ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΣΤΑΤΙΣΤΙΚΗΣ ΚΑΙ ΑΣΦΑΛΙΣΤΙΚΗΣ ΕΠΙΣΤΗΜΗΣ

ΜΟΝΤΕΛΑ ΤΑΞΙΝΟΜΗΣΗΣ ΚΑΙ ΕΦΑΡΜΟΓΕΣ

Γιώργος Ι. Δοντάς

Διατριβή

που υποβλήθηκε στο Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς ως μέρος των απαιτήσεων για την απόκτηση του Μεταπτυχιακού Διπλώματος Ειδίκευσης στην Εφαρμοσμένη Στατιστική

Πειραιάς
Μάιος 2012

UNIVERSITY OF PIRAEUS

DEPARTMENT OF STATISTICS AND INSURANCE SCIENCE

CLASSIFICATION MODELS AND APPLICATIONS

By

George J. Dontas

Thesis

submitted to the Department of Statistics and Insurance Science of the University of Piraeus in partial fulfilment of the requirements for the degree of Master of Science in Applied Statistics

Piraeus, Greece
May 2012

Η παρούσα Διπλωματική Εργασία εγκρίθηκε ομόφωνα από την Τριμελή Εξεταστική Επιτροπή που ορίσθηκε από το Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς στην υπ' αριθμ. …….. συνεδρίασή του σύμφωνα με τον Εσωτερικό Κανονισμό Λειτουργίας του Προγράμματος Μεταπτυχιακών Σπουδών.

Τα μέλη της Επιτροπής ήσαν:

- Καθ. Μ. Κούτρας (Επιβλέπων)
- Καθ. Κλ. Τσίμπος
- Αναπλ. Καθ. Γ. Ηλιόπουλος

Η έγκριση της Διπλωματική Εργασίας από το Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς δεν υποδηλώνει αποδοχή των γνωμών του συγγραφέα.

Στον πατέρα μου

Γιάννη

Ευχαριστίες

Οφείλω ένα μεγάλο ευχαριστώ στη γυναίκα μου και τα παιδιά μου για την υπομονή και την κατανόηση που έδειξαν όλο το χρονικό διάστημα που διήρκεσε το μεταπτυχιακό τμήμα και κυρίως στη σύνταξη αυτής της εργασίας, που με τη συμπαράστασή τους κατάφερα να ολοκληρώσω.

Περίληψη

Η μηχανική μάθηση έχει ως σκοπό τη δημιουργία αλγορίθμων ικανών να βελτιώνουν την απόδοσή τους, αξιοποιώντας προγενέστερη γνώση και εμπειρία, με σκοπό την εξαγωγή χρήσιμων συμπερασμάτων και την περιγραφή φαινομένων, μέσω της επεξεργασίας δεδομένων τεράστιου, πολλές φορές, όγκου. Το ζητούμενο στην περίπτωση της επιβλεπόμενης μάθησης είναι η κατασκευή ενός μοντέλου που αναπαριστά τη γνώση που αποκτήθηκε μέσω της εμπειρίας και το οποίο στη συνέχεια χρησιμοποιείται για την αξιολόγηση νέων παρατηρήσεων. Μία από τις πιο οικείες μεθόδους περιγραφής φαινομένων είναι η ταξινόμηση, η ένταξη δηλαδή κάθε παρατήρησης σε μία ομάδα, από ένα πεπερασμένο πλήθος υποψήφιων ομάδων. Η παρούσα εργασία επικεντρώνεται στην παρουσίαση ενός πολύ διαδεδομένου αλγορίθμου ταξινόμησης, προερχόμενου από τον τομέα της μηχανικής μάθησης, με το όνομα «μηχανή διανυσμάτων υποστήριξης» (Support Vector Machine - SVM). Η ανάπτυξη του θεωρητικού υποβάθρου του αλγορίθμου παρουσιάζεται σταδιακά, ώστε να γίνει κατανοητή από τον αναγνώστη όλη η διαδρομή, από τον πλέον στοιχειώδη αλγόριθμο ταξινόμησης, μέχρι τη βελτιστοποιημένη εκδοχή που αποτελεί η SVM. Στη διαδρομή αυτή, θα παρουσιαστούν εκτενώς δύο ακόμη διαδεδομένοι αλγόριθμοι ταξινόμησης, η «ομαλοποιημένη λογιστική παλινδρόμηση» και το «πολυστρωματικό νευρωνικό δίκτυο». Πέρα από τη θεωρητική παρουσίαση των αλγορίθμων, σκοπός της εργασίας είναι η προγραμματιστική ανάπτυξη αυτών – στις περιπτώσεις που αυτό δε θεωρείται ασύμφορο – για την αντιμετώπιση πρακτικών εφαρμογών, καθώς επίσης και η παρουσίαση του τρόπου χρήσης έτοιμων βιβλιοθηκών και ελεύθερα διαθέσιμων λογισμικών πακέτων. Τα δεδομένα που χρησιμοποιήθηκαν όπως και το σύνολο του κώδικα διατίθενται στον αναγνώστη για πειραματισμό. Η διεργασία της μηχανικής μάθησης δεν μπορεί βέβαια να είναι πλήρης, χωρίς την αξιολόγηση της γνώσης που αποκτάται. Για το λόγο αυτό, στο τελευταίο κεφάλαιο γίνεται μια αναφορά σε διαγνωστικούς ελέγχους και πρακτικές συμβουλές για την αξιολόγηση και βελτιστοποίηση του μοντέλου πριν αυτό τεθεί σε εφαρμογή.

Abstract

The aim of machine learning is to develop algorithms capable of improving their own performance, exploiting existing data, stored in huge databases, in order to discover knowledge and interpret several phenomena. Supervised learning aims in creating a model that takes into account the knowledge adapted by experience, and then uses it for evaluating new observations. One of the most common methods for describing phenomena is through classification, where a particular object is classified to one of several available classes of objects. The present thesis focuses on one of the most promising classification algorithms in the field of machine learning, the "support vector machine" (SVM). The presentation of the theoretical foundation advances gradually, starting from the most intuitive classification algorithm and reaching up to the optimized approach of SVM, so that it's easier for the reader to follow. During the presentation procedure, another two of the most popular classification algorithms are also highlighted: the "regularized logistic regression" and the "multi-layer perceptron". Beyond theoretical approach, this thesis aims developing appropriate algorithms, when possible, or otherwise to suggest how to use "of the shelf" and open-source software libraries. All the data used for the examples, as well as the whole of the implemented code, are available to the reader for experimentation. A machine learning process cannot be considered complete without having evaluated the model developed. For this reason, in the last chapter, we deemed it necessary to present several diagnostic tests and practical advice for model evaluation and optimization.

Περιεχόμενα

Κατάλογος Σχημάτων

Κ Ε Φ Α Λ Α Ι Ο 1

Ταξινόμηση και Μηχανική Μάθηση

1.1 Μηχανική Μάθηση

Οι πρόσφατες εξελίξεις της τεχνολογίας και των υπολογιστών έχουν διευκολύνει τη συλλογή πληθώρας δεδομένων, σε κάθε τομέα ενδιαφέροντος και η δουλειά του στατιστικού έγκειται στο να μπορέσει να τα εκμεταλλευτεί, εξάγοντας χρήσιμα πρότυπα και τάσεις, να αντιληφθεί αυτά που τα δεδομένα «έχουν να του πουν», ώστε εν τέλει να μπορέσει να τα αξιοποιήσει στην υποστήριξη διαδικασιών λήψης αποφάσεων. Η όλη αυτή διεργασία καλείται *«μάθηση μέσω των δεδομένων»*.

Η πρόκληση της μάθησης μέσω των δεδομένων έχει οδηγήσει τις στατιστικές επιστήμες σε μια επανάσταση. Δεδομένου μάλιστα ότι οι υπολογιστικές τεχνικές παίζουν καθοριστικό ρόλο στην επεξεργασία δεδομένων τεράστιου πολλές φορές όγκου, δε δημιουργεί έκπληξη το γεγονός ότι πολλές από τις νέες τεχνικές έχουν αναπτυχθεί στα πλαίσια του γνωστικού πεδίου της «επιστήμης υπολογιστών».

Ο σύγχρονος αυτός κόσμος της πληροφορίας έχει οδηγήσει στην ανάπτυξη υπολογιστικών τεχνικών και εργαλείων, ικανών να χειρίζονται και να αναλύσουν τεράστιους όγκους δεδομένων, μέσω ημιαυτόματων διεργασιών εξαγωγής χρήσιμων πληροφοριών στις οποίες αναφερόμαστε γενικά με τον όρο ***«Εξόρυξη Γνώσης»*** **(*Knowledge Discovery*)**. Με τον όρο ημιαυτόματες διεργασίες εννοούμε υπολογιστικά εργαλεία ανάλυσης, στα οποία όμως δεν παύει να είναι ουσιώδους σημασίας η επίβλεψη από τον αναλυτή, καθώς οι παρεμβάσεις που απαιτούνται από την πλευρά του είναι πολύ δύσκολο να αυτοματοποιηθούν. Για παράδειγμα, μόνο ο αναλυτής είναι σε θέση να αποφανθεί αν η εξαγόμενη πληροφορία έχει κάποια χρησιμότητα ή αν αποτυγχάνει να συνοψίσει τα δεδομένα με τρόπο διεισδυτικό και σαφή. Η μορφή που πολύ συχνά λαμβάνει αυτή η εξαγόμενη πληροφορία είναι επεξηγηματικά μοτίβα,

που αποκαλούνται «μοντέλα». Υπάρχουν διαφόρων ειδών μοντέλα. Για παράδειγμα υπάρχουν μοντέλα με τη μορφή κανόνων if-then-else, ή άλλα που υλοποιούνται με χρήση τεχνητών νευρωνικών δικτύων. Κοινό χαρακτηριστικό όλων αυτών των μοντέλων είναι η παράβλεψη των μη ουσιωδών λεπτομερειών με σκοπό τη σύνοψη των κυρίαρχων τάσεων στα δεδομένα.

Οι κυριότερες κατηγορίες των αλγορίθμων εξόρυξης γνώσης είναι δύο: οι αλγόριθμοι ***«μηχανικής μάθησης»*** **(*machine learning*)** και οι στατιστικές τεχνικές. Οι αλγόριθμοι μηχανικής μάθησης έχουν αναπτυχθεί στα πλαίσια της γνωστικής περιοχής της *«τεχνητής νοημοσύνης»* (*artificial intelligence*) που χρονολογείται από τα τέλη του 1950 και είχαν σχεδιαστεί αρχικά για την ανάπτυξη λογισμικού ικανού να μεταβάλλει αυτόματα τον τρόπο με τον οποίο επιτυγχάνει τους στόχους του (autonomous agent). Οι στατιστικές τεχνικές αναπτύχθηκαν στο πλαίσιο της θεωρίας πιθανοτήτων, στο τέλος του δέκατου ένατου αιώνα. Ωστόσο, μόλις στα τέλη της δεκαετίας του 1980 και στις αρχές της δεκαετίας του 1990 έγινε αντιληπτό ότι οι δύο αυτές επιστημονικές περιοχές προσπαθούν να αντιμετωπίσουν παρόμοια μεταξύ τους προβλήματα.

Με την έλευση της υπολογιστικής στατιστικής, τα σύνορα μεταξύ των δύο κλάδων έχουν σχεδόν εξαφανιστεί και οι στατιστικές τεχνικές που ασχολούνται με την κατασκευή μοντέλων και την εξαγωγή συμπερασμάτων δύσκολα διακρίνονται από αυτές της μηχανικής μάθησης, και το αντίστροφο. Η κυριότερη διαφορά μεταξύ των δύο προσεγγίσεων έχει να κάνει κυρίως με το σύνολο των υποθέσεων που προαπαιτεί κάθε τεχνική. Οι περισσότερες στατιστικές τεχνικές προϋποθέτουν ότι είτε τα ίδια τα δεδομένα, είτε τα σφάλματα μοντελοποίησης ακολουθούν κάποια παραμετρική κατανομή. Από την άλλη μεριά οι αλγόριθμοι μηχανικής μάθησης, σε γενικές γραμμές, δεν κάνουν τέτοιες υποθέσεις και ως εκ τούτου είναι σε θέση να παρέχουν πιο ακριβή μοντέλα, τουλάχιστον στις περιπτώσεις όπου η υπόθεση ύπαρξης κάποιας γνωστής κατανομής που περιγράφει τα δεδομένα δεν ευσταθεί. Και πάλι όμως, νέες μη παραμετρικές υπολογιστικές στατιστικές τεχνικές, όπως η bootstrap, έρχονται να επιτείνουν την ασάφεια στα όρια μεταξύ μηχανικής μάθησης και στατιστικής.

Η τεχνική δημιουργίας μοντέλων στην οποία επικεντρώνεται κατά κύριο λόγο η παρούσα εργασία εντάσσεται στον κλάδο της μηχανικής μάθησης και είναι γνωστή με την ονομασία ***«μηχανή διανυσμάτων υποστήριξης»*** **(*support vector machine - SVM*)**. Εκτενείς αναφορές θα γίνουν όμως και σε δύο ακόμη τεχνικές, ευρέως χρησιμοποιούμενες σε προβλήματα

ταξινόμησης, αυτή των πολυστρωματικών νευρωνικών δικτύων (*multi-layer perceptron*) και της *«ομαλοποιημένης λογιστικής παλινδρόμησης»* (*regularized logistic regression*).

Η μηχανική μάθηση αποτελεί μια περιοχή του πεδίου της τεχνητής νοημοσύνης που σχετίζεται με την εφαρμογή αλγορίθμων μάθησης με σκοπό την αυτόνομη απόκτηση και ενσωμάτωση γνώσης. Ως αποτέλεσμα έχει τη δημιουργία υπολογιστικών συστημάτων ικανών να μαθαίνουν κυρίως μέσω της εμπειρίας και της αναλυτικής παρατήρησης. Οι τεχνικές της μηχανικής μάθησης έχουν υιοθετηθεί από πολλούς επιστημονικούς κλάδους για την αυτοματοποίηση πολύπλοκων διεργασιών λήψης αποφάσεων και επίλυσης προβλημάτων.

Ακόμη όμως και μεταξύ των ειδημόνων στη μηχανική μάθηση, δεν υπάρχει κοινά αποδεκτός ορισμός του τι είναι και τι δεν είναι η μηχανική μάθηση. Ο Arthur Samuel (1959) έδωσε τον επόμενο:

- *Μηχανική Μάθηση: Το επιστημονικό πεδίο όπου δίδεται η δυνατότητα σε υπολογιστές να μαθαίνουν χωρίς να έχουν σαφώς προγραμματιστεί για κάτι τέτοιο.*

Ο Arthur Samuel, πρωτοπόρος στον τομέα της τεχνητής νοημοσύνης και της δημιουργίας λογισμικού παιχνιδιών, είχε κατασκευάσει ένα πρόγραμμα που έπαιζε το γνωστό παιχνίδι «Ντάμα» (Checkers). Το εντυπωσιακό σχετικά μ' αυτό το εγχείρημα ήταν το ότι, επειδή ο ίδιος ο Arthur Samuel δεν ήταν καλός παίχτης ντάμας, είχε την ιδέα να βάλει το πρόγραμμα να παίζει δεκάδες χιλιάδες παιχνίδια εναντίον του εαυτού του και να καταγράφει τις διατάξεις του ταμπλό που τείνουν να οδηγούν σε νικηφόρες παρτίδες και αντίστοιχα αυτές που τείνουν να οδηγούν σε ήττες. Το πρόγραμμα *«έμαθε»* λοιπόν με το πέρασμα του χρόνου ποιες είναι οι προτιμητέες διατάξεις στο ταμπλό και ποιες όχι, με αποτέλεσμα να τα καταφέρνει τελικά στο παιχνίδι της ντάμας πολύ καλύτερα απ' όσο ο δημιουργός του. Αυτό ήταν πράγματι πολύ εντυπωσιακό. Το γεγονός ότι ένας υπολογιστής έχει την υπομονή και ικανότητα να παίξει αναρίθμητα παιχνίδια με τον εαυτό του είναι κάτι που δεν ισχύει με τους ανθρώπους.

Ο ορισμός του Samuel είναι ανεπίσημος και κάπως παρωχημένος πια. Ένας πρόσφατος πιο αυστηρός ορισμός είναι ο επόμενος, ο οποίος προτάθηκε από τον Tom Mitchell (1998):

- *Καλώς ορισμένο πρόβλημα μάθησης: Ένα πρόγραμμα λέγεται ότι μαθαίνει από την εμπειρία **E** σχετικά με κάποια εργασία **T** και κάποιο μέτρο απόδοσης **P**, αν η απόδοσή του στην **T**, όπως μετριέται από το **P**, βελτιώνεται με την εμπειρία **E**.*

Στο προαναφερθέν πρόβλημα εκμάθησης του παιχνιδιού «Ντάμα», η εμπειρία ***E*** προέρχεται από το ότι το πρόγραμμα αναγκάζεται να παίξει δεκάδες χιλιάδες παιχνίδια με αντίπαλο τον εαυτό του. Η εργασία ***T*** είναι το ίδιο το παιχνίδι της ντάμας, ενώ το μέτρο απόδοσης ***P*** είναι η πιθανότητα νίκης στο επόμενο παιχνίδι εναντίον κάποιου νέου αντιπάλου.

Για κάθε φαινόμενο απ' αυτά που συμβαίνουν γύρω μας, του οποίου η συμπεριφορά μπορεί να παρατηρηθεί, το ερώτημα που τίθεται από τη σκοπιά της μηχανικής μάθησης και εν τέλει μπορεί να απαντηθεί μ' ένα ηχηρό «*Ναι*» είναι: «Μπορούν να ανακαλυφθούν και να περιγραφούν μοντέλα αυτής της συμπεριφοράς με τη χρήση ηλεκτρονικών υπολογιστών;»

1.2 Ταξινόμηση

Μια από τις πιο οικείες μεθόδους περιγραφής φαινομένων είναι η ταξινόμηση. Κάθε συγκεκριμένο αντικείμενο, είτε θα ανήκει σε μια κλάση αντικειμένων, είτε όχι. Θα μπορούσε να φανταστεί κανείς ότι υπάρχει μια διεργασία, σχετική με το φαινόμενο, η οποία αποδίδει τον χαρακτηρισμό *αληθές*, στα αντικείμενα που ανήκουν στη υπό εξέταση κλάση και *ψευδές*, σε αυτά που δεν ανήκουν. Στις περισσότερες περιπτώσεις η διαδικασία της ταξινόμησης δεν είναι απλή, καθώς δεν είναι εύκολη η πρόσβαση στη φυσική διεργασία που κατηγοριοποιεί τα αντικείμενα. Συνήθως μπορούμε μόνο να παρατηρήσουμε το αποτέλεσμα αυτής της διεργασίας: τον χαρακτηρισμό που έχει αποδοθεί σε κάθε αντικείμενο.

Ο στόχος της μηχανικής μάθησης είναι η δημιουργία ενός κατάλληλου μοντέλου αυτής της διεργασίας ταξινόμησης, το οποίο να προσεγγίζει την πραγματική διεργασία όσο το δυνατόν καλύτερα. Ο επόμενος ορισμός διατυπώνει πιο αυστηρά την έννοια της ταξινόμησης μέσω της μηχανικής μάθησης:

Ορισμός 1.1 (*Ταξινόμηση στη Μηχανική Μάθηση*) *Δοθέντος:*

- *ενός πληθυσμού δεδομένων* X
- *ενός δείγματος* $S \subset X$
- *μιας συνάρτησης-στόχου* (*διαδικασία ταξινόμησης*) $f : X \to \{true, false\}$
- *ενός ταξινομημένου συνόλου εκπαίδευσης* D, *όπου*
  $D = \{(\overline{x}, y) \mid \overline{x} \in S \text{ και } y = f(\overline{x})\}$

να βρεθεί συνάρτηση $\hat{f} : X \to \{true, false\}$ *κάνοντας χρήση του* D, *τέτοια ώστε*

$$\hat{f}(\overline{x}) \cong f(\overline{x}) \text{ για κάθε } \overline{x} \in X \qquad (1.1)$$

Στον παραπάνω ορισμό, ο πληθυσμός δεδομένων είναι το σύνολο των υπό εξέταση αντικειμένων. Το δείγμα S είναι ένα υποσύνολο του πληθυσμού. Το υποσύνολο αυτό είναι απαραίτητο, δεδομένου ότι στις περισσότερες περιπτώσεις οι πληθυσμοί που μας ενδιαφέρουν τείνουν να είναι πολύ μεγάλου, ή και άπειρου πλήθους και η δημιουργία μοντέλου θα καθίστατο πολύ αργή, ή και αδύνατη. Έτσι το δείγμα S χρησιμοποιείται ως αντιπροσωπευτικό του πληθυσμού, ώστε να καταστεί εφικτή η δημιουργία μοντέλου.

Η συνάρτηση-στόχος f είναι η διαδικασία που ταξινομεί τα αντικείμενα και θεωρείται ότι μπορεί να χαρακτηρίσει το κάθε αντικείμενο του συνόλου X με μια από τις τιμές $\{true, false\}$, εφόσον το αντικείμενο αυτό παρατηρηθεί. Έτσι, αν και δεν έχουμε άμεση πρόσβαση στη διαδικασία αυτή καθαυτή, μπορούμε σε κάθε περίπτωση να παρατηρήσουμε το πώς ταξινομείται από τη διαδικασία κάθε ένα από τα αντικείμενα του πληθυσμού.

Κάνοντας χρήση της πιο πάνω ιδιότητας της συνάρτησης-στόχου, είμαστε σε θέση να κατασκευάσουμε το ταξινομημένο ***«σύνολο εκπαίδευσης»*** (***training set***) D συμπεριλαμβάνοντας μαζί με όλα τα υπόλοιπα χαρακτηριστικά κάθε αντικειμένου του συνόλου S και την κλάση στην οποία αυτό ανήκει.

Μια από τις δύο κυριότερες κατηγορίες της μηχανικής μάθησης είναι η λεγόμενη ***«επιτηρούμενη μάθηση»*** (***supervised learning***), όπου για την κατασκευή του αλγορίθμου γίνεται χρήση ενός ταξινομημένου συνόλου εκπαίδευσης. Γενικότερα, επιτηρούμενη καλείται η διαδικασία μάθησης όποτε καθοδηγείται από μια *μεταβλητή απόκρισης*, είτε αυτή είναι διακριτή, είτε είναι συνεχής. Αντίστοιχα, όταν δε γίνεται χρήση ταξινομημένου συνόλου, ή γενικότερα δε διατίθεται κάποια μεταβλητή απόκρισης, διακριτή ή συνεχής, έχουμε τη ***«μη επιτηρούμενη μάθηση»*** (***unsupervised learning***), όπου ο στόχος είναι κυρίως να περιγραφεί ο τρόπος κατά τον οποίο τα δεδομένα οργανώνονται ή ομαδοποιούνται.

Τελικά, η (1.1) στον πιο πάνω ορισμό υποδηλώνει ότι η μάθηση μπορεί να εκληφθεί ως η διαδικασία υπολογισμού μιας συνάρτησης $\hat{f}$, η οποία αποτελεί μια προσέγγιση (ή ένα μοντέλο) της αρχικής συνάρτησης f, με βάση τις παρατηρήσεις του συνόλου εκπαίδευσης D. Σημειώνεται ότι η χρήση των χαρακτηρισμών $\{true, false\}$ είναι τυχαία. Αυτό που έχει σημασία είναι το σύνολο των κλάσεων να περιλαμβάνει δύο διακριτές τιμές. Θα μπορούσαν ασφαλώς να υπάρξουν και προβλήματα ταξινόμησης με περισσότερες των δύο κλάσεων. Η μόνη διαφορά θα ήταν ότι στο σύνολο τιμών της f και του μοντέλου της, $\hat{f}$, θα έπρεπε να περιλαμβάνεται αντίστοιχος αριθμός διακριτών χαρακτηρισμών.

Αφού κατασκευαστεί το μοντέλο $\hat{f}$, μπορεί πια να χρησιμοποιηθεί για την πρόβλεψη της κλάσης ενός νέου αντικειμένου από τον πληθυσμό X, όταν αυτή δεν είναι γνωστή. Μπορεί επιπλέον η μορφή του να μας δώσει και κάποια ιδέα για την ίδια τη φύση της πραγματικής διαδικασίας ταξινόμησης f. Στην ουσία προσπαθούμε μέσω της $\hat{f}$ να γενικεύσουμε τα συμπεράσματα που προκύπτουν από το σύνολο εκπαίδευσης D, στον πληθυσμό X. Για το λόγο αυτό η διαδικασία καλείται *«επαγωγική μάθηση»* (*inductive learning*). Είναι προφανές ότι, όσο περισσότερο αντιπροσωπευτικό του πληθυσμού είναι το σύνολο εκπαίδευσης, τόσο καλύτερα αποτελέσματα θα δίνει το μοντέλο.

Θα πρέπει να διευκρινιστεί ότι κάθε αντικείμενο του συνόλου S περιγράφεται από μια σειρά χαρακτηριστικών ή ανεξάρτητων μεταβλητών (attributes) και όταν λέμε ότι εφαρμόζουμε μια συνάρτηση ταξινόμησης σ' ένα αντικείμενο, εννοούμε προφανώς ότι την εφαρμόζουμε στο σύνολο των χαρακτηριστικών αυτού του αντικειμένου. Έτσι, το σύνολο S μπορεί να θεωρηθεί ως ένα υποσύνολο του Καρτεσιανού γινομένου των χαρακτηριστικών, ενώ το X ως γνήσιο υποσύνολο αυτού του Καρτεσιανού γινομένου.

Η μορφή που έχει συνήθως το σύνολο εκπαίδευσης είναι αυτή ενός «*πίνακα δεδομένων*» (*data table*) με γραμμές όσες και τα αντικείμενα του συνόλου και στήλες όσες τα χαρακτηριστικά που διατίθενται για κάθε αντικείμενο. Συνηθίζεται, στον ίδιο πίνακα δεδομένων, να συμπεριλαμβάνεται και η τάξη του κάθε αντικειμένου ως ένα επιπλέον χαρακτηριστικό (στήλη).

1.3 Μορφή του Μοντέλου

Οι μορφή που λαμβάνει η προσεγγιστική συνάρτηση (μοντέλο) $\hat{f}$ είναι συνήθως δύο ειδών:

1) Διαφανής διατύπωση (Transparent representation), π.χ.
   α) Κανόνες if-then-else
   β) Δέντρα αποφάσεων
2) Αδιαφανής διατύπωση (Nontransparent representation), π.χ.
   α) Τα βάρη στις συνάψεις μεταξύ των νευρώνων ενός τεχνητού νευρωνικού δικτύου.
   β) Ο γραμμικός συνδυασμός των διανυσμάτων σε μια μηχανή διανυσμάτων υποστήριξης.

Τα διαφανή μοντέλα αποτελούν διατυπώσεις οι οποίες μπορούν απευθείας να γίνουν αντιληπτές από τον άνθρωπο. Για παράδειγμα μπορούμε να αντιληφθούμε το νόημα ενός κανόνα if-then-else απλά διαβάζοντάς τον. Από την άλλη μεριά, ποτέ δε θα καταφέρουμε να αντιληφθούμε πλήρως το πώς ένα νευρωνικό δίκτυο αποθηκεύει την αποκτηθείσα γνώση, όσο και να εξετάζουμε τα βάρη που έχουν αποδοθεί στις συνάψεις του.

Η μορφή της προσεγγιστικής συνάρτησης είναι σημαντική γιατί από αυτήν εξαρτάται το πόσο καλά μπορεί να μοντελοποιηθεί η εκάστοτε συνάρτηση στόχος. Θεωρήστε για παράδειγμα ότι χρησιμοποιούμε ως μοντέλο τον ίδιο τον πίνακα δεδομένων. Ένα τέτοιο μοντέλο θα έχει τέλεια πληροφόρηση για τα αντικείμενα που περιλαμβάνονται στον πίνακα, ενώ θα είναι αδύνατον να παράσχει οποιαδήποτε εποικοδομητική πληροφορία για αντικείμενα εκτός του πίνακα. Το μοντέλο δηλαδή, *δεν μπορεί να γενικευθεί* πέραν των αντικειμένων του πίνακα δεδομένων και συνεπώς αποτελεί φτωχή επιλογή.

Στο άλλο άκρο βρίσκεται ένα μοντέλο που αποτελείται μόνο από μια σταθερά. Σε οποιοδήποτε αντικείμενο κι αν το χρησιμοποιήσουμε θα δίνει πάντα την ίδια σταθερά ως απόκριση. Αν έχουμε επιλέξει ως σταθερά την κλάση που αποτελεί την πλειοψηφία στο σύνολο εκπαίδευσης και αν το σύνολο αυτό είναι αντιπροσωπευτικό του πληθυσμού, τότε μπορούμε να πούμε ότι αυτό το απλοϊκό μοντέλο θα έχει αντίστοιχη επιτυχία στις προβλέψεις του για τα αντικείμενα του πληθυσμού με αυτή που έχει και στο σύνολο εκπαίδευσης, είναι δηλαδή, έως ένα βαθμό, *ικανό να γενικευθεί*.

Μοντέλα όπως τα δέντρα αποφάσεων, τα νευρωνικά δίκτυα και οι μηχανές διανυσμάτων υποστήριξης εμπίπτουν μεταξύ των δύο παραπάνω ακραίων περιπτώσεων. Έχει μάλιστα παρατηρηθεί ότι σε γενικές γραμμές τα μοντέλα με διαφανή διατύπωση υστερούν σε απόδοση σε σχέση με τα αδιαφανή. Η απαίτηση από ένα μοντέλο να είναι απευθείας ερμηνεύσιμο από τον άνθρωπο περιορίζει τη διαδικασία μοντελοποίησης, με αποτέλεσμα ένα διαφανές μοντέλο να μην είναι σε θέση να χαρακτηρίσει ορισμένα φαινόμενα τόσο αποτελεσματικά όσο ένα αδιαφανές.

1.4 Λογισμικά Εξόρυξης Γνώσης που θα χρησιμοποιηθούν

Όλα τα λογισμικά που θα χρησιμοποιηθούν στην παρούσα εργασία διατίθενται ελεύθερα στο διαδίκτυο. Είναι λοιπόν πολύ εύκολο να αποκτήσει κανείς πρόσβαση σε εκτενέστατα εγχειρίδια χρήσης και παραδείγματα, προκειμένου να εντρυφήσει στις δυνατότητες του

καθενός από αυτά. Στη συνέχεια γίνεται μόνο μια περιληπτική αναφορά στο καθένα απ' αυτά, καθώς είναι προφανές πως η όποια εκτενέστερη παρουσίαση είναι εκτός των σκοπών της παρούσας εργασίας.

1.4.1 Octave / Matlab

Η Octave (www.octave.org) είναι ένα πολύ ισχυρό προγραμματιζόμενο λογισμικό περιβάλλον, ανοικτού κώδικα, για αριθμητικούς υπολογισμούς και παραγωγή γραφικών. Ειδικότερα δε, ενδείκνυται για υπολογισμούς που σχετίζονται με πράξεις γραμμικής άλγεβρας. Η επικοινωνία με το λογισμικό γίνεται μέσω της γλώσσας προγραμματισμού που παρέχει, η οποία επιτρέπει στο σύστημα να επεκτείνει περεταίρω τις δυνατότητές του. Διατίθενται εκδόσεις για τα λειτουργικά συστήματα Linux, MAC OS X, Sun Solaris και Windows.

Η μεγάλη συμβατότητα της γλώσσας Octave με αυτήν του εμπορικού πακέτου MATLAB (www.mathworks.com), το οποίο χρησιμοποιείται σε μεγάλο βαθμό τόσο στη βιομηχανία όσο και ακαδημαϊκά, δίνει στο χρήστη την ευκαιρία να εξοικειωθεί με το συντακτικό και να αντιληφθεί τις δυνατότητες του MATLAB, όταν περιορισμοί χρηματοδότησης ή δικαιωμάτων δεν επιτρέπουν τη χρήση εμπορικών πακέτων.

Για την ανάπτυξη αλγορίθμων ταξινόμησης στη παρούσα εργασία, θα χρησιμοποιηθεί η γλώσσα Octave.

1.4.2 R Project

Το R Project (www.r-project.org) είναι ένα επίσης ελεύθερα διαθέσιμο προγραμματιστικό περιβάλλον, προσανατολισμένο περισσότερο στην επεξηγηματική ανάλυση δεδομένων, στην εφαρμογή διαφόρων στατιστικών μοντέλων και στην παραγωγή γραφικών. Η υποστήριξή του γίνεται μέσω της εθελοντικής συνεισφοράς πολλών ανθρώπων ανά τον κόσμο, οι οποίοι είναι και υπεύθυνοι για την ανάπτυξή του.

Η επικοινωνία με το πακέτο γίνεται κι εδώ αποκλειστικά μέσω της παρεχόμενης γλώσσας προγραμματισμού. Η γλώσσα αυτή, σε συνδυασμό με την ευέλικτη μηχανή γραφικών, δίνει τη δυνατότητα στον αναλυτή να παράγει εντυπωσιακά γραφήματα. Η ισχύς όμως αυτή παρέχεται με το αντίτιμο της όχι αμελητέας προσπάθειας για την εκμάθηση της γλώσσας και των δυνατοτήτων που αυτή παρέχει. Η διαθέσιμη βιβλιοθήκη συναρτήσεων είναι τεράστια και δεν παύει συνεχώς να διευρύνεται, με την προσθήκη όλο και περισσότερων νέων πακέτων

(packages), που διευκολύνουν την αντιμετώπιση όλο και μεγαλύτερης γκάμας προβλημάτων[1].

Στην R υποστηρίζονται ικανοποιητικά οι πράξεις μεταξύ πινάκων (αν και η βιβλιοθήκη συναρτήσεων γραμμικής άλγεβρας είναι σαφώς υποδεέστερη αυτής της Octave/Matlab), όπως και μεταξύ διανυσματικών και βαθμωτών μεγεθών. Το αντικείμενο που χρησιμοποιείται κυρίως για την αποθήκευση δεδομένων είναι το *dataframe*, το οποίο έχει τη μορφή πίνακα, με τη δυνατότητα όμως κάθε στήλη του να περιλαμβάνει μεταβλητές διαφορετικών μεταξύ τους τύπων δεδομένων (αριθμητικών, κατηγορικών, ημερομηνιών κλπ.).

Στα πλαίσια της παρούσας εργασίας θα παρουσιαστεί κυρίως ο τρόπος χρήσης κάποιων από τις διαθέσιμες συναρτήσεις που υλοποιούν τους αλγορίθμους που θα αναφερθούν.

Το R Project διατίθενται σε εκδόσεις για τα λειτουργικά συστήματα Linux, MAC OS X και Windows.

1.4.3 WEKA

Το WEKA (http://www.cs.waikato.ac.nz/ml/weka/) είναι ένα πακέτο που, σε αντίθεση με τα δύο προηγούμενα, προσφέρει γραφικό περιβάλλον επικοινωνίας με το χρήστη (GUI). Περιλαμβάνει μια πλούσια συλλογή αλγορίθμων εξόρυξης γνώσης, γραμμένων σε γλώσσα Java, από το πανεπιστήμιο του Waikato της Νέας Ζηλανδίας. Διατίθεται δε και αυτό ελεύθερα στο διαδίκτυο, σε εκδόσεις για Windows, Linux και Macintosh. Τα αρχικά του σημαίνουν Waikato Environment for Knowledge Analysis[2].

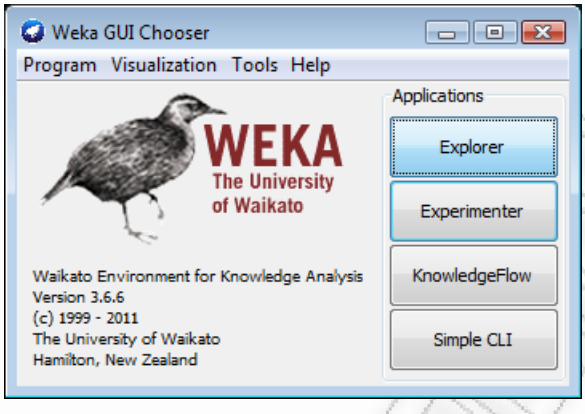
Αν και το γραφικό περιβάλλον του WEKA –που μάλιστα είναι περισσότερα του ενός– είναι ιδιαίτερα φιλικό, σε περιπτώσεις δεδομένων μεγάλου όγκου, ή όταν γίνεται χρήση υπολογιστικών πόρων απόμακρων δικτύων με τη μορφή υπηρεσίας (cloud computing), ενδείκνυται να γίνεται χρήση των αλγορίθμων που παρέχει μέσω της γραμμής εντολών. Αυτό γιατί αφενός έτσι απαιτείται σημαντικά λιγότερη μνήμη και αφετέρου γιατί μέσω της γραμμής εντολών παρέχεται επιπλέον λειτουργικότητα από αυτή που είναι διαθέσιμη μέσω του γραφικού περιβάλλοντος.

---

[1] βλ. http://cran.r-project.org/web/views/

[2] *weka* λέγεται επίσης ένα πουλί που συναντάται μόνο στα νησιά της Νέας Ζηλανδίας, το οποίο δεν μπορεί να πετάξει αλλά του αρέσει να εξερευνά.

Σχήμα 1-1
Η αρχική οθόνη του WEKA



Η αρχική οθόνη της εφαρμογής (Σχήμα 1-1) περιλαμβάνει τέσσερα κουμπιά που εκκινούν αντίστοιχα τις βασικές εφαρμογές που υποστηρίζει η πλατφόρμα:

- ***Explorer***: Το κυρίως περιβάλλον εξερεύνησης των δεδομένων.
- ***Experimenter***: Ένα περιβάλλον εκτέλεσης πειραμάτων και στατιστικών ελέγχων μεταξύ των συστημάτων μάθησης.
- ***KnowledgeFlow:*** Υποστηρίζει ουσιαστικά τις ίδιες λειτουργίες με τον Explorer αλλά με τη μορφή του εξαρχής καθορισμού (μέσω drag and drop) όλων των ενεργειών που θα εκτελεστούν επί των δεδομένων και της σύνδεσης μεταξύ τους, ώστε να προκύπτει μια συγκεκριμένη σειρά εκτέλεσης ενεργειών. Το σύνολο των ενεργειών και η παραγωγή των επιθυμητών αποτελεσμάτων και γραφημάτων, εκτελείται κατόπιν με το πάτημα ενός κουμπιού. Η διαδικασία αυτή απλουστεύει την επανεκτέλεση της ίδιας ακριβώς διαδοχής ενεργειών επί διαφορετικών σετ δεδομένων με κοινά χαρακτηριστικά, μιας και μπορεί να αποθηκευτεί. Οι ταξινομητές στο KnowledgeFlow υποστηρίζουν επιπλέον τη δυνατότητα επεξεργασίας των ανεξάρτητων cross-validation επαναλήψεων εν παραλλήλω[1] (multithreading). Ένα ακόμη πλεονέκτημα είναι ότι υποστηρίζονται και αλγόριθμοι εξελικτικής μάθησης (incremental learning).

[1] Π.χ. σε ένα 4-πύρηνο επεξεργαστή, δίνοντας τη τιμή 4 στο πεδίο “Execution slots”, κάθε πυρήνας εκτελεί και από μια επανάληψη (fold) για το cross-validation.

- ***SimpleCLI***: Παρέχει μια απλή διασύνδεση γραμμής εντολών που επιτρέπει την άμεση εκτέλεση των εντολών WEKA για λειτουργικά συστήματα που δεν παρέχουν το δικό τους περιβάλλον γραμμής εντολών.

Στην παρούσα εργασία θα χρησιμοποιηθεί μόνο η εφαρμογή Explorer από την πλατφόρμα εφαρμογών WEKA.

ΚΕΦΑΛΑΙΟ 2

Γραμμικές Επιφάνειες και Συναρτήσεις Αποφάσεων

2.1 Εισαγωγή

Ένα από τα βασικά ερωτήματα στα προβλήματα *δυαδικής ταξινόμησης* (*binary classification*) είναι το κατά πόσον τα αντικείμενα των δύο ομάδων είναι διαχωρίσιμα. Η πλέον προφανής προσέγγιση στην απάντηση αυτού του ερωτήματος είναι η κατασκευή μιας γραμμής, ή επιπέδου, ή υπερεπιπέδου (ανάλογα με τη διάσταση του χώρου του συνόλου εκπαίδευσης) που διαχωρίζει με τον καλύτερο δυνατό τρόπο τις δύο ομάδες. Στη περίπτωση της δυαδικής ταξινόμησης το υπερεπίπεδο αυτό καλείται *«γραμμική επιφάνεια απόφασης»* (*linear decision surface*). Η διαδικασία κατασκευής μιας τέτοιας επιφάνειας και η χρήση της για την ταξινόμηση άλλων σημείων του πληθυσμού μπορεί να θεωρηθεί επαγωγική μάθηση, αφού, με βάση τη διαχωρισιμότητα των αντικειμένων του συνόλου εκπαίδευσης, γενικεύουμε, εφαρμόζοντας την επιφάνεια απόφασης για την ταξινόμηση αντικειμένων από ολόκληρο τον πληθυσμό.

Είναι μάλιστα βολικότερο να χρησιμοποιούνται συναρτήσεις απόφασης αντί των επιφανειών. Οι συναρτήσεις αυτές μπορούν να θεωρηθούν ως προσεγγίσεις (ή μοντέλα) της αρχικής συνάρτησης στο πρόβλημα ταξινόμησης, όπως έχει ήδη τεθεί.

Έστω λοιπόν ένα πρόβλημα δυαδικής ταξινόμησης, όπου τα αντικείμενα έχουν χαρακτηριστεί ως $+1$ και -1. Θεωρούμε επίσης ότι όλα τα χαρακτηριστικά των αντικειμένων περιγράφονται από πραγματικούς αριθμούς. Κατ' αυτόν τον τρόπο, μπορούμε να αντιμετωπίσουμε κάθε αντικείμενο του πληθυσμού, ως ένα διάνυσμα θέσης ενός n-διάστατου χώρου εφοδιασμένου με εσωτερικό γινόμενο (dot product space) $\mathbb{R}^n$, όπου n ο αριθμός των

διαθέσιμων χαρακτηριστικών. Στα πλαίσια όσων αναφέρθηκαν προηγουμένως, το πρόβλημα μπορεί να τεθεί ως εξής:

Έστω:

- ότι ο χώρος με εσωτερικό γινόμενο $\mathbb{R}^n$ αποτελεί τον πληθυσμό δεδομένων, με αντικείμενα τα διανύσματα $\overline{x} \in \mathbb{R}^n$,
- ένα δείγμα $S \subset \mathbb{R}^n$,
- η συνάρτηση-στόχος $f : \mathbb{R}^n \to \{+1, -1\}$,
- και το ταξινομημένο σύνολο εκπαίδευσης $D = \{(\overline{x}, y) \mid \overline{x} \in S, y = f(\overline{x})\}$.

Να βρεθεί συνάρτηση $\hat{f} : \mathbb{R}^n \to \{+1, -1\}$ κάνοντας χρήση του D, τέτοια ώστε:

$$\hat{f}(x) \cong f(x) \text{ για κάθε } \overline{x} \in \mathbb{R}^n \qquad (2.1)$$

Προκειμένου να υπολογίσουμε τη συνάρτηση $\hat{f}$ η οποία αποτελεί μια προσέγγιση (ή ένα μοντέλο) της αρχικής συνάρτησης f, θα κατασκευάσουμε ένα υπερεπίπεδο που να διαχωρίζει τις κλάσεις $+1$ και -1 όσο το δυνατόν καλύτερα. Αν δεν υφίσταται υπερεπίπεδο τέτοιο ώστε να διαχωρίζει το σύνολο εκπαίδευσης, δεν μπορούμε να κατασκευάσουμε συνάρτηση απόφασης για το δοθέν πρόβλημα.

2.2 Γραμμικές Επιφάνειες Απόφασης Διερχόμενες από την Αρχή των Αξόνων

Έστω ότι ο πληθυσμός δεδομένων είναι ο $\mathbb{R}^2$ και διαθέτουμε ένα γραμμικώς διαχωρίσιμο σύνολο εκπαίδευσης. Αυτό σημαίνει ότι υπάρχει ευθεία που διαχωρίζει τέλεια τις δύο κλάσεις του συνόλου εκπαίδευσης. Υποθέτουμε αρχικά ότι η ευθεία αυτή, έστω g, διέρχεται από την αρχή των αξόνων, δηλαδή ότι η g είναι της μορφής:

$$g(\overline{x}) = w_1 x + w_2 y = 0$$

ή, αν αντιμετωπίσουμε το μεσαίο σκέλος ως εσωτερικό γινόμενο:

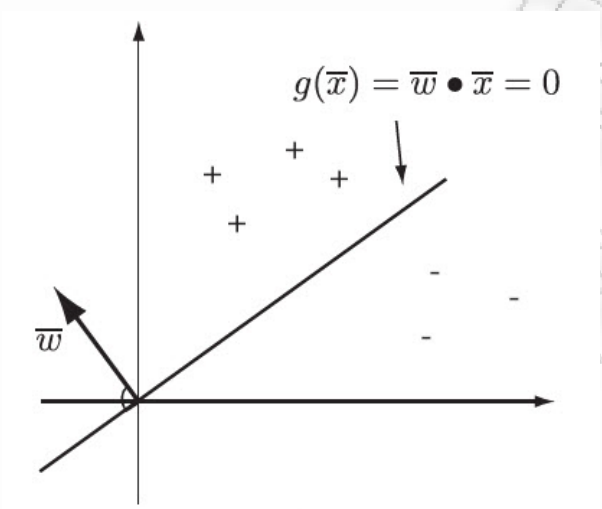
$$g(\overline{x}) = \overline{w} \bullet \overline{x} = 0 \qquad (2.2)$$

όπου $\overline{w} = (w_1, w_2)$, $\overline{x} = (x, y)$.

Η g διαχωρίζει πλήρως τα δεδομένα και καλείται επιφάνεια απόφασης. Το γεγονός, μάλιστα, ότι το εσωτερικό γινόμενο μεταξύ των δύο διανυσμάτων είναι μηδέν, συνεπάγεται

ότι τα διανύσματα $\overline{x}$ και $\overline{w}$ είναι μεταξύ τους ορθογώνια. Το Σχήμα 2-1 δείχνει την επιφάνεια απόφασης του $\mathbb{R}^2$, που διαχωρίζει τα σημεία του δείγματος εκπαίδευσης (με σύμβολα + και − αντίστοιχα).

Σχήμα 2-1
Επιφάνεια απόφασης διερχόμενη από την αρχή των αξόνων



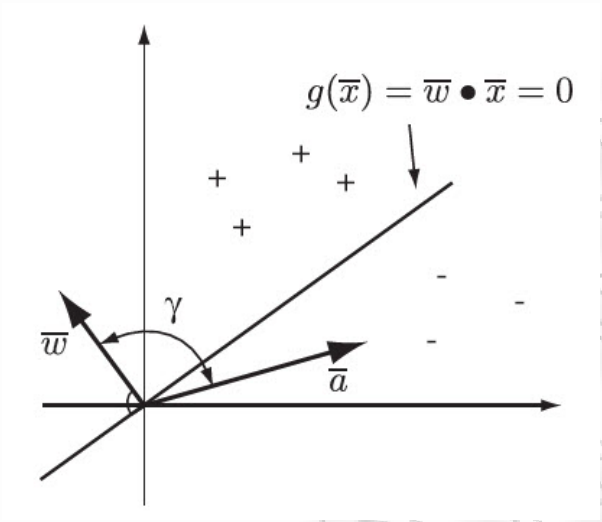
Παρατηρήστε ότι το κάθετο διάνυσμα $\overline{w}$ δείχνει προς την κατεύθυνση των σημείων με την ένδειξη $+$. Λέμε ότι τα σημεία αυτά είναι *«επάνω»* από την επιφάνεια απόφασης, ενώ αντίστοιχα τα «$-$» είναι *«κάτω»*.

Έχοντας ορίσει την επιφάνεια απόφασης, μπορούμε να τη χρησιμοποιήσουμε για να ταξινομήσουμε οποιοδήποτε νέο σημείο του πληθυσμού, ανάλογα με το αν βρίσκεται πάνω ή κάτω από την επιφάνεια απόφασης. Θεωρήστε για παράδειγμα ένα σημείο $\overline{a} \in \mathbb{R}^2$, με $\overline{a} \notin S$. Ανάλογα με το αν βρίσκεται πάνω ή κάτω από την επιφάνεια απόφασης, θα του δοθεί και ο χαρακτηρισμός $+$ και $-$ αντίστοιχα. Ο λόγος για τον οποίο άλλωστε το κάθετο διάνυσμα $\overline{w}$ επιλέγεται ώστε να δείχνει προς τη μεριά των σημείων με το χαρακτηρισμό $+1$, είναι επειδή βοηθά στην άμεση προσήμανση του νέου σημείου $\overline{a}$. Όταν εφαρμόζουμε την εξίσωση (2.2) στο σημείο $\overline{a}$ έχουμε:

$$g(\overline{a}) = \overline{w} \bullet \overline{a} = |\overline{w}||\overline{a}| cos(\gamma) = k \qquad (2.3)$$

όπου γ είναι η γωνία μεταξύ των διανυσμάτων $\overline{w}$ και $\overline{a}$ (βλ. Σχήμα 2-2). Προφανώς το k είναι θετικό αν το σημείο $\overline{a}$ βρίσκεται πάνω από την επιφάνεια ($\gamma \leq 90°$) και αρνητικό αν το $\overline{a}$ βρίσκεται από κάτω ($\gamma > 90°$).

Σχήμα 2-2
Ταξινόμηση σημείου $\overline{a}$ με χρήση της επιφάνειας απόφασης



Με βάση τα παραπάνω μπορούμε να κατασκευάσουμε μια *συνάρτηση απόφασης* $\hat{f}$ ως εξής:

$$\hat{f}(\overline{x}) = \begin{cases} +1 & \text{αν } g(\overline{x}) \geq 0 \\ -1 & \text{αν } g(\overline{x}) < 0 \end{cases} \tag{2.4}$$

για κάθε $\overline{x} \in \mathbb{R}^2$. Καθώς η $\hat{f}$ ταξινομεί κάθε σημείο του πληθυσμού, μπορεί να θεωρηθεί ως προσέγγιση της αρχικής συνάρτησης f. Έχουμε λοιπόν μια περίπτωση επαγωγικής μάθησης, υπό την έννοια ότι θεωρούμε το δείγμα του συνόλου εκπαίδευσης ως αντιπροσωπευτικό ολόκληρου του πληθυσμού.

2.3 Μετατοπισμένες Γραμμικές Επιφάνειες Απόφασης

Στη γενική περίπτωση η επιφάνεια απόφασης έχει τη μορφή:

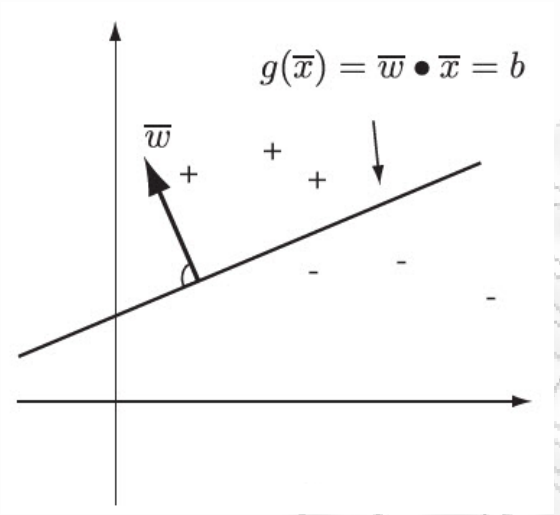
$$g(\overline{x}) = \overline{w} \bullet \overline{x} = b. \tag{2.5}$$

Και πάλι θεωρούμε το πρόβλημα δυαδικής ταξινόμησης, όπου τα αντικείμενα έχουν χαρακτηριστεί ως $+1$ και -1 και είναι τέλεια διαχωρίσιμα (βλ. Σχήμα 2-3).

Η μόνη διαφορά είναι ότι τώρα, για ένα νέο σημείο $\overline{a} \in \mathbb{R}^2$, η ταξινόμηση δεν μπορεί να γίνει απλά βάσει της τιμής που λαμβάνει το $\overline{w} \bullet \overline{a}$ επειδή η επιφάνεια απόφασης δε διέρχεται από την αρχή των αξόνων. Στη περίπτωση αυτή εργαζόμαστε ως εξής.

Σχήμα 2-3

Επιφάνεια απόφασης μη διερχόμενη από την αρχή των αξόνων



Επιλέγουμε τυχαίο σημείο, έστω $\overline{c}$ της επιφάνειας απόφασης:

$$g(\overline{c}) = \overline{w} \bullet \overline{c} = b \tag{2.6}$$

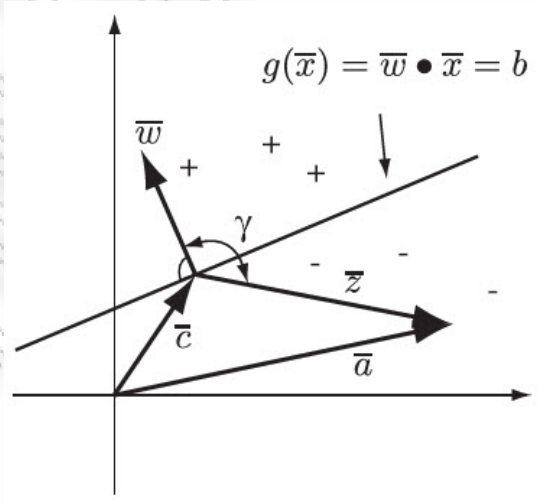
και ορίζουμε ένα διάνυσμα $\overline{z}$ τέτοιο ώστε $\overline{a} = \overline{c} + \overline{z}$ ή

$$\overline{z} = \overline{a} - \overline{c}. \tag{2.7}$$

Τέλος, θέτουμε το κάθετο διάνυσμα $\overline{w}$ έτσι ώστε η αρχή του να είναι στο σημείο $\overline{c}$. Η διάταξη αυτή παρουσιάζεται στο Σχήμα 2-4.

Σχήμα 2-4

Ταξινόμηση σημείου $\overline{a}$ με χρήση της επιφάνειας απόφασης



Αν θεωρήσουμε το $\bar{z}$ ως διάνυσμα θέσης του $\bar{a}$ ως προς το $\bar{c}$, το σημείο $\bar{c}$ μπορεί να χρησιμοποιηθεί όπως η αρχή των αξόνων στην περίπτωση που η επιφάνεια απόφασης περνούσε από αυτήν. Αναλυτικότερα το εσωτερικό γινόμενο

$$\bar{w}\bullet\bar{z} = \left|\bar{w}\right|\left|\bar{z}\right| cos\left(\gamma\right) = k \tag{2.8}$$

δίνει k θετικό αν το σημείο $\bar{a}$ βρίσκεται πάνω από την επιφάνεια ($\gamma \le 90°$) και αρνητικό αν το $\bar{a}$ βρίσκεται κάτω από αυτήν ($\gamma > 90°$).

Αντικαθιστώντας τις (2.5), (2.6) και (2.7) στην (2.8) έχουμε:

$$\bar{w}\bullet\bar{z} = \bar{w}\bullet\left(\bar{a}-\bar{c}\right) = \bar{w}\bullet\bar{a}-\bar{w}\bullet\bar{c} = \bar{w}\bullet\bar{a}-b = g\left(\bar{a}\right)-b.$$

Συνεπώς, το απαραίτητο, για τη διαδικασία ταξινόμησης, εσωτερικό γινόμενο σχετικά με το σημείο $\bar{a}$ μπορεί να υπολογιστεί εφαρμόζοντας τη g στο $\bar{a}$ και κατόπιν αφαιρώντας τη ***«μετάθεση»*** (***offset***) b [1]. Παρατηρήστε ότι το βοηθητικό σημείο $\bar{c}$ δεν είναι πλέον απαραίτητο.

Είμαστε λοιπόν και πάλι σε θέση να κατασκευάσουμε μια *συνάρτηση απόφασης* $\hat{f}$ ως εξής:

$$\hat{f}\left(\bar{x}\right) = \begin{cases} +1 & \text{αν } g\left(\bar{x}\right)-b \ge 0 \\ -1 & \text{αν } g\left(\bar{x}\right)-b < 0 \end{cases} \tag{2.9}$$

για κάθε $\bar{x} \in \mathbb{R}^2$. Για επιφάνειες απόφασης που διέρχονται από την αρχή των αξόνων, η (2.9) ανάγεται στην (2.4), η οποία μπορεί να θεωρηθεί έτσι ως ειδική περίπτωση.

Αν και για λόγους εποπτείας, περιοριστήκαμε στο χώρο $\mathbb{R}^2$, η παραπάνω διαδικασία μπορεί ομοίως να εφαρμοστεί σε n-διάστατους χώρους και να παραχθεί μια συνάρτηση απόφασης της μορφής:

$$\hat{f}\left(\bar{x}\right) = \text{sgn}\left(\bar{w}\bullet\bar{x}-b\right) \tag{2.10}$$

όπου $\bar{w},\bar{x} \in \mathbb{R}^n,\ b \in \mathbb{R}$ και

$$\text{sgn}\left(k\right) = \begin{cases} +1 & \alpha\nu\ k \ge 0 \\ -1 & \alpha\nu\ k < 0 \end{cases} \tag{2.11}$$

για κάθε $k \in \mathbb{R}$, η *βηματική συνάρτηση*.

---

[1] Παρατηρήστε ότι η *«αποτέμνουσα»* (intercept) ισούται με b/w_2

2.4 Απλός Αλγόριθμος Μάθησης

Ας δούμε τώρα έναν αλγόριθμο ο οποίος *κατασκευάζει* την επιφάνεια απόφασης. Δοθέντος ενός γραμμικώς διαχωρίσιμου συνόλου εκπαίδευσης[1] D, όπου:

$$D=\left\{\left(\bar{x}_1,y_1\right),\left(\bar{x}_2,y_2\right),\ldots,\left(\bar{x}_l,y_l\right)\right\}, \tag{2.12}$$

με $\bar{x}_i\in\mathbb{R}^2$ και $y_i\in\{+1,-1\}$, θα πρέπει πρώτα να κατασκευάσουμε μια επιφάνεια απόφασης της μορφής (2.5), υπολογίζοντας το διάνυσμα $\bar{w}$ και τη μετάθεση b με τη βοήθεια του συνόλου εκπαίδευσης και στη συνέχεια μια συνάρτηση απόφασης της μορφής (2.10).

Βήμα 1 Ξεκινάμε υπολογίζοντας τα κέντρα βάρους των δύο κλάσεων, τα οποία δεν είναι τίποτε άλλο από τα διανύσματα των μέσων τιμών, για όλες τις παρατηρήσεις κάθε κλάσης. Ορίζουμε το κέντρο βάρους για την κλάση $+1$ ως $\bar{c}_+$ και αντίστοιχα για την κλάση -1 ως $\bar{c}_-$. Αυτά υπολογίζονται ως εξής:

$$\bar{c}_+=\frac{1}{l_+}\sum_{(\bar{x}_i,+1)\in D}\bar{x}_i \tag{2.13}$$

$$\bar{c}_-=\frac{1}{l_-}\sum_{(\bar{x}_i,-1)\in D}\bar{x}_i \tag{2.14}$$

όπου τα

$$l_+=\left|\left\{\left(\bar{x},y\right)\,|\,\left(\bar{x},y\right)\in D,y=+1\right\}\right| \tag{2.15}$$

$$l_-=\left|\left\{\left(\bar{x},y\right)\,|\,\left(\bar{x},y\right)\in D,y=-1\right\}\right| \tag{2.16}$$

και δηλώνουν τον αριθμό των στοιχείων του D με χαρακτηρισμό $+1$ και -1 αντίστοιχα. Το Σχήμα 2-5(a) δείχνει τα κέντρα βάρους των δύο κλάσεων αντικειμένων.

Βήμα 2 Στη συνέχεια κατασκευάζουμε διάνυσμα $\bar{d}$ τέτοιο ώστε

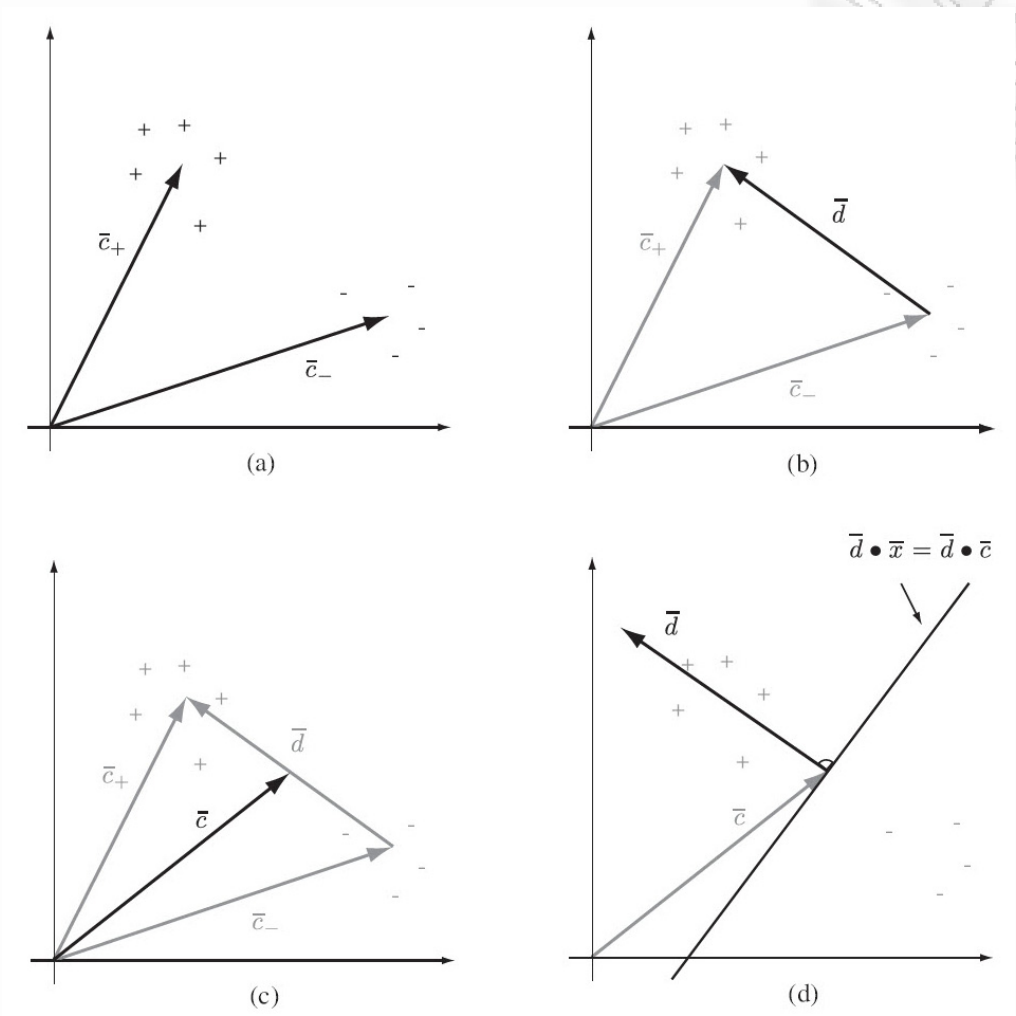
$$\bar{d}=\bar{c}_+-\bar{c}_- \tag{2.17}$$

όπως φαίνεται στο Σχήμα 2-5(b).

[1] Για το συμβολισμό του πλήθους του συνόλου εκπαίδευσης χρησιμοποιούνται συνήθως, στα πλαίσια της εργασίας, οι λατινικοί χαρακτήρες l και m.

Σχήμα 2-5

Γεωμετρική κατασκευή απλού αλγορίθμου μάθησης. (a) τα κέντρα βάρους $\overline{c}_+$ και $\overline{c}_-$ των δύο κλάσεων, (b) η διαφορά τους $\overline{d}$, (c) ο ολικός μέσος $\overline{c}$, (d) η επιφάνεια απόφασης $\overline{d} \bullet \overline{x} = \overline{d} \bullet \overline{c}$



Βήμα 3 Σε αυτό το βήμα υπολογίζουμε τον ολικό μέσο, έστω $\overline{c}$ μεταξύ των δύο κέντρων βάρους $\overline{c}_+$ και $\overline{c}_-$ ως εξής:

$$\overline{c} = \frac{1}{2}\left(\overline{c}_+ + \overline{c}_-\right) \tag{2.18}$$

Η κατασκευή του φαίνεται στο Σχήμα 2-5(c).

Βήμα 4 Μεταφέρουμε την αρχή του διανύσματος $\overline{d}$ στο $\overline{c}$ και κατασκευάζουμε ευθεία κάθετη στο $\overline{d}$ που διέρχεται από το $\overline{c}$ (Σχήμα 2-5 d). Η ευθεία αυτή αποτελεί την επιφάνεια απόφασης και κάνοντας χρήση της (2.5) με:

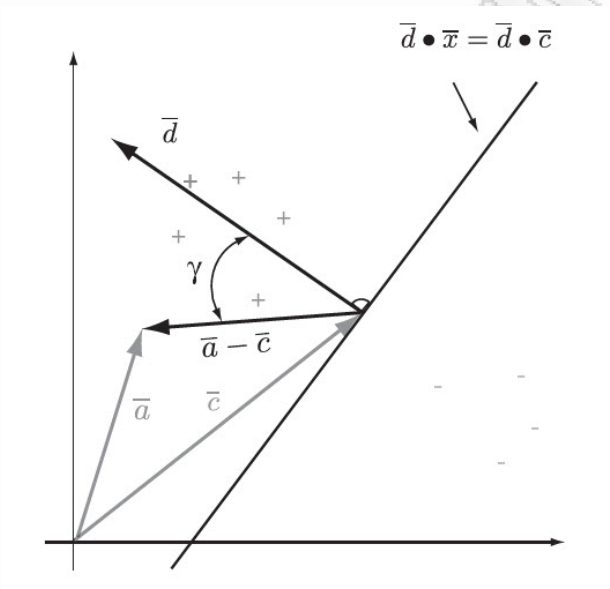
$$\overline{w} = \overline{d}\,, \tag{2.19}$$

$$b = \overline{d} \bullet \overline{c}\,, \tag{2.20}$$

λαμβάνουμε την εξίσωσή της:

$$\overline{d} \bullet \overline{x} = \overline{d} \bullet \overline{c} \tag{2.21}$$

Σχήμα 2-6
Ταξινόμηση ενός νέου σημείου $\overline{a}$



Βήμα 5 Τέλος, αντικαθιστώντας τις (2.19) και (2.20) στη γραμμική συνάρτηση απόφασης (2.10) λαμβάνουμε:

$$\hat{f}\left(\overline{x}\right) = \operatorname{sgn}\left(\overline{d} \bullet \overline{x} - \overline{d} \bullet \overline{c}\right) \tag{2.22}$$

για κάθε $\overline{x} \in \mathbb{R}^2$ και κάνοντας χρήση των ιδιοτήτων του εσωτερικού γινομένου καταλήγουμε στην παρακάτω εξίσωση για την επιφάνεια απόφασης:

$$\hat{f}\left(\overline{x}\right) = \operatorname{sgn}\left(\left(\overline{x} - \overline{c}\right) \bullet \overline{d}\right) \tag{2.23}$$

$$= \operatorname{sgn}\left(\left|\overline{x} - \overline{c}\right|\left|\overline{d}\right|\cos\gamma\right) \tag{2.24}$$

η οποία έχει και ενδιαφέρουσα γεωμετρική ερμηνεία. Η τιμή που λαμβάνει η συνάρτηση απόφασης για κάποιο σημείο $\overline{x}$ υπολογίζεται λαμβάνοντας το εσωτερικό γινόμενο μεταξύ του διανύσματος $\overline{x} - \overline{c}$ (το διάνυσμα θέσης του $\overline{x}$ ως προς το $\overline{c}$) και του κάθετου διανύσματος $\overline{d}$. Αν η γωνία γ μεταξύ των δύο διανυσμάτων είναι μικρότερη των $90°$, το

σημείο $\bar{x}$ είναι επάνω από την επιφάνεια απόφασης, αλλιώς βρίσκεται από κάτω. Το Σχήμα 2-6 απεικονίζει τα παραπάνω, για ένα νέο σημείο $\bar{a}$.

