Πανεπιστήμιο Ιωαννίνων
Σχολή Θετικών Επιστημών
Τμήμα Πληροφορικής

Πτυχιακή Εργασία

ΤΑΞΙΝΟΜΗΣΗ ΚΕΙΜΕΝΩΝ ΗΛΕΚΤΡΟΝΙΚΟΥ ΤΑΧΥΔΡΟΜΕΙΟΥ ΜΕ ΒΑΣΗ ΤΟ ΠΕΡΙΕΧΟΜΕΝΟ

ΓΡΗΓΟΡΙΟΣ ΤΖΩΡΤΖΗΣ

ΕΠΙΒΛΕΠΩΝ: Α. ΛΥΚΑΣ

Ιωάννινα, Ιούνιος 2006

Πρόλογος

Στην εργασία αυτή εξετάζεται το πρόβλημα της ταξινόμησης κειμένων ηλεκτρονικού ταχυδρομείου με βάση το περιεχόμενο. Στόχος της εργασίας είναι να παρουσιάσει ορισμένες μεθόδους μηχανικής μάθησης που εφαρμόζονται στην επίλυση αυτού. Συγκεκριμένα μελετάται η μέθοδος SVM (Support Vector Machine) καθώς και μέθοδοι ενεργητικής μάθησης με SVM. Πέρα από τις μεθόδους γίνεται και μία αναφορά σε φίλτρα που έχουν υλοποιηθεί και χρησιμοποιούνται ευρέως στην ταξινόμηση ηλεκτρονικού ταχυδρομείου.

Η παρούσα πτυχιακή εργασία εκπονήθηκε στα πλαίσια του προπτυχιακού προγράμματος σπουδών του Τμήματος Πληροφορικής του Πανεπιστημίου Ιωαννίνων. Πραγματοποιήθηκε υπό την επίβλεψη του Αν. Καθ. Α. Λύκα, τον οποίο ευχαριστώ θερμά για τη στενή συνεργασία και τον εποικοδομητικό διάλογο, που υπήρξαν καθοριστικής σημασίας για την ολοκλήρωσή της.

Ιωάννινα, Ιούνιος 2006

Γρηγόριος Τζώρτζης

Περιεχόμενα

Κεφάλαιο 1: Εισαγωγή

Η ολοένα και αυξανόμενη χρήση του ηλεκτρονικού ταχυδρομείου ιδιαίτερα τα τελευταία χρόνια σε συνδυασμό με το μηδενικό κόστος του έχει δώσει την ευκαιρία σε άτομα, γνωστά ως spammers, που θέλουν να διαφημίσουν τα προϊόντα και τις υπηρεσίες που παρέχουν να το πράττουν πολύ εύκολα. Τα άτομα αυτά αποστέλλουν μαζικά σε χιλιάδες χρήστες ταυτόχρονα ένα μήνυμα που περιέχει οτιδήποτε θέλουν να διαφημίσουν το οποίο μπορεί να είναι από διακοπές έως τρόποι για να γίνουν πλούσιοι ακόμη και πορνογραφικό υλικό. Τα μηνύματα αυτά αποκαλούνται spam.

Στόχος αυτών είναι να προσελκύσουν το ενδιαφέρον έστω και ενός μικρού αριθμού ατόμων απ' όλα τα άτομα προς τα οποία έχουν αποσταλεί καθώς όπως είπαμε το κόστος τους είναι αμελητέο. Μάλιστα ότι προβάλλεται σε αυτά μπορεί να είναι ψεύτικο και να αποσκοπεί στην εξαπάτηση του παραλήπτη. Η δουλειά των spammers γίνεται ακόμη πιο απλή από την ύπαρξη λογισμικού για μαζική αποστολή μηνυμάτων ηλεκτρονικού ταχυδρομείου καθώς και λιστών με διευθύνσεις χρηστών. Οι λίστες αυτές έχουν δημιουργηθεί από «κλοπή» διευθύνσεων μέσω ιστοσελίδων, από ομάδες συζητήσεων (newsgroups) κ.α.

Αυτή η μαζική αποστολή μηνυμάτων δημιουργεί μία σειρά από προβλήματα. Καταρχήν το γραμματοκιβώτιο κάθε χρήστη κατακλύζεται από δεκάδες spam μηνύματα το οποίο είναι πολύ ενοχλητικό καθώς πρέπει να «ψάξει» για να βρει τα μηνύματα που τον ενδιαφέρουν. Επίσης καταναλώνει το εύρος ζώνης και άλλους πόρους των παρόχων υπηρεσιών Internet (ISPs) και εκθέτει ανήλικους σε ακατάλληλο υλικό. Μάλιστα έρευνες έχουν δείξει ότι το πλήθος spam που λαμβάνει ένας χρήστης έχει διπλασιαστεί μεταξύ 2002 και 2006 και πλέον ξεπερνά τα 1600 μηνύματα ετησίως. Αυτά μας δείχνουν ότι είναι

απαραίτητο να λάβουμε μέτρα ενάντια στα spam μηνύματα και αυτό κυρίως γίνεται μέσω φίλτρων που ανακαλύπτουν τέτοια μηνύματα. Στην εργασία αυτή παρουσιάζουμε μεθόδους για κατασκευή ταξινομητών για το διαχωρισμό των μηνυμάτων που λαμβάνει ένας χρήστης ώστε η ζωή του να γίνει ευκολότερη.

1.1 Το πρόβλημα της ταξινόμησης κειμένων ηλεκτρονικού ταχυδρομείου

Στην εργασία αυτή μελετούμε το πρόβλημα της ταξινόμησης κειμένων ηλεκτρονικού ταχυδρομείου με βάση το περιεχόμενο. Το πρόβλημα αυτό αποτελεί μία εφαρμογή του γενικότερου προβλήματος της ταξινόμησης κειμένων (text classification).

Η ταξινόμηση κειμένων έχει να κάνει με την ανάθεση σε ένα κείμενο της κατηγορίας στην οποία αυτό ανήκει από ένα σύνολο προκαθορισμένων κατηγοριών. Στην περίπτωση των μηνυμάτων ηλεκτρονικού ταχυδρομείου υπάρχουν δύο κατηγορίες: η κατηγορία spam που υποδηλώνει ότι ένα μήνυμα είναι ανεπιθύμητο και προέρχεται από έναν αποστολέα με τον οποίο δεν θέλουμε να επικοινωνούμε και η κατηγορία μη-spam που υποδηλώνει ότι το μήνυμα πρέπει να παραδοθεί στα εισερχόμενα του χρήστη καθώς το περιεχόμενό του είναι χρήσιμο για τις δραστηριότητές του.

Στόχος μας είναι η παραπάνω διαδικασία να μπορεί να γίνεται αυτόματα ώστε οι χρήστες ηλεκτρονικού ταχυδρομείου να διευκολυνθούν στον διαχωρισμό της αλληλογραφίας τους. Ωστόσο πρέπει να γίνεται και με ασφάλεια, δηλαδή να μην καταλήγουν μηνύματα που ενδιαφέρουν τον χρήστη στην κατηγορία spam (false positives). Στην εργασία θα δούμε τρόπους για να επιτύχουμε και τα δύο.

1.2 Ορισμός του προβλήματος της ταξινόμησης κειμένων ηλεκτρονικού ταχυδρομείου

Στο σημείο αυτό επιχειρούμε να δώσουμε έναν τυπικό ορισμό (σύμφωνα με το [8]) του προβλήματος που μελετάμε με όρους της ταξινόμησης κειμένων. Έστω ένα σύνολο μηνυμάτων $D = \{d_1, d_2, \ldots, d_{|D|}\}$ και ένα σύνολο κατηγοριών $C = \{c_1 = \text{spam}, c_2 = \text{μη} - \text{spam}\}$. Στόχος μας είναι να αναθέσουμε σε κάθε ζεύγος $\langle d_j, c_i \rangle \in D \times C$ μία λογική τιμή. Η ανάθεση της τιμής T (αληθής) σημαίνει ότι το μήνυμα

d_j ανήκει στην κατηγορία c_i ενώ η ανάθεση της τιμής F (ψευδής) σημαίνει ότι το d_j δεν ανήκει στην c_i.

Ουσιαστικά θέλουμε με μία συνάρτηση $\Phi : D \times C \rightarrow \{T, F\}$ την οποία ονομάζουμε ταξινομητή (classifier) να προσεγγίσουμε μία άγνωστη συνάρτηση στόχο (target function) $\breve{\Phi} : D \times C \rightarrow \{T, F\}$ η οποία περιγράφει πως θα έπρεπε να ταξινομηθούν τα μηνύματα. Πρέπει ο ταξινομητής μας και η άγνωστη συνάρτηση να συμπίπτουν όσο το δυνατό περισσότερο.

Μάλιστα το πρόβλημα της ταξινόμησης μηνυμάτων ηλεκτρονικού ταχυδρομείου όπως είδαμε αποτελείται μόνο από δύο κατηγορίες και πρόκειται για δυαδικό πρόβλημα. Δηλαδή ένα μήνυμα είτε ανήκει στην μία κατηγορία είτε στην άλλη. Άρα μας αρκεί να ορίσουμε τον ταξινομητή μας ως: $\Phi_{spam} : D \rightarrow \{T, F\}$. Αν η απόφαση είναι T τότε το μήνυμα είναι spam ενώ αν είναι F είναι μη-spam.

1.3 Προσεγγίσεις στο φιλτράρισμα του ηλεκτρονικού ταχυδρομείου

Οι τεχνικές που χρησιμοποιούνται στο φιλτράρισμα των μηνυμάτων ηλεκτρονικού ταχυδρομείου προέρχονται από το ευρύτερο πεδίο της ταξινόμησης κειμένων όπου υπάρχουν δύο προσεγγίσεις. Η πρώτη εξ αυτών γνώρισε μεγάλη άνθιση τη δεκαετία του 1980 και είναι η μηχανική γνώσης (knowledge engineering - KE). Η δεύτερη είναι η μηχανική μάθηση (machine learning - ML) και από τις αρχές του 1990 έγινε πολύ δημοφιλής και πλέον αποτελεί το επίκεντρο των ερευνητικών προσπαθειών. Στη συνέχεια παρουσιάζουμε τα κύρια στοιχεία των προσεγγίσεων αυτών κυρίως από την σκοπιά του φιλτραρίσματος του ηλεκτρονικού ταχυδρομείου.

1.3.1 Μηχανική γνώσης

Η τεχνική αυτή στηρίζεται σε κανόνες (rules), έναν για κάθε κατηγορία, που είναι της μορφής:

if <DNF formula> **then** <category>

Η DNF (Disjunctive Normal Form) μορφή είναι ένα σύνολο από διαζεύξεις συζεύξεων. Ένα κείμενο θα ταξινομηθεί στην κατηγορία <category> εφόσον ικανοποιεί τουλάχιστον μία συζευκτική πρόταση καθώς υπάρχει διάζευξη μεταξύ των προτάσεων. Στην περίπτωση των μηνυμάτων ηλεκτρονικού ταχυδρομείου οι κανόνες αυτοί ψάχνουν τα διάφορα τμήματα ενός μηνύματος (όπως σώμα, θέμα) για να εντοπίσουν κάποιους όρους κλειδιά η ύπαρξη των οποίων υποδηλώνει ότι το μήνυμα είναι spam. Καθώς το πρόβλημα που μελετούμε είναι δυαδικό αν ορίσουμε μία DNF πρόταση για την κατηγορία spam αυτομάτως ότι δεν την επαληθεύει θα είναι μη-spam. Άρα μας αρκεί ένας τέτοιος κανόνας Ένα παράδειγμα τέτοιου κανόνα φαίνεται παρακάτω.

if (((*Subject*) "$$")	**or**
((*Body*) "Guarantee" **and** (*Body*) "satisfaction")	**or**
((*Subject*) "adults" **and** (*Body*)"viagra"))	**then** *spam* **else** *non-spam*

Σχήμα 1.1: Παράδειγμα κανόνων σε ένα φίλτρο.

Στο παράδειγμα μέσα σε παρένθεση υπάρχει το τμήμα του μηνύματος που ερευνά ο κανόνας και δίπλα η λέξη που αναζητά. Σε ένα πραγματικό φίλτρο ο παραπάνω κανόνας θα είχε δεκάδες διαζεύξεις (κάθε διάζευξη αποτελεί και έναν διαφορετικό κανόνα που συναντάμε στα φίλτρα στην πράξη). Βεβαίως οι κανόνες ερευνούν πέρα από λέξεις όπως φάνηκε πριν και για πιο σύνθετα στοιχεία όπως μέγεθος γραμματοσειράς, χαρακτηριστικά της HTML κ.α.

Ένα θέμα είναι ποιος δημιουργεί τους κανόνες αυτούς. Συνήθως απαιτείται η συμβολή ενός μηχανικού γνώσης (knowledge engineer) που γνωρίζει το πεδίο της εφαρμογής. Όσον αφορά τα φίλτρα ηλεκτρονικού ταχυδρομείου οι κανόνες αυτοί δημιουργούνται από τους κατασκευαστές τους. Πολλές φορές όμως απαιτείται και ο απλός χρήστης να δημιουργήσει δικούς του ώστε να προσαρμόσει το φίλτρο στις ανάγκες του.

1.3.2 Μηχανική μάθηση

Στην προσέγγιση αυτή μία γενική επαγωγική διαδικασία (learner) αυτόματα κατασκευάζει έναν ταξινομητή για την κατηγορία c_i «παρατηρώντας» τα χαρακτηριστικά του

συνόλου εκπαίδευσης (training set), του οποίου τα έγγραφα προηγουμένως έχει χειρωνακτικά ταξινομήσει στην κατηγορία τους ένας ειδικός του πεδίου (domain expert). Η διαδικασία αυτή συγκεντρώνει τα χαρακτηριστικά που πρέπει να διαθέτει ένα άγνωστο έγγραφο d_j ώστε να ταξινομηθεί στην κατηγορία c_i.

Όσον αφορά το πρόβλημα του ηλεκτρονικού ταχυδρομείου αρκεί η κατασκευή ενός μόνο ταξινομητή καθώς πρόκειται για δυαδικό πρόβλημα Επιπλέον το μόνο που απαιτείται είναι η χειρωνακτική ταξινόμηση ορισμένων μηνυμάτων σε spam και μη-spam. Πρόκειται για μία διαδικασία η οποία, σε αντίθεση με άλλα πεδία όπου εφαρμόζεται η μηχανική μάθηση και απαιτείται ο διαχωρισμός να γίνει από έναν ειδικό, μπορεί να γίνει από έναν απλό χρήστη του ηλεκτρονικού ταχυδρομείου πολύ εύκολα.

Τα παραπάνω αποτελούν μία μορφή μάθησης με επίβλεψη (supervised learning) καθώς η διαδικασία της μάθησης «επιβλέπεται» από τη γνώση που έχουμε για το ποιες είναι οι κατηγορίες και ποια έγγραφα ανήκουν σε κάθε μία εξ αυτών.

1.3.3 Σύγκριση των προσεγγίσεων

Η τεχνική της μηχανικής γνώσης φέρει το μειονέκτημα που καλείται συμφόρηση απόκτησης γνώσης (knowledge acquisition bottleneck) και είναι γνωστό από το πεδίο των έμπειρων συστημάτων. Αυτό συνίσταται στο ότι για την δημιουργία των κανόνων είναι απαραίτητη η εργασία ειδικών στο πεδίο της εφαρμογής αλλά και στο ότι αν αλλάξουν κάποιοι παράμετροι του προβλήματος (π.χ. η προσθήκη μίας νέας κατηγορίας) πρέπει οι ειδικοί να συνεργαστούν και πάλι ώστε να προσαρμόσουν τους κανόνες στα νέα δεδομένα.

Στο πρόβλημα που εμείς μελετάμε η παραπάνω προσέγγιση φέρει και ορισμένα άλλα μειονεκτήματα. Πολλές φορές απαιτείται από τους απλούς χρήστες να δημιουργούν κανόνες για να προσαρμόσουν τα φίλτρα στις ανάγκες τους. Κάτι τέτοιο προϋποθέτει ότι ο χρήστης θα δαπανήσει χρόνο και κυρίως ότι έχει κάποια εμπειρία και μπορεί να αναγνωρίσει χαρακτηριστικά spam στα μηνύματα που λαμβάνει. Επιπλέον είναι απαραίτητο και ο κατασκευαστής να παρέχει νέους κανόνες. Μάλιστα η στατική φύση των κανόνων σε συνδυασμό με το ότι αυτοί είναι διαθέσιμοι στο ευρύ κοινό δίνουν τη δυνατότητα σε έναν αποστολέα spam (spammer) να δοκιμάσει αν το μήνυμα που θα αποστείλει μπορεί να ξεγελάσει το φίλτρο και να το προσαρμόσει κατάλληλα.

Αντίθετα η τεχνική της μηχανικής μάθησης δεν απαιτεί τη συγγραφή κανόνων αλλά την ύπαρξη ενός αλγορίθμου για τη δημιουργία του ταξινομητή και ένα σύνολο εγγράφων ταξινομημένων χειρονακτικά. Έτσι αν γίνει κάποια αλλαγή στις παραμέτρους του

προβλήματος π.χ. προσθήκη μίας κατηγορίας, αρκεί ένα σύνολο εγγράφων για αυτή ώστε να τη «μάθει» ο ταξινομητής.

Στη περίπτωση του ηλεκτρονικού ταχυδρομείου οι ταξινομητές αυτοί μπορούν να προσαρμοσθούν σε νέα χαρακτηριστικά των spam μηνυμάτων αρκεί να έχουμε διαθέσιμο ένα σύνολο μηνυμάτων διαχωρισμένο μεταξύ των κατηγοριών. Επιπλέον καθώς κάθε χρήστης κατασκευάζει τον ταξινομητή με βάση τα μηνύματα που λαμβάνει αυτός είναι προσαρμοσμένος στα χαρακτηριστικά των δικών του μηνυμάτων. Άρα απαιτείται ο κατασκευαστής να παρέχει τον αλγόριθμο δημιουργίας του ταξινομητή και ο χρήστης τα διαχωρισμένα μεταξύ των κατηγοριών μηνύματα. Λόγω του τρόπου αυτού κατασκευής υπάρχει και μεγαλύτερη δυσκολία από την πλευρά του αποστολέα spam να ξεγελάσει το φίλτρο.

Από πλευράς απόδοσης η τεχνική της μηχανικής μάθησης έχει επιτύχει εντυπωσιακά αποτελέσματα όσον αφορά και το κόστος αλλά και την ορθότητα της ταξινόμησης με αποτέλεσμα πλέον να υπάρχουν φίλτρα που στηρίζονται σε αυτή και τα χρησιμοποιούμε στην καθημερινή μας ζωή. Ορισμένα τέτοια παρουσιάζονται στο κεφάλαιο 5.

1.4 Διάρθρωση της εργασίας

Στα κεφάλαια που ακολουθούν επικεντρωνόμαστε στη μελέτη μεθόδων μηχανικής μάθησης για την ταξινόμηση ηλεκτρονικού ταχυδρομείου. Στο κεφάλαιο 2 παρουσιάζουμε την προεπεξεργασία για τα μηνύματα ώστε να μετατραπούν σε μορφή κατάλληλη για τις μεθόδους καθώς και τον τρόπο κατασκευής ενός ταξινομητή και ορισμένα μέτρα αξιολόγησης. Στα επόμενα δύο κεφάλαια παρουσιάζουμε τις μεθόδους που μελετήσαμε. Συγκεκριμένα στο κεφάλαιο 3 περιγράφεται η μέθοδος SVM και στο κεφάλαιο 4 ασχολούμαστε με ενεργητική μάθηση με χρήση SVM και αναφέρουμε δύο μεθόδους. Στο κεφάλαιο 5 κάνουμε μία μικρή αναφορά σε φίλτρα που έχουν υλοποιηθεί και χρησιμοποιούνται από τους χρήστες ηλεκτρονικού ταχυδρομείου καθημερινά ώστε να δούμε πως από τη θεωρία και την έρευνα περνούμε στην πράξη. Περιγράφουμε τα κύρια χαρακτηριστικά των μηχανισμών που υλοποιούνται στα φίλτρα αυτά. Τέλος το κεφάλαιο 6 περιέχει τα πειραματικά αποτελέσματα που προέκυψαν από τη μελέτη των μεθόδων SVM και ενεργητικής μάθησης και καταλήγουμε παρουσιάζοντας ορισμένα συμπεράσματα και θέματα για μελλοντική έρευνα.

Κεφάλαιο 2: Προεπεξεργασία, εκπαίδευση και αξιολόγηση ταξινομητών

2.1 Γενικά

Με την προσέγγιση της μηχανικής μάθησης (machine learning) είδαμε ότι μπορούμε να κατασκευάσουμε ταξινομητές οι οποίοι αυτόματα θα μπορούν να αναθέτουν την κατηγορία ενός νέου άγνωστου εγγράφου και κατ' επέκταση ενός μηνύματος ηλεκτρονικού ταχυδρομείου. Για να μπορέσουμε όμως να κατασκευάσουμε τον ταξινομητή απαιτούνται τα παρακάτω:

- προεπεξεργασία των μηνυμάτων,
- εκπαίδευση του ταξινομητή.

Η φάση της προεπεξεργασίας είναι απαραίτητη ώστε τα μηνύματα να μετατραπούν σε μορφή την οποία να μπορεί να επεξεργαστεί ο ταξινομητής. Η εκπαίδευση χρειάζεται ώστε ο ταξινομητής να εξάγει τα χαρακτηριστικά των μηνυμάτων κάθε κατηγορίας μέσα από ένα σύνολο χειρωνακτικά ταξινομημένων μηνυμάτων (σύνολο εκπαίδευσης) ώστε να κατορθώνει στη συνέχεια να βρίσκει την κατηγορία νέων μηνυμάτων.

Τέλος είναι απαραίτητο να αξιολογήσουμε την επίδοση του ταξινομητή μας. Για να γίνει αυτό χρησιμοποιούμε ορισμένα μέτρα αξιολόγησης. Στο κεφάλαιο θα παρουσιάσουμε μερικά που χρησιμεύουν σε ταξινομητές ηλεκτρονικού ταχυδρομείου. Στη συνέχεια θα αναφερθούμε σε κάθε φάση λεπτομερώς.

