ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

$2^{η}$ ΣΕΙΡΑ ΓΡΑΠΤΩΝ ΑΣΚΗΣΕΩΝ

Ειρήνη Δόντη

ΑΜ: 03119839

8ο εξάμηνο

Αθήνα 2023

Άσκηση 1

1. $(p \Rightarrow \neg(p \vee q)) \Leftrightarrow ((r \wedge \neg q) \vee r)$

   Ξεκινάμε με το αριστερό μέρος: $(p \Rightarrow \neg(p \vee q))$ ή $(p \Rightarrow (\neg p \wedge \neg q))$ ή$(\neg p \vee (\neg p \wedge \neg q)) = (\neg p \wedge \neg q)$

   Συνεχίζουμε με το δεξί μέρος: $((r \wedge \neg q) \vee r)$ ή $((r \vee r) \wedge (r \vee \neg q))$

   ή $(r \wedge (r \vee \neg q))$ ή $(r \vee \neg q)$.

   Συνολικά: $(\neg(\neg p \wedge \neg q) \vee (r \vee \neg q)) \wedge ((\neg p \wedge \neg q) \vee \neg(r \vee \neg q)))$ ή $(p \vee q \vee r \vee \neg q) \wedge ((\neg p \wedge \neg q) \vee (\neg r \wedge q)))$ ή $(p \vee r) \wedge ((\neg p \vee (\neg r \wedge q)) \wedge (\neg q \vee (\neg r \wedge q)))$ ή $(p \vee r) \wedge ((\neg p \vee (\neg r \wedge q)) \wedge (\neg q \vee (\neg r \wedge q)))$ ή $(p \vee r) \wedge (((\neg p \vee \neg r) \wedge (\neg p \vee q))) \wedge ((\neg q \vee \neg r) \wedge (\neg q \vee q)))$

   ή $((p \vee r) \wedge (\neg p \vee \neg r) \wedge (\neg p \vee q) \wedge (\neg q \vee \neg r))$ το οποίο είναι σε CNF.

2. $\forall x, \forall y, \exists z. (\forall w. \left(p(x,y) \Rightarrow q(w)\right) \vee (p(y,z) \Rightarrow \neg q(w)))$

   Ισοδύναμα, χρησιμοποιώντας την τεχνική Skolemization για να απαλείψουμε τον υπαρξιακό ποσοδείκτη:

   $(\forall x, \forall y. \left(\forall w. \left(p(x,y) \Rightarrow q(w)\right)\right) \vee \left(\forall x, \forall y. \left(p(y, f(x,y)) \Rightarrow \neg q(w)\right)\right))$

   Αναλύουμε ξεχωριστά τους όρους: $\forall x, \forall y. \left(\forall w. \left(p(x,y) \Rightarrow q(w)\right)\right)$,

   $\forall x, \forall y. \left(\forall w. \left(\neg p(x,y) \vee q(w)\right)\right)$

   $\forall x, \forall y. \left(p(y, f(x,y)) \Rightarrow \neg q(w)\right)$ ή $\forall x, \forall y. \left(\neg p(y, f(x,y)) \vee \neg q(w)\right)$

   Οπότε, $\forall x, \forall y. (\forall w. \left(\neg p(x,y) \vee q(w)\right) \vee \left(\neg p(y, f(x,y)) \vee \neg q(w)\right))$ ή $\forall x, \forall y. ((\forall w. (p(x,y) \wedge \neg q(w)) \wedge p(y, f(x,y)) \wedge q(w))$ ή διαγράφοντας τους καθολικούς ποσοδείκτες η CNF μορφή είναι η εξής: $\{[p(x,y)], [\neg q(w)], [p(y, f(x,y))], [q(w')]\}$

Άσκηση 2

Δίνεται η γνώση K που αποτελείται από τις προτάσεις:

p(a, b)

p(b, a)

q(a)

q(b)

$\forall x. \forall y. \exists z. (\forall w. \big(p(x, z) \ \Rightarrow q(w)\big) \vee \ (p(y, z) \ \Rightarrow \ \neg q(w)))$

Εξετάζουμε τις προτάσεις της γνώσης K: Οι πρώτες τέσσερις προτάσεις θεωρούνται αληθείς, καθώς ορίζουμε το p ως μία σχέση που συνδέει τα στοιχεία α, β ή/και β, α αντίστοιχα. Επίσης, ορίζουμε το q ως μία σχέση που αναφέρεται στο στοιχείο α ή/και β.

