

Εισαγωγή στη Μηχανική Μάθηση



DALLE-3: An oil painted expensive art showing information flowing in a box, which represents a machine learning model that outputs knowledge

Γρηγόριος Τσουμάκας
Καθηγητής Μηχανικής Μάθησης και Ανακάλυψης Γνώσης
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

ΕΙΣΑΓΩΓΗ ΣΤΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Τίτλος πρωτότυπου: «Εισαγωγή στη Μηχανική Μάθηση»
Copyright © 2023



Το παρόν έργο διατίθεται με τους όρους της άδειας Creative Commons Αναφορά Δημιουργού – Μη Εμπορική Χρήση – Παρόμοια Διανομή 4.0. Για να δείτε τους όρους της άδειας αυτής επισκεφτείτε τον ιστότοπο <https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode.el>

Αν τυχόν κάποιο τμήμα του έργου διατίθεται με διαφορετικό καθεστώς αδειοδότησης, αυτό αναφέρεται ρητά και ειδικώς στην οικεία θέση.

Αφιερώνεται
στους φοιτητές του ΑΠΘ

ΠΕΡΙΕΧΟΜΕΝΑ

Πίνακας συντομεύσεων - ακρωνυμίων	ix
Πρόλογος	xi
1 Εισαγωγή	1
1.1 Εισαγωγή στη Μηχανική Μάθηση	1
1.1.1 Τι είναι η μηχανική μάθηση;	2
1.1.2 Γιατί να χρησιμοποιήσει κάποιος μηχανική μάθηση;	3
1.1.3 Γιατί να μην χρησιμοποιήσει κάποιος μηχανική μάθηση;	3
1.1.4 Κατηγορίες μηχανικής μάθησης	4
1.2 Μάθηση με Επίβλεψη	4
1.2.1 Ταξινόμηση και παλινδρόμηση	5
1.2.2 Οικογένειες αλγορίθμων	7
1.2.3 Συναρτήσεις απώλειας και κόστους	7
1.3 Περαιτέρω Μελέτη	8
2 Γραμμικά Μοντέλα	9
2.1 Γραμμική Παλινδρόμηση	9
2.1.1 Αναπαράσταση υποθέσεων	10
2.1.2 Αξιολόγηση υποθέσεων	10
2.1.3 Αναζήτηση της βέλτιστης υπόθεσης	12
2.1.4 Κανονική εξίσωση	13
2.2 Λογιστική Παλινδρόμηση	14
2.2.1 Αναπαράσταση υποθέσεων	14
2.2.2 Αξιολόγηση υποθέσεων	15
2.2.3 Αναζήτηση της βέλτιστης υπόθεσης	16
2.3 Ταξινόμηση σε Πάνω από Δύο Κλάσεις	17
2.3.1 Αποσύνθεση σε πολλές εργασίες 2 κλάσεων	17
2.3.2 Πολυχοτομική λογιστική παλινδρόμηση	17
2.4 Ρυθμός Μάθησης	20
2.5 Κλιμάκωση και Προτυποποίηση Χαρακτηριστικών	21

2.6 Στοχαστική Επικλινής Κάθοδος	22
2.7 Πολυωνυμικά Μοντέλα και Υπερπροσαρμογή	23
2.8 Περαιτέρω Μελέτη	25
2.9 Ασκήσεις	25
3 Δενδρικά Μοντέλα	27
3.1 Αναπαράσταση	27
3.2 Αναζήτηση και Αξιολόγηση	28
3.2.1 ID3	29
3.2.2 C4.5	35
3.2.3 CART	37
3.3 Υπερπροσαρμογή	38
3.4 Ερμηνευσιμότητα	39
3.5 Περαιτέρω Μελέτη	40
3.6 Ασκήσεις	40
4 Αξιολόγηση Μοντέλων	43
4.1 Διαδικασίες Αξιολόγησης	43
4.1.1 Κράτηση και σταυρωτή επικύρωση	44
4.1.2 Στρωματοποίηση και ομαδοποίηση	46
4.2 Μετρικές Αξιολόγησης	46
4.2.1 Παλινδρόμηση	46
4.2.2 Ταξινόμηση	48
4.3 Περαιτέρω Μελέτη	51
4.4 Ασκήσεις	51
5 Μοντέλα Κανόνων	53
5.1 Σύνολα Κανόνων	53
5.1.1 Ταξινομημένα σύνολα κανόνων	54
5.1.2 Μη ταξινομημένα σύνολα κανόνων	57
5.1.2.1 Χρήση παραδείγματος σπόρου	60
5.1.2.2 Ακτινωτή αναζήτηση	60
5.2 Εξελικτική Μάθηση Κανόνων	62
5.2.1 Γενετικοί αλγόριθμοι	62
5.2.2 Μέθοδοι επιλογής μελών	62
5.2.3 Εξέλιξη και μάθηση	63
5.2.4 Εξέλιξη συνόλων κανόνων	64
5.3 Περαιτέρω Μελέτη	64
5.4 Ασκήσεις	64
6 Μάθηση κατά Περίπτωση	67
6.1 Πλησιέστεροι Γείτονες	67
6.1.1 Σταθμισμένη απόσταση	69
6.1.2 Πολυπλοκότητα	70
6.1.3 Θέματα υπολογισμού απόστασης	70
6.1.4 Η κατάρα των διαστάσεων	70
6.1.5 Παρατηρήσεις	71
6.2 Περαιτέρω Μελέτη	71
7 Σύνολα Μοντέλων	73

7.1	Συγκερασμός Μοντέλων	74
7.1.1	Σωρευμένη γενίκευση	75
7.2	Τεχνικές Δειγματοληψίας	76
7.2.1	Αυτοδύναμη συνάθροιση	76
7.2.2	Μέθοδος του τυχαίου υποχώρου	77
7.3	Τεχνικές Ενίσχυσης	78
7.3.1	Προσαρμοστική ενίσχυση	78
7.3.2	Επικλινής ενίσχυση	79
7.4	Τυχαίο Δάσος	80
7.5	Περαιτέρω Μελέτη	80
7.6	Ασκήσεις	81
8	Ενισχυτική Μάθηση	83
8.1	Μαρκοβιανές Διαδικασίες Απόφασης	83
8.1.1	Ανταμοιβές και κέρδος	85
8.1.2	Συναρτήσεις Αξίας	87
8.1.3	Βέλτιστες συναρτήσεις αξίας	89
8.2	Δυναμικός Προγραμματισμός	90
8.2.1	Επανάληψη πολιτικής	90
8.2.2	Επανάληψη αξίας	93
8.2.3	Ασύγχρονος δυναμικός προγραμματισμός	94
8.3	Μέθοδοι Μόντε Κάρλο	95
8.3.1	Αξιολόγηση πολιτικής	95
8.3.2	Επανάληψη πολιτικής	96
8.3.3	Αυξητικοί υπολογισμοί	96
8.4	Μέθοδοι Χρονικών Διαφορών	97
8.4.1	Αξιολόγηση πολιτικής	98
8.4.2	Επανάληψη πολιτικής	99
8.5	Ασκήσεις	99
I	Παραρτήματα	103
A	Απαντήσεις ερωτήσεων - Λύσεις ασκήσεων	105
A.1	Κεφάλαιο 2	105
A.2	Κεφάλαιο 3	107
A.3	Κεφάλαιο 4	116
A.4	Κεφάλαιο 5	120
A.5	Κεφάλαιο 7	123
A.6	Κεφάλαιο 8	127

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

1.1	Το παραδοσιακό λογισμικό (Software 1.0) αναπτύσσεται από έναν ή παραπάνω προγραμματιστές. Το Software 1.0 δέχεται ως είσοδο δεδομένα και παράγει τη αντίστοιχη έξοδο. Η επόμενη γενιά λογισμικού (Software 2.0) αναπτύσσεται από έναν αλγόριθμο μηχανικής μάθησης, στον οποίο δίνουμε ως είσοδο τα δεδομένα και την επιθυμητή έξοδο του λογισμικού.	2
1.2	Παράδειγμα συνόλου εκπαίδευσης.	5
1.3	Η διαδικασία της επαγωγικής μάθησης.	5
1.4	Ταξινόμηση πελατών μιας τράπεζας σε χαμηλού και υψηλού ρίσκου.	6
1.5	Εκτίμηση τιμής μεταχειρισμένων αυτοκινήτων.	7
2.1	Γραμμικό μοντέλο εκτίμησης αξίας κατοικιών με βάση το εμβαδόν τους.	10
2.2	Η συνάρτηση κόστους σε μια εργασία παλινδρόμησης με μία μεταβλητή εισόδου.	11
2.3	Η συνάρτηση κόστους μιας εργασίας παλινδρόμησης ως προς μία μεταβλητή εισόδου.	12
2.4	Η σιγμοειδής συνάρτηση $\sigma(z) = \frac{1}{1+e^{-z}}$	14
2.5	Ένα απλό πρόβλημα δυαδικής ταξινόμησης με δύο μεταβλητές εισόδου.	15
2.6	Η συνάρτηση της λογαριθμικής απώλειας.	16
2.7	Η εκθετική συνάρτηση.	18
2.8	Γραμμική παλινδρόμηση για διαφορετικές τιμές της παραμέτρου η	21
3.1	Ένα δένδρο απόφασης που ταξινομεί μια ημέρα ως προς την καταλληλότητα για την διεξαγωγή ενός αγώνα τένις σύμφωνα με τις καιρικές συνθήκες που επικρατούν.	28
3.2	Γράφημα εντροπίας για δύο κλάσεις.	30
3.3	Γράφημα εντροπίας για τρεις κλάσεις.	31
3.4	Σύγκριση εντροπίας, Gini και το 1/2 της εντροπίας για δύο κλάσεις	38
3.5	Επιφάνεια απόφασης δένδρων με διαφορετικό μέγιστο βάθος για δύο χαρακτηριστικά του συνόλου δεδομένων Iris.	38
3.6	Οπτικοποίηση δένδρου απόφασης για το σύνολο δεδομένων Iris.	39
3.7	Σημαντικότητα μεταβλητών εισόδου στο σύνολο δεδομένων Iris.	40
4.1	Η διαδικασία της k -πλής σταυρωτής επικύρωσης.	45
6.1	Παράδειγμα ταξινόμησης με τη χρήση του αλγορίθμου k NN.	68
6.2	Διάγραμμα Voronoi για ένα σύνολο πέντε διδιάστατων παραδειγμάτων.	69

7.1	Σωρευμένη γενίκευση.	75
8.1	Η αλληλεπίδραση του πράκτορα με το περιβάλλον.	84
8.2	Γράφος μετάβασης της ΠΜΔΑ για το ρομπότ ανακύκλωσης.	86
8.3	Ένας κόσμος πλέγματος 5x5 και η αντίστοιχη συνάρτηση αξίας κατάστασης.	88
8.4	Βέλτιστη πολιτική και συνάρτηση αξίας στον κόσμο πλέγματος.	90
8.5	Κόσμος πλέγματος 4×4 .	91
8.6	Παράδειγμα της επαναληπτικής αξιολόγησης πολιτικής.	91
8.7	Επαναληπτική αξιολόγηση και βελτίωση πολιτικής.	93

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

2.1	Ένα εξαιρετικά μικρό σύνολο δεδομένων εκπαίδευσης για μια εργασία παλινδρόμησης με μία μεταβλητή εισόδου.	11
2.2	Παραδείγματα εκπαίδευσης ανά μοντέλο στις μεθόδους αποσύνθεσης μία εναντίων των υπολοίπων και μία εναντίον μίας	18
3.1	Ένα σύνολο παραδειγμάτων με τις καιρικές συνθήκες και την καταλληλότητα διεξαγωγής ενός αγώνα τένις.	31
3.2	Τιμές της μεταβλητής εισόδου θερμοκρασία και της μεταβλητής εξόδου σε ένα φύλλο.	36
4.1	Πίνακας σύγχυσης ενός μοντέλου ταξινόμησης μηνυμάτων ηλεκτρονικού ταχυδρομείου σε τρεις κλάσεις.	48
4.2	Πίνακας σύγχυσης για ένα πρόβλημα δυαδικής ταξινόμησης.	49
5.1	Σύνολο δεδομένων που ταξινομεί ένα θαλάσσιο κήτος ως δελφίνι σύμφωνα με κάποια χαρακτηριστικά.	55
5.2	Οι συνθήκες του συνόλου δεδομένων του Πίνακα 5.1 με τον αριθμό κάλυψης για την κλάση Δελφίνι.	55
5.3	Το σύνολο δεδομένων του Πίνακα 5.1. Με κόκκινο τα παραδείγματα που καλύπτονται από τον πρώτο κανόνα, με μπλε αυτά που καλύπτονται από τον δεύτερο κανόνα και με πράσινο αυτά του τρίτου κανόνα.	56
5.4	Οι συνθήκες του συνόλου δεδομένων του Πίνακα 5.1 με τον αριθμό κάλυψης για την κλάση Δελφίνι μετά τον πρώτο κανόνα.	56
5.5	Οι συνθήκες του συνόλου δεδομένων του Πίνακα 5.1 με τον αριθμό κάλυψης για την κλάση Δελφίνι μετά τον δεύτερο κανόνα.	57
5.6	Οι συνθήκες του συνόλου δεδομένων του Πίνακα 5.1 με τον αριθμό κάλυψης για την κλάση Δελφίνι=Ναι (1η επανάληψη).	59
5.7	Το σύνολο δεδομένων του Πίνακα 5.1. Με κόκκινο τα παραδείγματα που καλύπτονται από τον πρώτο κανόνα, με μπλε τα παραδείγματα του δεύτερου κανόνα και με πράσινο του τρίτου κανόνα για την κλάση Δελφίνι = Ναι.	59
5.8	Οι συνθήκες του συνόλου δεδομένων του Πίνακα 5.1 με τον αριθμό κάλυψης για την κλάση Δελφίνι=Ναι (2η Επανάληψη - 1η συνθήκη).	60

5.9 Οι συνθήκες του συνόλου δεδομένων του Πίνακα 5.1 με τον αριθμό κάλυψης για την κλάση Δελφίνι=Nai (2η Επανάληψη - 2η συνθήκη).	60
5.10 Οι συνθήκες του συνόλου δεδομένων του Πίνακα 5.1 με τον αριθμό κάλυψης για την κλάση Δελφίνι=Nai. Ο αριθμός των συνθηκών περιορίζεται με την χρήση του πρώτου παραδείγματος ως παράδειγμα σπόρος.	61
5.11 Το σύνολο δεδομένων του Πίνακα 5.1. Με κόκκινο τα παραδείγματα που καλύπτονται από τον πρώτο κανόνα, με μπλε τα παραδείγματα του δεύτερου κανόνα και με μωβ τα παραδείγματα που καλύπτονται και από τους δύο κανόνες.	61
5.12 Σ λύσεις και η καταλληλότητας τους.	63
5.13 Σ λύσεις και η καταλληλότητας τους.	63
7.1 Δύο σύνολα μοντέλων, καθένα από τα οποία αποτελείται από τρία μοντέλα δυαδικής ταξινόμησης, όπου το κάθε μοντέλο έχει ορθότητα 75%.	74
7.2 Παραδείγματα απλών συναρτήσεων συγκερασμού των προβλέψεων ενός συνόλου τριών μοντέλων παλινδρόμησης και ταξινόμησης (εργασία με τρεις κλάσεις).	75
7.3 Παράδειγμα συνόλων εκπαίδευσης που παράγονται με την εφαρμογή της αυτοδύναμης συνάθροισης.	77
7.4 Παραλλαγές ενός συνόλου εκπαίδευσης με τη μέθοδο των τυχαίων υποχώρων.	77
8.1 Δυναμική μιας ΠΜΔΑ για ένα ρομπότ ανακύκλωσης.	85
8.2 Πολιτική π_0 τυχαίας επιλογής ενέργειας σε κάθε κατάσταση και πολιτική π_1 έπειτα από ένα βήμα αξιολόγησης και βελτίωσης.	93
8.3 Κέρδος έπειτα από κάθε επεισόδιο για κάθε ζεύγος κατάστασης και ενέργειας στα επεισόδια.	95

ΠΙΝΑΚΑΣ ΣΥΝΤΟΜΕΥΣΕΩΝ - ΑΚΡΩΝΥΜΙΩΝ

APA	American Psychological Association
ΣΕΑΒ	Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών
ΚΟΥ	Κεντρική Ομάδα Υποστήριξης
ΕΥ	Επιστημονικά Υπεύθυνος
CART	Classification And Regression Trees
FN	False Negatives
FP	False Positives
ID3	Iterative Dichotomizer 3
k NN	k Nearest Neighbors
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
RAE	Relative Absolute Error
RMSE	Root Mean Squared Error
RRSE	Root Relative Squared Error
TN	True Negatives
TP	True Positives

ΠΡΟΛΟΓΟΣ

Οι σημειώσεις αυτές συνοδεύουν τις διαλέξεις του μαθήματος “Μηχανική Μάθηση” στο πρόγραμμα προπτυχιακών σπουδών του Τμήματος Πληροφορικής στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης. Θα ήθελα να ευχαριστήσω για την βοήθεια τους στην παραγωγή αυτών των σημειώσεων τους συνεργάτες μου Αθανάσιο Λαγόπουλο, Ιώαννη Μολλά και Νικόλαο Μυλωνά.

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο:

Θα αποκτήσουμε μια αρχική ιδέα αναφορικά με την επιστημονική περιοχή της μηχανικής μάθησης, τονίζοντας τη χρησιμότητά της, αλλά και τις αδυναμίες της. Επιπρόσθετα θα αναφερθούμε στις διαφορετικές κατηγορίες τεχνικών μηχανικής μάθησης, αναλύοντας σε περισσότερο βάθος την κατηγορία της μάθησης με επίβλεψη.

1.1 Εισαγωγή στη Μηχανική Μάθηση

Η τεχνολογία της μηχανικής μάθησης βρίσκεται ολόγυρά μας. Τη συναντάμε στην καθημερινότητά μας, πολλές φορές, χωρίς να το αντιλαμβανόμαστε. Είναι εκεί όταν το πρώι βλέπουμε στον παγκόσμιο ιστό τι καιρό θα κάνει κατά τη διάρκεια της ημέρας για να ντυθούμε κατάλληλα, όταν γράφουμε μηνύματα στο κινητό μας και εκείνο μας βοηθάει να ολοκληρώσουμε τις λέξεις και προτάσεις μας, όταν διαβάζουμε την ηλεκτρονική μας αλληλογραφία λαμβάνοντας ελάχιστα ανεπιθύμητα μηνύματα, καθώς και όταν επισκεπτόμαστε ένα ηλεκτρονικό κατάστημα και βλέπουμε προτάσεις σχετικών προϊόντων.

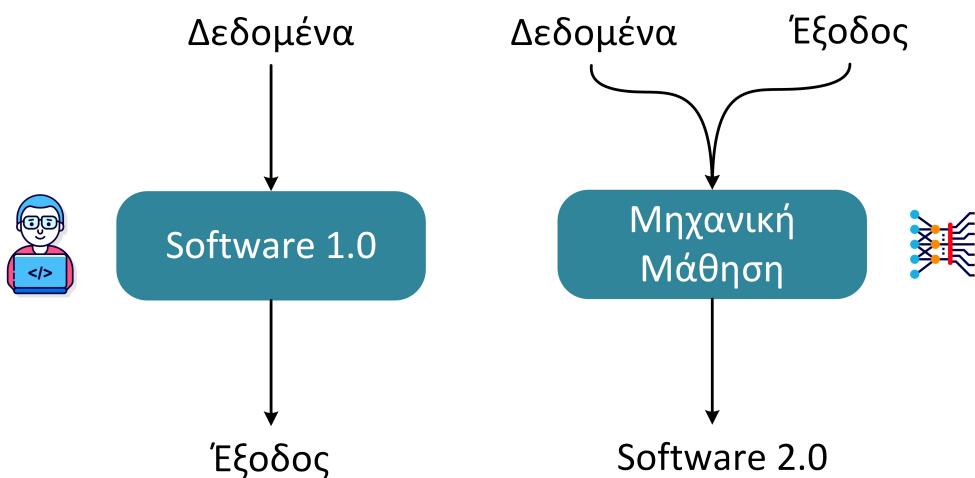
Παρά την ιστορία δεκαετιών της μηχανικής μάθησης, αυτή ήρθε στο προσκήνιο μετά το 2010 και γνωρίζει ραγδαία ανάπτυξη από τότε, με τις επιχειρήσεις να υιοθετούν πολύ έντονα τις μεθόδους της από το 2016 και μετά. Οι δύο κύριοι λόγοι που επέφεραν την εξάπλωση της μηχανικής μάθησης είναι η εκθετική αύξηση του όγκου των δεδομένων και η ραγδαία ανάπτυξη της τεχνολογίας επεξεργασίας μεγάλου όγκου δεδομένων (κάρτες γραφικών, υπολογιστικά πλέγματα). Η τεράστια ποσότητα δεδομένων που καταγράφεται και αποθηκεύεται (διαδίκτυο, έξυπνα τηλέφωνα, συσκευές με αισθητήρες συνδεδεμένες στο διαδίκτυο των αντικειμένων, κ.α.) μας επιτρέπει να εφαρμόσουμε τεχνικές μηχανικής μάθησης σε ένα μεγάλο εύρος από τομείς όπως η υγεία, η οικονομία, η ψυχαγωγία, η γεωργία, η ενέργεια και οι επικοινωνίες. Αντίστοιχα, η συνεχώς αυξανόμενη υπολογιστική ισχύς μας επιτρέπει να εφαρμόσουμε αλγορίθμους μηχανικής μάθησης σε ολοένα και πιο πολύπλοκα προβλήματα, όπως η κατανόηση της φυσικής γλώσσας, με ολοένα και μεγαλύτερη επιτυχία.

1.1.1 Τι είναι η μηχανική μάθηση;

Τι είναι όμως η μηχανική μάθηση; Ο όρος μηχανική μάθηση χρησιμοποιήθηκε για πρώτη φορά από τον Arthur Samuel το 1959, στο πλαίσιο της πρωτοποριακής, για την εποχή, έρευνάς του με στόχο την ανάπτυξη προγράμματος υπολογιστή με ικανότητα να παίζει το παιχνίδι ντάμα. Ο Samuel όρισε την μηχανική μάθηση ως εξής: «*η μηχανική μάθηση είναι μία περιοχή της επιστήμης των υπολογιστών, η οποία δίνει στους υπολογιστές την ικανότητα να μαθαίνουν χωρίς να έχουν προγραμματιστεί ρητά*».

Πολύ αργότερα, το 1997, ο Tom Mitchell όρισε την μηχανική μάθηση έτσι ώστε να συμπεριλαμβάνει κάθε πρόγραμμα υπολογιστή που βελτιώνει την απόδοση του πάνω σε μία συγκεκριμένη εργασία μέσω της εμπειρίας. Συγκεκριμένα έδωσε τον εξής ορισμό: «*Ένα πρόγραμμα υπολογιστή μαθαίνει από εμπειρία E ως προς κάποια κλάση εργασιών T και ενός μέτρου επίδοσης P, αν η επίδοσή του στις εργασίες της κλάσης T, όπως αποτιμάται από το μέτρο P, βελτιώνεται με την εμπειρία E*». Αναλογιστείτε έναν υπολογιστή που μαθαίνει να φίλτραρει την ανεπιθύμητη ηλεκτρονική αλληλογραφία σας. Σε αυτήν την περίπτωση η κλάση εργασιών, T, θα μπορούσε να είναι ο χαρακτηρισμός ενός email ως επιθυμητό ή ανεπιθύμητο, το μέτρο επίδοσης, P, θα μπορούσε να είναι το ποσοστό ενός συγκεκριμένου συνόλου από εισερχόμενα email που ο υπολογιστής χαρακτηρίζει σωστά ως επιθυμητά ή ανεπιθύμητα και η εμπειρία, E, θα μπορούσε να είναι ένα πλήθος από email που έχετε λάβει στο παρελθόν και τα οποία έχετε ήδη χαρακτηρίσει ως επιθυμητά ή ανεπιθύμητα.

Μια άλλη ενδιαφέρουσα οπτική γωνία του τι είναι η μηχανική μάθηση προσέφερε το 2017 ο Andrej Karpathy. Αναφερόμενος πιο συγκεκριμένα σε μια οικογένεια αλγορίθμων μηχανικής μάθησης, τα νευρωνικά δίκτυα, τα αποκάλεσε ως την επόμενη γενιά της τεχνολογίας λογισμικού, την οποία ονόμασε Software 2.0¹. Το παραδοσιακό λογισμικό (Software 1.0), το οποίο δέχεται ως είσοδο ένα σύνολο από δεδομένα και παράγει μια έξοδο, αναπτύσσεται από ανθρώπους (προγραμματιστές). Το Software 2.0 αναπτύσσεται από αλγορίθμους μηχανικής μάθησης, οι οποίοι δέχονται ως είσοδο τα δεδομένα και την έξοδο που τους αντιστοιχεί, και δίνουν στην έξοδο το λογισμικό (Εικόνα 1.1).



Σχήμα 1.1: Το παραδοσιακό λογισμικό (Software 1.0) αναπτύσσεται από έναν ή παραπάνω προγραμματιστές. Το Software 1.0 δέχεται ως είσοδο δεδομένα και παράγει τη αντίστοιχη έξοδο. Η επόμενη γενιά λογισμικού (Software 2.0) αναπτύσσεται από έναν αλγορίθμιμο μηχανικής μάθησης, στον οποίο δίνουμε ως είσοδο τα δεδομένα και την επιθυμητή έξοδο του λογισμικού.

Η μηχανική μάθηση μπορεί επίσης να θεωρηθεί ως ένα πρόβλημα **βελτιστοποίησης** (optimization). Σύμφωνα με αυτήν την θεώρηση, ένας αλγόριθμος μηχανικής μάθησης εκτελεί αναζήτηση σε έναν **χώρο υποθέσεων** (hypothesis space), H , για την εύρεση της **υπόθεσης** (hypothesis), $h \in H$, που ταιριάζει καλύτερα με τα δεδομένα τα οποία έχουμε στη διάθεση μας, καθώς και με ενδεχόμενη πρότερη γνώση σχετικά με το

¹<https://medium.com/@karpathy/software-2-0-a64152b37c35>

πρόβλημα που θέλουμε να λύσουμε. Στο πλαίσιο αυτό, κάθε αλγόριθμος μηχανικής μάθησης μπορεί να μελετήθει με βάση τρεις κύριους άξονες: α) τη μορφή **αναπαράστασης** (representation) των υποθέσεων, η οποία καθορίζει τον χώρο μέσα στον οποίο εκτελεί την αναζήτηση ο αλγόριθμος, β) τη συνάρτηση **αξιολόγησης** (evaluation) των υποθέσεων, μέσω της οποίας αποτιμά πόσο καλά ταιριάζουν οι υποθέσεις στα δεδομένα ή/και την πρότερη γνώση, και γ) τη μέθοδο **αναζήτησης** (search) για την υπόθεση με την βέλτιστη αξιολόγηση μέσα στον χώρο των υποθέσεων.

1.1.2 Γιατί να χρησιμοποιήσει κάποιος μηχανική μάθηση;

Υπάρχουν πολύπλοκα προβλήματα, όπως η οπτική αναγνώριση χαρακτήρων, η κατανόηση του περιεχομένου εικόνων, η αναγνώριση ομιλίας και η αυτόματη μετάφραση, για τα οποία η ανάπτυξη λογισμικού ικανοποιητικής επίδοσης με τον παραδοσιακό τρόπο είναι ανέφικτη. Σε αυτές τις περιπτώσεις η χρήση της μηχανικής μάθησης είναι μονόδρομος και οδηγεί συνήθως σε ταχύτερη επεξεργασία των δεδομένων σε σχέση με τον άνθρωπο, και ορισμένες φορές ακόμα και σε ποιοτικότερα αποτελέσματα.

Σε άλλα προβλήματα, όπως το φιλτράρισμα της ανεπιθύμητης αλληλογραφίας, η αποτίμηση του ρίσκου διαδικτυακών οικονομικών συναλλαγών και η εξαγωγή πληροφορίας από κείμενα, η ανάπτυξη παραδοσιακού λογισμικού είναι εφικτή. Προϋποθέτει όμως την κωδικοποίηση της γνώσης των αντίστοιχων ειδικών του προβλήματος, συνήθως στη μορφή κανόνων. Η γνώση αυτή δεν είναι εύκολο να αποκτηθεί, μπορεί να είναι υποκειμενική και απαιτεί αρκετό πειραματισμό και διορθωτικές κινήσεις προκειμένου να δουλέψει σωστά. Σε αυτές τις περιπτώσεις, η χρήση της μηχανικής μάθησης μπορεί να απαιτήσει τελικά λιγότερο κόπο, ο οποίος θα αφορά κυρίως στον χαρακτηρισμό των δεδομένων, και να οδηγήσει σε ποιοτικότερες και πιο αντικειμενικές λύσεις.

Επιπλέον, υπάρχουν πολλές εφαρμογές, στις οποίες δεν αρκεί μόνο η δημιουργία ενός ευφυούς συστήματος, αλλά απαιτείται η συνεχής συντήρηση και ενημέρωσή του ώστε να προσαρμόζεται στα νέα δεδομένα με την πάροδο του χρόνου. Τέτοιες εφαρμογές ανακύπτουν για παράδειγμα στον τομέα της ασφάλειας, όπου οι κακόβουλοι χρήστες διαρκώς δοκιμάζουν νέους τύπους επιθέσεων, καθώς και στον τομέα της κατανόησης επιστημονικού περιεχομένου, όπου η γλώσσα και η σημασιολογία των λέξεων και των εννοιών μεταβάλλεται με την πάροδο του χρόνου. Ένα σύστημα μηχανικής μάθησης είναι πολύ εύκολο να προσαρμοστεί σε νέα δεδομένα, σε αντίθεση με ένα παραδοσιακό λογισμικό, στο οποίο θα έπρεπε να γραφούν νέοι κανόνες και να τροποποιηθούν ή διαγραφούν υπάρχοντες κανόνες.

Τέλος, η μηχανική μάθηση μπορεί να εφαρμοστεί σε βάσεις δεδομένων με στόχο την ανακάλυψη νέας γνώσης. Συστήματα μηχανικής μάθησης έχουν χρησιμοποιηθεί στην Ιατρική για την ανακάλυψη παραγόντων κινδύνου που σχετίζονται με συγκεκριμένες ασθένειες, αλλά και στις επιχειρήσεις για την ανακάλυψη των αγοραστικών συνηθειών των πελατών, καθώς και των λόγων που οδηγούν στην επιτυχία ενός προϊόντος.

1.1.3 Γιατί να μην χρησιμοποιήσει κάποιος μηχανική μάθηση;

Είναι σημαντικό να τονίσουμε ότι η μηχανική μάθηση έχει συγκεκριμένες αδυναμίες που θα πρέπει κανείς να γνωρίζει. Καταρχάς, επειδή η γνώση ενός συστήματος μηχανικής μάθησης προέρχεται από τα δεδομένα, το σύστημα αναπόφευκτα θα αντικατοπτρίζει τα όποια προβλήματα απαντώνται στα αντίστοιχα δεδομένα, όπως ανθρώπινες προκαταλήψεις. Το 2015, το σύστημα κατανόησης φωτογραφιών της Google αναγνώρισε δύο μαύρους ανθρώπους που εμφανίζονταν σε μια φωτογραφία ως χιμπατζήδες². Το 2018, ανακαλύφθηκε πως η έκδοση του συστήματος Watson της IBM για τον τομέα της Ογκολογίας δίνει ακατάλληλες συστάσεις θεραπείας. Η κύρια αιτία ήταν ότι είχε τροφοδοτηθεί με συνθετικά δεδομένα, τα οποία δεν αντιστοιχούσαν σε πραγματικούς ασθενείς³. Το 2019, η πιστωτική κάρτα της Apple βρέθηκε να μεροληπτεί εναντίον των γυναικών, ορίζοντας ως και 20 φορές μικρότερο πιστωτικό όριο σε γυναίκες σε σχέση με τους συζύγους τους, παρά

²<https://www.wired.com/story/when-it-comes-to-gorillas-google-photos-remains-blind/>

³<https://www.theverge.com/2018/7/26/17619382/ibms-watson-ai-healthcare-science>

την κοινή τους φορολογική δήλωση⁴. Τα παραδείγματα αυτά φανερώνουν πως η ποιότητα των δεδομένων είναι ένας κρίσιμος παράγοντας για τη δημιουργία δίκαιων και αξιόπιστων συστημάτων μηχανικής μάθησης.

Επιπλέον, ορισμένοι τύποι συστημάτων μηχανικής μάθησης, ιδιαίτερα τα συστήματα που αποτελούνται από βαθιά νευρωνικά δίκτυα, είναι ευαίσθητα σε κακόβουλες επιθέσεις. Για παράδειγμα η ενσωμάτωση συγκεκριμένων χρωματικών προτύπων σε οδικά σήματα, μπορεί να ξεγελάσουν τα συστήματα αναγνώρισης οδικών σημάτων των αυτοκινήτων προκειμένου να προκαλέσουν ανεπιθύμητες καταστάσεις. Ομοίως συγκεκριμένα χρωματικά πρότυπα επάνω σε ρούχα μπορούν να ξεγελάσουν συστήματα αναγνώρισης ανθρώπων.

1.1.4 Κατηγορίες μηχανικής μάθησης

Μπορούμε να κατατάξουμε τους αλγορίθμους μηχανικής μάθησης σε τρεις κύριες κατηγορίες σύμφωνα με τον τύπο της επίβλεψης που επιδέχονται κατά την εκπαίδευση: **μάθηση με επίβλεψη** (supervised learning), **μάθηση χωρίς επίβλεψη** (unsupervised learning), και **ενισχυτική μάθηση** (reinforcement learning). Στην προηγούμενη ενότητα αναφέραμε πως ένας αλγόριθμος μηχανικής μάθησης μπορεί να θεωρηθεί πως εκτελεί αναζήτηση σε έναν χώρο υποθέσεων, προκειμένου να βρει την βέλτιστη υπόθεση. Στη μάθηση με επίβλεψη, οι εναλλακτικές υποθέσεις καλούνται και **μοντέλα** (models), στην μάθηση χωρίς επίβλεψη συχνά καλούνται **πρότυπα** (patterns), ενώ στην ενισχυτική μάθηση καλούνται **πολιτικές** (policies).

1.2 Μάθηση με Επίβλεψη

Η μάθηση με επίβλεψη μελετά το πρόβλημα της εύρεσης μιας άγνωστης συνάρτησης, η οποία καλείται **συνάρτηση στόχος** (target function), βάσει ενός συνόλου από ζευγάρια τιμών της εισόδου και της εξόδου της, το οποίο καλείται **σύνολο εκπαίδευσης** (training set). Η μεταβλητή εξόδου της συνάρτησης στόχος ονομάζεται και **εξαρτημένη μεταβλητή** (dependent variable) ή **μεταβλητή στόχος** (target variable). Οι μεταβλητές εισόδου της συνάρτησης στόχος ονομάζονται και **ανεξάρτητες μεταβλητές** (independent variables), **ιδιότητες** (attributes) ή **χαρακτηριστικά** (features). Το σύνολο των διαφορετικών δυνατών τιμών εισόδου της συνάρτησης στόχος, δηλαδή το πεδίο ορισμού της, ονομάζεται **χώρος των περιπτώσεων** (instance space), ενώ ένας συγκεκριμένο μέλος του συνόλου αυτού ονομάζεται **περίπτωση** (instance). Το σύνολο εκπαίδευσης περιλαμβάνει ένα υποσύνολο του χώρου των περιπτώσεων, για το οποίο η τιμή της μεταβλητής εξόδου είναι γνωστή. Τα μέλη του συνόλου εκπαίδευσης ονομάζονται **παραδείγματα εκπαίδευσης** (training examples) ή απλούστερα **παραδείγματα** (examples). Επειδή τις περισσότερες φορές οι αλγόριθμοι μηχανικής μάθησης δεν μπορούν να προσδιορίσουν επακριβώς την συνάρτηση στόχο, παρά μόνο να την προσεγγίσουν, η μάθηση με επίβλεψη καλείται και **προσέγγιση συνάρτησης** (function approximation).

Θα συμβολίζουμε το χώρο των περιπτώσεων ως \mathcal{X} και ένα σύνολο εκπαίδευσης ως $D = \{(\mathbf{x}^{(i)}, y^{(i)}) \mid i = 1 \dots m\}$, όπου m το πλήθος των παραδειγμάτων εκπαίδευσης, $y^{(i)}$ η τιμή της μεταβλητής εξόδου για το i -οστό παράδειγμα εκπαίδευσης και $\mathbf{x}^{(i)} = [x_1^{(i)}, \dots, x_n^{(i)}]$ οι αντίστοιχες τιμές των n μεταβλητών εισόδου. Η Εικόνα 1.2 παρουσιάζει ένα παραδειγμα συνόλου εκπαίδευσης, όπου η μεταβλητή εξόδου είναι η τιμή σε ευρώ ενός ακινήτου, και οι μεταβλητές εισόδου είναι η επιφάνεια σε τετραγωνικά μέτρα, ο αριθμός των δωματίων, ο αριθμός των ορόφων και η ηλικία του ακινήτου.

Η πιο απλή λύση σε ένα πρόβλημα μάθησης με επίβλεψη είναι η **μάθηση με απομνημόνευση** (rote learning). Ο αλγόριθμος μάθησης απλά αποθηκεύει το σύνολο εκπαίδευσης και μπορεί να δώσει την τιμή της μεταβλητής εξόδου για μια νέα περίπτωση, μόνο αν αυτή η ίδια είναι ήδη αποθηκευμένη. Είναι ευνόητο ότι η μάθηση με απομνημόνευση δεν είναι αποτελεσματική όταν το σύνολο των δεδομένων περιλαμβάνει ένα μικρό υποσύνολο του συνόλου των περιπτώσεων, όπως συμβαίνει στη πράξη. Απαιτείται επομένως η προσέγγιση της συνάρτησης στόχος μέσω μιας διαδικασίας επαγωγής. Εξετάζοντας μόνο ένα μέρος του χώρου των περιπτώσεων (το σύνολο εκπαίδευσης), ο αλγόριθμος μάθησης καλείται να επάγει μια συνάρτηση που θα ισχύει για όλο το χώρο των περιπτώσεων. Αυτή η λύση στο πρόβλημα της μάθησης με επίβλεψη καλείται **επαγωγική**

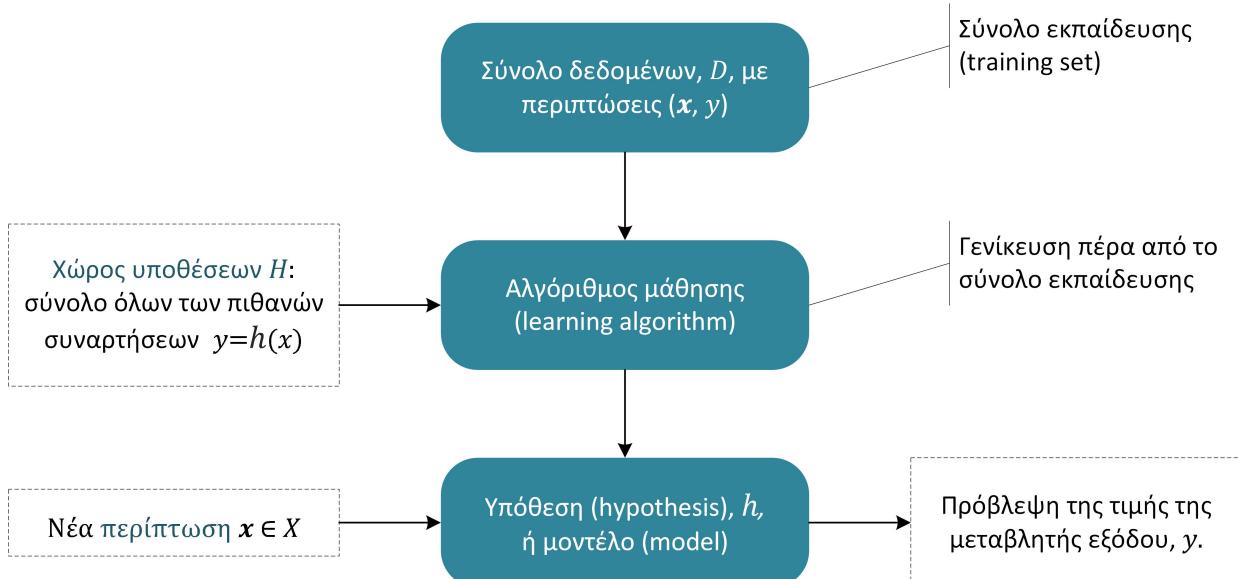
⁴<https://thenextweb.com/news/apple-cards-algorithm-under-investigation-for-sexist-credit-checks>

The diagram shows a table representing a dataset for house price prediction. The columns are labeled: Επιφάνεια (m^2), Δωμάτια, Όροφοι, Ηλικία, and Τιμή (€). There are four rows of data. The second row is highlighted with a red border and circled '40' in the Όροφοι column, indicating it is a training example. A bracket on the left indicates $m=3$, and a bracket at the bottom indicates $n=4$. Two boxes at the top define terms: 'Μεταβλητές εισόδου, ή χαρακτηριστικά ή ανεξάρτητες μεταβλητές' and 'Μεταβλητή εξόδου, ή μεταβλητή στόχος ή εξαρτημένη μεταβλητή'. An arrow points to the circled value '40' with the label 'Παράδειγμα εκπαίδευσης (training example)'. Another arrow points to the circled value with the label 'x_i⁽²⁾'.

Επιφάνεια (m^2)	Δωμάτια	Όροφοι	Ηλικία	Τιμή (€)
195	5	1	45	330.000
130	3	2	40	168.000
142	3	2	30	228.000

Σχήμα 1.2: Παράδειγμα συνόλου εκπαίδευσης.

μάθηση (inductive learning) και στηρίζεται σε μια υπόθεση που είναι γνωστή ως **υπόθεση της επαγωγικής μάθησης** (inductive learning hypothesis): Κάθε συνάρτηση που έχει βρεθεί να προσεγγίζει τη συνάρτηση στόχο καλά για ένα αρκετά μεγάλο σύνολο εκπαίδευσης, θα την προσεγγίζει το ίδιο καλά και για άλλες περιπτώσεις που δεν έχει εξετάσει. Κατά την επαγωγική μάθηση, οι αλγόριθμοι μηχανικής μάθησης με επίβλεψη εξετάζουν διάφορες εναλλακτικές συναρτήσεις, προκειμένου να βρουν εκείνη που προσεγγίζει καλύτερα τη συνάρτηση στόχο στο σύνολο εκπαίδευσης. Στην Εικόνα 1.3 βλέπουμε σχηματικά τη διαδικασία της επαγωγικής μάθησης.

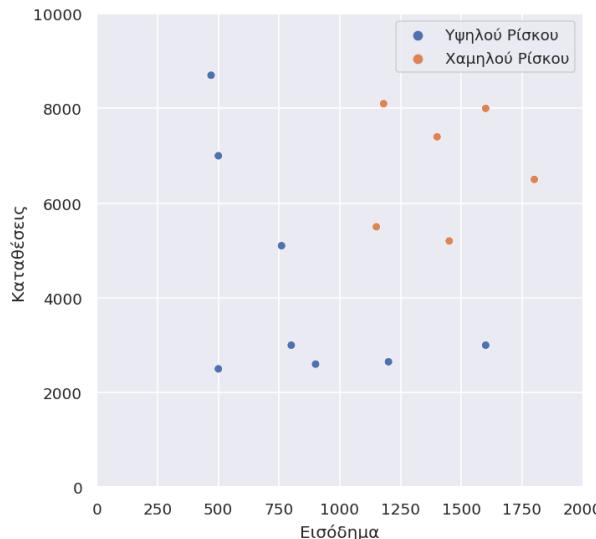


Σχήμα 1.3: Η διαδικασία της επαγωγικής μάθησης.

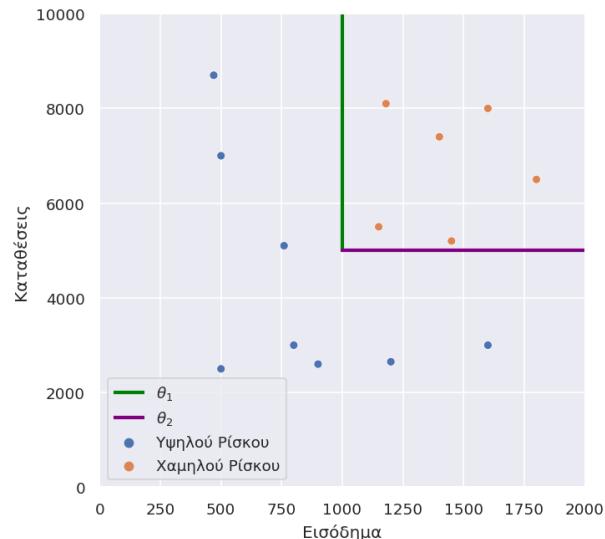
1.2.1 Ταξινόμηση και παλινδρόμηση

Στη μάθηση με επίβλεψη διακρίνονται δύο είδη προβλημάτων, τα προβλήματα **ταξινόμησης** (classification) και τα προβλήματα **παλινδρόμησης** (regression). Η ταξινόμηση αφορά στην δημιουργία μοντέλων πρόβλεψης διακριτών τιμών, ενώ η παλινδρόμηση αφορά στην δημιουργία μοντέλων πρόβλεψης συνεχών τιμών. Όταν η ταξινόμηση αφορά σε 2 μόνο κλάσεις, τότε καλείται **δυαδική ταξινόμηση** (binary classification), ενώ όταν αφορά σε πάνω από 2 κλάσεις καλείται **ταξινόμηση πολλαπλών κλάσεων** (multi-class classification).

Έστω για παράδειγμα μια τράπεζα, η οποία θέλει να υπολογίζει αν το ρίσκο έγκρισης δανείου ενός πελάτη της είναι υψηλό ή χαμηλό σύμφωνα με κάποια οικονομικά χαρακτηριστικά του πελάτη. Για να αυτοματοποιήσει την διαδικασία θέλει να εκπαιδεύσει ένα μοντέλο χρησιμοποιώντας μηχανική μάθηση. Ως μεταβλητές εισόδου ορίζουμε το **εισόδημα** και τις **καταθέσεις** του πελάτη, ενώ ως μεταβλητή εξόδου το ρίσκο έγκρισης δανείου, που μπορεί να πάρει μία από τις δύο διακριτές τιμές, **υψηλό ή χαμηλό** ρίσκο. Το πρόβλημα αυτό αποτελεί πρόβλημα ταξινόμησης καθώς επιθυμούμε την πρόβλεψη διακριτής τιμής. Η τράπεζα έχει ήδη αξιολογήσει κάποιους από τους περασμένους πελάτες στους οποίους έχει δώσει δάνειο ως πελάτες χαμηλού ή υψηλού ρίσκου.



(α) Δεδομένα.



(β) Δεδομένα και μοντέλο.

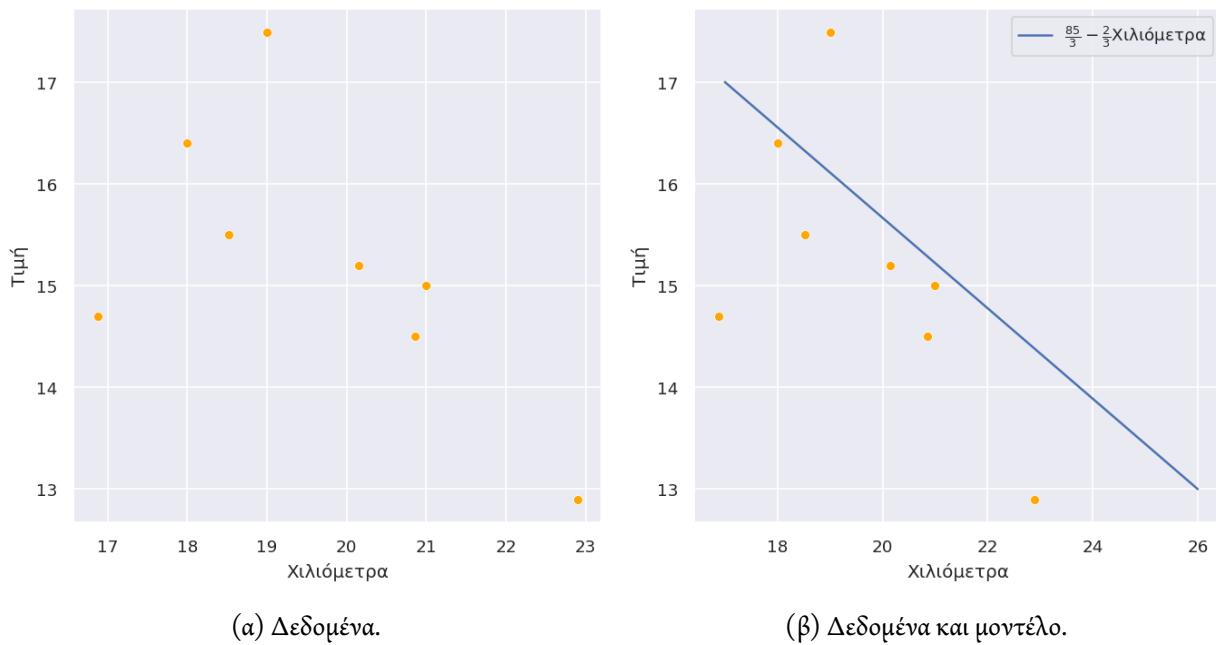
Σχήμα 1.4: Ταξινόμηση πελατών μιας τράπεζας σε χαμηλού ή υψηλού ρίσκου.

Το σχήμα 1.4α αναπαριστά τους πελάτες αυτούς χρησιμοποιώντας τα χαρακτηριστικά που αναφέρθηκαν παραπάνω, δηλαδή το εισόδημα και τις καταθέσεις κάθε πελάτη. Τα παραδείγματα αυτά αποτελούν το σύνολο εκπαίδευσης. Στην περίπτωση αυτή μπορούμε εύκολα να βρούμε ένα μοντέλο, το οποίο να προσεγγίζει απόλυτα τα δεδομένα εκπαίδευσης. Παρατηρούμε πως οι πελάτες χαμηλού ρίσκου είναι αυτοί που έχουν αρκετά υψηλό εισόδημα και μεγάλα ποσά καταθέσεων. Επομένως, μπορούμε να δημιουργήσουμε το παρακάτω μοντέλο με το οποίο θα αξιολογείται κάθε νέος πελάτης, το οποίο απεικονίζεται στο σχήμα 1.4β.

$$\begin{aligned} \text{ΑΝ } \text{εισόδημα} > \theta_1 \text{ ΚΑΙ } \text{αποταμιευμένο ποσό} > \theta_2 \\ \text{TOTE } \text{ρίσκο} = \text{χαμηλό } \textbf{ΑΛΛΙΩΣ } \text{ρίσκο} = \text{υψηλό} \end{aligned}$$

Ως ένα δεύτερο παράδειγμα, ας αναλογιστούμε μία μάντρα αυτοκινήτων, η οποία επιθυμεί να προβλέπει την τιμή πώλησης ενός μεταχειρισμένου αυτοκίνητου προκειμένου να ενημερώνει άμεσα τους πωλητές. Για το λόγο αυτό χρησιμοποιεί ένα μοντέλο μάθησης που έχει εκπαιδευτεί σε δεδομένα παλαιότερων πωλήσεων και προβλέπει την τιμή πώλησης (μεταβλητή εξόδου) ενός αυτοκίνητου. Ως μεταβλητή εισόδου έχει οριστεί ο αριθμός των χιλιομέτρων που έχει διανύσει αυτοκίνητο. Το πρόβλημα αυτό αποτελεί πρόβλημα παλινδρόμησης καθώς επιθυμούμε την πρόβλεψη συνεχούς τιμής.

Στο σχήμα 1.5α, απεικονίζονται οι τιμές και τα χιλιόμετρα ορισμένων Toyota Yaris του 2019 όπως βρέθηκαν στον ιστότοπο car.gr τον Απρίλιο του 2022. Ο οριζόντιος άξονας αναπαριστά την μεταβλητή εισόδου, δηλαδή τα χιλιόμετρα που έχει διανύσει ένα αυτοκίνητο (σε χιλιάδες), και ο κάθετος άξονας την μεταβλητή εξόδου, δηλαδή την τιμή πώλησης του αυτοκίνητου (σε χιλιάδες ευρώ). Τα παραδείγματα αυτά αποτελούν το σύνολο εκπαίδευσης. Στην περίπτωση αυτή μπορούμε εύκολα να βρούμε μια συνάρτηση, η οποία να προσεγγίζει τα δεδομένα εκπαίδευσης. Παρατηρούμε πως όσο αυξάνεται ο αριθμός των χιλιομέτρων ενός αυτοκίνητου



Σχήμα 1.5: Εκτίμηση τιμής μεταχειρισμένων αυτοκινήτων.

τόσο μειώνεται η τιμή του. Η μείωση αυτή είναι σχεδόν γραμμική. Επομένως, θα μπορούσαμε να πούμε πως μια ευθεία θα μπορούσε να προσεγγίσει αρκετά καλά τα δεδομένα εκπαίδευσης. Ένα τέτοιο γραμμικό μοντέλο που θα μπορούσε να προκύψει είναι το παρακάτω, το οποίο απεικονίζεται στο σχήμα 1.5β.

$$\text{Τιμή πώλησης} = \frac{85}{3} - \frac{2}{3} \text{ χιλιόμετρα}$$

1.2.2 Οικογένειες αλγορίθμων

Οι σημαντικότεροι αλγόριθμοι μάθησης με επίβλεψη μπορούν να χωριστούν στις παρακάτω οικογένειες αλγορίθμων, κυρίως με βάση την αναπαράσταση που υιοθετούν για τις υποθέσεις: **γραμμικά μοντέλα** (linear models), **δεντρικά μοντέλα** (tree models), **μάθηση κανόνων** (rule learning), **μάθηση κατά περίπτωση** (instance based learning), **πιθανοτικά μοντέλα** (probabilistic models), **νευρωνικά δίκτυα** (neural networks), **μηχανές διανυσμάτων υποστήριξης** (support vector machines) και **ομάδες μοντέλων** (ensemble learning). Οι περισσότεροι αλγόριθμοι μάθησης με επίβλεψη μπορούν να χρησιμοποιηθούν τόσο για ταξινόμηση όσο και για παλινδρόμηση.

1.2.3 Συναρτήσεις απώλειας και κόστους

Δοθέντος ενός παραδείγματος (x, y) και ενός μοντέλου μηχανικής μάθησης, h , μια **συνάρτηση απώλειας** (loss function) $L(h(x), y)$ υπολογίζει τη διαφορά μεταξύ της μεταβλητής εξόδου y του παραδείγματος και της πρόβλεψης του μοντέλου, $h(x)$. Δοθέντος ενός συνόλου παραδειγμάτων εκπαίδευσης $D = \{(x^{(i)}, y^{(i)}) \mid i = 1 \dots m\}$ και ενός μοντέλου μηχανικής μάθησης, h , μια **συνάρτηση κόστους** (cost function) $L(h, D)$ υπολογίζει την μέση απώλεια του μοντέλου στα παραδείγματα:

$$L(h, D) = \frac{1}{m} \sum_{i=1}^m L(h(x^{(i)}), y^{(i)}) \quad (1.1)$$

Οι συναρτήσεις κόστους χρησιμοποιούνται από κάποιες οικογένειες αλγορίθμων, όπως τα γραμμικά μοντέλα και τα νευρωνικά δίκτυα, για να αποτιμήσουν πόσο καλά ταιριάζει ένα μοντέλο στα δεδομένα εκπαίδευσης και αντιστοιχούν στον δεύτερο από τους τρεις κύριους άξονες ενός αλγορίθμου μηχανικής μάθησης που αναφέραμε στην ενότητα 1.1.1, τις συναρτήσεις αξιολόγησης.

1.3 Περαιτέρω Μελέτη

Ο Arthur Samuel δημοσίευσε την πρωτοποριακή του μελέτη επάνω στην μηχανική μάθηση στο πλαίσιο του παιχνιδιού της ντάμας το 1959 [1].

Για την οπτική γωνία της μηχανικής μάθησης ως αναζήτηση σε έναν χώρο υποθέσεων μπορεί κανείς να διαβάσει περισσότερα στο [2], ενώ για τους τρεις άξονες με βάση τους οποίους μπορούμε να εξετάσουμε κάθε αλγόριθμο μηχανικής μάθησης στο [3], καθώς και στο εκλαϊκευμένο βιβλίο μηχανικής μάθησης [4].

Βιβλιογραφία

- [1] A. L. Samuel. "Some Studies in Machine Learning Using the Game of Checkers". Στο: *IBM Journal of Research and Development* 3.3 (1959), σσ. 210–229. doi: [10.1147/rd.33.0210](https://doi.org/10.1147/rd.33.0210).
- [2] Tom M. Mitchell. *Machine learning, International Edition*. McGraw-Hill Series in Computer Science. McGraw-Hill, 1997. ISBN: 978-0-07-042807-2.
- [3] Pedro Domingos. "A few useful things to know about machine learning". Στο: *Communications of the ACM* 55.10 (Οκτ. 2012), σ. 78. ISSN: 00010782. doi: [10.1145/2347736.2347755](https://doi.org/10.1145/2347736.2347755). URL: <http://dl.acm.org/citation.cfm?doid=2347736.2347755>.
- [4] Pedro Domingos. *The Master Algorithm*. Penguin Books Ltd, 2017, σ. 352. ISBN: 9780141979243.

ΚΕΦΑΛΑΙΟ 2

ΓΡΑΜΜΙΚΑ ΜΟΝΤΕΛΑ

Στο κεφάλαιο αυτό:

Θα μελετήσουμε την οικογένεια των γραμμικών μοντέλων με έμφαση στη βελτιστοποίηση τους με τον αλγόριθμο της επικλινούς καθόδου. Αρχικά, θα παρουσιάσουμε τους αλγορίθμους της γραμμικής παλινδρόμησης και της λογιστικής παλινδρόμησης. Ο πρώτος αλγόριθμος χρησιμοποιείται σε εργασίες παλινδρόμησης, ενώ ο δεύτερος, παρά το γεγονός ότι στο όνομα του περιλαμβάνεται η λέξη παλινδρόμηση, χρησιμοποιείται σε εργασίες ταξινόμησης με δύο κλάσεις. Στη συνέχεια θα συζητήσουμε επεκτάσεις της λογιστικής παλινδρόμησης για πάνω από δύο κλάσεις. Έπειτα, θα συζητήσουμε για μια σημαντική παράμετρο αυτών των αλγορίθμων που ονομάζεται ρυθμός μάθησης, για το πως μπορούμε να επιταχύνουμε τη σύγκλιση τους κλιμακώνοντας τις μεταβλητές εισόδου, καθώς και για μια σημαντική παραλλαγή τους, που τους επιτρέπει να εφαρμοστούν σε τεράστιους όγκους και ροές δεδομένων. Τέλος θα δούμε πως μπορούμε να μάθουμε μη γραμμικές σχέσεις μεταξύ των μεταβλητών εισόδου και της μεταβλητής εξόδου, αλλά και πως μπορούμε να αντιμετωπίσουμε την υπερπροσαρμογή των γραμμικών μοντέλων στα δεδομένα.

Η οικογένεια των γραμμικών μοντέλων έχει τις ρίζες της στη στατιστική. Η μέθοδος των ελαχίστων τετραγώνων για την επίλυση ενός προβλήματος γραμμικής παλινδρόμησης είχε εφαρμοστεί στις αρχές του 19ου αιώνα από τους Legendre και Gauss για τον υπολογισμό της τροχιάς των πλανητών. Στο πλαίσιο της μηχανικής μάθησης, τα εξετάζουμε υπό το πρίσμα της ελαχιστοποίησης της τιμής της συνάρτησης κόστους με τον αλγόριθμο της επικλινούς καθόδου. Το όνομα τους οφείλεται, όπως θα διαπιστώσουμε παρακάτω, στο γεγονός πως η μεταβλητή εξόδου εξαρτάται από έναν γραμμικό συνδυασμό των μεταβλητών εισόδου. Γεωμετρικά, αναπαριστούν μια γραμμή στις δύο διαστάσεις, ένα επίπεδο στις τρεις διαστάσεις και ένα υπερεπίπεδο στις τέσσερις και πάνω διαστάσεις.

2.1 Γραμμική Παλινδρόμηση

Όπως αναφέραμε στην εισαγωγή (Ενότητα 1.1.1), ένας αλγόριθμος μηχανικής μάθησης περιλαμβάνει τρία βασικά συστατικά: μια μορφή αναπαράστασης των υποθέσεων, μια μετρική αξιολόγησης των υποθέσεων σε

Τσουμάκας Γ. (2023). «Εισαγωγή στη Μηχανική Μάθηση».

 Creative Commons Αναφορά Δημιουργού - Μη Εμπορική Χρήση - Παρόμοια Διανομή 4.0

σχέση με τα δεδομένα εκπαίδευσης και έναν αλγόριθμο αναζήτησης της βέλτιστης υπόθεσης. Στις ακόλουθες υποενότητες θα αναλύσουμε το κάθε ένα από αυτά τα συστατικά για τον αλγόριθμο της γραμμικής παλινδρόμησης (linear regression). Τέλος, στην υποενότητα 2.1.4 θα αναφερθούμε σε έναν εναλλακτικό τρόπο αναλυτικού υπολογισμού της βέλτιστης υπόθεσης.

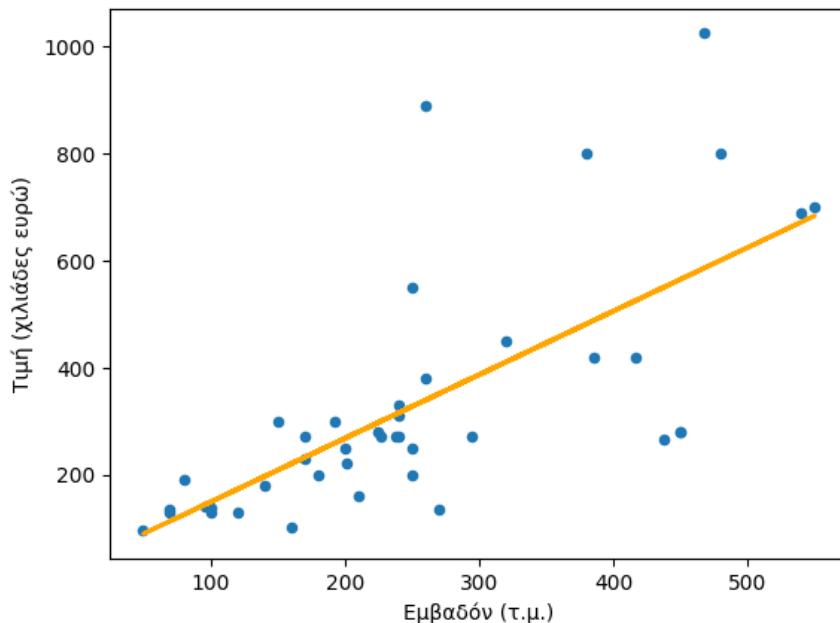
2.1.1 Αναπαράσταση υποθέσεων

Στη γραμμική παλινδρόμηση αναπαριστούμε μια υπόθεση, h , ως εξής:

$$h_{\theta}(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n \quad (2.1)$$

Οι σταθερές $\theta_0, \theta_1, \dots, \theta_n$ ονομάζονται παράμετροι. Συγκεκριμένα, η παράμετρος θ_0 ονομάζεται **προκατάληψη ή αποτέμνουσα** (bias ή intercept), ενώ οι παράμετροι $\theta_1, \dots, \theta_n$ ονομάζονται **βάρη** (weights).

Έστω για παράδειγμα, η υπόθεση $h_{\theta}(\mathbf{x}) = 3 + 2x_1 + 5x_2$. Η υπόθεση αυτή αφορά σε ένα πρόβλημα παλινδρόμησης με δύο μεταβλητές εισόδου και περιλαμβάνει τρεις παραμέτρους: την αποτέμνουσα $\theta_0 = 3$ και τα βάρη $\theta_1 = 2$ και $\theta_2 = 5$. Το σχήμα 2.1 παρουσιάζει το γραμμικό μοντέλο $h_{\theta}(\mathbf{x}) = 30 + 1.19x_1$, το οποίο έχει προκύψει από την εφαρμογή της γραμμικής παλινδρόμησης σε ένα σύνολο δεδομένων που περιλαμβάνει το εμβαδόν και την τιμή πώλησης κατοικιών στον Τρίλοφο Θεσσαλονίκης τον Μάρτιο του 2024.



Σχήμα 2.1: Γραμμικό μοντέλο εκτίμησης αξίας κατοικιών με βάση το εμβαδόν τους.

Για την οικονομία της παρουσίασης, μπορούμε να ορίσουμε μια τεχνητή μεταβλητή εισόδου x_0 , η οποία έχει πάντα τιμή 1 σε κάθε παράδειγμα του συνόλου εκπαίδευσης και σε κάθε νέα περίπτωση. Με αυτόν τον τρόπο, η υπόθεση μπορεί να αναπαρασταθεί συνοπτικά ως το εσωτερικό γινόμενο των διανυσμάτων θ και \mathbf{x} :

$$h_{\theta}(\mathbf{x}) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n = \boldsymbol{\theta} \cdot \mathbf{x} \quad (2.2)$$

2.1.2 Αξιολόγηση υποθέσεων

Ένα μοντέλο γραμμικής παλινδρόμησης αξιολογείται χρησιμοποιώντας ως συνάρτηση κόστους το μισό του μέσου τετραγωνικού σφάλματος (mean squared error):

$$J(\theta_0, \dots, \theta_n) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 \quad (2.3)$$

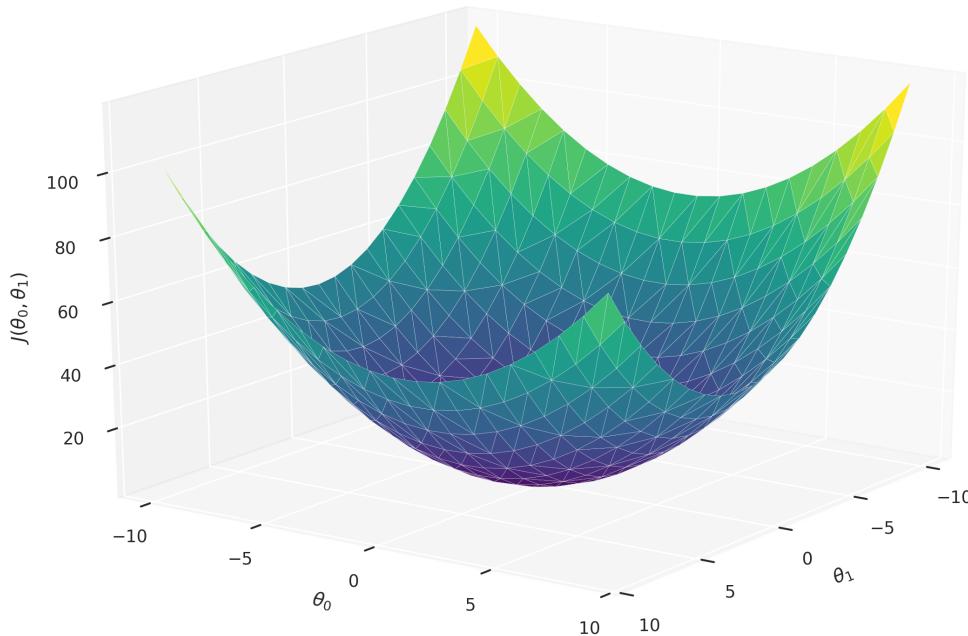
Έστω για παράδειγμα μια πολύ απλή εργασία παλινδρόμησης με μία μόνο μεταβλητή εισόδου και ένα εξαιρετικά μικρό σύνολο δεδομένων τριών παραδειγμάτων εκπαίδευσης που φαίνεται στον Πίνακα 2.1. Έστω επίσης ένα μοντέλο γραμμικής παλινδρόμησης για την συγκεκριμένη εργασία, στο οποίο $\theta_0 = 2$ και $\theta_1 = 1$, δηλαδή $h_{\theta}(x) = 2 + x_1$. Το κόστος αυτού του μοντέλου υπολογίζεται ως εξής:

$$J(\theta_0, \theta_1) = J(2, 1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 = \frac{1}{6} [1^2 + 0^2 + (-1)^2] = \frac{1}{3}$$

x_1	y
1	2
2	4
3	6

Πίνακας 2.1: Ένα εξαιρετικά μικρό σύνολο δεδομένων εκπαίδευσης για μια εργασία παλινδρόμησης με μία μεταβλητή εισόδου.