2.2 Διανυσματικό μοντέλο

Έστω $d_j \in D$ όπου D ένα σύνολο μηνυμάτων ηλεκτρονικού ταχυδρομείου. Η μορφή των μηνυμάτων αυτών είναι τέτοια που να μπορούν να είναι κατανοητά από τον άνθρωπο όταν τα διαβάζει. Ωστόσο αυτή δεν είναι κατάλληλη για έναν ταξινομητή. Οι περισσότεροι ταξινομητές μπορούν να διαχειριστούν μόνο αριθμητικά δεδομένα [13]. Γι' αυτό το λόγο είναι απαραίτητη μία διαδικασία αντιστοίχησης (indexing) του μηνύματος d_j σε μία συμπαγή μορφή κατάλληλη για τον ταξινομητή. Μία τέτοια μορφή, την οποία χρησιμοποιούμε στα πλαίσια αυτής της εργασίας, είναι η αναπαράσταση με χρήση διανύσματος (vector space model) (ενδεικτικά χρήση του στα [9-15]). Με βάση το μοντέλο αυτό κάθε μήνυμα d_j αναπαρίσταται ως εξής:

$$d_j = \langle w_{1j}, \ldots, w_{|T|j} \rangle$$

Κάθε συνιστώσα του παραπάνω διανύσματος αντιστοιχεί σε έναν όρο (term). Συνήθως ο όρος είναι μία λέξη. Περισσότερες λεπτομέρειες για τον καθορισμό των όρων θα αναφέρουμε στην επόμενη ενότητα. Το w_{kj} είναι το βάρος (weight) κάθε όρου το οποίο γενικά δείχνει πόσο σημαντικός είναι ο όρος για το μήνυμα d_j. Το σύνολο των όρων που αντιστοιχούν στις συνιστώσες του διανύσματος αποτελούν το λεξιλόγιο (vocabulary) και κάθε όρος εμφανίζεται τουλάχιστον μία φορά σε ένα μήνυμα του συνόλου μηνυμάτων D. Συνήθως τα διανύσματα που προκύπτουν είναι αραιά δηλαδή πολλές συνιστώσες έχουν την τιμή μηδέν.

Ένα σημαντικό θέμα που προκύπτει οποιαδήποτε και αν είναι η διαδικασία αντιστοίχησης είναι η απώλεια πληροφορίας. Οπότε πρέπει να είμαστε πολύ προσεκτικοί στο τι θα ορίσουμε ως όρο και με ποιο τρόπο θα υπολογίσουμε τα βάρη αυτών.

2.3 Καθορισμός των όρων

Στο σημείο αυτό θα ξεκαθαρίσουμε την έννοια του όρου, δηλαδή τι θεωρούμε ως όρους του διανύσματος αναπαράστασης.

2.3.1 Αντιστοίχηση όρων με λέξεις

Η πρώτη προσέγγιση που ακολουθείται στον καθορισμό των όρων είναι η αντιστοίχηση με λέξεις (bag of words model). Κάθε λέξη πρέπει να εμφανίζεται τουλάχιστον σε ένα μήνυμα του συνόλου εκπαίδευσης. Λέγοντας λέξη εννοούμε συνήθως ακολουθίες χαρακτήρων που στο μήνυμα διαχωρίζονται από λευκούς χαρακτήρες (π.χ. η λέξη "adult"). Ωστόσο μπορούμε να καθορίσουμε ότι μία λέξη θα αποτελείται μόνο από αριθμούς και γράμματα του αλφαβήτου και όλα τα άλλα να θεωρούνται διαχωριστικά μεταξύ των λέξεων. Είναι δυνατό να ορίσουμε ως μέρος μία λέξης και άλλα σύμβολα ανάλογα με το τι μας εξυπηρετεί κάθε φορά (π.χ. η λέξη "guar*an.tee").

Ένα άλλο θέμα είναι από ποια τμήματα του μηνύματος εξάγουμε τις λέξεις. Συνήθως λαμβάνουμε υπόψη ολόκληρο το μήνυμα αλλά είναι δυνατό να χρησιμοποιήσουμε μόνο μερικά τμήματα όπως το θέμα (subject) που φέρει πολύ πληροφορία για την κατηγορία του. Να τονίσουμε ότι λέξεις εξάγονται πιθανώς και από το HTML κομμάτι ενός μηνύματος καθώς και από τις επισυνάψεις (attachments). Την προσέγγιση αυτή ακολουθούμε στην παρούσα εργασία.

2.3.2 Εναλλακτικοί καθορισμοί των όρων

Μία εναλλακτική τεχνική είναι να αντιστοιχήσουμε τους όρους με φράσεις. Ο ορισμός της φράσης μπορεί να γίνει είτε με βάση τη γραμματική της γλώσσας (συντακτικός ορισμός της φράσης) είτε με βάση το ότι μία ακολουθία ή σύνολο λέξεων εμφανίζεται αρκετά συχνά στο σύνολο εκπαίδευσης (στατιστικός ορισμός της φράσης). Στην δεύτερη περίπτωση ένας όρος μπορεί να είναι π.χ. "be over 21". Μάλιστα η προσέγγιση n-gram (n-grams model) ανήκει στην κατηγορία αυτή και θεωρεί ως όρους ακολουθίες που αποτελούνται από 1, 2, … , n λέξεις [11,14].

Οι παραπάνω προσεγγίσεις συνήθως οδηγούν σε περισσότερους όρους (μεγαλύτερο λεξιλόγιο) από την προσέγγιση με λέξεις οι οποίοι ωστόσο έχουν μικρότερη συχνότητα εμφάνισης στο σύνολο εκπαίδευσης. Μάλιστα αυτές δεν έχουν πολύ καλύτερα αποτελέσματα στην ταξινόμηση απ' ότι η χρήση λέξεων. Πολλές φορές είναι και χειρότερες. Για το λόγο αυτό δεν τις μελετούμε περαιτέρω.

2.4 Καθορισμός των βαρών

Πλέον στρέφουμε την προσοχή μας στον τρόπο με τον οποίο θα καθορίσουμε το βάρος w_{kj} του όρου t_k στο μήνυμα d_j. Στη συνέχεια αναφέρουμε τρεις προσεγγίσεις οι οποίες είναι και οι πιο διαδεδομένες για ταξινόμηση ηλεκτρονικού ταχυδρομείου και όχι μόνο.

- ***Χρήση δυαδικών βαρών.*** Το βάρος κάθε όρου υποδηλώνει την ύπαρξή του ή όχι στο μήνυμα. Ουσιαστικά μας ενδιαφέρει κατά πόσο ένας όρος περιέχεται στο μήνυμα. Η ανάθεση των βαρών είναι:

  - $w_{kj} = 1$, αν ο όρος t_k υπάρχει στο μήνυμα d_j.
  - $w_{kj} = 0$, αν ο όρος t_k δεν υπάρχει στο μήνυμα d_j

- ***Χρήση της συχνότητας εμφάνισης.*** Σε αυτή την περίπτωση το βάρος κάθε όρου είναι ο αριθμός εμφανίσεών του στο μήνυμα. Η ανάθεση βαρών είναι:

$$w_{kj} = \#(t_k, d_j)$$

- ***Χρήση της συνάρτησης tfidf (term frequency inverse document frequency).*** Για τη συνάρτηση αυτή υπάρχουν αρκετοί ορισμοί. Στη συνέχεια παρουσιάζουμε δύο εξ αυτών:

$$w_{kj} = tfidf(t_k, d_j) = \#(t_k, d_j) \cdot \log \frac{|D|}{\#_D(t_k)}$$

$$w_{kj} = tfidf(t_k, d_j) = \log(\#(t_k, d_j) + 1) \cdot \log \frac{|D|}{\#_D(t_k)}$$

όπου $\#(t_k, d_j)$ είναι ο αριθμός εμφανίσεων του όρου t_k στο έγγραφο d_j, $|D|$ είναι το πλήθος των μηνυμάτων του συνόλου εκπαίδευσης D και $\#_D(t_k)$ είναι ο αριθμός μηνυμάτων στα οποία εμφανίζεται ο όρος t_k.

Η συνάρτηση αυτή μας δείχνει ότι ένας όρος όσες περισσότερες φορές εμφανίζεται σε ένα μήνυμα τόσο πιο αντιπροσωπευτικός του περιεχομένου του είναι (term frequency). Αντιθέτως σε όσα περισσότερα μηνύματα εμφανίζεται ο όρος τόσο

λιγότερο βοηθά στον διαχωρισμό τους δηλαδή παρέχει λιγότερη πληροφορία (inverse document frequency).

Είναι μάλιστα αρκετά συχνό τα βάρη που προκύπτουν από την *tfidf* να κανονικοποιούνται στο διάστημα [0,1]. Με αυτό τον τρόπο όλα τα διανύσματα έχουν ίσο μέτρο. Η κανονικοποίηση γίνεται ως εξής:

$$w_{kj} = \frac{tfidf(t_k , d_j)}{\sqrt{\sum_{s=1}^{T} (tfidf(t_s , d_j))^2}}$$

Η κανονικοποίηση αυτή ονομάζεται συνημιτονοειδής (cosine normalization).

2.5 Επιπλέον προεπεξεργασία

Στην ενότητα αυτή παρουσιάζουμε ορισμένα ακόμη θέματα στην προεπεξεργασία των μηνυμάτων. Μάλιστα η εφαρμογή όσων παρουσιάζονται παρακάτω είναι προαιρετική στη κατασκευή ενός ταξινομητή.

2.5.1 Αφαίρεση λέξεων

Στην αφαίρεση λέξεων συνήθως παραβλέπουμε τις λέξεις που είτε εμφανίζονται πολύ συχνά είτε πολύ σπάνια. Αυτές που εμφανίζονται συχνά αποκαλούνται συνηθισμένες λέξεις και συνήθως διαθέτουμε μια λίστα με τέτοιες (stop list). Οι λέξεις αυτές είναι άρθρα, σύνδεσμοι, προθέσεις κ.α. Το κίνητρο για την αφαίρεση αυτών είναι ότι δεν παρέχουν καμία πληροφορία σχετικά με την κατηγορία ενός μηνύματος καθώς τις συναντούμε τόσο σε spam όσο και μη-spam μηνύματα.

Όσον αφορά τις λέξεις που εμφανίζονται σπάνια αυτές μπορούν να αφαιρεθούν με βάση το πλήθος των διαφορετικών μηνυμάτων στα οποία υπάρχουν ή με βάση τον αριθμό εμφανίσεων σε όλα τα μηνύματα. Η ιδέα του να μην λαμβάνουμε υπόψη μας τέτοιες λέξεις στηρίζεται στο ότι δεν δίνουν πληροφορία για την κατηγορία του μηνύματος αφού η εμφάνισή τους δεν είναι «αρκετά» συχνή σε κάποια κατηγορία ώστε να τη χαρακτηρίζει.

Τα δύο παραπάνω βήματα συνήθως εφαρμόζονται κατά την προεπεξεργασία και ιδιαίτερα το πρώτο. Είναι εύκολα κατανοητό ότι έτσι το πλήθος των όρων του διανύσματος μειώνεται σημαντικά και συνεπώς η διάστασή του.

2.5.2 Εύρεση της μορφολογικής ρίζας

Σε ένα μήνυμα είναι συχνό να υπάρχουν πολλά παράγωγα μίας λέξης. Για παράδειγμα οι λέξεις "manage", "manager", "managing", "managed" είναι όλες παράγωγα της λέξης "manage" δηλαδή έχουν την ίδια μορφολογική ρίζα. Πλέον αντιμετωπίζουμε όλες τις παραπάνω λέξεις ως μία, την ρίζα τους.

Το κίνητρο για αυτό είναι ότι οι λέξεις αυτές παρέχουν την ίδια πληροφορία για τα αντικείμενα και υποκείμενα με τα οποία σχετίζονται και άρα μας αρκεί ένας όρος αντί τέσσερεις στο διάνυσμά μας. Μάλιστα είναι προφανές ότι η διάσταση των διανυσμάτων μειώνεται και η αναπαράσταση των μηνυμάτων γίνεται πιο συμπαγής καθώς αποφεύγεται το φαινόμενο πολλοί όροι να έχουν βάρος μηδέν.

Τα παραπάνω μας δίνουν την ελπίδα ότι με τον τρόπο αυτό μπορούμε να δημιουργήσουμε καλύτερους ταξινομητές. Ωστόσο στην πράξη κάτι τέτοιο δεν συμβαίνει καθώς είτε έχουμε πολύ μικρή βελτίωση είτε μείωση της ακρίβειας του ταξινομητή και άρα η εφαρμογή της εύρεσης της μορφολογικής ρίζας δεν γίνεται πάντα. Στην εργασία αυτή ούτε εμείς την χρησιμοποιούμε. Ένας γνωστός αλγόριθμος για μετασχηματισμό μορφολογικής ρίζας είναι του Porter [21].

2.5.3 Επιλογή των τμημάτων του μηνύματος

Ο κανόνας είναι ότι οι όροι εξάγονται από ολόκληρο το έγγραφο (υπό αυτή την έννοια λέμε ότι η φάση αυτή είναι προαιρετική). Ωστόσο στα μηνύματα ηλεκτρονικού ταχυδρομείου υπάρχουν πολλά τμήματα όπως κεφαλίδες, επισυνάψεις τα οποία είναι στην ευχέρειά μας αν θα τα χρησιμοποιήσουμε. Μάλιστα υπάρχουν τμήματα όπως είναι το θέμα τα οποία παρέχουν σημαντική πληροφορία για την κατηγορία και είναι δυνατό κάποιος να στηριχθεί μόνο σε αυτά για να δημιουργήσει ένα ταξινομητή. Μία άλλη δυνατή προσέγγιση είναι η χρήση μόνο του σώματος ή σε συνδυασμό με το θέμα. Πολλά μηνύματα περιέχουν HTML και άρα πρέπει να αποφασίσουμε αν θα εξάγουμε όρους από τις ετικέτες της. Οι επιλογές αυτές έχουν σημαντικό αντίκτυπο στο πλήθος των όρων του διανύσματος αλλά και

στην πολυπλοκότητα της προεπεξεργασίας καθώς τμήματα όπως οι επισυνάψεις απαιτούν αποκωδικοποίηση η οποία δεν είναι πάντα εύκολη.

2.6 Μείωση της διάστασης

Το πλήθος των όρων που προκύπτουν μετά τα βήματα που αναφέρθηκαν παραπάνω συνήθως είναι πάρα πολύ μεγάλο. Πολλοί απ' αυτούς δεν παρέχουν πληροφορία για την κατηγορία ενός μηνύματος και άρα επιθυμούμε να μην τους λάβουμε υπόψη στην κατασκευή του ταξινομητή.

2.6.1 Κίνητρα για μείωση της διάστασης

Όπως αναφέραμε είναι συνηθισμένο να προκύπτουν διανύσματα με μεγάλο πλήθος όρων. Αυτό δεν είναι επιθυμητό καθώς το πρόβλημα που είναι γνωστό ως κατάρα της διάστασης (curse of dimensionality) κάνει την εμφάνισή του. Το πρόβλημα αυτό έχει να κάνει με το ότι όσο μεγαλύτερη είναι η διάσταση των διανυσμάτων με τα οποία θα εκπαιδεύσουμε τον ταξινομητή τόσο περισσότερα μηνύματα απαιτούνται ώστε να επιτύχουμε καλή δυνατότητα ταξινόμησης, δηλαδή καλή εκπαίδευση του ταξινομητή. Ωστόσο η εύρεση μεγάλου αριθμού μηνυμάτων δεν είναι εύκολη και επιπλέον η χειρονακτική τους ταξινόμηση για να δημιουργήσουμε τον ταξινομητή γίνεται πιο επίπονη.

Ένα άλλο πρόβλημα που σχετίζεται με την μεγάλη διάσταση είναι η υπερεκπαίδευση (overfitting). Σύμφωνα με αυτό ένας ταξινομητής εμφανίζει μικρό σφάλμα στην ταξινόμηση των μηνυμάτων με τα οποία έχει εκπαιδευτεί και εξάγει τα χαρακτηριστικά τους αλλά μεγάλο σε άγνωστα μηνύματα. Αυτό συμβαίνει γιατί εκπαιδεύουμε τον ταξινομητή σε χαρακτηριστικά που σχετίζονται στενά με το σύνολο εκπαίδευσης και όχι με τις κατηγορίες γενικότερα.

Τέλος η μικρότερη διάσταση μειώνει το υπολογιστικό κόστος και τον χρόνο τόσο για την εκπαίδευση όσο και για την ταξινόμηση νέων μηνυμάτων.

2.6.2 Ορισμός και προσεγγίσεις

Ορίζουμε ως μείωση της διάστασης (dimensionality reduction) τη διαδικασία με την οποία το αρχικό πλήθος όρων $|T|$ μειώνεται σε $|T'|$ έτσι ώστε $|T'|<|T|$.

Οι προσεγγίσεις που ακολουθούνται είναι:

- μείωση με επιλογή όρων (term selection),
- μείωση με εξαγωγή όρων (term extraction).

Στην πρώτη προσέγγιση την οποία χρησιμοποιούμε στην παρούσα εργασία επιλέγουμε ένα υποσύνολο από τους αρχικούς όρους με βάση κάποια κριτήρια. Στη δεύτερη οι όροι που προκύπτουν δεν είναι του ίδιου τύπου με τους αρχικούς και προκύπτουν από συνδυασμούς ή μετασχηματισμούς των αρχικών. Μία συνηθισμένη προσέγγιση είναι οι νέοι όροι να προκύπτουν με γραμμικό συνδυασμό των υπαρχόντων. Στη συνέχεια θα περιγράψουμε ορισμένες μεθόδους που αφορούν τη μείωση με επιλογή. Να σημειώσουμε ότι η μείωση της διάστασης οδηγεί σε απώλεια πληροφορίας και άρα θα πρέπει να γίνεται πολύ προσεκτικά.

2.6.3 Μείωση με βάση τη συχνότητα εμφάνισης

Η μέθοδος αυτή ελέγχει τον αριθμό μηνυμάτων που εμφανίζεται ο κάθε όρος (document frequency) και επιλέγει εκείνους που εμφανίζονται στα περισσότερα μηνύματα. Αυτό με την πρώτη ματιά φαίνεται λίγο περίεργο καθώς λέξεις που εμφανίζονται συχνά είπαμε ότι δεν παρέχουν πολύ πληροφορία. Ωστόσο οι συνηθισμένες λέξεις αφαιρούνται πριν τη μείωση της διάστασης. Οι υπόλοιπες λέξεις που συναντάμε σε ένα σύνολο μηνυμάτων συνήθως έχουν μικρές συχνότητες εμφάνισης και άρα με την μέθοδο αυτή διατηρούμε όρους οι οποίοι έχουν μικρή συχνότητα εμφάνισης και διώχνουμε αυτούς που έχουν υπερβολικά μικρή και άρα είναι άχρηστοι.

Έχει δειχτεί [22] ότι με χρήση αυτής της μεθόδου είναι δυνατό να μειώσουμε έως και 10 φορές την διάσταση χωρίς να χάσουμε από την αποτελεσματικότητα του ταξινομητή.

2.6.4 Άλλες μέθοδοι

Πέρα από τη μέθοδο που στηρίζεται στη συχνότητα εμφάνισης των λέξεων υπάρχουν ορισμένες συναρτήσεις που χρησιμοποιούνται για τη μείωση της διάστασης οι οποίες έχουν μελετηθεί από πολλούς ερευνητές. Οι συναρτήσεις αυτές στηρίζονται στη διαίσθηση ότι οι όροι που περιέχουν την περισσότερη πληροφορία είναι αυτοί που η κατανομή τους μεταξύ των κατηγοριών είναι πολύ διαφορετική. Ουσιαστικά ψάχνουν για λέξεις που εμφανίζονται στην μία κατηγορία και όχι στην άλλη με αποτέλεσμα να

καταδεικνύουν ότι ένα μήνυμα ανήκει σε μία κατηγορία πολύ ισχυρά. Στον πίνακα που ακολουθεί παρουσιάζεται μία σύνοψη αυτών μαζί με τον μαθηματικό τους τύπο.

Function	Denoted by	Mathematical form
DIA association factor	$z(t_k, c_i)$	$P(c_i \mid t_k)$
Information gain	$IG(t_k, c_i)$	$\sum_{c \in \{c_i, \bar{c}_i\}} \sum_{t \in \{t_k, \bar{t}_k\}} P(t,c) \cdot \log \frac{P(t,c)}{P(t) \cdot P(c)}$
Mutual information	$MI(t_k, c_i)$	$\log \frac{P(t_k, c_i)}{P(t_k) \cdot P(c_i)}$
Chi-square	$\chi^2(t_k, c_i)$	$\frac{\lvert Tr \rvert \cdot [P(t_k, c_i) \cdot P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i) \cdot P(\bar{t}_k, c_i)]^2}{P(t_k) \cdot P(\bar{t}_k) \cdot P(c_i) \cdot P(\bar{c}_i)}$
NGL coefficient	$NGL(t_k, c_i)$	$\frac{\sqrt{\lvert Tr \rvert} \cdot [P(t_k, c_i) \cdot P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i) \cdot P(\bar{t}_k, c_i)]}{\sqrt{P(t_k) \cdot P(\bar{t}_k) \cdot P(c_i) \cdot P(\bar{c}_i)}}$
Relevancy score	$RS(t_k, c_i)$	$\log \frac{P(t_k \mid c_i) + d}{P(\bar{t}_k \mid \bar{c}_i) + d}$
Odds ratio	$OR(t_k, c_i)$	$\frac{P(t_k \mid c_i) \cdot (1 - P(t_k \mid \bar{c}_i))}{(1 - P(t_k \mid c_i)) \cdot P(t_k \mid \bar{c}_i)}$
GSS coefficient	$GSS(t_k, c_i)$	$P(t_k, c_i) \cdot P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i) \cdot P(\bar{t}_k, c_i)$

Πίνακας 2.1: Συναρτήσεις για μείωση της διάστασης.

Οι συναρτήσεις αυτές όπως φαίνεται περιέχουν πιθανότητες τις οποίες υπολογίζουμε με βάση των αριθμό εμφανίσεων των όρων στο σύνολο εκπαίδευσης. Στον παραπάνω πίνακα το $|Tr|$ αντιστοιχεί στο μέγεθος του συνόλου εκπαίδευσης ενώ το $P(t_k, \bar{c}_i)$ στην πιθανότητα το μήνυμα d_j να περιέχει τον όρο t_k και να μην ανήκει στην κατηγορία c_i.

Κάθε μία εκ των παραπάνω συναρτήσεων ορίζεται σχετικά με μόνο μία κατηγορία. Αυτό δεν μας ενοχλεί στο πρόβλημα που μελετούμε καθώς είναι δυαδικό ($c_i = \text{spam}, \bar{c}_i = \text{μη} - \text{spam}$). Σε προβλήματα που αποτελούνται από πολλές κατηγορίες για να βρούμε το πόσο χρήσιμος είναι ένας όρος υπολογίζουμε τη συνάρτηση για κάθε κατηγορία και στη συνέχεια μπορούμε να πάρουμε για παράδειγμα το άθροισμα για όλες τις κατηγορίες. Περισσότερες προσεγγίσεις γι' αυτό υπάρχουν στο [8].

Όσο μεγαλύτερη είναι η τιμή της συνάρτησης τόσο πιο σημαντικός είναι ο όρος. Άρα επιλέγουμε τους όρους με τη μεγαλύτερη τιμή. Σε μελέτες που έχουν γίνει η πλειονότητα των συναρτήσεων αυτών πετυχαίνουν καλύτερα αποτελέσματα από τη μείωση με βάση την συχνότητα [12].

