❖ Δένδρα Απόφασης

Πρόκειται για μια οικογένεια μη γραμμικών ταξινομητών. Είναι συστήματα απόφασης πολλών σταδίων (multistage), όπου οι κλάσεις απορρίπτονται διαδοχικά (sequentially), έως ότου φτάσουμε σε μια κλάση που θα είναι τελικά αποδεκτή. Για το λόγο αυτό:

- Ο χώρος των χαρακτηριστικών τεμαχίζεται σε μοναδικές περιοχές με έναν ακολουθιακό τρόπο.

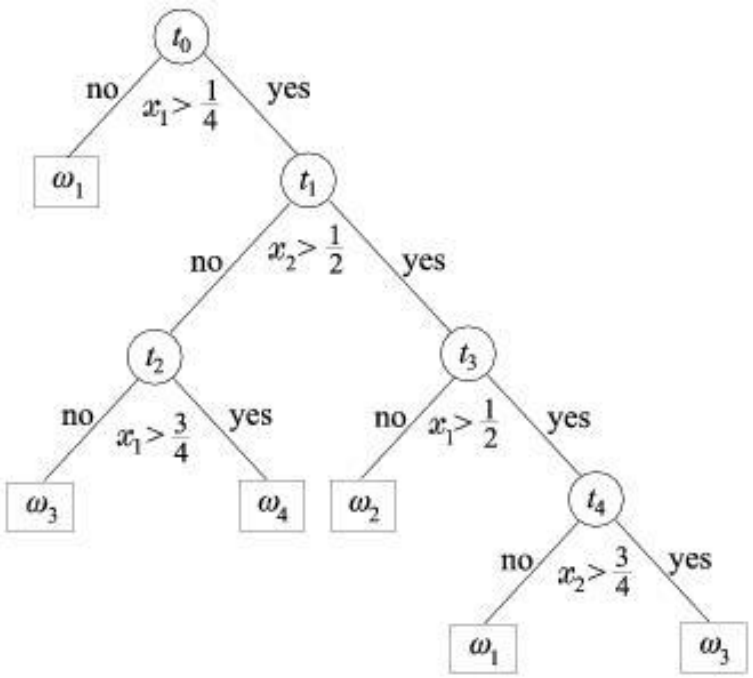
- Με την άφιξη ενός διανύσματος χαρακτηριστικών, λαμβάνονται διαδοχικές αποφάσεις καταχώρησης χαρακτηριστικών σε συγκεκριμένες περιοχές, ακολουθώντας ένα μονοπάτι κόμβων (nodes) ενός κατάλληλα κατασκευασμένου δένδρου.

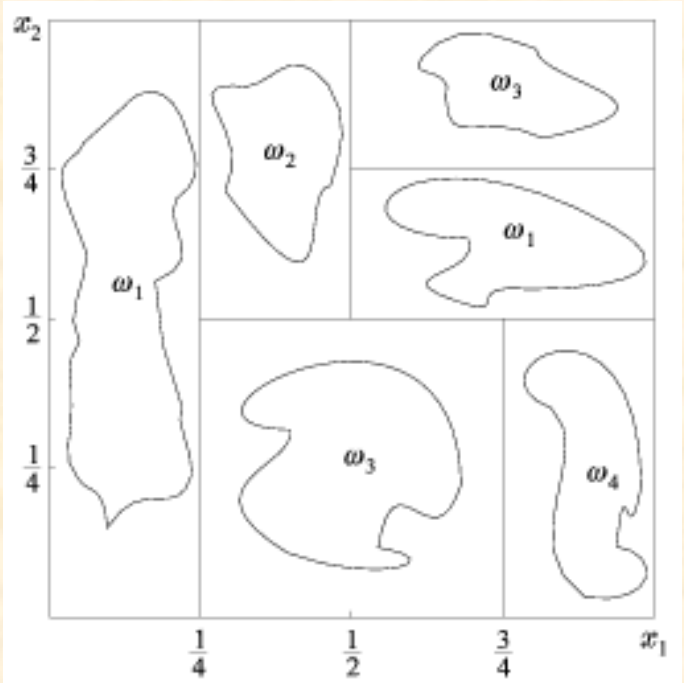
- Η ακολουθία των αποφάσεων εφαρμόζεται σε μεμονωμένα χαρακτηριστικά και οι ερωτήσεις που εξετάζονται σε κάθε κόμβο είναι του τύπου:

  είναι το χαρακτηριστικό $x_i \leq a$

  όπου a είναι ένα προεπιλεγμένο κατώφλι (επιλέγεται κατά τη διάρκεια της εκπαίδευσης).

