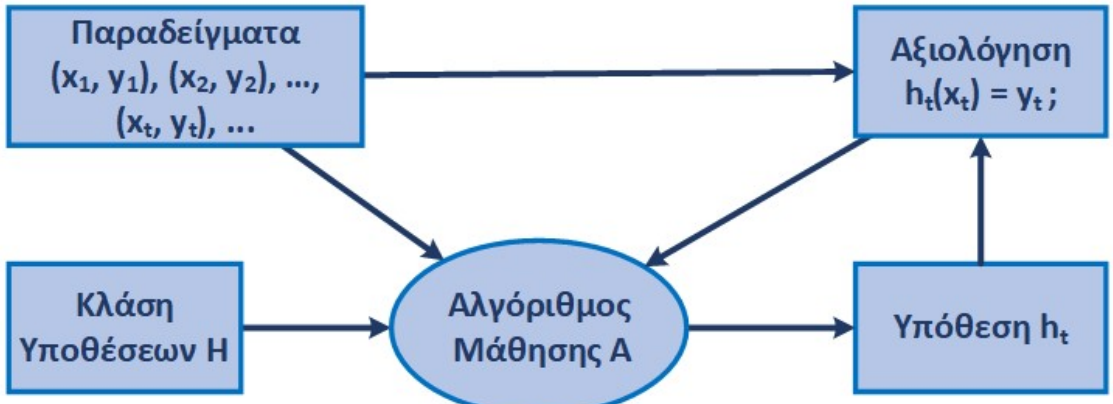
Βασικές Έννοιες

- **Σύμπαν** (domain) **X**: σύνολο (παρα)δειγμάτων για κατηγοριοποίηση, όπως περιγράφονται με βάση χαρακτηριστικά.
  - Έστω όλα τα μήλα: X = Βάρος x Όγκος x Περίμετρος x Χρώμα
- **Κατηγορίες** (labels) **Y** (εστιάζουμε σε |Y| = 2, π.χ. Y = {-1, +1} ή Y = {0, 1})
  - Για μήλα: Y = { Άνοστο, Νόστιμο }
- **Υπόθεση** (hypothesis, concept, classifier) **h : X → Y**
  - Κατηγοριοποιεί μήλο («παράδειγμα») ως άνοστο ή νόστιμο με βάση χαρακτηριστικά.
  - Για |Y| = 2, υπόθεση h ⊆ X (σύνολο νόστιμων μήλων).
- **Στόχος**: δεδομένων ορθά κατηγοριοποιημένων παραδειγμάτων, υπολογισμός υπόθεσης h : X → Y που κατηγοριοποιεί ορθά **όλα(;)** τα παραδείγματα στο X.
  - Δοκιμάζοντας **λίγα** μήλα, μαθαίνουμε να αποφεύγουμε **όλα** τα άνοστα!
  - Στατιστική (λίγα παραδείγματα) και υπολογιστική (ταχύτητα) **αποδοτικότητα**.

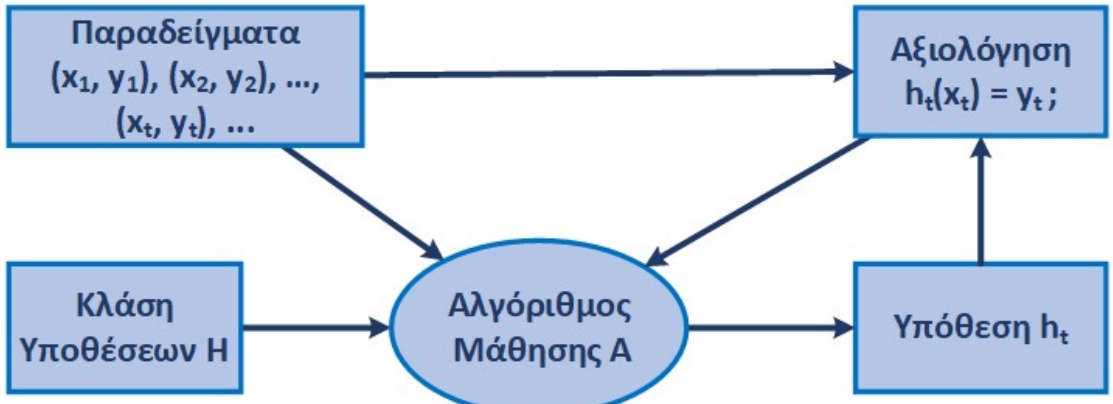
Άμεση Μάθηση (Online Learning)

- Κάθε χρονική στιγμή t = 1, 2, … (επ' άπειρον):
  - Επιλέγουμε υπόθεση $h_t : X \to Y$
  - Εμφανίζεται παράδειγμα $x_t \in X$
  - Τοποθετούμε παράδειγμα x_t στην **κατηγορία $z_t = h_t(x_t)$**
  - Πληροφορούμαστε (ορθή) **κατηγορία y_t** παραδείγματος x_t
  - Αν **$z_t \neq y_t$** , έχουμε λάθος (κόστος 1), διαφορετικά σωστό (κόστος 0)
- **Στόχος**: **πεπερασμένο** κόστος (για **άπειρη** ακολουθία παραδειγμάτων)!
  - «Μικρό» σύμπαν αποτελεί **τετριμμένη** περίπτωση (απομνημόνευση).
  - Χωρίς γνωστή «δομή», μπορεί **μη εφικτό** για «μεγάλο» (ή **άπειρο**) σύμπαν.



