On-line Learning

Μηχανική Μάθηση

ΔΠΜΣ Επιστήμη Δεδομένων και Μηχανική Μάθηση

Γιώργος Αλεξανδρίδης – gealexandri@islab.ntua.gr

Προσπέλαση δεδομένων κατά τη μάθηση

- **Μάθηση κατά δέσμη** (*batch learning*)
  - *Σταθερή* κατανομή δειγματοληψίας δεδομένων
  - Δειγματοληψία σημείων **i.i.d** (*independent & identically distributed*)
  - *Σύνολα* δεδομένων **εκπαίδευσης** και **ελέγχου**
  - PAC Learning: εύρεση υπόθεσης που ελαχιστοποιεί το σφάλμα γενίκευσης

- **On-line Μάθηση** (*On-line learning*)
  - Επεξεργασία *ενός* δείγματος δεδομένων κάθε χρονική στιγμή
    - Δεν γίνεται υπόθεση για την *κατανομή* των δεδομένων, ούτε ισχύουν οι **i.i.d θεωρήσεις**
  - «*Μεικτή*» εκπαίδευση και έλεγχος
  - Δεν υπάρχει η έννοια του **σφάλματος γενίκευσης**
    - Απόδοση μετριέται μέσω **μοντέλου σφάλματος** (*mistake model*) και της έννοιας της **μετάνοιας** (*regret*)
  - *Εγγυήσεις* μάθησης προκύπτουν από την **ανάλυση της χειρότερη περίπτωσης** (*worst-case analysis*)
    - Γνωστή και ως **μάθηση με αντιπαλότητα** (*adversarial learning*)

On-line μάθηση

- **Χαρακτηριστικά**
  - **Μικρό** υπολογιστικό *κόστος*
    - *Παραγωγή* προβλέψεων
    - *Ενημέρωση* παραμέτρων μοντέλου
  - **Ευκολία** υλοποίησης
- **Παραδείγματα χρήσης**
  - **Χρηματοοικονομικός** τομέας
    - *Πρόβλεψη σημερινής ισοτιμίας νομισμάτων/μετοχών* κλπ
  - **Ειδησεογραφικά** portal
    - *Ποιους* από τους αναγνώστες θα μπορούσε να *ενδιαφέρει* μια «*έκτακτη*» είδηση;
  - …

Γενικό μοντέλο on-line μάθησης

- T **γύροι** εκπαίδευσης
- Για $t = 1$ μέχρι T
  1. Λήψη **δείγματος** $x_t \in \mathcal{X}$
  2. **Πρόβλεψη** τιμής/ετικέτας $\hat{y}_t \in \mathcal{Y}$
  3. **Λήψη** τιμής/ετικέτας $y_t \in \mathcal{Y}$
  4. Υπολογισμός συνάρτησης **κόστους** $L(\hat{y}_t, y_t)$: $L: \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}_+$
- Ταξινόμηση: $Y = \{0,1\}, L(y, y') = |y - y'|$
- Παλινδρόμηση: $Y \in \mathbb{R}, L(y, y') = (y - y')^2$
- **Σκοπός** μάθησης
  - *Ελαχιστοποίηση* **σωρευτικής** απώλειας $\sum_{t=1}^{T} L(\hat{y}_t, y_t)$
- **Κατηγορίες**
  1. Πρόβλεψη με **συμβουλή «ειδικού»** (*expert advice*)
  2. *Γραμμική* ταξινόμηση

Πρόβλεψη με συμβουλή «ειδικού»

Prediction with expert advice

Πρόβλεψη με συμβουλή «ειδικού»

- Σε <u>κάθε γύρο</u>, εκτός του δείγματος δεδομένων, το μοντέλο *λαμβάνει* τις **συμβουλές** N «ειδικών»
- **Αλγόριθμος**
  - Για $t = 1$ μέχρι T
    1. **Λήψη** δείγματος $x_t \in \mathcal{X}$ και συμβουλών $y_{t,i} \in \mathcal{Y}, i \in [1, N]$
    2. **Πρόβλεψη** τιμής/ετικέτας $\hat{y}_t \in \mathcal{Y}$
    3. **Λήψη** τιμής/ετικέτας $y_t \in \mathcal{Y}$
    4. Υπολογισμός συνάρτησης **κόστους** $L(\hat{y}_t, y_t)$: $L: \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}_+$
- **Σκοπός μάθησης**
  - *Ελαχιστοποίηση* **μετάνοιας** (*regret*)
    - *διαφοράς* μεταξύ *σωρευτικής απώλειας* και απώλειας του *καλύτερου* «ειδικού»
    - Regret(T)= $R_T = \sum_{t=1}^{T} L(\hat{y}_t, y_t) - \min_{N} \sum_{t=1}^{T} L\left(y_{t,i}, y_t\right)$