- Τα παρακάτω σχήματα αντιστοιχούν σε τέτοια παραδείγματα. Τα δένδρα αυτού του τύπου είναι γνωστά ως Συνήθη Δυαδικά Δένδρα Ταξινόμησης (Ordinary Binary Classification Trees (OBCT)). Τα υπερπέπιπεδα απόφασης που διαιρούν το χώρο σε περιοχές, είναι παράλληλα στους άξονες του χώρου των δειγμάτων. Άλλοι τύποι διαίρεσης του χώροι είναι επίσης δυνατοί, παρότι είναι λιγότερο δημοφιλείς.





- Σχεδιαστικά στοιχεία που ορίζουν ένα δένδρο απόφασης.
  - Κάθε κόμβος, t, αντιστοιχεί σε ένα υποσύνολο $X_t \subseteq X$, όπου X είναι το σύνολο εκπαίδευσης. Σε κάθε κόμβο, το αντίστοιχο σύνολο, X_t διαιρείται σε δύο (δυαδική διαίρεση) ξένα μεταξύ τους υποσύνολα-απογόνους $X_{t,Y}$ and $X_{t,N}$, ώστε

$$X_{t,Y} \cap X_{t,N} = \emptyset$$
$$X_{t,Y} \cup X_{t,N} = X_t$$

$X_{t,Y}$ είναι το υποσύνολο του X_t για το οποίο η απάντηση στην ερώτηση του κόμβου t είναι ΝΑΙ. $X_{t,N}$ είναι το υποσύνολο που αντιστοιχεί στο ΟΧΙ. Η διαίρεση αποφασίζεται με βάση την ερώτηση (query) που υιοθετείται.

- Ένα κριτήριο διαμερισμού πρέπει να υιοθετηθεί προκειμένου να επιτευχθεί η βέλτιστη διαμέριση του X_t στα $X_{t,Y}$ και $X_{t,N}$.

- Ένα κριτήριο τερματισμού-διαμέρισης (stop-splitting) πρέπει να υιοθετηθεί προκειμένου να ελεγχθεί η ανάπτυξη του δένδρου έως τους τερματικούς κόμβους (φύλλα – leafs).

- Απαιτείται ένας κανόνας καταχώρησης ενός τερματικού κόμβου σε μία κατηγορία.

- Σύνολο ερωτήσεων: Στα δένδρα OBCT το σύνολο των ερωτήσεων είναι του τύπου

είναι $x_i \leq a$;

Η επιλογή του συγκεκριμένου χαρακτηριστικού x_i και της τιμής του κατωφλίου α, για κάθε κόμβο t, είναι το αποτέλεσμα αναζήτησης, κατά την εκπαίδευση, ανάμεσα στα χαρακτηριστικά και ένα σύνολο από δυνατές τιμές κατωφλίου. Ο τελικός συνδυασμός είναι αυτός που οδηγεί στη βέλτιστη τιμή ενός κατάλληλου κριτηρίου.

- Κριτήριο διαίρεσης: Η κύρια ιδέα πίσω από τη διαίρεση σε κάθε κόμβο είναι τα υποσύνολα-απόγονοι $X_{t,Y}$ και $X_{t,N}$ που θα προκύψουν να παρουσιάζουν μεγαλύτερο βαθμό ομογενοποίησης ως προς τις κλάσεις, σε σχέση με αυτόν του X_t. Έτσι το κριτήριο που θα επιλεγεί θα πρέπει να είναι σε συμφωνία με αυτόν το στόχο. Ένα συχνά χρησιμοποιούμενο κριτήριο είναι ο βαθμός μη-καθαρότητας ενός κόμβου (node impurity):

$$I(t) = -\sum_{i=1}^{M} P(\omega_i \mid t) \log_2 P(\omega_t \mid t)$$

και

$$P(\omega_i \mid t) \approx \frac{N_t^i}{N_t}$$

όπου N_t^i είναι ο αριθμός των στοιχείων τουσυνόλου X_t, τα οποία ανήκουν στην κλάση ω_i. Η μείωση της μη-καθαρότητας ενός κόμβου (decrease in node impurity) ορίζεται ως:

$$\Delta I(t) = I(t) - \frac{N_{t,\mathrm{Y}}}{N_t} I(t_{\mathrm{Y}}) - \frac{N_{t,N}}{N_t} I(t_N)$$

- Ο στόχος είναι να επιλεγούν σε κάθε κόμβο εκείνοι οι παράμετροι (χαρακτηριστικό και κατώφλι), που οδηγούν σε μία διαίρεση, η οποία παρουσιάζει τη μεγαλύτερη δυνατή μείωση της μη-καθαρότητας.

