)

Ίων Ανδρουτσόπουλος

http://www.aueb.gr/users/ion/

Οι διαφάνειες αυτής της διάλεξης βασίζονται:

- στο βιβλίο *Machine Learning* του T. Mitchell, McGraw-Hill, 1997,
- σε ύλη των διαλέξεων του μαθήματος Μηχανικής Μάθησης του A. Ng στο Πανεπιστήμιο Stanford (βλ. http://cs229.stanford.edu/).

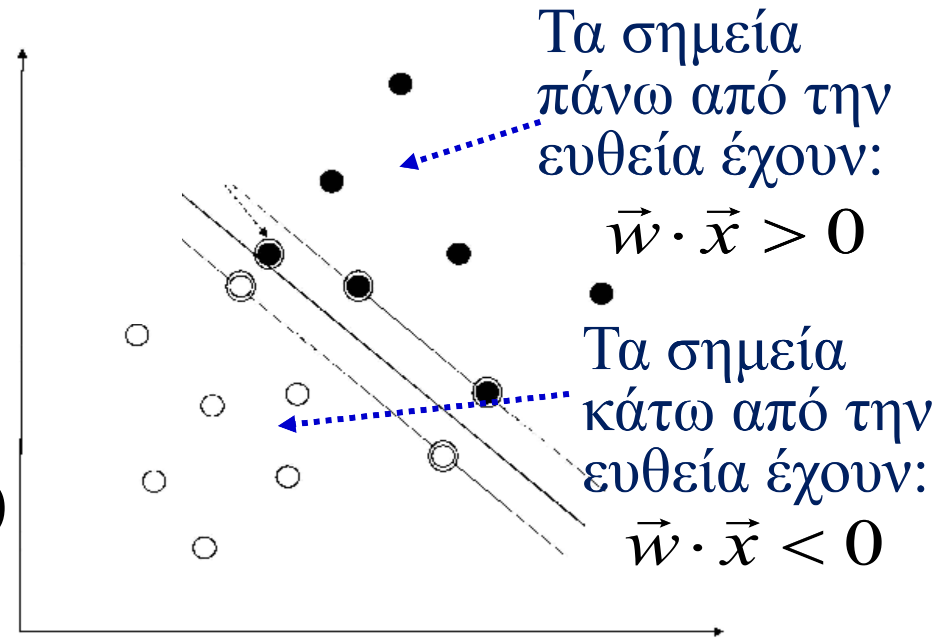
Τι θα ακούσετε σήμερα

- Γραμμικοί διαχωριστές.
- Ταξινομητές λογιστικής παλινδρόμησης.
- Μεγιστοποίηση πιθανοφάνειας με κατάβαση κλίσης.
- Ομαλοποίηση (regularization).
- Διαγνωστικοί έλεγχοι κατά τη χρήση επιβλεπόμενης μηχανικής μάθησης.

Γραμμικοί διαχωριστές

- Για δύο ιδιότητες x_1, x_2, προσπαθούμε να μάθουμε **ευθεία** που **διαχωρίζει** τις δύο κατηγορίες.

$$w_2 x_2 + w_1 x_1 + w_0 = 0$$



- Γενικότερα, για ιδιότητες x_1, x_2, …, x_n προσπαθούμε να μάθουμε ένα **υπερ-επίπεδο** που να διαχωρίζει τις δύο κατηγορίες.

Θεωρούμε πάλι ότι $x_0 = 1$.

$$w_n x_n + \ldots + w_1 x_1 + w_0 = \sum_{l=0}^{n} w_l x_l = \vec{w} \cdot \vec{x} = 0$$

- **Απόφαση** κατάταξης:

$$C = sign(\vec{w} \cdot \vec{x})$$

Γραμμικοί διαχωριστές – συνέχεια

- Συχνά θέλουμε ο ταξινομητής να επιστρέφει και ένα **βαθμό βεβαιότητας**.
  - Π.χ. πόσο **πιθανό** θεωρεί να **ανήκει** ένα προς κατάταξη κείμενο με διάνυσμα $\vec{x}$ στη **μία** ή την **άλλη κατηγορία**.
- Η προσημασμένη **απόσταση** $d_{\vec{w}}(\vec{x})$ από το υπερ-επίπεδο διαχωρισμού **<u>δεν</u>** είναι **καλός** βαθμός βεβαιότητας.