Μοντέλο Οριοθέτησης Λαθών

- **Mistake Bound Model**
  - *Πόσα* λάθη κάνει το μοντέλο μέχρι να μάθει μιας *έννοια*;
    - **Έννοια** (*concept*): *απεικόνιση* από τον χώρο των *δεδομένων* στον χώρο των *ετικετών*
- **Ορισμοί**
  - $M_{\mathcal{A}}(c) = \max_{x_1 \dots x_T} |\text{mistakes}(\mathcal{A}, c)|$
    - **Μέγιστος** αριθμός λαθών που κάνει ο *αλγόριθμος* $\mathcal{A}$ για να μάθει την *έννοια* c
  - $M_{\mathcal{A}}(\mathcal{C}) = \max_{c \in \mathcal{C}} M_{\mathcal{A}}(c)$
    - **Μέγιστος** αριθμός λαθών που κάνει ο *αλγόριθμος* $\mathcal{A}$ για να μάθει <u>οποιαδήποτε</u> *έννοια* στην κλάση εννοιών $\mathcal{C}$
- Επιθυμούμε να **θέσουμε όριο** M στο $M_{\mathcal{A}}(\mathcal{C})$

Αλγόριθμος HALVING

Halving($\mathcal{H}$)

$\mathcal{H}_1 \leftarrow \mathcal{H}$
for $t \leftarrow 1$ **to** T **do**
  Receive(x_t)
  $\widehat{y}_t \leftarrow$ MajorityVote($\mathcal{H}_t, x_t$)
  Receive(y_t)
  if $(\widehat{y}_t \neq y_t)$ **then**
    $\mathcal{H}_{t+1} \leftarrow \{c \in \mathcal{H}_t : c(x_t) = y_t\}$
  else $\mathcal{H}_{t+1} \leftarrow \mathcal{H}_t$
return $\mathcal{H}_{T+1}$

- **Απλός** αλγόριθμος, με αρκετά **καλά όρια**
  - $\mathcal{H}$ *πεπερασμένος* χώρος υποθέσεων
- **Όρια**
  - $M_{HALVING}(\mathcal{H}) \leq \log_2|\mathcal{H}|$
  - Αν $opt(\mathcal{H})$ βέλτιστο όριο λάθους ($opt(\mathcal{H}) \leq M_{HALVING}(\mathcal{H})$), επιπρόσθετα ισχύει $VCdim(\mathcal{H}) \leq opt(\mathcal{H})$

Αλγόριθμος Σταθμισμένης Πλειοψηφίας

```
WEIGHTED-MAJORITY(N)
for i ← 1 to N do
    w_{1,i} ← 1
for t ← 1 to T do
    RECEIVE(x_t)
    if ∑_{i: y_{t,i}=1} w_{t,i} ≥ ∑_{i: y_{t,i}=0} w_{t,i} then
        ŷ_t ← 1
    else ŷ_t ← 0
    RECEIVE(y_t)
    if (ŷ_t ≠ y_t) then
        for i ← 1 to N do
            if (y_{t,i} ≠ y_t) then
                w_{t+1,i} ← βw_{t,i}
            else w_{t+1,i} ← w_{t,i}
return w_{T+1}
```

- **Στάθμιση** της άποψης κάθε «*ειδικού*» ως προς το **ρυθμό** που κάνει λάθη
- **Αρχικά** οι N ειδικοί έχουν το ίδιο βάρος
- Πρόβλεψη ο **σταθμισμένος μέσος όρος** των προβλέψεων των «ειδικών»
- Σε περίπτωση που κάποιος κάνει λάθος, το **βάρος** του μειώνεται κατά *παράγοντα* β
  - Για $\beta = 0$ παίρνουμε τον *αλγόριθμο* HALVING