- Γιατί η μεγαλύτερη δυνατή μείωση; Παρατηρείστε ότι η μέγιστη τιμή για την $I(t)$ λαμβάνεται όταν όλες οι κλάσεις είναι ισοπίθανες, δηλ. όταν το X_t παρουσιάζει τον ελάχιστο δυνατό βαθμό ομογενοποίησης.

➢ Κανόνας τερματισμού διαίρεσης. Υιοθέτησε ένα κατώφλι T και σταμάτα την περαιτέρω διαίρεση ενός κόμβου (δηλ. καταχώρησέ τον σαν φύλλο-τερματικό κόμβο), αν η μείωση της μη-καθαρότητας είναι μικρότερη από T. Δηλ. όταν ο κόμβος t είναι "αρκετά καθαρός".

➢ Κανόνας αντιστοίχησης σε κλάση: Καταχώρησε ένα φύλλο στην κλάση ω_j , για την οποία: $j = \arg\max_i P(\omega_i \mid t)$

➢ Summary of an OBCT algorithmic scheme:

- Begin with the root node, i.e., $X_t = X$
- For each new node t
  - For every feature $x_k, k = 1, 2, \ldots, l$
    - For every value $\alpha_{kn}, n = 1, 2, \ldots, N_{tk}$
      - Generate X_{tY} and X_{tN} according to the answer in the question: is $x_k(i) \leq \alpha_{kn},\ i = 1, 2, \ldots, N_t$
      - Compute the impurity decrease
    - End
    - Choose α_{kn_0} leading to the maximum decrease w.r. to x_k
  - End
  - Choose x_{k_0} and associated $\alpha_{k_0 n_0}$ leading to the overall maximum decrease of impurity
  - If stop-splitting rule is met declare node t as a leaf and designate it with a class label
  - If not, generate two descendant nodes t_Y and t_N with associated subsets X_{tY} and X_{tN}, depending on the answer to the question: is $x_{k_0} \leq \alpha_{k_0 n_0}$
- End

➢ Παρατηρήσεις:

- Ένας κρίσιμος παράγοντας κατά τη φάση σχεδιασμού είναι το μέγεθος του δένδρου. Συνήθως το δένδρο αναπτύσσεται έως ότου φτάσει σε μεγάλο μέγεθος και στη συνέχεια εφαρμόζονται διάφορες τεχνικές κλαδέματος (pruning).
- Τα δένδρα απόφασης ανήκουν στην κατηγορία των ασταθών (unstable) ταξινομητών. Αυτό μπορεί να αντιμετωπιστεί με τεχνικές «μέσου όρου» ("averaging" techniques). Π.χ. χρησιμοποιώντας τεχνικές bootstrapping στο X, κατασκευάζονται διάφορα δένδρα, T_i, $i=1, 2, \ldots, B$. Η απόφαση ταξινόμησης λαμβάνεται με βάση έναν κανόνα πλειοψηφίας (majority voting).

❖ Συνδυάζοντας ταξινομητές

Η βασική φιλοσοφία πίσω από το συνδυασμό διαφορετικών ταξινομητών βασίζεται στο γεγονός ότι ακόμα και ο «καλύτερος» ταξινομητής αποτυγχάνει σε μερικά διανύσματα όπου άλλοι ταξινομητές μπορεί να δώσουν σωστή ταξινόμηση. Ο συνδυασμός ταξινομητών σκοπεύει στην εκμετάλλευση αυτής της συμπληρωματικής πληροφορίας (complementary information) που δίνεται από διάφορους ταξινομητές.

Έτσι κάποιος σχεδιάζει διαφορετικούς βέλτιστους ταξινομητές και στη συνέχεια συνδυάζει τα αποτελέσματα με βάση ένα συγκεκριμένο κανόνα.