Για να υπάρχει μοντέλο γνώσης, θέλουμε να αποδείξουμε ότι και η πέμπτη πρόταση είναι αληθής βάσει των παραπάνω. Αναλύουμε ξεχωριστά τους όρους:

$\forall w. \big(p(x, z) \ \Rightarrow q(w)\big)$: Για οποιοδήποτε w επιλέγεται αν η πρόταση $p(x, z)$ είναι αληθής, τότε αναγκαστικά η $q(w)$ πρέπει να είναι αληθής.

$(p(y, z) \ \Rightarrow \ \neg q(w))$: Για οποιοδήποτε w επιλέγεται αν η πρόταση $p(y, z)$ είναι αληθής, τότε αναγκαστικά η $q(w)$ πρέπει να είναι ψευδής.

Θεωρούμε τα y, z = a και x, w = b:

$\forall w. \big(p(b, a) \ \Rightarrow q(b)\big)$: Η πρόταση p(b, a) είναι αληθής, τότε και η q(b) πρέπει να είναι αληθής που ισχύει. Οπότε, αυτή η περίπτωση είναι αληθής.

$(p(\alpha, a) \ \Rightarrow \ \neg q(b))$: Η πρόταση p(a, a) είναι ψευδής, τότε και η q(b) πρέπει να είναι αληθής που ισχύει. Οπότε, αυτή η περίπτωση είναι αληθής.

Οπότε, υπάρχει το παραπάνω παράδειγμα μοντέλου γνώσης βάσει αυτών των προτάσεων.

<u>**Άσκηση 3**</u>

Δίνεται γνώση K που αποτελείται από τις προτάσεις:

p(a, b)

p(b, a)

q(a)

¬q(b)

$\forall x. \forall y. \exists z. (\forall w. (p(z, x) \Rightarrow q(w)) \vee (p(y, z) \Rightarrow \neg q(w)))$

Ελέγχουμε τη συνέπεια της K με τη βοήθεια του αλγορίθμου ανάλυσης:

Αρχικά, μετατρέπουμε σε ισοδύναμη πρόταση την πέμπτη πρόταση, αντίστοιχα με την άσκηση 1, υποερώτημα 2:

$\{[p(f(x, y), x)], [\neg q(w)], [p(y, f(x, y))], [q(w')]\}$

Οπότε, K = {p(a, b), p(b, a), q(a), ¬q(b), $p(f(x, y), x)$, $\neg q(w)$, $p(y, f(x, y))$, q(w')}

Επιλέγουμε τις δύο προτάσεις q(w'), $\neg q(w)$ και εφαρμόζουμε τον κανόνα ανάλυσης και συνεπώς παράγουμε το κενό σύνολο.

Παρατηρούμε, λοιπόν, ότι η γνώση K είναι αντιφατική και συνεπώς δεν υπάρχει μοντέλο που να ικανοποιεί όλες τις προτάσεις της K.

Άσκηση 4

Θέλουμε να βρούμε μία ερμηνεία που ικανοποιεί την πρόταση $\forall x.(\exists y.p(x,y) \Rightarrow q(a))$

Και δεν ικανοποιεί την πρόταση $(\forall x.\exists y.p(x,y)) \Rightarrow q(a))$, αλλιώς να αποδείξουμε ότι δεν υπάρχει αυτή η ερμηνεία.

Αν ισχύει η q(a), τότε ικανοποιούνται και οι δύο προτάσεις.

Αν δεν ισχύει η q(a), τότε, για να ικανοποιείται η πρώτη πρόταση, πρέπει να είναι αληθής η $\neg p(x,y1)$ για κάποιο y1. Στην περίπτωση αυτή ικανοποιείται και η δεύτερη πρόταση, καθώς ο όρος $\forall x.p(x,y1)$ είναι ψευδής για κάποιο y1 και συνεπώς δεν υπάρχει ερμηνεία που ικανοποιεί την πρώτη πρόταση και όχι τη δεύτερη.