Μια ενδιαφέρουσα ιδιότητα της συνάρτησης κόστους της γραμμικής παλινδρόμησης είναι ότι είναι κυρτή. Αυτό σημαίνει πως έχει ένα καθολικό ελάχιστο. Πιο συγκεκριμένα η συνάρτηση κόστους έχει παραβολική μορφή, επειδή εκφράζει ένα πολυώνυμο δευτέρου βαθμού ως προς την κάθε παράμετρο. Το Σχήμα 2.2 παρουσιάζει την συνάρτηση κόστους σε μια εργασία παλινδρόμησης με μία μεταβλητή εισόδου και επομένως με δύο παραμέτρους.



Σχήμα 2.2: Η συνάρτηση κόστους σε μια εργασία παλινδρόμησης με μία μεταβλητή εισόδου.

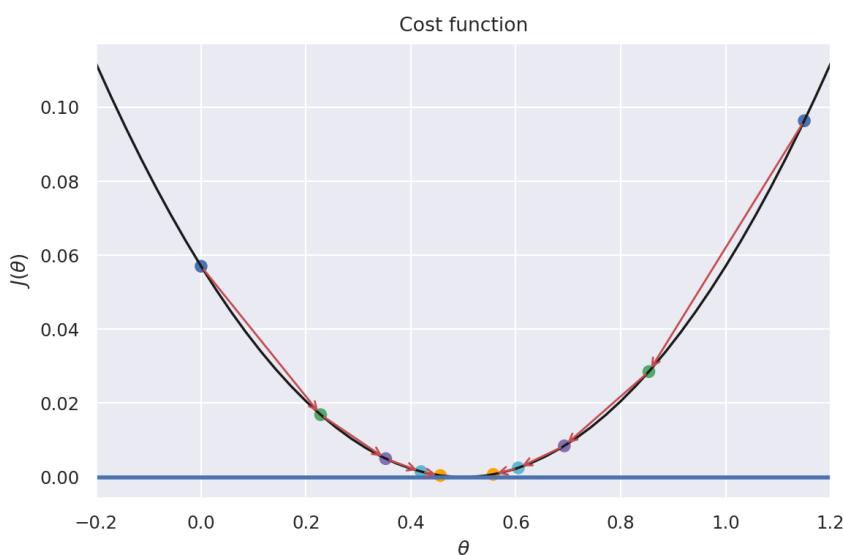
2.1.3 Αναζήτηση της βέλτιστης υπόθεσης

Ο αλγόριθμος με τον οποίο θα κάνουμε αναζήτηση για την βέλτιστη υπόθεση που ελαχιστοποιεί το κόστος ονομάζεται **επικλινής κάθοδος** (gradient descent). Ο αλγόριθμος αυτός είναι ένας επαναληπτικός αλγόριθμος βέλτιστοποίησης για την εύρεση του ελάχιστου μιας συνάρτησης. Στην περίπτωση ενός μοντέλου γραμμικής παλινδρόμησης, όπως το ορίσαμε στην Εξίσωση 2.2, θέλουμε να βρούμε τις παραμέτρους $\theta_0, \dots, \theta_n$, οι οποίες ελαχιστοποιούν την συνάρτηση κόστους $J(\theta_0, \dots, \theta_n)$ όπως την ορίσαμε στην Εξίσωση 2.3.

Ο αλγόριθμος ξεκινά αναθέτοντας τυχαίες τιμές στις παραμέτρους και στη συνέχεια μεταβάλλει επαναληπτικά την τιμή της κάθε παραμέτρου αντιστρόφως ανάλογα με την κλίση της συνάρτησης κόστους ως προς την παράμετρο αυτή στο σημείο που ορίζουν οι τρέχουσες τιμές των παραμέτρων. Αν η κλίση είναι θετική (αρνητική), τότε η τιμή της παραμέτρου μειώνεται (αυξάνεται), έτσι ώστε να οδηγηθούμε σε χαμηλότερες τιμές της συνάρτησης κόστους. Όσο μεγαλύτερη η κλίση της συνάρτησης κόστους ως προς μια παράμετρο, τόσο περισσότερο μεταβάλλουμε την τρέχουσα τιμή της. Η διαδικασία αυτή συνεχίζεται, είτε μέχρι οι τιμές των παραμέτρων να συγκλίνουν στο καθολικό ελάχιστο της συνάρτησης κόστους, είτε για συγκεκριμένο αριθμό επαναλήψεων.

Η ακριβής ποσότητα κατά την οποία αυξάνουμε ή μειώνουμε την τιμή κάθε παραμέτρου ισούται με το γινόμενο της απόλυτης τιμής της κλίσης της συνάρτησης κόστους ως προς την παράμετρο αυτή με έναν μικρό θετικό αριθμό η , ο οποίος ονομάζεται **ρυθμός μάθησης** (learning rate). Ο ρυθμός μάθησης αποτελεί παράμετρο του αλγορίθμου της γραμμικής παρεμβολής. Τέτοιου τύπου παραμέτρους τις αποκαλούμε συνήθως υπερ-παραμέτρους προς αποφυγή σύγχυσης σε σχέση με τις παραμέτρους, θ , οι οποίες αφορούν στην αναπάρασταση των υποθέσεων.

Το Σχήμα 2.3 δείχνει την παραπάνω διαδικασία, όπου για λόγους απλούστευσης της παρουσίασης στις 2 διαστάσεις εξετάζουμε την συνάρτηση κόστους ως προς μία μόνο παράμετρο θ . Βλέπουμε για παράδειγμα πως στο σημείο όπου η παράμετρος ισούται με 0, η κλίση της συνάρτησης κόστους είναι αρνητική και έτσι αυξάνουμε την τιμή της. Αντίθετα στο σημείο όπου η παράμετρος ισούται με 0.6, η συνάρτηση κόστους έχει θετική κλίση και έτσι μειώνουμε την τιμή της. Παρατηρούμε επίσης πως στο σημείο 0 η αρνητική κλίση είναι πολύ μεγαλύτερη της θετικής κλίσης στο σημείο 0.6, κάτι που αποτυπώνεται και στην διαφορετική ποσότητα κατά την οποία μεταβάλλουμε την τιμή της παραμέτρου στις δύο αυτές περιπτώσεις.



Σχήμα 2.3: Η συνάρτηση κόστους μιας εργασίας παλινδρόμησης ως προς μία μεταβλητή εισόδου.

Την κλίση της συνάρτησης κόστους ως προς μία παράμετρο στο σημείο που ορίζουν οι τρέχουσες τιμές των παραμέτρων μπορούμε να την λάβουμε από την τιμή της μερικής παράγωγου της συνάρτησης κόστους ως προς την παράμετρο αυτή. Ας υπολογίσουμε την μερική παράγωγο της συνάρτησης κόστους ως προς

μια παράμετρο θ_j . Αξιοποιώντας τον κανόνα της άθροισης¹, τον κανόνα της αλυσίδας² και δεδομένου ότι αν $f(x) = x^2$ τότε $f'(x) = 2x$ έχουμε:

$$\begin{aligned} \frac{\partial J}{\partial \theta_j} &= \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^m (h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)})^2 = \frac{1}{2m} \sum_{i=1}^m \frac{\partial}{\partial \theta_j} (h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)})^2 = \\ &= \frac{1}{m} \sum_{i=1}^m (h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)}) \frac{\partial}{\partial \theta_j} (h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)}) = \\ &= \frac{1}{m} \sum_{i=1}^m (h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)}) \frac{\partial}{\partial \theta_j} (\boldsymbol{\theta} \cdot \mathbf{x}^{(i)} - y^{(i)}) = \\ &= \frac{1}{m} \sum_{i=1}^m (h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)}) x_j^{(i)} \end{aligned} \quad (2.4)$$

Επομένως, σύμφωνα με τον αλγόριθμο της επικλινούς καθόδου, οι παράμετροι ενός μοντέλου γραμμικής παλινδρόμησης ενημερώνονται επαναληπτικά ως εξής:

- $\boldsymbol{\theta}' := \boldsymbol{\theta}$
- $\forall j \in \{0, \dots, n\}: \theta_j := \theta_j - \eta \frac{1}{m} \sum_{i=1}^m (h_{\boldsymbol{\theta}'}(\mathbf{x}^{(i)}) - y^{(i)}) x_j^{(i)}$

Προσέξτε πως σε ένα πρώτο βήμα, κρατάμε σε ένα βοηθητικό διάνυσμα $\boldsymbol{\theta}'$ τις προηγούμενες τιμές των παραμέτρων, προκειμένου στη συνέχεια να υπολογίζουμε τις σωστές τιμές των μερικών παραγώγων. Θα μπορύσαμε εναλλακτικά να θεωρήσουμε πως οι ενημερώσεις όλων των παραμέτρων γίνονται ταυτόχρονα.

Χρησιμοποιώντας σημειογραφία γραμμικής άλγεβρας μπορούμε να αναπαραστήσουμε τα δεδομένα με έναν πίνακα και να περιγράψουμε την ενημέρωση των παραμέτρων πιο απλά με μία εξίσωση που περιλαμβάνει πολλαπλασιασμούς πινάκων:

$$\boldsymbol{\theta} := \boldsymbol{\theta} - \eta \frac{1}{m} \mathbf{X}^\top (\mathbf{X} \boldsymbol{\theta} - \mathbf{y}) \quad (2.5)$$

Επιπλέον, με αυτόν τον τρόπο μπορούμε να αξιοποιήσουμε κατάλληλο υλικό (όπως κάρτες γραφικών) και λογισμικό για τον παράλληλο υπολογισμό των νέων τιμών όλων των παραμέτρων.

2.1.4 Κανονική εξίσωση

Οι παράμετροι που ελαχιστοποιούν την συνάρτηση κόστους στη γραμμική παλινδρόμηση, μπορούν να υπολογιστούν αναλυτικά με τη μέθοδο της **κανονικής εξίσωσης** (normal equation). Συγκεκριμένα, για κάθε παράμετρο, θα μπορύσαμε να πάρουμε τη μερική παράγωγο της συνάρτησης κόστους ως προς την παράμετρο αυτή και να την θέσουμε ίση με 0:

$$\frac{1}{m} \mathbf{X}^\top (\mathbf{X} \boldsymbol{\theta} - \mathbf{y}) = \vec{0} \Rightarrow \mathbf{X}^\top \mathbf{X} \boldsymbol{\theta} - \mathbf{X}^\top \mathbf{y} = \vec{0} \Rightarrow \mathbf{X}^\top \mathbf{X} \boldsymbol{\theta} = \mathbf{X}^\top \mathbf{y} \Rightarrow \boldsymbol{\theta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} \quad (2.6)$$

Η κανονική εξίσωση απαιτεί τον υπολογισμό του αντίστροφου του τετραγωνικού πίνακα $\mathbf{X}^\top \mathbf{X}$. Ωστόσο, ο πίνακας αυτός ενδέχεται να μην είναι αντιστρέψιμος. Για τον λόγο αυτό στην πράξη χρησιμοποιείται συνήθως η λύση με τον ψευδό-αντίστροφο πίνακα Moore-Penrose \mathbf{X}^\dagger του πίνακα \mathbf{X} :

$$\boldsymbol{\theta} = \mathbf{X}^\dagger \mathbf{y} \quad (2.7)$$

¹ $f(x) = g(x) + h(x) \Rightarrow f'(x) = g'(x) + h'(x)$

² $f(x) = g(h(x)) \Rightarrow f'(x) = g'(h(x))h'(x)$

2.2 Λογιστική Παλινδρόμηση

Ο αλγόριθμος της **λογιστικής παλινδρόμησης** (logistic regression) χρησιμοποιείται σε εργασίες δυαδικής ταξινόμησης. Θεωρούμε ότι η εξαρτημένη μεταβλητή y μπορεί να πάρει τιμές 0 ή 1, οι οποίες αντιστοιχούν στις δύο διαφορετικές κλάσεις της εργασίας δυαδικής ταξινόμησης.

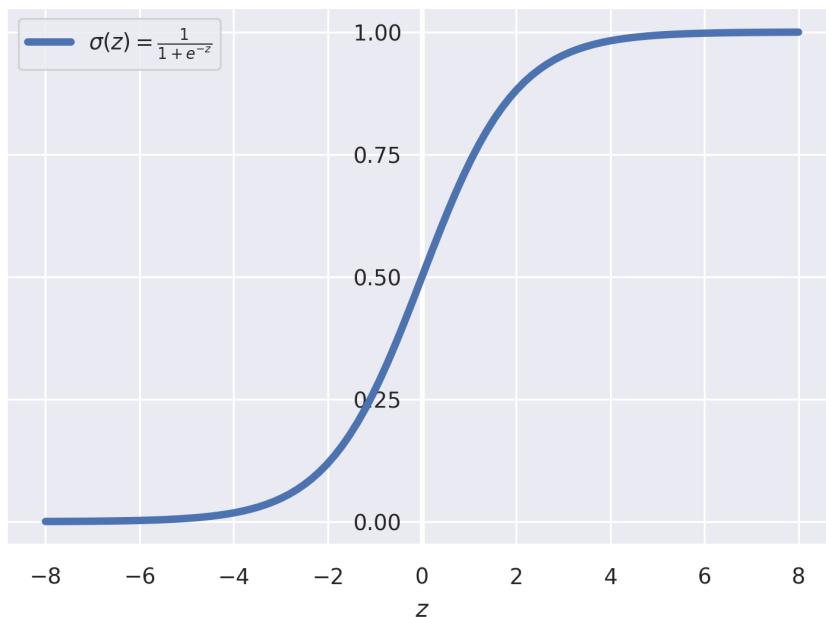
Όπως και στην περίπτωση της γραμμικής παλινδρόμησης, στις επόμενες 3 υποενότητες θα εξετάσουμε την μορφή της αναπαράστασης των υποθέσεων, την μετρική αξιολόγησης των υποθέσεων και τον αλγόριθμο αναζήτησης της λογιστικής παλινδρόμησης.

2.2.1 Αναπαράσταση υποθέσεων

Στη λογιστική παλινδρόμηση αναπαριστούμε μια υπόθεση, h , ως εξής:

$$h_{\theta}(\mathbf{x}) = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n) \quad (2.8)$$

όπου $\sigma(z) = \frac{1}{1+e^{-z}}$ η σιγμοειδής ή αλλιώς λογιστική συνάρτηση (Σχήμα 2.4), από την οποία έχει πάρει το όνομα του ο αλγόριθμος της λογιστικής παλινδρόμησης.



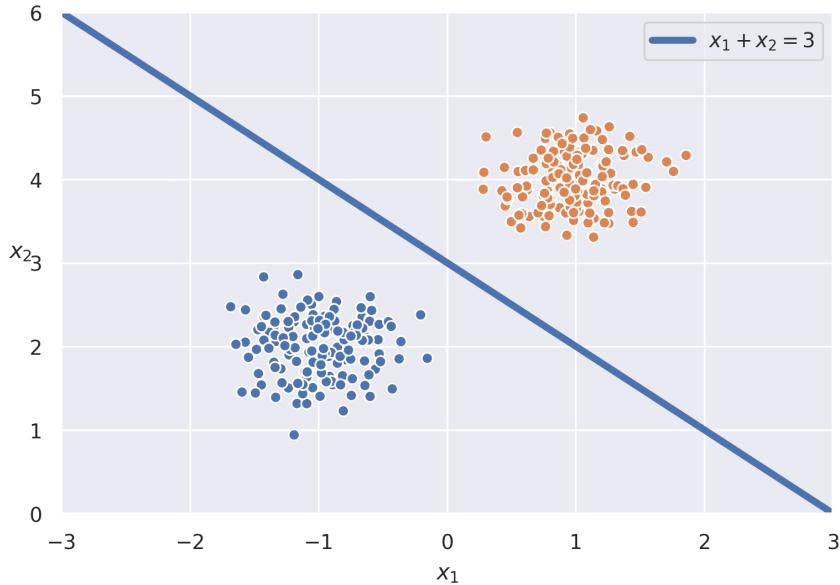
Σχήμα 2.4: Η σιγμοειδής συνάρτηση $\sigma(z) = \frac{1}{1+e^{-z}}$

Η έξοδος ενός μοντέλου λογιστικής παλινδρόμησης, $h_{\theta}(\mathbf{x})$, για μια περίπτωση \mathbf{x} ανήκει στο διάστημα $[0, 1]$ και ερμηνεύεται ως η πιθανότητα η περίπτωση \mathbf{x} να ανήκει στην κλάση 1, $p(y = 1 | \mathbf{x})$. Συμπληρωματικά, η πιθανότητα η περίπτωση \mathbf{x} να ανήκει στην κλάση 0, $p(y = 0 | \mathbf{x})$, ισούται με $1 - h_{\theta}(\mathbf{x})$. Προκειμένου να δώσουμε ως πρόβλεψη μία από τις δύο διακριτές κλάσεις 0 ή 1, εξετάζουμε αν η έξοδος του μοντέλου είναι μεγαλύτερη ή ίση από 0.5.

$$y = \begin{cases} 1, & \text{αν } p(y = 1 | \mathbf{x}) \geq 0.5 \\ 0, & \text{αλλιώς.} \end{cases}$$

Η έξοδος της λογιστικής συνάρτησης είναι μεγαλύτερη ή ίση από 0.5 όταν η είσοδος της είναι μεγαλύτερη ή ίση του μηδενός, δηλαδή όταν $\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n > 0$. Αυτό σημαίνει πως ένα μοντέλο λογιστικής παλινδρόμησης συνιστά ένα υπερεπίπεδο $n - 1$ διαστάσεων στον χώρο των n διαστάσεων των δεδομένων.

Ως παράδειγμα, μπορούμε να θεωρήσουμε το απλό πρόβλημα δυαδικής ταξινόμησης με δύο μεταβλητές εισόδου που φαίνεται στο Σχήμα 2.5. Έστω μοντέλο λογιστικής παλινδρόμησης με παραμέτρους $\theta_0 = -3$ και $\theta_1 = \theta_2 = 1$. Το μοντέλο θα δίνει στην έξοδο του την κλάση 1 όταν $-3 + x_1 + x_2 \geq 0$, δηλαδή όταν $x_1 + x_2 \geq 3$. Συμπληρωματικά το μοντέλο θα δίνει στην έξοδο του την κλάση 0 όταν $-3 + x_1 + x_2 < 0$, δηλαδή όταν $x_1 + x_2 < 3$. Επομένως το μοντέλο αυτό συνιστά την ευθεία $x_1 + x_2 = 3$, η οποία διαχωρίζει άπταιστα τα παραδείγματα του συγκεκριμένου συνόλου εκπαίδευσης.



Σχήμα 2.5: Ένα απλό πρόβλημα δυαδικής ταξινόμησης με δύο μεταβλητές εισόδου.

2.2.2 Αξιολόγηση υποθέσεων

Στον αλγόριθμο της λογιστικής παλινδρόμησης χρησιμοποιείται ως συνάρτηση απώλειας ο αρνητικός λογάριθμος της πιθανότητας που αποδίδει το μοντέλο στην κλάση ενός παραδείγματος, η οποία καλείται λογαριθμική απώλεια (logarithmic loss ή πιο απλά log loss), καθώς και δυαδική διασταυρωμένη εντροπία (binary cross-entropy):

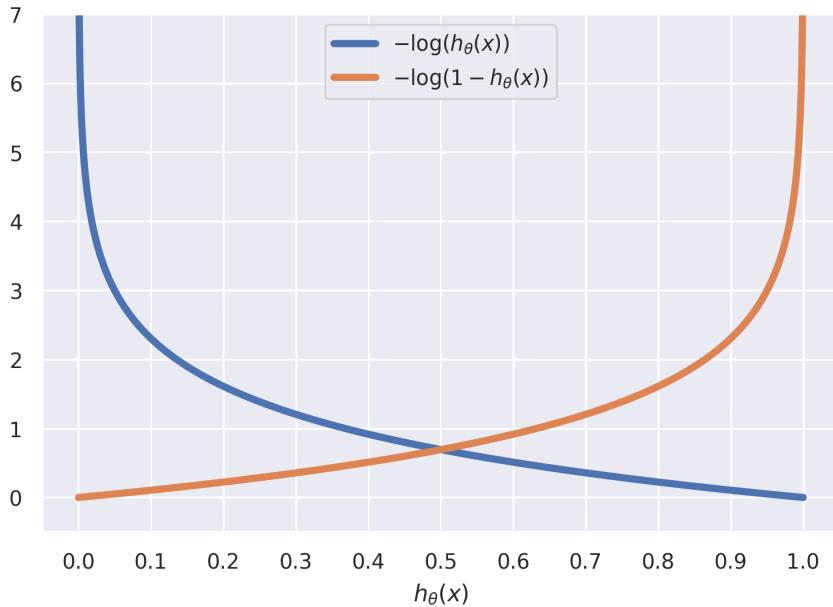
$$L_{\log}(h_{\theta}(\mathbf{x}), y) = -\log(p(y|\mathbf{x})) = \begin{cases} -\log(h_{\theta}(\mathbf{x})), & \text{av } y = 1 \\ -\log(1 - h_{\theta}(\mathbf{x})), & \text{av } y = 0. \end{cases} = -y \log(h_{\theta}(\mathbf{x})) - (1 - y) \log(1 - h_{\theta}(\mathbf{x})) \quad (2.9)$$

Στο Σχήμα 2.6 βλέπουμε μια γραφική απεικόνιση της συνάρτησης αυτής όταν η πραγματική κλάση y είναι 1 (με μπλε χρώμα) ή 0 (με πορτοκαλί χρώμα), όπου στον οριζόντιο άξονα έχουμε την έξοδο του μοντέλου, $h_{\theta}(\mathbf{x})$, και στον κατακόρυφο άξονα την αντίστοιχη έξοδο της συνάρτησης.

Η συνάρτηση κόστους της λογιστικής παλινδρόμησης υπολογίζει τον μέσο όρο της λογαριθμικής απώλειας στο σύνολο εκπαίδευσης:

$$J_{\log}(\theta) = \frac{1}{m} \sum_{i=1}^m L_{\log} \quad (2.10)$$

Όπως και στην περίπτωση της γραμμικής παλινδρόμησης, έτοι και εδώ, η συνάρτηση κόστους είναι κυρτή.



Σχήμα 2.6: Η συνάρτηση της λογαριθμικής απώλειας.

2.2.3 Αναζήτηση της βέλτιστης υπόθεσης

Η αναζήτηση στη λογιστική παλινδρόμηση εκτελείται, όπως και στην περίπτωση της γραμμικής παλινδρόμησης, με τον αλγόριθμο της επικλινούς καθόδου.

Ας υπολογίσουμε την μερική παράγωγο της συνάρτησης κόστους ως προς μια παράμετρο θ_j . Αξιοποιώντας και πάλι τους κανόνες της άθροισης και της αλυσίδας καθώς και ότι αν $f(x) = \log(x)$ τότε $f'(x) = \frac{1}{x}$ και αν $f(x) = \sigma(x)$ τότε $f'(x) = \sigma(x)(1 - \sigma(x))$ έχουμε:

$$\begin{aligned}
\frac{\partial J_{\log}}{\partial \theta_j} &= \frac{\partial}{\partial \theta_j} \frac{1}{m} \sum_{i=1}^m L_{\log} = \frac{1}{m} \sum_{i=1}^m \frac{\partial}{\partial \theta_j} - y^{(i)} \log(h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})) - (1 - y^{(i)}) \log(1 - h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})) = \\
&= \frac{1}{m} \sum_{i=1}^m -y^{(i)} \frac{1}{h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})} h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) (1 - h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})) x_j^{(i)} \\
&\quad - (1 - y^{(i)}) \frac{1}{1 - h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})} (-1) h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) (1 - h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})) x_j^{(i)} \\
&= \frac{1}{m} \sum_{i=1}^m -y^{(i)} (1 - h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})) x_j^{(i)} - (1 - y^{(i)}) (-1) h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) x_j^{(i)} = \\
&= \frac{1}{m} \sum_{i=1}^m -y^{(i)} x_j^{(i)} + y^{(i)} h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) x_j^{(i)} + h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) x_j^{(i)} - y^{(i)} h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) x_j^{(i)} \\
&= \frac{1}{m} \sum_{i=1}^m (h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)}) x_j^{(i)}
\end{aligned} \tag{2.11}$$

Παρατηρούμε ότι η μερική παράγωγος έχει την ίδια μορφή με την περίπτωση της γραμμικής παλινδρόμησης, με τη διαφορά ωστόσο να έγκειται στη διαφορετική αναπαράσταση του μοντέλου, $h_{\boldsymbol{\theta}}(x)$. Οι παράμετροι ενός μοντέλου λογιστικής παλινδρόμησης ενημερώνονται επαναληπτικά από τον αλγόριθμο της επικλινούς καθόδου ως εξής:

- $\boldsymbol{\theta}' := \boldsymbol{\theta}$

- $\forall j \in \{0, \dots, n\}: \theta_j = \theta_j - \eta \frac{1}{m} \sum_{i=1}^m (h_{\theta'}(\mathbf{x}^{(i)}) - y^{(i)}) x_j^{(i)}$

ή με σημειογραφία γραμμικής άλγεβρας:

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \frac{1}{m} \mathbf{X}^\top (\sigma(\mathbf{X}\boldsymbol{\theta}) - \mathbf{y}) \quad (2.12)$$

2.3 Ταξινόμηση σε Πάνω από Δύο Κλάσεις

Ένας περιορισμός της βασικής εκδοχής του αλγορίθμου της λογιστικής παλινδρόμησης είναι πως δεν μπορεί να χρησιμοποιηθεί σε εργασίες ταξινόμησης με πάνω από δύο κλάσεις (multi-class classification). Για την αντιμετώπιση αυτού του προβλήματος έχουν προταθεί δύο βασικές προσεγγίσεις: α) η μετατροπή της εργασίας ταξινόμησης σε πάνω από 2 κλάσεις σε πολλές εργασίες ταξινόμησης 2 κλάσεων, και β) η παραλλαγή του αλγορίθμου ώστε να μπορεί να χειριστεί δεδομένα με πάνω από 2 κλάσεις που καλείται **πολυχοτομική λογιστική παλινδρόμηση** (multinomial logistic regression).

2.3.1 Αποσύνθεση σε πολλές εργασίες 2 κλάσεων

Η πιο απλή προσέγγιση για να αποσυντεθεί μια εργασία ταξινόμησης πολλών κλάσεων σε πολλές εργασίες δυαδικής ταξινόμησης είναι η επονομαζόμενη **μία εναντίων των υπολοίπων** (one versus rest) ή **μία εναντίων όλων** (one versus all). Σύμφωνα με αυτήν την προσέγγιση, δημιουργούνται τόσα δυαδικά μοντέλα, όσα και οι κλάσεις. Για την εκπαίδευση του μοντέλου της κάθε κλάσης, θεωρούνται θετικά τα παραδείγματα της κλάσης αυτής και αρνητικά τα παραδείγματα των υπόλοιπων κλάσεων. Για την ταξινόμηση μιας νέας περίπτωσης, λαμβάνουμε την έξοδο όλων των μοντέλων και δίνουμε στην έξοδο την κλάση, το μοντέλο της οποίας επιστρέφει τη μεγαλύτερη πιθανότητα.

Μια εναλλακτική προσέγγιση για να επιτευχθεί η ίδια αποσύνθεση είναι η **μία εναντίων μίας** (one versus one), σύμφωνα με την οποία δημιουργείται ένα μοντέλο δυαδικής ταξινόμησης για κάθε ζεύγος κλάσεων. Για c κλάσεις, έχουμε $\frac{c(c-1)}{2}$ ζεύγη και αντίστοιχα μοντέλα δυαδικής ταξινόμησης. Για την εκπαίδευση του μοντέλου ενός ζεύγους κλάσεων, θεωρούνται θετικά τα παραδείγματα της μίας κλάσης, αρνητικά της άλλης κλάσης και αγνοούνται τα υπόλοιπα παραδείγματα. Για την ταξινόμηση μιας νέας περίπτωσης, λαμβάνουμε την έξοδο όλων των μοντέλων και δίνουμε στην έξοδο την κλάση με τις περισσότερες θετικές προβλέψεις. Ο μέγιστος αριθμός αυτών των θετικών προβλέψεων είναι $c - 1$, όσα και τα μοντέλα στα οποία συμμετέχει η κάθε κλάση. Εναλλακτικά μπορούμε να αθροίσουμε τις $c - 1$ πιθανότητες που λαμβάνει η κάθε κλάση από τα αντίστοιχα μοντέλα και να δώσουμε στην έξοδο την κλάση με το μεγαλύτερο άθροισμα.

Ο Πίνακας 2.2 παρουσιάζει ένα παράδειγμα των παραπάνω αποσυνθέσεων για μια εργασία τεσσάρων κλάσεων 0, 1, 2, 3, με δύο παραδείγματα εκπαίδευσης ανά κλάση. Η πρώτη στήλη δείχνει την κλάση κάθε ενός από τα 8 παραδείγματα εκπαίδευσης. Οι επόμενες 4 στήλες δείχνουν την κλάση αυτών των 8 παραδειγμάτων κατά την εκπαίδευση των αντίστοιχων τεσσάρων μοντέλων στην αποσύνθεση μία εναντίων των υπολοίπων. Οι τελευταίες 6 στήλες δείχνουν την κλάση αυτών των 8 παραδειγμάτων κατά την εκπαίδευση των αντίστοιχων $\frac{4(4-1)}{2} = 6$ μοντέλων στην αποσύνθεση μία εναντίον μίας. Αν κάποιο παράδειγμα δεν χρησιμοποιείται για την εκπαίδευση κάποιου μοντέλου τότε στο αντίστοιχο κελί του πίνακα εμφανίζεται μία παύλα.

2.3.2 Πολυχοτομική λογιστική παλινδρόμηση

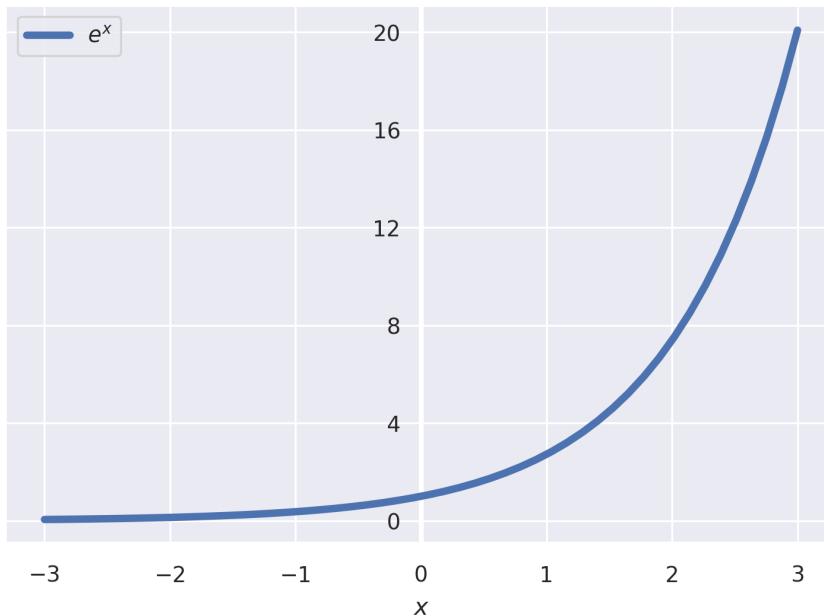
Στην πολυχοτομική λογιστική παλινδρόμηση μαθαίνουμε ένα ξεχωριστό διάνυσμα παραμέτρων θ_k για κάθε κλάση $k \in 1, \dots, c$. Θα ορίσουμε ως $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_c]$ τον πίνακα με όλες τις παραμέτρους όλων των κλάσεων. Η έξοδος του μοντέλου είναι μια κατανομή πιθανότητας για τις κλάσεις, η οποία προκύπτει από την εφαρμογή της συνάρτησης softmax:

Αρχικά	0	1	2	3	0 vs 1	0 vs 2	0 vs 3	1 vs 2	1 vs 3	2 vs 3
0	1	0	0	0	1	1	1	-	-	-
0	1	0	0	0	1	1	1	-	-	-
1	0	1	0	0	0	-	-	1	1	-
1	0	1	0	0	0	-	-	1	1	-
2	0	0	1	0	-	0	-	0	-	1
2	0	0	1	0	-	0	-	0	-	1
3	0	0	0	1	-	-	0	-	0	0
3	0	0	0	1	-	-	0	-	0	0

Πίνακας 2.2: Παραδείγματα εκπαίδευσης ανά μοντέλο στις μεθόδους αποσύνθεσης μία εναντίων των υπολοίπων και μία εναντίον μίας

$$h_{\Theta}(x) = \left(\frac{\exp(\theta_1 \cdot x)}{\sum_{l=1}^c \exp(\theta_l \cdot x)}, \dots, \frac{\exp(\theta_c \cdot x)}{\sum_{l=1}^c \exp(\theta_l \cdot x)} \right) \quad (2.13)$$

Η συνάρτηση αυτή υπολογίζει το εσωτερικό γινόμενο των παραμέτρων της κάθε κλάσης με το διάνυσμα των μεταβλητών εισόδου, το περνάει από την εκθετική συνάρτηση (Σχήμα 2.7), αποδίδοντας έτσι εκθετικά μεγαλύτερες τιμές σε σχέση με τα αντίστοιχα εσωτερικά γινόμενα, και τέλος διαιρεί τις τιμές αυτές με το άθροισμα τους, προκειμένου να δώσει μια κατανομή πιθανοτήτων. Για παράδειγμα, για εσωτερικά γινόμενα 5 κλάσεων ίσα με -1, 0, 1, 2 και 3, η κατανομή πιθανοτήτων που προκύπτει είναι αντίστοιχα [0.01, 0.03, 0.09, 0.24, 0.63].



Σχήμα 2.7: Η εκθετική συνάρτηση.

Στην υλοποίηση της συνάρτησης softmax σε κώδικα, χρησιμοποιείται ένα τρικ προκειμένου να αποφευχθούν προβλήματα αριθμητικής αστάθειας κατά τη διαίρεση των πολύ μεγάλων αριθμών που θα προκύψουν στον αριθμητή και παρονομαστή λόγω της εκθετικής συνάρτησης. Το τρικ αυτό συνίσταται στην αφαίρεση από κάθε εσωτερικό γινόμενο, του μεγαλύτερου από όλα τα εσωτερικά γινόμενα, έτσι ώστε να γίνουν όλα μικρότερα ή ίσα με το μηδέν, οπότε το πέρασμα τους από την εκθετική συνάρτηση θα δώσει έναν αριθμό μεταξύ

0 και 1. Στο προηγούμενο μας παράδειγμα, τα εσωτερικά γινόμενα θα γινόταν αντίστοιχα -4, -3, -2, -1 και 0, και η κατανομή πιθανοτήτων θα ήταν ίδια με πριν.

Στην πολυχοτομική λογιστική παλινδρόμηση, χρησιμοποιούμε ως συνάρτηση απώλειας την λογαριθμική απώλεια της πραγματικής κλάσης, όπως κάναμε και στην περίπτωση της λογιστικής παλινδρόμησης για προβλήματα δυαδικής ταξινόμησης:

$$L_{\log}(h_{\Theta}(\mathbf{x}), y) = -\log(p(y|\mathbf{x})) = -\log \left(\frac{\exp(\theta_y \cdot \mathbf{x})}{\sum_{l=1}^c \exp(\theta_l \cdot \mathbf{x})} \right) \quad (2.14)$$

Θα μπορούσαμε εναλλακτικά και ισοδύναμα να χρησιμοποιήσουμε τη συνάρτηση της διασταυρωμένης εντροπίας πολλαπλών κλάσεων (multi-class cross-entropy):

$$L_{CE}(h_{\Theta}(\mathbf{x}), y) = -\sum_{k=1}^c [y = k] \log(p(k|\mathbf{x})) = -\sum_{k=1}^c [y = k] \log \left(\frac{\exp(\theta_k \cdot \mathbf{x})}{\sum_{l=1}^c \exp(\theta_l \cdot \mathbf{x})} \right)$$

Ως συνάρτηση κόστους χρησιμοποιούμε όπως πάντα τον μέσο όρο της συνάρτησης απώλειας στα παραδείγματα εκπαίδευσης:

$$J_{\log}(\Theta) = \frac{1}{m} \sum_{i=1}^m L_{\log}(h_{\Theta}(\mathbf{x}^{(i)}, y^{(i)})) = -\frac{1}{m} \sum_{i=1}^m \log \left(\frac{\exp(\theta_{y^{(i)}} \cdot \mathbf{x}^{(i)})}{\sum_{l=1}^c \exp(\theta_l \cdot \mathbf{x}^{(i)})} \right) \quad (2.15)$$

Ισοδύναμα χρησιμοποιώντας την συνάρτηση της διασταυρωμένης εντροπίας πολλαπλών κλάσεων:

$$J_{CE}(\Theta) = \frac{1}{m} \sum_{i=1}^m L_{CE}(h_{\Theta}(\mathbf{x}^{(i)}, y^{(i)})) = -\frac{1}{m} \sum_{i=1}^m \sum_{k=1}^c [y = k] \log \left(\frac{\exp(\theta_k \cdot \mathbf{x})}{\sum_{l=1}^c \exp(\theta_l \cdot \mathbf{x})} \right)$$

Η αναζήτηση στην πολυχοτομική λογιστική παλινδρόμηση εκτελείται, όπως και στις περιπτώσεις της γραμμικής και λογιστικής παλινδρόμησης, με τον αλγόριθμο της επικλινούς καθόδου.

Ας υπολογίσουμε την μερική παράγωγο της συνάρτησης απώλειας της εξίσωσης 2.14 ως προς την j -οστή παράμετρο της κλάσης k , για ένα παράδειγμα εκπαίδευσης (\mathbf{x}, y) , αξιοποιώντας την ταυτότητα $\log(\frac{a}{b}) = \log(a) - \log(b)$:

$$\begin{aligned} \frac{\partial L_{\log}}{\partial \theta_{kj}} &= \frac{\partial}{\partial \theta_{kj}} -\log \left(\frac{\exp(\theta_y \cdot \mathbf{x})}{\sum_{l=1}^c \exp(\theta_l \cdot \mathbf{x})} \right) = \\ &= -\frac{\partial}{\partial \theta_{kj}} \log(\exp(\theta_y \cdot \mathbf{x})) - \log \left(\sum_{l=1}^c \exp(\theta_l \cdot \mathbf{x}) \right) = \\ &= -\frac{\partial}{\partial \theta_{kj}} \theta_y \cdot \mathbf{x} - \log \left(\sum_{l=1}^c \exp(\theta_l \cdot \mathbf{x}) \right) \end{aligned} \quad (2.16)$$

Για τον πρώτο όρο της διαφοράς έχουμε:

$$\frac{\partial}{\partial \theta_{kj}} \theta_y \cdot \mathbf{x} = [y = k] x_j$$

Για τον δεύτερο όρο της διαφοράς, αξιοποιώντας τον κανόνα της αλυσίδας, και δεδομένου ότι $\log'(x) = \frac{1}{x}$ και $\exp'(x) = \exp(x)$ έχουμε:

$$\frac{\partial}{\partial \theta_{kj}} \log \left(\sum_{l=1}^c \exp(\boldsymbol{\theta}_l \cdot \mathbf{x}) \right) = \frac{\frac{\partial}{\partial \theta_{kj}} \sum_{l=1}^c \exp(\boldsymbol{\theta}_l \cdot \mathbf{x})}{\sum_{l=1}^c \exp(\boldsymbol{\theta}_l \cdot \mathbf{x})} = \frac{\exp(\boldsymbol{\theta}_k \cdot \mathbf{x})}{\sum_{l=1}^c \exp(\boldsymbol{\theta}_l \cdot \mathbf{x})} x_j = p(k|\mathbf{x}) x_j$$

Βάζοντας μαζί τους δύο όρους έχουμε:

$$\frac{\partial L_{\log}}{\partial \theta_{kj}} = -([y = k] - p(k|\mathbf{x})) x_j$$

Άρα η αντίστοιχη μερική παράγωγος της συνάρτησης κόστους της εξίσωσης 2.15 είναι:

$$\frac{\partial J_{\log}}{\partial \theta_{kj}} = -\frac{1}{m} \sum_{i=1}^m ([y = k] - p(k|\mathbf{x})) x_j \quad (2.17)$$

Σύμφωνα με τον αλγόριθμο της επικλινούς καθόδου, οι παράμετροι του μοντέλου θα ενημερώνονται επαναληπτικά ως εξής:

- $\forall k \in \{1, \dots, c\}: \theta'_k = \theta_k$
- $\forall j \in \{0, \dots, n\}, \forall k \in \{1, \dots, c\}: \theta_{kj} = \theta_{kj} - \eta \frac{1}{m} \sum_{i=1}^m \left(\frac{\exp(\boldsymbol{\theta}'_k \cdot \mathbf{x}^{(i)})}{\sum_{l=1}^c \exp(\boldsymbol{\theta}'_l \cdot \mathbf{x}^{(i)})} - [y^{(i)} = k] \right) x_j^{(i)}$

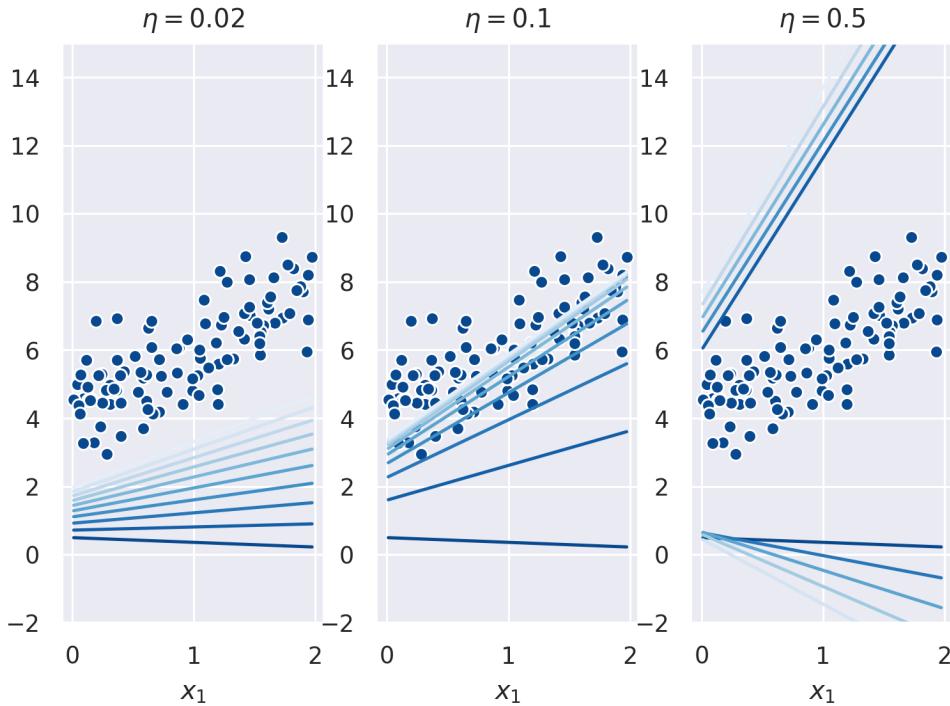
2.4 Ρυθμός Μάθησης

Ο ρυθμός μάθησης, η , καθορίζει τόσο τον χρόνο που θα χρειαστεί η επικλινής κάθοδος για να συγκλίνει στην ελάχιστη τιμή της συνάρτησης κόστους, όσο και το αν θα καταφέρει να συγκλίνει ή όχι. Αν η τιμή του ρυθμού μάθησης τεθεί σε μια πολύ μικρή τιμή, τότε η σύγκλιση θα επιτευχθεί, αλλά θα απαιτήσει πάρα πολλές επαναλήψεις και αντίστοιχα πολύ μεγάλη χρονική καθυστέρηση. Από την άλλη, αν η τιμή του ρυθμού μάθησης τεθεί σε μια πολύ μεγάλη τιμή, τότε μπορεί η σύγκλιση να είναι αδύνατον να επιτευχθεί ή ακόμα και να έχουμε αποκλίνουσα συμπεριφορά, κατά την οποία πηγαίνουμε σε μεγαλύτερες τιμές της συνάρτησης κόστους σε κάθε επανάληψη.

Στο Σχήμα 2.8 παρουσιάζονται οι 10 πρώτες επαναλήψεις του αλγορίθμου της επικλινούς καθόδου για τρεις διαφορετικές τιμές του ρυθμού μάθησης. Ξεκινώντας με τυχαίες τιμές για τα θ_0 και θ_1 , βλέπουμε τα μοντέλα που αντιστοιχούν σε κάθε επανάληψη, ξεκινώντας από το σκούρο μπλε και συνεχίζοντας στο πιο ανοιχτό. Παρατηρούμε πως η παράμετρος θ_0 επηρεάζει την δυνατότητα και την ταχύτητα σύγκλισης στα δεδομένα μας. Για μια μικρή τιμή της παραμέτρου $\eta = 0.02$ η σύγκλιση καθυστερεί, ενώ για μια μεγαλύτερη τιμή $\eta = 0.1$ η σύγκλιση επιτυγχάνεται πολύ γρήγορα. Αντίστοιχα, μια μεγάλη τιμή $\eta = 0.5$ προκαλεί μια αποκλίνουσα συμπεριφορά.

Προκειμένου να βρούμε μια κατάλληλη τιμή για τον ρυθμό μάθησης, πρέπει να πειραματιστούμε με διάφορες τιμές και να μελετήσουμε το διάγραμμα που απεικονίζει το κόστος στον άξονα των y και τις επαναλήψεις στον άξονα των x . Αν δούμε ότι το κόστος ανεβαίνει ή ανεβοκατεβαίνει με τις επαναλήψεις, τότε πρέπει να επιλέξουμε έναν μικρότερο ρυθμό μάθησης, ενώ αν δούμε ότι κατεβαίνει με πολύ αργό ρυθμό, τότε πρέπει να επιλέξουμε έναν μεγαλύτερο ρυθμό μάθησης. Συνήθως δοκιμάζουμε τιμές ανά δύναμη του 10 με μία ενδιάμεση μεταξύ των τιμών, π.χ. 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1 κτλ.

Ένα σχετικό ερώτημα που συνδέεται με τον ρυθμό μάθησης αφορά στο πόσες επαναλήψεις ενημερώσεων παραμέτρων θα πρέπει να κάνουμε. Μπορούμε να θέσουμε έναν μεγάλο αριθμό επαναλήψεων, ελέγχοντας όμως ταυτόχρονα μήπως ο αλγόριθμος έχει συγκλίνει στο ελάχιστο κόστος. Αυτό ο ελεγχος, θα μπορούσε να γίνει εποπτικά, εξετάζοντας τη γραφική απεικόνιση του κόστους με την πάροδο του χρόνου. Αν παρατηρήσουμε ότι το κόστος σταματήσει να μειώνεται σημαντικά, αυτό σημαίνει πως ο αλγόριθμος έχει συγκλίνει. Στην πράξη, μπορούμε να εξετάζουμε έπειτα από κάθε επανάληψη την διαφορά στην τιμή του κόστους και να



Σχήμα 2.8: Γραμμική παλινδρόμηση για διαφορετικές τιμές της παραμέτρου η

σταματήσουμε όταν αυτή γίνει μικρότερη από από έναν μικρό θετικό αριθμό, π.χ. 10^{-4} . Εναλλακτικά μπορούμε να σταματήσουμε όταν η νόρμα του διανύσματος των κλίσεων γίνει μικρότερη από έναν μικρό θετικό αριθμό, π.χ. 10^{-4} .

2.5 Κλιμάκωση και Προτυποποίηση Χαρακτηριστικών

Η σύγκλιση της επικλινούς καθόδου καθυστερεί όταν οι τιμές των μεταβλητών εισόδου ανήκουν σε αρκετά διαφορετικές κλίμακες, όπως για παράδειγμα οι μεταβλητές ηλικία και καταθέσεις ενός συνόλου πελατών μιας τράπεζας με εύρος τιμών από 15 έως 95 έτη και από 500 ευρώ έως 800 χιλιάδες ευρώ. Αυτό οφείλεται στο γεγονός ότι ένας κατάλληλος ρυθμός μάθησης για μια μεταβλητή με μικρό εύρος τιμών, θα είναι αρκετά μικρός για μια άλλη μεταβλητή με μεγάλο εύρος τιμών. Προκειμένου να αποτρέψουμε κάτι τέτοιο, μετασχηματίζουμε τις μεταβλητές εισόδου έτσι ώστε να βρίσκονται όλες στην ίδια κλίμακα. Μπορούμε να φανταστούμε την μορφή της συνάρτηση κόστους ως μια γόνδολα για τις παραμέτρους που αντιστοιχούν σε δύο τέτοιες μεταβλητές, και ως μπολ για δύο παραμέτρους, οι μεταβλητές των οποίων βρίσκονται στην ίδια κλίμακα.

Δύο τεχνικές που φέρνουν ένα σύνολο μεταβλητών εισόδου στην ίδια κλίμακα είναι η **κλιμάκωση** (scaling) και η **προτυποποίηση** (standardization). Στην κλιμάκωση, εφαρμόζουμε σε κάθε μεταβλητή τον εξής μετασχηματισμό (Εξίσωση 2.18): αφαιρούμε από την τιμή κάθε παραδείγματος την ελάχιστη τιμή της μεταβλητής και διαιρούμε με το εύρος τιμών της μεταβλητής (διαφορά της μέγιστης μείον την ελάχιστη τιμή της μεταβλητής). Με αυτόν τον τρόπο, όλες οι μεταβλητές αποκτούν εύρος τιμών από 0 έως 1. Ένα μειονέκτημα αυτής της μεθόδου είναι ότι αν υπάρχουν έκτοπες τιμές (outliers), τότε ο μεγαλύτερος όγκος της κατανομής των τιμών της μεταβλητής θα μαζευτεί σε ένα μικρότερο εύρος.

$$x_j^{(i)} := \frac{x_j^{(i)} - \min_i x_j^{(i)}}{\max_i x_j^{(i)} - \min_i x_j^{(i)}} \quad (2.18)$$

Στην προτυποποίηση εφαρμόζουμε σε κάθε μεταβλητή τον εξής μετασχηματισμό (Εξίσωση 2.19): αφαι-

ρούμε από την τιμή κάθε παραδείγματος, την μέση τιμή της μεταβλητής και διαιρούμε με την τυπική απόκλισή της. Έτσι όλες οι μεταβλητές αποκτούν την ίδια μέση τιμή, 0, και μοναδιαία διακύμανση. Σε αντίθεση με την κλιμάκωση, η τεχνική αυτή δεν επηρεάζεται αρνητικά από την παρουσία έκτοπων τιμών. Ωστόσο, το νέο εύρος τιμών των μεταβλητών δεν είναι περιορισμένο εντός συγκεκριμένων προκαθορισμένων ορίων όπως στην περίπτωση της κλιμάκωσης.

$$x_j^{(i)} := \frac{x_j^{(i)} - \mu_j}{\sigma_j} \quad \mu_j = \frac{1}{m} \sum_i x_j^{(i)} \quad \sigma_j = \sqrt{\frac{\sum_i (x_j^{(i)} - \mu_j)^2}{m}} \quad (2.19)$$

2.6 Στοχαστική Επικλινής Κάθοδος

Όσο μεγαλύτερο είναι το σύνολο των δεδομένων εκπαίδευσης, τόσο περισσότερο καθυστερεί η επικλινής κάθοδος να συγκλίνει, καθώς η ενημέρωση των παραμέτρων σε κάθε επανάληψη απαιτεί τον υπολογισμό της εξόδου του μοντέλου για κάθε παράδειγμα εκπαίδευσης. Μια λύση σε αυτό το μειονέκτημα αποτελεί η **στοχαστική επικλινής κάθοδος** (stochastic gradient descent), η οποία ενημερώνει τις παραμέτρους με βάση ένα μόνο παράδειγμα εκπαίδευσης. Οι ενημερώσεις στην γραμμική και λογιστική παλινδρόμηση για ένα παράδειγμα $x^{(i)}$ γίνονται επομένως:

- $\theta' := \theta$
- $\forall j \in \{0, \dots, n\}: \theta_j = \theta_j - \eta (h_{\theta'}(\mathbf{x}^{(i)}) - y^{(i)}) x_j^{(i)}$

ή χρησιμοποιώντας σημειογραφία γραμμικής άλγεβρας:

$$\theta = \theta - \eta \mathbf{x}^{(i)} (h_{\theta}(\mathbf{x}^{(i)}) - y^{(i)}) \quad (2.20)$$

Έστω για παράδειγμα ένα πρόβλημα ταξινόμησης μιας αίτησης για καταναλωτικό δάνειο, με βάση δύο μεταβλητές εισόδου: το μηνιαίο εισόδημα του πελάτη που αιτείται το δάνειο και τις αποταμιεύσεις του στην τράπεζα σε χιλιάδες ευρώ. Έστω ότι οι αρχικές τιμές των παραμέτρων του γραμμικού μοντέλου μας είναι $\theta_0 = \theta_1 = \theta_2 = 0$ και ο ρυθμός μάθησης είναι $\eta = 0.1$. Έστω παράδειγμα εκπαίδευσης (\mathbf{x}, y) στο οποίο $x_1 = 3$, $x_2 = 2$ και $y = 0$. Οι παράμετροι θα ενημερωθούν ως εξής:

$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix} - \eta \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} (h_{\theta}(\mathbf{x}) - y) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} - 0.1 \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} (0.5 - 0) = \begin{bmatrix} -0.05 \\ -0.15 \\ -0.1 \end{bmatrix}$$

Παρατηρούμε πως οι τιμές των παραμέτρων μειώθηκαν. Αυτό είναι κάτι αναμενόμενο, αφού για να μειωθεί η λογαριθμική απώλεια στο παράδειγμα αυτό, θα πρέπει να μειωθεί η έξοδος του μοντέλου, που ισούται με 0.5 με βάση τις αρχικές παραμέτρους, η οποία για να μειωθεί θα πρέπει να μειωθεί το εσωτερικό γινόμενο των παραμέτρων με τις θετικές τιμές εισόδου του παραδείγματος και την θετική προκαθορισμένη είσοδο 1 που αντιστοιχεί στην αποτέμνουσα.

Η εφαρμογή αυτής της παραλλαγής, απαιτεί το τυχαίο ανακάτεμα των δεδομένων εκπαίδευσης, πριν από κάθε πέρασμα από όλα τα δεδομένα εκπαίδευσης, το οποίο καλείται **εποχή** (epoch). Το γεγονός ότι εξετάζει ένα παράδειγμα κάθε φορά, επιτρέπει την μάθηση από τεράστιους όγκους δεδομένων που δεν χωρούν στη μνήμη, καθώς και από συνεχείς ροές δεδομένων όπως αναρτήσεις σε κοινωνικά μέσα και χρηματιστηριακές συναλλαγές. Η έκδοση της επικλινούς καθόδου που είχαμε δει μέχρι αυτήν την ενότητα, η οποία σε αντίθεση λαμβάνει υπόψη όλα τα δεδομένα σε κάθε βήμα ενημέρωσης, καλείται και επικλινής κάθοδος **κατά δέσμες** (batch).

Η στοχαστική επικλινής κάθοδος ελαχιστοποιεί μια προσέγγιση της πραγματικής συνάρτησης κόστους. Αυτό έχει ως αποτέλεσμα να μην εμφανίζει την ομαλή συμπεριφορά της επικλινούς καθόδου κατά δέσμες. Το

κόστος σταδιακά θα ελαττώνεται, όμως ενδέχεται να ανεβοκατεβαίνει από επανάληψη σε επανάληψη. Αυτή η συμπεριφορά είναι επιθυμητή σε περιπτώσεις συναρτήσεων κόστους με παραπάνω από ένα ελάχιστα, όπως στα νευρωνικά δίκτυα, καθώς μπορεί να βοηθήσει τον αλγόριθμο να ξεφύγει από τοπικά ελάχιστα. Ωστόσο στα γραμμικά μοντέλα, το αποτέλεσμα είναι πως δεν καταφέρνει ποτέ να συγκλίνει στην ελάχιστη τιμή. Για τον λόγο αυτό, στην στοχαστική επικλινή κάθοδο ο ρυθμός μάθησης μειώνεται με την πάροδο του χρόνου, προκειμένου να επιτευχθεί η σύγκλιση στην ελάχιστη τιμή. Αυτό είναι περιττό στην περίπτωση της επικλινούς καθόδου κατά δέσμες, όπου ούτως ή άλλως μειώνεται σταδιακά η κλίση, επομένως και το βήμα της ενημέρωσης, με αποτέλεσμα να επιτυγχάνεται η σύγκλιση.

Ανάμεσα στην κατά δέσμες και τη στοχαστική βρίσκεται η παραλλαγή της επικλινούς καθόδου κατά **μίνι δέσμες** (mini-batch). Σε αυτήν την παραλλαγή η ενημέρωση των βαρών στηρίζεται σε έναν μικρό αριθμό τυχαία επιλεγμένων παραδειγμάτων. Τα δεδομένα εκπαίδευσης ανακατεύονται τυχαία πριν από κάθε εποχή, όπως και στην περίπτωση της στοχαστικής επικλινούς καθόδου, και στη συνέχεια επιλέγεται διαδοχικά ένας αριθμός από συνεχόμενα δεδομένα. Η παραλλαγή αυτή, μπορεί να αξιοποιήσει κατάλληλο υλικό (όπως κάρτες γραφικών) και λογισμικό προκειμένου οι ενημερώσεις των βαρών να γίνουν παράλληλα για όλα τα παραδείγματα της μίνι δέσμης και έτσι να επιταχυνθεί η διαδικασία της μάθησης.

2.7 Πολυωνυμικά Μοντέλα και Υπερπροσαρμογή

Σε πολλές εργασίες μάθησης με επίβλεψη, η σχέση της μεταβλητής εξόδου με τις μεταβλητές εισόδου δεν είναι γραμμική. Για παράδειγμα, η σχέση της αξίας ενός αυτοκινήτου σε σχέση με την ηλικία του είναι φθίνουσα, όμως συνήθως τα πρώτα χρόνια η αξία μειώνεται πιο απότομα, ενώ μετά από αρκετά χρόνια, η αξία μειώνεται με πολύ πιο αργό ρυθμό. Ομοίως ο διαχωρισμός των σημείων μιας μεγάλης πόλης της Ελλάδας που θα βρέξει μια συγκεκριμένη ώρα της επόμενης ημέρας, σε σχέση με εκείνα που δεν θα βρέξει, με βάση το γεωγραφικό μήκος και πλάτος τους, αναμένεται να μην μπορεί να επιτευχθεί ικανοποιητικά με μία γραμμή. Μια παραβολική συνάρτηση ή γενικότερα ένα πολυώνυμο μεγαλύτερου βαθμού από 1, θα μπορούσε να μοντελοποιήσει καλύτερα τέτοιες μη γραμμικές σχέσεις μεταξύ των μεταβλητών εισόδου και της μεταβλητής εξόδου. Μπορούμε ωστόσο να παραμείνουμε στη χρήση γραμμικών μοντέλων, υιοθετώντας το εξής τρικ. Θα προσθέσουμε επιπλέον μεταβλητές εισόδου, οι οποίες θα αντιστοιχούν σε πολυωνυμικούς παράγοντες των υπαρχουσών μεταβλητών.

Έστω για παράδειγμα, ότι θέλουμε να υπολογίσουμε την αξία ενός ορθογώνιου αγροτεμαχίου με βάση το μήκος (x_1) και πλάτος του (x_2). Ένα γραμμικό μοντέλο παλινδρόμησης θα είχε τη μορφή: $h_1(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$. Ωστόσο, συνήθως η αξία σχετίζεται γραμμικά σε σχέση με το εμβαδόν ενός αγροτεμαχίου. Θα μπορούσαμε επομένως να προσθέσουμε μεταβλητή $x_3 = x_1 * x_2$. Ένα γραμμικό μοντέλο με τη μορφή $h_2(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$ ή ακόμα και $h_3(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_1^2$ θα μπορέσει να μοντελοποιήσει πολύ καλύτερα αυτήν τη μη γραμμική σχέση της αξίας σε σχέση με το μήκος και το πλάτος του οικοπέδου, από ότι το μοντέλο h_1 . Ομοίως για την καλύτερη μοντελοποίηση της αξίας ενός αυτοκινήτου σε σχέση με την ηλικία του (x_1) μέσω ενός γραμμικού μοντέλου, θα μπορούσαμε να προσθέσουμε μεταβλητές $x_2 = x_1^2$ ή ακόμα και $x_3 = x_1^3$. Μια τέτοια προσέγγιση σε εργασίες παλινδρόμησης ονομάζεται **πολυωνυμική παλινδρόμηση** (polynomial regression). Σε τέτοιες περιπτώσεις θα πρέπει οπωσδήποτε να εφαρμόζεται κλιμάκωση των μεταβλητών εισόδου, καθώς οι τιμές των πολυωνυμικών παραγόντων που εισάγονται θα βρίσκονται σίγουρα σε διαφορετική κλίμακα σε σχέση με τις προϋπάρχουσες μεταβλητές.

Ένας κίνδυνος που έλλογχεύει από την προσθήκη πολυωνυμικών παραγόντων στα γραμμικά μοντέλα, είναι η **υπερπροσαρμογή** (overfitting). Είναι μια σημαντική έννοια στη μηχανική μάθηση που αφορά στην υπερβολική προσαρμογή ενός μοντέλου στα δεδομένα εκπαίδευσης, σε βαθμό που βλάπτε την ικανότητα του μοντέλου να δίνει ορθές προβλέψεις σε άλλα δεδομένα που δεν έχει δει, δηλαδή στην ικανότητά του να γενικεύει πέρα από τα δεδομένα εκπαίδευσης. Μια σχετική έννοια είναι αυτή της **πολυπλοκότητας ενός μοντέλου** (model complexity), δηλαδή της ικανότητας του να προσαρμόζεται στα δεδομένα. Η μεγαλύτερη πολυπλοκότητα δίνει περισσότερες δυνατότητες καλύτερης προσαρμογής ενός μοντέλου στα δεδομένα, όμως αυξάνει και τον

κίνδυνο της υπερπροσαρμογής. Ένα μοντέλο με μικρή πολυπλοκότητα δεν κινδυνεύει από υπερπροσαρμογή, αλλά αν η πολυπλοκότητα του δεν είναι αρκετή για να προσαρμοστεί καλά στα δεδομένα εκπαίδευσης, τότε έχουμε το αντίθετο φαινόμενο της **υπόπροσαρμογής** (underfitting). Ένα πετυχημένο μοντέλο θα πρέπει να ισορροπεί μεταξύ της υπόπροσαρμογής και της υπέρπροσαρμογής στα δεδομένα εκπαίδευσης.

Η αντιμετώπιση της υπερπροσαρμογής, μπορεί να επιτευχθεί με τους παρακάτω τρόπους: α) **επιλογή χαρακτηριστικών** (feature selection), β) **ομαλοποίηση** (regularization), γ) **πρόωρη διακοπή** (early stopping). Η επιλογή χαρακτηριστικών αποτελεί έναν ολόκληρο κλάδο της μηχανικής μάθησης που αφορά στην αυτόματη επιλογή ενός υποσυνόλου μεταβλητών εισόδου, το οποίο οδηγεί στην μείωση της πολυπλοκότητας του μοντέλου. Στη ομαλοποίηση διατηρούνται όλες οι μεταβλητές εισόδου, όμως αποθαρρύνονται οι μεγάλες τιμές παραμέτρων, οι οποίες οδηγούν σε πιο απότομες αλλαγές της εξόδου σε σχέση με την είσοδο, και άρα πιο πολύπλοκες συναρτήσεις και επιφάνειες απόφασης. Η πρόωρη διακοπή αφορά επαναληπτικές διαδικασίες προσαρμογής στα δεδομένα, όπως η επικλινή κάθοδος, και συνίσταται στη διακοπή της προσαρμογής νωρίτερα από το αναμενόμενο. Στην επικλινή κάθοδο, αυτό σημαίνει διακοπή της ενημέρωσης των παραμέτρων πριν την επίτευξη της σύγκλισης.

Η ομαλοποίηση επιτυγχάνεται προσθέτοντας στη συνάρτηση κόστους μια συνάρτηση των παραμέτρων. Οι δυο πιο δημοφιλείς συναρτήσεις είναι το άθροισμα των τετραγώνων τους και το άθροισμα της απόλυτης τιμής τους. Στην πρώτη περίπτωση, η ομαλοποίηση ονομάζεται ομαλοποίηση L2 και η (λογιστική) παλινδρόμηση καλείται (λογιστική) **παλινδρόμηση κορυφογραμμής** (ridge regression). Στην δεύτερη περίπτωση, η ομαλοποίηση ονομάζεται ομαλοποίηση L1 και η (λογιστική) παλινδρόμηση καλείται (λογιστική) **παλινδρόμηση λάσο** (lasso regression).

Η Εξίσωση 2.21 δείχνει την προσθήκη της συνάρτησης των παραμέτρων $R(\theta)$, η οποία πολλαπλασιάζεται με μία σταθερά α . Αυτή η σταθερά ρυθμίζει την ένταση της ομαλοποίησης. Μεγάλες τιμές της ενδέχεται να οδηγήσουν σε υποπροσαρμογή, ενώ μικρές ενδέχεται να μην μπορέσουν να αντιμετωπίσουν την υπερπροσαρμογή.

$$J(\boldsymbol{\theta}) = \frac{1}{m} \sum_{i=1}^m L(h_{\boldsymbol{\theta}}(\mathbf{x}), y^{(i)}) + \alpha R(\boldsymbol{\theta}) \quad (2.21)$$

Στην περίπτωση της ομαλοποίησης L2 έχουμε $R(\boldsymbol{\theta}) = \sum_{j=1}^n \theta_j^2$, ενώ σε αυτήν της ομαλοποίησης L1 έχουμε $R(\boldsymbol{\theta}) = \sum_{j=1}^n |\theta_j|$. Παρατηρήστε πως η συνάρτηση R δεν αφορά στην αποτέλεσμα θ_0 , καθώς αυτή δεν επηρεάζει την υπερπροσαρμογή του μοντέλου στα δεδομένα.

Η προσθήκη του όρου $\alpha R(\boldsymbol{\theta})$ στη συνάρτηση κόστους έχει ως αποτέλεσμα την αντίστοιχη προσθήκη της μερικής παραγώγου του όρου αυτού ως προς την εκάστοτε παράμετρο θ_j για $j > 0$. Στην περίπτωση της ομαλοποίησης L2, η παράγωγος είναι $2\alpha\theta_j$. Στην περίπτωση της ομαλοποίησης L1, η παράγωγος δεν ορίζεται για $\theta_j = 0$, ενώ για $\theta_j \neq 0$ ισούται με $\alpha \frac{\theta_j}{|\theta_j|}$. Μια αντιμετώπιση της ασυνέχειας συνίσταται στο να θέσουμε την τιμή της μερικής παραγώγου για την αντίστοιχη παράμετρο ίση με το μηδέν. Επομένως, μπορούμε συνολικά να θεωρήσουμε πως η παράγωγος ισούται με τη συνάρτηση $\alpha \text{sign}(\theta_j)$:

$$\text{sign}(\theta_j) = \begin{cases} -1, & \text{αν } \theta_j < 0 \\ 0, & \text{αν } \theta_j = 0 \\ 1, & \text{αν } \theta_j > 0. \end{cases} \quad (2.22)$$

Στην πράξη δεν θεωρείται η ίδια η σταθερά α ως υπερπαράμετρος στην (λογιστική) παλινδρόμηση, αλλά μια κανονικοποιημένη εκδοχή της σε σχέση με το πλήθος των παραδειγμάτων m . Συγκεκριμένα, στην ομαλοποίηση L2 θέτουμε $\alpha = \frac{\lambda}{2m}$ και η μερική παράγωγος του $\alpha R(\boldsymbol{\theta})$ ως προς την παράμετρο θ_j γίνεται $\frac{\lambda}{m}\theta_j$. Στην ομαλοποίηση L1 θέτουμε $\alpha = \frac{\lambda}{m}$ και η μερική παράγωγος του $\alpha R(\boldsymbol{\theta})$ ως προς την παράμετρο θ_j γίνεται $\frac{\lambda}{m} \text{sign}(\theta_j)$.