$$d_{\vec{w}}(\vec{x}) = \frac{\vec{w} \cdot \vec{x}}{\|\vec{w}\|}$$

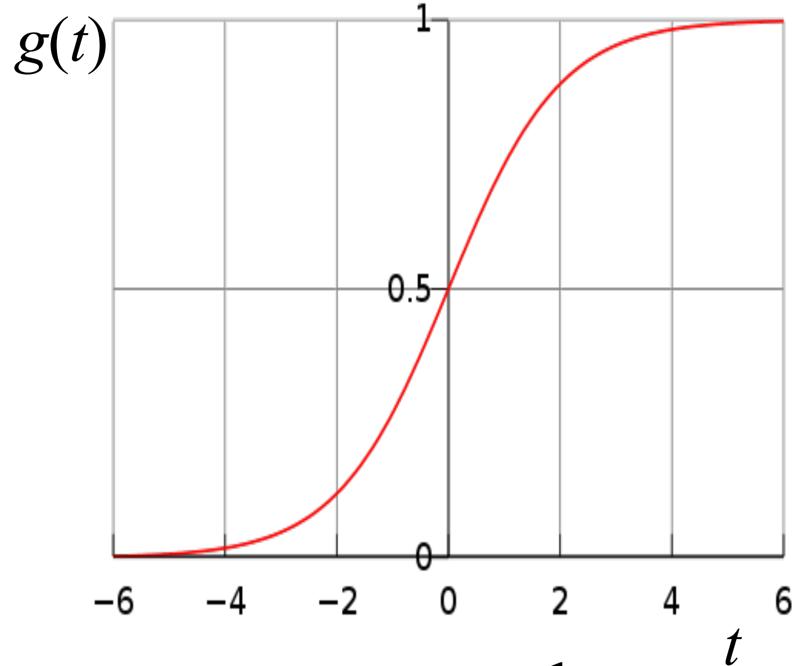
Χωρίς το w_0.

  - **Δεν** είναι **περιορισμένη** στο **[0, 1].**
  - Για **μεγάλες** (θετικές ή αρνητικές) **αποστάσεις** θέλουμε η βεβαιότητα να **τείνει στο 1**.
  - Για **μικρές αποστάσεις** θέλουμε η βεβαιότητα να **τείνει στο 0**.

Σιγμοειδής συνάρτηση (logistic function)

- Στην περίπτωσή μας το t θα είναι η **προσημασμένη** (και μη κανονικοποιημένη) **απόσταση** από το **υπερ-επίπεδο** διαχωρισμού:

$$t = \vec{w} \cdot \vec{x}$$



Πηγή εικόνας: http://en.wikipedia.org/wiki/Logistic_function

$$g(t) = \frac{1}{1+e^{-t}}$$

- **Πιθανότητα** το $\vec{x}$ να ανήκει στη **θετική** κατηγορία:

$$P(c_+ \mid \vec{x}) = \frac{1}{1+e^{-\vec{w}\cdot\vec{x}}}$$

- **Πιθανότητα** να ανήκει στην **αρνητική** κατηγορία:

$$P(c_- \mid \vec{x}) = 1 - P(c_+ \mid \vec{x})$$

Ταξινομητές λογιστικής παλινδρόμησης (logistic regression classifiers)

$$P(c_+ \mid \vec{x}) = \frac{1}{1+e^{-\vec{w}\cdot\vec{x}}}, \quad P(c_- \mid \vec{x}) = 1 - P(c_+ \mid \vec{x}) = \frac{e^{-\vec{w}\cdot\vec{x}}}{1+e^{-\vec{w}\cdot\vec{x}}}$$

- Κατά την **εκπαίδευση**, επιλέγουν το $\vec{w}$ που κάνει τον ταξινομητή πιο βέβαιο ότι τα **παραδείγματα εκπαίδευσης** ανήκουν στις **σωστές κατηγορίες**.
  - Μεγιστοποιούν τη (δεσμευμένη) «**πιθανοφάνεια**» των παραδειγμάτων.

$$L(\vec{w}) = P(y^{(1)}, \ldots, y^{(m)} \mid \vec{x}^{(1)}, \ldots, \vec{x}^{(m)}; \vec{w})$$

Οι σωστές κατηγορίες των παραδειγμάτων εκπαίδευσης.

Τα παραδείγματα εκπαίδευσης.

Μεγιστοποίηση πιθανοφάνειας

- Θεωρώντας ότι τα παραδείγματα εκπαίδευσης έχουν επιλεγεί από τον **ίδιο πληθυσμό** και είναι **ανεξάρτητα**:

$$L(\vec{w}) = P(y^{(1)}, \ldots, y^{(m)} \mid \vec{x}^{(1)}, \ldots, \vec{x}^{(m)}; \vec{w})$$

$$= \prod_{i=1}^{m} P(y^{(i)} \mid \vec{x}^{(i)}; \vec{w})$$

- Αντί να μεγιστοποιήσουμε την $L(\vec{w})$, βολεύει να μεγιστοποιήσουμε τη (γνησίως αύξουσα):

$$l(\vec{w}) = \log L(\vec{w}) = \sum_{i=1}^{m} \log P(y^{(i)} \mid \vec{x}^{(i)}; \vec{w})$$

  - Είναι **κοίλη συνάρτηση**, δεν υπάρχει κίνδυνος να φτάσω σε τοπικό μέγιστο.