Άσκηση 5

Παρακάτω, παρουσιάζεται το λογικό διάγραμμα:

parent(x, y) ← father(x, y). (1)

parent(x, y) ← mother(x, y). (2)

sibling(y, z) ← parent(y, x), parent(z, x). (3)

sibling(x, y) ← sibling(y, x) (4)

grandparent(x, z) ← parent(x, y), parent(y, z). (5)

cousin(y, z) ← grandparent(y, x), grandparent(z, x). (6)

mother(a, b) ← . (7)

father(b, c) ← . (8)

mother(d, b) ← . (9)

mother(e, c) ← . (11)

mother(f, b) ← . (12)

1. Γράφουμε το σύμπαν και τη βάση Herbrand του λογικού προγράμματος:

   Σύμπαν Herbrand UP ενός προγράμματος P: Είναι το σύνολο όλων των βασικών όρων που μπορούν να κατασκευαστούν από τις στθερές και τις συναρτήσεις του P.
   UP = {a, b, c, d, e, f}

   Βάση Herbrand BP: Είναι το σύνολο όλων των βασικών ατόμων που μπορούν να κατασκευαστούν από το UP και τα κατηγορήματα του P.
   BP = {parent(a, b), father(a, b), mother(a, b), parent(a, c), father(a, c), mother(a,c),…, sibling(a, b), sibling(a, c), sibling(b, a),…, grandparent(a, b), grandparent(a, c),…, cousin(a, b), cousin(a, c),…}

2.
   (a) Forward Chaining
       Εφαρμόζουμε τους κανόνες του προγράμματος αρχίζοντας από τα γνωστά γεγονότα για την παραγωγή νέων γεγονότων μέχρι και αν καταλήξουμε στα ζητούμενα ερωτήματα cousin(a,e), cousin(a,f), sibling(a,e):

       cousin(a, e)
       Από (6): grandparent(a, c), grandparent(e, c) παράγεται cousin(a, e)
       Από (5): parent(a, b), parent(b, c) παράγεται grandparent(a, c) και
       parent(e, b), parent(b, c) παράγεται grandparent(e, c)
       Από (2): mother(a, b) παράγεται parent(a, b)
       Από (1): father(b, c) παράγεται parent(b, c)

Παρατηρούμε ότι το ερώτημα parent(e, b) δεν μπορεί να παραχθεί, οπότε ανακοινώνεται *αποτυχία*.

<u>cousin(a, f)</u>

Από (6): grandparent(a, c), grandparent(f, c) παράγεται cousin(a, f)

Από (5): parent(a, b), parent(b, c) παράγεται grandparent(a, c) και parent(f, b), parent(b, c) παράγεται grandparent(f, c)

Από (2): mother(f, b) παράγεται parent(f, b)

Από (1): father(b, c) παράγεται parent(b, c)

Από (2): mother(a, b) παράγεται parent(a, b)

Από (1): father(b, c) παράγεται parent(b, c)

Παρατηρούμε ότι το ερώτημα cousin(a, f)
μπορεί να παραχθεί, οπότε ανακοινώνεται *επιτυχία*.

<u>sibling(a, e)</u>

Από (4): sibling(e, a) παράγεται sibling(a, e)

Από (3): parent(e, c), parent(a, c) παράγεται sibling(e, a)

Από (2): mother(e, c) παράγεται parent(e, c)

Παρατηρούμε ότι το ερώτημα parent(a, c) δεν μπορεί να παραχθεί, οπότε ανακοινώνεται *αποτυχία*.

(b) <u>Backward Chaining</u>

Ο αλγόριθμος βρίσκει τις συνεπαγωγές της βάσης γνώσης των οποίων το συμπέρασμα είναι cousin(a,e), cousin(a,f), sibling(a,e).