Αλγόριθμος Σταθμισμένης Πλειοψηφίας: Όρια

- m_T: αριθμός λαθών του **αλγορίθμου** μετά από $T \geq 1$ γύρους
- m_T^*: αριθμός λαθών του καλύτερου από τους N «*ειδικούς*»
- Ισχύει: $m_T \leq \frac{\log N + m_T^* \log\frac{1}{\beta}}{\log\frac{2}{1+\beta}}$
  - Η παραπάνω σχέση <u>εγγυάται</u> **άνω όριο** της μορφής
    - $m_T \leq O(\log N) + K m_T^*, K$ σταθερά
    - Από τη στιγμή που ο *πρώτος όρος* μεταβάλλεται *λογαριθμικά* σε σύγκριση με το πλήθος των «ειδικών», στην πράξη το **πλήθος** των λαθών του αλγορίθμου *εξαρτάται* από μια **σταθερά** επί το **πλήθος** των λαθών του καλύτερου «ειδικού»
- Ωστόσο, οι **ντετερμινιστικοί αλγόριθμοι** δεν επιτυγχάνουν καλά **όρια** για τη *μετάνοια*
  - Δεν μπορούν να τη «*φράξουν*» **γραμμικά** συναρτήσει του **πλήθους** των γύρων ($R_T \neq o(T)$)
  - <u>Λύση</u>: χρήση **randomized** αλγορίθμων

Randomized Αλγόριθμος Σταθμισμένης Πλειοψηφίας

Randomized-Weighted-Majority (N)

for $i \leftarrow 1$ **to** N **do**
    $w_{1,i} \leftarrow 1$
    $p_{1,i} \leftarrow 1/N$
for $t \leftarrow 1$ **to** T **do**
    Receive$(\mathbf{l}_t)$
    for $i \leftarrow 1$ **to** N **do**
        if $(l_{t,i} = 1)$ **then**
            $w_{t+1,i} \leftarrow \beta w_{t,i}$
        else $w_{t+1,i} \leftarrow w_{t,i}$
    $W_{t+1} \leftarrow \sum_{i=1}^{N} w_{t+1,i}$
    for $i \leftarrow 1$ **to** N **do**
        $p_{t+1,i} \leftarrow w_{t+1,i}/W_{t+1}$
return $\mathbf{w}_{T+1}$

- Στην περίπτωση αυτή, ο αλγόριθμος καλείται να **επιλέξει** από N *πιθανές δράσεις* (όχι «ειδικοί»)
- Σε κάθε γύρο, υπολογισμός *κατανομής* $\boldsymbol{p}_t$ πάνω στις δράσεις
  - Αρχικά ομοιόμορφη
- Λήψη διανύσματος απώλειας $\boldsymbol{l}_t$
- *Επαναπροσδιορισμός* **συνεισφοράς δράσης** όπως στην περίπτωση του ντετερμινιστικού αλγορίθμου σταθμισμένης πλειοψηφίας

Randomized Αλγόριθμος Σταθμισμένης Πλειοψηφίας: Όρια

- Λάθος στον γύρο t: $L_t = \sum_{i=1}^{N} p_{t,i} l_{t,i}$
  - Θεωρούμε δυαδική απώλεια $l_{t,i} \in \{0,1\}$
- Λάθος μετά από T γύρους: $\mathcal{L}_T = \sum_{t=1}^{T} L_t$
- Λάθος ενέργειας i μετά από T γύρους: $\mathcal{L}_{T,i} = \sum_{t=1}^{T} l_{t,i}$
- Ελάχιστο λάθος ενεργειών: $\mathcal{L}_T^{\min} = \min_i \mathcal{L}_{T,i}$
- «Μετάνοια»: $R_T = \mathcal{L}_T - \mathcal{L}_T^{\min}$

- **Θεώρημα**
  - Για $\beta \in [\frac{1}{2}, 1)$ και για $T \geq 1$ **ισχύει**: $\mathcal{L}_T \leq \frac{\log N}{1-\beta} + (2-\beta)\mathcal{L}_T^{\min}$
  - Επιπρόσθετα για $\beta = \max\{\frac{1}{2}, 1 - \sqrt{\frac{\log N}{T}}\}$ **ισχύει** $\mathcal{L}_T \leq \mathcal{L}_T^{\min} + 2\sqrt{T \log N}$
    - Άρα $R_T \leq 2\sqrt{T \log N}$ οπότε $R_T = O(\sqrt{T})$ και η *μέση μετάνοια ανά γύρο* $\frac{R_T}{T}$ μειώνεται με ρυθμό $O(\frac{1}{\sqrt{T}})$