2.8 Περαιτέρω Μελέτη

Μια εξαιρετική παρουσίαση των γραμμικών μοντέλων με έμφαση στον αλγόριθμο της επικλινούς καθόδου μπορεί κανείς να παρακολουθήσει στο διαδικτυακό μάθημα επάνω στην Μηχανική Μάθηση του Andrew Ng³. Ένα βιβλίο που ακολουθεί την ίδια φιλοσοφία με το παραπάνω μάθημα είναι το [1]. Περισσότερες πληροφορίες για στατιστικές μεθόδους αναλυτικού υπολογισμού των παραμέτρων γραμμικών μοντέλων, μπορεί κανείς να βρει στα [2, 3].

2.9 Ασκήσεις

- Έστω στοχαστική επικλινής κάθοδος, με ρυθμό μάθησης $\eta = 0.1$, για την εκπαίδευση ενός γραμμικού μοντέλου παλινδρόμησης με δύο μεταβλητές εισόδου. Κάποια χρονική στιγμή της εκπαίδευσης οι παράμετροι έχουν τις τιμές $\theta_0 = 1, \theta_1 = -1, \theta_2 = 2$. Ποια η επόμενη τιμή των παραμέτρων αν στον αλγόριθμο δοθεί το παράδειγμα $(x, y) = ([0.5, 0.25], 3)$;
- Έστω το παρακάτω σύνολο εκπαίδευσης σε ένα πρόβλημα παλινδρόμησης. Οι παράμετροι ενός γραμμικού μοντέλου είναι $\theta_0 = 0$ και $\theta_1 = 1$. Ποιο είναι το κόστος του μοντέλου αυτού;

x_1	y
0	1
1	2
-1	0
0	1

- Έστω το παρακάτω σύνολο εκπαίδευσης σε ένα πρόβλημα γραμμικής παλινδρόμησης με δύο μεταβλητές εισόδου, x_1 και x_2 . Ποια η τιμή της μεταβλητής x_1 στο 4ο παράδειγμα, έπειτα από τον μετασχηματισμό των δεδομένων με την τεχνική της κλιμάκωσης;

x_1	x_2	y
10	-5	100
20	0	98
30	5	102
40	-2	87

- Έστω ένα πρόβλημα ταξινόμησης με 5 κλάσεις και ένα σύνολο εκπαίδευσης με 10, 20, 30, 40 και 50 παραδείγματα για κάθε κλάση αντίστοιχα. Εφαρμόζουμε τις μεθόδους μία εναντίων μίας και μία εναντίων όλων. Πόσα μοντέλα θα εκπαιδεύσουμε και πόσα παραδείγματα θα έχει το κάθε αντίστοιχο σύνολο εκπαίδευσης; Ποια μέθοδος περιλαμβάνει συνολικά το μεγαλύτερο άθροισμα παραδειγμάτων των αντίστοιχων συνόλων εκπαίδευσης;

Βιβλιογραφία

- [1] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

³<https://www.coursera.org/learn/machine-learning>

- [2] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer New York, 2006. ISBN: 978-0-387-31073-2.
- [3] Trevor Hastie, Robert Tibshirani και Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2016. ISBN: 978-0-387-84858-7.

ΚΕΦΑΛΑΙΟ 3

ΔΕΝΔΡΙΚΑ ΜΟΝΤΕΛΑ

Στο κεφάλαιο αυτό:

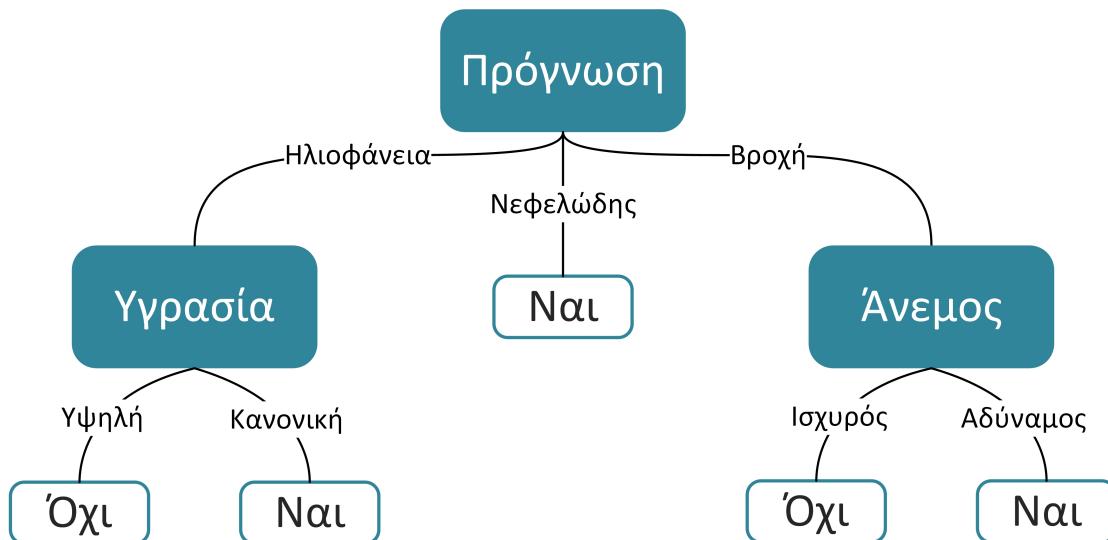
Θα μελετήσουμε την οικογένεια των δενδρικών μοντέλων. Αρχικά θα μελετήσουμε την αναπαράσταση των δενδρικών μοντέλων, και στη συνέχεια θα δούμε τρεις από τους πιο διαδεδομένους αλγορίθμους μάθησης δενδρικών μοντέλων, τους ID3, C4.5 και CART. Έπειτα θα συζητήσουμε τις τεχνικές που αξιοποιούνται στα δενδρικά μοντέλα προκειμένου να αντιμετωπιστεί το φαινόμενο της υπερπροσαρμογής. Τέλος θα αναφερθούμε στην έμφυτη ερμηνευσιμότητα των δενδρικών μοντέλων.

Τα δενδρικά μοντέλα είναι από τις πιο δημοφιλείς και πολύπλευρες οικογένειες μοντέλων μηχανικής μάθησης. Αποτελούν το βασικό συστατικό στοιχείο πολύ αποτελεσματικών τεχνικών για την παραγωγή και τον συγκερασμό συνόλων από μοντέλα πρόβλεψης¹, όπως το τυχαίο δάσος (random forest) και η επικλινής ενίσχυση (gradient boosting). Επιπλέον, τα δενδρικά μοντέλα είναι ερμηνεύσιμα, καθώς είναι ισοδυνάμα με ένα σύνολο κανόνων αν-τότε (if-then), με τους οποίους ο άνθρωπος είναι ιδιαίτερα εξοικειωμένος. Μπορούν να χρησιμοποιηθούν τόσο για προβλήματα ταξινόμησης, όσο και παλινδρόμησης, ακόμα και ομαδοποίησης. Τα δενδρικά μοντέλα ταξινόμησης ονομάζονται **δένδρα απόφασης** (decision trees).

3.1 Αναπαράσταση

Τα δενδρικά μοντέλα έχουν πάρει το όνομα τους από την αναπαράσταση τους. Αυτή έχει τη μορφή ενός δένδρου, στο οποίο οι εσωτερικοί κόμβοι αντιστοιχούν σε μεταβλητές εισόδου, τα κλαδιά αντιστοιχούν σε αμοιβαίως αποκλειόμενα υποσύνολα τιμών της μεταβλητής στην οποία αντιστοιχεί ο κόμβος από τον οποίο εκτείνονται, και τα φύλλα αντιστοιχούν σε προβλέψεις για την μεταβλητή εξόδου. Το Σχήμα 3.1 αναπαριστά ένα δένδρο απόφασης, για ένα πρόβλημα δυαδικής ταξινόμησης με μεταβλητές εισόδου την υγρασία (κανονική, υψηλή), τον άνεμο (αδύναμος, ισχυρός) και την πρόγνωση του καιρού (ηλιοφάνεια, συννεφιά, βροχή) για μια ημέρα και μεταβλητή εξόδου το αν αυτή η ημέρα είναι κατάλληλη για τένις (όχι, ναι).

¹ Δείτε κεφάλαιο 7.



Σχήμα 3.1: Ένα δένδρο απόφασης που ταξινομεί μια ημέρα ως προς την καταλληλότητα για την διεξαγωγή ενός αγώνα τένις σύμφωνα με τις καιρικές συνθήκες που επικρατούν.

Για να προβλέψουμε την μεταβλητή εξόδου για μια νέα περίπτωση, ξεκινάμε από τη ρίζα και ακολουθούμε το κλαδί που περιλαμβάνει την τιμή της περίπτωσης για την μεταβλητή εισόδου που αντιστοιχεί στη ρίζα. Συνεχίζουμε επαναληπτικά ώσπου να καταλήξουμε σε κάποιο φύλλο. Έστω για παράδειγμα μια νέα ημέρα, η οποία θέλουμε να δούμε αν είναι κατάλληλη για τένις, και για την οποία γνωρίζουμε ότι η υγρασία είναι υψηλή, ο άνεμος ισχυρός και η πρόγνωση του καιρού είναι βροχή. Ξεκινώντας από τη ρίζα θα ακολουθήσουμε το πρώτο από δεξιά κλαδί και στη συνέχεια το αριστερό κλαδί για να καταλήξουμε στο δεύτερο φύλλο από τα δεξιά, το οποίο προβλέπει την τιμή όχι. Τα δενδρικά μοντέλα μπορούν να αναπαρασταθούν ισοδύναμα ως ένα σύνολο από κανόνες αν-τότε, έναν για κάθε διαδρομή από τη ρίζα του δένδρου ως κάθε φύλλο. Οι συνθήκες των κανόνων αντιστοιχούν στους εσωτερικούς κόμβους του δένδρου και εκφράζουν περιορισμούς ως προς τις επιτρεπτές τιμές των μεταβλητών εισόδου και συνδέονται με σύζευξη, ενώ τα συμπεράσματα αντιστοιχούν σε φύλλα και εκφράζουν την τιμή της μεταβλητής εξόδου. Το δένδρο απόφασης του Σχήματος 3.1 αντιστοιχεί στο παρακάτω σύνολο κανόνων:

$$\begin{aligned}
 \text{Πρόγνωση} &= \text{Ηλιοφάνεια} \wedge \text{Υγρασία} = \text{Κανονική} \Rightarrow \text{Téniς} = \text{Ναι} \\
 \text{Πρόγνωση} &= \text{Ηλιοφάνεια} \wedge \text{Υγρασία} = \text{Υψηλή} \Rightarrow \text{Téniς} = \text{Όχι} \\
 \text{Πρόγνωση} &= \text{Συννεφιά} \Rightarrow \text{Téniς} = \text{Ναι} \\
 \text{Πρόγνωση} &= \text{Βροχή} \wedge \text{Άνεμος} = \text{Ισχυρός} \Rightarrow \text{Téniς} = \text{Όχι} \\
 \text{Πρόγνωση} &= \text{Βροχή} \wedge \text{Άνεμος} = \text{Αδύναμος} \Rightarrow \text{Téniς} = \text{Ναι}
 \end{aligned}$$

Στα δενδρικά μοντέλα, ο χώρος των υποθέσεων περιλαμβάνει όλα τα δένδρα που μπορούν να προκύψουν από ένα σύνολο δεδομένων εκπαίδευσης.

3.2 Αναζήτηση και Αξιολόγηση

Στην ενότητα αυτή θα μελετήσουμε τρεις από τους πιο διαδεδομένους αλγορίθμους μάθησης δενδρικών μοντέλων: τον ID3, τον C4.5 που αποτελεί επέκταση του ID3, και τον CART. Οι τρεις αυτοί αλγόριθμοι, καθώς και οι περισσότεροι αλγόριθμοι μάθησης δενδρικών μοντέλων, αποτελούν παραλλαγές ενός βασικού αλγορίθμου που εκτελεί αναρρίχηση λόφων στον χώρο όλων των δυνατών δένδρων. Ο αλγόριθμος αυτός ξεκινάει από ένα δένδρο ενός φύλλου (ρίζα), στο οποίο αρχικά βρίσκονται όλα τα δεδομένα εκπαίδευσης, και επεκτείνει επαναληπτικά κάθε φύλλο του δένδρου, εξετάζοντας ολοένα και μεγαλύτερα δένδρα. Η επέκταση ενός φύλλου συνίσταται στην επιλογή μιας μεταβλητής εισόδου, στην προσθήκη κόμβων παιδιών που αντιστοιχούν

σε υποσύνολα των τιμών της μεταβλητής αυτής, και στην διανομή των παραδειγμάτων εκπαίδευσης του φύλλου στα παιδιά του με βάση την τιμή που έχουν στην μεταβλητή αυτή. Η επιλογή της μεταβλητής και των υποσυνόλων των τιμών της γίνεται άπληστα με στόχο την μείωση της ανομοιογένειας των παραδειγμάτων του φύλλου. Μέσα από αυτή τη διαδικασία, το φύλλο μετατρέπεται σε εσωτερικό κόμβο και τα παιδιά του σε φύλλα.

3.2.1 ID3

Ο αλγόριθμος ID3 (Iterative Dichotomizer 3) είναι ένας από τους πρώτους και δημοφιλέστερους αλγορίθμους που αναπτύχθηκαν για τη μάθηση δένδρων απόφασης. Μπορεί να χειριστεί μόνο διακριτές μεταβλητές εισόδου. Για την επέκταση ενός φύλλου, υποψήφιες είναι όλες εκείνες οι μεταβλητές εισόδου που δεν έχουν χρησιμοποιηθεί ήδη για την επέκταση κάποιου προγόνου του. Για κάθε μία από αυτές τις μεταβλητές ο ID3 υπολογίζει την μείωση που θα επιτευχθεί στην ανομοιογένεια του τρέχοντος φύλλου, αν αυτό επεκταθεί προσθέτοντας τόσους κόμβους παιδιά όσες οι τιμές της μεταβλητής, και επιλέγει εκείνη την μεταβλητή που οδηγεί στην μεγαλύτερη μείωση.

Ως μετρική της ανομοιογένειας ενός συνόλου παραδειγμάτων ο ID3 χρησιμοποιεί την **εντροπία** (entropy). Έστω πρόβλημα ταξινόμησης με c κλάσεις, $\{0, 1, 2, \dots, c - 1\}$, σύνολο παραδειγμάτων S και υποσύνολο αυτών των παραδειγμάτων S_k που ανήκουν στην κλάση $k = 0, \dots, c - 1$.

$$\text{Entropy}(S) = \sum_{k=0}^{c-1} -\frac{|S_k|}{|S|} \log_2 \left(\frac{|S_k|}{|S|} \right) \quad (3.1)$$

Στους υπολογισμούς της εντροπίας θα θεωρούμε ότι $\log_2(0) = 0$. Ας δούμε ορισμένα παραδείγματα. Έστω πρόβλημα δυαδικής ταξινόμησης και φύλλο S με 8 παραδείγματα, 4 εκ των οποίων ανήκουν στην κλάση 0, και 4 στην κλάση 1. Σε αυτό το φύλλο η ανομοιογένεια είναι στο μέγιστο της.

$$\text{Entropy}(S) = -\frac{4}{8} \log_2 \left(\frac{4}{8} \right) - \frac{4}{8} \log_2 \left(\frac{4}{8} \right) = -\frac{1}{2}(-1) - \frac{1}{2}(-1) = 1.$$

Έστω πρόβλημα δυαδικής ταξινόμησης και φύλλο S με 8 παραδείγματα, εκ των οποίων όλα ανήκουν στην κλάση 1. Αυτό το φύλλο είναι απόλυτα ομοιογενές. Τέτοια φύλλα καλούνται αμιγή.

$$\text{Entropy}(S) = -\frac{0}{8} \log_2 \left(\frac{0}{8} \right) - \frac{8}{8} \log_2 \left(\frac{8}{8} \right) = -0 - 1 \log_2(1) = 0$$

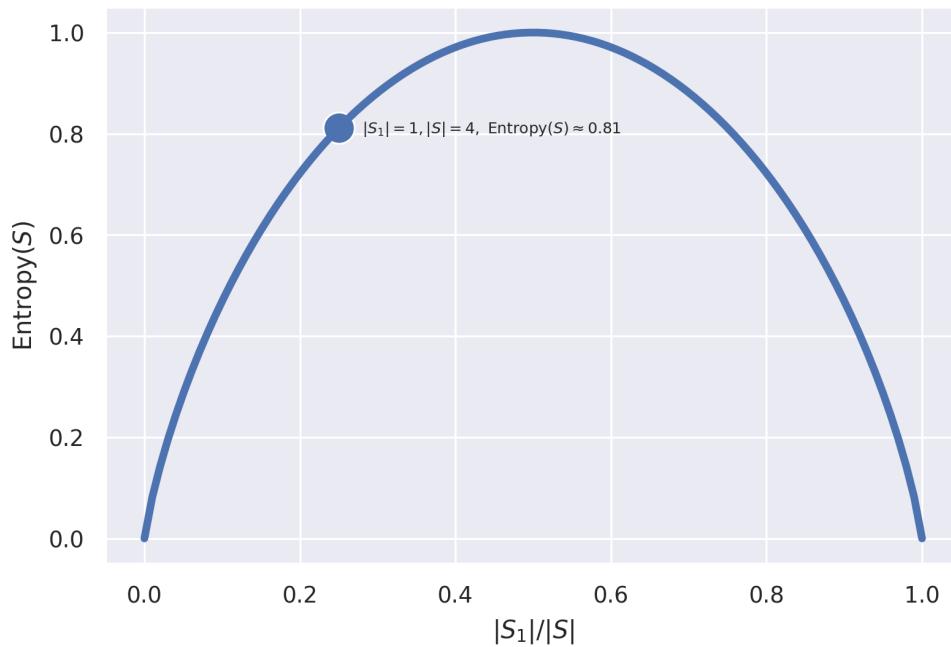
Έστω πρόβλημα δυαδικής ταξινόμησης και φύλλο S με 4 παραδείγματα, 3 εκ των οποίων ανήκουν στην κλάση 0, και 1 στην κλάση 1.

$$\text{Entropy}(S) = -\frac{3}{4} \log_2 \left(\frac{3}{4} \right) - \frac{1}{4} \log_2 \left(\frac{1}{4} \right) \approx -\frac{3}{4}(-0.415) - \frac{1}{4}(-2) \approx 0.811$$

Το Σχήμα 3.2 παρουσιάζει ένα γράφημα της τιμής της εντροπίας για ένα πρόβλημα δυαδικής ταξινόμησης, σε σχέση με το ποσοστό των παραδειγμάτων που ανήκουν στην κλάση 1. Βλέπουμε πως η εντροπία παίρνει την μέγιστη τιμή της, ίση με 1, όταν τα παραδείγματα είναι εξίσου μοιρασμένα στις 2 κλάσεις. Στο γράφημα βλέπουμε επίσης το σημείο το οποίο αντιστοιχεί στο τελευταίο από τα παραπάνω παραδείγματα.

Το Σχήμα 3.3 παρουσιάζει ένα γράφημα με την εντροπία για ένα πρόβλημα ταξινόμησης τριών κλάσεων σε σχέση με το ποσοστό των παραδειγμάτων που ανήκουν στις κλάσεις 0 και 1. Παρατηρούμε ότι η μέγιστη τιμή της εντροπίας μεγαλώνει με τον αριθμό των κλάσεων c , αφού όταν έχουμε ισοκατανεμημένα παραδείγματα στις κλάσεις, αυτή ισούται με $-\log_2(\frac{1}{c})$. Στο σχήμα αυτό προκύπτει για τιμές ίσες με $1/3$ στους άξονες του επιπέδου.

Ο αλγόριθμος ID3 υπολογίζει την μείωση της εντροπίας κατά την επέκταση ενός φύλλου. Αυτή η μείωση ονομάζεται **πληροφοριακό κέρδος** (information gain). Έστω ότι σε ένα φύλλο έχουμε ένα σύνολο S με παραδειγμάτα εκπαίδευσης (x, y) . Έστω μεταβλητή εισόδου A με σύνολο διακριτών τιμών $V(A)$. Για



Σχήμα 3.2: Γράφημα εντροπίας για δύο κλάσεις.

συγκεκριμένη τιμή $v \in V(A)$, ορίζουμε $S_v = \{(x, y) \in S | x_A = v\}$ ως το υποσύνολο του S με τιμή v στην μεταβλητή A . Το πληροφοριακό κέρδος από την επέκταση του φύλλου αυτού με βάση την μεταβλητή A ορίζεται ως εξής:

$$\text{InformationGain}(S, A) = \text{Entropy}(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v) \quad (3.2)$$

Η επαναληπτική διαδικασία επέκτασης του δένδρου συνεχίζεται μέχρι: α) να προκύψει αμιγής κόμβος (με παραδείγματα της ίδιας κλάσης), ο οποίος μετατρέπεται σε φύλλο που αντιστοιχεί στην κλάση των παραδειγμάτων του, β) να προκύψει κόμβος δίχως παραδείγματα εκπαίδευσης, ο οποίος μετατρέπεται σε φύλλο που αντιστοιχεί στην πιο συχνή κλάση στα παραδείγματα του γονέα του, γ) να έχουνε εξεταστεί όλες οι μεταβλητές εισόδου σε πρόγονους κόμβους, οπότε ο τρέχοντας κόμβος μετατρέπεται σε φύλλο που αντιστοιχεί στην πιο συχνή κλάση των παραδειγμάτων του. Στις περιπτώσεις (β) και (γ), από τη συχνότητα των κλάσεων μπορούμε να υπολογίσουμε και μια κατανομή πιθανοτήτων για τις κλάσεις.

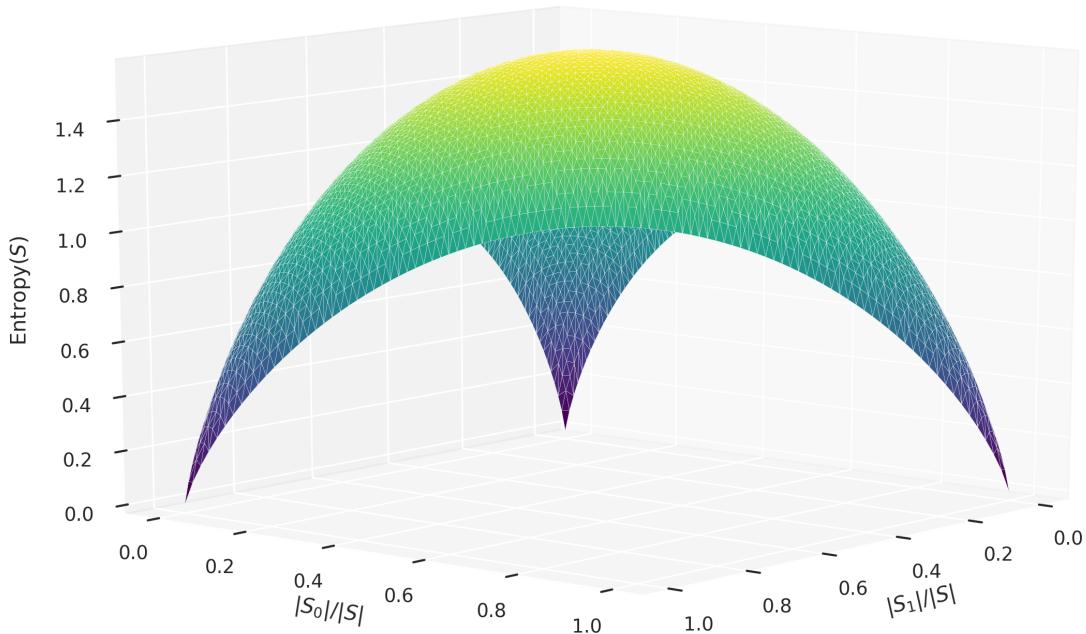
Έστω για παράδειγμα τα δεδομένα του Πίνακα 3.1, από τα οποία έχει προκύψει το δένδρο του Σχήματος 3.1 έπειτα από εφαρμογή του αλγορίθμου ID3. Ξεκινώντας από την ρίζα, έχουμε 9 (5) παραδείγματα της κλάσης Ναι ('Οχι), επομένως η εντροπία είναι:

$$\text{Entropy}(S) = -\frac{9}{14} \log_2 \left(\frac{9}{14} \right) - \frac{5}{14} \log_2 \left(\frac{5}{14} \right) \approx -0.643 (-0.637) - 0.357 (-1.485) \approx 0.940$$

Δεν υπάρχει πρόγονος κόμβος, επομένως θα υπολογίσουμε το πληροφοριακό κέρδος για όλες τις μεταβλητές εισόδου. Για την μεταβλητή πρόγνωση θα υπολογίσουμε την εντροπία στα υποσύνολα για τις τιμές ηλιοφάνεια, συννεφιά και βροχή. Για την ηλιοφάνεια έχουμε:

$$\text{Entropy}(S_H) = -\frac{2}{5} \log_2 \left(\frac{2}{5} \right) - \frac{3}{5} \log_2 \left(\frac{3}{5} \right) \approx -0.4 (-1.322) - 0.6 (-0.737) \approx 0.971 \quad (3.3)$$

Για την συννεφιά έχουμε:



Σχήμα 3.3: Γράφημα εντροπίας για τρεις κλάσεις.

Ημέρα	Πρόγνωση	Θερμοκρασία	Υγρασία	Άνεμος	Τένις
1η	Ηλιοφάνεια	Υψηλή	Υψηλή	Αδύναμος	Όχι
2η	Ηλιοφάνεια	Υψηλή	Υψηλή	Ισχυρός	Όχι
3η	Συννεφιά	Υψηλή	Υψηλή	Αδύναμος	Ναι
4η	Βροχή	Μέτρια	Υψηλή	Αδύναμος	Ναι
5η	Βροχή	Χαμηλή	Κανονική	Αδύναμος	Ναι
6η	Βροχή	Χαμηλή	Κανονική	Ισχυρός	Όχι
7η	Συννεφιά	Χαμηλή	Κανονική	Ισχυρός	Ναι
8η	Ηλιοφάνεια	Μέτρια	Υψηλή	Αδύναμος	Όχι
9η	Ηλιοφάνεια	Χαμηλή	Κανονική	Αδύναμος	Ναι
10η	Βροχή	Μέτρια	Κανονική	Αδύναμος	Ναι
11η	Ηλιοφάνεια	Μέτρια	Κανονική	Ισχυρός	Ναι
12η	Συννεφιά	Μέτρια	Υψηλή	Ισχυρός	Ναι
13η	Συννεφιά	Υψηλή	Κανονική	Αδύναμος	Ναι
14η	Βροχή	Μέτρια	Υψηλή	Ισχυρός	Όχι

Πίνακας 3.1: Ένα σύνολο παραδειγμάτων με τις καυρικές συνθήκες και την καταλληλότητα διεξαγωγής ενός αγώνα τένις.

$$\text{Entropy}(S_{\Sigma}) = -\frac{4}{4} \log_2 \left(\frac{4}{4} \right) - \frac{0}{4} \log_2 \left(\frac{0}{4} \right) = -0 - 0 = 0$$

Τέλος για την βροχή έχουμε:

$$\text{Entropy}(S_B) = -\frac{3}{5} \log_2 \left(\frac{3}{5} \right) - \frac{2}{5} \log_2 \left(\frac{2}{5} \right) \approx -0.6 (-0.737) - 0.4 (-1.322) \approx 0.971 \quad (3.4)$$

Επομένως το πληροφοριακό κέρδος αν επεκτείνουμε το δένδρο με βάση τη μεταβλητή πρόγνωση είναι:

$$\text{InformationGain}(S, \text{Πρόγνωση}) = 0.940 - \frac{5}{14} 0.971 - \frac{4}{14} 0 - \frac{5}{14} 0.971 \approx 0.940 - 0.694 = 0.247$$

Για την μεταβλητή θερμοκρασία θα υπολογίσουμε την εντροπία στα υποσύνολα για τις τιμές υψηλή, μέτρια και χαμηλή. Για την υψηλή έχουμε:

$$\text{Entropy}(S_T) = -\frac{2}{4} \log_2 \left(\frac{2}{4} \right) - \frac{2}{4} \log_2 \left(\frac{2}{4} \right) = -0.5 (-1) - 0.5 (-1) = 1$$

Για την μέτρια έχουμε:

$$\text{Entropy}(S_M) = -\frac{4}{6} \log_2 \left(\frac{4}{6} \right) - \frac{2}{6} \log_2 \left(\frac{2}{6} \right) \approx -0.667 (-0.585) - 0.333 (-1.585) \approx 0.918$$

Τέλος για την χαμηλή έχουμε:

$$\text{Entropy}(S_X) = -\frac{3}{4} \log_2 \left(\frac{3}{4} \right) - \frac{1}{4} \log_2 \left(\frac{1}{4} \right) \approx -0.75 (-0.415) - 0.25 (-2) \approx 0.811$$

Επομένως το πληροφοριακό κέρδος αν επεκτείνουμε το δένδρο με βάση τη μεταβλητή θερμοκρασία είναι:

$$\text{InformationGain}(S, \text{Θερμοκρασία}) = 0.940 - \frac{4}{14} 1 - \frac{6}{14} 0.918 - \frac{4}{14} 0.811 \approx 0.940 - 0.911 = 0.029$$

Για την μεταβλητή υγρασία θα υπολογίσουμε την εντροπία στα υποσύνολα για τις τιμές υψηλή και κανονική. Για την υψηλή έχουμε:

$$\text{Entropy}(S_T) = -\frac{3}{7} \log_2 \left(\frac{3}{7} \right) - \frac{4}{7} \log_2 \left(\frac{4}{7} \right) = -0.429 (-1.222) - 0.571 (-0.807) = 0.985$$

Για την κανονική έχουμε:

$$\text{Entropy}(S_K) = -\frac{6}{7} \log_2 \left(\frac{6}{7} \right) - \frac{1}{7} \log_2 \left(\frac{1}{7} \right) \approx -0.857 (-0.222) - 0.143 (-2.807) \approx 0.592$$

Επομένως το πληροφοριακό κέρδος αν επεκτείνουμε το δένδρο με βάση τη μεταβλητή υγρασία είναι:

$$\text{InformationGain}(S, \text{Υγρασία}) = 0.940 - \frac{7}{14} 0.985 - \frac{7}{14} 0.592 \approx 0.940 - 0.788 = 0.152$$

Τέλος για την μεταβλητή άνεμος θα υπολογίσουμε την εντροπία στα υποσύνολα για τις τιμές αδύναμος και ισχυρός. Για την αδύναμος έχουμε:

$$\text{Entropy}(S_A) = -\frac{6}{8} \log_2 \left(\frac{6}{8} \right) - \frac{2}{8} \log_2 \left(\frac{2}{8} \right) = -0.429 (-1.222) - 0.571 (-0.807) = 0.985$$

Για την ισχυρός έχουμε:

$$\text{Entropy}(S_I) = -\frac{3}{6} \log_2 \left(\frac{3}{6} \right) - \frac{3}{6} \log_2 \left(\frac{3}{6} \right) \approx -0.5(-1) - 0.5(-1) \approx 0.5 + 0.5 = 1$$

Επομένως το πληροφοριακό κέρδος αν επεκτείνουμε το δένδρο με βάση τη μεταβλητή άνεμος είναι:

$$\text{InformationGain}(S, \text{Άνεμος}) = 0.940 - \frac{8}{14} 0.985 - \frac{6}{14} 1 \approx 0.940 - 0.892 = 0.048$$

Παρατηρούμε ότι το μέγιστο πληροφοριακό κέρδος το έχουμε αν επιλέξουμε την μεταβλητή πρόγνωση για την ρίζα του δένδρου. Θα δημιουργήσουμε ένα παιδί για κάθε τιμή της μεταβλητής αυτής, θα διανείμουμε τα αντίστοιχα παραδείγματα εκπαίδευσης στο κάθε παιδί, και θα συνεχίσουμε την διαδικασία επέκτασης για κάθε ένα από αυτά τα παιδιά.

Στον κόμβο που αντιστοιχεί στην ήλιοφάνεια έχουμε 5 παραδείγματα εκπαίδευσης, εκ των οποίων τα 2 (3) θετικά (αρνητικά). Όπως είδαμε στην Εξίσωση 3.3, η εντροπία του κόμβου αυτού ισούται με 0.971. Θα υπολογίσουμε το πληροφοριακό κέρδος για όλες τις μεταβλητές εισόδου εκτός από την πρόγνωση που έχει ήδη εξεταστεί στον πρόγονο κόμβο που αντιστοιχεί στη ρίζα του δένδρου. Για την μεταβλητή θερμοκρασία θα υπολογίσουμε την εντροπία στα υποσύνολα για τις τιμές υψηλή, μέτρια και χαμηλή. Για την υψηλή έχουμε:

$$\text{Entropy}(S_Y) = -\frac{0}{2} \log_2 \left(\frac{0}{2} \right) - \frac{2}{2} \log_2 \left(\frac{2}{2} \right) = -0 - 0 = 0$$

Για την μέτρια έχουμε:

$$\text{Entropy}(S_M) = -\frac{1}{2} \log_2 \left(\frac{1}{2} \right) - \frac{1}{2} \log_2 \left(\frac{1}{2} \right) \approx -0.5(-1) - 0.5(-1) = 1$$

Τέλος για την χαμηλή έχουμε:

$$\text{Entropy}(S_X) = -\frac{1}{1} \log_2 \left(\frac{1}{1} \right) - \frac{0}{1} \log_2 \left(\frac{0}{1} \right) = -0 - 0 = 0$$

Επομένως το πληροφοριακό κέρδος αν επεκτείνουμε το δένδρο με βάση τη μεταβλητή θερμοκρασία είναι:

$$\text{InformationGain}(S, \text{Θερμοκρασία}) = 0.971 - \frac{2}{5} 0 - \frac{2}{5} 1 - \frac{1}{5} 0 = 0.971 - 0.4 = 0.571$$

Για την μεταβλητή υγρασία θα υπολογίσουμε την εντροπία στα υποσύνολα για τις τιμές υψηλή και κανονική. Για την υψηλή έχουμε:

$$\text{Entropy}(S_Y) = -\frac{0}{3} \log_2 \left(\frac{0}{3} \right) - \frac{3}{3} \log_2 \left(\frac{3}{3} \right) = -0 - 0 = 0$$

Για την κανονική έχουμε:

$$\text{Entropy}(S_K) = -\frac{2}{2} \log_2 \left(\frac{2}{2} \right) - \frac{0}{2} \log_2 \left(\frac{0}{2} \right) = -0 - 0 = 0$$

Επομένως το πληροφοριακό κέρδος αν επεκτείνουμε το δένδρο με βάση τη μεταβλητή υγρασία είναι:

$$\text{InformationGain}(S, \text{Υγρασία}) = 0.971 - \frac{3}{5} 0 - \frac{2}{5} 0 = 0.971$$

Τέλος για την μεταβλητή άνεμος θα υπολογίσουμε την εντροπία στα υποσύνολα για τις τιμές αδύναμος και ισχυρός. Για την αδύναμος έχουμε:

$$\text{Entropy}(S_A) = -\frac{1}{3} \log_2 \left(\frac{1}{3} \right) - \frac{2}{3} \log_2 \left(\frac{2}{3} \right) = -0.333 (-1.585) - 0.667 (-0.585) = 0.528$$

Για την ισχυρός έχουμε:

$$\text{Entropy}(S_I) = -\frac{1}{2} \log_2 \left(\frac{1}{2} \right) - \frac{1}{2} \log_2 \left(\frac{1}{2} \right) \approx -0.5(-1) - 0.5(-1) \approx 0.5 + 0.5 = 1$$

Επομένως το πληροφοριακό κέρδος αν επεκτείνουμε το δένδρο με βάση τη μεταβλητή άνεμος είναι:

$$\text{InformationGain}(S, \text{Άνεμος}) = 0.971 - \frac{3}{5}0.528 - \frac{2}{5}1 \approx 0.971 - 0.951 = 0.020$$

Παρατηρούμε ότι το μέγιστο πληροφοριακό κέρδος το έχουμε αν επιλέξουμε την μεταβλητή υγρασία για τον κόμβο αυτό. Θα δημιουργήσουμε ένα παιδί για κάθε τιμή της μεταβλητής αυτής και θα διανείμουμε τα αντίστοιχα παραδείγματα εκπαίδευσης στο κάθε παιδί. Παρατηρούμε ότι και οι δύο κόμβοι είναι αμιγείς, επομένως θα μετατραπούν σε φύλλα που θα αντιστοιχούν στις κλάσεις όχι και ναι, για τις τιμές υψηλή και κανονική της μεταβλητής υγρασία αντίστοιχα.

Στον κόμβο που αντιστοιχεί στην συννεφιά έχουμε 4 παραδείγματα, τα οποία όλα είναι θετικά, επομένως ο κόμβος αυτό θα μετατραπεί σε φύλλο που θα αντιστοιχεί στην κλάση ναι.

Τέλος στον κόμβο που αντιστοιχεί στην βροχή έχουμε 5 παραδείγματα εκπαίδευσης, εκ των οποίων τα 3 (2) θετικά (αρνητικά). Όπως είδαμε στην Εξίσωση 3.4, η εντροπία του κόμβου αυτού ισούται με 0.971. Θα υπολογίσουμε το πληροφοριακό κέρδος για όλες τις μεταβλητές εισόδου εκτός από την πρόγνωση που έχει ήδη εξεταστεί στον πρόγονο κόμβο που αντιστοιχεί στη ρίζα του δένδρου. Για την μεταβλητή θερμοκρασία θα υπολογίσουμε την εντροπία στα υποσύνολα για τις τιμές μέτρια και χαμηλή, καθώς για την τιμή υψηλή δεν έχουμε κανένα παράδειγμα. Για την μέτρια έχουμε:

$$\text{Entropy}(S_M) = -\frac{2}{3} \log_2 \left(\frac{2}{3} \right) - \frac{1}{3} \log_2 \left(\frac{1}{3} \right) \approx -0.667(-0.585) - 0.333(-1.585) = 0.918$$

Για την χαμηλή έχουμε:

$$\text{Entropy}(S_X) = -\frac{1}{2} \log_2 \left(\frac{1}{2} \right) - \frac{1}{2} \log_2 \left(\frac{1}{2} \right) = -0.5(-1) - 0.5(-1) \approx 0.5 + 0.5 = 1$$

Επομένως το πληροφοριακό κέρδος αν επεκτείνουμε το δένδρο με βάση τη μεταβλητή θερμοκρασία είναι:

$$\text{InformationGain}(S, \text{Θερμοκρασία}) = 0.971 - \frac{3}{5}0.918 - \frac{2}{5}1 = 0.971 - 0.951 = 0.020$$

Για την μεταβλητή υγρασία θα υπολογίσουμε την εντροπία στα υποσύνολα για τις τιμές υψηλή και κανονική. Για την υψηλή έχουμε:

$$\text{Entropy}(S_Y) = -\frac{1}{2} \log_2 \left(\frac{1}{2} \right) - \frac{1}{2} \log_2 \left(\frac{1}{2} \right) = -0.5(-1) - 0.5(-1) \approx 0.5 + 0.5 = 1$$

Για την κανονική έχουμε:

$$\text{Entropy}(S_K) = -\frac{2}{3} \log_2 \left(\frac{2}{3} \right) - \frac{1}{3} \log_2 \left(\frac{1}{3} \right) \approx -0.667(-0.585) - 0.333(-1.585) = 0.918$$

Επομένως το πληροφοριακό κέρδος αν επεκτείνουμε το δένδρο με βάση τη μεταβλητή υγρασία είναι:

$$\text{InformationGain}(S, \text{Υγρασία}) = 0.971 - \frac{2}{5}1 - \frac{3}{5}0.918 = 0.971 - 0.951 = 0.020$$

Τέλος για την μεταβλητή άνεμος θα υπολογίσουμε την εντροπία στα υποσύνολα για τις τιμές αδύναμος και ισχυρός. Για την αδύναμος έχουμε:

$$\text{Entropy}(S_A) = -\frac{3}{3} \log_2 \left(\frac{3}{3} \right) - \frac{0}{3} \log_2 \left(\frac{0}{3} \right) = -0 - 0 = 0$$

Για την ισχυρός έχουμε:

$$\text{Entropy}(S_I) = -\frac{0}{2} \log_2 \left(\frac{0}{2} \right) - \frac{2}{2} \log_2 \left(\frac{2}{2} \right) = -0 - 0 = 0$$

Επομένως το πληροφοριακό κέρδος αν επεκτείνουμε το δένδρο με βάση τη μεταβλητή άνεμος είναι:

$$\text{InformationGain}(S, \text{Άνεμος}) = 0.971 - \frac{3}{5}0 - \frac{2}{5}0 \approx 0.971 - 0 = 0.971$$

Παρατηρούμε ότι το μέγιστο πληροφοριακό κέρδος το έχουμε αν επιλέξουμε την μεταβλητή άνεμος για τον κόμβο αυτό. Θα δημιουργήσουμε ένα παιδί για κάθε τιμή της μεταβλητής αυτής και θα διανείμουμε τα αντίστοιχα παραδείγματα εκπαίδευσης στο κάθε παιδί. Παρατηρούμε ότι και οι δύο κόμβοι είναι αμιγείς, επομένως θα μετατραπούν σε φύλλα που θα αντιστοιχούν στις κλάσεις όχι και ναι, για τις τιμές ισχυρός και αδύναμος της μεταβλητής άνεμος αντίστοιχα.

Πλέον δεν απομένει άλλο φύλλο προς επέκταση αφού όλα είναι αμιγή, και έτσι έχουμε καταλήξει στο δένδρο του Σχήματος 3.1.

3.2.2 C4.5

Ο αλγόριθμος C4.5 αποτελεί επέκταση του ID3. Μια βασική διαφορά του από τον ID3 είναι ότι χρησιμοποιεί μια εναλλακτική συνάρτηση αξιολόγησης της μείωσης της ανομοιογένειας που καλείται λόγος κέρδους (gain ratio). Ο λόγος κέρδους ορίζεται με βάση μια βοηθητική συνάρτηση που καλείται πληροφορία διαχωρισμού (split information) και ορίζεται ως εξής:

$$\text{SplitInformation}(S, A) = - \sum_{v \in V(A)} \frac{|S_v|}{|S|} \log_2 \left(\frac{|S_v|}{|S|} \right)$$

Η συνάρτηση αυτή μετράει την εντροπία ενός συνόλου παραδειγμάτων αλλά σε σχέση με μια μεταβλητή εισόδου, αντί για την μεταβλητή εξόδου. Με βάση αυτήν τη συνάρτηση ο λόγος κέρδους ορίζεται ως εξής:

$$\text{GainRatio}(S, A) = \frac{\text{InformationGain}(S, A)}{\text{SplitInformation}(S, A)} \quad (3.5)$$

Η συνάρτηση αυτή προτάθηκε προκειμένου να αντιμετωπιστεί ένα μειονέκτημα του πληροφοριακού κέρδους, το οποίο έχει να κάνει με την προτίμηση μεταβλητών εισόδου με πολλές τιμές έναντι άλλων με λιγότερες, επειδή οι πρώτες οδηγούν από τη φύση τους σε μικρότερα υποσύνολα δεδομένων με χαμηλότερη εντροπία. Σαν ένα ακραίο παράδειγμα αναλογιστείτε να είχαμε ως μεταβλητή εισόδου τον αύξοντα αριθμό του κάθε παραδείγματος εκπαίδευσης. Η επέκταση της ρίζας με βάση αυτήν την μεταβλητή θα οδηγούσε σε μηδενισμό της εντροπίας, αλλά ταυτόχρονα σε ένα κοντό και φαρδύ δένδρο που δεν θα είχε καμία πρακτική χρησιμότητα.

Μια από τις σημαντικότερες δυνατότητες που προσθέτει ο C4.5 στον ID3 είναι αυτή του χειρισμού συνεχών μεταβλητών, για τις οποίες υπολογίζει την μείωση της ανομοιογένειας για μια επέκταση σε δύο παιδιά με βάση ένα αριθμητικό κατώφλι. Για την εύρεση του κατωφλίου αυτού, ταξινομούνται τα παραδείγματα του φύλλου σε αύξουσα σειρά και εξετάζονται ως υποψήφια κατώφλια εκείνα που ισούνται με τον μέσο όρο δύο γειτονικών στη σειρά παραδειγμάτων τα οποία ανήκουν σε διαφορετική κλάση.

Έστω για παράδειγμα ότι στα δεδομένα του Πίνακα 3.1, η μεταβλητή θερμοκρασία είναι συνεχής. Έστω επίσης ότι σε ένα φύλλο έχουμε τα 6 παραδείγματα εκπαίδευσης που βλέπουμε στον Πίνακα 3.2α. Ο Πίνακας 3.2β δείχνει τα ίδια δεδομένα, αλλά έπειτα από ταξινόμηση των τιμών της μεταβλητής θερμοκρασία σε αύξουσα

σειρά. Γειτονικά παραδείγματα με διαφορετική τιμή στην κλάση είναι το δεύτερο και το τρίτο, καθώς και το πέμπτο με το έκτο. Επομένως τα υποψήφια κατώφλια είναι 13 και 30 αντίστοιχα.

Θερμοκρασία	Τένις	Θερμοκρασία	Τένις
33	'Όχι	4	'Όχι
16	Ναι	10	'Όχι
10	'Όχι	16	Ναι
22	Ναι	22	Ναι
27	Ναι	27	Ναι
4	'Όχι	33	'Όχι

(a) Αρχικός πίνακας.

(β) Μετά την ταξινόμηση σε σειρά.

Πίνακας 3.2: Τιμές της μεταβλητής εισόδου θερμοκρασία και της μεταβλητής εξόδου σε ένα φύλλο.

Μια επιπλέον δυνατότητα που υποστηρίζει ο C4.5 σε σχέση με τον ID3, είναι ο χειρισμός παραδειγμάτων εκπαίδευσης και περιπτώσεων με ελλιπείς τιμές για μία ή παραπάνω μεταβλητές εισόδου. Έστω για παράδειγμα ότι θέλουμε να φτιάξουμε ένα δένδρο απόφασης για τη διάγνωση μιας ασθένειας και ότι μία από τις μεταβλητές εισόδου αφορά στην τιμή της κακής χοληστερίνης. Αυτή η τιμή μπορεί να μην είναι διαθέσιμη σε κάποια από τα παραδείγματα εκπαίδευσης ή σε μια νέα περίπτωση που θέλουμε να ταξινομήσουμε με ένα εκπαιδευμένο δένδρο. Αυτό είναι ένα γενικότερο πρόβλημα στη μηχανική μάθηση.

Ένας πρώτος τρόπος να αντιμετωπιστεί αυτό κατά την εκπαίδευση ενός δενδρικού μοντέλου είναι να υποθέσουμε πως η τιμή που λείπει ισούται με την πιο κοινή τιμή (για διακριτές μεταβλητές) ή τον μέσο όρο των τιμών (για συνεχείς μεταβλητές) στα παραδείγματα εκπαίδευσης του τρέχοντος κόμβου του δένδρου. Εναλλακτικά μπορούμε να εστιάσουμε στο υποσύνολο των παραδειγμάτων εκπαίδευσης του τρέχοντος κόμβου που έχουν την ίδια κλάση με το παράδειγμα στο οποίο λείπει η τιμή.

Ο τρόπος που ο C4.5 χειρίζεται μεταβλητές με ελλιπείς τιμές έχει ως εξής. Καταρχάς, από τον υπολογισμό του κέρδους πληροφορίας για μια μεταβλητή εισόδου, εξαιρούνται τα παραδείγματα στα οποία η τιμή της μεταβλητής αυτής λείπει. Τόσο η εντροπία του τρέχοντος κόμβου, όσο και η εντροπία των παιδιών υπολογίζεται με βάση μόνο τα παραδείγματα που έχουν τιμή στην μεταβλητή ενδιαφέροντος. Το πληροφοριακό κέρδος που θα υπολογιστεί με αυτόν τον τρόπο σταθμίζεται έπειτα με το ποσοστό των παραδειγμάτων του κόμβου στα οποία δεν λείπει τιμή για τη συγκεκριμένη μεταβλητή.

Έστω για παράδειγμα η δυαδική μεταβλητή εισόδου υψηλός πυρετός, για την οποία σε έναν κόμβο έχουμε 8 παραδείγματα με τιμή ναι, 2 παραδείγματα με τιμή όχι, και 2 παραδείγματα στα οποία η τιμή λείπει. Έστω δυαδική μεταβλητή εξόδου Covid-19, η οποία έχει τιμή ναι στα 4 από τα 8 παραδείγματα όπου η μεταβλητή υψηλός πυρετός έχει την τιμή ναι, σε κανένα από τα 2 παραδείγματα που έχει την τιμή όχι και σε 1 από τα 2 παραδείγματα στα οποία η τιμή λείπει. Η εντροπία του κόμβου θα είναι:

$$\text{Entropy}(S) = -\frac{4}{10} \log_2 \left(\frac{4}{10} \right) - \frac{6}{10} \log_2 \left(\frac{6}{10} \right) = 0.971$$

Για την τιμή ναι της μεταβλητής υψηλός πυρετός η εντροπία είναι:

$$\text{Entropy}(S_N) = -\frac{4}{8} \log_2 \left(\frac{4}{8} \right) - \frac{4}{8} \log_2 \left(\frac{4}{8} \right) = 1$$

Για την τιμή όχι της μεταβλητής υψηλός πυρετός η εντροπία είναι:

$$\text{Entropy}(S_O) = -\frac{0}{2} \log_2 \left(\frac{0}{2} \right) - \frac{2}{2} \log_2 \left(\frac{2}{2} \right) = 0$$

Άρα το τελικό πληροφοριακό κέρδος είναι:

$$\frac{10}{12} \text{InformationGain}(S, \text{Υψηλός Πυρετός}) = \frac{10}{12} \left[0.971 - \frac{8}{10}1 - \frac{2}{10}0 \right] = \frac{10}{12} 0.171 = 0.143$$

Επιπλέον στον υπολογισμό της πληροφορίας διαχωρισμού, τα παραδείγματα με ελλιπείς τιμές θεωρούνται ότι ανήκουν σε μια νέα ξεχωριστή διακριτή τιμή:

$$\text{SplitInformation}(S, \text{Υψηλός Πυρετός}) = -\frac{8}{12} \log_2 \left(\frac{8}{12} \right) - \frac{2}{12} \log_2 \left(\frac{2}{12} \right) - \frac{2}{12} \log_2 \left(\frac{2}{12} \right)$$

Αν μια μεταβλητή με ελλιπείς τιμές επιλεγεί σε έναν κόμβο ως βέλτιστη, τα παραδείγματα με τις ελλιπείς τιμές διανέμονται σε όλα τα παιδιά, σταθμισμένα με το ποσοστό των αντίστοιχων παραδειγμάτων χωρίς ελλιπείς τιμές. Αν στο παραπάνω παράδειγμα μας επιλεγεί η μεταβλητή υψηλός πυρετός για τον κόμβο, τότε τα 2 παραδείγματα με ελλιπείς τιμές για την μεταβλητή αυτή θα διανεμηθούν και στα 2 παιδιά. Στο παιδί που αφορά την τιμή *nai* θα σταθμιστούν με 0.8 και στο παιδί που αφορά την τιμή *όχι* θα σταθμιστούν με 0.2.

Κατά την χρήση του δένδρου για την πρόβλεψη μιας νέας περίπτωσης με ελλιπείς τιμές, αν η περίπτωση δεν έχει τιμή για την μεταβλητή του τρέχοντα κόμβου του δένδρου, τότε η περίπτωση θα διανεμηθεί σε όλα τα παιδιά σταθμισμένη με τον ίδιο τρόπο, όπως αναφέραμε και προηγουμένως. Η τελική κατανομή πιθανοτήτων για την περίπτωση αυτή θα προκύψει από το σταθμισμένο άθροισμα των κατανομών πιθανοτήτων όλων των φύλλων στα οποία θα καταλήξει η περίπτωση.

3.2.3 CART

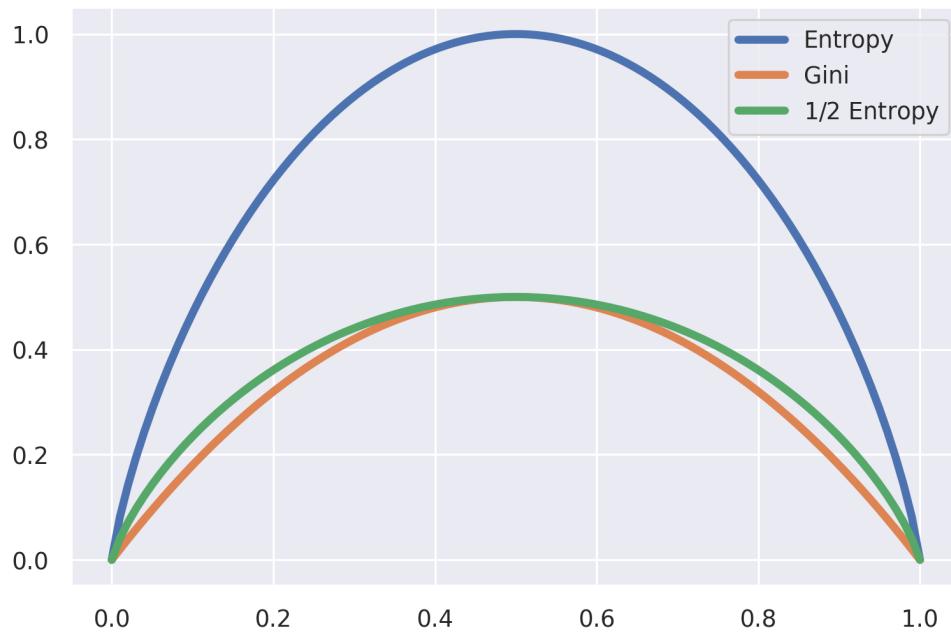
Ο αλγόριθμος CART (Classification And Regression Trees) μπορεί να χειριστεί τόσο διακριτές όσο και συνεχείς μεταβλητές. Στη θέση της εντροπίας, χρησιμοποιεί την συνάρτηση Gini, η οποία έχει παρόμοια συμπεριφορά με την εντροπία, όπως μπορούμε να δούμε στο Σχήμα 3.4. Η συνάρτηση Gini ορίζεται ως εξής:

$$\text{Gini}(S) = 1 - \sum_{k=0}^{c-1} \left(\frac{|S_k|}{|S|} \right)^2 \quad (3.6)$$

Ο αλγόριθμος CART μπορεί να χρησιμοποιηθεί και σε προβλήματα παλινδρόμησης, χρησιμοποιώντας ως μετρική ανομοιογένειας τη διακύμανση της μεταβλητής εξόδου των παραδειγμάτων στα φύλλα:

$$\text{Var}(S) = \frac{1}{|S|} \sum_{i=1}^{|S|} (y^{(i)} - \bar{y})^2 \quad \bar{y} = \frac{1}{|S|} \sum_{i=1}^{|S|} y^{(i)} \quad (3.7)$$

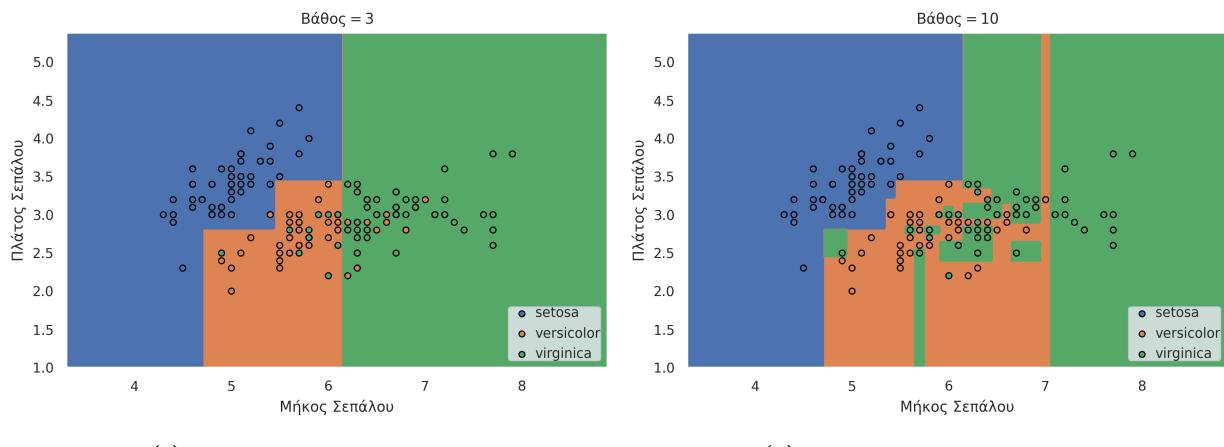
Στον αλγόριθμο CART η επέκταση ενός φύλλου οδηγεί πάντοτε σε δύο παιδιά. Στην περίπτωση των δυαδικών διακριτών και των συνεχών μεταβλητών εισόδου, η επέκταση δεν διαφέρει από αυτό που έχουμε ήδη δει για τον C4.5. Στην περίπτωση όμως διακριτών μεταβλητών με 3 ή παραπάνω τιμές, τότε θα πρέπει να εξεταστούν όλοι οι συνδυασμοί των τιμών αυτών σε δύο υποσύνολα. Για μικρό αριθμό τιμών, αυτό θα μπορούσε να γίνει εξαντλητικά. Για μεγάλο όμως αριθμό τιμών, το υπολογιστικό κόστος γίνεται απαγορευτικό. Σε αυτές τις περιπτώσεις, αν η μεταβλητή εξόδου είναι δυαδική, τότε οι τιμές της μεταβλητής εισόδου μπορούν να ταξινομηθούν σε σειρά με βάση το πλήθος των παραδειγμάτων που ανήκουν σε μία από τις δύο κλάσεις. Στη συνέχεια εξετάζουμε υποψήφιους διαχωρισμούς σε δύο υποσύνολα με αυτήν την σειρά. Η ίδια διαδικασία ακολουθείται και για συνεχή μεταβλητή εξόδου, έπειτα από ταξινόμηση σε σειρά των τιμών της μεταβλητής εισόδου με βάση την μέση τιμή της μεταβλητής εξόδου στα αντίστοιχα παραδείγματα. Στην περίπτωση διακριτής μεταβλητής εξόδου με παραπάνω από 2 τιμές, δεν μπορούν να ακολουθηθούν αντίστοιχες διαδικασίες απλοποίησης του υπολογιστικού κόστους.



Σχήμα 3.4: Σύγκριση εντροπίας, Gini και το 1/2 της εντροπίας για δύο κλάσεις

3.3 Υπερπροσαρμογή

Όσο περισσότερο επεκτείνεται ένα δένδρο, τόσο πιο πολύπλοκο γίνεται με κίνδυνο να προσαρμοστεί υπερβολικά στα δεδομένα εκπαίδευσης. Το Σχήμα 3.5 συγκρίνει την επιφάνεια απόφασης δύο δένδρων με διαφορετικά μέγιστα βάθη. Τα δένδρα αυτά έχουν εκπαιδευτεί στο σύνολο δεδομένων Iris², ένα διάσημο σύνολο δεδομένων που αφορά στην ταξινόμηση ανθών από λουλούδια του γένους Iris σε 3 διαφορετικά είδη (Setosa, Versicolor, Virginica) με βάση το μήκος και το πλάτος των σεπάλων (κάλυκας) και των πετάλων (στεφάνη) τους. Το σύνολο αυτό περιλαμβάνει 50 παραδείγματα από την κάθε κλάση. Παρατηρούμε πως το δένδρο με το μεγαλύτερο βάθος (αυτό στα δεξιά) προσαρμόζεται υπερβολικά στα δεδομένα. Για την αντιμετώπιση του φαινομένου αυτού, είτε αφήνουμε το δένδρο να επεκταθεί και στη συνέχεια το κλαδεύουμε, είτε σταματάμε νωρίτερα την επέκτασή του.



(α) Μέγιστο βάθος δένδρου 3.

(β) Μέγιστο βάθος δένδρου 10.

Σχήμα 3.5: Επιφάνεια απόφασης δένδρων με διαφορετικό μέγιστο βάθος για δύο χαρακτηριστικά του συνόλου δεδομένων Iris.

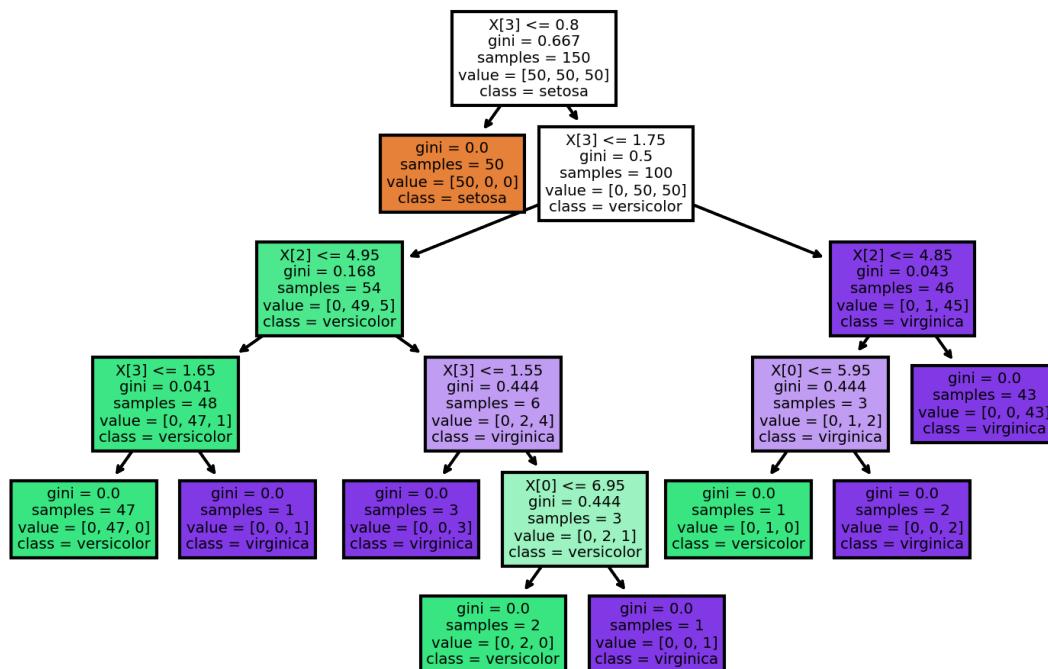
²<https://archive.ics.uci.edu/ml/datasets/iris>

Η επέκταση του μοντέλου μπορεί να τερματιστεί νωρίτερα από τις κλασικές τερματικές του συνθήκες ορίζοντας ένα μέγιστο βάθος δένδρου. Επιπλέον μπορούμε να απαιτήσουμε έναν ελάχιστο αριθμό από παραδείγματα εκπαίδευσης σε έναν κόμβο προκειμένου να συνεχιστεί η επέκταση του ή να απαιτήσουμε έναν ελάχιστο αριθμό από παραδείγματα στα φύλλα. Επιπρόσθετα, μπορούμε να απαιτήσουμε η μείωση της εντροπίας σε έναν κόμβο να είναι μεγαλύτερη από ένα ελάχιστο κατώφλι για να συνεχιστεί η επέκταση του. Τέλος θα μπορούσαμε να ορίσουμε έναν μέγιστο αριθμό από φύλλα. Στην περίπτωση αυτή το δένδρο αναπτύσσεται συνήθως επεκτείνοντας κάθε φορά εκείνον τον κόμβο που μειώνει περισσότερο την ανομοιογένεια (best-first).

Η μέθοδος reduced-error pruning, προϋποθέτει την ύπαρξη ενός ξεχωριστού συνόλου δεδομένων, που καλείται σύνολο επικύρωσης. Αφού επεκταθεί κανονικά το δένδρο, κάθε εσωτερικός κόμβος του θεωρείται υποψήφιος για κλάδεμα. Το κλάδεμα ενός εσωτερικού κόμβου επιφέρει σαν αποτέλεσμα την αντικατάσταση του κόμβου, μαζί με το υποδένδρο που κρέμεται από αυτόν, με ένα φύλλο. Οι εσωτερικοί κόμβοι κλαδεύονται ένας ένας επαναληπτικά, επιλέγοντας κάθε φορά εκείνον που οδηγεί στην μεγαλύτερη μείωση του σφάλματος του μοντέλου στα δεδομένα επικύρωσης, μέχρις ότου το κλάδεμα ενός εσωτερικού κόμβου να οδηγήσει σε αύξηση του σφάλματος στα δεδομένα επικύρωσης.

3.4 Ερμηνευσιμότητα

Τα δενδρικά μοντέλα μπορούν να οπτικοποιηθούν, προκειμένου να μελετήσουμε την γνώση που αναπαριστούν. Ωστόσο, όσο μεγαλύτερο το βάθος του δένδρου, τόσο δυσκολότερο για έναν άνθρωπο να το παρακολουθήσει. Δείτε για παράδειγμα πόσο δύσκολη είναι η κατανόηση του δενδρικού μοντέλου που παρουσιάζει το Σχήμα 3.6.

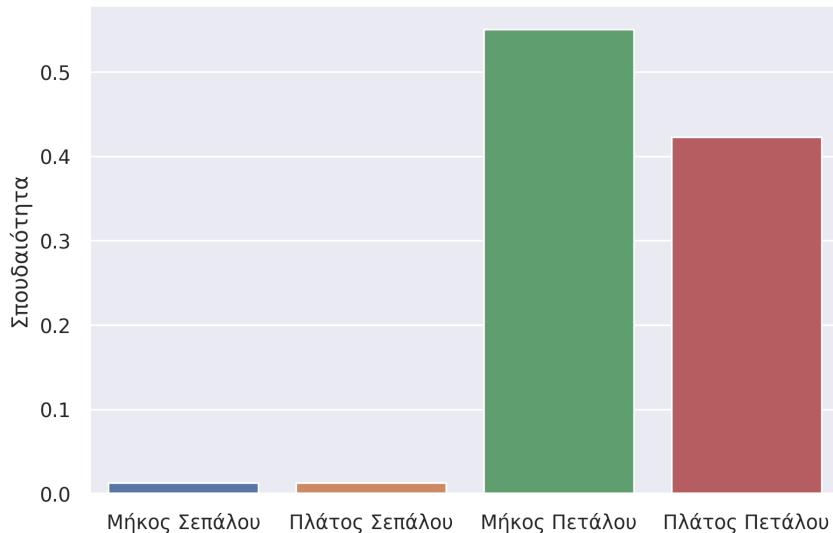


Σχήμα 3.6: Οπτικοποίηση δένδρου απόφασης για το σύνολο δεδομένων Iris.

Σε τέτοιες περιπτώσεις, μπορούμε να υπολογίσουμε μια περίληψη του δένδρου με τη μορφή ενός σκορ σημαντικότητας για κάθε μεταβλητή εισόδου. Συγκεκριμένα, υπολογίζουμε το ποσοστό της μείωσης της ανομοιογένειας των παραδειγμάτων εκπαίδευσης από το συνολικό δένδρο που οφείλεται σε κάθε κόμβο, πολλαπλασιάζοντας την μείωση της εντροπίας από την επέκταση του συγκεκριμένου κόμβου με το ποσοστό των παραδειγμάτων του. Για παράδειγμα, στον αλγόριθμο ID3, το ποσοστό αυτό θα υπολογιζόταν ως εξής:

$$\frac{|S|}{m} \left(\text{Entropy}(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v) \right)$$

Στη συνέχεια αθροίζουμε τις τιμές αυτές για τους κόμβους που αφορούν στην ίδια μεταβλητή εισόδου. Τέλος κανονικοποιούμε τις τιμές αυτές ώστε να είναι μεταξύ 0 και 1 και να αθροίζουν στην μονάδα, διαιρώντας με το άθροισμα των τιμών αυτών για όλες τις μεταβλητές εισόδου. Το Σχήμα 3.7 δείχνει την σημαντικότητα των μεταβλητών εισόδου στο σύνολο δεδομένων Iris, όπως αυτή προκύπτει από ένα δενδρικό μοντέλο.