<u>cousin(a, e)</u>

Από (6): cousin(a, e) παράγεται από grandparent(a, x1), grandparent(e, x1)

Από (5): grandparent(a, x1) παράγεται από parent(a, y1), parent(y1, x1) και grandparent(e, x1) παράγεται από parent(e, y1), parent(y1, x1)

Από (2): parent(a, y1) παράγεται από mother(a, b) για y1 = b

Από (1): parent(y1 = b, x1) παράγεται από father(b, c) για x1 = c

Παρατηρούμε ότι το ερώτημα parent(e, y1 = b) δεν μπορεί να παραχθεί, οπότε ανακοινώνεται *αποτυχία*.

<u>cousin(a, f)</u>

Από (6): cousin(a, f) παράγεται από grandparent(a, x1), grandparent(f, x1)

Από (5): grandparent(a, x1) παράγεται από parent(a, y1), parent(y1, x1)

grandparent(f, x1) παράγεται από parent(f, y1), parent(y1, x1)

Από (2): parent(f, y1) παράγεται από mother(f, b) για y1 = b

Από (1): parent(y1 = b, x1) παράγεται father(b, c) για x1 = c

Από (2): parent(a, y1 = b) παράγεται από mother(a, b)

Από (1): parent(y1 = b, x1 = c) παράγεται από father(b, c)

Παρατηρούμε ότι το ερώτημα cousin(a, f)

μπορεί να παραχθεί, οπότε ανακοινώνεται *επιτυχία*.

<u>sibling(a, e)</u>

Από (4): sibling(a, e) παράγεται από sibling(e, a)

Από (3): sibling(e, a) παράγεται parent(e, x1), parent(a, x1)

Από (2): parent(e, x1) παράγεται από mother(e, c) για x1 = c

Παρατηρούμε ότι το ερώτημα parent(a, x1 = c) δεν μπορεί να παραχθεί, οπότε ανακοινώνεται *αποτυχία*.

3. Οι αλγόριθμοι Forward Chaining και Backward Chaining δεν παρέχουν την αναμενόμενη απάντηση, διότι διαισθητικά οι a και f είναι αδέρφια, οπότε το cousin(a, f) δεν πρέπει να επιστρέφει επιτυχία. Οπότε, μπορούμε να προσθέσουμε την πρόταση cousin(y, z) ← ¬sibling(y, z) ώστε να είναι διαισθητικά ορθότερη.

Άσκηση 6

1. Θα χρησιμοποιήσουμε επαγωγή για $w_0 = 0$:
   Βάση k = 1: Θεωρούμε ότι $w_0^1 = w_0 = w_0^2$, οπότε οι perceptrons μπορούν να προβλέψουν την ίδια κλάση για κάθε παράδειγμα μέχρι να περιλάβουν το πρώτο λάθος. Οπότε, $w_1^1 = w_0 + y_{11}x_{11} = y_{12}x_{12} = y_{11}x_{11}$ και επιπλέον $w_1^2 = w_0 + \eta y_{12}x_{12} = \eta y_{12}x_{12} = \eta y_{11}x_{11} = \eta w_1^1$
   Μετά το k-οστό λάθος και οι δύο perceptrons θα πρέπει να προβλέψουν την ίδια κλάση. Οπότε, $w_{k+1}^2 = w_k^2 + \eta y_{k+1(2)}x_{k+1(2)} = w_k^2 + \eta y_{k+1(1)}x_{k+1(1)} = \eta w_{k+1}^1$, καθώς $w_{k+1}^1 = w_k^1 + y_{k+1(1)}x_{k+1(1)}$.

   Οπότε, γενικά, ισχύει ότι $w_1 = \frac{w_2}{\eta}$

2. Ο αριθμός των απαιτούμενων επαναλήψεων μέχρι τη σύγκλιση του αλγορίθμου μπορεί να είναι διαφορετικός από τον βασικό αλγόριθμο. Η αλλαγή αυτή εξαρτάται από την τιμή της παραμέτρου η. Όταν η τιμή της η είναι μεγαλύτερη, ο αλγόριθμος μπορεί να συγκλίνει γρηγορότερα, ενώ όταν η τιμή της η είναι μικρότερη, μπορεί να απαιτηθούν περισσότερες επαναλήψεις για την σύγκλιση.

