ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Τεχνητή Νοημοσύνη

17η διάλεξη (2024-25)

Ίων Ανδρουτσόπουλος

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

Οι διαφάνειες αυτής της διάλεξης βασίζονται:

- στο βιβλίο *Artificial Intelligence – A Modern Approach* των S. Russel και P. Norvig, 2η και 4η έκδοση, Prentice Hall, 2003 και 2020,
- 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

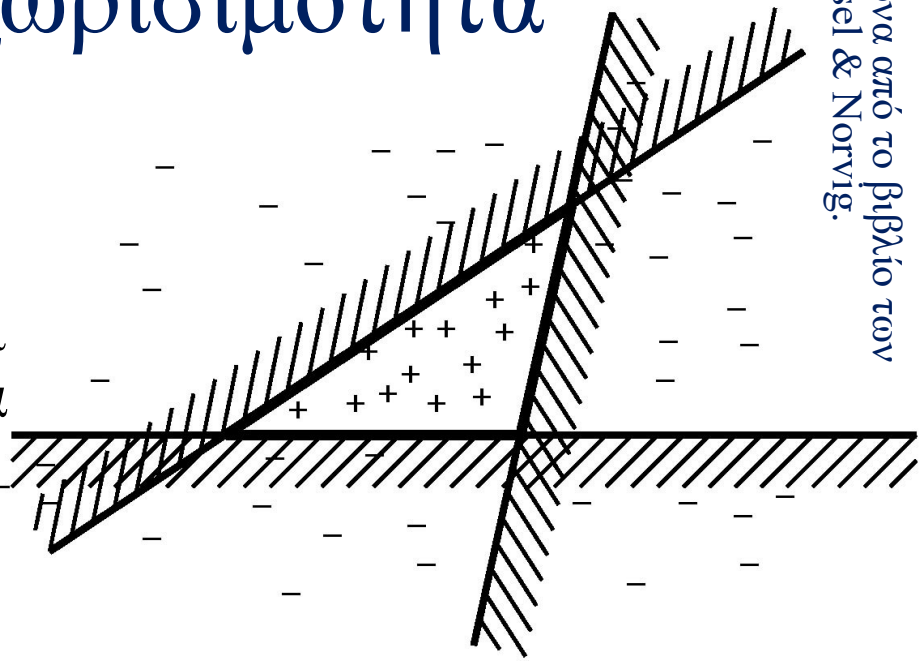
- Περισσότερα περί συλλογικής μάθησης (ensemble learning).
- Ενδυνάμωση (boosting) και αλγόριθμος ADABOOST.
- Γραμμική παλινδρόμηση.
- Κατάβαση κλίσης (gradient descent).

Συλλογική μάθηση (ensemble learning)

- Μαθαίνουμε πολλές **διαφορετικές υποθέσεις**.
  - Π.χ. με διαφορετικά **δεδομένα εκπαίδευσης**
  - ή με διαφορετικά **σύνολα ιδιοτήτων**
  - ή με διαφορετικές **τιμές υπερ-παραμέτρων**
  - ή με διαφορετικούς **αλγόριθμους μάθησης**.
- **Συνδυάζουμε** τις αποκρίσεις τους.
  - Π.χ. ακολουθούμε την άποψη της πλειοψηφίας.
  - **Αν τα λάθη κάθε υπόθεσης είναι ανεξάρτητα** από τα λάθη των άλλων και **κάθε υπόθεση κάνει λάθος με πιθανότητα p = 0.1**, τότε π.χ. με **συνδυασμό 3 υποθέσεων, ο πλειοψηφικός συνδυασμός κάνει λάθος με p'** $= p^3 + 3 \cdot p^2 \cdot (1 - p)$ (πρέπει να κάνουν λάθος και οι τρεις ή δύο από τις τρεις) = **0.028 < 0.1**.
  - Αν οι υποθέσεις κάνουν συχνά τα **ίδια λάθη**, **τότε δεν έχουμε ανεξαρτησία**. Αν τα λάθη, όμως, που κάνουν διαφέρουν σε κάποιο βαθμό και πάλι συχνά έχουμε βελτίωση.

Γραμμική διαχωρισιμότητα

Έστω ότι έχουμε δύο κατηγορίες (+ και -) και δύο ιδιότητες (x, y). Το πρόβλημα μάθησης λέγεται **γραμμικά διαχωρίσιμο** ανν υπάρχει **ευθεία** που χωρίζει τα παραδείγματα των δύο κατηγοριών. Για περισσότερες ιδιότητες, ανν υπάρχει **υπερ-επίπεδο** που τα διαχωρίζει.