Σχήμα 3.7: Σημαντικότητα μεταβλητών εισόδου στο σύνολο δεδομένων Iris.

3.5 Περαιτέρω Μελέτη

Για περισσότερη μελέτη γύρω από τις βασικές έννοιες των δενδρικών μοντέλων και τους αλγορίθμους ID3 και C4.5 μπορεί κανείς να διαβάσει περισσότερα στο [1]. Ένας πλήρης οδηγός του αλγορίθμου C4.5 από τον δημιουργό του αλγορίθμου, τον Ross Quinlan, βρίσκεται στο [2]. Για τον αλγόριθμο CART περισσότερες λεπτομέρειες δίνονται στο ομώνυμο βιβλίο από τους δημιουργούς του [3]. Μια πιο συνοπτική περιγραφή του CART υπάρχει στην ενότητα 9.2 του [4].

3.6 Ασκήσεις

1. Έστω το παρακάτω σύνολο εκπαίδευσης. Ποιο χαρακτηριστικό θα επιλεγεί στη ρίζα του δένδρου από τον αλγόριθμο ID3;

x_1	x_2	x_3	x_4	x_5	y
a	b	a	a	a	0
a	a	b	a	b	1
b	b	c	a	b	0
b	c	a	b	c	1
c	b	b	c	c	0

2. Έστω το παρακάτω σύνολο εκπαίδευσης ενός δενδρικού μοντέλου. Με βάση ποια τιμή θα γίνει ο πρώτος έλεγχος στο δένδρο;

x_1	y
2	0
4	1
6	0
8	0
10	0
12	1

3. Έστω πρόβλημα δυαδικής ταξινόμησης με δύο δυαδικές μεταβλητές εισόδου και το ακόλουθο σύνολο εκπαίδευσης. Ποια μεταβλητή εισόδου θα ελεγχθεί στην κορυφή του δένδρου απόφασης που θα προκύψει εφαρμόζοντας τον αλγόριθμο ID3; Ποιο το αντίστοιχο πληροφοριακό κέρδος (information gain);

x_1	x_2	y
1	1	1
1	1	1
1	0	0
0	0	1
0	1	0
0	1	0

4. Έστω το παρακάτω σύνολο δεδομένων. Η εντροπία των παραδειγμάτων με $x_2 = 0$ είναι 0 και η εντροπία των παραδειγμάτων με $x_1 = 0$ είναι 1. Επίσης, είναι γνωστό ότι ο αλγόριθμος ID3, βρίσκει τη x_2 να είναι καταλληλότερη μεταβλητή για τον έλεγχο στη ρίζα. Η μεταβλητή εξόδου είναι δίτιμη. Συμπληρώστε τις τιμές που λείπουν από τον πίνακα, αιτιολογώντας την απάντησή σας.

x_1	x_2	y
1	0	
1	1	
0	0	0
0	1	

5. Έστω ότι έχουμε το παρακάτω σύνολο δεδομένων που αφορά την πρόβλεψη αν ένα άτομο θα αγοράσει ή όχι ένα προϊόν, με βάση την ηλικία και την εισόδηματική του κατάσταση:

Ηλικία	Εισόδημα	Θα αγοράσει
25	Υψηλό	'Οχι
30	Χαμηλό	'Οχι
35	Υψηλό	'Οχι
40	Χαμηλό	'Οχι
45	Υψηλό	Ναι
50	Χαμηλό	'Οχι
55	Υψηλό	Ναι
60	Χαμηλό	Ναι

- (α) Υπολογίστε την εντροπία του αρχικού συνόλου δεδομένων (δηλαδή την εντροπία για την έξοδο “Θα αγοράσει”).
- (β) Υπολογίστε την εντροπία για κάθε χαρακτηριστικό (Ηλικία, Εισόδημα).
- (γ) Χρησιμοποιήστε την έννοια του Gain Ratio για να αποφασίσετε ποιο χαρακτηριστικό (Ηλικία ή Εισόδημα) είναι το καλύτερο για τη διαίρεση του συνόλου δεδομένων.
- (δ) Με βάση ποιο χαρακτηριστικό θα γίνει ο διαχωρισμός στη ρίζα του δέντρου;

Βιβλιογραφία

- [1] Tom M. Mitchell. *Machine learning, International Edition*. McGraw-Hill Series in Computer Science. McGraw-Hill, 1997. ISBN: 978-0-07-042807-2.
- [2] Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [3] Leo Breiman, Jerome H Friedman, R A Olshen και Charles J Stone. *Classification and Regression Trees*. Chapman και Hall/CRC, 1984, σ. 368. ISBN: 0412048418.
- [4] Trevor Hastie, Robert Tibshirani και Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2016. ISBN: 978-0-387-84858-7.

ΚΕΦΑΛΑΙΟ 4

ΑΞΙΟΛΟΓΗΣΗ ΜΟΝΤΕΛΩΝ

Στο κεφάλαιο αυτό:

Θα ασχοληθούμε με την αξιολόγηση μοντέλων μηχανικής μάθησης. Αρχικά θα δούμε τις διαδικασίες που ακολουθούμε και έπειτα τις μετρικές που χρησιμοποιούμε για την αξιολόγηση μοντέλων.

Η αξιολόγηση των μοντέλων μηχανικής μάθησης είναι ιδιαίτερα σημαντική για την επίτευξη πραγματικής προόδου στη μηχανική μάθηση. Μας δίνει τη δυνατότητα να συγκρίνουμε δύο ή παραπάνω μοντέλα, καθώς και να επιλέξουμε τις καλύτερες υπερ-παραμέτρους για έναν αλγόριθμο. Αυτή η διαδικασία ονομάζεται **επιλογή μοντέλου** (model selection). Από την άλλη εξίσου σημαντικό είναι να μπορέσουμε να έχουμε μια εκτίμηση της επίδοσης που θα έχει ένα μοντέλο σε άγνωστα δεδομένα. Αυτό καλείται **εκτίμηση μοντέλου** (model assessment). Και στις δύο περιπτώσεις θα πρέπει να χρησιμοποιήσουμε κάποια συγκεκριμένη μετρική αξιολόγησης. Στις δύο ενότητες που ακολουθούν, θα δούμε αρχικά διάφορες διαδικασίες που ακολουθούμε για την αξιολόγηση μοντέλου και έπειτα διάφορες μετρικές αξιολόγησης.

4.1 Διαδικασίες Αξιολόγησης

Ένας πρώτος τρόπος με τον οποίο θα μπορούσε κανείς να αξιολογήσει ένα μοντέλο μηχανικής μάθησης, είναι εξετάζοντας την ακρίβεια του στο **σύνολο εκπαίδευσης** (training set). Το σφάλμα ενός μοντέλου στα δεδομένα από τα οποία εκπαίδευτηκε καλείται **σφάλμα επαναντικατάστασης** (resubstitution error). Προφανώς, το σφάλμα αυτό θα είναι μια ιδιαίτερα αισιόδοξη εκτίμηση του σφάλματος που θα κάνει το μοντέλο σε νέα δεδομένα, τα οποία δεν έχει ξαναδεί. Επιπλέον, το σφάλμα αυτό δεν μπορεί να χρησιμοποιηθεί για επιλογή μοντέλου, καθώς θα προτιμήσει μοντέλα που υπερπροσαρμόζονται στα δεδομένα εκπαίδευσης.

Για τον λόγο αυτό, συνήθως χωρίζουμε το αρχικό σύνολο δεδομένων εκπαίδευσης σε δύο ξένα μεταξύ τους υποσύνολα, ένα σύνολο εκπαίδευσης και ένα **σύνολο ελέγχου** (test set). Το σύνολο ελέγχου μπορεί να χρησιμοποιηθεί είτε για εκτίμηση μοντέλου, είτε για επιλογή μοντέλου. Δεν πρέπει ωστόσο να χρησιμοποιείται ταυτόχρονα και για τους δύο αυτούς σκοπούς, γιατί ελλοχεύει το πρόβλημα της **υπερπροσαρμογής κατά την επιλογή μοντέλου** (overfitting in model selection): όσο περισσότερα τα διαφορετικά μοντέλα μεταξύ

των οποίων επιλέγουμε, και όσο λιγότερα τα δεδομένα ελέγχου, τόσο αυξάνει η πιθανότητα το καλύτερο μοντέλο με βάση το σύνολο ελέγχου να έτυχε να ταιριάζει καλύτερα στο συγκεκριμένο σύνολο ελέγχου. Αυτό θα έχει ως αποτέλεσμα το σφάλμα που καταγράψαμε να είναι υποεκτίμηση του σφάλματος σε άλλα άγνωστα δεδομένα.

Έστω για παράδειγμα ένας αλγόριθμος μάθησης δενδρικών μοντέλων, για τον οποίο θέλουμε να αποφασίσουμε την υπερ-παράμετρο του μέγιστου βάθους μεταξύ των τιμών 2 έως 10 και επιπλέον να εκτιμήσουμε την επίδοσή του σε άγνωστα δεδομένα. Η αφελής προσέγγιση περιλαμβάνει την εκπαίδευση 9 δενδρικών μοντέλων, ένα για κάθε διαφορετικό μέγιστο βάθος χρησιμοποιώντας το σύνολο εκπαίδευσης, και την καταγραφή της επίδοσης αυτών των μοντέλων στο σύνολο ελέγχου. Έστω ότι βέλτιστη επίδοση έδωσε το δένδρο με μέγιστο βάθος 5 (επιλογή μοντέλου). Είναι επικίνδυνο να πούμε ότι η επίδοση που παρατηρήσαμε για αυτό το μοντέλο, θα είναι και αυτή που θα πετύχει το μοντέλο όταν εφαρμοστεί σε άγνωστα δεδομένα (εκτίμηση μοντέλου), λόγω της πιθανής υπερπροσαρμογής κατά την επιλογή μοντέλου.

Η λύση στο παραπάνω πρόβλημα ακούει στο όνομα **σύνολο επικύρωσης** (validation set). Το αρχικό σύνολο δεδομένων εκπαίδευσης το χωρίζουμε σε τρία ξένα μεταξύ τους υποσύνολα, ένα σύνολο εκπαίδευσης, ένα σύνολο επικύρωσης και ένα σύνολο ελέγχου. Το σύνολο επικύρωσης χρησιμοποιείται για την επιλογή μοντέλου, ενώ το σύνολο ελέγχου για την εκτίμηση μοντέλου. Υιοθετώντας αυτήν την προσέγγιση στο προηγούμενο παράδειγμα, θα εκπαιδεύσουμε όπως και πριν τα 9 δενδρικά μοντέλα χρησιμοποιώντας το σύνολο εκπαίδευσης, αλλά θα μετρήσουμε την επίδοση τους στο σύνολο επικύρωσης. Στη συνέχεια θα επιλέξουμε το μοντέλο με την καλύτερη επίδοση και θα μετρήσουμε την επίδοση του στο σύνολο ελέγχου.

Μετά από την επιλογή μοντέλου, θα μπορούσαμε να εκπαιδεύσουμε ένα μοντέλο με τις βέλτιστες παραμέτρους στην ένωση των συνόλων εκπαίδευσης και επικύρωσης και στην συνέχεια να μετρήσουμε την απόδοση αυτού του μοντέλου στο σύνολο ελέγχου, στο πλαίσιο της διαδικασίας εκτίμησης μοντέλου. Ομοίως μετά και από την εκτίμηση μοντέλου, θα μπορούσαμε να εκπαιδεύσουμε ένα μοντέλο με τις βέλτιστες παραμέτρους στην ένωση των συνόλων εκπαίδευσης, επικύρωσης και ελέγχου, δηλαδή σε όλα τα αρχικά μας δεδομένα, και να χρησιμοποιήσουμε το μοντέλο αυτό στην παραγωγή για να λάβουμε προβλέψεις σε άγνωστα δεδομένα. Οι παραπάνω πρακτικές είναι ακατάλληλες για οικογένειες αλγορίθμων, όπως τα νευρωνικά δίκτυα, που είναι ασταθείς, δηλαδή ευαίσθητοι ακόμη και σε μικρές αλλαγές στα δεδομένα εκπαίδευσης. Θα μπορούσαν ωστόσο να υιοθετηθούν στην περίπτωση σταθερών αλγορίθμων, όπως τα γραμμικά μοντέλα, ειδικά όταν το πλήθος των διαθέσιμων δεδομένων εκπαίδευσης δεν είναι επαρκές.

4.1.1 Κράτηση και σταυρωτή επικύρωση

Ένα σημαντικό ερώτημα εδώ είναι πόσο μεγάλο θα πρέπει να είναι το κάθε ένα από τα σύνολα εκπαίδευσης, επικύρωσης και ελέγχου. Προφανώς, όσο μεγαλύτερα είναι, τόσο το καλύτερο. Αν το πλήθος των αρχικών δεδομένων εκπαίδευσης είναι αρκετά μεγάλο, η τεχνική της **κράτησης** (holdout), κρατάει στην άκρη ένα ποσοστό τους για επικύρωση και έλεγχο. Π.χ. μπορούμε να χρησιμοποιήσουμε 60% για εκπαίδευση, 20% για επικύρωση και άλλο 20% για έλεγχο. Όταν ωστόσο τα δεδομένα είναι λίγα, τότε η μείωση των συνόλου εκπαίδευσης κατά 40% μπορεί να έχει σημαντική επίδραση στην ακρίβεια του μοντέλου που θα εκπαιδευτεί. Καθώς η επίδραση αυτή είναι σημαντικότερη από την επίδραση που έχει ο μειωμένος αριθμός παραδειγμάτων των σύνολων ελέγχου ή/και επικύρωσης στην αξιοπιστία της εκτίμησης, μπορούμε να τροποποιήσουμε τα ποσοστά σε 80%, 10% και 10% ή ακόμα και 90%, 5% και 5%.

Για τη βελτίωση της αξιοπιστίας της εκτίμησης της παραπάνω διαδικασίας μπορεί να χρησιμοποιηθεί η μέθοδος της **επαναλαμβανόμενης κράτησης** (repeated holdout). Η διαδικασία της κράτησης επαναλαμβάνεται πολλές φορές (π.χ. 10), επιλέγοντας κάθε φορά με τυχαία δειγματοληψία έναν διαφορετικό διαχωρισμό των δεδομένων σε σύνολα εκπαίδευσης, επικύρωσης και ελέγχου, και υπολογίζουμε τον μέσο όρο των αποτελεσμάτων τους. Ένα μειονέκτημα αυτής της προσέγγισης είναι πως τα σύνολα ελέγχου (και επικύρωσης) διαφορετικών επαναλήψεων ενδέχεται να επικαλύπτονται, επομένως οι επί μέρους εκτιμήσεις δεν θα είναι ανεξάρτητες μεταξύ τους, και ο μέσος όρος τους λιγότερο αξιόπιστος.

Εναλλακτικά μπορεί να χρησιμοποιηθεί η μέθοδος της ***k*-πλήρης σταυρωτής επικύρωσης** (*k*-fold cross-

validation). Σύμφωνα με αυτήν, τα δεδομένα εκπαίδευσης D χωρίζονται σε k ισομερέθη ξένα μεταξύ τους υποσύνολα D_1, D_2, \dots, D_k . Επαναλαμβάνουμε έπειτα για κάθε υποσύνολο την ακόλουθη διαδικασία που παρουσιάζεται γραφικά στο Σχήμα 4.1. Εκπαιδεύουμε ένα μοντέλο από την ένωση των υπόλοιπων υποσυνόλων και εφαρμόζουμε το μοντέλο αυτό στο τρέχον υποσύνολο. Στο τέλος παίρνουμε τον μέσο όρο των k αποτελεσμάτων. Στην βιβλιογραφία συνήθως το k τίθεται στην τιμή 10, εκτός και αν το υπολογιστικό κόστος είναι μεγάλο, οπότε τίθεται σε μικρότερες τιμές (π.χ. 3 ή 4). Μικρές τιμές του k επηρεάζουν περισσότερο τα αποτελέσματα της εκτίμησης μοντέλου και λιγότερο της επιλογής μοντέλου.



Σχήμα 4.1: Η διαδικασία της k -πλής σταυρωτής επικύρωσης.

Αν θέλουμε να κάνουμε ταυτόχρονα επιλογή και εκτίμηση μοντέλου, τότε η διαδικασία της k -πλής σταυρωτής επικύρωσης θα πρέπει να γίνεται φωλιασμένα. Μία εξωτερική σταυρωτή επικύρωση θα οδηγήσει σε k εκτιμήσεις του σφάλματος του μοντέλου. Στο σύνολο εκπαίδευσης κάθε επανάληψης αυτής της διαδικασίας, εφαρμόζεται μια εσωτερική σταυρωτή επικύρωση προκειμένου να επιτευχθεί η επιλογή μοντέλου. Αυτό έχει σαν αποτέλεσμα $k^2 + k$ εκπαιδεύσεις μοντέλων. Για την βελτίωση της αξιοπιστίας της παραπάνω εκτίμησης μπορεί η διαδικασία να επαναληφθεί πολλές φορές. Για παράδειγμα, αν εφαρμόσουμε 10 φορές 10-πλη φωλιασμένη σταυρωτή επικύρωση, τότε θα εκπαιδεύσουμε συνολικά 1100 μοντέλα.

Μια ακραία εκδοχή της σταυρωτής επικύρωσης χωρίζει το σύνολο εκπαίδευσης σε τόσα υποσύνολα ώστα τα παραδείγματα εκπαίδευσης. Επομένως έχουμε m επανάληψεις της διαδικασίας αξιολόγησης με 1 παραδείγμα στο σύνολο ελέγχου ή επικύρωσης και $m - 1$ παραδείγματα στο σύνολο εκπαίδευσης. Αυτή η διαδικασία καλείται **σταυρωτή επικύρωση άφησε ένα εκτός** (leave-one-out cross-validation). Η διαδικασία αυτή έχει το θετικό πως αξιοποιεί το μεγαλύτερο δυνατό μέρος των δεδομένων για εκπαίδευση, αλλά και το σημαντικό αρνητικό του υψηλού υπολογιστικού κόστους. Για τους παραπάνω λόγους ενδείκνυται σε περιπτώσεις πολύ μικρών συνόλων εκπαίδευσης. Τέλος, σε αντίθεση με την κλασική σταυρωτή επικύρωση, η διαδικασία αυτή είναι ντετερμινιστική, καθώς δεν εκτελούμε κάποια τυχαία δειγματοληψία των δεδομένων για την παραγωγή των υποσυνόλων.

Μια άλλη διαδικασία που χρησιμοποιείται όταν το σύνολο των δεδομένων εκπαίδευσης έχει μικρό μέγεθος, είναι η **χρήση ίδιων δυνάμεων** (bootstrap). Σε αυτή τη διαδικασία εκτελούμε δειγματοληψία με επανατοποθέτηση προκειμένου να λάβουμε m παραδείγματα για εκπαίδευση (όπου m το μέγεθος του συνόλου των δεδομένων μας). Στη συνέχεια συνδυάζουμε την επίδοση του μοντέλου στο σύνολο εκπαίδευσης e_{train} και στο σύνολο των παραδειγμάτων που δεν επιλέχθηκαν κατά τη διαδικασία της δειγματοληψίας e_{test} ως εξής:

$$e_{bootstrap} = 0.632e_{test} + 0.368e_{train} \quad (4.1)$$

Όταν τα δεδομένα είναι λίγα η εκτίμηση στο σύνολο ελέγχου θα αναμένεται να είναι απαισιόδοξη, ενώ αντίθετα η εκτίμηση στα δεδομένα εκπαίδευσης που είναι ήδη γνωστά στον αλγόριθμο αναμένεται να είναι

αισιόδοξη. Τα αντίστοιχα βάρη 0.632 και 0.368 έρχονται να συγκεράσουν τις δύο αυτές παρατηρήσεις προκειμένου να πάρουμε μια αξιόπιστη εκτίμηση. Οι τιμές προέρχονται από την πιθανότητα επιλογής (0.632) ή μη (0.368) ενός δείγματος στο σύνολο εκπαίδευσης από τη διαδικασία δειγματοληψίας με επανατοποθέτηση όταν το πλήθος των δεδομένων, m , τείνει στο άπειρο.

4.1.2 Στρωματοποίηση και ομαδοποίηση

Σε προβλήματα ταξινόμησης, ο διαχωρισμός των δεδομένων μπορεί να γίνει με τη διαδικασία της **στρωματοποίησης** (stratification) με βάση την κλάση. Η στρωματοποίηση φροντίζει το ποσοστό παραδειγμάτων της κάθε κλάσης στα υποσύνολα εκπαίδευσης, επικύρωσης και ελέγχου να είναι το ίδιο όπως και στο αρχικό σύνολο εκπαίδευσης. Έστω για παράδειγμα ένα αρχικό σύνολο εκπαίδευσης με 100 παραδείγματα εκ των οποίων τα 10 ανήκουν στην κλάση 0 και στα 90 στην κλάση 1. Έστω ότι θα το χωρίσουμε σε υποσύνολα εκπαίδευσης (60%), επικύρωσης (20%) και ελέγχου (20%). Κάνοντας χρήση στρωματοποιημένης δειγματοληψίας, στα υποσύνολα αυτά θα καταλήξουν 6, 2 και 2 παραδείγματα της κλάσης 0 αντίστοιχα. Με τυχαία δειγματοληψία, η κατανομή αυτή δεν είναι εγγυημένη. Στην πράξη η στρωματοποίηση οδηγεί σε καλύτερες επιλογές και εκτιμήσεις μοντέλου.

Είναι πολύ σημαντικό ο τρόπος με τον οποίο θα χωρίσουμε το αρχικό μας σύνολο εκπαίδευσης σε υποσύνολα εκπαίδευσης, επικύρωσης και ελέγχου να συνάδει με τον τρόπο με τον οποίο θα χρησιμοποιηθεί το μοντέλο στην πράξη. Έστω για παράδειγμα ότι τα παραδείγματά μας αντιστοιχούν σε 100 καταστήματα λιανικής πώλησης ηλεκτρικών συσκευών, τα οποία είναι κατανεμημένα γεωγραφικά σε 6 διαφορετικές χώρες της Ευρώπης. Έστω ότι θέλουμε να εκτιμήσουμε τα έσοδα που θα έχουμε από ένα νέο κατάστημα που σκοπεύουμε να ανοίξουμε σε μια νέα, 7η, χώρα της Ευρώπης. Μπορούμε να χτίσουμε ένα μοντέλο παλινδρόμησης λαμβάνοντας ως μεταβλητές εισόδου οικονομικά, γεωγραφικά, κλιματικά, και άλλα χαρακτηριστικά της περιοχής κάθε καταστήματος. Ένας ρεαλιστικός διαχωρισμός των δεδομένων μας σε σύνολο εκπαίδευσης, επικύρωσης και ελέγχου, θα φρόντιζε να μην υπάρχουν καταστήματα της ίδιας χώρας σε διαφορετικά σύνολα, καθώς στην συγκεκριμένη εφαρμογή το νέο κατάστημα θα βρίσκεται σε μια νέα χώρα. Θα μπορούσαμε π.χ. να έχουμε τα καταστήματα των 4 από τις 6 χώρες στο σύνολο εκπαίδευσης, της 5ης στο σύνολο επικύρωσης και της 6ης στο σύνολο ελέγχου. Παρόμοια, στην ανακάλυψη φαρμάκων, ο διαχωρισμός των χημικών ουσιών σε σύνολα εκπαίδευσης, επικύρωσης και ελέγχου, περιλαμβάνει ένα αρχικό στάδιο ομαδοποίησης τους, προκειμένου να μην τοποθετηθούν σε διαφορετικά υποσύνολα αρκετά όμοιες χημικές ουσίες, καθώς οι νέες χημικές ουσίες στις οποίες δίνουν προβλέψεις τα μοντέλα αυτά, είναι συνήθως αρκετά διαφορετικές από αυτές που το μοντέλο έχει εκπαιδευτεί. Γενικότερα, μπορούμε να θεωρήσουμε πως υπάρχει κάποια μεταβλητή, με την οποία τα δεδομένα εκπαίδευσης ομαδοποιούνται (χώρα στο πρώτο παράδειγμα, ομάδα που προέκυψε από την διαδικασία ομαδοποίησης στο δεύτερο παράδειγμα). Οι διαδικασίες διαχωρισμού των δεδομένων σε σύνολα εκπαίδευσης, επικύρωσης και ελέγχου μπορούν να πάρουν αυτήν την μεταβλητή ως παράμετρο, προκειμένου να φροντίσουν ώστε όλα τα δεδομένα κάθε ομάδας να είναι στο ίδιο σύνολο.

4.2 Μετρικές Αξιολόγησης

Στην ενότητα αυτή θα μελετήσουμε διάφορες μετρικές αξιολόγησης που υπάρχουν στη βιβλιογραφία για μοντέλα παλινδρόμησης και ταξινόμησης. Για τον ορισμό αυτών των μετρικών θα θεωρήσουμε ότι έχουμε ένα σύνολο δεδομένων ελέγχου $D = \{(x^{(i)}, y^{(i)}), \dots, (x^{(m)}, y^{(m)})\}$ και ένα μοντέλο h .

4.2.1 Παλινδρόμηση

Δύο βασικές μετρικές αξιολόγησης είναι το **μέσο απόλυτο σφάλμα** (mean absolute error - MAE) και η **τετραγωνική ρίζα** του **μέσου τετραγωνικού σφάλματος** (root mean squared error - RMSE):

$$MAE(h, D) = \frac{1}{m} \sum_{i=1}^m |h(\mathbf{x}^{(i)}) - y^{(i)}| \quad (4.2)$$

$$RMSE(h, D) = \sqrt{\frac{1}{m} \sum_{i=1}^m (h(\mathbf{x}^{(i)}) - y^{(i)})^2} \quad (4.3)$$

Τόσο η MAE όσο και η RMSE είναι στην ίδια κλίμακα με την μεταβλητή εξόδου, οπότε είναι εύκολο να κατανοήσουμε τη φυσική τους σημασία. Σε σχέση με τη MAE, η RMSE τιμωρεί περισσότερο τα μεγαλύτερα σφάλματα απ' ότι τα μικρότερα. Αυτό είναι συχνά κάτι επιθυμητό. Ωστόσο, για τον ίδιο λόγο η RMSE είναι πιο ευαίσθητη στην παρουσία έκτοπων τιμών της μεταβλητής εξόδου. Τέτοιες τιμές αναμένουμε να οδηγήσουν σε μεγάλα σφάλματα, τα οποία όμως δεν θα αντιπροσωπεύουν την πραγματική επίδοση του μοντέλου. Σε τέτοιες περιπτώσεις, η MAE επηρεάζεται λιγότερο.

Οι παραπάνω μετρικές μπορούν να υπολογιστούν και σε σχέση με ένα απλοϊκό μοντέλο, το οποίο δίνει ως έξοδο την μέση τιμή της μεταβλητής εξόδου στο σύνολο εκπαίδευσης, \bar{y}_{train} . Ονομάζονται αντίστοιχα **σχετικό απόλυτο σφάλμα** (relative absolute error - RAE) και **τετραγωνική ρίζα του σχετικού τετραγωνικού σφάλματος** (root relative squared error - RRSE):

$$RAE(h, D) = \frac{\sum_{i=1}^m |h(\mathbf{x}^{(i)}) - y^{(i)}|}{\sum_{i=1}^m |y^{(i)} - \bar{y}_{train}|} \quad (4.4)$$

$$RRSE(h, D) = \sqrt{\frac{\sum_{i=1}^m (h(\mathbf{x}^{(i)}) - y^{(i)})^2}{\sum_{i=1}^m (y^{(i)} - \bar{y}_{train})^2}} \quad (4.5)$$

Οι μετρικές RAE και RRSE δείχνουν πόσο έχει καταφέρει να μάθει ένα μοντέλο σε σχέση με ένα απλοϊκό μοντέλο, κάτι που δεν είναι προφανές από τις μετρικές MAE και RMSE.

Μια διαδεδομένη μετρική που θυμίζει ως ένα βαθμό το RRSE είναι ο **συντελεστής προσδιορισμού** (coefficient of determination) ή R^2 :

$$R^2(h, D) = 1 - \frac{\sum_{i=1}^m (h(\mathbf{x}^{(i)}) - y^{(i)})^2}{\sum_{i=1}^m (y^{(i)} - \bar{y})^2} \quad (4.6)$$

όπου $\bar{y} = \frac{1}{m} \sum_{i=1}^m y^{(i)}$ είναι η μέση τιμή της εξαρτημένης μεταβλητής στο σύνολο ελέγχου. Ο συντελεστής προσδιορισμού αποτιμά το ποσοστό της διακύμανσης της εξαρτημένης μεταβλητής που μπορεί να προβλεφθεί από τις ανεξάρτητες μεταβλητές. Η τιμή του συντελεστή προσδιορισμού κυμαίνεται τυπικά από 0 ως 1, αν και μπορεί να πάρει και αρνητικές τιμές για μοντέλα που απέχουν πολύ από τα πραγματικά δεδομένα. Όσο πιο κοντά στο 1, τόσο καλύτερο το μοντέλο.

Το **μέσο απόλυτο ποσοστιαίο σφάλμα** (mean absolute percentage error - MAPE) είναι μια μετρική που υπολογίζει το απόλυτο σφάλμα σε σχέση με την πραγματική τιμή της εξαρτημένης μεταβλητής:

$$MAPE(h, D) = \frac{100}{m} \sum_{i=1}^m \left| \frac{y^{(i)} - h(\mathbf{x}^{(i)})}{y^{(i)}} \right| \quad (4.7)$$

Το MAPE χρησιμοποιείται όταν αυτό που μας ενδιαφέρει είναι το ποσοστό του σφάλματος και όχι η απόλυτη τιμή του. Για παράδειγμα, αν ένα απόλυτο σφάλμα των 50 μονάδων για μια πραγματική τιμή των 500 μονάδων, είναι το ίδιο σημαντικό με ένα απόλυτο σφάλμα της μίας μονάδας για μια πραγματική τιμή των 10 μονάδων, τότε το MAPE είναι μια κατάλληλη μετρική καθώς θα υπολογίσει ένα ποσοστιαίο σφάλμα 10% και στις δύο περιπτώσεις.

Ένα μειονέκτημα του MAPE είναι το γεγονός ότι δεν ορίζεται όταν ο παρονομαστής είναι μηδέν. Κάτι τέτοιο μπορεί να προκύψει σε μια εφαρμογή πρόβλεψης μελλοντικής ζήτησης προϊόντων ή υπηρεσιών. Ένα

εξεζητημένο προϊόν ενός συύπερ μάρκετ για παράδειγμα, μπορεί να έχει μηδενικές πωλήσεις ορισμένες ημέρες. Οι μετρικές RAE, RRSE και R^2 δεν πάσχουν από αυτό το πρόβλημα.

Σε αντίθεση με τις MAE και RMSE που εκτιμούν απευθείας τη διαφορά μεταξύ της πρόβλεψης και της πραγματικής τιμής της εξαρτημένης μεταβλητής, οι RAE, RRSE, R^2 και MAPE είναι συγκριτικές μετρικές, καθώς λαμβάνουν υπόψη την παραπάνω διαφορά σε σχέση με μια άλλη ποσότητα.

4.2.2 Ταξινόμηση

Μια από τις πιο συχνά χρησιμοποιούμενες μετρικές αξιολόγησης για μοντέλα ταξινόμησης είναι η **ορθότητα** (accuracy), η οποία ισούται με τον λόγο του πλήθους των σωστών προβλέψεων προς το σύνολο των παραδειγμάτων ελέγχου:

$$\text{Ορθότητα} = \frac{|\{(x^{(i)}, y^{(i)}) \in D : h(x^{(i)}) = y^{(i)}\}|}{m} \quad (4.8)$$

Ένας καλός τρόπος για να παρατηρήσουμε εποπτικά την επίδοση ενός μοντέλου ταξινόμησης είναι μέσω ενός **πίνακα σύγχυσης** (confusion matrix). Ο Πίνακας 4.1 παρουσιάζει έναν πίνακα σύγχυσης για μια εργασία ταξινόμησης μηνυμάτων ηλεκτρονικού ταχυδρομείου. Το μοντέλο ταξινομεί ένα μήνυμα ως επείγον, κανονικό ή ανεπιθύμητο.

		Πραγματική		
		επείγον	κανονικό	ανεπιθύμητο
Πρόβλεψη	επείγον	8	10	1
	κανονικό	5	60	50
	ανεπιθύμητο	3	30	200

Πίνακας 4.1: Πίνακας σύγχυσης ενός μοντέλου ταξινόμησης μηνυμάτων ηλεκτρονικού ταχυδρομείου σε τρεις κλάσεις.

Ο πίνακας σύγχυσης είναι ένας τετραγωνικός πίνακας με αριθμό στηλών και γραμμών ίσο με τον αριθμό των κλάσεων της εργασίας ταξινόμησης. Οι γραμμές ενός πίνακα σύγχυσης αντιστοιχούν στις κλάσεις που δίνονται ως πρόβλεψη από το μοντέλο για ένα παράδειγμα ελέγχου και οι στήλες στην πραγματική κλάση του. Τα κελιά, c_{ij} , ενός πίνακα σύγχυσης απαριθμούν τα παραδείγματα ελέγχου για κάθε συνδυασμό πραγματικής κλάσης και κλάσης που δόθηκε ως πρόβλεψη από το μοντέλο:

$$c_{ij} = |\{(x^{(k)}, y^{(k)}) \in D : h(x^{(k)}) = i, y^{(k)} = j\}| \quad (4.9)$$

Παρατηρώντας έναν πίνακα σύγχυσης, στην διαγώνιο του θα δούμε τον αριθμό των σωστών προβλέψεων του μοντέλου, ενώ εκτός διαγωνίου τα σφάλματά του. Με βάση έναν πίνακα σύγχυσης, η ορθότητα υπολογίζεται ως εξής:

$$\text{Ορθότητα} = \frac{\sum_{k=1}^c c_{kk}}{m} \quad (4.10)$$

Η ορθότητα αξιολογεί ένα μοντέλο συνολικά για όλες τις κλάσεις. Πολλές φορές ωστόσο θέλουμε να γνωρίζουμε την επίδοση του μοντέλου για μία μόνο από τις κλάσεις ή για κάθε μία από τις κλάσεις ξεχωριστά. Σε μια εφαρμογή διάγνωσης της ασθένειας COVID-19 από ακτινογραφία θώρακα για παράδειγμα, είναι πιθανότερο να μας ενδιαφέρει περισσότερο η κλάση που αντιστοιχεί σε άτομα που πάσχουν από την ασθένεια, παρά εκείνη που αντιστοιχεί σε άτομα που είναι υγή.

Σε εφαρμογές δυαδικής ταξινόμησης σαν και αυτή που αναφέραμε παραπάνω, η κλάση ενδιαφέροντος καλείται συχνά **θετική** και αντίστοιχα ή άλλη κλάση καλείται **αρνητική**. Για τον λόγο αυτό, οι τιμές των τεσσάρων κελιών ενός πίνακα σύγχυσης δυαδικής ταξινόμησης καλούνται **αληθώς θετικά** (true positive - TP),

αληθώς αρνητικά (true negative - TN), **ψευδώς θετικά** (false positive - FP) και **ψευδώς αρνητικά** (false negative - FN), όπως φαίνεται και στον Πίνακα 4.2.

		Πραγματική	Θετική	Αρνητική
		Θετική	TP	FP
Πρόβλεψη	Αρνητική	FN	TN	

Πίνακας 4.2: Πίνακας σύγχυσης για ένα πρόβλημα δυαδικής ταξινόμησης.

Δύο δημοφιλείς μετρικές αξιολόγησης της θετικής κλάσης είναι η **ακρίβεια** (precision) και η **ανάκληση** (recall). Η ακρίβεια μετράει τον αριθμό των παραδειγμάτων που ταξινομήθηκαν ορθά ως θετικά προς όλα τα παραδείγματα που ταξινομήθηκαν ως θετικά:

$$\text{Ακρίβεια} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4.11)$$

Η ανάκληση, η οποία καλείται επίσης **ενασθησία** (sensitivity) και **ρυθμός αληθώς θετικών** (true positive rate), μετράει τον αριθμό των παραδειγμάτων που ταξινομήθηκαν ορθά ως θετικά προς όλα τα παραδείγματα που είναι πραγματικά θετικά:

$$\text{Ανάκληση} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4.12)$$

Συχνά μας ενδιαφέρει η αντίστοιχη μετρική της ανάκλησης, αλλά για την αρνητική κλάση, η οποία καλείται **εξειδίκευση** (specificity) και **ρυθμός αληθώς αρνητικών** (true negative rate). Η εξειδίκευση μετράει τον αριθμό των παραδειγμάτων που ταξινομήθηκαν ορθά ως αρνητικά προς όλα τα παραδείγματα που είναι πραγματικά αρνητικά:

$$\text{Εξειδίκευση} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (4.13)$$

Η ακρίβεια και η ανάκληση συχνά συνδυάζονται στην επονομαζόμενη **μετρική F** (F-measure ή F_1 score), η οποία αποτελεί τον αρμονικό μέσο όρο τους, έναν από τους τρεις μέσους όρους του Πυθαγόρα, μαζί με τον αριθμητικό και τον γεωμετρικό μέσο όρο:

$$\begin{aligned} \text{F-measure} &= \left(\frac{\text{Ακρίβεια}^{-1} + \text{Ανάκληση}^{-1}}{2} \right)^{-1} = \frac{2}{\frac{1}{\text{Ακρίβεια}} + \frac{1}{\text{Ανάκληση}}} = \frac{2 \cdot \text{Ακρίβεια} \cdot \text{Ανάκληση}}{\text{Ακρίβεια} + \text{Ανάκληση}} = \\ &= \frac{2 \cdot \frac{\text{TP}}{\text{TP} + \text{FP}} \cdot \frac{\text{TP}}{\text{TP} + \text{FN}}}{\frac{\text{TP}}{\text{TP} + \text{FP}} + \frac{\text{TP}}{\text{TP} + \text{FN}}} = \frac{\frac{2 \cdot \text{TP}^2}{(\text{TP} + \text{FP}) \cdot (\text{TP} + \text{FN})}}{\frac{\text{TP} \cdot (\text{TP} + \text{FN}) + \text{TP} \cdot (\text{TP} + \text{FP})}{(\text{TP} + \text{FP}) \cdot (\text{TP} + \text{FN})}} = \frac{2 \text{TP}}{2 \text{TP} + \text{FN} + \text{FP}} \end{aligned} \quad (4.14)$$

Ο δείκτης 1 στο F_1 score, αποτελεί στην πραγματικότητα την τιμή της παραμέτρου β , στην F_β , μια γενικότερη εκδοχή της μετρικής F, όπου μπορούμε να ρυθμίσουμε την σημαντικότητα της ανάκλησης σε σχέση με την ακρίβεια:

$$F_\beta = \frac{(1 + \beta^2) \cdot \text{Ακρίβεια} \cdot \text{Ανάκληση}}{\beta^2 \cdot \text{Ακρίβεια} + \text{Ανάκληση}} = \frac{(1 + \beta^2) \cdot \text{TP}}{(1 + \beta^2) \cdot \text{TP} + \beta^2 \cdot \text{FN} + \text{FP}} \quad (4.15)$$

Η τιμή 1 που χρησιμοποιείται στην μετρική F σημαίνει πως τόσο η ανάκληση όσο και η ακρίβεια είναι το ίδιο σημαντικές. Δυο άλλες τιμές που χρησιμοποιούνται συχνά είναι 2 και 0.5. Η πρώτη σημαίνει πως η ανάκληση είναι δύο φορές σημαντικότερη από την ακρίβεια, ενώ το 0.5 το αντίθετο.

Η ακρίβεια, η ανάκληση και η μετρική F μπορούν να υπολογιστούν και για την αρνητική κλάση, αν αντιστρέψουμε τους ρόλους των δύο κλάσεων, δηλαδή αν θεωρήσουμε την αρνητική κλάση ως θετική και την θετική ως αρνητική. Επιπλέον, οι παραπάνω μετρικές μπορούν να υπολογιστούν για οποιαδήποτε κλάση k μιας εργασίας ταξινόμησης με $c > 2$ κλάσεις, λαμβάνοντας υπόψη όλες τις υπόλοιπες κλάσεις μαζί ως μία αρνητική κλάση, ως εξής:

$$\text{TP}_k = c_{kk} \quad (4.16)$$

$$\text{FP}_k = \sum_{i=1}^c c_{ki} - \text{TP}_k. \quad (4.17)$$

$$\text{FN}_k = \sum_{i=1}^c c_{ik} - \text{TP}_k. \quad (4.18)$$

$$\text{TN}_k = m - \text{TP}_k - \text{FP}_k - \text{FN}_k. \quad (4.19)$$

Ο υπολογισμός μιας καθολικής ακρίβειας, ανάκλησης και μετρικής F που να αφορά σε όλες τις κλάσεις, μπορεί να γίνει με τους εξής δύο τρόπους:

- υπολογισμός μέσου όρου των τιμών που αντιστοιχούν στις μετρικές για κάθε κλάση k , που ονομάζεται **μακρό-υπολογισμός του μέσου όρου** (macro-averaging) και θα συμβολίζεται με M
- υπολογισμός μίας τιμής της κάθε μετρικής με βάση τα αθροίσματα των TP_k , TN_k , FP_k και FN_k όλων των κλάσεων, που ονομάζεται **μίκρο-υπολογισμός του μέσου όρου** (micro-averaging) και θα συμβολίζεται με μ

Ο μακρό-υπολογισμός του μέσου όρου θεωρεί πως όλες οι κλάσεις έχουν την ίδια σημαντικότητα, ενώ ο μίκρο-υπολογισμός του μέσου όρου επηρεάζεται από την συχνότητα των κλάσεων, δίνοντας μεγαλύτερο (μικρότερο) βάρος στις συχνές (σπάνιες) κλάσεις. Παρακάτω ορίζονται οι καθολικές εκδοχές της ακρίβειας, ανάκλησης και μετρικής F.

$$\text{Ακρίβεια}_M = \frac{1}{c} \sum_{k=1}^c \frac{\text{TP}_k}{\text{TP}_k + \text{FP}_k} \quad (4.20)$$

$$\text{Ανάκληση}_M = \frac{1}{c} \sum_{k=1}^c \frac{\text{TP}_k}{\text{TP}_k + \text{FN}_k} \quad (4.21)$$

$$\text{F-measure}_M = \frac{1}{c} \sum_{k=1}^c \frac{2\text{TP}_k}{2\text{TP}_k + \text{FN}_k + \text{FP}_k} \quad (4.22)$$

$$\text{Ακρίβεια}_\mu = \text{Ανάκληση}_\mu = \text{F-measure}_\mu = \frac{\sum_{k=1}^c \text{TP}_k}{m} \quad (4.23)$$

Υπάρχουν μετρικές που λαμβάνουν υπόψη την κατανομή πιθανότητας $[h(0|\mathbf{x}), h(1|\mathbf{x}), \dots, h(c-1|\mathbf{x})]$ που μπορεί να δώσει ένα μοντέλο ταξινόμησης στην έξοδο του για c κλάσεις. Μια τέτοια μετρική είναι το Brier Score, το οποίο για εργασίες δυαδικής ταξινόμησης ισούται με το μέσο τετραγωνικό σφάλμα μεταξύ της πραγματικής τιμής της κλάσης (0 ή 1) και της πιθανότητας που δίνει το μοντέλο για την κλάση 1:

$$\text{BrierScore} = \frac{1}{m} \sum_{i=1}^m (h(1|\mathbf{x}^{(i)}) - y^{(i)})^2 \quad (4.24)$$

To Brier Score μπορεί να υπολογιστεί επίσης και για εργασίες ταξινόμησης σε πάνω από 2 κλάσεις ως εξής:

$$\text{BrierScore} = \frac{1}{m} \sum_{i=1}^m \sum_{j=0}^{c-1} \left(\hat{y}_j^{(i)} - y_j^{(i)} \right)^2 = \frac{1}{m} \sum_{i=1}^m \left(1 - 2h(y|x^{(i)}) + \sum_{j=0}^{c-1} \hat{y}_j^{(i)} \right) \quad (4.25)$$

όπου $y_j^{(i)} = 1$ αν $y^{(i)} = j$ και $y_j^{(i)} = 0$ αλλιώς.

Στην περίπτωση των 2 κλάσεων η τιμή αυτής της έκδοσης του Brier Score ισούται με το διπλάσιο της έκδοσης για εργασίες δυαδικής ταξινόμησης.

4.3 Περαιτέρω Μελέτη

Περισσότερες πληροφορίες για διαδικασίες και μετρικές αξιολόγησης μοντέλων ταξινόμησης μπορεί να βρει κανείς στο [1]. Η αναλυτική μελέτη της μετρικής F [2], παρουσιάζει την ιστορία της μετρικής, τις ιδιότητές της, την κριτική που έχει ασκηθεί σε αυτήν, συστάσεις για ορθή χρήση της και εναλλακτικές της. Το όνομα της μετρικής Brier Score προέρχεται από τον Glenn Brier [3].

4.4 Ασκήσεις

- Έστω ότι έχετε στη διάθεση σας 1000 παραδείγματα καλών και 1000 παραδείγματα κακών πελατών μιας τράπεζας για έγκριση δανείου. Θέλετε να εκπαιδεύσετε ένα δέντρο απόφασης, για το οποίο θέλετε να επιλέξετε την παράμετρο του βάθους, και επιπλέον θέλετε να εκτιμήσετε το μελλοντικό σφάλμα που θα κάνει το μοντέλο αυτό με τη βέλτιστη παράμετρο σε νέους πελάτες. Τι θα κάνετε;
- Έστω ότι στο παρακάτω σύνολο 12 παραδειγμάτων εκπαίδευσης εκτελείτε στρωματοποιημένη 4-πλή σταυρωτή επικύρωση (stratified 4-fold cross validation). Πόσα παραδείγματα της κλάσης 0 θα υπάρχουν σε κάθε ένα από τα σύνολα ελέγχου που θα δημιουργηθούν;

$$y \mid 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1$$

- Έστω ο παρακάτω πίνακας σύγχυσης για 4 κλάσεις. Θυμηθείτε ότι οι γραμμές αφορούν τις προβλέψεις του μοντέλου, ενώ οι στήλες τις πραγματικές κλάσεις. Σε ποια κλάση έχουμε τη μεγαλύτερη ακρίβεια (precision);

	0	1	2	3
0	8	0	4	3
1	2	11	1	0
2	3	2	12	0
3	0	4	0	12

- Έστω ο παρακάτω πίνακας σύγχυσης για 4 κλάσεις. Θυμηθείτε ότι οι γραμμές αφορούν τις προβλέψεις του μοντέλου, ενώ οι στήλες τις πραγματικές κλάσεις. Σε ποια κλάση έχουμε τη μεγαλύτερη ανάκληση (recall);

	0	1	2	3
0	9	1	4	2
1	2	12	3	4
2	3	2	11	0
3	0	4	1	9

5. Δείξτε ότι όταν εφαρμόσουμε την εξίσωση 4.25 σε εργασίες δυαδικής ταξινόμησης, το αποτέλεσμα ισούται με το διπλάσιο εκείνου της εξίσωσης 4.24.
6. Έστω ότι έχουμε ένα σύστημα δυαδικής κατάταξης το οποίο αξιολογήθηκε σε 1000 δείγματα. Η θετική κλάση εμφανίζεται σε 200 δείγματα. Το μοντέλο πέτυχε ακρίβεια 0.80 και ανάκληση 0.50.
 - (α) Συμπληρώστε τον πίνακα σύγχυσης.
 - (β) Υπολογίστε το F_1 score.

Βιβλιογραφία

- [1] Nathalie Japkowicz και Mohak Shah. *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, 2011.
- [2] Peter Christen, David J. Hand και Nishadi Kirielle. “A Review of the F-Measure: Its History, Properties, Criticism, and Alternatives”. Στο: *ACM Comput. Surv.* 56.3 (Οκτ. 2023). ISSN: 0360-0300. doi: [10.1145/3606367](https://doi.org/10.1145/3606367). URL: <https://doi.org/10.1145/3606367>.
- [3] Glenn W. Brier. “Verification of Forecasts Expressed in Terms of Probability”. Στο: *Monthly Weather Review* 78 (1950), σσ. 1–3.

ΚΕΦΑΛΑΙΟ 5

ΜΟΝΤΕΛΑ ΚΑΝΟΝΩΝ

Στο κεφάλαιο αυτό:

Θα μελετήσουμε την οικογένεια των μοντέλων που στηρίζονται σε κανόνες. Αρχικά θα μελετήσουμε την διαδικασία μάθησης συνόλων κανόνων, και στη συνέχεια θα δούμε τους γενετικούς αλγορίθμους και πως μπορούμε να τους χρησιμοποιήσουμε για να βελτιστοποιήσουμε σύνολα κανόνων.

Τα μοντέλα κανόνων είναι μοντέλα μηχανικής μάθησης που στηρίζονται σε κανόνες **αν-τότε** (if-then). Η αναπαράσταση τους συνίσταται σε ένα σύνολο κανόνων που αποτελούνται από συζεύξεις περιορισμών στις μεταβλητές εισόδου. Η αναπαράσταση των μοντέλων κανόνων είναι απλή και κατανοητή από τον άνθρωπο, ο οποίος είναι εξοικειωμένος με σύνολα κανόνων αν-τότε. Η ιδιότητά τους αυτή, τα καθιστά ιδιαίτερα δημοφιλή έναντι άλλων αλγορίθμων που μπορεί να πετυχαίνουν υψηλότερη ακρίβεια. Στις δύο ενότητες που ακολουθούν, θα μελετήσουμε αρχικά τη διαδικασία απευθείας μάθησης ενός συνόλου κανόνων, καθώς και τους δύο τύπους συνόλων κανόνων, τα ταξινομημένα (ordered) και τα μη ταξινομημένα (unordered), και έπειτα θα δούμε πως μπορούμε να βελτιστοποιήσουμε σύνολα κανόνων με τη χρήση γενετικών αλγορίθμων.

5.1 Σύνολα Κανόνων

Τα δενδρικά μοντέλα ισοδυναμούν με ένα σύνολο αμοιβαία αποκλειόμενων κανόνων, όπου κάθε φύλλο αποτελεί και έναν κανόνα. Επομένως, ένας πρώτος τρόπος για την μάθηση ενός συνόλου κανόνων είναι η εκπαίδευση ενός δενδρικού μοντέλου και η μετατροπή του σε ένα ισοδύναμο σύνολο κανόνων.

Ένας δεύτερος, και πιο ευέλικτος, τρόπος είναι να μάθουμε απευθείας ένα σύνολο κανόνων. Σε αυτήν την ενότητα θα μελετήσουμε μια τέτοια οικογένεια αλγορίθμων για μάθηση κανόνων που ονομάζονται αλγόριθμοι **ακολουθιακής κάλυψης** (sequential covering) ή αλγόριθμοι **διαχώρισε και βασίλευε** (separate and conquer). Οι αλγόριθμοι αυτοί είναι επαναληπτικοί, όπου σε κάθε επανάληψη μαθαίνουμε έναν καινούργιο κανόνα. Αφού μάθουμε έναν κανόνα αφαιρούμε τα παραδείγματα τα οποία ικανοποιούν τις συνθήκες του κανόνα.

Algorithm 1 Δημιουργία λίστας κανόνων

```

1: procedure LEARNRULELIST( $D$ )
2:    $R = []$  ▷ Κενό σύνολο κανόνων
3:   while  $D \neq \emptyset$  do
4:      $r \leftarrow \text{LearnRule}(D)$  ▷ Μάθηση κανόνα
5:      $R \leftarrow R \cup \{r\}$  ▷ Προσθήκη κανόνα στο σύνολο
6:      $D \leftarrow D \setminus \{x \in D \mid x \text{ covered by } r\}$  ▷ Αφαίρεση από το σύνολο δεδομένων των παραδειγμάτων που καλύπτονται από τον κανόνα  $r$ 
7:   end while
8:   return  $R$ 
9: end procedure
10: procedure LEARNRULE( $D$ )
11:    $b \leftarrow \text{true}$  ▷ Αρχικοποίηση κανόνα που ικανοποιείται πάντα
12:    $L \leftarrow \text{set of available literals}$  ▷ Σύνολο όλων των συνθηκών
13:   while not Homogeneous( $D$ ) do
14:      $l \leftarrow \text{BestLiteral}(L, D)$  ▷ Εύρεση καλύτερης συνθήκης
15:      $b \leftarrow b \wedge l$  ▷ Προσθήκη συνθήκης στον κανόνα
16:      $D \leftarrow \{x \in D \mid x \text{ covered by } b\}$  ▷ Αφαίρεση από το σύνολο δεδομένων των παραδειγμάτων που καλύπτονται από τον κανόνα  $b$ 
17:      $L \leftarrow L \setminus \{l' \in L \mid l' \text{ uses same feature as } l\}$  ▷ Αφαίρεση όλων των συνθηκών από το σύνολο που αφορούν το ίδιο χαρακτηριστικό με την συνθήκη  $l$ 
18:   end while
19:   return "if  $b$  then Label( $D$ )"
20: end procedure

```

5.1.1 Ταξινομημένα σύνολα κανόνων

Τα ταξινομημένα σύνολα κανόνων ονομάζονται και **λίστες κανόνων** (rule lists). Κύριο χαρακτηριστικό αυτών των συνόλων είναι πως παίζει ρολό η σειρά με την οποία εξετάζεται κάθε κανόνας. Η βασική ιδέα αυτού του αλγορίθμου μάθησης κανόνων είναι η επαναληπτική προσθήκη μιας συνθήκης, ενός συνδυασμού μεταβλητής και τιμής, στο σώμα του κανόνα με κριτήριο τη βελτίωση της ομοιογένειάς των παραδειγμάτων εκπαίδευσης που ικανοποιούν τις συνθήκες του. Ο αλγόριθμος 1 δείχνει τη διαδικασία για τη μάθηση μιας λίστας κανόνων και την επί μέρους διαδικασία μάθησης ενός κανόνα. Η έννοια διαχώρισε και βασίλευε έχει να κάνει όχι μόνο με το πως μαθαίνουμε τον ένα κανόνα πίσω από τον άλλο και αφαιρούμε παραδείγματα, αλλά και κατά τη μάθηση ενός κανόνα όπου και πάλι έχουμε επανάληψη με διαχωρισμό και αφαίρεση παραδειγμάτων μετά την προσθήκη μιας συνθήκης στο σώμα του κανόνα.

Η διαδικασία μάθησης του κανόνα παρομοιάζει τη διαδικασία μάθησης ενός δέντρου, όπου εξετάζουμε ένα συνεχώς μικρότερο μέρος του συνόλου παραδειγμάτων. Ωστόσο, σε αντίθεση με τα δέντρα όπου σε κάθε επανάληψη ο κόμβος-συνθήκη στάπει σε πολλά παιδιά, στην περίπτωση των κανόνων μάθησης η ανάπτυξη του κανόνα συνεχίζει σε ένα κλαδί μέχρι να βρεθεί φύλλο. Έπειτα, η διαδικασία επαναλαμβάνεται από την αρχή στο σύνολο των παραδειγμάτων που έχει απομείνει.

Ο αλγόριθμος μπορεί να γίνει καλύτερα κατανοητός μέσα από ένα παράδειγμα. Ο Πίνακας 5.1 περιέχει ένα σύνολο δεδομένων που ταξινομεί ένα θαλάσσιο κήτος ως δελφίνι ή όχι σε σχέση με το μήκος, τα βράγχια, το ρύγχος και τα δόντια του. Θεωρούμε πως η μεταβλητή μήκος είναι και αυτή διακριτή για λόγους απλότητας. Ξεκινάμε τη διαδικασία καταγράφοντας τον αριθμό των δελφινιών και μη δελφινιών για κάθε συνθήκη (συνδυασμό μεταβλητής και τιμής) που εμφανίζεται στο σύνολο δεδομένων μας, όπως φαίνεται και στον Πίνακα 5.2. Έπειτα, επιλέγουμε τη συνθήκη με τη μεγαλύτερη ομοιογένεια με την οποία θα ξεκινήσουμε τη δημιουργία του κανόνα. Στην προκειμένη περίπτωση βλέπουμε πως υπάρχουν τρεις διαφορετικές συνθήκες που

Μήκος	Βράγχια	Ρύγχος	Δόντια	Δελφίνι
3	'Οχι	Ναι	Πολλά	Ναι
4	'Οχι	Ναι	Πολλά	Ναι
3	'Οχι	Ναι	Λίγα	Ναι
5	'Οχι	Ναι	Πολλά	Ναι
5	'Οχι	Ναι	Λίγα	Ναι
5	Ναι	Ναι	Πολλά	'Οχι
4	Ναι	Ναι	Πολλά	'Οχι
5	Ναι	'Οχι	Πολλά	'Οχι
4	Ναι	'Οχι	Πολλά	'Οχι
4	'Οχι	Ναι	Λίγα	'Οχι

Πίνακας 5.1: Σύνολο δεδομένων που ταξινομεί ένα θαλάσσιο κήτος ως δελφίνι σύμφωνα με κάποια χαρακτηριστικά.

ικανοποιούνται από παραδείγματα της ίδιας κλάσης (πλήρης ομοιογένεια): Μήκος = 3, Βράγχια = Ναι και Ρύγχος = 'Οχι. Θα διαλέξουμε τη συνθήκη που ικανοποιεί τα περισσότερα παραδείγματα, δηλαδή τη συνθήκη Βράγχια = Ναι.

Συνθήκη	Ναι	'Οχι
Μήκος = 3	2	0
Μήκος = 4	1	3
Μήκος = 5	2	2
Βράγχια = Ναι	0	4
Βράγχια = 'Οχι	5	1
Ρύγχος = Ναι	5	3
Ρύγχος = 'Οχι	0	2
Δόντια = Λίγα	2	1
Δόντια = Πολλά	3	4

Πίνακας 5.2: Οι συνθήκες του συνόλου δεδομένων του Πίνακα 5.1 με τον αριθμό κάλυψης για την κλάση Δελφίνι.

Η επιλογή της βέλτιστης συνθήκης όπως περιγράφεται παραπάνω, γίνεται μέσω της συνάρτησης BestLiteral στη γραμμή 14 στον αλγόριθμο 1. Η ομοιογένεια μπορεί να υπολογιστεί με τους τύπους της Εντροπίας (3.1) ή του Gini (3.6). Στην πραγματικότητα, όποιον τύπο και να διαλέξουμε το αποτέλεσμα θα είναι το ίδιο. Σε αντίθεση με τα δενδρικά μοντέλα όπου γίνεται υπολογισμός της μέσης τιμής της ανομοιογένειας των διαφορετικών τιμών ενός χαρακτηριστικού, στην περίπτωση των κανόνων μάθησης ελέγχουμε την ανομοιογένεια μιας συγκεκριμένης συνθήκης. Τα αποτελέσματα δεν είναι ίδια ως προς την τιμή της ανομοιογένειας αλλά ως προς το ποια συνθήκη είναι η καλύτερη.

Δεδομένου ότι τα παραδείγματα που καλύπτονται από την πρώτη συνθήκη είναι ομογενή, δεν προχωράμε σε προσθήκη περαιτέρω συνθηκών και επιστρέφουμε τον κανόνα με την παραπάνω συνθήκη και με το συμπέρασμα που αντιστοιχεί στην κλάση που ανήκουν τα παραδείγματα αυτά:

Αν Βράγχια = Ναι τότε Δελφίνι = 'Οχι

Αφαιρούμε τα 4 παραδείγματα, τα οποία καλύπτει ο κανόνας αυτός (με κόκκινο χρώμα στον Πίνακα 5.3) και επαναλαμβάνουμε τη διαδικασία με τα εναπομέιναντα παραδείγματα. Ο πίνακας 5.4 καταγράφει τον αριθμό

των παραδειγμάτων για τις δύο τιμές (Ναι, Όχι) της κλάσης δελφίνι και για κάθε συνθήκη που προκύπτει από τα 6 εναπομείναντα παραδείγματα.

Μήκος	Βράγχια	Ρύγχος	Δόντια	Δελφίνι
3	Όχι	Ναι	Πολλά	Ναι
4	Όχι	Ναι	Πολλά	Ναι
3	Όχι	Ναι	Λίγα	Ναι
5	Όχι	Ναι	Πολλά	Ναι
5	Όχι	Ναι	Λίγα	Ναι
5	Ναι	Ναι	Πολλά	Όχι
4	Ναι	Ναι	Πολλά	Όχι
5	Ναι	Όχι	Πολλά	Όχι
4	Ναι	Όχι	Πολλά	Όχι
4	Όχι	Ναι	Λίγα	Όχι

Πίνακας 5.3: Το σύνολο δεδομένων του Πίνακα 5.1. Με κόκκινο τα παραδείγματα που καλύπτονται από τον πρώτο κανόνα, με μπλε αυτά που καλύπτονται από τον δεύτερο κανόνα και με πράσινο αυτά του τρίτου κανόνα.

Συνθήκη	Ναι	Όχι
Μήκος = 3	2	0
Μήκος = 4	1	1
Μήκος = 5	2	0
Βράγχια = Όχι	5	1
Ρύγχος = Ναι	5	1
Δόντια = Λίγα	2	1
Δόντια = Πολλά	3	0

Πίνακας 5.4: Οι συνθήκες του συνόλου δεδομένων του Πίνακα 5.1 με τον αριθμό κάλυψης για την κλάση Δελφίνι μετά τον πρώτο κανόνα.

Από το πίνακα παρατηρούμε πως οι συνθήκες με τη μεγαλύτερη ομοιογένεια είναι οι Μήκος=3, Μήκος=5 και Δόντια=Πολλά, εκ των οποίων η τελευταία καλύπτει τα περισσότερα παραδείγματα. Επομένως, ο δεύτερος κανόνας θα περιλαμβάνει και αυτός μόνο την παραπάνω συνθήκη:

$$\text{Αν Δόντια = Πολλά τότε Δελφίνι = Ναι}$$

Αν εξετάσουμε τον δεύτερο κανόνα από μόνο του παρατηρούμε πως καλύπτει και παραδείγματα που αφαιρέθηκαν από τον προηγούμενο κανόνα. Όπως αναφέραμε και στην αρχή, στα ταξινομένα σύνολα κανόνων, παίζει ρόλο η σειρά των κανόνων. Ο δεύτερος κανόνας, θα εξεταστεί μόνο αν ο πρώτος δεν ικανοποιηθεί. Επομένως, για την ένωση των δύο κανόνων χρησιμοποιούμε το ”αλλιώς” (else) και η λίστα κανόνων μέχρι στιγμής γίνεται:

$$\begin{aligned} \text{Αν Βράγχια = Ναι τότε Δελφίνι = Όχι} \\ \text{Αλλιώς Αν Δόντια = Πολλά τότε Δελφίνι = Ναι} \end{aligned}$$

Επαναλαμβάνουμε τη διαδικασία αφαιρώντας τα 3 παραδείγματα που καλύπτει ο δεύτερος κανόνας (με μπλε χρώμα στον Πίνακα 5.3). Ο Πίνακας 5.5 καταγράφει τον αριθμό των παραδειγμάτων για κάθε τιμής της κλάσης δελφίνι και για κάθε συνθήκη που προκύπτει από τα 3 εναπομείναντα παραδείγματα. Παρατηρούμε

Συνθήκη	Ναι	Όχι
Μήκος = 3	1	0
Μήκος = 4	0	1
Μήκος = 5	1	0
Βράγχια = 'Όχι	2	1
Ρύγχος = Ναι	2	1
Δόντια = Λίγα	2	1

Πίνακας 5.5: Οι συνθήκες του συνόλου δεδομένων του Πίνακα 5.1 με τον αριθμό κάλυψης για την κλάση Δελφίνι μετά τον δεύτερο κανόνα.

πως σε αυτήν την περίπτωση έχουμε τρεις ομοιογενείς συνθήκες (Μήκος = 3, Μήκος = 4 και Μήκος = 5) με τον ίδιο αριθμό παραδειγμάτων. Θα επιλέξουμε τη συνθήκη Μήκος = 4 καθώς με αυτήν την επιλογή στο επόμενο βήμα θα απομείνουν δύο παραδείγματα όπου η κλάση Δελφίνι έχει την ίδια τιμή ίση με Ναι (με μαύρο χρώμα στον πίνακα 5.3) και η μάθηση κανόνων μπορεί να ολοκληρωθεί. Η τελική λίστα με τους κανόνες είναι η εξής:

Αν Βράγχια = Ναι τότε Δελφίνι = 'Όχι
Αλλιώς Αν Δόντια = Πολλά τότε Δελφίνι = Ναι
Αλλιώς Αν Μήκος = 4 τότε Δελφίνι = 'Όχι
Αλλιώς Δελφίνι = Ναι

Ο αλγόριθμος 1 μπορεί να παραμετροποιηθεί ώστε να καλύπτει και άλλες περιπτώσεις συνόλων δεδομένων. Σε περιπτώσεις όπου το σύνολο δεδομένων περιέχει παραδείγματα με θόρυβο μπορούμε να περιορίσουμε την ανάπτυξη των κανόνων σε ένα συγκεκριμένο μέγεθος αλλάζοντας τον έλεγχο στη γραμμή 3, όπου η επανάληψη θα σταματάει αν το σύνολο των κανόνων φτάσει σε μέγεθος έναν προκαθορισμένο αριθμό. Ένας τετοιος περιορισμός μειώνει την πιθανότητα υπερπροσαρμογής στα δεδομένα. Αντίστοιχα, στη διαδικασία μάθησης κανόνων μπορούμε να αλλάξουμε τη συνθήκη για απόλυτη ομοιογένεια (γραμμή 13) με μια λιγότερο απόλυτη συνθήκη, όπως ορισμός κατωφλιού στην εντροπία ή περιορισμός των παραδειγμάτων που πρέπει να ικανοποιούν τη συνθήκη του κανόνα.

5.1.2 Μη ταξινομημένα σύνολα κανόνων

Τα μη ταξινομημένα σύνολα κανόνων (unordered rule sets) προσφέρουν μεγαλύτερη ευελιξία καθώς η σειρά με την οποία εξετάζονται οι κανόνες δεν επηρεάζει το αποτέλεσμα. Μπορούμε να μετατρέψουμε τη λίστα κανόνων που προέκυψε στο προηγούμενο παράδειγμα σε ένα μη ταξινομημένο σύνολο κανόνων προσθέτοντας σε κάθε κανόνα την άρνηση των κανόνων που προηγούνται του κανόνα αυτού στη λίστα. Επομένως, το ταξινομημένο σύνολο κανόνων του παραδείγματος της προηγούμενης ενότητας μετατρέπεται σε:

Αν Βράγχια = Ναι τότε Δελφίνι = 'Όχι
Αν Βράγχια = 'Όχι και Δόντια = Πολλά τότε Δελφίνι = Ναι
Αν Βράγχια = 'Όχι και Δόντια = Λίγα και Μήκος = 4 τότε Δελφίνι = 'Όχι
Αν Βράγχια = 'Όχι και Δόντια = Λίγα και Μήκος ≠ 4 τότε Δελφίνι = Ναι

Παρά την ευελιξία που προσφέρουν τα μη ταξινομημένα σύνολα κανόνων, καθιστούν τους κανόνες πιο περίπλοκους από ότι τα ταξινομημένα σύνολα κανόνων, μειώνοντας την ερμηνευσιμότητα τους. Η δημιουργία μη ταξινομημένων συνόλων γίνεται ακόμη πιο περίπλοκη στην περίπτωση που οι κανόνες αποτελούνται από περισσότερες συνθήκες.