Μπορούμε να προχωρήσουμε ένα βήμα παραπέρα ώστε να καταλήξουμε σε μια αλγεβρική μορφή της συνάρτησης απόφασης, συναρτήσει των κέντρων βάρους των κλάσεων του συνόλου εκπαίδευσης, αντικαθιστώντας τις (2.17) και (2.18) στην (2.23). Έτσι παίρνουμε

$$\begin{aligned} \hat{f}(\bar{x}) &= \operatorname{sgn}\left((\bar{x}-\bar{c})\bullet\bar{d}\right) \\ &= \operatorname{sgn}\left(\left[\bar{x}-\frac{1}{2}(\bar{c}_{+}+\bar{c}_{-})\right]\bullet(\bar{c}_{+}-\bar{c}_{-})\right) \end{aligned} \tag{2.25}$$

2.5 Ανακεφαλαίωση

Με την παραπάνω μεθοδολογία αναπτύξαμε από μια καθαρά μαθηματική σκοπιά τις επιφάνειες και συναρτήσεις απόφασης και στη συνέχεια τις χρησιμοποιήσαμε σε έναν απλό αλγόριθμο μάθησης. Στον αλγόριθμο αυτό κάθε σημείο του συνόλου εκπαίδευσης συνεισφέρει εξίσου στον υπολογισμό του αντίστοιχου κεντροβαρικού σημείου. Αυτό δυνητικά θα μπορούσε να προκαλέσει προβλήματα, στην περίπτωση που το σύνολο εκπαίδευσης περιελάμβανε ακραίες παρατηρήσεις (outliers).

Οι ακραίες παρατηρήσεις δύνανται να μεταβάλουν τον προσανατολισμό της επιφάνειας απόφασης, κάτι που θα μπορούσε να οδηγήσει σε λάθος ταξινομήσεις (misclassifications) σημείων εκτός του συνόλου εκπαίδευσης.

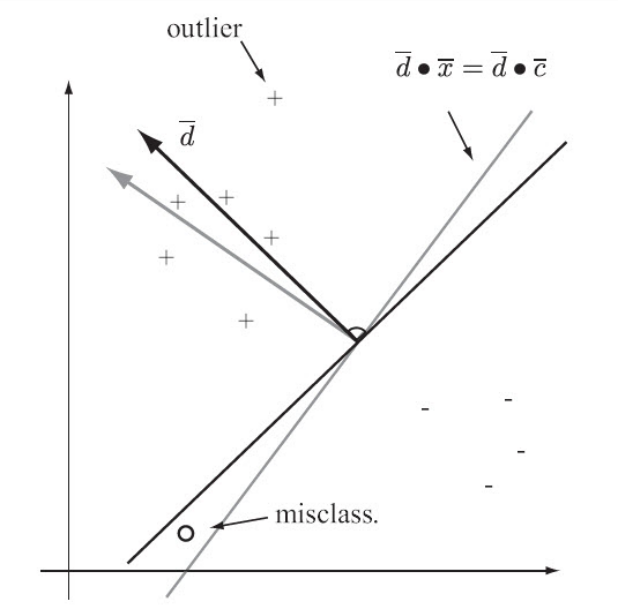
Η περίπτωση αυτή περιγράφεται οπτικά στο Σχήμα 2-7 για μια και μόνο ακραία παρατήρηση στη κλάση +1. Αν η ακραία παρατήρηση αγνοηθεί, προκύπτει η επιφάνεια απόφασης χρώματος γκρι. Με χρήση της επιφάνειας αυτής παρατηρήστε ότι το σημείο που σημειώνεται με κύκλο θα βρεθεί επάνω από την επιφάνεια αντί κάτω από αυτήν.

Προκειμένου να αποφευχθούν τέτοια σφάλματα ταξινόμησης θα πρέπει να αγνοηθούν οι ακραίες παρατηρήσεις και να χρησιμοποιηθούν μόνο σημεία που πραγματικά αντιπροσωπεύουν τις δύο κλάσεις.

Θα δούμε στη συνέχεια ότι καλύτερα μελετημένοι αλγόριθμοι όπως ο *perceptron* και η μηχανή διανυσμάτων υποστήριξης λύνουν αυτό το πρόβλημα, ελαχιστοποιώντας την επίδραση ακραίων παρατηρήσεων στην κατασκευή των επιφανειών απόφασης.

Σχήμα 2-7

Οι ακραίες παρατηρήσεις μπορεί να μεταβάλουν τον προσανατολισμό της επιφάνειας απόφασης και να οδηγήσουν σε λάθη ταξινόμησης



Προηγουμένως όμως, θα αναφερθούμε σε ένα πολύ δημοφιλή και ευρέως χρησιμοποιούμενο ταξινομητή, προερχόμενο από το πεδίο της στατιστικής. Αυτόν της λογιστικής παλινδρόμησης, όπου για την αποφυγή των σφαλμάτων ταξινόμησης λόγω ακραίων παρατηρήσεων θα εφαρμοστεί η τεχνική της *ομαλοποίησης*.

ΚΕΦΑΛΑΙΟ 3

Ομαλοποιημένη Λογιστική Παλινδρόμηση

3.1 Λογιστική Παλινδρόμηση

Παραμένοντας στο πρόβλημα της δυαδικής ταξινόμησης, θεωρούμε ότι η μεταβλητή απόκρισης y μπορεί να λάβει μόνο τις τιμές 0 ή 1. Αργότερα, στο παρόν κεφάλαιο, θα δούμε ότι είναι πολύ εύκολη η γενίκευση σε περιπτώσεις όπου η μεταβλητή απόκρισης περιλαμβάνει περισσότερες των δύο κατηγοριών. Αν προσπαθούσαμε να κατασκευάσουμε ένα ταξινομητή πού θα χαρακτηρίζει τα εισερχόμενα μηνύματα ηλεκτρονικής αλληλογραφίας ως ανεπιθύμητα (spam) ή μη, τότε το διάνυσμα χαρακτηριστικών για την i-οστή παρατήρηση $\bar{x}^{(i)}$, πιθανότατα θα περιελάμβανε μια σειρά χαρακτηριστικών γνωρισμάτων του μηνύματος, ενώ η μεταβλητή απόκρισης y ή αλλιώς ο ***χαρακτηρισμός*** (***label***) του μηνύματος, θα έπαιρνε την τιμή 1, στην περίπτωση spam και 0 αλλιώς.

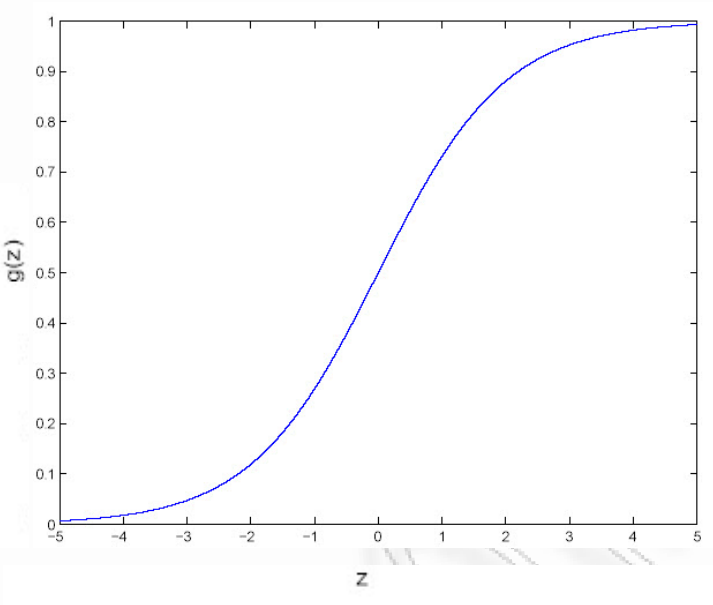
Η μορφή του μοντέλου $h_{\vartheta}(\bar{x})$ που υιοθετούμε στην περίπτωση αυτή είναι:

$$h_{\vartheta}\left(\bar{x}\right) = g\left(\bar{\vartheta} \bullet \bar{x}\right) = \frac{1}{1+e^{-\bar{\vartheta}\bullet\bar{x}}} \text{ , όπου: } g\left(z\right) = \frac{1}{1+e^{-z}}$$

η λεγόμενη ***λογιστική*** ή ***σιγμοειδής*** συνάρτηση. Η γραφική παράσταση της λογιστικής συνάρτησης δίνεται στο Σχήμα 3-1. Δεδομένου ότι η $g(z)$ ταυτίζεται με τη συνάρτηση κατανομής της τυπικής λογιστικής κατανομής, η τιμή της τείνει στο 1 καθώς το $z \to \infty$ και στο 0 καθώς το $z \to -\infty$. Επιπλέον η $g(z)$ και φυσικά η $h_{\vartheta}(\bar{x})$ είναι φραγμένη στο διάστημα $(0,1)$. Θα γίνει επίσης η παραδοχή ότι $\bar{x} = (x_0, x_1, \ldots, x_n)$ όπου $x_0 = 1$, οπότε

$$\bar{\vartheta} \bullet \bar{x} = \vartheta_0 + \sum_{j=1}^{n} \vartheta_j x_j$$

Σχήμα 3-1
Η γραφική παράσταση της λογιστικής συνάρτησης



Η ερμηνεία της τιμής που παράγει η λογιστική συνάρτηση, με δεδομένο το διάνυσμα τιμών $\bar{x}^{(i)}$ της i-οστής παρατήρησης, είναι η ***πιθανότητα*** να λάβει η παρατήρηση αυτή το χαρακτηρισμό $y=1$, δηλαδή

$$h_{\vartheta}(\bar{x}) = p\left(y=1 \mid \bar{x}; \bar{\vartheta}\right).$$

Δεδομένου ότι θα πρέπει εντέλει να ταξινομηθεί κάθε παρατήρηση είτε ως 0 είτε ως 1, είναι λογικό να υιοθετήσουμε την εξής παραδοχή: αν $h_{\vartheta}(\bar{x}) \geq 0.5$ η πρόβλεψη είναι "$y=1$", ενώ αν $h_{\vartheta}(\bar{x}) < 0.5$ η πρόβλεψη είναι "$y=0$". Σύμφωνα με το Σχήμα 3-1 λοιπόν, τιμές του $z \geq 0$ έχουν σαν αποτέλεσμα $g(z) \geq 0.5$. Συνεπώς:

$$h_{\vartheta}(\bar{x}) = g\left(\bar{\vartheta} \bullet \bar{x}\right) \geq 0.5 \text{ όταν } \bar{\vartheta} \bullet \bar{x} \geq 0 \text{ και}$$

$$h_{\vartheta}(\bar{x}) = g\left(\bar{\vartheta} \bullet \bar{x}\right) < 0.5 \text{ όταν } \bar{\vartheta} \bullet \bar{x} < 0.$$

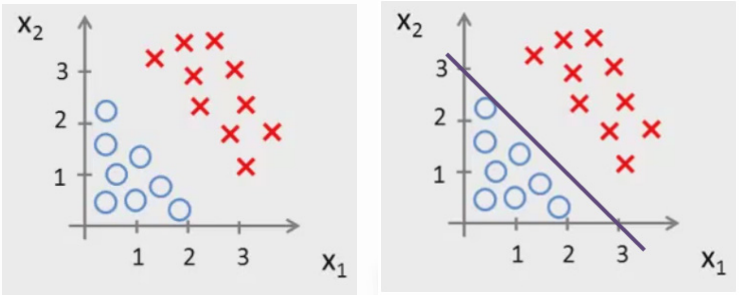
Ένα απλό παράδειγμα για τον τρόπο με τον οποίο το μοντέλο αυτό κατασκευάζει επιφάνειες απόφασης και καταλήγει σε προβλέψεις δίνεται στο Σχήμα 3-2 για το εμφανιζόμενο αριστερά ταξινομημένο σύνολο εκπαίδευσης. Η μορφή του μοντέλου στην περίπτωση αυτή είναι η εξής:

$$h_{\vartheta}(\bar{x}) = g\left(\vartheta_0 + \vartheta_1 x_1 + \vartheta_2 x_2\right)$$

Προτού αναφερθούμε στον τρόπο εκτίμησης των τιμών των παραμέτρων $\bar{\vartheta}$, ας υποθέσουμε ότι έχουμε καταλήξει στο διάνυσμα παραμέτρων $\bar{\vartheta} = [-3,1,1]^T$. Η επιφάνεια

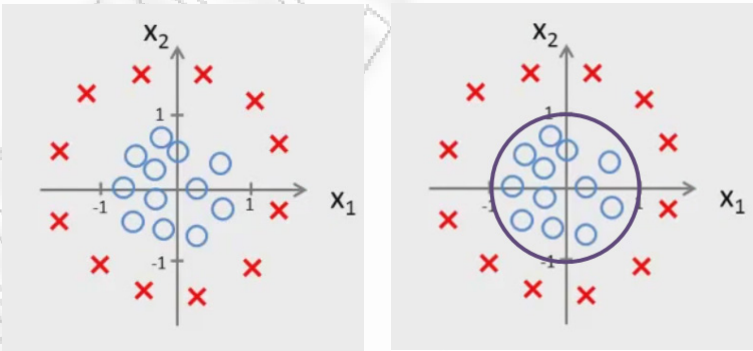
απόφασης $\bar{\vartheta} \bullet \bar{x} = 0$, που αντιστοιχεί σε $h_g(\bar{x}) = 0.5$, έχει τη μορφή ευθείας με εξίσωση: $-3 + x_1 + x_2 = 0$, η οποία φαίνεται στο Σχήμα 3-2 (δεξιά). Έτσι για κάθε ζευγάρι τιμών όπου $x_1 + x_2 \geq 3$ το μοντέλο θα προβλέπει "$y = 1$", αλλιώς "$y = 0$".

Σχήμα 3-2
Γραμμική επιφάνεια απόφασης στη λογιστική παλινδρόμηση



Σε περιπτώσεις όπου μια γραμμική επιφάνεια απόφασης δεν επαρκεί για το διαχωρισμό των δεδομένων, όπως για παράδειγμα για το σύνολο εκπαίδευσης που εμφανίζεται στο Σχήμα 3-3, μπορούμε να δημιουργήσουμε όρους δεύτερης ή και ανώτερης τάξης από τα διαθέσιμα χαρακτηριστικά με σκοπό να καταλήξουμε σε μια επιφάνεια απόφασης πιο σύνθετης μορφής.

Σχήμα 3-3
Μη γραμμική επιφάνεια απόφασης στη λογιστική παλινδρόμηση



Στη συγκεκριμένη περίπτωση θα μπορούσαμε να επιλέξουμε ένα μοντέλο της μορφής:

$$h_\vartheta(\bar{x}) = g\left(\vartheta_0 + \vartheta_1 x_1 + \vartheta_2 x_2 + \vartheta_3 x_1^2 + \vartheta_4 x_2^2\right)$$

και έστω πάλι ότι έχουμε καταλήξει στο διάνυσμα παραμέτρων $\vartheta = [-1, 0, 0, 1, 1]^T$.

Η επιφάνεια απόφασης $\bar{\vartheta} \bullet \bar{x} = 0$ έχει τώρα τη μορφή κύκλου με εξίσωση: $-1 + x_1^2 + x_2^2 = 0$ Για κάθε ζεύγος σημείων που θα ισχύει $x_1^2 + x_2^2 \geq 1$ το μοντέλο θα προβλέπει "$y = 1$", αλλιώς "$y = 0$".

3.2 Εκτίμηση των Παραμέτρων του Μοντέλου

Σύμφωνα με τα προαναφερθέντα, σχετικά με την ερμηνεία της τιμής που παράγει η λογιστική συνάρτηση και δεδομένου ότι κάθε αντικείμενο χαρακτηρίζεται αποκλειστικά με μία από τις τιμές 0 ή 1, μπορούμε να γράψουμε:

$$p\left(y = 1 \mid \bar{x}; \bar{\vartheta}\right) = h_\vartheta\left(\bar{x}\right)$$

$$p\left(y = 0 \mid \bar{x}; \bar{\vartheta}\right) = 1 - h_\vartheta\left(\bar{x}\right).$$

Οι παραπάνω σχέσεις μπορούν να γραφούν πιο συνεπτυγμένα ως εξής:

$$p\left(y \mid \bar{x}; \bar{\vartheta}\right) = \left(h_\vartheta\left(\bar{x}\right)\right)^y \left(1 - h_\vartheta\left(\bar{x}\right)\right)^{1-y}.$$

Θεωρώντας ότι τα m δείγματα του συνόλου εκπαίδευσης είναι μεταξύ τους ανεξάρτητα, μπορούμε να γράψουμε την «*πιθανοφάνεια*» (*Likelihood*) των παραμέτρων ως εξής:

$$L\left(\bar{\vartheta}\right) = p\left(\bar{y} \mid S; \bar{\vartheta}\right) = \prod_{i=1}^{m} p\left(y^{(i)} \mid \bar{x}^{(i)}; \bar{\vartheta}\right) = \prod_{i=1}^{m} \left(h_\vartheta\left(\bar{x}^{(i)}\right)\right)^{y^{(i)}} \left(1 - h_\vartheta\left(\bar{x}^{(i)}\right)\right)^{1-y^{(i)}}$$

όπου S ο πίνακας των χαρακτηριστικών του συνόλου εκπαίδευσης.

Καθώς είναι ευκολότερο να εργαστούμε με τη λογαριθμοπιθανοφάνεια (log-likelihood), λαμβάνουμε:

$$l\left(\bar{\vartheta}\right) = \log L\left(\bar{\vartheta}\right)$$

$$= \sum_{i=1}^{m} y^{(i)} \log\left(h_\vartheta\left(\bar{x}^{(i)}\right)\right) + \left(1 - y^{(i)}\right) \log\left(1 - h_\vartheta\left(\bar{x}^{(i)}\right)\right).$$

Δεδομένου ότι η συνήθης πρακτική στις μεθόδους μηχανικής μάθησης είναι να ελαχιστοποιείται μια συνάρτηση κόστους, αντί να μεγιστοποιήσουμε την πιθανοφάνεια θα ορίσουμε την αντίθετή της ως συνάρτηση κόστους με στόχο να βρούμε τις τιμές των παραμέτρων $\bar{\vartheta}$ για τις οποίες ελαχιστοποιείται:

$$J\left(\bar{\vartheta}\right) = \sum_{i=1}^{m} \left[-y^{(i)} \log\left(h_\vartheta\left(\bar{x}^{(i)}\right)\right) - \left(1 - y^{(i)}\right) \log\left(1 - h_\vartheta\left(\bar{x}^{(i)}\right)\right) \right]. \qquad (3.1)$$

Πριν προχωρήσουμε, ας δούμε μια χρήσιμη ιδιότητα της παραγώγου $g'(z)$ της σιγμοειδούς συνάρτησης:

$$g'(z) = \frac{d}{dz}\frac{1}{1+e^{-z}} = \frac{1}{\left(1+e^{-z}\right)^2}\left(e^{-z}\right)$$

$$= \frac{1}{1+e^{-z}}\left(1-\frac{1}{1+e^{-z}}\right) = g(z)\left(1-g(z)\right). \qquad (3.2)$$

Για την ελαχιστοποίηση της συνάρτησης κόστους μπορεί να χρησιμοποιηθεί η μέθοδος της «***Επικλινούς καθόδου***» (***gradient descent***). Σύμφωνα με τη μέθοδο αυτή, ξεκινώντας από κάποια αρχική τιμή των παραμέτρων $\bar{\vartheta}$, εκτελούμε επαναληπτικά την παρακάτω διόρθωση[1]:

$$\vartheta_j \leftarrow \vartheta_j - \alpha\frac{\partial}{\partial\vartheta_j}J\left(\bar{\vartheta}\right).$$

Με κάθε επανάληψη του παραπάνω αλγόριθμου οδηγούμαστε και ένα βήμα προς την κατεύθυνση όπου η γραφική παράσταση της J παρουσιάζει την πιο απότομη κλίση. Το μήκος του βήματος μειώνεται όσο πλησιάζουμε στο σημείο του ολικού ελαχίστου (καθώς μειώνεται συνεχώς η τιμή της μερικής παραγώγου) κατά ένα μέγεθος που εξαρτάται και από την επιλογή της τιμής της παραμέτρου α η οποία καλείται «***ρυθμός εκμάθησης***» (***learning rate***). Προκειμένου να χρησιμοποιηθεί ο αλγόριθμος αυτός, θα πρέπει να υπολογιστεί η τιμή της μερικής παραγώγου της συνάρτησης κόστους J ως προς ϑ_j. Ας δούμε τη διαδικασία για ένα μόνο αντικείμενο του συνόλου εκπαίδευσης $(\bar{x}, y)$:

$$\frac{\partial}{\partial\vartheta_j}J\left(\bar{\vartheta}\right) = \left(-y\frac{1}{g\left(\bar{\vartheta}\bullet\bar{x}\right)} + (1-y)\frac{1}{1-g\left(\bar{\vartheta}\bullet\bar{x}\right)}\right)\frac{\partial}{\partial\vartheta_j}g\left(\bar{\vartheta}\bullet\bar{x}\right)$$

$$= \left(-y\frac{1}{g\left(\bar{\vartheta}\bullet\bar{x}\right)} + (1-y)\frac{1}{1-g\left(\bar{\vartheta}\bullet\bar{x}\right)}\right)g\left(\bar{\vartheta}\bullet\bar{x}\right)\left(1-g\left(\bar{\vartheta}\bullet\bar{x}\right)\right)\frac{\partial}{\partial\vartheta_j}\left(\bar{\vartheta}\bullet\bar{x}\right)$$

$$= \left(-y\left(1-g\left(\bar{\vartheta}\bullet\bar{x}\right)\right) + (1-y)g\left(\bar{\vartheta}\bullet\bar{x}\right)\right)x_j$$

$$= \left(h_\vartheta\left(\bar{x}\right)-y\right)x_j. \qquad (3.3)$$

Στα παραπάνω έγινε χρήση της ιδιότητας που δείχθηκε με τη σχέση (3.2). Συνεπώς, ο κανόνας με βάση τον οποίο γίνεται η επαναληπτική διόρθωση των τιμών των παραμέτρων $\bar{\vartheta}$ για ένα αντικείμενο του συνόλου εκπαίδευσης διαμορφώνεται ως εξής:

[1] Τονίζεται ότι η ενημέρωση αυτή γίνεται ταυτόχρονα για όλες τις τιμές του $j = 0,\ldots,n$ σε κάθε βήμα.

$$\vartheta_j \leftarrow \vartheta_j - \alpha\left(h_\vartheta(\bar{x}) - y\right)x_j.$$

Θα πρέπει εδώ να τονιστεί ότι η *τυποποίηση* των τιμών των χαρακτηριστικών, στην περίπτωση που οι μονάδες μέτρησής τους διαφέρουν μεταξύ τους, βοηθά στην ταχύτερη σύγκλιση του αλγορίθμου. Τέλος σημειώνεται ότι, όπως θα δούμε και παρακάτω, η συνάρτηση κόστους $J(\bar{\vartheta})$ της σχέσης (3.1), συνηθίζεται να λαμβάνεται πολλαπλασιασμένη με τον όρο $1/m$.

3.3 Ομαλοποιημένη Λογιστική Παλινδρόμηση

Αν και ο αλγόριθμος της λογιστικής παλινδρόμησης δουλεύει ικανοποιητικά σε πληθώρα προβλημάτων, όταν εφαρμοστεί σε συγκεκριμένες περιπτώσεις προβλημάτων μηχανικής μάθησης δεν αποκλείεται να υποπέσει στο σφάλμα που καλείται «*υπερπροσαρμογή*» (overfitting) με αποτέλεσμα τη δραστική μείωση της αποτελεσματικότητάς του.

Συγκεκριμένα, λέγοντας υπερπροσαρμογή εννοούμε το φαινόμενο όπου το μοντέλο λαμβάνει μια υπερβολικά περίτεχνη μορφή, προκειμένου να επεξηγήσει όσο το δυνατόν πιο πιστά την όποια συμπεριφορά εμφανίζει το περιορισμένο πλήθος των δεδομένων εκπαίδευσης. Στη πραγματικότητα όμως, τα δεδομένα που αποτελούν το σύνολο εκπαίδευσης συχνά εμπεριέχουν κάποιο βαθμό σφάλματος ή τυχαίου θορύβου. Έτσι, δημιουργώντας ένα μοντέλο που αναπαράγει σε υπερβολικό βαθμό τη συμπεριφορά αυτών των, έστω και ελαφρώς, ανακριβών δεδομένων, αυξάνεται ο κίνδυνος μειωμένης προβλεπτικής ικανότητάς του (γενίκευσης), όταν εφαρμόζεται σε δεδομένα εκτός του δείγματος εκπαίδευσης. Γι' αυτό άλλωστε είναι απαραίτητος ο έλεγχος κάθε μοντέλου σε δεδομένα, εκτός του δείγματος που χρησιμοποιήθηκε για την κατασκευή του.

Εναλλακτικά, στο φαινόμενο της υπερπροσαρμογής αναφερόμαστε λέγοντας ότι το μοντέλο έχει *«υψηλή διακύμανση»* (high variance). Αντίστοιχα στην περίπτωση της υποπροσαρμογής (under-fitting), λέμε ότι το μοντέλο χαρακτηρίζεται από *«υψηλή μεροληψία»* (high bias) [1].

Στις περισσότερες περιπτώσεις το μεγάλο πλήθος των διαθέσιμων χαρακτηριστικών για τις παρατηρήσεις του συνόλου εκπαίδευσης, καθιστά αδύνατο τον οπτικό έλεγχο ενδεχόμενης

[1] Για περισσότερα βλ. Κεφ. 7.

υπερπροσαρμογής του μοντέλου. Για την αποφυγή της υπερπροσαρμογής λοιπόν, οι διαθέσιμες επιλογές είναι οι εξής δύο:

1. **Μείωση του αριθμού των χαρακτηριστικών.**
   - Επιλογή των πλέον σημαντικών χαρακτηριστικών (χειρονακτικά)
   - Επιλογή χαρακτηριστικών μέσω αλγορίθμου
2. ***«Ομαλοποίηση»*** **(Regularization)**
   - Διατήρηση όλων των χαρακτηριστικών αλλά μείωση των τιμών που λαμβάνουν οι παράμετροι. Η μέθοδος αυτή δουλεύει ικανοποιητικά όταν το πλήθος των χαρακτηριστικών είναι μεγάλο και ταυτόχρονα όλα τα χαρακτηριστικά θεωρείται ότι έχουν έστω και μικρή συνεισφορά στην πρόβλεψη της εξαρτημένης μεταβλητής.

Η ιδέα της μείωσης των χαρακτηριστικών μπορεί πράγματι να δουλέψει ικανοποιητικά και να αποφευχθεί η υπερπροσαρμογή. Το πρόβλημα όμως είναι ότι αγνοώντας συγκεκριμένα χαρακτηριστικά, χάνεται πληροφορία ενδεχομένως χρήσιμη για το υπ' όψη πρόβλημα.

Από την άλλη μεριά, η βασική ιδέα της ομαλοποίησης είναι σε γενικές γραμμές η εξής: αν διατηρηθούν σε χαμηλά επίπεδα τα μεγέθη των τιμών των παραμέτρων, μπορεί να αποδειχθεί ότι η μορφή της συνάρτησης *εξομαλύνεται* [30], με συνέπεια να καταλήγουμε εν γένει σε απλούστερης μορφής μοντέλα, λιγότερο επιρρεπή σε υπερπροσαρμογή. Στην πράξη λοιπόν, αυτό που κάνουμε με την ομαλοποίηση, είναι ότι τροποποιούμε την εκάστοτε συνάρτηση κόστους, προσθέτοντας έναν επιπλέον όρο σε αυτήν, ώστε να μειωθούν οι τιμές όλων των παραμέτρων, εξαιρουμένης συνήθως της παραμέτρου ϑ_0.

Στην περίπτωση της «***ομαλοποιημένης λογιστικής παλινδρόμησης***» (***Regularized Logistic Regression - RLR***) μπορούμε να εκτιμήσουμε τις τιμές των παραμέτρων $\overline{\vartheta}$, όπως ήδη έχει αναφερθεί, ελαχιστοποιώντας την παρακάτω συνάρτηση κόστους (αντί να μεγιστοποιήσουμε τη log-likelihood):

$$J\left(\overline{\vartheta}\right)=\frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log\left(h_{\vartheta}\left(\overline{x}^{(i)}\right)\right)-\left(1-y^{(i)}\right)\log\left(1-h_{\vartheta}\left(\overline{x}^{(i)}\right)\right)\right]$$

όπου m το πλήθος του συνόλου εκπαίδευσης και

$$h_{\vartheta}\left(\overline{x}\right)=g\left(\overline{\vartheta}\bullet\overline{x}\right)=\frac{1}{1+e^{-\overline{\vartheta}\bullet\overline{x}}}$$

η λογιστική συνάρτηση. Στην ομαλοποιημένη μορφή της παραπάνω συνάρτησης, απλά προστίθεται ένας επιπλέον όρος:

$$J\left(\bar{\vartheta}\right)=\frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log\left(h_{\vartheta}\left(\bar{x}^{(i)}\right)\right)-\left(1-y^{(i)}\right)\log\left(1-h_{\vartheta}\left(\bar{x}^{(i)}\right)\right)\right]+\frac{\lambda}{2m}\sum_{j=1}^{n}\vartheta_j^2 \quad (3.4)$$

όπου n το πλήθος των παραμέτρων (εκτός της σταθεράς). Ο επιπλέον αυτός όρος εν τέλει λειτουργεί αποτρεπτικά στο να λαμβάνουν υψηλές τιμές οι παράμετροι $\vartheta_1,\ldots,\vartheta_n$, με τον όρο ϑ_0 να εξαιρείται, κατά σύμβαση, από τον περιορισμό του να μη λάβει υψηλή τιμή.

Ο όρος λ καλείται παράμετρος ομαλοποίησης και η επιλογή της τιμής του σχετίζεται με την αντιστάθμιση των δύο αντικρουόμενων επιδιώξεων δηλαδή, από τη μια μεριά, την όσο το δυνατόν καλύτερη προσαρμογή στα δεδομένα εκπαίδευσης (που εκφράζεται από τον πρώτο όρο του αθροίσματος) και από την άλλη, τη διατήρηση σε χαμηλά επίπεδα των τιμών των παραμέτρων ώστε να αποφευχθεί η υπερπροσαρμογή (που εκφράζεται από τον δεύτερο όρο).

Είναι προφανές ότι η επιλογή πολύ μεγάλων τιμών για το λ μπορεί να οδηγήσει σε τέτοιας τάξης μείωση των τιμών των παραμέτρων, ώστε το μοντέλο να οδηγηθεί στο άλλο άκρο, αυτό της υποπροσαρμογής. Τελικά, το αποτέλεσμα αυτής της απλής παρέμβασης στην έκφραση της συνάρτησης κόστους είναι ότι, ακόμη κι αν χρησιμοποιείται ένας μεγάλος αριθμός παραμέτρων ή μια υψηλού βαθμού πολυωνυμική συνάρτηση αυτών, η εφαρμογή της ομαλοποίησης θα οδηγήσει κατά πάσα πιθανότητα σε μια μορφή μοντέλου που δεν υπερπροσαρμόζεται στα δεδομένα εκπαίδευσης.

3.4 Αλγόριθμοι Βελτιστοποίησης

Για την ελαχιστοποίηση της συνάρτησης κόστους, είναι δυνατόν (και σχετικά απλό) να υλοποιηθεί κώδικας που να εφαρμόζει τον αλγόριθμο gradient descent [23], ο οποίος περιγράφηκε στην Παράγραφο 3.2. Περισσότερες λεπτομέρειες για το συγκεκριμένο αλγόριθμο δε θα δοθούν εδώ, μιας και στην πράξη, τις περισσότερες φορές πλέον, χρησιμοποιούνται περισσότερο προηγμένοι αλγόριθμοι βελτιστοποίησης, οι οποίοι είναι ταχύτεροι από τον gradient descent, ενώ ανταποκρίνονται καλύτερα σε σύνολα εκπαίδευσης πολύ μεγάλων διαστάσεων.

Συνήθως οι αλγόριθμοι αυτοί απαιτούν, ως μια από τις εισόδους τους, την υπόδειξη μιας συνάρτησης που επιστρέφει την τιμή που λαμβάνει η συνάρτηση κόστους $J(\bar{\vartheta})$ και, εφόσον

αυτό είναι δυνατόν, το *«διάνυσμα κλίσης»* (gradient vector), δηλαδή τις μερικές παραγώγους αυτής της συνάρτησης κόστους ως προς τις παραμέτρους $\bar{\vartheta}$:

$$\frac{\partial}{\partial \vartheta_j} J\left(\bar{\vartheta}\right) \text{ , για } j = 0,1,\ldots,n.$$

Ως παραδείγματα αλγορίθμων που χρησιμοποιούν περισσότερο «έξυπνες» μεθόδους απ' αυτήν του gradient descent για την ελαχιστοποίηση συναρτήσεων, αναφέρονται ενδεικτικά οι:

- Conjugate gradient
- BFGS[1]
- L-BFGS[2].

Οι λεπτομέρειες της λειτουργίας των παραπάνω αλγορίθμων είναι πέραν του αντικειμένου αυτής της εργασίας (μπορούν άλλωστε να χρησιμοποιηθούν επιτυχώς, χωρίς να απαιτείται η γνώση των λεπτομερειών λειτουργίας τους). Οι βασικότερες ιδιότητές τους πάντως, είναι οι εξής:

Πλεονεκτήματα

- Δεν απαιτείται επιλογή κάποιου ρυθμού εκμάθησης, όπως του α που απαιτεί ο αλγόριθμος gradient descent.
- Συγκλίνουν συνήθως γρηγορότερα στη βέλτιστη λύση από τον gradient descent.

Μειονεκτήματα

- Παρουσιάζουν μεγαλύτερη πολυπλοκότητα στην υλοποίηση.

Εξαιτίας του μειονεκτήματος της πολυπλοκότητας, για τη χρήση τέτοιων αλγορίθμων θα πρέπει κανείς να καταφύγει σε κατάλληλες, έτοιμες βιβλιοθήκες λογισμικού, μια και η εξαρχής υλοποίησή τους απαιτεί εξειδικευμένες γνώσεις. Στην περίπτωση του παραδείγματος της επόμενης παραγράφου, χρησιμοποιήθηκε ένας τέτοιος αλγόριθμος, όπως αυτός έχει υλοποιηθεί στην open source γλώσσα προγραμματισμού ***Octave***. Ο αλγόριθμος ενεργοποιείται μέσω της συνάρτησης ***fminunc***[3].

---

[1] BFGS: Μέθοδος των Broyden-Fletcher-Goldfarb-Shanno για την επίλυση μη γραμμικών προβλημάτων βελτιστοποίησης, άνευ περιορισμών.

[2] Limited memory BFGS: Εκδοχή του BFGS με περιορισμένες απαιτήσεις μνήμης.

[3] Η ονομασία της συνοψίζει τη φράση: «Find minimum of unconstrained multivariable function» Βλ. http://www.mathworks.com/help/toolbox/optim/ug/fminunc.html

Η γλώσσα Octave επιλέχθηκε ως περιβάλλον υλοποίησης γιατί έχει το πλεονέκτημα ότι τα αντικείμενά της είναι εξ ορισμού πίνακες και οι τελεστές της υποστηρίζουν εγγενώς πράξεις γραμμικής άλγεβρας. Έτσι, αν γίνει σωστή εκμετάλλευση των πλεονεκτημάτων αυτών μέσω τεχνικών διανυσματοποίησης των υπολογισμών (vectorized code), ο κώδικας λαμβάνει μια πολύ πιο συμπυκνωμένη και κομψή μορφή αλλά κυρίως γίνεται **κατά πολύ ταχύτερος**.

Στο παράδειγμα που ακολουθεί, για λόγους σύγκρισης, έγινε υλοποίηση και με τη γλώσσα προγραμματισμού ***R***, ώστε να γίνει κατανοητός ο τρόπος χρήσης της αντίστοιχης συνάρτησης βελτιστοποίησης ***optim***[1].

3.5 Εφαρμογή RLR: Πρόβλεψη Καταλληλότητας Προϊόντος

Η ομαλοποιημένη λογιστική παλινδρόμηση εφαρμόστηκε σε μια μελέτη περίπτωσης, σύμφωνα με την οποία ο διευθυντής παραγωγής ενός εργοστασίου κατασκευής μικροεπεξεργαστών χρειάζεται, κατά τον έλεγχο διασφάλισης ποιότητας, ένα μοντέλο πρόβλεψης της καταλληλότητας των παραγόμενων μικροεπεξεργαστών βασισμένο στα αποτελέσματα δύο δοκιμασιών στις οποίες αυτοί υποβάλλονται.

Προκειμένου να προχωρήσουμε στην κατασκευή ενός μοντέλου λογιστικής παλινδρόμησης, είναι διαθέσιμο ένα σύνολο 118 μικροεπεξεργαστών ταξινομημένων ως αποδεκτών ή μη, μαζί με τα αποτελέσματα των αντίστοιχων δοκιμασιών.

Πρόσβαση στο συγκεκριμένο αρχείο δεδομένων αλλά και στο πλήρες σετ των αρχείων με τον κώδικα της εφαρμογής παρέχεται μέσω του συνδέσμου: http://bit.ly/xugR1l .

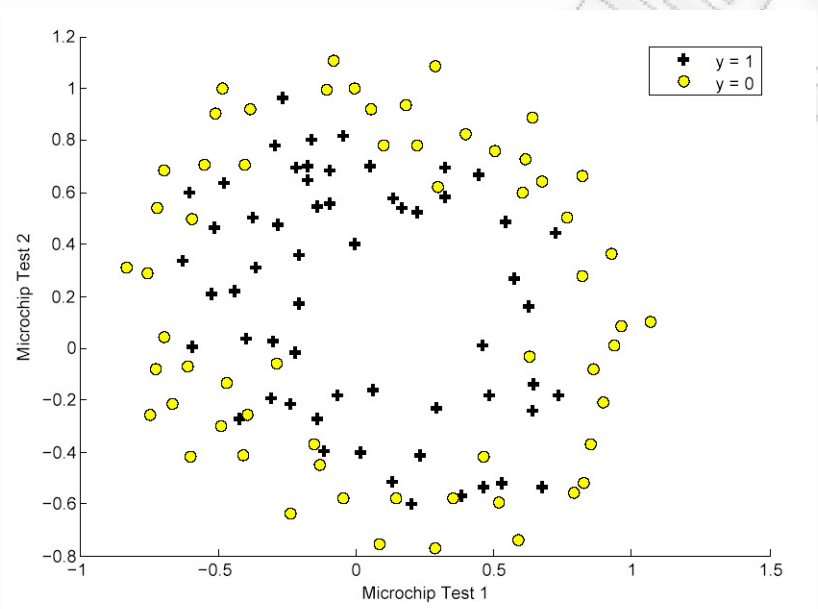
Στο Σχήμα 3-4 εμφανίζεται ένα διάγραμμα διασποράς των τιμών που έχουν ληφθεί για κάθε μία από τις δύο δοκιμασίες στους άξονες x_1 και x_2. Επίσης, με διαφορετική σήμανση διακρίνονται οι περιπτώσεις μικροεπεξεργαστών που έγιναν αποδεκτοί ($y = 1$), καθώς και αυτοί που απορρίφθηκαν ($y = 0$).

Από το διάγραμμα αυτό είναι προφανές ότι είναι αδύνατος ο διαχωρισμός των αποδεκτών από τις μη αποδεκτές παρατηρήσεις με μια ευθεία γραμμή. Συνεπώς η απλή εφαρμογή της λογιστικής παλινδρόμησης δε θα έχει ικανοποιητικά αποτελέσματα στη συγκεκριμένη περίπτωση, αφού το αποτέλεσμά της θα ήταν μια γραμμική επιφάνεια απόφασης.

[1] Βλ. http://finzi.psych.upenn.edu/R/library/stats/html/optim.html , καθώς και το πακέτο *optimx* [24]

Ένας τρόπος επίτευξης καλύτερης προσαρμογής είναι μέσω της δημιουργίας περισσότερων χαρακτηριστικών. Έτσι, από τις αρχικές μεταβλητές x_1, x_2 του συνόλου εκπαίδευσης δημιουργήθηκαν όλες οι μεταβλητές που αντιστοιχούν στους όρους ενός πολυωνύμου 6^{ου} βαθμού, δηλαδή: $[1, x_1, x_2, x_1^2, x_1x_2, x_2^2, x_1^3, \ldots, x_1x_2^5, x_2^6]^T$.

Σχήμα 3-4
Διάγραμμα διασποράς του συνόλου εκπαίδευσης



Ως αποτέλεσμα λοιπόν, ξεκινώντας από ένα διάνυσμα δύο μεταβλητών (με τα σκορ των δύο δοκιμασιών), καταλήξαμε σε ένα διάνυσμα 28 μεταβλητών. Ένας ταξινομητής λογιστικής παλινδρόμησης ο οποίος θα έχει «εκπαιδευτεί» σε ένα διάνυσμα χαρακτηριστικών τέτοιας διάστασης, σίγουρα θα δώσει μια επιφάνεια απόφασης με πολύ πιο περίπλοκη μορφή, ικανή να διαχωρίσει με αντίστοιχα μεγαλύτερη επιτυχία, τα δεδομένα του υπόψη δείγματος.

Φυσικά, ενώ το νέο διάνυσμα χαρακτηριστικών μάς επιτρέπει να κατασκευάσουμε ένα ταξινομητή πολύ πιο ευπροσάρμοστο, η όλη διαδικασία γίνεται ιδιαίτερα ευάλωτη στον κίνδυνο της υπερπροσαρμογής.

Για την επίλυση του εν λόγω προβλήματος κάνοντας χρήση της ομαλοποίησης, σύμφωνα με αυτά που αναφέρθηκαν και στην προηγούμενη παράγραφο, απαιτείται η δημιουργία μιας συνάρτησης που θα επιστρέφει, αφενός την τιμή που λαμβάνει η συνάρτηση κόστους (ως προς το διάνυσμα των παραμέτρων $\vec{\vartheta}$) και αφετέρου το διάνυσμα κλίσης (gradient vector).

Η μορφή της ομαλοποιημένης συνάρτησης κόστους της λογιστικής παλινδρόμησης έχει ήδη δοθεί στη σχέση (3.4).

Το στοιχεία του διανύσματος κλίσης της συνάρτησης αυτής, με βάση τη σχέση (3.3), θα πρέπει να υπολογιστούν σύμφωνα με τους παρακάτω τύπους[1]:

$$\frac{\partial J(\vec{\vartheta})}{\partial \vartheta_j} = \frac{1}{m}\sum_{i=1}^{m}\left(h_\vartheta\left(\vec{x}^{(i)}\right) - y^{(i)}\right)x_j^{(i)} \quad \text{για } j = 0,$$

$$\frac{\partial J(\vec{\vartheta})}{\partial \vartheta_j} = \frac{1}{m}\left(\sum_{i=1}^{m}\left(h_\vartheta\left(\vec{x}^{(i)}\right) - y^{(i)}\right)x_j^{(i)} + \lambda\vartheta_j\right) \quad \text{για } j \geq 1.$$

Αφού δημιουργηθεί αυτή η συνάρτηση, θα πρέπει να περαστεί ως παράμετρος στη συνάρτηση *fminunc* της Octave. Η συνάρτηση fminunc βρίσκει το σημείο όπου ελαχιστοποιείται μια συνάρτηση άνευ περιορισμών[2]. Επιπλέον απαιτείται ως είσοδος στη συνάρτηση, το διάνυσμα των αρχικών τιμών των προς βελτιστοποίηση παραμέτρων και προαιρετικά, ο καθορισμός των τιμών κάποιων παραμέτρων λειτουργίας.

Συγκεκριμένα, ο τρόπος κλήσης της δίδεται στο ακόλουθο πλαίσιο:

```
% Ορισμός του 'GradObj' σε 'on' αφού η συνάρτηση κόστους επιστρέφει το gradient
% Επιπλέον ορίζουμε το μέγιστο αριθμό βημάτων έως τον τερματισμό μέσω της 'MaxIter'
options = optimset('GradObj', 'on', 'MaxIter', 400);

% Optimize
[theta, J, exit_flag] = ...
    fminunc(@(t)(costFunctionReg(t, X, y, lambda)), initial_theta, options);
```

Για την υπόδειξη της συνάρτησης που θα ελαχιστοποιηθεί (εδώ της costFunctionReg), την οποία ασφαλώς θα πρέπει να έχουμε ήδη δημιουργήσει, χρησιμοποιείται η τεχνική του ορισμού μιας *ανώνυμης*[3] συνάρτησης με παράμετρο *t*, η οποία καλεί την costFunctionReg. Η τιμή του *exit_flag* υποδηλώνει το λόγο για τον οποίο ο αλγόριθμος τερματίστηκε. Περισσότερες λεπτομέρειες δίνονται στην τεκμηρίωση της συνάρτησης η οποία περιλαμβάνεται στο σύνδεσμο (link) που έχει ήδη δοθεί.

Για την καλύτερη οπτική απεικόνιση του μοντέλου που προήλθε από τη διαδικασία «μάθησης», δίδεται στο Σχήμα 3-5 το όριο απόφασης το οποίο διαχωρίζει τις αποδεκτές από τις μη αποδεκτές παρατηρήσεις.

---

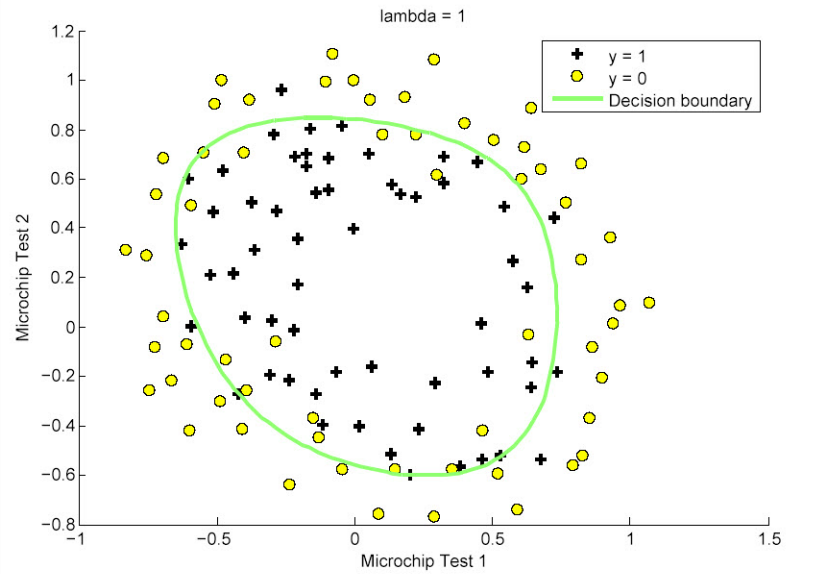
[1] Υπενθυμίζεται ότι, κατά σύμβαση, ο σταθερός όρος εξαιρείται από τη διαδικασία ομαλοποίησης.

[2] Οι περιορισμοί στα προβλήματα βελτιστοποίησης αναφέρονται στα όρια που επιτρέπεται να λάβουν οι πιθανές τιμές των παραμέτρων ή σε άλλου είδους σχέσεις που επιβάλλεται να ικανοποιούν. Η λογιστική παλινδρόμηση δεν επιβάλει τέτοιους περιορισμούς, αφού οι παράμετροι μπορούν να λάβουν οποιαδήποτε πραγματική τιμή.

[3] Μέσω των ανώνυμων συναρτήσεων παρέχεται η δυνατότητα άμεσης δημιουργίας απλών συναρτήσεων χωρίς να απαιτείται η κατασκευή *.m* αρχείων. Βλ. http://www.mathworks.com/help/techdoc/matlab_prog/f4-70115.html

Για τη σχεδίαση αυτού του ορίου υπολογίστηκαν οι τιμές που λαμβάνει ο όρος $\overline{\vartheta} \bullet \overline{x}$ σε ένα πλέγμα σημείων του επιπέδου και στη συνέχεια σχεδιάστηκε η ισοϋψής καμπύλη που αντιστοιχεί στη στάθμη 0, ή με άλλα λόγια στην πρόβλεψη $p = 0.5$.

Σχήμα 3-5
Δεδομένα εκπαίδευσης και όριο διαχωρισμού για $\lambda = 1$



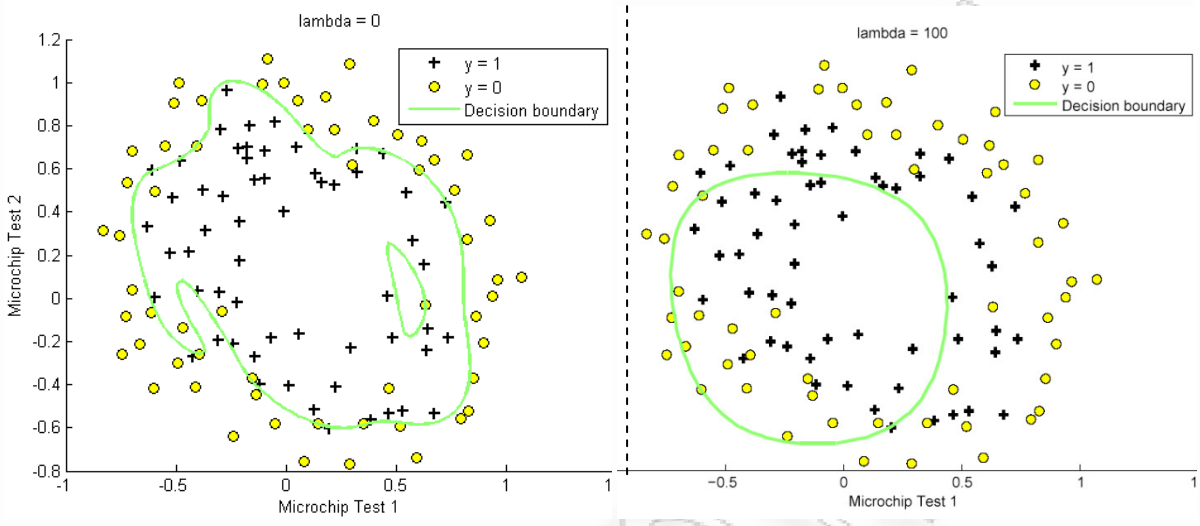
Δοκιμάζοντας τη δημιουργία μοντέλου χωρίς την εφαρμογή ομαλοποίησης (κάτι που αντιστοιχεί σε μια τιμή του συντελεστή $\lambda = 0$), το μοντέλο στο οποίο καταλήξαμε υπερπροσαρμοζόταν στα δεδομένα και το όριο απόφασης είχε τη μορφή που φαίνεται στο αριστερό τμήμα, στο Σχήμα 3-6.

Αντίστοιχα, η επιλογή μιας μεγάλης τιμής για το λ, π.χ. $\lambda = 100$ είχε τα αντίθετα αποτελέσματα. Όπως φαίνεται και στο δεξί τμήμα στο Σχήμα 3-6, η προσαρμογή παύει πλέον να είναι ικανοποιητική και το όριο απόφασης δε διαχωρίζει ικανοποιητικά τα δεδομένα. Έχουμε με άλλα λόγια, ένα ξεκάθαρο παράδειγμα υποπροσαρμογής.

Σημειώνεται εδώ ότι οι υλοποιήσεις πολύπλοκων αλγορίθμων όπως ο BFGS εμφανίζουν διαφορές από γλώσσα σε γλώσσα. Η μορφή της επιφάνειας απόφασης που φαίνεται αριστερά στο Σχήμα 3-6 προήλθε από την υλοποίηση της fminunc στην έκδοση 2008a του MATLAB όπου η συνάρτηση κόστους έλαβε την τιμή $J = 0.220$.

Σχήμα 3-6

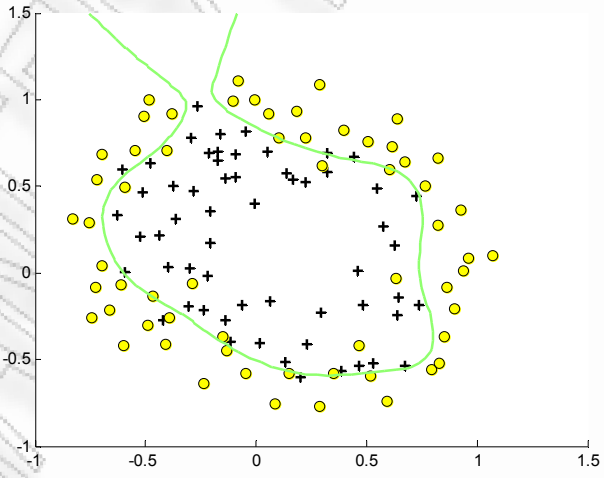
Δεδομένα εκπαίδευσης και όριο διαχωρισμού για $\lambda = 0$ και $\lambda = 100$ (MATLAB)



Στο Σχήμα 3-7 φαίνεται η επιφάνεια απόφασης που προέκυψε από την εφαρμογή του ίδιου κώδικα στην έκδοση 3.2.4 της Octave. Η συνάρτηση κόστους έλαβε εκεί την τιμή $J = 0.272$. Στην ίδια ουσιαστικά λύση κατέληξε και μια υλοποίηση με τη γλώσσα R (έκδοση 2.13.1) και την αντίστοιχη συνάρτηση βελτιστοποίησης ***optim*** ($J = 0.275$).

Σχήμα 3-7

Δεδομένα εκπαίδευσης και όριο διαχωρισμού για $\lambda = 0$ (Octave / R)



Σε κάθε περίπτωση πάντως, είναι προφανές το πόσο επιρρεπές καθίσταται το μοντέλο χωρίς ομαλοποίηση σε εξόφθαλμα σφάλματα κατά την ταξινόμηση νέων παρατηρήσεων.

Ως ένδειξη των επιπτώσεων της ομαλοποίησης, ζητήθηκε η πρόβλεψη ταξινόμησης του συνόλου των δεδομένων εκπαίδευσης, βάσει των μοντέλων με $\lambda = 1, \lambda = 0$ και $\lambda = 100$. Τα ποσοστά ακρίβειας[1] των προβλέψεων που σημειώθηκαν αντίστοιχα ήταν τα εξής: 83.05%, 88.98% και 61.02%.

Αφού προσδιοριστούν οι παράμετροι του μοντέλου, είμαστε πλέον σε θέση να το χρησιμοποιήσουμε για την πρόβλεψη της αποδοχής ή μη μικροεπεξεργαστών, βάσει των αποτελεσμάτων στα δύο τεστ.

Τα κυριότερα σημεία της υλοποίησης της συγκεκριμένης εφαρμογής δίνονται, σε γλώσσα Octave αλλά και σε R, στο Παράρτημα της σελ. 163.

Από την παραπάνω εφαρμογή γίνεται αντιληπτό ότι μέσω της τεχνητής δημιουργίας επιπλέον χαρακτηριστικών και της μη γραμμικής επιφάνειας απόφασης στην οποία καταλήξαμε μέσω αυτών, καταφέραμε να αντιμετωπίσουμε ικανοποιητικά το πρόβλημά μας με ένα μοντέλο λογιστικής παλινδρόμησης. Η τεχνική αυτή πράγματι δουλεύει καλά σε περιπτώσεις όπου τα διαθέσιμα χαρακτηριστικά είναι δύο, αλλά για τις περισσότερες εφαρμογές της πράξης τα χαρακτηριστικά που έχουμε στη διάθεσή μας είναι, κατά κανόνα, πολύ περισσότερα.

Σε περιπτώσεις όπου το πλήθος των διαθέσιμων χαρακτηριστικών είναι μεγάλο και ταυτόχρονα είναι επιθυμητή μια μη γραμμική επιφάνεια απόφασης, ακόμη κι αν εισαγάγουμε όρους αντίστοιχους με ενός πολυωνύμου μόλις $2^{ης}$ τάξης, το πλήθος των χαρακτηριστικών θα αυξηθεί κατά πολύ. Για παράδειγμα αν $n = 100$ θα καταλήγαμε σε περίπου 5000 χαρακτηριστικά, καθώς ασυμπτωτικά ο αριθμός των δευτεροβάθμιων όρων είναι τάξης $O(n^2)$ και ισούται περίπου με $n^2/2$. Βέβαια θα μπορούσαμε να χρησιμοποιήσουμε μόνο ένα υποσύνολο αυτών των όρων, π.χ. $x_1^2, x_2^2, \ldots, x_{100}^2$. Το σύνολο των χαρακτηριστικών τότε θα ήταν πράγματι κατά πολύ μικρότερο, αλλά όχι αρκετό για τη δημιουργία διαχωριστικών επιφανειών περισσότερο πολύπλοκων από επιφάνειες της μορφής υπερελλειψοειδών.

Κατ' επέκταση, αν επιχειρήσει κανείς να προχωρήσει στη δημιουργία όρων αντίστοιχων με ενός πολυωνύμου $3^{ης}$ τάξης, ο αριθμός των χαρακτηριστικών, που είναι πια τάξης $O(n^3)$, καθίσταται απαγορευτικός (π.χ. στην περίπτωση όπου $n = 100$ θα καταλήγαμε σε περίπου 170.000 χαρακτηριστικά).

[1] Με τον όρο *ακρίβεια* αναφερόμαστε στο ποσοστό των σωστών προβλέψεων. Η ακρίβεια αποτελεί ένα από τα πιο συνηθισμένα μέτρα της απόδοσης ενός μοντέλου.

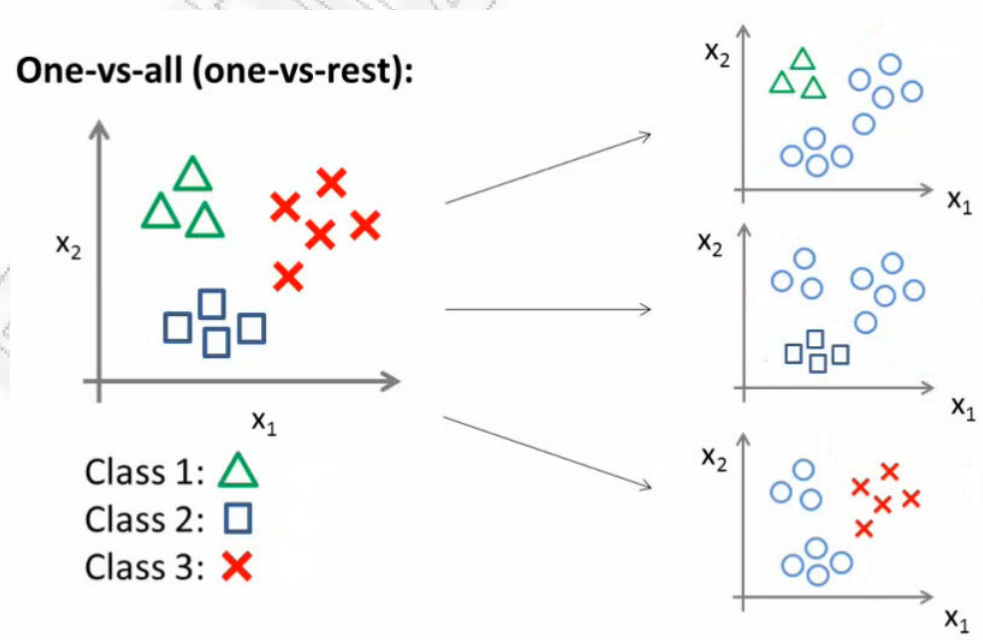
Το πρόβλημα γίνεται ανυπέρβλητο με μια τέτοια προσέγγιση όταν εργαζόμαστε σε εφαρμογές όπως λ.χ. η όραση υπολογιστών, όπου το πλήθος των αρχικά διαθέσιμων χαρακτηριστικών είναι της τάξης των αρκετών χιλιάδων. Κατά συνέπεια, σε τέτοιες περιπτώσεις, η μέθοδος της λογιστικής παλινδρόμησης παύει να αποτελεί αποδεκτό εργαλείο επίλυσης προβλημάτων ταξινόμησης μη γραμμικώς διαχωρίσιμων δεδομένων.

Πριν προχωρήσουμε στην παρουσίαση πολύ πιο αποτελεσματικών μεθόδων αντιμετώπισης τέτοιου είδους προβλημάτων (όπως είναι αυτή των νευρωνικών δικτύων), θα παρουσιαστεί μια εφαρμογή της ομαλοποιημένης λογιστικής παλινδρόμησης σε πρόβλημα πολλαπλής κατηγοριοποίησης, το οποίο θα αντιμετωπιστεί και σε επόμενα κεφάλαια με άλλους αλγορίθμους.

3.6 Εφαρμογή RLR: Ανάγνωση Χειρόγραφων Ψηφίων

Η εφαρμογή του μοντέλου της ομαλοποιημένης λογιστικής παλινδρόμησης σε περιπτώσεις όπου τα υπόψη αντικείμενα ανήκουν σε περισσότερες από δύο κλάσεις, γίνεται μέσω μιας απλής γενίκευσής του, με την ονομασία ***«ταξινόμηση ενός εναντίον όλων»*** **(*one-vs-all classification*)**. Στο Σχήμα 3-8 απεικονίζεται σχηματικά η ιδέα αυτή, για την περίπτωση τριών κλάσεων.

Σχήμα 3-8
Η ιδέα του one-vs-all classification



Όπως βλέπουμε, τα σημεία του συνόλου εκπαίδευσης συμβολίζονται αντίστοιχα με τρίγωνα, τετράγωνα και σταυρούς ανάλογα με την κλάση στην οποία ανήκουν. Σύμφωνα με τη μέθοδο one-vs-all, αυτό που κάνουμε είναι να δημιουργήσουμε, από το αρχικό σύνολο εκπαίδευσης, τόσα ξεχωριστά προβλήματα δυαδικής ταξινόμησης όσες και οι διαθέσιμες κλάσεις. Έτσι σαν πρώτο βήμα δημιουργούμε ένα ψευδοσύνολο εκπαίδευσης όπου στα σημεία που αντιστοιχούν στα τρίγωνα αντιστοιχίζουμε την κλάση 1 («επιτυχία») και σε όλα τα υπόλοιπα την κλάση 0 («αποτυχία»). Στη συνέχεια, με αντίστοιχο τρόπο, δημιουργούμε άλλα δύο ψευδοσύνολα όπου η κλάση 1 αντιστοιχίζεται διαδοχικά στα τετράγωνα και στους σταυρούς. Για όλα αυτά τα σύνολα εκπαίδευσης προσαρμόζουμε αντίστοιχα μοντέλα λογιστικής παλινδρόμησης, σύμφωνα με όσα έχουν αναφερθεί έως τώρα και εν τέλει καταλήγουμε σε τρεις ταξινομητές (όσες και οι διαθέσιμες κλάσεις), καθένας εκ των οποίων έχει εκπαιδευτεί να αναγνωρίζει κάθε μια εκ των τριών αυτών κλάσεων.

Προκειμένου να προβούμε σε πρόβλεψη κλάσης για μια νέα παρατήρηση, εφαρμόζουμε και τα τρία μοντέλα που δημιουργήσαμε και λαμβάνουμε αντίστοιχα τρείς πιθανότητες οι οποίες αντιστοιχούν στις πιθανότητες να ανήκει η νέα παρατήρηση σε κάθε μια από τις υπόψη κλάσεις. Η παρατήρηση τελικά ταξινομείται στην κλάση που αντιστοιχεί στην «επιτυχία» για το μοντέλο από το οποίο ελήφθη η μεγαλύτερη πιθανότητα επιτυχίας.

Στη μελέτη περίπτωσης που έχει υλοποιηθεί εφαρμόστηκε η παραπάνω μέθοδος προκειμένου να ταξινομηθούν ψηφιοποιημένες εικόνες χειρόγραφων αριθμητικών ψηφίων (από το 0 έως το 9) σε 10 κλάσεις οι οποίες αντιστοιχούν σε καθένα από αυτά τα ψηφία. Πρόκειται δηλαδή για μια εφαρμογή οπτικής αναγνώρισης χαρακτήρων (OCR). Η αυτοματοποιημένη αναγνώριση χειρόγραφων ψηφίων χρησιμοποιείται ευρέως σήμερα – από την αναγνώριση ταχυδρομικών κωδικών σε φακέλους αλληλογραφίας έως την αναγνώριση ποσών γραμμένων σε τραπεζικές επιταγές.

Το σύνολο των δεδομένων εκπαίδευσης περιλαμβάνει 5000 χειρόγραφα ψηφία, όπου καθένα από αυτά δεν είναι παρά μια εικόνα κάποιου ψηφίου, διαστάσεων 20x20 pixel, σε αποχρώσεις του γκρι[1]. Πρόσβαση στο συγκεκριμένο αρχείο δεδομένων αλλά και στο πλήρες σετ των αρχείων με τον κώδικα της εφαρμογής παρέχεται μέσω του συνδέσμου: http://bit.ly/w4uRzj. Κάθε pixel παριστάνεται από έναν πραγματικό αριθμό ο οποίος

[1] Σημειώνεται ότι το αρχείο δεδομένων που χρησιμοποιήθηκε είναι ένα υποσύνολο του αρχικού αρχείου χειρόγραφων ψηφίων MNIST (http://yann.lecun.com/exdb/mnist/)

αντιστοιχεί στην ένταση του γκρι χρώματος σε εκείνο το σημείο. Ένα τυχαίο δείγμα 100 τέτοιων εικόνων δίνεται στο Σχήμα 3-9.

Κάθε πλέγμα των 20x20 pixel έχει μετατραπεί σε ένα διάνυσμα 400 τιμών. Καθένα από αυτά τα διανύσματα, που αντιστοιχούν στα δείγματα εκπαίδευσης, αντιστοιχεί σε μία γραμμή του πίνακα δεδομένων του παραδείγματος, ο οποίος έχει διάσταση 5000 x 400.

Η μετατροπή αυτή του πίνακα των χαρακτηριστικών σε διάνυσμα δε γίνεται απλά με σκοπό τη μείωση των διαστάσεων του πίνακα δεδομένων, αλλά κυρίως γιατί οι συναρτήσεις βελτιστοποίησης (τύπου fminunc) απαιτούν ως είσοδο ένα *διάνυσμα* παραμέτρων και όχι έναν *πίνακα*. Κατά τον ίδιο τρόπο επιστρέφουν το gradient ως *διάνυσμα* και όχι ως *πίνακα*.

Σχήμα 3-9
Δείγμα 100 εικόνων με χειρόγραφα ψηφία

Τέλος, στα δεδομένα περιλαμβάνεται και ένα διάνυσμα 5000 τιμών, με τα αριθμητικά ψηφία στα οποία πραγματικά αντιστοιχεί ο χειρόγραφος αριθμός κάθε εικόνας. Στο συγκεκριμένο διάνυσμα, για λόγους προγραμματιστικής ευκολίας, τα ψηφία «1» έως «9» έχουν χαρακτηριστεί με τους αριθμούς «1» έως «9» στη φυσική τους σειρά, ενώ το ψηφίο «0» χαρακτηρίζεται με την τιμή «10».