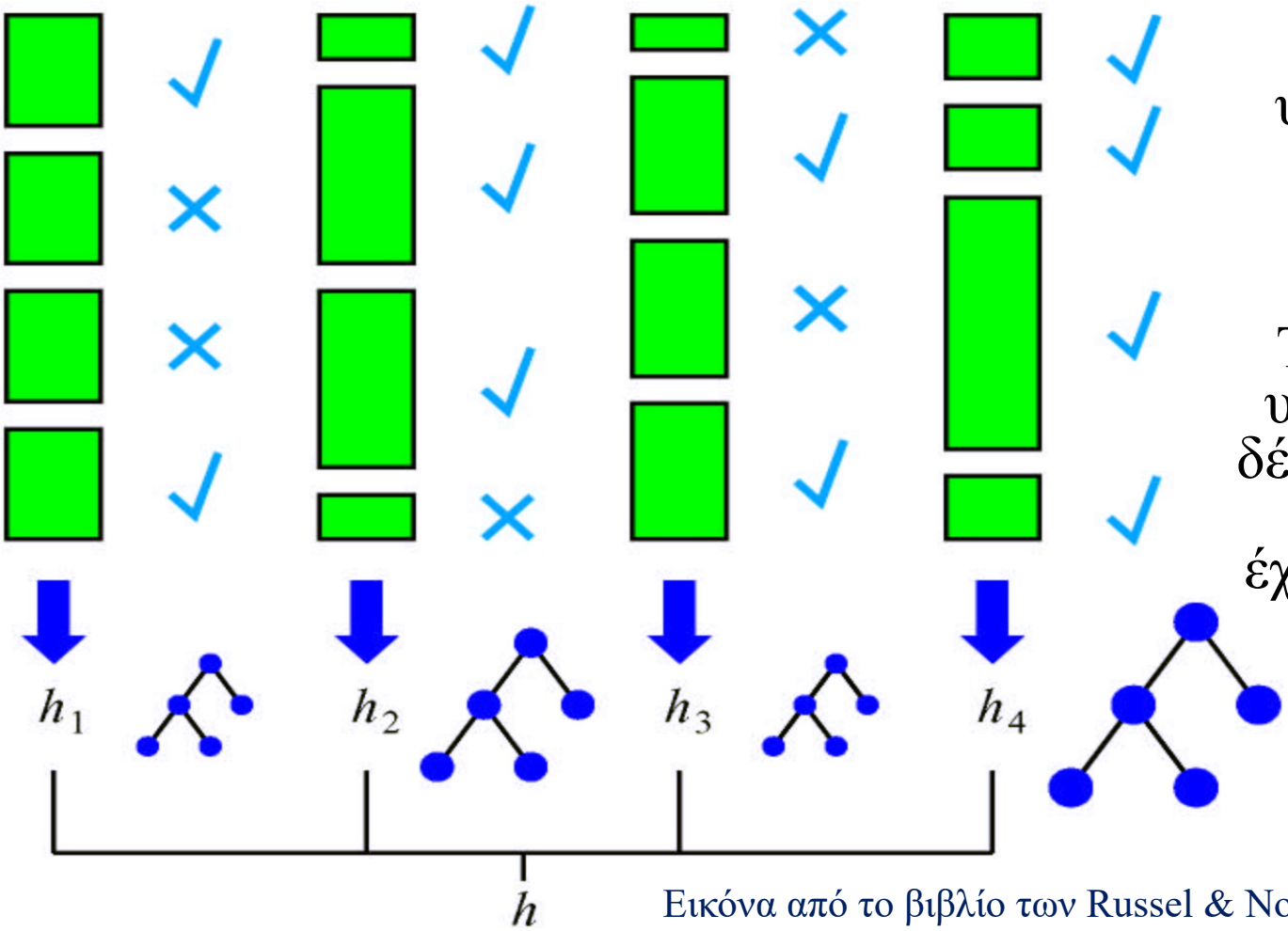
Εικόνα από το βιβλίο των Russel & Norvig.

- Υπάρχουν αλγόριθμοι μάθησης που ο χώρος αναζήτησής τους περιέχει μόνο **γραμμικούς διαχωριστές**.
  - Μαθαίνουν συναρτήσεις της μορφής: $f(\vec{x}) = \sum_i w_i x_i + b = \vec{w} \cdot \vec{x} + b$
  - Κατατάσσουν ανάλογα με το πρόσημο του $f(\vec{x})$.
- Με γραμμικούς διαχωριστές δεν μπορούμε να διαχωρίσουμε τα παραδείγματα του σχήματος, αλλά με **συνδυασμό** τους μπορούμε.