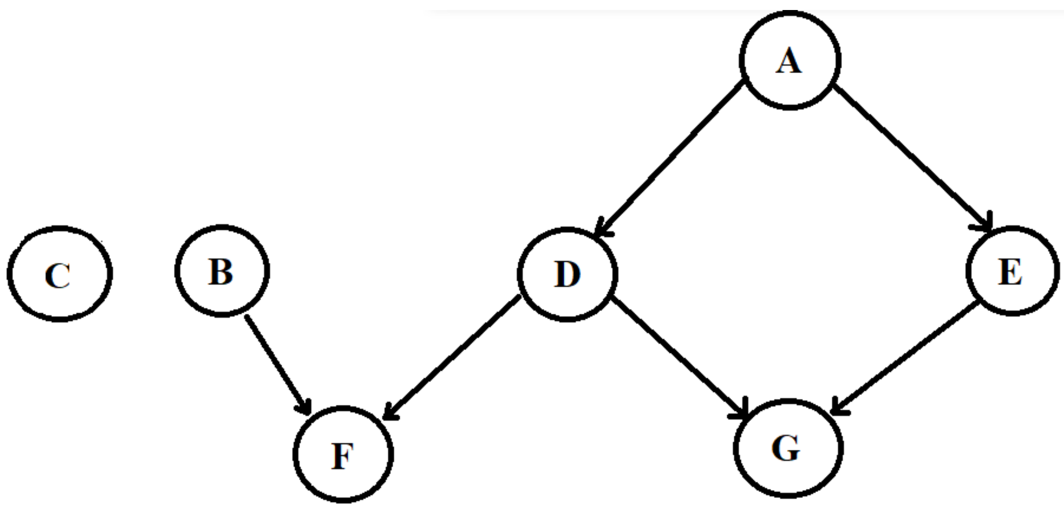
Άσκηση 7

1. Η σχεδίαση των δέντρων αποφάσεων στο πρόβλημα δυαδικής ταξινόμησης γίνεται με βάση τα χαρακτηριστικά των δεδομένων (που περιλαμβάνουν την είσοδο) και την ανάγκη να διαχωριστούν οι κλάσεις 0 και 1. Κάθε εσωτερικός κόμβος του δέντρου αντιστοιχεί σε ένα χαρακτηριστικό και ένα κατώφλι (threshold) για την απόφαση. Οι κλάδοι του δέντρου αντιπροσωπεύουν τις διάφορες πιθανές αποφάσεις και τις τελικές κλάσεις.

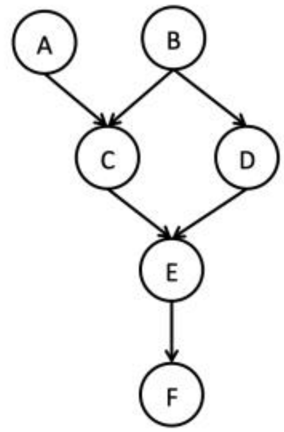
2. Το μέγιστο βάθος των δέντρων αποφάσεων εξαρτάται από τον αριθμό των χαρακτηριστικών (d) που χρησιμοποιούνται για την απόφαση. Το μέγιστο βάθος είναι d, καθώς κάθε εσωτερικός κόμβος αντιστοιχεί σε ένα χαρακτηριστικό.
3. Η VC-διάσταση της κλάσης των δέντρων αποφάσεων στον χώρο $\{0, 1\}^d$ είναι 2^d όσος και ο αριθμός των φύλλων εκ των οποίων μπορούμε να κατασκευάσουμε για είσοδο μέχρι 2^d διαφορετικών δειγμάτων.

Άσκηση 8

1. Σχεδιάζουμε το δίκτυο πίστης που αντιστοιχεί στην κατανομή:
   P(A, B, C, D, E, F,G) = P(A)P(B)P(C)P(D|A)P(E|A)P(F|B,D)P(G|D,E)



2. Η $J(<A, B, C, D, E, F>) = Pr(A)Pr(B)Pr(D|B)Pr(C|A\wedge B)Pr(E|C\wedge D)Pr(F|E)$ αντιστοιχεί στο παρακάτω δίκτυο πίστης.



3. Για να προσδιοριστεί πλήρως το δίκτυο πίστης, πρέπει να γνωρίζουμε 1 + 1 + 2 + 4 + 4 + 2 = 14 πιθανότητες.