Για την υλοποίηση του ταξινομητή, από θεωρητικής πλευράς, δεν απαιτείται τίποτε περισσότερο από αυτά που έχουν ήδη ειπωθεί. Το μόνο που θα παρατηρήσει κανείς στην

υλοποίηση είναι ότι η συνάρτηση ελαχιστοποίησης που χρησιμοποιείται αυτή τη φορά δεν είναι η fminunc αλλά η ***fmincg*** [25], η οποία δουλεύει κατά τον ίδιο τρόπο, αλλά έχει ταχύτερη απόκριση στις περιπτώσεις όπου ο αριθμός των παραμέτρων είναι μεγάλος.

Αφού προσδιοριστούν οι παράμετροι των μοντέλων, είμαστε σε θέση να χρησιμοποιήσουμε τον αλγόριθμο one-vs-all για την ανάγνωση (πρόβλεψη) του ψηφίου που απεικονίζει κάθε τέτοια νέα εικόνα, με βάση τις τιμές της έντασης του γκρι χρώματος των pixel που την αποτελούν (βλ. Σχήμα 3-10).

Ως ένδειξη της επίδοσης του μοντέλου ζητήθηκε η πρόβλεψη ταξινόμησης του συνόλου των δεδομένων εκπαίδευσης. Το ποσοστό ακρίβειας των προβλέψεων ήταν 94.90% (95.14% στο MATLAB).

Σχήμα 3-10
Εφαρμογή του one-vs-all για την ανάγνωση χειρόγραφου ψηφίου

Τα κυριότερα σημεία της υλοποίησης της συγκεκριμένης εφαρμογής δίνονται, σε γλώσσα Octave, στο Παράρτημα της σελ. 166.

Κ Ε Φ Α Λ Α Ι Ο 4

Τα Μοντέλα Perceptron και MLP

4.1 Εισαγωγή

Το μοντέλο perceptron μπορεί να θεωρηθεί ως ο πρόγονος των σύγχρονων τεχνητών νευρωνικών δικτύων. Αποτελείται από έναν και μοναδικό νευρώνα ο οποίος χρησιμοποιεί μια δυαδική συνάρτηση απόφασης βασισμένη σε μια γραμμική επιφάνεια απόφασης. Η «μάθηση» στη περίπτωση του απλού αυτού δικτύου έγκειται στην εκτίμηση, μέσω ενός αλγορίθμου εκπαίδευσης, του καθέτου διανύσματος και της μετάθεσης, που καθορίζουν τη γραμμική επιφάνεια απόφασης σύμφωνα με όσα αναφέρθηκαν στο Κεφ. 2. Αφού ολοκληρωθεί η διαδικασία εκπαίδευσης, το μοντέλο perceptron μπορεί να χρησιμοποιηθεί για την ταξινόμηση νέων παρατηρήσεων του πληθυσμού.

Η μέθοδος που χρησιμοποιείται για την εκπαίδευση του perceptron είναι *ευρετική*[1] (*heuristic*). Αναζητά δηλαδή *κάποια* επιφάνεια απόφασης, όχι απαραίτητα βέλτιστη, για ένα γραμμικώς διαχωρίσιμο σύνολο εκπαίδευσης, μέσω εμπειρικών μεθόδων και προσεγγίσεων, σε ολόκληρο το χώρο των πιθανών κάθετων διανυσμάτων και όρων μετάθεσης.

Μια ιδιαίτερα ενδιαφέρουσα ιδιότητα του μοντέλου perceptron είναι ότι εκτός του συνήθους αλγορίθμου εκπαίδευσης, μπορεί να διατυπωθεί μια δυϊκή μορφή αυτού. Η μάθηση στην περίπτωση αυτή έγκειται στον εντοπισμό, πέραν του όρου μετάθεσης, μιας σειράς συντελεστών επιρροής. Οι συντελεστές αυτοί είναι αντιπροσωπευτικοί του βαθμού επιρροής κάθε σημείου του συνόλου εκπαίδευσης στη τελική θέση της επιφάνειας απόφασης. Αυτοί οι συντελεστές επιρροής μαζί με τον όρο μετάθεσης είναι που καθορίζουν την επιφάνεια

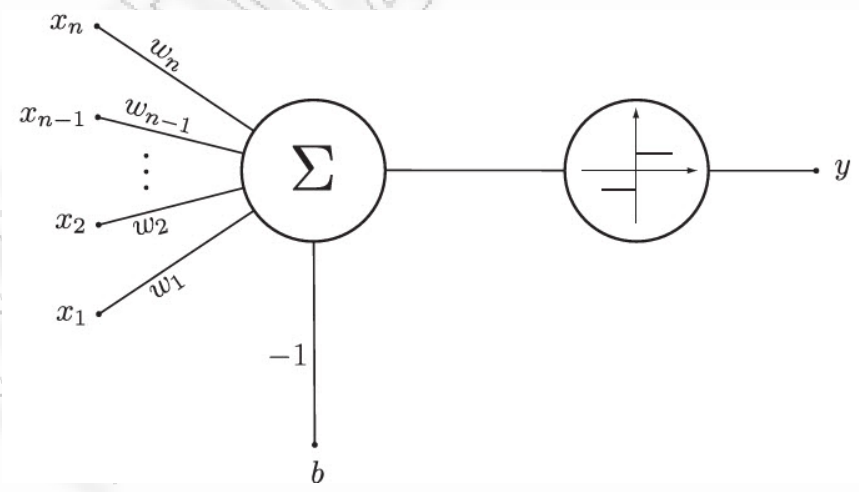
[1] Μια ευρετική μέθοδος είναι ένας αλγόριθμος που χρησιμοποιεί εμπειρικούς κανόνες και προσεγγίσεις για να εντοπίσει *κάποια* λύση σε ένα δεδομένο πρόβλημα και όχι τη βέλτιστη λύση.

απόφασης, σε αντίθεση με τη βασική προσέγγιση, κατά την οποία η επιφάνεια απόφασης καθορίζεται από ένα κάθετο διάνυσμα και έναν όρο μετάθεσης. Είναι αξιοσημείωτο ότι με τη δυϊκή μορφή του αλγορίθμου εκπαίδευσης του perceptron, καθίσταται σαφές το ποια ακριβώς σημεία περιορίζουν τη θέση της επιφάνειας απόφασης και ποια όχι. Αυτό αποτελεί μια πρώτη προσέγγιση για κάποιες από τις δομές που χρησιμοποιούνται στις μηχανές διανυσμάτων υποστήριξης.

4.2 Αρχιτεκτονική και Αλγόριθμος Εκπαίδευσης

Στο Κεφ. 2 είδαμε ότι σε ένα πρόβλημα δυαδικής ταξινόμησης, μπορούμε να υπολογίσουμε μια επιφάνεια απόφασης και παράλληλα την αντίστοιχη συνάρτηση απόφασης, βασισμένοι στα κέντρα βάρους των δύο κλάσεων. Στη δεκαετία του '50, ο Frank Rosenblatt πρότεινε μια εντελώς διαφορετική προσέγγιση για την εύρεση της συνάρτησης απόφασης. Πρότεινε μια μηχανή (το perceptron) που κωδικοποιεί τη δομή της συνάρτησης απόφασης, με βάση μια υποκείμενη γραμμική επιφάνεια απόφασης. Σήμερα η αρχιτεκτονική αυτή περιγράφεται ως ένας τεχνητός νευρώνας, αποτελούμενος από δύο υπολογιστικά μέρη: μια συνάρτηση συνάθροισης (aggregation function) και μια συνάρτηση ενεργοποίησης (activation function). Η δομή ενός τέτοιου νευρώνα απεικονίζεται στο Σχήμα 4-1.

Σχήμα 4-1
Η αρχιτεκτονική του δικτύου perceptron



Το perceptron συλλέγει στο στοιχείο του αθροιστή n σταθμισμένα σήματα εισόδου $x_1, x_2, \ldots, x_n$ με αντίστοιχα βάρη $w_1, w_2, \ldots, w_n$ και τα αθροίζει. Παρατηρήστε ότι στον

αθροιστή καταλήγει και ένας όρος b με σταθερό βάρος -1 που καλείται πόλωση (bias)[1]. Το άθροισμα κατόπιν περνάει στη συνάρτηση ενεργοποίησης, που δεν είναι παρά η βηματική συνάρτηση (sgn), η οποία επιστρέφει την τιμή -1 αν η είσοδός του είναι αρνητική και $+1$ αν είναι θετική. Δεν είναι δύσκολο να εκφράσουμε τη λειτουργία του perceptron με την ακόλουθη μορφή:

$$y = \operatorname{sgn}\left(\left[\sum_{k=1}^{n} w_k x_k\right] + (-1)b\right) = \operatorname{sgn}\left(\bar{w} \bullet \bar{x} - b\right). \tag{4.1}$$

Στην παραπάνω σχέση παρατηρούμε ότι το διάνυσμα των βαρών αντιστοιχεί στο κάθετο διάνυσμα μιας επιφάνειας απόφασης, ενώ η πόλωση στην αντίστοιχη μετάθεση, σύμφωνα με τα όσα έχουν αναφερθεί στο Κεφ. 2. Επιπλέον, μέσω της βηματικής συνάρτησης λαμβάνεται και ο χαρακτηρισμός (label) κάθε αντικειμένου. Το μοντέλο perceptron λοιπόν είναι αντίστοιχο με μια συνάρτηση απόφασης της μορφής

$$\hat{f}\left(\bar{x}\right) = y = \operatorname{sgn}\left(\bar{w} \bullet \bar{x} - b\right). \tag{4.2}$$

Οι παράμετροι $\bar{w}$ και b καλούνται ελεύθερες παράμετροι της συνάρτησης απόφασης, με την έννοια ότι καθορίζονται από τα δεδομένα εκπαίδευσης. Αντί να χρησιμοποιήσει κάποια στατιστική τεχνική για τον υπολογισμό αυτών των παραμέτρων, ο Rosenblatt χρησιμοποίησε μια εμπειρική διαδικασία αναζήτησης κατάλληλων τιμών γι' αυτές, μέσω διαδοχικών βελτιώσεων επί μιας αρχικής τυχαίας επιφάνειας, μέχρι αυτή να γίνει αποδεκτή ως επιφάνεια απόφασης. Έστω ότι το αρχικό, γραμμικά διαχωρίσιμο σύνολο εκπαίδευσης έχει τη μορφή

$$D = \left\{\left(\bar{x}_1, y_1\right), \left(\bar{x}_2, y_2\right), \ldots, \left(\bar{x}_l, y_l\right)\right\}\ , \tag{4.3}$$

με $\bar{x}_i \in \mathbb{R}^n$ και $y_i \in \{+1, -1\}$. Ο αλγόριθμος εκπαίδευσης μπορεί συνοπτικά να δοθεί ως εξής:

Αρχικοποίησε τα $\bar{w}$ και b με τυχαίες τιμές
repeat
  for each $(\bar{x}_i, y_i) \in D$ **do**
    if $\hat{f}(\bar{x}_i) \neq y_i$ **then**
      Διόρθωσε τις τιμές των $\bar{w}$ και b
    end if
  end for
until το D να είναι τέλεια ταξινομημένο
return $\bar{w}$ και b

[1] Στην ορολογία των νευρωνικών δικτύων ο όρος bias χρησιμοποιείται με την έννοια της αποτέμνουσας (intercept).

Ο αλγόριθμος ελέγχει τη συνάρτηση απόφασης $\hat{f}$ για κάθε αντικείμενο του συνόλου εκπαίδευσης και στην περίπτωση που αποτύχει στη σωστή ταξινόμησή του, προσαρμόζει τις ελεύθερες παραμέτρους αυξομειώνοντας τις τρέχουσες τιμές τους. Η διαδικασία αυτή συνεχίζεται μέχρι να ταξινομηθούν σωστά όλα τα αντικείμενα του συνόλου εκπαίδευσης. Εδώ, η υπόθεση του γραμμικώς διαχωρίσιμου συνόλου παίζει πολύ σημαντικό ρόλο, καθώς ο αλγόριθμος αυτός εξασφαλίζει τη σύγκλιση[1] μόνο όταν τα δεδομένα του συνόλου εκπαίδευσης είναι γραμμικά διαχωρίσιμα (αλλιώς θα εισέλθει σε ατέρμονα βρόγχο).

Χρησιμοποιώντας τεχνική ορολογία θα λέγαμε ότι ο αλγόριθμος αυτός συνιστά μια ευρετική μέθοδο τύπου *«άπληστης αναζήτησης»* (greedy search) στο χώρο $\bar{w}$ - b, όπου οι τιμές του διανύσματος $\bar{w}$ και της μετάθεσης b επανακαθορίζονται μέχρι να βρεθεί μια κατάλληλη επιφάνεια απόφασης. Η ονομασία αυτού του τύπου αναζήτησης σχετίζεται με το γεγονός ότι η τρέχουσα λύση βελτιώνεται διαρκώς, κατά την εξέταση κάθε στοιχείου του χώρου αναζήτησης, χωρίς ο αλγόριθμος να επιστρέφει ποτέ σε προηγούμενο στάδιο για να διερευνήσει τυχόν εναλλακτικές λύσεις[2].

Αλγόριθμος 4.1

let $D = \{(\bar{x}_1, y_1), (\bar{x}_2, y_2), \ldots, (\bar{x}_l, y_l)\} \subset \mathbb{R}^n \times \{+1, -1\}$
let $0 < \eta < 1$
$\bar{w} \leftarrow \bar{0}$
$b \leftarrow 0$
$r \leftarrow \max\{|\bar{x}| \,|\, (\bar{x}, y) \in D\}$
repeat
  for $i = 1$ **to** l
    if $\text{sgn}(\bar{w} \bullet \bar{x}_i - b) \neq y_i$ **then**
      $\bar{w} \leftarrow \bar{w} + \eta y_i \bar{x}_i$
      $b \leftarrow b - \eta y_i r^2$
    end if
  end for
until $\text{sgn}(\bar{w} \bullet \bar{x}_j - b) = y_j$ με $j = 1, \ldots, l$
return $(\bar{w}, b)$

[1] Για την απόδειξη ότι ο αλγόριθμος θα βρει μια λύση σε πεπερασμένο αριθμό επαναλήψεων βλ. [16], Κεφ. 10, §5.5

[2] Αυτή η τεχνική αποτελεί πρόδρομο αντίστοιχων τεχνικών που χρησιμοποιούνται στα σύγχρονα πολυεπίπεδα νευρωνικά δίκτυα, όπως η backpropagation.

Ο Αλγόριθμος 4.1 δείχνει λεπτομερέστερα τη διαδικασία εκπαίδευσης του perceptron. Η ποσότητα r καλείται *ακτίνα* του συνόλου εκπαίδευσης και στην ουσία αποτελεί την ακτίνα της υπερσφαίρας με κέντρο την αρχή των αξόνων η οποία περικλείει όλα τα σημεία του συνόλου εκπαίδευσης. Στην περίπτωση όπου ο πληθυσμός δεδομένων είναι ο $\mathbb{R}^n$, πρόκειται απλά για το μήκος του διανύσματος θέσης του πιο απομακρυσμένου, από την αρχή των αξόνων, σημείου στο σύνολο εκπαίδευσης. Η ποσότητα η καλείται *ρυθμός εκμάθησης* και ελέγχει την ταχύτητα σύγκλισης της διαδικασίας αναζήτησης επιφάνειας απόφασης. Στη καρδιά του αλγορίθμου υπάρχουν οι δύο παρακάτω κανόνες αναπροσαρμογής:

$$\bar{w} \leftarrow \bar{w} + \eta y_i \bar{x}_i \tag{4.4}$$

$$b \leftarrow b - \eta y_i r^2. \tag{4.5}$$

Στην περίπτωση εσφαλμένης ταξινόμησης σημείου, οι δύο αυτοί κανόνες επιχειρούν να διορθώσουν τη θέση της επιφάνειας απόφασης έτσι ώστε το εν λόγω σημείο να ταξινομείται σωστά.

Θεωρήστε για παράδειγμα το $(\bar{x}_i, y_i)$ με $y_i = +1$. Αν το σημείο αυτό δεν ταξινομείται σωστά με βάση την τρέχουσα επιφάνεια απόφασης, θα λαμβάνει το χαρακτηρισμό -1 αντί του σωστού $+1$. Ο πρώτος κανόνας (4.4) επιχειρεί να διορθώσει το σφάλμα *περιστρέφοντας* την επιφάνεια απόφασης προς τη κατεύθυνση του $\bar{x}_i$. Η περιστροφή πραγματοποιείται προσθέτοντας μια σταθμισμένη εκδοχή του $\bar{x}_i$ στο κάθετο διάνυσμα $\bar{w}$. Ο συντελεστής στάθμισης ισούται με το ρυθμό εκμάθησης η. Αυτό φαίνεται καλύτερα αν ξαναγράψουμε τον κανόνα (4.4) ως εξής:

$$\bar{w} \leftarrow \bar{w} + \eta \bar{x}_i \tag{4.6}$$

όπου το y_i έχει λάβει την τιμή $+1$. Η επίδραση της διόρθωσης αυτής σε μια επιφάνεια απόφασης στο $\mathbb{R}^2$ φαίνεται στο Σχήμα 4-2. Εδώ το σημείο $\bar{x}_i$ βρίσκεται κάτω από την αρχική επιφάνεια απόφασης (γκρι) και, μετά τη διόρθωση, ταξινομείται σωστά αφού πλέον βρίσκεται πάνω από την περιεστρεμμένη (μαύρη).

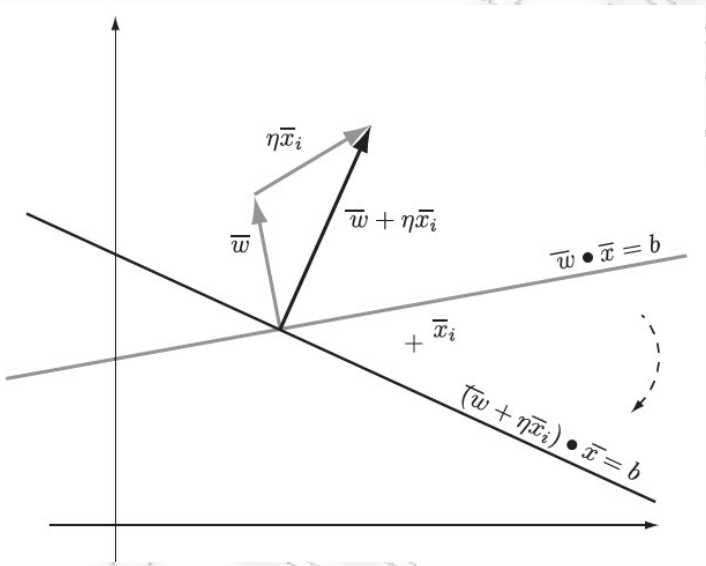
Ανάλογα θα αντιμετωπιστεί και εσφαλμένα ταξινομημένο σημείο με $y_i = -1$. Τώρα η σταθμισμένη εκδοχή του $\bar{x}_i$ θα αφαιρεθεί από το κάθετο διάνυσμα $\bar{w}$, προκαλώντας περιστροφή στην αντίθετη κατεύθυνση.

Ο δεύτερος κανόνας (4.5) προσπαθεί να διορθώσει το λάθος *μετατοπίζοντας* την επιφάνεια απόφασης. Παρατηρήστε ότι η μετατόπιση της επιφάνειας θα γίνει προς την αντίθετη κατεύθυνση από αυτήν που βρίσκεται η επιθυμητή για το σημείο κλάση.

Αυτό φαίνεται καλύτερα αν ξαναγράψουμε τον κανόνα (4.5) ως εξής:

$$b \leftarrow b + (-y_i)\eta r^2. \tag{4.7}$$

Σχήμα 4-2
Επίδραση του κανόνα μεταβολής του $\overline{w}$

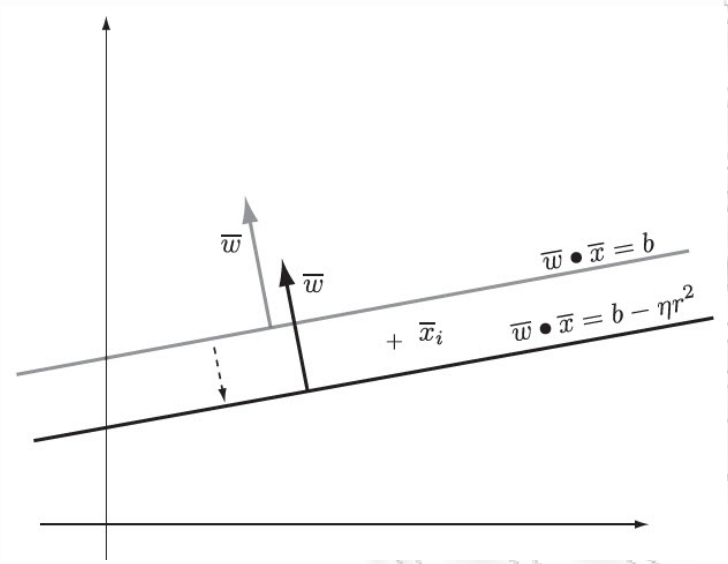


Προκειμένου να γίνει κατανοητό γιατί η διόρθωση αυτή είναι η κατάλληλη, ας θεωρήσουμε ότι η τρέχουσα επιφάνεια απόφασης ταξινομεί εσφαλμένα ένα σημείο $(\overline{x}_i, y_i)$ με $y_i = +1$.

Αυτό σημαίνει ότι το σημείο βρίσκεται κάτω από την επιφάνεια απόφασης και μια διόρθωση θα απαιτούσε τη μετατόπιση της επιφάνειας απόφασης σε αντίθετη κατεύθυνση από αυτή που δείχνει το κάθετο διάνυσμα $\overline{w}$, ή με άλλα λόγια στην αντίθετη κατεύθυνση από αυτή που βρίσκονται τα αντικείμενα με χαρακτηρισμό $+1$. Το μέγεθος της μετατόπισης εξαρτάται από το ρυθμό εκμάθησης και την ακτίνα του συνόλου εκπαίδευσης. Στο Σχήμα 4-3 φαίνεται η επίδραση της εφαρμογής αυτού του κανόνα σε μια επιφάνεια απόφασης στο $\mathbb{R}^2$. Συγκεκριμένα, το σημείο $\overline{x}_i$ ταξινομείται εσφαλμένα από την αρχική επιφάνεια απόφασης (γκρι), αλλά σωστά από τη μετατεθειμένη (μαύρη).

Σχήμα 4-3

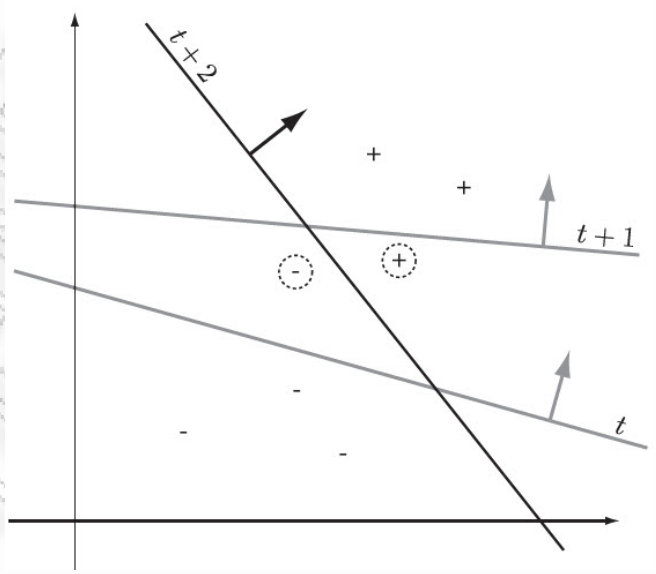
Επίδραση του κανόνα μεταβολής του όρου μετάθεσης b



Η συνολική επίδραση και των δύο αυτών κανόνων για μια επιφάνεια απόφασης του $\mathbb{R}^2$ παρουσιάζεται στο Σχήμα 4-4. Κατά τη χρονική στιγμή του βήματος t η επιφάνεια ταξινομεί εσφαλμένα το σημείο στον κύκλο με χαρακτηρισμό -1 και ο αλγόριθμος εφαρμόζει τους δύο παραπάνω κανόνες αναπροσαρμογής.

Σχήμα 4-4

Η θέση της επιφάνειας απόφασης στα βήματα $t, t+1, t+2$



Η τελική επιφάνεια, μετά την περιστροφή και μετάθεση φαίνεται για τη χρονική στιγμή $t+1$. Μετά τη διόρθωση αυτή όμως, ένα από τα σημεία με χαρακτηρισμό $+1$ (που επίσης εμφανίζεται σε κύκλο) βρίσκεται να είναι λάθος ταξινομημένο. Οι διορθωτικοί κανόνες ενεργοποιούνται εκ νέου προκειμένου να διορθωθεί το νέο σφάλμα, οδηγώντας σε μια νέα επιφάνεια (στη θέση $t+2$), όπου όλα τα σημεία ταξινομούνται σωστά, με αποτέλεσμα ο αλγόριθμος να τερματίσει.

Μετά από αυτή τη σύντομη παρουσίαση του τρόπου λειτουργίας του αλγορίθμου, θα πρέπει να είναι μάλλον προφανές ότι οι περισσότερες εφαρμογές των διορθωτικών αυτών κανόνων θα συμβούν για τα σημεία που βρίσκονται κοντά στα όρια μεταξύ των δύο κλάσεων, καθώς αυτά παρουσιάζουν τη μεγαλύτερη δυσκολία να διαχωριστούν και να ταξινομηθούν σωστά. Με άλλα λόγια θα πρέπει να γίνεται αντιληπτό ότι η τελική επιφάνεια απόφασης θα πρέπει να είναι κοντά στο όριο μεταξύ των δύο κλάσεων και τα σημεία που βρίσκονται κοντά στο όριο θα ταξινομούνται εσφαλμένα πολύ πιο συχνά από τα υπόλοιπα. Κατά συνέπεια τα σημεία αυτά θα προκαλούν τη μεταβολή της επιφάνειας απόφασης, κατά τη διάρκεια της εκπαίδευσης του αλγορίθμου, πολύ πιο συχνά από αυτά που βρίσκονται μακριά από αυτό το όριο. Αυτό διαφέρει σημαντικά από ότι συνέβαινε στον απλό αλγόριθμο μάθησης του Κεφ. 2, όπου κάθε σημείο συνεισέφερε το ίδιο στον καθορισμό της θέσης της επιφάνεια απόφασης, συμμετέχοντας απλώς στον υπολογισμό του αντίστοιχου κέντρου βάρους. Το γεγονός αυτό, ότι δηλαδή τα σημεία κοντά στο όριο συνεισφέρουν περισσότερο στην αναζήτηση μιας κατάλληλης επιφάνειας απόφασης δίνει την αφορμή για μια εναλλακτική ή *δυϊκή μορφή* του αλγορίθμου εκπαίδευσης perceptron.

4.3 Δυϊσμός

Είδαμε ότι ο αλγόριθμος perceptron επαναλαμβάνεται έως ότου δεν υπάρχει καμία λανθασμένη ταξινόμηση στο δείγμα. Υπενθυμίζεται ότι βάσει του κανόνα διόρθωσης του $\overline{w}$ (4.4), προστίθεται σε αυτό η ποσότητα $\eta y_i \overline{x}_i$ κάθε φορά που το σημείο $\overline{x}_i$ ταξινομείται εσφαλμένα και αυτό επιφέρει μια περιστροφή της επιφάνειας απόφασης. Αν ένα συγκεκριμένο $\overline{x}_i$ παρουσιάζει ιδιαίτερη δυσκολία στο να ταξινομηθεί σωστά, θα πρέπει ενδεχομένως να εφαρμοστεί ο κανόνας διόρθωσης του $\overline{w}$ πολλές φορές έως ότου η επιφάνεια απόφασης στραφεί αρκετά ώστε να το ταξινομήσει σωστά. Το ίδιο ισχύει και για τον κανόνα μετατόπισης (4.5) όπου επίσης μπορεί να χρειαστεί να μετακινηθεί η επιφάνεια απόφασης

πολλές φορές έως ότου το σημείο να ταξινομηθεί σωστά. Έτσι, σημεία που είναι δύσκολο να ταξινομηθούν σωστά συνεισφέρουν περισσότερο στον υπολογισμό της επιφάνειας απόφασης απ' ότι όσα είναι εύκολο να ταξινομηθούν.

Ας θεωρήσουμε τώρα ότι εισάγουμε ένα μετρητή του αριθμού των εσφαλμένων ταξινομήσεων κάθε σημείου του συνόλου εκπαίδευσης. Θα περιμέναμε ότι σημεία δύσκολο να ταξινομηθούν θα είχαν μεγάλη τιμή στο μετρητή, ενώ σημεία εύκολα ταξινομήσιμα θα είχαν είτε μηδενική τιμή, είτε μια τιμή κοντά στο 0. Έστω ότι ο μετρητής έχει τη μορφή του διανύσματος $\bar{\alpha} = (\alpha_1, \ldots, \alpha_l)$, με κάθε του στοιχείο να αντιστοιχεί και σε ένα σημείο του συνόλου εκπαίδευσης. Το στοιχείο α_i θα μετράει το πόσες φορές ταξινομήθηκε εσφαλμένα το σημείο $\bar{x}_i$. Μπορούμε τώρα να τροποποιήσουμε τον αλγόριθμο perceptron ως εξής:

Αρχικοποίησε τα $\bar{\alpha}$ και b με 0
repeat
  for each $(\bar{x}_i, y_i) \in D$ **do**
    if $\hat{f}(\bar{x}_i) \neq y_i$ **then**
      Αύξησε το α_i κατά 1
      Διόρθωσε την τιμή του b
    end if
  end for
until το D να είναι τέλεια ταξινομημένο
return $\bar{\alpha}$ και b

Παρατηρήστε ότι στον εσώτερο βρόγχο του αλγορίθμου έχει αντικατασταθεί ο κανόνας διόρθωσης του κάθετου διανύσματος $\bar{w}$ με ένα κανόνα που αυξάνει την τιμή του στοιχείου α_i. Ο αλγόριθμος αυτός όμως δεν υπολογίζει την απαραίτητη για τον καθορισμό της συνάρτησης $\hat{f}$, παράμετρο $\bar{w}$. Παρόλα αυτά είναι δυνατόν αυτή να ανακτηθεί, με τη βοήθεια του διανύσματος-μετρητή $\bar{\alpha}$. Αναλογιζόμενοι το γεγονός ότι το κάθετο διάνυσμα $\bar{w}$ προκύπτει από ένα γραμμικό συνδυασμό σταθμισμένων εκδοχών των εσφαλμένα ταξινομημένων σημείων, θα ισχύει:

$$\bar{w} = \sum_{i=1}^{l} \eta \alpha_i y_i \bar{x}_i = \eta \sum_{i=1}^{l} \alpha_i y_i \bar{x}_i. \qquad (4.8)$$

Καθώς μόνο σημεία με σφάλματα ταξινόμησης έχουν μη μηδενική τιμή α, η σχέση (4.8) εκφράζει γραμμικό συνδυασμό μόνο σημείων εσφαλμένα ταξινομημένων. Στην έκφραση αυτή ο ρυθμός εκμάθησης η αποτελεί απλά μια σταθερά που σταθμίζει τις τιμές των στοιχείων του διανύσματος $\bar{w}$ και καθώς αυτό που μας ενδιαφέρει είναι κυρίως η

κατεύθυνση του διανύσματος, ο συντελεστής αυτός συνήθως αγνοείται και η σχέση (4.8) γράφεται:

$$\bar{w} = \sum_{i=1}^{l} \alpha_i y_i \bar{x}_i \tag{4.9}$$

με $\alpha_i \approx 0$ για «εύκολα» σημεία,

$\alpha_i \gg 1$ για «δύσκολα» σημεία.

Με βάση τα παραπάνω η συνάρτηση απόφασης $\hat{f}$ μπορεί να γραφεί στη μορφή:

$$\hat{f}(\bar{x}) = \operatorname{sgn}(\bar{w} \bullet \bar{x} - b) = \operatorname{sgn}\left(\sum_{j=1}^{l} \alpha_j y_j \bar{x}_j \bullet \bar{x} - b \right). \tag{4.10}$$

Στην παραπάνω έκφραση οι ελεύθερες παράμετροι είναι πλέον οι $\bar{\alpha}$ και b, και ο τροποποιημένος αλγόριθμος υπολογίζει κατάλληλες τιμές για τις δύο παραμέτρους αυτές. Με τη νέα αυτή μορφή του αλγορίθμου το πρόβλημα εύρεσης των $\bar{w}$ και b μετατράπηκε σε ένα πρόβλημα εκτίμησης των $\bar{\alpha}$ και b. Το αρχικά διατυπωμένο πρόβλημα ως προς το $\bar{w}$ καλείται *πρωταρχικό πρόβλημα* με *πρωταρχική παράμετρο* το $\bar{w}$. Το τροποποιημένο πρόβλημα ως προς το $\bar{\alpha}$ καλείται *δυϊκό πρόβλημα* με *δυϊκή παράμετρο* το $\bar{\alpha}$. Καθώς δυνάμεθα να χρησιμοποιήσουμε τη δυϊκή παράμετρο προκειμένου να κατασκευάσουμε μια λύση για το πρωταρχικό πρόβλημα, είναι προφανές ότι μπορεί να χρησιμοποιηθεί οποιοσδήποτε από τους δύο αλγορίθμους για να βρεθεί μια κατάλληλη επιφάνεια απόφασης.

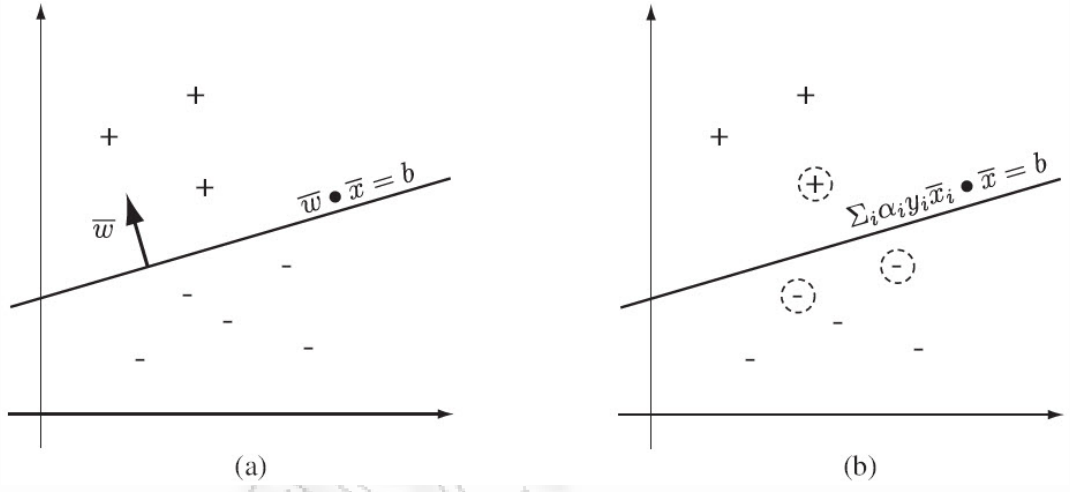
Αλγόριθμος 4.2

let $D = \{(\bar{x}_1, y_1), (\bar{x}_2, y_2), \ldots, (\bar{x}_l, y_l)\} \subset \mathbb{R}^n \times \{+1, -1\}$
let $0 < \eta < 1$
$\bar{\alpha} \leftarrow \bar{0}$
$b \leftarrow 0$
$r \leftarrow \max\{\|\bar{x}\| \mid (\bar{x}, y) \in D\}$
repeat
  for $i = 1$ **to** l
    if $\operatorname{sgn}\left(\sum_{j=1}^{l} \alpha_j y_j \bar{x}_j \bullet \bar{x}_i - b\right) \neq y_i$ **then**
      $\alpha_i \leftarrow \alpha_i + 1$
      $b \leftarrow b - \eta y_i r^2$
    end if
  end for
until $\operatorname{sgn}\left(\sum_{j=1}^{l} \alpha_j y_j \bar{x}_j \bullet \bar{x}_k - b\right) = y_k$ με $k = 1, \ldots, l$
return $(\bar{\alpha}, b)$

Ο Αλγόριθμος 4.2 εμφανίζει αναλυτικά αυτή τη δυϊκή εκδοχή του αλγορίθμου εκπαίδευσης perceptron.

Εν κατακλείδι, η πρωταρχική προσέγγιση στην εκπαίδευση του perceptron αναζητά λύση στο χώρο $\overline{w}$ - b και καταλήγει σε μια διατύπωση της επιφάνειας απόφασης αναφορικά με ένα κάθετο διάνυσμα $\overline{w}$ και έναν όρο μετάθεσης b (Σχήμα 4-5a). Στη δυϊκή προσέγγιση για την εκπαίδευση του perceptron ο αλγόριθμος αναζητά λύση στο χώρο $\overline{\alpha}$ - b και κατασκευάζει την επιφάνεια απόφασης χρησιμοποιώντας το μετρητή $\overline{\alpha}$ και τον όρο μετάθεσης b (Σχήμα 4-5b).

Σχήμα 4-5
Μορφή της επιφάνειας απόφασης perceptron: a) πρωταρχική, b) δυϊκή



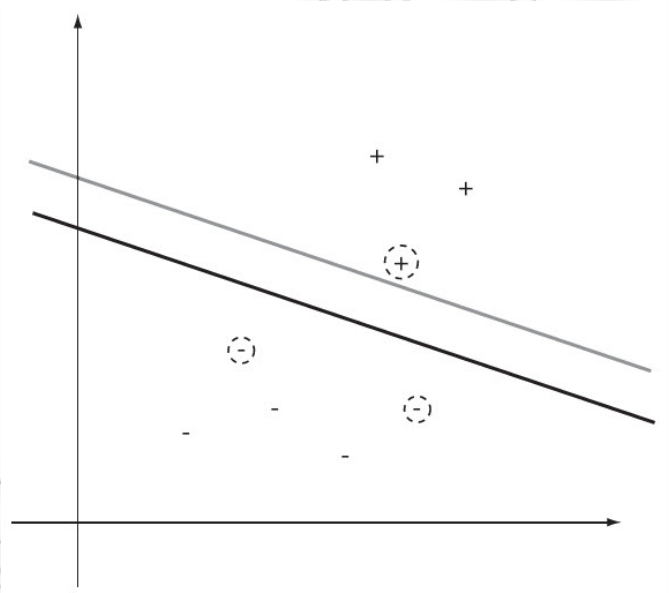
Στη δυϊκή μορφή του αλγορίθμου καθίσταται προφανές το ποια σημεία θέτουν τους περισσότερους περιορισμούς στον προσδιορισμό της επιφάνειας απόφασης. Τα σημεία αυτά έχουν τιμές $\alpha \gg 1$ και παρουσιάζονται εντός κύκλου στο παραπάνω σχήμα.

Ο δυϊσμός, ή η δυϊκή προσέγγιση στο πρόβλημα ταξινόμησης, έχει ενδιαφέρουσες επιπτώσεις στο σχεδιασμό των αλγορίθμων, καθώς φέρνει στην επιφάνεια κρυμμένους περιορισμούς του προβλήματος ταξινόμησης. Στη συγκεκριμένη περίπτωση κατέστη σαφές ότι η δυϊκή λύση διέπεται από τους περιορισμούς που επιβάλλουν τα σημεία που βρίσκονται στη γειτονιά του ορίου μεταξύ των κλάσεων. Κατά συνέπεια, ο δυϊσμός είναι ένα αποτελεσματικό εργαλείο σχεδιασμού αλγορίθμων που επιτρέπει τη διερεύνηση διαφορετικών εναλλακτικών προσεγγίσεων. Αυτό θα αποδειχθεί εξαιρετικά σημαντικό κατά την ανάπτυξη των μηχανών διανυσμάτων υποστήριξης.

Είναι επίσης σαφές ότι ο αλγόριθμος perceptron αντιμετωπίζει το πρόβλημα των ακραίων παρατηρήσεων που αναφέρθηκε στο Κεφ. 2, καθώς μόνο τα σημεία κοντά στο όριο επιδρούν στον καθορισμό της επιφάνειας απόφασης.

Ένα επακόλουθο της ευρετικής φύσης και των δύο μορφών του αλγορίθμου είναι ότι η αναζήτηση της επιφάνειας απόφασης τερματίζεται αμέσως μόλις βρεθεί μια επιφάνεια που διαχωρίζει επιτυχώς το σύνολο εκπαίδευσης. Αυτό μπορεί να έχει ως αποτέλεσμα την επιλογή υποβέλτιστων επιφανειών, τοποθετημένων αδικαιολόγητα κοντά σε κάποια από τα σημεία του συνόλου, με αποτέλεσμα να υπάρχει μεγαλύτερη πιθανότητα σφαλμάτων κατά την ταξινόμηση νέων σημείων του πληθυσμού.

Σχήμα 4-6
Υποβέλτιστη επιφάνεια (γκρι) και εναλλακτική (μαύρη) τοποθετημένη στο μέσο της απόστασης μεταξύ των σημείων των δύο κλάσεων με τις υψηλότερες τιμές α



Στο Σχήμα 4-6 η γκρι επιφάνεια είναι τοποθετημένη πολύ κοντά σε σημείο του δείγματος αν και υπάρχει πολύς χώρος για διαφορετική τοποθέτηση. Διαισθητικά θα περιμέναμε μια πιο αποτελεσματική επιφάνεια να βρίσκεται στο μέσο του διαστήματος μεταξύ των σημείων με υψηλές τιμές του α. Οι ταξινομητές μεγίστου περιθωρίου, που θα παρουσιαστούν στο Κεφ. 5, αντιμετωπίζουν αυτό το πρόβλημα υπό την έννοια ότι οι αντίστοιχοι αλγόριθμοι καταλήγουν σίγουρα στην επιφάνεια απόφασης που βρίσκεται στο μέσο μεταξύ των ορίων των δύο κλάσεων.

4.4 Το Δίκτυο MLP

Όπως είδαμε, οι δυνατότητες καθορισμού διαχωριστικών επιφανειών είναι περιορισμένες στο δίκτυο perceptron, καθώς έχοντας ένα μόνο νευρώνα μπορεί να παραστήσει μόνο επίπεδες επιφάνειες. Πριν προχωρήσουμε στη χρήση συναρτήσεων πυρήνα για τη διαμόρφωση μη γραμμικών επιφανειών, θα αναφερθούμε σε ένα προγενέστερο αλγόριθμο μη γραμμικής ταξινόμησης, ο οποίος συγκαταλέγεται μεταξύ των πλέον αποδοτικών σήμερα, το ***«πολυστρωματικό perceptron»*** **(*Multi Layer Perceptron - MLP*)**, το οποίο δεν είναι παρά ή διασύνδεση πολλών απλών δικτύων τύπου perceptron σε μια ιεραρχική δομή. Με μια τέτοια διάταξη αίρεται ο περιορισμός διαμόρφωσης γραμμικών επιφανειών του perceptron και μπορούμε εύκολα να δημιουργήσουμε μη γραμμικές επιφάνειες απόφασης.

Στις προηγούμενες παραγράφους είδαμε ότι το perceptron αντιστοιχεί σε ένα υπερεπίπεδο του χώρου των δεδομένων εκπαίδευσης και πολλές φορές το μοντέλο αυτό περιγράφεται ως ένας τεχνητός «νευρώνας». Ως γνωστόν, οι εγκέφαλοι των ανθρώπων και των ζώων έχουν τη δυνατότητα να φέρουν επιτυχώς εις πέρας πολύπλοκα προβλήματα ταξινόμησης, όπως για παράδειγμα την αναγνώριση εικόνων. Είναι δε προφανές ότι κανένας μεμονωμένος νευρώνας του εγκεφάλου δεν έχει από μόνος του τη δυνατότητα να εκτελέσει κάτι τόσο πολύπλοκο. Η επίλυση τέτοιων προβλημάτων γίνεται μέσω της εκτενούς διασύνδεσης μεταξύ πολλών νευρώνων, η οποία δίνει τη δυνατότητα να αναλυθεί ένα πρόβλημα σε υποπροβλήματα ικανά να αντιμετωπιστούν σε επίπεδο νευρώνα. Αυτή η διαπίστωση έδωσε την έμπνευση για την ανάπτυξη των τεχνητών νευρωνικών δικτύων.

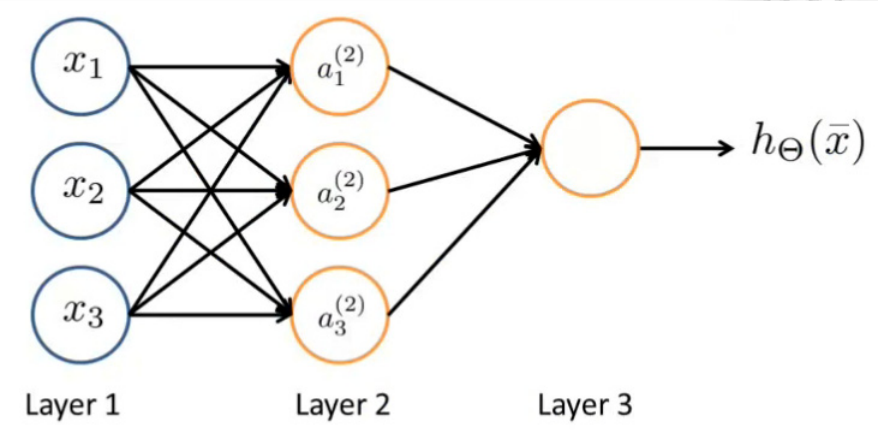
Στη συνέχεια θα παρουσιαστούν τα βασικότερα σημεία που είναι απαραίτητα για την κατασκευή και εκπαίδευση ενός μοντέλου ταξινόμησης MLP, ώστε να καταστεί εφικτή η προγραμματιστική υλοποίησή του για την αντιμετώπιση μιας μελέτης περίπτωσης.

Στο Σχήμα 4-7 φαίνεται ένα παράδειγμα ενός δικτύου MLP τριών στρωμάτων. Στο Στρώμα 1 βλέπουμε τις μονάδες εισόδου x_1, x_2, x_3. Στις περισσότερες περιπτώσεις δεν σχεδιάζεται (αλλά εννοείται) μία ακόμη μονάδα x_0, αυτή της πόλωσης (bias), η οποία επίσης συνδέεται με όλους τους κόμβους του επόμενου στρώματος και που στο εξής θα θεωρούμε ότι λαμβάνει πάντα την τιμή $+1$.

Το Στρώμα 1 αναφέρεται ως ***στρώμα εισόδου***, καθώς από εκεί εισέρχονται στο δίκτυο τα διαθέσιμα χαρακτηριστικά των δεδομένων εκπαίδευσης. Πολλές φορές στη βιβλιογραφία το

στρώμα εισόδου δε λογίζεται καν ως *στρώμα* καθώς οι κόμβοι της εισόδου δε λειτουργούν ως νευρώνες αλλά απλώς μεταδίδουν τις τιμές x_1, x_2, x_3 στο επόμενο στρώμα.

Σχήμα 4-7
Δίκτυο MLP τριών στρωμάτων



Στο Στρώμα 2 διακρίνουμε τρεις μονάδες χαρακτηρισμένες ως $\alpha_1^{(2)}, \alpha_2^{(2)}, \alpha_3^{(2)}$, ενώ και πάλι εξυπακούεται η ύπαρξη μιας επιπλέον μονάδας πόλωσης $\alpha_0^{(2)}$, πάντα με τιμή $+1$. Τέλος στο Στρώμα 3 βλέπουμε μια μονάδα η οποία εξάγει το τελικό αποτέλεσμα, ανάλογα με την εκάστοτε συνάρτηση ενεργοποίησης.

Το Στρώμα 3 αναφέρεται ως ***στρώμα εξόδου*** καθώς εκεί λαμβάνεται το τελικό αποτέλεσμα της όλης διεργασίας, ενώ το ενδιάμεσο (Στρώμα 2) καλείται ***κρυφό στρώμα***. Σε ένα δίκτυο MLP είναι δυνατόν να περιλαμβάνονται περισσότερα του ενός κρυφά στρώματα, αν και σπάνια θα δει κανείς περισσότερα των δύο. Ο όρος «κρυφό» έχει να κάνει με το γεγονός ότι στην περίπτωση της επιτηρούμενης μάθησης είναι γνωστές τόσο οι τιμές των χαρακτηριστικών του στρώματος εισόδου, όσο και οι «σωστές» τιμές του στρώματος εξόδου. Αντίθετα στα κρυφά στρώματα εμφανίζονται τιμές που ουδέποτε είναι διαθέσιμες σε ένα σύνολο εκπαίδευσης.

Ως προς τον αριθμό των νευρώνων που επιλέγονται για κάθε στρώμα, η απάντηση είναι αυτονόητη όσον αφορά τα στρώματα εισόδου και εξόδου. Στο στρώμα εισόδου υπάρχουν τόσες μονάδες όσες και ο αριθμός των χαρακτηριστικών των δειγμάτων εκπαίδευσης, ενώ στο στρώμα εξόδου θα έχουμε είτε μια μονάδα, αν πρόκειται για πρόβλημα δυαδικής ταξινόμησης, είτε – όπως θα δούμε και παρακάτω – τόσες μονάδες όσες και οι διαθέσιμες κλάσεις, αν πρόκειται για πρόβλημα πολλαπλής ταξινόμησης.

Εκεί που τίθεται θέμα επιλογής αριθμού μονάδων, είναι στα κρυφά στρώματα του δικτύου. Κατ' αρχάς μια λογική προεπιλογή είναι να χρησιμοποιηθεί ένα μόνο κρυφό στρώμα. Αυτός είναι και ο πιο κοινός τύπος δικτύων που χρησιμοποιούνται στην πράξη. Αν παρόλα αυτά κάποιος επιλέξει να χρησιμοποιήσει περισσότερα του ενός κρυφά στρώματα, συνιστάται να έχουν όλα τον ίδιο αριθμό νευρώνων.

Σχετικά με τον αριθμό των νευρώνων οι οποίοι θα αποτελούν ένα κρυφό στρώμα, συνήθως όσο περισσότεροι επιλεγούν τόσο το καλύτερο. Στις περισσότερες περιπτώσεις ο αριθμός τους είναι της ίδιας τάξης μεγέθους με το πλήθος των χαρακτηριστικών, αν και στην περίπτωση που ο αριθμός των χαρακτηριστικών είναι μικρός, ο αριθμός των μονάδων στο κρυφό στρώμα μπορεί να επιλεγεί ως το τριπλάσιο ή και το τετραπλάσιο του αριθμού των χαρακτηριστικών.

Στη συνέχεια θα χρησιμοποιηθεί η εξής ορολογία:

$a_i^{(j)}$: η ενεργοποίηση (ή ***έξοδος***) της μονάδας i του στρώματος j.

$\Theta^{(j)}$: ο πίνακας με τα ***συναπτικά βάρη*** (**synaptic weights**), ή απλά τις παραμέτρους, που συνδέουν τους νευρώνες του στρώματος j με αυτούς του στρώματος $j+1$.

Για να γίνουν καλύτερα κατανοητά τα παραπάνω ας δούμε αναλυτικά τους υπολογισμούς που υποδηλώνει η διάταξη του δικτύου στο Σχήμα 4-7.

$$\begin{aligned}
\alpha_1^{(2)} &= g\left(\Theta_{10}^{(1)}x_0 + \Theta_{11}^{(1)}x_1 + \Theta_{12}^{(1)}x_2 + \Theta_{13}^{(1)}x_3\right) = g\left(z_1^{(2)}\right) \\
\alpha_2^{(2)} &= g\left(\Theta_{20}^{(1)}x_0 + \Theta_{21}^{(1)}x_1 + \Theta_{22}^{(1)}x_2 + \Theta_{23}^{(1)}x_3\right) = g\left(z_2^{(2)}\right) \\
\alpha_3^{(2)} &= g\left(\Theta_{30}^{(1)}x_0 + \Theta_{31}^{(1)}x_1 + \Theta_{32}^{(1)}x_2 + \Theta_{33}^{(1)}x_3\right) = g\left(z_3^{(2)}\right) \\
h_\Theta\left(\overline{x}\right) &= \alpha_1^{(3)} = g\left(\Theta_{10}^{(2)}\alpha_0^{(2)} + \Theta_{11}^{(2)}\alpha_1^{(2)} + \Theta_{12}^{(2)}\alpha_2^{(2)} + \Theta_{13}^{(2)}\alpha_3^{(2)}\right) = g\left(z_1^{(3)}\right).
\end{aligned}$$

Στις παραπάνω σχέσεις το $z_i^{(j)}$ συμβολίζει τη ***«δικτυακή διέγερση»*** **(*net-input*)** του νευρώνα i στο στρώμα j. Από τα παραπάνω διαπιστώνουμε ότι π.χ. ο πίνακας $\Theta^{(1)} \in \mathbb{R}^{3\times 4}$. Γενικότερα, αν ένα δίκτυο έχει s_j μονάδες στο στρώμα j και s_{j+1} μονάδες στο στρώμα $j+1$, τότε ο πίνακας $\Theta^{(j)}$ θα είναι διαστάσεων $s_{j+1}\times(s_j+1)$. Εν κατακλείδι το δίκτυο αναπαριστά μια συνάρτηση h από το χώρο των εισόδων x σε ένα χώρο προβλέψεων y με παραμέτρους Θ.

Ως συνάρτηση ενεργοποίησης g αντί της βηματικής θα χρησιμοποιηθεί η ***λογιστική***, η οποία σχεδόν μονοπωλείται στα δίκτυα MLP, καθώς έχει το πλεονέκτημα ότι, όντας

παραγωγίσιμη, μπορεί να χρησιμοποιηθεί σε μεθόδους βελτιστοποίησης που εκμεταλλεύονται τις παραγώγους, όπως αυτές που έχουμε ήδη συναντήσει.

Τα δίκτυα perceptron πολλών στρωμάτων που ενσωματώνουν τη λογιστική συνάρτηση, έχουν σημαντικές δυνατότητες αναπαράστασης συναρτήσεων και συγκεκριμένα αποδεικνύεται ότι μπορούν να προσεγγίσουν οποιαδήποτε ομαλή συνάρτηση, όσο καλά επιθυμούμε [9] και μάλιστα με τη χρήση μόλις ενός κρυφού στρώματος [4]. Για το λόγο αυτό καλούνται και ***«Καθολικοί Προσεγγιστές»*** **(*Universal Approximators*)**.

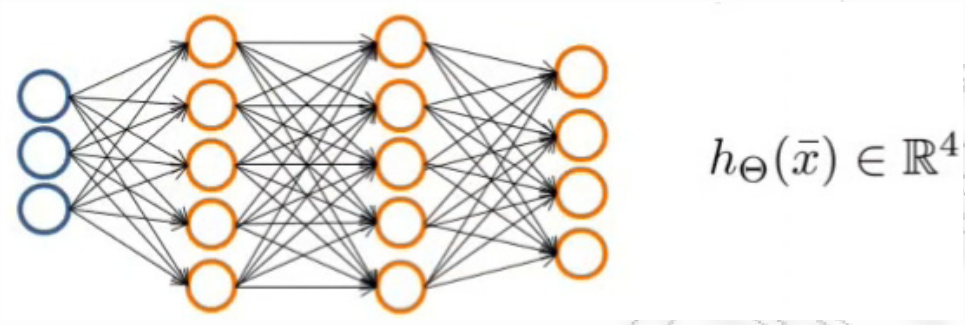
Οι προαναφερθέντες υπολογισμοί που αναφέρονται στο Σχήμα 4-7, μπορούν να αποδοθούν πιο συνεκτικά ως ακολούθως:

$$\overline{z}^{(2)} = \Theta^{(1)}\overline{x} = \Theta^{(1)}\overline{\alpha}^{(1)}$$
$$\overline{\alpha}^{(2)} = g\left(\overline{z}^{(2)}\right)$$
$$\overline{z}^{(3)} = \Theta^{(2)}\overline{\alpha}^{(2)}$$
$$h_{\Theta}\left(\overline{x}\right) = \overline{\alpha}^{(3)} = g\left(\overline{z}^{(3)}\right)$$

Η παραπάνω διαδικασία υπολογισμού της $h_{\Theta}(\overline{x})$ λέγεται ***«ανάκληση»*** **(*forward propagation*)** και η απόδοσή της στην παραπάνω συνεκτική μορφή μας εξυπηρετεί ιδιαίτερα, προκειμένου να αναπτυχθεί βάσει αυτής, κώδικας σε διανυσματική μορφή.

Η διάταξη με ένα μόνο νευρώνα στο στρώμα εξόδου είναι προφανώς κατάλληλη για την αντιμετώπιση προβλημάτων ταξινόμησης όπου ο αριθμός των κλάσεων είναι δύο. Ο τρόπος αντιμετώπισης προβλημάτων πολλαπλής κατηγοριοποίησης μέσω δικτύου MLP δεν είναι παρά μια προέκταση της μεθόδου «one-vs-all», που είδαμε στο Κεφ. 3. Συγκεκριμένα, επιλέγεται μια διάταξη δικτύου όπου το στρώμα εξόδου αποτελείται από τόσες μονάδες όσες και οι διαθέσιμες κλάσεις. Για παράδειγμα, μια δυνατή διάταξη δικτύου για την περίπτωση που το σύνολο των κατηγοριών στις οποίες ανήκουν τα αντικείμενα έχει διάσταση τέσσερα, θα μπορούσε να είναι αυτή που φαίνεται στο Σχήμα 4-8, όπου κάθε μία από τις μονάδες εξόδου αποτελεί και έναν ταξινομητή λογιστικής παλινδρόμησης. Ένα δίκτυο σαν το παραπάνω θα έδινε στην έξοδό του ως αποτέλεσμα, ένα διάνυσμα τεσσάρων αριθμών, τρεις εκ των οποίων θα είχαν την τιμή 0 και ένας την τιμή 1. Κάθε μονάδα εξόδου δηλαδή θα *απαντούσε* στο ερώτημα εάν τα συγκεκριμένα χαρακτηριστικά με τα οποία τροφοδοτήθηκαν οι μονάδες εισόδου, αφορούν αντικείμενο της αντίστοιχης κλάσης (1), ή όχι (0).

Σχήμα 4-8
Δίκτυο MLP για πολλαπλή κατηγοριοποίηση



Στην πράξη απαιτείται προσοχή στο σημείο αυτό, καθώς το διάνυσμα $\bar{y}$ των δεδομένων εκπαίδευσης, το οποίο περιλαμβάνει τον χαρακτηρισμό της κλάσης στην οποία ανήκει το κάθε αντικείμενο του συνόλου εκπαίδευσης - και έστω στο παράδειγμά μας $\bar{y} \in \{1,2,3,4\}$, θα πρέπει να μετατραπεί σε πίνακα, έτσι ώστε κάθε στοιχείο $y^{(i)}$ του διανύσματος να αντικατασταθεί με ένα από τα παρακάτω διανύσματα:

$$\begin{pmatrix}1\\0\\0\\0\end{pmatrix}, \begin{pmatrix}0\\1\\0\\0\end{pmatrix}, \begin{pmatrix}0\\0\\1\\0\end{pmatrix}, \begin{pmatrix}0\\0\\0\\1\end{pmatrix}$$

ανάλογα με την κλάση κάθε αντικειμένου.

4.5 Εκπαίδευση του Δικτύου MLP

Η εκπαίδευση ενός δικτύου πολλών στρωμάτων είναι η διαδικασία ρύθμισης των συναπτικών βαρών του, έτσι ώστε να ικανοποιείται κάποιο κριτήριο καταλληλότητας. Αυτό που κάνει την εκπαίδευση ενός δικτύου MLP πολύ πιο ενδιαφέρουσα είναι η ιδιότητα του καθολικού προσεγγιστή, που αναφέραμε παραπάνω. Αυτή λέει, με απλά λόγια, πως αν έχουμε το κατάλληλο σε μέγεθος δίκτυο μπορούμε να το εκπαιδεύσουμε να μάθει οποιαδήποτε συνάρτηση εμείς επιθυμούμε με οποιαδήποτε ποιότητα προσέγγισης επιθυμούμε. Θυμίζουμε ότι το απλό δίκτυο perceptron μπορεί να υλοποιήσει μόνο γραμμικές διαχωριστικές επιφάνειες. Αυτό αιτιολογεί και τη δημοτικότητα των αλγορίθμων εκπαίδευσης του MLP, με κυριότερο εκπρόσωπο τον αλγόριθμο ***«οπισθοδιάδοσης»*** **(*backpropagation*)** που θα περιγραφεί συνοπτικά στη συνέχεια.

Ο αλγόριθμος backpropagation προτάθηκε από τον Paul Werbos[1] στη δεκαετία του 1970 στα πλαίσια της ανάλυσης μοντέλων οικονομικής και πολιτικής πρόβλεψης, τα οποία όμως ουδόλως θύμιζαν νευρωνικά δίκτυα. Αργότερα, στη δεκαετία του 1980, έγινε αντιληπτό ότι η μέθοδος μπορούσε να μεταφερθεί αυτούσια στην εκπαίδευση νευρωνικών δικτύων πολλών στρωμάτων και έκτοτε έγινε η πιο γνωστή και πιο διαδεδομένη μέθοδος για το σκοπό αυτό.

Ας θεωρήσουμε γενικά ένα σύνολο εκπαίδευσης $\{(\bar{x}^{(1)}, y^{(1)}), (\bar{x}^{(2)}, y^{(2)}), \ldots, (\bar{x}^{(m)}, y^{(m)})\}$ και ένα δίκτυο MLP με L στρώματα, καθένα από τα οποία έχει s_l μονάδες (μη περιλαμβανομένης της μονάδας πόλωσης), με $l = 1, 2, \ldots, L$. Για ευκολία, ο αριθμός των μονάδων στο στρώμα εξόδου θα συμβολίζεται με K. Έτσι, στην περίπτωση της δυαδικής ταξινόμησης, θα ισχύει προφανώς $h_\Theta(\bar{x}) \in \mathbb{R}$ και $s_L = 1$ (αφού $K = 1$), ενώ στην περίπτωση πολλαπλής ταξινόμησης $(K \geq 3)$ θα έχουμε $s_L = K$ και $h_\Theta(\bar{x}) \in \mathbb{R}^K$.

Η συνάρτηση κόστους που θα πρέπει να ελαχιστοποιηθεί στην περίπτωση αυτή, αποτελεί γενίκευση της αντίστοιχης στην περίπτωση της λογιστικής παλινδρόμησης και θα έχει την παρακάτω μορφή

$$J(\Theta) = \frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\left[-y_k^{(i)}\log\left(\left(h_\Theta(x^{(i)})\right)_k\right) - \left(1-y_k^{(i)}\right)\log\left(\left(h_\Theta(x^{(i)})\right)_k\right)\right] +$$

$$+\frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(\Theta_{ji}^{(l)}\right)^2, \qquad (4.11)$$

όπου m το πλήθος σημείων του συνόλου εκπαίδευσης, $(h_\Theta(\bar{x}))_i$ το i-οστό στοιχείο του διανύσματος εξόδου, ενώ στη 2^η γραμμή αναγνωρίζουμε τον όρο ομαλοποίησης. Από τις παραμέτρους που αθροίζονται σε αυτό τον όρο, κατ' αναλογία με ό,τι κάναμε στη λογιστική παλινδρόμηση, συνηθίζεται να εξαιρούνται όσες αντιστοιχούν στις μονάδες πόλωσης[2]. Συγκεκριμένα, πρόκειται για αυτές που έχουν δείκτη $i = 0$.

Για την ελαχιστοποίηση της παραπάνω συνάρτησης κόστους θα εφαρμόσουμε τον αλγόριθμο backpropagation. Κατ' αναλογία με την περίπτωση της λογιστικής παλινδρόμησης, θα πρέπει να υλοποιηθεί προγραμματιστικά μια συνάρτηση που θα επιστρέφει, αφενός την τιμή που λαμβάνει η παραπάνω συνάρτηση κόστους (ως προς το

[1] P. Werbos. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. PhD thesis, Harvard, Cambridge, MA, August 1974.

[2] Η πρακτική της εξαίρεσης των παραμέτρων που αντιστοιχούν στους όρους πόλωσης ακολουθείται κατά σύμβαση. Παρόλα αυτά, στην περίπτωση που οι όροι αυτοί δεν εξαιρεθούν από την άθροιση, η διαφοροποίηση που θα παρουσιαστεί στο αποτέλεσμα δε θα είναι σημαντική.

διάνυσμα των παραμέτρων Θ) και αφετέρου το διάνυσμα κλίσης (gradient vector) που θα αποτελείται από τους όρους $\frac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta)$.

Για τον υπολογισμό της τιμής που λαμβάνει η συνάρτηση κόστους αρκεί η εφαρμογή του τύπου (4.11). Αυτό που παρουσιάζει μεγαλύτερη δυσκολία είναι ο υπολογισμός των όρων των μερικών παραγώγων. Προκειμένου να απλοποιηθούν οι συμβολισμοί θα θεωρήσουμε στη συνέχεια ότι το δείγμα των δεδομένων εκπαίδευσης περιλαμβάνει ένα και μοναδικό ζεύγος $(\overline{x}, y)$ και θα περιγράψουμε τη διαδικασία μόνο γι' αυτό. Επίσης θα θεωρήσουμε ότι το υπό εκπαίδευση δίκτυο έχει τη μορφή αυτού που εμφανίζεται στο Σχήμα 4-8.

Αρχικά θα εκτελέσουμε τη διαδικασία ανάκλησης κατά τα γνωστά, για κάποιες αρχικές τιμές των παραμέτρων Θ:

$$\begin{aligned}
&\overline{\alpha}^{(1)} = \overline{x} \\
&\overline{z}^{(2)} = \Theta^{(1)} \overline{\alpha}^{(1)} \\
&\overline{\alpha}^{(2)} = g\left(\overline{z}^{(2)}\right) \quad (add\ \alpha_0^{(2)}) \\
&\overline{z}^{(3)} = \Theta^{(2)} \overline{\alpha}^{(2)} \\
&\overline{\alpha}^{(3)} = g\left(\overline{z}^{(3)}\right) \quad (add\ \alpha_0^{(3)}) \\
&\overline{\alpha}^{(4)} = h_\Theta\left(\overline{x}\right) = g\left(\overline{z}^{(4)}\right).
\end{aligned}$$

Στη συνέχεια θα υπολογιστούν οι μερικές παράγωγοι μέσω του αλγορίθμου backpropagation. Η ιδέα στην οποία βασίζεται αυτός ο αλγόριθμος είναι ότι για κάθε νευρώνα ορίζεται ένας όρος $\delta_j^{(l)}$, ο οποίος κατά κάποιον τρόπο αντιπροσωπεύει το «σφάλμα» της ενεργοποίησης $\alpha_j^{(l)}$ του νευρώνα j στο στρώμα l και σχετίζεται με τις μερικές παραγώγους του κόστους ως προς τα συναπτικά βάρη του νευρώνα j. Έτσι στο παράδειγμά μας, ξεκινώντας από το στρώμα εξόδου θα έχουμε:

$$\delta_j^{(4)} = \alpha_j^{(4)} - y_j = \left(h_\Theta\left(x\right)\right)_j - y_j.$$

Δεδομένου ότι τα δ, α, y είναι διανύσματα, η προηγούμενη σχέση μπορεί να γραφεί στη μορφή

$$\overline{\delta}^{(4)} = \overline{\alpha}^{(4)} - \overline{y}\ .$$

Έχοντας υπολογίσει τους όρους δ για το τελευταίο στρώμα του δικτύου, μπορούμε να πάμε προς τα πίσω και να υπολογίσουμε τους όρους δ για τα προηγούμενα στρώματα, που αποδεικνύεται ([9],[20]) ότι προκύπτουν ως εξής:

$$\bar{\delta}^{(3)} = \left(\Theta^{(3)}\right)^T \bar{\delta}^{(4)} .* g'\left(\bar{z}^{(3)}\right)$$
$$\bar{\delta}^{(2)} = \left(\Theta^{(2)}\right)^T \bar{\delta}^{(3)} .* g'\left(\bar{z}^{(2)}\right).$$

Στους πιο πάνω τύπους το σύμβολο .* αντιστοιχεί στο πολλαπλασιασμό στοιχείου προς στοιχείο μεταξύ των δύο διανυσμάτων και ο συμβολισμός αυτός ταυτίζεται με αυτόν που χρησιμοποιείται στη γλώσσα Octave για τον ίδιο σκοπό.