Ενδυνάμωση (boosting)

Βάρη των παραδειγμάτων εκπαίδευσης στην 1η επανάληψη.

Στην επόμενη επανάληψη αυξάνει το βάρος των παραδειγμάτων που κατετάγησαν πριν λανθασμένα.



Δεν είναι απαραίτητο κάποια από τις υποθέσεις να είναι απολύτως συνεπής.

Το μέγεθος των υποθέσεων (εδώ δέντρων) αναλογεί στο βάρος που έχουν στην τελική ψηφοφορία.

Εικόνα από το βιβλίο των Russel & Norvig.

Ενδυνάμωση (boosting)

- Κάθε παράδειγμα εκπαίδευσης φέρει ένα **βάρος** $w_j \geq 0$.
  - Παριστάνει τη **σπουδαιότητα** που έχει το παράδειγμα.
  - Οι περισσότεροι αλγόριθμοι **επεκτείνονται** εύκολα, ώστε να υποστηρίζουν παραδείγματα εκπαίδευσης με βάρη.
  - Ή αντιμετωπίζουμε κάθε παράδειγμα σαν να εμφανίζεται $\mathbf{w_j}$ **φορές** στα δεδομένα εκπαίδευσης.
- **Εκπαίδευση**: μαθαίνουμε **διαδοχικά** Μ υποθέσεις:
  - Την πρώτη φορά, όλα τα παραδείγματα έχουν ίσο βάρος.
  - Συνήθως πριν κάθε επανάληψη, τα **βάρη** των παραδειγμάτων που κατέταξε **λάθος** η τελευταία υπόθεση **αυξάνονται** και αυτών που κατέταξε **σωστά μειώνονται**.
  - Δίνουμε έμφαση στα παραδείγματα που κατετάγησαν λάθος.
- **Κατά τη χρήση**: ψηφοφορία μεταξύ των Μ υποθέσεων.
  - Οι υποθέσεις που ήταν συνεπείς με περισσότερα παραδείγματα εκπαίδευσης έχουν **ψήφους** με μεγαλύτερα **βάρη**.

Αλγόριθμος ADABOOST

Ένας από τους πιο γνωστούς αλγορίθμους ενδυνάμωσης.

συνάρτηση adaboost(*παραδείγματα*, *L*, *M*)

είσοδοι:

παραδείγματα: σύνολο $\{(x_1, y_1), ..., (x_N, y_N)\}$ παραδειγμάτων x_i με τις ορθές αποκρίσεις τους y_i.

L: βασικός αλγόριθμος μάθησης

M: αριθμός υποθέσεων που θα δημιουργηθούν

τοπικές μεταβλητές:

w: διάνυσμα με τα βάρη των *N* παραδειγμάτων, αρχικά όλα *1/N*

h : διάνυσμα όπου εισάγουμε τις *M* υποθέσεις που μαθαίνουμε

z : διάνυσμα όπου εισάγουμε τα βάρη ψήφων των *M* υποθέσεων

... συνεχίζεται ...

Αλγόριθμος ADABOOST (συνέχεια)

για $m = 1$ **ως** M

  $h[m] \leftarrow L(παραδείγματα, w)$

  $σφάλμα \leftarrow 0$

  για $j = 1$ **ως** N

    αν $h[m](x_j) \neq y_j$ **τότε** $σφάλμα \leftarrow σφάλμα + w[j]$

  αν $σφάλμα \geq 0.5$ **τότε** m-- και βγες από τον βρόχο

  για $j = 1$ **ως** N

    αν $h[m](x_j) = y_j$ **τότε** $w[j] \leftarrow w[j] \cdot σφάλμα / (1 - σφάλμα)$

  $w \leftarrow normalize(w)$

  $z[m] \leftarrow ½ \log[(1 - σφάλμα) / σφάλμα]$

επίστρεψε weighted-majority(h, m, z)

Μαθαίνουμε μια νέα υπόθεση (ένα νέο μέλος της επιτροπής).

Υπολογίζουμε το συνολικό σφάλμα της νέας υπόθεσης. Το άθροισμα των βαρών w των παραδειγμάτων είναι 1.

Για σφάλμα < 0.5, τα βάρη των παραδειγμάτων που κατετάγησαν σωστά μειώνονται.

Ώστε τα παραδείγματα να έχουν πάλι άθροισμα βαρών 1.

Δίνουμε μεγάλο βάρος σε μια υπόθεση αν τα πήγε καλά.