Αλγόριθμος Εκθετικής Σταθμισμένης Πλειοψηφίας

Exponential-Weighted-Average (N)

for $i \leftarrow 1$ **to** N **do**
  $w_{1,i} \leftarrow 1$
for $t \leftarrow 1$ **to** T **do**
  Receive(x_t)
  $\widehat{y}_t \leftarrow \frac{\sum_{i=1}^{N} w_{t,i} y_{t,i}}{\sum_{i=1}^{N} w_{t,i}}$
  Receive(y_t)
  for $i \leftarrow 1$ **to** N **do**
    $w_{t+1,i} \leftarrow w_{t,i}\, e^{-\eta L(\widehat{y}_{t,i}, y_t)}$
return $\mathbf{w}_{T+1}$

- **Ντετερμινιστικός** αλγόριθμος
- L **κυρτή** (*convex*) συνάρτηση σφάλματος και λαμβάνει τιμές στο [0,1]
  - $L_{t,i}$ συνολικό σφάλμα i-οστού «ειδικού» μετά από t γύρους

Αλγόριθμος Εκθετικής Σταθμισμένης Πλειοψηφίας: Όρια

- Για *κυρτή* συνάρτηση **σφάλματος** $L \in [0,1]$, για $\eta > 0$ και <u>για κάθε</u> ακολουθία ετικετών $y_1 \ldots y_T \in \mathcal{Y}$, η «*μετάνοια*» του αλγορίθμου ικανοποιεί τη σχέση $R_T \leq \frac{\log N}{\eta} + \frac{\eta T}{8}$
- **Βέλτιστο** η
  - $\eta = \sqrt{\frac{8 \log N}{T}},\ R_T \leq \sqrt{\frac{T \log N}{2}}$ και $\frac{R_T}{T} = O(\sqrt{\frac{\log N}{T}})$
  - Μειονέκτημα ότι εξαρτάται από το **πλήθος** των γύρων (ορίζοντα) T
  - **Τρυκ διπλασιασμού** (*doubling trick*)
    - Χωρίζουμε τον χρόνο σε *περιόδους* $[2^k, 2^{k+1} - 1]$ διάρκειας $2^k, k \in [0, n]$, θέτουμε $T \geq 2^n - 1$ και επιλέγουμε $\eta_k = \sqrt{\frac{8 \log N}{2^k}}$ σε κάθε περίοδο
    - Τότε ισχύει $R_T \leq \frac{\sqrt{2}}{\sqrt{2}-1}\sqrt{\frac{T \log N}{2}} + \sqrt{\frac{\log N}{2}}$