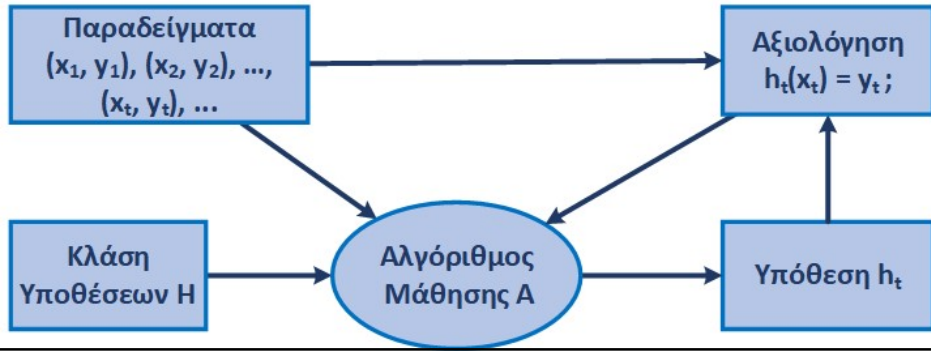
Άμεση Μάθηση (Online Learning)

- Κάθε χρονική στιγμή t = 1, 2, … (επ' άπειρον):
  - Επιλέγουμε υπόθεση $h_t : X \to Y$ και εμφανίζεται παράδειγμα $(x_t, y_t) \in (X \times Y)$
  - Αν $\mathbf{h_t(x_t) \neq y_t}$, έχουμε λάθος (κόστος 1), διαφορετικά σωστό (κόστος 0)
- **Στόχος**: **πεπερασμένο** κόστος (για **άπειρη** ακολουθία παραδειγμάτων)!
  - Χωρίς γνωστή «δομή», μπορεί **μη εφικτό** για «μεγάλο» (ή **άπειρο**) σύμπαν.
- **Κλάση υποθέσεων** (hypothesis class) $H \subseteq 2^X$ (ή γενικά $H \subseteq Y^X$) και υπάρχει απόλυτα **συνεπής** υπόθεση $\mathbf{f \in H}$ ώστε $\mathbf{y_t = f(x_t)}$, για κάθε $x_t \in X$.
  - «Ορθή» κατηγοριοποίηση f για X: **πραγματοποιήσιμη** (realizable) περίπτωση.
  - Απαιτούνται **υποθέσεις** για κλάση H ή/και για μέγεθος περιγραφής f.



Άμεση Μάθηση με Φράγμα Λαθών

- **Κλάση υποθέσεων H** μαθαίνεται με **M λάθη**, αν υπάρχει αλγόριθμος με ≤ M λάθη για κάθε ακολουθία παραδειγμάτων κατηγοριοποιημένων από κάποια $f \in H$.
  - Επιθυμητή πολυωνυμική (π.χ., στο $\log_2|H|$) υπολογιστική πολυπλοκότητα!
- **Παράδειγμα**: εκμάθηση λογικών διαζεύξεων σε ≤ n μεταβλητές.
  - Σύμπαν $X = \{0, 1\}^n$, κλάση $\mathbf{H_{disj\text{-}n}}$ περιέχει όλες λογικές διαζεύξεις ≤ n μεταβλητών. Π.χ., $h_{\{1,3,7\}}(x) = x_1 \vee x_3 \vee x_7$. $h_S(x) = \bigvee_{i \in S} x_i\,,\ \forall S \subseteq \{1, \ldots, n\}$
  - Αρχικά $S = \{ 1, \ldots, n \}$ και υπόθεση h_S. Αμετάβλητη: $h_S(x) \geq f(x)$.
  - Για κάθε **λάθος** $h_S(x) = 1$ και $f(x) = 0$, αφαιρούμε από S κάθε θέση i όπου $x_i = 1$.
  - Σε κάθε λάθος, |S| μικραίνει τουλάχιστον κατά 1: συνολικά ≤ n λάθη.
  - Κλάση λογικών διαζεύξεων $H_{disj\text{-}n}$ μαθαίνεται **με n λάθη** (βέλτιστο στη χειρ. περ.).
- **Συντηρητικός** αλγόριθμος: ενημερώνει κατάσταση του **μόνο όταν** κάνει λάθος.
- **Perceptron** μαθαίνει υπερεπίπεδο διαχωρισμού με $O(1/\gamma^2)$ λάθη (γ περιθώριο).