Επιστρέφει την άποψη της πλειοψηφίας των m υποθέσεων του h, με βάρη ψήφων εκείνα του z.

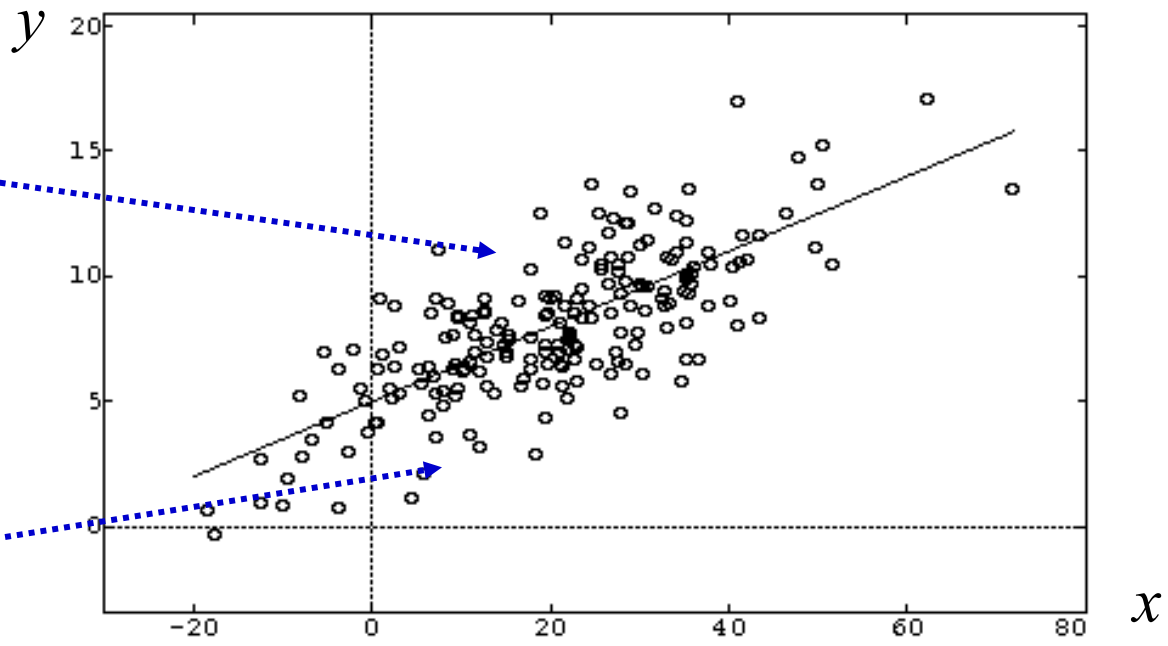
Χαρακτηριστικά ADABOOST

- Αν ο **βασικός αλγόριθμος** L επιτυγχάνει πάντα ***σφάλμα* < 0.5**, η συνολική επιστρεφόμενη υπόθεση γίνεται **συνεπής** με τα δεδομένα εκπαίδευσης για **μεγάλα M**.
  - Αν δεν υπάρχουν ασυνεπή παραδείγματα. (Η απόδειξη παραλείπεται.)
  - Άρα μπορούμε να μάθουμε **οποιαδήποτε συνάρτηση** για μεγάλα M (π.χ. μη γραμμικούς διαχωριστές).
  - Συχνά χρησιμοποιούνται απλοϊκοί βασικοί αλγόριθμοι, π.χ. **δέντρα απόφασης ενός μόνο επιπέδου** (decision stumps).
- **Καθώς το M αυξάνει**, η συνολική υπόθεση κάποια στιγμή γίνεται **συνεπής** με τα δεδομένα εκπαίδευσης (αν δεν έχουμε ασυνεπή παραδείγματα εκπαίδευσης).
  - Αλλά αν **αυξήσουμε το M περαιτέρω** (πιο περίπλοκη υπόθεση), συνήθως επιτυγχάνουμε ακόμα υψηλότερα ποσοστά ορθότητας στα δεδομένα αξιολόγησης (**καλύτερη γενίκευση**).
  - **Αντίθετα** από ό,τι προτείνει το ξυράφι του **Ockham**!
  - Πιθανή εξήγηση: με μεγαλύτερα M, η συνολική υπόθεση μαθαίνει να διαχωρίζει τις κατηγορίες με **μεγαλύτερο περιθώριο**, που τη βοηθά να κατατάσσει ακριβέστερα νέες περιπτώσεις.