Γραμμική ταξινόμηση

Linear Classification

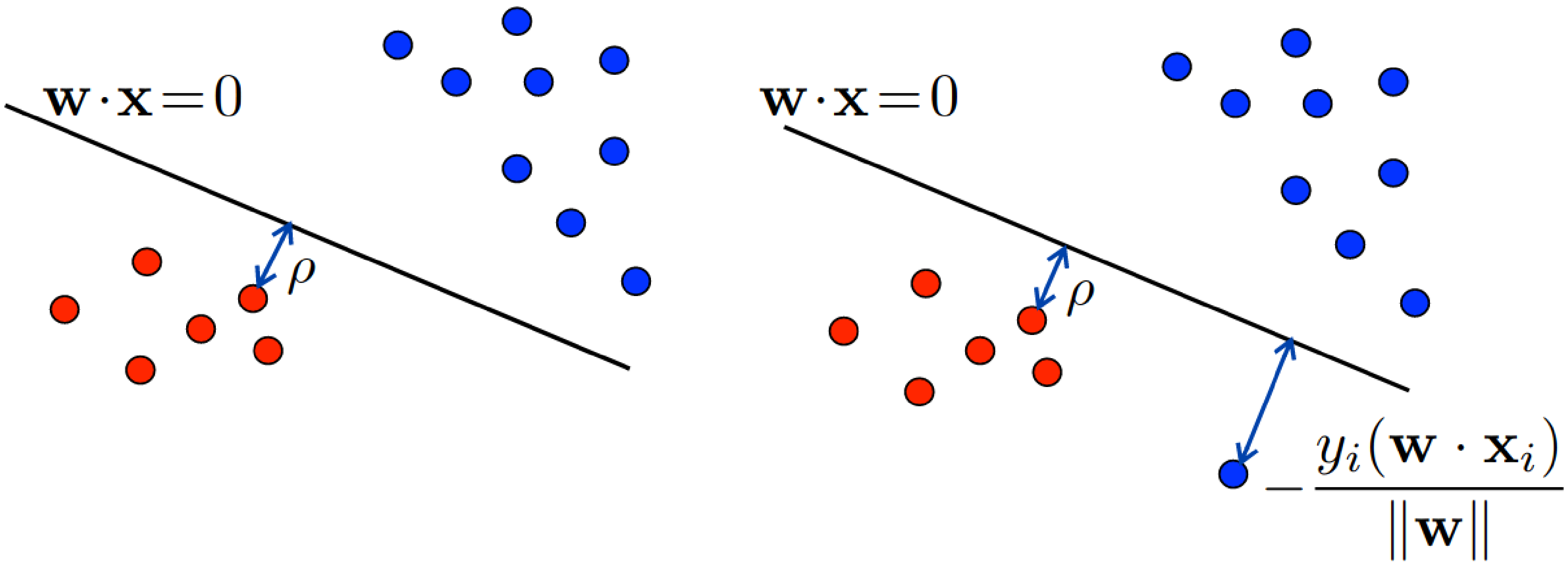
Perceptron

PERCEPTRON($\mathbf{w}_0$)

$\mathbf{w}_1 \leftarrow \mathbf{w}_0$ $\quad \triangleright$ typically $\mathbf{w}_0 = \mathbf{0}$
for $t \leftarrow 1$ **to** T **do**
  RECEIVE($\mathbf{x}_t$)
  $\widehat{y}_t \leftarrow \operatorname{sgn}(\mathbf{w}_t \cdot \mathbf{x}_t)$
  RECEIVE(y_t)
  if $(\widehat{y}_t \neq y_t)$ **then**
    $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_t \mathbf{x}_t$ $\quad \triangleright$ more generally $\eta y_t \mathbf{x}_t, \eta > 0$.
  else $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t$
return $\mathbf{w}_{T+1}$

- **On-line** έκδοση του **αλγορίθμου**
- $\boldsymbol{w}_t$: διάνυσμα βαρών
  - *Επίπεδο* **διαχωρισμού** κλάσεων
  - *Ενημέρωση* <u>μόνο</u> σε περίπτωση σφάλματος
- Ο αλγόριθμος *βελτιστοποιεί* **αντικειμενική συνάρτηση**
  - $F(\boldsymbol{w}) = \frac{1}{T}\sum_{t=1}^{T} \max(0, -y_t(\boldsymbol{w}\boldsymbol{x}_t))$
  - *Κυρτή*, αλλά όχι *διαφορίσημη*

Perceptron: Επίπεδο διαχωρισμού



Perceptron: Όρια για γραμμικά διαχωρίσιμες κλάσεις

- Έστω $\boldsymbol{x}_1 \dots \boldsymbol{x}_T \in \mathbb{R}^N$ ακολουθία T σημείων με $\|\boldsymbol{x}_t\| \leq r, \forall t \in [1, T]$ για $r > 0$. Υποθέτουμε ότι **υπάρχει** $\rho > 0$ και $\boldsymbol{v} \in \mathbb{R}^N$ έτσι ώστε $\rho \leq \frac{y_t(\boldsymbol{v}\boldsymbol{x}_t)}{\|\boldsymbol{v}\|}$. Τότε το **πλήθος** των ενημερώσεων που πραγματοποιεί ο αλγόριθμος είναι *φραγμένο* από τον λόγο $\frac{r^2}{\rho^2}$
  - **Κανονικοποιημένο** περιθώριο (*margin*)
- **Παρατηρήσεις**
  - **Αυστηρό** όριο, που <u>δεν εξαρτάται</u> από το **πλήθος** των διαστάσεων των δεδομένων, ούτε από τη σειρά τους
  - Σε περίπτωση που οι **κλάσεις** δεν είναι γραμμικά **διαχωρίσιμες**, ο αλγόριθμος δεν συγκλίνει
  - Για *μικρά περιθώρια*, η σύγκλιση μπορεί να είναι **πολύ αργή** $(\Omega(2^N))$