Μεγιστοποίηση πιθανοφάνειας – συνέχεια

- Αν παραστήσουμε τις (σωστές) κατηγορίες με $\boldsymbol{y} = \mathbf{1}$ (**θετική** κατηγορία) και $\boldsymbol{y} = \mathbf{0}$ (**αρνητική**), τότε:

$$P(y \mid \vec{x}; \vec{w}) = P(c_+ \mid \vec{x}; \vec{w})^y \cdot P(c_- \mid \vec{x}; \vec{w})^{(1-y)}$$

  - Για $y = 1$ (θετική κατηγορία), ο 2ος όρος εξαφανίζεται.
  - Για $y = 0$ (αρνητική), ο 1ος όρος εξαφανίζεται.

- Οπότε:

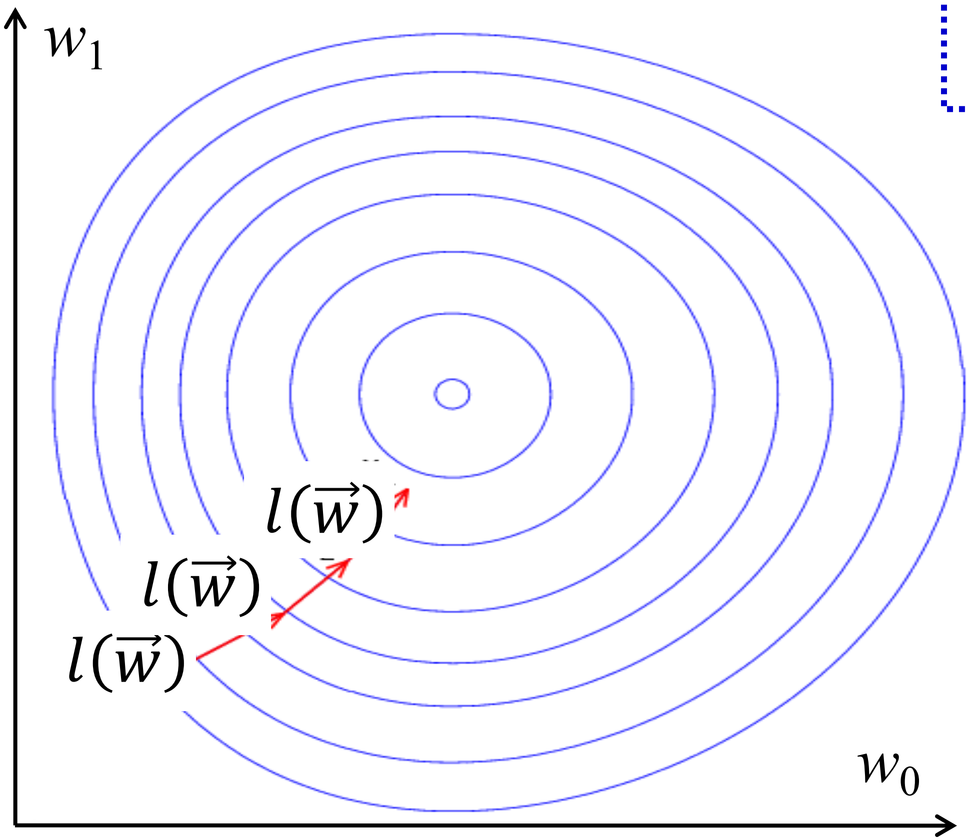
$$l(\vec{w}) = \sum_{i=1}^{m} \log P(c_+ \mid \vec{x}^{(i)}; \vec{w})^{y^{(i)}} + \log P(c_- \mid \vec{x}^{(i)}; \vec{w})^{(1-y^{(i)})}$$

$$= \sum_{i=1}^{m} y^{(i)} \log P(c_+ \mid \vec{x}^{(i)}; \vec{w}) + (1 - y^{(i)}) \log P(c_- \mid \vec{x}^{(i)}; \vec{w})$$

Ίδιο με το άθροισμα της (αρνητικής) **διασταυρωμένης εντροπίας** των παραδειγμάτων (επόμενες διαλέξεις).

Ανάβαση κλίσης (gradient ascent)

Ξεκινώ με τυχαία βάρη $\vec{w}$. Μετράω επίδοση $l(\vec{w})$ στα παραδείγματα εκπαίδευσης. Προς τα πού να μεταβάλω τα βάρη;



Η **κλίση** $\nabla l(\vec{w})$ είναι ένα διάνυσμα που δείχνει προς την κατεύθυνση μεταβολής των βαρών που οδηγεί στη μεγαλύτερη **αύξηση** του $l(\vec{w})$.

Σε κάθε βήμα, τροποποιούμε το $\vec{w}$ κατά η προς την κατεύθυνση που προκαλεί τη μεγαλύτερη μείωση του σφάλματος:

$$\vec{w} \leftarrow \vec{w} + \eta \nabla l(\vec{w})$$

Ανάβαση λόφου με συνάρτηση αξιολόγησης το $l(\vec{w})$.