Γραμμική παλινδρόμηση

Τα σημεία πάνω από τη γραμμή της $f(x)$ έχουν:
$y > w_1 x + w_0$

Τα σημεία κάτω από τη γραμμή της $f(x)$ έχουν:
$y < w_1 x + w_0$



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

- Θέλουμε να **μάθουμε την $f(x)$** από ένα **δείγμα** (τελείες).
- Περιοριζόμαστε σε **γραμμικές υποθέσεις** (συναρτήσεις):

$$y = f_{w_1, w_0}(x) = w_1 x + w_0$$

- Άρα ψάχνουμε τα **καλύτερα**: w_1, w_0

Γραμμική παλινδρόμηση – συνέχεια

- Αν έχουμε **δύο ιδιότητες** x_1, x_2, οι γραμμικές μας υποθέσεις αντιστοιχούν σε **επίπεδα** του τριδιάστατου χώρου:

$$y = f_{w_2, w_1, w_0}(x_1, x_2) = w_2 x_2 + w_1 x_1 + w_0$$

- Γενικότερα, αν έχουμε ιδιότητες $x_1, x_2, \ldots, x_n$, οι γραμμικές μας υποθέσεις αντιστοιχούν σε **υπερ-επίπεδα** του $(n+1)$-διάστατου χώρου:

$$y = f_{w_n, \ldots, w_0}(x_1, \ldots, x_n) = w_n x_n + \ldots w_1 x_1 + w_0$$

$$= \sum_{l=0}^{n} w_l x_l = \langle w_0, w_1, \ldots, w_n \rangle \cdot \langle x_0, x_1, \ldots x_n \rangle$$

$$f_{\vec{w}}(\vec{x}) = \vec{w} \cdot \vec{x} = W^T X$$

Θεωρούμε ότι πάντα $x_0 = 1$.

Αν θεωρήσουμε κάθε διάνυσμα ως πίνακα μιας στήλης.

και ψάχνουμε το **καλύτερο** $\vec{w}$.

Συνάρτηση αξιολόγησης

- Ο **χώρος αναζήτησης** περιλαμβάνει τα δυνατά $\vec{w}$.
- Για να **αξιολογήσουμε** κάθε **κατάσταση** $\vec{w}$, θα χρησιμοποιήσουμε το **τετραγωνικό σφάλμα**:

$$E(\vec{w}) = \frac{1}{2}\sum_{i=1}^{m}[f_{\vec{w}}(\vec{x}^{(i)}) - y^{(i)}]^2$$

όπου:

$(\vec{x}^{(i)}, y^{(i)})$ τα **παραδείγματα εκπαίδευσης** (δείγμα),

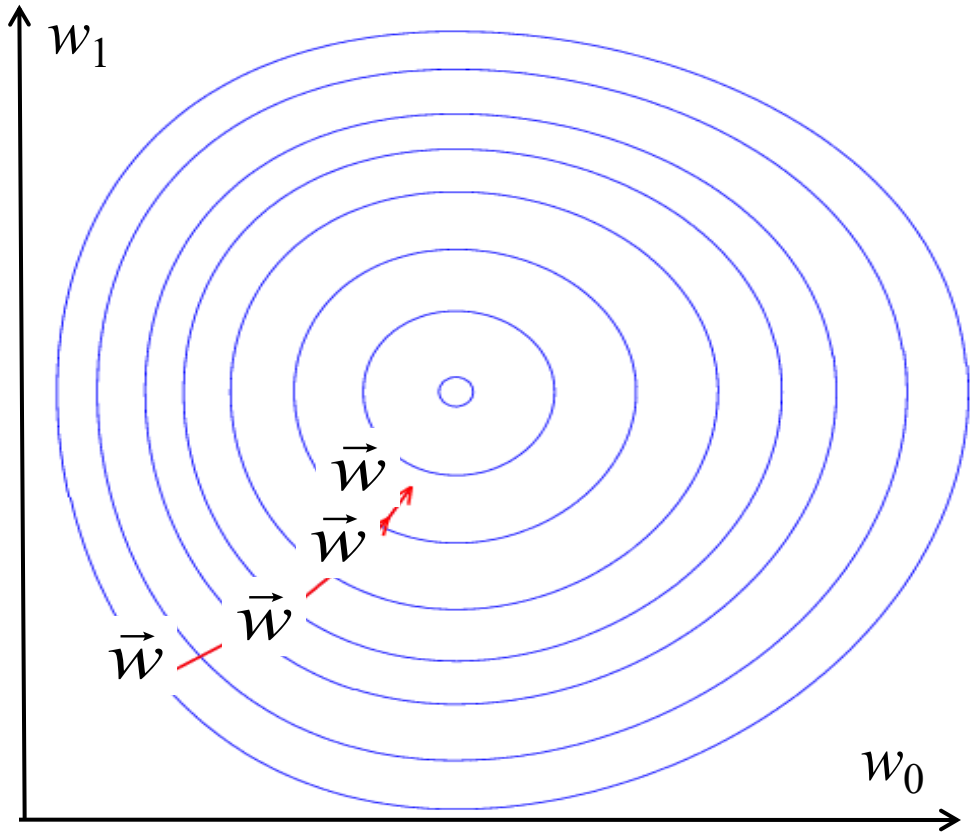
$y^{(i)}$ η **ορθή απόκριση** για είσοδο $\vec{x}^{(i)}$.