Μη ταξινομημένα σύνολα κανόνων μπορούν να δημιουργηθούν και κατευθείαν από ένα σύνολο δεδομένων. Ο αλγόριθμος 2 δείχνει τη διαδικασία μάθησης μη ταξινομημένων συνόλων κανόνων. Κύριο στοιχείο αυτού

Algorithm 2 Δημιουργία μη ταξινομημένου συνόλου κανόνων

```

1: procedure LEARNRULESET( $D$ )
2:    $R = []$  ▷ Κενό σύνολο κανόνων
3:   for every class  $C_i$  do
4:      $D_i \leftarrow D$ 
5:     while  $D_i$  contains examples of  $C_i$  do
6:        $r \leftarrow \text{LearnRule}(D_i, C_i)$  ▷ Μάθηση κανόνα
7:        $R \leftarrow R \cup \{r\}$  ▷ Προσθήκη κανόνα στο σύνολο
8:        $D_i \leftarrow D_i \setminus \{x \in C_i | x \text{ covered by } r\}$  ▷ Αφαίρεση από το σύνολο δεδομένων των παραδειγμάτων της κλάσης  $C_i$  που καλύπτονται από τον κανόνα  $r$ 
9:     end while
10:   end for
11:   return  $R$ 
12: end procedure
13: procedure LEARNRULE( $D, C$ )
14:    $b \leftarrow \text{true}$  ▷ Αρχικοποίηση κανόνα που ικανοποιείται πάντα
15:    $L \leftarrow \text{set of available literals}$  ▷ Σύνολο όλων των συνθηκών
16:   while not Homogeneous( $D$ ) do
17:      $l \leftarrow \text{BestLiteral}(L, D, C)$  ▷ Εύρεση καλύτερης συνθήκης
18:      $b \leftarrow b \wedge l$  ▷ Προσθήκη συνθήκης στον κανόνα
19:      $D \leftarrow \{x \in D | x \text{ covered by } b\}$ 
20:      $L \leftarrow L \setminus \{l' \in L | l' \text{ uses same feature as } l\}$  ▷ Αφαίρεση όλων των συνθηκών από το σύνολο που αφορούν το ίδιο χαρακτηριστικό με την συνθήκη  $l$ 
21:   end while
22:   return "if  $b$  then Class= $C$ "
23: end procedure

```

του αλγορίθμου είναι πως μαθαίνουμε κανόνες για κάθε κλάση ξεχωριστά. Η φιλοσοφία του αλγορίθμου είναι η ίδια με τον αλγόριθμο μάθησης λίστας κανόνων, δηλαδή διαχώρισε και βασίλευε.

Για την καλύτερη κατανόηση του αλγορίθμου μάθησης μη ταξινομημένων συνόλων κανόνων χρησιμοποιούμε το ίδιο παράδειγμα με προηγουμένων (Πίνακας 5.1). Ξεκινάμε την επαναληπτική διαδικασία του αλγορίθμου για την τιμή Ναι της κλάσης Δελφίνι. Για κάθε συνθήκη που υπάρχει στο σύνολο δεδομένων μας καταμετρούμε τον αριθμό των παραδειγμάτων με κλάση Δελφίνι = Ναι, το πλήθος των παραδειγμάτων που καλύπτονται από τη συνθήκη αυτή και το λόγο των δύο αυτών αριθμών, δηλαδή την ακρίβεια (precision), όπως φαίνεται στον Πίνακα 5.6.

Σε κάθε επανάληψη για τη μάθηση του κανόνα επιλέγουμε τη συνθήκη με τη μεγαλύτερη ακρίβεια. Στην πρώτη επανάληψη αυτή η συνθήκη είναι Μήκος = 3, όπου όλα τα παραδείγματα που καλύπτει αυτή η συνθήκη είναι για την κλάση Δελφίνι=Ναι. Επομένως, η συνθήκη αυτή ολοκληρώνει και τον κανόνα, ο οποίος είναι:

Αν Μήκος = 3 τότε Δελφίνι = Ναι

Αφαιρούμε τα παραδείγματα που καλύπτει ο κανόνας αυτός (με κόκκινο χρώμα στον Πίνακα 5.7) και επαναλαμβάνουμε τη διαδικασία για τα υπόλοιπα παραδείγματα. Στον Πίνακα 5.8 παρατηρούμε πως η συνθήκη Βράγχια = 'Όχι έχει τη μεγαλύτερη ακρίβεια και την προσθέτουμε στον κανόνα. Ωστόσο, ο κάνονας μέχρι στιγμής καλύπτει 4 παραδείγματα όπου ένα παράδειγμα ανήκει στην κλάση Δελφίνι='Όχι (το τελευταίο παράδειγμα στον Πίνακα 5.7). Για το λόγο αυτό επαναλαμβάνουμε τη διαδικασία μάθησης ενός κανόνα μόνο για τα 4 αυτά παραδείγματα.

Ο Πίνακας 5.9 παρουσιάζει την κάλυψη για τα 4 αυτά παραδείγματα. Η ακρίβεια μεγιστοποιείται σε δύο περιπτώσεις: για Μήκος = 5 και για Δόντια = Πολλά. Σε αυτήν την περίπτωση επιλέγουμε τυχαία μία από τις

Συνθήκη	Ναι	Κάλυψη	Ναι/Κάλυψη
Μήκος = 3	2	2	1.00
Μήκος = 4	1	4	0.25
Μήκος = 5	2	4	0.50
Βράγχια = Ναι	0	4	0.00
Βράγχια = Όχι	5	6	0.83
Ρύγχος = Ναι	5	8	0.63
Ρύγχος = Όχι	0	2	0.00
Δόντια = Λίγα	2	3	0.67
Δόντια = Πολλά	3	7	0.43

Πίνακας 5.6: Οι συνθήκες του συνόλου δεδομένων του Πίνακα 5.1 με τον αριθμό κάλυψης για την κλάση Δελφίνι=Ναι (1η επανάληψη).

Μήκος	Βράγχια	Ρύγχος	Δόντια	Δελφίνι
3	Όχι	Ναι	Πολλά	Ναι
4	Όχι	Ναι	Πολλά	Ναι
3	Όχι	Ναι	Λίγα	Ναι
5	Όχι	Ναι	Πολλά	Ναι
5	Όχι	Ναι	Λίγα	Ναι
5	Ναι	Ναι	Πολλά	Όχι
4	Ναι	Ναι	Πολλά	Όχι
5	Ναι	Όχι	Πολλά	Όχι
4	Ναι	Όχι	Πολλά	Όχι
4	Όχι	Ναι	Λίγα	Όχι

Πίνακας 5.7: Το σύνολο δεδομένων του Πίνακα 5.1. Με κόκκινο τα παραδείγματα που καλύπτονται από τον πρώτο κανόνα, με μπλε τα παραδείγματα του δεύτερου κανόνα και με πράσινο τον τρίτου κανόνα για την κλάση Δελφίνι = Ναι.

δύο συνθήκες (Μήκος = 5). Η επιλογή της συνθήκης αυτής μας οδηγεί σε ομοιογενή κατάσταση. Ο δεύτερος κανόνας είναι:

Αν Βράγχια = Όχι και Μήκος = 5 τότε Δελφίνι = Ναι

Η παραπάνω διαδικασία επαναλαμβάνεται μέχρι να καλυφθούν όλα τα παραδείγματα της κλάσης Δελφίνι = Ναι. Το σύνολο κανόνων για την κλάση Δελφίνι = Ναι είναι:

Αν Μήκος = 3 τότε Δελφίνι = Ναι

Αν Βράγχια = Όχι και Μήκος = 5 τότε Δελφίνι = Ναι

Αν Βράγχια = Όχι και Δόντια = Πολλά τότε Δελφίνι = Ναι

Η ίδια διαδικασία που έγινε παραπάνω επαναλαμβάνεται για την κλάση Δελφίνι = Όχι και προκύπτει το παρακάτω σύνολο κανόνων:

Αν Βράγχια = Ναι τότε Δελφίνι = Όχι

Αν Μήκος = 4 και Δόντια = Λίγα τότε Δελφίνι = Όχι

Συνθήκη	Ναι	Κάλυψη	Ναι/Κάλυψη
Μήκος = 4	1	4	0.25
Μήκος = 5	2	4	0.50
Βράγχια = Ναι	0	4	0.00
Βράγχια = 'Οχι	3	4	0.75
Ρύγχος = Ναι	3	6	0.50
Ρύγχος = 'Οχι	0	2	0.00
Δόντια = Λίγα	1	2	0.50
Δόντια = Πολλά	2	6	0.33

Πίνακας 5.8: Οι συνθήκες του συνόλου δεδομένων του Πίνακα 5.1 με τον αριθμό κάλυψης για την κλάση Δελφίνι=Ναι (2η Επανάληψη - 1η συνθήκη).

Συνθήκη	Ναι	Κάλυψη	Ναι/Κάλυψη
Μήκος = 4	1	2	0.50
Μήκος = 5	2	2	1.00
Ρύγχος = Ναι	3	4	0.75
Δόντια = Λίγα	1	2	0.50
Δόντια = Πολλά	2	2	1.00

Πίνακας 5.9: Οι συνθήκες του συνόλου δεδομένων του Πίνακα 5.1 με τον αριθμό κάλυψης για την κλάση Δελφίνι=Ναι (2η Επανάληψη - 2η συνθήκη).

Η ένωση των δύο συνόλων για τις κλάσεις Δελφίνι = Ναι και Δελφίνι = 'Οχι είναι το μη ταξινομημένο σύνολο κανόνων του συνόλου δεδομένων.

Έστω ένα νέο παράδειγμα για ένα κύτος με χαρακτηριστικά Βράγχια = Ναι, Μήκος = 3, Δόντια = Λίγα και Ρύγχος = Ναι. Σύμφωνα με το μοντέλο μας, το παράδειγμα αυτό καλύπτεται από τους κανόνες "Αν Μήκος = 3 τότε Δελφίνι = Ναι" και "Αν Βράγχια = Ναι τότε Δελφίνι = 'Οχι". Σε αυτήν την περίπτωση δεν μπορούμε να επιλέξουμε την κλάση στην οποία ανήκει το παράδειγμα αυτό. Αυτό συμβαίνει λόγω της απλότητας του παραδείγματος που αναλύουμε. Στη γενική περίπτωση ένας κανόνας συνοδεύεται από μία ακρίβεια. Η ακρίβεια αυτή προέρχεται από τους περιορισμούς που ορίζουμε στην ομοιογένεια κάθε κανόνα, ενώ για την ταξινόμηση παραδειγμάτων επιλέγουμε τον κανόνα με την υψηλότερη ακρίβεια ή τους κανόνες με το μεγαλύτερο άθροισμα ακρίβειας.

5.1.2.1 Χρήση παραδείγματος σπόρου

Μια παραλλαγή του παραπάνω αλγορίθμου μάθησης κανόνων είναι ο περιορισμός των συνθηκών που ελέγχουμε σε κάθε επανάληψη για τη δημιουργία κανόνων. Αυτό μπορεί να γίνει με τη χρήση ενός **παραδείγματος σπόρου** (seed example), δηλαδή ενός παραδείγματος που επιλέγουμε στην αρχή της διαδικασίας και μας περιορίζει τις συνθήκες σύμφωνα με τις τιμές των χαρακτηριστικών του. Επιλέγοντας ως παράδειγμα σπόρο το πρώτο παράδειγμα του Πίνακα 5.1, οι συνθήκες που πρέπει να ελεγχθούν σε κάθε επανάληψη περιορίζονται σε 4 όπως φαίνεται και στον Πίνακα 5.10. Χρησιμοποιώντας ένα παραδειγμα σπόρο μειώνεται η πολυπλοκότητα του αλγορίθμου αλλά αυξάνεται η πιθανότητα μείωσης της ακρίβειας του μοντέλου μας.

5.1.2.2 Ακτινωτή αναζήτηση

Μια δεύτερη παραλλαγή του αλγορίθμου μάθησης κανόνων είναι η χρήση **ακτινωτής αναζήτησης** (beam search), όπου κατά τη διαδικασία μάθησης κανόνα εξετάζουμε παραπάνω από μία συνθήκη. Στο παραπάνω

Συνθήκη	Ναι	Κάλυψη	Ναι/Κάλυψη
Μήκος = 3	2	2	1.00
Μήκος = 4	1	4	0.25
Μήκος = 5	2	4	0.50
Βράγχια = Ναι	0	4	0.00
Βράγχια = Όχι	5	6	0.83
Ρύγχος = Ναι	5	8	0.63
Ρύγχος = Όχι	0	2	0.00
Δόντια = Λίγα	2	3	0.67
Δόντια = Πολλά	3	7	0.43

Πίνακας 5.10: Οι συνθήκες του συνόλου δεδομένων του Πίνακα 5.1 με τον αριθμό κάλυψης για την κλάση Δελφίνι=Ναι. Ο αριθμός των συνθηκών περιορίζεται με την χρήση του πρώτου παραδείγματος ως παράδειγμα σπόρος.

παράδειγμα πέρα από τη συνθήκη Μήκος = 3, που έχει την μεγαλύτερη ακρίβεια (1.0), επιλέγουμε και τη συνθήκη με τη δεύτερη μεγαλύτερη ακρίβεια που είναι τα Βράγχια = Όχι (0.83). Αυτή η επιλογή κάνει τη διαδικασία λιγότερο άπληστη. Μετά την πρώτη επανάληψη, για την κλάση Δελφίνι = Ναι, του αλγορίθμου μάθησης κανόνα καταλήγουμε με δύο διαφορετικούς κανόνες βασισμένους στις παραπάνω δύο συνθήκες. Οι κανόνες είναι:

Αν Μήκος = 3 τότε Δελφίνι = Ναι

Αν Βράγχια = Όχι και Δόντια = Πολλά τότε Δελφίνι = Ναι

Θα πρέπει να επιλέξουμε έναν από τους δύο κανόνες για να συνεχίζουμε τη διαδικασία μάθησης του συνόλου κανόνων. Ο πίνακας 5.11 παρουσιάζει τα παραδείγματα που καλύπτει κάθε κανόνας. Επιλέγουμε τον δεύτερο κανόνα (μπλε) καθώς καλύπτει 3 παραδείγματα σε αντίθεση με τον πρώτο κανόνα που καλύπτει 2. Η διαδικασία μάθησης του συνόλου κανόνων συνεχίζεται με τον ίδιο τρόπο όπου σε κάθε επανάληψη ελέγχουμε δύο διαφορετικές αρχικές συνθήκες και επιλέγουμε κάθε φορά την πιο κατάλληλη.

Μήκος	Βράγχια	Ρύγχος	Δόντια	Δελφίνι
3	Όχι	Ναι	Πολλά	Ναι
4	Όχι	Ναι	Πολλά	Ναι
3	Όχι	Ναι	Λίγα	Ναι
5	Όχι	Ναι	Πολλά	Ναι
5	Όχι	Ναι	Λίγα	Ναι
5	Ναι	Ναι	Πολλά	Όχι
4	Ναι	Ναι	Πολλά	Όχι
5	Ναι	Όχι	Πολλά	Όχι
4	Ναι	Όχι	Πολλά	Όχι
4	Όχι	Ναι	Λίγα	Όχι

Πίνακας 5.11: Το σύνολο δεδομένων του Πίνακα 5.1. Με κόκκινο τα παραδείγματα που καλύπτονται από τον πρώτο κανόνα, με μπλε τα παραδείγματα του δεύτερου κανόνα και με μωβ τα παραδείγματα που καλύπτονται και από τους δύο κανόνες.

5.2 Εξελικτική Μάθηση Κανόνων

5.2.1 Γενετικοί αλγόριθμοι

Οι γενετικοί αλγόριθμοι (genetic algorithms) αποτελούν μέθοδο επίλυσης προβλημάτων βελτιστοποίησης. Στηρίζονται στην προσομοίωση του φυσικού φαινομένου της εξέλιξης. Η διαδικασία της εξέλιξης ξεκινά από έναν τυχαίο πληθυσμό υποψήφιων λύσεων και προχωρά επαναληπτικά ως εξής: α) Αξιολογείται η καταλληλότητα (fitness) κάθε λύσης, β) επιλέγεται στοχαστικά ένα υποσύνολο των λύσεων με βάση την καταλληλότητα τους, και γ) από αυτό το υποσύνολο παράγεται η νέα γενιά υποψήφιων λύσεων μέσω διαδικασιών της βιολογικής εξέλιξης όπως η διασταύρωση (crossover) και η τυχαία μετάλλαξη (random mutation). Η νέα γενιά που προκύπτει αποτελεί τον πληθυσμό για την επόμενη επανάληψη της διαδικασίας. Η διαδικασία τερματίζει είτε όταν ολοκληρωθεί ένας μέγιστος αριθμός επαναλήψεων, είτε όταν βρεθεί κάποια λύση με ικανοποιητική καταλληλότητα.

Τα βασικά στοιχεία/παράμετροι ενός τυπικού γενετικού αλγόριθμου είναι η συνάρτηση καταλληλότητας (fitness function), ένα όριο της για τον τερματισμό του αλγορίθμου (*threshold*), το μέγεθος του πληθυσμού, το ποσοστό του πληθυσμού που αντικαθίσταται σε κάθε βήμα (r) και ο ρυθμός μετάλλαξης (m).

Ένας τυπικός γενετικός αλγόριθμος ξεκινά με τη δημιουργία ενός αριθμού, p , χρωμοσωμάτων, τα οποία αποτελούν τον αρχικό πληθυσμό. Το μέγεθος του πληθυσμού εξαρτάται από τη φύση του προβλήματος, αλλά συνήθως φτάνει τις εκατοντάδες ή χιλιάδες χρωμοσωμάτων. Σχεδόν πάντα ο αρχικός πληθυσμός δημιουργείται τυχαία προκειμένου να καλύψει όλο το εύρος των δυνατών χρωμοσωμάτων (όλο το χώρο αναζήτησης). Ωστόσο, όταν γνωρίζουμε ότι σε κάποιες περιοχές του χώρου αυτού ενδέχεται να υπάρχουν καλές λύσεις, μπορούμε να επιλέξουμε περισσότερα μέλη του αρχικού πληθυσμού από εκεί.

Σε κάθε επανάληψη παράγεται μια νέα γενιά p λύσεων που προκύπτει από τον υπάρχοντα πληθυσμό ως εξής. Ένας αριθμός $(1 - r)p$, όπου $0 < r \leq 1$, των μελών του νέου πληθυσμού θα προκύψει από αντιγραφή μελών του υπάρχοντος πληθυσμού. Η επιλογή των μελών αυτών γίνεται στοχαστικά με βάση την καταλληλότητα τους. Υπάρχουν διάφορες μέθοδοι, οι οποίες περιγράφονται στην παρακάτω ενότητα. Τα υπόλοιπα $r p$ μέλη θα προκύψουν εφαρμόζοντας τον γενετικό τελεστή της διασταύρωσης. Τέλος, τα μέλη που προκύπτουν υπόκεινται με κάποια πιθανότητα σε μετάλλαξη.

5.2.2 Μέθοδοι επιλογής μελών

Μία από τις πιο διαδεδομένες μεθόδους επιλογής των λύσεων είναι η επιλογή ανάλογα με την καταλληλότητα (fitness proportionate selection), γνωστή και ως μέθοδος της ρουλέτας (roulette wheel), όπου η πιθανότητα επιλογής μιας λύσης h_i είναι ευθέως ανάλογη της καταλληλότητας της:

$$P(h_i) = \frac{\text{fitness}(h_i)}{\sum_{j=1}^p \text{fitness}(h_j)} \quad (5.1)$$

Η μέθοδος της ρουλέτας παρουσιάζει πρόβλημα, είτε όταν όλα τα μέλη του πληθυσμού έχουν παραπλήσια καταλληλότητα, είτε όταν ένα ή δύο μέλη έχουν πολύ μεγαλύτερη καταλληλότητα από τα υπόλοιπα. Στην πρώτη περίπτωση, η εξέλιξη εκφυλίζεται σε τυχαία αναζήτηση με γενετική παρέκκλιση (genetic drift), με τελικό αποτέλεσμα τη σύγκλιση του πληθυσμού σε κάποια χρωμοσώματα, χωρίς όμως αυτά να είναι απαραίτητα πιο καταλληλα. Στη δεύτερη περίπτωση, ο πληθυσμός θα συγκλίνει πολύ γρήγορα στο μέλος με την πολύ μεγαλύτερη καταλληλότητα, με αποτέλεσμα να γίνει μηδαμινή εξερεύνηση του χώρου αναζήτησης.

Μια εναλλακτική μέθοδος που αντιμετωπίζει τα παραπάνω προβλήματα είναι η επιλογή κατάταξης (rank selection). Στην επιλογή κατάταξης, οι λύσεις ταξινομούνται σε σειρά με βάση την καταλληλότητά τους και η πιθανότητα επιλογής μιας λύσης γίνεται πλέον ανάλογη της σειράς της στην κατάταξη αυτή.

Για παράδειγμα, έστω ότι ο πληθυσμός αποτελείται από 5 λύσεις, των οποίων η καταλληλότητα φαίνεται στον Πίνακα 5.12. Η πιθανότητα επιλογής της κάθε λύσης, με βάση τις μεθόδους της ρουλέτας και της κατάταξης φαίνονται στον Πίνακα 5.13.

	h_1	h_2	h_3	h_4	h_5
καταλληλότητα	2	3	1	40	4

Πίνακας 5.12: 5 λύσεις και η καταλληλότητας τους.

	h_1	h_2	h_3	h_4	h_5
Μέθοδος ρουλέτας	2/50	3/50	1/50	40/50	4/50
Μέθοδος κατάταξης	2/15	3/15	1/15	5/15	4/15

Πίνακας 5.13: 5 λύσεις και η καταλληλότητας τους.

Ακόμα μια διαδεδομένη μέθοδος επιλογής λύσεων ονομάζεται επιλογή τουρνουά (tournament selection) και λειτουργεί ως εξής. Αρχικά επιλέγονται τυχαία 2 λύσεις. Από αυτές, επιλέγεται η καταλληλότερη με προκαθορισμένη πιθανότητα q και η άλλη με πιθανότητα $(1 - q)$. Η μέθοδος αυτή έχει το πλεονέκτημα ότι εκτελείται γρηγορότερα και μπορεί να εκτελεστεί εύκολα παράλληλα, αφού δεν απαιτεί τον υπολογισμό της καταλληλότητας όλων των μελών του πληθυσμού.

Ανεξάρτητα από τη μέθοδο επιλογής, είναι συχνά χρήσιμο, να διατηρείται η βέλτιστη τρέχουσα λύση στην επόμενη γενιά. Η στρατηγική αυτή ονομάζεται ελιτισμός και εγγυάται ότι δεν θα χαθεί η βέλτιστη τρέχουσα λύση ενώ συνήθως έχει ως αποτέλεσμα πιο ομαλές βελτιώσεις της καταλληλότητας του πληθυσμού.

Η δημιουργία απογόνων στους ΓΑ καθορίζεται από ένα σύνολο τελεστών που ανασυνδυάζουν (recombine) και μεταλλάσσουν (mutate) επιλεγμένα μέλη του πληθυσμού. Οι τελεστές αυτοί αποτελούν εξιδανικευμένες παραλλαγές των γενετικών λειτουργιών στη βιολογική εξέλιξη. Οι πιο συνηθισμένοι είναι η διασταύρωση και η μετάλλαξη

5.2.3 Εξέλιξη και μάθηση

Στη φύση, πολλοί οργανισμοί μαθαίνουν να προσαρμόζονται σε σημαντικό βαθμό κατά τη διάρκεια της ζωής τους. Ταυτόχρονα όμως, βιολογικές και κοινωνικές διαδικασίες επιτρέπουν στα έμβια όντα να προσαρμόζονται σε χρονική διάρκεια πολλών γενεών. Τίθεται λοιπόν το ερώτημα αν υπάρχει κάποια σχέση που να συνδέει τις δύο αυτές διαδικασίες. Υπάρχουν δύο σχετικές θεωρίες, οι οποίες περιγράφονται στη συνέχεια αυτής της ενότητας.

Ο Lamarck υποστηρίζει την άποψη πως η εξέλιξη των ειδών σε χρονική διάρκεια πολλών γενεών επηρεάζεται άμεσα από τις εμπειρίες των οργανισμών του είδους κατά τη διάρκεια της ζωής τους. Συγκεκριμένα πρότεινε ότι οι εμπειρίες της ζωής ενός οργανισμού επηρεάζουν άμεσα το γενετικό υλικό των απογόνων του.

Πρόκειται για μια ενδιαφέρουσα άποψη, καθώς με τον τρόπο αυτό η διαδικασία της εξέλιξης θα μπορούσε να επιταχυνθεί. Ωστόσο η θεωρία αυτή δεν είναι επιστημονικά αποδεκτή. Ισχύει η άποψη ότι το γενετικό υλικό ενός οργανισμού δεν επηρεάζεται καθόλου από τις εμπειρίες των βιολογικών γονέων του. Ωστόσο, μελέτες έχουν δείξει ότι αυτού του είδους η θεωρία μπορεί να βελτιώσει την επίδοση των γενετικών αλγορίθμων.

Ο Baldwin υποστηρίζει την άποψη πως όταν ένα είδος εξελίσσεται σε ένα περιβάλλον το οποίο αλλάζει, τότε θα υπάρχει μια εξελικτική πίεση υπέρ των μελών του πληθυσμού με ικανότητες μάθησης. Έστω για παράδειγμα ότι εμφανίζεται ένα νέο είδος κυνηγού του είδους που μελετάμε. Τα πιο ικανά (δυνατά, γρήγορα, κτλ) μέλη θα επιβιώσουν και θα μεταδώσουν το γενετικό τους υλικό στις επόμενες γενιές. Το ίδιο όμως θα κάνουν και τα μέλη που έχουν ανεπτυγμένες ικανότητες μάθησης, καθώς θα μάθουν πως να αποφεύγουν αυτό το νέο είδος κυνηγού.

Τα μέλη που μπορούν να μαθαίνουν ικανότητες εξαρτώνται λιγότερο από το γενετικό τους υλικό. Άρα είναι λογικό να εμφανίζουν μεγαλύτερη ποικιλία στο γενετικό τους υλικό. Έτσι αυξάνεται η συνολική ποικιλία του πληθυσμού και τελικά επιτυγχάνεται πιο γρήγορη εξέλιξη, αφού αποφεύγεται ο συνωστισμός. Το φαινόμενο Baldwin παρέχει έναν έμμεσο μηχανισμό επηρεασμού της εξέλιξης μέσω της μάθησης κατά τη διάρκεια της ζωής.

5.2.4 Εξέλιξη συνόλων κανόνων

Μπορούμε να εξελίξουμε ένα σύνολο κανόνων αν το αναπαραστήσουμε με μια ακολουθία από 0 και 1. Θα ξεκινήσουμε από την αναπαράσταση ενός κανόνα, και πιο συγκεκριμένα από την αναπαράσταση των συνθηκών του κανόνα.

Κάθε διαφορετική διακριτή τιμή κάθε μεταβλητής εισόδου μπορεί να αναπαρασταθεί με ένα bit. Έστω για παράδειγμα η μεταβλητή βράγχια με τιμές όχι και ναι. Η ακολουθία bit 10 αναπαραριστά την συνθήκη βράγχια=όχι, η ακολουθία 01 την συνθήκη βράγχια=ναι και η ακολουθία 11 την συνθήκη βράγχια=όχι ή βράγχια=ναι, δηλαδή στην πράξη την απουσία συνθήκης για την μεταβλητή βράγχια. Η ακολουθία 00 που περισσεύει δεν αντιστοιχεί σε κάποια αναπαράσταση συνθήκης για την μεταβλητή και επομένως, θα πρέπει αν προκύπτει να απορρίπτεται ως άκυρο χρωμόσωμα.

Για την αναπαράσταση συζεύξεων δύο ή περισσότερων συνθηκών μπορούμε να συνενώσουμε τις ακολουθίες τους. Έστω για παράδειγμα η ακόλουθη σύζευξη δύο συνθηκών: βράγχια=όχι και ράμφος=ναι. Θα την αναπαραστήσουμε ως 10 01.

Το συμπέρασμα ενός κανόνα θα μπορούσε να αναπαρασταθεί με τον ίδιο τρόπο και να συνενωθεί με τις ακολουθίες των συνθηκών. Για παράδειγμα ο κανόνας αν βράγχια=όχι και ράμφος=ναι τότε δελφίνι=ναι, θα μπορούσε να αναπαρασταθεί ως 10 01 01. Για να αποφευχθούν συμπεράσματα δίχως νόημα, όπως 00 και 11, είναι προτιμότερο να χρησιμοποιήσουμε ένα bit για το συμπέρασμα, αναπαριστώντας τον παραπάνω κανόνα ως 10 01 1.

Σύνολα κανόνων αναπαριστώνται συνενώνοντας τις ακολουθίες όλων των κανόνων που περιλαμβάνουν. Για να έχει νόημα ο τελεστής της διασταύρωσης, καταρχήν απαιτούμε ο κάθε κανόνας να έχει σταθερό μήκος ακολουθίας. Αυτό μπορεί να επιτευχθεί αν συμπεριλάβουμε στην αναπαράσταση όλες τις μεταβλητές εισόδου, ακόμα και αν δεν συμμετέχουν στις συνθήκες του κανόνα. Έστω για παράδειγμα οι ακόλουθες μεταβλητές εισόδου με τα σύνολα τιμών τους: μήκος {3, 4, 5}, βράγχια {όχι, ναι}, ράμφος {όχι, ναι}, δόντια {λίγα, πολλά}. Ο κανόνας αν βράγχια=όχι τότε δελφίνι=ναι θα αναπαρασταθεί ως 111 10 11 11 1.

Ως συνάρτηση καταλληλότητας μπορούμε να ορίσουμε την ακρίβεια πρόβλεψης του συνόλου κανόνων σε ένα σύνολο εκπαίδευσης. Αν θέλαμε να αποφύγουμε την υπερπροσαρμογή, θα μπορούσαμε να προσθέσουμε και κάποια ποινή στην συνάρτηση με βάση το πλήθος των κανόνων ή/και τον αριθμό των συνθηκών τους.

5.3 Περαιτέρω Μελέτη

Για την εξελικτική μάθηση κανόνων μπορεί να διαβάσει κανείς περισσότερα στο [1].

5.4 Ασκήσεις

- Έστω πρόβλημα δυαδικής ταξινόμησης με δύο δυαδικές μεταβλητές εισόδου και το ακόλουθο σύνολο εκπαίδευσης. Εφαρμόστε έναν αλγόριθμο μάθησης συνόλου ταξινομημένων κανόνων, δηλαδή λίστας κανόνων. Μετατρέψτε τη λίστα κανόνων σε σύνολο μη ταξινομημένων κανόνων. Εφαρμόστε έναν αλγόριθμο μάθησης συνόλου μη ταξινομημένων κανόνων.

x_1	x_2	y
1	1	+
1	1	+
1	0	-
0	0	+
0	1	-
0	1	-

2. Έστω το παρακάτω σύνολο δεδομένων εκπαίδευσης. Δώστε τον πρώτο κανόνα που θα μάθαινε ένας αλγόριθμος ακολουθιακής κάλυψης που μαθαίνει λίστες κανόνων.

x_1	x_2	y
1	2	0
2	0	1
1	1	0
0	2	1
1	0	1

3. Ο παρακάτω πίνακας δίνει την καταλληλότητα 5 συνόλων κανόνων που εξελίσσονται με γενετικό αλγόριθμο. Ποια η πιθανότητα επιλογής του συνόλου 2 με τη μέθοδο της ρουλέτας;

Σύνολο	1	2	3	4	5
Καταλληλότητα	15	10	20	5	50

4. Σε μια επιχείρηση, υπάρχουν δύο χαρακτηριστικά που επηρεάζουν αν ένας πελάτης θα παραγγείλει ή όχι από την ιστοσελίδα: Ο τύπος πελάτη που μπορεί να είναι απλός ή συνδρομητής (X1) και η συχνότητα επίσκεψης του χρήστη και παίρνει τιμές 1, 2, 3 και 4 (X2). Το αποτέλεσμα Υ είναι “Ναι” ή “Όχι”, που δηλώνει αν ο πελάτης έκανε παραγγελία ή όχι.

Έχετε τα παρακάτω δεδομένα:

X1(Τύπος Πελάτη)	X2(Συχνότητα Επίσκεψης)	Υ(Παραγγελία)
Απλός	1	Ναι
Συνδρομητής	2	Όχι
Συνδρομητής	1	Ναι
Απλός	2	Ναι
Απλός	3	Όχι
Απλός	4	Ναι
Συνδρομητής	3	Όχι

- α) Βρείτε τον πρώτο κανόνα που θα μάθει ένας αλγόριθμος μάθησης συνόλου ταξινομημένων κανόνων, αν ξεκινήσει με την πρώτη παρατήρηση.
β) Χρησιμοποιώντας τη μέθοδο των “μη ταξινομημένων κανόνων”, ποια θα είναι η πρόβλεψη του μοντέλου για έναν πελάτη τύπου συνδρομητή και συχνότητα επίσκεψης 2;

Βιβλιογραφία

- [1] Stefano. Nolfi. *Evolutionary robotics : the biology, intelligence, and technology of self-organizing machines*. eng. Intelligent robots and autonomous agents. Cambridge, Mass: MIT Press, 2000. ISBN: 0262140705.

ΚΕΦΑΛΑΙΟ 6

ΜΑΘΗΣΗ ΚΑΤΑ ΠΕΡΙΠΤΩΣΗ

Στο κεφάλαιο αυτό:

Θα παρουσιάσουμε ένα από τους πιο διαδεδομένους αλγορίθμους μηχανικής μάθησης - τον αλγόριθμο των k Πλησιέστερων Γειτόνων.

Σε αντίθεση με τις περισσότερες μεθόδους μάθησης με επίβλεψη που αναζητούν μια γενική περιγραφή των δεδομένων εκπαίδευσης, οι αλγόριθμοι της **μάθησης κατά περίπτωση** (instance-based learning) απλά αποθηκεύουν τα παραδείγματα του συνόλου εκπαίδευσης. Στο γεγονός αυτό οφείλεται και η εναλλακτική ονομασία αυτής της κατηγορίας αλγορίθμων μάθησης, η οποία είναι **μάθηση μέσω απομνημόνευσης** (memory-based learning). Η μάθηση αναβάλλεται μέχρι να χρειαστεί να γίνει πρόβλεψη για μια νέα περίπτωση. Για το λόγο αυτό, η μάθηση κατά περίπτωση καλείται και **αναβλητική μάθηση** (lazy learning). Η πρόβλεψη της τιμής της εξαρτημένης μεταβλητής μιας νέας περίπτωσης, στηρίζεται σε ένα υποσύνολο των αποθηκευμένων παραδειγμάτων που μοιάζουν με τη νέα περίπτωση.

Ακολουθεί η παρουσίαση του πιο διαδεδομένου αλγορίθμου μάθησης κατά περίπτωση.

6.1 Πλησιέστεροι Γείτονες

Ο αλγόριθμος των k **Πλησιέστερων Γειτόνων** (k Nearest Neighbor - k NN) κάνει πρόβλεψη για μια νέα περίπτωση με βάση τα k πλησιέστερα αποθηκευμένα παραδείγματα, όπου k είναι μια παράμετρος που καθορίζει ο χρήστης.

Αρχικά πρέπει να ανακτηθούν τα k πλησιέστερα παραδείγματα. Αυτό γίνεται υπολογίζοντας την απόσταση κάθε αποθηκευμένου παραδείγματος με την νέα περίπτωση, και επιλέγοντας τα k παραδείγματα με την μικρότερη απόσταση.

Έστω πρόβλημα μάθησης με επίβλεψη με n μεταβλητές εισόδου, σύνολο εκπαίδευσης $D = (\mathbf{x}^{(i)}, y^{(i)})$, όπου $i = 1 \dots n$ και $\mathbf{x}^{(i)} = [x_1^{(i)}, \dots, x_n^{(i)}]$, και μια νέα περίπτωση $\mathbf{x}' = [x'_1, \dots, x'_n]$. Τυπικά μέτρα απόστασης είναι:

- Η Ευκλείδεια απόσταση

$$d(\mathbf{x}^{(i)}, \mathbf{x}') = \sqrt{\sum_{j=1}^n (x_j^{(i)} - x'_j)^2} \quad (6.1)$$

- Η απόσταση Manhattan ή city-block

$$d(\mathbf{x}^{(i)}, \mathbf{x}') = \sum_{j=1}^n |x_j^{(i)} - x'_j| \quad (6.2)$$

- Η απόσταση Chebyshev

$$d(\mathbf{x}^{(i)}, \mathbf{x}') = \max_j (|x_j^{(i)} - x'_j|) \quad (6.3)$$

Στη συνέχεια γίνεται η πρόβλεψη. Ο αλγόριθμος k NN μπορεί να χρησιμοποιηθεί σε εργασίες τόσο ταξινόμησης όσο και παλινδρόμησης. Η πρόβλεψη της τιμής μιας νέας περίπτωσης \mathbf{x}' με βάση τα k πλησιέστερα παραδείγματα $(\bar{\mathbf{x}}^{(1)}, \bar{y}^{(1)}), \dots, (\bar{\mathbf{x}}^{(k)}, \bar{y}^{(k)})$ ισούται με:

- Τον μέσο όρο της τιμής της εξαρτημένης μεταβλητής στα k πλησιέστερα παραδείγματα για προβλήματα παλινδρόμησης

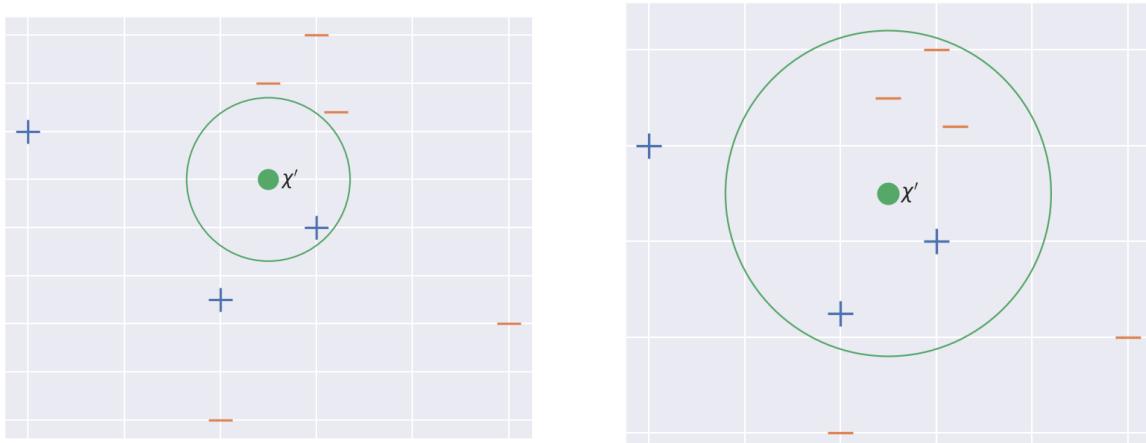
$$\hat{y}' = \frac{\sum_{i=1}^k \bar{y}^{(i)}}{k} \quad (6.4)$$

- Την πιο κοινή κατηγορία $0 \dots c-1$ ανάμεσα στα πλησιέστερα παραδείγματα για προβλήματα ταξινόμησης σε c κλάσεις

$$\hat{y}' = \arg \max_{j \in \{0,1,\dots,c-1\}} \sum_{i=1}^k \mathbb{1}[\bar{y}^{(i)} = j] \quad (6.5)$$

όπου $\mathbb{1}[\cdot]$ ισούται με 1 αν η παράσταση · είναι αληθής και με 0 αν είναι ψευδής.

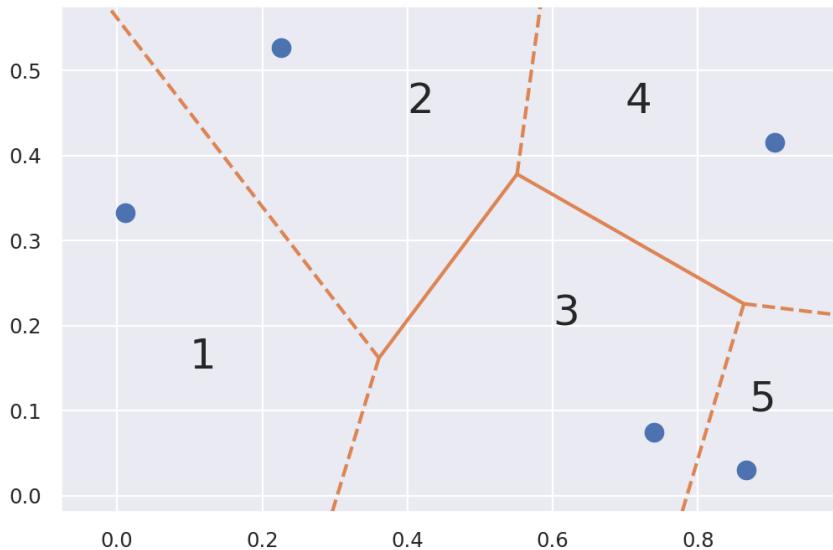
Το Σχήμα 6.1 δείχνει ένα σύνολο από διδιάστατα δεδομένα εκπαίδευσης για ένα πρόβλημα ταξινόμησης με δύο τάξεις (+ και -) καθώς και μια νέα περίπτωση \mathbf{x}' , η οποία πρόκειται να ταξινομηθεί σύμφωνα με τον αλγόριθμο k NN. Παρατηρούμε ότι για $k = 1$ (αριστερά) η πρόβλεψη του αλγορίθμου είναι η κατηγορία +, ενώ για $k = 5$ (δεξιά) έχουμε την κατηγορία -.



Σχήμα 6.1: Παράδειγμα ταξινόμησης με τη χρήση του αλγορίθμου k NN.

Ο αλγόριθμος k NN δεν παράγει στην πραγματικότητα ποτέ ένα μοντέλο της συνάρτησης στόχου από το σύνολο εκπαίδευσης. Αντίθετα, υπολογίζει την τιμή της εξαρτημένης μεταβλητής τη στιγμή εκείνη που ζητείται πρόβλεψη για μια νέα περίπτωση. Ωστόσο, έμμεσα υπάρχει ένα μοντέλο που αποτελείται από τις προβλέψεις του αλγορίθμου για κάθε δυνατή είσοδο.

Για διδιάστατα δεδομένα ενός προβλήματος ταξινόμησης μπορούμε να απεικονίσουμε το μοντέλο του 1NN χρησιμοποιώντας ένα διάγραμμα Voronoi. Στο σχήμα 6.2, βλέπουμε το διάγραμμα Voronoi, για ένα σύνολο πέντε διδιάστατων παραδειγμάτων. Το κυρτό πολύγωνο που περιβάλλει κάθε παράδειγμα οριοθετεί την περιοχή των περιπτώσεων που βρίσκονται πιο κοντά του, και που άρα θα ταξινομηθούν σύμφωνα με την κατηγορία στην οποία ανήκει αυτό το παράδειγμα. Η διακεκομένες γραμμές τείνουν στο άπειρο.



Σχήμα 6.2: Διάγραμμα Voronoi για ένα σύνολο πέντε διδιάστατων παραδειγμάτων.

6.1.1 Σταθμισμένη απόσταση

Μια διαδεδομένη επέκταση του k NN είναι η στάθμιση της συνεισφοράς του κάθε γείτονα, έτσι ώστε όσο πιο κοντά βρίσκεται ένας γείτονας, τόσο μεγαλύτερο να είναι το βάρος του. Για μια νέα περίπτωση \bar{x}' , μπορούμε να ορίσουμε το βάρος ενός γείτονα $\bar{x}^{(i)}$ ως:

$$w^{(i)} = \frac{1}{d(\bar{x}', \bar{x}^{(i)})} \quad (6.6)$$

Η σταθμισμένη απόφαση του k NN λαμβάνεται ως εξής:

- Σε εργασίες παλινδρόμησης:

$$\hat{y}' = \frac{\sum_{i=1}^k w^{(i)} \bar{y}^{(i)}}{\sum_{i=1}^k w^{(i)}} \quad (6.7)$$

- Σε εργασίες ταξινόμησης:

$$\hat{y}' = \arg \max_{j \in \{0, 1, \dots, c-1\}} \sum_{i=1}^k w^{(i)} \mathbb{1}[y^{(i)} = j] \quad (6.8)$$

6.1.2 Πολυπλοκότητα

Στον k NN, όλη η επεξεργασία γίνεται κατά την πρόβλεψη και η ταξινόμηση μιας περίπτωσης απαιτεί πολύ χρόνο. Για να αντιμετωπιστεί αυτό το γεγονός, γίνεται χρήση δομών δεικτοδότησης (indexes), οι οποίες προσθέτουν κάποιο (ανεκτό) κόστος σε μνήμη, αλλά επιταχύνουν σημαντικά τον εντοπισμό των πλησιέστερων γειτόνων.

Για παράδειγμα, θα μπορούσε να γίνει χρήση μιας δεντρικής δομής δεικτοδότησης, όπου τα παραδείγματα εκπαίδευσης αποθηκεύονται στα φύλλα, έτσι ώστε κοντινά παραδείγματα να βρίσκονται σε κοντινά φύλλα. Οι εσωτερικοί κόμβοι του δένδρου κατευθύνουν μια νέα περίπτωση σε κοντινό παράδειγμα ελέγχοντας τις τιμές κάποιων χαρακτηριστικών.

Μια δομή που χρησιμοποιείται στην πράξη είναι το kd -tree. Ωστόσο αν τα χαρακτηριστικά είναι πολλά, τότε η μέθοδος αυτή δεν προσφέρει κάποιο ιδιαίτερο πλεονέκτημα.

6.1.3 Θέματα υπολογισμού απόστασης

Όταν δεν έχουν όλες οι συνεχείς μεταβλητές το ίδιο εύρος τιμών, τότε οι διαφορές των τιμών τους δεν θα είναι ισότιμες μεταξύ των μεταβλητών. Για παράδειγμα η διαφορά της μεταβλητής μισθός, με τιμές στην τάξη των χιλιάδων, θα κυριαρχήσει στον υπολογισμό της συνολικής απόστασης σε σχέση με τη μεταβλητή ηλικία, με τιμές στην τάξη των δεκάδων. Επομένως απαιτείται να φέρουμε όλες τις μεταβλητές στο ίδιο εύρος τιμών, χρησιμοποιώντας μετασχηματισμούς όπως η κλιμάκωση και η προτυποποίηση που είδαμε στην Ενότητα ??.

Οι συνήθεις μετρικές απόστασης, όπως η Ευκλείδεια, υποθέτουν ότι οι μεταβλητές είναι συνεχείς. Πως όμως μπορεί να υπολογιστεί η απόσταση μεταξύ διακριτών ή βαθμωτών μεταβλητών; Έστω για παράδειγμα η διακριτή μεταβλητή επάγγελμα. Θα μπορούσαμε να πούμε ότι η απόσταση μεταξύ δύο επαγγελμάτων είναι 0 όταν αυτά ισούνται και 1 όταν διαφέρουν. Εναλλακτικά θα μπορούσαμε να εξετάσουμε την ομοιότητα των δύο επαγγελμάτων. Για παράδειγμα η διαφορά του επαγγέλματος καθηγητής λυκείου και καθηγητής πανεπιστήμιου θα μπορούσε να είναι μικρότερη του 1. Αν διαθέταμε μια ιεραρχική δομή επαγγελμάτων, τότε θα μπορούσαμε να λάβουμε υπόψη την απόσταση των δύο επαγγελμάτων σε αυτήν την ιεραρχία. Για παράδειγμα το επάγγελμα σεφ θα βρίσκεται πιο κοντά στο επάγγελμα ρεσεψιονίστ, απ' ότι στο επάγγελμα μηχανικός λογισμικού.

Στις βαθμωτές μεταβλητές, η απόσταση είναι πιο εύκολο να καθοριστεί. Έστω για παράδειγμα η βαθμωτή μεταβλητή ηλικιακή ομάδα με τιμές νέος, μεσήλικας και υπερήλικας. Προφανώς η απόσταση ανάμεσα στις τιμές νέος και μεσήλικας θα πρέπει να είναι μικρότερη σε σχέση με τις τιμές νέος και υπερήλικας. Γενικότερα για βαθμωτή μεταβλητή με διατεταγμένες τιμές v_1, \dots, v_l , μπορούμε να πούμε ότι η απόσταση μεταξύ δύο τιμών, v_a και v_b , όπου $a < b$ ισούται με $(b - a)/l - 1$.

6.1.4 Η κατάρα των διαστάσεων

Ο αλγόριθμος k NN υπολογίζει την απόσταση μεταξύ των περιπτώσεων με βάση όλα τα χαρακτηριστικά. Αυτό μπορεί να μειώσει σημαντικά την ακρίβεια του αλγορίθμου, όταν υπάρχουν πολλά χαρακτηριστικά που δεν επηρεάζουν την εξαρτημένη μεταβλητή. Έστω για παράδειγμα ένα πρόβλημα με 20 χαρακτηριστικά, εκ των οποίων μόνο τα 2 είναι σημαντικά για την πρόβλεψη νέων περιπτώσεων. Τότε μια περίπτωση και ένα παράδειγμα που έχουν ίδιες τιμές στα 2 αυτά χαρακτηριστικά μπορεί να έχουν πολύ μεγάλη Ευκλείδεια απόσταση. Επομένως, η πρόβλεψη με τον αλγόριθμο k NN δεν θα είναι αποτελεσματική.

Η εκθετική αύξηση της δυσκολίας μάθησης σε σχέση με την αύξηση του αριθμού των χαρακτηριστικών ονομάζεται **κατάρα των διαστάσεων** (curse of dimensionality). Για να αντιμετωπιστεί αυτό το πρόβλημα έχουν προταθεί μέθοδοι τόσο για τη στάθμιση των χαρακτηριστικών, όσο και για την επιλογή ενός υποσυνόλου χαρακτηριστικών. Για παράδειγμα θα μπορούσαν να χρησιμοποιηθούν βάρη στα χαρακτηριστικά ανάλογα με τη σημαντικότητά τους. Όσο μεγαλύτερο το βάρος ενός χαρακτηριστικού, τόσο μεγαλύτερη η διαφορά ανάμεσα σε δύο τιμές του, και επομένως τόσο μεγαλύτερη συνολική απόσταση ανάμεσα σε μια περίπτωση και σε ένα παράδειγμα. Η πλήρης απάλειψη των λιγότερο σημαντικών χαρακτηριστικών αποτελεί πιο

δραστική λύση. Υπάρχει ολόκληρη περιοχή της μηχανικής μάθησης που ονομάζεται **επιλογή χαρακτηριστικών** (feature selection) και μελετά αυτό ακριβώς το αντικείμενο.

6.1.5 Παρατηρήσεις

Τα πλεονεκτήματα του k NN είναι τα εξής: Αφενός, η εκπαίδευση του είναι ταχύτατη, αφού απλά αποθηκεύει τα παραδείγματα εκπαίδευσης. Επιπλέον μπορεί να μάθει πολύπλοκες συναρτήσεις επειδή επικεντρώνεται σε τοπικές (επομένως πιο απλές) περιοχές του χώρου των περιπτώσεων. Τέλος δεν έχει απώλεια πληροφορίας επειδή αξιοποιεί τα διαθέσιμα δεδομένα ως έχουν. Τα μειονεκτήματα του είναι ότι καθυστερεί κατά την πρόβλεψη και ότι λαμβάνει υπόψη ασήμαντα χαρακτηριστικά. Η επαγωγική μεροληψία του έγκειται στην παραδοχή ότι η τιμή της εξαρτημένης μεταβλητής είναι παραπλήσια για περιπτώσεις που είναι κοντινές με βάσει την απόστασή τους.

6.2 Περαιτέρω Μελέτη

Μια από τις πρώτες ολοκληρωμένες μελέτες του αλγορίθμου των k πλησιέστερων γειτόνων αποτελεί η [1].

Βιβλιογραφία

- [1] T. M. Cover και P. E. Hart. “Nearest Neighbor Pattern Classification”. Στο: *IEEE Transactions on Information Theory* 13.1 (1967), σσ. 21–27. doi: 10.1109/TIT.1967.1053964.

ΚΕΦΑΛΑΙΟ 7

ΣΥΝΟΛΑ ΜΟΝΤΕΛΩΝ

Στο κεφάλαιο αυτό:

Θα μελετήσουμε την περιοχή της μηχανικής μάθησης που ασχολείται με σύνολα από μοντέλα πρόβλεψης. Αρχικά θα δούμε τους τρόπους με τους οποίους μπορούμε να συγκεράσουμε τις προβλέψεις ενός συνόλου από μοντέλα. Στη συνέχεια θα δούμε πως μπορούμε να παράξουμε ένα σύνολο από μοντέλα μηχανικής μάθησης της ίδιας οικογένειας χρησιμοποιώντας τεχνικές δειγματοληψίας και ενίσχυσης. Θα ολοκληρώσουμε το κεφάλαιο παρουσιάζοντας τον αλγόριθμον του τυχαίου δάσους, έναν δημοφιλή αλγόριθμο για την μάθηση ενός συνόλου από δενδρικά μοντέλα.

Η περιοχή της μηχανικής μάθησης που ασχολείται με τη δημιουργία και τον συγκερασμό ενός συνόλου από μοντέλα μηχανικής μάθησης καλείται **μέθοδοι συνόλου** (ensemble methods) ή **μάθηση συνόλου** (ensemble learning). Πρόκειται για ένα πολύ σημαντικό πρακτικό αντικείμενο της μηχανικής μάθησης, καθώς ένα σύνολο από μοντέλα συνήθως καταφέρνει να πετύχει μεγαλύτερη ορθότητα σε σχέση με ένα μεμονωμένο μοντέλο. Ένα από τα πιο διάσημα σύνολα μοντέλων είναι εκείνο που κέρδισε το 2009 τον διαγωνισμό του ενός εκατομμυρίου δολαρίων της Netflix. Ο διαγωνισμός αυτός αναζητούσε την προσέγγιση που θα βελτίωνε κατά 10% τον αλγόριθμο συστάσεων που χρησιμοποιούσε η Netflix για να προτείνει ταινίες στους πελάτες της. Η λύση της νικήτριας ομάδας περιελάμβανε 107 επί μέρους μοντέλα. Η ιδέα του συγκερασμού ενός συνόλου μοντέλων μηχανικής μάθησης, βρίσκει την αναλογία της και στον κόσμο των ανθρώπων, όπου η **σοφία του πλήθους** ή **συλλογική νοημοσύνη**, δηλαδή ο συγκερασμός των απαντήσεων ενός μεγάλου αριθμού ανθρώπων σε ένα πρόβλημα, μπορεί να οδηγήσει σε παρόμοια ή καλύτερα αποτελέσματα σε σχέση με την απάντηση ενός μόνο ειδικού.

Για να καταφέρει ένα σύνολο μοντέλων να έχει μεγαλύτερη ορθότητα από τα επί μέρους μοντέλα από τα οποία αποτελείται, θα πρέπει να περιλαμβάνει μοντέλα ικανοποιητικής ορθότητας (π.χ. μεγαλύτερης του 50% σε ένα πρόβλημα δυαδικής ταξινόμησης), τα οποία να διαφέρουν μεταξύ τους ως προς τις περιπτώσεις στις οποίες σφάλλουν. Αυτή η διαφορετικότητα των μοντέλων ενός συνόλου, καλείται **ποικιλία** (diversity). Θα τονίσουμε την σημασία της ποικιλίας στην ορθότητα ενός συνόλου με το ακόλουθο παράδειγμα. Έστω δύο σύνολα μοντέλων, το καθένα από τα οποία περιλαμβάνει τρία μοντέλα δυαδικής ταξινόμησης. Τα σύνολα αυτά

y	M1	M2	M3	Σύνολο
0	0	0	0	0
1	1	1	1	1
0	1	1	1	1
1	1	1	1	1
Ορθότητα	0.75	0.75	0.75	0.75

(α) Χωρίς ποικιλία.

y	M1	M2	M3	Σύνολο
0	0	0	0	0
1	1	1	0	1
0	1	0	0	0
1	1	0	1	1
Ορθότητα	0.75	0.75	0.75	1

(β) Με ποικιλία.

Πίνακας 7.1: Δύο σύνολα μοντέλων, καθένα από τα οποία αποτελείται από τρία μοντέλα δυαδικής ταξινόμησης, όπου το κάθε μοντέλο έχει ορθότητα 75%.

ταξινομούν μια περίπτωση στην κλάση που δίνουν στην εξόδο τους τουλάχιστον 2 από τα 3 επί μέρους μοντέλα τους. Η πρώτη στήλη των Πινάκων 7.1α και 7.1β περιέχει τις πραγματικές τιμές της δίτιμης μεταβλητής εξόδου για τέσσερα παραδείγματα ενός συνόλου ελέγχου, οι επόμενες τρεις στήλες περιέχουν τις αποφάσεις των τριών μοντέλων στα παραδείγματα αυτά, και η τελευταία στήλη τις αποφάσεις του συνόλου των μοντέλων. Η τελευταία γραμμή των πινάκων δείχνει την ορθότητα των τριών μοντέλων καθώς και του συνόλου των μοντέλων. Παρατηρούμε πως και τα δύο σύνολα μοντέλων, τόσο αυτό του Πίνακα 7.1α όσο και αυτό του Πίνακα 7.1β, αποτελούνται από μοντέλα με ικανοποιητική ορθότητα 75%. Ωστόσο το σύνολο μοντέλων του Πίνακα 7.1α δεν έχει καθόλου ποικιλία, καθώς και τα τρία μοντέλα σφάλλουν στο ίδιο, τρίτο, παράδειγμα του συνόλου ελέγχου. Αντίθετα το σύνολο μοντέλων του Πίνακα 7.1β χαρακτηρίζεται από υψηλή ποικιλία, καθώς τα τρία μοντέλα σφάλλουν σε διαφορετικά παραδείγματα του συνόλου ελέγχου. Το μοντέλο M1 σφάλει στο τρίτο παράδειγμα, το M2 στο τέταρτο και το M3 στο δεύτερο. Έτσι το σύνολο αυτό καταφέρνει να πετύχει βελτιωμένη ορθότητα σε σχέση με τα επί μέρους μοντέλα του, σε αντίθεση με το σύνολο του Πίνακα 7.1α.

Οι τεχνικές μηχανικής μάθησης για σύνολα μοντέλων ασχολούνται με δύο βασικές δραστηριότητες: α) το πως θα παραχθούν τα μοντέλα του συνόλου, και β) το πως θα συγκεραστούν οι προβλέψεις των μοντέλων του συνόλου. Ένας πολύ απλός τρόπος παραγωγής πολλών και διαφορετικών μεταξύ τους μοντέλων είναι η χρήση διαφορετικών αλγορίθμων μάθησης. Σε αυτήν την περίπτωση τα σύνολα καλούνται **ετερογενή**. Αντίθετα, όταν τα μοντέλα παράγονται χρησιμοποιώντας τον ίδιο αλγόριθμο, τότε τα σύνολα καλούνται **ομοιογενή**. Για την παραγωγή ομοιογενών συνόλων μοντέλων, θα μελετήσουμε στις επόμενες ενότητες τις τεχνικές δειγματοληψίας και ενίσχυσης, καθώς και τον αλγόριθμο του τυχαίου δάσους. Πριν από αυτό όμως θα δούμε τους τρόπους με τους οποίους μπορούμε να συγκεράσουμε τις προβλέψεις ενός συνόλου από μοντέλα.

7.1 Συγκερασμός Μοντέλων

Ο συγκερασμός ενός συνόλου από μοντέλα, μπορεί να επιτευχθεί είτε με απλές συναρτήσεις συγκερασμού των προβλέψεων τους, είτε με πολύπλοκες συναρτήσεις, οι οποίες μπορεί να αποτελούν οι ίδιες ένα μοντέλο μηχανικής μάθησης, όπως στην τεχνική της σωρευμένης γενίκευσης που θα μελετήσουμε στο τέλος αυτής της ενότητας.

Σε εργασίες παλινδρόμησης, μια απλή συνάρτηση συγκερασμού των διαφορετικών προβλέψεων ενός συνόλου μοντέλων παλινδρόμησης είναι ο μέσος όρος τους. Σε εργασίες ταξινόμησης, μπορούμε αντίστοιχα να δώσουμε ως συγκερασμό την κλάση που δίνουν στην εξόδο τους η πλειοψηφία των μοντέλων του συνόλου. Αυτός ο τρόπος συγκερασμού ονομάζεται **σκληρή ψηφοφορία** (hard voting). Εναλλακτικά, εφόσον τα μοντέλα δίνουν στην εξόδο τους μια κατανομή πιθανοτήτων στις κλάσεις, θα μπορούσαμε να πάρουμε τους μέσους όρους της πιθανότητας κάθε κλάσης, οι οποίοι θα αποτελούσαν επίσης μια κατανομή πιθανοτήτων. Αυτός ο τρόπος συγκερασμού, σε αντιστοιχία με τον προηγούμενο, ονομάζεται **μαλακή ψηφοφορία** (soft voting).

Αν κάθε ένα από τα μοντέλα συνοδεύεται από κάποιον θετικό πραγματικό αριθμό αντίστοιχο της σημαντι-

κότητας τους, όπως για παράδειγμα η ορθότητα που πετυχαίνουν σε κάποιο σύνολο επικύρωσης, τότε μπορούμε να σταθμίσουμε τον συγκερασμό με αυτούς τους αριθμούς που καλούμε βάρη. Αυτός ο τρόπος συγκερασμού ονομάζεται σταθμισμένη ψηφοφορία στην περίπτωση της ταξινόμησης και σταθμισμένος μέσος όρος στην περίπτωση της παλινδρόμησης.

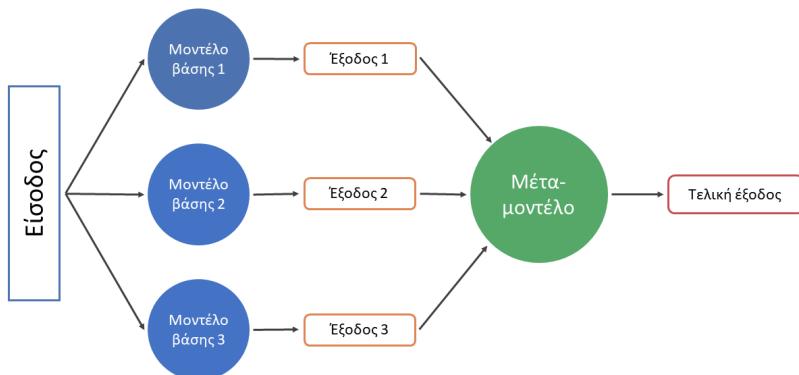
Ο Πίνακας 7.2 παρουσιάζει παραδείγματα απλών συναρτήσεων συγκερασμού των προβλέψεων ενός συνόλου τριών μοντέλων παλινδρόμησης και ταξινόμησης (εργασία με τρεις κλάσεις). Ο σταθμισμένος συγκερασμός στην περίπτωση της παλινδρόμησης θα υπολογιστεί ως εξής: $\frac{108 \cdot 0.4 + 123 \cdot 0.6 + 102 \cdot 0.3}{0.4 + 0.6 + 0.3} = 113.54$.

	Ταξινόμηση			
	Παλινδρόμηση	Κλάσεις	Πιθανότητες	Βάρος
Μοντέλο 1	108.0	0	0.60, 0.30, 0.10	0.4
Μοντέλο 2	123.0	1	0.20, 0.80, 0.00	0.6
Μοντέλο 3	102.0	0	0.70, 0.10, 0.20	0.3
Συγκερασμός	111.0	0	0.50, 0.40, 0.10	
Σταθμισμένος Συγκερασμός	113.54	0	0.44, 0.48, 0.08	

Πίνακας 7.2: Παραδείγματα απλών συναρτήσεων συγκερασμού των προβλέψεων ενός συνόλου τριών μοντέλων παλινδρόμησης και ταξινόμησης (εργασία με τρεις κλάσεις).

7.1.1 Σωρευμένη γενίκευση

Η **σωρευμένη γενίκευση** (stacked generalization) ή απλά **σώρευση** (stacking) συνδυάζει ένα σύνολο από μοντέλα πρόβλεψης, που ονομάζονται μοντέλα βάσης, χρησιμοποιώντας ένα επιπλέον μοντέλο πρόβλεψης, το οποίο ονομάζεται μέτα-μοντέλο. Το μέτα-μοντέλο έχει ως είσοδο την έξοδο των μοντέλων βάσης. Σε εργασίες ταξινόμησης, η είσοδος του μέτα-μοντέλου μπορεί να είναι είτε η πιθανότερη κλάση, είτε η κατανομή πιθανοτήτων όλων των κλάσεων. Το Σχήμα 7.1 παρουσιάζει την σώρευση με γραφικό τρόπο.



Σχήμα 7.1: Σωρευμένη γενίκευση.

Η εκπαίδευση του μέτα-μοντέλου μπορεί να γίνει με δεδομένα εκπαίδευσης που προκύπτουν από την εφαρμογή των μοντέλων βάσης, στα δεδομένα με τα οποία εκπαίδευτηκαν (in-sample). Για ένα αρχικό σύνολο εκπαίδευσης $D = \{(x^{(i)}, y^{(i)}) \mid i = 1, \dots, m\}$, και N μοντέλα βάσης h_1, \dots, h_N , το σύνολο εκπαίδευσης για το μέτα-μοντέλο είναι $D_{meta} = \{(\mathbf{h}^{(i)}, y^{(i)}) \mid i = 1 \dots m\}$, όπου $\mathbf{h}^{(i)} = [h_1(x^{(i)}), \dots, h_N(x^{(i)})]$.

Ένα μειονέκτημα αυτού του τρόπου εκπαίδευσης του μέτα-μοντέλου είναι πως κατά την εφαρμογή του σε νέες περιπτώσεις, οι τιμές εισόδου που θα του δίνονται, δηλαδή οι προβλέψεις των μοντέλων βάσης, αναμένεται να ακολουθούν διαφορετική κατανομή από αυτές που είχε εκπαιδευτεί. Συγκεκριμένα, θα χαρακτηρίζονται

από μεγαλύτερη αβεβαιότητα. Αυτό αντιβαίνει σε μια βασική υπόθεση της μάθησης με επίβλεψη, σύμφωνα με την οποία τα δεδομένα στα οποία εφαρμόζεται ένα μοντέλο, προέρχονται από την ίδια κατανομή που προέρχονται και τα δεδομένα από τα οποία εκπαιδεύτηκε. Για την αποφυγή αυτού του προβλήματος, συστήνεται τα παραδείγματα εκπαίδευσης του μέτα-μοντέλου να προκύπτουν από την εφαρμογή των μοντέλων βάσης σε διαφορετικά παραδείγματα από αυτά που έχουν εκπαιδευτεί (out-of-sample).

Ένας τρόπος για να το πετύχουμε αυτό είναι με τον διαχωρισμό του συνόλου εκπαίδευσης σε δύο υποσύνολα. Το πρώτο υποσύνολο που προκύπτει χρησιμοποιείται για την εκπαίδευση των μοντέλων βάσης ενώ το δεύτερο υποσύνολο δίνεται ως είσοδος στα ήδη εκπαίδευμένα μοντέλα βάσης προκειμένου να προκύψει το σύνολο εκπαίδευσης του μέτα-μοντέλου. Μια επέκταση της τεχνικής αυτής είναι η χρήση k -πλης σταυρωτής επικύρωσης όπου τα $k-1$ μέρη είναι το σύνολο εκπαίδευσης των μοντέλων βάσης και το εναπομένων μέρος χρησιμοποιείται για την παραγωγή του συνόλου εκπαίδευσης του μέτα-μοντέλου. Η διαδικασία επαναλαμβάνεται k φορές και έτσι παράγεται ένα σύνολο εκπαίδευσης για το μέτα-μοντέλο με ίδιο πλήθος παραδειγμάτων όσο και το αρχικό σύνολο εκπαίδευσης.

Η σώρευση χρησιμοποιείται συνήθως για τον συγκερασμό ετερογενών συνόλων μοντέλων. Επιπλέον, η σώρευση μπορεί να χρησιμοποιηθεί σε εργασίες με πολυτροπικά (multimodal) δεδομένα (εικόνα, ήχος, κείμενο κλπ.) όπου κάθε μοντέλο βάσης εκπαιδεύεται με ένα διαφορετικό τύπο δεδομένων. Για παράδειγμα, σε ένα πρόβλημα ταξινόμησης των εικόνων που εμφανίζονται μέσα σε ένα βιβλίο σε κατηγορίες όπως γραφική παράσταση, σχεδιάγραμμα, φωτογραφία, κ.α., ένα μοντέλο βάσης μπορεί να έχει ως είσοδο την εικόνα και ένα άλλο μοντέλο βάσης τη λεξάντα που τη συνοδεύει.