Μεγιστοποίηση πιθανοφάνειας – συνέχεια

Τώρα **μεγιστοποιούμε** το $l(\vec{w})$, αντί να ελαχιστοποιούμε το $E(\vec{w})$.

- Με <u>**ανάβαση**</u> **κλίσης**:

$$\vec{w} \leftarrow \vec{w} + \eta \cdot \nabla l(\vec{w})$$

καταλήγουμε (άσκηση μελέτης) στον κανόνα ενημέρωσης:

$$w_l \leftarrow w_l + \eta \cdot \sum_{i=1}^{m} [y^{(i)} - P(c_+ \mid \vec{x}^{(i)})] \cdot x_l^{(i)}$$

- Αν έχουμε πολλά παραδείγματα εκπαίδευσης, χρησιμοποιούμε **στοχαστική ανάβαση κλίσης**.
  - Ένα μόνο παράδειγμα εκπαίδευσης $\langle x^{(i)}, y^{(i)} \rangle$ (ή ένα mini-batch) σε κάθε βήμα, κάνουμε βήμα προς το $\nabla l_i(\vec{w})$.
  - Δεν υπάρχει κλειστή λύση.

Ομαλοποίηση (regularization)

- Στην πράξη αντί για το:

$$l(\vec{w}) = \sum_{i=1}^{m} \log P(y^{(i)} \mid \vec{x}^{(i)}; \vec{w})$$

συνήθως μεγιστοποιούμε το:

$$l(\vec{w}) - \lambda \cdot \|\vec{w}\|^2 = l(\vec{w}) - \lambda \cdot \sum_{l=0}^{n} w_l^2$$

L2 regularization (“ridge regression”)

δηλ. επιβραβεύουμε υποψήφια $\vec{w}$ με πολλά μικρά βάρη.

- Υπάρχει έτσι **μικρότερος κίνδυνος υπερ-εφαρμογής**.
  - Π.χ. αν πολλά βάρη w_l είναι πολύ μικρά, οι αντίστοιχες ιδιότητες ουσιαστικά δεν χρησιμοποιούνται. Με **λιγότερες ιδιότητες** έχουμε **μικρότερο κίνδυνο υπερ-εφαρμογής**.
  - $\lambda > 0$. Η τιμή επιλέγεται με δοκιμές σε δεδομένα επικύρωσης.

Το L1 regularization (“lasso regression”) χρησιμοποιεί τη νόρμα L1, δηλ. προσθέτει $-\lambda \sum_{l=0}^{n} |w_l|$. Οδηγεί σε πιο αραιά $\vec{w}$ (με πολλά μηδενικά).

Important advice

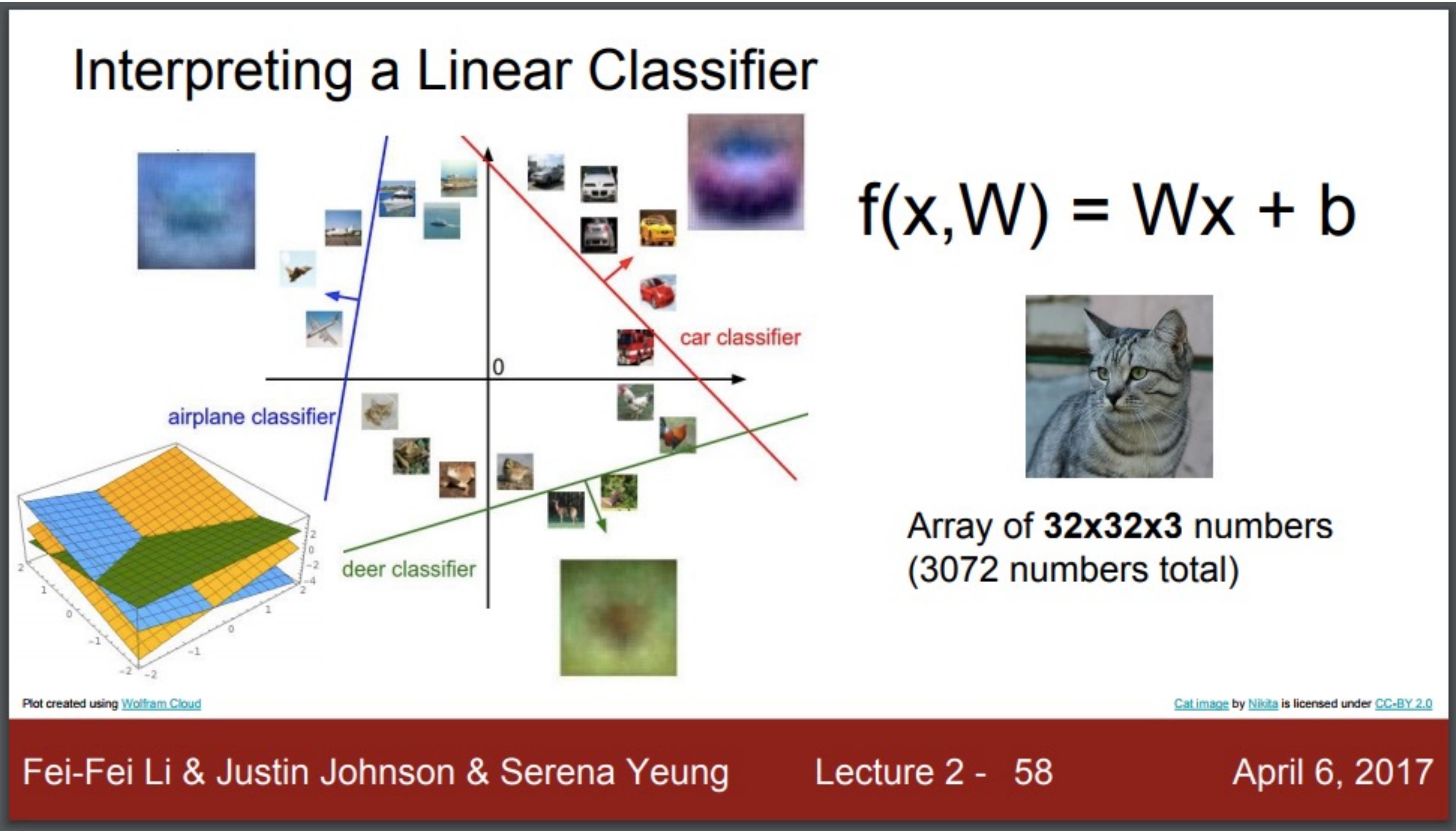
- For each attribute X_i, **normalize** assuming normal distribution:

$$X_i \leftarrow \frac{X_i - \mu_i}{\sigma_i}$$

Mean and standard deviation of X_i in the training data.

- These tricks are **also needed** in **linear regression**.
- Also important: **start** with **random small weights**.
    - E.g., sample them from a **zero-centered Gaussian with small $\boldsymbol{\sigma}$**.

Multinomial logistic regression



From Stanford's course "Convolutional Neural Networks for Visual Recognition", F.F. Li, J. Johnson, S. Yeung, 2017. http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture2.pdf

Multinomial Logistic Regression

- Extension for **multiple (non-overlapping) classes** $c_1, \ldots, c_K$.
  - Intuitively, we learn a **separate linear separator** for **each class** c_j, with its **own weights** vector $\vec{w}_j$.
  - We obtain a **score** $\boldsymbol{z_j = \vec{w}_j \cdot \vec{x}}$ from the separator of **each class** $\boldsymbol{c_j}$, and we **apply** the **softmax** function $\frac{\exp(z_j)}{\sum_{j'} \exp(z_{j'})}$ to the **scores** to turn them into **probabilities** that sum up to 1.

probability that $\vec{x}$ belongs in c_j

$$P(c_j \mid \vec{x}) = \frac{e^{\vec{w}_j \cdot \vec{x}}}{\sum_{j'=1}^{K} e^{\vec{w}_{j'} \cdot \vec{x}}}$$