Αλγόριθμος Υποδιπλασιασμού (Halving Algorithm)

- Υποδιπλασιασμός για **πεπερασμένη** κλάση υποθέσεων H:
  - Αρχικά $\mathbf{S_0 = H}$. Διατηρούμε σύνολο S_t **συνεπών** υποθέσεων μέχρι δείγμα t.
  - Για t = 1, 2, …, κλάση $\mathbf{z_t}$ δείγματος $\mathbf{x_t}$ με **πλειοψηφία** σε **συνεπείς** υποθέσεις $\mathbf{h \in S_{t-1}}$
  - $\mathbf{S_t = \{ h \in S_{t-1} \mid h(x_t) = y_t \}}$ (υποθέσεις συνεπείς για t πρώτα παραδείγματα).
  - Συνολικά $\mathbf{\leq \log_2(|H|)}$ λάθη: αν έχουμε λάθος, $|S_t| \leq |S_{t-1}| / 2$, λόγω πλειοψηφίας.
  - Αν έχουμε prior p_h = πιθανότητα υπόθεση h να είναι απολύτως ορθή.
  - Πλειοψηφία με βάση πιθανότητες p_h. Λάθη ≤ **εντροπία** αντίστοιχης κατανομής.
- Χρονική πολυπλοκότητα O(|H|) – **εκθετικός** σε μέγεθος περιγραφής h ∈ H!
  - Αποδοτικές υλοποιήσεις: π.χ., αλγόριθμος **ελλειψοειδούς** μαθαίνει **υπερεπίπεδο** διαχωρισμού σε n-πλέγμα διάστασης d με $\mathbf{O(d^2 \log(n))}$ λάθη, αντί O(d log(n)) λαθών.
- Αν κλάση H **δεν** περιέχει απολύτως ορθή υπόθεση για X:
  αγνωστική (agnostic) περίπτωση.

Επιλέγοντας Συμβουλή Ειδικού

- Έχουμε |H| ειδικούς, έναν για κάθε $h \in H$, που προβλέπουν αν βρέξει ή όχι.
- Κάθε πρωί t = 1, 2, …., T, βλέπουμε χαρακτηριστικά καιρού x_t και επιλέγουμε h_t
- Αν $h_t(x_t) = y_t$, έχουμε κόστος 0, αλλιώς κόστος 1.
- **Στόχος**: κόστος συγκρίσιμο με κόστος καλύτερου ειδικού (εκ των υστέρων).

$$\text{regret}(T) = \sum_{t=1}^{T} (h_t(x_t) \neq y_t) - \min_{h \in H} \sum_{t=1}^{T} (h(x_t) \neq y_t)$$

  - Αν τέλειος ειδικός, πλειοψηφία αλάνθαστων μέχρι στιγμής εγγυάται **$\leq \log_2|H|$** λάθη
  - Αν **όχι**, φάσεις με πλειοψηφία αλάνθαστων με επανεκκίνηση όταν όλοι ≥ 1 λάθος.
  - Αν L = #λαθών καλύτερου ειδικού, έχουμε **$\leq L+1$** φάσεις, και **$\leq \log_2|H|$** λάθη/φάση.
  - Συνολικός #λαθών **$\leq \log_2|H| (L + 1)$**.
  - «Ξεχνάμε» εντελώς προηγούμενες φάσεις: περιθώριο σημαντικής βελτίωσης!

Πλειοψηφία με Βάρη Εμπιστοσύνης (WMA)

- Λάθος δεν αποκλείει κάποιον ειδικό h, αλλά μειώνει βάρος εμπιστοσύνης w(h).
  - Αρχικά $\mathbf{w_1(h) = 1}$ για κάθε $h \in H$. Σε κάθε βήμα t = 1, 2, …, T:
  - Για παρατήρηση x_t, υιοθετούμε πρόταση $\mathbf{z_t}$ που συγκεντρώνει βεβαρημένη πλειοψηφία.

$$\sum_{h \in H : h(x_t) = z_t} w_t(h) \geq W_t(H)/2$$

  - Για κάθε $h \in H$ με $h(x_t) \neq y_t$, $\mathbf{w_{t+1}(h) = w_t(h)/2}$ (γενικότερα $w_{t+1}(h) = (1 - \varepsilon) w_t(h)$).
- L = #λαθών καλύτερου ειδικού, M = #λαθών αλγόριθμου, W = συνολικό βάρος.
  - Αρχικά $W_1 = |H|$, μειώνεται κατά 25% σε κάθε λάθος: $W_{t+1} \leq 3\, W_t / 4$
  - Βάρος καλύτερου ειδικού ≤ συνολικό βάρος

$$(1/2)^L \leq |H|(3/4)^M$$
$$-L \leq \log(|H|) - M \log(4/3)$$
$$M \leq 2.41(L + \log(|H|))$$