Παρατηρήστε ότι στη διαδικασία που περιγράφηκε δεν επιχειρήθηκε υπολογισμός του όρου $\bar{\delta}^{(1)}$, καθώς το πρώτο στρώμα αντιστοιχεί στο στρώμα εισόδου όπου εκεί έχουμε τις παρατηρηθείσες τιμές για τα χαρακτηριστικά των δεδομένων εκπαίδευσης και προφανώς δεν υπάρχει κάποιο «σφάλμα» σχετικά με αυτές τις τιμές, ούτε και κανένας λόγος να επιχειρήσουμε να τις μεταβάλουμε.

Ο όρος $g'(\bar{z}^{(l)})$ αντιστοιχεί στην παράγωγο της συνάρτησης ενεργοποίησης g υπολογισμένη για τις τιμές της δικτυακής διέγερσης $\bar{z}^{(l)}$. Δεδομένου ότι για την περίπτωση της λογιστικής συνάρτησης η παράγωγος μπορεί να γραφεί στη μορφή

$$f'(u) = f(u)\left(1 - f(u)\right)$$

οι παραπάνω τύποι υπολογισμού των όρων δ μπορούν να γραφούν:

$$\bar{\delta}^{(3)} = \left(\Theta^{(3)}\right)^T \bar{\delta}^{(4)} .* \bar{\alpha}^{(3)} .* \left(1 - \bar{\alpha}^{(3)}\right)$$
$$\bar{\delta}^{(2)} = \left(\Theta^{(2)}\right)^T \bar{\delta}^{(3)} .* \bar{\alpha}^{(2)} .* \left(1 - \bar{\alpha}^{(2)}\right).$$

Ο όρος backpropagation έχει λοιπόν να κάνει με αυτή την *οπισθοδιάδοση* των σφαλμάτων γι' αυτό και πολλές φορές αναφέρεται ως *backpropagation of errors algorithm*. Μπορεί να αποδειχθεί ότι, αγνοώντας τον όρο ομαλοποίησης (ή αντίστοιχα θέτοντας $\lambda = 0$), ισχύει η παρακάτω σχέση:

$$\frac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta) = \alpha_j^{(l)} \delta_i^{(l+1)}.$$

Ας ανακεφαλαιώσουμε επαναλαμβάνοντας τα βήματα του αλγόριθμου backpropagation στη γενικότερη περίπτωση (και όχι για ένα μόνο ζευγάρι δεδομένων εκπαίδευσης). Ο αλγόριθμος αυτός, για λόγους καλύτερης εποπτείας, έχει διαμορφωθεί ώστε να γίνεται χρήση ενός επαναληπτικού βρόγχου (***for*** loop). Κάλλιστα όμως μπορεί να διανυσματοποιηθεί πλήρως και να αποφευχθεί εντελώς ο επαναληπτικός βρόγχος, με αποτέλεσμα να εκτελείται κατά πολύ ταχύτερα.

Αλγόριθμος 4.3

let $D = \{(\bar{x}^{(1)}, y^{(1)}), (\bar{x}^{(2)}, y^{(2)}), \ldots, (\bar{x}^{(m)}, y^{(m)})\}$
Αρχικοποίησε τις παραμέτρους Θ
$\Delta_{ij}^{(l)} \leftarrow 0$ (για όλα τα l, i, j)
for $t = 1$ **to** m
$\bar{\alpha}^{(1)} \leftarrow \bar{x}^{(t)}$
Υπολόγισε με τη διαδικασία ανάκλησης τα $\bar{\alpha}^{(l)}$ για $l = 2, 3, \ldots, L$
$\bar{\delta}^{(L)} \leftarrow \bar{\alpha}^{(L)} - \bar{y}^{(t)}$
Υπολόγισε τα $\bar{\delta}^{(L-1)}, \bar{\delta}^{(L-2)}, \ldots, \bar{\delta}^{(2)}$
$\Delta_{ij}^{(l)} \leftarrow \Delta_{ij}^{(l)} + \alpha_j^{(l)} \delta_i^{(l+1)}$
end for
if $j \neq 0$ **then** $D_{ij}^{(l)} \leftarrow \frac{1}{m}\Delta_{ij}^{(l)} + \frac{\lambda}{m}\Theta_{ij}^{(l)}$ **else** $D_{ij}^{(l)} \leftarrow \frac{1}{m}\Delta_{ij}^{(l)}$ (bias terms) **end if**
return $D_{ij}^{(l)}$

Στο παράδειγμα που έχει υλοποιηθεί στα πλαίσια της παρούσας εργασίας, ο αλγόριθμος του backpropagation δίνεται σε πλήρως διανυσματοποιημένη μορφή. Προσομοιώνει όμως δίκτυα αποκλειστικά ενός κρυφού στρώματος.

4.6 Εφαρμογή MLP: Ανάγνωση Χειρόγραφων Ψηφίων

Το μοντέλο του δικτύου MLP εφαρμόστηκε στα δεδομένα του παραδείγματος της Ενότητας 3.6, που είχε σαν σκοπό την ταξινόμηση ψηφιοποιημένων εικόνων χειρόγραφων ψηφίων (από το 0 έως το 9) σε 10 κλάσεις. Αναλυτική περιγραφή των δεδομένων έχει ήδη δοθεί στην Ενότητα 3.6 και ο αναγνώστης μπορεί να ανατρέξει εκεί για περισσότερες λεπτομέρειες.

Το δίκτυο που υλοποιήθηκε αποτελείται από 400 μονάδες στο στρώμα εισόδου, 10 μονάδες στο στρώμα εξόδου, ενώ για το μοναδικό κρυφό στρώμα επιλέχθηκαν 25 μονάδες. Πρόσβαση στο αρχείο δεδομένων αλλά και στο πλήρες σετ των αρχείων με τον κώδικα της εφαρμογής παρέχεται μέσω του συνδέσμου: http://bit.ly/v2gItB.

Αφού προσδιοριστούν οι παράμετροι του μοντέλου, είμαστε σε θέση να χρησιμοποιήσουμε το MLP για την ανάγνωση (πρόβλεψη) του ψηφίου που απεικονίζει κάθε τέτοια νέα εικόνα, με βάση τις τιμές της έντασης του γκρι χρώματος των pixel που την αποτελούν.

Ως ένδειξη της επίδοσης του μοντέλου ζητήθηκε η πρόβλεψη ταξινόμησης του συνόλου των δεδομένων εκπαίδευσης. Το ποσοστό ακρίβειας των προβλέψεων ήταν **99.54%**, ενώ με κατάλληλες ρυθμίσεις είναι δυνατόν να επιτύχουμε τέλεια προσαρμογή στο σύνολο των δεδομένων εκπαίδευσης. Τα κυριότερα σημεία της υλοποίησης της συγκεκριμένης εφαρμογής δίνονται, σε γλώσσα Octave, στο Παράρτημα της σελ. 168.

Στη συνέχεια θα αναφερθούμε σε κάποιες τεχνικές λεπτομέρειες σχετικές με την υλοποίηση του συγκεκριμένου μοντέλου στη γλώσσα Octave.

4.6.1 Μετατροπή πινάκων σε διάνυσμα και αντίστροφα

Όπως έχουμε ήδη αναφέρει, ο τρόπος χρήσης των προηγμένων αλγορίθμων βελτιστοποίησης που χρησιμοποιούνται από συναρτήσεις όπως η ***fminunc***, απαιτεί τη δημιουργία μιας συνάρτησης η οποία επιστρέφει το κόστος και το διάνυσμα κλίσης και έχει τη γενική μορφή:

```
function  [jVal, gradient] = costFunction(theta)
  …
```

Κατόπιν, οι βέλτιστες τιμές για τις παραμέτρους λαμβάνονται με μια κλήση της μορφής:

```
optTheta = fminunc(@costFunction, initialTheta, options)
```

Το πρόβλημα εδώ έγκειται στο γεγονός ότι οι αλγόριθμοι βελτιστοποίησης *απαιτούν* οι παράμετροι **theta**, **initialTheta** και **gradient** να έχουν τη μορφή διανύσματος. Στην περίπτωση του μοντέλου MLP όπως είδαμε όμως, οι παράμετροι $\Theta^{(l)}$ καθώς και οι όροι με τις μερικές παραγώγους $D^{(l)}$ είναι στην πραγματικότητα πίνακες.

Πρέπει λοιπόν να μεριμνήσουμε να μετατρέψουμε τους πίνακες αυτούς σε διανύσματα, όταν γίνεται χρήση αυτών των συναρτήσεων και στη συνέχεια να τους επαναφέρουμε στη μορφή του αρχικού πίνακα, από το διάνυσμα που προσωρινά δημιουργήσαμε.

Έστω $\Theta^{(1)} \in \mathbb{R}^{10\times 11}, \Theta^{(2)} \in \mathbb{R}^{10\times 11}, \Theta^{(3)} \in \mathbb{R}^{1\times 11}$ και $D^{(1)} \in \mathbb{R}^{10\times 11}, D^{(2)} \in \mathbb{R}^{10\times 11}, D^{(3)} \in \mathbb{R}^{1\times 11}$. Στην Octave η μετατροπή των πινάκων Θ και D σε δύο διανύσματα, γίνεται ως εξής:

```
thetaVec = [ Theta1(:) ; Theta2(:) ; Theta3(:) ];
Dvec = [ D1(:) ; D2(:) ; D3(:) ];
```

Αντίστοιχα η επαναφορά τους στην αρχική μορφή γίνεται ως εξής:

```
Theta1 = reshape(thetaVec(1:110), 10, 11);
Theta2 = reshape(thetaVec(111:220), 10, 11);
Theta3 = reshape(thetaVec(221:231), 1, 11);
```

Ανάλογα δουλεύουμε και για τους πίνακες D.

4.6.2 Επαλήθευση με τη μέθοδο gradient checking

Ένα δεύτερο θέμα που είναι σημαντικό έχει να κάνει με την, όχι αμελητέα πιθανότητα, να παρεισφρύσουν σφάλματα στον κώδικα που υλοποιεί τον αλγόριθμο backpropagation. Ενδέχεται μάλιστα ο αλγόριθμος να δείχνει ότι πράγματι δουλεύει, ακόμη κι αν υπάρχει κάποιο μικρό σφάλμα στον κώδικα (π.χ. η συνάρτηση κόστους μπορεί να συνεχίζει να μειώνεται από βήμα σε βήμα), στην πραγματικότητα όμως το δίκτυο δε θα έχει ποτέ την επίδοση που θα είχε αν ο κώδικας ήταν αλάνθαστος.

Για την αντιμετώπιση αυτού του προβλήματος ενδείκνυται να χρησιμοποιείται η μέθοδος ***gradient checking***, η οποία αποτελεί ένα είδος επαλήθευσης, προκειμένου να είμαστε σίγουροι ότι η υλοποίησή μας δεν έχει λάθος και ότι πράγματι υπολογίζουμε σωστά τις παραγώγους της συνάρτησης κόστους.

Η όλη ιδέα είναι πολύ απλή και βασίζεται στον παράλληλο αριθμητικό υπολογισμό της τιμής της παραγώγου σε κάθε σημείο, σύμφωνα με τον προσεγγιστικό τύπο:

$$\frac{d}{d\vartheta}J(\vartheta)\approx\frac{J(\vartheta+\varepsilon)-J(\vartheta-\varepsilon)}{2\varepsilon}$$

για μια αρκετά μικρή τιμή του ε (π.χ. $\varepsilon=10^{-4}$).

Στη γενικότερη περίπτωση όπου το ϑ είναι διάνυσμα ($\overline{\vartheta}=[\vartheta_1,\vartheta_2,\vartheta_3,\ldots,\vartheta_n]$) οι τύποι υπολογισμού των μερικών παραγώγων λαμβάνουν την παρακάτω μορφή:

$$\frac{\partial}{\partial\vartheta_1}J(\overline{\vartheta})\approx\frac{J(\vartheta_1+\varepsilon,\vartheta_2,\vartheta_3,\ldots,\vartheta_n)-J(\vartheta_1-\varepsilon,\vartheta_2,\vartheta_3,\ldots,\vartheta_n)}{2\varepsilon}$$

$$\frac{\partial}{\partial\vartheta_2}J(\overline{\vartheta})\approx\frac{J(\vartheta_1,\vartheta_2+\varepsilon,\vartheta_3,\ldots,\vartheta_n)-J(\vartheta_1,\vartheta_2-\varepsilon,\vartheta_3,\ldots,\vartheta_n)}{2\varepsilon}$$

$$\vdots$$

$$\frac{\partial}{\partial\vartheta_n}J(\overline{\vartheta})\approx\frac{J(\vartheta_1,\vartheta_2,\vartheta_3,\ldots,\vartheta_n+\varepsilon)-J(\vartheta_1,\vartheta_2,\vartheta_3,\ldots,\vartheta_n-\varepsilon)}{2\varepsilon}.$$

Τα παραπάνω μπορούν να υλοποιηθούν στην Octave με κώδικα που θα έχει συνοπτικά την παρακάτω μορφή:

```
for i = 1:n,
    thetaPlus = theta;
    thetaPlus(i) =  thetaPlus(i) + EPSILON;
    thetaMinus = theta;
    thetaMinus(i) = thetaMinus(i) - EPSILON;
    gradApprox(i) = ( J(thetaPlus) – J(thetaMinus) ) / (2*EPSILON)
end;
```

όπου n η διάσταση του διανύσματος των παραμέτρων **theta**.

Ο έλεγχος στη συνέχεια έγκειται στο κατά πόσον ταυτίζονται (με κάποια δεδομένη προσέγγιση φυσικά) τα στοιχεία των διανυσμάτων **gradApprox** και **Dvec** (το οποίο περιλαμβάνει τις μερικές παραγώγους που υπολογίστηκαν μέσω του backpropagation). Για $\varepsilon = 10^{-4}$ οι τιμές συνήθως θα συμφωνούν τουλάχιστον στα 4 πρώτα δεκαδικά ψηφία (και συχνά σε πολύ περισσότερα).

Ο κώδικας που εκτελεί τον αριθμητικό υπολογισμό της παραγώγου θα πρέπει να εφαρμοστεί σε *δοκιμαστικό* δίκτυο ελέγχου και σε καμία περίπτωση δεν πρέπει να εκτελείται κατά τη διαδικασία μάθησης των *πραγματικών* μας δικτύων διαφορετικά θα γίνει ***πολύ*** αργός. Αυτό συμβαίνει γιατί η αριθμητική προσέγγιση που περιγράφηκε αποτελεί έναν τρομερά υπολογιστικά δαπανηρό τρόπο εκτίμησης των μερικών παραγώγων, σε σχέση με τον αλγόριθμο backpropagation.

Για λόγους οικονομίας ο έλεγχος gradient checking δεν περιλαμβάνεται στον κώδικα του Παραρτήματος της σελ. 168, έχει όμως υλοποιηθεί και τα σχετικά αρχεία περιλαμβάνονται στο σύνδεσμο με τον κώδικα και τα δεδομένα της εφαρμογής.

4.7 Αρχικοποίηση Παραμέτρων

Κατά την εκτέλεση της διαδικασίας ελαχιστοποίησης της συνάρτησης κόστους, είτε μέσω του αλγορίθμου gradient descent, είτε με χρήση των προηγμένων αλγορίθμων, που έχουν προαναφερθεί, απαιτείται η επιλογή κάποιων αρχικών τιμών για τις παραμέτρους Θ. Μάλιστα, όταν κάναμε χρήση των συναρτήσεων αυτών στα μοντέλα της λογιστικής παλινδρόμησης αρχικοποιούσαμε τις τιμές των παραμέτρων με μηδενικά και δεν υπήρχε κανένα απολύτως πρόβλημα με αυτό.

Αντίθετα όμως, στην περίπτωση των νευρωνικών δικτύων δεν πρέπει να χρησιμοποιηθούν μηδενικές τιμές για την αρχικοποίηση των τιμών των παραμέτρων, καθώς οι τιμές ενεργοποίησης κάθε νευρώνα $\alpha^{(l)}$ ενός στρώματος θα είναι μεταξύ τους ίδιες, για όλα τα δεδομένα του δείγματος εκπαίδευσης.

Το ίδιο θα ισχύει και με τις τιμές $\delta^{(l)}$ αλλά και με τα συναπτικά βάρη κάθε νευρώνα που θα υπολογίζονται σε κάθε κύκλο βελτιστοποίησης. Γίνεται λοιπόν εύκολα αντιληπτό ότι ένα τέτοιο δίκτυο δε θα μπορέσει να προσομοιώσει συναρτήσεις πραγματικά ενδιαφέρουσες, αφού ανεξάρτητα με το πλήθος των νευρώνων ενός στρώματος, όλοι θα υπολογίζουν ακριβώς την ίδια συνάρτηση των μεταβλητών εισόδου.

Προκειμένου να αποφευχθεί το πρόβλημα αυτό των «συμμετρικών βαρών», θα πρέπει να φροντίσουμε να αρχικοποιούμε τις παραμέτρους ενός νευρωνικού δικτύου με τυχαίες τιμές (π.χ. $-\varepsilon \le \Theta_{ij}^{(l)} \le \varepsilon$,).

Κάτι που θα πρέπει επίσης να έχουμε υπόψη είναι ότι, επειδή η συνάρτηση κόστους του MLP *δεν* είναι κυρτή, υπάρχει πιθανότητα η διαδικασία ελαχιστοποίησης να καταλήξει σε **τοπικά ελάχιστα**, τα οποία προφανώς αποτελούν υποδεέστερες λύσεις του προβλήματος σε σχέση με το ολικό ελάχιστο.

Παρόλα αυτά η πράξη έχει δείξει ότι αν ο αλγόριθμος δεν σταματήσει από την αρχή σε ένα τοπικό ελάχιστο και αν εξελιχθεί κανονικά για ένα ικανό χρονικό διάστημα, το τοπικό ελάχιστο που θα προκύψει στο τέλος σπανίως δημιουργεί ουσιαστικό πρόβλημα καθώς τις περισσότερες φορές αντιστοιχεί σε πολύ ικανοποιητική τιμή κόστους. Δεν είναι ξεκάθαρο γιατί συμβαίνει αυτό, ούτε είναι απόλυτα σίγουρο ότι θα συμβεί έτσι. Είναι όμως, και για το λόγο αυτό, πολύ σημαντικό να δοθεί προσοχή στις αρχικές τιμές που θα δοθούν στις παραμέτρους, ώστε να αποφευχθεί το «κόλλημα» σε τοπικά ελάχιστα στην αρχή του αλγορίθμου.

Οι τιμές των παραμέτρων πρέπει να αρχικοποιούνται σε ***μικρές κατ' απόλυτη τιμή τυχαίες τιμές.*** Και αυτό γιατί οι μεγάλες τιμές οδηγούν σε μεγάλες απόλυτες τιμές των δικτυακών διεγέρσεων, οι οποίες οδηγούν με τη σειρά τους σε τιμές ενεργοποίησης νευρώνα πολύ κοντά στον κορεσμό, δηλαδή κοντά στο 0 ή στο 1. Τέτοιες τιμές ενεργοποίησης δίνουν πολύ μικρή παράγωγο και συνεπώς η τιμή του σφάλματος δ του νευρώνα είναι σχεδόν μηδενική. Καθώς η διόρθωση των βαρών είναι ανάλογη του δ, η διόρθωση που επιτυγχάνεται είναι σχεδόν αμελητέα και το δίκτυο μπορεί να κολλήσει από την αρχή σε ένα τοπικό ελάχιστο, ή να παραμείνει ουσιαστικά στο ίδιο σημείο για πολλές επαναλήψεις.

Ένας αποτελεσματικός τρόπος για την επιλογή της τιμής ε [25], ο οποίος εφαρμόστηκε και στο παράδειγμα που υλοποιήθηκε, είναι με βάση το πλήθος των νευρώνων στα γειτονικά στρώματα του κάθε πίνακα $\Theta^{(l)}$ και συγκεκριμένα:

$$\varepsilon = \frac{\sqrt{6}}{\sqrt{L_{in} + L_{out}}}, \text{ όπου } L_{in} = s_l \text{ και } L_{out} = s_{l+1}$$

4.8 Ρουτίνες MLP στα Πακέτα R και WEKA

4.8.1 MLP στο R Project

Εκπαίδευση νευρωνικών δικτύων – αποκλειστικά ενός και μόνον κρυφού στρώματος – επιτυγχάνεται μέσω του πακέτου ***nnet***[1].

Αρχικά εξάγουμε από το περιβάλλον της Octave τα δεδομένα του παραδείγματος για την ανάγνωση χειρόγραφων ψηφίων, σαν αρχείο τύπου csv (comma separated values):

```
% Εξαγωγή δεδομένων σε μορφή csv (από το περιβάλλον της Octave)
csvwrite ('imagedata.csv', [X y]);
```

Στο περιβάλλον R πια και αφού οριστεί ως τρέχων κατάλογος αυτός στον οποίο βρίσκεται το παραπάνω csv αρχείο[2]:

```
# Εισαγωγή δεδομένων
dataset <- read.csv('imagedata.csv',header=F)

# Προσθήκη κεφαλίδων
names(dataset) <- c(paste('pixel.',rep(1:400),sep=''),'digit')

# Διαχωρισμός των χαρακτηριστικών (X) από της κατηγορίες (y)
X <- dataset[,-401]
y <- dataset[,401]

# Φόρτωμα του πακέτου που θα χρησιμοποιηθεί
library(nnet)
```

Η διαδικασία μετατροπής του κάθε ψηφίου, από το 1 έως το 10, που χαρακτηρίζει την κλάση κάθε εγγραφής στο διάνυσμα $\bar{y}$ των δεδομένων εκπαίδευσης, σε ένα διάνυσμα διάστασης 10, πραγματοποιείται με την παρακάτω εντολή:

```
# Μετατροπή επιπέδων του παράγοντα y σε διανύσματα μόνο με 0 και 1
targets <- class.ind(y)
```

Ακολουθεί η εντολή εκπαίδευσης, όπου δίνονται ως παράμετροι: το σύνολο εκπαίδευσης, οι κλάσεις που αντιστοιχούν στις εγγραφές του συνόλου, ο αριθμός των μονάδων του κρυφού στρώματος (***size***), η τιμή για την αρχικοποίηση των παραμέτρων του δικτύου[3] (***rang***), ο συντελεστής μείωσης του ρυθμού εκμάθησης της συνάρτησης βελτιστοποίησης[4] (***decay***), ο

---

[1] Feed-forward Neural Networks and Multinomial Log-Linear Models (βλ. http://bit.ly/vZfgtY)

[2] Αν δε χρησιμοποιείται ως περιβάλλον εργασίας το RStudio (http://rstudio.org/), όπου η επιλογή καταλόγου μπορεί να γίνει από το μενού, ένας εύκολος τρόπος για τον ορισμό του τρέχοντος καταλόγου είναι μέσω της εντολής **`setwd(dirname(file.choose()))`**. Στο πλαίσιο διαλόγου που θα ανοίξει μεταφερθείτε στον επιθυμητό κατάλογο και επιλέξτε οποιοδήποτε αρχείο σε αυτόν.

[3] Πρόκειται για την τιμή του ε, σύμφωνα με τα αναφερόμενα στη Ενότητα 4.7

[4] Η βέλτιστη τιμή του θα πρέπει να προκύψει με πειραματισμό. Βλ. Ενότητα 6.11 για ένα τρόπο ρύθμισης της παραμέτρου μέσου του πακέτου ***caret***.

μέγιστος αριθμός βημάτων έως τον τερματισμό (***maxit***) και ο μέγιστος αριθμός των επιτρεπτών βαρών[1], στην περίπτωση που αυτός είναι μεγαλύτερος της προκαθορισμένης τιμής 1000 (***MaxNWts***).

```
# Εκπαίδευση νευρωνικού δικτύου
digits_nn <- nnet(X, targets, size = 25, rang = 0.12, decay = 0.1, maxit = 400,
MaxNWts = 20000)
```

Η συνάρτηση ***nnet*** παρέχει τη δυνατότητα περαιτέρω παραμετροποίησης. Για περισσότερες λεπτομέρειες ο αναγνώστης παραπέμπεται στην τεκμηρίωση της συνάρτησης. Το ποσοστό ακρίβειας των προβλέψεων επί του συνόλου εκπαίδευσης ήταν **99.68%**

```
# Υπολογισμός ποσοστού επιτυχών προβλέψεων επί του συνόλου εκπαίδευσης
res <- predict(digits_nn, X)
pred.y <- apply(res,1,which.is.max)
mean(pred.y==y) * 100
[1] 99.68
```

Για την κατασκευή «***μήτρας σύγχυσης***» (***confusion matrix***) υπάρχουν προφανώς πολλές διαθέσιμες εναλλακτικές επιλογές στα διάφορα πακέτα της R. Μία από αυτές είναι μέσω του πακέτου ***caret*** και της συνάρτησης ***confusionMatrix***[2].

```
library(caret)
confusionMatrix(pred.y,y)
```

4.8.2 MLP στο WEKA

Κατ' αρχήν θα δημιουργηθεί αρχείο τύπου ARFF[3] (Attribute-Relation File Format), μέσα από το περιβάλλον της R, κάνοντας χρήση του πακέτου ***foreign***. Η τυποποίηση ARFF αποτελεί τη φυσική μέθοδο αποθήκευσης δεδομένων στο WEKA.

```
% Εξαγωγή δεδομένων σε μορφή arff (από το περιβάλλον της R)
library(foreign)
write.arff(dataset,'imagedata.arff')
```

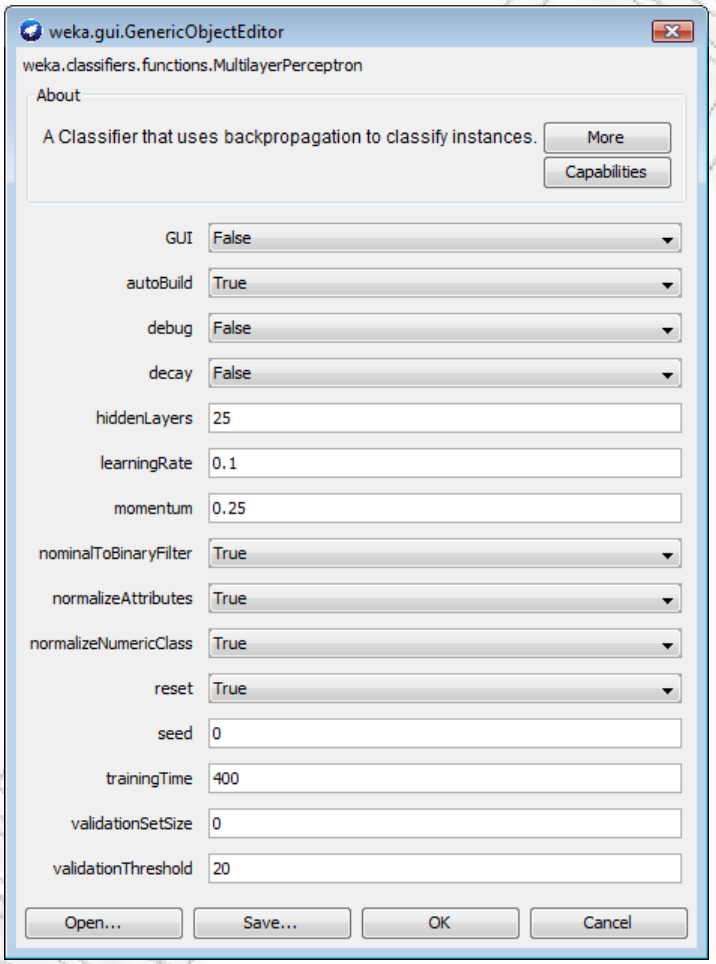
Στη συνέχεια, από το παράθυρο Explorer του WEKA, φορτώνουμε το αρχείο imagedata.arff. Εάν η κλάση (attribute 401) έχει τύπο *Numeric*, τη μετατρέπουμε σε κατηγορική κάνοντας χρήση του φίλτρου ***NumericToNominal***. Στην καρτέλα Classify, μέσω του κουμπιού "*Choose*", επιλέγουμε στην κατηγορία "*functions*" το ***MultilayerPerceptron***.

Στο παράθυρο ορισμού των παραμέτρων (βλ. Σχήμα 4-9) η βασικότερη παράμετρος είναι η ***hiddenLayers***. Στο πεδίο αυτό δίνεται μια λίστα ακέραιων θετικών τιμών, χωρισμένων με

[1] Στη συγκεκριμένη περίπτωση ο αριθμός αυτός είναι 401*25+25*10 = 10285.
[2] Βλ. http://finzi.psych.upenn.edu/R/library/caret/html/confusionMatrix.html
[3] Βλ. http://weka.wikispaces.com/ARFF

κόμμα, τόσες όσα και τα κρυφά στρώματα, με κάθε τιμή να δηλώνει τον αριθμό των μονάδων του αντίστοιχου κρυφού στρώματος. Αν επιθυμούμε να μην υπάρχει κανένα κρυφό στρώμα, δίνουμε 0. Στη περίπτωσή μας δίνουμε μόνο την τιμή 25 (δηλ. ένα κρυφό στρώμα 25 μονάδων). Προκειμένου να ληφθούν υπόψη οι τιμές που θα δοθούν εδώ, θα πρέπει η παράμετρος ***autobuild*** να είναι **True** (προεπιλεγμένη τιμή).

Σχήμα 4-9
WEKA: Ορισμός παραμέτρων για το MultilayerPerceptron



Στο ***trainingTime*** που δηλώνει τον αριθμό βημάτων δίνουμε, όπως πριν, την τιμή 400. Αναλυτική περιγραφή για κάθε μία από αυτές δίνεται στο παράθυρο "*Information*" που ανοίγει με το κουμπί "*More*".

Σημειώνεται ότι μπορεί να ζητηθεί η επιλογή βέλτιστων τιμών για επιλεγμένες παραμέτρους. Ένας από τους τρόπους για να γίνει αυτό στο WEKA αναφέρεται στην

Ενότητα 6.9.3. Για τις παραμέτρους ***learningRate*** και ***momentum*** οι τιμές που χρησιμοποιήθηκαν προέκυψαν από μια τέτοια διαδικασία.

Στη συνέχεια πατάμε OK για να κλείσει το παράθυρο, επιλέγουμε "*Use training set*" στο πλαίσιο "*Test options*", ώστε να ελέγξουμε την ακρίβεια του μοντέλου επί του συνόλου εκπαίδευσης, σε αντιστοιχία με τα προηγούμενα και πατάμε "*Start*".

Στο παράθυρο "*Classifier output*" θα πάρουμε τα αποτελέσματα, σύμφωνα με τα οποία, το ποσοστό ακρίβειας των προβλέψεων επί του συνόλου εκπαίδευσης είναι **98,96%**.

Για περισσότερες λεπτομέρειες σχετικά με τον τρόπο υλοποίησης του ταξινομητή MultilayerPerceptron στο WEKA, τις παραμέτρους που δέχεται αλλά και τα μέτρα αξιολόγησης της απόδοσης του μοντέλου, που αναγράφονται στα αποτελέσματα, βλ. [36].

Κ Ε Φ Α Λ Α Ι Ο 5

Ταξινομητές Μεγίστου Περιθωρίου

5.1 Εισαγωγή

Οι προσεγγίσεις για την αντιμετώπιση του προβλήματος δυαδικής ταξινόμησης των προηγούμενων κεφαλαίων μπορεί να οδηγήσουν σε επιφάνειες απόφασης που ενέχουν τον κίνδυνο εσφαλμένης ταξινόμησης δεδομένων που δεν είναι μέρος του συνόλου εκπαίδευσης. Για τον απλό αλγόριθμο μάθησης του Κεφ. 2, οι στρεβλώσεις της επιφάνειας απόφασης που ενδεχομένως να οδηγήσουν σε σφάλματα ταξινόμησης, μπορεί να προκύψουν λόγω πιθανών ακραίων παρατηρήσεων.

Στην εκμάθηση perceptron του Κεφ. 4, ο αλγόριθμος εκπαίδευσης θα τερματίσει αμέσως όταν βρεθεί η πρώτη αποδεκτή επιφάνεια απόφασης για το δεδομένο σύνολο εκπαίδευσης. Η τεχνική που ακολουθείται δεν παρέχει καμία εγγύηση ότι η επιφάνεια αυτή θα έχει αντίστοιχη επιτυχία στις προβλέψεις για τα υπόλοιπα αντικείμενα του πληθυσμού, με εκείνη που έχει για τα στοιχεία του συνόλου εκπαίδευσης, κάτι που και πάλι μπορεί να οδηγήσει σε εσφαλμένες ταξινομήσεις των νέων δεδομένων.

Στο παρόν κεφάλαιο θα παρουσιαστεί μια νέα μέθοδος που προσπαθεί να αποφύγει αυτό το μειονέκτημα. Η μέθοδος αυτή βασίζεται στην αναζήτηση μιας επιφάνειας η οποία θα ισαπέχει από τα όρια των δύο κλάσεων, στην περιοχή εκείνη όπου η απόσταση μεταξύ των ορίων των δύο κλάσεων είναι η μικρότερη. Ταυτόχρονα μεγιστοποιεί τις αποστάσεις από τα όρια αυτά. Ορίζοντας με αυτό τον τρόπο την επιφάνεια απόφασης μειώνεται η πιθανότητα εσφαλμένων ταξινομήσεων. Τα μοντέλα που υλοποιούν αυτή τη μεθοδολογία καλούνται «***ταξινομητές μεγίστου περιθωρίου***» (***maximum-margin classifiers***).

Το γεγονός ότι αναζητούμε τη βέλτιστη επιφάνεια υπό ένα κριτήριο σημαίνει ότι έχουμε να κάνουμε με ένα *πρόβλημα βελτιστοποίησης* (optimization problem) και όπως θα δούμε, η κατασκευή ενός ταξινομητή μεγίστου περιθωρίου είναι στην πραγματικότητα ένα πρόβλημα «***κυρτής βελτιστοποίησης***» (***convex optimization***) που μπορεί να λυθεί με τεχνικές τετραγωνικού προγραμματισμού (quadratic programming - QP).

Τα σημεία του συνόλου εκπαίδευσης που αντιπροσωπεύουν τους μεγαλύτερους περιορισμούς σχετικά με τη θέση μιας τέτοιας βέλτιστης επιφάνειας και τα οποία καλούνται «***διανύσματα υποστήριξης***» (***support vectors***), σχετίζονται με τα σημεία με υψηλές τιμές του α στη δυϊκή μορφή του αλγορίθμου perceptron.

5.2 Προβλήματα Βελτιστοποίησης

Τα προβλήματα στα οποία είναι επιθυμητή η επιλογή μιας βέλτιστης λύσης από έναν αριθμό εφικτών λύσεων καλούνται προβλήματα βελτιστοποίησης. Συνήθως οι εφικτές λύσεις κατατάσσονται με βάση μία αντικειμενική συνάρτηση και ο στόχος είναι να βρεθεί η εφικτή λύση που ελαχιστοποιεί (ή μεγιστοποιεί) την τιμή αυτής της συνάρτησης. Στα περισσότερα προβλήματα βελτιστοποίησης τίθενται επίσης και μια σειρά περιορισμών οι οποίοι μειώνουν ακόμη περισσότερο τις εφικτές λύσεις. Ένα πρόβλημα βελτιστοποίησης μπορεί να διατυπωθεί ως εξής:

$$\min_{\bar{x}} \phi(\bar{x}), \tag{5.1}$$

έτσι ώστε

$$h_i(\bar{x}) \geq c_i, \text{ με } i = 1, \ldots, l \tag{5.2}$$

για όλα τα $\bar{x} \in \mathbb{R}^n$. Εδώ η συνάρτηση $\phi : \mathbb{R}^n \to \mathbb{R}$ είναι η *αντικειμενική συνάρτηση* και κάθε συνάρτηση $h_i : \mathbb{R}^n \to \mathbb{R}$ καλείται *περιορισμός* με *όριο* το c_i. Κάθε $\bar{x} \in \mathbb{R}^n$ που ικανοποιεί τους περιορισμούς καλείται *εφικτή* λύση. Η βελτιστοποίηση έχει ως στόχο να βρει την εφικτή λύση $\bar{x}^*$, η οποία ελαχιστοποιεί την αντικειμενική συνάρτηση έτσι ώστε για κάθε άλλη εφικτή λύση $\bar{q} \in \mathbb{R}^n$ να ισχύει

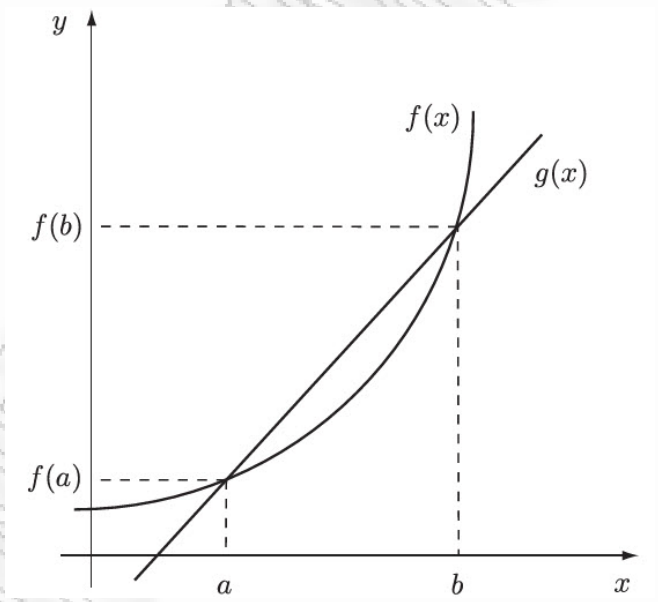
$$\phi(\bar{x}^*) \leq \phi(\bar{q}). \tag{5.3}$$

Τα προβλήματα βελτιστοποίησης ταξινομούνται ανάλογα με τις ιδιότητες των αντίστοιχων αντικειμενικών συναρτήσεων και περιορισμών. Για παράδειγμα, ένα πρόβλημα γραμμικής

βελτιστοποίησης έχει γραμμική αντικειμενική συνάρτηση και γραμμικούς περιορισμούς. Δηλαδή τόσο η αντικειμενική συνάρτηση όσο και οι περιορισμοί παριστάνονται με γραμμές, επίπεδα ή υπερεπίπεδα στους αντίστοιχους χώρους. Αν η αντικειμενική συνάρτηση ή οι περιορισμοί είναι μη γραμμικοί, τότε και το πρόβλημα βελτιστοποίησης καλείται μη γραμμικό.

Εδώ θα μας απασχολήσουν προβλήματα *κυρτής βελτιστοποίησης*. Ένα πρόβλημα κυρτής βελτιστοποίησης έχει κυρτή αντικειμενική συνάρτηση και γραμμικούς περιορισμούς. Τα προβλήματα κυρτής βελτιστοποίησης έχουν ιδιαίτερα καλή συμπεριφορά, καθώς η αντικειμενική συνάρτηση παρουσιάζει ολικό ελάχιστο και η επιφάνεια της συνάρτησης είναι ομαλή, υπό την έννοια ότι μπορεί να χαραχθεί ευθεία από ένα σημείο της επιφάνειας προς οποιοδήποτε άλλο σημείο αυτής, χωρίς στο ενδιάμεσο η ευθεία αυτή να διασχίσει ποτέ την επιφάνεια.

Σχήμα 5-1
Κυρτή συνάρτηση



Για να φανεί καλύτερα αυτό, θεωρήστε τη συνάρτηση $f:\mathbb{R}\to\mathbb{R}$ (Σχήμα 5-1). Έστω $a,b\in\mathbb{R}$, με $a<b$ και $g:\mathbb{R}\to\mathbb{R}$ μια γραμμική συνάρτηση τέτοια ώστε $g(a)=f(a)$ και $g(b)=f(b)$. Η g είναι δηλαδή μια ευθεία που τέμνει τη γραφική παράσταση της f στα σημεία $(a,f(a))$ και $(b,f(b))$. Η συνάρτηση f θα λέγεται κυρτή αν $f(x)\le g(x)$ για όλα τα $x\in\mathbb{R}$ με $a<x<b$.

Παραδείγματα κυρτών συναρτήσεων είναι οι συναρτήσεις που το όρισμά τους υψώνεται σε θετική άρτια και ακέραια δύναμη (π.χ. $f(x) = x^2$). Υπάρχουν δε αποδοτικοί αλγόριθμοι, οι οποίοι επωφελούνται από τις ιδιότητες μιας κυρτής αντικειμενικής συνάρτησης προκειμένου να επιλύσουν ένα πρόβλημα κυρτής βελτιστοποίησης. Μια τέτοια τεχνική είναι αυτή του τετραγωνικού προγραμματισμού, που θα συζητηθεί στην Ενότητα 5.5.

5.3 Μέγιστα Περιθώρια

Γίνεται ίσως διαισθητικά αντιληπτό ότι, για ένα γραμμικώς διαχωρίσιμο σύνολο εκπαίδευσης σε ένα πρόβλημα δυαδικής ταξινόμησης, η βέλτιστη επιφάνεια απόφασης ισαπέχει από τα όρια των κλάσεων. Αυτό μπορεί άτυπα να δικαιολογηθεί με το επιχείρημα ότι, καθώς το σύνολο εκπαίδευσης απεικονίζει μόνο κατά προσέγγιση τον πληθυσμό των δεδομένων, θέτοντας την επιφάνεια απόφασης έτσι ώστε να ισαπέχει από τα όρια των δύο κλάσεων, αυξάνεται η πιθανότητα σωστής ταξινόμησης νέων σημείων. Μάλιστα, η πιθανότητα αυτή θα αυξηθεί ακόμη περισσότερο αν μεγιστοποιηθούν οι αποστάσεις των ορίων από την επιφάνεια απόφασης[1]. Στο Σχήμα 5-2 επιχειρείται μια απεικόνιση των παραπάνω στο χώρο $\mathbb{R}^2$.

Προκειμένου να κατασκευαστούν τέτοιες βέλτιστες επιφάνειες είναι απαραίτητο να οριστούν δύο επιπλέον έννοιες:

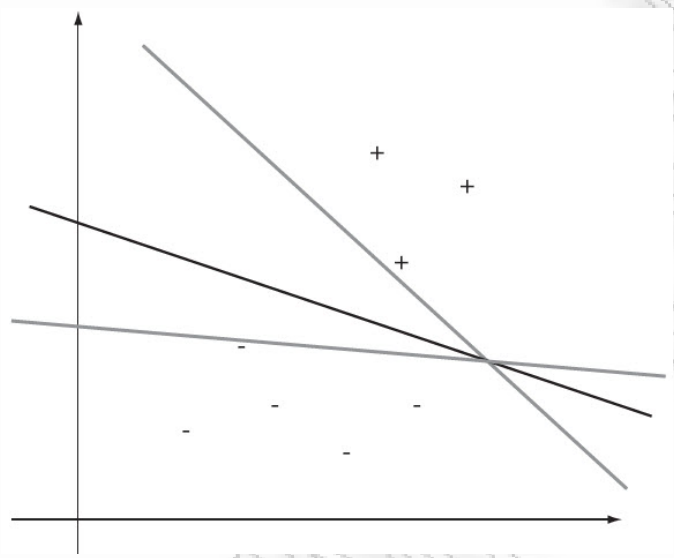
Ορισμός 5.1 *Ένα υπερεπίπεδο υποστηρίζει μία κλάση αν είναι παράλληλο σε μια* (*γραμμική*) *επιφάνεια απόφασης και όλα τα σημεία της αντίστοιχης κλάσης είναι είτε επάνω, είτε κάτω από αυτό. Ένα τέτοιο υπερεπίπεδο καλείται «**υπερεπίπεδο στήριξης**»* (***supporting hyperplane***).

Ένα τέτοιο υπερεπίπεδο θα μπορούσε να προκύψει αν μετακινούσαμε ένα αντίγραφο της επιφάνειας απόφασης έως ότου αγγίξει το όριο μιας από τις κλάσεις. Στο πρόβλημα δυαδικής ταξινόμησης υπάρχουν δύο υπερεπίπεδα στήριξης: ένα που θα προκύψει από τη μεταφορά προς την κατεύθυνση της κλάσης $+1$ και ένα από τη μεταφορά προς την κατεύθυνση της κλάσης -1.

[1] Για την απόδειξη αυτού του ισχυρισμού βλ. [10], Κεφ. 10.

Η δεύτερη έννοια που απαιτείται είναι αυτή του περιθωρίου, η οποία είναι κρίσιμη για τη συγκεκριμένη διαδικασία κατασκευής βέλτιστης επιφάνειας απόφασης.

Σχήμα 5-2
Βέλτιστη (μαύρη) και υποβέλτιστες (γκρι) επιφάνειες απόφασης



Ορισμός 5.2 *Σε ένα πρόβλημα δυαδικής ταξινόμησης η απόσταση μεταξύ των δύο φερόντων υπερεπιπέδων καλείται «**περιθώριο**» (**margin**).*

Με τη βοήθεια των δύο αυτών εννοιών μπορούμε πλέον να διατυπώσουμε με ποσοτικούς όρους το κριτήριο που κάνει μια επιφάνεια απόφασης βέλτιστη.

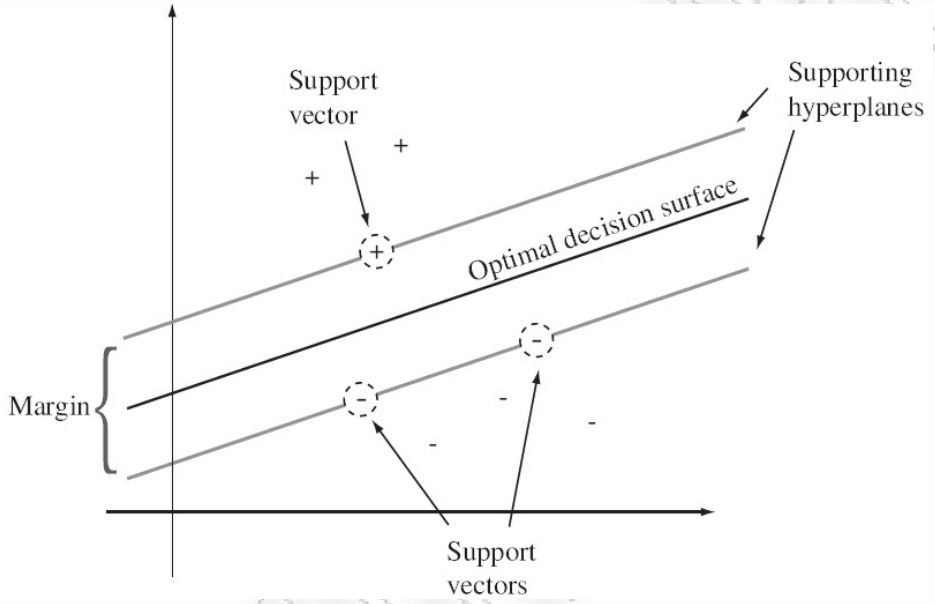
Ορισμός 5.3 *Μία επιφάνεια απόφασης για ένα πρόβλημα δυαδικής ταξινόμησης είναι «**βέλτιστη**» αν ισαπέχει από τα υπερεπίπεδα στήριξης και μεγιστοποιεί το περιθώριό τους.*

Αυτό σημαίνει ότι το πρόβλημα βελτιστοποίησης περιλαμβάνει την εξεύρεση μιας επιφάνειας απόφασης που να επιτρέπει στα υπερεπίπεδα στήριξης να μετατεθούν όσο το δυνατόν μακρύτερα από αυτή, μεγιστοποιώντας έτσι το περιθώριο και κατόπιν την τοποθέτηση της επιφάνειας απόφασης έτσι ώστε να ισαπέχει από τα υπερεπίπεδα στήριξης.

Τα προαναφερθέντα απεικονίζονται στο Σχήμα 5-3, όπου βλέπουμε τα δύο υπερεπίπεδα στήριξης να έχουν μετατεθεί τόσο ώστε ίσα που να εφάπτονται στα αντίστοιχα όρια. Η απόσταση μεταξύ των υπερεπιπέδων είναι το περιθώριο και η βέλτιστη επιφάνεια απόφασης βρίσκεται στο μέσον του. Παρατηρήστε ότι το μέγεθος του περιθωρίου περιορίζεται από τα σημεία κάθε κλάσης που εμφανίζονται σε κύκλο και καλούνται «***διανύσματα υποστήριξης***»

(***support vectors***). Παρατηρήστε επίσης ότι το περιθώριο έχει τη μέγιστη δυνατή τιμή, καθώς κάθε περιστροφή ή μετάθεση της επιφάνειας απόφασης θα είχε σαν αποτέλεσμα ένα μικρότερο περιθώριο. Ως εκ τούτου, ο στόχος ενός ταξινομητή μεγίστου περιθωρίου είναι να βρει τη θέση της επιφάνειας απόφασης για την οποία μεγιστοποιείται το περιθώριο.

Σχήμα 5-3
Βέλτιστη επιφάνεια και υπερεπίπεδα στήριξης



5.4 Βελτιστοποίηση του Περιθωρίου

Η εξεύρεση μιας επιφάνειας απόφασης που να μεγιστοποιεί το περιθώριο μεταξύ δύο φερόντων υπερεπιπέδων αποτελεί ένα πρόβλημα βελτιστοποίησης, στο οποίο οι εφικτές λύσεις περιλαμβάνουν όλες τις δυνατές επιφάνειες απόφασης, με τα αντίστοιχα κάθε φορά υπερεπίπεδα στήριξης. Για κάθε μία από αυτές τις εφικτές λύσεις, η αντικειμενική συνάρτηση του προβλήματος υπολογίζει το μέγεθος του περιθωρίου που της αντιστοιχεί. Μεγιστοποιώντας την αντικειμενική συνάρτηση θα καταλήξουμε στο ζητούμενο μέγιστο περιθώριο. Οι περιορισμοί στην περίπτωση αυτή είναι οι θέσεις των φερόντων υπερεπιπέδων, τα οποία δεν επιτρέπεται να διασχίσουν τα όρια των αντίστοιχων κλάσεων. Αυτό μπορεί να διατυπωθεί ως εξής

$$m^* = \max \phi(\bar{w}, b), \tag{5.4}$$

υπό τους περιορισμούς που τίθενται από τα υπερεπίπεδα στήριξης.

Η αντικειμενική συνάρτηση $\phi(\bar{w},b)$ υπολογίζει το περιθώριο για μια δοθείσα επιφάνεια απόφασης $\bar{w}\bullet\bar{x}=b$. Το μέγιστο περιθώριο m^* προκύπτει από μια βέλτιστη επιφάνεια απόφασης $\bar{w}^*\bullet\bar{x}=b^*$. Προκειμένου να καταστεί δυνατή η επίλυση αυτού το προβλήματος θα πρέπει να βρεθεί μια κατάλληλη έκφραση για την αντικειμενική συνάρτηση ϕ. Για το σκοπό αυτό θα χρειαστεί η έννοια της *προβολής*.

Ορισμός 5.4 *Έστω $\bar{a}$ και $\bar{b}$ διανύσματα του $\mathbb{R}^n$ που σχηματίζουν γωνία γ μεταξύ τους. Η «**προβολή**» p_a του $\bar{a}$ στην κατεύθυνση του $\bar{b}$ δίνεται από τον τύπο*

$$p_a=|\bar{a}|\cos\gamma=\frac{\bar{a}\bullet\bar{b}}{|\bar{b}|}. \tag{5.5}$$

Μπορούμε τώρα να συνάγουμε τη μορφή της αντικειμενικής συνάρτησης. Έστω το γραμμικά διαχωρίσιμο σύνολο εκπαίδευσης

$$D=\{(\bar{x}_1,y_1),(\bar{x}_2,y_2),\ldots,(\bar{x}_l,y_l)\}\subseteq\mathbb{R}^n\times\{+1,-1\}. \tag{5.6}$$

Ας υποθέσουμε επίσης ότι γνωρίζουμε τη βέλτιστη επιφάνεια απόφασης για το σύνολο αυτό,

$$\bar{w}^*\bullet\bar{x}=b^*. \tag{5.7}$$

Θα ισχύουν τότε οι παρακάτω ταυτότητες:

$$m^*=\phi(\bar{w}^*,b^*)=\max\phi(\bar{w},b). \tag{5.8}$$

Το βέλτιστο περιθώριο m^* προκύπτει από την αντικειμενική συνάρτηση ϕ δοθεισών των παραμέτρων $\bar{w}^*$ και b^* της βέλτιστης επιφάνειας απόφασης.

Καθώς η επιφάνεια απόφασης της σχέσης (5.7) είναι μια επιφάνεια μεγίστου περιθωρίου, θα υπάρχουν δύο υπερεπίπεδα στήριξης σε ίσες αποστάσεις από αυτήν, δηλαδή

$$\bar{w}^*\bullet\bar{x}=b^*+k \tag{5.9}$$

$$\bar{w}^*\bullet\bar{x}=b^*-k \tag{5.10}$$

Το πρώτο από αυτά είναι το υπερεπίπεδο στήριξης για την κλάση $+1$ και βρίσκεται πάνω από την επιφάνεια απόφασης, ενώ το δεύτερο αντιστοιχεί στην κλάση -1 και είναι από κάτω (βλ. Σχήμα 5-4).

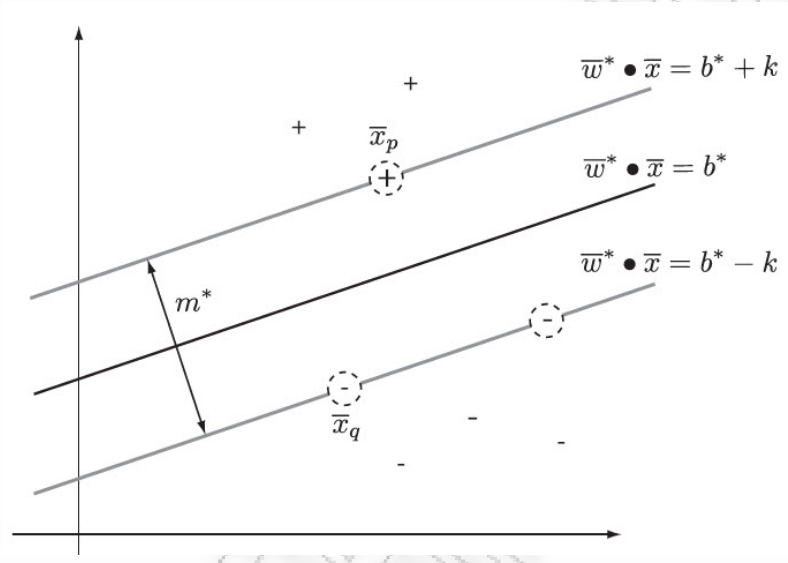
Επιπρόσθετα, καθώς η επιφάνεια (5.7) είναι η βέλτιστη επιφάνεια απόφασης, τα υπερεπίπεδα στήριξης θα διέρχονται από ορισμένα διανύσματα υποστήριξης. Έστω $(\bar{x}_p,+1)\in D$ ένα διάνυσμα υποστήριξης για την κλάση $+1$ με

$$\bar{w}^*\bullet\bar{x}_p=b^*+k \tag{5.11}$$

επί του υπερεπιπέδου στήριξης (5.9). Ομοίως, έστω το $(\overline{x}_q, -1) \in D$ ένα διάνυσμα υποστήριξης για την κλάση -1, επί του υπερεπιπέδου στήριξης (5.10), με

$$\overline{w}^* \bullet \overline{x}_q = b^* - k \tag{5.12}$$

Σχήμα 5-4
Περιθώριο m^* για τη βέλτιστη επιφάνεια $\overline{w}^* \bullet \overline{x} = b^*$



Εξ' ορισμού, η απόσταση μεταξύ των δύο φερόντων επιπέδων ισούται με το περιθώριο m^*. Η απόσταση αυτή μπορεί να υπολογιστεί ως η προβολή του διανύσματος $\overline{x}_p - \overline{x}_q$ στην κατεύθυνση του διανύσματος $\overline{w}^*$, που είναι κάθετο στην επιφάνεια απόφασης (Σχήμα 5-5), δηλαδή

$$m^* = \left|\overline{x}_p - \overline{x}_q\right| \cos\gamma = \frac{\overline{w}^* \bullet \left(\overline{x}_p - \overline{x}_q\right)}{\left|\overline{w}^*\right|} \qquad \text{από την (5.5)}$$

$$= \frac{\overline{w}^* \bullet \overline{x}_p - \overline{w}^* \bullet \overline{x}_q}{\left|\overline{w}^*\right|} = \frac{\left(b^* + k\right) - \left(b^* - k\right)}{\left|\overline{w}^*\right|} \qquad \text{από τις (5.11) και (5.12)}$$

$$= \frac{2k}{\left|\overline{w}^*\right|} \tag{5.13}$$

όπου γ η γωνία μεταξύ των διανυσμάτων $\overline{w}^*$ και $\overline{x}_p - \overline{x}_q$. Έτσι καταλήγουμε στην έκφραση βελτιστοποίησης

$$m^* = \max \frac{2k}{|\overline{w}|} \tag{5.14}$$

η οποία μπορεί να εκφραστεί ως ελαχιστοποίηση και να λάβει τη μορφή:

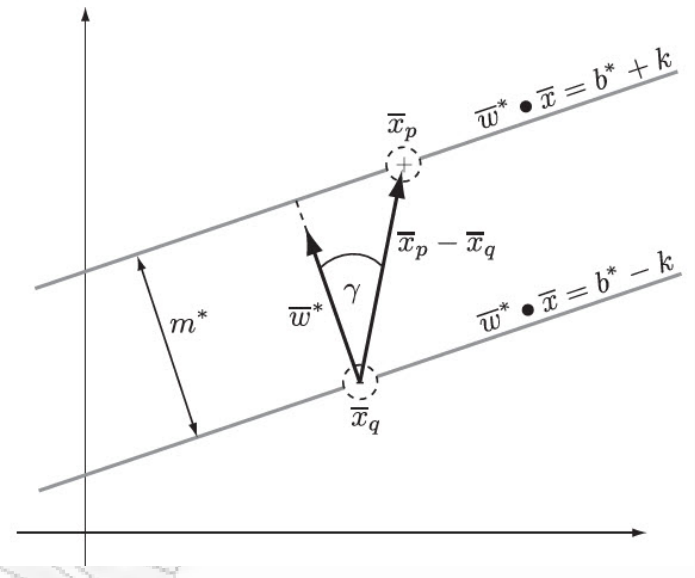
$$m^* = \min \frac{|\overline{w}|}{2k}. \tag{5.15}$$

Για την μεγιστοποίηση του περιθωρίου μπορούμε ισοδύναμα να χρησιμοποιήσουμε την αντικειμενική συνάρτηση

$$\phi(\overline{w}, b) = \frac{1}{2}\overline{w} \bullet \overline{w}, \tag{5.16}$$

καθώς έτσι οδηγούμαστε στην αντιμετώπιση ενός δευτεροβάθμιου προγράμματος. Δεδομένου ότι η βελτιστοποίηση δεν επηρεάζεται από αλλαγή κλίμακας, μπορούμε να επιλέξουμε μια βολική τιμή για τη σταθερά k. Στην περίπτωσή μας επιλέχθηκε η τιμή $k = 1$.

Σχήμα 5-5
Υπολογισμός του περιθωρίου μεταξύ δύο φερόντων επιπέδων



Ίσως να φαίνεται περίεργο το γεγονός ότι η αντικειμενική συνάρτηση δεν περιλαμβάνει τον όρο b προς βελτιστοποίηση. Θα δούμε όμως ότι ο όρος μετάθεσης θα εμφανιστεί στη διατύπωση των περιορισμών.

Ας επικεντρωθούμε τώρα στους περιορισμούς του προβλήματος βελτιστοποίησης. Η ιδέα είναι ότι τα υπερεπίπεδα στήριξης των αντίστοιχων κλάσεων θα πρέπει να παραμένουν ως τέτοια καθ' όλη τη διάρκεια των βελτιστοποιήσεων. Δηλαδή δεν πρέπει σε καμία περίπτωση

να επιτραπεί σε αυτά να διασχίσουν τα αντίστοιχα όρια των κλάσεων. Αυτό σημαίνει ότι για τα υπερεπίπεδα στήριξης (5.9) και (5.10) θα πρέπει να ισχύουν αντίστοιχα:

$$\bar{w}^* \bullet \bar{x}_i \geq b^* + k \quad \text{για όλα τα } (\bar{x}_i, y_i) \in D \text{ με } y_i = +1, \tag{5.17}$$

$$\bar{w}^* \bullet \bar{x}_i \leq b^* - k \quad \text{για όλα τα } (\bar{x}_i, y_i) \in D \text{ με } y_i = -1. \tag{5.18}$$

Λαμβάνοντας υπόψη την απλουστευτική παραδοχή $k = 1$ στην (5.15), έχουμε

$$\bar{w} \bullet \bar{x}_i \geq 1 + b \quad \text{για όλα τα } (\bar{x}_i, y_i) \in D \text{ με } y_i = +1, \tag{5.19}$$

$$\bar{w} \bullet (-\bar{x}_i) \geq 1 - b \quad \text{για όλα τα } (\bar{x}_i, y_i) \in D \text{ με } y_i = -1 \tag{5.20}$$

που μπορούν να γραφούν σε μια πιο συμπαγή μορφή ως εξής:

$$\bar{w} \bullet (y_i \bar{x}_i) \geq 1 + y_i b \quad \text{για όλα τα } (\bar{x}_i, y_i) \in D. \tag{5.21}$$

Στη μορφή αυτή είναι άμεσα ορατό το γεγονός ότι *όλα* τα σημεία του συνόλου εκπαίδευσης επιφέρουν περιορισμούς. Η ακόλουθη πρόταση για τον υπολογισμό της βέλτιστης επιφάνειας απόφασης με το μέγιστο περιθώριο συνοψίζει όλα τα προηγούμενα, κάνοντας χρήση των (5.16) και (5.21).

***Πρόταση 5.1* (*Ταξινομητής Μεγίστου Περιθωρίου*)** *Δοθέντος ενός γραμμικά διαχωρίσιμου συνόλου εκπαίδευσης*

$$D = \{(\bar{x}_1, y_1), (\bar{x}_2, y_2), \ldots, (\bar{x}_l, y_l)\} \subseteq \mathbb{R}^n \times \{+1, -1\}$$

μπορούμε να υπολογίσουμε μια επιφάνεια απόφασης μεγίστου περιθωρίου $\bar{w}^* \bullet \bar{x} = b^*$ *με τη βελτιστοποίηση*

$$\min \phi(\bar{w}, b) = \min_{\bar{w}, b} \frac{1}{2} \bar{w} \bullet \bar{w} \tag{5.22}$$

υπό τους περιορισμούς

$$\bar{w} \bullet (y_i \bar{x}_i) \geq 1 + y_i b \quad \text{για όλα τα } (\bar{x}_i, y_i) \in D. \tag{5.23}$$

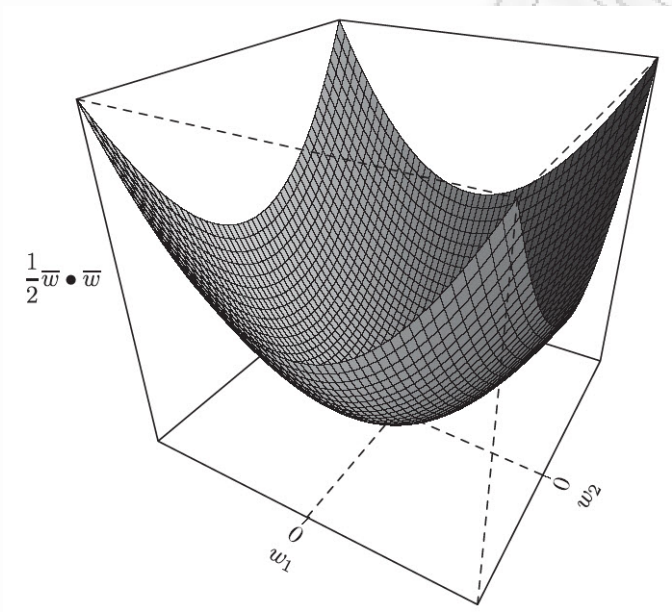
5.5 Τετραγωνικός Προγραμματισμός

Είναι εύκολο να διαπιστωθεί ότι η αντικειμενική συνάρτηση της (5.22) είναι κυρτή,

$$\phi(\bar{w}, b) = \frac{1}{2} \bar{w} \bullet \bar{w} = \frac{1}{2} \left(w_1^2 + \ldots + w_n^2 \right) \tag{5.24}$$

όπου $\overline{w} = (w_1, \ldots, w_n)$. Η γραφική της παράσταση στο χώρο $\mathbb{R}^2$ δίνεται στο Σχήμα 5-6. Η κυρτότητα συνεπάγεται την παρουσία ολικού ελαχίστου στην αντικειμενική συνάρτηση. Με άλλα λόγια, από ένα σύνολο εφικτών λύσεων μπορούμε να βρούμε τη λύση για την οποία η αντικειμενική συνάρτηση δίνει την ελάχιστη τιμή.

Σχήμα 5-6
Αντικειμενική συνάρτηση $\frac{1}{2}\overline{w} \bullet \overline{w}$ στο $\mathbb{R}^2$



Ένας αποτελεσματικός τρόπος για την επίλυση προβλημάτων κυρτής βελτιστοποίησης, της μορφής που αντιμετωπίζουμε εδώ, είναι μέσω *τετραγωνικού προγραμματισμού*. Οι περισσότεροι λύτες τετραγωνικού προγραμματισμού είναι συναρτήσεις της μορφής

$$\overline{w}^* = solve(\mathrm{Q}, \overline{q}, \mathrm{X}, \overline{c}) \tag{5.25}$$

Με την παραπάνω μορφή διατυπώνεται γενικά το πρόβλημα της κυρτής βελτιστοποίησης, το οποίο επίσης συχνά αναφέρεται και ως *τετραγωνικός προγραμματισμός*,

$$\overline{w}^* = \arg\min_{\overline{w}} \left(\frac{1}{2} \overline{w}^T \mathrm{Q} \overline{w} - \overline{q} \bullet \overline{w} \right) \tag{5.26}$$

υπό τους περιορισμούς

$$\mathrm{X}^{\mathrm{T}} \overline{w} \geq \overline{c} \tag{5.27}$$

Εδώ το Q είναι πίνακας διαστάσεων $n \times n$, το X πίνακας διαστάσεων $n \times l$, τα $\overline{w}^*, \overline{w}, \overline{q}$ είναι $n-$διάστατα διανύσματα και το $\overline{c}$ είναι $l-$διάστατο διάνυσμα.