Κέρδος πληροφορίας (Information Gain)

Η συνάρτηση αυτή έχει χρησιμοποιηθεί ευρέως στην μείωση της διάστασης όσον αφορά την ταξινόμηση του ηλεκτρονικού ταχυδρομείου [9-12] και γενικότερα στη ταξινόμηση κειμένων και έχει πετύχει πολύ καλά αποτελέσματα. Την παρουσιάζουμε αναλυτικότερα καθώς την έχουμε χρησιμοποιήσει στα πλαίσια της εργασίας αυτής για να μειώσουμε τη διάσταση των μηνυμάτων.

Οι πιθανότητες που υπάρχουν στη συνάρτηση αυτή υπολογίζονται με βάση τις συχνότητες στο σύνολο εκπαίδευσης. Συγκεκριμένα η πιθανότητα της κατηγορίας c_i υπολογίζεται συνήθως ως ο λόγος των μηνυμάτων που ανήκουν στην κατηγορία c_i προς το σύνολο των μηνυμάτων του συνόλου εκπαίδευσης. Η πιθανότητα $P(t_k, c_i)$ με βάση το πολυωνυμικό (multinomial) μοντέλο που παρουσιάζεται στο [14].

Ένας εναλλακτικός ορισμός [10] που μας βοηθά στην κατανόησή της είναι ο παρακάτω:

$$IG(t_k, c_i) = H(c) - \sum_{t \in \{t_k, \bar{t}_k\}} P(t) \cdot H(c \mid t)$$

$$H(c) = - \sum_{c \in \{c_i, \bar{c}_i\}} P(c) \cdot \log P(c)$$

Σύμφωνα με αυτόν το $H(c)$ είναι η εντροπία της κατηγορίας ενός μηνύματος και άρα μετρά την αβεβαιότητα της κατηγορίας. Το $H(c \mid t)$ μετρά την αβεβαιότητα για την κατηγορία δεδομένου ότι γνωρίζουμε ότι ο όρος υπάρχει ή δεν υπάρχει στο μήνυμα. Άρα η παραπάνω εξίσωση μας δείχνει πόσο μειώνεται η αβεβαιότητα για την κατηγορία αν γνωρίζουμε την παρουσία ή απουσία του όρου. Όσο μεγαλύτερη η τιμή της συνάρτησης τόσο σημαντικότερος είναι ο όρος.

2.7 Παράδειγμα προεπεξεργασίας

Στο σημείο αυτό παρουσιάζουμε ένα παράδειγμα ώστε η διαδικασία της προεπεξεργασίας να γίνει καλύτερα κατανοητή. Έστω το μήνυμα που φαίνεται στο σχήμα 2.1.

Subject: German corpora I am looking for on-line corpora of modern German. Any information would be appreciated.

Σχήμα 2.1: Μήνυμα πριν την προεπεξεργασία.

Πρώτο βήμα στη προεπεξεργασία είναι η εύρεση των λέξεων. Θεωρούμε πως μία λέξη αποτελείται μόνο από χαρακτήρες και οποιοδήποτε άλλο σύμβολο αποτελεί διαχωριστικό μεταξύ των λέξεων. Είναι συνηθισμένο να γίνεται μετατροπή των χαρακτήρων σε πεζούς και αυτό κάνουμε και εδώ. Εξάγουμε τις λέξεις από ολόκληρο το μήνυμα. Οι λέξεις είναι οι εξής:

subject	line
german	of
corpora	modern
i	any
am	information
looking	would
for	be
on	appreciated

Σχήμα 2.2: Λέξεις του μηνύματος.

Στη συνέχεια αφαιρούνται οι συνηθισμένες λέξεις του μηνύματος όπως είναι το "for". Αυτό έχει ως αποτέλεσμα το λεξιλόγιο να μειωθεί και οι λέξεις πλέον να είναι:

subject	modern
german	information
corpora	would
looking	be
line	appreciated

Σχήμα 2.3: Λέξεις μετά την αφαίρεση συνηθισμένων λέξεων.

Έστω ότι δεν εφαρμόζουμε την εύρεση της μορφολογικής ρίζας και ότι διατηρούμε τις λέξεις με το μεγαλύτερο κέρδος πληροφορίας στο σύνολο των μηνυμάτων που

διαθέτουμε. Έστω ότι στο συγκεκριμένο μήνυμα υπάρχουν τέσσερεις εξ αυτών. Οι λέξεις που απομένουν είναι:

german corpora	modern appreciated

Σχήμα 2.4: Λέξεις μετά τη μείωση της διάστασης.

Το τελευταίο βήμα έχει να κάνει με την ανάθεση των βαρών στους όρους του μηνύματος. Έστω ότι τα βάρη είναι η συχνότητα εμφάνισης του όρου. Τότε το διάνυσμα που περιγράφει το μήνυμα είναι:

Όρος	**Βάρος**
german	2
corpora	2
modern	1
appreciated	1

Σχήμα 2.5: Διάνυσμα του μηνύματος.

Το διάνυσμα αυτό πλέον θα δοθεί στον ταξινομητή ώστε να τον εκπαιδεύσουμε και να μάθει τα χαρακτηριστικά των κατηγοριών.

2.8 Κατασκευή ταξινομητή

Η τεχνική της μηχανικής μάθησης όπως αναφέραμε απαιτεί την ύπαρξη ενός συνόλου μηνυμάτων τα οποία έχουμε χειρονακτικά ταξινομήσει στις κατηγορίες τους. Το σύνολο αυτό συνήθως διαιρείται σε δύο μέρη, το σύνολο εκπαίδευσης (training set) και το σύνολο ελέγχου (test set). Για να χρησιμοποιήσουμε τα σύνολα αυτά πρέπει πρώτα να τα επεξεργαστούμε όπως αναφέρθηκε στις προηγούμενες ενότητες.

Το σύνολο εκπαίδευσης περιέχει τα μηνύματα από τα οποία ο ταξινομητής θα εξάγει τα χαρακτηριστικά και θα δημιουργήσει ένα μοντέλο με βάση το οποίο θα μπορεί να ταξινομεί νέα άγνωστα μηνύματα. Καθώς ο ταξινομητής γνωρίζει την κατηγορία των

μηνυμάτων του συνόλου αυτού επιχειρεί να δημιουργήσει ένα μοντέλο που θα τα ταξινομεί ορθά. Έτσι έχουμε πλέον κατασκευάσει ένα ταξινομητή αυτόματα μέσω ενός συνόλου εκπαίδευσης και ενός αλγορίθμου που εξάγει τα χαρακτηριστικά και δημιουργεί το μοντέλο.

Το σύνολο ελέγχου χρειάζεται για να αξιολογήσουμε την αποδοτικότητα του ταξινομητή που κατασκευάσαμε. Τα μηνύματα του συνόλου αυτού δίνονται ως είσοδος στον ταξινομητή και αυτός αποφασίζει για την κατηγορία τους. Στη συνέχεια ελέγχουμε αν η απόφαση αυτή είναι σωστή ή λανθασμένη. Αν είναι συχνά ορθή τότε ο ταξινομητής μας μπορεί και κάνει καλό διαχωρισμό και συνεπώς μπορούμε να εμπιστευτούμε τις αποφάσεις του.

Πολλές φορές οι αλγόριθμοι που χρησιμοποιούνται για να εκπαιδεύσουμε τους ταξινομητές έχουν κάποιες εσωτερικές παραμέτρους τις οποίες θέλουμε να βελτιστοποιήσουμε ώστε να λάβουμε καλύτερα αποτελέσματα. Για να το πετύχουμε αυτό χρειαζόμαστε και ένα σύνολο επικύρωσης (validation set) το οποίο επίσης αποτελείται από μηνύματα που γνωρίζουμε την κατηγορία τους. Σε αυτό το σύνολο ελέγχουμε την επίδραση που έχουν διαφορετικές τιμές των παραμέτρων κατά την εκπαίδευση. Επιλέγουμε τις παραμέτρους που δίνουν τα καλύτερα αποτελέσματα στο σύνολο επικύρωσης.

Να τονίσουμε πως τα τρία αυτά σύνολα πρέπει να είναι ανεξάρτητα μεταξύ τους γιατί διαφορετικά δεν μπορούμε να βγάλουμε σωστά συμπεράσματα για το πόσο «καλός» είναι ο ταξινομητής μας. Για παράδειγμα αν υπάρχουν στο σύνολο ελέγχου μηνύματα που συναντούμε και στο σύνολο εκπαίδευσης θα λάβουμε πολύ ενθαρρυντικά αποτελέσματα αφού αυτά δεν είναι άγνωστα στον ταξινομητή.

2.8.1 K-fold cross validation

Το k-fold cross validation αποτελεί μία εναλλακτική προσέγγιση στο θέμα της εκπαίδευσης του ταξινομητή. Έχει χρησιμοποιηθεί ευρέως ως τρόπος για να πετύχουμε πιο ακριβή πειραματικά αποτελέσματα σε σχέση με την προηγούμενη προσέγγιση κατά την αξιολόγηση των ταξινομητών [9-14].

Στην περίπτωση αυτή το αρχικό σύνολο μηνυμάτων δεν χωρίζεται σε σύνολο εκπαίδευσης και ελέγχου. Αντίθετα χωρίζεται σε k υποσύνολα τα οποία δεν έχουν κοινά σημεία. Στη συνέχεια k-1 υποσύνολα χρησιμεύουν ως σύνολο εκπαίδευσης και ένα υποσύνολο ως σύνολο ελέγχου. Η εκπαίδευση στο σημείο αυτό είναι ακριβώς ίδια με πριν. Ωστόσο η διαδικασία επαναλαμβάνεται k φορές με αποτέλεσμα όλα τα υποσύνολα να

βρεθούν τόσο στο σύνολο εκπαίδευσης όσο και στο σύνολο ελέγχου. Για να δούμε πόσο καλός είναι ο ταξινομητής παίρνουμε τον μέσο όρο από τις k επαναλήψεις.

Συνήθως προσπαθούμε σε κάθε ένα από τα επιμέρους υποσύνολα να διατηρούμε την ίδια αναλογία από spam και μη-spam μηνύματα που υπάρχει στο αρχικό σύνολο μηνυμάτων ώστε τα αποτελέσματά μας να είναι όσο το δυνατό πιο ακριβή.

Τόσο σε αυτή όσο και στην προηγούμενη προσέγγιση αν πέρα από την αξιολόγηση του ταξινομητή επιθυμούμε να τον χρησιμοποιήσουμε και στην πράξη συνήθως μετά την αξιολόγηση τον εκπαιδεύουμε με χρήση όλων των μηνυμάτων που έχουμε στην διάθεσή μας. Κυρίως το k-fold cross validation έχει χρησιμοποιηθεί στα πλαίσια αυτής της εργασίας.

2.9 Μέτρα αξιολόγησης ταξινομητών ηλεκτρονικού ταχυδρομείου

Όπως αναφέραμε όταν εκπαιδεύουμε έναν ταξινομητή στη συνέχεια πρέπει να αξιολογήσουμε την απόδοσή του μέσω του συνόλου ελέγχου. Στην ενότητα αυτή παρουσιάζουμε ορισμένα μέτρα που μας βοηθούν στην διαδικασία αυτή.

Το πρόβλημα της ταξινόμησης κειμένων ηλεκτρονικού ταχυδρομείου αν και είναι μία μορφή του προβλήματος της ταξινόμησης κειμένων διαφέρει σημαντικά σε ένα σημείο. Τα λάθη που μπορούν να συμβούν στην ταξινόμηση δεν είναι ίδιας βαρύτητας. Συγκεκριμένα το να θεωρήσουμε ένα spam μήνυμα ως μη-spam (false negative) είναι λιγότερο σοβαρό από το αντίστροφο σφάλμα (false positive). Ένας χρήστης δεν θα ενοχληθεί σημαντικά αν κάποια στιγμή καταλήξει ένα spam μήνυμα στον κατάλογο με τα εισερχόμενα. Αντίθετα αν ένα μη-spam μήνυμα καταλήξει στο φάκελο "spam" ή ακόμη χειρότερα διαγραφεί αυτόματα από το φίλτρο που χρησιμοποιεί πρόκειται για κάτι απαράδεκτο ακόμη και όταν συμβαίνει πολύ σπάνια.

Τέτοια φαινόμενα στην ταξινόμηση ηλεκτρονικού ταχυδρομείου δεν πρέπει να συμβαίνουν γιατί ουσιαστικά κάνουν τον χρήστη να σκέφτεται «καλύτερα να απενεργοποιήσω το φίλτρο ώστε να μη χάσω τα σημαντικά μου μηνύματα» καθιστώντας το έτσι άχρηστο στην πράξη. Γι' αυτό το λόγο πρέπει να είμαστε πολύ προσεκτικοί στο τρόπο με τον οποίο αξιολογούμε τους ταξινομητές μας ώστε αυτοί να μπορούν να βρουν εφαρμογή στην καθημερινή ζωή.

Δύο πολύ γνωστά μέτρα τα οποία χρησιμοποιούνται είναι τα spam recall (*R*) και spam precision (*P*). Οι ορισμοί τους είναι:

$$R = \frac{N_{S \to S}}{N_{S \to S} + N_{S \to NS}}$$

$$P = \frac{N_{S \to S}}{N_{S \to S} + N_{NS \to S}}$$

Στους παραπάνω ορισμούς τα $N_{X \to Y}$ συμβολίζουν το πλήθος των μηνυμάτων τα οποία ανήκουν στην κατηγορία X και ταξινομούνται στην Y. Οι δύο κατηγορίες συμβολίζονται ως S και NS (spam, non-spam). Το spam recall μετρά το ποσοστό των spam μηνυμάτων που ο ταξινομητής καταφέρνει να αναγνωρίσει. Ουσιαστικά είναι η πιθανότητα αν ένα μήνυμα είναι spam ο ταξινομητής να λάβει την απόφαση αυτή. Διαισθητικά μπορούμε να πούμε ότι μετρά την αποτελεσματικότητα. Το spam precision μετρά το ποσοστό των μηνυμάτων που ταξινομούνται ως spam και όντως είναι τέτοια. Ουσιαστικά είναι η πιθανότητα αν ένα μήνυμα ταξινομηθεί ως spam η απόφαση αυτή να είναι η σωστή. Διαισθητικά θα λέγαμε ότι μετρά το πόσο ασφαλής είναι ο ταξινομητής.

Καθώς τα δύο παραπάνω μέτρα είναι αντικρουόμενα με την έννοια ότι όταν το spam recall μεγαλώνει το spam precision μειώνεται και το αντίστροφο υπάρχει ένα μέτρο που τα συνδυάζει το F_1 που ορίζεται ως:

$$F_1 = \frac{2RP}{R + P}$$

Ένα άλλο μέτρο το οποίο χρησιμοποιούμε και στην εργασία μας είναι το accuracy (*Acc*) το οποίο ορίζεται ως:

$$Acc = \frac{N_{NS \to NS} + N_{S \to S}}{N_{NS} + N_S}$$

Αυτό μετρά το ποσοστό των μηνυμάτων και από τις δύο κατηγορίες που έχουν ταξινομηθεί σωστά και δίνει μία καλή εικόνα για την απόδοση του ταξινομητή. Τα N_{NS} και N_S είναι το σύνολο των μη-spam και spam μηνυμάτων που θέλουμε να ταξινομήσουμε.

Όπως είπαμε υπάρχει διαφορά μεταξύ των λαθών που συμβαίνουν στην ταξινόμηση του ηλεκτρονικού ταχυδρομείου. Για το λόγο αυτό οι Androutsopoulos et al. [9] εισήγαγαν μία σταθμισμένη εκδοχή του μέτρου accuracy (*WAcc*) η οποία είναι:

$$WAcc = \frac{\lambda \cdot N_{NS \to NS} + N_{S \to S}}{\lambda \cdot N_{NS} + N_S}$$

Σύμφωνα με τον ορισμό μοιάζει να θεωρούμε κάθε μη-spam μήνυμα ως λ διαφορετικά μηνύματα. Αυτό σημαίνει ότι η ορθή ταξινόμηση ενός τέτοιου μηνύματος αντιστοιχεί σε λ επιτυχίες και αντίστοιχα η λανθασμένη ταξινόμηση σε λ αποτυχίες. Με αυτό τον τρόπο έχουμε κατορθώσει να δώσουμε έμφαση στην ορθή ταξινόμηση των μη-spam μηνυμάτων.

Η τιμή του λ συνήθως ορίζεται με βάση το πώς μεταχειριζόμαστε μηνύματα που ταξινομούμε ως spam. Αν τα μηνύματα αυτά διαγράφονται αυτόματα τότε απαιτείται μεγάλο λ (π.χ. λ=999) ώστε να μην συμβεί ποτέ το ανεπιθύμητο σενάριο ο χρήστης να χάσει ένα μη-spam μήνυμα. Αντίθετα αν απλώς προστίθεται στο μήνυμα μία ετικέτα "spam" τότε το λ θα είναι μικρό (π.χ. λ=1).

Κεφάλαιο 3: Η μέθοδος SVM

3.1 Γενικά

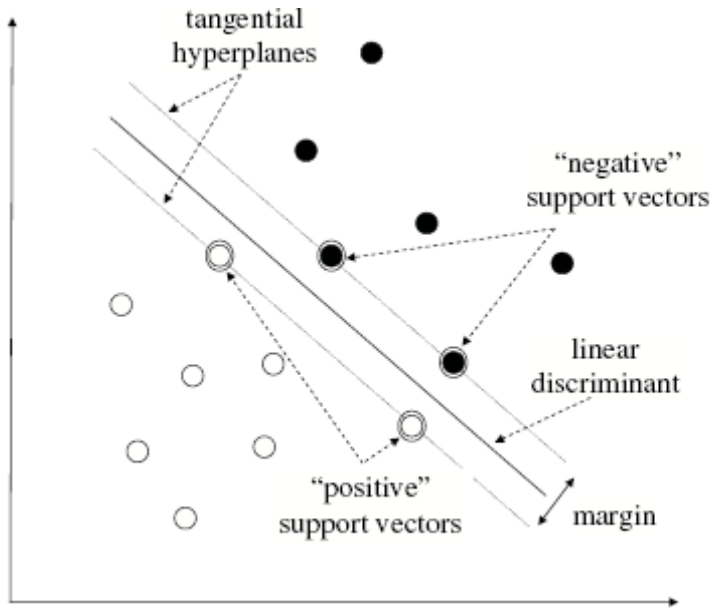
Η μέθοδος SVM (Support Vector Machine) αναπτύχθηκε από τον V. Vapnik και είναι μία από τις ευρέως χρησιμοποιούμενες τεχνικές μηχανικής μάθησης. Έχει βρει εφαρμογή στην αναγνώριση χειρόγραφων ψηφίων (handwritten digit recognition), στην ταξινόμηση κειμένων (text classification) κ.α. Η μέθοδος SVM στηρίζεται σε ένα ισχυρό θεωρητικό υπόβαθρο, τη θεωρία στατιστικής μάθησης (statistical learning theory).

Πρόκειται για μία μέθοδο η οποία διαχωρίζει τα δεδομένα δύο κατηγοριών δημιουργώντας ένα υπερεπίπεδο. Αν και η επιφάνεια αυτή είναι γραμμική πετυχαίνει να διαχωρίζει και μη γραμμικά διαχωρίσιμα δεδομένα καθώς τα αναπαριστά σε ένα χώρο μεγαλύτερης διάστασης από τον αρχικό και σε αυτόν φέρει το υπερεπίπεδο. Να τονίσουμε ότι το υπερεπίπεδο που δημιουργεί είναι αυτό που διαχωρίζει τις κατηγορίες με το βέλτιστο τρόπο και όχι ένα οποιοδήποτε. Τέλος οι ταξινομητές SVM για να φέρουν το υπερεπίπεδο χρησιμοποιούν ένα μέρος του συνόλου εκπαίδευσης τα λεγόμενα support vectors απ' όπου προήλθε και το όνομα της μεθόδου.

3.2 Υπερεπίπεδο μέγιστου εύρους

Όπως είπαμε η μέθοδος SVM προσπαθεί να διαχωρίσει τα δεδομένα των δύο κατηγοριών βρίσκοντας ένα υπερεπίπεδο απόφασης (decision hyperplane). Έστω το σχήμα

3.1 το οποίο αναπαριστά τα δεδομένα δύο κατηγοριών για την δυσδιάστατη περίπτωση στην οποία το υπερεπίπεδο είναι ευθεία.



Σχήμα 3.1: Υπερεπίπεδο απόφασης σε γραμμικά διαχωρίσιμα δεδομένα.

Όπως είναι φανερό υπάρχει ένα μεγάλο πλήθος υπερεπιπέδων τα οποία μπορούν να διαχωρίσουν τα δεδομένα των δύο κατηγοριών και μάλιστα χωρίς να κάνουν κανένα λάθος, καθώς τα δεδομένα είναι γραμμικά διαχωρίσιμα. Αν και αυτά τα υπερεπίπεδα κατορθώνουν στο σύνολο εκπαίδευσης να έχουν μηδενικό σφάλμα δεν υπάρχει καμία εγγύηση ότι θα έχουν την ίδια ικανότητα ταξινόμησης νέων άγνωστων δεδομένων.

Για τον παραπάνω λόγο η μέθοδος SVM προσπαθεί να εντοπίσει το «καλύτερο» απ' όλα τα υπερεπίπεδα. Η έννοια του «καλύτερου» αντιπροσωπεύεται από αυτό που ονομάζουμε υπερεπίπεδο μέγιστου εύρους (maximum margin hyperplane). Ουσιαστικά προσπαθούμε να βρούμε το υπερεπίπεδο που διαχωρίζει τα δεδομένα των δύο κατηγοριών του συνόλου εκπαίδευσης με τη μεγαλύτερη δυνατή απόσταση. Για να γίνει αυτό επιλέγουμε δύο παράλληλα υπερεπίπεδα που το καθένα περιλαμβάνει τουλάχιστον ένα στιγμιότυπο καθένα για διαφορετική κατηγορία (tangential hyperplanes). Τα υπερεπίπεδα είναι τέτοια ώστε «ανάμεσά» τους να μην υπάρχουν δεδομένα του συνόλου εκπαίδευσης και άρα να το διαχωρίζουν τέλεια. Τα στιγμιότυπα που βρίσκονται πάνω σε αυτά ονομάζονται support vectors. Η απόσταση των παράλληλων υπερεπιπέδων είναι το εύρος (margin). Επιλέγουμε

αυτά με το μεγαλύτερο εύρος. Το υπερεπίπεδο απόφασης είναι αυτό που βρίσκεται στο μέσο των παράλληλων υπερεπιπέδων με το μεγαλύτερο εύρος..

Το κίνητρο για την εύρεση του συγκεκριμένου υπερεπιπέδου είναι ότι δίνει την δυνατότητα ο ταξινομητής που δημιουργούμε να μπορεί να δίνει ορθά αποτελέσματα και για άγνωστα δεδομένα. Συγκεκριμένα ταξινομητές με μεγάλο εύρος εμφανίζουν λιγότερο το πρόβλημα της υπερεκπαίδευσης (overfitting).