different weights vector per class

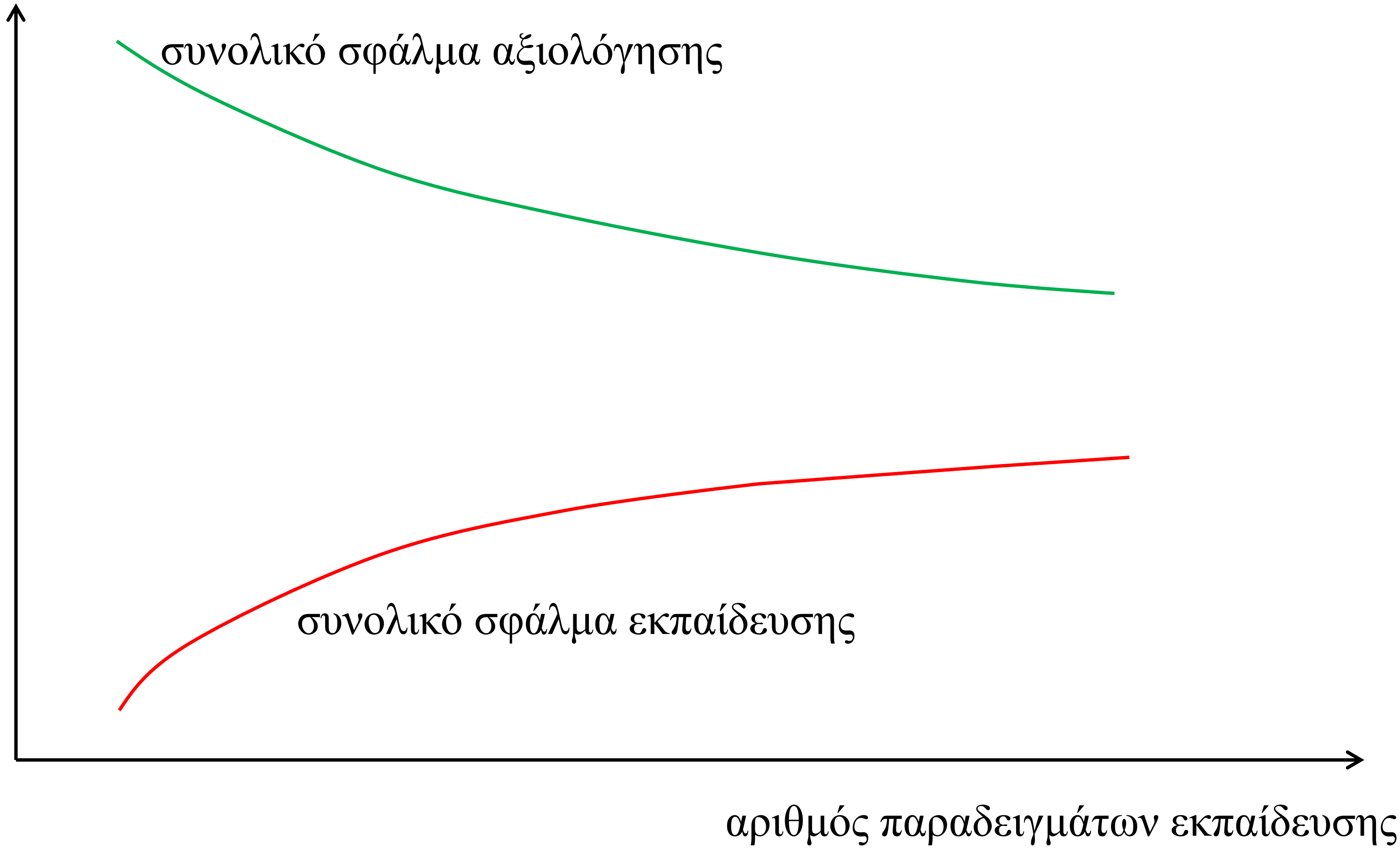
normalization factor

- We **train** by **maximizing** the conditional **log-likelihood**.
  - Same as **minimizing** the **cross-entropy** of the training examples.

Συμβουλές χρήσης επιβλεπόμενης ΜΜ
(βασισμένες σε συμβουλές του A. Ng)

- Στους περισσότερους αλγορίθμους **επιβλεπόμενης μηχανικής μάθησης** το συνολικό **σφάλμα στα δεδομένα εκπαίδευσης** είναι **χαμηλότερο** από το συνολικό σφάλμα στα δεδομένα αξιολόγησης.
  - Σφάλμα εκπαίδευσης: Πόσο καλά τα πάμε στα **ίδια** **δεδομένα** που χρησιμοποιήσαμε για εκπαίδευση.
  - Σφάλμα αξιολόγησης: Πόσο καλά τα πάμε σε **διαφορετικά** **δεδομένα** από εκείνα που χρησιμοποιήσαμε για εκπαίδευση.
- Το σφάλμα εκπαίδευσης συχνά είναι μια χρήσιμη ένδειξη του **πόσο καλά μπορούμε να ελπίζουμε ότι θα τα πάμε** κατά την αξιολόγηση.
- Παραστάσεις των δύο ειδών σφαλμάτων συχνά βοηθούν να **διαγνώσουμε** τι δεν πάει καλά με το σύστημά μας.