Μπορούμε να μετασχηματίσουμε το γενικό πρόβλημα βελτιστοποίησης έτσι ώστε να έρθει στη μορφή της βελτιστοποίησης περιθωρίου της Πρότασης 5.1, αν θέσουμε όπου Q τον μοναδιαίο πίνακα I και $\bar{q} = \bar{0}$, οπότε θα έχουμε

$$\bar{w}^* = \arg\min_{\bar{w}} \left(\frac{1}{2} \bar{w}^T \mathrm{I} \bar{w} - \bar{0} \bullet \bar{w} \right) = \arg\min_{\bar{w}} \left(\frac{1}{2} \bar{w} \bullet \bar{w} \right) \tag{5.28}$$

Η διαφορά μεταξύ της αρχικής διατύπωσης του προβλήματος βελτιστοποίησης περιθωρίου και αυτής που χρησιμοποιεί ένα λύτη τετραγωνικού προγραμματισμού είναι ότι η τελευταία επιστρέφει το όρισμα που ελαχιστοποιεί την αντικειμενική συνάρτηση αντί για την ελαχιστοποιημένη τιμή της αντικειμενικής συνάρτησης (εξ ου και ο τελεστής argmin αντί του min). Αυτό δεν αποτελεί πρόβλημα, αφού μπορούμε πάντα να καταλήξουμε στο βέλτιστο περιθώριο εφαρμόζοντας τη σχέση (5.13) για την τιμή του βέλτιστου διανύσματος $\bar{w}^*$.

Έχοντας καταλήξει στη μορφή της αντικειμενικής συνάρτησης του τετραγωνικού προγράμματος, στρεφόμαστε τώρα στους περιορισμούς. Σε αντίθεση με τη μορφή που αυτοί είχαν στο αρχικά τεθέν πρόβλημα βελτιστοποίησης, οι περιορισμοί στην περίπτωση των τετραγωνικών προγραμμάτων εκφράζονται με τη μορφή της (5.27). Ωστόσο, με κάποια επεξεργασία μπορούμε να φέρουμε τους περιορισμούς μας σε αυτή τη ζητούμενη μορφή. Η σχέση (5.23) μπορεί να γραφεί:

$$\left(y_i \bar{x}_i \right) \bullet \bar{w} \geq 1 + y_i b \tag{5.29}$$

για όλα τα $(\bar{x}_i, y_i) \in D$ με $i = 1, \ldots, l$ και $\bar{x}_i = (x_i^{(1)}, \ldots, x_i^{(n)})^{\mathrm{T}}$. Βάσει αυτών, μπορεί να οριστεί ο πίνακας X ως

$$\mathrm{X} = \begin{pmatrix} y_1 x_1^{(1)} & \ldots & y_i x_i^{(1)} & \ldots & y_l x_l^{(1)} \\ \vdots & & \vdots & & \vdots \\ y_1 x_1^{(n)} & \ldots & y_i x_i^{(n)} & \ldots & y_l x_l^{(n)} \end{pmatrix}. \tag{5.30}$$

Με άλλα λόγια, κάθε στήλη του X ισούται με το διάνυσμα $y_i \bar{x}_i = (y_i x_i^{(1)}, \ldots, y_i x_i^{(n)})^{\mathrm{T}}$. Από την άλλη μεριά, το διάνυσμα $\bar{c}$ μπορεί να λάβει τη μορφή

$$\bar{c} = \begin{pmatrix} 1 + y_1 b \\ 1 + y_2 b \\ \vdots \\ 1 + y_l b \end{pmatrix}. \tag{5.31}$$

Με βάσει τους προηγούμενους ορισμούς, είναι απλό να φανεί ότι η διατύπωση των περιορισμών με τη μορφή της (5.27) είναι μια περισσότερο συμπαγής μορφή των περιορισμών που διατυπώνονται στην (5.29).

Παρατηρήστε ότι η ελεύθερη παράμετρος b έχει εισαχθεί στο τετραγωνικό πρόβλημα μέσω του διανύσματος $\overline{c}$. Συνεπώς, μέσω της διαδικασίας βελτιστοποίησης, θα πρέπει να ελαχιστοποιηθεί η αντικειμενική συνάρτηση ως προς τα $\overline{w}$ και b.

Η ακόλουθη πρόταση συνοψίζει όλα τα παραπάνω και εκφράζει τη βελτιστοποίηση μεγίστου περιθωρίου με όρους προβλήματος τετραγωνικού προγραμματισμού:

Πρόταση 5.2 *Δοθέντος ενός γραμμικά διαχωρίσιμου συνόλου εκπαίδευσης*

$$D=\left\{\left(\overline{x}_1,y_1\right),\left(\overline{x}_2,y_2\right),\ldots,\left(\overline{x}_l,y_l\right)\right\}\subseteq\mathbb{R}^n\times\left\{+1,-1\right\}$$

μπορεί να υπολογιστεί η επιφάνεια απόφασης μεγίστου περιθωρίου $\overline{w}^*\bullet\overline{x}=b^*$ *μέσω τετραγωνικού προγραμματισμού που επιλύει το γενικευμένο πρόβλημα βελτιστοποίησης*

$$\left(\overline{w}^*,b^*\right)=\arg\min_{\overline{w},b}\left(\frac{1}{2}\overline{w}^T\mathrm{Q}\overline{w}-\overline{q}\bullet\overline{w}\right) \tag{5.32}$$

υπό τους περιορισμούς

$$\mathrm{X}^{\mathrm{T}}\overline{w}\geq\overline{c} \tag{5.33}$$

με $\mathrm{Q}=\mathrm{I},\overline{q}=\overline{0}$ *και όπου τα* X *και* $\overline{c}$ *κατασκευάζονται σύμφωνα με τις (5.30) και (5.31) αντίστοιχα.*

Ο Αλγόριθμος 5.1 παρουσιάζει τη διαδικασία υπολογισμού μιας επιφάνειας απόφασης μεγίστου περιθωρίου, με χρήση λύτη τετραγωνικού προγράμματος.

Η ποσότητα r είναι η ακτίνα του συνόλου εκπαίδευσης D. Η σταθερά q καθορίζει το μέγεθος του διαστήματος στο οποίο θα αναζητηθεί τιμή για τον όρο μετάθεσης και της έχει τεθεί η τιμή 1000. Ωστόσο, ακριβέστερη εκτίμηση της τιμής της θα πρέπει να προσδιοριστεί πειραματικά, καθώς αυτή είναι άκρως εξαρτημένη από τα εκάστοτε διαθέσιμα δεδομένα εκπαίδευσης.

Για τις βέλτιστες τιμές του διανύσματος $\overline{w}^*$ και του όρου μετάθεσης b^* δεν ορίζονται αρχικές τιμές και παραμένουν ακαθόριστες έως ότου ο αλγόριθμος βρει κάποια λύση που ικανοποιεί τους περιορισμούς.

Αλγόριθμος 5.1

let $D = \{(\bar{x}_1, y_1), (\bar{x}_2, y_2), \ldots, (\bar{x}_l, y_l)\} \subset \mathbb{R}^n \times \{+1, -1\}$
$r \leftarrow \max\{|\bar{x}| \mid (\bar{x}, y) \in D\}$
$q \leftarrow 1000$
let $\bar{w}^*$ και b^* μη προσδιορισμένα
Δημιούργησε το X σύμφωνα με την (5.30), βάσει του D.
for each $b \in [-q, q]$ **do**
  Δημιούργησε το $\bar{c}$ σύμφωνα με την (5.31), χρησιμοποιώντας το b.
  $\bar{w} = solve(\mathrm{I}, \bar{0}, \mathrm{X}, \bar{c})$
  if ($\bar{w}$ προσδιορισμένο **and** $\bar{w}^*$ μη προσδιορισμένο) **or**
    ($\bar{w}$ προσδιορισμένο **and** $|\bar{w}| < |\bar{w}^*|$) **then**
      $\bar{w}^* \leftarrow \bar{w}$
      $b^* \leftarrow b$
  end if
end for
if $\bar{w}^*$ μη προσδιορισμένο **then**
  stop οι περιορισμοί δεν είναι ικανοποιήσιμοι
else if $|\bar{w}^*| > q/r$ **then**
  stop η υπόθεση οριοθέτησης του $|\bar{w}|$ παραβιάζεται
end if
return $(\bar{w}^*, b^*)$

Δεδομένου ότι ο όρος μετάθεσης b αποτελεί ελεύθερη παράμετρο, χρειάζεται να επιλεγούν κατάλληλες τιμές για αυτόν. Προκειμένου να καταλήξουμε σε ένα εύλογο διάστημα τιμών του b, θεωρούμε την περιγραφή μιας επιφάνειας απόφασης από την εξίσωση

$$b = \bar{w} \bullet \bar{x} \tag{5.34}$$

η οποία μπορεί να γραφεί ως εξής

$$b = |\bar{w}||\bar{x}| \cos\gamma \tag{5.35}$$

όπου γ η γωνία μεταξύ των διανυσμάτων $\bar{w}$ και $\bar{x}$. Για $0 \le \gamma \le \pi$ έχουμε

$$-|\bar{w}||\bar{x}| \le b \le |\bar{w}||\bar{x}|. \tag{5.36}$$

Θεωρώντας μόνο σημεία εντός της υπερσφαίρας με ακτίνα r, δηλαδή μόνο σημεία με $|\bar{x}| \le r$

$$-|\bar{w}| r \le b \le |\bar{w}| r. \tag{5.37}$$

Ατυχώς όμως το $|\overline{w}|$ είναι μη φραγμένο, καθιστώντας έτσι τα όρια του b, όπως ορίστηκαν προηγουμένως, άχρηστα. Θεωρώντας όμως ότι σε ένα σύνολο εκπαίδευσης ακτίνας r το μέγιστο δυνατό περιθώριο είναι $2r$ και εισάγοντας αυτό στην εξίσωση (5.13), λαμβάνουμε

$$\frac{2}{|\overline{w}|} \leq 2r \tag{5.38}$$

δεχόμενοι και πάλι μια τιμή για το $k=1$, όπως προηγουμένως. Από τη σχέση αυτή προκύπτει ένα κάτω όριο για το $|\overline{w}|$

$$\frac{1}{r} \leq |\overline{w}| \tag{5.39}$$

Αυτό σημαίνει ότι $|\overline{w}| = 1/r$ για το μέγιστο δυνατό περιθώριο και $|\overline{w}| > 1/r$ για μικρότερα μεγέθη περιθωρίων. Για απειροελάχιστα μεγέθη περιθωρίων θα ισχύει $|\overline{w}| \to \infty$. Παρόλα αυτά όμως, εμείς ενδιαφερόμαστε για τη μεγιστοποίηση του περιθωρίου. Συνεπώς, επιφάνειες απόφασης με περιθώρια μικρότερα από κάποιο όριο δεν μπορούν να θεωρούνται μέρος των εφικτών λύσεων. Για να το εκφράσουμε αυτό περιορίζουμε τις τιμές του $|\overline{w}|$ ως ακολούθως:

$$\frac{1}{r} \leq |\overline{w}| \leq \frac{q}{r} \tag{5.40}$$

όπου q σταθερά που φράσσει την τιμή του $|\overline{w}|$ σε ένα πολλαπλάσιο του $1/r$. Επιλέγοντας $q=1000$, όπως εμφανίζεται στον αλγόριθμο, το ελάχιστο μέγεθος του περιθωρίου το οποίο μπορεί να θεωρηθεί ως εφικτή λύση είναι 1000 φορές μικρότερο του μέγιστου δυνατού, εντός ενός συνόλου εκπαίδευσης ακτίνας r. Αντικαθιστώντας την (5.40) στην (5.37) έχουμε

$$-q \leq b \leq q \tag{5.41}$$

που αντιστοιχεί στα όρια που καθορίζονται για το b στον αλγόριθμο.

Ο Αλγόριθμος 5.1 μπορεί να αποτύχει να καταλήξει σε λύση για δύο λόγους. Ο πρώτος είναι στην περίπτωση που δεν μπορούν να ικανοποιηθούν οι τεθέντες περιορισμοί για καμία τιμή του b στο προκαθορισμένο διάστημα. Ο δεύτερος είναι στην περίπτωση που παραβιάζεται η υπόθεση του ορίου του $|\overline{w}|$. Δεδομένου ότι έχουμε υποθέσει ένα γραμμικώς διαχωρίσιμο σύνολο εκπαίδευσης είναι εξασφαλισμένο ότι υφίσταται λύση και οι τυχόν αστοχίες υποδηλώνουν ότι οι παραδοχές οριοθέτησης για το b δεν είναι σωστές και ότι οι οριακές του τιμές θα πρέπει να αυξηθούν.

5.6 Ανακεφαλαίωση

Ορίζοντας το μέγιστο περιθώριο ως κριτήριο για την επιλογή επιφανειών απόφασης, έχουμε επιτύχει το στόχο να αποφύγουμε την επιλογή μιας υποβέλτιστης επιφάνειας απόφασης ως μοντέλου σε προβλήματα δυαδικής ταξινόμησης. Ωστόσο, κατά τον υπολογισμό τέτοιων ταξινομητών μεγίστου περιθωρίου με τη χρήση λυτών τετραγωνικών προγραμμάτων, καταλήξαμε στο συμπέρασμα ότι οι λύσεις εξαρτώνται από τις τιμές που επιλέγουμε για τον όρο μετάθεσης, ο οποίος αποτελεί ελεύθερη παράμετρο.

Αν και επιχειρήθηκε η παροχή κάποιων κατευθυντήριων γραμμών για τον τρόπο αναζήτησης του όρου μετάθεσης ο οποίος οδηγεί στο μέγιστο περιθώριο, η ακριβής του τιμή μπορεί να καθοριστεί μόνο μέσω πειραματισμών. Η αναζήτηση βέλτιστης τιμής μιας ελεύθερης παραμέτρου ενός μοντέλου δεν είναι κάτι ασυνήθιστο στις μεθόδους μηχανικής μάθησης. Πολλοί σύνθετοι αλγόριθμοι μάθησης περιλαμβάνουν ελεύθερες παραμέτρους που πρέπει να εκτιμηθούν πειραματικά για το εκάστοτε σύνολο εκπαίδευσης.

Η δυϊκή μορφή του αλγορίθμου μεγίστου περιθωρίου, ο οποίος παρουσιάστηκε εδώ, αποτελεί μια αξιοσημείωτη εξαίρεση στα παραπάνω. Ο δυϊκός αυτός αλγόριθμος κατασκευάζει επίσης έναν ταξινομητή μεγίστου περιθωρίου, με τη διαφορά όμως ότι δεν περιλαμβάνει ελεύθερες παραμέτρους. Αυτός καλείται «***γραμμική μηχανή διανυσμάτων υποστήριξης***» (***linear support vector machine***) και θα αναπτυχθεί αναλυτικά στο Κεφ. 6.

ΚΕΦΑΛΑΙΟ 6

Μηχανές Διανυσμάτων Υποστήριξης

6.1 Εισαγωγή

Ο αλγόριθμος της «***μηχανής διανυσμάτων υποστήριξης***» (***support vector machine - SVM***) είναι στην ουσία η δυϊκή μορφή του αλγορίθμου μεγίστου περιθωρίου, ο οποίος αναπτύχθηκε στο Κεφ. 5. Η δυϊκή αυτή μορφή λαμβάνεται με εφαρμογή της θεωρίας βελτιστοποίησης κατά Lagrange στο πρόβλημα που αφορά την κατασκευή ενός ταξινομητή μεγίστου περιθωρίου και έχει ορισμένα ενδιαφέροντα επακόλουθα. Ένα από αυτά είναι ότι οι γραμμικοί ταξινομητές που βασίζονται στην SVM μπορούν εύκολα να επεκταθούν σε μη γραμμικούς, διευρύνοντας έτσι τρομακτικά την εφαρμοσιμότητα της μεθόδου. Στη καρδιά αυτής της γενίκευσης, από τους γραμμικούς στους μη γραμμικούς ταξινομητές, βρίσκεται η ιδέα των «***συναρτήσεων-πυρήνων***» (***kernel functions***). Με την εφαρμογή σε μία γραμμική SVM αυτού που συχνά αναφέρεται ως το «***τέχνασμα του πυρήνα***» (***kernel trick***), λαμβάνουμε ένα μη γραμμικό ταξινομητή. Είναι αξιοσημείωτο ότι οι μη γραμμικές SVM διατηρούν την αποδοτικότητα στην εξεύρεση γραμμικών επιφανειών απόφασης, αλλά παρέχουν επιπλέον τη δυνατότητα εφαρμογής του ταξινομητή επί συνόλων εκπαίδευσης μη γραμμικώς διαχωρίσιμων.

Προς το τέλος της παρουσίασης του αλγορίθμου θα γίνει αναφορά σε μια γενίκευσή του, η οποία επιτρέπει στον ταξινομητή να υποπίπτει σε σφάλματα ταξινόμησης των δεδομένων του συνόλου εκπαίδευσης, μέσω της εισαγωγής των λεγόμενων «***χαλαρών μεταβλητών***» (***slack variables***). Αυτό έχει ως σκοπό την αποφυγή της υπερπροσαρμογής, σε αντιστοιχία με την τεχνική της ομαλοποίησης, η οποία χρησιμοποιήθηκε σε προηγούμενα κεφάλαια κατά την ανάπτυξη των αλγορίθμων της λογιστικής παλινδρόμησης και των νευρωνικών δικτύων. Οι

ταξινομητές μεγίστου περιθωρίου που συμπεριλαμβάνουν χαλαρές μεταβλητές καλούνται «***ταξινομητές εύκαμπτου περιθωρίου***» (***soft margin classifiers***).

Η κλασσική θεωρία των μηχανών διανυσμάτων υποστήριξης αντιμετωπίζει μόνο προβλήματα δυαδικής ταξινόμησης οπότε η θεωρητική παρουσίαση που θα κάνουμε θα επικεντρωθεί μόνο σε αυτά. Άλλωστε οι τεχνικές επέκτασής της σε προβλήματα πολλαπλής ταξινόμησης βασίζεται σε ιδέες όπως η *one-vs-all classification* για την οποία έχει ήδη γίνει αναφορά στην Ενότητα 3.6. Για περισσότερα βλ. [1], [10], [21].

Αρχικά θα παρουσιαστούν κάποια γενικά στοιχεία της θεωρίας βελτιστοποίησης κατά Lagrange και του δυϊκού κατά Lagrange προβλήματος και στη συνέχεια θα γίνει εφαρμογή της θεωρίας αυτής για την παραγωγή του αλγορίθμου SVM. Το κεφάλαιο θα ολοκληρωθεί με την αναλυτική παρουσίαση πρακτικών εφαρμογών του αλγορίθμου, με χρήση των λογισμικών ***Octave***, ***R*** και ***WEKA***.

6.2 Το Δυϊκό κατά Lagrange Πρόβλημα

Η παραγωγή της δυϊκής μορφής ενός προβλήματος βελτιστοποίησης δίνει συχνά νέες ιδέες σχετικά με το προς επίλυση πρόβλημα οι οποίες μπορεί να οδηγήσουν σε νέες τεχνικές για την επίλυσή του, όπως θα διαπιστωθεί και στη συγκεκριμένη περίπτωση. Μια ιδιαιτέρως βολική τεχνική που χρησιμοποιείται συχνά, είναι η χρήση του «***δυϊκού κατά Lagrange***» (***Lagrangian dual***) προβλήματος.

Έστω ένα πρόβλημα βελτιστοποίησης της μορφής:

$$\min_{\bar{x}} \phi(\bar{x}), \tag{6.1}$$

έτσι ώστε

$$g_i(\bar{x}) \geq 0, \text{ με } i = 1,\ldots,l \tag{6.2}$$

για όλα τα $\bar{x} \in \mathbb{R}^n$. Εδώ υποθέτουμε ότι η αντικειμενική συνάρτηση ϕ είναι κυρτή και οι περιορισμοί g_i γραμμικοί. Η διατύπωση αυτή είναι όμοια με αυτή της (5.2) αν θεωρήσουμε ότι $g_i(\bar{x}) = h_i(\bar{x}) - c_i$. Συνήθως αυτή η διατύπωση αναφέρεται ως το *πρωταρχικό πρόβλημα βελτιστοποίησης*.

Στη συνέχεια δημιουργούμε ένα νέο πρόβλημα βελτιστοποίησης, το λεγόμενο *λαγκρανζιανό*, με βάση το πρωταρχικό μας πρόβλημα, ως εξής:

$$\max_{\bar{\alpha}} \min_{\bar{x}} L(\bar{\alpha}, \bar{x}) = \max_{\bar{\alpha}} \min_{\bar{x}} \left(\phi(\bar{x}) - \sum_{i=1}^{l} \alpha_i g_i(\bar{x}) \right), \tag{6.3}$$

έτσι ώστε

$$\alpha_i \geq 0, \text{ με } i = 1, \ldots, l \tag{6.4}$$

για όλα τα $\bar{x} \in \mathbb{R}^n$. Η νέα αντικειμενική συνάρτηση $L(\bar{\alpha}, \bar{x})$ καλείται *λαγκρανζιανή* και ενσωματώνει την αρχική αντικειμενική συνάρτηση ϕ μαζί με ένα γραμμικό συνδυασμό των περιορισμών g_i. Οι τιμές $\alpha_1, \ldots, \alpha_l$ (μία για τον κάθε περιορισμό) καλούνται πολλαπλασιαστές Lagrange και μπορεί στη συνέχεια να αναγράφονται με τη μορφή διανύσματος

$$\bar{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_l) \tag{6.5}$$

Η μεταβλητή $\bar{x}$ καλείται *πρωταρχική*, ενώ η $\bar{\alpha}$ *δυϊκή*.

Λύσεις του παραπάνω προβλήματος αποτελούν τα σημεία τα οποία ταυτόχρονα μεγιστοποιούν την τιμή της συνάρτησης $L(\bar{\alpha}, \bar{x})$ ως προς τη δυϊκή μεταβλητή $\bar{\alpha}$ και την ελαχιστοποιούν ως προς την πρωταρχική μεταβλητή $\bar{x}$. Αυτό σημαίνει ότι τα σημεία αυτά είναι «***σαγματικά σημεία***» (***saddle points***) του γραφήματος της $L(\bar{\alpha}, \bar{x})$. Δεδομένου ότι η πρωταρχική αντικειμενική συνάρτηση $\phi(\bar{x})$ είναι κυρτή και οι περιορισμοί $g_i(\bar{x})$ είναι γραμμικοί, υπάρχει ένα και μοναδικό σαγματικό σημείο. Καθώς το σημείο αυτό αποτελεί λύση, η L θα ελαχιστοποιείται ως προς $\bar{x}$ και η μερική της παράγωγος θα είναι μηδέν:

$$\frac{\partial L}{\partial \bar{x}} = \bar{0}. \tag{6.6}$$

Έστω $\bar{x}^*$ η τιμή του $\bar{x}$ στο σαγματικό σημείο της L. Υπολογίζοντας τη μερική παράγωγό της ως προς $\bar{x}$ στο σημείο αυτό, θα ισχύει

$$\frac{\partial L}{\partial \bar{x}}(\bar{\alpha}, \bar{x}^*) = \bar{0}. \tag{6.7}$$

Μια από τις ενδιαφέρουσες και κρίσιμες για την περίπτωσή μας, ιδιότητες της βελτιστοποίησης κατά Lagrange είναι ότι, κάτω από ορισμένες συνθήκες, μια λύση του λαγκρανζιανού προβλήματος αποτελεί επίσης λύση για το πρωταρχικό. Για να φανεί καλύτερα αυτό, έστω $\bar{\alpha}^*$ και $\bar{x}^*$ μία λύση του λαγκρανζιανού, τέτοια ώστε

$$\max_{\bar{\alpha}} \min_{\bar{x}} L(\bar{\alpha}, \bar{x}) = L(\bar{\alpha}^*, \bar{x}^*) = \phi(\bar{x}^*) - \sum_{i=1}^{l} \alpha_i^* g_i(\bar{x}^*). \tag{6.8}$$

Τότε το $\bar{x}^*$ θα αποτελεί λύση της πρωταρχικής αντικειμενικής συνάρτησης αν και μόνο αν ισχύουν τα παρακάτω:

$$\frac{\partial L}{\partial \bar{x}}\left(\bar{\alpha}^*, \bar{x}^*\right) = \bar{0}, \tag{6.9}$$

$$\alpha_i^* g_i\left(\bar{x}^*\right) = 0, \tag{6.10}$$

$$g_i\left(\bar{x}^*\right) \geq 0, \tag{6.11}$$

$$\alpha_i^* \geq 0 \tag{6.12}$$

για $i = 1, \ldots, l$. Η πλέον ενδιαφέρουσα από τις παραπάνω σχέσεις είναι η *συνθήκη συμπληρωματικότητας* (6.10). Το ότι απαιτείται αυτή να ισχύει μπορεί να φανεί από τη σχέση (6.8) όπου ο όρος του αθροίσματος θα πρέπει να μηδενιστεί, ώστε η $L(\bar{\alpha}^*, \bar{x}^*) = \phi(\bar{x}^*)$. Καθώς όμως πρόκειται για άθροισμα γινομένων μη αρνητικών ποσοτήτων είναι σαφές ότι, προκειμένου να μηδενίζεται το άθροισμα, κάθε όρος του θα πρέπει να ισούται με μηδέν[1]. Από τις άλλες δύο σχέσεις, η (6.9) εξασφαλίζει ότι η τιμή $\bar{x}^*$ βρίσκεται στο σαγματικό σημείο, ενώ οι (6.11) και (6.12) είναι οι περιορισμοί του πρωταρχικού και του λαγκρανζιανού προβλήματος αντίστοιχα. Οι τέσσερις αυτές συνθήκες είναι γνωστές ως συνθήκες των ***Karush-Kuhn-Tucker*** (***KKT conditions***).

Η επίλυση του λαγκρανζιανού προβλήματος βελτιστοποίησης, στην περίπτωση που η πρωταρχική αντικειμενική συνάρτηση είναι κυρτή, μπορεί να απλοποιηθεί χρησιμοποιώντας το γεγονός ότι το βέλτιστο $\bar{x}^*$ πρέπει να βρίσκεται στο μοναδικό σαγματικό σημείο της λαγκρανζιανής. Κατά συνέπεια, υπολογίζοντας την έκφραση του $\bar{x}^*$ ως προς $\bar{\alpha}$ από την (6.7) , μπορούμε να αναδιατυπώσουμε το αρχικό μας πρόβλημα ως προς τη δυϊκή μεταβλητή μόνο. Θέτοντας $L(\bar{\alpha}, \bar{x}^*) = \phi'(\bar{\alpha})$ έχουμε:

$$\max_{\bar{\alpha}} \phi'\left(\bar{\alpha}\right), \tag{6.13}$$

έτσι ώστε

$$\alpha_i \geq 0 \text{, για } i = 1, \ldots, l. \tag{6.14}$$

Η συνάρτηση ϕ' καλείται «***δυϊκή κατά Lagrange***» (***Lagrangian dual***) και μέσω αυτής μπορεί να επιλυθεί το πρωταρχικό πρόβλημα βελτιστοποίησης, καθώς:

[1] Η συνθήκη αυτή είναι γνωστή ως ***complementary slackness***, που σημαίνει ότι ο κάθε πολλαπλασιαστής και ο αντίστοιχος περιορισμός δεν μπορούν ταυτόχρονα να λαμβάνουν μη μηδενικές τιμές. Με άλλα λόγια τα διανύσματα των πολλαπλασιαστών και των περιορισμών έχουν *συμπληρωματικά μοτίβα σποραδικότητας* [6], [5].

$$\max_{\bar{\alpha}} \phi'(\bar{\alpha}) = \phi'(\bar{\alpha}^*) = L(\bar{\alpha}^*, \bar{x}^*) = \phi(\bar{x}^*), \tag{6.15}$$

όπου τα $\bar{x}^*$ και $\bar{\alpha}^*$ πρέπει να ικανοποιούν τις συνθήκες ΚΚΤ.

6.3 Το Δυϊκό Πρόβλημα Βελτιστοποίησης Μεγίστου Περιθωρίου

Όπως έχει ήδη αναφερθεί, η μηχανή διανυσμάτων υποστήριξης αποτελεί το δυϊκό πρόβλημα των ταξινομητών μεγίστου περιθωρίου. Στη συνέχεια θα διατυπωθεί το πρόβλημα αυτό εφαρμόζοντας την τεχνική της δυϊκής κατά Lagrange συνάρτησης στο πρωταρχικό πρόβλημα των ταξινομητών μεγίστου περιθωρίου. Για δοθέν γραμμικώς διαχωρίσιμο σύνολο εκπαίδευσης της μορφής

$$D = \{(\bar{x}_1, y_1), (\bar{x}_2, y_2), \ldots, (\bar{x}_l, y_l)\} \subseteq \mathbb{R}^n \times \{+1, -1\} \tag{6.16}$$

σύμφωνα με την Πρόταση 5.1, το πρόβλημα βελτιστοποίησης μπορεί να γραφεί:

$$\min_{\bar{w},b} \phi(\bar{w}, b) = \min_{\bar{w},b} \frac{1}{2} \bar{w} \bullet \bar{w} \tag{6.17}$$

υπό τους περιορισμούς

$$g_i(\bar{w}, b) = y_i(\bar{w} \bullet \bar{x}_i - b) - 1 \geq 0 \tag{6.18}$$

για $i = 1, \ldots, l$. Κατασκευάζουμε την αντίστοιχη λαγκρανζιανή ως εξής

$$\begin{aligned} L(\bar{\alpha}, \bar{w}, b) &= \phi(\bar{w}, b) - \sum_{i=1}^{l} \alpha_i g_i(\bar{w}, b) \\ &= \frac{1}{2} \bar{w} \bullet \bar{w} - \sum_{i=1}^{l} \alpha_i \left(y_i(\bar{w} \bullet \bar{x}_i - b) - 1 \right) \\ &= \frac{1}{2} \bar{w} \bullet \bar{w} - \sum_{i=1}^{l} \alpha_i y_i \bar{w} \bullet \bar{x}_i + b \sum_{i=1}^{l} \alpha_i y_i + \sum_{i=1}^{l} \alpha_i \end{aligned} \tag{6.19}$$

και το κατά Lagrange πρόβλημα βελτιστοποίησης γίνεται

$$\max_{\bar{\alpha}} \min_{\bar{w},b} L(\bar{\alpha}, \bar{w}, b), \tag{6.20}$$

υπό τους περιορισμούς

$$\alpha_i \geq 0, \text{ για } i = 1, \ldots, l \tag{6.21}$$

Έστω $\bar{\alpha}^*, \bar{w}^*$ και b^* η λύση του προβλήματος αυτού, ώστε

$$\max_{\bar{\alpha}} \min_{\bar{w},b} L(\bar{\alpha}, \bar{w}, b) = L(\bar{\alpha}^*, \bar{w}^*, b^*). \tag{6.22}$$

Τότε, αφού η ϕ είναι κυρτή και οι περιορισμοί g_i γραμμικοί, η λύση $\bar{\alpha}^*, \bar{w}^*$ και b^* θα ικανοποιεί τις παρακάτω συνθήκες ΚΚΤ:

$$\frac{\partial L}{\partial \bar{w}}\left(\bar{\alpha}^*, \bar{w}^*, b^*\right) = \bar{0}, \tag{6.23}$$

$$\frac{\partial L}{\partial b}\left(\bar{\alpha}^*, \bar{w}^*, b^*\right) = 0, \tag{6.24}$$

$$\alpha_i^*\left(y_i\left(\bar{w}^* \bullet \bar{x}_i - b^*\right) - 1\right) = 0, \tag{6.25}$$

$$y_i\left(\bar{w}^* \bullet \bar{x}_i - b^*\right) - 1 \geq 0, \tag{6.26}$$

$$\alpha_i^* \geq 0 \tag{6.27}$$

για $i = 1, \ldots, l$. Οι σχέσεις (6.23) και (6.24) διασφαλίζουν ότι τα $\bar{w}^*$ και b^* βρίσκονται στο σαγματικό σημείο της λαγκρανζιανής ενώ η συνθήκη συμπληρωματικότητας (6.25) υποδηλώνει ότι τα $\bar{w}^*$ και b^* αποτελούν επίσης λύσεις του πρωταρχικού προβλήματος

$$\max_{\bar{\alpha}} \min_{\bar{w}, b} L\left(\bar{\alpha}, \bar{w}, b\right) = L\left(\bar{\alpha}^*, \bar{w}^*, b^*\right) = \phi\left(\bar{w}^*, b^*\right) \tag{6.28}$$

και κατά συνέπεια μπορεί να χρησιμοποιηθεί η βελτιστοποίηση κατά Lagrange για τον υπολογισμό του μεγίστου περιθωρίου. Τέλος, οι σχέσεις (6.26) και (6.27) διασφαλίζουν ότι η λύση βρίσκεται στα αντίστοιχα χωρία των εφικτών λύσεων.

Για την επίλυση του προβλήματος θα κατασκευάσουμε τη δυϊκή κατά Lagrange συνάρτηση. Αρχικά θα υπολογιστούν οι εκφράσεις των σημείων $\bar{w}^*$ και b^*, τα οποία θα πρέπει να βρίσκονται στο σαγματικό σημείο της λαγκρανζιανής. Εφαρμόζοντας την πρώτη συνθήκη ΚΚΤ (6.23), υπολογίζουμε τη μερική παράγωγο της L ως προς την πρωταρχική μεταβλητή $\bar{w}$ στο σαγματικό σημείο $\bar{w}^*$ και τη θέτουμε ίση με μηδέν:

$$\frac{\partial L}{\partial \bar{w}}\left(\bar{\alpha}, \bar{w}^*, b\right) = \bar{w}^* - \sum_{i=1}^{l} \alpha_i y_i \bar{x}_i = \bar{0}, \tag{6.29}$$

$$\bar{w}^* = \sum_{i=1}^{l} \alpha_i y_i \bar{x}_i. \tag{6.30}$$

Αντίστοιχα εργαζόμαστε και για τη δεύτερη πρωταρχική μεταβλητή b:

$$\frac{\partial L}{\partial b}\left(\bar{\alpha}, \bar{w}, b^*\right) = \sum_{i=1}^{l} \alpha_i y_i = 0. \tag{6.31}$$

Έχει ενδιαφέρον ότι στην περίπτωση αυτή δε λαμβάνουμε μια σχέση για το b^* αλλά έναν περιορισμό, ότι δηλαδή στο σαγματικό σημείο b^* θα πρέπει να ισχύει $\sum_{i=1}^{l} \alpha_i y_i = 0$. Παρόλα

αυτά η τιμή του b^* μπορεί να ανακτηθεί μέσω των δεδομένων του συνόλου εκπαίδευσης και του διανύσματος $\bar{w}^*$. Όπως έχει αναφερθεί για την περίπτωση των ταξινομητών μεγίστου περιθωρίου, το υπερεπίπεδο στήριξης για την κλάση +1 θα πρέπει να διέρχεται από κάποιο σημείο $(\bar{x}_p, +1) \in D$ στο όριο της κλάσης +1 (βλ. Σχήμα 5-4). Το γεγονός αυτό μας επιτρέπει να υπολογίσουμε τη μετάθεση b^+ αυτού του υπερεπιπέδου.

$$b^+ = \bar{w}^* \bullet \bar{x}_p. \tag{6.32}$$

Πιο συγκεκριμένα, αναγνωρίζοντας το γεγονός ότι για δοθέν $\bar{w}^*$, το πλησιέστερο στο όριο σημείο $\bar{x}_p$ θα έχει και τη μικρότερη μετάθεση, μπορούμε να υπολογίσουμε το b^+ ως εξής:

$$b^+ = \min\left\{\bar{w}^* \bullet \bar{x} \,|\, (\bar{x}, y) \in D,\, y = +1\right\}. \tag{6.33}$$

Εφαρμόζοντας παρόμοια συλλογιστική για την άλλη κλάση λαμβάνουμε

$$b^- = \max\left\{\bar{w}^* \bullet \bar{x} \,|\, (\bar{x}, y) \in D,\, y = -1\right\}. \tag{6.34}$$

Καθώς η επιφάνεια απόφασης βρίσκεται ακριβώς στο μέσον της απόστασης μεταξύ των δύο φερόντων υπερεπιπέδων θα ισχύει:

$$b^* = \frac{b^+ + b^-}{2}. \tag{6.35}$$

Είμαστε έτοιμοι πια να διαμορφώσουμε τη δυϊκή κατά Lagrange συνάρτηση. Αντικαθιστώντας την (6.30) στην (6.19) και εφαρμόζοντας τον περιορισμό (6.31) λαμβάνουμε:

$$\phi'(\bar{\alpha}) = L(\bar{\alpha}, \bar{w}^*, b^*) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \bar{x}_i \bullet \bar{x}_j. \tag{6.36}$$

Συνεπώς, το δυϊκό πρόβλημα βελτιστοποίησης για ταξινομητές μεγίστου περιθωρίου διατυπώνεται με την ακόλουθη πρόταση:

***Πρόταση 6.1* (*Δυϊκό κατά Lagrange πρόβλημα Μεγίστου Περιθωρίου*)** *Δοθέντος προβλήματος βελτιστοποίησης μεγίστου περιθωρίου, όπως αυτό της Πρότασης 5.1, το κατά Lagrange δυϊκό του πρόβλημα λαμβάνει τη μορφή:*

$$\max_{\bar{\alpha}} \phi'(\bar{\alpha}) = \max_{\bar{\alpha}} \left(\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \bar{x}_i \bullet \bar{x}_j \right), \tag{6.37}$$

υπό τους περιορισμούς

$$\sum_{i=1}^{l} \alpha_i y_i = 0, \tag{6.38}$$

$$\alpha_i \geq 0\,,\ \mu\varepsilon\ i = 1,\ldots,l \tag{6.39}$$

Ενδιαφέρον παρουσιάζουν οι επιπτώσεις της συνθήκης συμπληρωματικότητας (6.25) για δοθείσα λύση $\bar{\alpha}^*$ του δυϊκού προβλήματος βελτιστοποίησης. Η συνθήκη αυτή ικανοποιείται για κάθε $i = 1,\ldots,l$ μόνο εάν είτε το $\alpha_i^* = 0,$ είτε το $y_i(\bar{w}^* \bullet \bar{x}_i - b^*) - 1 = 0$. Έστω η περίπτωση όπου $\alpha_j^* > 0$ για κάποιο σημείο $(\bar{x}_j, y_j) \in D$. Προκειμένου η συνθήκη να ισχύει, θα πρέπει $y_j(\bar{w}^* \bullet \bar{x}_j - b^*) - 1 = 0$, ή

$$\bar{w}^* \bullet \bar{x}_j = b^* + 1 \quad \text{για } y_j = +1, \tag{6.40}$$

$$\bar{w}^* \bullet \bar{x}_j = b^* - 1 \quad \text{για } y_j = -1. \tag{6.41}$$

Οι παραπάνω εξισώσεις όμως είναι αυτές των φερόντων υπερεπιπέδων στην περίπτωση της βέλτιστης επιφάνειας απόφασης $\bar{w}^* \bullet \bar{x} = b^*$ (βλ. εξισώσεις (5.9) και (5.10) για $k = 1$). Αυτό σημαίνει ότι το συγκεκριμένο σημείο $(\bar{x}_j, y_j)$ με μη μηδενικό πολλαπλασιαστή Lagrange $\alpha_j^* > 0$ βρίσκεται επί ενός εκ των δύο φερόντων υπερεπιπέδων, ανάλογα με την τιμή του χαρακτηρισμού του, y_j. Το σημείο αυτό αποτελεί περιορισμό για το περιθώριο, υπό την έννοια ότι το υπερεπίπεδο στήριξης δεν μπορεί να μετακινηθεί πέρα από αυτό.

Σημεία με μη μηδενική τιμή του πολλαπλασιαστή Lagrange καλούνται *διανύσματα υποστήριξης* και μια προσεκτική εξέταση των εξισώσεων (6.30) και (6.35) οδηγεί στο συμπέρασμα ότι μόνο τα διανύσματα υποστήριξης συνεισφέρουν στη λύση του δυϊκού προβλήματος βελτιστοποίησης μεγίστου περιθωρίου.

Έστω τώρα $\alpha_j^* = 0$ για κάποιο σημείο $(\bar{x}_j, y_j) \in D$. Αυτό σημαίνει ότι το σημείο $\bar{x}_j$ δε βρίσκεται στην περιοχή του ορίου της κλάσης, καθώς τώρα έχουμε ότι $y_j(\bar{w}^* \bullet \bar{x}_j - b^*) - 1 > 0$ ή αλλιώς:

$$\bar{w}^* \bullet \bar{x}_j > b^* + 1 \quad \text{για } y_j = +1, \tag{6.42}$$

$$\bar{w}^* \bullet \bar{x}_j < b^* - 1 \quad \text{για } y_j = -1. \tag{6.43}$$

Με άλλα λόγια, σημεία με μηδενικό πολλαπλασιαστή Lagrange δεν περιορίζουν το μέγεθος του περιθωρίου. Υπενθυμίζεται ότι το πρωταρχικό πρόβλημα βελτιστοποίησης μεγίστου

περιθωρίου βρίσκει τα πλέον απομακρυσμένα μεταξύ τους υπερεπίπεδα στήριξης, αυτά δηλαδή με το μέγιστο μεταξύ τους περιθώριο. Τα σημεία του συνόλου εκπαίδευσης τα οποία περιόριζαν το μέγεθος αυτού του περιθωρίου καλούνταν διανύσματα υποστήριξης. Μπορούμε λοιπόν να κάνουμε τον ακόλουθο συσχετισμό:

Το πρωταρχικό πρόβλημα βελτιστοποίησης μεγίστου περιθωρίου υπολογίζει τα υπερεπίπεδα στήριξης, το περιθώριο μεταξύ των οποίων περιορίζεται από τα διανύσματα υποστήριξης. Το δυϊκό του πρόβλημα υπολογίζει τα διανύσματα υποστήριξης, τα οποία περιορίζουν το μέγεθος του περιθωρίου μεταξύ των υπερεπιπέδων στήριξης.

Η αναγνώριση του γεγονότος ότι μόνο τα διανύσματα υποστήριξης συνεισφέρουν στη δυϊκή λύση μας επιτρέπει να εκφράσουμε την τιμή του b^* με πιο κομψό τρόπο. Αντί να αναζητούμε τα σημεία κάθε κλάσης που βρίσκονται εγγύτερα στην επιφάνεια απόφασης, όπως κάναμε προηγουμένως, τώρα γνωρίζουμε ποια είναι τα σημεία αυτά που αποτελούν όριο για τα υπερεπίπεδα στήριξης: αυτά με μη μηδενικούς πολλαπλασιαστές Lagrange. Αν επιλέξουμε ένα διάνυσμα υποστήριξης από το σύνολο εκπαίδευσης, έστω $(\bar{x}_{sv+}, y_{sv+})$ με $y_{sv+} = +1$, τότε συνδυάζοντας την (6.40) με την (6.30), μπορούμε να υπολογίσουμε το b^* ως εξής:

$$b^* = \bar{w}^* \bullet \bar{x}_{sv+} - 1 = \sum_{i=1}^{l} \alpha_i^* y_i \bar{x}_i \bullet \bar{x}_{sv+} - 1 \tag{6.44}$$

6.3.1 Η δυϊκή συνάρτηση απόφασης

Σύμφωνα με τα όσα έχουν αναφερθεί σε προηγούμενα κεφάλαια, οι συναρτήσεις απόφασης γραμμικών ταξινομητών βασίζονται σε γραμμικές επιφάνειες απόφασης και επιστρέφουν το χαρακτηρισμό $+1$ για σημεία που βρίσκονται πάνω από την επιφάνεια απόφασης και το χαρακτηρισμό -1 για σημεία κάτω από αυτή. Έστω $\bar{\alpha}^*, \bar{w}^*$ και b^* μία λύση του λαγκρανζιανού προβλήματος βελτιστοποίησης. Τότε η βέλτιστη επιφάνεια απόφασης ορίζεται ως

$$\bar{w}^* \bullet \bar{x} = b^*. \tag{6.45}$$

Αυτή μας οδηγεί στην παρακάτω συνάρτηση απόφασης μεγίστου περιθωρίου:

$$\hat{f}(\bar{x}) = \operatorname{sgn}\left(\bar{w}^* \bullet \bar{x} - b^*\right). \tag{6.46}$$

Αντικαθιστώντας τα $\overline{w}^*$ και b^* με βάση τις σχέσεις (6.30) και (6.44), η συνάρτηση απόφασης λαμβάνει τη μορφή:

$$\hat{f}\left(\overline{x}\right)=\operatorname{sgn}\left(\sum_{i=1}^{l}\alpha_i^* y_i \overline{x}_i \bullet \overline{x}-\sum_{i=1}^{l}\alpha_i^* y_i \overline{x}_i \bullet \overline{x}_{sv+}+1\right). \tag{6.47}$$

Δηλαδή, ο δυϊκός ταξινομητής μεγίστου περιθωρίου καθορίζεται πλήρως από τα διανύσματα υποστήριξης, ή με άλλα λόγια από τα σημεία που αποτελούν όριο για το μέγεθος του περιθωρίου μεταξύ των φερόντων υπερεπιπέδων. Λόγω αυτού του χαρακτηριστικού, η συνάρτηση απόφασης του δυϊκού ταξινομητή μεγίστου περιθωρίου καλείται επίσης ***μηχανή διανυσμάτων υποστήριξης***. Επιπλέον, θεωρείται ως γραμμική μηχανή διανυσμάτων υποστήριξης, δεδομένου ότι βασίζεται σε μια γραμμική επιφάνεια απόφασης.

6.4 Γραμμικές Μηχανές Διανυσμάτων Υποστήριξης

Το πρόβλημα δυαδικής ταξινόμησης υπό το πρίσμα των μηχανών διανυσμάτων υποστήριξης διατυπώνεται ως εξής:

Έστω:

- ότι ο χώρος με εσωτερικό γινόμενο $\mathbb{R}^n$ αποτελεί τον πληθυσμό δεδομένων, με αντικείμενα τα διανύσματα $\overline{x}\in\mathbb{R}^n$
- μία συνάρτηση-στόχος $f:\mathbb{R}^n\to\{+1,-1\}$
- ένα χαρακτηρισμένο και γραμμικώς διαχωρίσιμο σύνολο εκπαίδευσης

  $$D=\left\{\left(\overline{x}_1,y_1\right),\left(\overline{x}_2,y_2\right),\ldots,\left(\overline{x}_l,y_l\right)\right\}\subseteq\mathbb{R}^n\times\{+1,-1\},$$

  όπου $y_i=f\left(\overline{x}_i\right)$.

Να βρεθεί συνάρτηση $\hat{f}:\mathbb{R}^n\to\{+1,-1\}$ κάνοντας χρήση του D, τέτοια ώστε:

$$\hat{f}(x)\cong f(x) \tag{6.48}$$

για κάθε $\overline{x}\in\mathbb{R}^n$. Χρησιμοποιώντας ως μοντέλο τη γραμμική μηχανή διανυσμάτων υποστήριξης από τη σχέση (6.47) έχουμε,

$$\hat{f}\left(\overline{x}\right)=\operatorname{sgn}\left(\sum_{i=1}^{l}\alpha_i^* y_i \overline{x}_i \bullet \overline{x}-\sum_{i=1}^{l}\alpha_i^* y_i \overline{x}_i \bullet \overline{x}_{sv+}+1\right), \tag{6.49}$$

όπου τα σημεία $(\bar{x}_i, y_i) \in D$ είναι διανύσματα υποστήριξης αν οι αντίστοιχοι πολλαπλασιαστές Lagrange είναι μη μηδενικοί, δηλαδή $\alpha_i^* > 0$. Στην παραπάνω σχέση χρησιμοποιούμε ένα εκ των διανυσμάτων υποστήριξης που έχουν προκύψει,

$$(\bar{x}_{sv+}, +1) \in \left\{ (\bar{x}_i, +1) \,|\, (\bar{x}_i, +1) \in D, \alpha_i^* > 0 \right\} \tag{6.50}$$

προκειμένου να υπολογίσουμε τον όρο μετάθεσης. Η εκπαίδευση του μοντέλου γίνεται με τη δυϊκή κατά Lagrange βελτιστοποίηση μεγίστου περιθωρίου, σύμφωνα με την Πρόταση 6.1:

$$\bar{\alpha}^* = \arg\max_{\bar{\alpha}} \left(\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \bar{x}_i \bullet \bar{x}_j \right), \tag{6.51}$$

υπό τους περιορισμούς

$$\sum_{i=1}^{l} \alpha_i y_i = 0, \tag{6.52}$$

$$\alpha_i \geq 0, \text{ με } i = 1, \ldots, l \tag{6.53}$$

Στην περίπτωση λοιπόν που το σύνολο εκπαίδευσης είναι γραμμικώς διαχωρίσιμο, το πρόβλημα ταξινόμησης μπορεί να αντιμετωπιστεί με τη χρήση γραμμικών μηχανών διανυσμάτων υποστήριξης. Στην πράξη βέβαια, τα γραμμικώς διαχωρίσιμα σύνολα δεδομένων είναι πολύ λίγα.

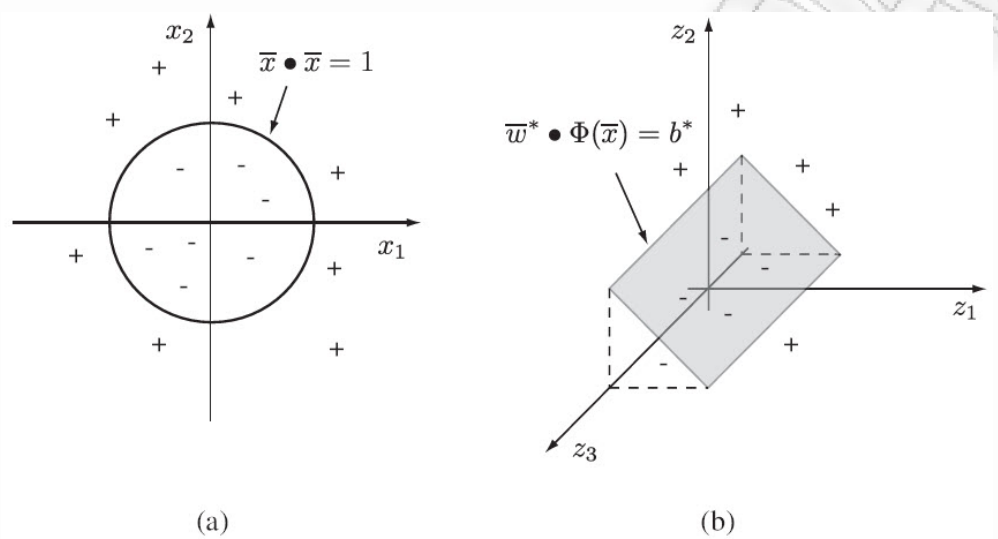
6.5 Μη Γραμμικές Μηχανές Διανυσμάτων Υποστήριξης

Αυτό που καθιστά τόσο αξιόλογο τον αλγόριθμο SVM είναι το ότι το βασικό αυτό γραμμικό πλαίσιο που αναπτύχθηκε στις προηγούμενες παραγράφους μπορεί εύκολα να επεκταθεί για την περίπτωση όπου το σύνολο εκπαίδευσης δεν είναι γραμμικώς διαχωρίσιμο. Η βασική ιδέα πίσω από αυτή την επέκταση είναι η μετατροπή του χώρου εισόδου, στον οποίο τα δεδομένα δεν είναι γραμμικώς διαχωρίσιμα, σε ένα χώρο μεγαλύτερης διάστασης ο οποίος καλείται «***χώρος χαρακτηριστικών***» (***feature space***), όπου τα δεδομένα θα είναι γραμμικώς διαχωρίσιμα. Αξίζει να σημειωθεί ότι αν η μετατροπή αυτή επιλεγεί προσεκτικά, όλοι οι υπολογισμοί που συνδέονται με το χώρο των χαρακτηριστικών μπορούν να γίνουν στο χώρο εισόδου. Δηλαδή, αν και ο χώρος εισόδου μετασχηματίζεται έτσι ώστε να καταστούν γραμμικώς διαχωρίσιμα τα δεδομένα, δεν απαιτείται η δαπάνη του υπολογιστικού κόστους που συνεπάγεται ο μετασχηματισμός αυτός. Οι συναρτήσεις που σχετίζονται με τις μετατροπές αυτές καλούνται «συναρτήσεις-πυρήνες» και η διαδικασία της χρήσης των

συναρτήσεων αυτών, για το πέρασμα από μια γραμμική σε μια μη γραμμική SVM, αναφέρεται ως το «τέχνασμα του πυρήνα». Οι πυρήνες μπορούν να ερμηνευθούν ως μέτρα ομοιότητας (similarity) μεταξύ δύο διανυσμάτων στο χώρο εισόδου.

Σχήμα 6-1

Απεικόνιση μη γραμμικού συνόλου με $\overline{x} \in \mathbb{R}^2$ (a), σε χώρο χαρακτηριστικών με $\overline{x} \in \mathbb{R}^3$ (b)



Ας θεωρήσουμε το ακόλουθο παράδειγμα. Το σύνολο δεδομένων που απεικονίζεται στο Σχήμα 6-1(a) ανήκει στο δισδιάστατο χώρο με εσωτερικό γινόμενο, $\mathbb{R}^2$ και είναι προφανές ότι δεν υφίσταται γραμμική επιφάνεια απόφασης της μορφής

$$\overline{w} \bullet \overline{x} = b \tag{6.54}$$

η οποία μπορεί να διαχωρίσει, χωρίς σφάλμα, τα αντικείμενα των δύο κλάσεων. Αντιθέτως, η μη γραμμική επιφάνεια απόφασης

$$\overline{x} \bullet \overline{x} = 1 \tag{6.55}$$

με $\overline{x} \in \mathbb{R}^2$, όπως φαίνεται στο Σχήμα 6-1(a), μπορεί. Αντί όμως για την κατασκευή μιας συνάρτησης απόφασης που θα βασίζεται σε μία επιφάνεια απόφασης του χώρου εισόδου, ας θεωρήσουμε μια συνάρτηση απόφασης η οποία, αφού αρχικά απεικονίσει τα σημεία $\overline{x} \in \mathbb{R}^2$ σε κάποιο χώρο (εφοδιασμένο με εσωτερικό γινόμενο) μεγαλύτερης διάστασης, έστω στον $\mathbb{R}^3$, στη συνέχεια θα χρησιμοποιεί μια επιφάνεια απόφασης αυτού του χώρου προκειμένου να ταξινομεί τα σημεία

$$\hat{f}\left(\overline{x}\right) = \operatorname{sgn}\left(\overline{w} \bullet \Phi\left(\overline{x}\right) - b\right) \tag{6.56}$$

όπου η απεικόνιση από το χώρο εισόδου στο χώρο χαρακτηριστικών $\Phi : \mathbb{R}^2 \to \mathbb{R}^3$ ορίζεται:

$$\Phi(\bar{x}) = \Phi(x_1, x_2) = \left(x_1^2, x_2^2, \sqrt{2}x_1x_2\right) = (z_1, z_2, z_3) = \bar{z} \tag{6.57}$$

με $\bar{x} \in \mathbb{R}^2$ και $\bar{z} \in \mathbb{R}^3$. Μέσω της παραπάνω συνάρτησης κάθε σημείο της μη γραμμικής επιφάνειας απόφασης (6.55) στο χώρο εισόδου απεικονίζεται σε ένα επίπεδο στο χώρο χαρακτηριστικών της μορφής:

$$\bar{w}^* \bullet \Phi(\bar{x}) = b^* \tag{6.58}$$

με $\bar{w}^* = (w_1^*, w_2^*, w_3^*) = (1,1,0)$ και $b^* = 1$. Επιπρόσθετα, όλα τα σημεία του χώρου εισόδου με χαρακτηρισμό $+1$ θα απεικονίζεται σε σημεία πάνω από το επίπεδο αυτό, ενώ όλα τα σημεία του χώρου εισόδου με χαρακτηρισμό -1 θα απεικονίζεται σε σημεία κάτω από αυτό. Το γεγονός ότι το επίπεδο της (6.58) διαχωρίζει τις κλάσεις στο χώρο χαρακτηριστικών σημαίνει ότι αποτελεί μια γραμμική επιφάνεια απόφασης. Αυτό απεικονίζεται στο Σχήμα 6-1(b). Κατά συνέπεια η απεικόνιση Φ μετατρέπει το μη γραμμικό πρόβλημα απόφασης του χώρου εισόδου σε γραμμικό στο χώρο χαρακτηριστικών.

Με δεδομένη την επιφάνεια απόφασης (6.58) του χώρου χαρακτηριστικών μπορούμε να ορίσουμε τη συνάρτηση απόφασης:

$$\hat{f}(\bar{x}) = \operatorname{sgn}\left(\bar{w}^* \bullet \Phi(\bar{x}) - b^*\right). \tag{6.59}$$

Η συνάρτηση αυτή πρώτα απεικονίζει το κάθε σημείο του χώρου εισόδου $\bar{x} \in \mathbb{R}^2$ σε ένα σημείο του χώρου χαρακτηριστικών $\bar{z} = \Phi(\bar{x}) \in \mathbb{R}^3$ και στη συνέχεια χρησιμοποιεί τη γραμμική επιφάνεια απόφασης του χώρου χαρακτηριστικών για να υπολογίσει την κλάση του σημείου. Χρησιμοποιώντας τη σχέση (6.57), η συνάρτηση αυτή μπορεί να γραφτεί ως:

$$\begin{aligned}\hat{f}(\bar{x}) &= \operatorname{sgn}\left(\bar{w}^* \bullet \Phi(\bar{x}) - b^*\right) = \operatorname{sgn}\left(w_1^* x_1^2 + w_2^* x_2^2 + w_3^* \sqrt{2} x_1 x_2 - b^*\right) \\ &= \operatorname{sgn}\left(\bar{w}^* \bullet \bar{z} - b^*\right) = \operatorname{sgn}\left(\sum_{i=1}^{3} w_i^* z_i - b^*\right).\end{aligned} \tag{6.60}$$

Από την παραπάνω σχέση γίνεται εμφανές ότι η πολυπλοκότητα της συνάρτησης απόφασης σχετίζεται ευθέως με τη διάσταση του χώρου χαρακτηριστικών. Συνεπώς, πολυπλοκότερες μη γραμμικές επιφάνειες απόφασης στο χώρο εισόδου θα απαιτούν όλο και μεγαλύτερης διάστασης χώρους χαρακτηριστικών, προκειμένου να καταστεί εφικτή η κατασκευή γραμμικών επιφανειών απόφασης, ενώ οι αντίστοιχες εκφράσεις των συναρτήσεων απόφασης

$$\hat{f}(\bar{x}) = \operatorname{sgn}\left(\sum_{i=1}^{d} w_i z_i - b\right) \tag{6.61}$$

θα αυξάνουν σε πολυπλοκότητα ανάλογα με τον αριθμό των διαστάσεων d του χώρου χαρακτηριστικών.

6.5.1 Το τέχνασμα του πυρήνα

Η δυϊκή μορφή του κάθετου διανύσματος $\overline{w}^*$ στη συνάρτηση απόφασης (6.59), σύμφωνα με τη σχέση (6.30), θα είναι

$$\overline{w}^* = \sum_{i=1}^{l} \alpha_i^* y_i \Phi(\overline{x}_i) \tag{6.62}$$

με $\Phi : \mathbb{R}^2 \to \mathbb{R}^3$, όπως ορίστηκε στην (6.57) και με το l να δηλώνει τον αριθμό των δεδομένων εκπαίδευσης στο χώρο εισόδου. Οι τιμές α_i^* αντιπροσωπεύουν τους κατάλληλους πολλαπλασιαστές Lagrange της δυϊκής αυτής διατύπωσης. Παρατηρήστε ότι ο μετασχηματισμός $\Phi(\overline{x}_i)$ των δεδομένων εκπαίδευσης είναι απαραίτητος καθώς το $\overline{w}^*$ είναι το κάθετο διάνυσμα στο χώρο χαρακτηριστικών. Αντικαθιστώντας το $\overline{w}^*$ στη συνάρτηση απόφασης, σύμφωνα με την παραπάνω σχέση λαμβάνουμε

$$\begin{aligned} \hat{f}(\overline{x}) &= \operatorname{sgn}\left(\overline{w}^* \bullet \Phi(\overline{x}) - b^*\right) \\ &= \operatorname{sgn}\left(\sum_{i=1}^{l} \alpha_i^* y_i \Phi(\overline{x}_i) \bullet \Phi(\overline{x}) - b^*\right). \end{aligned} \tag{6.63}$$

Με τον τρόπο όμως που ορίστηκε η Φ, η μορφή που λαμβάνει ένα εσωτερικό γινόμενο της μορφής $\Phi(\overline{x}) \bullet \Phi(\overline{y})$ είναι η εξής

$$\begin{aligned} \Phi(\overline{x}) \bullet \Phi(\overline{y}) &= \left(x_1^2, x_2^2, \sqrt{2}x_1x_2\right) \bullet \left(y_1^2, y_2^2, \sqrt{2}y_1y_2\right) \\ &= x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2 = \left(x_1y_1 + x_2y_2\right)\left(x_1y_1 + x_2y_2\right) \\ &= \left(\overline{x} \bullet \overline{y}\right)\left(\overline{x} \bullet \overline{y}\right) = \left(\overline{x} \bullet \overline{y}\right)^2. \end{aligned}$$

Κατά συνέπεια, η σχέση (6.63) γράφεται:

$$\hat{f}(\overline{x}) = \operatorname{sgn}\left(\sum_{i=1}^{l} \alpha_i^* y_i \left(\overline{x}_i \bullet \overline{x}\right)^2 - b^*\right). \tag{6.64}$$

Δηλαδή, αντί να λάβουμε μια συνάρτηση της οποίας η πολυπλοκότητα είναι ανάλογη με τον αριθμό των διαστάσεων του χώρου χαρακτηριστικών, έχουμε καταλήξει σε μια έκφραση με πολυπλοκότητα ανάλογη με τον αριθμό των διανυσμάτων υποστήριξης.

Επιπρόσθετα, έχουμε μια έκφραση που υπολογίζει το αποτέλεσμα ενός εσωτερικού γινομένου του χώρου χαρακτηριστικών, στο χώρο εισόδου, καθιστώντας έτσι περιττούς τους υπολογισμούς του μετασχηματισμού Φ. Αυτό βέβαια ήταν αποτέλεσμα της προσεκτικής επιλογής της μορφής του μετασχηματισμού Φ. Ας δούμε λοιπόν αναλυτικότερα τους μετασχηματισμούς αυτούς.

Δεδομένου ενός κατάλληλου μετασχηματισμού $\Phi : \mathbb{R}^n \to \mathbb{R}^m$ με $m \geq n$, συναρτήσεις της μορφής

$$k(\bar{x}, \bar{y}) = \Phi(\bar{x}) \bullet \Phi(\bar{y}), \tag{6.65}$$

με $\bar{x}, \bar{y} \in \mathbb{R}^n$, καλούνται *συναρτήσεις-πυρήνες* ή *πυρήνες*. Οι πυρήνες υπολογίζουν ένα εσωτερικό γινόμενο του χώρου χαρακτηριστικών και το καθοριστικό τους γνώρισμα είναι ότι η τιμή αυτού του εσωτερικού γινομένου στην πραγματικότητα υπολογίζεται στο χώρο εισόδου.

Η συνάρτηση απόφασης της (6.63) μπορεί λοιπόν, σύμφωνα με τα παραπάνω, να γραφεί

$$\hat{f}(\bar{x}) = \operatorname{sgn}\left(\sum_{i=1}^{l} \alpha_i^* y_i k(\bar{x}_i, \bar{x}) - b^* \right). \tag{6.66}$$

Αυτό είναι το λεγόμενο *τέχνασμα του πυρήνα*. Χρησιμοποιώντας δηλαδή μια κατάλληλη συνάρτηση-πυρήνα μπορούμε να εκμεταλλευτούμε τα οφέλη των μετασχηματισμών σε χώρους χαρακτηριστικών, χωρίς να απαιτείται ο αναλυτικός υπολογισμός των μετασχηματισμών αυτών, αφού οι υπολογισμοί στο χώρο χαρακτηριστικών απλοποιούνται σε υπολογισμούς στο χώρο εισόδου. Με συνετή επιλογή της συνάρτησης-πυρήνα μπορούμε να ελέγξουμε τη πολυπλοκότητα του μοντέλου και το *τέχνασμα* έγκειται στην επιλογή της κατάλληλης συνάρτησης πυρήνα για το εκάστοτε διαθέσιμο σύνολο εκπαίδευσης.

Ήδη έχουμε δει δύο συναρτήσεις-πυρήνες. Στη πρώτη περίπτωση έστω $\Phi : \mathbb{R}^n \to \mathbb{R}^n$, η ταυτοτική συνάρτηση στο $\mathbb{R}^n$. Τότε

$$k(\bar{x}, \bar{y}) = \Phi(\bar{x}) \bullet \Phi(\bar{y}) = \bar{x} \bullet \bar{y} \tag{6.67}$$

με $\bar{x}, \bar{y} \in \mathbb{R}^n$. Ο πυρήνας αυτός καλείται *γραμμικός* και εδώ ο χώρος χαρακτηριστικών ταυτίζεται με το χώρο εισόδου. Θα δούμε αργότερα ότι ένας τέτοιος πυρήνας είναι χρήσιμος σε περιπτώσεις συνόλων δεδομένων μεγάλης διάστασης σε συνδυασμό με ταξινομητές εύκαμπτου περιθωρίου.

Στη δεύτερη περίπτωση ο πυρήνας μας βασίστηκε στην απεικόνιση $\Phi:\mathbb{R}^2\to\mathbb{R}^3$ με $\Phi(x_1,x_2)=(x_1^2,x_2^2,\sqrt{2}x_1x_2)$, έτσι ώστε

$$k(\bar{x},\bar{y})=\Phi(\bar{x})\bullet\Phi(\bar{y})=(\bar{x}\bullet\bar{y})^2 \tag{6.68}$$

με τα $\bar{x},\bar{y}\in\mathbb{R}^2$. Ο πυρήνας αυτός καλείται *ομογενής πολυωνυμικός δευτέρου βαθμού* και απεικονίζει δευτεροβάθμιες επιφάνειες απόφασης του χώρου εισόδου σε γραμμικές στο χώρο χαρακτηριστικών. Μπορεί δε εύκολα να επεκταθεί σε χώρους εισόδου οποιασδήποτε διάστασης όπου $\bar{x},\bar{y}\in\mathbb{R}^n$. Ο ΠΙΝΑΚΑΣ 6-1 περιλαμβάνει μια λίστα των πιο διαδεδομένων συναρτήσεων-πυρήνων[1].