- Έστω ότι καθένας από τους, L ταξινομητές που σχεδιάστηκαν δίνει στην έξοδό του τις εκ των υστέρων πιθανότητας

$$P(\omega_i \mid \underline{x}), i = 1, 2, \ldots, M$$

- Κανόνας γινομένου: Καταχώρησε το $\underline{x}$ στην κλάση ω_i:

$$i = \arg\max_k \prod_{j=1}^{L} P_j(\omega_k \mid \underline{x})$$

όπου $P_j(\omega_k \mid \underline{x})$ είναι η αντίστοιχη εκ των υστέρων πιθανότητα του j^{th} ταξινομητή.

- Κανόνας άθροισης: Καταχώρησε το $\underline{x}$ στην κλάση ω_i :

$$i = \arg\max_k \sum_{j=1}^{L} P_j(\omega_k \mid \underline{x})$$

- Κανόνας πλειοψηφίας: Καταχώρησε το $\underline{x}$ στην κλάση για την οποία υπάρχει ομοφωνία ή όταν τουλάχιστον ℓ_c από τους ταξινομητές συμφωνούν στην κλάση

$$\ell_c = \begin{cases} \frac{L}{2}+1, \text{ L even} \\ \frac{L+1}{2}, \text{ L odd} \end{cases}$$

Διαφορετικά η απόφαση είναι απόρριψη (rejection), δηλ. δεν λαμβάνεται καμία απόφαση.

Έτσι σωστή απόφαση λαμβάνεται όταν η πλειοψηφία των ταξινομητών συμφωνεί με τη σωστή κατηγορία και λάθος όταν η πλειοψηφία συμφωνεί με μία λάθος κατηγορία.

- Εξαρτημένοι ή ανεξάρτητοι ταξινομητές;

  - Παρότι δεν υπάρχουν γενικά θεωρητικά αποτελέσματα, το πείραμα έδειξε ότι όσο πιο ανεξάρτητοι είναι οι ταξινομητές ως προς τις αποφάσεις τους, τόσο πιο πολύ αναμένεται η λήψη βελτιωμένων αποτελεσμάτων μετά το συνδυασμό των επιμέρους αποφάσεων. Ωστόσο, δεν υπάρχει εγγύηση ότι ο συνδυασμός ταξινομητών οδηγεί σε καλύτερη απόδοση συγκρινόμενος με την απόδοση του "καλύτερου" ανάμεσα στους ταξινομητές.

- Προς την ανεξαρτησία: Δυνατά σενάρια

  - Εκπαίδευση μεμονωμένων ταξινομητών χρησιμοποιώντας διαφορετικά διανύσματα δεδομένων. Εδώ κάποιος έχει αρκετές επιλογές:
    - Bootstrapping: Πρόκειται για μία δημοφιλή τεχνική για το συνδυασμό ασταθών ταξινομητών, όπως είναι τα δένδρα απόφασης.

- Stacking: Εκπαίδευση του συνδυαστή με δεδομένα που δεν έχουν χρησιμοποιηθεί για την εκπαίδευση των μεμονωμένων ταξινομητών.
- Χρήση διαφορετικών υποχώρων για την εκπαίδευση μεμονωμένων ταξινομητών: Σύμφωνα μ' αυτή τη μέθοδο, κάθε μεμονωμένος ταξινομητής εκπαιδεύεται σε διαφορετικό υπόχωρο του χώρου των χαρακτηριστικών. Δηλ. χρησιμοποιούνται διαφορετικά χαρακτηριστικά για κάθε ταξινομητή.

➢ Παρατηρήσεις:

- Οι κανόνες πλειοψηφία και αθροίσματος συγκαταλέγονται ανάμεσα στα πιο δημοφιλή συνδυαστικά σχήματα.

- Η εκπαίδευση των μεμονωμένων ταξινομητών σε διαφορετικούς υποχώρους του χώρου των χαρακτηριστικών φαίνεται να οδηγεί σε σημαντικά καλύτερα αποτελέσματα σε σχέση με την περίπτωση όπου οι ταξινομητές εκπαιδεύονται στον ίδιο υπόχωρο.

- Εκτός από τους τρεις παραπάνω κανόνες, μπορούν να υιοθετηθούν και άλλοι όπως η τιμή της διαμέσου (Median value) των εξόδων των μεμονωμένων ταξινομητών.