3.2.1 Τυπικός ορισμός υπερεπιπέδου

Στο σημείο αυτό δίνουμε την εξίσωση του υπερεπιπέδου και των παράλληλων σε αυτό καθώς και το μέγεθος του εύρους. Η παρουσίαση θα γίνει με όρους που αφορούν το πρόβλημα που μελετάμε στην εργασία αυτή.

Έστω $D = \{d_1, d_2, \ldots, d_{|D|}\}$ το σύνολο εκπαίδευσης για τον ταξινομητή μας και $C = \{c_1 = \text{spam}, c_2 = \text{μη−spam}\}$ το σύνολο των κατηγοριών. Σε κάθε μήνυμα d_i του συνόλου εκπαίδευσης αντιστοιχεί και μία κατηγορία $c_i \in \{1, -1\}$ όπου το c_i=1 σημαίνει ότι το μήνυμα είναι spam και c_i=-1 σημαίνει ότι το μήνυμα είναι μη-spam. Καθώς πρόκειται για γραμμική επιφάνεια η εξίσωσή της είναι:

$$\mathbf{w} \cdot \mathbf{d} + b = 0$$

όπου **w** και b είναι παράμετροι του μοντέλου. Μάλιστα το διάνυσμα **w** είναι κάθετο στο υπερεπίπεδο που χωρίζει τις κατηγορίες. Τα μηνύματα που το διάνυσμά τους βρίσκεται πάνω στο υπερεπίπεδο θα επαληθεύουν την εξίσωση αυτή. Για τα υπόλοιπα θα ισχύει:

$$\mathbf{w} \cdot \mathbf{d} + b = k$$

Όπου $k > 0$ για όσα θα βρίσκονται στη πλευρά προς την οποία το **w** έχει φορά (τα μηνύματα spam εδώ) και $k < 0$ για όσα θα είναι στην άλλη. Μάλιστα έχουμε τη δυνατότητα να ορίσουμε τις παραμέτρους του μοντέλου έτσι ώστε τα παράλληλα υπερεπίπεδα να εκφράζονται ως (canonical hyperplanes):

$$\mathbf{w} \cdot \mathbf{d} + b = 1$$
$$\mathbf{w} \cdot \mathbf{d} + b = -1$$

Αυτό μπορεί να γίνει καθώς το υπερεπίπεδο με παραμέτρους (**w**,*b*) είναι το ίδιο με αυτό που έχει παραμέτρους (m**w**,m*b*). Από τις δύο παραπάνω εξισώσεις είναι εύκολο θεωρώντας ένα σημείο σε κάθε παράλληλο υπερεπίπεδο d_1, d_2 να βρούμε το εύρος του ταξινομητή.

$$\mathbf{w} \cdot (\mathbf{d}_1 - \mathbf{d}_2) = 2 \Rightarrow \|\mathbf{w}\| \times \text{margin} = 2 \Rightarrow \text{margin} = \frac{2}{\|\mathbf{w}\|}$$

3.3 Εκπαίδευση ταξινομητή SVM

3.3.1 Γραμμικό SVM

Ξεκινάμε την παρουσίασή μας σχετικά με την εκπαίδευση ενός SVM ταξινομητή θεωρώντας την περίπτωση που το σύνολο εκπαίδευσης είναι γραμμικά διαχωρίσιμο. Η εκπαίδευση συνίσταται στην εύρεση των παραμέτρων (**w**,*b*) του υπερεπιπέδου απόφασης. Συγκεκριμένα, η εκπαίδευση αφορά την λύση του εξής προβλήματος βελτιστοποίησης με περιορισμούς (constraint optimization problem):

$$\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2}$$

$$\text{subject to } c_i(\mathbf{w} \cdot \mathbf{d}_i + b) \geq 1, \quad i = 1, 2, ..., |D|$$

Από τις παραπάνω εξισώσεις προκύπτει ότι προσπαθούμε να βρούμε το υπερεπίπεδο με το μεγαλύτερο εύρος ενώ ταυτόχρονα θέλουμε τα μηνύματα της κατηγορίας c_i=1 να βρίσκονται πάνω από το υπερεπίπεδο $\mathbf{w} \cdot \mathbf{d} + b = 1$ ενώ της c_i=-1 κάτω από το $\mathbf{w} \cdot \mathbf{d} + b = -1$ και άρα να διαχωρίζονται τέλεια. Το πρόβλημα επιλύεται με χρήση πολλαπλασιαστών Lagrange. Για το σκοπό αυτό ξαναγράφουμε τη συνάρτηση που θέλουμε να ελαχιστοποιήσουμε σε μορφή που να λαμβάνει υπόψη τους περιορισμούς (η μορφή είναι γνωστή ως Lagrangian):

$$L_p = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{|D|} \lambda_i (c_i(\mathbf{w} \cdot \mathbf{d}_i + b) - 1)$$

Τα λ_i είναι οι πολλαπλασιαστές Lagrange και είναι φανερό ότι τα λάθη στην ταξινόμηση του συνόλου εκπαίδευσης έχουν ως αποτέλεσμα την αύξηση της τιμής της συνάρτησης καθώς $\lambda_i \geq 0$. Για να την ελαχιστοποιήσουμε παίρνουμε τις παραγώγους της ως προς τις παραμέτρους:

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{|D|} \lambda_i c_i \mathbf{d}_i$$

$$\frac{\partial L_p}{\partial b} = 0 \Rightarrow \sum_{i=1}^{|D|} \lambda_i c_i = 0$$

Καθώς δεν γνωρίζουμε τις τιμές για τους πολλαπλασιαστές Lagrange δεν μπορούμε να βρούμε τις τιμές για τις παραμέτρους ($\mathbf{w}$,b). Μάλιστα καθώς οι περιορισμοί εκφράζονται με ανισότητες και όχι ισότητες δεν μπορούν να χρησιμεύσουν στην εύρεση της λύσης. Συνεπώς τους μετατρέπουμε σε ισότητες με χρήση των συνθηκών Karush-Kuhn-Tucker (KKT conditions) που η λύση για αυτές είναι ισοδύναμη με την αρχική:

$$\lambda_i \geq 0$$
$$\lambda_i(c_i(\mathbf{w} \cdot \mathbf{d}_i + b) - 1) = 0$$

Αξίζει να σημειώσουμε ότι σύμφωνα με τους περιορισμούς αυτούς οι πολλαπλασιαστές Lagrange είναι διάφοροι του μηδενός μόνο για τα μηνύματα του συνόλου εκπαίδευσης που βρίσκονται ακριβώς πάνω στα παράλληλα υπερεπίπεδα δηλαδή για τα support vectors. Μάλιστα οι παράμετροι ($\mathbf{w}$,b) όπως φαίνεται από τους τελευταίους περιορισμούς και από την παράγωγο ως προς $\mathbf{w}$ καθορίζονται αποκλειστικά από τα support vectors. Τελικά για να επιλύσουμε το πρόβλημα το μετατρέπουμε στο ισοδύναμο δυικό (dual problem), καθώς δεν είναι εύκολη η χρήση των KKT συνθηκών, όπου έχουμε αντί της Lagrangian μορφής μία που περιέχει μόνο τους πολλαπλασιαστές Lagrange:

$$L_D = \sum_{i=1}^{|D|} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j c_i c_j \mathbf{d}_i \cdot \mathbf{d}_j$$

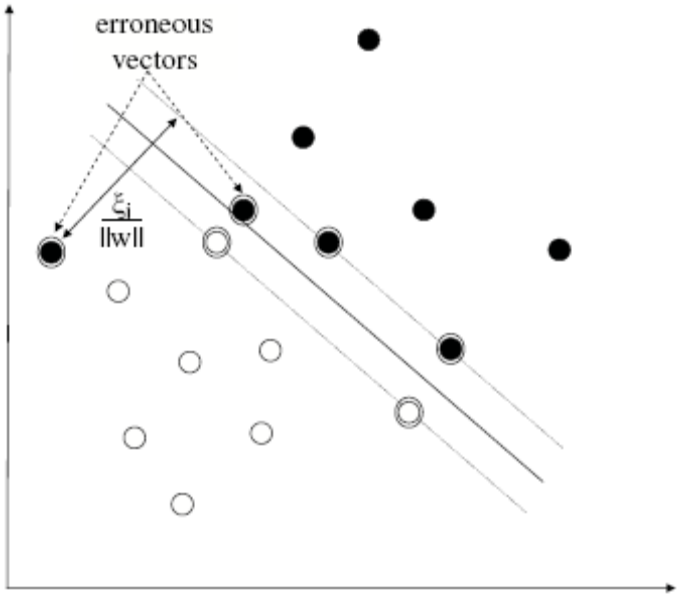
$$\text{subject to } \lambda_i \geq 0 \text{ and } \sum_{i=1}^{|D|} \lambda_i c_i = 0$$

Με την επίλυση του δυικού προβλήματος (το οποίο σε αντίθεση με το αρχικό είναι πρόβλημα μεγιστοποίησης) παίρνουμε τα λ_i (μόνο αυτά που αντιστοιχούν στα support vectors δεν είναι μηδέν) και με τη βοήθεια των ΚΚΤ συνθηκών και της παραγώγου ως προς **w** παίρνουμε τα (**w**,*b*). Η εύρεση της λύσης γίνεται συνήθως με τεχνικές τετραγωνικού προγραμματισμού (quadratic programming) και η επιφάνεια απόφασης εκφράζεται ως:

$$\left(\sum_{i=1}^{|D|} \lambda_i c_i \mathbf{d}_i \cdot \mathbf{d} \right) + b = 0$$

3.3.2 Γραμμικό SVM - Μη διαχωρίσιμη περίπτωση

Στην παραπάνω ενότητα είδαμε πως δημιουργούμε ένα SVM ταξινομητή που δεν κάνει λάθη στο σύνολο εκπαίδευσης. Πολλές φορές ωστόσο είναι αδύνατο να δημιουργήσουμε ένα υπερεπίπεδο που να το πετυχαίνει αυτό. Για το λόγο αυτό υπάρχει μία λίγο διαφορετική μορφή του SVM που λέγεται χαλαρό SVM (soft margin) και επιτρέπει να βρίσκουμε ένα υπερεπίπεδο και σε τέτοιες περιπτώσεις. Αυτό είναι επιθυμητό και για έναν άλλο λόγο, μπορούμε να επιτρέψουμε μερικά σφάλματα με στόχο να πετύχουμε μεγαλύτερο εύρος μεταξύ των κατηγοριών και άρα καλύτερη ικανότητα γενίκευσης. Παράδειγμα μίας τέτοιας περίπτωσης φαίνεται στο σχήμα 3.2.



Σχήμα 3.2: Παράδειγμα μη διαχωρίσιμου SVM.

Πλέον η εκπαίδευση έγκειται στη επίλυση του εξής προβλήματος βελτιστοποίησης με περιορισμούς:

$$\min_{\mathbf{w},\xi}\left(\frac{\|\mathbf{w}\|^2}{2}+C\sum_{i=1}^{|D|}\xi_i\right)$$

$$\text{subject to } \xi_i \geq 0 \text{ and } c_i(\mathbf{w}\cdot\mathbf{d}_i+b)\geq 1-\xi_i, \quad i=1,2,...,|D|$$

Στην περίπτωση αυτή υπάρχουν επιπλέον τα ξ_i καθώς και το C. Οι μεταβλητές ξ είναι απαραίτητες ώστε να επιτρέψουμε πλέον να συμβαίνουν σφάλματα στο σύνολο εκπαίδευσης. Τα ξ_i μετρούν το σφάλμα για κάθε μήνυμα και συγκεκριμένα η απόσταση μεταξύ του στιγμιότυπου και του παράλληλου υπερεπιπέδου που αντιστοιχεί στην κατηγορία αυτού είναι $\xi_i/\|\mathbf{w}\|$ όπως φαίνεται και στο σχήμα 3.2. Για τα μηνύματα που βρίσκονται στην σωστή πλευρά των παράλληλων υπερεπιπέδων που τους αντιστοιχούν οι μεταβλητές αυτές έχουν τιμή μηδέν. Μάλιστα για να περιορίσουμε τον αριθμό σφαλμάτων που γίνονται αυτά λαμβάνονται υπόψη και στην συνάρτηση που βελτιστοποιούμε. Η μεταβλητή C καθορίζεται από τον χρήστη και όσο μεγαλύτερη είναι τόσο πιο αυστηροί είμαστε ως προς τα λάθη. Η Lagrangian μορφή, όπου τα μ_i είναι οι πολλαπλασιαστές Lagrange που έχουν να κάνουν με το ότι τα $\xi_i \geq 0$, είναι:

$$L_p=\frac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{|D|}\xi_i-\sum_{i=1}^{|D|}\lambda_i(c_i(\mathbf{w}\cdot\mathbf{d}_i+b)-1+\xi_i)-\sum_{i=1}^{|D|}\mu_i\xi_i$$

Παίρνοντας τις παραγώγους ως προς $\mathbf{w}$, b, ξ_i έχουμε:

$$\frac{\partial L_p}{\partial \mathbf{w}}=0\Rightarrow \mathbf{w}=\sum_{i=1}^{|D|}\lambda_i c_i \mathbf{d}_i$$

$$\frac{\partial L_p}{\partial b}=0\Rightarrow \sum_{i=1}^{|D|}\lambda_i c_i=0$$

$$\frac{\partial L_p}{\partial \xi_i}=0\Rightarrow \lambda_i+\mu_i=C$$

Για τους ίδιους λόγους με πριν επιθυμούμε να μετατρέψουμε τις ανισότητες των περιορισμών σε ισότητες. Αυτό γίνεται με χρήση των ΚΚΤ συνθηκών, η λύση για τις οποίες είναι επίσης όπως αναφέραμε λύση και για το αρχικό μας πρόβλημα, που είναι:

$$\lambda_i \geq 0, \mu_i \geq 0, \xi_i \geq 0$$
$$\lambda_i(c_i(\mathbf{w} \cdot \mathbf{d}_i + b) - 1 + \xi_i) = 0$$
$$\mu_i \xi_i = 0$$

Αξίζει να σημειώσουμε ότι στη μη-διαχωρίσιμη περίπτωση τα λ_i είναι μη μηδενικά για τα στιγμιότυπα που είτε βρίσκονται πάνω στο παράλληλο υπερεπίπεδο που αντιστοιχεί στην κατηγορία τους είτε έχουν $\xi_i > 0$ ενώ τα μ_i είναι μηδέν μόνο γι' αυτά με $\xi_i > 0$. Τελικά η επίλυση του προβλήματος γίνεται μέσω της μεγιστοποίησης του δυικού το οποίο είναι:

$$L_D = \sum_{i=1}^{|D|} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j c_i c_j \mathbf{d}_i \cdot \mathbf{d}_j$$

$$\text{subject to } 0 \leq \lambda_i \leq C \text{ and } \sum_{i=1}^{|D|} \lambda_i c_i = 0$$

Η μόνη διαφορά με την προηγούμενη περίπτωση είναι το άνω όριο που τίθεται στα λ_i. Μάλιστα παρατηρούμε πως στο δυικό δεν εμφανίζονται ούτε τα ξ_i ούτε τα μ_i. Η εύρεση των παραμέτρων ($\mathbf{w}$,b) γίνεται όπως και πριν. Η επιφάνεια απόφασης έχει ακριβώς την ίδια μορφή με την προηγούμενη περίπτωση δηλαδή:

$$\left(\sum_{i=1}^{|D|} \lambda_i c_i \mathbf{d}_i \cdot \mathbf{d} \right) + b = 0$$

3.3.3 Μη γραμμικό SVM

Οι δύο περιπτώσεις που εξετάσαμε στις προηγούμενες ενότητες επιτρέπουν την εύρεση ενός υπερεπιπέδου απόφασης όταν τα δεδομένα μας είναι γραμμικά διαχωρίσιμα. Για να κατορθώσουμε να επεκτείνουμε το SVM και στη μη γραμμική περίπτωση αυτό που κάνουμε είναι να απεικονίσουμε τα δεδομένα μας σε ένα χώρο μεγαλύτερης διάστασης από

τον αρχικό και να τα διαχωρίσουμε γραμμικά σε αυτόν. Αν κατορθώσουμε αυτό αρκεί να εφαρμόσουμε την προηγούμενη μεθοδολογία για να εκπαιδεύσουμε τον ταξινομητή μας.

Το βασικό θέμα που προκύπτει είναι με ποιο τρόπο οδηγούμαστε στην μεγαλύτερη διάσταση. Ουσιαστικά είναι δύσκολο να αποφασίσουμε τι είδους συνάρτηση θα χρησιμοποιήσουμε ώστε να οδηγηθούμε στην μεγαλύτερη διάσταση και να κατορθώσουμε σε αυτή να πετύχουμε γραμμικό διαχωρισμό. Μάλιστα η επίλυση ενός προβλήματος βελτιστοποίησης σε μεγάλη διάσταση απαιτεί υψηλό υπολογιστικό κόστος.

Προς το παρόν έστω ότι έχουμε μία συνάρτηση $\mathbf{\Phi(d)}$ η οποία απεικονίζει τα μηνύματα μας στον μεγαλύτερης διάστασης χώρο. Σε αυτή τη περίπτωση η γραμμική επιφάνεια απόφασης θα είναι της μορφής:

$$\mathbf{w} \cdot \mathbf{\Phi(d)} + b = 0$$

Η εκπαίδευση έγκειται στη επίλυση του εξής προβλήματος βελτιστοποίησης με περιορισμούς:

$$\min_{\mathbf{w},\xi} \left(\frac{\|\mathbf{w}\|^2}{2} + C \sum_{i=1}^{|D|} \xi_i \right)$$

$$\text{subject to } \xi_i \geq 0 \text{ and } c_i(\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{d}_i) + b) \geq 1 - \xi_i, \quad i = 1, 2, ..., |D|$$

Είναι φανερή πλέον η ομοιότητα μεταξύ της γραμμικής και μη περίπτωσης και ανάλογα με πριν καταλήγουμε στην επίλυση του εξής δυικού προβλήματος:

$$L_D = \sum_{i=1}^{|D|} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j c_i c_j \mathbf{\Phi}(\mathbf{d}_i) \cdot \mathbf{\Phi}(\mathbf{d}_j)$$

$$\text{subject to } 0 \leq \lambda_i \leq C \text{ and } \sum_{i=1}^{|D|} \lambda_i c_i = 0$$

Αφού βρούμε τα λ_i για να υπολογίσουμε τις παραμέτρους $(\mathbf{w},b)$ χρησιμοποιούμε τις ΚΚΤ συνθήκες και τη παράγωγο ως προς $\mathbf{w}$ που δίνονται από τις εξισώσεις:

$$\mathbf{w} = \sum_{i=1}^{|D|} \lambda_i c_i \mathbf{\Phi}(\mathbf{d}_i)$$

$$\lambda_i(c_i(\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{d}_i) + b) - 1 + \xi_i) = 0 \Rightarrow \lambda_i(c_i(\sum_{j=1}^{|D|} \lambda_j c_j \mathbf{\Phi}(\mathbf{d}_j) \cdot \mathbf{\Phi}(\mathbf{d}_i) + b) - 1 + \xi_i) = 0$$

Η επιφάνεια απόφασης που παίρνουμε έχει την παρακάτω μορφή που είναι ίδια με πριν αλλά τα δεδομένα απεικονίζονται στον μεγαλύτερης διάστασης χώρο:

$$\left(\sum_{i=1}^{|D|} \lambda_i c_i \mathbf{\Phi}(\mathbf{d}_i) \cdot \mathbf{\Phi}(\mathbf{d}) \right) + b = 0$$

3.3.4 Χρήση πυρήνων

Στην μη γραμμική περίπτωση αφήσαμε ένα σκοτεινό σημείο κατά την εκπαίδευση, ποια είναι η συνάρτηση $\mathbf{\Phi}(\mathbf{d})$. Την απάντηση σε αυτό δίνει η χρήση πυρήνων (kernel trick). Αν παρατηρήσουμε προσεκτικά τη μη γραμμική περίπτωση θα διαπιστώσουμε ότι κατά την εκπαίδευση απαιτείται ο υπολογισμός μόνο εσωτερικών γινομένων στο νέο χώρο και όχι απευθείας η τιμή της ίδιας της $\mathbf{\Phi}(\mathbf{d})$. Το εσωτερικό γινόμενο όπως γνωρίζουμε εκφράζει ομοιότητα μεταξύ δύο διανυσμάτων. Η χρήση πυρήνων μας δίνει τη δυνατότητα να υπολογίσουμε την ομοιότητα στο νέο χώρο με χρήση του αρχικού χώρου. Μπορούμε να γράψουμε τον πυρήνα ως:

$$K(\mathbf{u}, \mathbf{v}) = \mathbf{\Phi}(\mathbf{u}) \cdot \mathbf{\Phi}(\mathbf{v})$$

Ουσιαστικά ο πυρήνας είναι μία συνάρτηση που μας επιτρέπει να υπολογίσουμε το εσωτερικό γινόμενο στον μεγαλύτερης διάστασης χώρο κάνοντας χρήση των διανυσμάτων όπως αυτά είναι ορισμένα στο αρχικό χώρο. Ο πυρήνας έτσι μας δίνει τη δυνατότητα να αποφύγουμε τον ορισμό της συνάρτησης $\mathbf{\Phi}(\mathbf{d})$ καθώς στους υπολογισμούς μας θέλουμε μόνο εσωτερικά γινόμενα. Οι πυρήνες ικανοποιούν το θεώρημα του Mercer (Mercer's theorem) το οποίο μας εξασφαλίζει ότι αυτοί μπορούν να εκφραστούν ως εσωτερικά γινόμενα σε ένα χώρο μεγάλης διάστασης. Μάλιστα καθώς οι υπολογισμοί μας γίνονται στον αρχικό χώρο το υπολογιστικό κόστος δεν αυξάνεται και αποφεύγουμε και το

πρόβλημα της κατάρας της διάστασης (curse of dimensionality). Πλέον με τη χρήση των πυρήνων μπορούμε να γράψουμε το υπερεπίπεδο απόφασης ως εξής:

$$\left(\sum_{i=1}^{|D|} \lambda_i c_i K(\mathbf{d}_i, \mathbf{d}) \right) + b = 0$$

Στη συνέχεια παρουσιάζονται ορισμένοι γνωστοί και ευρέως χρησιμοποιούμενοι πυρήνες:

- γραμμικός: $K(\mathbf{u}, \mathbf{v}) = \mathbf{u} \cdot \mathbf{v}$
- πολυωνυμικός: $K(\mathbf{u}, \mathbf{v}) = (\gamma \mathbf{u} \cdot \mathbf{v} + r)^p, \gamma > 0$
- ακτινωτής συνάρτησης βάσης (RBF): $K(\mathbf{u}, \mathbf{v}) = e^{-\gamma \|\mathbf{u}-\mathbf{v}\|^2}$
- σιγμοειδής: $K(\mathbf{u}, \mathbf{v}) = \tanh(\gamma \mathbf{u} \cdot \mathbf{v} + r)$

Ο αναγνώστης παραπέμπεται και στα [23,24] όπου υπάρχει αναλυτική παρουσίαση της διαδικασίας εκπαίδευσης ενός SVM ταξινομητή.