  - #λαθών $\mathbf{\leq 2.41(L + \log_2|H|)}$, αντί του $\mathbf{\log_2|H|\ (L + 1)}$ για πλειοψηφία με φάσεις.
  - $\mathbf{2.5\log_2|H|}$ λάθη αρχικά, και μετά **2.5** λάθη αλγόριθμου για κάθε **αναπόφευκτο** λάθος!
  - Όχι ικανοποιητικό για «δύσκολες» προβλέψεις με «μεγάλο» L. Μπορούμε καλύτερα;

Αναλογικότητα με Βάρη Εμπιστοσύνης (RWMA)

- Αντίστοιχα, αλλά χρησιμοποιούμε τα **βάρη** ως **πιθανότητες**!
  - Αρχικά $\mathbf{w_1(h) = 1}$ για κάθε $h \in H$. Σε κάθε βήμα $t = 1, 2, \ldots, T$:
  - Για παρατήρηση x_t, υιοθετούμε πρόταση $\mathbf{z_t}$ με πιθανότητα ανάλογη βάρους υποστήριξης. $\frac{\sum_{h \in H: h(x_t) = z_t} w_t(h)}{W_t(H)}$
  - Για κάθε $h \in H$ με $h(x_t) \neq y_t$, $\mathbf{w_{t+1}(h) = (1 - \varepsilon)w_t(h)}$.
- L = #λαθών καλύτερου ειδικού, M = **αναμενόμενο** #λαθών αλγόριθμου, F_t = πιθανότητα λάθους σε βήμα t, και $M = F_1 + F_2 + \ldots + F_t + \ldots$
  - $W_{t+1} = W_t$[σωστό] + $(1 - \varepsilon)W_t$[λάθος]
  - Αφαιρούμε εF_t από συνολικό βάρος W_t σε κάθε βήμα: $\mathbf{W_{t+1} = W_t(1 - \varepsilon F_t)}$
  - #λαθών $\mathbf{\leq (1+\varepsilon/2)L + \log_2|H|/\varepsilon}$, $\forall \varepsilon > 0$.
  - **Regret** = #λαθών – βέλτιστος #λαθών
    Regret / L $\rightarrow$ 0, καθώς L (ή T) μεγαλώνει.
  - **No-regret** αλγόριθμοι: πρακτικά **βέλτιστος** μέσος #λαθών, καθώς T μεγαλώνει!

$$(1-\varepsilon)^L \leq |H| \prod_t (1 - \varepsilon F_t)$$

$$-L\ln(1-\varepsilon) \leq \log(|H|) - \varepsilon \sum_t F_t$$

$$-L\ln(1-\varepsilon) \leq \log(|H|) - \varepsilon M$$

$$M \approx (1+\varepsilon/2)L + \log(|H|)/\varepsilon$$

$$\varepsilon = \sqrt{\log(|H|)/L} \Rightarrow M \leq L + 2\sqrt{L\log(|H|)}$$

Γενικεύσεις – Εφαρμογές

- Γενικεύσεις – Παραλλαγές:
  - Πλαίσιο **πολλαπλασιαστικής ενημέρωσης βαρών** (multiplicative weight updates).
  - Fictitious play, winnow (αυξομείωση βαρών), Hedge όπου $w_{t+1}(h) = w_t(h)\exp(-\eta \text{cost})$, AdaBoost για συνδυασμό ταξινομητών, …
  - **Bandits**: επιλογή μόνο ενός h_t, για κάθε t, και μαθαίνουμε μόνο για $h_t(x_t) = y_t$
  - **Εκθετικά** μεγάλο |H|: μείωση διάστασης, πλαίσιο **Follow the (Regularized) Leader**.
- Εφαρμογές:
  - Μηχανική μάθηση (γραμμικός διαχωρισμός, boosting, …).
  - Θεωρία παιγνίων: 2-person 0-sum games, coarse correlated equilibrium
  - Εξελικτική θεωρία παιγνίων: replicator dynamics.
  - Βελτιστοποίηση: stochastic gradient descent, άμεση κυρτή βελτιστοποίηση
  - Προσεγγιστική επίλυση packing / covering γραμμικών προγραμμάτων.
  - Άμεσοι και προσεγγιστικοί αλγόριθμοι, αλγοριθμικός σχεδιασμός μηχανισμών
  - (Arora, Hazan, Kale, ToC 2012, https://theoryofcomputing.org/articles/v008a006/v008a006.pdf)