Διαγνωστικοί έλεγχοι: **υπερ-εφαρμογή**



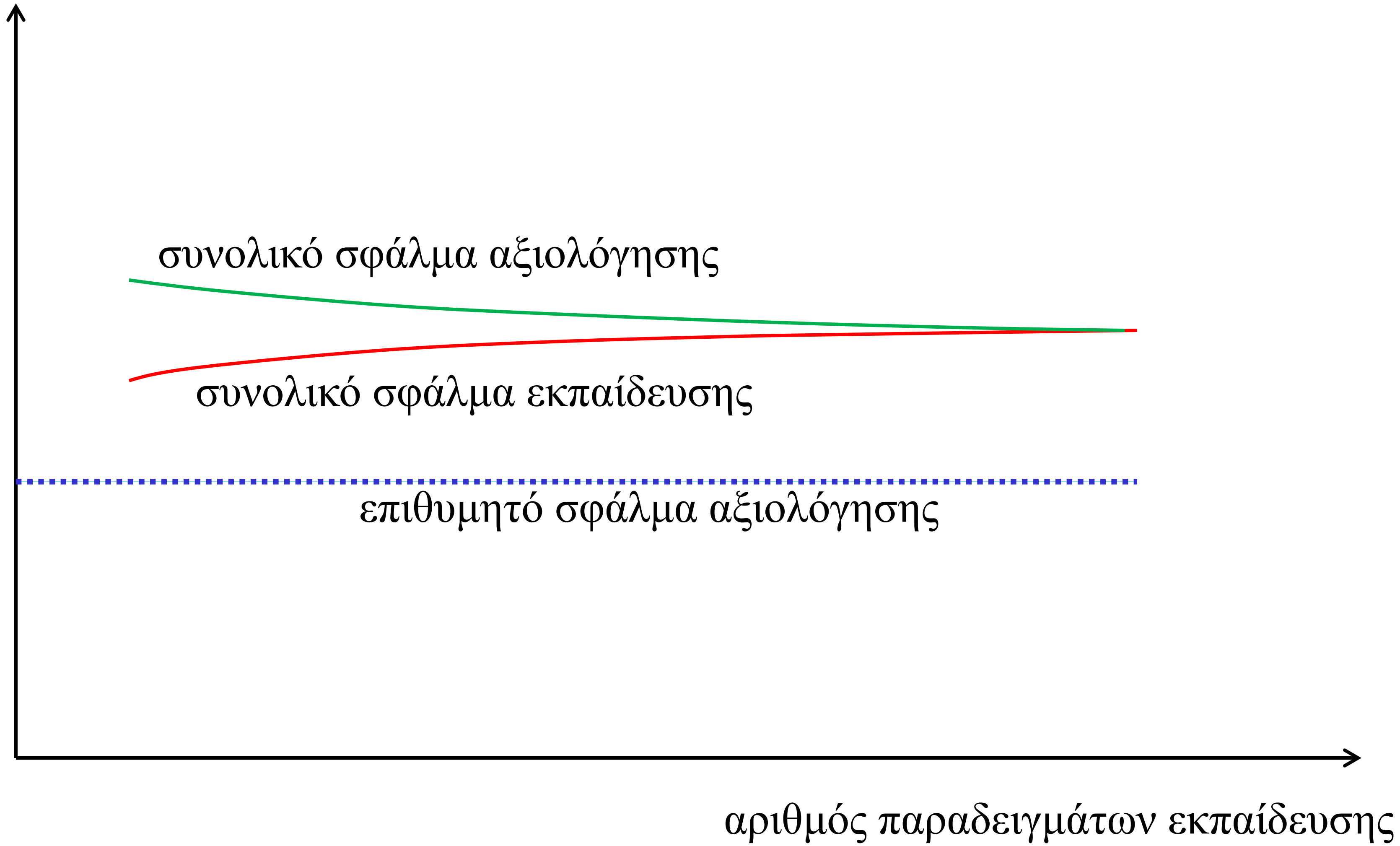
Διαγνωστικοί έλεγχοι: **υπερ-εφαρμογή**

- Αν **παρατηρούμε** τα εξής:
  - Το συνολικό σφάλμα εκπαίδευσης αυξάνεται (**χειροτερεύει**) **απότομα** όσο προσθέτουμε παραδείγματα εκπαίδευσης.
  - Το συνολικό σφάλμα αξιολόγησης μειώνεται (**βελτιώνεται**) **απότομα** όσο προσθέτουμε παραδείγματα εκπαίδευσης.
  - **Κυρίως**: υπάρχει **μεγάλη διαφορά** μεταξύ των δύο σφαλμάτων.
- Μπορεί το σύστημα να πάσχει από **υπερ-εφαρμογή**:
  - Τα πηγαίνει πολύ καλύτερα στα δεδομένα εκπαίδευσης από ό,τι στα δεδομένα αξιολόγησης, γιατί μαθαίνει **ιδιαιτερότητες** των **παραδειγμάτων αξιολόγησης.**
  - Ευκολότερο να συμβεί με **λίγα δεδομένα εκπαίδευσης**.
  - Όσο **αυξάνονται** τα **δεδομένα εκπαίδευσης**, τόσο **δυσκολότερο** γίνεται να μάθει **ιδιαιτερότητές** τους. Πετυχαίνει καλύτερη γενίκευση, οπότε τα πηγαίνει και **καλύτερα** στα δεδομένα αξιολόγησης.

Διαγνωστικοί έλεγχοι: **υπερ-εφαρμογή**

- Τι **μπορεί** να βοηθήσει:
  - **Περισσότερα δεδομένα εκπαίδευσης**.
  - **Λιγότερες** (καλύτερες) **ιδιότητες** (επιλογή ιδιοτήτων, SVD).
  - **Μεγαλύτερο** λ στη λογιστική παλινδρόμηση.
  - **Απλούστερο μοντέλο υποθέσεων** (π.χ. γραμμικές αντί για πολυωνυμικές υποθέσεις υψηλότερου βαθμού ή αντί για μη παραμετρικό μοντέλο όπως ο *k*-NN).
- Τι **δεν** θα βοηθήσει μάλλον:
  - **Περισσότερες ιδιότητες**.
  - **Πιο περίπλοκο μοντέλο** (π.χ. πιο πολλά επίπεδα ή περισσότεροι νευρώνες ανά επίπεδο σε ένα νευρωνικό δίκτυο).
  - **Περισσότερα δέντρα στα Τυχαία Δάση**.