3.4 Ταξινόμηση άγνωστου μηνύματος

Έχοντας πλέον εκπαιδεύσει τον ταξινομητή επιθυμούμε να τον χρησιμοποιήσουμε ώστε να βρει την κατηγορία ενός άγνωστου μηνύματος. Η απόφαση αυτή λαμβάνεται με βάση την πλευρά του υπερεπιπέδου απόφασης στην οποία βρίσκεται το διάνυσμα του μηνύματος δηλαδή με βάση το πρόσημό του. Έστω το μήνυμα d τότε η απόφαση του ταξινομητή είναι:

$$f(\mathbf{d}) = sign\left(\left(\sum_{i=1}^{|D|} \lambda_i c_i K(\mathbf{d}_i, \mathbf{d}) \right) + b \right)$$

Αν η τιμή είναι θετική τότε c=1 δηλαδή το μήνυμα είναι spam, αν είναι αρνητική τότε c=-1 και άρα το μήνυμα είναι μη-spam. Στην εξίσωση απόφασης αν έχουμε χρησιμοποιήσει γραμμικό πυρήνα τότε δημιουργούμε ένα γραμμικό SVM καθώς ο πυρήνας αυτός δεν κάνει καμία μετατροπή στην αρχική διάσταση των μηνυμάτων.

3.5 Χαρακτηριστικά SVM ταξινομητών

Οι ταξινομητές SVM έχουν ένα πλήθος από χαρακτηριστικά τα οποία κάνουν ελκυστική τη χρήση τους. Καταρχήν όπως είπαμε μπορούν να δημιουργήσουν ένα υπερεπίπεδο και στη περίπτωση που τα δεδομένα μας δεν είναι γραμμικά διαχωρίσιμα. Μάλιστα αυτό είναι το βέλτιστο υπερεπίπεδο το οποίο προσφέρει τον καλύτερο δυνατό διαχωρισμό μεταξύ των κατηγοριών.

Το γεγονός ότι το υπερεπίπεδο περιγράφεται από ένα υποσύνολο του συνόλου εκπαίδευσης, τα support vectors, έχει ως αποτέλεσμα η περιγραφή αυτού να είναι αρκετά συμπαγής. Το χαρακτηριστικό αυτό κάνει τα SVM κατάλληλα και για αυξητική μάθηση (incremental learning) σε περιπτώσεις που το σύνολο δεδομένων είναι πολύ μεγάλο [19].

Σε αντίθεση με άλλες μεθόδους ταξινόμησης όπως τα νευρωνικά δίκτυα τα οποία βρίσκουν τοπικά βέλτιστες λύσεις τα SVM βρίσκουν την ολικά βέλτιστη και μάλιστα με αλγορίθμους που το κόστος τους είναι αρκετά μικρό. Ακόμη δεν απαιτούν μεγάλη μείωση της διάστασης κατά την προεπεξεργασία καθώς μπορούν να διαχειριστούν δεδομένα με πολλούς όρους χωρίς να εμφανίζεται υπερεκπαίδευση. Γενικά τα SVM έχουν πολύ καλή απόδοση στην ταξινόμηση κειμένων και το ίδιο έχει παρατηρηθεί και για την ταξινόμηση ηλεκτρονικού ταχυδρομείου. Σε πειράματα που έχουν γίνει [11,12] βρίσκονται πάντα ανάμεσα στους ταξινομητές με τη μεγαλύτερη ακρίβεια ξεπερνώντας και τον πολύ γνωστό Naïve Bayes.

Πέρα από τα παραπάνω πλεονεκτήματα υπάρχουν και ορισμένα μειονεκτήματα που οφείλονται στις παραμέτρους του προβλήματος που πρέπει να καθορίσει ο χρήστης κατά την εκπαίδευση. Συγκεκριμένα αναφερόμαστε στο C το οποίο καθορίζει πόσο αυστηροί είμαστε με τα λάθη καθώς και στην επιλογή του πυρήνα. Μάλιστα οι πυρήνες διαθέτουν και οι ίδιοι παραμέτρους που πρέπει να καθορίσει ο χρήστης. Η επιλογή τιμών γι' αυτές είναι καθοριστικής σημασίας για την απόδοση του ταξινομητή μας. Για να βρούμε τις ιδανικές τιμές χρειαζόμαστε ένα σύνολο επικύρωσης στο οποίο θα δοκιμάζουμε τα αποτελέσματα της εκπαίδευσης με διαφορετικές παραμέτρους. Συνήθως δοκιμάζουμε συνδυασμούς των παραμέτρων και επιλέγουμε αυτόν με την καλύτερη απόδοση στο σύνολο επικύρωσης κάνοντας αυτό που ονομάζουμε αναζήτηση πλέγματος (grid search).

Από το παραπάνω βλέπουμε ότι ο ταξινομητής βρίσκει την βέλτιστη λύση για ένα συγκεκριμένο σύνολο παραμέτρων. Αν οι παράμετροι δεν έχουν επιλεγεί προσεκτικά πιθανώς η βέλτιστη λύση να μην είναι και η καλύτερη για το πρόβλημά μας. Παρόλα αυτά τα SVM χρησιμοποιούνται σε διάφορα πεδία με μεγάλη επιτυχία.

Κεφάλαιο 4: Ενεργητική μάθηση

4.1 Γενικά

Σε πολλές εφαρμογές της μηχανικής μάθησης η διαδικασία της δημιουργίας του συνόλου εκπαίδευσης, δηλαδή της χειρονακτικής ανάθεσης της κατηγορίας στα δεδομένα, είναι χρονοβόρα και κοστίζει. Η ενεργητική μάθηση (active learning) έχει ως στόχο να μειώσει τον απαιτούμενο αριθμό από χειρονακτικά ταξινομημένα δεδομένα. Συγκεκριμένα έστω ότι έχουμε διαθέσιμο ένα σύνολο δεδομένων (pool) στα οποία δεν έχουμε αναθέσει την κατηγορία που ανήκει το καθένα. Πλέον δίνουμε τη δυνατότητα στον ίδιο τον ταξινομητή να επιλέξει τα δεδομένα με τα οποία θα τον εκπαιδεύσουμε και άρα να μας «ζητήσει» την κατηγορία μόνο αυτών. Το κύριο μέλημα της ενεργητικής μάθησης είναι πως θα βρει τα «καλύτερα» δεδομένα από τα διαθέσιμα έτσι ώστε με όσο το δυνατό λιγότερα να πετυχαίνει ο ταξινομητής μας καλή απόδοση εφάμιλλη με αυτή που θα είχε αν γνώριζε την κατηγορία για ολόκληρο το σύνολο δεδομένων.

Όσον αφορά το πρόβλημα της ταξινόμησης ηλεκτρονικού ταχυδρομείου η ενεργητική μάθηση μπορεί να βοηθήσει ως εξής: ο ταξινομητής έχει πρόσβαση στα μηνύματα που έχει λάβει ο χρήστης και τον ρωτά για την κατηγορία ορισμένων εξ αυτών. Ανάλογα με την απάντηση που λαμβάνει ρωτά την κατηγορία και άλλων μηνυμάτων. Η διαδικασία επαναλαμβάνεται μερικές φορές και το αποτέλεσμα είναι η δημιουργία ενός φίλτρου προσαρμοσμένου στο συγκεκριμένο χρήστη. Μάλιστα το πλήθος μηνυμάτων που θα χρειαστεί να αναθέσει την κατηγορία ο χρήστης θα είναι αρκετά μικρότερο σε σχέση με το πλήθος όλων των μηνυμάτων που βρίσκονται στο γραμματοκιβώτιό του. Αποτέλεσμα

αυτού είναι να μειωθεί αρκετά ο κόπος που καταβάλει για την εκπαίδευση του φίλτρου ενώ ταυτόχρονα δεν θυσιάζει την απόδοσή του.

Στο κεφάλαιο αυτό παρουσιάζουμε ορισμένους αλγορίθμους για ενεργητική μάθηση με χρήση SVM. Πριν από αυτό ωστόσο παρέχουμε ορισμένα θεωρητικά κίνητρα που μας οδηγούν σε αυτούς ώστε να γίνουν καλύτερα κατανοητοί.

4.2 Θεωρητικό υπόβαθρο

Έστω ένα σύνολο από δεδομένα $D = \{d_1, d_2, \ldots, d_{|D|}\}$ στα οποία έχουμε αναθέσει την κατηγορία (c_i=1 (spam), c_i=-1 (μη-spam)) και με αυτά θέλουμε να εκπαιδεύσουμε τον SVM ταξινομητή. Όπως είπαμε και στο κεφάλαιο 3 υπάρχει ένα σύνολο από υπερεπίπεδα που διαχωρίζουν τα δεδομένα στο χώρο των χαρακτηριστικών F (feature space) χωρίς να κάνουν σφάλματα (θεωρούμε τη διαχωρίσιμη περίπτωση του SVM). Ο χώρος F είναι αυτός στον οποίο μας οδηγεί ο πυρήνας που χρησιμοποιούμε στο SVM μοντέλο μας (συνήθως είναι μεγαλύτερης διάστασης από τον αρχικό). Το σύνολο των υπερεπιπέδων μπορεί να γραφεί ως:

$$H = \left\{ f \mid f(\mathbf{d}) = \frac{\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{d})}{\|\mathbf{w}\|} \ \ where \ \ \mathbf{w} \in W \right\}$$

όπου W είναι ο χώρος των παραμέτρων (parameter space) και είναι ακριβώς ίδιος με τον χώρο των χαρακτηριστικών. Από το σύνολο των υπερεπιπέδων αυτά που διαχωρίζουν τέλεια τα δεδομένα μας λέμε ότι ανήκουν στο version space το οποίο ορίζουμε ως:

$$V = \left\{ f \in H \mid c_i f(\mathbf{d}_i) > 0, \ i = 1, 2, \ldots, |D| \right\}$$

Καθώς το H είναι ένα σύνολο από υπερεπίπεδα υπάρχει μία ένα προς ένα αντιστοιχία μεταξύ των $\mathbf{w}$ με μέτρο ένα και των υπερεπιπέδων αυτού και μπορούμε να ορίσουμε το version space ως:

$$V = \left\{ \mathbf{w} \in W \mid \|\mathbf{w}\| = 1, \ c_i(\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{d}_i)) > 0, \ i = 1, 2, \ldots, |D| \right\}$$

Από τους παραπάνω ορισμούς φαίνεται να απουσιάζει η παράμετρος b του υπερεπιπέδου. Αυτό έχει ως αποτέλεσμα τα υπερεπίπεδα να περνούν από την αρχή των αξόνων στο F. Ωστόσο αυτό αντισταθμίζεται από την χρήση πυρήνων οι οποίοι οδηγούν σε υπερεπίπεδα τα οποία δεν περνούν από την αρχή των αξόνων στον αρχικό χώρο των δεδομένων. Επίσης στα πλαίσια αυτής της θεωρητικής παρουσίασης θεωρούμε ότι όλα μας τα δεδομένα έχουν ίσα μέτρα δηλαδή $\|\mathbf{\Phi}(\mathbf{d}_i)\| = \lambda$. Αξίζει να σημειώσουμε ότι αυτό ισχύει πάντα για τον RBF πυρήνα. Το version space είναι το κεντρικό σημείο αυτής της παρουσίασης καθώς σε αυτό βρίσκεται η λύση που αναζητά η μέθοδος SVM.

Ο χώρος των χαρακτηριστικών F και ο χώρος των παραμέτρων W είναι δυικοί καθώς σημεία στον ένα αντιστοιχούν σε υπερεπίπεδα στον άλλο. Είναι προφανές ότι σημεία στον W δηλαδή παράμετροι $\mathbf{w}$ αντιστοιχούν σε υπερεπίπεδα στον F. Για το αντίστροφο διαισθητικά μπορούμε να πούμε ότι κάθε μήνυμα του συνόλου εκπαίδευσης περιορίζει τα υπερεπίπεδα που μπορούμε να χρησιμοποιήσουμε σε αυτά που το ταξινομούν σωστά. Μάλιστα τα $\mathbf{w} \in W$ που επιτρέπεται να χρησιμοποιήσουμε θα βρίσκονται μόνο στην μία πλευρά του υπερεπιπέδου που ορίζεται στο W από ένα μήνυμα του συνόλου εκπαίδευσης. Πιο αναλυτικά έστω ότι έχουμε το μήνυμα d_i του συνόλου εκπαίδευσης το οποίο ανήκει στην κατηγορία c_i. Τότε οποιοδήποτε υπερεπίπεδο διαχωρίζει τα δεδομένα μας θα πρέπει να ικανοποιεί τη σχέση: $c_i(\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{d}_i)) > 0$. Τώρα αντί να θεωρήσουμε το $\mathbf{w}$ ως το κάθετο διάνυσμα ενός υπερεπιπέδου στο F θεωρήσουμε το $\mathbf{\Phi}(\mathbf{d}_i)$ ως το κάθετο διάνυσμα ενός υπερεπιπέδου στο W μπορούμε να δούμε ότι η παραπάνω σχέση μας περιορίζει στο μισό του W.

Όπως είδαμε στο κεφάλαιο 3 τα SVM βρίσκουν το υπερεπίπεδο με το μέγιστο εύρος στο F. Μία περιγραφή του προβλήματος βελτιστοποίησης είναι η παρακάτω:

$$\max_{\mathbf{w}} \left(\min_{i} \left(c_i(\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{d}_i)) \right) \right)$$

$$\text{subject to } \|\mathbf{w}\| = 1 \text{ and } c_i(\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{d}_i)) > 0 \;\; i = 1,\ 2,\ \ldots,\ |D|$$

Σύμφωνα με αυτή επιθυμούμε να βρούμε το υπερεπίπεδο που μας δίνει τη μεγαλύτερη δυνατή απόσταση από το κοντινότερο μήνυμα του συνόλου εκπαίδευσης. Μάλιστα οι περιορισμοί μας εξασφαλίζουν ότι η λύση θα ανήκει στο V.

Καθώς θεωρούμε ότι όλα τα δεδομένα έχουν ίσα μέτρα το $\mathbf{\Phi}(\mathbf{d}_i)/\lambda$ είναι το κάθετο διάνυσμα μέτρου ένα του υπερεπιπέδου στο χώρο των παραμέτρων. Έτσι μπορούμε

να πούμε ότι η έκφραση $c_i(\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{d}_i))$ αντιστοιχεί σε λ φορές την απόσταση ανάμεσα στο σημείο $\mathbf{w}$ και το υπερεπίπεδο με κάθετο διάνυσμα $\mathbf{\Phi}(\mathbf{d}_i)$. Αυτό σημαίνει ότι το SVM προσπαθεί να βρει το σημείο στο V που έχει τη μεγαλύτερη δυνατή απόσταση από τα υπερεπίπεδα που δημιουργούνται στο χώρο των παραμέτρων από το σύνολο εκπαίδευσης. Ουσιαστικά βρίσκει το κέντρο της υπερσφαίρας με τη μεγαλύτερη ακτίνα η οποία δεν τέμνει τα υπερεπίπεδα που αντιστοιχούν στο σύνολο εκπαίδευσης. Στο χώρο των χαρακτηριστικών τα υπερεπίπεδα στα οποία ακουμπά η υπερσφαίρα αντιστοιχούν στα support vectors.

Από τα παραπάνω εύκολα συμπεραίνουμε ότι η ακτίνα της υπερσφαίρας είναι:

$$\frac{c_i(\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{d}_i))}{\lambda}$$

όπου το d_i αντιστοιχεί σε ένα support vector. Μάλιστα το μέγεθος της ακτίνας είναι ίσο με $1/\lambda$ φορές το εύρος του ταξινομητή.

4.3 Αλγόριθμοι για ενεργητική μάθηση

4.3.1 Στόχος των αλγορίθμων

Όπως αναφέραμε η ενεργητική μάθηση έχει ως στόχο τη μείωση των χειρονακτικά ταξινομημένων μηνυμάτων που απαιτούνται χωρίς ωστόσο να μειωθεί η απόδοση του ταξινομητή που δημιουργούμε. Ουσιαστικά μπορούμε να πούμε ότι η ενεργητική μάθηση αναζητά στα δεδομένα που δεν έχουν ανατεθεί κατηγορίες εκείνα που θα οδηγήσουν όσο το δυνατό γρηγορότερα στην καλύτερη προσέγγιση του $\mathbf{w}$ που θα λαμβάναμε αν ξέραμε την κατηγορία όλων των δεδομένων. Λαμβάνοντας υπόψη όσα είπαμε στην ενότητα 4.2 για να γίνει αυτό επιλέγουμε δεδομένα που τα υπερεπίπεδά τους μειώνουν όσο το δυνατό ταχύτερα το version space. Τώρα θα δώσουμε έναν ορισμό για να γίνει καλύτερα κατανοητό αυτό:

Ορισμός: Έστω V_i το version space για τον ταξινομητή μας αφού έχει λάβει την κατηγορία για i μηνύματα. Όταν ο ταξινομητής ρωτήσει την κατηγορία για το i+1 μήνυμα το version space θα γίνει ίσο με ένα εκ των δύο παρακάτω ανάλογα με την κατηγορία στην οποία αυτό θα ανήκει:

$$V_i^+ = V_i \cap \{\mathbf{w} \in W \mid +(\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{d}_{i+1})) > 0\}$$
$$V_i^- = V_i \cap \{\mathbf{w} \in W \mid -(\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{d}_{i+1})) > 0\}$$

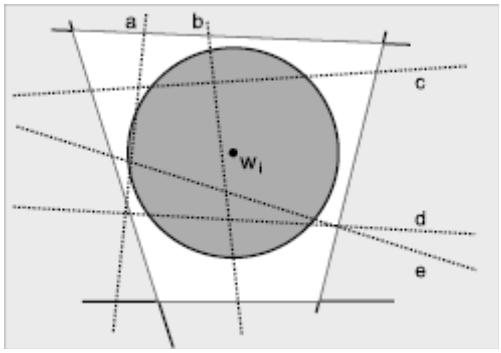
Καθώς θέλουμε η μείωση του V να είναι όσο το δυνατό μεγαλύτερη κάθε φορά θέλουμε ο ταξινομητής να ρωτά την κατηγορία για το μήνυμα εκείνο που τα V_i^+, V_i^- είναι ίσα σε μέγεθος ώστε όποια και να είναι η κατηγορία του να μειώνουμε το V όσο περισσότερο γίνεται. Ουσιαστικά προσπαθούμε με κάθε ερώτηση να μειώνουμε στο μισό το V. Μάλιστα αποδεικνύεται [17] ότι με τον τρόπο αυτό έχουμε τη ταχύτερη μείωση του V. Έτσι με κάθε ερώτηση περιορίζουμε τις δυνατές λύσεις όσο περισσότερο μπορούμε και άρα με ένα μικρό σύνολο εκπαίδευσης πετυχαίνουμε ίδια αποτελέσματα με τη χρήση ενός μεγαλύτερου. Καθώς δεν είναι εύκολο να υπολογίσουμε ακριβώς το μέγεθος για το version space οι αλγόριθμοι που παρουσιάζονται στη συνέχεια προσπαθούν να το προσεγγίσουν για να επιλέξουν το μήνυμα και να ρωτήσουν την κατηγορία του.

4.3.2 Αλγόριθμος simple margin

Έστω V_i το version space για τον ταξινομητή μας αφού έχει λάβει την κατηγορία για *i* μηνύματα. Όπως είπαμε το SVM θα βρει το κέντρο της υπερσφαίρας $\mathbf{w}_i$ που χωρά στο V_i δηλαδή δεν τέμνει τα υπερεπίπεδα που δημιουργούνται από τα *i* μηνύματα. Η θέση του V_i στην οποία βρίσκεται το κέντρο της υπερσφαίρας εξαρτάται από το σχήμα του. Ωστόσο συνήθως βρίσκεται περίπου στο κέντρο του V_i. Αυτό μας δίνει τη δυνατότητα να επιλέγουμε το μήνυμα που θα ρωτήσουμε την κατηγορία του ως εξής: για όλα τα μηνύματα ελέγχουμε πόσο κοντά βρίσκονται τα υπερεπίπεδα που δημιουργούν στο χώρο των παραμέτρων στο $\mathbf{w}_i$. Όσο πιο κοντά τόσο πιο κεντρικά θα είναι τοποθετημένα στο V_i και άρα θα το χωρίζουν στη μέση ικανοποιώντας έτσι αυτά που είπαμε για γρήγορη μείωση του version space. Η απόσταση αυτή μπορεί πολύ εύκολα να υπολογιστεί ως: $|\mathbf{w}_i \cdot \mathbf{\Phi}(\mathbf{d})|$. Έτσι βρίσκουμε την απόσταση για όλα τα μηνύματα που δεν έχουμε αναθέσει κατηγορία και επιλέγουμε αυτό με τη μικρότερη. Ο αλγόριθμος συνοψίζεται ως εξής:

1. Εκπαίδευση του SVM ταξινομητή με τα μηνύματα που γνωρίζουμε την κατηγορία τους.
2. Εύρεση της απόστασης για τα μηνύματα που δεν γνωρίζουμε την κατηγορία
3. Επιλογή αυτού με την μικρότερη απόσταση. Πήγαινε στο 1.

Στο παρακάτω σχήμα φαίνεται το version space το οποίο περιορίζεται από τα υπερεπίπεδα με συνεχή γραμμή και η υπερσφαίρα που σχηματίζεται. Οι γραμμές με κουκίδες αντιστοιχούν σε μηνύματα τα οποία δεν γνωρίζουμε την κατηγορία τους και από αυτά θέλουμε να επιλέξουμε το επόμενο που θα ρωτήσουμε. Το κοντινότερο στο κέντρο είναι το **b** και αυτό επιλέγουμε.



Σχήμα 4.1: Ο αλγόριθμος simple margin επιλέγει το b.

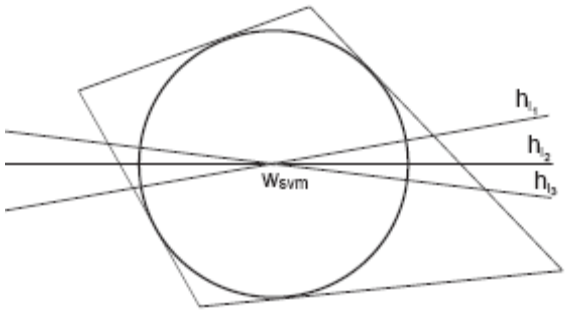
Ο αλγόριθμος αυτός που προτείνεται από τους Tong et al. [17] στηρίζεται στο ότι το version space είναι συμμετρικό και αυτό είναι ένα από τα μειονεκτήματά του. Ωστόσο πετυχαίνει πολύ καλά αποτελέσματα στα πειράματα τους.

4.3.3 Αλγόριθμος για ομάδες ενεργά επιλεγμένων μηνυμάτων

Ο προηγούμενος αλγόριθμος σε κάθε επανάληψη επιλέγει ένα μήνυμα και στη συνέχεια εκπαιδεύει ξανά τον SVM ταξινομητή. Κάτι τέτοιο έχει υψηλό υπολογιστικό κόστος ειδικά όταν το σύνολο εκπαίδευσης είναι μεγάλο. Για το λόγο αυτό είναι προτιμότερο να επιλέγουμε περισσότερα του ενός μηνύματα, ομάδες (batches), πριν εκπαιδεύσουμε ξανά τον ταξινομητή μας.