Perceptron: Όρια για μη-γραμμικά διαχωρίσιμες κλάσεις

- Έστω $\boldsymbol{x}_1 \dots \boldsymbol{x}_T \in \mathbb{R}^N$ ακολουθία T σημείων με $\|\boldsymbol{x}_t\| \leq r, \forall t \in [1, T]$ για $r > 0$. Έστω $\mathcal{I}$ το *σύνολο* των *επαναλήψεων* κατά τις οποίες ο αλγόριθμος **ενημέρωσε** τα *βάρη* του. Τότε το πλήθος των ενημερώσεων $M_T = |\mathcal{I}|$ που πραγματοποιεί ο αλγόριθμος **φράσσεται** από το όριο
  - $M_T \leq \inf_{\rho>0, \|\boldsymbol{v}\|_2 \leq 1} \left(\frac{r}{\rho} + \sqrt{\left\|\mathbf{l}_\rho\right\|_1} \right)^2, \boldsymbol{l}_\rho = \left(\max\left\{0, 1 - \frac{y_t(\boldsymbol{v}\boldsymbol{x}_t)}{\rho}\right\} \right)_{t \in \mathcal{I}}$
    - $\mathbf{l}_\rho$: Απώλεια ρ-Hinge
    - $\boldsymbol{v}$: διάνυσμα βαρών
    - $\inf(S)$: **μέγιστο κάτω φράγμα** (*infimum*) συνόλου S
      - ο μεγαλύτερος πραγματικός αριθμός ο οποίος είναι μικρότερος ή ίσος με κάθε αριθμό που περιέχεται στο S
  - Ένα **λιγότερο ισχυρό όριο** ισχύει και για τη L_2 νόρμα
    - $M_T \leq \inf_{\rho>0, \|\boldsymbol{v}\|_2 \leq 1} \left(\frac{r}{\rho} + \sqrt{\left\|\mathbf{l}_\rho\right\|_2} \right)^2$

Dual Perceptron

DualPerceptron($\boldsymbol{\alpha}_0$)

$\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha}_0$ $\quad \triangleright$ typically $\boldsymbol{\alpha}_0 = \mathbf{0}$
for $t \leftarrow 1$ **to** T **do**
 Receive($\mathbf{x}_t$)
 $\widehat{y}_t \leftarrow \operatorname{sgn}(\sum_{s=1}^{T} \alpha_s y_s (\mathbf{x}_s \cdot \mathbf{x}_t))$
 Receive(y_t)
 if $(\widehat{y}_t \neq y_t)$ **then**
  $\alpha_t \leftarrow \alpha_t + 1$
 else $\alpha_t \leftarrow \alpha_t$
return $\boldsymbol{\alpha}$

- **Επέκταση** *γραμμικής διαχωρισιμότητας* σε χώρους **μεγαλύτερων** διαστάσεων
- Προϋποθέτει ότι έχουμε **διαθέσιμα** T δείγματα με τις *ετικέτες* τους
- Διάνυσμα συντελεστών $\boldsymbol{\alpha} \in \mathbb{R}^T$ που ανατίθεται σε κάθε δείγμα $\boldsymbol{x}_t$
- $\boldsymbol{w} = \sum_{s=1}^{T} \alpha_s y_s \boldsymbol{x}_s$
  - Ενημέρωση ισοδύναμη με προσθήκη όρου $y_t \boldsymbol{x}_t$ στο $\boldsymbol{w}$
  - Συσχέτιση με Perceptron

Kernel Perceptron

KernelPerceptron($\boldsymbol{\alpha}_0$)