ΠΙΝΑΚΑΣ 6-1 ΤΥΠΙΚΟΙ ΠΥΡΗΝΕΣ ΚΑΙ ΕΛΕΥΘΕΡΕΣ ΠΑΡΑΜΕΤΡΟΙ

Ονομασία Πυρήνα	Συνάρτηση Πυρήνα [α]	Ελεύθερες Παράμετροι
Γραμμικός	$k(\bar{x},\bar{y})=\bar{x}\bullet\bar{y}$	Δεν υπάρχουν
Ομογενής πολυωνυμικός	$k(\bar{x},\bar{y})=(\bar{x}\bullet\bar{y})^d$	$d\geq 2$
Μη-ομογενής πολυωνυμικός	$k(\bar{x},\bar{y})=(\bar{x}\bullet\bar{y}+c)^d$	$d\geq 2, c>0$
Γκαουσιανός (Gaussian)	$k(\bar{x},\bar{y})=e^{-\left(\|\bar{x}-\bar{y}\|^2/2\sigma^2\right)}$	$\sigma>0$

[α] $\bar{x},\bar{y}\in\mathbb{R}^n$

Στις προηγούμενες παραγράφους επικεντρωθήκαμε στη δυϊκή διατύπωση του κάθετου στην επιφάνεια απόφασης διανύσματος $\bar{w}^*$. Όμως και η δυϊκή διατύπωση του όρου μετάθεσης b^* [βλ. (6.44)] μπορεί επίσης να αποδοθεί με βάση τη συνάρτηση-πυρήνα:

$$\begin{aligned} b^* &= \bar{w}^*\bullet\Phi\left(\bar{x}_{sv+}\right)-1 \\ &= \sum_{i=1}^{l}\alpha_i^* y_i\Phi\left(\bar{x}_i\right)\bullet\Phi\left(\bar{x}_{sv+}\right)-1 \\ &= \sum_{i=1}^{l}\alpha_i^* y_i k\left(\bar{x}_i,\bar{x}_{sv+}\right)-1 \end{aligned}$$

Προκειμένου να βρεθούν τα διανύσματα υποστήριξης στο χώρο χαρακτηριστικών, θα πρέπει να εφαρμοστεί το τέχνασμα του πυρήνα και στον αλγόριθμο εκπαίδευσης:

[1] Για περισσότερες συναρτήσεις-πυρήνες βλ. [1], §2.3.2.7. Για τεχνικές κατασκευής νέων πυρήνων από προϋπάρχοντες βλ. [4], §6.2.

$$\bar{\alpha}^* = \arg\max_{\bar{\alpha}} \left(\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j k\left(\bar{x}_i, \bar{x}_j\right) \right) \tag{6.69}$$

υπό τους περιορισμούς

$$\sum_{i=1}^{l} \alpha_i y_i = 0, \tag{6.70}$$

$$\alpha_i \geq 0, \quad i = 1, \ldots, l \tag{6.71}$$

για την εύρεση των βέλτιστων πολλαπλασιαστών Lagrange $\bar{\alpha}^*$. Η συνάρτηση $k(\bar{x}, \bar{y})$ που θα χρησιμοποιηθεί θα πρέπει προφανώς να ταυτίζεται με τη συνάρτηση-πυρήνα που χρησιμοποιήθηκε στο μοντέλο. Παρατηρήστε ότι οι περιορισμοί δεν επηρεάζονται από το τέχνασμα του πυρήνα και είναι ανεξάρτητοι από το χώρο χαρακτηριστικών που θα επιλεγεί. Για ακόμη πιο πολύπλοκες επιφάνειες απόφασης του χώρου εισόδου θα μπορούσε να δοκιμαστεί η χρήση πολυωνυμικών πυρήνων ανωτέρου βαθμού ή και ακόμη πιο σύνθετων, όπως του γκαουσιανού.

Δεν είναι πάντα προφανές το ποιος πυρήνας είναι καταλληλότερος για κάθε συγκεκριμένο πρόβλημα. Αν δεν υπάρχει πρότερη γνώση, ή δε βρεθούν αναφορές σε σχετική βιβλιογραφία, η καλύτερη προσέγγιση είναι η επιλογή είτε του γκαουσιανού, είτε χαμηλού βαθμού πολυωνυμικού (με $d = 1$ ή 2).

Επίσης, η διαδικασία επιλογής τιμών για τις ελεύθερες παραμέτρους του πυρήνα δεν είναι, γενικά, τετριμμένη καθώς αντικατοπτρίζει την επιθυμητή στάθμιση μεταξύ των απαιτήσεων πολυπλοκότητας και ακρίβειας του μοντέλου. Στην ενότητα 6.9 θα παρουσιαστούν κάποιοι τρόποι για την επιλογή τιμών των ελεύθερων παραμέτρων με τα λογισμικά πακέτα που χρησιμοποιούνται στην εργασία.

6.5.2 Εμβαθύνοντας στις συναρτήσεις πυρήνα

Στην περίπτωση του ομογενούς πολυωνυμικού πυρήνα $2^{ου}$ βαθμού είδαμε ότι με προσεκτικό ορισμό της απεικόνισης Φ και του χώρου χαρακτηριστικών, ισχύει η ιδιότητα $\Phi(\bar{x}) \bullet \Phi(\bar{y}) = (\bar{x} \bullet \bar{y})^2$ και το εσωτερικό γινόμενο μπορεί να υπολογιστεί στο χώρο εισόδου. Στην ενότητα αυτή θα δειχθεί ότι για κάθε πυρήνα υπάρχει μια αντίστοιχη *πρότυπη* (ή *κανονική*) απεικόνιση καθώς και ένας χώρος χαρακτηριστικών. Την ύπαρξη των πρότυπων αυτών δομών εγγυάται ένα σύνολο υποθέσεων σχετικά με τον πυρήνα.

Η ακόλουθη ιδιότητα είναι σημαντική καθώς εξασφαλίζει ότι το εσωτερικό γινόμενο ορίζεται στο χώρο χαρακτηριστικών. Αποτελεί μάλιστα το κύριο χαρακτηριστικό των πυρήνων που χρησιμοποιούνται στις μηχανές διανυσμάτων υποστήριξης:

Ορισμός 6.1 **(*Θετικά ορισμένος πυρήνας*)** *Μία συνάρτηση* $k:\mathbb{R}^n\times\mathbb{R}^n\to\mathbb{R}$ *τέτοια ώστε να ισχύει:*

$$\sum_{i=1}^{l}\sum_{j=1}^{l}\theta_i\theta_j k\left(\overline{x}_i,\overline{x}_j\right)\geq 0 \tag{6.72}$$

με $\theta_i,\theta_j\in\mathbb{R}$*, και* $\overline{x}_1,\ldots,\overline{x}_l$ *σύνολο σημείων στον* $\mathbb{R}^n$*, καλείται «**θετικά ορισμένος πυρήνας**»* (***positive-definite kernel***)[1].

Προκειμένου να κατασκευάσουμε τους πρότυπους χώρους χαρακτηριστικών μας είναι απαραίτητη μία ακόμα ιδιότητα των πυρήνων. Θα χρειαστεί μάλιστα η εισαγωγή ενός νέου συμβολισμού. Έστω $k:\mathbb{R}^n\times\mathbb{R}^n\to\mathbb{R}$ ένας πυρήνας. Τότε $k(\cdot,\overline{x})$ θα είναι ένας μερικά αποτιμημένος πυρήνας με $\overline{x}\in\mathbb{R}^n$, ο οποίος παριστάνει μια συνάρτηση $\mathbb{R}^n\to\mathbb{R}$, ως προς το πρώτο όρισμα και με δείκτη το δεύτερο.

Θεώρημα 6.2 **(*Ιδιότητα αναπαράστασης πυρήνα*)** *Έστω* $k:\mathbb{R}^n\times\mathbb{R}^n\to\mathbb{R}$ *θετικά ορισμένος πυρήνας. Τότε, για κάθε* $\overline{x},\overline{y}\in\mathbb{R}^n$ *ισχύει*

$$k\left(\overline{x},\overline{y}\right)=k\left(\overline{x},\cdot\right)\bullet k\left(\cdot,\overline{y}\right). \tag{6.73}$$

Με άλλα λόγια η τιμή ενός πυρήνα μπορεί να υπολογιστεί στα σημεία $\overline{x}$ και $\overline{y}$ μέσω του εσωτερικού γινομένου δύο μερικά αποτιμημένων πυρήνων στα σημεία αυτά.

Τέλος, η επόμενη γνωστή ανισότητα θα αποβεί χρήσιμη στη διερεύνηση της δομής των πρότυπων χώρων χαρακτηριστικών:

Θεώρημα 6.3 **(*Ανισότητα Cauchy-Schwarz*)** *Για κάθε* $\overline{x},\overline{y}\in\mathbb{R}^n$

$$\left(\overline{x}\bullet\overline{y}\right)^2\leq\left(\overline{x}\bullet\overline{x}\right)\left(\overline{y}\bullet\overline{y}\right) \tag{6.74}$$

[1] Η συνθήκη (6.72) καλείται ***συνθήκη του Mercer*** (βλ. π.χ. [1]). Γι αυτό και δεν αποτελούν όλα τα μέτρα ομοιότητας έγκυρες συναρτήσεις πυρήνα, καθώς θα πρέπει να ικανοποιείται τη συνθήκη του Mercer.

Μπορούμε πλέον να δείξουμε ότι, για κάθε πυρήνα, υπάρχει ένας πρότυπος χώρος χαρακτηριστικών[1]. Δηλαδή, για κάθε θετικά ορισμένο πυρήνα $k:\mathbb{R}^n\times\mathbb{R}^n\to\mathbb{R}$ μπορεί να κατασκευαστεί ένας πρότυπος χώρος χαρακτηριστικών Z με μία αντίστοιχη απεικόνιση $\Phi:\mathbb{R}^n\to Z$ έτσι ώστε να ισχύει η σχέση:

$$k\left(\bar{x},\bar{y}\right)=\Phi\left(\bar{x}\right)\bullet\Phi\left(\bar{y}\right) \tag{6.75}$$

για όλα τα $\bar{x},\bar{y}\in\mathbb{R}^n$. Για τη διαδικασία κατασκευής τού πρότυπου χώρου χαρακτηριστικών θα ακολουθηθούν τα παρακάτω βήματα:

- Ορισμός ενός χώρου χαρακτηριστικών και κατασκευή της απεικόνισης Φ.
- Μετατροπή του χώρου χαρακτηριστικών σε διανυσματικό χώρο.
- Ορισμός του εσωτερικού γινομένου στο χώρο αυτό.
- Απόδειξη ότι το εσωτερικό γινόμενο ικανοποιεί την (6.75).

Δοθέντος ενός θετικά ορισμένου πυρήνα $k:\mathbb{R}^n\times\mathbb{R}^n\to\mathbb{R}$ και ενός συνόλου σημείων του χώρου εισόδου $x_1,\ldots,x_l\in\mathbb{R}^n$, ορίζουμε το χώρο χαρακτηριστικών Z ως το σύνολο όλων των *συναρτήσεων* που απεικονίζουν σημεία του χώρου εισόδου $\mathbb{R}^n$ σε πραγματικούς αριθμούς.

$$Z=\left\{f:\mathbb{R}^n\to\mathbb{R}\right\}. \tag{6.76}$$

Παρατηρήστε ότι ο χώρος χαρακτηριστικών αποτελείται από συναρτήσεις και όχι από διανύσματα. Ορίζουμε την απεικόνιση από το χώρο εισόδου στο χώρο χαρακτηριστικών $\Phi:\mathbb{R}^n\to Z$ ως:

$$\Phi\left(\bar{x}\right)=k\left(\cdot,\bar{x}\right) \tag{6.77}$$

για όλα τα $\bar{x}\in\mathbb{R}^n$. Δηλαδή η Φ απεικονίζει το σημείο $\bar{x}\in\mathbb{R}^n$ στη συνάρτηση $k\left(\cdot,\bar{x}\right)\in Z$.

Για να γίνει αυτός ο χώρος χαρακτηριστικών διανυσματικός, θα επιτρέψουμε την έκφραση κάθε τυχούσας συνάρτησης του χώρου, ως γραμμικού συνδυασμού των μερικά αποτιμημένων πυρήνων στα δοσμένα σημεία.

Δηλαδή, μία συνάρτηση $h:\mathbb{R}^n\to\mathbb{R}$ μπορεί να εκφραστεί ως

[1] Ο πρότυπος χώρος χαρακτηριστικών ενός πυρήνα είναι ένας «***χώρος Hilbert αναπαράστασης πυρήνα***» (***Reproducing Kernel Hilbert Space - RKHS***) και, κατά μία έννοια, είναι ο *μικρότερος* χώρος χαρακτηριστικών αυτού του πυρήνα. Για περισσότερα βλέπε [32], [12], [30].

$$h = \sum_{i=1}^{l} \theta_i k\left(\cdot, \overline{x}_i\right) \tag{6.78}$$

όπου $\theta_i \in \mathbb{R}$. Θα μπορούσε δηλαδή να θεωρήσει κανείς ότι οι μερικά αποτιμημένοι πυρήνες για τα δοθέντα σημεία του χώρου εισόδου αποτελούν μια «βάση» του χώρου χαρακτηριστικών. Για τον ορισμό του εσωτερικού γινομένου στο χώρο αυτό, έστω $g = \sum_{j=1}^{l} \gamma_j k\left(\cdot, \overline{x}_j\right)$, με $\gamma_j \in \mathbb{R}$, μία άλλη συνάρτηση του χώρου χαρακτηριστικών. Το εσωτερικό γινόμενο ορίζεται τότε ως

$$h \bullet g = \sum_{i=1}^{l} \sum_{j=1}^{l} \theta_i \gamma_j k\left(\cdot, \overline{x}_i\right) \bullet k\left(\cdot, \overline{x}_j\right) = \sum_{i=1}^{l} \sum_{j=1}^{l} \theta_i \gamma_j k\left(\overline{x}_i, \overline{x}_j\right). \tag{6.79}$$

Για κάθε συνάρτηση $f, g, h \in Z$ και σταθερές $p, q \in \mathbb{R}$, απαιτείται επιπλέον να δειχθούν και οι παρακάτω αλγεβρικές ιδιότητες του εσωτερικού γινομένου:

1. $f \bullet g = g \bullet f$ (συμμετρία)
2. $(pf + qg) \bullet h = pf \bullet h + qg \bullet h$ (γραμμικότητα)
3. $f \bullet f \geq 0$ (μη-αρνητικότητα)
4. $f \bullet f = 0$ αν και μόνο αν $f = 0$ (μη-εκφυλισμός)

Η (1) προκύπτει άμεσα από τη συμμετρία του πυρήνα k [βλ. και (6.79)]. Με την h όπως ορίστηκε στην (6.78), δείχνουμε στη συνέχεια ότι ισχύει η (2):

$$\begin{aligned}
\left(pf + qg\right) \bullet h &= \left(pf + qg\right) \bullet \sum_{i=1}^{l} \theta_i k\left(\cdot, \overline{x}_i\right) \\
&= \sum_{i=1}^{l} \theta_i \left(pf + qg\right) \bullet k\left(\cdot, \overline{x}_i\right) \\
&= \sum_{i=1}^{l} \theta_i \left(pf \bullet k\left(\cdot, \overline{x}_i\right) + qg \bullet k\left(\cdot, \overline{x}_i\right)\right) \\
&= \sum_{i=1}^{l} \theta_i pf \bullet k\left(\cdot, \overline{x}_i\right) + \sum_{i=1}^{l} \theta_i qg \bullet k\left(\cdot, \overline{x}_i\right) \\
&= pf \bullet \sum_{i=1}^{l} \theta_i k\left(\cdot, \overline{x}_i\right) + qg \bullet \sum_{i=1}^{l} \theta_i k\left(\cdot, \overline{x}_i\right) \\
&= pf \bullet h + qg \bullet h.
\end{aligned}$$

Για τις (3) και (4), έστω ότι $f = \sum_{i=1}^{l} \theta_i k(\cdot, \overline{x}_i)$. Η (3) προκύπτει άμεσα λόγω της (6.72):

$$f \bullet f = \sum_{i=1}^{l} \sum_{j=1}^{l} \theta_i \theta_j k\left(\overline{x}_i, \overline{x}_j\right) \geq 0. \tag{6.80}$$

Για την (4), αν $f = 0$ τότε προφανώς ισχύει $f \bullet f = 0$. Για να δειχθεί η αντίστροφη σχέση και λαμβάνοντας υπόψη την ιδιότητα αναπαράστασης, εργαζόμαστε ως εξής:

$$f\left(\overline{x}\right) = \sum_{i=1}^{l} \theta_i k\left(\overline{x}, \overline{x}_i\right) = \sum_{i=1}^{l} \theta_i k\left(\cdot, \overline{x}\right) \bullet k\left(\cdot, \overline{x}_i\right) = k\left(\cdot, \overline{x}\right) \bullet f. \tag{6.81}$$

Κάνοντας χρήση και της ανισότητας Cauchy-Schwarz, έχουμε:

$$\left(k\left(\cdot, \overline{x}\right) \bullet f\right)^2 \leq \left(k\left(\cdot, \overline{x}\right) \bullet k\left(\cdot, \overline{x}\right)\right)\left(f \bullet f\right) \tag{6.82}$$

οπότε

$$\left(f\left(\overline{x}\right)\right)^2 \leq k\left(\overline{x}, \overline{x}\right)\left(f \bullet f\right) \tag{6.83}$$

για όλα τα $\overline{x} \in \mathbb{R}^n$. Ως εκ τούτου έπεται ότι αν $f \bullet f = 0$ συνεπάγεται $f = 0$. Το εσωτερικό γινόμενο λοιπόν είναι καλά ορισμένο.

Τέλος, θα πρέπει να δείξουμε ότι τηρείται και η προϋπόθεση της (6.75). Από τον τρόπο που ορίστηκε η απεικόνιση, σε συνδυασμό με την ιδιότητα αναπαράστασης, έχουμε:

$$\Phi\left(\overline{x}\right) \bullet \Phi\left(\overline{y}\right) = k\left(\cdot, \overline{x}\right) \bullet k\left(\cdot, \overline{y}\right) = k\left(\overline{x}, \overline{y}\right). \tag{6.84}$$

Ένα άμεσο επακόλουθο της παραπάνω κατασκευής είναι ότι οι χώροι χαρακτηριστικών των πυρήνων δεν είναι μοναδικοί. Για να γίνει καλύτερα κατανοητό αυτό, θα δούμε ένα παράδειγμα όπου θα χρησιμοποιήσουμε τον ομογενή πολυωνυμικό πυρήνα δευτέρου βαθμού, δηλαδή $k(\overline{x}, \overline{y}) = (\overline{x} \bullet \overline{y})^2$ με $\overline{x}, \overline{y} \in \mathbb{R}^2$.

Έστω λοιπόν $\Phi : \mathbb{R}^2 \to \mathbb{R}^3$ και $\Psi : \mathbb{R}^2 \to \{f : \mathbb{R}^2 \to \mathbb{R}\}$ δύο απεικονίσεις από το χώρο εισόδου σε δύο διαφορετικούς χώρους χαρακτηριστικών, έτσι ώστε

$$\Phi\left(\overline{x}\right) = \Phi\left(x_1, x_2\right) = \left(x_1^2, x_2^2, \sqrt{2} x_1 x_2\right) \tag{6.85}$$

και

$$\Psi\left(\overline{x}\right) = k\left(\cdot, \overline{x}\right) = \left((\cdot) \bullet \overline{x}\right)^2. \tag{6.86}$$

Τότε

$$\begin{aligned} \Phi\left(\overline{x}\right) \bullet \Phi\left(\overline{y}\right) &= \left(x_1^2, x_2^2, \sqrt{2} x_1 x_2\right) \bullet \left(y_1^2, y_2^2, \sqrt{2} y_1 y_2\right) = \left(x \bullet y\right)^2 \\ &= k\left(\overline{x}, \overline{y}\right) = k\left(\cdot, \overline{x}\right) \bullet k\left(\cdot, \overline{y}\right) = \left((\cdot) \bullet \overline{x}\right)^2 \bullet \left((\cdot) \bullet \overline{y}\right)^2 \\ &= \Psi\left(\overline{x}\right) \bullet \Psi\left(\overline{y}\right). \end{aligned}$$

Από τα παραπάνω φαίνεται ότι οι χώροι χαρακτηριστικών δεν είναι μοναδικοί, παρόλα αυτά όμως οι τιμές των εσωτερικών γινομένων που υπολογίζουν είναι. Δηλαδή, για κάθε ζευγάρι στοιχείων του χώρου εισόδου, τα εσωτερικά γινόμενα των διαφόρων χώρων χαρακτηριστικών οδηγούν πάντα στο ίδιο αποτέλεσμα για το δεδομένο ζευγάρι στοιχείων.

Ας δούμε σαν παράδειγμα την πρότυπη απεικόνιση και τον αντίστοιχο χώρο χαρακτηριστικών για τον ομογενή πολυωνυμικό πυρήνα δευτέρου βαθμού και την απεικόνιση $\Phi : \mathbb{R}^2 \to \mathbb{R}^3$ της (6.85), όπου

$$\overline{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \mapsto \quad \Phi(\overline{x}) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1 x_2 \end{bmatrix}$$

και

$$k(\overline{x}, \overline{y}) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1 x_2 \end{bmatrix} \bullet \begin{bmatrix} y_1^2 \\ y_2^2 \\ \sqrt{2}y_1 y_2 \end{bmatrix}.$$

Έστω μια συνάρτηση των πολυωνύμων δεύτερης τάξης ως προς $\overline{x}$:

$$f(\overline{x}) = ax_1^2 + bx_2^2 + c\sqrt{2}x_1 x_2.$$

Η συνάρτηση αυτή ανήκει στο χώρο των συναρτήσεων $f : \mathbb{R}^2 \to \mathbb{R}$. Συμβολίζουμε το χώρο αυτό με H.

Ας ορίσουμε την ακόλουθη ισοδύναμη έκφραση για την f :

$$f(.) = \begin{bmatrix} a \\ b \\ c \end{bmatrix}. \tag{6.87}$$

Η έκφραση αυτή αναφέρεται στην ίδια τη συνάρτηση κατά τρόπο αφηρημένο. Ορισμένες φορές μάλιστα, όταν δεν προκαλείται σύγχυση, γράφουμε μόνο f αντί για $f(.)$.

Ο συμβολισμός $f(\overline{x}) \in \mathbb{R}$ αναφέρεται στην αποτίμηση της συνάρτησης σε ένα συγκεκριμένο σημείο (και δεν είναι παρά ένας πραγματικός αριθμός). Σύμφωνα με τα παραπάνω μπορούμε να γράψουμε:

$$f(\overline{x}) = f(.) \bullet \Phi(\overline{x}).$$

Με άλλα λόγια, η αποτίμηση της f στο $\bar{x}$ μπορεί να γραφεί με τη μορφή εσωτερικού γινομένου στο χώρο χαρακτηριστικών. Συχνά χρησιμοποιείται ο παρακάτω συμβολισμός με τον οποίο γίνεται άμεση αναφορά στο σχετικό χώρο των συναρτήσεων f

$$f(\bar{x}) = \langle f(.), \Phi(\bar{x}) \rangle_H .$$

Αυτή η συλλογιστική μάς οδηγεί σε ένα συμπέρασμα που αρχικά μοιάζει αντιφατικό: Η ίδια η $\Phi(\bar{x})$ μπορεί να θεωρηθεί ως μια *συνάρτηση* που απεικονίζει το $\mathbb{R}^2$ στο $\mathbb{R}$. Χρησιμοποιώντας το συμβολισμό της (6.87) μπορούμε να γράψουμε

$$k(.,\bar{y}) = \begin{bmatrix} y_1^2 \\ y_2^2 \\ \sqrt{2}y_1y_2 \end{bmatrix} = \Phi(\bar{y})$$

και στη συνέχεια

$$k(.,\bar{y}) \bullet \Phi(\bar{x}) = ax_1^2 + bx_2^2 + c\sqrt{2}x_1x_2$$

όπου $a = y_1^2, b = y_2^2$ και $c = \sqrt{2}y_1y_2$. Λόγω της συμμετρίας των ορισμάτων θα μπορούσαμε ισοδύναμα να είχαμε γράψει

$$\begin{aligned} k(.,\bar{x}) \bullet \Phi(\bar{y}) &= uy_1^2 + vy_2^2 + w\sqrt{2}y_1y_2 \\ &= k(\bar{x},\bar{y}). \end{aligned}$$

Μπορούμε δηλαδή, χωρίς να προκαλείται αντίφαση, να γράψουμε: $\Phi(\bar{x}) = k(.,\bar{x})$ και $\Phi(\bar{y}) = k(.,\bar{y})$. Τότε, η απεικόνιση Φ καλείται πρότυπη απεικόνιση και ο χώρος H πρότυπος χώρος χαρακτηριστικών ([32], Lemma 4.19).

Το παράδειγμα αυτό κατέδειξε τα δύο καθοριστικά γνωρίσματα του πρότυπου χώρου χαρακτηριστικών:

- Το γεγονός ότι η απεικόνιση κάθε σημείου του χώρου εισόδου περιλαμβάνεται στον πρότυπο χώρο χαρακτηριστικών: $\forall \bar{x} \in \mathbb{R}^2, k(.,\bar{x}) \in H$
- Την ιδιότητα αναπαράστασης: $\forall \bar{x} \in \mathbb{R}^2, \forall f \in H, \quad f \bullet k(.,\bar{x}) = f(\bar{x})$

Συγκεκριμένα, για κάθε $\bar{x}, \bar{y} \in \mathbb{R}^2$,

$$k(\bar{x},\bar{y}) = k(\bar{x},\cdot) \bullet k(\cdot,\bar{y}).$$

6.6 Ταξινομητές Εύκαμπτου Περιθωρίου

Τα δεδομένα εκπαίδευσης σε πραγματικά προβλήματα περιλαμβάνουν συχνά θόρυβο οφειλόμενο κυρίως σε σφάλματα μέτρησης ή σε σφάλματα κατά την καταχώρηση των δεδομένων. Στην ενότητα αυτή θα παρουσιαστεί μια γενίκευση του ταξινομητή μεγίστου περιθωρίου που αντιμετωπίζει την ύπαρξη θορύβου στα δεδομένα εκπαίδευσης. Αυτό γίνεται αγνοώντας συγκεκριμένες παρατηρήσεις οι οποίες θεωρείται ότι οφείλονται σε θόρυβο και καταλήγοντας εν τέλει σε απλούστερες επιφάνειες απόφασης από αυτές που θα προέκυπταν κανονικά. Αυτό είναι άλλωστε θεμιτό καθώς, ως γνωστόν, απλούστερα μοντέλα τείνουν να γενικεύουν καλύτερα.

Σύμφωνα με όσα έχουν έως τώρα αναφερθεί, τα μοντέλα των ταξινομητών μεγίστου περιθωρίου είναι της μορφής

$$\hat{f}\left(\bar{x}\right)=\operatorname{sgn}\left(\bar{w}\bullet\bar{x}-b\right) \tag{6.88}$$

όπου το κάθετο διάνυσμα $\bar{w}$ και ο όρος μετάθεσης b της επιφάνειας απόφασης προσδιορίζονται από την επίλυση του πρωταρχικού προβλήματος βελτιστοποίησης:

$$\min\phi\left(\bar{w},b\right)=\min\frac{1}{2}\bar{w}\bullet\bar{w}, \tag{6.89}$$

υπό τους περιορισμούς

$$y_i\left(\bar{w}\bullet\bar{x}_i-b\right)-1\geq 0,\ \text{με}\ i=1,\ldots,l \tag{6.90}$$

και με δεδομένο ένα σύνολο εκπαίδευσης $(\bar{x}_1,y_1),\ldots,(\bar{x}_l,y_l)\in\mathbb{R}^n\times\{+1,-1\}$.

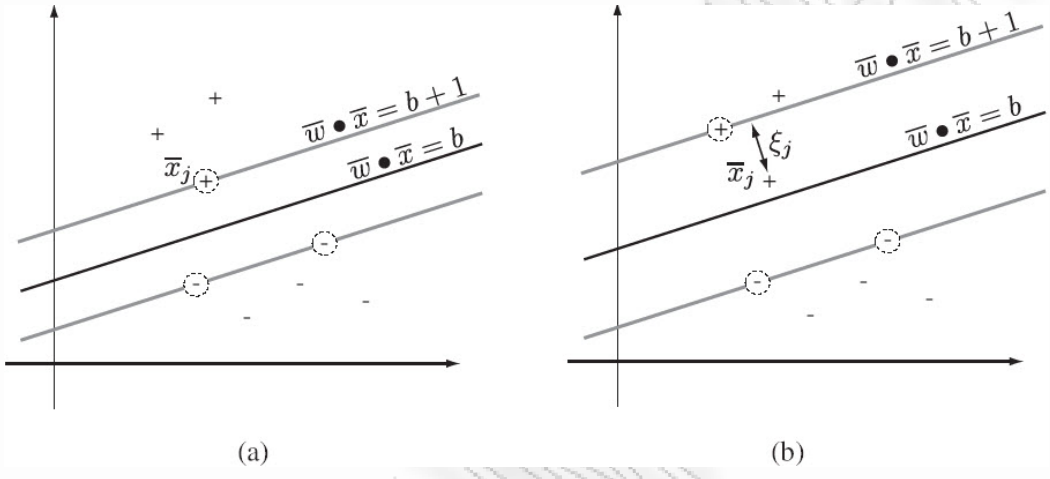
Κατά τη διαδικασία της βελτιστοποίησης προκύπτει ένας ταξινομητής μεγίστου περιθωρίου ο οποίος εξασφαλίζει ότι τα υπερεπίπεδα στήριξης θα τοποθετηθούν σε όσο το δυνατόν μεγαλύτερη απόσταση από την επιφάνεια απόφασης, έτσι που ίσα να εφάπτονται στα όρια των κλάσεων.

Η κατασκευή αυτή είναι όμως εν μέρει μόνο επιτυχής στην περίπτωση των δεδομένων εκπαίδευσης που εμπεριέχουν θόρυβο, όπου το μέγεθος του περιθωρίου περιορίζεται λόγω μερικών μόνο – ενδεχομένως εσφαλμένων – σημείων. Ωστόσο, θα ήταν εφικτό να μειώσουμε την επίδραση που επιφέρουν τα σημεία αυτά στο μέγεθος του περιθωρίου αν τους επιτρέπαμε να παραμείνουν στη «λάθος» πλευρά του αντίστοιχου υπερεπιπέδου στήριξης, μέσω της εισαγωγής των λεγόμενων *χαλαρών μεταβλητών*. Οι μεταβλητές αυτές δεν είναι παρά κάποιοι όροι σφάλματος οι οποίοι μετρούν πόσο απέχει ένα συγκεκριμένο σημείο που βρίσκεται στη λάθος πλευρά του αντίστοιχου υπερεπιπέδου στήριξης από αυτό ή, με άλλα λόγια, το πόσο

μεγάλο είναι το σφάλμα που διαπράττεται αν επιτραπεί στο σημείο αυτό να πάψει να αποτελεί περιορισμό για τη θέση του υπερεπιπέδου στήριξης. Τα προηγούμενα απεικονίζονται στο Σχήμα 6-2.

Σχήμα 6-2

(a) Το περιθώριο περιορίζεται από το σημείο $(\overline{x}_j, +1)$. (b) Το περιθώριο δεν περιορίζεται από το σημείο $(\overline{x}_j, +1)$ και το προκύπτον σφάλμα προσμετρείται από τη μεταβλητή ξ_j.



Παρατηρήστε ότι στην περίπτωση (b) του παραπάνω σχήματος καταστρατηγείται ο περιορισμός του προβλήματος βελτιστοποίησης: $\overline{w} \bullet \overline{x}_j - b - 1 \geq 0$. Παρόλα αυτά μπορούμε να επανακτήσουμε ένα λειτουργικό περιορισμό αν λάβουμε υπόψη τη χαλαρή μεταβλητή, $\overline{w} \bullet \overline{x}_j - b + \xi_j - 1 \geq 0$, με $\xi_j \geq 0$ (δηλ. το σφάλμα προσμετράται πάντα ως θετική ποσότητα). Αν λοιπόν εισάγουμε για κάθε σημείο του συνόλου εκπαίδευσης μια τέτοια μεταβλητή, οι αντίστοιχοι τροποποιημένοι περιορισμοί θα λάβουν τη μορφή:

$$y_i\left(\overline{w} \bullet \overline{x}_i - b\right) + \xi_i - 1 \geq 0 \qquad (6.91)$$

με $\xi_i \geq 0$. Προφανώς η μεταβλητή αυτή ισούται με 0 για όλα τα σημεία που δεν συνιστούν περιορισμό για τη θέση του αντίστοιχου υπερεπιπέδου στήριξης, συνεπώς στις περιπτώσεις αυτές μπορεί να διατηρηθεί ο περιορισμός στην αρχική του μορφή, $y_i(\overline{w} \bullet \overline{x}_i - b) - 1 \geq 0$.

Έτσι, η αντικειμενική συνάρτηση του προβλήματος μεγίστου περιθωρίου, στην οποία λαμβάνονται υπόψη και οι χαλαρές μεταβλητές, μπορεί να ξαναγραφεί σύμφωνα με την παρακάτω πρόταση.

***Πρόταση 6.2* (*Βελτιστοποίηση Εύκαμπτου Περιθωρίου*)** *Δοθέντος ενός συνόλου εκπαίδευσης*

$$D=\left\{\left(\bar{x}_1,y_1\right),\left(\bar{x}_2,y_2\right),\ldots,\left(\bar{x}_l,y_l\right)\right\}\subseteq\mathbb{R}^n\times\left\{+1,-1\right\},$$

μπορούμε να υπολογίσουμε την επιφάνεια απόφασης εύκαμπτου περιθωρίου $\bar{w}^*\bullet\bar{x}=b^*$ *λύνοντας το επόμενο πρόβλημα βελτιστοποίησης:*

$$\min_{\bar{w},\bar{\xi},b}\phi\left(\bar{w},\bar{\xi},b\right)=\min_{\bar{w},\bar{\xi},b}\left(\frac{1}{2}\bar{w}\bullet\bar{w}+C\sum_{i=1}^{l}\xi_i\right), \tag{6.92}$$

υπό τους περιορισμούς

$$y_i\left(\bar{w}\bullet\bar{x}_i-b\right)+\xi_i-1\geq 0, \tag{6.93}$$

$$\xi_i\geq 0, \tag{6.94}$$

με $i=1,\ldots,l,\quad\bar{\xi}=(\xi_1,\ldots,\xi_l)$ *και* $C>0$.

Καθώς όλοι οι όροι σφάλματος λαμβάνουν κατά σύμβαση θετικές τιμές, το πρόβλημα παραμένει ένα πρόβλημα κυρτής βελτιστοποίησης.

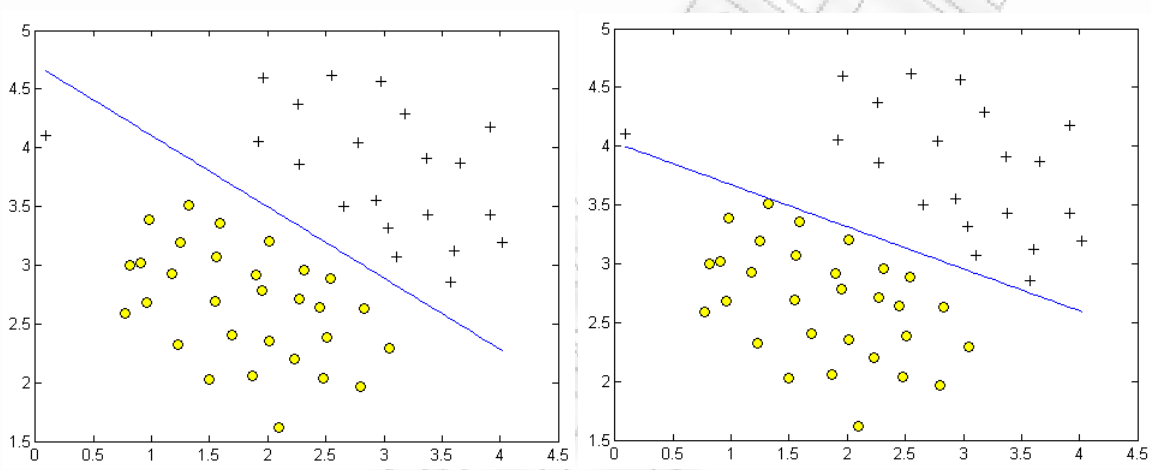
Για την αποφυγή τετριμμένων λύσεων, στις οποίες το σύνολο των δεδομένων εκπαίδευσης θεωρείται ως αποτέλεσμα θορύβου, οι χαλαρές μεταβλητές έχουν ληφθεί υπόψη με τη μορφή του όρου ποινής $C\sum_{i=1}^{l}\xi_i$ στην αντικειμενική συνάρτηση. Με τον τρόπο αυτό η βελτιστοποίηση της αντικειμενικής συνάρτησης ανάγεται σε ένα αντιστάθμισμα μεταξύ του μεγέθους του περιθωρίου και του μεγέθους του σφάλματος, όπου ως σφάλμα νοείται το άθροισμα των τιμών των χαλαρών μεταβλητών. Όσο μεγαλύτερο επιτρέπουμε να γίνει το περιθώριο, τόσο περισσότερα σημεία του συνόλου εκπαίδευσης θα βρίσκονται στη λάθος πλευρά του αντίστοιχου υπερεπιπέδου στήριξης και συνεπώς τόσο μεγαλύτερο θα γίνεται στο σφάλμα και αντίστροφα.

Πιο συγκεκριμένα, η επιλογή μεγάλου περιθωρίου θα έχει πιθανά ως αποτέλεσμα την εισαγωγή ενός μεγάλου αριθμού μη μηδενικών χαλαρών μεταβλητών. Αντίθετα, η επιλογή μικρού περιθωρίου θα μειώσει τον αριθμό των μη μηδενικών χαλαρών μεταβλητών, αλλά θα επιτρέψει σε σημεία που ενδεχομένως αποτελούν θόρυβο να καθορίσουν τη θέση της επιφάνειας απόφασης. Η σταθερά C (από τη λέξη *Cost* = κόστος) μας επιτρέπει να ελέγξουμε αυτό το αντιστάθμισμα μεταξύ του μεγέθους του περιθωρίου και του σφάλματος. Η σταθερά C είναι θετική και δεν μπορεί να ισούται με μηδέν, που σημαίνει ότι δεν μπορούμε να την αγνοήσουμε πλήρως. Μεγάλη τιμή της μεταβλητής C θα οδηγήσει σε λύση με όσο το

δυνατόν μικρότερο αριθμό μη μηδενικών χαλαρών μεταβλητών, επειδή τα σφάλματα συνεπάγονται μεγάλο κόστος, λόγω του C. Με άλλα λόγια, μεγάλες τιμές της C οδηγούν σε λύσεις με μικρά μεγέθη περιθωρίου[1]. Αν αντίθετα δοθεί μικρή τιμή στη μεταβλητή C, η εισαγωγή μη μηδενικών χαλαρών μεταβλητών δεν αποφέρει απαγορευτικό κόστος και μπορούν να βρεθούν λύσεις με μεγαλύτερο μέγεθος περιθωρίου, αγνοώντας κάποια από τα σημεία που βρίσκονται πολύ κοντά στην επιφάνεια απόφασης (βλ. Σχήμα 6-3).

Σχήμα 6-3

Γραμμική διαχωριστική επιφάνεια για μικρή (αριστερά) και μεγάλη (δεξιά) τιμή του C



Έτσι καταλήγουμε στις παρακάτω σχέσεις μεταξύ του κόστους και του μεγέθους περιθωρίου:

μεγάλο $C \sim$ μικρό περιθώριο,
μικρό $C \sim$ μεγάλο περιθώριο. (6.95)

Μία λύση λοιπόν $\overline{w}^*, \overline{\xi}^*, b^*$ του προβλήματος βελτιστοποίησης

$$\min_{\overline{w},\overline{\xi},b} \phi\left(\overline{w},\overline{\xi},b\right) = \frac{1}{2}\overline{w}^* \bullet \overline{w}^* + C\sum_{i=1}^{l}\xi_i^* = m^*$$

αποτελεί ένα αντιστάθμισμα μεταξύ του μεγέθους m^* του περιθωρίου και του μεγέθους του σφάλματος $\sum_{i=1}^{l}\xi_i^*$ για δεδομένη τιμή του κόστους C.

[1] Παρατηρήστε ότι ο συντελεστής C έχει επίδραση ανάλογη του $1/\lambda$, όπου λ ο συντελεστής ομαλοποίησης που χρησιμοποιήθηκε στη λογιστική παλινδρόμηση και στο νευρωνικό δίκτυο.

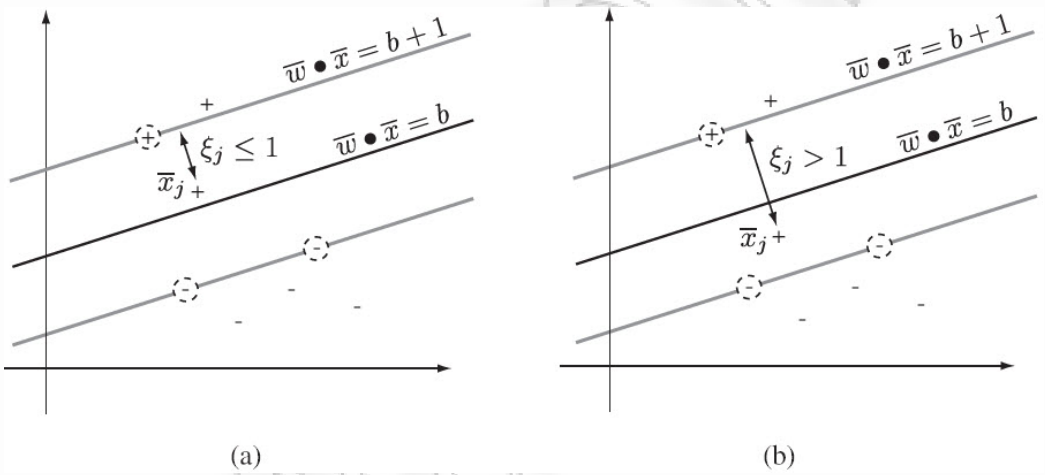
Καθώς οι χαλαρές μεταβλητές εμφανίζονται μόνον ως μέρος του αλγορίθμου εκπαίδευσης, το μοντέλο του ταξινομητή μεγίστου περιθωρίου παραμένει αμετάβλητο:

$$\hat{f}\left(\overline{x}\right)=\operatorname{sgn}\left(\overline{w}^{*}\bullet\overline{x}-b^{*}\right). \tag{6.96}$$

Η μόνη διαφορά είναι ότι τώρα επιτρέπεται στο μοντέλο μας να υποπέσει σε συγκεκριμένο αριθμό εσφαλμένων ταξινομήσεων, εξαρτώμενο από την τιμή της μεταβλητής κόστους C. Για να γίνουν καλύτερα αντιληπτά αυτά τα σφάλματα ταξινόμησης, θα εξετάσουμε αναλυτικότερα τους όρους σφάλματος που οφείλονται στις χαλαρές μεταβλητές.

Σχήμα 6-4

Το σημείο $(\overline{x}_j,+1)$ με $\xi_j\le 1$ βρίσκεται πάνω από την επιφάνεια απόφασης και ταξινομείται σωστά (a), ενώ με $\xi_j>1$ βρίσκεται κάτω από αυτήν και ταξινομείται εσφαλμένα (b).



(a) (b)

Αν σε ένα συγκεκριμένο σημείο $(\overline{x}_j,y_j)$ αντιστοιχεί μια χαλαρή μεταβλητή με τιμή μικρότερη ή ίση της μονάδας $(\xi_j\le 1)$, το σημείο αυτό θα ταξινομηθεί σωστά από τη συνάρτηση απόφασης, έστω κι αν βρίσκεται εντός του περιθωρίου. Αν από την άλλη μεριά, το σημείο έχει μια χαλαρή μεταβλητή με τιμή $\xi_j>1$, θα ταξινομηθεί εσφαλμένα από τη συνάρτηση απόφασης. Ακριβέστερα, αν υποθέσουμε ότι το σημείο είναι το $(\overline{x}_j,+1)$ και ξαναγράψουμε τον περιορισμό της (6.93) στη μορφή:

$$\overline{w}\bullet\overline{x}_j=b+\left(1-\xi_j\right) \tag{6.97}$$

το σημείο θα βρίσκεται πάνω από την επιφάνεια απόφασης $\overline{w}\bullet\overline{x}=b$ όσο η ποσότητα $1-\xi_j\ge 0$, δηλαδή όσο $\xi_j\le 1$, αλλιώς θα βρίσκεται κάτω από αυτήν. Αυτό ακριβώς απεικονίζεται στο Σχήμα 6-4. Η εισαγωγή λοιπόν των χαλαρών μεταβλητών μπορεί να

οδηγήσει το μοντέλο μας στο να υποπέσει σε σφάλματα ταξινόμησης, κάτι όμως που δεν είναι κατ' ανάγκην ανεπιθύμητο όταν δεχτούμε ότι στο σύνολο εκπαίδευσης περιλαμβάνονται παρατηρήσεις που μπορούν να εκληφθούν ως προϊόντα θορύβου.

6.6.1 Η δυϊκή διατύπωση του ταξινομητή εύκαμπτου περιθωρίου

Όπως έχουμε διαπιστώσει κατά την ανάπτυξη του αλγορίθμου των ταξινομητών μεγίστου περιθωρίου, το πρόβλημα στην πρωταρχική του διατύπωση έχει περιορισμένη χρησιμότητα, καθώς δεν επιτρέπει την εφαρμογή του τεχνάσματος πυρήνα που θα επέκτεινε τη χρήση γραμμικών ταξινομητών σε προβλήματα μη γραμμικά. Έτσι, για να γενικεύσουμε την ιδέα του ταξινομητή εύκαμπτου περιθωρίου ώστε να έχει εφαρμογή και σε μη γραμμικά προβλήματα, θα αναπτύξουμε εδώ το δυϊκό κατά Lagrange πρόβλημα. Θα ξεκινήσουμε με την πρωταρχική αντικειμενική συνάρτηση (6.92), με τους περιορισμούς της (6.93) και (6.94), και θα τη ξαναγράψουμε ως λαγκρανζιανή σύμφωνα με την (6.3):

$$L\left(\bar{\alpha},\bar{\beta},\bar{w},\bar{\xi},b\right)=\frac{1}{2}\bar{w}\bullet\bar{w}+C\sum_{i=1}^{l}\xi_i$$

$$-\sum_{i=1}^{l}\alpha_i\left(y_i\left(\bar{w}\bullet\bar{x}_i-b\right)+\xi_i-1\right)-\sum_{i=1}^{l}\beta_i\xi_i. \qquad (6.98)$$

Αυτή η λαγκρανζιανή συνάρτηση περιέχει μία πρόσθετη πρωταρχική μεταβλητή $\bar{\xi}$, λόγω των χαλαρών μεταβλητών και μια πρόσθετη δυϊκή, τη $\bar{\beta}=(\beta_1,\ldots,\beta_l)$ που περιλαμβάνει τους πολλαπλασιαστές Lagrange για τους περιορισμούς $\xi_i\geq 0$. Το λαγκρανζιανό πρόβλημα βελτιστοποίησης λαμβάνει λοιπόν τη μορφή:

$$\max_{\bar{\alpha},\bar{\beta}}\min_{\bar{w},\bar{\xi},b}L\left(\bar{\alpha},\bar{\beta},\bar{w},\bar{\xi},b\right), \qquad (6.99)$$

υπό τους περιορισμούς

$$\alpha_i\geq 0, \qquad (6.100)$$

$$\beta_i\geq 0 \qquad (6.101)$$

για $i=1,\ldots,l$.

Καθώς η πρωταρχική αντικειμενική συνάρτηση είναι κυρτή, η λαγκρανζιανή έχει ένα και μοναδικό σαγματικό σημείο και συνεπώς, μια λύση, $\bar{\alpha}^*,\bar{\beta}^*,\bar{w}^*,\bar{\xi}^*,b^*$, που θα πρέπει να ικανοποιεί τις συνθήκες ΚΚΤ.

$$\frac{\partial L}{\partial \overline{w}}\left(\overline{\alpha},\overline{\beta},\overline{w}^*,\overline{\xi},b\right)=\overline{0}, \tag{6.102}$$

$$\frac{\partial L}{\partial \xi_i}\left(\overline{\alpha},\overline{\beta},\overline{w},\xi_i^*,b\right)=0, \tag{6.103}$$

$$\frac{\partial L}{\partial b}\left(\overline{\alpha},\overline{\beta},\overline{w},\overline{\xi},b^*\right)=0, \tag{6.104}$$

$$\alpha_i^*\left(y_i\left(\overline{w}^*\bullet\overline{x}_i-b^*\right)+\xi_i^*-1\right)=0, \tag{6.105}$$

$$\beta_i^*\xi_i^*=0, \tag{6.106}$$

$$y_i\left(\overline{w}^*\bullet\overline{x}_i-b^*\right)+\xi_i^*-1\geq 0, \tag{6.107}$$

$$\alpha_i^*\geq 0, \tag{6.108}$$

$$\beta_i^*\geq 0, \tag{6.109}$$

$$\xi_i^*\geq 0 \tag{6.110}$$

για $i=1,\ldots,l$.

Οι τρεις πρώτες σχέσεις διασφαλίζουν ότι οι λύσεις για τις πρωταρχικές μεταβλητές βρίσκονται στο σαγματικό σημείο της λαγκρανζιανής. Οι σχέσεις (6.105) και (6.106) είναι οι συνθήκες συμπληρωματικότητας. Οι τελευταίες τέσσερεις σχέσεις αποτελούν τους περιορισμούς του πρωταρχικού και του λαγκρανζιανού προβλήματος. Μπορούμε λοιπόν και πάλι, μέσω της βελτιστοποίησης κατά Lagrange, να επιλύσουμε το πρωταρχικό μας πρόβλημα βελτιστοποίησης

$$\begin{aligned}\max_{\overline{\alpha},\overline{\beta}}\min_{\overline{w},\overline{\xi},b} L\left(\overline{\alpha},\overline{\beta},\overline{w},\overline{\xi},b\right)&=L\left(\overline{\alpha}^*,\overline{\beta}^*,\overline{w}^*,\overline{\xi}^*,b^*\right)\\&=\frac{1}{2}\overline{w}^*\bullet\overline{w}^*+C\sum_{i=1}^{l}\xi_i^*.\end{aligned}$$

Όπως και στην περίπτωση του άκαμπτου περιθωρίου, μπορούμε να λύσουμε το πρόβλημα αυτό υπολογίζοντας τη δυϊκή κατά Lagrange συνάρτηση. Εφαρμόζοντας τις συνθήκες ΚΚΤ, αρχικά θα υπολογίσουμε τις μερικές παραγώγους της L ως προς τις πρωταρχικές μεταβλητές στο σαγματικό σημείο της λαγκρανζιανής.

$$\frac{\partial L}{\partial \overline{w}}\left(\overline{\alpha},\overline{\beta},\overline{w}^*,\overline{\xi},b\right)=\overline{w}^*-\sum_{i=1}^{l}\alpha_i y_i\overline{x}_i=\overline{0}, \tag{6.111}$$

$$\overline{w}^*=\sum_{i=1}^{l}\alpha_i y_i\overline{x}_i \tag{6.112}$$

$$\frac{\partial L}{\partial b}\left(\bar{\alpha},\bar{\beta},\bar{w},\bar{\xi},b^*\right)=\sum_{i=1}^{l}\alpha_i y_i=0. \tag{6.113}$$

Παρατηρούμε ότι τα αποτελέσματα ως εδώ δε διαφέρουν από την περίπτωση του άκαμπτου περιθωρίου. Τέλος παραγωγίζοντας τη λαγκρανζιανή ως προς κάθε μία από τις χαλαρές μεταβλητές ξ_i και υπολογίζοντας τις παραγώγους στο ξ_i^*

$$\frac{\partial L}{\partial \xi_i}\left(\bar{\alpha},\bar{\beta},\bar{w},\xi_i^*,b\right)=C-\alpha_i-\beta_i=0, \tag{6.114}$$

λαμβάνουμε τους νέους περιορισμούς,

$$\alpha_i=C-\beta_i,\ \ i=1,\ldots,l. \tag{6.115}$$

Εισάγοντας τους όρους που προέκυψαν από τις μερικές παραγωγίσεις πίσω στη λαγκρανζιανή και εφαρμόζοντας τους περιορισμούς, λαμβάνουμε τη δυϊκή κατά Lagrange συνάρτηση

$$\phi'\left(\bar{\alpha}\right)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \bar{x}_i\bullet\bar{x}_j. \tag{6.116}$$

Αξίζει να σημειωθεί ότι η παραπάνω αντικειμενική συνάρτηση έχει ακριβώς την ίδια δομή όπως η δυϊκή συνάρτηση του ταξινομητή άκαμπτου περιθωρίου που δόθηκε στην (6.36). Το συμπέρασμα που συνάγεται είναι ότι η φύση του προβλήματος βελτιστοποίησης δεν έχει αλλάξει –μόνο οι περιορισμοί άλλαξαν. Τώρα έχουμε επιπλέον τους περιορισμούς της σχέσης (6.115) λόγω της πρόσθετης πρωταρχικής μεταβλητής $\bar{\xi}$. Λαμβάνοντας υπόψη ότι οι πολλαπλασιαστές Lagrange δεν μπορούν να λάβουν αρνητικές τιμές, μπορούμε να γράψουμε τους περιορισμούς αυτούς ως εξής:

$$0\le\alpha_i\le C. \tag{6.117}$$

$$0\le\beta_i\le C. \tag{6.118}$$

Συνεπώς, το δυϊκό πρόβλημα βελτιστοποίησης για ταξινομητές εύκαμπτου περιθωρίου διατυπώνεται όπως στην ακόλουθη πρόταση:

***Πρόταση 6.3* (*Δυϊκό κατά Lagrange πρόβλημα Εύκαμπτου Περιθωρίου*)** *Δοθέντος προβλήματος βελτιστοποίησης εύκαμπτου περιθωρίου, όπως αυτό της Πρότασης 6.2, το κατά Lagrange δυϊκό του πρόβλημα είναι το εξής:*

$$\max_{\bar{\alpha}}\phi'\left(\bar{\alpha}\right)=\max_{\bar{\alpha}}\left(\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \bar{x}_i\bullet\bar{x}_j\right), \tag{6.119}$$

υπό τους περιορισμούς

$$\sum_{i=1}^{l} \alpha_i y_i = 0, \tag{6.120}$$

$$C \geq \alpha_i \geq 0, \quad i = 1, \ldots, l \tag{6.121}$$

όπου C *η σταθερά κόστους.*

Είναι αξιοσημείωτο το γεγονός ότι η μοναδική διαφορά μεταξύ του προβλήματος βελτιστοποίησης μεγίστου περιθωρίου της Πρότασης 6.1 και του προβλήματος βελτιστοποίησης εύκαμπτου περιθωρίου που διατυπώθηκε εδώ, είναι ότι οι πολλαπλασιαστές Lagrange του τελευταίου περιορίζονται από την τιμή της σταθεράς κόστους C.

Για την ερμηνεία αυτού του αποτελέσματος, ας δούμε τη συνθήκη συμπληρωματικότητας (6.106) και τους περιορισμούς της (6.115). Η συνθήκη συμπληρωματικότητας εξασφαλίζει ότι για ένα σημείο $\overline{x}_i$ με μη μηδενική χαλαρή μεταβλητή $\xi_i > 0$, ο αντίστοιχος πολλαπλασιαστής Lagrange θα πρέπει να είναι μηδέν $(\beta_i = 0)$. Από τον περιορισμό (6.115) προκύπτει ότι $\alpha_i = C$. Δηλαδή, για κάθε σημείο που βρίσκεται στη λάθος πλευρά του αντίστοιχου υπερεπιπέδου στήριξης, ο πολλαπλασιαστής Lagrange έχει ως όριο την τιμή C, δηλαδή η επίδραση του σημείου στην επιφάνεια απόφασης περιορίζεται από την τιμή αυτής της σταθεράς. Από την άλλη μεριά, αν στο σημείο $\overline{x}_i$ αντιστοιχεί μια μηδενική χαλαρή μεταβλητή, $\xi_i = 0$ το β_i μπορεί να λάβει οποιαδήποτε τιμή ώστε να ισχύει $0 < \beta_i \leq C$. Από τον περιορισμό (6.115) προκύπτει τότε ότι $0 \leq \alpha_i < C$. Αυτό σημαίνει ότι για σημεία $\overline{x}_i$ με μηδενική χαλαρή μεταβλητή που είναι διανύσματα υποστήριξης, το εύρος των αντίστοιχων πολλαπλασιαστών Lagrange είναι $0 < \alpha_i < C$.

Είναι ξεκάθαρο ότι όσο μεγαλύτερη τιμή δίνουμε στη σταθερά C, τόσο μεγαλύτερη είναι και η επίδραση που έχουν στη θέση της επιφάνειας απόφασης τα σημεία που βρίσκονται στη λάθος πλευρά του αντίστοιχου υπερεπιπέδου στήριξης και αντίστροφα. Μάλιστα η συμπεριφορά αυτή της σταθεράς C δε διαφοροποιείται μεταξύ της πρωταρχικής και της δυϊκής μορφής του προβλήματος βελτιστοποίησης: Μεγάλες τιμές του C περιορίζουν τη θέση της επιφάνειας απόφασης, ενώ μικρές τιμές του C δίνουν μεγαλύτερη ελευθερία στην τοποθέτησή της.

Θα πρέπει να είμαστε ιδιαίτερα προσεκτικοί κατά τον υπολογισμό του βέλτιστου όρου μετάθεσης b^* στην περίπτωση του ταξινομητή εύκαμπτου περιθωρίου. Στην περίπτωση του άκαμπτου περιθωρίου είχαμε την ευχέρεια της επιλογής ενός οποιουδήποτε διανύσματος

υποστήριξης και βάσει αυτού να υπολογίσουμε τον όρο μετάθεσης. Στην περίπτωση του εύκαμπτου περιθωρίου μπορούμε και πάλι να επιλέξουμε ένα διάνυσμα υποστήριξης για την εκτέλεση των υπολογισμών, αλλά θα πρέπει να αποφύγουμε διανύσματα υποστήριξης των οποίων ο αντίστοιχος πολλαπλασιαστής Lagrange ισούται με την τιμή της σταθεράς C, καθώς αυτό υποδηλώνει ότι στο συγκεκριμένο διάνυσμα αντιστοιχεί μία μη μηδενική χαλαρή μεταβλητή και άρα αυτό βρίσκεται στη λάθος πλευρά του αντίστοιχου υπερεπιπέδου στήριξης. Μια τέτοια επιλογή θα οδηγούσε σε εσφαλμένη τιμή για το b^*. Κατά συνέπεια μπορούμε να επιλέξουμε ένα σημείο $(\overline{x}_{sv+}, +1)$ με μηδενική χαλαρή μεταβλητή $\xi^*_{sv+} = 0$ του οποίου ο αντίστοιχος πολλαπλασιαστής Lagrange α^*_{sv+} να έχει τιμή στο εύρος $0 < \alpha^*_{sv+} < C$. Το σημείο αυτό θα βρίσκεται στο υπερεπίπεδο στήριξης αφού ο αντίστοιχος πολλαπλασιαστής Lagrange είναι μεγαλύτερος του μηδενός και ταυτόχρονα μικρότερος από την τιμή της σταθεράς κόστους, ενώ το $\xi^*_{sv+} = 0$. Από τη συνθήκη συμπληρωματικότητας (6.105), σε συνδυασμό με την (6.112), έχουμε

$$\sum_{i=1}^{l} \alpha_i^* y_i \overline{x}_i \bullet \overline{x}_{sv+} - b^* - 1 = 0 \tag{6.122}$$

και λύνοντας ως προς b^* λαμβάνουμε

$$b^* = \sum_{i=1}^{l} \alpha_i^* y_i \overline{x}_i \bullet \overline{x}_{sv+} - 1. \tag{6.123}$$

Αν επιλεγεί προσεκτικά το διάνυσμα υποστήριξης, ο υπολογισμός για το b^* είναι ακριβώς ίδιος με αυτόν του άκαμπτου περιθωρίου [βλ. (6.44)].

Η συνάρτηση απόφασης επίσης δεν επηρεάζεται από τη χρήση των χαλαρών μεταβλητών και είναι ίδια με αυτή της περίπτωσης άκαμπτου περιθωρίου [βλ. (6.47)]. Από τη σκοπιά του μοντέλου, η εισαγωγή των χαλαρών μεταβλητών σημαίνει απλά ότι οι πολλαπλασιαστές Lagrange α_i περιορίζονται από την τιμή της μεταβλητής C, χωρίς να υπάρχει καμία άλλη επίδραση στη δομή του μοντέλου. Αυτό σημαίνει ότι η μόνη διαφορά μεταξύ μηχανών διανυσμάτων υποστήριξης άκαμπτου περιθωρίου και αυτών του εύκαμπτου περιθωρίου είναι ότι οι τελευταίοι έχουν μία επιπλέον ελεύθερη παράμετρο, τη σταθερά κόστους C, για την οποία ο χρήστης θα πρέπει να επιλέξει τιμή πριν την εκκίνηση του αλγορίθμου εκπαίδευσης.

Τέλος, παρατηρήστε ότι, τόσο ο αλγόριθμος εκπαίδευσης του ταξινομητή εύκαμπτου περιθωρίου που δίνεται στην Πρόταση 6.3, όσο και το μοντέλο που δίνεται στην (6.47) εκφράζονται με όρους εσωτερικών γινομένων μεταξύ σημείων του χώρου εισόδου $\mathbb{R}^n$. Αυτό

σημαίνει ότι μπορεί να εφαρμοστεί το τέχνασμα του πυρήνα στη δυϊκή διατύπωση του ταξινομητή εύκαμπτου περιθωρίου και να κατασκευαστούν μη γραμμικοί ταξινομητές εύκαμπτου περιθωρίου. Έχει ίσως ενδιαφέρον να σημειωθεί ότι συγκεκριμένα σύνολα δεδομένων, μη γραμμικώς διαχωρίσιμα, μπορούν να αντιμετωπιστούν με γραμμικό ταξινομητή εύκαμπτου περιθωρίου, εφόσον τα σημεία που προκαλούν τη μη γραμμικότητα θεωρούνται ως αποτέλεσμα θορύβου.

6.7 Χρήση του Ταξινομητή SVM με Πακέτα Λογισμικού

Για την επίλυση του προβλήματος βελτιστοποίησης που απαιτεί ο ταξινομητής SVM, σε αντίθεση με την περίπτωση του δικτύου MLP ή του μοντέλου λογιστικής παλινδρόμησης, δεν προτείνεται σε καμία περίπτωση να επιχειρηθεί ανάπτυξη κώδικα από πλευράς χρήστη. Και αυτό γιατί ο αλγόριθμος, πέραν της πολυπλοκότητάς του, απαιτεί την εφαρμογή διαφόρων τεχνικών αριθμητικής βελτιστοποίησης, προκειμένου να καταστεί αποδοτικός.

Κατά τον ίδιο τρόπο λοιπόν που δε θα σκεφτόμαστε ποτέ να υλοποιήσουμε μια συνάρτηση που να αντιστρέφει ένα πίνακα ή που να υπολογίζει μια τετραγωνική ρίζα, έτσι και προκειμένου να χρησιμοποιήσουμε τον ταξινομητή SVM πάντα καταφεύγουμε σε έτοιμες ρουτίνες, οι οποίες έχουν δοκιμαστεί για χρόνια προκειμένου να βελτιστοποιηθούν. Παραδείγματα τέτοιων λογισμικών αποτελούν τα ακόλουθα:

- LIBLINEAR[1]
- LIBSVM[2]
- SVMLight[3]

Αυτό που πρέπει να γίνει από το χρήστη είναι η επιλογή του πυρήνα και της παραμέτρου C. Ως προς την επιλογή πυρήνα, στη συντριπτική πλειονότητα των περιπτώσεων καταφεύγουμε είτε στον γκαουσιανό πυρήνα, είτε στη μη χρήση πυρήνα (που όπως έχει ήδη ειπωθεί, αναφέρεται και ως γραμμικός πυρήνας). Για την επιλογή, κύριο ρόλο παίζει η σχέση μεταξύ του μεγέθους του δείγματος m και του πλήθους των χαρακτηριστικών n. Όσο μικρότερο είναι το n σε σχέση με το m, τόσο περισσότερο αποτελεσματικός είναι ο

[1] http://www.csie.ntu.edu.tw/~cjlin/liblinear/
[2] http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[3] http://svmlight.joachims.org/

γκαουσιανός πυρήνας. Αντίθετα, αν το n είναι μεγάλο σε σχέση με το m, προτιμάται ο γραμμικός πυρήνας.

Στην περίπτωση που οι μονάδες μέτρησης των χαρακτηριστικών διαφέρουν μεταξύ τους, είναι σημαντικό να προηγηθεί *τυποποίηση των τιμών των χαρακτηριστικών* πριν εφαρμοστεί ο αλγόριθμος, προκειμένου να αποφευχθεί η κυριαρχία των χαρακτηριστικών με μεγαλύτερα εύρη τιμών επί αυτών με μικρότερα εύρη. Στην περίπτωση αυτή θα πρέπει να δοθεί ιδιαίτερη προσοχή ώστε να χρησιμοποιηθούν οι ίδιοι σταθμικοί συντελεστές, τόσο στα σύνολα εκπαίδευσης, επικύρωσης και ελέγχου[1], όσο και σε νέα δεδομένα για τα οποία μελλοντικά θα ζητηθούν προβλέψεις. Επίσης, καθώς ο SVM απαιτεί τα χαρακτηριστικά να είναι εκφρασμένα ως πραγματικοί αριθμοί, εάν υπάρχουν κατηγορικές μεταβλητές θα πρέπει αυτές να μετατραπούν σε αριθμητικές στη φάση της προ-επεξεργασίας των δεδομένων (συνήθως μια μεταβλητή k-κατηγοριών αντικαθίσταται από k νέες δυαδικές).

Προκειμένου να καταλήξουμε στη βέλτιστη επιλογή τιμών για τις παραμέτρους C και σ, θα πρέπει να καταφύγουμε σε δοκιμές με τη βοήθεια ενός συνόλου δεδομένων επικύρωσης. Και για τις δύο αυτές παραμέτρους προτείνεται να δοκιμαστούν τιμές με πολλαπλασιαστική σχέση μεταξύ τους, π.χ. $(0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30)$ και ο έλεγχος να γίνει για κάθε δυνατό ζεύγος τιμών. Εν τέλει θα επιλεγεί το ζεύγος στο οποίο θα αντιστοιχεί το μικρότερο ποσοστό εσφαλμένων ταξινομήσεων.

Αν κάποιος επιθυμεί να κάνει χρήση της Octave (ή του MATLAB) ενδέχεται να χρειαστεί, ανάλογα και με τη ρουτίνα SVM που θα χρησιμοποιήσει, να υλοποιήσει και την εκάστοτε συνάρτηση-πυρήνα.

Στα αρχεία που συνοδεύουν την εφαρμογή της Ενότητας 6.10 περιλαμβάνεται, κυρίως για εκπαιδευτικούς σκοπούς, μια απλοποιημένη υλοποίηση του αλγορίθμου SMO σε Octave με όνομα ***svmTrain*** [25], καθώς επίσης και η απαιτούμενη συνάρτηση-πυρήνας με όνομα ***gaussianKernel***, η οποία έχει την παρακάτω μορφή:

```
function sim = gaussianKernel(x1, x2, sigma)
%Επιστρέφει τον Radial Basis Function kernel των διανυσμάτων x1 και x2

% Εξασφάλισε ότι τα x1 και x2 είναι διανύσματα στήλη
x1 = x1(:); x2 = x2(:);
sim = exp(-norm(x1-x2)^2/2/sigma^2);
end
```

[1] Βλ. Κεφ. 7

Στην πράξη βέβαια είναι προτιμότερο να εγκατασταθεί και να χρησιμοποιείται και στην Octave μια βελτιστοποιημένη βιβλιοθήκη όπως η LIBSVM.