Ένας άμεσος τρόπος για να το κάνουμε αυτό θα ήταν για παράδειγμα να χρησιμοποιήσουμε τον αλγόριθμο simple margin αλλά σε κάθε βήμα να επιλέγουμε τα b>1 κοντινότερα μηνύματα. Οποιοσδήποτε όμως αλγόριθμος θα επέλεγε μηνύματα κατ' αυτό τον τρόπο θα παραβίαζε την αρχή ότι κάθε επιλεγμένο μήνυμα πρέπει να μειώνει στο μισό το version space. Με αυτό το τρόπο το πιθανότερο είναι ότι τα παραπάνω μηνύματα που επιλέγουμε σε κάθε βήμα μειώνουν πολύ λίγο το version space καθώς τα υπερεπίπεδά τους

σχηματίζουν πολύ μικρές γωνίες. Το φαινόμενο αυτό απεικονίζεται στο σχήμα που ακολουθεί.



Σχήμα 4.2: Επιλογή h>1 μηνυμάτων με βάση την απόσταση.

Για να καταφέρουμε να επιλέξουμε h>1 μηνύματα σε κάθε βήμα προτείνεται [18] ένας αλγόριθμος ο οποίος πέρα από την απόσταση λαμβάνει υπόψη του και τις γωνίες που σχηματίζονται μεταξύ των υπερεπιπέδων στο χώρο των παραμέτρων. Ουσιαστικά προσπαθεί να επιλέξει μηνύματα που τα υπερεπίπεδά τους σχηματίζουν όσο το δυνατό μεγαλύτερες γωνίες μεταξύ τους. Έτσι καταφέρνουμε να είμαστε αρκετά πιστοί στην αρχή ότι με κάθε μήνυμα πρέπει να μειώνουμε το version space στο μισό.

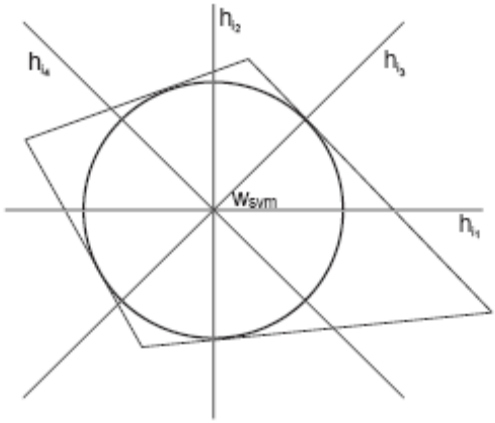
Έστω h_i , h_j τα υπερεπίπεδα στο χώρο των παραμέτρων που αντιστοιχούν στα μηνύματα d_i , d_j. Η μεταξύ τους γωνία υπολογίζεται ως:

$$\left|\cos(\angle(h_i,h_j))\right| = \frac{\left|\mathbf{\Phi}(\mathbf{d}_i)\cdot\mathbf{\Phi}(\mathbf{d}_j)\right|}{\left\|\mathbf{\Phi}(\mathbf{d}_i)\right\|\left\|\mathbf{\Phi}(\mathbf{d}_j)\right\|} = \frac{\left|K(\mathbf{d}_i,\mathbf{d}_j)\right|}{\sqrt{K(\mathbf{d}_i,\mathbf{d}_i)K(\mathbf{d}_j,\mathbf{d}_j)}}$$

Για να κατορθώσουμε να έχουμε όσο το δυνατό μεγαλύτερες γωνίες για κάθε νέο μήνυμα που επιλέγουμε για τη τρέχουσα ομάδα προσπαθούμε να μεγιστοποιήσουμε την ελάχιστη γωνία με τα μηνύματα που έχουμε ήδη επιλέξει στη τρέχουσα ομάδα. Αυτό μπορεί να εκφραστεί ως η ελαχιστοποίηση της εξής ποσότητας όπου το l περιλαμβάνει τα μηνύματα που έχουμε ήδη επιλέξει στη τρέχουσα ομάδα:

$$\max_{l\in\{1,\ldots,j-1\}}\left(\frac{\left|K(\mathbf{d}_l,\mathbf{d}_j)\right|}{\sqrt{K(\mathbf{d}_l,\mathbf{d}_l)K(\mathbf{d}_j,\mathbf{d}_j)}}\right)$$

Ουσιαστικά το επόμενο μήνυμα που επιλέγουμε είναι αυτό που ελαχιστοποιεί την ποσότητα αυτή. Η προσέγγιση αυτή οδηγεί σε επιλογή υπερεπιπέδων που στο χώρο των παραμέτρων θα έχουν διάταξη σαν και αυτή που φαίνεται στο παρακάτω σχήμα.



Σχήμα 4.3: Χρήση γωνιών κατά την επιλογή νέων μηνυμάτων.

Στην τελική του μορφή ο αλγόριθμος συνδυάζει την τεχνική με γωνίες με αυτή της μικρότερης απόστασης ώστε να πετυχαίνουμε τη μεγαλύτερη δυνατή μείωση. Έστω D^* το σύνολο με τα μηνύματα στα οποία δεν έχουμε αναθέσει ακόμη την κατηγορία τους και S είναι το σύνολο με τα μηνύματα που επιλέγουμε να ρωτήσουμε την κατηγορία τους για ένα βήμα του αλγορίθμου. Η διαδικασία μπορεί να περιγραφεί ως εξής:

1. Εκπαίδευση του SVM ταξινομητή με τα μηνύματα που γνωρίζουμε την κατηγορία τους.
2. $S = \text{Ø}$

   repeat

   $$t = \arg\min_{i \in D^* \setminus S} \left(\lambda \left| \mathbf{w} \cdot \mathbf{\Phi}(\mathbf{d}_i) \right| + (1-\lambda) \max_{j \in S} \left(\frac{\left| K(\mathbf{d}_i, \mathbf{d}_j) \right|}{\sqrt{K(\mathbf{d}_i, \mathbf{d}_i) K(\mathbf{d}_j, \mathbf{d}_j)}} \right) \right)$$

   $S = S \cup \{d_t\}$

   until $|S| = b$
3. Προσθήκη του S στο σύνολο εκπαίδευσης. Πήγαινε στο 1.

Το $\lambda \in [0,1]$ είναι μία παράμετρος η οποία καθορίζει πόση βαρύτητα θα δίνουμε στην απόσταση και πόση στις γωνίες.

Στο σημείο αυτό δίνουμε τον ψευδοκώδικα για την επιλογή μηνυμάτων μίας ομάδας [18]. Έστω D το σύνολο όλων των μηνυμάτων με τα πρώτα s-1 να αντιστοιχούν σε αυτά που έχει ανατεθεί η κατηγορία και τα υπόλοιπα να είναι υποψήφια για επιλογή στο τρέχον βήμα. Στόχος είναι η επιλογή ακόμη h μηνυμάτων για το σύνολο εκπαίδευσης.

Αλγόριθμος: Επιλογή μηνυμάτων ομάδας

είσοδος:

λ	(βαρύτητα μεταξύ απόστασης-γωνιών)
h	(μέγεθος ομάδας)
s	(πρώτη θέση για μήνυμα χωρίς κατηγορία)
D	(σύνολο δεδομένων)

έξοδος:

D	(σύνολο δεδομένων)

```
distance = array[D−s+1]
maxCos = array[D−s+1]

for j = 0 to D−s
    distance(j) = |w · Φ(d_{s+j})|
    maxCos(j) = 0
end for

for k = 0 to h−1
    minIndex = k
    minValue = +∞
    for j = k to D−s
        if distance(j) < 1
          value = λ*distance(j)+(1−λ)*maxCos(j)
          if value < minValue
            minIndex = j
            minValue = value
          end if
        end if
    end for

swap ( d_{s+minIndex} , d_{s+k} )

    for j = k+1 to D−s
        cos = |K(d_{s+k}, d_{s+j})| / sqrt(K(d_{s+k}, d_{s+k}) K(d_{s+j}, d_{s+j}))
        if cos > maxCos(j)
          maxCos(j) = cos
        end if
    end for
end for
```

Στον παραπάνω αλγόριθμο απαιτούμε η απόσταση να είναι μικρότερη του ένα ώστε το υπερεπίπεδο στο χώρο των παραμέτρων να τέμνει το version space.

Στη μέθοδο αυτή προσπαθούμε, με λίγο παραπάνω υπολογιστικό κόστος λόγω του υπολογισμού των γωνιών, να πετύχουμε ίδια ακρίβεια με μικρότερο σύνολο εκπαίδευσης σε σχέση με την επιλογή με μόνο κριτήριο την απόσταση για $b>1$. Ένα σημείο που δεν απαντάται στο [18] είναι πως επιλέγουμε την παράμετρο λ το οποίο είναι μειονέκτημα για τη μέθοδο. Στην εργασία αυτή μελετάμε τόσο τη μέθοδο αυτή όσο και τη simple margin.

Κεφάλαιο 5: Υλοποιήσεις φίλτρων για αναγνώριση spam μηνυμάτων

5.1 Γενικά

Το πρόβλημα της μαζικής αποστολής μηνυμάτων ηλεκτρονικού ταχυδρομείου ολοένα και οξύνεται με αποτέλεσμα η ζωή των χρηστών ηλεκτρονικού ταχυδρομείου να γίνεται δύσκολη καθώς καθημερινά στο γραμματοκιβώτιό τους φθάνουν πολλά spam μηνύματα. Κάτι τέτοιο μπορεί να έχει ως αποτέλεσμα το να τεθεί υπό αμφισβήτηση η χρησιμότητα του ηλεκτρονικού ταχυδρομείου ως υπηρεσίας μέσω της οποίας ο άνθρωπος επικοινωνεί και κανονίζει τα επαγγελματικά ραντεβού του. Λόγω των παραπάνω έχουν πλέον υλοποιηθεί αρκετά φίλτρα με σκοπό να διευκολύνουν τους χρήστες στο να διαχειριστούν τα μηνύματα που λαμβάνουν.

Πολλές εταιρείες έχουν δημιουργήσει φίλτρα τα οποία ενσωματώνουν στις εφαρμογές τους για αποστολή και λήψη μηνυμάτων (e-mail clients). Άλλα φίλτρα έχουν δημιουργηθεί ανεξάρτητα από τις εφαρμογές για αποστολή και λήψη μηνυμάτων και αυτές μπορούν να τα χρησιμοποιήσουν ώστε να ανακαλύψουν spam μηνύματα. Συνήθως πρόκειται για φίλτρα τα οποία προσθέτουν κάποιες κεφαλίδες στα μηνύματα που είναι spam και στη συνέχεια η εφαρμογή του χρήστη μπορεί να τα διαχειριστεί κατά το δοκούν όπως π.χ. να τα διαγράψει ή να τα μετακινήσει σε έναν κατάλογο με το όνομα "spam".

Σκοπός του κεφαλαίου αυτού είναι να παρουσιάσει ορισμένες από τις λεπτομέρειες που κρύβονται πίσω από μερικά γνωστά και ευρέως χρησιμοποιούμενα φίλτρα. Αυτά μπορούν να διαχωριστούν σε δύο κατηγορίες: φίλτρα που βασίζονται σε κανόνες (rules) και φίλτρα που στηρίζονται σε τεχνικές μηχανικής μάθησης (machine learning - ML).

Τα φίλτρα που βασίζονται σε κανόνες ακολουθούν τη προσέγγιση της μηχανικής γνώσης (knowledge engineering - KE) και προσπαθούν να ελέγξουν κατά πόσο ένα μήνυμα ικανοποιεί τους κανόνες. Οι κανόνες είναι της μορφής:

if <συνθήκη> **then** <κατηγορία = spam>

Τέτοια φίλτρα απαιτούν από τους χρήστες να δημιουργούν τους δικούς τους κανόνες και διαθέτουν εξ αρχής ορισμένους έτοιμους από τον κατασκευαστή του φίλτρου οι οποίοι ανιχνεύουν χαρακτηριστικά που είναι πολύ συνηθισμένα σε spam μηνύματα.

Τα φίλτρα που βασίζονται στη μηχανική μάθηση έχουν τη δυνατότητα να ανακαλύπτουν αυτόματα τα χαρακτηριστικά εκείνα τα οποία διαχωρίζουν τα επιθυμητά από τα ανεπιθύμητα μηνύματα. Συνήθως απαιτείται ο χρήστης να τα εκπαιδεύσει δίνοντας ως είσοδο μηνύματα και των δύο κατηγοριών. Λόγω αυτού έχουν τη δυνατότητα να προσαρμόζονται στα χαρακτηριστικά των μηνυμάτων μου λαμβάνει ο κάθε χρήστης σε αντίθεση με τα προηγούμενα όπου θα πρέπει αυτός να ανακαλύψει τα χαρακτηριστικά των μηνυμάτων που λαμβάνει ώστε να δημιουργήσει κατάλληλους κανόνες.

Αρκετά συνηθισμένο και για τις δύο παραπάνω κατηγορίες είναι να παρέχεται στο χρήστη η επιλογή να δημιουργεί λευκές λίστες (whitelists), δηλαδή λίστες με διευθύνσεις απ' όπου θέλει να λαμβάνει τα μηνύματα ανεξαρτήτως περιεχομένου καθώς και μαύρες λίστες (blacklists), δηλαδή λίστες με διευθύνσεις απ' όπου τα μηνύματα δεν είναι ευπρόσδεκτα. Ουσιαστικά τα μηνύματα με διευθύνσεις της πρώτης λίστας παραδίδονται πάντα στο γραμματοκιβώτιο του χρήστη ενώ της δεύτερης χαρακτηρίζονται πάντα spam.

5.2 Χαρακτηριστικά υλοποιημένων φίλτρων

Στην ενότητα αυτή παρουσιάζονται τα χαρακτηριστικά ορισμένων γνωστών φίλτρων. Συγκεκριμένα αναλύονται οι μέθοδοι που αυτά χρησιμοποιούν για να βρουν την κατηγορία ενός μηνύματος. Τα φίλτρα που παρουσιάζονται είναι τα: Microsoft Outlook 2000 και 2003 Junk E-mail Filter, Mozilla E-mail Client Junk Filter και SpamAssassin.

5.2.1 Microsoft Outlook 2000 Junk E-mail Filter

Το φίλτρο αυτό αποτελεί εκπρόσωπο της πρώτης κατηγορίας, δηλαδή στηρίζεται σε κανόνες [1] ώστε να βρει την κατηγορία ενός νέου μηνύματος. Μάλιστα το Outlook 2000 διαθέτει δύο φίλτρα, ένα που έχει ως στόχο την ανακάλυψη μηνυμάτων που έχουν εμπορικό περιεχόμενο (junk e-mail filter) και ένα για την εύρεση μηνυμάτων που απευθύνονται σε ενήλικες (adult content filter). Τα δύο αυτά φίλτρα μπορούν να ενεργοποιηθούν από τον χρήστη του Outlook 2000 αν το επιθυμεί αλλά οι κανόνες που περιέχουν δεν είναι δυνατό ούτε να τροποποιηθούν αλλά ούτε παρέχονται ενημερώσεις (updates) αυτών μέσω της Microsoft. Οι ενημερώσεις θα ήταν χρήσιμες καθώς το φίλτρο θα μπορούσε να προσαρμοστεί σε νέα πρότυπα spam μηνυμάτων π.χ. να ελέγχει για την έκφραση "money back " αλλά και για την "m0ney back ". Παράδειγμα αυτών των κανόνων φαίνεται στον παρακάτω πίνακα.

Τμήμα μηνύματος	**Κανόνας**	**Φίλτρο**
Body	"money back "	junk e-mail filter
Body	"Dear friend"	junk e-mail filter
Body	"Guarantee" AND ("satisfaction" OR "absolute")	junk e-mail filter
Subject	"$$"	junk e-mail filter
From	First 8 characters are digits	junk e-mail filter
Body	"must be 18"	adult content filter
Body	"18+"	adult content filter
Body	" xxx!"	adult content filter
Subject	"free" AND "adult"	adult content filter
Subject	"adults only"	adult content filter

Πίνακας 5.1: Παράδειγμα κανόνων του Outlook 2000.

Στον πίνακα παρουσιάζονται το τμήμα του μηνύματος το οποίο ελέγχει ο κάθε κανόνας καθώς και σε πιο φίλτρο αντιστοιχεί. Στους κανόνες που εμφανίζεται σύζευξη (AND) πρέπει να ικανοποιούνται όλες οι συνθήκες για να επαληθευτούν ενώ σε αυτούς με διάζευξη (OR) αρκεί μόνο μία. Αν η περιοχή περιλαμβάνει τις αναζητούμενες εκφράσεις το μήνυμα είναι spam. Να τονίσουμε πως αρκεί ένας μόνο κανόνας να ισχύει και η έκφραση του να εμφανίζεται μία φορά για να χαρακτηριστεί το μήνυμα ανεπιθύμητο. Δηλαδή αν το μήνυμα εμφανίζει στο σώμα μία φορά την έκφραση "Dear friend" τότε είναι spam.

Ακόμη παρέχεται η δυνατότητα ο χρήστης να δημιουργήσει δικούς του κανόνες μέσω ενός μενού που έχει δημιουργηθεί για αυτό το σκοπό (rules wizard). Τα βασικά βήματα σε αυτή τη διαδικασία είναι ο καθορισμός του τμήματος που ο κανόνας θα ελέγχει και βεβαίως οι λέξεις που θα προσπαθεί να εντοπίσει. Να τονιστεί πως όλοι οι κανόνες κάνουν διάκριση μεταξύ κεφαλαίων και πεζών (case sensitive).

Επιπλέον υπάρχει η επιλογή οι χρήστες να δημιουργούν λίστες με επιθυμητούς και ανεπιθύμητους αποστολείς, δηλαδή λευκές και μαύρες λίστες. Η πρώτη λίστα ονομάζεται exception list και περιλαμβάνει διευθύνσεις απ' όπου τα μηνύματα πρέπει πάντα να παραδίδονται στα εισερχόμενα. Η δεύτερη λίστα χωρίζεται σε δύο την junk senders και την adult content senders που προφανώς σχετίζονται με το περιεχόμενο του μηνύματος. Ωστόσο διευθύνσεις που βρίσκονται σε αυτές είναι μη επιθυμητές και τα μηνύματα χαρακτηρίζονται αυτομάτως spam.

Τέλος εφόσον ένα μήνυμα χαρακτηριστεί ως spam υπάρχουν τρεις επιλογές για τη διαχείρισή του: αυτόματη διαγραφή, μετακίνηση σε επιθυμητό κατάλογο και τέλος παράδοση στα εισερχόμενα με χρωματική επισήμανση.

5.2.2 Microsoft Outlook 2003 Junk E-mail Filter

Σε αυτή τη νεότερη έκδοση του Outlook η Microsoft εγκατέλειψε τη προσέγγιση με κανόνες και πέρασε στη μηχανική μάθηση [2]. Πρόκειται για στατιστική μέθοδο που στηρίζεται στην εύρεση της πιθανότητας ένα μήνυμα να είναι spam με βάση το περιεχόμενο του.

Το φίλτρο βασικά αποτελείται από ένα λεξικό που περιλαμβάνει τις λέξεις τις οποίες αναζητά καθώς και το βάρος της καθεμίας. Το βάρος κάθε λέξης έχει εξαχθεί από την Microsoft με βάση στατιστικές μεθόδους και δεν μπορεί να αλλάξει. Έχουν χρησιμοποιηθεί μηνύματα που έχουν αποσταλεί στη Microsoft από την spam fighting community τα οποία έχει επεξεργαστεί και εξάγει το λεξικό και τα βάρη.

Πιο αναλυτικά όταν ο χρήστης του Outlook το εγκαθιστά παρέχεται ένα αρχείο που περιέχει τις αντιστοιχίες λέξεων και βαρών. Βεβαίως στο αρχείο αντί για τις λέξεις υπάρχει ο κώδικας κατακερματισμού τους ώστε να γίνεται η αντιστοιχία με αυτές ενός νέου μηνύματος. Μάλιστα πολλές λέξεις εμφανίζονται δύο φορές στο λεξικό καθώς τα βάρη διαφέρουν ανάλογα με το τμήμα του μηνύματος που αυτές εντοπίζονται και συγκεκριμένα στο σώμα (body) ή στο θέμα (subject). Στη συνέχεια φαίνεται ένα παράδειγμα αντιστοιχίας λέξεων – βαρών.

Λέξη	Βάρος	Τμήμα μηνύματος	Κώδικας κατακερματισμού
screensavers	-0.020422	Body	dbd50355e5e865c5fc6d97cb4c8eaa28
screensavers	0.065368	Subject	716e204ac490ac3f26af4e49535cd5ea
linux	0.001100	Body	9796b4300be5b94e840969d695ba5797
linux	-0.021932	Subject	cd2be8031b4bfe90c5756100bbea3d0f
free	0.072806	Body	1ae8d55aaeee41e162b11a8aa3681b2d
free	0.069711	Subject	df6cdc35bc39daf2cb5e63b29fc8faad

Πίνακας 5.2: Λεξικό Outlook 2003, αντιστοιχία λέξεων με βάρη.

Το λεξικό πέρα από αγγλικές περιέχει και λέξεις από άλλες γλώσσες ώστε να είναι δυνατό το επιτυχημένο φιλτράρισμα μηνυμάτων που δεν έχουν γραφεί στα αγγλικά. Επίσης υπάρχουν λέξεις που σχετίζονται με την HTML δηλαδή ετικέτες, μέγεθος γραμματοσειράς και χρώμα γραμματοσειράς. Ο εντοπισμός ορισμένων χαρακτηριστικών HTML όπως μεγάλη γραμματοσειρά, έντονα χρώματα αποτελούν ισχυρές ενδείξεις ότι το μήνυμα είναι spam και άρα η ύπαρξη τέτοιων λέξεων στο λεξικό είναι επιθυμητή.

Όταν ένα νέο μήνυμα φθάσει στο γραμματοκιβώτιο του χρήστη αυτό επεξεργάζεται ώστε να βρεθούν οι λέξεις που περιέχει. Αν τυχόν περιέχει και HTML, όπως αναφέραμε και πριν, λαμβάνεται και αυτή υπόψη. Δηλαδή για κάθε λέξη υπολογίζεται ο κώδικας κατακερματισμού της και στη συνέχεια εντοπίζεται το βάρος που της αντιστοιχεί. Τα βάρη όλων των λέξεων αθροίζονται ώστε να υπολογιστεί το συνολικό βάρος του μηνύματος. Οι λέξεις με θετικό βάρος ενισχύουν τη πεποίθηση ότι πρόκειται για spam ενώ αυτές με αρνητικό την εξασθενούν. Στην περίπτωση που μία λέξη δεν βρεθεί στο λεξικό το βάρος που λαμβάνει είναι μηδενικό και άρα δεν συμβάλει στην τελική απόφαση.