7.2 Τεχνικές Δειγματοληψίας

7.2.1 Αυτοδύναμη συνάθροιση

Στην **αυτοδύναμη συνάθροιση** (bootstrap aggregating, εν συντομίᾳ bagging), η παραγωγή των μοντέλων του συνόλου επιτυγχάνεται μέσω της εφαρμογής του ίδιου αλγορίθμου μηχανικής μάθησης σε διαφορετικές παραλλαγές του συνόλου εκπαίδευσης, οι οποίες προκύπτουν χρησιμοποιώντας δειγματοληψία με επανατοποθέτηση. Ο συγκερασμός των μοντέλων του συνόλου επιτυγχάνεται μέσω σκληρής ψηφοφορίας στην περίπτωση της ταξινόμησης και μέσω του υπολογισμού του μέσου όρου των προβλέψεων των μοντέλων στην περίπτωση της παλινδρόμησης.

Για ένα σύνολο εκπαίδευσης m παραδειγμάτων, η πιθανότητα να επιλεγεί ένα από τα m παραδείγματα σε κάθε μία από τις δειγματοληψίες με επανατοποθέτηση είναι $\frac{1}{m}$, ενώ το να μην επιλεγεί είναι $1 - \frac{1}{m}$. Επομένως, η πιθανότητα να μην επιλεγεί καθόλου ένα παράδειγμα σε μια παραλλαγή του συνόλου εκπαίδευσης είναι $(1 - \frac{1}{m})^m$ και η πιθανότητα να επιλεγεί είναι $1 - (1 - \frac{1}{m})^m$. Όταν το m τείνει στο άπειρο, η πιθανότητα επιλογής ενός παραδείγματος σε μια παραλλαγή του συνόλου εκπαίδευσης τείνει στο $1 - \frac{1}{e} \approx 0.632$. Δηλαδή περίπου 0.632 m από τα παραδείγματα του αρχικού συνόλου εκπαίδευσης συμμετέχουν σε κάθε παραλλαγή του.

Ο Πίνακας 7.3 παρουσιάζει ένα σύνολο με 10 παραδείγματα εκπαίδευσης και 3 παραλλαγές του που έχουν προκύψει με τη διαδικασία της αυτοδύναμης συνάθροισης. Η πρώτη στήλη παρουσιάζει τον κωδικό του παραδείγματος (1 έως 10) και οι επόμενες το πλήθος κάθε παραδείγματος στο αντίστοιχο σύνολο εκπαίδευσης. Στο αρχικό σύνολο κάθε παράδειγμα υπάρχει μία μόνο φορά. Στην παραλλαγή 1, τα παραδείγματα 2, 3 και 6 υπάρχουν 2 φορές, ενώ τα παραδείγματα 4, 8 και 10 καθόλου.

Η αυτοδύναμη συνάθροιση δουλεύει καλά για ασταθείς αλγορίθμους μάθησης, δηλαδή αλγορίθμους που παράγουν αρκετά διαφορετικά μεταξύ τους μοντέλα όταν τους δοθούν παραπλήσια σύνολα δεδομένων. Τέτοιοι είναι οι αλγόριθμοι μάθησης δενδρικών μοντέλων, ειδικά όταν αυτά έχουν αναπτυχθεί σε πλήρες βάθος, σε αντίθεση με τους αλγορίθμους μάθησης γραμμικών μοντέλων. Με άλλα λόγια, δουλεύει καλά για μοντέλα που υπερπροσαρμόζονται στα δεδομένα εκπαίδευσης, δηλαδή για μοντέλα υψηλής πολυπλοκότητας που παρουσιάζουν υψηλή διακύμανση και χαμηλή προκατάληψη. Συγκεκριμένα, η αυτοδύναμη συνάθροιση μειώνει την διακύμανση, ενώ έχει ελάχιστη επιρροή στην προκατάληψη. Στην ιδανική περίπτωση η διακύμανση μειώ-

Παράδειγμα	Αρχικό Σύνολο	Παραλλαγή 1	Παραλλαγή 2	Παραλλαγή 3
1	1	1	0	2
2	1	2	1	0
3	1	2	0	1
4	1	0	2	1
5	1	1	0	2
6	1	2	1	0
7	1	1	0	2
8	1	0	2	1
9	1	1	2	0
10	1	0	2	1

Πίνακας 7.3: Παράδειγμα συνόλων εκπαίδευσης που παράγονται με την εφαρμογή της αυτοδύναμης συνάθροισης.

νεται γραμμικά σε σχέση με το πλήθος N των μοντέλων:

$$\text{Variance}(h_{bagging}(\mathbf{x})) = \frac{\text{Variance}(h(\mathbf{x}))}{N}$$

Τα παραδείγματα του αρχικού συνόλου εκπαίδευσης που δεν περιλαμβάνονται σε μια παραλλαγή του που δημιουργείται με την μέθοδο της αυτοδύναμης συνάθροισης καλούνται **δείγματα εκτός σάκου** (out of bag samples). Τα δείγματα αυτά μπορούν να χρησιμεύσουν για διάφορους υπολογισμούς, όπως π.χ. για ρύθμιση των υπερπαραμέτρων του μοντέλου που θα εκπαιδευτεί από το αντίστοιχο σύνολο εκπαίδευσης ή για την εκτίμηση του σφάλματος του, στο πλαίσιο μιας διαδικασίας σταθμισμένου συγκερασμού.

Στην αυτοδύναμη συνάθροιση, τα μοντέλα του συνόλου μπορούν τόσο να εκπαιδευτούν όσο και να χρησιμοποιηθούν για προβλέψεις ανεξάρτητα το ένα από το άλλο. Επομένως, θα μπορούσε να γίνει χρήση πολλαπλών υπολογιστικών πόρων για την επιτάχυνση των διαδικασιών εκπαίδευσης και προβλεψης.

7.2.2 Μέθοδος του τυχαίου υποχώρου

Ένας άλλος τρόπος για την δημιουργία παραλλαγών του συνόλου εκπαίδευσης αφορά στην τυχαία επιλογή ενός υποσυνόλου των μεταβλητών εισόδου. Η μέθοδος αυτή ονομάζεται **μέθοδος του τυχαίου υποχώρου** (random subspace method).

Έστω για παράδειγμα ένα σύνολο εκπαίδευσης με 5 μεταβλητές εισόδου. Ορίζουμε ένα συγκεκριμένο αριθμό μεταβλητών, π.χ. 3, τις οποίες θα επιλέξουμε τυχαία από τις 5 για κάθε μοντέλο. Ο πίνακας 7.4 δείχνει τις μεταβλητές που συμμετέχουν (1) ή όχι (0) σε 7 παραλλαγές του αρχικού συνόλου με βάση την μέθοδο των τυχαίων υποχώρων.

	Μεταβλητή 1	Μεταβλητή 2	Μεταβλητή 3	Μεταβλητή 4	Μεταβλητή 5
Αρχικό Σύνολο	1	1	1	1	1
Παραλλαγή 1	1	1	1	0	0
Παραλλαγή 2	0	1	1	1	0
Παραλλαγή 3	0	0	1	1	1
Παραλλαγή 4	1	0	0	1	1
Παραλλαγή 5	1	1	0	0	1
Παραλλαγή 6	0	1	0	1	1
Παραλλαγή 7	1	0	1	0	1

Πίνακας 7.4: Παραλλαγές ενός συνόλου εκπαίδευσης με τη μέθοδο των τυχαίων υποχώρων.

7.3 Τεχνικές Ενίσχυσης

Η βασική ιδέα στις τεχνικές **ενίσχυσης** (boosting) είναι πως τα μοντέλα εκπαιδεύονται διαδοχικά και κάθε μοντέλο προσπαθεί να διορθώσει τα λάθη των προηγούμενων μοντέλων. Οι τεχνικές ενίσχυσης είναι ιδιαίτερα δημοφιλείς λόγω της υψηλής ακριβείας που πετυχαίνουν σε πολλά προβλήματα μάθησης. Ωστόσο, λόγω της διαδοχικότητας στη διαδικασία μάθησης, αυτή δεν μπορεί να γίνει παράλληλα, με αποτέλεσμα να απαιτείται μεγαλύτερος χρόνος για την ολοκλήρωση της μάθησης στις τεχνικές αυτές.

Δύο από τις πιο διαδεδομένες τεχνικές ενίσχυσης είναι η **προσαρμοστική ενίσχυση** (adaptive boosting, εν συντομίᾳ AdaBoost) και η **επικλινής ενίσχυση** (gradient boosting), οι οποίες αναλύονται στις παρακάτω ενότητες.

7.3.1 Προσαρμοστική ενίσχυση

Στην προσαρμοστική ενίσχυση, ο τρόπος με τον οποίο κάνουμε ένα μοντέλο να προσπαθήσει να διορθώσει τα λάθη των προηγούμενων μοντέλων είναι δίνοντας μεγαλύτερη προσοχή στα παραδείγματα εκπαίδευσης όπου τα προηγούμενα μοντέλα εμφανίζουν τη μεγαλύτερη απόκλιση από τη μεταβλητή στόχο.

Η αρχική έκδοση της προσαρμοστικής ενίσχυσης αφορούσε εργασίες δυαδικής ταξινόμησης. Έπειτα παρουσιάστηκαν επεκτάσεις για εργασίες παλινδρόμησης (AdaBoost.R) και ταξινόμησης σε πάνω από δύο κλάσεις (AdaBoost.M1, AdaBoost.M2). Εμείς θα μελετήσουμε στη συνέχεια την AdaBoost.M1 (Αλγόριθμος 3), η οποία είναι από τις πιο διαδεδομένες εκδόσεις της προσαρμοστικής ενίσχυσης.

Algorithm 3 Η μέθοδος AdaBoost.M1

```

1: procedure ADABOOST.M1_TRAIN( $D, N$ )  $\triangleright D = \{(\mathbf{x}^{(i)}, y^{(i)}) \mid i = 1, \dots, m\}$ 
2:   for  $i \leftarrow 1$  to  $m$  do
3:      $w^{(i)} = 1/m$   $\triangleright$  Αρχικοποίηση βαρών παραδειγμάτων
4:   end for
5:   for  $t \leftarrow 1$  to  $N$  do
6:      $h_t \leftarrow \text{TrainModel}(w, D)$   $\triangleright$  Εκπαίδευση  $h_t$  στο σταθμισμένο σύνολο εκπαίδευσης
7:      $e_t \leftarrow \sum_{i=1}^m w^{(i)}[y^{(i)} \neq h_t(\mathbf{x}^{(i)})]$   $\triangleright$  Υπολογισμός σταθμισμένου σφάλματος
8:     if ( $e_t \geq 0.5$ )  $\vee$  ( $e_t == 0$ ) then
9:       break
10:    end if
11:    for  $i \leftarrow 1$  to  $m$  do
12:      if ( $y^{(i)} == h_t(\mathbf{x}^{(i)})$ ) then
13:         $w^{(i)} \leftarrow \frac{e_t}{1-e_t} w^{(i)}$   $\triangleright$  Ενημέρωση των βαρών
14:      end if
15:    end for
16:    for  $i \leftarrow 1$  to  $m$  do
17:       $w^{(i)} \leftarrow \frac{w^{(i)}}{\sum_{i=1}^m w^{(i)}}$   $\triangleright$  Κανονικοποίηση βαρών
18:    end for
19:  end for
20: end procedure
21: procedure ADABOOST.M1_PREDICT( $H_t, \mathbf{x}'$ )  $\triangleright H_t = \{h_t \mid t = 1, \dots, N\}$ 
22:    $h(\mathbf{x}') = \underset{c \in C}{\operatorname{argmax}} \sum_{t=1}^N \log \frac{1-e_t}{e_t}$   $\triangleright$  Επιλογή κλάσης που συγκεντρώνει την πλειοψηφία των σταθμισμένων ψήφων
23: end procedure

```

Ο αλγόριθμος AdaBoost.M1 κατά την εκπαίδευση δέχεται ως είσοδο ένα σύνολο δεδομένων $D = \{(\mathbf{x}^{(i)}, y^{(i)}) \mid$

$i = 1, \dots, m\}$ και τον συνολικό αριθμό των μοντέλων μάθησης N . Η διαδικασία ξεκινάει με την αρχικοποίηση των βαρών κάθε παραδειγμάτος εκπαίδευσής σε $\frac{1}{m}$ (γραμμή 3). Έπειτα, εκπαιδεύεται ένα μοντέλο μάθησης h_t στο σταθμισμένο σύνολο εκπαίδευσής (γραμμή 6), υπολογίζεται το σταθμισμένο σφάλμα $e_t = \sum_{i=1}^m w^{(i)}[y^{(i)} \neq h_t(\mathbf{x}^{(i)})]$ (γραμμή 7) και γίνεται ενημέρωση των βαρών των παραδειγμάτων, για τα οποία το μοντέλο προέβλεψε σωστά τη μεταβλητή στόχο ($y^{(i)} == h_t(\mathbf{x}^{(i)})$), σύμφωνα με τον τύπο $w^{(i)} = \frac{e_t}{1-e_t} w^{(i)}$ (γραμμή 13). Τέλος, γίνεται κανονικοποίηση των βαρών (γραμμή 17). Η διαδικασία αυτή επαναλαμβάνεται για τον σύνολο των μοντέλων, N , ή μέχρι το σφάλμα e_t μηδενιστεί ή γίνει μεγαλύτερο ή ίσο του 0.5. Κατά την πρόβλεψη, η AdaBoost.M1 υπολογίζει τις προβλέψεις όλων των μοντέλων και τις σταθμίζει κατά $\log \frac{1-e_t}{e_t}$ σύμφωνα με το σφάλμα του κάθε μοντέλου e_t . Η μέθοδος επιλέγει την κλάση που συγκεντρώνει την πλειοψηφία των σταθμισμένων ψήφων (γραμμή 22).

Η μέθοδος της προσαρμοστικής ενίσχυσης προϋποθέτει πως ο αλγόριθμος μάθησης που χρησιμοποιείται μπορεί να χειριστεί βάρη. Στις περισσότερες περιπτώσεις οι αλγόριθμοι μάθησης υποστηρίζουν βάρη. Για παράδειγμα, στα γραμμικά μοντέλα η συνάρτηση κόστους μπορεί να πολλαπλασιάζει το κόστος κάθε παραδειγμάτος με το βάρος του, ενώ στα δενδρικά μοντέλα, η εντροπία μπορεί να υπολογίζεται με βάση τα βάρη των παραδειγμάτων κάθε κλάσης, αντί για το πλήθος τους. Σε αντίθετη περίπτωση μπορεί να γίνει δειγματοληψία με επαναποτθέτηση (bootstrap sampling) με πιθανότητα επιλογής ενός παραδειγμάτους ανάλογη του βάρους του.

7.3.2 Επικλινής ενίσχυση

Η επικλινής ενίσχυση λειτουργεί με την ίδια φιλοσοφία όπως και η προσαρμοστική ενίσχυση, εκπαιδεύοντας διαδοχικά μοντέλα μάθησης όπου κάθε μοντέλο διορθώνει τα λάθη των προηγούμενων μοντέλων. Ωστόσο, σε αντίθεση με την προσαρμοστική ενίσχυση, δεν ανανεώνει τα βάρη των παραδειγμάτων σε κάθε επανάληψη, αλλά εκπαιδεύει ένα νέο μοντέλο χρησιμοποιώντας ως μεταβλητή στόχο την διαφορά μεταξύ της τρέχουσας εκτίμησης του συνόλου και της πραγματικής τιμής της μεταβλητής στόχου.

Η επικλινής ενίσχυση γίνεται πιο εύκολα κατανοητή αν την μελετήσουμε αρχικά σε μια εργασία παλινδρόμησης. Έστω ένα σύνολο εκπαίδευσης $D = \{(\mathbf{x}^{(i)}, y^{(i)}) \mid i = 1, \dots, m\}$. Θα ξεκινήσουμε με ένα μοντέλο $h_0(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m y^{(i)}$, το οποίο προβλέπει απλά την μέση τιμής της μεταβλητής εξόδου. Θα συνεχίσουμε μαθαίνοντας μοντέλα h_1, h_2, \dots με τιμή για την εξαρτημένη μεταβλητή το υπόλοιπο (residual) της αφαίρεσης της πρόβλεψης του προηγούμενου συνόλου μοντέλων από την πραγματική τιμή της μεταβλητής στόχου. Η πρόβλεψη $H_N(\mathbf{x})$ ενός συνόλου N εκπαιδευμένων μοντέλων ορίζεται ως εξής:

$$H_N(\mathbf{x}) = \sum_{t=0}^N h_t(\mathbf{x})$$

ή με τη χρήση αναδρομικού ορισμού:

$$H_N(\mathbf{x}) = H_{N-1}(\mathbf{x}) + h_N(\mathbf{x})$$

Οι τιμές της εξαρτημένης μεταβλητής για το μοντέλο h_t και το παράδειγμα $\mathbf{x}^{(i)}$ είναι ίσες με το υπόλοιπο $y^{(i)} - H_{t-1}(\mathbf{x}^{(i)})$. Ο ρόλος του μοντέλου h_t είναι να αντισταθμίζει τις αδυναμίες του τρέχοντος συνόλου μοντέλων H_{t-1} .

Το πλήθος των μοντέλων που θα εκπαιδεύσουμε, N , αποτελεί την βασική υπερ-παράμετρο της επικλινούς ενίσχυσης. Όσο περισσότερα μοντέλα εκπαιδεύουμε, τόσο καλύτερα θα πλησιάζουμε την πραγματική τιμή της μεταβλητής εξόδου, όμως ελλοχεύει ο κίνδυνος της υπερ-προσαρμογής στα δεδομένα εκπαίδευσης. Για την αντιμετώπιση αυτού του κινδύνου, στην επικλινή ενίσχυση υπάρχει μια ακόμα υπερ-παράμετρος που ονομάζεται **συρρίκνωση** (shrinkage), $\eta \in (0, 1)$, και καθορίζει το πόσο λαμβάνεται υπόψη κάθε επόμενο μοντέλο στην συνολική πρόβλεψη:

$$H_t(\mathbf{x}) = H_{t-1}(\mathbf{x}) + \eta h_t(\mathbf{x}) \quad (7.1)$$

Η τεχνική της επικλινούς ενίσχυσης παίρνει το όνομα της από το γεγονός ότι συνδυάζει την επικλινή κάθοδο και την ενίσχυση. Για να το δούμε αυτό θα ξαναγράψουμε την εξίσωση 7.1, ως εξής:

$$H_t(\mathbf{x}) = H_{t-1}(\mathbf{x}) - \eta(-h_t(\mathbf{x})) \quad (7.2)$$

Δεδομένου ότι το μοντέλο h_t προσεγγίζει το υπόλοιπο $y^{(i)} - H_{t-1}(\mathbf{x}^{(i)})$, ξαναγράφουμε την εξίσωση ως:

$$H_t(\mathbf{x}) \approx H_{t-1}(\mathbf{x}) - \eta[-(y^{(i)} - H_{t-1}(\mathbf{x}^{(i)}))] \quad (7.3)$$

Ο όρος μέσα στην αγκύλη ισούται με την μερική παράγωγο της συνάρτησης απώλειας του μισού τετραγωνικού σφάλματος ως προς το προηγούμενο σύνολο μοντέλων:

$$\frac{\partial}{\partial H_{t-1}} \frac{1}{2} (y^{(i)} - H_{t-1}(\mathbf{x}^{(i)}))^2 = -(y^{(i)} - H_{t-1}(\mathbf{x}^{(i)}))$$

Επομένως, η επικλινής ενίσχυση προσεγγίζει την εκτέλεση επικλινούς καθόδου στον χώρο των μοντέλων ή προβλέψεων, καθώς ενημερώνει τις προβλέψεις του τρέχοντος συνόλου μοντέλων με βάση το τετραγωνικό σφάλμα του.

Το γεγονός αυτό καθιστά την επικλινή ενίσχυση μια πολύ ισχυρή τεχνική, καθώς μπορούμε να αντικαταστήσουμε το τετραγωνικό σφάλμα με όποια συνάρτηση απώλειας ταιριάζει καλύτερα στην εργασία που μας απασχολεί, συμπεριλαμβάνοντας φυσικά και κλασικές εργασίες ταξινόμησης μέσω των συναρτήσεων απώλειας που είδαμε στα γραμμικά μοντέλα.

Η επικλινής ενίσχυση μπορεί θεωρητικά να συνδυαστεί με οποιοδήποτε τύπο μοντέλων. Στην πράξη ωστόσο συνδυάζεται με δενδρικά μοντέλα.

7.4 Τυχαίο Δάσος

Ένας άλλος τρόπος παραγωγής διαφορετικών μοντέλων, είναι η εισαγωγή της τυχαιότητας στην λειτουργία των αλγορίθμων μάθησης. Θα εστιάσουμε συγκεκριμένα σε δενδρικά μοντέλα και στην τεχνική του **τυχαίου δάσους** (random forest).

Η τεχνική του τυχαίου δάσους εισάγει την τυχαιότητα στην μάθηση δενδρικών μοντέλων, επιλέγοντας το καλύτερο ζεύγος μεταβλητής εισόδου και τιμής σε κάθε κόμβο μεταξύ όχι όλων των διαθέσιμων μεταβλητών εισόδου, αλλά ενός τυχαία επιλεγμένου υποσυνόλου τους. Μια καλή προκαθορισμένη τιμή για το μέγεθος του υποσυνόλου αυτού είναι $\log_2(n) + 1$ ή \sqrt{n} , όπου n το πλήθος των μεταβλητών εισόδου.

Επιπλέον, πριν η τεχνική του τυχαίου δάσους ξεκινήσει την εκπαίδευση ενός δενδρικού μοντέλου αξιοποιεί και τεχνική δειγματοληψίας και συγκεκριμένα δειγματοληψία των παραδειγμάτων εκπαίδευσης με επανατοποθέτηση, ακριβώς όπως στην αυτοδύναμη συνάθροιση.

7.5 Περαιτέρω Μελέτη

Περισσότερες πληροφορίες για τα σύνολα μοντέλων, μπορεί να βρει κανείς σε δύο βιβλία που εστιάζουν αποκλειστικά σε αυτήν την περιοχή της μηχανικής μάθηση [1, 2]. Μια περιγραφή του διαγωνισμού της NetFlix και της προόδου που είχε επιτευχθεί ως το 2007 μπορεί να βρει κανείς στο [3]. Για εφαρμογές συνόλων από δενδρικά μοντέλα στην επεξεργασία εικόνας και βίντεο, μπορεί κανείς να διαβάσει το [4]. Η φιλοσοφία της παραγωγής και του συγκερασμού ενός συνόλου από μοντέλα, έχει βρει εφαρμογή πέρα από την μάθηση με επίβλεψη και σε άλλες εργασίες μηχανικής μάθησης, σε πολύ μικρότερο βαθμό ωστόσο, όπως στην ομαδοποίηση [5] και στην επιλογή χαρακτηριστικών [6]. Για την προσαρμοστική ενίσχυση, περισσότερες πληροφορίες μπορεί να αντλήσει κανείς στην εργασία [7].

7.6 Ασκήσεις

1. Στον παρακάτω πίνακα, η πρώτη γραμμή μας δείχνει την κλάση (θετική ή αρνητική) για τα 6 παραδείγματα ενός συνόλου εκπαίδευσης. Η δεύτερη γραμμή μας δείχνει τις προβλέψεις του πρώτου μοντέλου που εκπαιδεύτηκε από αυτό το σύνολο εκπαίδευσης με την τεχνική AdaBoost.M1. Ποια θα είναι τα βάρη του κάθε παραδείγματος εκπαίδευσης κατά την εκπαίδευση του δεύτερου μοντέλου;

y	+	+	-	+	-	-
h_1	+	+	+	-	-	-

2. Έστω το παρακάτω σύνολο εκπαίδευσης στο οποίο εφαρμόζουμε τον αλγόριθμο Gradient Boosting. Τι τιμή θα δώσει στην έξοδό του, το πρώτο μοντέλο (h_0); Τι τιμή θα έχει η μεταβλητή εξόδου στα δεδομένα εκπαίδευσης του δεύτερου μοντέλου (h_1) για το παράδειγμα όπου $x_1 = 10$; Αν η έξοδος του συνόλου των δύο μοντέλων για κάποιο παράδειγμα είναι ίση με 120 και η συρρίκνωση $\eta = 0.1$, με τι ισούται η έξοδος του δεύτερου μοντέλου (h_1) στο παράδειγμα αυτό;

x_1	y
10	10
20	30
30	50
40	90
50	170
60	250

3. Έστω ένα σύνολο 6 παραδειγμάτων εκπαίδευσης. Ο παρακάτω πίνακας δείχνει τη συχνότητα καθενός από αυτά σε μια παραλλαγή του συνόλου εκπαίδευσης που έχει προκύψει με τη μέθοδο της αυτοδύναμης συνάθροισης (bagging), εκτός από ένα. Υπολογίστε τη συχνότητα που λείπει.

Παράδειγμα	1	2	3	4	5	6
Συχνότητα	0	1	?	2	0	3

4. Ο παρακάτω πίνακας περιλαμβάνει ένα σύνολο δεδομένων ελέγχου (x, y) καθώς και προβλέψεις από δύο διαφορετικά μοντέλα, h_1 και h_2 , για κάθε παράδειγμα του συνόλου. Χρησιμοποιώντας την τεχνική της σωρευμένης γενίκευσης έχουμε εκπαιδεύσει ένα γραμμικό μέτα-μοντέλο παλινδρόμησης με παραμέτρους $\theta_0 = 0.1, \theta_1 = 0.5, \theta_2 = 0.5$. Υπολογίστε την έξοδο του μετα-μοντέλου για το 1ο παράδειγμα ελέγχου.

x	y	h_1	h_2
0	2	1	2
-1	1	0.8	1.2
1	3	2.5	2.5
0	1.5	1.6	1

5. Έστω ότι έχουμε ένα σύνολο εκπαίδευσης 8 παραδειγμάτων (1 έως 8) και παράγουμε με τη χρήση της τεχνικής bagging (αυτοδύναμη συνάθροιση) τρεις παραλλαγές του με δειγματοληψία με επανατοποθέτηση. Οι συχνότητες εμφάνισης των παραδειγμάτων σε κάθε παραλλαγή δίνονται εν μέρει στον παρακάτω πίνακα.

Παράδειγμα	Παραλλαγή 1	Παραλλαγή 2	Παραλλαγή 3
1	1	0	2
2	0	2	?
3	2	1	0
4	0	0	1
5	3	?	0
6	?	2	1
7	0	1	?
8	2	0	2

- (α) Συμπλήρωσε τις τιμές που λείπουν στον πίνακα.
- (β) Ποια παραδείγματα είναι εκτός σάκου (out-of-bag) για κάθε παραλλαγή?

Βιβλιογραφία

- [1] Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall/CRC, 2012. ISBN: 1439830037.
- [2] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, 2004.
- [3] James Bennett, Stan Lanning και Netflix Netflix. “The Netflix Prize”. Στο: *KDD Cup and Workshop in conjunction with KDD*. 2007.
- [4] A. Criminisi και J. Shotton. *Decision Forests for Computer Vision and Medical Image Analysis*. Springer Publishing Company, Incorporated, 2013. ISBN: 1447149289.
- [5] Alexander Strehl και Joydeep Ghosh. “Cluster Ensembles — A Knowledge Reuse Framework for Combining Multiple Partitions”. Στο: *J. Mach. Learn. Res.* 3 (2002), σσ. 583–617.
- [6] Opeyemi A. Osanaiye κ.ά. “Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing”. Στο: *EURASIP Journal on Wireless Communications and Networking* 2016 (2016), σσ. 1–10.
- [7] Yoav Freund και Robert E. Schapire. “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting”. Στο: *Journal of Computer and System Sciences* 55 (1 1997). ISSN: 00220000. doi: [10.1006/jcss.1997.1504](https://doi.org/10.1006/jcss.1997.1504).

ΚΕΦΑΛΑΙΟ 8

ΕΝΙΣΧΥΤΙΚΗ ΜΑΘΗΣΗ

Στο κεφάλαιο αυτό:

Θα εστιάσουμε στην ενισχυτική μάθηση. Θα ξεκινήσουμε μελετώντας τις Μαρκοβιανές διαδικασίες απόφασης, που αποτελούν το μαθηματικό της υπόβαθρο, και τις τεχνικές δυναμικού προγραμματισμού που χρησιμοποιούνται για την επίλυση τους. Στη συνέχεια θα μελετήσουμε τις τεχνικές Μόντε Κάρλο και έπειτα τις τεχνικές χρονικών διαφορών.

Η ενισχυτική μάθηση (reinforcement learning) ασχολείται με τη μάθηση της συμπεριφοράς ενός πράκτορα, δηλαδή με τη μάθηση του τι ενέργειες πρέπει να εκτελεί με βάση την κατάσταση στην οποία βρίσκεται. Η μάθηση επιτυγχάνεται μέσω της αλληλεπίδρασης του πράκτορα με το περιβάλλον του, κατά την οποία εκτελεί ενέργειες και παρατηρεί ένα σήμα (καθυστερημένης) ανταμοιβής. Η μάθηση συνίσταται στην αναζήτηση της συμπεριφοράς που μεγιστοποιεί το άθροισμα των ανταμοιβών που λαμβάνει μακροπρόθεσμα.

8.1 Μαρκοβιανές Διαδικασίες Απόφασης

Οι εργασίες ενισχυτικής μάθησης μοντελοποιούνται μαθηματικά ως Μαρκοβιανές διαδικασίες απόφασης (ΜΔΑ). Ο πράκτορας αλληλεπιδρά με το περιβάλλον σε διακριτά χρονικά βήματα t , $t = 0, 1, 2, 3, \dots$. Σε κάθε χρονικό βήμα t , ο πράκτορας βρίσκεται σε μια **κατάσταση** $S_t \in \mathcal{S}$. Με βάση την τρέχουσα κατάσταση, ο πράκτορας επιλέγει μια **ενέργεια** $A_t \in \mathcal{A}(S_t)$. Ένα χρονικό βήμα αργότερα, εν μέρει ως αποτέλεσμα της ενέργειας του, ο πράκτορας λαμβάνει μια **ανταμοιβή** $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$ και βρίσκεται σε μια νέα κατάσταση S_{t+1} . Η αληλεπίδραση αυτή απεικονίζεται στο Σχήμα 8.1.

Σε μια πεπερασμένη Μαρκοβιανή διαδικασία απόφασης (ΠΜΔΑ), τα σύνολα \mathcal{S} , $\mathcal{A}(S_t) \forall S_t \in \mathcal{S}$ και \mathcal{R} είναι πεπερασμένα. Επιπλέον, στις ΜΔΑ θεωρούμε πως όλες οι καταστάσεις χαρακτηρίζονται από την ιδιότητα Markov, δηλαδή περιλαμβάνουν κάθε πληροφορία της παρελθοντικής αλληλεπίδρασης του πράκτορα με το περιβάλλον, η οποία μπορεί να είναι χρήσιμη για την επιλογή ενεργειών στο μέλλον:

$$P(S_{t+1} = s', R_{t+1} = r | S_t, A_t, R_t, \dots, R_1, S_0, A_0) = P(S_{t+1} = s', R_{t+1} = r | S_t, A_t)$$

Η δυναμική μιας MDA, καθορίζει την πιθανότητα μετάβασης του πράκτορα στο επόμενο χρονικό βήμα $t + 1$ σε μια κατάσταση s' και την ταυτόχρονη λήψη ανταμοιβής r , όταν στο χρονικό βήμα t βρίσκεται στην κατάσταση s και επιλέγει ενέργεια a :

$$p(s', r | s, a) : P(S_{t+1} = s', R_{t+1} = r | S_t = s, A_t = a)$$

Από την δυναμική αυτή, μπορούμε με τη χρήση παραγοντοποίησης να λάβουμε τις πιθανότητες μετάβασης:

$$p(s' | s, a) : P(S_{t+1} = s' | S_t = s, A_t = a) = \sum_{r \in \mathcal{R}} p(s', r | s, a)$$

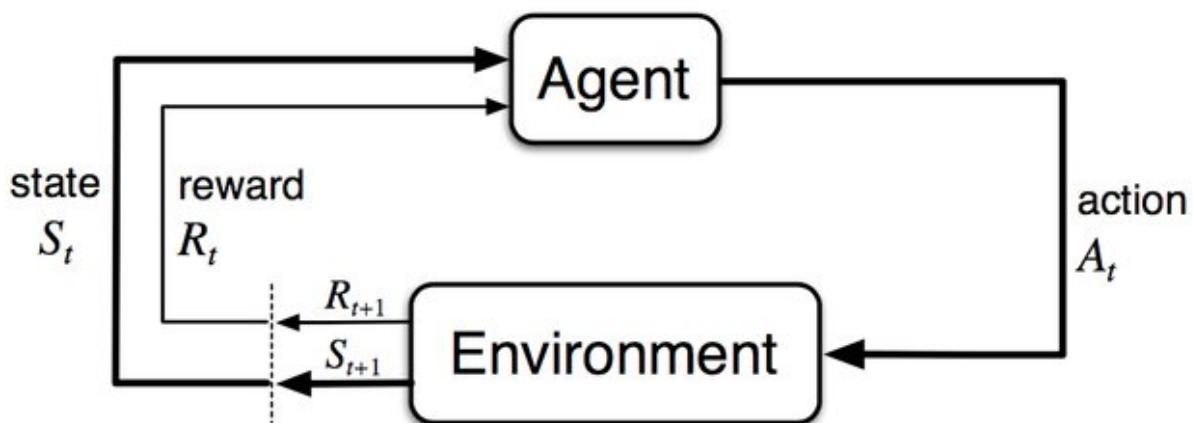
καθώς και την αναμενόμενη ανταμοιβή για μια τριάδα κατάστασης, ενέργειας και επόμενης κατάστασης:

$$r(s, a, s') : \mathbb{E}[R_{t+1} | S_t = s, A_t = a, S_{t+1} = s'] = \sum_{r \in \mathcal{R}} r \frac{p(s', r | s, a)}{p(s' | s, a)}$$

Μια εναλλακτική πιο συμπυκνωμένη αναπαράσταση των πιθανοτήτων μετάβασης και αναμενόμενων ανταμοιβών είναι $P_{ss'}^a$ και $R_{ss'}^a$ αντίστοιχα.

Η μοντελοποίηση της αλληλεπίδρασης του πράκτορα με το περιβάλλον ως MDA είναι αρκετά ευέλικτη. Τα χρονικά βήματα της αλληλεπίδρασης, δεν είναι απαραίτητο να αντιστοιχούν σε τακτά χρονικά διαστήματα. Μπορούν να αντιστοιχούν σε αυθαίρετα διαδοχικά βήματα επιλογής ενέργειας. Η κατάσταση σε ένα χρονικό βήμα t , μπορεί να περιλαμβάνει οποιαδήποτε πληροφορία είναι διαθέσιμη στον πράκτορα από το περιβάλλον του σε εκείνο το βήμα, όπως άμεσα ερεθίσματα, επεξεργασμένα ερεθίσματα, ή δομές που ενημερώνονται με την πάροδο του χρόνου από την ακολουθία ερεθισμάτων που δέχεται ο πράκτορας. Οι ενέργειες μπορεί να είναι από χαμηλού επιπέδου, όπως η τάση του ρεύματος που πρέπει να εφαρμοστεί στην μηχανή ενός ρομποτικού βραχίονα, έως υψηλού επιπέδου, όπως η επιλογή καταστήματος για μια βραδινή έξοδο, ή η επιλογή μεταπτυχιακού προγράμματος σπουδών.

Έστω για παράδειγμα η ακόλουθη PIMDA που αφορά ένα ρομπότ, το οποίο συλλέγει άδεια τενεκεδάκια αναψυκτικών. Υπάρχουν δύο καταστάσεις: Χαμηλή στάθμη μπαταρίας (low), γεμάτη μπαταρία (high), επομένως $\mathcal{S} = \{high, low\}$. Υπάρχουν τρεις διαθέσιμες ενέργειες: Αναμονή για τενεκεδάκι (wait), αναζήτηση για τενεκεδάκι (search), επιστροφή στη βάση για φόρτιση (recharge), όπου $\mathcal{A}(high) = \{search, wait\}$



Σχήμα 8.1: Η αλληλεπίδραση του πράκτορα με το περιβάλλον.

και $\mathcal{A}(low) = \{search, wait, recharge\}$. Οι ανταμοιβές που λαμβάνει το ρομπότ είναι -3 αν αδειάσει η μπαταρία του, και +1 για κάθε τενεκεδάκι που μαζεύει, με R^{search} και R^{wait} να είναι ο αναμενόμενος αριθμός από τενεκεδάκια που θα μαζέψει όταν τα αναζητά ή τα αναμένει αντίστοιχα ($R^{search} > R^{wait}$). Όταν η μπαταρία του είναι γεμάτη και εκτελεί αναζήτηση για τενεκεδάκι, η πιθανότητα η μπαταρία να συνεχίσει να είναι γεμάτη είναι c ενώ η πιθανότητα η στάθμη της μπαταρίας να γίνει χαμηλή είναι $1 - c$. Όταν η στάθμη της μπαταρίας είναι χαμηλή και εκτελεί αναζήτηση για τενεκεδάκι, η πιθανότητα να μην εξαντληθεί η μπαταρία είναι d , ενώ με πιθανότητα $1 - d$ η μπαταρία εξαντλείται. Σε αυτήν την περίπτωση, μαζεύουμε εμείς το ρομπότ και το τοποθετούμε στη βάση του για φόρτιση. Οι πιθανότητες μετάβασης και οι αναμενόμενες ανταμοιβές συνοψίζονται στον πίνακα 8.1.

$s_t = s$	$a_t = a$	$s_{t+1} = s'$	$p(s' s, a)$	$r(s, a, s')$
high	search	high	c	R^{search}
high	search	low	$1 - c$	R^{search}
low	search	high	$1 - d$	-3
low	search	low	d	R^{search}
high	wait	high	1	R^{wait}
high	wait	low	0	R^{wait}
low	wait	high	0	R^{wait}
low	wait	low	1	R^{wait}
low	recharge	high	1	0
low	recharge	low	0	0

Πίνακας 8.1: Δυναμική μιας ΠΜΔΑ για ένα ρομπότ ανακύκλωσης.

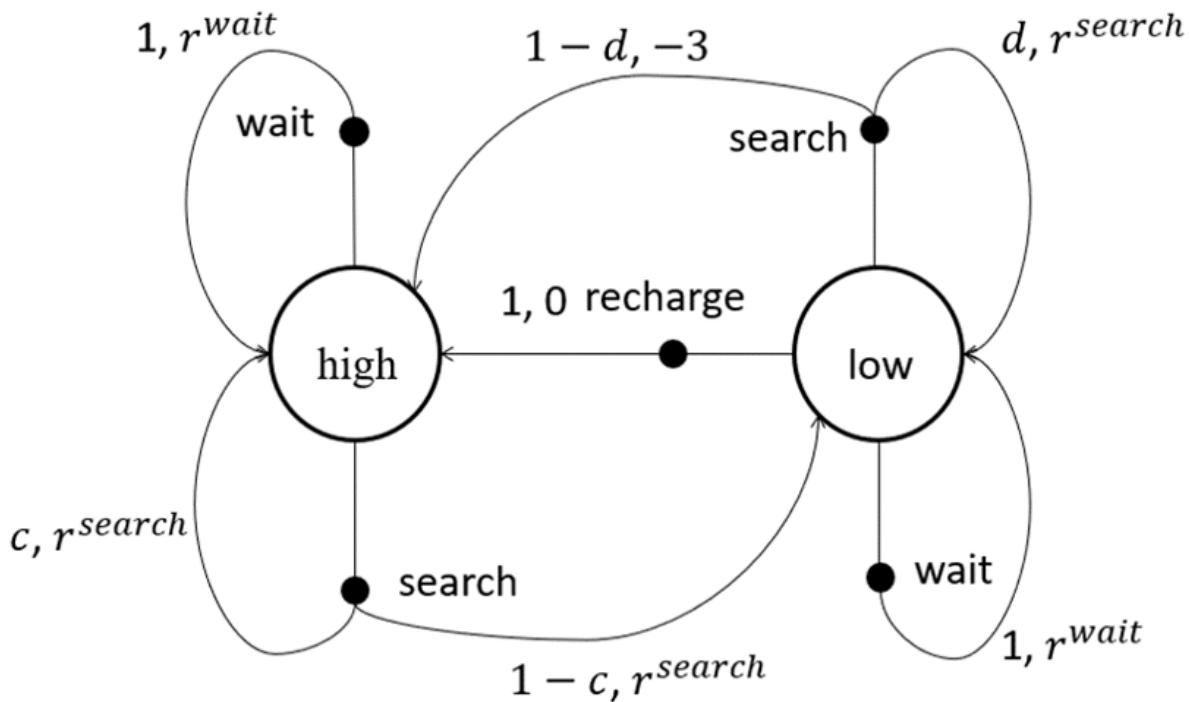
Ο γράφος μετάβασης (transition graph) είναι ένας τρόπος γραφικής αναπαράστασης μιας ΠΜΔΑ. Αποτελείται από κόμβους καταστάσεων και κόμβους ενεργειών. Οι κόμβοι καταστάσεων αναπαρίστανται με έναν μεγάλο κύκλο με το όνομα της κατάστασης ως ετικέτα. Σε έναν γράφο μετάβασης υπάρχει ένας κόμβος κατάστασης για κάθε κατάσταση. Οι κόμβοι ενεργειών αναπαρίστανται με μία μικρή βούλα με το όνομα της ενέργειας ως ετικέτα. Σε έναν γράφο μετάβασης υπάρχει ένας κόμβος ενέργειας για κάθε δυνατό ζεύγος κατάστασης - ενέργειας. Ξεκινώντας από μία κατάσταση s και εκτελώντας την ενέργεια a μεταβαίνουμε στον κόμβο ενέργειας (s, a) . Η μετάβαση αυτή αναπαρίσταται με μια γραμμή, η οποία συνδέει τον κόμβο κατάστασης s με τον κόμβο ενέργειας (s, a) . Το περιβάλλον απαντά με μια μετάβαση στην επόμενη κατάσταση s' , η οποία αναπαρίσταται με ένα βέλος από τον κόμβο ενέργειας (s, a) στην κατάσταση s' . Κάθε βέλος αντιστοιχεί σε μια τριάδα (s, a, s') και έχει ως ετικέτα ένα ζεύγος τιμών με την πιθανότητα μετάβασης $p(s' | s, a)$, και την αναμενόμενη τιμή της ανταμοιβής για αυτή τη μετάβαση $r(s, a, s')$. Το άθροισμα των πιθανοτήτων μετάβασης που υπάρχουν ως ετικέτες στα βέλη που φεύγουν από έναν κόμβο ενέργειας ισούται με 1. Το Σχήμα 8.2 δείχνει το γράφο μετάβασης για την ΠΜΔΑ του προηγούμενου παραδείγματος.

8.1.1 Ανταμοιβές και κέρδος

Ο στόχος ενός πράκτορα ενισχυτικής μάθησης καθορίζεται από τις ανταμοιβές. Ο πράκτορας προσπαθεί να μεγιστοποιήσει το άθροισμα των ανταμοιβών που θα λάβει μακροπρόθεσμα από το χρονικό βήμα t που βρίσκεται και έπειτα. Το άθροισμα αυτό ονομάζεται αναμενόμενο **κέρδος** (return):

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \dots \quad (8.1)$$

Για παράδειγμα, αν θέλουμε ένα ρομπότ να μάθει να περπατάει, θα μπορούσαμε να ορίσουμε θετική ανταμοιβή ανάλογη του πόσο πολύ προχώρησε σε κάθε χρονικό βήμα. Αν θέλουμε να μάθει να βρίσκει την έξοδο σε έναν λαβύρινθο, θα μπορούσαμε να ορίσουμε ανταμοιβή 1 αν βγει και 0 σε κάθε άλλη μετάβαση. Αυτό ωστόσο δεν θα ωθούσε τον πράκτορα να μάθει να βγαίνει γρήγορα, καθώς η ανταμοιβή είναι η ίδια ανεξάρτητα από



Σχήμα 8.2: Γράφος μετάβασης της ΠΜΔΑ για το ρομπότ ανακύκλωσης.

τα χρονικά βήματα που θα χρειαστεί. Ένας καλύτερος ορισμός ανταμοιβών θα ήταν 0 αν βγει και -1 σε κάθε άλλη μετάβαση.

Οι ανταμοιβές δεν θα πρέπει να καθορίζουν το πως θα πετύχει ο πράκτορας αυτό που επιθυμούμε, αλλά το τι είναι αυτό που επιθυμούμε να πετύχει. Για παράδειγμα, αν θέλουμε ένας πράκτορας να μάθει να κερδίζει μια παρτίδα σκάκι, θα μπορούσαμε να του δώσουμε ανταμοιβή 1 αν νικήσει, -1 αν χάσει και 0 αν η παρτίδα λήξει ισόπαλη. Δεν θα ήταν ωστόσο σωστό το να δώσουμε ανταμοιβές ανάλογες των κομματιών του αντιπάλου που θα κερδίσει ή βάσει στρατηγικών επιτευγμάτων (π.χ. έλεγχος του κέντρου της σκακιέρας).

Υπάρχουν δύο βασικές κατηγορίες εργασιών ενισχυτικής μάθησης, αυτές που η αλληλεπίδραση του πράκτορα με το περιβάλλον χωρίζεται από τη φύση της σε υπακολουθίες που καλούνται **επεισόδια** και οι **συνεχείς**. Παραδείγματα επεισοδίων είναι μια παρτίδα ενός παιχνιδιού και μία διαδρομή ως την έξοδο από έναν λαβύρινθο. Σε αυτήν την κατηγορία εργασιών υπάρχει τονλάχιστον μία **τερματική κατάσταση** όπου ο πράκτορας είναι εγγυημένο να φτάσει. Αντίθετα, ο έλεγχος ενός πυρηνικού αντιδραστήρα και ο έλεγχος ενός ρομπότ με μακρά διάρκεια λειτουργίας αποτελούν παραδείγματα συνεχών εργασιών. Στις εργασίες που χωρίζονται σε επεισόδια, υπάρχει πάντα ένα τελικό χρονικό βήμα T στο οποίο η αλληλεπίδραση τελειώνει. Επομένως το κέρδος σε αυτήν την κατηγορία εργασιών, μπορεί να οριστεί και ως εξής:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \dots + R_T \quad (8.2)$$

Σε συνεχείς εργασίες όμως, όπου το πλήθος των αποφάσεων θεωρείται άπειρο, το κέρδος θα είναι επίσης άπειρο, με αποτέλεσμα να μην μπορούμε να διακρίνουμε μεταξύ καλύτερων και χειρότερων ενεργειών. Για να αντιμετωπιστεί αυτό το πρόβλημα, το κέρδος σε συνεχείς εργασίες ορίζεται μέσω μιας βοηθητικής έννοιας, της **έκπτωσης** (discounting):

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (8.3)$$

όπου γ μια παράμετρος με τιμές $0 \leq \gamma < 1$, η οποία ονομάζεται **ρυθμός έκπτωσης** (discount rate). Η έκπτωση μας εξασφαλίζει ότι αν οι ανταμοιβές είναι φραγμένες, τότε και το κέρδος είναι φραγμένο:

$$\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \leq \sum_{k=0}^{\infty} \gamma^k R_{\max} = \frac{R_{\max}}{1 - \gamma} \quad (8.4)$$

Θα ήταν χρήσιμο να μπορούμε να ορίσουμε με έναν ενιαίο τρόπο το κέρδος τόσο για συνεχείς εργασίες όσο και για εργασίες με επεισόδια. Ένας πρώτος τρόπος για να επιτευχθεί αυτό είναι ορίζοντας την έννοια της **απορροφητικής κατάστασης**. Σε μια τέτοια κατάσταση, ανεξάρτητα από την επιλεγόμενη ενέργεια, ο πράκτορας παραμένει στην ίδια κατάσταση και πάιρνει ανταμοιβή 0. Θεωρώντας τις τερματικές καταστάσεις σε εργασίες με επεισόδια ως απορροφητικές, το κέρδος μπορεί να οριστεί ως εξής:

$$G_t = \inf_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (8.5)$$

όπου ο γ επιτρέπεται να είναι και 1 σε εργασίες με επεισόδια

Χωρίς την έννοια της απορροφητικής κατάστασης, μπορούμε να ορίσουμε το κέρδος ενιαία ως εξής:

$$G_t = \sum_{k=t+1}^T \gamma^{k-t-1} R_k = \sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1} \quad (8.6)$$

όπου ο γ επιτρέπεται να είναι 1 και το T να είναι άπειρο, αλλά όχι ταυτόχρονα.

8.1.2 Συναρτήσεις Αξίας

Η **πολιτική** (policy) ενός πράκτορα είναι μία συνάρτηση απεικόνισης $\pi(s)$ μιας κατάστασης $s \in \mathcal{S}$ σε μία ενέργεια που δύναται να εκτελεστεί σε αυτήν την κατάσταση $a \in \mathcal{A}(s)$. Δηλαδή, η πολιτική ενός πράκτορα καθορίζει τις αποφάσεις που αυτός θα λάβει σε κάθε χρονικό βήμα. Θεωρείστε ότι οι πολιτικές μπορεί να είναι στοχαστικές. Στην ίδια κατάσταση ενδέχεται να επιλεχθεί διαφορετική ενέργεια με κάποια πιθανότητα. Η πολιτική π μας δίνει αυτήν την πιθανότητα $\pi(a|s)$.

Οι συναρτήσεις αξίας δίνουν μια εκτίμηση του πόσο καλό είναι για έναν πράκτορα, δεδομένης μιας πολιτικής που αυτός ακολουθεί: i) να βρίσκεται σε μια κατάσταση, και ii) να επιλέγει μια ενέργεια όταν βρίσκεται σε μια κατάσταση. Η εκτίμηση αυτή στηρίζεται στο αναμενόμενο κέρδος, το οποίο εξαρτάται από τις ενέργειες που θα εκτελέσει ο πράκτορας στη συνέχεια, οι οποίες εξαρτώνται από την πολιτική του πράκτορα, αλλά και που αυτές θα καταλήξουν.

Η συνάρτηση αξίας κατάστασης δίνεται από την παρακάτω εξίσωση:

$$v_{\pi}(s) = \mathbb{E}_{\pi} \{ G_t | S_t = s \} = \mathbb{E}_{\pi} \left\{ \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s \right\} \quad (8.7)$$

Η συνάρτηση αξίας ενέργειας δίνεται από την παρακάτω εξίσωση:

$$\begin{aligned} q_{\pi}(s, a) &= \mathbb{E}_{\pi} \{ G_t | S_t = s, A_t = a \} \\ &= \mathbb{E}_{\pi} \left\{ \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a \right\} \end{aligned} \quad (8.8)$$

$E_{\pi} \{ \}$ είναι η αναμενόμενη τιμή της παράστασης μέσα στις αγκύλες δεδομένου ότι ο πράκτορας θα ακολουθήσει την πολιτική π .

Οι συναρτήσεις αξίας ικανοποιούν μια αναδρομική σχέση που ονομάζεται **εξίσωση του Bellman**. Η εξίσωση του Bellman για την v_{π} δίνεται από την παρακάτω εξίσωση:

$$\begin{aligned}
v_\pi(s) &= \mathbb{E}_\pi \{G_t | S_t = s\} = \mathbb{E}_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \middle| S_t = s \right\} \\
&= \mathbb{E}_\pi \left\{ R_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k R_{t+k+2} \middle| S_t = s \right\} \\
&= \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a \left[R_{ss'}^a + \gamma E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \middle| s_{t+1} = s' \right\} \right] \\
&= \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma v_\pi(s')] \tag{8.9}
\end{aligned}$$

Η εξίσωση του Bellman για την v_π ορίζει μια σχέση ανάμεσα στην αξία μιας κατάστασης και την αξία της επόμενης κατάστασης. Ορίζει ότι η αξία μιας κατάστασης θα πρέπει να ισούται με την αξία της επόμενης κατάστασης (με έκπτωση) και της ανταμοιβής που προκύπτει. Λαμβάνει υπόψη όλες τις επόμενες καταστάσεις με βάση την πιθανότητα που έχουν να προκύψουν (σταθμισμένο άθροισμα).

Έστω για παράδειγμα ο κόσμος πλέγματος 5×5 που φαίνεται στο Σχήμα 8.3(a). Οι καταστάσεις είναι τα κελιά του πλέγματος. Δεν υπάρχουν τερματικές καταστάσεις ($\gamma = 0.9$). Υπάρχουν 4 δυνατές ενέργειες (επάνω, κάτω, δεξιά, αριστερά) που οδηγούν αιτιοκρατικά στις αντίστοιχες καταστάσεις, εκτός από εκείνες που οδηγούν εκτός πλέγματος, στις οποίες ο πράκτορας παραμένει στην κατάσταση που βρίσκεται αλλά παίρνει -1 ως ανταμοιβή. Επιπλέον από την κατάσταση A (B) ο πράκτορας μεταβαίνει στην κατάσταση A' (B') και παίρνει ως ανταμοιβή +10 (+5). Η ανταμοιβή είναι 0 για κάθε άλλη μετάβαση. Το Σχήμα 8.3(β) δείχνει την τιμή της v_π (με ακρίβεια ενός δεκαδικού ψηφίου), όπου η πολιτική τυχαίας επιλογής ενέργειας με ίδια πιθανότητα.

	A		B	
			+5	
	+10		B'	
	A'			

(a)

3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

(β)

Σχήμα 8.3: Ένας κόσμος πλέγματος 5×5 και η αντίστοιχη συνάρτηση αξίας κατάστασης.

Ας επαληθεύσουμε ότι η εξίσωση Bellman ισχύει για την κατάσταση στο κέντρο του πλέγματος. Η εξίσωση Bellman για την κατάσταση στο κέντρο (γραμμή 3, στήλη 3) έχει ως εξής:

$$v_\pi(3, 3) = \sum_a \pi((3, 3), a) \sum_{s'} P_{(3,3)s'}^a [R_{(3,3)s'}^a + \gamma v_\pi(s')] \tag{8.10}$$

Επειδή η πολιτική δίνει ίση πιθανότητα σε κάθε ενέργεια $a \in \{\text{επάνω, δεξιά, κάτω, αριστερά}\}$, έχουμε $\pi((3, 3), a) = \frac{1}{4}$. Επειδή οι ενέργειες οδηγούν αιτιοκρατικά στις αντίστοιχες καταστάσεις έχουμε $P_{(3,3)(2,3)}^{\text{επάνω}} = 1$ και $P_{(3,3)s'}^{\text{επάνω}} = 0$ για κάθε άλλη κατάσταση $s' \neq (2, 3)$, $P_{(3,3)(3,4)}^{\text{δεξιά}} = 1$ και $P_{(3,3)s'}^{\text{δεξιά}} = 0$ για κάθε άλλη κατάσταση $s' \neq (3, 4)$, $P_{(3,3)(4,3)}^{\text{κάτω}} = 1$ και $P_{(3,3)s'}^{\text{κάτω}} = 0$ για κάθε άλλη κατάσταση $s' \neq (4, 3)$ και τέλος $P_{(3,3)(3,2)}^{\text{αριστερά}} = 1$ και $P_{(3,3)s'}^{\text{αριστερά}} = 0$ για κάθε άλλη κατάσταση $s' \neq (3, 2)$.

Επομένως έχουμε:

$$\begin{aligned}
 v_{\pi(3,3)} &= \frac{1}{4} \cdot 1 \cdot [R_{(3,3)(2,3)}^{\text{επάνω}} + 0.9v_{\pi}(2,3)] + \frac{1}{4} \cdot 1 \cdot [R_{(3,3)(3,4)}^{\delta\text{εξία}} + 0.9v_{\pi}(3,4)] + \\
 &\quad \frac{1}{4} \cdot 1 \cdot [R_{(3,3)(4,3)}^{\kappa\text{άτω}} + 0.9v_{\pi}(4,3)] + \frac{1}{4} \cdot 1 \cdot [R_{(3,3)(3,2)}^{\alpha\text{ριστερά}} + 0.9v_{\pi}(3,2)] \\
 &= \frac{1}{4}[0 + 0.9 \cdot 2.3] + \frac{1}{4}[0 + 0.9 \cdot 0.4] + \\
 &\quad \frac{1}{4}[0 + 0.9 \cdot (-0.4)] + \frac{1}{4}[0 + 0.9 \cdot 0.7] \\
 &= \frac{1}{4} \cdot 0.9 \cdot [2.3 + 0.4 - 0.4 + 0.7] = \frac{1}{4} \cdot 0.9 \cdot 3 = 0.675
 \end{aligned}$$

Με ακρίβεια ενός δεκαδικού ψηφίου το 0.675 γίνεται 0.7, ίσο δηλαδή με την τιμή της $v_{\pi}(3,3)$, επομένως η εξίσωση Bellman ισχύει.

8.1.3 Βέλτιστες συναρτήσεις αξίας

Μια πολιτική π είναι καλύτερη ή ίδια σε σχέση με μία άλλη πολιτική π' αν και μόνο αν το αναμενόμενο κέρδος της είναι μεγαλύτερο ή ίδιο με το κέρδος της π' για όλες τις καταστάσεις: $\pi \geq \pi' \Leftrightarrow v_{\pi}(s) \geq v_{\pi'}(s) \forall s \in S$. Στις ΠΜΔΑ υπάρχει τουλάχιστον μία πολιτική που είναι καλύτερη ή το ίδιο καλή σε σχέση με τις υπόλοιπες. Ονομάζεται **βέλτιστη πολιτική** και συμβολίζεται ως π_* . Στα προβλήματα λήψης ακολουθιακών αποφάσεων, στόχος είναι η εύρεση της βέλτιστης πολιτικής, ή αλλιώς η επίλυση της ΠΜΔΑ.

Στις βέλτιστες πολιτικές αντιστοιχούν βέλτιστες συναρτήσεις αξίας κατάστασης, οι οποίες συμβολίζονται ως v_* και ορίζονται ως εξής:

$$v_*(s) = \max_{\pi} v_{\pi}(s) \quad \forall s \in S \quad (8.11)$$

Στις βέλτιστες πολιτικές αντιστοιχούν βέλτιστες συναρτήσεις αξίας ενέργειας, οι οποίες συμβολίζονται ως q_* και ορίζονται ως εξής:

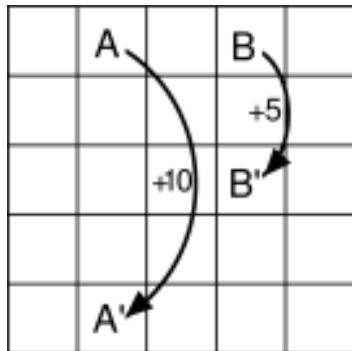
$$q_*(s, a) = \max_{\pi} Q_{\pi}(s, a) \quad \forall s \in S, a \in A(s) \quad (8.12)$$

Ως συναρτήσεις αξίας μιας πολιτικής, οι βέλτιστες συναρτήσεις αξίας θα πρέπει να ικανοποιούν τις εξισώσεις του Bellman. Για τη v_* η εξίσωση εκφράζει το γεγονός ότι η αξία μιας κατάστασης κάτω από μια βέλτιστη πολιτική θα πρέπει να ισούται με το αναμενόμενο κέρδος αν εκτελέσουμε την καλύτερη ενέργεια από εκείνη την κατάσταση: $v_*(s) = \max_a \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma v_*(s')]$. Για τη q_* η εξίσωση εκφράζει το γεγονός ότι η αξία ενός ζεύγους κατάστασης - ενέργειας κάτω από μια βέλτιστη πολιτική θα πρέπει να ισούται με το αναμενόμενο κέρδος αν λάβουμε την καλύτερη ενέργεια από την κατάσταση στην οποία θα βρεθούμε: $q_*(s, a) = \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma \max_{a'} q_*(s', a')]$.

Δεδομένης της q_* ο πράκτορας ακολουθεί μια βέλτιστη πολιτική επιλέγοντας σε κάθε κατάσταση s εκείνη την ενέργεια a που μεγιστοποιεί την ποσότητα $q_*(s, a)$. Δεδομένης της v_* ο πράκτορας ακολουθεί μια βέλτιστη πολιτική επιλέγοντας σε κάθε κατάσταση s εκείνη την ενέργεια η οποία οδηγεί στο μέγιστο αναμενόμενο κέρδος στο επόμενο βήμα σύμφωνα με τη βέλτιστη εξίσωση Bellman. Εφόσον οι εξισώσεις είναι βέλτιστες, η επιλογή ενέργειας σε μια κατάσταση είναι και μακροπρόθεσμα βέλτιστη. Επομένως η εύρεση της βέλτιστης πολιτικής στα προβλήματα λήψης ακολουθιακών αποφάσεων, μπορεί να προκύψει από τον υπολογισμό μίας εκ των v_* και q_* .

Το Σχήμα 8.4 δείχνει: α) τον κόσμο πλέγματος από το προηγούμενο παράδειγμα, β) την τιμή της βέλτιστης συνάρτησης αξίας κατάστασης v_* , και γ) τη βέλτιστη πολιτική π^* . Παρατηρούμε πως η τιμή της καλύτερης κατάστασης είναι 24.4. Ας υπολογίσουμε αυτήν την τιμή με ακρίβεια 3 δεκαδικών δεδομένου ότι εκφράζει το αναμενόμενο κέρδος κάτω από τη βέλτιστη πολιτική. Από την κατάσταση με την καλύτερη τιμή και ακολουθώντας τη βέλτιστη πολιτική θα παίρνουμε ανταμοιβές $v_*(1, 2) = 10 + 0.9 \cdot 0 + 0.9^2 \cdot 0 + 0.9^3 \cdot 0 + 0.9^4$.

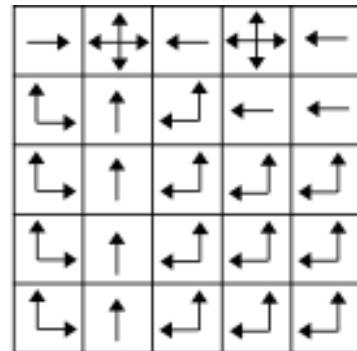
$$0 + 0.9^5 \cdot 10 + \dots = 10 + 0.9^5 \cdot 10 + 0.9^{10} \cdot 10 + 0.9^{15} \cdot 10 + \dots = 10(1 + 0.9^5 + 0.9^{10} + 0.9^{15} + \dots) = 10[(0.9^5)^0 + (0.9^5)^1 + (0.9^5)^2 + (0.9^5)^3 + \dots] = \frac{10}{1 - 0.9^5} = \frac{10}{1 - 0.59049} = 24.419.$$



α)

22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

β)



γ)

Σχήμα 8.4: Βέλτιστη πολιτική και συνάρτηση αξίας στον κόσμο πλέγματος.

Για την εύρεση της βέλτιστης πολιτικής μπορεί να εκτελεστεί αναζήτηση. Όμως αν έχουμε $|\mathcal{S}|$ καταστάσεις και $|\mathcal{A}|$ ενέργειες σε κάθε κατάσταση, τότε η χρονική πολυπλοκότητα είναι $O(|\mathcal{A}|^{|S|})$. Εναλλακτικά, μπορεί να εφαρμοστεί γραμμικός προγραμματισμός, δεδομένου ότι οι βέλτιστες εξισώσεις Bellman είναι στην ουσία συστήματα $|S|$ μη γραμμικών εξισώσεων (λόγω του τελεστή \max) με $|S|$ αγνώστους. Η χρονική πολυπλοκότητα του γραμμικού προγραμματισμού είναι πολυωνυμικής τάξης. Στη συνέχεια θα δούμε δυο αλγορίθμους δυναμικού προγραμματισμού, οι οποίοι έχουν επίσης πολυωνυμική τάξη χρονικής πολυπλοκότητας, όμως στην πράξη κλιμακώνονται καλύτερα από το γραμμικό προγραμματισμό και μπορούν να χρησιμοποιηθούν για την επίλυση ΠΜΔΑ με εκατομμύρια καταστάσεις.

8.2 Δυναμικός Προγραμματισμός

Ο **δυναμικός προγραμματισμός** (dynamic programming) είναι μια οικογένεια αλγορίθμων για τον υπολογισμό της βέλτιστης πολιτικής σε μια ΠΜΔΑ. Η κεντρική ιδέα στον δυναμικό προγραμματισμό (ΔP) είναι η αναζήτηση καλών πολιτικών με βάση τις συναρτήσεις αξίας. Αν βρούμε τις βέλτιστες συναρτήσεις αξίας, τότε εύκολα βρίσκουμε και τη βέλτιστη πολιτική.

Θα μελετήσουμε τους παρακάτω αλγορίθμους ΔP :

- Επανάληψη Πολιτικής
- Επανάληψη Αξίας

8.2.1 Επανάληψη πολιτικής

Ο αλγόριθμος της **επανάληψης πολιτικής** (policy iteration) έχει ως στόχο την προσέγγιση της βέλτιστης πολιτικής π^* , ξεκινώντας από μία τυχαία πολιτική π και συνεχίζοντας με επαναληπτική εκτέλεση του παρακάτω ζεύγους διαδικασιών:

- Υπολογισμός της v_π ή/και της q_π , για την τρέχουσα πολιτική π , διαδικασία που ονομάζεται **αξιολόγηση πολιτικής** (policy evaluation)
- Βελτίωση της τρέχουσας πολιτικής, διαδικασία που ονομάζεται **βελτίωση πολιτικής** (policy improvement)

Ένας τρόπος για αξιολόγηση πολιτικής είναι βάσει τεχνικών της γραμμικής άλγεβρας. Η εξίσωση Bellman για την v_π ορίζει ένα σύστημα $|\mathcal{S}|$ γραμμικών εξισώσεων με $|\mathcal{S}|$ αγνώστους. Θα μπορούσε να χρησιμοποιηθεί μια μέθοδος για την επίλυση τέτοιων συστημάτων, όπως η μέθοδος απαλοιφής κατά Gauss, η οποία όμως έχει υπολογιστική πολυπλοκότητα $O(|\mathcal{S}|^3)$.

Οι αλγόριθμοι ΔΠ που θα εξεταστούν υπολογίζουν την v_π με επαναληπτικό τρόπο. Έστω μια ακολουθία v_0, v_1, v_2, \dots από προσεγγιστικές συναρτήσεις αξίας κατάστασης, $v_i : S \rightarrow \mathbb{R}$. Η αρχική προσέγγιση v_0 επιλέγεται τυχαία. Αν υπάρχουν τελικές καταστάσεις, ορίζουμε την τιμή της v_0 ίση με 0 για τις καταστάσεις αυτές. Η κάθε επόμενη προσέγγιση υπολογίζεται από την προηγούμενη χρησιμοποιώντας την εξίσωση Bellman ως κανόνα ενημέρωσης για κάθε $s \in \mathcal{S}$, όπως φαίνεται στην εξίσωση 8.13. Η ακολουθία $\{v_k\}$ συγκλίνει στην v_π καθώς $k \rightarrow \infty$.

$$V_{k+1}(s) \leftarrow \sum_{\alpha} \pi(s, \alpha) \sum_{s'} P_{ss'}^\alpha [R_{ss'}^\alpha + \gamma V_k(s')] \quad (8.13)$$

Έστω για παράδειγμα ο κόσμος πλέγματος 4×4 που φαίνεται στο Σχήμα 8.5. Οι καταστάσεις είναι τα κελιά του πλέγματος. Οι τερματικές καταστάσεις είναι τα γκρι κελιά ($\gamma = 1$). Υπάρχουν 4 δυνατές ενέργειες (επάνω, κάτω, δεξιά, αριστερά) που οδηγούν αιτιοκρατικά στις αντίστοιχες καταστάσεις, εκτός από εκείνες που οδηγούν εκτός πλέγματος, στις οποίες ο πράκτορας παραμένει στην κατάσταση που βρίσκεται. Η ανταμοιβή είναι -1 για κάθε μετάβαση μέχρι και την τερματική κατάσταση.

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Σχήμα 8.5: Κόσμος πλέγματος 4×4 .

Έστω πολιτική, σύμφωνα με την οποία ο πράκτορας επιλέγει τυχαία μια από τις 4 διαθέσιμες ενέργειες σε κάθε κατάσταση, δηλαδή $\pi(s, \text{επάνω}) = \pi(s, \text{κάτω}) = \pi(s, \text{αριστερά}) = \pi(s, \text{δεξιά}) = \frac{1}{4}$ για κάθε $s \in S$. Θα ξεκινήσουμε από μια αρχική προσέγγιση v_0 της συνάρτησης αξίας κατάστασης, σύμφωνα με την οποία η αξία κάθε κατάστασης είναι 0, δηλαδή $v_0(s) = 0$ για κάθε $s \in S$. Το Σχήμα 8.6 δείχνει τις διαδοχικές εκτιμήσεις της συνάρτησης αξίας κατάστασης έπειτα από $k = 1, 2, 3$ και 10 βήματα, καθώς και τις πραγματικές τιμές της, στις οποίες θα συγκλίνει η διαδικασία της επαναληπτικής αξιολόγησης πολιτικής έπειτα από άπειρα βήματα.