- «Γραμμική παλινδρόμηση **ελαχίστων τετραγώνων**».
  - Αξιολογούμε αθροίζοντας τα **τετράγωνα** των **διαφορών** των **αποκρίσεων** από τις **επιθυμητές** τιμές.

Παράσταση της E για διαφορετικά $\vec{w}$

$$E(\vec{w}) = \frac{1}{2}\sum_{i=1}^{m}[f_{\vec{w}}(\vec{x}^{(i)}) - y^{(i)}]^2$$

Η επιφάνεια που προκύπτει είναι **κυρτή**, άρα δεν έχει τοπικά ελάχιστα.



Ψάχνουμε το διάνυσμα βαρών $\vec{w}$ για το οποίο το συνολικό σφάλμα

$$E(\vec{w})$$

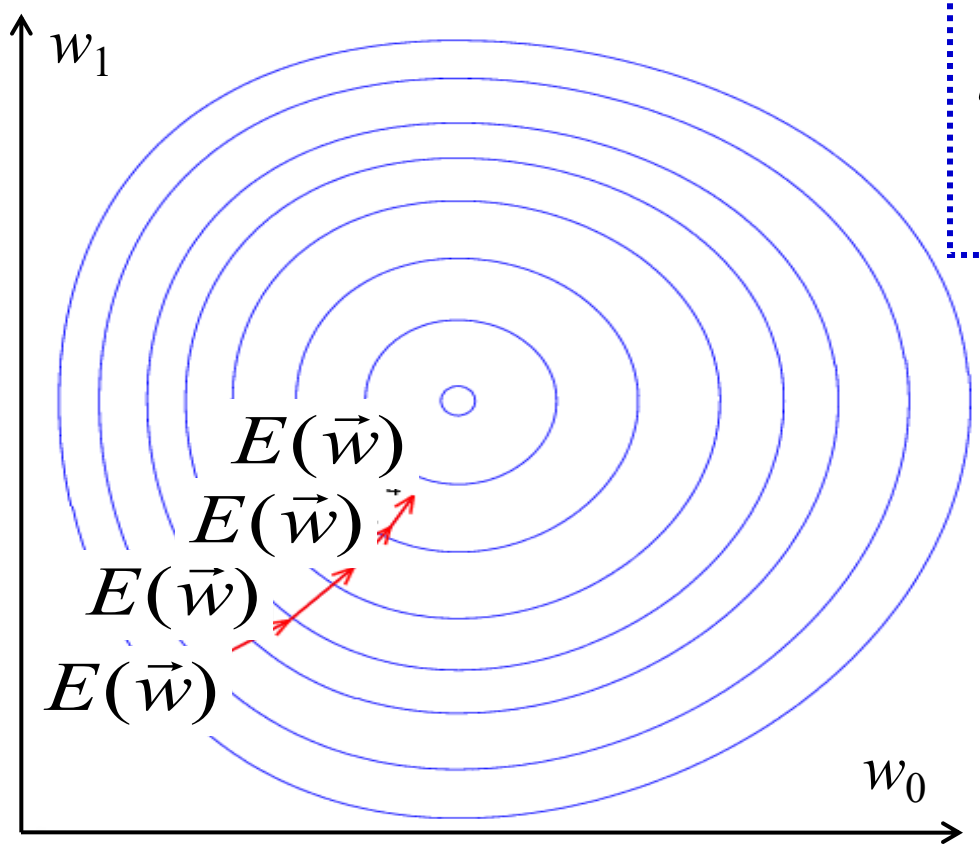
στα παραδείγματα εκπαίδευσης είναι **ελάχιστο**.

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

Κατάβαση κλίσης (gradient descent)

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

$$E(\vec{w}) = \frac{1}{2}\sum_{i=1}^{m}[f_{\vec{w}}(\vec{x}^{(i)}) - y^{(i)}]^2$$



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

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

$$\vec{w} \leftarrow \vec{w} - \eta \cdot \nabla E(\vec{w})$$

Κατάβαση λόφου με συνάρτηση αξιολόγησης E.