Στα αρχεία που συνοδεύουν την εφαρμογή της Ενότητας 6.10 περιλαμβάνεται μεταξύ άλλων, μια συνάρτηση με όνομα ***LIBSVM_Params***, μέσω της οποίας επιτυγχάνεται η βέλτιστη επιλογή τιμών για τις παραμέτρους C και σ στην Octave (αφού προηγουμένως έχει εγκατασταθεί η βιβλιοθήκη LIBSVM, σύμφωνα με τις οδηγίες στην ιστοσελίδα που έχει δοθεί). Επίσης περιλαμβάνεται η συνάρτηση ***svmParams***, η οποία είναι αντίστοιχη με την προηγούμενη, με τη διαφορά ότι κάνει χρήση της svmTrain για την επιλογή βέλτιστων τιμών για τις παραμέτρους C και σ.

Στο R project, η ρουτίνα ***svm*** του πακέτου ***e1071***[1] αποτελεί μια διασύνδεση με τη βιβλιοθήκη LIBSVM, ενώ το πακέτο ***kernlab***[2] παρέχει ένα ευέλικτο πλαίσιο εκπαίδευσης μέσω μεθόδων πυρήνα και βασίζεται στις βιβλιοθήκες LIBSVM και BSVM[3]. Για ταξινόμηση με χρήση του αλγορίθμου SVM στο πακέτο kernlab, χρησιμοποιείται η ρουτίνα ***ksvm***. Επίσης πρόσβαση στην υλοποίηση SVMlight (μόνο για ταξινόμηση one-vs-all) παρέχεται μέσω του πακέτου ***klaR***[4].

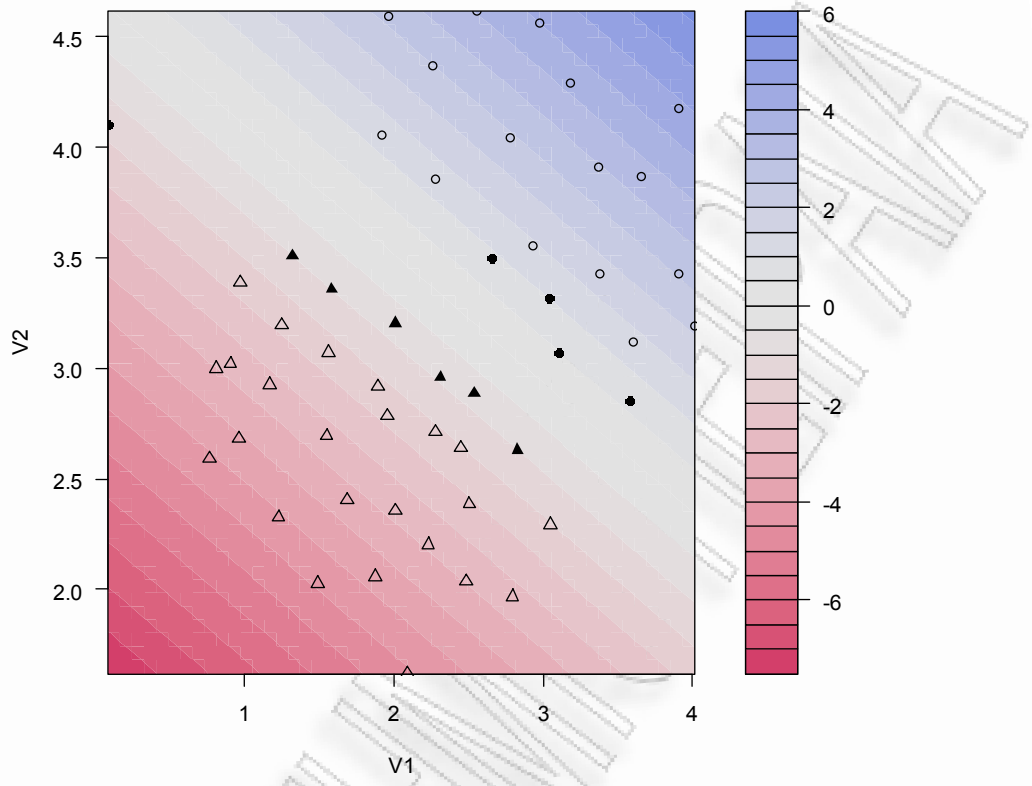
Για παράδειγμα, στην περίπτωση δυαδικής ταξινόμησης με δύο διαθέσιμα χαρακτηριστικά όπως στο Σχήμα 6-3, μπορούμε μέσω του πακέτου kernlab της R να εργαστούμε ως εξής:

```
# Φόρτωσε το πακέτο "Kernel-based Machine Learning Lab"
library(kernlab)

# V3: η μεταβλητή απόκρισης στο dataframe ds, τύπου factor
# Η επιλογή vanilladot υποδηλώνει Linear kernel
svp <- ksvm(V3~V2+V1, data=ds, kernel="vanilladot", C=1)

# Χρήση του μοντέλου για πρόβλεψη
pred <- predict(svp, ds, type="response")

# Contour plot
plot(svp,data=ds)
```

Με την τελευταία εντολή λαμβάνουμε γράφημα με τις ισοϋψείς (contour plot) των τιμών απόφασης, στο οποίο καταδεικνύονται με μαύρο χρώμα και τα διανύσματα υποστήριξης (βλ. Σχήμα 6-5).

[1] e1071: http://cran.r-project.org/web/packages/e1071/index.html
[2] kernlab: Kernel-based Machine Learning Lab, http://cran.r-project.org/web/packages/kernlab/index.html
[3] http://www.csie.ntu.edu.tw/~cjlin/bsvm
[4] klaR: Classification and visualization, http://cran.r-project.org/web/packages/klaR/index.html

Σχήμα 6-5
SVM Contour plot



Για τη βέλτιστη επιλογή των παραμέτρων C και σ στην R υπάρχουν διαφορετικές επιλογές ανάλογα με τον αλγόριθμο που εφαρμόζουμε. Κατ' αρχάς μπορούμε πάντα να δημιουργήσουμε μια συνάρτηση σαν την LIBSVM_Params που αναφέρθηκε προηγουμένως. Στην περίπτωση που χρησιμοποιούμε το πακέτο **kernlab**, μπορούμε να ζητήσουμε αυτόματη εκτίμηση της βέλτιστης τιμής για την παράμετρο σ, δίνοντας *kpar = "automatic"* (που αποτελεί και την default επιλογή).

Αν γίνεται χρήση του πακέτου **e1071** μπορεί να χρησιμοποιηθεί η ρουτίνα ***tune***. Στη συνέχεια με την **plot.tune** μπορούμε να δούμε και οπτικά για ποια ζευγάρια τιμών έχουμε το μικρότερο ποσοστό εσφαλμένων ταξινομήσεων.

Για παράδειγμα, χρησιμοποιώντας το γνωστό σετ δεδομένων ***iris***[1]:

```
library(e1071)
data(iris)
obj <- tune.svm(Species~., data = iris,
                gamma = 2^c(-8,-4,0,4), cost = 2^c(-8,-4,-2,0))
plot(obj, transform.x = log2, transform.y = log2)
```

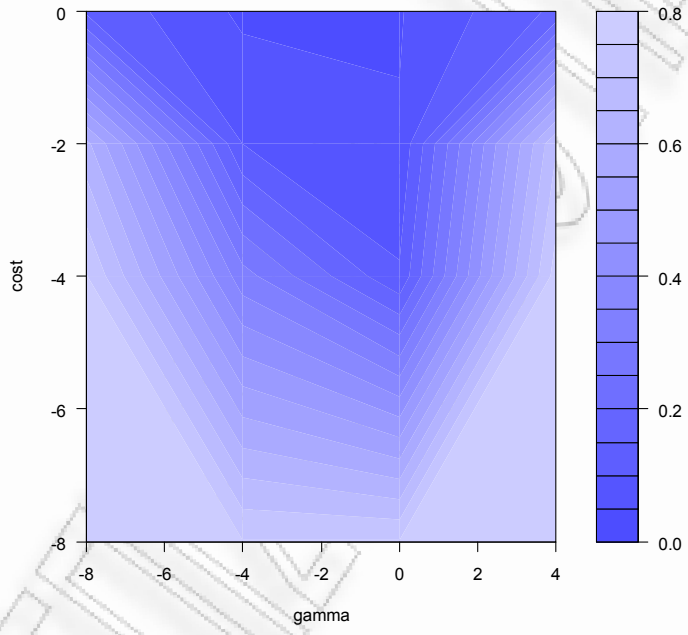
[1] http://finzi.psych.upenn.edu/R/library/datasets/html/iris.html

λαμβάνουμε τις βέλτιστες τιμές των παραμέτρων:

```
- best parameters:
  gamma cost
 0.0625    1
```

Με την εντολή plot λαμβάνουμε το γράφημα που φαίνεται στο Σχήμα 6-6, για μια καλύτερη οπτική απεικόνιση των αποτελεσμάτων. Όσο μεγαλύτερη είναι η ένταση του μπλε τόσο καλύτερος ο συνδυασμός τιμών.

Σχήμα 6-6
Βέλτιστες τιμές C και gamma



Σημειώνεται ότι σε διάφορες υλοποιήσεις αντί για την παράμετρο σ μπορεί να γίνεται αναφορά στην παράμετρο *gamma*. Γενικά ισχύει ότι: $gamma = 1/(2\sigma^2)$

Τέλος, ρύθμιση των παραμέτρων με ταυτόχρονη σύγκριση μεταξύ μοντέλων μπορεί να επιτευχθεί στην R μέσω του πακέτου ***caret***. Παράδειγμα της χρήσης του θα παρουσιαστεί στην Ενότητα 6.11.

Από την άλλη μεριά, στο WEKA για την εκπαίδευση του ταξινομητή SVM υπάρχει προεγκατεστημένη μια υλοποίηση του αλγορίθμου SMO [28], ενώ για τη χρήση των βιβλιοθηκών LIBSVM και LIBLINEAR θα πρέπει να εγκατασταθούν οι αντίστοιχες *wrapper* κλάσεις για το WEKA, οι οποίες διατίθενται στις αντίστοιχες ιστοσελίδες που αναφέρθηκαν

προηγουμένως. Η χρήση των δυο τελευταίων προτείνεται ανεπιφύλακτα έναντι του SMO, καθώς είναι *κατά πολύ* ταχύτερες.

Για να γίνουν ορατές οι κλάσεις αυτές από την εφαρμογή, θα πρέπει να βρίσκονται στο CLASSPATH[1]. Ο απλούστερος τρόπος να γίνει αυτό είναι μέσω μιας μικρής παρέμβασης στο αρχείο **RunWeka.ini**. Συγκεκριμένα, αν τα αρχεία με κατάληξη **.jar** έχουν για παράδειγμα τοποθετηθεί στον κατάλογο *C:/libraries*, προσθέτουμε μια γραμμή όπως η παρακάτω στο τέλος του αρχείου RunWeka.ini, το οποίο βρίσκεται στον κατάλογο που έχει εγκατασταθεί το WEKA:

```
cp=%CLASSPATH%;C:/libraries/liblinear-1.7-with-deps.jar;C:/libraries/libsvm.jar;C:/libraries/wlsvm.jar
```

Ως προς τη βελτιστοποίηση των τιμών των παραμέτρων με χρήση του WEKA, ο αναγνώστης μπορεί να βρει αναλυτικές πληροφορίες στη ***WEKA Wiki*** [2].

6.8 Παράδειγμα Χρήσης της LIBSVM με Gaussian Kernel

Το σύνολο των δεδομένων εκπαίδευσης που θα χρησιμοποιηθεί αποτελείται από δύο μόνον χαρακτηριστικά, για λόγους εποπτείας. Το αρχείο δεδομένων είναι σε μορφή .mat, πρόκειται δηλαδή για το αποτέλεσμα εξαγωγής των μεταβλητών από ένα workspace της Octave ή του Matlab. Το αρχείο αυτό είναι διαθέσιμο μέσω του συνδέσμου http://bit.ly/sgvMHv.

Στην επόμενη παράγραφο θα παρουσιαστεί μια, ακόμη πιο ολοκληρωμένη, αντιμετώπιση αντίστοιχου παραδείγματος, η οποία θα περιλαμβάνει βέλτιστη επιλογή των παραμέτρων C και σ, και μάλιστα με τρία διαφορετικά λογισμικά (Octave, R και WEKA).

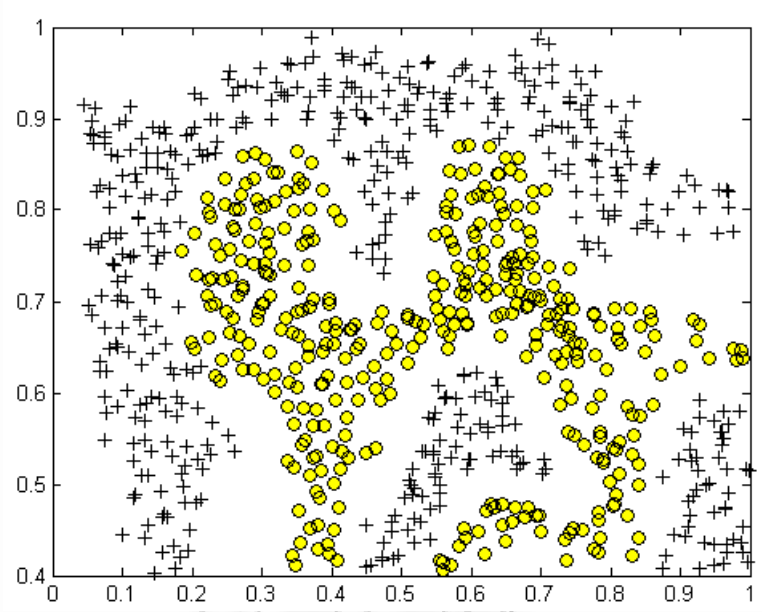
```
% Φόρτωμα αρχείου
% Θα δημιουργηθούν οι μεταβλητές X, y στο περιβάλλον εργασίας
load('ex6data2.mat');

% Εκτύπωση δεδομένων (βλέπε ορισμό της plotData παρακάτω)
plotData(X, y);
```

[1] Αυτό που κάνει το classpath είναι να υποδεικνύει στην εικονική μηχανή (JVM) πού μπορεί να βρει τις απαραίτητες κλάσεις ώστε να τρέξει ένα πρόγραμμα. Ένα πρόγραμμα Java αποτελείται εξ' ολοκλήρου από κλάσεις (class). Οτιδήποτε δηλώνεται και εκτελείται βρίσκεται μέσα σε μία ή περισσότερες κλάσεις.

[2] Π.χ. http://weka.wikispaces.com/Optimizing+parameters

Με την τελευταία εντολή εμφανίζεται το γράφημα των σημείων στο επίπεδο. Τα σημεία με χαρακτηρισμό $y = 0$ εμφανίζονται με κίτρινους κύκλους, ενώ αυτά με $y = 1$ εμφανίζονται με μαύρους σταυρούς (βλ. Σχήμα 6-7).

Σχήμα 6-7
Παράδειγμα χρήσης LIBSVM.



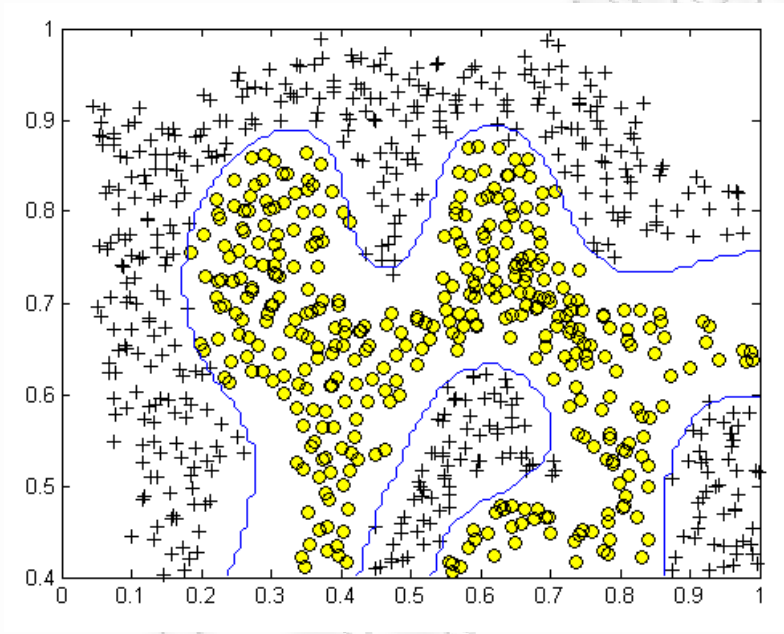
Είναι φανερό ότι δεν υπάρχει γραμμικό όριο απόφασης που να διαχωρίζει τις δύο ομάδες. Ωστόσο, με τη χρήση γκαουσιανού πυρήνα στον αλγόριθμο SVM θα καταφέρουμε να διαμορφώσουμε ένα μη γραμμικό όριο απόφασης που να διαχωρίζει ικανοποιητικά τα δεδομένα.

```
% SVM Parameters (προς το παρόν δεν αναζητούμε βέλτιστες τιμές)
C = 1; sigma = 0.1;

% Χρήση LIBSVM
% Προσθήκη του path για τη βιβλιοθήκη LIBSVM
addpath ('C:\libraries\libsvm-3.11\matlab');

% Δημιουργία μοντέλου
model_LIBSVM=svmtrain(y, X, ['-c ',num2str(C),' -t 2 -g ',num2str(1/2/sigma^2)]);

% Προβλέψεις, επίπεδο ακρίβειας, σφάλμα
[predicted_label, accuracy, decision_values] = svmpredict(y, X, model_LIBSVM);
fprintf('\nTraining Set Error: %f%%\n', mean(double(predicted_label ~= y)) * 100);

% Εκτύπωση δεδομένων με τη διαχωριστική επιφάνεια
% (βλέπε ορισμό της LIBSVMvisualizeBoundary αργότερα)
LIBSVMvisualizeBoundary(X, y, model_LIBSVM)
```

Με την παράμετρο ***-c*** περνάμε την τιμή του κόστους C, με την παράμετρο ***-t 2*** δηλώνουμε ότι επιθυμούμε γκαουσιανό πυρήνα, ενώ με την παράμετρο ***-g*** περνάμε την τιμή gamma[1].

Μέσω της βοηθητικής συνάρτησης LIBSVMvisualizeBoundary λαμβάνουμε το αποτέλεσμα που φαίνεται στο Σχήμα 6-8.

Σχήμα 6-8
SVM, Gaussian kernel ($C = 1, \sigma = 0.1$)



Στη συνέχεια δίνεται ο κώδικας των βοηθητικών συναρτήσεων plotData και LIBSVMvisualizeBoundary:

```
function plotData(X, y)
%   Απεικονίζει τα σημεία με + (για y=1) και ο (για y=0).
%   Το X θεωρείται πίνακας Mx2.

% Βρες τις θέσεις των y=1 και y=0
pos = find(y == 1); neg = find(y == 0);

% Plot
plot(X(pos, 1), X(pos, 2), 'k+','LineWidth', 1, 'MarkerSize', 7)
hold on;
plot(X(neg, 1), X(neg, 2), 'ko', 'MarkerFaceColor', 'y', 'MarkerSize', 7)
hold off;

end
```

[1] Αναλυτικές οδηγίες χρήσης περιλαμβάνονται στο αρχείο README στον κατάλογο όπου έχει εγκατασταθεί η βιβλιοθήκη LIBSVM.

```
function LIBSVMvisualizeBoundary(X, y, model, varargin)
% Απεικονίζει τη μη γραμμική επιφάνεια απόφασης από την εκμάθηση του SVM
% Εμφάνιση των δεδομένων εκπαίδευσης
plotData(X, y)

% Υπολόγισε προβλέψεις επί ενός πλέγματος τιμών
x1plot = linspace(min(X(:,1)), max(X(:,1)), 100)';
x2plot = linspace(min(X(:,2)), max(X(:,2)), 100)';
[X1, X2] = meshgrid(x1plot, x2plot);
vals = zeros(size(X1));
for i = 1:size(X1, 2)
   this_X = [X1(:, i), X2(:, i)];
   vals(:, i) = svmpredict(zeros(100,1), this_X, model);
end
% Απεικόνισε την επιφάνεια
hold on
contour(X1, X2, vals, [0 0], 'Color', 'b');
hold off;
end
```

6.9 Παράδειγμα Επιλογής Βέλτιστων *C* και *σ* (Octave, R, WEKA)

Στο παράδειγμα αυτό, επίσης για λόγους εποπτείας, θα αντιμετωπιστεί ένα σύνολο εκπαίδευσης με δύο μόνον διαθέσιμα χαρακτηριστικά. Σε αντίθεση με το προηγούμενο παράδειγμα, εδώ είναι διαθέσιμο κι ένα δεύτερο σύνολο δεδομένων (validation set) το οποίο θα χρησιμοποιηθεί για τη βέλτιστη επιλογή των παραμέτρων. Το αρχείο δεδομένων είναι επίσης σε μορφή .mat, πρόκειται δηλαδή για το αποτέλεσμα εξαγωγής των μεταβλητών ενός workspace της Octave ή του Matlab. Το αρχείο αυτό είναι διαθέσιμο online, μέσω του συνδέσμου http://bit.ly/s06U3k. Αφού ολοκληρωθεί η αντιμετώπιση με Octave, τα δεδομένα θα αποθηκευτούν σε μορφές κατάλληλες να διαβαστούν από τα άλλα δύο πακέτα.

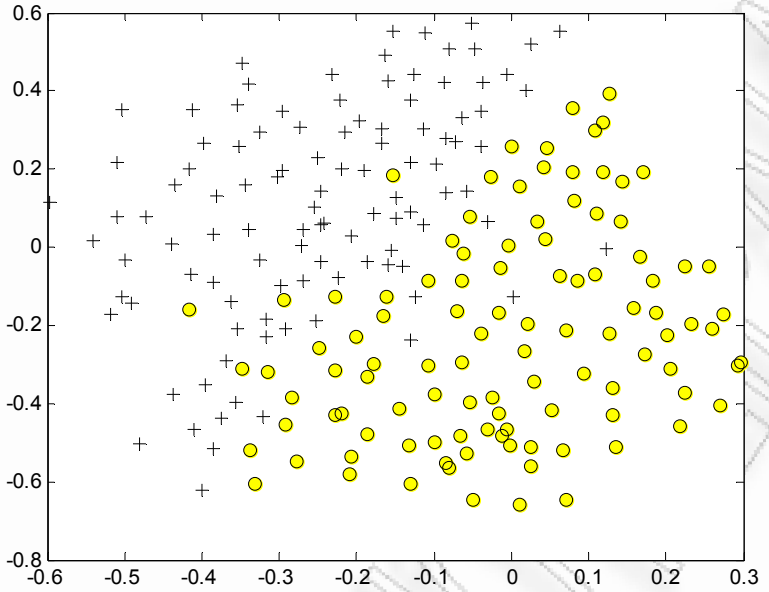
6.9.1 Octave/Matlab

```
% Φόρτωμα αρχείου. (Θα δημιουργηθούν οι μεταβλητές X, y, Xval, yval)
load('ex6data3.mat');

% Plot training data
plotData(X, y);

% Χρήση LIBSVM. % Προσθήκη του path για τη βιβλιοθήκη LIBSVM
addpath ('C:\libraries\libsvm-3.11\matlab');

% Επιλογή βέλτιστων τιμών των παραμέτρων
[C, sigma] = LIBSVM_Params(X, y, Xval, yval);
```

Με την εντολή plotData, η οποία δόθηκε στην προηγούμενη παράγραφο, εμφανίζεται το γράφημα των σημείων στο επίπεδο. Τα σημεία με χαρακτηρισμό $y = 0$ εμφανίζονται με κίτρινους κύκλους, ενώ αυτά με $y = 1$ εμφανίζονται με μαύρους σταυρούς (Σχήμα 6-9).

Σχήμα 6-9

Παράδειγμα για την επιλογή παραμέτρων C, σ

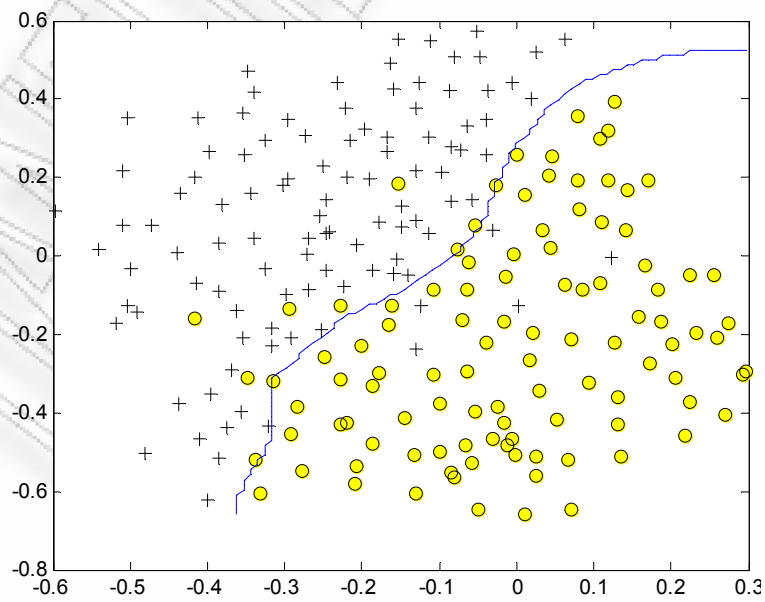


Οι τιμές που επιλέχθηκαν ήταν $C = 1, \quad \sigma = 0.1$.

```
% Δημιουργία μοντέλου (όπως περιγράφηκε στην προηγούμενη παράγραφο)
model = svmtrain(y, X, ['-c ',num2str(C),' -g ',num2str(1/2/sigma^2),' -t 2']);

% Εκτύπωση δεδομένων με τη διαχωριστική επιφάνεια
% (βλέπε ορισμό της LIBSVMvisualizeBoundary στην προηγούμενη παράγραφο)
visualizeBoundaryLIBSVM(X, y, model);
```

Σχήμα 6-10

Διαχωριστική επιφάνεια βάσει των βέλτιστων C, σ (Octave)



Και πάλι μέσω της βοηθητικής συνάρτησης LIBSVMvisualizeBoundary λαμβάνουμε το αποτέλεσμα που φαίνεται στο Σχήμα 6-10.

Η συνάρτηση LIBSVM_Params, ο κώδικας της οποίας δίνεται στη συνέχεια, υπολογίζει το σφάλμα ταξινόμησης που προκύπτει για κάθε συνδυασμό των παραμέτρων C και σ, για τις οποίες ορίζονται αντίστοιχα διανύσματα πιθανών τιμών και επιστρέφει εκείνο το ζευγάρι τιμών για το οποίο ελαχιστοποιείται το σφάλμα.

```
function [C, sigma] = LIBSVM_Params(X, y, Xval, yval)
% επιστρέφει τα βέλτιστα C και sigma για χρήση με RBF kernel,
% βασισμένα στο validation set.

C_vals = [0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30];
sigma_vals = [0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30];
dimC = size(C_vals,2);
res = zeros(size(C_vals,2)*size(sigma_vals,2),3);
for i=1:size(C_vals,2)
    for j=1:size(sigma_vals,2)
       C = C_vals(i);
       sigma=sigma_vals(j);
       model= svmtrain(y,X,['-c ',num2str(C),' -g ',num2str(1/2/sigma^2),' -t 2']);
       [predicted_label,accuracy,decision_values] = svmpredict(yval, Xval, model);
       err = mean(double(predicted_label ~= yval));
       res((i-1)*dimC+j,:)=[C sigma err];
    end
end
[val,ind]=min(res(:,3),[],1);
C=res(ind,1);
sigma=res(ind,2);

% =========================================================================
end
```

6.9.2 R project

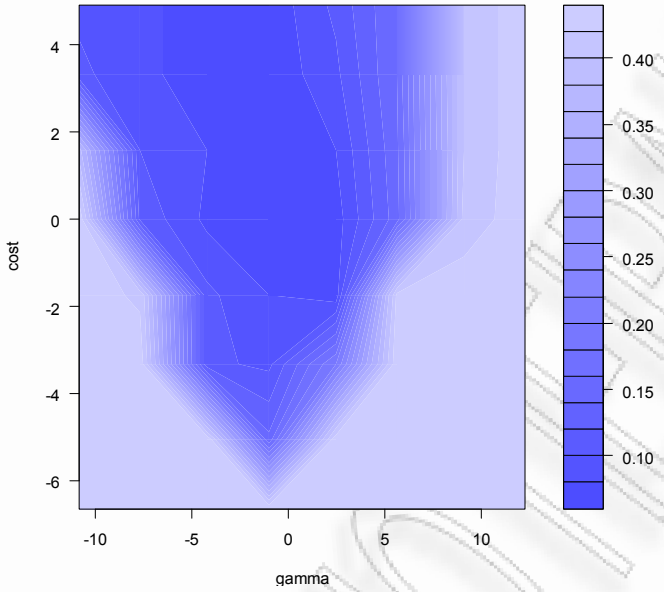
Αρχικά εξάγουμε από το περιβάλλον της Octave τα δεδομένα training και validation σαν δύο αρχεία τύπου csv (comma separated values)

```
% Εξαγωγή δεδομένων σε μορφή csv (από το περιβάλλον της Octave)
csvwrite ('ex6data3_TRAIN.csv', [X y]);
csvwrite ('ex6data3_VAL.csv', [Xval yval]);
```

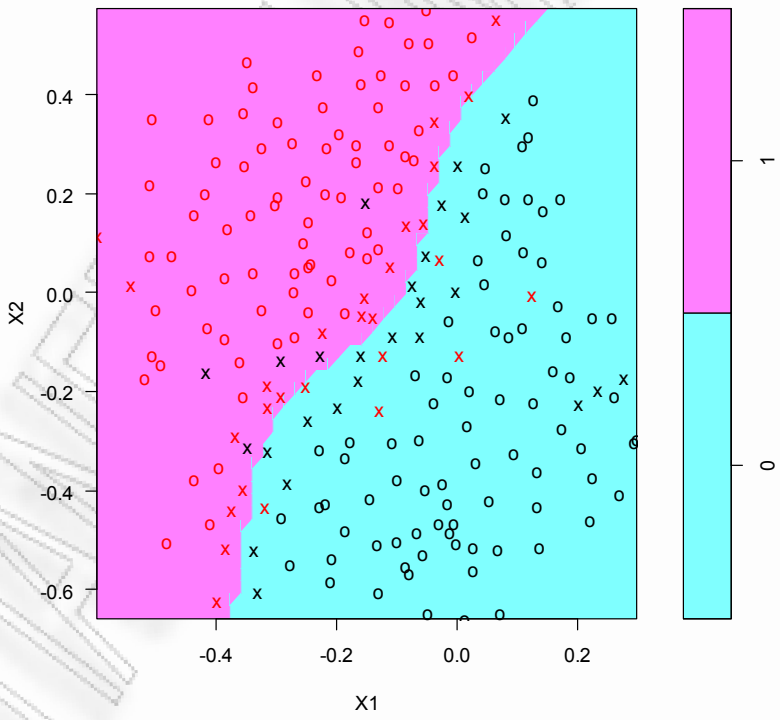
Στο περιβάλλον R πια, και αφού οριστεί ως τρέχων κατάλογος αυτός στον οποίο βρίσκονται τα παραπάνω csv αρχεία:

```
# Εισαγωγή δεδομένων
dataset_train <- read.csv('ex6data3_TRAIN.csv',header=F)
dataset_val <- read.csv('ex6data3_VAL.csv',header=F)

# Προσθήκη κεφαλίδων
names(dataset_train) <- c('X1','X2','Y')
names(dataset_val) <- c('X1','X2','Y')

# Μετατροπή της εξαρτημένης μεταβλητής σε factor
dataset_train$Y <- factor(dataset_train$Y)
dataset_val$Y <- factor(dataset_val$Y)
```

Σχήμα 6-11
Οπτική απεικόνιση των αποτελεσμάτων βελτιστοποίησης



Σχήμα 6-12
Διαχωριστική επιφάνεια βάσει των βέλτιστων C, σ (R project)




```
# Φόρτωμα του πακέτου που θα χρησιμοποιηθεί
library(e1071)
# Ορισμός των τιμών C, gamma μεταξύ των οποίων θα γίνει επιλογή
gamma.vals <- 1/2/c(0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30)^2
cost.vals <- c(0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30)
```

```
# Βελτιστοποίηση C, gamma και οπτική απεικόνιση αποτελεσμάτων
obj <- tune.svm(Y~., data = dataset_val, gamma = gamma.vals, cost = cost.vals)
plot(obj, transform.x = log2, transform.y = log2)

# Δημιουργία μοντέλου και εκτύπωση αποτελεσμάτων ταξινόμησης
m <- svm(Y~X2+X1, data = dataset_train, kernel = "radial",
         gamma = obj$best.parameters$gamma, cost = obj$best.parameters$cost)
plot(m, dataset_train)
```

Οι βέλτιστες τιμές που επιλέχτηκαν μέσω αυτής της διαδικασίας ήταν $C=10,\ \sigma=1$ (είναι αμφότερες δεκαπλάσιες από τις προηγούμενες, ωστόσο η μεταξύ τους σχέση διατηρείται σταθερή). Το αποτέλεσμα της συνάρτησης plot.tune εμφανίζεται στο Σχήμα 6-11, ενώ αυτό της plot.svm, στο Σχήμα 6-12.

6.9.3 WEKA

Κατ' αρχήν θα δημιουργηθούν κατάλληλα αρχεία τύπου ARFF (Attribute-Relation File Format), μέσα από το περιβάλλον της R, κάνοντας χρήση του πακέτου ***foreign***.

```
% Εξαγωγή δεδομένων σε μορφή arff (από το περιβάλλον της R)
library(foreign)

write.arff(dataset_train,'ex6data3_TRAIN.arff')
write.arff(dataset_val,'ex6data3_VAL.arff')
```

Στη συνέχεια από το παράθυρο Explorer του WEKA φορτώνουμε το αρχείο ex6data3_VAL.arff. Στην καρτέλα Classify επιλέγουμε την κλάση επιλογής παραμέτρων μέσω Cross-Validation: ***weka.classifiers.meta.CVParameterSelection.*** Στις παραμέτρους της ορίζουμε ως ταξινομητή τον ***weka.classifiers.functions.LibSVM***. Επιβεβαιώνουμε στις παραμέτρους του ταξινομητή ότι το kerneltype είναι το RBF.

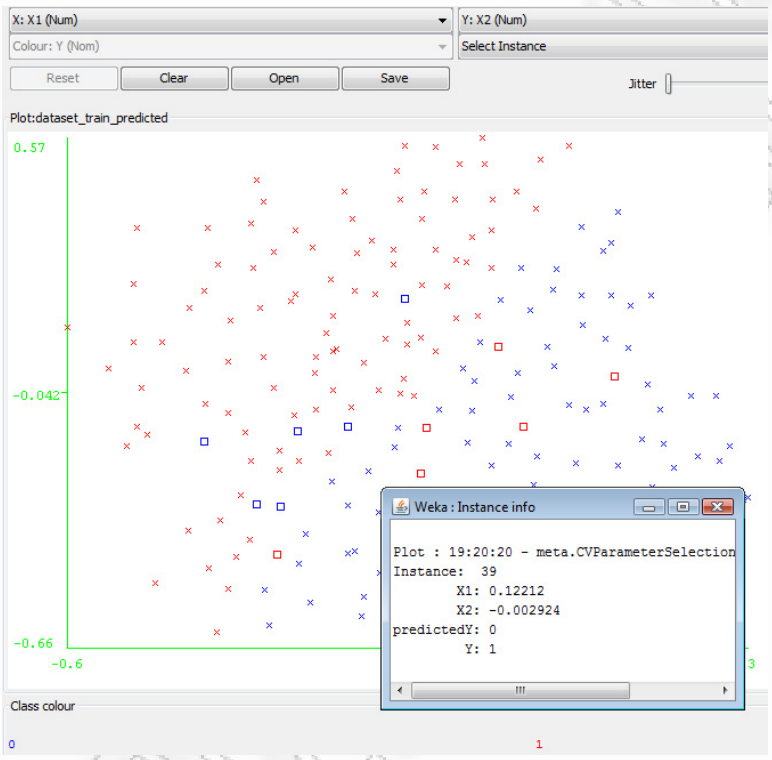
Στο πεδίο CVParameters εισάγουμε τόσες εγγραφές όσες και οι παράμετροι για τις οποίες ζητάμε βέλτιστες τιμές. Αρχικά δίνεται ο κωδικός της παραμέτρου, το εύρος τιμών και τέλος ο αριθμός των βημάτων. Στη περίπτωσή μας δίνουμε π.χ. αντίστοιχα για το C και gamma:

```
C 0.01 30.0 30

G 1.0 5000.0 30
```

Τέλος κλείνουμε το παράθυρο και εκτελούμε τη διαδικασία. Το αποτέλεσμα που λαμβάνουμε είναι παρόμοιο με το παρακάτω. Οι βέλτιστες τιμές που επιλέχθηκαν εμφανίζονται τονισμένες στο παρακάτω πλαίσιο.

```
Cross-validated Parameter selection.
Classifier: weka.classifiers.functions.LibSVM
Cross-validation Parameter: '-G' ranged from 1.0 to 5000.0 with 30.0 steps
Cross-validation Parameter: '-C' ranged from 0.01 to 30.0 with 30.0 steps
Classifier Options: -G 1 -C 24.8293 -S 0 -K 2 -D 3 -R 0.0 -N 0.5 -M 40.0 -E 0.0010 -P 0.1
```

Σχήμα 6-13
WEKA: Απεικόνιση των σφαλμάτων ταξινόμησης



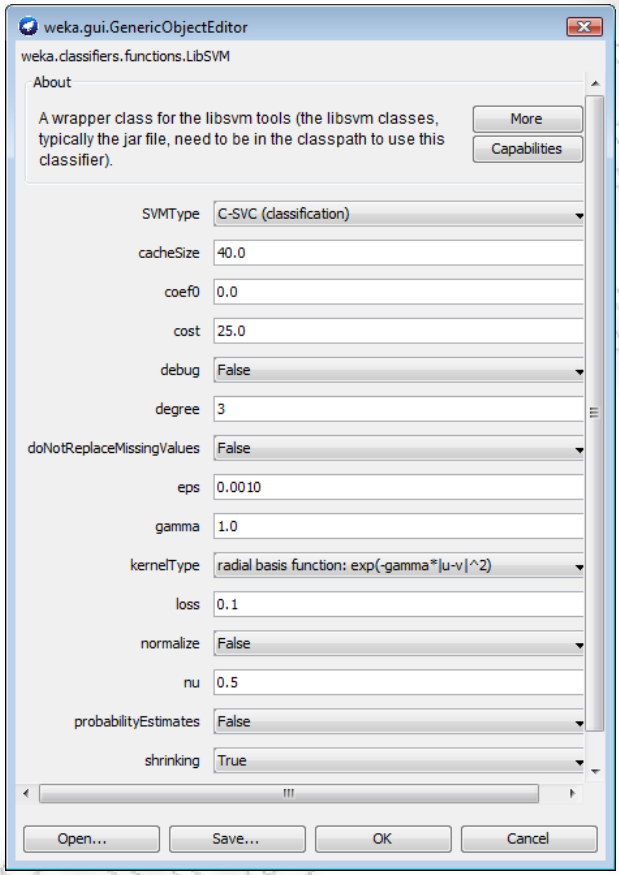
Στο ίδιο πλαίσιο (Classifier output), παρατίθενται στη συνέχεια τα αναλυτικά αποτελέσματα από την εφαρμογή του αλγορίθμου για τις βέλτιστες αυτές τιμές.

Με δεξί κλικ στη σχετική εγγραφή στο Result list και επιλογή *Visualize classifier errors,* λαμβάνουμε την εικόνα στο Σχήμα 6-13. Με κόκκινα και μπλε × εμφανίζονται οι σωστά ταξινομημένες εγγραφές, με κλάση 1 και 0 αντίστοιχα, ενώ με κόκκινα και μπλε □ οι αντίστοιχες εσφαλμένες ταξινομήσεις. Με κλικ σε κάθε εγγραφή ανοίγει παράθυρο με σχετικές πληροφορίες, όπως φαίνεται στο σχήμα.

Στην περίπτωση που θέλουμε να εκτελέσουμε τον ταξινομητή με δικές μας επιλογές στις τιμές των παραμέτρων, στην καρτέλα Classify επιλέγουμε απευθείας τον ταξινομητή ***weka.classifiers.functions.LibSVM***.

Οι βασικότερες παράμετροι που πρέπει να οριστούν είναι οι *cost*, *gamma* και *kerneltype*, όπως φαίνεται στο Σχήμα 6-14.

Σχήμα 6-14
WEKA: Ορισμός παραμέτρων ταξινομητή LibSVM



6.10 Εφαρμογή SVM: Ταξινόμηση Ανεπιθύμητης Αλληλογραφίας

Πολλές υπηρεσίες e-mail σήμερα παρέχουν ***φίλτρα ανεπιθύμητης ηλεκτρονικής αλληλογραφίας*** **(*spam e-mail filters*)** τα οποία ταξινομούν τα εισερχόμενα μηνύματα ηλεκτρονικού ταχυδρομείου σε ανεπιθύμητα ή μη με μεγάλο βαθμό ακρίβειας. Η εφαρμογή που θα παρουσιαστεί έχει σαν σκοπό την υλοποίηση ενός spam filter με χρήση του αλγορίθμου SVM. Συγκεκριμένα ο αλγόριθμος ταξινόμησης θα εκπαιδευτεί ώστε να αποδίδει τον χαρακτηρισμό spam ($y = 1$) ή non-spam ($y = 0$) σε ένα e-mail, δεδομένου του (αγγλικού) κειμένου που περιέχει, αφού προηγουμένως το κείμενο αυτό μετατραπεί σε ένα κατάλληλο διάνυσμα χαρακτηριστικών.

Το σύνολο δεδομένων που έχει χρησιμοποιηθεί βασίζεται σε ένα υποσύνολο του *SpamAssassin public mail corpus*[1]. Πρόσβαση στο πλήρες σετ των αρχείων δεδομένων και κώδικα της εφαρμογής παρέχεται μέσω του συνδέσμου http://bit.ly/twDKXG. Τα κυριότερα σημεία της υλοποίησης της συγκεκριμένης εφαρμογής δίνονται σε γλώσσα Octave στο Παράρτημα της σελ. 171.

Στο Σχήμα 6-15 φαίνεται ένα δείγμα μηνύματος στο οποίο περιλαμβάνονται ένα URL (διαδικτυακή διεύθυνση), μια διεύθυνση ηλεκτρονικού ταχυδρομείου, αριθμοί και ποσά σε δολάρια κλπ.

Σχήμα 6-15
Δείγμα Περιεχομένων Ηλεκτρονικού Μηνύματος

```
> Anyone knows how much it costs to host a web portal ?
>
Well, it depends on how many visitors youre expecting.  This can be
anywhere from less than 10 bucks a month to a couple of $100.  You
should checkout http://www.rackspace.com/ or perhaps Amazon EC2 if
youre running something big..

To unsubscribe yourself from this mailing list, send an email to:
groupname-unsubscribe@egroups.com
```

Αν και πολλά μηνύματα μπορεί να περιέχουν παρόμοιες οντότητες, τα χαρακτηριστικά τους προφανώς θα διαφέρουν σε κάθε περίπτωση. Έτσι, μια μεθοδολογία που συνήθως χρησιμοποιείται κατά την προεπεξεργασία των μηνυμάτων είναι αυτή της «κανονικοποίησης» οντοτήτων σαν τις παραπάνω, έτσι ώστε όλες οι URL να αντιμετωπίζονται με τον ίδιο τρόπο, όλοι οι αριθμοί επίσης, κλπ. Για παράδειγμα θα μπορούσε να αντικατασταθεί κάθε URL που εμφανίζεται στο μήνυμα με το αλφαριθμητικό «*httpaddr*» που θα υποδηλώνει την ύπαρξη κάποιας URL διεύθυνσης.

Με την πρακτική αυτή, η απόφαση του ταξινομητή θα βασιστεί στο αν *κάποια* URL περιλαμβάνεται στο μήνυμα και όχι στο αν περιλαμβάνεται η οποιαδήποτε συγκεκριμένη URL. Αυτό κατά κανόνα βελτιώνει την απόδοση του ταξινομητή, καθώς οι αποστολείς

[1] The SpamAssassin public mail corpus: Μια επιλογή μηνυμάτων ηλεκτρονικού ταχυδρομείου κατάλληλη για χρήση σε δοκιμές συστημάτων φιλτραρίσματος ανεπιθύμητων μηνυμάτων.
http://spamassassin.apache.org/publiccorpus/

τέτοιων μηνυμάτων συνήθως τυχαιοποιούν το λεκτικό των διευθύνσεων και ως εκ τούτου η πιθανότητα να επαναληφθεί κάποια συγκεκριμένη διεύθυνση URL είναι πολύ μικρή.

Στη συνάρτηση **processEmail.m** η οποία περιλαμβάνεται στα αρχεία που απαρτίζουν την εφαρμογή, έχει αναπτυχθεί μια σειρά τέτοιων κανονικοποιήσεων, όπως:

- μετατροπή όλων των χαρακτήρων σε πεζούς,
- διαγραφή των οδηγιών (tags) της HTML,
- αντικατάσταση κάθε URL με το κείμενο «httpaddr»,
- αντικατάσταση κάθε διεύθυνσης e-mail με το κείμενο «emailaddr»,
- αντικατάσταση κάθε αριθμού με το κείμενο «number»,
- αντικατάσταση των χαρακτήρων δολαρίου ($) με το κείμενο «dollar»,
- Διατήρηση μόνο της ρίζας από κάθε λέξη (αφαιρώντας την κατάληξη) με χρήση του αλγορίθμου του Porter [29].

Το αποτέλεσμα αυτής της προεπεξεργασίας στο κείμενο του μηνύματος που εμφανίζεται στο Σχήμα 6-15, είναι η σειρά από «λέξεις» που παρουσιάζεται στο Σχήμα 6-16.

Σχήμα 6-16
Περιεχόμενα μηνύματος μετά την προεπεξεργασία

```
anyon know how much it cost to host a web portal well it depend on how
mani visitor your expect thi can be anywher from less than number buck
a month to a coupl of dollarnumb you should checkout httpaddr or perhap
amazon ecnumb if your run someth big to unsubscrib yourself from thi
mail list send an email to emailaddr
```

Το επόμενο βήμα είναι να επιλέξουμε ποιες λέξεις θα θέλαμε να ληφθούν υπόψη στον αλγόριθμο ταξινόμησης. Με άλλα λόγια, ποιο θα είναι το «λεξικό» που θα χρησιμοποιηθεί. Στην εφαρμογή αυτή έχουν επιλεγεί ως λεξικό μόνον οι 1899 πιο συχνά εμφανιζόμενες λέξεις του SpamAssassin mail corpus. Η λίστα των λέξεων αυτών βρίσκεται στο αρχείο **vocab.txt**, το οποίο επίσης περιλαμβάνεται στα αρχεία που απαρτίζουν την εφαρμογή. Στη πράξη βέβαια χρησιμοποιούνται λεξικά με πολύ μεγαλύτερο αριθμό λέξεων (συνήθως 10.000 έως 50.000).

Δοθέντος του λεξικού που θα χρησιμοποιηθεί, μπορούμε να αντιστοιχίσουμε κάθε λέξη του προεπεξεργασμένου μηνύματος με τον αύξοντα αριθμό ο οποίος αντιστοιχεί στην αντίστοιχη λέξη στο λεξικό. Το αποτέλεσμα αυτής της αντικατάστασης για τις λέξεις του

μηνύματος στο Σχήμα 6-16, εμφανίζεται στο Σχήμα 6-17. Για κάθε κείμενο μηνύματος με το οποίο τροφοδοτείται η συνάρτηση **processEmail.m**, επιστρέφει ένα διάνυσμα με αριθμούς σαν τους παρακάτω.

Σχήμα 6-17
Δείκτες των λέξεων του προεπεξεργασμένου μηνύματος

```
86 916 794 1077 883
370 1699 790 1822
1831 883 431 1171
794 1002 1893 1364
592 1676 238 162 89
688 945 1663 1120
1062 1699 375 1162
479 1893 1510 799
1182 1237 810 1895
1440 1547 181 1699
1758 1896 688 1676
992 961 1477 71 530
1699 531
```

Στη συνέχεια, μέσω της συνάρτησης **emailFeatures.m**, δημιουργείται το διάνυσμα των χαρακτηριστικών με διάσταση όση και ο αριθμός των λέξεων στο λεξικό, ενώ η τιμή κάθε χαρακτηριστικού ανήκει στο $\{0,1\}$. Συγκεκριμένα, $x_i = 1$ αν η $i-$οστή λέξη περιλαμβάνεται στο μήνυμα και $x_i = 0$ αν δεν περιλαμβάνεται. Έτσι, για ένα τυπικό μήνυμα, το διάνυσμα των χαρακτηριστικών θα έχει τη παρακάτω μορφή:

$$x = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{1899}$$

Για την εκπαίδευση του ταξινομητή διατίθεται ένα σύνολο από 4000 προεπεξεργασμένα μηνύματα (spam και non-spam), τα οποία βρίσκονται στο αρχείο **spamTrain.mat**. Σε ένα δεύτερο αντίστοιχο αρχείο με όνομα **spamTest.mat**, βρίσκονται 1000 ακόμη προεπεξεργασμένα μηνύματα, επί των οποίων θα δοκιμαστεί η απόδοση του ταξινομητή. Οι

προβλέψεις παρέχονται μέσω της συνάρτησης **svmPredict.m** [25] (όταν η εκπαίδευση γίνεται με την **svmTrain**). Η ακρίβεια του ταξινομητή στα δεδομένα του συνόλου εκπαίδευσης ήταν **99.825%**.

Στη συνέχεια γίνεται χρήση της βιβλιοθήκης LIBSVM μέσω της Octave (συνάρτηση **svmtrain**) και λαμβάνεται επίσης ακρίβεια **99.825%** επί του συνόλου δεδομένων εκπαίδευσης, με χρήση της αντίστοιχης συνάρτησης **svmpredict**.

Κατόπιν εφαρμόζουμε τα μοντέλα που έχουν δημιουργηθεί μέσω των δύο εναλλακτικών υλοποιήσεων του ταξινομητή SVM επί του συνόλου των 1000 test μηνυμάτων και λαμβάνουμε αντίστοιχα ποσοστά σωστών προβλέψεων:

svmTrain: **98.8%**

LIBSVM: **98.9%**

Καθώς η εκπαίδευση πραγματοποιήθηκε με χρήση γραμμικού πυρήνα, μπορούμε μέσω των παραμέτρων του μοντέλου να δούμε ποιες λέξεις εν τέλει θεωρούνται από τον ταξινομητή ως οι πλέον ενδεικτικές για την ταξινόμηση ενός μηνύματος ως spam. Ζητώντας τις λέξεις που αντιστοιχούν στις 15 μεγαλύτερες τιμές των παραμέτρων λαμβάνουμε το αποτέλεσμα που φαίνεται στο Σχήμα 6-18.

Η ύπαρξη λοιπόν σε μήνυμα λέξεων όπως *«our», «click», «remove», «guarantee»* και *«visit»* είναι πολύ πιθανόν να οδηγήσουν τον αλγόριθμο στην απόφαση να ταξινομήσει το μήνυμα ως spam.

Σχήμα 6-18
Κυριότεροι «προγνώστες» ανεπιθύμητων μηνυμάτων

```
our click remov guarante visit basenumb dollar will price pleas nbsp
most lo ga dollarnumb
```

Στο σύνολο των αρχείων της εφαρμογής περιλαμβάνονται τέσσερα δοκιμαστικά μηνύματα (δύο spam και δύο non-spam), που μπορούν να ελεγχθούν με τον αλγόριθμο που δημιουργήσαμε. Ασφαλώς ο αλγόριθμος μπορεί κάλλιστα να χρησιμοποιηθεί από τον αναγνώστη για τον έλεγχο οποιουδήποτε δικού του μηνύματος, γραμμένου σε αγγλική γλώσσα.

6.11 Σύγκριση Μοντέλων στην R – Ανάγνωση Χειρόγραφων Ψηφίων

Στην παρούσα παράγραφο θα γίνει μια περιληπτική παρουσίαση των δυνατοτήτων που προσφέρει το πακέτο ***caret***[1] στην R για τη σύγκριση μεταξύ μοντέλων ως προς την προβλεπτική τους ικανότητα. Το σύνολο δεδομένων που θα χρησιμοποιηθεί είναι αυτό για την ανάγνωση χειρόγραφων ψηφίων, που παρουσιάστηκε σε προηγούμενα κεφάλαια. Συγκεκριμένα θα χρησιμοποιηθεί η συνάρτηση ***train***, η οποία δημιουργεί μοντέλα πρόβλεψης (χρησιμοποιώντας σχετικές ρουτίνες από άλλα πακέτα), με ταυτόχρονη βελτιστοποίηση μιας σειράς παραμέτρων (εφόσον αυτό ζητηθεί φυσικά), τις οποίες οι αντίστοιχες ρουτίνες λαμβάνουν ως ορίσματα. Η επιλογή των βέλτιστων τιμών για τις παραμέτρους κάθε μοντέλου γίνεται αυτόματα μετά από σύγκριση κάποιων μέτρων απόδοσης. Η απόδοση κάθε πιθανού συνδυασμού τιμών των παραμέτρων προκύπτει μετά την εφαρμογή μιας διαδικασίας αναδειγματοληψίας (π.χ. cross-validation), τα χαρακτηριστικά της οποίας επίσης καθορίζονται από το χρήστη.

Οι αλγόριθμοι που θα χρησιμοποιηθούν για σύγκριση είναι: Το νευρωνικό δίκτυο με ένα κρυφό στρώμα (***nnet***), η μηχανή διανυσμάτων υποστήριξης με γκαουσιανό πυρήνα (***svmRadial***) και η πολυωνυμική παλινδρόμηση (***multinom***) [34]. Επισημαίνεται ότι, καθώς ζητείται βελτιστοποίηση παραμέτρων και στις τρεις περιπτώσεις, η παρακάτω διαδικασία καθίσταται ιδιαίτερα χρονοβόρα.

Μετά τη δημιουργία του μοντέλου με τις βέλτιστες τιμές παραμέτρων για κάθε αλγόριθμο, η μεταξύ τους σύγκριση γίνεται εφαρμόζοντάς τα επί υποσυνόλου των δεδομένων που εξαιρέθηκαν από τη διαδικασία εκπαίδευσης (***test set***). Γι' αυτό, πριν τη διαδικασία εκπαίδευσης, το αρχικό σύνολο των 5000 δεδομένων χωρίζεται σε δύο μέρη, μεγέθους 4000 (training) και 1000 (test) αντίστοιχα.

```
# Εισαγωγή δεδομένων
dataset <- read.csv('imagedata.csv',header=F)

# Προσθήκη κεφαλίδων
names(dataset) <- c(paste('pixel.',rep(1:400),sep=''),'digit')
# Εύρεση αριθμού υποδειγμάτων
nobs <- nrow(dataset)

# Διαχωρισμός σε training και test σετ
train.idc <- sample(nrow(dataset), 0.8*nobs)
test.idc <- setdiff(seq_len(nobs), train.idc)

train.X <- dataset[train.idc,-401]
test.X <- dataset[test.idc,-401]
```

[1] caret: Classification and Regression Training, http://cran.r-project.org/web/packages/caret/index.html

```
train.y <- factor(dataset[train.idc,401])
test.y <- factor(dataset[test.idc,401])
```

Για την επιλογή βέλτιστων τιμών ελέγχονται όλοι οι συνδυασμοί που περιλαμβάνονται στον πίνακα που δίνεται ως όρισμα στην παράμετρο **tuneGrid**.

```
library(caret)

# Δημιουργία dataframe με τιμές των παραμέτρων decay και size του νευρωνικού δικτ.
nnGrid <- expand.grid(.decay = seq(1e-3,0.2,.05), .size = seq(20,30,2))

# Εκπαίδευση δικτύου
nnFit  <- train(train.X, factor(train.y), "nnet", rang = 0.12, maxit = 400,
          MaxNWts = 20000, tuneGrid = nnGrid,
          trControl = trainControl(method = "cv"))
```

Αντίστοιχα, για το πολυωνυμικό μοντέλο:

```
multinomFit <- train(train.X, factor(train.y), "multinom" , trace=FALSE,
               MaxNWts = 20000, maxit=400,
               tuneGrid = data.frame(.decay = seq(1e-3,0.2,.05)),
               trControl = trainControl(method = "cv"))
```

Τέλος, για τη μηχανή διανυσμάτων υποστήριξης:

```
# Δημιουργία data frame με πιθανές τιμές των παραμέτρων
svmGrid <- expand.grid(.sigma = 2^seq(-8,2,1), .C = 1.5^seq(-2,10,1))

# Εκπαίδευση μοντέλου
svmFit <- train(train.X, factor(train.y), "svmRadial",
          tuneGrid = svmGrid, trControl = trainControl(method = "cv"))
```

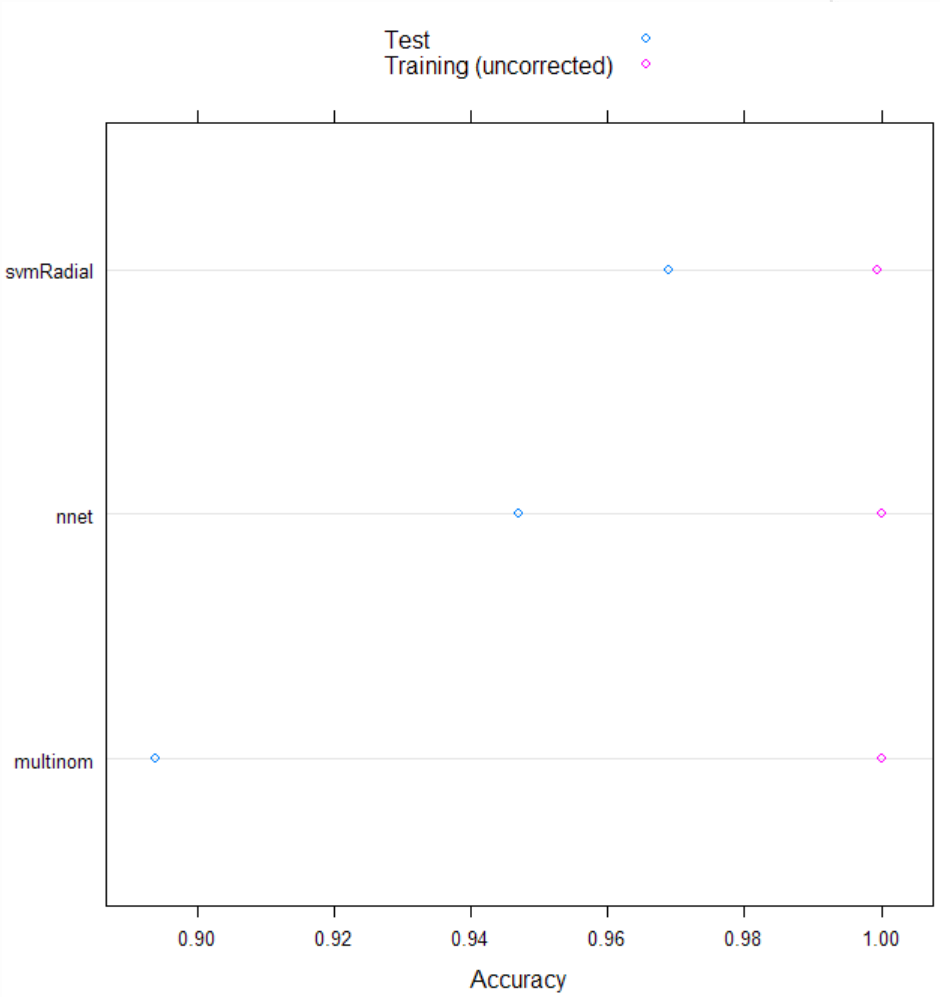
Με την εντολή ***extractPrediction*** λαμβάνουμε ένα αντικείμενο που περιλαμβάνει τόσο τις προβλέψεις των μοντέλων όσο και τις παρατηρηθείσες (πραγματικές) τιμές και για τα τρία μοντέλα.

```
# Εφαρμογή μοντέλων στο σύνολο ελέγχου και δημιουργία προβλέψεων
predTargets <- extractPrediction(list(nnFit, multinomFit, svmFit),
                                 testX = test.X,
                                 testY = factor(test.y))
```

Τέλος, με την εντολή ***plotObsVsPred*** δημιουργείται ένα dot-plot (βλ. Σχήμα 6-19) με τα αποτελέσματα της σύγκρισης των επιδόσεων πρόβλεψης, τόσο επί των δεδομένων του συνόλου εκπαίδευσης (training set), όσο και επί του συνόλου ελέγχου (test set)

```
# Δημιουργία συγκριτικού γραφήματος (dot-plot)
plotObsVsPred(predTargets,auto.key=TRUE)
```

Σχήμα 6-19
Σύγκριση της επίδοσης μεταξύ μοντέλων



Παρατηρούμε ότι τελικά το μοντέλο SVM υπερτερεί έναντι των υπολοίπων στην ταξινόμηση νέων παρατηρήσεων.

ΚΕΦΑΛΑΙΟ 7

Αξιολόγηση της Απόδοσης του Μοντέλου

7.1 Διαγνωστικοί Έλεγχοι

Σκοπός του κεφαλαίου αυτού είναι να δοθούν κάποιες πρακτικές συμβουλές σε σχέση με τις διαθέσιμες μεθόδους που μπορούν να χρησιμοποιηθούν με σκοπό τη βελτίωση της απόδοσης του υπό κατασκευή μοντέλου, καθώς και για τον τρόπο επιλογής των καταλληλότερων από αυτές ανάλογα με την περίπτωση.

Τα συχνότερα χρησιμοποιούμενα μέτρα με τα οποία ελέγχεται η απόδοση ενός μοντέλου είναι η *ακρίβεια* (*accuracy*), που όπως έχει ήδη αναφερθεί προκύπτει ως ο λόγος του αριθμού των σωστών προβλέψεων προς το συνολικό αριθμό των παρατηρήσεων, το *σφάλμα* (*error*) που ισούται με 1 – *ακρίβεια*, η *μήτρα σύγχυσης* κλπ.[1]

Έστω λοιπόν ότι έχουμε στη διάθεσή μας ένα μοντέλο το οποίο, κατά τον έλεγχο της απόδοσής του επί νέων δεδομένων, αποδείχθηκε ότι δεν επιτυγχάνει ικανοποιητική ακρίβεια στις προβλέψεις που παρέχει. Οι κυριότερες από τις διαθέσιμες επιλογές μέσω των οποίων μπορούμε να παρέμβουμε ώστε να βελτιώσουμε την απόδοση του μοντέλου παρατίθενται στη συνέχεια:

- Εξασφάλιση μεγαλύτερου αριθμού δεδομένων εκπαίδευσης.
- Εκπαίδευση του μοντέλου με χρήση ενός υποσυνόλου των διαθέσιμων χαρακτηριστικών (κυρίως στην περίπτωση που είναι διαθέσιμα πάρα πολλά),

[1] Για αναλυτική παρουσίαση των μέτρων απόδοσης βλ. [35], [10], [2], [36].

μέσω μιας – χειροκίνητης ή αυτοματοποιημένης – διαδικασίας επιλογής των πλέον χρήσιμων.

- Εξασφάλιση μεγαλύτερου αριθμού χαρακτηριστικών, στην περίπτωση που θεωρείται ότι τα διαθέσιμα χαρακτηριστικά δεν είναι αρκετά, ή δεν είναι τα πλέον κατάλληλα.
- Προσθήκη τεχνητά δημιουργημένων χαρακτηριστικών όπως πολυωνυμικών, της μορφής $x_1^2, x_2^2, x_1 x_2$, ή αύξηση του αριθμού των κρυφών στρωμάτων ενός νευρωνικού δικτύου κλπ.
- Μεταβολή της τιμής του συντελεστή ομαλοποίησης λ, ή του συντελεστή κόστους C αν πρόκειται για μοντέλο SVM.

Κάποιες από τις παραπάνω επιλογές, όπως π.χ. η συλλογή μεγαλύτερου αριθμού δεδομένων εκπαίδευσης, μπορεί στην πράξη να αποβούν ιδιαίτερα χρονοβόρες και μερικές φορές ενδέχεται να συνεισφέρουν ελάχιστα ή και καθόλου στη βελτίωση της απόδοσης του μοντέλου. Είναι λοιπόν ιδιαίτερα σημαντικό η επιλογή στην οποία θα καταλήξουμε να μη γίνει διαισθητικά ή στην τύχη, αλλά μετά από πειραματική διερεύνηση. Και, όπως θα δούμε στη συνέχεια, υπάρχει μια απλή και σχετικά γρήγορη τεχνική για να απορρίψουμε άμεσα αρκετές από τις προαναφερθείσες επιλογές.

Τέτοιου είδους τεχνικές καλούνται διαγνωστικοί έλεγχοι και χρησιμοποιούνται προκειμένου να μας βοηθήσουν να αντιληφθούμε τι πάει ή δεν πάει καλά με τον υπόψη αλγόριθμο εκπαίδευσης, αλλά και να μας υποδείξουν τους τρόπους με τους οποίους θα πρέπει να παρέμβουμε ώστε να τον βελτιώσουμε. Δεν αποκλείεται κάποιοι από τους ελέγχους αυτούς να απαιτούν κάποιο χρόνο για την υλοποίησή τους. Το σίγουρο όμως είναι ότι αυτή η σπατάλη χρόνου αξίζει τον κόπο.

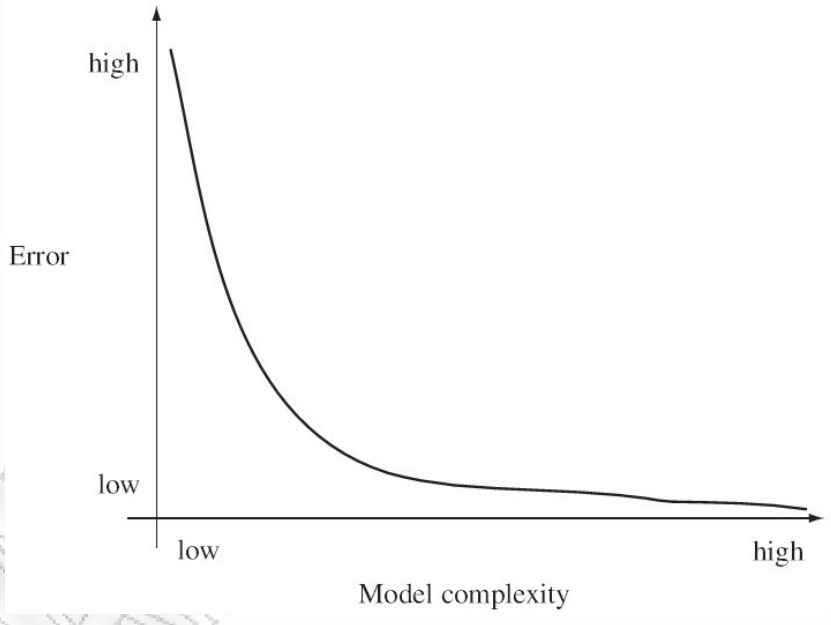
7.2 Αξιολόγηση του Μοντέλου

Κατά τη δημιουργία ενός μοντέλου είναι πιθανόν να δοθεί ιδιαίτερη βαρύτητα στην επιλογή των παραμέτρων εκείνων που θα οδηγούν σε όσο το δυνατόν μεγαλύτερη ακρίβεια (ή αντίστοιχα όσο το δυνατόν μικρότερο σφάλμα) κατά την ταξινόμηση των δεδομένων του συνόλου εκπαίδευσης, μία επιλογή που φυσικά θα προκύψει μετά από μια σειρά διαδοχικών δοκιμών και ρυθμίσεων, έως ότου να ληφθεί το καλύτερο δυνατό αποτέλεσμα. Καθώς το

σύνολο εκπαίδευσης χρησιμοποιείται τόσο για την κατασκευή του μοντέλου όσο και για τον υπολογισμό του σφάλματος, αναφερόμαστε στο σφάλμα αυτό με τον όρο «***σφάλμα εκπαίδευσης***» (***training error***). Κατά τη διαδικασία αξιολόγησης του μοντέλου στη συγκεκριμένη περίπτωση, ο στόχος θα είναι λοιπόν η ελαχιστοποίηση αυτού του σφάλματος εκπαίδευσης.