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

(α) $k=0$

0	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	0

(β) $k=1$

0	-1.7	-2	-2
-1.7	-2	-2	-2
-2	-2	-2	-1.7
-2	-2	-1.7	0

(γ) $k=2$

0	-2.4	-2.9	-3
-2.4	-2.9	-3	-2.9
-2.9	-3	-2.9	-2.4
-3	-2.9	-2.4	0

(δ) $k=3$

0	-6.1	-8.4	-9
-6.1	-7.7	-8.4	-8.4
-8.4	-8.4	-7.7	-6.1
-9	-8.4	-6.1	0

(ε) $k=10$

0	-14	-20	-22
-14	-18	-20	-20
-20	-20	-18	-14
-22	-20	-14	0

(στ) $k=\infty$

Σχήμα 8.6: Παράδειγμα της επαναληπτικής αξιολόγησης πολιτικής.

Για την υλοποίηση της παραπάνω διαδικασίας στον υπολογιστή, απαιτείται ένας πίνακας με την τρέχουσα

εκτίμηση v_k της v_π για κάθε κατάσταση. Επίσης, απαιτείται και ένας δεύτερος πίνακας στον οποίο θα αποθηκεύεται η νέα εκτίμηση v_{k+1} της v_π κατά την εφαρμογή της εξίσωσης 8.13. Αφού εφαρμοστεί η εξίσωση 8.13 για όλες τις καταστάσεις, οι τιμές του δεύτερου πίνακα αντιγράφονται στον πρώτο και η διαδικασία επαναλαμβάνεται. Ωστόσο, οι αλγόριθμοι ΔΠ χρησιμοποιούν συνήθως έναν και μόνο πίνακα, οπότε η ενημέρωση των τιμών γίνεται **επιτόπια** (in place) βάσει των τιμών που περιέχει ο πίνακας εκείνη τη στιγμή. Καθώς εφαρμόζεται η εξίσωση 8.13 για κάθε μία από τις καταστάσεις, οι τιμές του πίνακα αρχικά αντιστοιχούν σε εκείνες της προηγούμενης εκτίμησης, όμως στην πορεία, λαμβάνονται υπόψη και νέες τιμές που υπολογίστηκαν νωρίτερα κατά τη διάρκεια αυτού του βήματος ενημέρωσης. Χρησιμοποιώντας αυτήν την υλοποίηση, η ακολουθία $\{v_k\}$ συγκλίνει και πάλι στην v_π και μάλιστα γρηγορότερα.

Ένα άλλο ζήτημα υλοποίησης αφορά το πόσες επαναλήψεις θα πρέπει να εκτελούνται στην πράξη, αφού θεωρητικά απαιτούνται άπειρες για να επιτευχθεί η σύγκλιση. Συνήθως, σταματάμε τις επαναλήψεις όταν η μεγαλύτερη διαφορά στην τιμή της συνάρτησης αξίας κατάστασης μέσα σε ένα βήμα, δηλαδή η ποσότητα $\max_{s \in \mathcal{S}} |v_{k+1}(s) - v_k(s)|$, είναι αρκετά μικρή. Ο αλγόριθμος 4 υλοποιεί τη διαδικασία της επαναληπτικής αξιολόγησης πολιτικής.

Algorithm 4 Ο αλγόριθμος επαναληπτική αξιολόγηση πολιτικής.

- 1: **Είσοδος:** ΠΜΔΑ $\langle \mathcal{S}, \mathcal{A}, P, R \rangle$, πολιτική π , ρυθμός έκπτωσης γ , μικρός θετικός αριθμός θ
 - 2: **Έξοδος:** Συνάρτηση αξίας κατάστασης v_π
 - 3: Τυχαία αρχικοποίηση της v_π
 - 4: **repeat**
 - 5: $\Delta \leftarrow 0$
 - 6: **for** $s \in S$ **do**
 - 7: $v \leftarrow v_\pi(s)$
 - 8: $v_\pi(s) \leftarrow \sum_a \pi(s, a) \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma v_\pi(s')]$
 - 9: $\Delta \leftarrow \max(\Delta, |v - v_\pi(s)|)$
 - 10: **end for**
 - 11: **until** $\Delta < \theta$
 - 12: **return** v_π
-

Ο λόγος για τον οποίο κάνουμε αξιολόγηση πολιτικής είναι για να μπορέσουμε να βρούμε καλύτερες πολιτικές. Ένας τρόπος να το πετύχουμε αυτό είναι να αλλάξουμε μια πολιτική έτσι ώστε να επιλέγει ενέργειες άπληστα σε κάθε κατάσταση σύμφωνα με τη συνάρτηση αξίας ενέργειας που υπολογίστηκε από αυτήν την πολιτική. Έτσι, για κάθε κατάσταση $s \in \mathcal{S}$, η νέα βελτιωμένη πολιτική που μπορούμε να πάρουμε είναι:

$$\pi'(s) = \arg \max_{\alpha} q_{\pi}(s, \alpha) = \arg \max_{\alpha} \sum_{s'} P_{ss'}^{\alpha} [R_{ss'}^{\alpha} + \gamma v(s')] \quad (8.14)$$

Στον κόσμο πλέγματος που εξετάζουμε ως παράδειγμα σε αυτήν την ενότητα, ξεκινήσαμε από μια πολιτική, έστω π_0 , η οποία επιλέγει σε κάθε κατάσταση ισοπίθανα όλες τις δυνατές ενέργειες και υπολογίσαμε τη συνάρτηση αξίας κατάστασης για αυτήν την πολιτική. Προχωρώντας στη διαδικασία της βελτίωσης αυτής της πολιτικής, εφαρμόζουμε την εξίσωση 8.14 και παίρνουμε μια νέα πολιτική, έστω π_1 . Ο Πίνακας 8.2 δείχνει την αναλυτική μορφή των δύο πολιτικών.

Ας εξετάσουμε λεπτομερώς πως υπολογίστηκε η νέα πολιτική για την κατάσταση 5. Έχουμε:

$$\begin{aligned} Q(5, \text{επάνω}) &= -1 + V(1) = -1 + (-14) = -15 \\ Q(5, \text{δεξιά}) &= -1 + V(6) = -1 + (-20) = -21 \\ Q(5, \text{κάτω}) &= -1 + V(9) = -1 + (-20) = -21 \\ Q(5, \text{αριστερά}) &= -1 + V(4) = -1 + (-14) = -15 \end{aligned}$$

s	π_0				π_1			
	επάνω	δεξιά	κάτω	αριστερά	επάνω	δεξιά	κάτω	αριστερά
1	0.25	0.25	0.25	0.25	0	0	0	1
2	0.25	0.25	0.25	0.25	0	0	0	1
3	0.25	0.25	0.25	0.25	0	0	0.5	0.5
4	0.25	0.25	0.25	0.25	1	0	0	0
5	0.25	0.25	0.25	0.25	0.5	0	0	0.5
6	0.25	0.25	0.25	0.25	0	0	0.5	0.5
7	0.25	0.25	0.25	0.25	0	0	1	0
8	0.25	0.25	0.25	0.25	1	0	0	0
9	0.25	0.25	0.25	0.25	0.5	0.5	0	0
10	0.25	0.25	0.25	0.25	0	0.5	0.5	0
11	0.25	0.25	0.25	0.25	0	0	1	0
12	0.25	0.25	0.25	0.25	0.5	0.5	0	0
13	0.25	0.25	0.25	0.25	0	1	0	0
14	0.25	0.25	0.25	0.25	0	1	0	0

Πίνακας 8.2: Πολιτική π_0 τυχαίας επιλογής ενέργειας σε κάθε κατάσταση και πολιτική π_1 έπειτα από ένα βήμα αξιολόγησης και βελτίωσης.

Την καλύτερη αναμενόμενη απολαβή την δίνουν δύο ενέργειες, η επάνω και η αριστερά. Επομένως, στην κατάσταση S , ο πράκτορας θα επιλέγει στοχαστικά με πιθανότητα 0.5 μεταξύ αυτών των δύο ενεργειών, ενώ η πιθανότητα επιλογής των υπόλοιπων δύο ενεργειών θα είναι 0.

Έπειτα από τη βελτίωση μιας πολιτικής π_0 , ή οποία θα μας δώσει μια νέα πολιτική π_1 , μπορούμε να αξιολογήσουμε τη νέα πολιτική, δηλαδή να υπολογίσουμε την v_{π_1} , και στη συνέχεια να τη βελτιώσουμε ώστε να μας δώσει μια νέα πολιτική π_2 . Μπορούμε δηλαδή επαναληπτικά να βρίσκουμε ολοένα και καλύτερες συναρτήσεις αξίας και πολιτικές, μέχρι να συγκλίνουμε στις βέλτιστες. Η επαναληπτική αυτή διαδικασία, η οποία όπως είπαμε και στην αρχή αυτής της ενότητας ονομάζεται επανάληψη πολιτικής, απεικονίζεται στο Σχήμα 8.7. Ο αλγόριθμος 5 υλοποιεί τη διαδικασία αυτή.

$$\pi_0 \xrightarrow{\text{Αξ.}} V^{\pi_0} \xrightarrow{\text{Βε.}} \pi_1 \xrightarrow{\text{Αξ.}} V^{\pi_1} \xrightarrow{\text{Βε.}} \pi_2 \xrightarrow{\text{Αξ.}} \dots \xrightarrow{\text{Βε.}} \pi^* \xrightarrow{\text{Αξ.}} V^{\pi^*}$$

Σχήμα 8.7: Επαναληπτική αξιολόγηση και βελτίωση πολιτικής.

8.2.2 Επανάληψη αξίας

Ένα μειονέκτημα της επανάληψης πολιτικής είναι ότι η αξιολόγηση της πολιτικής είναι η ίδια μια επαναληπτική διαδικασία, η οποία απαιτεί πολλά βήματα (θεωρητικά συγκλίνει στο άπειρο). Θα μπορούσαμε ίσως να σταματήσουμε τη διαδικασία αυτή νωρίτερα, δηλαδή να κάνουμε ορισμένα μόνο βήματα της αξιολόγησης, με αποτέλεσμα βέβαια να έχουμε μία προσέγγιση της v_{π} .

Μια ειδική περίπτωση αυτής της ιδέας είναι να κάνουμε μόνο ένα βήμα αξιολόγησης πολιτικής. Σε αυτήν την ειδική περίπτωση ο αλγόριθμος ονομάζεται **επανάληψη αξίας** (value iteration). Μπορεί να γραφεί σαν ένα απλό βήμα ενημέρωσης το οποίο συνδυάζει την βελτίωση πολιτικής και την αξιολόγηση πολιτικής σε ένα βήμα για κάθε $s \in S$:

$$\begin{aligned} V_{k+1}(s) &\leftarrow \max_a E\{r_{t+1} + \gamma V_k(s_{t+1}) \mid s_t = s, a_t = a\} \\ V_{k+1}(s) &\leftarrow \max_a \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V_k(s')] \end{aligned} \quad (8.15)$$

Algorithm 5 Ο αλγόριθμος επανάληψη πολιτικής.

```

1: Είσοδος: ΠΜΔΑ  $\langle \mathcal{S}, \mathcal{A}, P, R \rangle$ , ρυθμός έκπτωσης  $\gamma$ , μικρός θετικός αριθμός  $\theta$ 
2: Έξοδος: Βέλτιστη πολιτική  $\pi$ 
3:  $\pi \leftarrow$  τυχαία πολιτική
4: ΣταθερήΠολιτική  $\leftarrow false$ 
5: while  $Stable = false$  do
6:    $V \leftarrow$  Επαναληπτικη-Αξιολόγηση-Πολιτικής( $\langle \mathcal{S}, \mathcal{A}, P, R \rangle, \pi, \gamma, \theta$ )
7:    $Stable \leftarrow true$ 
8:   for  $s \in S$  do
9:      $b \leftarrow \pi(s)$ 
10:     $\pi(s) \leftarrow \arg \max_a \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V(s')]$ 
11:    if  $b \neq \pi(s)$  then
12:       $Stable = false$ 
13:    end if
14:   end for
15: end while
16: return  $\pi$ 

```

Η διαδικασία αυτή θα συγκλίνει στο άπειρο, αλλά πρακτικά μπορούμε να σταματήσουμε όταν η διαφορά της συνάρτησης αξίας από επανάληψη σε επανάληψη είναι μικρή. Ο αλγόριθμος 6 υλοποιεί τη διαδικασία αυτή.

Algorithm 6 Ο αλγόριθμος επανάληψη αξίας.

```

1: Είσοδος: ΠΜΔΑ  $\langle \mathcal{S}, \mathcal{A}, P, R \rangle$ , ρυθμός έκπτωσης  $\gamma$ , μικρός θετικός αριθμός  $\theta$ 
2: Έξοδος: Βέλτιστη πολιτική  $\pi$ 
3:
4: for  $s \in S$  do
5:    $V(s) \leftarrow$  τυχαίος πραγματικός αριθμός
6: end for
7: repeat
8:    $\Delta \leftarrow 0$ 
9:   for  $s \in S$  do
10:     $v \leftarrow V(s)$ 
11:     $V(s) \leftarrow \max_a \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V(s')]$ 
12:     $\Delta \leftarrow \max(\Delta, |v - V(s)|)$ 
13:   end for
14: until  $\Delta < \theta$ 
15: for  $s \in S$  do
16:    $\pi(s) \leftarrow \arg \max_a \sum_{s'} P_{ss'}^a [R_{ss'}^a + \gamma V(s')]$ 
17: end for
18: return  $\pi$ 

```

8.2.3 Ασύγχρονος δυναμικός προγραμματισμός

Ένα μεγάλο μειονέκτημα των μεθόδων δυναμικού προγραμματισμού είναι ότι απαιτούν υπολογισμούς επάνω σε όλο το χώρο των καταστάσεων. Αν ο χώρος είναι πολύ μεγάλος (π.χ. 10^{20} στο τάβλι) τότε το υπολογιστικό κόστος είναι πολύ μεγάλο. Οι αλγόριθμοι **ασύγχρονον δυναμικού προγραμματισμού** δεν ενημερώ-

νουν όλες τις καταστάσεις με κάποιο συστηματικό τρόπο, αλλά με μια οποιαδήποτε σειρά, με αποτέλεσμα κάποιες καταστάσεις να ενημερωθούν περισσότερες φορές από κάποιες άλλες. Με τη χρήση του ασύγχρονου δυναμικού προγραμματισμού μπορούμε να εστιάσουμε την ενημέρωση σε εκείνες τις καταστάσεις τις οποίες επισκέπτεται ο πράκτορας, και επομένως είναι πιο σχετικές με τη συμπεριφορά του. Έχουμε δηλαδή ανάμιξη του υπολογισμού της πολιτικής και της αλληλεπίδρασης του πράκτορα με το περιβάλλον.

8.3 Μέθοδοι Μόντε Κάρλο

Οι μέθοδοι Μόντε Κάρλο (MK) δεν απαιτούν τη γνώση της δυναμικής του περιβάλλοντος, $p(s'|s, a)$ και $r(s, a, s')$. Το μόνο που απαιτούν είναι εμπειρία, δηλαδή ακολουθίες από καταστάσεις, ενέργειες και ανταμοιβές, είτε από την πραγματική αλληλεπίδραση του πράκτορα με το περιβάλλον του είτε από μια προσομοίωση της. Στην τελευταία περίπτωση, δεν απαιτείται πλήρης γνώση της δυναμικής του περιβάλλοντος, αλλά μόνο ένα μοντέλο του περιβάλλοντος που μπορεί να παράξει ακολουθίες αλληλεπίδρασης.

Οι μέθοδοι MK στηρίζονται στον υπολογισμό της μέσης τιμής του κέρδους στο πλαίσιο ενός επεισόδιου. Επιπλέον, οι μέθοδοι MK ακολουθούν την φιλοσοφία της γενικευμένης επανάληψης πολιτικής. Λόγω του ότι η δυναμική του περιβάλλοντος δεν είναι διαθέσιμη, Θα επικεντρωθούμε στη συνάρτηση αξίας ενέργειας.

8.3.1 Αξιολόγηση πολιτικής

Δεδομένης μιας πολιτικής π , Θέλουμε να υπολογίσουμε τις αξίες των ενεργειών. Θα εκτελέσουμε έναν αριθμό από επεισόδια. Στο τέλος κάθε επεισόδιου, θα καταγράψουμε το κέρδος $G(s, a)$ έπειτα από κάθε ζεύγος κατάστασης και ενέργειας στο επεισόδιο αυτό. Για κάθε ζεύγος κατάστασης και ενέργειας στο πρόβλημα, θα υπολογίσουμε τον μέσο όρο των κερδών που έχουμε καταγράψει. Όσο περισσότερα τα δείγματα των κερδών ενός ζεύγους κατάστασης και ενέργειας, τόσο καλύτερη η προσέγγιση της αξίας του.

Έστω για παράδειγμα το ακόλουθο επεισόδιο: $s_1, a_1, -1, s_2, a_2, -1, s_1, a_1, -1, s_2, a_1, -1, s_4$. Στην μέθοδο MK πρώτης επίσκεψης, καταγράφουμε το κέρδος έπειτα από την πρώτη επίσκεψη σε κάθε ζεύγος κατάστασης και ενέργειας στο επεισόδιο, άρα έχουμε $G(s_1, a_1) = -4$, $G(s_2, a_2) = -3$ και $G(s_2, a_1) = -1$. Στην μέθοδο MK κάθε επίσκεψης, καταγράφουμε τα κέρδη έπειτα από την κάθε επίσκεψη σε κάθε ζεύγος κατάσταση και ενέργειας στο επεισόδιο, άρα έχουμε $G(s_1, a_1) = \{-4, -2\}$, $G(s_2, a_2) = -3$ και $G(s_2, a_1) = -1$.

Έστω για παράδειγμα τα ακόλουθα επεισόδια:

- $s_1, a_1, -1, s_2, a_2, -1, s_1, a_1, -1, s_2, a_1, -1, s_4$
- $s_2, a_1, -1, s_1, a_2, -1, s_1, a_1, -1, s_4$
- $s_1, a_2, -1, s_2, a_1, -1, s_2, a_2, -1, s_2, a_1, -1, s_4$

Θα υπολογίσουμε τις εκτιμήσεις των αξιών των ενεργειών από την MK πρώτης επίσκεψης. Ο πίνακας 8.3 δείχνει το κέρδος για κάθε ζεύγος κατάστασης και ενέργειας έπειτα από την πρώτη επίσκεψη σε κάθε επεισόδιο, καθώς και την τρέχουσα εκτίμηση της αξίας τους στο τέλος των τριών επεισοδίων.

Κέρδος				
Ζεύγος	1	2	3	M.0.
s_1, a_1	-4	-1		-2.5
s_1, a_2		-2	-4	-3
s_2, a_1	-1	-3	-3	-2.3
s_2, a_2	-3		-2	-2.5

Πίνακας 8.3: Κέρδος έπειτα από κάθε επεισόδιο για κάθε ζεύγος κατάστασης και ενέργειας στα επεισόδια.

Ένα πρόβλημα με την MK πρώτης επίσκεψης είναι πως αν η πολιτική που αξιολογούμε είναι ντετερμινιστική, τότε θα βλέπουμε το ίδιο επεισόδιο κάθε φορά. Έτσι όμως δεν θα παίρνουμε εκτιμήσεις της αξίας των υπολοίπων ενεργειών σε κάθε κατάσταση, και άρα ο αλγόριθμος δεν θα μαθαίνει. Αυτό είναι σημαντικό πρόβλημα, καθώς ακόμα και αν ξεκινήσουμε με μια τυχαία στοχαστική πολιτική, η βελτίωση της συνήθως περιλαμβάνει την επιλογή της βέλτιστης ενέργειας σε κάθε κατάσταση, οδηγώντας έτσι σε μια ντετερμινιστική πολιτική.

Ένας πρώτος τρόπος να αντιμετωπιστεί αυτό είναι οι **εξερευνητικές εκκινήσεις** (exploring starts). Τα επεισόδια θα ξεκινάνε σε ένα ζεύγος κατάστασης και ενέργειας, αντί για την κλασική περίπτωση που ξεκινάνε σε μια αρχική κατάσταση. Κάθε ζεύγος κατάστασης και ενέργειας θα έχει μη μηδενική πιθανότητα να επιλεγεί ως αρχικό ζεύγος. Αυτό εγγυάται πως ο πράκτορας θα επισκεφτεί όλα τα ζεύγη κατάστασης και ενέργειας άπειρες φορές με την προϋπόθεση της εκτέλεσης άπειρων επεισοδίων.

8.3.2 Επανάληψη πολιτικής

Ένας πρώτος αλγόριθμος MK επανάληψης πολιτικής μπορεί να οριστεί ως εξής: εκτελούμε αξιολόγηση πολιτικής μέσω MK πρώτης επίσκεψης με εξερευνητικές εκκινήσεις για ένα επεισόδιο. Βελτιώνουμε την πολιτική μέσω άπληστης επιλογής της καλύτερης ενέργειας σε κάθε κατάσταση: $\pi_{k+1}(s) = \arg \max_a q_{\pi_k}(s, a)$.

Ένα μειονέκτημα των εξερευνητικών εκκινήσεων είναι πως δεν μπορεί να χρησιμοποιηθεί όταν έχουμε έναν πράκτορα που αλληλεπιδρά με ένα πραγματικό περιβάλλον. Π.χ. αν έχουμε ένα ρομπότ που εξερευνά σωληνώσεις ύδρευσης για προβλήματα, δεν μπορούμε να το ξεκινήσουμε από οποιοδήποτε σημείο μέσα στο δίκτυο, παρά μόνο σε συγκεκριμένες αρχικές καταστάσεις.

Μια εναλλακτική λύση είναι να απαιτούμε οι πολιτικές να είναι στοχαστικές και να αποδίδουν μη μηδενική πιθανότητα σε κάθε ενέργεια κάθε κατάστασης. Τέτοιες πολιτικές ονομάζονται **μαλακές** (soft): $\pi(s, a) > 0 \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$. Μια ϵ -μαλακή πολιτική αποδίδει σε κάθε κατάσταση πιθανότητα ίση η μεγαλύτερη από μια μικρή πιθανότητα ϵ δια το πλήθος των διαθέσιμων ενέργειών στην κατάσταση αυτή: $\pi(s, a) > \frac{\epsilon}{|\mathcal{A}(s)|} \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$. Μια ϵ -άπληστη πολιτική αποδίδει σε κάθε κατάσταση πιθανότητα ίση με μικρή πιθανότητα ϵ δια το πλήθος των διαθέσιμων ενέργειών στην κατάσταση αυτή, ενώ στην καλύτερη ενέργεια της κατάστασης αποδίδει πιθανότητα $1 - \epsilon$.

Ένας αλγόριθμος επανάληψης πολιτικής που στηρίζεται σε αυτήν την λύση, είναι ο αλγόριθμος της ϵ -μαλακής MK πρώτης επίσκεψης που παρουσιάζεται παρακάτω.

8.3.3 Αυξητικοί υπολογισμοί

Στο τέλος κάθε επεισοδίου, υπολογίζουμε τον μέσο όρο των κερδών για τα ζεύγη καταστάσεων και ενέργειών που επισκεφτήκαμε στο επεισόδιο. Αυτό απαιτεί την διατήρηση στη μνήμη όλων των κερδών που έχουν παρατηρηθεί για όλα τα ζεύγη καταστάσεων και ενέργειών. Για την μείωση των αναγκών σε μνήμη μπορούμε να υπολογίζουμε τον μέσο όρο αυξητικά. Μια πρώτη λύση είναι να κρατάμε το άθροισμα και το πλήθος των κερδών για κάθε ζεύγος κατάστασης και ενέργειας. Το μόνο πρόβλημα εδώ είναι ότι το άθροισμα αυτό μπορεί να μεγαλώσει πολύ και να απαιτήσει περισσότερη μνήμη. Μια δεύτερη λύση είναι ο αυξητικός υπολογισμός του μέσου όρου.

Έστω AG_{k+1} ο μέσος όρος $k + 1$ κερδών. Ο μέσος όρος αυτός μπορεί να εκφραστεί αναδρομικά σε σχέση με τον προηγούμενο μέσο όρο και το τελευταίο κέρδος ως εξής:

$$AG_{k+1} = \frac{1}{k+1} \sum_{i=1}^{k+1} G_i = \frac{1}{k+1} \left(G_{k+1} + \sum_{i=1}^k G_i \right) = \frac{1}{k+1} (G_{k+1} + kAG_k) \Leftrightarrow$$

$$AG_{k+1} = \frac{1}{k+1} (G_{k+1} + kAG_k + AG_k - AG_k) = \frac{1}{k+1} (G_{k+1} + (k+1)AG_k - AG_k) \Leftrightarrow$$

Algorithm 7 Ο αλγόριθμος ε-μαλακή MK πρώτης επίσκεψης.

```

1: Είσοδος: μικρός θετικός αριθμός  $\epsilon$ 
2: Έξοδος: Βέλτιστη πολιτική  $\pi$ 
3:
4: for  $s \in \mathcal{S}, a \in \mathcal{A}(s)$  do
5:    $\pi(s) \leftarrow$  τυχαία ε-μαλακή πολιτική
6:    $Returns(s, a) \leftarrow \emptyset$ 
7:    $Q(s, a) \leftarrow$  τυχαίος πραγματικός αριθμός
8: end for
9: repeat
10:   Δημιουργία επεισοδίου ακολουθώντας την  $\pi$ 
11:   for  $s \in \mathcal{S}, a \in \mathcal{A}(s)$  do
12:      $G \leftarrow$  το κέρδος έπειτα από την πρώτη επίσκεψη στο  $s, a$ 
13:      $Returns(s, a) \leftarrow Returns(s, a) \wedge G$ 
14:      $Q(s, a) \leftarrow \text{average}(Returns(s, a))$ 
15:   end for
16:   for κάθε  $s$  στο επεισόδιο do
17:      $a_* = \arg \max_a Q(s, a)$ 
18:      $\pi(s, a) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}(s)|}, & \text{if } a = a_* \\ \frac{\epsilon}{|\mathcal{A}(s)|}, & \text{if } a \neq a_* \end{cases}$ 
19:   end for
20: until
21: return  $\pi$ 

```

$$AG_{k+1} = AG_k + \frac{1}{k+1} (G_{k+1} - AG_k)$$

Η ποσότητα $\frac{1}{k+1}$ καλείται μέγεθος βήματος και μικραίνει όσο αυξάνονται οι εκτιμήσεις των κερδών. Ωστόσο, σε μη στατικά προβλήματα είναι προτιμότερο το μέγεθος αυτό να είναι ένας σταθερός μικρός αριθμός α , $0 < \alpha \leq 1$:

$$AG_{k+1} = AG_k + \alpha (G_{k+1} - AG_k)$$

Μη στατικά είναι τα προβλήματα όπου οι αξίες των καταστάσεων και ενεργειών αλλάζουν στην πάροδο του χρόνου, όπως για παράδειγμα όταν ένα θήραμα προσαρμόζεται στην συμπεριφορά του θηρευτή του. Στη διαδικασία της επανάληψης πολιτικής, τα κέρδη για κάθε ζεύγος κατάστασης και ενέργειας, εξαρτώνται από την πολιτική του πράκτορα η οποία μεταβάλλεται (βελτιώνεται) στην πάροδο του χρόνου. Σε τέτοιες περιπτώσεις με το σταθερό μέγεθος βήματος δίνουμε περισσότερο βάρος κατά τον υπολογισμό του μέσου όρου στις πιο πρόσφατες ανταμοιβές και κέρδη, αντί να τα σταθμίζουμε ισοδύναμα:

$$AG_{k+1} = AG_k + \alpha(G_{k+1} - AG_k) = \alpha AG_{k+1} + (1-\alpha) AG_k = \alpha AG_{k+1} + (1-\alpha)\alpha G_k + (1-\alpha)^2 AG_{k-1} \Leftrightarrow$$

$$AG_{k+1} = \alpha G_{k+1} + (1-\alpha)\alpha G_k + (1-\alpha)^2 \alpha G_{k-1} + (1-\alpha)^3 AG_{k-2} = \dots$$

8.4 Μέθοδοι Χρονικών Διαφορών

Οι μέθοδοι χρονικών διαφορών (ΧΔ) αποτελούν μία από τις πιο κεντρικές και καινοτόμες ιδέες στην ενισχυτική μάθηση. Δεν απαιτούν μοντέλο του περιβάλλοντος, παρά μόνο εμπειρία (δείγματα αλληλεπιδρασης με το περιβάλλον), όπως και οι μέθοδοι MK. Σε αντίθεση με αυτές, οι μέθοδοι ΧΔ ενημερώνουν τις εκτιμήσεις

για την αξία μιας κατάστασης (ή ενός ζεύγους κατάστασης και ενέργειας) με βάση τις εκτιμήσεις των αξιών άλλων καταστάσεων (ή ζευγών κατάστασης και ενέργειας). Αυτή η διαφορά, επιτρέπει στις μεθόδους XΔ να μαθαίνουν κατά τη διάρκεια ενός επεισοδίου.

8.4.1 Αξιολόγηση πολιτικής

Στις μεθόδους MK η ενημέρωση των αξιών γίνεται ως εξής:

$$V(S_t) \leftarrow V(S_t) + \alpha[G_t - V(S_t)]$$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[G_t - Q(S_t, A_t)]$$

Στις μεθόδους XΔ ενός βήματος, που ονομάζονται XΔ(0) η ενημέρωση των αξιών γίνεται ως εξής:

$$V(S_t) \leftarrow V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]$$

Ο ψευδοκώδικας της μεθόδου XΔ(0) για αξιολόγηση πολιτικής με βάση τις αξίες των καταστάσεων παρουσιάζεται στον αλγόριθμο 8.

Algorithm 8 Ο αλγόριθμος XΔ(0).

```

1: Είσοδος: πολιτική  $\pi$  προς αξιολόγηση,  $\alpha \in (0, 1]$ ,  $\gamma \in [0, 1]$ 
2: Έξοδος: αξίες καταστάσεων  $V(s) \forall s \in \mathcal{S}$ 
3:
4: for  $s \in \mathcal{S}$  do
5:    $V(s) \leftarrow$  τυχαία τιμή (0 για τερματικές καταστάσεις)
6: end for
7: for κάθε επεισόδιο do
8:   Αρχικοποίηση  $S$ 
9:   for κάθε βήμα του επεισοδίου do
10:    Εκτέλεση ενέργειας  $A$  με βάση  $\pi(S)$ , παρατήρηση ανταμοιβής  $R$  και επόμενης κατάστασης  $S'$ 
11:     $V(S) \leftarrow V(S) + \alpha[R + \gamma V(S') - V(S)]$ 
12:     $S \leftarrow S'$ 
13:   end for
14: end for
15: return  $V$ 

```

Οι μέθοδοι ΔΠ, MK και XΔ μπορούν να συγκριθούν με βάση τον τρόπο που προσεγγίζουν την εξίσωση του Bellman:

$$v_\pi(s) = \mathbb{E}_\pi \{G_t | S_t = s\} = \mathbb{E}_\pi \{R_{t+1} + \gamma G_{t+1} | S_t = s\} = \mathbb{E}_\pi \{R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s\}$$

Οι μέθοδοι MK χρησιμοποιούν μια εκτίμηση του G_t , ενώ οι μέθοδοι ΔΠ και XΔ μια εκτίμηση του $R_{t+1} + \gamma v_\pi(S_{t+1})$. Στις μεθόδους ΔΠ η εκτίμηση αφορά μόνο στην ποσότητα $v_\pi(S_{t+1})$ ενώ στις μεθόδους XΔ και στην ποσότητα R_{t+1} . Οι μέθοδοι XΔ και MK εκτελούν δειγματοληπτικές ενημερώσεις που στηρίζονται μόνο σε μία επόμενη κατάσταση (ή ζεύγος κατάστασης και ενέργειας). Οι μέθοδοι ΔΠ εκτελούν αναμενόμενες ενημερώσεις, οι οποίες στηρίζονται σε όλες τις επόμενες καταστάσεις (ή ζεύγη κατάστασης και ενέργειας).

8.4.2 Επανάληψη πολιτικής

Ένας πρώτος αλγόριθμος επανάληψης πολιτικής, ο SARSA, ενημερώνει τις αξίες των ενεργειών έπειτα από κάθε μετάβαση από μία μη τερματική κατάσταση ως εξής:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]$$

Για τερματικές καταστάσεις S , ισχύει $Q(S, A) = 0$. Επιπλέον έπειτα από κάθε μετάβαση, βελτιώνουμε την τρέχουσα ε-μαλακή πολιτική κάνοντας την ε-άπληστη σε σχέση με τις Q . Ο αλγόριθμος αυτός συγκλίνει στην βέλτιστη πολιτική για άπειρες επισκέψεις σε κάθε ζεύγος κατάστασης και ενέργειας και με την προϋπόθεση πως η πολιτική θα γίνει άπληστη στο άπειρο. Αυτό μπορεί να επιτευχθεί θέτοντας $\epsilon = \frac{1}{t}$. Ο ψευδοκώδικας του SARSA παρουσιάζεται στον αλγόριθμο 9.

Algorithm 9 Ο αλγόριθμος SARSA.

```

1: Είσοδος:  $\alpha \in (0, 1]$ ,  $\gamma \in [0, 1]$ 
2: Έξοδος: βέλτιστη συνάρτηση αξίας ενέργειας  $Q \approx q_*$ 
3:
4: for  $s \in \mathcal{S}, a \in \mathcal{A}(s)$  do
5:    $Q(s, a) \leftarrow$  τυχαία τιμή (0 για τερματικές καταστάσεις)
6: end for
7: for κάθε επεισόδιο do
8:   Αρχικοποίηση  $S$ 
9:   Επιλογή ενέργειας  $A$  ε-άπληστα βάση των  $Q(S, A)$ 
10:  for κάθε βήμα του επεισοδίου do
11:    Εκτέλεση ενέργειας  $A$ , παρατήρηση ανταμοιβής  $R$  και επόμενης κατάστασης  $S'$ 
12:    Επιλογή ενέργειας  $A'$  ε-άπληστα βάση των  $Q(S', A')$ 
13:     $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma Q(S', A') - Q(S, A)]$ 
14:     $S \leftarrow S', A \leftarrow A'$ 
15:  end for
16: end for
17: return  $Q$ 

```

Ο αλγόριθμος SARSA είναι ένας αλγόριθμος **εντός πολιτικής** (on policy) με την έννοια ότι βελτιώνει την πολιτική, την οποία ακολουθεί κατά τη διάρκεια της αλληλεπίδρασης με το περιβάλλον. Σε αντίθεση, οι αλγόριθμοι **εκτός πολιτικής** (off policy) βελτιώνουν μια διαφορετική πολιτική από αυτήν που ακολουθούν. Αυτό έχει το πλεονέκτημα να μπορεί η πολιτική που βελτιώνεται να είναι άπληστη, όσο η πολιτική συμπεριφοράς εξακολουθεί να είναι ε-μαλακή ώστε να μπορεί ο αλγόριθμος να μαθαίνει. Στη συνέχεια θα δούμε ένα αλγόριθμο XΔ εκτός πολιτικής, τον Q-Learning, ο οποίος αποτελεί μια από τις μεγαλύτερες επιτυχίες της ενισχυτικής μάθησης. Ο αλγόριθμος αυτός συγκλίνει στη βέλτιστη πολιτική εφόσον ο πράκτορας επισκεφτεί κάθε ζεύγος κατάστασης και ενέργειας άπειρες φορές. Ο ψευδοκώδικας του Q-Learning παρουσιάζεται στον αλγόριθμο 10.

8.5 Ασκήσεις

- Έστω ότι $\gamma = 0.7$ και ότι παρατηρούμε την εξής ακολουθία από ανταμοιβές από την αρχή ως το τέλος ενός επεισοδίου: $R_1 = 2, R_2 = 1, R_3 = -1$ και $R_4 = 1$. Υπολογίστε τα κέρδη G_0, G_1, G_2, G_3 και G_4 .
- Έστω ότι $\gamma = 0.8$ και ότι την αρχική ανταμοιβή $R_1 = 1$ ακολουθεί μια άπειρη ακολουθία από ανταμοιβές ίσες με 2. Υπολογίστε τα κέρδη G_0 και G_1 .

Algorithm 10 Ο αλγόριθμος *Q-Learning*.

```

1: Είσοδος:  $\alpha \in (0, 1]$ ,  $\gamma \in [0, 1]$ 
2: Έξοδος: βέλτιστη συνάρτηση αξίας ενέργειας  $Q \approx q_*$ 
3:
4: for  $s \in \mathcal{S}, a \in \mathcal{A}(s)$  do
5:    $Q(s, a) \leftarrow$  τυχαία τιμή (0 για τερματικές καταστάσεις)
6: end for
7: for κάθε επεισόδιο do
8:   Αρχικοποίηση  $S$ 
9:   for κάθε βήμα του επεισοδίου do
10:    Επιλογή ενέργειας  $A$  ε-άπληστα βάση των  $Q(S, A)$ 
11:    Εκτέλεση ενέργειας  $A$ , παρατήρηση ανταμοιβής  $R$  και επόμενης κατάστασης  $S'$ 
12:     $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_{A'} Q(S', A') - Q(S, A)]$ 
13:     $S \leftarrow S'$ 
14:   end for
15: end for
16: return  $Q$ 

```

3. Το Σχήμα 8.5 δείχνει τον κόσμο πλέγματος που παρουσιάστηκε στην ενότητα 8.2.1 και το Σχήμα 8.6(στ) την τιμή v_π για αυτόν τον κόσμο, όπου π η πολιτική τυχαίας επιλογής ενέργειας. Αν προσθέσουμε κάτω από την κατάσταση 13 μια κατάσταση 16, για την οποία οι ενέργειες αριστερά, επάνω, δεξιά και κάτω οδηγούν στις καταστάσεις 12, 13, 14 και 16 αντίστοιχα, ποια είναι η τιμή $v_\pi(16)$ αν οι υπόλοιπες μεταβάσεις μείνουν ως έχουν; Αν τώρα θεωρήσουμε ότι η ενέργεια κάτω στην κατάσταση 13 έχει ως αποτέλεσμα την μετάβαση στην κατάσταση 16, ποια είναι η τιμή $v_\pi(16)$;
4. Δώστε την εξίσωση ενημέρωσης για τη διαδοχική προσέγγιση της συνάρτησης αξίας ενέργειας Q^π .
5. Δώστε τις βέλτιστες εξισώσεις του Bellman για το ρομπότ ανακύκλωσης.
6. Στον κόσμο πλέγματος που παρουσιάστηκε στην ενότητα 8.2.1, πόσες επαναλήψεις του αλγορίθμου επανάληψη πολιτικής χρειάζονται για να καταλήξουμε σε βέλτιστη πολιτική, ξεκινώντας από μια πολιτική που επιλέγει ενέργειες τυχαία σε κάθε κατάσταση;
7. Έστω ο παρακάτω κόσμος πλέγματος, στον οποίο ο πράκτορας εκκινεί από την κατάσταση A και έχει να επιλέξει μεταξύ δύο ενεργειών, επάνω και κάτω, οι οποίες των οδηγούν αιτιοκρατικά στις καταστάσεις B και Γ αντίστοιχα λαμβάνοντας ανταμοιβή +50 και -50 αντίστοιχα. Από τις B και Γ έχει στη διάθεση τον μόνο μία ενέργεια, δεξιά, η οποία τον οδηγεί αιτιοκρατικά ένα βήμα δεξιά λαμβάνοντας ανταμοιβή -1 και +1 αντίστοιχα. Οι καταστάσεις K1.100 και K2.100 είναι τερματικές. Δώστε μια ανίσωση ως προς την τιμή του ρυθμού έκπτωσης γ , σύμφωνα με την οποία ο πράκτορας θα πρέπει αρχικά να επιλέξει την ενέργεια επάνω έναντι της κάτω.

B	K1.1	...	K1.100
A			
B	K2.1	...	K2.100

8. Έστω ένας πράκτορας, ο οποίος δραστηριοποιείται σε έναν κόσμο με δύο καταστάσεις K1 και K2. Σε κάθε κατάσταση έχει δύο ενέργειες στη διάθεση του, τις E1 και E2. Όταν επιλέγει την E1 τότε με πιθανότητα 0,8 παραμένει στην κατάσταση που βρίσκεται και παίρνει ανταμοιβή 0 και με πιθανότητα 0,2

μεταβαίνει στην άλλη κατάσταση και πάρνει ανταμοιβή +5. Όταν επιλέγει την E2, τότε με πιθανότητα 1 μεταβαίνει στην άλλη κατάσταση και πάρνει ανταμοιβή +1. (α) Σχεδιάστε το γράφο μετάβασης της Μαρκοβιανής διαδικασίας απόφασης του παραπάνω πράκτορα. (β) Θεωρώντας ότι $\gamma=0.9$ και ότι η πολιτική του πράκτορα είναι: $\pi(K1,E1)=0.2, \pi(K2,E2)=0.6$, εφαρμόστε μια επανάληψη του αλγορίθμου επαναληπτική αξιολόγηση πολιτικής, ξεκινώντας από αρχικές τιμές $V(s)=0$ για κάθε s. (γ) Βελτιώστε την πολιτική του πράκτορα βάσει των τιμών της V που υπολογίσατε στο προηγούμενο βήμα.

9. Έστω μια Μαρκοβιανή διαδικασία απόφασης με τρεις καταστάσεις K1, K2, K3. Η κατάσταση K3 είναι τερματική. Στις άλλες δύο καταστάσεις διατίθενται δύο ενέργειες E1 και E2. Στην κατάσταση K1, όταν ο πράκτορας επιλέγει την ενέργεια E1, τότε με πιθανότητα 0,8 μεταβαίνει στην κατάσταση K2 και πάρνει ανταμοιβή -2, ενώ με πιθανότητα 0,2 παραμένει στην K1 και πάρνει ανταμοιβή -1. Όταν επιλέγει την E2, τότε με με πιθανότητα 0,1 μεταβαίνει στην κατάσταση K3 και πάρνει ανταμοιβή 0, ενώ με πιθανότητα 0,9 παραμένει στην K1 και πάρνει ανταμοιβή -1. Στην κατάσταση K2, όταν ο πράκτορας επιλέγει την ενέργεια E1, τότε με πιθανότητα 0,8 μεταβαίνει στην κατάσταση K1 και πάρνει ανταμοιβή -1, ενώ με πιθανότητα 0,2 παραμένει στην K2 και πάρνει ανταμοιβή -2. Όταν επιλέγει την E2, τότε με με πιθανότητα 0,1 μεταβαίνει στην κατάσταση K3 και πάρνει ανταμοιβή 0, ενώ με πιθανότητα 0,9 παραμένει στην K2 και πάρνει ανταμοιβή -2. (α) Σχεδιάστε το γράφο μετάβασης της παραπάνω Μαρκοβιανής διαδικασίας απόφασης. (β) Θεωρώντας ότι $\gamma=1$ εφαρμόστε μια επανάληψη του αλγορίθμου επανάληψη αξίας, ξεκινώντας από αρχικές τιμές $V(s)=0$ για κάθε s.

10. Έστω τα ακόλουθα δύο επεισόδια:

- $s_1, a_1, -1, s_2, a_2, -1, s_1, a_1, -1, s_2, a_1, -1, s_4$
- $s_2, a_1, -1, s_1, a_2, -1, s_1, a_1, -1, s_4$

Εφαρμόστε τον αλγόριθμο ε-μαλακή MK πρώτης επίσκεψης με αυξητικούς υπολογισμούς και δείξτε την πολιτική στο τέλος αυτών των δύο επεισοδίων, με μέγεθος βήματος $\alpha = 0.1, \gamma = 1$ και $\epsilon = 0.1$.

11. Έστω τα ακόλουθα δύο επεισόδια:

- $s_1, a_1, -1, s_2, a_2, -1, s_1, a_1, -1, s_2, a_1, -1, s_4$
- $s_2, a_1, -1, s_1, a_2, -1, s_1, a_1, -1, s_4$

Εφαρμόστε τους αλγορίθμους SARSA και Q-Learning και δείξτε τις αξίες των ενεργειών στο τέλος των δύο αυτών επεισοδίων, ($\alpha = 0.1, \gamma = 1, \epsilon = 0.1$).

12. Έστω ένας πράκτορας που δραστηριοποιείται σε ένα περιβάλλον με τρεις καταστάσεις s_1, s_2, s_3 , όπου η κατάσταση s_3 είναι τερματική. Στις καταστάσεις s_1 και s_2 υπάρχουν διαθέσιμες δύο ενέργειες: a_1 και a_2 . Η δυναμική του περιβάλλοντος έχει ως εξής: Στην κατάσταση s_1 αν επιλέξει a_1 με πιθανότητα 0.7 μεταβαίνει στην s_2 και λαμβάνει ανταμοιβή -1 και με πιθανότητα 0.3 παραμένει στην s_1 και λαμβάνει ανταμοιβή -2. Αν επιλέξει a_2 μεταβαίνει απευθείας στην s_3 με ανταμοιβή +5. Στην κατάσταση s_2 αν επιλέξει a_1 μεταβαίνει στην s_3 με ανταμοιβή +2, ενώ αν επιλέξει a_2 παραμένει στην s_2 με ανταμοιβή -1.

- (α) Να κατασκευάσετε τον γράφο μετάβασης της παραπάνω ΠΜΔΑ, σημειώνοντας τις πιθανότητες μετάβασης και τις αντίστοιχες ανταμοιβές.
- (β) Έστω πολιτική π , η οποία επιλέγει πάντα a_1 σε κάθε μη-τερματική κατάσταση. Υπολογίστε τις τιμές $V^\pi(s_1)$ και $V^\pi(s_2)$, με συντελεστή έκπτωσης $\gamma = 0.9$ και $V(s_3) = 0$.
- (γ) Βάσει των τιμών που υπολογίστηκαν στο (β), ποια ενέργεια είναι προτιμότερη σε κάθε κατάσταση; Δηλαδή, ποια είναι η βελτιωμένη πολιτική;

Μέρος Ι

ΠΑΡΑΡΤΗΜΑΤΑ

ΠΑΡΑΡΤΗΜΑ Α

ΑΠΑΝΤΗΣΕΙΣ ΕΡΩΤΗΣΕΩΝ - ΛΥΣΕΙΣ ΑΣΚΗΣΕΩΝ

A.1 Κεφάλαιο 2

1. Από την εκφώνηση καταλαβαίνουμε πως εκπαιδεύουμε ένα γραμμικό μοντέλο παλινδρόμησης της μορφής $h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$. Μας δίνεται ο ρυθμός μάθησης $\eta = 0.1$ και οι τρέχουσες τιμές των παραμέτρων $\theta_0 = 1, \theta_1 = -1$ και $\theta_2 = 2$.

Θεωρώντας τεχνητή μεταβλητή εισόδου x_0 , η οποία έχει πάντα τιμή 1, οι ενημερώσεις των παραμέτρων στον στοχαστικό αλγόριθμο επικλινής καθόδου για ένα παράδειγμα (x, y) γίνονται ως εξής: $\theta_j := \theta_j - \eta(h_{\theta}(x) - y)x_j$.

Δεδομένου του παραδείγματος $(x, y) = ([0.5, 0.25], 3)$, υπολογίζουμε τις ενημερώσεις των παραμέτρων ως εξής:

$$\theta_0 = 1 - (0.1)(-2)(1) = 1.2$$

$$\theta_1 = -1 - (0.1)(-2)(0.5) = -0.9$$

$$\theta_2 = 2 - (0.1)(-2)(0.25) = 2.05$$

Άρα, οι νέες τιμές των παραμέτρων είναι $\theta_0 = 1.2, \theta_1 = -0.9$ και $\theta_2 = 2.05$.

2. Το κόστος ενός γραμμικού μοντέλου δίνεται από τη συνάρτηση:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x_i) - y_i)^2$$

Μας δίνεται γραμμικό μοντέλο $h_{\theta}(x) = x$.

Για τα παρακάτω δεδομένα:

x_1	y
0	1
1	2
-1	0
0	1

Υπολογίζουμε το κόστος ως εξής:

$$J(\theta) = \frac{1}{2 \cdot 4} [(0-1)^2 + (1-2)^2 + (-1-0)^2 + (0-1)^2]$$

$$J(\theta) = \frac{1}{8} [1+1+1+1] = \frac{1}{8} \cdot 4 = 0.5$$

Άρα, το κόστος είναι $J(\theta) = 0.5$.

3. Στην τεχνική της κλιμάκωσης, ο μετασχηματισμός δίνεται από τον τύπο:

$$x'_j = \frac{x_j - \min(x_j)}{\max(x_j) - \min(x_j)}$$

Για τη μεταβλητή x_1 και τα δεδομένα:

$$\{10, 20, 30, 40\}$$

Έχουμε $\min(x_1) = 10$ και $\max(x_1) = 40$

Η μετασχηματισμένη τιμή του x_1 στο 4ο παράδειγμα είναι:

$$x'_1 = \frac{40-10}{40-10} = \frac{30}{30} = 1$$

Άρα, η τιμή του x_1 στο 4ο παράδειγμα μετά την κλιμάκωση 0-1 είναι:

$$x'_1 = 1$$

4. Μία εναντίων όλων

- Απαιτούνται 5 μοντέλα, ένα για κάθε κλάση.
- Σε κάθε μοντέλο, τα παραδείγματα της θετικής κλάσης παραμένουν ως έχουν, ενώ τα παραδείγματα των άλλων 4 κλάσεων θεωρούνται ως αρνητικά.
- Επομένως, για κάθε μοντέλο, το πλήθος των παραδειγμάτων είναι το συνολικό πλήθος:

$$m = m_1 + m_2 + m_3 + m_4 + m_5 = 10 + 20 + 30 + 40 + 50 = 150$$

- Επειδή εκπαιδεύουμε 5 μοντέλα, το συνολικό άθροισμα παραδειγμάτων είναι:

$$5 \times 150 = 750$$

Μία εναντίων μίας

- Απαιτούνται μοντέλα για κάθε ζεύγος κλάσεων. Ο αριθμός των ζευγών είναι:

$$\binom{5}{2} = \frac{5 \cdot 4}{2} = 10$$

- Για κάθε ζεύγος κλάσεων (i, j) , χρησιμοποιούνται μόνο τα παραδείγματα των δύο αυτών κλάσεων. Το πλήθος παραδειγμάτων για κάθε μοντέλο είναι:

- $(1, 2) \rightarrow n_1 + n_2 = 10 + 20 = 30$
- $(1, 3) \rightarrow n_1 + n_3 = 10 + 30 = 40$
- $(1, 4) \rightarrow n_1 + n_4 = 10 + 40 = 50$
- $(1, 5) \rightarrow n_1 + n_5 = 10 + 50 = 60$
- $(2, 3) \rightarrow n_2 + n_3 = 20 + 30 = 50$
- $(2, 4) \rightarrow n_2 + n_4 = 20 + 40 = 60$
- $(2, 5) \rightarrow n_2 + n_5 = 20 + 50 = 70$
- $(3, 4) \rightarrow n_3 + n_4 = 30 + 40 = 70$
- $(3, 5) \rightarrow n_3 + n_5 = 30 + 50 = 80$
- $(4, 5) \rightarrow n_4 + n_5 = 40 + 50 = 90$

- Το συνολικό άθροισμα παραδειγμάτων για όλα τα μοντέλα είναι:

$$30 + 40 + 50 + 60 + 50 + 60 + 70 + 70 + 70 + 80 + 90 = 600$$

Στο ίδιο αποτέλεσμα μπορούμε να φτάσουμε και αν σκεφτούμε πως το κάθε ένα από τα 150 παραδείγματα εκπαιδεύνται θα συμμετάσχει σε 4 μοντέλα, τα οποία αφορούν τα αντίστοιχα ζεύγη της κλάσης στην οποία ανήκει με όλες τις υπόλοιπες κλάσεις. 150×4 μας κάνει επίσης 600.

Σύγκριση μεθόδων:

- Μία εναντίων όλων: 5 μοντέλα, συνολικά 750 παραδείγματα.
- Μία εναντίων μίας: 10 μοντέλα, συνολικά 600 παραδείγματα.
- Άρα, η μέθοδος μία εναντίων όλων περιλαμβάνει το μεγαλύτερο άθροισμα παραδειγμάτων.

A.2 Κεφάλαιο 3

1. Η εντροπία ορίζεται ως:

$$H(S) = - \sum p_i \log_2 p_i \quad (\text{A.1})$$

Από το σύνολο δεδομένων, έχουμε:

- $y = 0$ σε 3 περιπτώσεις
- $y = 1$ σε 2 περιπτώσεις

Οι πιθανότητες είναι:

$$p(0) = \frac{3}{5}, \quad p(1) = \frac{2}{5} \quad (\text{A.2})$$

Οπότε η εντροπία του συνόλου είναι:

$$H(S) = - \left(\frac{3}{5} \log_2 \frac{3}{5} + \frac{2}{5} \log_2 \frac{2}{5} \right) \quad (\text{A.3})$$

Υπολογίζοντας:

$$H(S) \approx -(0.6 \times (-0.737) + 0.4 \times (-1.322)) = 0.971 \quad (\text{A.4})$$

Για το x_1 :

Οι πιθανότητες εμφάνισης είναι:

$$P(a) = \frac{2}{5}, \quad P(b) = \frac{2}{5}, \quad P(c) = \frac{1}{5} \quad (\text{A.5})$$

Υπολογίζουμε τις επιμέρους εντροπίες:

$$\begin{aligned} H(S_a) &= 1 \\ H(S_b) &= 1 \\ H(S_c) &= 0 \end{aligned}$$

Η σταθμισμένη εντροπία για το x_1 είναι:

$$\begin{aligned} H(S_{x_1}) &= \left(\frac{2}{5} \times 1\right) + \left(\frac{2}{5} \times 1\right) + \left(\frac{1}{5} \times 0\right) \\ &= 0.8 \end{aligned}$$

Για το x_2 :

Οι πιθανότητες εμφάνισης είναι:

$$P(a) = \frac{1}{5}, \quad P(b) = \frac{3}{5}, \quad P(c) = \frac{1}{5} \quad (\text{A.6})$$

Υπολογίζουμε τις επιμέρους εντροπίες:

$$\begin{aligned} H(S_a) &= 0 \\ H(S_b) &= 0 \\ H(S_c) &= 0 \end{aligned}$$

Η σταθμισμένη εντροπία για το x_2 είναι:

$$H(S_{x_2}) = \left(\frac{1}{5} \times 0\right) + \left(\frac{3}{5} \times 0\right) + \left(\frac{1}{5} \times 0\right) = 0$$

Για το x_3 :

$$\begin{aligned} H(S_a) &= 1 \\ H(S_b) &= 1 \\ H(S_c) &= 0 \end{aligned}$$

Άρα η σταθμισμένη εντροπία για το x_3 είναι:

$$H(S_{x_3}) = \frac{2}{5} \cdot 1 + \frac{2}{5} \cdot 1 + \frac{1}{5} \cdot 0 = 0.8$$

Για το x_4 :

$$\begin{aligned} H(S_a) &= 0.918 \\ H(S_b) &= 1 \\ H(S_c) &= 0 \end{aligned}$$

Άρα η σταθμισμένη εντροπία για το x_4 είναι:

$$H(S_{x_4}) = \frac{3}{5} \cdot 0.918 + \frac{1}{5} \cdot 1 + \frac{1}{5} \cdot 0 = 0.7508$$

Για το x_5 :

$$\begin{aligned} H(S_a) &= 0 \\ H(S_b) &= 1 \\ H(S_c) &= 1 \end{aligned}$$

Άρα η σταθμισμένη εντροπία για το x_5 είναι:

$$H(S_{x_5}) = \frac{1}{5} \cdot 0 + \frac{2}{5} \cdot 1 + \frac{2}{5} \cdot 1 = 0.8$$

Το χαρακτηριστικό x_2 έχει τη μικρότερη σταθμισμένη εντροπία ($H(S_{x_2}) = 0$) και κατά συνέπεια το μεγαλύτερο IG. Επομένως το χαρακτηριστικό x_2 επιλέγεται ως η ρίζα του δέντρου από τον αλγόριθμο ID3.

2. Η εντροπία ορίζεται ως:

$$H(S) = - \sum p_i \log_2 p_i \quad (\text{A.7})$$

Από το σύνολο δεδομένων, έχουμε:

- $y = 0$ σε 4 περιπτώσεις
- $y = 1$ σε 2 περιπτώσεις

Οι πιθανότητες είναι:

$$p(0) = \frac{4}{6}, \quad p(1) = \frac{2}{6} \quad (\text{A.8})$$

Οπότε η εντροπία του συνόλου είναι:

$$H(S) = - \left(\frac{4}{6} \log_2 \frac{4}{6} + \frac{2}{6} \log_2 \frac{2}{6} \right) \quad (\text{A.9})$$

Υπολογίζοντας:

$$H(S) \approx 0.918 \quad (\text{A.10})$$

Γειτονικά παραδείγματα με διαφορετική τιμή στην κλάση είναι το πρώτο και το δεύτερο, το δεύτερο και το τρίτο καθώς και το πέμπτο με το έκτο.

Επομένως τα υποψήφια κατώφλια είναι:

$$s_1 = \frac{2+4}{2} = 3, \quad s_2 = \frac{4+6}{2} = 5, \quad s_3 = \frac{10+12}{2} = 11 \quad (\text{A.11})$$

Για κάθε s_i , υπολογίζουμε την εντροπία των δύο υποσυνόλων:

$$\left(\frac{|S_{\text{left}}|}{|S|} H(S_{\text{left}}) + \frac{|S_{\text{right}}|}{|S|} H(S_{\text{right}}) \right) \quad (\text{A.12})$$

Υπολογίζουμε τις εντροπίες για κάθε πιθανή τιμή διαχωρισμού:

Για $s_1 = 3$:

$$\begin{aligned} S_{\text{left}} &= \{2\} \Rightarrow H(S_{\text{left}}) = 0 \\ S_{\text{right}} &= \{4, 6, 8, 10, 12\} \\ H(S_{\text{right}}) &= -\left(\frac{3}{5} \log_2 \frac{3}{5} + \frac{2}{5} \log_2 \frac{2}{5}\right) \approx 0.971 \end{aligned}$$

Η εντροπία των δύο υποσυνόλων για το s_1 είναι:

$$\left(\frac{|S_{\text{left}}|}{|S|} H(S_{\text{left}}) + \frac{|S_{\text{right}}|}{|S|} H(S_{\text{right}}) \right) = \left(\frac{1}{6} \times 0 + \frac{5}{6} \times 0.971 \right) \approx 0.809 \quad (\text{A.13})$$

Για $s_2 = 5$:

Τα δεδομένα χωρίζονται σε δύο υποσύνολα:

- S_{left} (όπου $x_1 \leq 5$): Τα σημεία $\{2, 4\}$ με ετικέτες $y = 0$ και $y = 1$.
- S_{right} (όπου $x_1 > 5$): Τα σημεία $\{6, 8, 10, 12\}$ με ετικέτες $y = 0, 0, 0, 1$.

$$H(S_{\text{left}}) = -\left(\frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2}\right) = 1$$

$$H(S_{\text{right}}) = -\left(\frac{3}{4} \log_2 \frac{3}{4} + \frac{1}{4} \log_2 \frac{1}{4}\right)$$

Υπολογίζοντας:

$$H(S_{\text{right}}) \approx 0.811$$

Η εντροπία των δύο υποσυνόλων για το s_2 είναι:

$$\left(\frac{|S_{\text{left}}|}{|S|} H(S_{\text{left}}) + \frac{|S_{\text{right}}|}{|S|} H(S_{\text{right}}) \right) = \left(\frac{2}{6} \times 1 + \frac{4}{6} \times 0.811 \right) = (0.333 + 0.541) = 0.874$$

Για $s_3 = 11$:

Τα δεδομένα χωρίζονται σε δύο υποσύνολα:

- S_{left} (όπου $x_1 \leq 11$): Τα σημεία $\{2, 4, 6, 8, 10\}$ με ετικέτες $y = 0, 1, 0, 0, 0$.
- S_{right} (όπου $x_1 > 11$): Το σημείο $\{12\}$ με ετικέτα $y = 1$.

Το υποσύνολο S_{left} περιέχει τα σημεία $\{2, 4, 6, 8, 10\}$ με τις ετικέτες $y = 0, 1, 0, 0, 0$.

- Η πιθανότητα για $y = 0$ είναι $p(0) = \frac{4}{5}$ - Η πιθανότητα για $y = 1$ είναι $p(1) = \frac{1}{5}$

Η εντροπία για το S_{left} είναι:

$$H(S_{\text{left}}) = - \left(\frac{4}{5} \log_2 \frac{4}{5} + \frac{1}{5} \log_2 \frac{1}{5} \right) \approx 0.722$$

Το υποσύνολο S_{right} περιέχει το σημείο $\{12\}$ με την ετικέτα $y = 1$.

- Η πιθανότητα για $y = 1$ είναι $p(1) = 1$ - Η εντροπία για το S_{right} είναι:

$$H(S_{\text{right}}) = 0$$

Η εντροπία των δύο υποσυνόλων για το s_3 είναι:

$$\left(\frac{|S_{\text{left}}|}{|S|} H(S_{\text{left}}) + \frac{|S_{\text{right}}|}{|S|} H(S_{\text{right}}) \right) = \left(\frac{5}{6} \times 0.722 + \frac{1}{6} \times 0 \right) = 0.602$$

Με βάση τους υπολογισμούς που έχουμε κάνει για την εντροπία των δύο υποσυνόλων, το καλύτερο σημείο διαχωρισμού είναι το $s_3 = 11$, καθώς έχει τη μικρότερη σταθμισμένη εντροπία.

Το δέντρο θα χωρίσει τα δεδομένα σε δύο υποσύνολα:

- Ένα υποσύνολο με τα σημεία $\{2, 4, 6, 8, 10\}$, για τα οποία $x_1 \leq 11$.
- Ένα υποσύνολο με το σημείο $\{12\}$, για το οποίο $x_1 > 11$.

3. Η αρχική εντροπία του στόχου y είναι:

$$H(S) = - \sum p_i \log_2 p_i \quad (\text{A.14})$$

Από το σύνολο δεδομένων, έχουμε:

- $y = 1$ σε 3 περιπτώσεις
- $y = 0$ σε 3 περιπτώσεις

Οπότε,

$$p(1) = \frac{3}{6}, \quad p(0) = \frac{3}{6} \quad (\text{A.15})$$

Άρα η εντροπία του συνόλου είναι:

$$H(S) = -\left(\frac{3}{6} \log_2 \frac{3}{6} + \frac{3}{6} \log_2 \frac{3}{6}\right) \quad (\text{A.16})$$

Υπολογίζοντας,

$$H(S) = -(0.5 \times (-1) + 0.5 \times (-1)) = 1 \quad (\text{A.17})$$

Για το x_1

Διαχωρίζουμε το σύνολο δεδομένων με βάση το x_1 :

- Για $x_1 = 1$: $\{(1, 1, 1), (1, 1, 1), (1, 0, 0)\}$, όπου $p(1) = \frac{2}{3}, p(0) = \frac{1}{3}$
- Για $x_1 = 0$: $\{(0, 0, 1), (0, 1, 0), (0, 1, 0)\}$, όπου $p(1) = \frac{1}{3}, p(0) = \frac{2}{3}$

Η εντροπία για κάθε ομάδα είναι:

$$H(S_{x_1=1}) = -\left(\frac{2}{3} \log_2 \frac{2}{3} + \frac{1}{3} \log_2 \frac{1}{3}\right) \approx 0.918 \quad (\text{A.18})$$

$$H(S_{x_1=0}) = -\left(\frac{1}{3} \log_2 \frac{1}{3} + \frac{2}{3} \log_2 \frac{2}{3}\right) \approx 0.918 \quad (\text{A.19})$$

Άρα η σταθμισμένη εντροπία για το x_1 είναι:

$$\frac{3}{6} \times 0.918 + \frac{3}{6} \times 0.918 \quad (\text{A.20})$$

Για το x_2

Διαχωρίζουμε το σύνολο δεδομένων με βάση το x_2 :

- Για $x_2 = 1$: $\{(1, 1, 1), (1, 1, 1), (0, 1, 0), (0, 1, 0)\}$, όπου $p(1) = \frac{2}{4}, p(0) = \frac{2}{4}$
- Για $x_2 = 0$: $\{(1, 0, 0), (0, 0, 1)\}$, όπου $p(1) = \frac{1}{2}, p(0) = \frac{1}{2}$

Η εντροπία για κάθε ομάδα είναι:

$$H(S_{x_2=1}) = -\left(\frac{2}{4} \log_2 \frac{2}{4} + \frac{2}{4} \log_2 \frac{2}{4}\right) = 1 \quad (\text{A.21})$$

$$H(S_{x_2=0}) = -\left(\frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2}\right) = 1 \quad (\text{A.22})$$

Άρα η σταθμισμένη εντροπία για το x_2 είναι:

$$\frac{4}{6} \times 1 + \frac{2}{6} \times 1 \quad (\text{A.23})$$

Επιλέγουμε το χαρακτηριστικό x_1 καθώς έχει τη μικρότερη σταθμισμένη εντροπία. Το Information Gain (IG) του x_1 είναι:

$$IG(x_1) = 0.918 = 0.082 \quad (\text{A.24})$$

4. Είναι γνωστό ότι:

- Η εντροπία των παραδειγμάτων με $x_2 = 0$ είναι 0, άρα όλα έχουν την ίδια τιμή εξόδου y .
- Η εντροπία των παραδειγμάτων με $x_1 = 0$ είναι 1, που σημαίνει ότι υπάρχουν και τα δύο δυνατά αποτελέσματα για το y .
- Ο αλγόριθμος ID3 επιλέγει το x_2 ως καταληλότερη μεταβλητή για τον έλεγχο στη ρίζα, άρα το πληροφοριακό κέρδος από το x_2 είναι υψηλότερο από το x_1 .

Από την πρώτη πληροφορία ($H(S_{x_2=0}) = 0$), όλα τα δείγματα όπου $x_2 = 0$ έχουν την ίδια τιμή y . Το τρίτο δείγμα έχει $y = 0$, άρα και το πρώτο δείγμα πρέπει να έχει $y = 0$.

$$y^{(1)} = 0 \quad (\text{A.25})$$

Πα τα παραδείγματα όπου $x_1 = 0$, γνωρίζουμε ότι η εντροπία είναι 1, πράγμα που σημαίνει ότι υπάρχουν τόσο $y = 0$ όσο και $y = 1$. Ένα από αυτά ($x_1 = 0, x_2 = 0$) είναι ήδη $y = 0$, επομένως το άλλο ($x_1 = 0, x_2 = 1$) πρέπει να είναι $y = 1$.

$$y^{(4)} = 1 \quad (\text{A.26})$$

Αν το $y^{(2)} = 0$ τότε η σταθμισμένη εντροπία για το x_1 είναι $\frac{2}{4} \times 0 + \frac{2}{4} \times 1 = 0.5$ και για το x_2 είναι $\frac{2}{4} \times 0 + \frac{2}{4} \times 1 = 0.5$. Αν το $y^{(2)} = 1$ τότε η σταθμισμένη εντροπία για το x_1 είναι $\frac{2}{4} \times 1 + \frac{2}{4} \times 0 = 1$ και για το x_2 είναι $\frac{2}{4} \times 0 + \frac{2}{4} \times 0 = 0$.

Τέλος, γνωρίζουμε ότι ο ID3 επέλεξε το x_2 ως ρίζα. Αυτό σημαίνει ότι η εντροπία πριν από τη διάσπαση ήταν μεγαλύτερη απ' ότι μετά τη διάσπαση. Επομένως, το $y^{(2)} = 1$. Άρα ο τελικός συμπληρωμένος πίνακας θα είναι:

x_1	x_2	y
1	0	0
1	1	1
0	0	0
0	1	1

5. α) Στο δεδομένο σύνολο έχουμε:

$$p_{N\alpha} = \frac{3}{8}$$

$$p_{O\chi} = \frac{5}{8}$$

Άρα:

$$H(S) = -\left(\frac{3}{8} \log_2\left(\frac{3}{8}\right) + \frac{5}{8} \log_2\left(\frac{5}{8}\right)\right) = 0.954$$

β) Για το χαρακτηριστικό Ήλικία:

Γειτονικά παραδείγματα με διαφορετική τιμή στην κλάση είναι το τέταρτο με το πέμπτο, το πέμπτο με το έκτο και το έκτο με το έβδομο.