$\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha}_0$ $\quad \triangleright$ typically $\boldsymbol{\alpha}_0 = \mathbf{0}$
for $t \leftarrow 1$ **to** T **do**
$\quad$ Receive(x_t)
$\quad$ $\widehat{y}_t \leftarrow \operatorname{sgn}(\sum_{s=1}^{T} \alpha_s y_s K(x_s, x_t))$
$\quad$ Receive(y_t)
$\quad$ **if** $(\widehat{y}_t \neq y_t)$ **then**
$\quad\quad$ $\alpha_t \leftarrow \alpha_t + 1$
$\quad$ **else** $\alpha_t \leftarrow \alpha_t$
return $\boldsymbol{\alpha}$

- **Dual Perceptron**
  - υπολογίζει *εσωτερικά* γινόμενα μεταξύ των *δειγμάτων* δεδομένων
- **Kernel Perceptron**
  - *επέκταση* της έννοιας του *εσωτερικού γινομένου* με τη χρήση **συμμετρικού και θετικά ορισμένου πυρήνα** (*Positive Definite Symmetric – PDS*)

Winnow

$\textsc{Winnow}(\eta)$

$w_1 \leftarrow \mathbf{1}/N$
for $t \leftarrow 1$ **to** T **do**
  $\textsc{Receive}(\mathbf{x}_t)$
  $\widehat{y}_t \leftarrow \operatorname{sgn}(\mathbf{w}_t \cdot \mathbf{x}_t)$
  $\textsc{Receive}(y_t)$
  if $(\widehat{y}_t \neq y_t)$ **then**
    $Z_t \leftarrow \sum_{i=1}^{N} w_{t,i} \exp(\eta y_t x_{t,i})$
    for $i \leftarrow 1$ **to** N **do**
      $w_{t+1,i} \leftarrow \frac{w_{t,i} \exp(\eta y_t x_{t,i})}{Z_t}$
  else $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t$
return $\mathbf{w}_{T+1}$

- **Επέκταση** Perceptron με *πολλαπλασιαστική* μεταβολή των βαρών
- Z_t: Παράγοντας *κανονικοποίησης*
- Αν y_tκαι $\boldsymbol{x}_{t,i}$ έχουν ίδιο πρόσημο $\boldsymbol{w}_{t,i}$ αυξάνεται
  - Αλλιώς μειώνεται

Winnow: Όρια

- *Σχέση* **Winnow** με τον **Αλγόριθμο Σταθμισμένης Πλειοψηφίας**
  - Αν $\boldsymbol{x}_{t,i} \in \{-1, +1\}$ το $\text{sgn}(\boldsymbol{w}_t \boldsymbol{x}_t)$ *συμπίπτει* με την ψήφο της **πλειοψηφίας**
  - Ο *πολλαπλασιασμός* με e^{η} ή $e^{-\eta}$ των βαρών των ορθών/εσφαλμένων **προβλέψεων** είναι <u>ισοδύναμος</u> με τον *πολλαπλασιασμό* με $\beta = e^{-2\eta}$ των **βαρών** των λανθασμένων προβλέψεων του Α.Σ.Π.
- **Αντίστοιχες** *συσχετίσεις* βρίσκονται και με *άλλους* αλγορίθμους
  - *Boosting*, Perceptron
- Έστω $\boldsymbol{x}_1 \dots \boldsymbol{x}_T \in \mathbb{R}^N$ ακολουθία T σημείων με $\|\boldsymbol{x}_t\|_\infty \leq r_\infty, \forall t \in [1, T]$ για $r_\infty > 0$. Υποθέτουμε ότι υπάρχει $\rho_\infty > 0$ και $\boldsymbol{v} \in \mathbb{R}^N, \boldsymbol{v} \geq 0$ έτσι ώστε $\rho_\infty \leq \frac{y_t(\boldsymbol{v}\boldsymbol{x}_t)}{\|\boldsymbol{v}\|}$. Τότε το **πλήθος** των *ενημερώσεων* που πραγματοποιεί ο αλγόριθμος είναι φραγμένο από τον λόγο $2\frac{r_\infty^2}{\rho_\infty^2}\log N$
  - **Παρόμοιο** όριο με *Perceptron* αλλά για <u>διαφορετικές νόρμες</u>
    - L_1 για το *Perceptron* και L_∞ για *Winnow*

Βιβλιογραφία

1. *M. Mohri, A. Rostamizadeh, A. Talwalker* – **Foundations of Machine Learning**
   - Κεφάλαιο 8