Πέρα από τα παραπάνω υπάρχουν και ορισμένα ακόμη τεστ που γίνονται και με τη σειρά τους επηρεάζουν το τελικό βάρος. Ορισμένα εξ αυτών είναι:

- έλεγχος του χρόνου αποστολής και λήψης,
- έλεγχος στο θέμα για κεφαλαία,
- έλεγχος για μη αλφαριθμητικούς χαρακτήρες στο θέμα,
- έλεγχος διπλών χαρακτήρων στο θέμα.

Το πρώτο τεστ ερευνά την ημέρα αποστολής του μηνύματος, την ώρα που αυτό παρελήφθη καθώς και το χρόνο που μεσολάβησε μεταξύ αποστολής και λήψης του. Αν ένα μήνυμα έχει μεγάλη διαφορά μεταξύ αποστολής και λήψης έχει μεγαλύτερη πιθανότητα να

είναι spam. Το δεύτερο βρίσκει την αναλογία λέξεων στο θέμα που όλα τα γράμματά τους είναι κεφαλαία. Αν το ποσοστό είναι πάνω από 25% προστίθεται *-0.015324* στο βάρος [2]. Το τρίτο υπολογίζει το ποσοστό των χαρακτήρων του θέματος που δεν είναι γράμματα ή αριθμοί και αν υπερβαίνει το 8% τότε προστίθεται *-0.011104*. Τέλος το τέταρτο ελέγχει για διπλούς χαρακτήρες στο θέμα δηλαδή που είναι ίδιοι και εμφανίζονται συνεχόμενα μέσα σε μία λέξη. Όσο συχνότερο είναι το μοτίβο αυτό τόσο πιθανότερο να είναι spam το μήνυμα που λάβαμε. Το δεύτερο και τρίτο τεστ μας παραξενεύουν οφείλουμε να ομολογήσουμε καθώς τα βάρη που αναθέτουν θα περιμέναμε να είναι θετικά και όχι αρνητικά καθώς τέτοια χαρακτηριστικά παραπέμπουν σε spam μηνύματα κυρίως.

Αφού πραγματοποιηθούν και τα παραπάνω τεστ και το βάρος τους προστεθεί σε αυτό που έχει προκύψει από τις λέξεις του σώματος και του θέματος λαμβάνουμε το τελικό βάρος του μηνύματος. Στη συνέχεια το βάρος κανονικοποιείται στο διάστημα [0,1]. Μπορούμε να πούμε πως η τιμή που λαμβάνει εκφράζει την πιθανότητα το μήνυμα να είναι spam. Σαν τελευταίο βήμα η Microsoft έχει ορίσει δέκα επίπεδα στο παραπάνω διάστημα (επίπεδα 0 , 1 , … , 9) και ανάλογα με το κανονικοποιημένο βάρος γίνεται η αντιστοιχία σε ένα εξ αυτών. Το επίπεδο αποτελεί το τελικό αποτέλεσμα του φιλτραρίσματος. Παρακάτω παρουσιάζονται τα δέκα επίπεδα.

Επίπεδο	**Κανονικοποιημένο βάρος**
0	0.00000000
1	0.30000001
2	0.56000000
3	0.67100000
4	0.73000002
5	0.80000001
6	0.93099999
7	0.94999999
8	0.95999998
9	0.98000002

Πίνακας 5.3: Αντιστοιχία επιπέδων με κανονικοποιημένα βάρη.

Το επίπεδο σε συνδυασμό με τη ρύθμιση της ευαισθησίας του φίλτρου από τον χρήστη καθορίζουν αν το μήνυμα θα χαρακτηριστεί spam. Το φίλτρο διαθέτει τέσσερεις επιλογές λειτουργίας. Στην πρώτη μόνο μηνύματα από τη λίστα με απαγορευμένες διευθύνσεις χαρακτηρίζονται spam. Ουσιαστικά δεν γίνεται φιλτράρισμα των νέων μηνυμάτων (no automatic filtering). Στη δεύτερη η ευαισθησία είναι χαμηλή (low) και

μόνο τα μηνύματα που το επίπεδό τους ξεπερνά το 6 είναι spam. Στη τρίτη η ευαισθησία είναι υψηλή (high) και μηνύματα με επίπεδο 3 ή μικρότερο καταλήγουν στα εισερχόμενα. Η τελευταία επιλογή είναι και η αυστηρότερη καθώς μόνο μηνύματα που προέρχονται από αποστολείς που είναι εγγεγραμμένοι στη λίστα με επιτρεπτές διευθύνσεις γίνονται δεκτά, δηλαδή αποστολείς που βρίσκονται στη λευκή λίστα.

Όπως γίνεται φανερό και από πιο πάνω το φίλτρο διαθέτει λίστες όπου ο χρήστης μπορεί να καταχωρεί διευθύνσεις απ' όπου επιθυμεί πάντα τα μηνύματα να παραδίδονται στα εισερχόμενα, δηλαδή λευκές λίστες (safe senders, safe recipients) καθώς και διευθύνσεις που να οδηγούν κατευθείαν σε χαρακτηρισμό ως ανεπιθύμητο, δηλαδή μαύρες λίστες (blocked senders). Ανεξάρτητα από τη ρύθμιση του φίλτρου αυτό παρακάμπτεται για διευθύνσεις που ανήκουν σε κάποια από τις ανωτέρω λίστες.

Από την πλευρά του χρήστη αυτός δεν μπορεί ούτε να αλλάξει τα βάρη, αλλά ούτε και να εκπαιδεύσει το φίλτρο ώστε να προσαρμοστεί στα μηνύματα που αυτός λαμβάνει. Το μόνο που του μένει είναι να καθορίσει την ευαισθησία του (ή να το απενεργοποιήσει) και να κατεβάζει τις ενημερώσεις που παρέχει ο κατασκευαστής, οι οποίες εφοδιάζουν το φίλτρο με νέο λεξικό ώστε να ανιχνεύονται νέα χαρακτηριστικά spam μηνυμάτων.

5.2.3 Mozilla E-mail Client Junk Filter

Η εφαρμογή για αποστολή και λήψη μηνυμάτων του Mozilla έχει εφοδιαστεί με ένα φίλτρο για εύρεση των spam μηνυμάτων που λαμβάνει ο χρήστης και βασίζεται στην μηχανική μάθηση [3]. Στηρίζεται σε στατιστικές μεθόδους και συγκεκριμένα σε μία μορφή ταξινομητή Bayes που έχει προτείνει ο P. Graham [4].

Ο χρήστης πρέπει να εκπαιδεύσει το φίλτρο ώστε αυτό να εξάγει για κάθε λέξη που περιλαμβάνεται στα μηνύματα του συνόλου εκπαίδευσης μία πιθανότητα που να δείχνει κατά πόσο αυτή αποτελεί ένδειξη spam μηνύματος. Με αυτό τον τρόπο δημιουργείται ένα λεξικό που αντιπροσωπεύει τα μηνύματα ενός συγκεκριμένου χρήστη.

Η εκπαίδευση του φίλτρου γίνεται ως εξής: ο χρήστης πρέπει να τροφοδοτήσει το φίλτρο με έναν αριθμό μηνυμάτων που να τα έχει ο ίδιος χειρονακτικά προηγουμένως διαχωρίσει μεταξύ των δύο κατηγοριών. Στη συνέχεια εξάγονται οι λέξεις από τα μηνύματα και δημιουργούνται δύο πίνακες κατακερματισμού ένας για κάθε κατηγορία. Κάθε πίνακας περιέχει την αντιστοιχία των λέξεων με τον αριθμό εμφανίσεων αυτών στο σύνολο μηνυμάτων της κατηγορίας. Για παράδειγμα η λέξη "madam" εμφανίζεται 30 φορές στην

κατηγορία spam (όχι σε 30 διαφορετικά μηνύματα αλλά 30 φορές πιθανώς σε ένα μόνο μήνυμα).

Κατά την ανάλυση των μηνυμάτων ως μέρος μίας λέξης εκλαμβάνονται αλφαριθμητικοί χαρακτήρες, παύλες (“ – ”), απόστροφοι (“ ’ ”) και δολάρια (“ $ ”). Όλα τα άλλα σύμβολα θεωρείται ότι διαχωρίζουν τις λέξεις μεταξύ τους. Μάλιστα λέξεις που τυχόν αποτελούνται μονάχα από αριθμούς δεν θεωρούνται μέρος του λεξικού. Το ίδιο ισχύει και για τα σχόλια της HTML.

Τα τμήματα του μηνύματος που ο αλγόριθμος ελέγχει για να ανακαλύψει λέξεις είναι το σώμα, το θέμα καθώς και όλες οι άλλες κεφαλίδες. Επίσης και η HTML θεωρείται μέρος του μηνύματος και άρα ελέγχεται και αυτή. Ακόμη λαμβάνεται υπόψη η χρήση java (javascripts) καθώς και οποιεσδήποτε επισυνάψεις (attachments) είναι δυνατό να αποκωδικοποιηθούν ως απλό κείμενο.

Ως επόμενο βήμα στην εκπαίδευση του φίλτρου έχουμε τη δημιουργία ενός τρίτου πίνακα κατακερματισμού όπου λαμβάνοντας υπόψη τον αριθμό εμφανίσεων μίας λέξης και στις δύο κατηγορίες εξάγουμε την πιθανότητα ένα μήνυμα να είναι spam εφόσον περιέχει τη λέξη αυτή. Η πιθανότητα υπολογίζεται ως εξής:

$$P(spam \mid \text{word}) = \frac{\frac{b}{nbad}}{\frac{g}{ngood} + \frac{b}{nbad}}$$

Στην παραπάνω εξίσωση τα *b*, *g* αντιστοιχούν στον αριθμό εμφανίσεων μίας λέξης στην κατηγορία spam και μη-spam αντίστοιχα ενώ τα *nbad*, *ngood* αντιστοιχούν στο πλήθος μηνυμάτων της κάθε κατηγορίας. Να τονίσουμε ότι το *g* είναι ο αριθμός εμφανίσεων της λέξης στην κατηγορία μη-spam πολλαπλασιασμένος επί δύο ώστε να υπάρχει μεροληψία υπέρ αυτής ώστε να μειωθεί ο αριθμός μη-spam μηνυμάτων που θεωρούνται λανθασμένα ως spam (false positives). Επίσης λέξεις που εμφανίζονται μόνο σε μία εκ των δύο κατηγοριών του συνόλου εκπαίδευσης λαμβάνουν πιθανότητα *0.01* και *0.99* αν ανήκουν στην κατηγορία μη-spam και spam αντίστοιχα.

Αφού ολοκληρωθεί η εκπαίδευση όταν ένα νέο μήνυμα φθάσει στο γραμματοκιβώτιο του χρήστη εξάγονται οι λέξεις που αυτό περιέχει και αντιστοιχίζονται σε αυτές του λεξικού. Αν μία λέξη δε βρίσκεται στο λεξικό λαμβάνει πιθανότητα *0.40*. Στη συνέχεια επιλέγονται οι δεκαπέντε (15) λέξεις των οποίων η πιθανότητα εμφανίζει τη μεγαλύτερη διαφορά από το *0.50* (το μεγαλύτερο δέλτα). Το *0.50* δείχνει ότι μία λέξη δεν

μας βοηθά να διαχωρίσουμε τις κατηγορίες καθώς τους αναθέτει ίδια πιθανότητα γι' αυτό και επιλέγουμε αυτές με το μεγαλύτερο δέλτα από την τιμή αυτή. Στη συνέχεια η πιθανότητα του μηνύματος υπολογίζεται με μία τροποποίηση του νόμου του Bayes η οποία είναι:

$$P(spam|w_1 \cap ... \cap w_{15}) = \frac{P(spam|w_1)...P(spam|w_{15})}{P(spam|w_1)...P(spam|w_{15}) + (1 - P(spam|w_1))...(1 - P(spam|w_{15}))}$$

Όπως μπορούμε να δούμε η πιθανότητα υπολογίζεται πολλαπλασιάζοντας και διαιρώντας με τις πιθανότητες των δεκαπέντε επιλεγμένων λέξεων. Αν $P(spam \mid w_1 \cap ... \cap w_{15}) > 0.90$ τότε το μήνυμα χαρακτηρίζεται spam από τον αλγόριθμο.

Μάλιστα καθώς ο χρήστης χρησιμοποιεί το φίλτρο μπορεί εφόσον παρατηρήσει ότι ένα μήνυμα έχει ταξινομηθεί λανθασμένα να διορθώσει την κατηγορία του και αυτό, καθώς διαθέτει δυνατότητα αυξητικής μάθησης (incremental learning), να ενημερώσει τις πιθανότητες των λέξεων κατάλληλα χωρίς να απαιτείται εκπαίδευση με ολόκληρη τη συλλογή μηνυμάτων από την αρχή.

Ακόμη μπορούν να δημιουργηθούν λευκές λίστες με διευθύνσεις απ' όπου ο χρήστης επιθυμεί να λαμβάνει πάντα τα μηνύματα και μαύρες λίστες απ' όπου δεν θέλει να τα λαμβάνει (whitelists και blacklists). Αν ο αποστολέας ανήκει σε μία εξ αυτών τότε το φίλτρο παρακάμπτεται.

Συνολικά μπορούμε να πούμε ότι το φίλτρο αυτό στηρίζεται αποκλειστικά σε μηχανική μάθηση, απαιτεί στην αρχή εκπαίδευση ώστε να καταστεί λειτουργικό και έχει την ικανότητα να μαθαίνει συνεχώς όταν διορθώνεται η κατηγορία ενός μηνύματος.

5.2.4 SpamAssassin

Το SpamAssassin [5] είναι ένα φίλτρο το οποίο σε αντίθεση με τα προηγούμενα δεν έχει δημιουργηθεί ώστε να συνοδεύει μία εφαρμογή αποστολής και λήψης μηνυμάτων αλλά μπορεί να εγκατασταθεί στον υπολογιστή ενός χρήστη ή ακόμη και σε έναν εξυπηρετητή και να το χρησιμοποιήσει οποιαδήποτε τέτοια εφαρμογή. Μάλιστα συνδυάζει και τις δύο διαθέσιμες τεχνικές, τόσο αυτή με κανόνες όσο και αυτή με μηχανική μάθηση. Επιπλέον χρησιμοποιεί και μαύρες λίστες που βρίσκονται στο διαδίκτυο (on-line blacklists) καθώς και τεχνικές που στηρίζονται σε άθροισμα ελέγχου (checksum based techniques). Οι δύο τελευταίοι έλεγχοι ανήκουν στους αποκαλούμενους ελέγχους στο δίκτυο (net tests).

Για κάθε έλεγχο που επαληθεύεται το φίλτρο αναθέτει ένα βάρος. Το συνολικό βάρος που λαμβάνει ένα μήνυμα προκύπτει από το άθροισμα των βαρών όλων των κανόνων που ενεργοποιήθηκαν. Μάλιστα υπάρχουν συνήθως τέσσερα διαφορετικά βάρη για κάθε κανόνα ανάλογα με το αν ο χρήστης έχει ενεργοποιήσει ή όχι τους ελέγχους στο δίκτυο και αν έχει ενεργοποιήσει ή όχι τη μηχανική μάθηση. Μερικές φορές υπάρχει μόνο ένα βάρος που εφαρμόζεται και για τις τέσσερεις περιπτώσεις. Τα βάρη έχουν προκύψει με τη χρήση γενετικού αλγορίθμου (genetic algorithm).

Όσον αφορά το κομμάτι της μηχανικής μάθησης το SpamAssassin στηρίζεται σε μία μορφή ταξινομητή Bayes. Πρόκειται βασικά για τον ίδιο αλγόριθμο που χρησιμοποιεί ο Mozilla και ο αναγνώστης παραπέμπεται στην προηγούμενη ενότητα για λεπτομέρειες καθώς και στο [4]. Ωστόσο σε αυτή την περίπτωση είναι δυνατό να γίνει και αυτόματη εκπαίδευση του φίλτρου. Τότε τα μηνύματα που χρησιμοποιούνται είναι αυτά για τα οποία υπάρχει «μεγάλη» σιγουριά για την κατηγορία στην οποία ανήκουν. Λέγοντας «μεγάλη» σιγουριά εννοούμε πως είτε έχουν ενεργοποιήσει πολλούς κανόνες, άρα είμαστε πεπεισμένοι ότι είναι spam, είτε λίγους, άρα δεν είναι spam.

Οι κανόνες που χρησιμοποιεί το φίλτρο αυτό διαφέρουν εντελώς από τους πολύ απλούς κανόνες που είδαμε στο Outlook 2000. Πρόκειται για κανόνες που όταν επαληθεύονται προσθέτουν το βάρος που τους αντιστοιχεί στο συνολικό αντί να οδηγούν απευθείας στο χαρακτηρισμό ως spam. Αυτοί περιλαμβάνουν ελέγχους όπως:

- ποσοστό κενών γραμμών και HTML στο σώμα,
- αναφορά σε φάρμακα και δοσολογίες με πιθανά κενά στο όνομα τους,
- απουσία κεφαλίδων στο μήνυμα,
- ύπαρξη λέξεων με κεφαλαία γράμματα,
- διαφορά μεταξύ ημερομηνίας αποστολής και λήψης,
- ύπαρξη συγκεκριμένων λέξεων σε διάφορα τμήματα του μηνύματος,
- μέγεθος γραμματοσειράς,
- ύποπτη μορφή της ταυτότητας του μηνύματος (Message-ID),
- χρώμα στην HTML κ.α.

Όταν ένα νέο μήνυμα καταφθάσει αυτό ελέγχεται με βάση τους κανόνες αυτούς, οι οποίοι ξεπερνούν τους 400, και το βάρος όσων ενεργοποιήθηκαν προστίθεται. Να σημειώσουμε ότι υπάρχουν κανόνες με θετικό και αρνητικό βάρος που καταδεικνύουν αντίστοιχα spam και μη-spam χαρακτηριστικά. Αν το σύνολο αυτό ξεπερνά ένα κατώφλι, η προεπιλεγμένη τιμή γι' αυτό είναι *5* και μπορεί να αλλάξει από τον χρήστη, τότε το μήνυμα

χαρακτηρίζεται ως spam. Όταν αποδοθεί αυτός ο χαρακτηρισμός το SpamAssassin προσθέτει ορισμένες κεφαλίδες στο μήνυμα και πιθανώς μία αναφορά με όλους τους κανόνες που επαληθεύθηκαν και το βάρος που αναλογεί στον καθένα. Μάλιστα είναι δυνατό στο θέμα να προσαρτηθεί η ακολουθία "*****SPAM*****". Στη συνέχεια είναι δουλειά της εφαρμογής του χρήστη να επεξεργασθεί αυτά τα μηνύματα περαιτέρω αλλιώς θα καταλήξουν στα εισερχόμενα.

Ως μέρος των κανόνων θεωρούνται και οι έλεγχοι στο δίκτυο. Όπως αναφέραμε αυτοί περιλαμβάνουν μαύρες λίστες στο διαδίκτυο καθώς και ελέγχους με άθροισμα ελέγχου. Το φίλτρο αυτό διαθέτει πρόσβαση σε ορισμένες μαύρες λίστες (π.χ. NJABL, SORBS) και αυτό που κάνει όταν λαμβάνει ένα μήνυμα είναι να αποστείλει ένα ερώτημα προς αυτές αν η IP του αποστολέα είναι καταχωρημένη. Αν αυτό συμβαίνει η λίστα απαντά με μία διεύθυνση της μορφής *127.0.0.x* όπου το *x* καταδεικνύει τη βάση δεδομένων της λίστας όπου βρέθηκε. Κάθε μία τέτοια βάση περιέχει διευθύνσεις που είναι ύποπτες εξαιτίας συγκεκριμένων λόγων (λεπτομέρειες στο [6]).

Όσον αφορά τα τεστ με άθροισμα ελέγχου αυτά στηρίζονται στην εξής ιδέα: αν κάποιος λάβει ένα spam τότε μπορεί να το αναφέρει σε μία βάση δεδομένων στο διαδίκτυο υπολογίζοντας ένα άθροισμα ελέγχου για το μήνυμα και αποστέλλοντας το σε αυτή. Στη συνέχεια όταν παραληφθεί ένα νέο μήνυμα υπολογίζεται το άθροισμα ελέγχου του και συγκρίνεται με τα καταχωρημένα στη βάση δεδομένων. Αν υπάρξει αντιστοιχία το βάρος αυξάνεται. Το SpamAssassin χρησιμοποιεί τα εξής τεστ: DCC (Distributed Checksum Clearinghouse), Pyzor, Vipul's Razor (λεπτομέρειες στο [7]).

Διατίθενται επίσης λευκές και μαύρες λίστες που μπορεί να δημιουργεί ο χρήστης και διευθύνσεις που βρίσκονται σε αυτές έχουν ως αποτέλεσμα τα μηνύματα είτε να καταλήγουν πάντα στα εισερχόμενα είτε να χαρακτηρίζονται ανεπιθύμητα.

Επιπλέον ο χρήστης έχει την δυνατότητα να δημιουργήσει δικούς του κανόνες καθώς και να τροποποιήσει τα βάρη από τους προκαθορισμένους. Όπως αναφέραμε ήδη μπορεί να απενεργοποιήσει τη μηχανική μάθηση και τους ελέγχους στο δίκτυο. Παρακάτω παρουσιάζονται ορισμένοι κανόνες και η αντιστοιχία τους με βάρη. Πλήρης λίστα των κανόνων υπάρχει στο [5]. Στον πίνακα 5.4 στη στήλη βάρη ακολουθείται η εξής σύμβαση: *<έλεγχοι δικτύου>/<μηχανική μάθηση>*.

Κατηγορία ελέγχου	Τμήμα μηνύματος	Περιγραφή	Βάρος			
			off/off	on/off	off/on	on/on
Machine Learning	Body	Bayesian spam probability is 95 to 99%	0.0001	0.0001	3.0	3.0
On-line blacklist	Header	SORBS: sender is on a hijacked network	0	0.240	0	0.258
Checksum based	Full	Listed in DCC	0	1.37	0	2.17
Rule	Body	Message is 0% to 10% HTML	1.232	0.642	1.996	0.795
Rule	Header	Message-ID is unusually long	0.899	0.267	1.188	1.204

Πίνακας 5.4: Παράδειγμα ελέγχων του SpamAssassin.

5.3 Σύγκριση υλοποιημένων φίλτρων

Σε αυτή την ενότητα γίνεται μία απόπειρα σύγκρισης των φίλτρων που παρουσιάστηκαν προηγουμένως παρουσιάζοντας ορισμένα από τα πλεονεκτήματα και μειονεκτήματα του καθενός.