Κανόνας ενημέρωσης βαρών

$$E(\vec{w}) = \frac{1}{2}\sum_{i=1}^{m}[f_{\vec{w}}(\vec{x}^{(i)}) - y^{(i)}]^2$$

$$\nabla E(\vec{w}) = \left\langle \frac{\partial E(\vec{w})}{\partial w_0}, \frac{\partial E(\vec{w})}{\partial w_1}, \ldots, \frac{\partial E(\vec{w})}{\partial w_l}, \ldots, \frac{\partial E(\vec{w})}{\partial w_n} \right\rangle$$

$$\frac{\partial E(\vec{w})}{\partial w_l} = \sum_{i=1}^{m}[f_{\vec{w}}(\vec{x}^{(i)}) - y^{(i)}] \cdot \frac{\partial (f_{\vec{w}}(\vec{x}^{(i)}) - y^{(i)})}{\partial w_l}$$

$$= \sum_{i=1}^{m}[f_{\vec{w}}(\vec{x}^{(i)}) - y^{(i)}] \cdot \frac{\partial (\sum_{j=0}^{n} w_j x_j^{(i)} - y^{(i)})}{\partial w_l}$$

Κανόνας ενημέρωσης βαρών – συνέχεια

$$E(\vec{w}) = \frac{1}{2}\sum_{i=1}^{m}[f_{\vec{w}}(\vec{x}^{(i)}) - y^{(i)}]^2$$

$$\nabla E(\vec{w}) = \left\langle \frac{\partial E(\vec{w})}{\partial w_0}, \frac{\partial E(\vec{w})}{\partial w_1}, \ldots, \frac{\partial E(\vec{w})}{\partial w_l}, \ldots, \frac{\partial E(\vec{w})}{\partial w_n} \right\rangle$$

$$\frac{\partial E(\vec{w})}{\partial w_l} = \ldots = \sum_{i=1}^{m}[f_{\vec{w}}(x^{(i)}) - y^{(i)}] \cdot x_l^{(i)}$$

Άρα:

$$\nabla E(\vec{w}) = \sum_{i=1}^{m}[f_{\vec{w}}(\vec{x}^{(i)}) - y^{(i)}] \left\langle x_0^{(i)}, \ldots x_l^{(i)}, \ldots, x_n^{(i)} \right\rangle$$

$$= \sum_{i=1}^{m}[f_{\vec{w}}(\vec{x}^{(i)}) - y^{(i)}] \cdot \vec{x}^{(i)}$$

Κανόνας ενημέρωσης βαρών – συνέχεια

Άρα ο κανόνας ενημέρωσης βαρών:

$$\vec{w} \leftarrow \vec{w} - \eta \cdot \nabla E(\vec{w})$$

γίνεται:

$$\vec{w} \leftarrow \vec{w} - \eta \cdot \sum_{i=1}^{m} [f_{\vec{w}}(\vec{x}^{(i)}) - y^{(i)}] \cdot \vec{x}^{(i)}$$

και κάθε βάρος ενημερώνεται ως εξής:

$$w_l \leftarrow w_l - \eta \cdot \sum_{i=1}^{m} [f_{\vec{w}}(\vec{x}^{(i)}) - y^{(i)}] \cdot x_l{}^{(i)}$$

Αλγόριθμος κατάβασης κλίσης
(batch gradient descent)

1. Ξεκίνα με τυχαία βάρη $\vec{w}$.
2. Όσο το $E(\vec{w})$ δεν έχει συγκλίνει:
3. Ενημέρωσε τα βάρη:

$$w_l \leftarrow w_l - \eta \cdot \sum_{i=1}^{m} [f_{\vec{w}}(\vec{x}^{(i)}) - y^{(i)}] \cdot x_l{}^{(i)}$$

4. Πήγαινε στο βήμα 2.

Το η είναι μικρή θετική σταθερά. Εναλλακτικά προσαρμόζεται σε κάθε επανάληψη.