Επομένως τα υποψήφια κατώφλια είναι:

$$s_1 = \frac{40 + 45}{2} = 42,5, \quad s_2 = \frac{45 + 50}{2} = 47,5, \quad s_3 = \frac{50 + 55}{2} = 52,5 \quad (\text{A.27})$$

Για κάθε s_i , υπολογίζουμε την εντροπία των δύο υποσυνόλων:

$$\left(\frac{|S_{\text{left}}|}{|S|} H(S_{\text{left}}) + \frac{|S_{\text{right}}|}{|S|} H(S_{\text{right}}) \right) \quad (\text{A.28})$$

Υπολογίζουμε τις εντροπίες για κάθε πιθανή τιμή διαχωρισμού:

Για $s_1 = 42,5$:

$$\begin{aligned} S_{\text{left}} &= \{25, 30, 35, 40\} \\ S_{\text{right}} &= \{45, 50, 55, 60\} \end{aligned}$$

$$H(S_{\text{left}}) = 0$$

$$H(S_{\text{right}}) = -\left(\frac{3}{4} \log_2 \frac{3}{4} + \frac{1}{4} \log_2 \frac{1}{4}\right) \approx 0.811$$

Η εντροπία των δύο υποσυνόλων για το s_1 είναι:

$$\left(\frac{|S_{\text{left}}|}{|S|} H(S_{\text{left}}) + \frac{|S_{\text{right}}|}{|S|} H(S_{\text{right}}) \right) = \left(\frac{4}{8} \times 0 + \frac{4}{8} \times 0.811 \right) \approx 0.406 \quad (\text{A.29})$$

Για $s_2 = 47.5$:

$$\begin{aligned} S_{\text{left}} &= \{25, 30, 35, 40, 45\} \\ S_{\text{right}} &= \{50, 55, 60\} \end{aligned}$$

$$H(S_{\text{left}}) = -\left(\frac{4}{5} \log_2 \frac{4}{5} + \frac{1}{5} \log_2 \frac{1}{5}\right) \approx 0.7219$$

$$H(S_{\text{right}}) = -\left(\frac{1}{3} \log_2 \frac{1}{3} + \frac{2}{3} \log_2 \frac{2}{3}\right) \approx 0.9183$$

Η εντροπία των δύο υποσυνόλων για το s_2 είναι:

$$\left(\frac{|S_{\text{left}}|}{|S|} H(S_{\text{left}}) + \frac{|S_{\text{right}}|}{|S|} H(S_{\text{right}}) \right) = \left(\frac{5}{8} \times 0.7219 + \frac{3}{8} \times 0.9183 \right) \approx 0.7944 \quad (\text{A.30})$$

Για $s_3 = 52.5$:

$$\begin{aligned} S_{\text{left}} &= \{25, 30, 35, 40, 45, 50\} \\ S_{\text{right}} &= \{55, 60\} \end{aligned}$$

$$H(S_{\text{left}}) = -\left(\frac{5}{6} \log_2 \frac{5}{6} + \frac{1}{6} \log_2 \frac{1}{6}\right) \approx 0.649$$

$$H(S_{\text{right}}) = 0$$

Η εντροπία των δύο υποσυνόλων για το s_3 είναι:

$$\left(\frac{|S_{\text{left}}|}{|S|} H(S_{\text{left}}) + \frac{|S_{\text{right}}|}{|S|} H(S_{\text{right}}) \right) = \left(\frac{6}{8} \times 0.649 + \frac{2}{8} \times 0 \right) \approx 0.4868 \quad (\text{A.31})$$

Για το χαρακτηριστικό Εισόδημα:

$$H(S_{\text{ψηλό}}) = -\left(\frac{2}{4} \log_2 \frac{2}{4} + \frac{2}{4} \log_2 \frac{2}{4}\right) = 1$$

$$H(S_{\text{χαμηλό}}) = -\left(\frac{3}{4} \log_2 \frac{3}{4} + \frac{1}{4} \log_2 \frac{1}{4}\right) = 0.811$$

Η σταθμισμένη εντροπία είναι:

$$\left(\frac{|S_{\text{ψηλό}}|}{|S|} H(S_{\text{ψηλό}}) + \frac{|S_{\text{χαμηλό}}|}{|S|} H(S_{\text{χαμηλό}}) \right) = \left(\frac{4}{8} \times 1 + \frac{4}{8} \times 0.811 \right) \approx 0.906 \quad (\text{A.32})$$

γ) Για να επιλέξουμε το καλύτερο χαρακτηριστικό, υπολογίζουμε το Gain Ratio. Το Gain Ratio υπολογίζεται με την εξής φόρμουλα:

$$\text{Gain Ratio} = \frac{\text{Information Gain}}{\text{Split Information}}$$

Η *Information Gain* υπολογίζεται ως η διαφορά μεταξύ της εντροπίας του αρχικού συνόλου και της μέσης εντροπίας μετά τον διαχωρισμό:

$$\text{Information Gain} = H(S) - H(\text{Χαρακτηριστικό})$$

Για το χαρακτηριστικό Ηλικία:

$$\text{Information Gain}_{\text{Ηλικία}} = 0.954 - 0.406 = 0.548$$

$$\text{Split Information}_{\text{Ηλικία}} = -\left(\frac{4}{8} \log_2 \frac{4}{8} + \frac{4}{8} \log_2 \frac{4}{8}\right) = 1$$

$$\text{Gain Ratio}_{\text{Ηλικία}} = \frac{0.548}{1} = 0.548$$

Για το χαρακτηριστικό Εισόδημα:

$$\text{Information Gain}_{\text{Εισόδημα}} = 0.954 - 0.906 = 0.048$$

$$\text{Split Information}_{\text{Εισόδημα}} = -\left(\frac{4}{8} \log_2 \frac{4}{8} + \frac{4}{8} \log_2 \frac{4}{8}\right) = 1$$

$$\text{Gain Ratio}_{\text{Εισόδημα}} = \frac{0.048}{1} = 0.048$$

Επομένως, το χαρακτηριστικό Εισόδημα είναι το καλύτερο χαρακτηριστικό με το μεγαλύτερο Gain και, συνεπώς, επιλέγεται για τη διαίρεση.

δ) Το χαρακτηριστικό Ηλικία έχει το μεγαλύτερο Gain Ratio. Άρα το χαρακτηριστικό Ηλικία είναι στη ρίζα του δέντρου.

A.3 Κεφάλαιο 4

1. Αρχικά χωρίζουμε τα 2000 παραδείγματα σε εκπαίδευση και έλεγχο. Μια συνηθισμένη διάσπαση είναι 80% εκπαίδευση (1600 δείγματα) και 20% έλεγχος (400 δείγματα). Στη συνέχεια επιλέγουμε το βέλτιστο βάθος του δέντρου, εκτελώντας cross-validation στο σύνολο εκπαίδευσης. Επιλέγουμε το βάθος που έχει το μικρότερο σφάλμα επικύρωσης. Αφού βρούμε το βέλτιστο βάθος, εκπαιδεύουμε ένα νέο δέντρο απόφασης με αυτή την παράμετρο σε όλο το σύνολο εκπαίδευσης (1600 δείγματα) και υπολογίζουμε το σφάλμα στο σύνολο ελέγχου (400 δείγματα). Το σφάλμα αυτό προσεγγίζει το μελλοντικό σφάλμα που θα έκανε το μοντέλο σε νέους πελάτες.

2. Έχουμε 12 παραδείγματα και εκτελούμε στρωματοποιημένη 4-πλή σταυρωτή επικύρωση (stratified 4-fold cross-validation). Αυτό σημαίνει ότι διατηρούμε την αναλογία των κλάσεων σε κάθε fold.

Από το δεδομένο για:

Κλάση 1 ($y = 1$): 8 παραδείγματα.

Κλάση 0 ($y = 0$): 4 παραδείγματα.

Κάθε fold πρέπει να έχει $12 / 4 = 3$ παραδείγματα ελέγχου.

Επειδή η επικύρωση είναι στρωματοποιημένη, η αναλογία των κλάσεων πρέπει να διατηρείται σε κάθε fold. Η συνολική αναλογία των κλάσεων στο dataset είναι:

Κλάση 1: $8/12 \approx 67\%$

Κλάση 0: $4/12 \approx 33\%$.

Σε κάθε fold των 3 παραδειγμάτων, διατηρούμε αυτή την αναλογία:

67% από 3 δείγματα \rightarrow περίπου 2 δείγματα από την κλάση 1

33% από 3 δείγματα \rightarrow περίπου 1 δείγμα από την κλάση 0.

Άρα σε κάθε ένα από τα 4 σύνολα ελέγχου, θα υπάρχουν 1 δείγμα της κλάσης 0 και 2 δείγματα της κλάσης 1.

3. Η ακρίβεια (Precision) για κάθε κλάση i δίνεται από τον τύπο:

$$\text{Precision} = \frac{TP}{TP + FP}$$

όπου:

True Positives (TP): Τα στοιχεία που ανήκουν στην κλάση i και το μοντέλο τα προέβλεψε σωστά ως i

False Positives (FP): Τα στοιχεία που το μοντέλο προέβλεψε ως i , αλλά ανήκουν σε άλλη κλάση.

Ο δεδομένος πίνακας σύγχυσης είναι:

$$\begin{bmatrix} 8 & 0 & 4 & 3 \\ 2 & 11 & 1 & 0 \\ 3 & 2 & 12 & 0 \\ 0 & 4 & 0 & 12 \end{bmatrix}$$

Οι τιμές TP για κάθε κλάση είναι τα στοιχεία της διαγωνίου:

$$TP_0 = 8, \quad TP_1 = 11, \quad TP_2 = 12, \quad TP_3 = 12$$

Τα FP για κάθε κλάση υπολογίζονται αθροίζοντας τα στοιχεία της αντίστοιχης γραμμής (προβλεπόμενη κλάση) εκτός της διαγωνίου:

$$\begin{aligned} FP_0 &= 0 + 4 + 3 = 7 \\ FP_1 &= 2 + 1 + 0 = 3 \\ FP_2 &= 3 + 2 + 0 = 5 \\ FP_3 &= 0 + 4 + 0 = 4 \end{aligned}$$

Τώρα, η ακρίβεια για κάθε κλάση είναι:

$$\begin{aligned} Precision_0 &= \frac{8}{8+7} = \frac{8}{15} \approx 0.533 \\ Precision_1 &= \frac{11}{11+3} = \frac{11}{14} \approx 0.786 \\ Precision_2 &= \frac{12}{12+5} = \frac{12}{17} \approx 0.706 \\ Precision_3 &= \frac{12}{12+4} = \frac{12}{16} = 0.75 \end{aligned}$$

Η κλάση με τη μεγαλύτερη ακρίβεια είναι η κλάση 1 με $Precision \approx 0.786$.

4. Η ανάκληση για κάθε κλάση i δίνεται από τον τύπο:

$$Recall = \frac{TP}{TP + FN}$$

όπου:

True Positives (TP): Τα στοιχεία που ανήκουν στην κλάση i και το μοντέλο τα προέβλεψε σωστά ως i

False Positives (FP): Τα στοιχεία που το μοντέλο προέβλεψε ως i , αλλά ανήκουν σε άλλη κλάση

False Negatives (FN): Τα στοιχεία που ανήκουν στην κλάση i αλλά το μοντέλο τα προέβλεψε ως άλλη κλάση.

Ο νέος πίνακας σύγχυσης είναι:

$$\begin{bmatrix} 9 & 1 & 4 & 2 \\ 2 & 12 & 3 & 4 \\ 3 & 2 & 11 & 0 \\ 0 & 4 & 1 & 9 \end{bmatrix}$$

Οι τιμές TP για κάθε κλάση είναι τα στοιχεία της διαγωνίου:

$$TP_0 = 9, \quad TP_1 = 12, \quad TP_2 = 11, \quad TP_3 = 9$$

Τα FN για κάθε κλάση υπολογίζονται αθροίζοντας τα στοιχεία της αντίστοιχης στήλης (πραγματική κλάση) εκτός της διαγωνίου:

$$\begin{aligned} FN_0 &= 2 + 3 + 0 = 5 \\ FN_1 &= 1 + 2 + 4 = 7 \\ FN_2 &= 4 + 3 + 1 = 8 \\ FN_3 &= 2 + 4 + 0 = 6 \end{aligned}$$

Τώρα, η ανάκληση για κάθε κλάση είναι:

$$\begin{aligned} Recall_0 &= \frac{9}{9+5} = \frac{9}{14} \approx 0.643 \\ Recall_1 &= \frac{12}{12+7} = \frac{12}{19} = 0.63 \\ Recall_2 &= \frac{11}{11+8} = \frac{11}{19} \approx 0.579 \\ Recall_3 &= \frac{9}{9+6} = \frac{9}{15} = 0.6 \end{aligned}$$

Η κλάση με τη μεγαλύτερη ανάκληση είναι η κλάση 0 με $Recall \approx 0.643$.

5. Εξίσωση 4.25:

$$\text{BrierScore} = \frac{1}{m} \sum_{i=1}^m \sum_{j=0}^c \left(\hat{y}_j^{(i)} - y_j^{(i)} \right)^2$$

Εξίσωση 4.24:

$$\text{BrierScore} = \frac{1}{m} \sum_{i=1}^m \left(\hat{y}_1^{(i)} - y_1^{(i)} \right)^2$$

Για δυαδική ταξινόμηση ($c = 1$), η εξίσωση 4.25 γίνεται:

$$\text{BrierScore}_{2 \text{ categories}} = \frac{1}{m} \sum_{i=1}^m \left[\left(\hat{y}_0^{(i)} - y_0^{(i)} \right)^2 + \left(\hat{y}_1^{(i)} - y_1^{(i)} \right)^2 \right]$$

Αντικαθιστούμε τις τιμές για τις κατηγορίες 0 και 1 ($\hat{y}_0^{(i)} = 1 - \hat{y}_1^{(i)}$ και $y_0^{(i)} = 1 - y_1^{(i)}$):

$$= \frac{1}{m} \sum_{i=1}^m \left[\left((1 - \hat{y}_1^{(i)}) - (1 - y_1^{(i)}) \right)^2 + \left(\hat{y}_1^{(i)} - y_1^{(i)} \right)^2 \right]$$

Απλοποιούμε την πρώτη παράσταση:

$$= \frac{1}{m} \sum_{i=1}^m \left[\left(\hat{y}_1^{(i)} - y_1^{(i)} \right)^2 + \left(\hat{y}_1^{(i)} - y_1^{(i)} \right)^2 \right]$$

Αυτό είναι το ίδιο με:

$$= \frac{1}{m} \sum_{i=1}^m 2 \left(\hat{y}_1^{(i)} - y_1^{(i)} \right)^2$$

Άρα, έχουμε:

$$\text{BrierScore}_{2 \text{ categories}} = 2 \times \frac{1}{m} \sum_{i=1}^m \left(\hat{y}_1^{(i)} - y_1^{(i)} \right)^2$$

Αυτό αποδεικνύει ότι η εξίσωση 4.25 σε περίπτωση δυαδικής ταξινόμησης ισούται με το διπλάσιο της εξίσωσης 4.24.

6. α) Υπολογισμός TP και FN από την ανάκληση:

$$\text{Ανάκληση} = \frac{TP}{TP + FN} = 0.50$$

Εφόσον $TP + FN = 200$ (τα πραγματικά θετικά παραδείγματα), έχουμε:

$$\frac{TP}{200} = 0.50 \Rightarrow TP = 0.50 \cdot 200 = 100$$

$$FN = 200 - TP = 200 - 100 = 100$$

Υπολογισμός FP από την ακρίβεια:

$$\text{Precision} = \frac{TP}{TP + FP} = 0.80$$

$$\frac{100}{100 + FP} = 0.80 \Rightarrow 100 = 80 + 0.80 \cdot FP \Rightarrow 20 = 0.80 \cdot FP \Rightarrow FP = \frac{20}{0.80} = 25$$

Υπολογισμός TN:

$$TN = 1000 - TP - FN - FP = 1000 - 100 - 100 - 25 = 775$$

Πίνακας Σύγχυσης:

	Προβλέφθηκε Θετικό	Προβλέφθηκε Αρνητικό
Πραγματικά Θετικό	TP = 100	FN = 100
Πραγματικά Αρνητικό	FP = 25	TN = 775

- β) Υπολογισμός F_1 score:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \cdot \frac{0.80 \cdot 0.50}{0.80 + 0.50} = 2 \cdot \frac{0.40}{1.30} = \frac{0.80}{1.30} \approx 0.615$$

Άρα: $F_1 \approx 0.615$ ή 61.5%

A.4 Κεφάλαιο 5

1. Εξαγωγή Κανόνων με Ταξινόμηση (Ordered Rules):

Οι κανόνες εφαρμόζονται με σειρά, και μόλις ικανοποιηθεί κάποιος, η διαδικασία τερματίζεται.

- Παρατηρούμε ότι όταν $x_1 = 1$ και $x_2 = 1$, το αποτέλεσμα είναι πάντα $y = +$.
- Όταν $x_1 = 0$ και $x_2 = 0$, το αποτέλεσμα είναι επίσης $y = +$.
- Όταν $x_1 = 0$ και $x_2 = 1$, το αποτέλεσμα είναι πάντα $y = -$.

Με βάση τα παραπάνω, μπορούμε να γράψουμε:

- (a) Αν $x_1 = 1$ και $x_2 = 1$, τότε $y = +$.
- (β) Άλλιώς αν $x_1 = 0$ και $x_2 = 0$, τότε $y = +$.
- (γ) Άλλιώς, $y = -$.

Οι παραπάνω κανόνες καλύπτουν σωστά όλα τα παραδείγματα, με οικονομία στον αριθμό τους χάρη στη χρήση προτεραιότητας.

Εξαγωγή Κανόνων χωρίς Ταξινόμηση (Unordered Rules):

Σε αυτή την περίπτωση, οι κανόνες είναι ανεξάρτητοι και δεν εφαρμόζονται με σειρά. Πρέπει να καλύπτουν όλες τις περιπτώσεις χωρίς επικαλύψεις.

Πιθανοί συνδυασμοί των εισόδων:

x_1	x_2	y
1	1	+
1	0	+
0	0	+
0	1	-

Για κάθε συνδυασμό, ορίζουμε έναν ξεχωριστό κανόνα:

- (a) Αν $x_1 = 1$ και $x_2 = 1$, τότε $y = +$.
- (β) Αν $x_1 = 1$ και $x_2 = 0$, τότε $y = +$.
- (γ) Αν $x_1 = 0$ και $x_2 = 0$, τότε $y = +$.
- (δ) Αν $x_1 = 0$ και $x_2 = 1$, τότε $y = -$.

Έτσι διασφαλίζεται ότι κάθε είσοδος αντιστοιχεί σε ακριβώς έναν κανόνα, χωρίς να απαιτείται default rule.

2. Συγκεντρωτικός Πίνακας Υποσυνθηκών:

Για κάθε τιμή των x_1 και x_2 , καταγράφουμε πόσες φορές εμφανίζεται με έξοδο $y = 0$ ή $y = 1$:

Συνθήκη	$y = 0$	$y = 1$
$x_1 = 0$	0	1
$x_1 = 1$	2	1
$x_1 = 2$	0	1
$x_2 = 0$	0	2
$x_2 = 1$	1	0
$x_2 = 2$	1	1

Επιλογή Καλύτερου Κανόνα:

Ψάχνουμε για τη συνθήκη που:

- Είναι όσο το δυνατόν πιο ομοιογενής, δηλαδή να δίνει πάντα την ίδια τιμή για το y .
- Καλύπτει τα περισσότερα παραδείγματα.

Ανάλυση:

- $x_2 = 0$: 2 παραδείγματα με $y = 1 \rightarrow$ απόλυτα ομοιογενής.
- $x_2 = 1$: 1 παράδειγμα με $y = 0 \rightarrow$ ομοιογενής αλλά μικρότερη κάλυψη.
- $x_1 = 1$: 3 παραδείγματα με $y = 0, 0, 1 \rightarrow$ όχι ομοιογενής.

Άρα, επιλέγουμε: $x_2 = 0$

Τελικός Κανόνας:

$$\boxed{\text{Av } x_2 = 0 \text{ τότε } y = 1}$$

Ο παραπάνω κανόνας είναι ο πρώτος που θα μάθει ο αλγόριθμος επειδή:

- Καλύπτει 2 παραδείγματα.
- Και τα δύο έχουν την ίδια έξοδο $y = 1$.
- Είναι ο πιο καθαρός (ομοιογενής) με τη μεγαλύτερη κάλυψη.

3. Ο παρακάτω πίνακας δείχνει την καταλληλότητα 5 συνόλων κανόνων:

Σύνολο	Καταλληλότητα
1	15
2	10
3	20
4	5
5	50

Το άθροισμα των καταλληλοτήτων είναι:

$$S = 15 + 10 + 20 + 5 + 50 = 100 \quad (\text{A.33})$$

Η πιθανότητα επιλογής του συνόλου 2 υπολογίζεται ως:

$$P(2) = \frac{10}{100} = 0.1 \quad (\text{A.34})$$

4. α) Βήμα-Βήμα Κατασκευή Κανόνων:

(α) Πρώτη παρατήρηση: (Απλός, 1) → Ναι

Κανόνας:

$$\text{Αν } X1 = \text{Απλός και } X2 = 1, \text{ τότε } Y = \text{Ναι}$$

Αφαιρείται το παράδειγμα 1.

(β) Επόμενο μη καλυμμένο παράδειγμα: (Συνδρομητής, 2) → 'Οχι

Κανόνας:

$$\text{Αν } X1 = \text{Συνδρομητής και } X2 = 2, \text{ τότε } Y = \text{'Οχι}$$

Αφαιρείται το παράδειγμα 2.

(γ) Επόμενο: (Συνδρομητής, 3) → 'Οχι

Κανόνας:

$$\text{Αν } X1 = \text{Συνδρομητής και } X2 = 3, \text{ τότε } Y = \text{'Οχι}$$

Αφαιρείται το παράδειγμα 3.

(δ) Επόμενο: (Απλός, 4) → Ναι

Κανόνας:

$$\text{Αν } X1 = \text{Απλός και } X2 = 4, \text{ τότε } Y = \text{Ναι}$$

Αφαιρείται το παράδειγμα 4.

(ε) Επόμενο: (Απλός, 2) → Ναι

Κανόνας:

$$\text{Αν } X1 = \text{Απλός και } X2 = 2, \text{ τότε } Y = \text{Ναι}$$

Αφαιρείται το παράδειγμα 5.

(στ) Τέλος: (Συνδρομητής, 4) → Ναι

Κανόνας:

$$\text{Αν } X1 = \text{Συνδρομητής και } X2 = 4, \text{ τότε } Y = \text{Ναι}$$

Τελική Λίστα Ταξινομημένων Κανόνων:

(α) Αν $X1 = \text{Απλός}$ και $X2 = 1$, τότε $Y = \text{Ναι}$

(β) Αν $X1 = \text{Συνδρομητής}$ και $X2 = 2$, τότε $Y = \text{'Οχι}$

(γ) Αν $X1 = \text{Συνδρομητής}$ και $X2 = 3$, τότε $Y = \text{'Οχι}$

(δ) Αν $X1 = \text{Απλός}$ και $X2 = 4$, τότε $Y = \text{Ναι}$

(ε) Αν $X1 = \text{Απλός}$ και $X2 = 2$, τότε $Y = \text{Ναι}$

(στ) Αν $X1 = \text{Συνδρομητής}$ και $X2 = 4$, τότε $Y = \text{Ναι}$

β) Χρησιμοποιώντας τη μέθοδο των "μη ταξινομημένων κανόνων", πρέπει να προβλεπτούν οι κανόνες χωρίς προηγούμενη ταξινόμηση. Η πρόβλεψη για έναν πελάτη τύπου συνδρομητή και συχνότητα επίσκεψης 2 προκύπτει από τη συνδυασμένη συχνότητα επίσκεψης και τον τύπο πελάτη στα δεδομένα. Από τον πίνακα, οι παρατηρήσεις για πελάτες τύπου συνδρομητή και συχνότητα επίσκεψης 2 είναι:

Συνδρομητής, Συχνότητα Επίσκεψης 2: 'Οχι

Συνεπώς, βάσει των "μη ταξινομημένων κανόνων", η πρόβλεψη για έναν πελάτη τύπου συνδρομητή και συχνότητα επίσκεψης 2 είναι "Οχι".

A.5 Κεφάλαιο 7

- Υπολογισμός βαρών στο AdaBoost.M1 (μετά το πρώτο μοντέλο)

Δεδομένα:

- Πραγματικές ετικέτες: $y = [+, +, -, +, -, -]$
- Προβλέψεις του πρώτου μοντέλου: $h_1 = [+, +, +, -, -, -]$

Βήμα 1: Αρχικοποίηση βαρών

$$w^{(1)}(i) = \frac{1}{6}, \quad \text{για κάθε } i = 1, \dots, 6$$

Βήμα 2: Υπολογισμός σταθμισμένου σφάλματος του h_1

Τα λάθη είναι στα παραδείγματα 3 και 4.

$$e_1 = w^{(1)}(3) + w^{(1)}(4) = \frac{1}{6} + \frac{1}{6} = \frac{1}{3}$$

Βήμα 3: Ενημέρωση βαρών

Για τα σωστά ταξινομημένα παραδείγματα:

$$w^{(2)}(i) = w^{(1)}(i) \cdot \frac{e_1}{1 - e_1} = \frac{1}{6} \cdot \frac{1/3}{2/3} = \frac{1}{12}$$

Για τα λάθος ταξινομημένα (3 και 4), το βάρος παραμένει:

$$w^{(2)}(3) = w^{(2)}(4) = \frac{1}{6}$$

Βήμα 4: Κανονικοποίηση βαρών

$$\sum w^{(2)} = \frac{1}{12} + \frac{1}{12} + \frac{1}{6} + \frac{1}{6} + \frac{1}{12} + \frac{1}{12} = \frac{8}{12} = \frac{2}{3}$$

Διαιρεση κάθε βάρους με 2/3:

$$w_{\kappa\alphaνoν.}^{(2)}(i) = \frac{w^{(2)}(i)}{2/3}$$

Τελικά βάρη:

Παράδειγμα	Κανονικοποιημένο Βάρος
1	$\frac{1}{12}/\frac{2}{3} = \frac{1}{8}$
2	$\frac{1}{12}/\frac{2}{3} = \frac{1}{8}$
3	$\frac{1}{6}/\frac{2}{3} = \frac{1}{4}$
4	$\frac{1}{6}/\frac{2}{3} = \frac{1}{4}$
5	$\frac{1}{12}/\frac{2}{3} = \frac{1}{8}$
6	$\frac{1}{12}/\frac{2}{3} = \frac{1}{8}$

Άρα, τα παραδείγματα 3 και 4 (που ταξινομήθηκαν λάθος) έχουν τα μεγαλύτερα βάρη κατά την εκπαίδευση του δεύτερου μοντέλου.

2. Η αρχική πρόβλεψη h_0 στο Gradient Boosting είναι η μέση τιμή των τιμών εξόδου y :

$$h_0 = \frac{1}{N} \sum_{i=1}^N y_i$$

Για το δοσμένο σύνολο εκπαίδευσης:

$$y = \{10, 30, 50, 90, 170, 250\}$$

Υπολογίζουμε το μέσο όρο:

$$h_0 = \frac{10 + 30 + 50 + 90 + 170 + 250}{6} = \frac{600}{6} = 100$$

Άρα, η αρχική πρόβλεψη h_0 είναι:

$$h_0 = 100$$

Τα κατάλοιπα (residuals) υπολογίζονται ως:

$$r_i = y_i - h_0$$

Για κάθε παράδειγμα εκπαίδευσης:

$$r_1 = 10 - 100 = -90$$

$$r_2 = 30 - 100 = -70$$

$$r_3 = 50 - 100 = -50$$

$$r_4 = 90 - 100 = -10$$

$$r_5 = 170 - 100 = 70$$

$$r_6 = 250 - 100 = 150$$

Άρα, για το παράδειγμα όπου $x_1 = 10$, το υπόλοιπο (η έξοδος του h_1 πριν την εφαρμογή του συντελεστή συρρίκνωσης) είναι:

$$r_1 = -90$$

Η έξοδος του συνόλου των δύο μοντέλων δίνεται από τη σχέση:

$$F_1(x) = h_0 + \eta h_1(x)$$

Εφόσον για κάποιο παράδειγμα έχουμε $F_1(x) = 120$, αντικαθιστούμε $h_0 = 100$:

$$120 = 100 + 0.1 \cdot h_1(x)$$

Λύνοντας για $h_1(x)$:

$$h_1(x) = \frac{120 - 100}{0.1} = \frac{20}{0.1} = 200$$

Άρα, η έξοδος του δεύτερου μοντέλου h_1 στο συγκεκριμένο παράδειγμα είναι:

$$h_1(x) = 200$$

3. Στη μέθοδο της αυτοδύναμης συνάθροισης (bagging), δημιουργούμε ένα νέο σύνολο εκπαίδευσης επιλέγοντας τυχαία N δείγματα από το αρχικό σύνολο με επαναποθέτηση. Αυτό σημαίνει ότι κάθε αρχικό δείγμα μπορεί να εμφανιστεί 0, 1, 2, ... φορές στο νέο σύνολο.

Για ένα σύνολο εκπαίδευσης μεγέθους N , η συνολική συχνότητα των δειγμάτων στο νέο σύνολο πρέπει να είναι επίσης N , αφού επιλέγουμε N δείγματα.

Έστω ότι έχουμε $N = 6$ παραδείγματα. Από τον πίνακα:

Παράδειγμα	1	2	3	4	5	6
Συχνότητα	0	1	?	2	0	3

Το άθροισμα των συχνοτήτων πρέπει να είναι ίσο με 6:

$$0 + 1 + x + 2 + 0 + 3 = 6$$

Απλοποιώντας:

$$x + 6 = 6$$

$$x = 0$$

Η αγνοούμενη συχνότητα για το τρίτο παράδειγμα είναι:

$$\mathbf{0}$$

Αυτό σημαίνει ότι το τρίτο παράδειγμα δεν επιλέχθηκε καθόλου στη συγκεκριμένη παραλλαγή του συνόλου εκπαίδευσης.

4. Δίνεται ένα σύνολο δεδομένων ελέγχου (x, y) καθώς και οι προβλέψεις από δύο μοντέλα h_1 και h_2 . Το μετα-μοντέλο είναι ένα γραμμικό μοντέλο παλινδρόμησης με τις παραμέτρους:

$$\theta_0 = 0.1, \quad \theta_1 = 0.5, \quad \theta_2 = 0.5$$

Η έξοδος του μετα-μοντέλου υπολογίζεται ως εξής:

$$\hat{y} = \theta_0 + \theta_1 h_1 + \theta_2 h_2$$

Από τον πίνακα, για το 1ο παράδειγμα έχουμε:

$$h_1 = 1, \quad h_2 = 2$$

Άρα, η έξοδος του μετα-μοντέλου είναι:

$$\hat{y} = 0.1 + 0.5(1) + 0.5(2)$$

$$\hat{y} = 0.1 + 0.5 + 1$$

$$\hat{y} = 1.6$$

Η έξοδος του μετα-μοντέλου για το 1ο παράδειγμα ελέγχου είναι:

1.6

5. (α) Κάθε παραλλαγή πρέπει να περιλαμβάνει ακριβώς 8 δείγματα (με επαναλήψεις). Για κάθε στήλη προσθέτουμε τις γνωστές τιμές και αφαιρούμε από το 8 για να βρούμε τις άγνωστες.

- Παραλλαγή 1: $1 + 0 + 2 + 0 + 3 + x + 0 + 2 = 8 \Rightarrow x = 0$
- Παραλλαγή 2: $0 + 2 + 1 + 0 + x + 2 + 1 + 0 = 8 \Rightarrow x = 2$
- Παραλλαγή 3: $2 + x + 0 + 1 + 0 + 1 + y + 2 = 8 \Rightarrow x + y = 2$, άρα οι δυνατές λύσεις είναι $\{(x = 0, y = 2), (x = 2, y = 0), (x = 1, y = 1)\}$, εκ των οποίων επιλέγουμε $x = 0, y = 2$

Άρα ο συμπληρωμένος πίνακας είναι:

Παράδειγμα	Παραλλαγή 1	Παραλλαγή 2	Παραλλαγή 3
1	1	0	2
2	0	2	0
3	2	1	0
4	0	0	1
5	3	2	0
6	0	2	1
7	0	1	2
8	2	0	2

(β) Είναι όσα έχουν συχνότητα 0 σε κάθε παραλλαγή.

- Παραλλαγή 1: Παραδείγματα 2, 4, 6, 7
- Παραλλαγή 2: Παραδείγματα 1, 4, 5, 8
- Παραλλαγή 3: Παραδείγματα 2, 3, 5

A.6 Κεφάλαιο 8

1. Έστω ότι ο συντελεστής έκπτωσης είναι $\gamma = 0.7$ και οι ανταμοιβές είναι:

$$\begin{aligned} R_1 &= 2, \\ R_2 &= 1, \\ R_3 &= -1, \\ R_4 &= 1. \end{aligned}$$

Το κέρδος G_t ορίζεται ως:

$$G_t = \sum_{k=0}^{+\infty} \gamma^k R_{t+k+1} = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots$$

Υπολογίζουμε διαδοχικά:

$$\begin{aligned} G_0 &= R_1 + \gamma R_2 + \gamma^2 R_3 + \gamma^3 R_4 \\ &= 2 + (0.7 \times 1) + (0.7^2 \times (-1)) + (0.7^3 \times 1) \\ &= 2 + 0.7 - 0.49 + 0.343 \\ &= 2.553. \end{aligned}$$

$$\begin{aligned} G_1 &= R_2 + \gamma R_3 + \gamma^2 R_4 \\ &= 1 + (0.7 \times (-1)) + (0.7^2 \times 1) \\ &= 1 - 0.7 + 0.49 \\ &= 0.79. \end{aligned}$$

$$\begin{aligned} G_2 &= R_3 + \gamma R_4 \\ &= -1 + (0.7 \times 1) \\ &= -1 + 0.7 \\ &= -0.3. \end{aligned}$$

$$G_3 = R_4 = 1.$$

$$G_4 = 0 \quad (\text{δεν υπάρχουν μελλοντικές ανταμοιβές}).$$

Άρα τα κέρδη είναι:

$$\begin{aligned} G_0 &= 2.553, \\ G_1 &= 0.79, \\ G_2 &= -0.3, \\ G_3 &= 1, \\ G_4 &= 0. \end{aligned}$$

2. Έστω ότι ο συντελεστής έκπτωσης είναι $\gamma = 0.8$ και η ακολουθία ανταμοιβών είναι:

$$\begin{aligned} R_1 &= 1, \\ R_2 &= 2, \\ R_3 &= 2, \\ &\vdots \end{aligned}$$

Το κέρδος G_t ορίζεται ως:

$$G_t = \sum_{k=0}^{+\infty} \gamma^k R_{t+k+1} = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots$$

Υπολογίζουμε:

$$\begin{aligned} G_0 &= R_1 + \gamma R_2 + \gamma^2 R_3 + \gamma^3 R_4 + \dots \\ &= 1 + 0.8 \times 2 + 0.8^2 \times 2 + 0.8^3 \times 2 + \dots \\ &= 1 + 2(0.8 + 0.8^2 + 0.8^3 + \dots). \end{aligned}$$

Η άπειρη γεωμετρική σειρά $S = 0.8 + 0.8^2 + 0.8^3 + \dots$ έχει άθροισμα:

$$S = \frac{0.8}{1 - 0.8} = 4.$$

Άρα,

$$G_0 = 1 + 2 \times 4 = 9.$$

Για το G_1 :

$$\begin{aligned} G_1 &= R_2 + \gamma R_3 + \gamma^2 R_4 + \dots \\ &= 2 + 0.8 \times 2 + 0.8^2 \times 2 + \dots \\ &= 2(1 + 0.8 + 0.8^2 + \dots). \end{aligned}$$

Χρησιμοποιώντας ξανά το άθροισμα της άπειρης γεωμετρικής σειράς:

$$G_1 = 2 \times \frac{1}{1 - 0.8} = 2 \times 5 = 10.$$

Άρα τα κέρδη είναι:

$$\begin{aligned} G_0 &= 9, \\ G_1 &= 10. \end{aligned}$$

3. Θέλουμε να υπολογίσουμε την τιμή $v_\pi(16)$ σε δύο περιπτώσεις:

Περίπτωση 1: Προσθήκη της κατάστασης 16 κάτω από την 13

Η νέα κατάσταση 16 έχει 4 ενέργειες (αριστερά, επάνω, δεξιά, κάτω) που οδηγούν στις καταστάσεις 12, 13, 14 και 16 αντίστοιχα. Θεωρούμε τυχαία πολιτική (πιθανότητα 1/4 ανά ενέργεια) και $\gamma = 1$. Όλες οι μεταβάσεις έχουν ανταμοιβή -1 .

Από το Σχήμα 8.6(στ) έχουμε:

$$v_\pi(12) = -22, \quad v_\pi(13) = -14, \quad v_\pi(14) = -20$$

Η εξίσωση Bellman για $v_\pi(16)$ δίνει:

$$\begin{aligned} v_\pi(16) &= \frac{1}{4}[-1 + v_\pi(12)] + \frac{1}{4}[-1 + v_\pi(13)] + \frac{1}{4}[-1 + v_\pi(14)] + \frac{1}{4}[-1 + v_\pi(16)] \\ &= \frac{1}{4}(-23 - 15 - 21 - 1 + v_\pi(16)) \\ &= \frac{-60 + v_\pi(16)}{4} \end{aligned}$$

Λύνοντας:

$$v_\pi(16) = \frac{-60 + v_\pi(16)}{4} \Rightarrow 4v_\pi(16) = -60 + v_\pi(16) \Rightarrow 3v_\pi(16) = -60 \Rightarrow v_\pi(16) = -20$$

Περίπτωση 2: Επιπλέον, η ενέργεια “κάτω” από την κατάσταση 13 οδηγεί στην 16

Θέλουμε τώρα να υπολογίσουμε τη νέα τιμή $v_\pi(13)$, αφού η “κάτω” ενέργεια πλέον οδηγεί στην 16 (αντί να παραμένει στην 13). Χρησιμοποιούμε πάλι την εξίσωση Bellman:

Από τα δεδομένα:

$$v_\pi(9) = -20, \quad v_\pi(14) = -20, \quad v_\pi(12) = -22, \quad v_\pi(16) = -20$$

Άρα $v_\pi(16) = -20$ και στις δύο περιπτώσεις.

4. Η εξίσωση ενημέρωσης για τη συνάρτηση αξίας ενέργειας $Q^\pi(s_t, a_t)$ στη διαδοχική προσέγγιση είναι η εξής:

$$Q^\pi(s_t, a_t) \leftarrow Q^\pi(s_t, a_t) + \alpha [r_{t+1} + \gamma Q^\pi(s_{t+1}, a_{t+1}) - Q^\pi(s_t, a_t)]$$

όπου:

- $Q^\pi(s_t, a_t)$ είναι η εκτιμώμενη τιμή της αξίας ενέργειας για την κατάσταση s_t και την ενέργεια a_t ,
- r_{t+1} είναι η άμεση ανταμοιβή που προκύπτει από την εκτέλεση της ενέργειας a_t στην κατάσταση s_t ,
- γ είναι ο συντελεστής απόσβεσης (discount factor),
- α είναι το ποσοστό εκμάθησης (learning rate),
- s_{t+1} και a_{t+1} είναι η επόμενη κατάσταση και ενέργεια αντίστοιχα.

5. Οι εξισώσεις του Bellman για το ρομπότ ανακύκλωσης είναι ένα κρίσιμο κομμάτι της διαδικασίας μάθησης, ειδικά αν το σύστημα επιδιώκει να βελτιστοποιήσει την απόδοσή του στον εντοπισμό και την κατηγοριοποίηση υλικών ανακύκλωσης.

Για να καθορίσουμε τις βέλτιστες εξισώσεις Bellman, πρέπει να λάβουμε υπόψη μας την κατάσταση s_t του ρομπότ (δηλαδή, την τρέχουσα θέση ή κατάσταση του ρομπότ στον περιβάλλον του) και την ενέργεια a_t που εκτελεί σε αυτήν την κατάσταση. Οι εξισώσεις Bellman για το πρόβλημα του ρομπότ

ανακύκλωσης θα είναι για μια αξία πολιτικής $Q^\pi(s_t, a_t)$ και θα εξαρτώνται από την ανταμοιβή που λαμβάνει το ρομπότ και τις μελλοντικές καταστάσεις του.

Βέλτιστες Εξισώσεις Bellman:

Για τη συνάρτηση αξίας πολιτικής $Q^\pi(s_t, a_t)$:

$$Q^\pi(s_t, a_t) = \mathbb{E} [r_{t+1} + \gamma Q^\pi(s_{t+1}, a_{t+1})]$$

Όπου:

- $Q^\pi(s_t, a_t)$ είναι η αξία της εκτέλεσης της ενέργειας a_t στην κατάσταση s_t .
- r_{t+1} είναι η άμεση ανταμοιβή που λαμβάνει το ρομπότ για την εκτέλεση της ενέργειας a_t .
- γ είναι ο συντελεστής απόσβεσης (discount factor), που καθορίζει την προτεραιότητα των μελλοντικών ανταμοιβών σε σχέση με τις άμεσες.
- $Q^\pi(s_{t+1}, a_{t+1})$ είναι η αξία της ενέργειας στην επόμενη κατάσταση s_{t+1} .

Για τη βέλτιστη πολιτική π^* , η οποία μεγιστοποιεί το Q -συνάρτημα:

$$Q^*(s_t, a_t) = \mathbb{E} \left[r_{t+1} + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1}) \right]$$

Στην περίπτωση αυτή:

- $Q^*(s_t, a_t)$ είναι η βέλτιστη αξία ενέργειας για την κατάσταση s_t και ενέργεια a_t , και επιδιώκει τη μεγιστοποίηση του μελλοντικού κέρδους.

Εξίσωση του Bellman για την αξία της κατάστασης $V^\pi(s_t)$:

$$V^\pi(s_t) = \mathbb{E} [r_{t+1} + \gamma V^\pi(s_{t+1})]$$

6. Στην ενότητα 8.2.1, εφαρμόζεται ο αλγόριθμος επανάληψη πολιτικής σε έναν κόσμο πλέγματος 4×4 , με δύο τερματικές καταστάσεις. Ξεκινώντας από μια πολιτική π_0 που επιλέγει ενέργειες ισοπίθανα, κάθε επανάληψη του αλγορίθμου περιλαμβάνει:

- Αξιολόγηση πολιτικής (υπολογισμός της v_π για την τρέχουσα πολιτική).
- Βελτίωση πολιτικής (επιλογή ενεργειών που μεγιστοποιούν την αναμενόμενη αξία).

Από τον πίνακα 8.2, παρατηρούμε ότι μετά από μία μόνο επανάληψη (δηλαδή μετά την παραγωγή της π_1 από την π_0), η πολιτική σταθεροποιείται, πράγμα που σημαίνει ότι:

$$\pi_1 = \pi^*$$

Άρα, ο αλγόριθμος καταλήγει στη βέλτιστη πολιτική μετά από μία επανάληψη.

Απάντηση: Μία (1) επανάληψη αρκεί για να καταλήξουμε σε βέλτιστη πολιτική στον συγκεκριμένο κόσμο πλέγματος.

7. Αν ο πράκτορας επιλέξει:

- Επάνω: μεταβαίνει από A → B με ανταμοιβή +50, και στη συνέχεια κάνει 99 μετακινήσεις με ανταμοιβή -1 ανά βήμα.

- Κάτω: μεταβαίνει από $A \rightarrow G$ με ανταμοιβή -50 , και στη συνέχεια κάνει 99 μετακινήσεις με ανταμοιβή $+1$ ανά βήμα.

Το συνολικό αναμενόμενο κέρδος για κάθε διαδρομή δίνεται από:

$$G_{\text{επάνω}} = 50 - \gamma \cdot \frac{1 - \gamma^{99}}{1 - \gamma}$$

$$G_{\text{κάτω}} = -50 + \gamma \cdot \frac{1 - \gamma^{99}}{1 - \gamma}$$

Για να είναι προτιμότερη η ενέργεια επάνω, πρέπει να ισχύει:

$$G_{\text{επάνω}} > G_{\text{κάτω}}$$

Δηλαδή:

$$50 - \gamma \cdot \frac{1 - \gamma^{99}}{1 - \gamma} > -50 + \gamma \cdot \frac{1 - \gamma^{99}}{1 - \gamma}$$

Μεταφέροντας τους όρους έχουμε:

$$100 > 2\gamma \cdot \frac{1 - \gamma^{99}}{1 - \gamma}$$

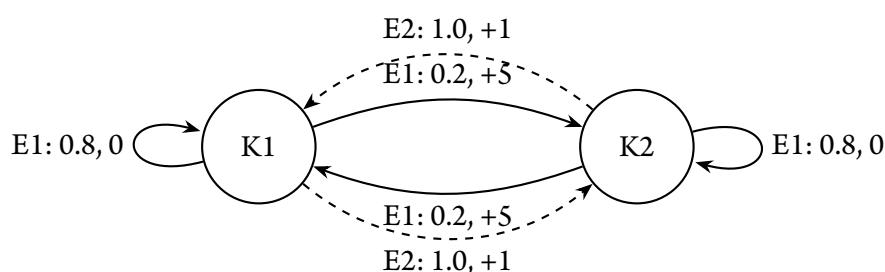
Διαιρώντας με 2γ και αναδιατάσσοντας, προκύπτει η τελική ανίσωση:

$$\frac{1 - \gamma^{99}}{1 - \gamma} < \frac{50}{\gamma}$$

Απάντηση: Ο πράκτορας θα επιλέξει την ενέργεια επάνω, αν ισχύει:

$$\boxed{\frac{1 - \gamma^{99}}{1 - \gamma} < \frac{50}{\gamma}}$$

8. α)



β) Η εξίσωση Bellman για την ενημέρωση είναι:

$$V_{k+1}(s) = \max_a \sum_{s'} P(s'|s, a) [R(s, a, s') + \gamma V_k(s')]$$

Αρχικά:

$$V_0(K1) = V_0(K2) = 0$$

Για K1:

$$Q(K1, E1) = 0.8(0 + 0.9 \cdot 0) + 0.2(5 + 0.9 \cdot 0) = 1.0$$

$$Q(K1, E2) = 1.0(1 + 0.9 \cdot 0) = 1.0$$

$$V_1(K1) = \max(1.0, 1.0) = 1.0$$

Για K2:

$$Q(K2, E1) = 0.8(0 + 0.9 \cdot 0) + 0.2(5 + 0.9 \cdot 0) = 1.0$$

$$Q(K2, E2) = 1.0(1 + 0.9 \cdot 0) = 1.0$$

$$V_1(K2) = \max(1.0, 1.0) = 1.0$$

Άρα:

$$V_1(K1) = V_1(K2) = 1.0$$

γ) Υπολογίζουμε την αναμενόμενη απόδοση για κάθε ενέργεια με $V(K1) = V(K2) = 1.0$.

Για K1:

$$Q(K1, E1) = 0.8(0 + 0.9 \cdot 1.0) + 0.2(5 + 0.9 \cdot 1.0) = 0.72 + 1.18 = 1.90$$

$$Q(K1, E2) = 1.0(1 + 0.9 \cdot 1.0) = 1.9$$

Για K2:

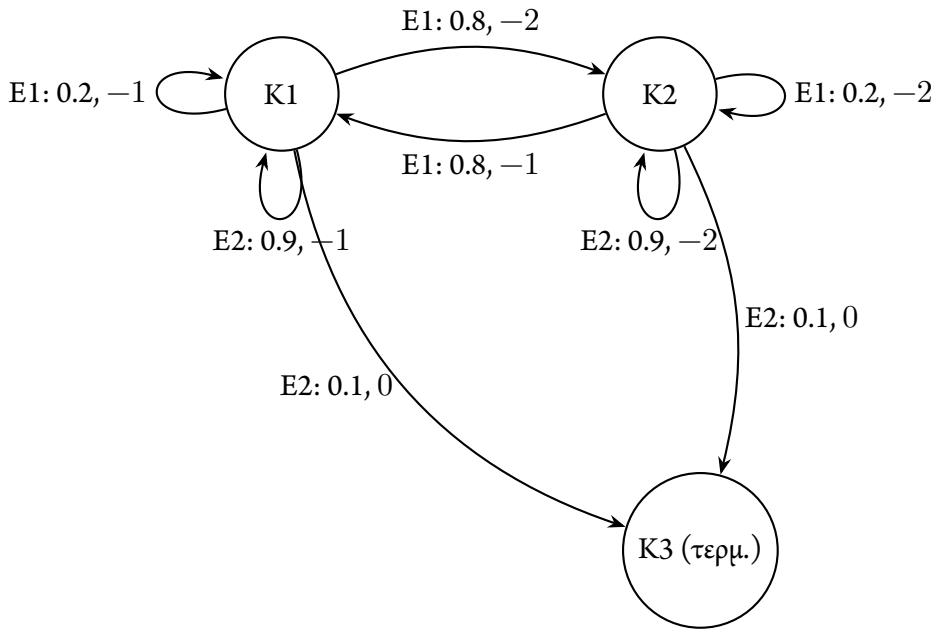
$$Q(K2, E1) = 0.8(0 + 0.9 \cdot 1.0) + 0.2(5 + 0.9 \cdot 1.0) = 1.90$$

$$Q(K2, E2) = 1.0(1 + 0.9 \cdot 1.0) = 1.9$$

Συμπέρασμα: Εφόσον οι δύο ενέργειες έχουν ίση αναμενόμενη αξία και στις δύο καταστάσεις, η νέα πολιτική μπορεί να είναι στοχαστική:

$$\pi'(K1) = \pi'(K2) = \text{ισοπίθανα μεταξύ E1 και E2 (π.χ. } 50\% - 50\%)$$

9. α)



β) Η κατάσταση K3 είναι τερματική. Άρα:

$$V(K3) = 0$$

γ) Αρχικά: $V_0(K1) = V_0(K2) = 0$

$$\text{Για } K1: Q(K1, E1) = 0.8(-2 + 0.9 \cdot V_0(K2)) + 0.2(-1 + 0.9 \cdot V_0(K1)) = 0.8 \cdot (-2) + 0.2 \cdot (-1) = -1.6 - 0.2 = -1.8, Q(K1, E2) = 0.1(0 + 0.9 \cdot V_0(K3)) + 0.9(-1 + 0.9 \cdot V_0(K1)) = 0 + 0.9 \cdot (-1) = -0.9$$

$$V_1(K1) = \max(-1.8, -0.9) = -0.9$$

$$\text{Για } K2: Q(K2, E1) = 0.8(-1 + 0.9 \cdot V_0(K1)) + 0.2(-2 + 0.9 \cdot V_0(K2)) = 0.8 \cdot (-1) + 0.2 \cdot (-2) = -0.8 - 0.4 = -1.2$$

$$V_1(K2) = -1.2$$

Τελικά αποτελέσματα μετά από μία επανάληψη:

$$V(K1) = -0.9, \quad V(K2) = -1.2, \quad V(K3) = 0$$

δ) Επιλέγουμε την ενέργεια με τη μεγαλύτερη τιμή $Q(s, a)$.

Για K1:

$$Q(K1, E1) = 0.8(-2 + 0.9 \cdot (-1.2)) + 0.2(-1 + 0.9 \cdot (-0.9)) = -1.8 - 0.2 \cdot (1.81) = \approx -2.162$$

$$Q(K1, E2) = 0.1(0 + 0.9 \cdot 0) + 0.9(-1 + 0.9 \cdot (-0.9)) = 0 + 0.9(-1.81) = -1.629$$

Άρα:

$$\pi(K1) = E2$$

$$\text{Για } K2: Q(K2, E1) = 0.8(-1 + 0.9 \cdot (-0.9)) + 0.2(-2 + 0.9 \cdot (-1.2)) = -0.8 - 0.648 + (-0.4 - 0.216) = -2.064$$

'Apa:

$\pi(K2) = \text{El}$ (μοναδική)

Τελική βελτιωμένη πολιτική:

$\pi(K1) = E2$, $\pi(K2) = E1$, $\pi(K3) = \tau\epsilon\rho\mu\alpha\tau i\kappa\gamma$ (καμία ενέργεια)

10. Επεισόδιο 1:

$$s_1, a_1, -1, \quad s_2, a_2, -1, \quad s_1, a_1, -1, \quad s_2, a_1, -1, \quad s_4$$

Επεισόδιο 2:

$$s_2, a_1, -1, \quad s_1, a_2, -1, \quad s_1, a_1, -1, \quad s_4$$

Αρχικοποίηση: Όλες οι εκτιμήσεις $Q(s,a)$ ξεκινούν από 0.

Ενημέρωση μετά το 1ο επεισόδιο:

Πρώτες εμφανίσεις:

- (s_1, a_1) : εμφανίζεται πρώτη φορά στο βήμα 1 \rightarrow κέρδος $G = -4$
 - (s_2, a_2) : στο βήμα 2 $\rightarrow G = -3$
 - (s_2, a_1) : στο βήμα 4 $\rightarrow G = -1$

Αυξητική ενημέρωση:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \cdot (G - Q(s, a))$$

$$Q(s_1, a_1) = 0 + 0.1 \cdot (-4 - 0) = -0.4$$

$$Q(s_2, a_2) = 0 + 0.1 \cdot (-3 - 0) = -0.3$$

$$Q(s_2, a_1) = 0 + 0.1 \cdot (-1 - 0) = -0.1$$

Ενημέρωση μετά το 2ο επεισόδιο:

Πρώτες εμφανίσεις:

- (s_2, a_1) : βήμα $1 \rightarrow G = -3$
 - (s_1, a_2) : βήμα $2 \rightarrow G = -2$
 - (s_1, a_1) : βήμα $3 \rightarrow G = -1$

$$\begin{aligned}
 Q(s_2, a_1) &= -0.1 + 0.1 \cdot (-3 - (-0.1)) = -0.1 + 0.1 \cdot (-2.9) = -0.1 - 0.29 = -0.39 \\
 Q(s_1, a_2) &= 0 + 0.1 \cdot (-2 - 0) = -0.2 \\
 Q(s_1, a_1) &= -0.4 + 0.1 \cdot (-1 + 0.4) = -0.46
 \end{aligned}$$

Τελικές τιμές $Q(s, a)$

Κατάσταση	Ενέργεια	$Q(s, a)$
s_1	a_1	-0.46
s_1	a_2	-0.2
s_2	a_1	-0.39
s_2	a_2	-0.3

Τελική πολιτική με ε -μαλακή επιλογή ($\varepsilon=0.1$)

Η πιθανότητα επιλογής της καλύτερης ενέργειας είναι:

$$\pi(a|s) = \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{|A(s)|}, & \text{αν } a = \arg \max_{a'} Q(s, a') \\ \frac{\varepsilon}{|A(s)|}, & \text{αλλιώς} \end{cases}$$

Έστω ότι $|A(s)| = 2$.

Για s_1 , το a_2 είναι καλύτερο ($-0.2 > -0.46$):

$$\pi(a_2|s_1) = 0.95, \quad \pi(a_1|s_1) = 0.05$$

Για s_2 , το a_2 είναι επίσης καλύτερο ($-0.3 > -0.39$):

$$\pi(a_2|s_2) = 0.95, \quad \pi(a_1|s_2) = 0.05$$

Τελική πολιτική (στοχαστική):

$$\pi(s_1) = \begin{cases} a_2 & \text{με πιθανότητα } 0.95 \\ a_1 & \text{με πιθανότητα } 0.05 \end{cases} \quad \pi(s_2) = \begin{cases} a_2 & \text{με πιθανότητα } 0.95 \\ a_1 & \text{με πιθανότητα } 0.05 \end{cases}$$

11. Αλγόριθμος SARSA

Η ενημέρωση έχει τη μορφή:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

Αρχικοποίηση: Όλα τα $Q(s, a) = 0$

Επεισόδιο 1:

(α) $(s_1, a_1) \rightarrow -1 \rightarrow (s_2, a_2)$

$$Q(s_1, a_1) = 0 + 0.1[-1 + 0 - 0] = -0.1$$

(β) $(s_2, a_2) \rightarrow -1 \rightarrow (s_1, a_1)$

$$Q(s_2, a_2) = 0 + 0.1[-1 + (-0.1) - 0] = -0.11$$

(γ) $(s_1, a_1) \rightarrow -1 \rightarrow (s_2, a_1)$

$$Q(s_1, a_1) = -0.1 + 0.1[-1 + 0 - (-0.1)] = -0.19$$

(δ) $(s_2, a_1) \rightarrow -1 \rightarrow (s_4)$ (τερματική)

$$Q(s_2, a_1) = 0 + 0.1[-1 + 0 - 0] = -0.1$$

Επεισόδιο 2:

(α) $(s_2, a_1) \rightarrow -1 \rightarrow (s_1, a_2)$

$$Q(s_2, a_1) = -0.1 + 0.1[-1 + 0 - (-0.1)] = -0.19$$

(β) $(s_1, a_2) \rightarrow -1 \rightarrow (s_1, a_1)$

$$Q(s_1, a_2) = 0 + 0.1[-1 + (-0.19) - 0] = -0.119$$

(γ) $(s_1, a_1) \rightarrow -1 \rightarrow (s_4)$

$$Q(s_1, a_1) = -0.19 + 0.1[-1 + 0 - (-0.19)] = -0.271$$

Τελικός πίνακας Q (SARSA):

Κατάσταση	Ενέργεια	$Q(s,a)$
s_1	a_1	-0.271
s_1	a_2	-0.119
s_2	a_1	-0.19
s_2	a_2	-0.11

Αλγόριθμος Q-Learning:

Η ενημέρωση έχει τη μορφή:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a') - Q(s_t, a_t)]$$

Επεισόδιο 1:

(α) $(s_1, a_1) \rightarrow s_2$

$$Q(s_1, a_1) = 0 + 0.1[-1 + 0 - 0] = -0.1$$

(β) $(s_2, a_2) \rightarrow s_1$

$$Q(s_2, a_2) = 0 + 0.1[-1 + \max(-0.1, 0) - 0] = -0.1$$

(γ) $(s_1, a_1) \rightarrow s_2$

$$Q(s_1, a_1) = -0.1 + 0.1[-1 + \max(0, -0.1) - (-0.1)] = -0.19$$

(δ) $(s_2, a_1) \rightarrow s_4$

$$Q(s_2, a_1) = 0 + 0.1[-1 + 0 - 0] = -0.1$$

Επεισόδιο 2:

(α) $(s_2, a_1) \rightarrow s_1$

$$Q(s_2, a_1) = -0.1 + 0.1[-1 + \max(-0.2, 0) - (-0.1)] = -0.19$$

(β) $(s_1, a_2) \rightarrow s_1$

$$Q(s_1, a_2) = 0 + 0.1[-1 + \max(-0.2, 0) - 0] = -0.1$$

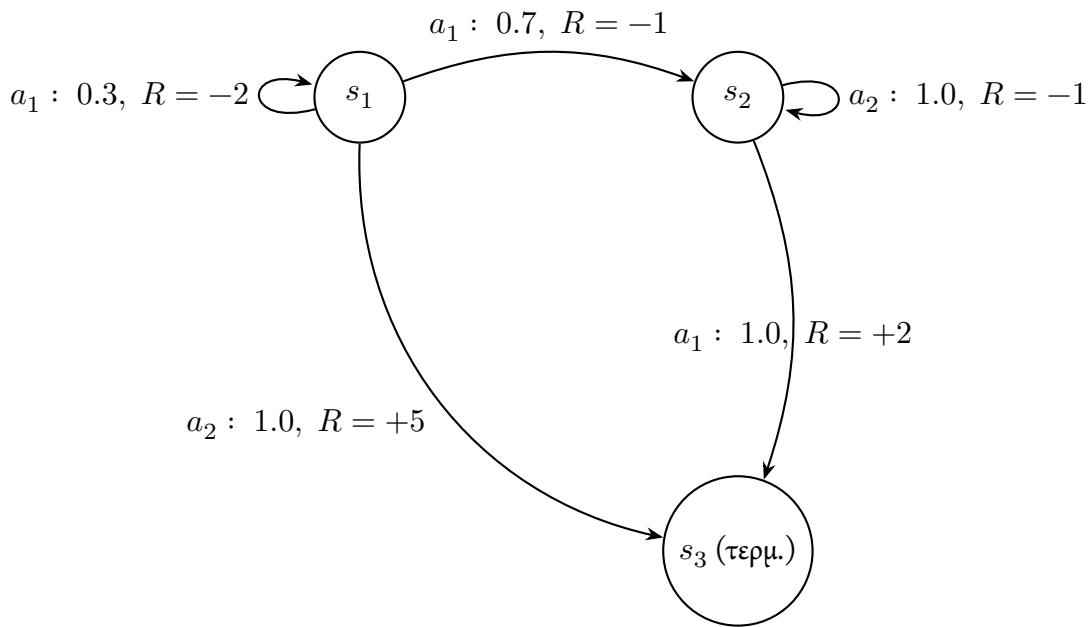
(γ) $(s_1, a_1) \rightarrow s_4$

$$Q(s_1, a_1) = -0.2 + 0.1[-1 + 0 - (-0.2)] = -0.28$$

Τελικός πίνακας Q (Q-Learning):

Κατάσταση	Ενέργεια	$Q(s,a)$
s_1	a_1	-0.28
s_1	a_2	-0.1
s_2	a_1	-0.19
s_2	a_2	-0.1

12. α) Ο παρακάτω γράφος απεικονίζει τη Μαρκοβιανή διαδικασία απόφασης (ΠΜΔΑ) του περιβάλλοντος, με τις πιθανότητες μετάβασης και τις αντίστοιχες ανταμοιβές:



β) Για s_2 (με a_1):

$$V^\pi(s_2) = \mathbb{E}[r + \gamma V(s')] = 1.0 \cdot (2 + 0.9 \cdot 0) = 2$$

Για s_1 (με a_1):

$$\begin{aligned} V^\pi(s_1) &= 0.7 \cdot (-1 + 0.9 \cdot V(s_2)) + 0.3 \cdot (-2 + 0.9 \cdot V(s_1)) \\ &= 0.7(-1 + 0.9 \cdot 2) + 0.3(-2 + 0.9 \cdot V(s_1)) \\ &= 0.7(-1 + 1.8) + 0.3(-2 + 0.9V(s_1)) \\ &= 0.7(0.8) + 0.3(-2 + 0.9V(s_1)) \\ &= 0.56 + 0.3(-2 + 0.9V(s_1)) \\ &= 0.56 - 0.6 + 0.27V(s_1) \\ V^\pi(s_1) &= -0.04 + 0.27V(s_1) \end{aligned}$$

Λύνοντας ως εξίσωση:

$$V^\pi(s_1) - 0.27V(s_1) = -0.04 \Rightarrow 0.73V(s_1) = -0.04 \Rightarrow V(s_1) \approx -0.0548$$

Άρα:

$$V^\pi(s_1) \approx -0.055, \quad V^\pi(s_2) = 2, \quad V(s_3) = 0$$

γ) Υπολογίζουμε την αναμενόμενη απόδοση για κάθε ενέργεια και κατάσταση:

Για s_1 :

- a_1 : όπως παραπάνω $\rightarrow \approx -0.055$
- a_2 : μετάβαση απευθείας σε s_3 με ανταμοιβή +5:

$$Q(s_1, a_2) = 1.0 \cdot (5 + 0.9 \cdot 0) = 5$$

Για s_2 :

- a_1 : όπως παραπάνω $\rightarrow Q(s_2, a_1) = 2$
- a_2 : παραμονή στο s_2 :

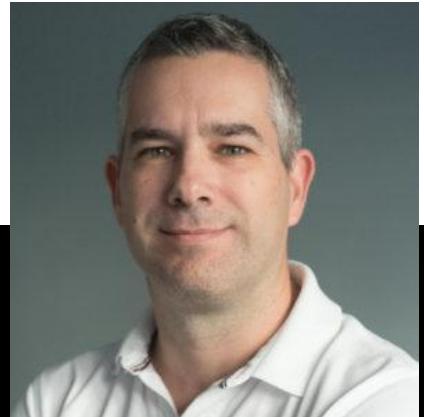
$$Q(s_2, a_2) = -1 + 0.9 \cdot V(s_2) = -1 + 0.9 \cdot 2 = 0.8$$

Συμπέρασμα: Η βελτιωμένη πολιτική είναι $\pi'(s_1) = a_2$ (επιλέγει +5 αντί για αρνητική αναμενόμενη), $\pi'(s_2) = a_1$ (επιλέγει +2 αντί για 0.8)

Εισαγωγή στη Μηχανική Μάθηση

Τι θα μάθουμε από αυτό το βιβλίο:

- Θα αποκτήσουμε μια αρχική ιδέα αναφορικά με την επιστημονική περιοχή της μηχανικής μάθησης,
- Θα μελετήσουμε την οικογένεια των γραμμικών μοντέλων,
- Θα μελετήσουμε την οικογένεια των δενδρικών μοντέλων,
- Θα ασχοληθούμε με την αξιολόγηση μοντέλων μηχανικής μάθησης,
- Θα μελετήσουμε την οικογένεια των μοντέλων που στηρίζονται σε κανόνες,
- Θα παρουσιάσουμε έναν από τους πιο διαδεδομένους αλγορίθμους μηχανικής μάθησης – τον αλγόριθμο των k Πλησιέστερων Γειτόνων,
- Θα μελετήσουμε την περιοχή της μηχανικής μάθησης που ασχολείται με σύνολα από μοντέλα πρόβλεψης,
- και θα εστιάσουμε στην ενισχυτική μάθηση.



Γρηγόριος Τσουμάκας

Καθηγητής Μηχανικής Μάθησης και Ανακάλυψης Γνώσης
Αριστοτελείο Πανεπιστήμιο Θεσσαλονίκης

Ο Γρηγόριος Τσουμάκας είναι Καθηγητής Μηχανικής Μάθησης και Ανακάλυψης Γνώσης στο Τμήμα Πληροφορικής του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης (ΑΠΘ). Έλαβε πτυχίο Πληροφορικής από το ΑΠΘ το 1999, μεταπτυχιακό στην Τεχνητή Νοημοσύνη από το Πανεπιστήμιο του Εδιμβούργου το 2000 και διδακτορικό στην Πληροφορική από το ΑΠΘ το 2005. Η ερευνητική του εμπειρία επικεντρώνεται σε τεχνικές μάθησης με επίβλεψη και στην επεξεργασία φυσικής γλώσσας. Έχει δημοσιεύσει περισσότερα από 150 άρθρα και σύμφωνα με το Google Scholar το έργο του έχει λάβει περισσότερες από 20.000 αναφορές και το h-index του είναι 52. Ο Δρ. Τσουμάκας είναι πρεσβύτερο μέλος των IEEE και ACM και επίτιμο μέλος της Ελληνικής Εταιρίας Τεχνητής Νοημοσύνης. Στις τιμητικές του διακρίσεις περιλαμβάνονται το βραβείο δεκαετούς διάρκειας στον χρόνο το 2017 από το Ευρωπαϊκό Συνέδριο για τη Μηχανική Μάθηση και τις Αρχές και την Πρακτική της Ανακάλυψης Γνώσης από Βάσεις Δεδομένων (ECML PKDD) και το βραβείο καλύτερης εργασίας Marco Ramoni από το 19ο Διεθνές Συνέδριο για την Τεχνητή Νοημοσύνη στην Ιατρική (AIME 2021). Επιπλέον είναι συνιδρυτής και επιστημονικός σύμβουλος της Medoid AI, τεχνοβλαστού του ΑΠΘ που ιδρύθηκε το 2019 και αναπτύσσει λύσεις τεχνητής νοημοσύνης.