❖ The Boosting Approach

- The origins: Is it possible a weak learning algorithm (one that performs slightly better than a random guessing) to be boosted into a strong algorithm? (Villiant 1984).

- The procedure to achieve it:

  - Adopt a weak classifier known as the base classifier.
  - Employing the base classifier, design a series of classifiers, in a hierarchical fashion, each time employing a different weighting of the training samples. Emphasis in the weighting is given on the hardest samples, i.e., the ones that keep "failing".
  - Combine the hierarchically designed classifiers by a weighted average procedure.

➢ The AdaBoost Algorithm.

Construct an optimally designed classifier of the form:

$$f(\underline{x}) = \text{sign}\left\{F(\underline{x})\right\}$$

where:

$$F(\underline{x}) = \sum_{k=1}^{K} a_k \varphi\left(\underline{x}; \underline{\vartheta}_k\right)$$

where $\varphi\left(\underline{x}; \underline{\vartheta}_k\right)$ denotes the <u>base classifier</u> that returns a binary class label:

$$\varphi\left(\underline{x}; \underline{\vartheta}_k\right) \in \{-1, 1\}$$

$\underline{\vartheta}$ is a parameter vector.

- The essence of the method.

  Design the series of classifiers:

$$\varphi(\underline{x};\underline{\vartheta}_1), \varphi(\underline{x};\underline{\vartheta}_2), \ldots, \varphi(\underline{x};\underline{\vartheta}_k)$$

  The parameter vectors

$$\underline{\vartheta}_k, k = 1, 2, \ldots, K$$

  are optimally computed so as:

  – To minimize the error rate on the training set.
  – Each time, the training samples are re-weighted so that the weight of each sample depends on its history. Hard samples that "insist" on failing to be predicted correctly, by the previously designed classifiers, are more heavily weighted.

- Updating the weights for each sample $\underline{x}_i, i = 1, 2, \ldots, N$

$$w_i^{(m+1)} = \frac{w_i^m \exp\left(-y_i a_m \varphi\left(\underline{x}_i; \underline{\vartheta}_m\right)\right)}{Z_m}$$

  - Z_m is a normalizing factor common for all samples.
  - $a_m = \frac{1}{2} \ln \frac{1 - P_m}{P_m}$

    where P_m<0.5 (by assumption) is the error rate of the optimal classifier $\varphi\left(\underline{x}; \underline{\vartheta}_m\right)$ at stage m. Thus $\alpha_m > 0$.
  - The term: $\exp\left(-y_i a_m \varphi\left(\underline{x}_i; \underline{\vartheta}_m\right)\right)$

    takes a large value if $y_i \varphi\left(\underline{x}_i; \underline{\vartheta}_m\right) < 0$ (wrong classification) and a small value in the case of correct classification $\{y_i \varphi\left(\underline{x}_i; \underline{\vartheta}_m\right) > 0\}$
  - The update equation is of a multiplicative nature. That is, successive large values of weights (hard samples) result in larger weight for the next iteration

- The algorithm

- Initialize: $w_i^{(1)} = \frac{1}{N},\ i = 1, 2 \ldots, N$
- Initialize: $m = 1$
- Repeat
  - Compute optimum $\boldsymbol{\theta}_m$ in $\phi(\cdot; \boldsymbol{\theta}_m)$ by minimizing P_m
  - Compute the optimum P_m
  - $\alpha_m = \frac{1}{2} \ln \frac{1-P_m}{P_m}$
  - $Z_m = 0.0$
  - For $i = 1$ to N
    - $w_i^{(m+1)} = w_i^{(m)} \exp\left(-y_i \alpha_m \phi(\boldsymbol{x}_i; \boldsymbol{\theta}_m)\right)$
    - $Z_m = Z_m + w_i^{(m+1)}$
  - End{For}
  - For $i = 1$ to N
    - $w_i^{(m+1)} = w_i^{(m+1)} / Z_m$
  - End {For}
  - $K = m$
  - $m = m + 1$
- Until a termination criterion is met.
- $f(\cdot) = \text{sign}(\sum_{k=1}^{K} \alpha_k \phi(\cdot, \theta_k))$

➢ Remarks:

- Training error rate tends to zero after a few iterations. The test error levels to some value.

- AdaBoost is greedy in reducing the margin that samples leave from the decision surface.