Υπο-εφαρμογή (underfitting)



Διαγνωστικοί έλεγχοι: **υπο-εφαρμογή**

- Αν **παρατηρούμε** τα εξής:
  - Το συνολικό σφάλμα εκπαίδευσης **αυξάνεται (χειροτερεύει) πολύ λίγο** όσο προσθέτουμε παραδείγματα εκπαίδευσης.
  - Το συνολικό σφάλμα αξιολόγησης **μειώνεται (βελτιώνεται) πολύ λίγο** όσο προσθέτουμε παραδείγματα εκπαίδευσης.
  - **Κυρίως**: υπάρχει πολύ **μικρή διαφορά** μεταξύ των δύο σφαλμάτων (και δεν έχουμε φτάσει στο επιθυμητό επίπεδο σφάλματος).
- Ίσως ο **χώρος αναζήτησης** είναι υπερβολικά **περιορισμένος**:
  - Το σύστημα ίσως **δεν μπορεί να μάθει** αυτό που θέλουμε, γιατί δεν περιλαμβάνεται στο χώρο αναζήτησης.
  - Οι **υποθέσεις** του χώρου ίσως είναι **υπερβολικά απλοϊκές**, για να γενικεύσουν τα δεδομένα εκπαίδευσης.

Διαγνωστικοί έλεγχοι: **υπο-εφαρμογή**

- Τι **<u>μπορεί</u> να βοηθήσει**:
  - **Περισσότερες ιδιότητες** (π.χ. νέες πληροφορίες ή προσθήκη συνδυασμών ιδιοτήτων, όπως λογικό ΚΑΙ ζευγών ιδιοτήτων στον Naive Bayes και στη λογιστική παλινδρόμηση, που δεν μαθαίνουν μόνοι τους τέτοιους συνδυασμούς).
  - **Πιο περίπλοκο μοντέλο υποθέσεων** (π.χ. περισσότερα επίπεδα ή περισσότεροι νευρώνες ανά επίπεδο σε ένα νευρωνικό δίκτυο).
  - **Περισσότερα δέντρα** στα **Τυχαία Δάση**.
  - **Μικρότερο** λ στη λογιστική παλινδρόμηση.
- Τι **<u>δεν</u> θα βοηθήσει** μάλλον:
  - **Περισσότερα δεδομένα εκπαίδευσης**.
  - **Λιγότερες ιδιότητες** (π.χ. με επιλογή ιδιοτήτων, SVD).

Βιβλιογραφία

- Russel & Norvig (4η έκδοση): ενότητες 19.6.4, 19.6.5.
  - Όσοι ενδιαφέρονται μπορούν να διαβάσουν προαιρετικά και τις υπόλοιπες ενότητες του κεφαλαίου 19.
- Βλαχάβας κ.ά: ενότητες 18.3.3.
- Συμβουλευτείτε τις σημειώσεις «Linear regression, classification and logistic regression, generalized linear models» του A. Ng.
  - Βλ. https://sgfin.github.io/files/notes/CS229_Lecture_Notes.pdf, σελ. 1–7, 16–19.
  - Όσοι ενδιαφέρονται μπορούν να διαβάσουν προαιρετικά (εκτός εξεταστέας ύλης) και τις υπόλοιπες ενότητες. Το Perceptron της ενότητας 6 θα το συναντήσουμε και στην επόμενη διάλεξη.

Βιβλιογραφία – συνέχεια

- Οι ταξινομητές λογιστικής παλινδρόμησης περιγράφονται και σε πρόσθετο (ηλεκτρονικό, δωρεάν) κεφάλαιο του βιβλίου «Machine Learning» του T. Mitchell.
  - Βλ. http://www.cs.cmu.edu/~tom/NewChapters.html.
  - Βλ. εισαγωγή ενότητας 3 και υπο-ενότητες 3.2 και 3.3.

- Οι Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines, SVM) είναι άλλη μια σημαντική μέθοδος επιβλεπόμενης μάθησης.
  - Περιγράφονται στην ενότητα 19.7.5 των Russel & Norvig ($4^{η}$ έκδοση) και στην ενότητα 18.9 των Βλαχάβα κ.ά.
  - Περιγράφονται επίσης στο κεφάλαιο 15 του βιβλίου «An Introduction to Information Retrieval» των C.D. Manning, P. Raghavan και H. Schütze, το οποίο διατίθεται ελεύθερα (βλ. http://www-nlp.stanford.edu/IR-book/).