Στοχαστική κατάβαση κλίσης

1. Ξεκίνα με τυχαία βάρη $\vec{w}$.
2. Θέσε $i \leftarrow 1$ και $s \leftarrow 0$. Ανακάτεψε τα παραδείγματα.
3. Υπολόγισε το $E_i(\vec{w}) = \frac{1}{2}[f_{\vec{w}}(\vec{x}^{(i)}) - y^{(i)}]^2$
   μόνο στο <u>τρέχον</u> (i-στό) παράδειγμα εκπαίδευσης.
4. $s \leftarrow s + E_i(\vec{w})$
5. Ενημέρωσε τα βάρη: $\vec{w} \leftarrow \vec{w} - \eta \cdot \nabla E_i(\vec{w})$
   δηλαδή: $w_l \leftarrow w_l - \eta \cdot [f_{\vec{w}}(\vec{x}^{(i)}) - y^{(i)}] \cdot x_l^{(i)}$
6. Αν υπάρχει (i+1)-στό παράδειγμα, θέσε $i \leftarrow i + 1$ και πήγαινε στο βήμα 3.
7. Αν το s δεν έχει συγκλίνει και δεν υπερβήκαμε το μέγιστο αριθμό επαναλήψεων, πήγαινε στο βήμα 2.

Προκύπτει υπολογίζοντας τις μερικές παραγώγους...

Στοχαστική κατάβαση κλίσης – συνέχεια

- **Μικρότερο υπολογιστικό κόστος**.
  - Το **σφάλμα** υπολογίζεται κάθε φορά σε **ένα μόνο παράδειγμα** εκπαίδευσης. Στην πράξη (ιδιαίτερα με GPUs) σε ένα **mini-batch** (π.χ. μερικές δεκάδες) παραδειγμάτων εκπαίδευσης.
- Τα βήματα **δεν πηγαίνουν** πάντα **προς το ελάχιστο** του $E(\vec{w})$. Κάθε βήμα πάει προς το ελάχιστο του $E_i(\vec{w})$.
  - **Φαίνεται** σαν να κάνει και **τυχαία βήματα**.
  - Με **μεγαλύτερα mini-batches**, η κλίση του $E_i(\vec{w})$ **προσεγγίζει περισσότερο** την **κλίση του** $\boldsymbol{E(\vec{w})}$, επιτρέπει **μεγαλύτερο** $\boldsymbol{\eta}$.
- Ενδέχεται να **μη φτάσει ακριβώς** στο ελάχιστο $E(\vec{w})$, αλλά να **περιφέρεται** γύρω (και κοντά) από αυτό.
  - Αλλά **στην πράξη** φτάνει συνήθως **πολύ κοντά** στο ελάχιστο.
  - Δείτε και https://aclanthology.org/2024.eacl-long.157/.

Κλειστή λύση γραμμικής παλινδρόμησης

- Υπάρχει και **κλειστή λύση** για την εύρεση των βαρών που ελαχιστοποιούν το συνολικό σφάλμα.

$$E(\vec{w}) = \frac{1}{2}\sum_{i=1}^{m}[f_{\vec{w}}(\vec{x}^{(i)}) - y^{(i)}]^2$$

- Θέτουμε $\nabla E(\vec{w}) = \vec{0}$ και λύνουμε ως προς $\vec{w}$.
- Η λύση που προκύπτει είναι: $W^* = (X^T X)^{-1} X^T Y$

όπου:

$$X = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_n^{(2)} \\ \dots & \dots & \dots & \dots \\ x_1^{(m)} & x_2^{(m)} & \dots & x_n^{(m)} \end{bmatrix} \quad Y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \dots \\ y^{(m)} \end{bmatrix}$$

Κλειστή λύση – συνέχεια

- Η κλειστή λύση όμως απαιτεί την **αντιστροφή** του **πίνακα** $(X^T X)$.
  - Αν έχουμε πολύ μεγάλο αριθμό παραδειγμάτων και ιδιοτήτων, μπορεί να είναι **χρονοβόρα**.
- Η **κατάβαση κλίσης** μπορεί να χρησιμοποιηθεί **και σε άλλα προβλήματα**, όπου **δεν υπάρχει κλειστή λύση**.
  - Θα συναντήσουμε τέτοια προβλήματα στη συνέχεια.

Βιβλιογραφία

- Russel & Norvig (4η έκδοση): ενότητα 19.6 ως και υπο-ενότητα 19.6.2, εισαγωγή ενότητας 19.8, υπο-ενότητα 19.8.4.
- Βλαχάβας κ.ά: ενότητες 18.3 και 18.11.1, 18.11.2.
  - Προαιρετικά μελετήστε και την ενότητα 18.11.3.
- Συμβουλευτείτε και τις σημειώσεις «Linear regression, classification and logistic regression, generalized linear models» του A. Ng του Πανεπιστημίου Stanford.
  - Βλ. https://sgfin.github.io/files/notes/CS229_Lecture_Notes.pdf, σελ. 1–7.