Όσο αυξάνουμε την πολυπλοκότητα του μοντέλου είναι λογικό να περιμένουμε ότι το σφάλμα εκπαίδευσης θα μειώνεται. Για παράδειγμα, σε ένα μοντέλο μηχανής διανυσμάτων υποστήριξης η πολυπλοκότητα μπορεί να αυξηθεί μέσω κατάλληλης ρύθμισης τριών παραμέτρων: Της επιλογής συνάρτησης πυρήνα, των τιμών για τις ελεύθερες παραμέτρους αυτής και την τιμή της σταθεράς κόστους C. Η σχέση μεταξύ του σφάλματος εκπαίδευσης και της πολυπλοκότητας ενός SVM μοντέλου, για δεδομένο σύνολο εκπαίδευσης, φαίνεται στο Σχήμα 7-1[1].

Σχήμα 7-1
Τυπική καμπύλη του σφάλματος εκπαίδευσης σε μοντέλα SVM



Στα αριστερά του οριζόντιου άξονα βρίσκονται μοντέλα χαμηλής πολυπλοκότητας ενώ στα δεξιά, υψηλής. Ο κατακόρυφος άξονας απεικονίζει το αντίστοιχο σφάλμα εκπαίδευσης.

[1] Η μορφή της καμπύλης στο Σχήμα 7-1 παρουσιάζεται με μια ομαλοποιημένη μορφή. Σε πραγματικά προβλήματα είναι πιθανό να παρατηρηθούν πολλά τοπικά μέγιστα και ελάχιστα.

Είναι προφανές ότι, όπως αναμέναμε, το σφάλμα εκπαίδευσης μειώνεται με την αύξηση της πολυπλοκότητας του μοντέλου.

Για τα περισσότερα σύνολα εκπαίδευσης δεν είναι δύσκολο να βρεθούν κατάλληλες τιμές των παραμέτρων του μοντέλου, ώστε το σφάλμα εκπαίδευσης να προσεγγίζει το μηδέν. Το γεγονός αυτό θα μας ήταν ιδιαιτέρως χρήσιμο εάν το διαθέσιμο σύνολο εκπαίδευσης μπορούσε να θεωρηθεί ως ένας απολύτως αντιπροσωπευτικός εκπρόσωπος του συνολικού πληθυσμού. Δυστυχώς όμως, τα σύνολα εκπαίδευσης ποτέ δεν αποτελούν τέλειους εκπροσώπους των αντίστοιχων πληθυσμών, μιας και συνήθως αποτελούν μόνο ένα μικρό μέρος αυτών. Κατά συνέπεια, το γεγονός ότι μπορούμε να ελαχιστοποιήσουμε το σφάλμα εκπαίδευσης είναι άνευ σημασίας, αφού δεν μας επιτρέπει να καταλήξουμε σε ασφαλή συμπεράσματα σχετικά με την απόδοση του μοντέλου στα υπόλοιπα δεδομένα του πληθυσμού.

Πέρα από το γεγονός ότι το σύνολο εκπαίδευσης αποτελεί συνήθως ένα μικρό μόνο μέρος του πληθυσμού, υπάρχουν μια σειρά ακόμη σφάλματα που ενδέχεται να υποβαθμίσουν την αντιπροσωπευτικότητα των δεδομένων εκπαίδευσης. Μια τέτοια πηγή σφάλματος είναι η λεγόμενη *μεροληψία της δειγματοληψίας* (*sampling bias*), με την έννοια ότι κατά τη λήψη του δείγματος που θα χρησιμοποιηθεί ως σύνολο εκπαίδευσης, είναι πιθανό να αποτύχουμε να συμπεριλάβουμε κάποιες παρατηρήσεις αποφασιστικής σημασίας. Μία ακόμη πηγή σφάλματος αποτελεί ο *θόρυβος* (*noise*), ο οποίος μπορεί να οφείλεται σε πάσης φύσεως λάθη που μπορεί να προκύψουν π.χ. σε ένα πρόβλημα ταξινόμησης κατά τη διαδικασία απόδοσης της κλάσης στα αντικείμενα, από τη συνάρτηση-στόχο.

Έχοντας ως δεδομένο το γεγονός ότι τα σύνολα εκπαίδευσης αποτελούν ελλιπείς εκπροσωπήσεις των αντίστοιχων πληθυσμών, η ελαχιστοποίηση του σφάλματος εκπαίδευσης καλείται «***υπερπροσαρμογή***» (***over-fitting***). Ένα υπερπροσαρμοσμένο μοντέλο είναι ένα μοντέλο που προσεγγίζει το τέλειο για τα δεδομένα του συνόλου εκπαίδευσης αφού το σφάλμα εκπαίδευσης προσεγγίζει το μηδέν. Το μεγαλύτερο δε μέρος της πολυπλοκότητάς του είναι αποτέλεσμα μεροληψιών δειγματοληψίας και θορύβου στα δεδομένα εκπαίδευσης. Η επιπλέον αυτή πολυπλοκότητα μπορεί όμως να οδηγήσει σε σφάλματα ταξινόμησης όταν το μοντέλο εφαρμοστεί σε δεδομένα του πληθυσμού τα οποία δεν περιλαμβάνονταν στο σύνολο εκπαίδευσης. Αυτό σημαίνει ότι τα υπερπροσαρμοσμένα μοντέλα έχουν *μειωμένη ικανότητα γενίκευσης*. Στην επόμενη παράγραφο θα δούμε μια μεθοδολογία για τον περιορισμό της επίδρασης του θορύβου στο σύνολο εκπαίδευσης.

7.3 Διαμερισμός των Διαθέσιμων Δεδομένων

Κατά τη διαδικασία κατασκευής μοντέλων πρόβλεψης, όπως τα μοντέλα ταξινόμησης και ιδιαίτερα στις περιπτώσεις όπου το πλήθος των διαθέσιμων δεδομένων δεν είναι πολύ περιορισμένο, συνηθίζεται η διάσπασή τους σε τρία ανεξάρτητα μέρη. Το πρώτο καλείται «***σύνολο εκπαίδευσης***» (***training set***), το δεύτερο «***σύνολο επικύρωσης***» (***validation set***) και το τρίτο «***σύνολο ελέγχου***» (***test set***). Η διαδικασία αυτού του διαμερισμού εκτελείται μέσω τεχνικών τυχαιοποίησης, ώστε να διασφαλίζεται ότι κάθε ένα από αυτά τα σύνολα είναι αντιπροσωπευτικό του πλήθους των διαθέσιμων δεδομένων.

Το *σύνολο εκπαίδευσης*, με το οποίο κυρίως έχουμε ασχοληθεί στα πλαίσια αυτής της εργασίας, χρησιμοποιείται από τους διάφορους αλγορίθμους εκπαίδευσης ώστε να προσδιοριστούν οι βέλτιστες τιμές των παραμέτρων του αντίστοιχου μοντέλου ταξινόμησης (π.χ. για την εύρεση των συναπτικών βαρών ενός νευρωνικού δικτύου). Ο έλεγχος της απόδοσης του μοντέλου επί αυτού του συνόλου δεδομένων, όπως ήδη έχει αναλυθεί, δεν είναι ενδεικτικός της ικανότητάς του να γενικεύει τα συμπεράσματα, από το σύνολο εκπαίδευσης στον πληθυσμό.

Το *σύνολο επικύρωσης* χρησιμοποιείται για τον έλεγχο της απόδοσης του μοντέλου επί νέων δεδομένων με σκοπό τη βελτιστοποίηση των τιμών που θα δοθούν στις ελεύθερες παραμέτρους του (όπως π.χ. την εύρεση του βέλτιστου αριθμού νευρώνων, την επιλογή της παραμέτρου κόστους C μιας μηχανής διανυσμάτων υποστήριξης, ή του συντελεστή ομαλοποίησης λ ενός μοντέλου λογιστικής παλινδρόμησης), έχοντας σαν κριτήριο την απόδοση του μοντέλου στο σύνολο δεδομένων επικύρωσης. Κατ' αυτή την έννοια, το σύνολο επικύρωσης *χρησιμοποιείται* προκειμένου να οριστικοποιηθεί η μορφή του μοντέλου. Επομένως, η εκτίμηση της απόδοσης του μοντέλου επί των δεδομένων του συνόλου επικύρωσης, δε θα είναι και πάλι αμερόληπτη, καθώς τα ίδια δεδομένα χρησιμοποιήθηκαν για τη ρύθμιση κάποιων από τις παραμέτρους του.

Τέλος, *το σύνολο ελέγχου* χρησιμοποιείται μόνο στο τελευταίο στάδιο για τον έλεγχο της απόδοσης του μοντέλου, όπως αυτό έχει διαμορφωθεί μετά τη διαδικασία επικύρωσης. Έτσι, καθώς τα δεδομένα στα οποία γίνεται ο έλεγχος δεν έχουν χρησιμοποιηθεί ξανά στην όλη διαδικασία, η εκτίμηση της προβλεπτικής ικανότητάς (γενίκευσης) του μοντέλου γίνεται όσο το δυνατόν πιο αμερόληπτη.

Δεν υπάρχει κοινά αποδεκτή πρακτική για το μέγεθος του καθενός από τα παραπάνω σύνολα, ως ποσοστού επί των συνολικών δεδομένων. Τυπικοί διαμερισμοί είναι π.χ. 40/30/30, 50/25/25, 60/20/20 ή 70/15/15.

Η δημιουργία συνόλου επικύρωσης κάποιες φορές δεν κρίνεται απαραίτητη, όπως όταν π.χ. χρησιμοποιούνται έτοιμες ρουτίνες κατασκευής μοντέλων στην R, οι οποίες επιστρέφουν την τιμή που λαμβάνουν τα διάφορα μέτρα απόδοσης, εκτελώντας αυτόματα τη διαδικασία της *διασταυρούμενης επικύρωσης* (*cross-validation*), ή αν π.χ. επιλεγεί "*Cross-validation*" στο πλαίσιο "*Test options*", όταν χρησιμοποιούνται ρουτίνες του WEKA.

Η ιδέα του cross-validation είναι απλή: Ένα δεδομένο σύνολο εκπαίδευσης διαμερίζεται σε έναν αυθαίρετο αριθμό υποσυνόλων, π.χ. δέκα, μέσω τυχαίας δειγματοληψίας. Στη συνέχεια εκτελείται ο αλγόριθμος εκπαίδευσης του μοντέλου χρησιμοποιώντας τα εννέα από αυτά, ως σύνολο εκπαίδευσης. Τέλος μετράται η απόδοση του μοντέλου στα δεδομένα του δέκατου υποσυνόλου που δεν χρησιμοποιήθηκε κατά την εκπαίδευση και παίζει το ρόλο του συνόλου επικύρωσης. Η διαδικασία αυτή κατόπιν επαναλαμβάνεται άλλες εννέα φορές, όπου κάθε φορά ένα διαφορετικό υποσύνολο από τα δέκα παίζει το ρόλο του συνόλου επικύρωσης. Οι τελικές τιμές των μέτρων απόδοσης του μοντέλου προκύπτουν ως ο μέσος όρος των μέτρων που υπολογίστηκαν σε καθένα από τα στάδια (folds) της διαδικασίας.

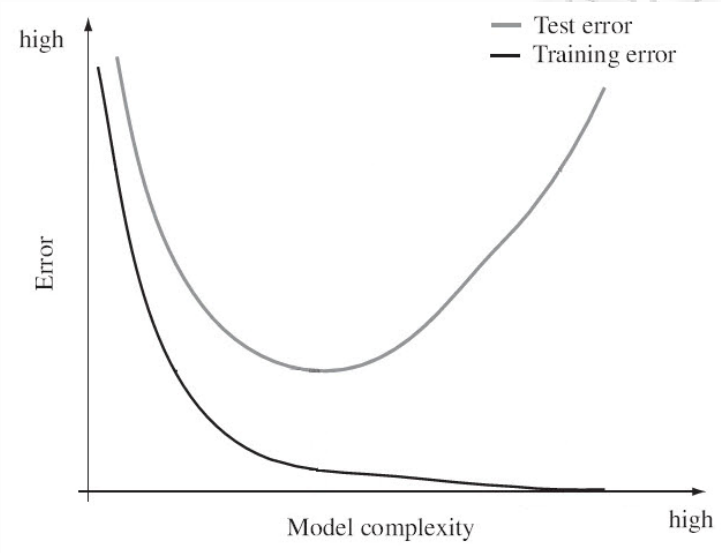
7.4 Μεροληψία και Διακύμανση Μοντέλου

Όπως ήδη αναφέρθηκε στο Κεφ. 3, συχνά αναφερόμαστε στο φαινόμενο της υπερπροσαρμογής λέγοντας ότι το μοντέλο έχει *«υψηλή διακύμανση»* (high variance). Από την άλλη μεριά, στην περίπτωση της υποπροσαρμογής (under-fitting), δηλαδή όταν το μοντέλο είναι περισσότερο απ' όσο πρέπει απλό, με αποτέλεσμα και τα σφάλματα εκπαίδευσης και τα σφάλματα ελέγχου να είναι μεγάλα, λέμε ότι το μοντέλο χαρακτηρίζεται από *«υψηλή μεροληψία»* (high bias). Στο σύνολο των περιπτώσεων όπου το μοντέλο που προκύπτει από έναν αλγόριθμο μάθησης δεν αποδίδει ικανοποιητικά, αυτό οφείλεται είτε σε πρόβλημα υψηλής διακύμανσης είτε σε πρόβλημα υψηλής μεροληψίας. Το να διαγνωστεί λοιπόν τι από τα δύο συμβαίνει θα δώσει μια σαφή ένδειξη για την κατεύθυνση στην οποία θα πρέπει να κινηθούμε για την αντιμετώπιση του προβλήματος.

Στα αριστερά του οριζόντιου άξονα στο Σχήμα 7-1 βρίσκονται τα απλούστερα από τα μοντέλα, τα οποία και διακρίνονται από υψηλή μεροληψία, ενώ τα πλέον πολύπλοκα, στα

δεξιά, χαρακτηρίζονται από υψηλή διακύμανση. Αν επιδιώκαμε να απεικονίσουμε στο ίδιο διάγραμμα και το σφάλμα κάθε μοντέλου κατά την εφαρμογή του στο σύνολο ελέγχου[1] θα λαμβάναμε ένα διάγραμμα όπως αυτό στο Σχήμα 7-2, με τις χαρακτηριστικές καμπύλες σφάλματος εκπαίδευσης και ελέγχου.

Σχήμα 7-2
Τυπικές καμπύλες σφάλματος εκπαίδευσης και ελέγχου μοντέλων SVM



Αυτό που παρατηρούμε άμεσα στο διάγραμμα αυτό, είναι ότι το μοντέλο που ελαχιστοποιεί το σφάλμα εκπαίδευσης δεν ταυτίζεται με αυτό που ελαχιστοποιεί το σφάλμα ελέγχου. Ενώ το σφάλμα εκπαίδευσης μειώνεται με την αύξηση της πολυπλοκότητας, το σφάλμα ελέγχου παρουσιάζει μια πολύ διαφορετική συμπεριφορά. Ξεκινώντας από τα μοντέλα χαμηλής πολυπλοκότητας στα αριστερά (περιοχή υψηλής μεροληψίας), το σφάλμα ελέγχου μειώνεται σταδιακά καθώς η πολυπλοκότητα αυξάνει. Όμως, από ένα συγκεκριμένο σημείο και μετά, αρχίζει και πάλι να αυξάνεται με την αύξηση της πολυπλοκότητας (περιοχή υψηλής διακύμανσης). Το σημείο αυτό αντιστοιχεί στο μοντέλο με το βέλτιστο σφάλμα ελέγχου, ενώ ταυτόχρονα σηματοδοτεί την έναρξη του φαινομένου της υπερπροσαρμογής των μοντέλων με ακόμη μεγαλύτερη πολυπλοκότητα. Ο βαθμός υπερπροσαρμογής εκδηλώνεται με την όλο και φτωχότερη απόδοση των μοντέλων στο σύνολο ελέγχου. Η υποβαθμισμένη αυτή απόδοση οφείλεται στο γεγονός ότι στα μοντέλα αυτά απαιτήθηκε

[1] Για την ακρίβεια και σύμφωνα με τα προαναφερθέντα, το μοντέλο κατόπιν δοκιμάζεται στο σύνολο επικύρωσης και αυτό το σφάλμα είναι που στην ουσία απεικονίζεται στο γράφημα. Για λόγους απλότητας όμως θα γίνεται αναφορά στη συνέχεια μόνο σε σφάλμα εκπαίδευσης και ελέγχου.

αυξημένη πολυπλοκότητα, προκειμένου να αποδώσει μια δομή των δεδομένων εκπαίδευσης η οποία προφανώς δεν υφίσταται στα δεδομένα ελέγχου. Η αυξημένη αυτή πολυπλοκότητα λοιπόν, κάθε άλλο παρά συνεισφέρει στην επιδιωκόμενη γενίκευση. Ο ΠΙΝΑΚΑΣ 7-1 αποτελεί μια εναλλακτική απεικόνιση της σύγκρισης μεταξύ μοντέλων και της επιλογής του μοντέλου με το βέλτιστο σφάλμα ελέγχου.

ΠΙΝΑΚΑΣ 7-1. Καταγραφή σφάλματος εκπαίδευσης και ελέγχου κάθε μοντέλου

A/A	CLASSIFIER	TRAIN ERROR	TEST ERROR	ΕΠΙΛΟΓΗ
1	F1			
2	F2			
3	F3			
4	F4			ΚΑΛΥΤΕΡΟ
5	F5			
6	F6			
7	F7			

Εξαιτίας του ότι το ανεξάρτητο σύνολο ελέγχου μπορεί να θεωρηθεί ως ένα αντιπροσωπευτικό δείγμα δεδομένων από το συνολικό πληθυσμό, η απόδοση του μοντέλου στα δεδομένα του συνόλου αυτού μας επιτρέπει να αντλήσουμε πολύτιμα συμπεράσματα σχετικά με την απόδοση του μοντέλου στο σύνολο του πληθυσμού. Να αποτιμήσουμε δηλαδή, την ικανότητά του να γενικεύει.

Συμπερασματικά, αντί να επιδιώκουμε την ελαχιστοποίηση του σφάλματος εκπαίδευσης κατά τη διαδικασία αξιολόγησης του μοντέλου, θα πρέπει να επιδιώκουμε την ελαχιστοποίηση του σφάλματος ελέγχου. Με τον τρόπο αυτό αποκτούμε την πλέον ρεαλιστική και αμερόληπτη εκτίμηση για την απόδοση του μοντέλου στο συνολικό πληθυσμό δεδομένων.

Έστω λοιπόν ότι το μοντέλο μας παρουσιάζει μειωμένη απόδοση (υψηλό σφάλμα) όταν εφαρμόζεται στο σύνολο ελέγχου. Θα πρέπει πρωτίστως να διαπιστώσουμε αν αυτό οφείλεται σε υψηλή διακύμανση ή σε υψηλή μεροληψία. Η ιδέα προκύπτει άμεσα, παρατηρώντας το διάγραμμα στο Σχήμα 7-2. Αν το σφάλμα εκπαίδευσης είναι επίσης υψηλό, βρισκόμαστε στην αριστερή περιοχή του διαγράμματος, αυτήν της υψηλής μεροληψίας. Αν αντίθετα το σφάλμα εκπαίδευσης είναι πολύ μικρότερο του σφάλματος ελέγχου, τότε βρισκόμαστε στη δεξιά πλευρά του διαγράμματος, αυτή των μοντέλων υψηλής διακύμανσης.

7.4.1 Μεροληψία και διακύμανση συναρτήσει της παραμέτρου ομαλοποίησης

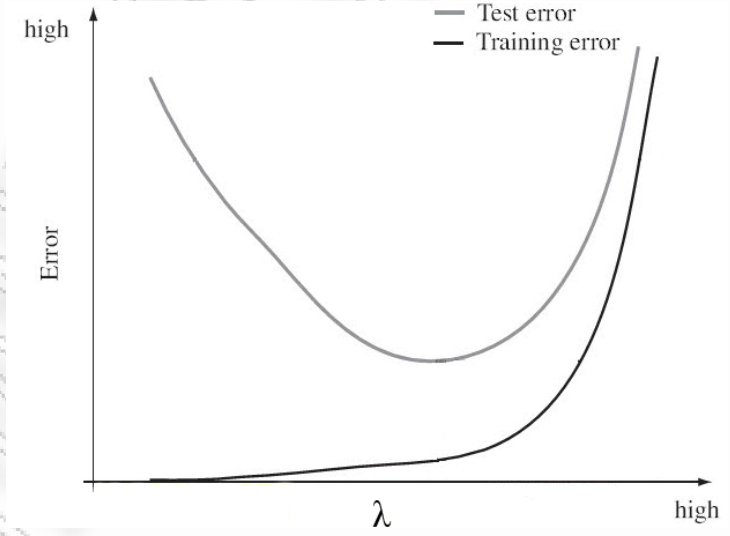
Στο Κεφ. 3, κατά την παρουσίαση της ομαλοποιημένης λογιστικής παλινδρόμησης, είδαμε το ρόλο της παραμέτρου ομαλοποίησης λ στην πρόληψη της υπερπροσαρμογής. Εδώ θα δούμε πώς η μεταβολή της τιμής της παραμέτρου αυτής επηρεάζει τη μεροληψία και τη διακύμανση ενός μοντέλου.

Όπως έχει ήδη αναφερθεί, μια πολύ αυξημένη τιμή της παραμέτρου λ σε ένα μοντέλο π.χ. πολυωνυμικής μορφής, θα έχει σαν αποτέλεσμα οι τιμές όλων των παραμέτρων, πλην του σταθερού όρου, να λάβουν τιμές κοντά στο μηδέν. Το μοντέλο τότε ουσιαστικά ταυτίζεται με την τιμή αυτού του σταθερού όρου και συνεπώς, καθώς υποπροσαρμόζεται έντονα στα δεδομένα, χαρακτηρίζεται από υψηλή μεροληψία.

Στο αντίθετο άκρο, το να δοθεί στη παράμετρο λ μια πολύ μικρή ή και μηδενική τιμή, θα έχει ως αποτέλεσμα την υπερπροσαρμογή του μοντέλου στα δεδομένα εκπαίδευσης και κατά συνέπεια το χαρακτηρισμό του ως μοντέλου υψηλής διακύμανσης. Προφανώς η τιμή του λ που αντιστοιχεί στο βέλτιστο μοντέλο, σύμφωνα π.χ. με τη διάταξη που απεικονίζει ο ΠΙΝΑΚΑΣ 7-1, θα είναι μια ενδιάμεση τιμή, ούτε πολύ μεγάλη ούτε πολύ μικρή.

Σχήμα 7-3

Σχέση μεταξύ του λ και του σφάλματος εκπαίδευσης και ελέγχου



Σύμφωνα με μια συνήθη πρακτική, για την επιλογή μιας καλής τιμής για την παράμετρο λ καθορίζεται εξ' αρχής ένα σύνολο υποψήφιων τιμών, όπως π.χ. $(0, 0.1, 0.2, 0.4, \ldots, 10)$, όπου η κάθε μία είναι περίπου διπλάσια της προηγούμενης. Οι τιμές αυτές αντιστοιχούν σε ισάριθμα

μοντέλα από τα οποία θα επιλεγεί το καταλληλότερο με βάση την τιμή του σφάλματος που θα προκύψει από την εφαρμογή του στα δεδομένα του συνόλου ελέγχου.

Αν επιχειρήσουμε να κατασκευάσουμε ένα διάγραμμα που να απεικονίζει τα παραπάνω, θα πάρουμε καμπύλες παρόμοιες με αυτές που φαίνονται στο Σχήμα 7-3 (παρατηρήστε ότι αυτές μοιάζουν με κατοπτρικές εικόνες αυτών που εμφανίζονται στο Σχήμα 7-2). Εδώ, η περιοχή υψηλής μεροληψίας βρίσκεται στα δεξιά, ενώ στα αριστερά βρίσκεται η περιοχή υψηλής διακύμανσης. Με τη βοήθεια ενός τέτοιου διαγράμματος είναι εύκολο να εντοπίσουμε τι πραγματικά συμβαίνει και αν πράγματι έχουμε επιλέξει μια καλή τιμή για το συντελεστή ομαλοποίησης λ.

7.5 Καμπύλες Μάθησης

Μια εναλλακτική μεθοδολογία που χρησιμοποιείται συχνά για να εξακριβωθεί αν βρισκόμαστε αντιμέτωποι με πρόβλημα υψηλής διακύμανσης ή πρόβλημα υψηλής μεροληψίας, είναι η αποτύπωση των καμπύλων σφάλματος εκπαίδευσης και ελέγχου, όχι πια ως συναρτήσεις της πολυπλοκότητας του μοντέλου ή της τιμής του συντελεστή ομαλοποίησης, αλλά ως συναρτήσεις του πλήθους των δεδομένων στο σύνολο εκπαίδευσης.

Φυσικά, αφού το πλήθος των δεδομένων εκπαίδευσης που έχουμε στη διάθεσή μας είναι σταθερό, αυτό που κάνουμε είναι να χρησιμοποιούμε σκοπίμως υποσύνολα των δεδομένων εκπαίδευσης και να υπολογίζουμε για το καθένα απ' αυτά τις αντίστοιχες τιμές των σφαλμάτων.

Η ιδέα στην οποία βασίζεται η μεθοδολογία αυτή είναι ότι όσο μικρότερο είναι το πλήθος του συνόλου εκπαίδευσης τόσο πιο επιτυχής θα είναι η προσαρμογή ενός συγκεκριμένου μοντέλου και τόσο μικρότερο το αντίστοιχο σφάλμα εκπαίδευσης. Αντίθετα, όσο μεγαλώνει το πλήθος των δεδομένων εκπαίδευσης τόσο θα περιορίζεται ο βαθμός προσαρμογής του μοντέλου και αντίστοιχα θα αυξάνει το σφάλμα εκπαίδευσης.

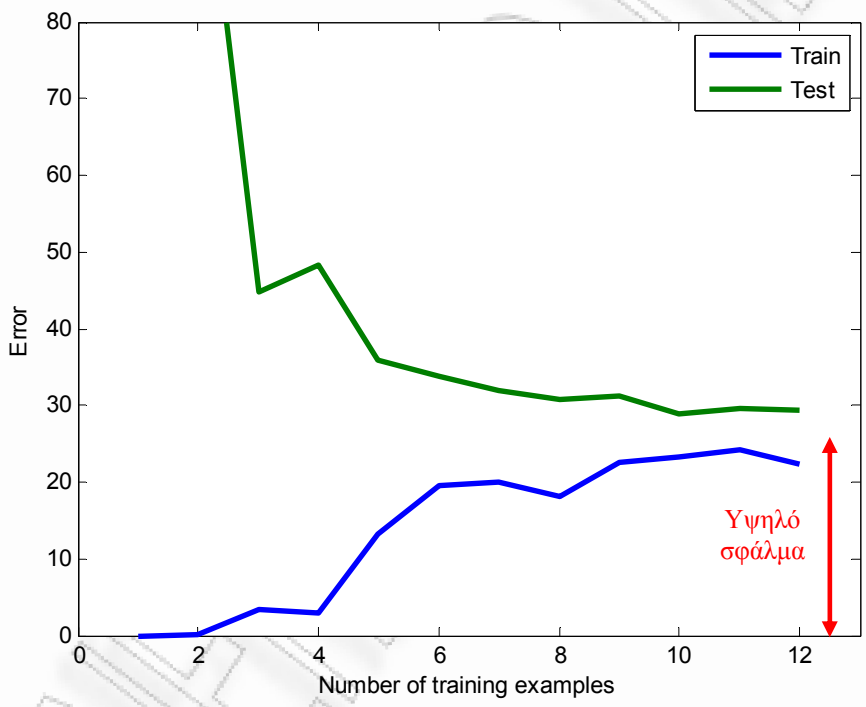
Το αντίθετο προφανώς θα συμβαίνει με το σφάλμα ελέγχου. Όσο μικρότερο είναι το πλήθος του συνόλου εκπαίδευσης τόσο πιο μειωμένη ικανότητα γενίκευσης θα παρουσιάζει το μοντέλο και τόσο μεγαλύτερο θα είναι το σφάλμα ελέγχου. Αντίθετα, όσο μεγαλώνει το πλήθος των δεδομένων εκπαίδευσης τόσο θα μειώνεται στο σφάλμα ελέγχου, αφού όσο περισσότερα είναι τα δεδομένα στα οποία εκπαιδεύτηκε το μοντέλο, τόσο μεγαλύτερη ικανότητα γενίκευσης θα επιδεικνύει, κατά την εφαρμογή του σε νέα δεδομένα.

Τονίζεται εδώ ότι για τον υπολογισμό του σφάλματος εκπαίδευσης λαμβάνεται υπόψη μόνο το πλήθος των δεδομένων του εκάστοτε συνόλου εκπαίδευσης, ενώ για τον υπολογισμό του σφάλματος ελέγχου λαμβάνεται κάθε φορά υπόψη ο συνολικός αριθμός των δεδομένων ελέγχου (ή επικύρωσης).

Στην περίπτωση που είμαστε αντιμέτωποι με πρόβλημα υψηλής μεροληψίας, το σφάλμα ελέγχου θα αρχίσει να μειώνεται με τη σταδιακή αύξηση των δεδομένων εκπαίδευσης, αλλά από έναν αριθμό δεδομένων και μετά, η βελτίωση της απόδοσης που θα παρουσιάζει ένα υποπροσαρμοσμένο μοντέλο θα είναι μηδαμινή.

Σχήμα 7-4

Καμπύλες μάθησης στην περίπτωση υψηλής μεροληψίας



Όσον αφορά το σφάλμα εκπαίδευσης, αυτό θα ξεκινήσει από χαμηλές τιμές, για μικρά μεγέθη του συνόλου εκπαίδευσης και θα καταλήξει σε τιμές παραπλήσιες του σφάλματος ελέγχου. Το πρόβλημα της υψηλής μεροληψίας αντανακλάται στην *υψηλή τιμή στην οποία καταλήγουν να έχουν, τόσο το σφάλμα ελέγχου όσο και το σφάλμα εκπαίδευσης*. Ένα χαρακτηριστικό γράφημα των καμπύλων μάθησης στη περίπτωση υψηλής μεροληψίας

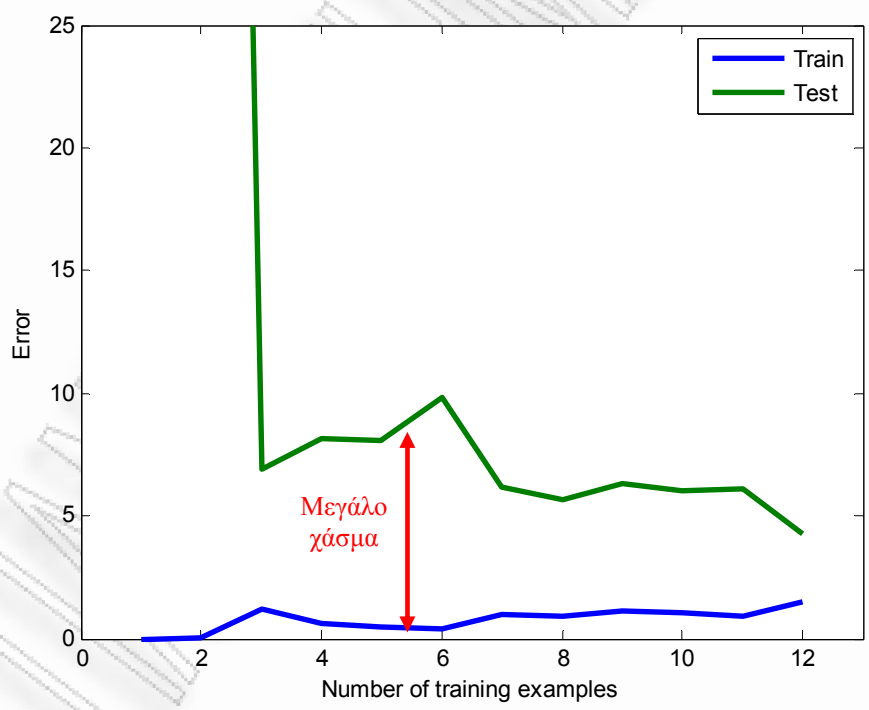
δίνεται στο Σχήμα 7-4, όπου βλέπουμε και τα δύο σφάλματα να συγκλίνουν σε παραπλήσιες τιμές, της τάξης των 25 μονάδων[1].

Το συμπέρασμα στο οποίο καταλήγουμε είναι ότι σε περίπτωση που η μη ικανοποιητική απόδοση του μοντέλου μας οφείλεται σε υψηλή μεροληψία, η εξασφάλιση μεγαλύτερου αριθμού δεδομένων εκπαίδευσης, από μόνη της, δε θα προσφέρει αξιοσημείωτη βελτίωση των τιμών του σφάλματος ελέγχου (όπως και εκπαίδευσης). Αυτό το συμπέρασμα είναι ιδιαίτερα χρήσιμο γιατί θα μας αποτρέψει από τη σπατάλη χρόνου και κόπου που συνεπάγεται μια ενδεχόμενη απόπειρα συλλογής περισσότερων δεδομένων εκπαίδευσης, με την ελπίδα ότι μέσω αυτών θα βελτιωθεί η απόδοση του μοντέλου.

Στην περίπτωση που είμαστε αντιμέτωποι με ένα πρόβλημα υψηλής διακύμανσης, το σφάλμα εκπαίδευσης του υπερπροσαρμοσμένου μοντέλου θα ξεκινήσει από πολύ χαμηλές τιμές και καθώς τα δεδομένα εκπαίδευσης αυξάνουν, θα συνεχίσει να αποδίδει ικανοποιητικά παρουσιάζοντας μικρή σχετικά αύξηση των τιμών του σφάλματος εκπαίδευσης.

Σχήμα 7-5

Καμπύλες μάθησης στην περίπτωση υψηλής διακύμανσης



[1] Οι τιμές του σφάλματος στον κατακόρυφο άξονα αναφέρονται στο τετραγωνικό σφάλμα ενός μοντέλου απλής γραμμικής παλινδρόμησης, το οποίο έχει χρησιμοποιηθεί για την παραγωγή των διαγραμμάτων.

Από τη άλλη μεριά, το σφάλμα ελέγχου ενός υπερπροσαρμοσμένου μοντέλου θα παραμείνει σε υψηλές τιμές ακόμη και για αρκετά μεγάλα μεγέθη του συνόλου εκπαίδευσης, παρουσιάζοντας όμως μία πτωτική τάση. Το ενδεικτικό διαγνωστικό στοιχείο για την περίπτωση υψηλής διακύμανσης είναι η *ύπαρξη ενός μεγάλου χάσματος μεταξύ των καμπύλων εκπαίδευσης και ελέγχου*, όπως φαίνεται στο Σχήμα 7-5.

Αν συλλέγαμε περισσότερα δεδομένα έτσι ώστε να μπορούσαμε να επεκτείνουμε το Σχήμα 7-5 προς τα δεξιά, θα βλέπαμε τις δύο καμπύλες να συγκλίνουν σταδιακά. Στη περίπτωση αυτή λοιπόν, η συλλογή περισσότερων δεδομένων εκπαίδευσης μπορεί πράγματι να βελτιώσει την απόδοση του μοντέλου.

Επισημαίνεται εδώ ότι τα ΣΧΗΜΑΤΑ 7-4 και 7-5 έχουν προέλθει από την αντιμετώπιση του ίδιου προβλήματος με δύο διαφορετικά όμως μοντέλα, συνεπώς οι τιμές των σφαλμάτων στον κατακόρυφο άξονα είναι μεταξύ τους συγκρίσιμες.

Στην πρώτη ενότητα του παρόντος κεφαλαίου παρατέθηκε μια σειρά παρεμβάσεων με σκοπό τη βελτίωση της απόδοσης ενός μοντέλου το οποίο παρουσιάζει μη αποδεκτά σφάλματα ελέγχου. Ας δούμε επιγραμματικά, σύμφωνα με τα έως τώρα αναφερθέντα, ποιες είναι οι πλέον αποτελεσματικές παρεμβάσεις σε κάθε περίπτωση.

Περίπτωση υψηλής μεροληψίας

- *Εξασφάλιση μεγαλύτερου αριθμού χαρακτηριστικών*
- *Προσθήκη τεχνητά δημιουργημένων χαρακτηριστικών*
- *Μείωση του συντελεστή ομαλοποίησης*

Περίπτωση υψηλής διακύμανσης

- *Εξασφάλιση μεγαλύτερου αριθμού δεδομένων εκπαίδευσης*
- *Χρήση ενός υποσυνόλου των διαθέσιμων χαρακτηριστικών*
- *Αύξηση του συντελεστή ομαλοποίησης*

7.6 Διαστήματα Εμπιστοσύνης Σφάλματος

Όσο προσεκτικά και αν έχει επιλεγεί το σύνολο εκπαίδευσης D, πάντα θα υπάρχει ένας βαθμός αβεβαιότητας κατά πόσον το σύνολο αυτό είναι αντιπροσωπευτικό του πληθυσμού. Κατ' επέκταση, πάντα θα υπάρχει ένας βαθμός αβεβαιότητας ως προς το υπολογιζόμενο σφάλμα που θα προκύψει από την επεξεργασία του συνόλου αυτού. Στην παράγραφο αυτή θα

παρουσιαστεί η κατασκευή διαστημάτων εμπιστοσύνης για το σφάλμα, μέσω των οποίων ποσοτικοποιείται αυτή η αβεβαιότητα.

Έστω err_D το σφάλμα ενός μοντέλου, υπολογισμένο με βάση ένα σύνολο δεδομένων D. Το $p\%$ διάστημα εμπιστοσύνης καθορίζει ένα φάσμα τιμών για το err_D μεταξύ ενός ανώτερου (ub) κι ενός κατώτερου (lb) ορίου και η ερμηνεία του είναι η εξής: Αν ληφθεί μεγάλος αριθμός δειγμάτων από τον πληθυσμό και για κάθε ένα από τα δείγματα αυτά υπολογιστεί ένα διάστημα εμπιστοσύνης, το $p\%$ των διαστημάτων αυτών θα περιλαμβάνουν την πραγματική τιμή του err_D.

Για $p = 95\%$, το διάστημα $[\text{lb}, \text{ub}]$ ως γνωστόν, καλείται το *95% διάστημα εμπιστοσύνης*.

Ένας ιδιαίτερα αποτελεσματικός και υπολογιστικά ξεκάθαρος τρόπος εκτίμησης των ορίων του διαστήματος είναι μέσω της τεχνικής *bootstrap*. Σύμφωνα με την τεχνική αυτή, χρησιμοποιείται το σύνολο D για τον προσδιορισμό της αβεβαιότητας που αυτό το ίδιο παρουσιάζει στην αναπαράσταση του συνολικού πληθυσμού. Με την τεχνική bootstrap δημιουργούμε b αντίγραφα του συνόλου D μέσω δειγματοληψίας με επανάθεση. Αυτό σημαίνει ότι κάθε αντίγραφο B_i του D με $i = 1, \ldots, b$, το οποίο καλείται δείγμα bootstrap, κατά πάσα πιθανότητα θα διαφέρει από το D λόγω της φύσης της δειγματοληψίας. Κάθε δείγμα bootstrap λοιπόν αποτελεί έναν εναλλακτικό τρόπο δόμησης ενός συνόλου εκπαίδευσης, κάνοντας χρήση δεδομένων από το συνολικό πληθυσμό. Η ποικιλία των δειγμάτων εξασφαλίζει την ανίχνευση του βαθμού αβεβαιότητας για το κατά πόσον ένα και μοναδικό από αυτά, το D, αποτελεί αντιπροσωπευτικό δείγμα του πληθυσμού. Ως προς το πλήθος των δειγμάτων bootstrap που θα δημιουργηθούν, ένας γενικός κανόνας είναι ότι όσο περισσότερα είναι αυτά, τόσο ακριβέστερη γίνεται η εκτίμηση των ορίων του διαστήματος εμπιστοσύνης. Στην πράξη, την απόφαση για το πλήθος των δειγμάτων καθορίζουν η διαθέσιμη υπολογιστική ισχύς και το πόσο κρίσιμη είναι η ακριβής εκτίμηση των ορίων του διαστήματος.

Μετά τη δημιουργία των δειγμάτων bootstrap, χρησιμοποιούμε το καθένα από αυτά για τον υπολογισμό του αντίστοιχου σφάλματος του μοντέλου (π.χ. του σφάλματος επικύρωσης). Επισημαίνεται ότι κατά τη διάρκεια της διαδικασίας αυτής οι παράμετροι του μοντέλου (π.χ. η συνάρτηση-πυρήνας, οι ελεύθερες παράμετροι αυτής και ο συντελεστής κόστους C, για ένα μοντέλο SVM) διατηρούνται σταθερές σε όλα τα δείγματα bootstrap. Συνήθως οι

παράμετροι αυτές είναι οι βέλτιστες, όπως έχουν προσδιοριστεί με βάσει τα όσα έχουν μέχρι τώρα αναφερθεί.

Για να υπολογιστεί το διάστημα εμπιστοσύνης, οι τιμές των σφαλμάτων που λήφθηκαν από καθένα από τα δείγματα bootstrap διατάσσονται κατ' αύξουσα σειρά και υπολογίζεται το άνω και κάτω όριο του διαστήματος, με βάση τα ποσοστημόρια που προκύπτουν από το επιθυμητό επίπεδο εμπιστοσύνης.

Στο σημείο αυτό έχουμε επιτύχει τους εξής στόχους: Έχουμε αντιμετωπίσει τον κίνδυνο μεροληψίας ή διακύμανσης του μοντέλου και έχουμε προσδιορίσει την αβεβαιότητα του σφάλματος εξαιτίας του συγκεκριμένου συνόλου δεδομένων D.

7.6.1 Σύγκριση μοντέλων

Συχνά συμβαίνει στην πράξη να διαθέτουμε δύο ή περισσότερα διαφορετικά μοντέλα για ένα συγκεκριμένο πρόβλημα εξόρυξης γνώσης. Αυτά είναι μοντέλα που πιθανόν έχουν προκύψει από διαφορετικούς αλγορίθμους μάθησης και που όλα έχουν ρυθμιστεί ώστε να αποδίδουν ικανοποιητικά, αλλά που μόνο ένα από αυτά θα πρέπει να επιλεγεί για να υλοποιηθεί. Πιθανόν να έχουμε τη δυνατότητα να επιλέξουμε μεταξύ ενός λιγότερο πολύπλοκου μοντέλου με μεγαλύτερο όμως σφάλμα κι ενός πιο σύνθετου με μικρότερο σφάλμα. Το πρόβλημα αυτό δεν είναι τόσο τετριμμένο γιατί τα λιγότερο πολύπλοκα μοντέλα είναι συνήθως περισσότερο ελκυστικά. Η ερώτηση την οποία καλούμαστε να απαντήσουμε είναι η εξής: *Οι αποδόσεις των δύο μοντέλων διαφέρουν σημαντικά μεταξύ τους*;

Για την απάντηση στο ερώτημα αυτό μπορεί να κατασκευαστεί ένα πολλαπλό διάστημα εμπιστοσύνης του σφάλματος (για παράδειγμα με τη μέθοδο Bonferonni), στο επιθυμητό επίπεδο εμπιστοσύνης. Αν τα διαστήματα που αντιστοιχούν στα δύο μοντέλα δεν επικαλύπτονται, οι αποδόσεις τους θεωρείται ότι διαφέρουν σημαντικά μεταξύ τους και θα πρέπει να επιλεγεί το μοντέλο που παρουσιάζει την καλύτερη απόδοση. Αν από την άλλη μεριά τα δύο διαστήματα επικαλύπτονται, οι αποδόσεις των δύο μοντέλων *δε* θεωρούνται σημαντικά διαφορετικές και η επιλογή μπορεί να βασιστεί σε άλλα κριτήρια, όπως η πολυπλοκότητα του μοντέλου.

Παράρτημα

Εφαρμογή RLR: Πρόβλεψη Καταλληλότητας Προϊόντος, (Σελ. 34)

Υλοποίηση σε Octave

Regularized_Logistic.m

```
%% Initialization
clear all; close all; clc

%% Load Data
data = load('ex2data2.txt');
X = data(:, [1, 2]); y = data(:, 3);

plotData(X, y);

% Put some labels
hold on;

% Labels and Legend
xlabel('Microchip Test 1')
ylabel('Microchip Test 2')

% Specified in plot order
legend('y = 1', 'y = 0')
hold off;

% Add Polynomial Features. Note that mapFeature also adds a column of ones, so
% the intercept term is handled
X = mapFeature(X(:,1), X(:,2));

% Initialize fitting parameters
initial_theta = zeros(size(X, 2), 1);
% Set regularization parameter lambda to 1 (you should vary this)
lambda = 1;

% Set Options
options = optimset('GradObj', 'on', 'MaxIter', 50000, 'MaxFunEvals', 50000);

% Optimize
[theta, J, exit_flag] = ...
    fminunc(@(t)(costFunctionReg(t, X, y, lambda)), initial_theta, options);
```

sigmoid.m

```
function g = sigmoid(z)
% sigmoid computes the sigmoid of z.

g = 1./(1+exp(-z));

end
```

costFunctionReg.m

```
function [J, grad] = costFunctionReg(theta, X, y, lambda)
%Compute cost and gradient for logistic regression with regularization
%   J = COSTFUNCTIONREG(theta, X, y, lambda) computes the cost of using
%   theta as the parameter for regularized logistic regression and the
%   gradient of the cost w.r.t. the parameters.

m = length(y); % number of training examples

% COST FUNCTION
J = 1/m * sum(-y.*log(sigmoid(X*theta))- ...
    (1-y).*log(1-sigmoid(X*theta))) + ...
    lambda/2/m * sum(theta(2:end).^2) ;

%   gradient
grad = 1/m * X' * (sigmoid(X*theta) - y) + ...
    lambda/m * [0;theta(2:end)];

end
```

mapFeature.m

```
function out = mapFeature(X1, X2)
% Feature mapping function to polynomial features
%
%   MAPFEATURE(X1, X2) maps the two input features to quadratic
features.
%   Returns a new feature array with more features, comprising of
%   X1, X2, X1.^2, X2.^2, X1*X2, X1*X2.^2, etc..
%
%   Inputs X1, X2 must be the same size
%

degree = 6;
out = ones(size(X1(:,1)));
for i = 1:degree
    for j = 0:i
        out(:, end+1) = (X1.^(i-j)).*(X2.^j);
    end
end

end
```

Υλοποίηση σε R

```
#####################################################
### FUNCTIONS

sigmoid <- function(z) {
    return(1/(1 + exp(-z)))
}

mapFeature <- function(X1, X2) {
    degree <- 6
    out <- rep(1, length(X1))
    for (i in 1:degree) {
        for (j in 0:i) {
            out <- cbind(out, (X1^(i - j)) * (X2^j))
        }
    }
    return(out)
}

## Cost Function
fr <- function(theta, X, y, lambda) {
    m <- length(y)
    return(1/m * sum(-y * log(sigmoid(X %*% theta)) - (1 - y) *
       log(1 - sigmoid(X %*% theta))) + lambda/2/m * sum(theta[-1]^2))
}

## Gradient
grr <- function(theta, X, y, lambda) {
    return(1/m * t(X) %*% (sigmoid(X %*% theta) - y) + lambda/m *
       c(0, theta[-1]))
}

#####################################################
```

```
data <- read.csv("ex2data2.txt", header = F)

X = data[,c(1,2)]
y = data[,3]
X = mapFeature(X[,1],X[,2])

m <- nrow(X)
n <- ncol(X)

initial_theta = rep(0, n)

lambda <- 1

res <- optim(initial_theta, fr, grr, X, y, lambda,
    method = "BFGS", control = list(maxit = 100000))
```

Εφαρμογή RLR: Ανάγνωση Χειρόγραφων Ψηφίων, (Σελ.40)

Υλοποίηση σε Octave

Regularized_one_vs_all_Logistic.m

```
%% Initialization
clear ; close all; clc

%% Setup the parameters
input_layer_size  = 400;  % 20x20 Input Images of Digits
num_labels = 10;          % 10 labels, from 1 to 10
                          % (note that "0" is mapped to label 10)

%% =========== Part 1: Loading and Visualizing Data =============
% Load Training Data
load('ex3data1.mat'); % training data stored in arrays X, y
m = size(X, 1);

% Randomly select 100 data points to display
rand_indices = randperm(m);
sel = X(rand_indices(1:100), :);

displayData(sel);

%% ============ Part 2: One-vs-All Logistic Regression ============
lambda = 0.1;
[all_theta] = oneVsAll(X, y, num_labels, lambda);

%% ================ Part 3: Predict for One-Vs-All ================
pred = predictOneVsAll(all_theta, X);

fprintf('\nTraining Set Accuracy: %f\n', mean(double(pred == y)) * 100);

%  Run through the examples one at the a time to see what it is predicting.

%  Randomly permute examples
rp = randperm(m);

for i = 1:m
    % Display
    fprintf('\nDisplaying Example Image\n');
    displayData(X(rp(i), :));

    pred = predictOneVsAll(all_theta, X(rp(i),:));
    fprintf('\nOne-vs-All: %d (digit %d)\n', pred, mod(pred, 10));

    % Pause
    fprintf('Program paused. Press enter to continue.\n');
    pause;
end
```

oneVsAll.m

```
function [all_theta] = oneVsAll(X, y, num_labels, lambda)
%ONEVSALL trains num_labels logistic regression classifiers and returns all
%the classifiers in a matrix all_theta, where the i-th row of all_theta
%corresponds to the classifier for label i

m = size(X, 1);
n = size(X, 2);

% You need to return the following variable
all_theta = zeros(num_labels, n + 1);

% Add ones to the X data matrix
X = [ones(m, 1) X];

for c = 1:num_labels

    % Set Initial theta
    initial_theta = zeros(n + 1, 1);

    % Set options for fmincg
    options = optimset('GradObj', 'on', 'MaxIter', 50);

    % Run fmincg to obtain the optimal theta
    % This function will return theta and the cost
    [theta] = ...
        fmincg (@(t)(lrCostFunction(t, X, (y == c), lambda)), ...
                initial_theta, options);

    all_theta(c,:) = theta';
end

end
```

predictOneVsAll.m

```
function p = predictOneVsAll(all_theta, X)
%PREDICT Predict the label for a trained one-vs-all classifier. The labels
%are in the range 1..K, where K = size(all_theta, 1).

m = size(X, 1);

% Add ones to the X data matrix
X = [ones(m, 1) X];

% Get the matrix of probabilities. Rows are cases, Columns are classes.
all_p = sigmoid(X*all_theta');

% Get the max value and index for each row (2) of the matrix
% p contains the indices of the maximum values
[C,p] = max(all_p,[],2);

end
```

Εφαρμογή MLP: Ανάγνωση Χειρόγραφων Ψηφίων, (σελ.65)

Υλοποίηση σε Octave

regularized_nn_training.m

```
%% Initialization
clear ; close all; clc

%% Setup the parameters you will use
input_layer_size  = 400;  % 20x20 Input Images of Digits
hidden_layer_size = 25;   % 25 hidden units
num_labels = 10;          % 10 labels, from 1 to 10
                          % (note that we have mapped "0" to label 10)

% Load Training Data
load('ex4data1.mat');
m = size(X, 1);

%% ================ Initializing Pameters ================
initial_Theta1 = randInitializeWeights(input_layer_size, hidden_layer_size);
initial_Theta2 = randInitializeWeights(hidden_layer_size, num_labels);

% Unroll parameters
initial_nn_params = [initial_Theta1(:) ; initial_Theta2(:)];

%% =============== Train the Network ===============
%  To train the neural network, we will now use "fmincg", which
%  is a function which works similarly to "fminunc". Recall that these
%  advanced optimizers are able to train our cost functions efficiently as
%  long as we provide them with the gradient computations.

%  Change the MaxIter to a larger values to see how more training helps.
options = optimset('MaxIter', 400);
%  You should also try different values of lambda
lambda = 1;
% Create "short hand" for the cost function to be minimized
costFunction = @(p) nnCostFunction(p, ...
                                   input_layer_size, ...
                                   hidden_layer_size, ...
                                   num_labels, X, y, lambda);

% Now, costFunction is a function that takes in only one argument (the
% neural network parameters)
[nn_params, cost] = fmincg(costFunction, initial_nn_params, options);

% Obtain Theta1 and Theta2 back from nn_params
Theta1 = reshape(nn_params(1:hidden_layer_size * (input_layer_size + 1)), ...
                 hidden_layer_size, (input_layer_size + 1));

Theta2 = reshape(nn_params((1 + (hidden_layer_size * (input_layer_size + 1))):end), ...
                 num_labels, (hidden_layer_size + 1));

%% ================= Implement Predict =================
%  After training the neural network, we would like to use it to predict
%  the labels. This lets you compute the training set accuracy.

pred = predict(Theta1, Theta2, X);

fprintf('\nTraining Set Accuracy: %f\n', mean(double(pred == y)) * 100);
```

nnCostFunction.m

```
function [J grad] = nnCostFunction(nn_params, ...
                                   input_layer_size, ...
                                   hidden_layer_size, ...
                                   num_labels, ...
                                   X, y, lambda)

% Computes the cost and gradient of the neural network. The parameters for the
% neural network are "unrolled" into the vector nn_params and need to be converted back
% into the weight matrices. The returned parameter grad should be a "unrolled" vector
% of the partial derivatives of the neural network.
% Reshape nn_params back into the parameters Theta1 and Theta2, the weight matrices
% for our 2 layer neural network
Theta1 = reshape(nn_params(1:hidden_layer_size * (input_layer_size + 1)), ...
                 hidden_layer_size, (input_layer_size + 1));

Theta2 = reshape(nn_params((1 + (hidden_layer_size * (input_layer_size + 1))):end), ...
                 num_labels, (hidden_layer_size + 1));

% Setup some useful variables
m = size(X, 1);

% You need to return the following variables correctly
J = 0;
Theta1_grad = zeros(size(Theta1));
Theta2_grad = zeros(size(Theta2));

% Feedforward the neural network and return the cost in the variable J.

% Recode the labes as vectors to contain only 0 and 1
new_y = eye(num_labels)(y,:);

% Add ones to the X data matrix (bias terms)
X = [ones(m, 1) X];

% get from layer 1 to layer 2
z2 = Theta1*X';
a2 = sigmoid(z2);  % (25x5000)

% add a row of ones for the bias terms
a2 = [ones(1,m); a2]; % (26x5000)

% get from layer 2 to output layer (3)
z3 = Theta2*a2;
a3 = sigmoid(z3); % (10x5000)

% Cost
% size(new_y)        = 5000  x   10
% size(a3)           =   10  x 5000
% size(new_y' .* a3) =   10  x 5000
% add by columns to get   1  x 5000  using sum(...,1) then sum the resulting values

J = 1/m * sum(sum(-new_y' .* log(a3) - (1-new_y') .* log(1-a3),1)) + ...
lambda/2/m * (sum(sum(Theta1(:,2:end).^2)) + sum(sum(Theta2(:,2:end).^2)));

% Implement the backpropagation algorithm to compute the gradients Theta1_grad and
% Theta2_grad. You should return the partial derivatives of the cost function with
% respect to Theta1 and Theta2 in Theta1_grad and Theta2_grad, respectively.
d3 = a3 - new_y';
d2 = (Theta2' * d3)(2:end,:) .* sigmoidGradient(z2);  % (25x5000)
Delta_2 = d3 * a2';  % (10x26)
Delta_1 = d2 * X;    % (25x401)
Theta2_grad = 1/m * Delta_2 + lambda/m * [zeros(num_labels,1),Theta2(:,2:end)];
Theta1_grad = 1/m * Delta_1 + lambda/m * [zeros(hidden_layer_size,1),Theta1(:,2:end)];

% Unroll gradients
grad = [Theta1_grad(:) ; Theta2_grad(:)];
% =========================================================================
end
```

sigmoidGradient.m

```
function g = sigmoidGradient(z)

%   g = sigmoidGradient(z) computes the gradient of the sigmoid function
%   evaluated at z. This should work regardless if z is a matrix or a
%   vector. In particular, if z is a vector or matrix, you should return
%   the gradient for each element.

g = sigmoid(z) .* (1 - sigmoid(z));

% =============================================================
end
```

sigmoid.m

```
function g = sigmoid(z)
%g = sigmoid(z) computes the sigmoid of z.

g = 1.0 ./ (1.0 + exp(-z));
end
```

randInitializeWeights.m

```
function W = randInitializeWeights(L_in, L_out)
% Randomly initialize the weights of a layer with L_in
% incoming connections and L_out outgoing connections
% Note: The first column of W corresponds to the parameters for the bias units

epsilon_init = epsilon_init = sqrt(6)/sqrt(L_in + L_out);
W = rand(L_out, 1 + L_in) * 2 * epsilon_init - epsilon_init;

% =============================================================
end
```

predict.m

```
function p = predict(Theta1, Theta2, X)
%   p = predict(Theta1, Theta2, X) outputs the predicted label of X given the
%   trained weights of a neural network (Theta1, Theta2)

m = size(X, 1);
num_labels = size(Theta2, 1);

p = zeros(size(X, 1), 1);

h1 = sigmoid([ones(m, 1) X] * Theta1');
h2 = sigmoid([ones(m, 1) h1] * Theta2');
[dummy, p] = max(h2, [], 2);

% =============================================================
end
```

Εφαρμογή SVM: Ταξινόμηση Ανεπιθύμητης Αλληλογραφίας, (σελ.138)

Υλοποίηση σε Octave

svm_spam_filter.m

```
%% Spam Classification with SVMs

clear ; close all; clc

%% ==================== Email Preprocessing ====================
%  To use an SVM to classify emails into Spam v.s. Non-Spam, you first need
%  to convert each email into a vector of features. In this part, I will
%  implement the preprocessing steps for each email.

% Extract Features
file_contents = readFile('emailSample1.txt');
word_indices  = processEmail(file_contents);

% Print Stats
fprintf('Word Indices: \n');
fprintf(' %d', word_indices);
fprintf('\n\n');

features = emailFeatures(word_indices);

% Print Stats
fprintf('Length of feature vector: %d\n', length(features));
fprintf('Number of non-zero entries: %d\n', sum(features > 0));

%% =========== Train Linear SVM for Spam Classification ========
%  Train a linear classifier to determine if an
%  email is Spam or Not-Spam.

% Load the Spam Email dataset
% You will have X, y in your environment
load('spamTrain.mat');

fprintf('\nTraining Linear SVM (Spam Classification)\n')
fprintf('(this may take 1 to 2 minutes) ...\n')

C = 0.1;
model = svmTrain(X, y, C, @linearKernel);
p = svmPredict(model, X);

fprintf('Training Accuracy: %f\n', mean(double(p == y)) * 100);

% Add the path for the LIBSVM library
addpath ('C:\libraries\libsvm-3.11\matlab');

model_LSVM = svmtrain(y, X, ['-c ',num2str(C), ' -t 0']);
p_LSVM = svmpredict(y, X, model_LSVM);

fprintf('Training Accuracy LIBSVM: %f\n', mean(double(p_LSVM == y)) * 100);
```

svm_spam_filter.m (ΣΥΝΕΧΕΙΑ)

```
%% ==================== Test Spam Classification ================
%  After training the classifier, we can evaluate it on a test set
%  (spamTest.mat)

% Load the test dataset
% You will have Xtest, ytest in your environment
load('spamTest.mat');

fprintf('\nEvaluating the trained Linear SVM on a test set ...\n')

p = svmPredict(model, Xtest);

fprintf('Test Accuracy: %f\n', mean(double(p == ytest)) * 100);

[p_LSVM, accuracy, decision_values] = svmpredict(ytest, Xtest, model_LSVM);
fprintf('Test Accuracy LIBSVM: %f\n', mean(double(p_LSVM == ytest)) * 100);

%% ================= Top Predictors of Spam ====================
%  Since the model we are training is a linear SVM, we can inspect the
%  weights learned by the model to understand better how it is determining
%  whether an email is spam or not. The following code finds the words with
%  the highest weights in the classifier. Informally, the classifier
%  'thinks' that these words are the most likely indicators of spam.
%

% Sort the weights and obtin the vocabulary list
[weight, idx] = sort(model.w, 'descend');
vocabList = getVocabList();

fprintf('\nTop predictors of spam: \n');
for i = 1:15
    fprintf(' %-15s (%f) \n', vocabList{idx(i)}, weight(i));
end


%% ==================== Try Your Own Emails =====================
%  Now that you've trained the spam classifier, you can use it on your own
%  emails! In the example folder I have included spamSample1.txt,
%  spamSample2.txt, emailSample1.txt and emailSample2.txt as examples.
%  The following code reads in one of these emails and then uses your
%  learned SVM classifier to determine whether the email is Spam or
%  Not Spam

% Set the file to be read in (change this to spamSample2.txt,
% emailSample1.txt or emailSample2.txt to see different predictions on
% different emails types). Try your own emails as well!
filename = 'emailSample1.txt';

% Read and predict
file_contents = readFile(filename);
word_indices  = processEmail(file_contents);
x             = emailFeatures(word_indices);
p = svmPredict(model, x);

fprintf('\nProcessed %s\n\nSpam Classification: %d\n', filename, p);
fprintf('(1 indicates spam, 0 indicates not spam)\n\n');

p_LSVM = svmpredict(0, x', model_LSVM);

fprintf('\nSpam Classification: %d\n', p_LSVM);
fprintf('(1 indicates spam, 0 indicates not spam)\n\n');
```

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] ABE, S. *Support Vector Machines for Pattern Classification*. Second Ed. Springer, 2010.

[2] ALPAYDIN, E. *Introduction to Machine Learning*. Second Ed. MIT Press, 2010.

[3] BENNETT, C. J., CAMPBEL, C. Support Vector Machines: Hype or Hallelujah? *SIGKDD Explorations*, Vol.2, No. 2, 1-13, 2000.

[4] BISHOP, M. C. *Pattern Recognition and Machine Learning*. Springer, 2006.

[5] BOYD, P. S. Lecture Notes for SEE course: Convex Optimization I. Available in http://see.stanford.edu/see/courseInfo.aspx?coll=2db7ced4-39d1-4fdb-90e8-364129597c87. Stanford University, 2008.

[6] BOYD, P. S., VANDENBERGHE, L. *Convex Optimization*. (e-Book, Lecture slides, etc. can be found in http://www.stanford.edu/~boyd/cvxbook/). Cambridge University Press, 2004.

[7] CHANG, C. C., LIN, J. C. LIBSVM: a Library for Support Vector Machines. [Online]. Software available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm, detailed documentation (algorithms, formulae, ...) can be found in http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.ps.gz , 2001.

[8] COOK, D., SWAYNE, F. D. *Interactive and Dynamic Graphics for Data Analysis With R and GGobi*. Springer, 2007.

[9] DIAMANTARAS, I. K. *Artificial Neural Networks (in Greek)*. Klidarithmos, 2007.

[10] HAMEL, L. *Knowledge Discovery with Support Vector Machines*. Wiley, 2009.

[11] HASTIE, T., ROSSET, S., TIBSHIRANI, R., ZHU, J. The Entire Regularization Path for the Support Vector Machine. *Journal of Machine Learning Research* 5, no.1: p. 1391-1415, 2004.

[12] HASTIE, T., TIBSHIRANI, R., FRIEDMAN, J. *The Elements of Statistical Learning*. Second Ed. Springer-Verlag, 2009.

[13] HAYKIN, S. *Neural Networks – A Comprehensive Foundation*. Second Ed. Prentice-Hall, 1999.

[14] HEARST, A. M., SCHOLKOPF, B. S., DUMAIS, S., OSUNA, E., PLATT, J. Trends and Controversies – Support Vector machines. *IEEE Intelligent Systems*, 13(4):18-28, 1998.

[15] HSU, C. W., CHANG, C. C., LIN, J. C. A Practical Guide to Support Vector Classification. *Bioinformatics* 1, no. 1, p.1-16, 2010.

[16] IZENMANN, J. A. *Modern Multivariate Statistical Techniques*. Springer, 2008.

[17] JENSEN, S. An Introduction to Lagrange Multipliers. [Online]: http://www.slimy.com/~steuard/teaching/tutorials/Lagrange.html

[18] KARATZOGLOU, A., MEYER, D., HORNIK, K. Support Vector Machines in R. *Journal of Statistical Software*, Vol. 15, Issue 9, http://www.jstatsoft.org/v15/i09/ April 2006.

[19] KUHN, M. Building predictive models in R using the caret package. *Journal of Statistical Software*, Vol. 28, Issue 5, http://www.jstatsoft.org/v28/i05/ Nov. 2008.

[20] KULKARNI, S., HARMAN, G. *An Elementary Introduction to Statistical Learning Theory*. Wiley, 2011.

[21] LU, C. Probabilistic machine learning approaches to medical classification problems. PhD thesis, Faculty of Engineering, Katholieke Universiteit Leuven, 2005.

[22] MEYER, D. Support vector machines: The interface to libsvm in package e1071. Technische Universität Wien, 2009.

[23] MITCHELL, M. T. *Machine Learning*. McGraw-Hill, 1997.

[24] NASH, C. J., VARADHAN, R. Unifying Optimization Algorithms to Aid Software System Users: optimx for R. *Journal of Statistical Software*, Vol. 43, Issue 9, http://www.jstatsoft.org/v43/i09/ Aug. 2011.

[25] NG, Y. A. Lecture Notes for SEE course: Introduction to Machine Learning. Available in http://ml-class.org. Stanford University, 2011.

[26] NG, Y. A. Lecture Notes for CS229 SEE course: Machine Learning. Available in http://see.stanford.edu/see/courseinfo.aspx?coll=348ca38a-3a6d-4052-937d-cb017338d7b1. Stanford University, 2008.

[27] NOBLE, S. W. What is a support vector machine? *Nature Biotechnology* 24, 1565 - 1567, 2006.

[28] PLATT, C. J. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. In Technical Report MST-TR-98-14. Microsoft Research, 1998.

[29] PORTER, F. M. An algorithm for suffix stripping. *Program*, Vol. 14, no. 3, p. 130-137, 1980.

[30] SCHOLKOPF, B. S., SMOLA, J. A. *Learning with Kernels – Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002.

[31] SHAWE-TAYLOR, J., CRISTIANINI, N. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[32] STEINWART, I., CHRISTMANN, A. *Support Vector Machines*. Springer, 2008.

[33] STEWART, J. *Calculus*. Fourth Ed. Brooks/Cole Publishing Company, 1999.

[34] VENABLES, N. W., RIPLEY, D. B. *Modern Applied Statistics with S*. Fourth Ed. Springer, 2002.

[35] WILLIAMS, G. *Data Mining with Rattle and R*. Springer, 2011.

[36] WITTEN, H. I., FRANK, E., HALL, A. M. *Data Mining. Practical Machine Learning Tools and Techniques*. Third Ed. Morgan Kaufmann, 2011.