Το Outlook 2000 όπως αναφέραμε ακολουθεί την προσέγγιση της μηχανικής γνώσης και βασίζεται σε ορισμένους απλούς κανόνες. Η χρήση κανόνων απαιτεί από τον χρήστη να δημιουργεί δικούς του, γεγονός που σημαίνει πως αυτός πρέπει να διαθέτει χρόνο και εμπειρία για να ανακαλύπτει τα χαρακτηριστικά των μηνυμάτων που λαμβάνει. Επίσης καθώς οι κανόνες που είναι διαθέσιμοι από τον κατασκευαστή είναι πολύ απλοί και διαθέσιμοι στο διαδίκτυο είναι πολύ εύκολο για τους αποστολείς spam (spammers) να δημιουργήσουν μηνύματα που να ξεπερνούν το εμπόδιο αυτών. Αυτό ενισχύει την άποψη ότι ο χρήστης πρέπει να φτιάξει δικούς του κανόνες και καλό θα ήταν να δημιουργήσει και μία λευκή λίστα για να αποφύγει λάθη από τους κανόνες. Συνολικά θα λέγαμε ότι το φίλτρο αυτό δεν μπορεί να κάνει και πολλά στην αντιμετώπιση του προβλήματος των spam.

Το Outlook 2003 ακολουθεί την προσέγγιση της μηχανικής μάθησης. Μάλιστα δεν απαιτεί κανένα κόπο από την πλευρά του χρήστη και είναι λειτουργικό από την πρώτη στιγμή καθώς διαθέτει έτοιμο λεξικό. Ωστόσο περιορίζει σημαντικά τον χρήστη καθώς δεν μπορεί να το εκπαιδεύσει ώστε να μειωθούν τυχόν λάθη και να προσαρμοστεί στα χαρακτηριστικά των μηνυμάτων που λαμβάνει. Εν μέρει αυτό αντισταθμίζεται από την παροχή ενημερώσεων μέσω του κατασκευαστή εφοδιάζοντας έτσι το φίλτρο με πληρέστερο λεξικό προσαρμοσμένο στις νέες μορφές των spam. Τέλος το γεγονός ότι το λεξικό είναι

ίδιο για όλους δίνει τη δυνατότητα στους αποστολείς spam να ξεπερνούν ευκολότερα το εμπόδιό του.

Το φίλτρο που χρησιμοποιεί ο Mozilla βασίζεται και αυτό σε μηχανική μάθηση. Αντίθετα όμως με το προηγούμενο ο χρήστης πρέπει στην αρχή να το εκπαιδεύσει καθώς δεν παρέχεται κάποιο λεξικό. Η διαδικασία αυτή ωστόσο είναι πολύ απλούστερη από την συγγραφή κανόνων καθώς απαιτείται χρόνος ώστε χειρονακτικά να διαχωριστούν ορισμένα μηνύματα και να δοθούν για εκπαίδευση και τίποτα παραπάνω. Ένα σημαντικό πλεονέκτημα που απορρέει από αυτό είναι ότι το φίλτρο προσαρμόζεται στα χαρακτηριστικά των μηνυμάτων του κάθε χρήστη. Επίσης αφού το λεξικό δεν είναι ίδιο για όλους η δουλειά του αποστολέα spam γίνεται αρκετά δυσκολότερη ώστε να κατορθώσει να το ξεπεράσει. Ακόμη το φίλτρο μπορεί να διορθώνει λάθη που κάνει αφού υποστηρίζει αυξητική μάθηση με αποτέλεσμα την καλύτερη προσαρμογή σε κάθε χρήστη. Μπορούμε να πούμε πως το φίλτρο αυτό είναι καλύτερο από τα δύο προαναφερθέντα λόγω των δυνατοτήτων αυτών.

Το SpamAssassin όπως έχουμε δει αποτελεί συνδυασμό των δύο προσεγγίσεων. Τα πλεονεκτήματα της μηχανικής μάθησης είναι ίδια με αυτά του Mozilla καθώς γίνεται χρήση του ίδιου αλγορίθμου. Το κομμάτι με τους κανόνες έχει το μειονέκτημα ότι είναι ίδιο για όλους τους χρήστες ωστόσο πρόκειται για κανόνες πολύ πιο σύνθετους απ' αυτούς του Outlook 2000. Οι κανόνες ψάχνουν το μήνυμα για «κόλπα» που χρησιμοποιούν οι αποστολείς spam ώστε να ξεγελάσουν τα φίλτρα και να δελεάσουν τους παραλήπτες. Επίσης ελέγχονται οι κεφαλίδες για ύποπτα χαρακτηριστικά. Οι κανόνες σε συνδυασμό με τη μηχανική μάθηση ίσως αποτελούν την καλύτερη λύση στο πρόβλημα. Ακόμη η χρήση των ελέγχων στο δίκτυο βοηθούν στην ανίχνευση spam που χρησιμοποιούν για παράδειγμα κλεμμένες διευθύνσεις ή που ήδη έχουν γίνει αντιληπτά από άλλους χρήστες και έχουν αναφερθεί.

Συνολικά μπορούμε να πούμε πως τα δύο τελευταία φίλτρα αποτελούν καλύτερες λύσεις στο πρόβλημα του spam σε σχέση με αυτά της Microsoft. Μάλιστα η κατεύθυνση του SpamAssassin που συνδυάζει μηχανική γνώσης με μηχανική μάθηση ίσως να αποτελέσει και την καλύτερη λύση στο πρόβλημα αυτό.

Κεφάλαιο 6: Πειραματική μελέτη

6.1 Γενικά

Στα πλαίσια της εργασίας πραγματοποιήθηκε μία σειρά πειραμάτων με στόχο την αξιολόγηση των μεθόδων που παρουσιάζονται στα κεφάλαια 3 και 4 στην ταξινόμηση ηλεκτρονικού ταχυδρομείου. Συγκεκριμένα μία ομάδα πειραμάτων είχε ως στόχο την αξιολόγηση του SVM ταξινομητή και μία άλλη την αξιολόγηση του όταν συνδυάζεται με ενεργητική μάθηση. Όσον αφορά την ενεργητική μάθηση μελετήθηκε τόσο ο αλγόριθμος simple margin όσο και αυτός για επιλογή ομάδων (batches) μηνυμάτων. Το μέτρο με βάση το οποίο έγινε η αξιολόγηση είναι η ακρίβεια (accuracy). Στη συνέχεια παρουσιάζουμε την υποδομή της πειραματικής μελέτης, τις συλλογές δεδομένων που χρησιμοποιήσαμε, τα αποτελέσματα των πειραμάτων και τα συμπεράσματα που προκύπτουν.

6.2 Υποδομή πειραματικής μελέτης

Όπως έχουμε ήδη αναφέρει στο κεφάλαιο 2 οι συλλογές δεδομένων είναι σε μορφή η οποία δεν είναι κατάλληλη για έναν ταξινομητή καθώς αυτός συνήθως διαχειρίζεται μόνο αριθμητικά δεδομένα. Για το λόγο αυτό γίνεται προεπεξεργασία των συλλογών δεδομένων. Στην εργασία αυτή έχουμε χρησιμοποιήσει το λογισμικό Rainbow [25] για την προεπεξεργασία των μηνυμάτων το οποίο δημιουργεί το διάνυσμα κάθε μηνύματος. Το λογισμικό αυτό πραγματοποιεί τα εξής βήματα:

- ***Εύρεση λεξιλογίου***. Για το σύνολο των μηνυμάτων εξάγουμε τους όρους που υπάρχουν σε αυτά. Στα πειράματα μας κάθε όρος αντιστοιχεί σε μία λέξη που αποτελείται μόνο από γράμματα (βλέπε §2.3.1).
- ***Αφαίρεση συνηθισμένων λέξεων***. Οι λέξεις αυτές εμφανίζονται και στις δύο κατηγορίες. Το Rainbow παρέχει μία λίστα τέτοιων λέξεων και η εφαρμογή της είναι προαιρετική. Εμείς τη χρησιμοποιούμε σε όλα τα πειράματά μας.
- ***Εύρεση μορφολογικής ρίζας***. Η εφαρμογή της είναι προαιρετική και στα πειράματα δεν τη χρησιμοποιούμε.
- ***Αφαίρεση ετικετών HTML***. Σε μηνύματα που περιέχουν HTML επιλέξαμε να την αφαιρέσουμε.
- ***Αφαίρεση λέξεων με βάση τη συχνότητα εμφάνισης***. Παρέχεται η δυνατότητα αφαίρεσης λέξεων που εμφανίζονται το πολύ σε x μηνύματα. Η επιλογή αυτή χρησιμοποιείται για να μειωθεί η διάσταση.
- ***Χρήση της συνάρτησης κέρδος πληροφορίας (IG)***. Η συνάρτηση αυτή βοηθά στη μείωση της διάστασης διατηρώντας τους χρήσιμους όρους. Στη μελέτη μας τη χρησιμοποιούμε.
- ***Δημιουργία διανύσματος***. Το Rainbow μετά τα παραπάνω βήματα επιστρέφει το διάνυσμα κάθε μηνύματος με βάρη τη συχνότητα εμφάνισης κάθε όρου. Στα πειράματα χρησιμοποιούμε τα βάρη αυτά.

Η διαδικασία της προεπεξεργασίας παρουσιάζεται αναλυτικά στο κεφάλαιο 2 και εκεί παραπέμπεται ο αναγνώστης. Εδώ απλώς θέλαμε να δείξουμε τι μας παρέχει το λογισμικό που χρησιμοποιήσαμε και τις επιλογές που κάναμε.

Όσον αφορά τον ταξινομητή SVM για την υλοποίησή του χρησιμοποιήσαμε το πακέτο λογισμικού LIBSVM [26]. Σε αυτό δίνουμε ως είσοδο τα διανύσματα των μηνυμάτων όπως προκύπτουν μετά την προεπεξεργασία και ως έξοδο λαμβάνουμε ένα μοντέλο με τα support vectors και τις τιμές των πολλαπλασιαστών Lagrange που αντιστοιχούν σε αυτά. Κατά την εκπαίδευση του ταξινομητή μπορούμε να επιλέξουμε τον πυρήνα που θα χρησιμοποιήσουμε. Στα περισσότερα πειράματα έχουμε επιλέξει τον RBF πυρήνα. Μάλιστα το πακέτο αυτό παρέχει έτοιμα προγράμματα που επιτρέπουν την αξιολόγηση του ταξινομητή και επίσης την εύρεση των βέλτιστων παραμέτρων για το C (το πόσο αυστηροί είμαστε με τα σφάλματα) και τους πυρήνες κάνοντας αναζήτηση σε πλέγμα.

Για την ενεργητική μάθηση έχουμε υλοποιήσει τις δύο μεθόδους σε C. Συγκεκριμένα στον κώδικά μας επιλέγουμε ποια μηνύματα θα προστεθούν στο σύνολο

εκπαίδευσης και στη συνέχεια χρησιμοποιούμε τη LIBSVM για να εκπαιδεύσουμε και να αξιολογήσουμε τον ταξινομητή.

6.3 Συλλογές μηνυμάτων ηλεκτρονικού ταχυδρομείου

Για την αξιολόγηση των μεθόδων χρησιμοποιήθηκαν δύο συλλογές μηνυμάτων στις οποίες έχει γίνει πλήθος πειραμάτων από πολλούς ερευνητές για την αξιολόγηση ταξινομητών ηλεκτρονικού ταχυδρομείου. Πρόκειται για τις συλλογές Ling-Spam και SpamAssassin τις οποίες παρουσιάζουμε στη συνέχεια.

Ling-Spam: Η συλλογή Ling-Spam [27] περιλαμβάνει 2893 μηνύματα ηλεκτρονικού ταχυδρομείου. Τα 2412 είναι μη-spam και προέρχονται από μία λίστα γλωσσολογικών θεμάτων (linguistic mailing list). Τα υπόλοιπα 481 spam μηνύματα προέρχονται από το γραμματοκιβώτιο ενός εκ των δημιουργών της [9] και δίνουν αναλογία 16.6% σε spam μηνύματα. Στη συλλογή έχει γίνει ένα είδος προεπεξεργασίας και έχουν αφαιρεθεί όλες οι κεφαλίδες πλην του θέματος καθώς και ετικέτες HTML, επισυνάψεις. Ουσιαστικά κάθε μήνυμα αποτελείται μόνο από το θέμα και το σώμα. Είναι διαθέσιμη σε τέσσερεις εκδόσεις ανάλογα με το αν κατά την προεπεξεργασία έχει χρησιμοποιηθεί λίστα συνηθισμένων λέξεων και εύρεση μορφολογικής ρίζας. Στα πειράματά μας επιλέξαμε την έκδοση που καμία επιλογή εκ των δύο δεν είναι ενεργοποιημένη ώστε η προεπεξεργασία να γίνει με το Rainbow. Μάλιστα είναι ήδη χωρισμένη σε δέκα μέρη για 10-fold cross validation.

SpamAssassin: Η συλλογή SpamAssassin [28] αποτελείται από 6047 μηνύματα που έχουν παραχωρηθεί από διάφορους χρήστες ηλεκτρονικού ταχυδρομείου. Τα 4150 μηνύματα είναι μη-spam και τα υπόλοιπα 1897 είναι spam. Αυτό οδηγεί σε μία αναλογία 31.3% spam μηνυμάτων. Σε αντίθεση με την προηγούμενη συλλογή σε αυτή τα μηνύματα δεν έχουν υποστεί καμία προεπεξεργασία και είναι στην αρχική τους μορφή δηλαδή περιλαμβάνουν όλες τις κεφαλίδες, ετικέτες HTML και επισυνάψεις. Αυτό μας δίνει τη δυνατότητα στα πειράματά μας να εκμεταλλευτούμε όλη τη διαθέσιμη πληροφορία ενός μηνύματος. Μάλιστα υπάρχουν 250 μη-spam μηνύματα που θεωρούνται δύσκολα στη ταξινόμηση καθώς περιέχουν κοινά στοιχεία με spam μηνύματα όπως είναι η χρήση HTML. Αυτά περιλαμβάνονται στον κατάλογο hard_ham και με αυτό το όνομα θα τα αναφέρουμε στην συνέχεια.

6.4 Πειράματα

Στο σημείο αυτό θα παρουσιάσουμε τα πειραματικά αποτελέσματα που προέκυψαν από την αξιολόγηση των μεθόδων SVM και της ενεργητικής μάθησης με χρήση των δύο συλλογών μηνυμάτων που προαναφέρθηκαν. Για όλα τα πειράματα που ακολουθούν ισχύουν τα εξής: κάθε όρος του διανύσματος με το οποίο αναπαριστούμε τα μηνύματα αντιστοιχεί σε μία λέξη, έχουν αφαιρεθεί οι συνηθισμένες λέξεις, δεν χρησιμοποιούμε εύρεση της μορφολογικής ρίζας και στα μηνύματα της συλλογής SpamAssassin που περιέχουν HTML αυτή έχει αφαιρεθεί. Η αξιολόγηση των μεθόδων έγινε με το μέτρο ακρίβεια (accuracy) και δώσαμε την ίδια βαρύτητα τόσο στα spam όσο και στα μη-spam μηνύματα, δηλαδή λ=1 (βλέπε §2.9).

6.4.1 Αξιολόγηση μεθόδου SVM

Μία σειρά πειραμάτων που πραγματοποιήθηκε αποσκοπούσε στη μελέτη της απόδοσης της μεθόδου SVM. Σε όλες τις περιπτώσεις έχει χρησιμοποιηθεί ο RBF πυρήνας και τα βάρη στα διανύσματα αντιστοιχούν στη συχνότητα εμφάνισης κάθε λέξης στο μήνυμα. Επίσης η αφαίρεση λέξεων που εμφανίζονται σε λιγότερα από x μηνύματα και της συνάρτησης κέρδος πληροφορίας (IG) για μείωση της διάστασης εφαρμόστηκαν διαφορετικά για κάθε πείραμα. Μάλιστα η εκτίμηση της απόδοσης έγινε τόσο με χρήση όλων των τμημάτων των μηνυμάτων αλλά και με χρήση μόνο του θέματος καθώς αυτό περιλαμβάνει πολύ πληροφορία σχετικά με το περιεχόμενο και κατ' επέκταση για την κατηγορία.

Τα πειράματά μας ήταν τα εξής: μελέτη στη συλλογή Ling-Spam με χρήση 10-fold cross validation (βλέπε §2.8.1). Για κάθε μία εκ των δέκα επαναλήψεων βρήκαμε με χρήση της αναζήτησης σε πλέγμα τις βέλτιστες παραμέτρους. Πρόκειται για την παράμετρο C που έχει να κάνει με το πόσο αυστηροί είμαστε με τα λάθη στην εκπαίδευση και τη γ του RBF πυρήνα. Μάλιστα δοκιμάσαμε μεγάλο πλήθος τιμών ώστε να βρούμε τις βέλτιστες. Μελέτη στη συλλογή SpamAssassin με τον ίδιο τρόπο με πριν με μόνη διαφορά ότι και για τις δέκα επαναλήψεις χρησιμοποιήσαμε τον ίδιο συνδυασμό παραμέτρων. Ωστόσο και πάλι αναζητήσαμε το βέλτιστο. Επίσης στη συλλογή αυτή κάναμε το ίδιο πείραμα εξαιρώντας τα hard_ham μηνύματα και ένα άλλο όπου εκπαιδεύσαμε τον ταξινομητή με όλα τα μηνύματα πλην των hard_ham και τον αξιολογήσαμε στην ταξινόμηση αυτών. Στους πίνακες που

ακολουθούν παραθέτουμε τα αποτελέσματα αναφέροντας ταυτόχρονα και τις επιλογές τόσο για τις παραμέτρους όσο και για την προεπεξεργασία. Η στήλη «σπάνιες λέξεις» έχει να κάνει με το ότι διώχνουμε τους όρους που εμφανίζονται σε λιγότερα από x μηνύματα.

	Σπάνιες λέξεις	**IG**	***C***	***γ***	**Λεξικό**	**Ακρίβεια**
Ling-Spam	2	Όχι	$2^{9},\ldots,2^{11}$	$2^{-10},\ldots,2^{-7}$	16300-17100	97.27%
SpamAssassin	3	Ναι	2^{6}	2^{-11}	10000	98.67%
SpamAssassin (*χωρίς hard_ham*)	3	Ναι	2^{7}	2^{-13}	10000	99.39%
SpamAssassin (*μόνο hard_ham*)	3	Ναι	2^{7}	2^{-13}	10000	24.4%

Πίνακας 6.1: Μέθοδος SVM με χρήση όλων των τμημάτων κάθε μηνύματος.

	Σπάνιες λέξεις	**IG**	***C***	***γ***	**Λεξικό**	**Ακρίβεια**
Ling-Spam	Όχι	Όχι	$2^{3},\ldots,2^{4}$	$2^{-5},\ldots,2^{-3}$	2670-2730	90.7%
SpamAssassin	2	Ναι	2^{1}	2^{-1}	1500	90.04%
SpamAssassin (*χωρίς hard_ham*)	2	Ναι	2^{1}	2^{-3}	1500	90.04%
SpamAssassin (*μόνο hard_ham*)	2	Ναι	2^{1}	2^{-3}	1500	80.8%

Πίνακας 6.2: Μέθοδος SVM με χρήση μόνο του θέματος κάθε μηνύματος.

Από τα παραπάνω αποτελέσματα βλέπουμε ότι η επίδοση της μεθόδου SVM είναι πάρα πολύ καλή ακόμη και στη περίπτωση που η ταξινόμηση γίνεται μόνο με το θέμα των μηνυμάτων. Λογικό είναι ότι με χρήση μόνο του θέματος η ακρίβεια μειώνεται καθώς η πληροφορία για την κατηγορία σαφώς και είναι λιγότερη. Σημαντική είναι η διαφορά στις τιμές των παραμέτρων *C*, *γ* μεταξύ των δύο πινάκων. Ως συνέπεια αυτού σε όλα τα πειράματα όπου χρησιμοποιούμε μόνο το θέμα το πλήθος των support vectors είναι μεγαλύτερο. Στα πειράματα της Ling-Spam είπαμε ότι για κάθε επανάληψη από τις δέκα βρήκαμε ξεχωριστές παραμέτρους γι' αυτό και αναφέρεται το εύρος των τιμών. Η επίδοση και στις δύο συλλογές είναι παρόμοια παρότι η SpamAssassin θεωρείται δυσκολότερη και κυρίως αυτό οφείλεται στο ότι είναι πολύ μεγάλη (6047 μηνύματα) και άρα το σύνολο εκπαίδευσης είναι μεγάλο δίνοντας πολύ καλή ακρίβεια.

Η μόνη περίπτωση όπου η ακρίβεια είναι πολύ χαμηλή (μόλις 24.4%) είναι όταν χρησιμοποιούμε ως σύνολο ελέγχου τα μηνύματα hard_ham και η εκπαίδευση γίνεται με τα υπόλοιπα μηνύματα της συλλογής. Ο λόγος που συμβαίνει αυτό είναι διττός. Πρώτον τα μηνύματα αυτά έχουν χαρακτηριστικά που παραπέμπουν σε spam και άρα είναι δύσκολο ο ταξινομητής να τα ξεχωρίσει. Ωστόσο αυτό θα σήμαινε ότι και στη περίπτωση που χρησιμοποιούμε μόνο το θέμα η ακρίβεια θα έπρεπε να είναι χαμηλή. Κάτι τέτοιο όμως δεν συμβαίνει. Ο δεύτερος λόγος και κυρίαρχος είναι η επιλογή παραμέτρων και κυρίως του *γ*. Ενώ η πολύ μικρή του τιμή (2^{-13}) είναι βέλτιστη στο σύνολο εκπαίδευσης στο σύνολο ελέγχου οδηγεί σε μικρή ακρίβεια καθώς ο αριθμός των support vectors είναι μικρός με αποτέλεσμα να μην είναι δυνατό ο ταξινομητής να αναθέσει σωστά την κατηγορία στα «δύσκολα» αυτά μηνύματα. Αυτό επιβεβαιώθηκε και μέσω μίας σειράς πειραμάτων με μικρότερο *γ* που δεν τα παρουσιάζουμε εδώ.

6.4.2 Σύγκριση πυρήνων για τη μέθοδο SVM

Τα πειράματα που παρουσιάζονται στην ενότητα αυτή έχουν ως στόχο να συγκρίνουν διάφορους πυρήνες στη συλλογή μηνυμάτων SpamAssassin. Και εδώ χρησιμοποιούμε 10-fold cross validation, τα βάρη των διανυσμάτων αντιστοιχούν στη συχνότητα εμφάνισης του κάθε όρου και το κάθε πείραμα γίνεται τόσο με ολόκληρο το μήνυμα όσο και μόνο με το θέμα. Μάλιστα η προεπεξεργασία έχει γίνει όπως ακριβώς παρουσιάστηκε στους πίνακες 6.1 και 6.2 στην περίπτωση που χρησιμοποιούμε ολόκληρη τη συλλογή (δεύτερη γραμμή στους πίνακες).

Στην ενότητα 3.3.4 αναφέραμε λεπτομερώς τι είναι ο πυρήνας και τον ορίσαμε ως το εσωτερικό γινόμενο δύο διανυσμάτων που έχουν μετατραπεί μέσω μίας συνάρτησης $\mathbf{\Phi(d)}$ την οποία συνήθως δεν γνωρίζουμε. Στους πυρήνες που μελετάμε στη συνέχεια γνωρίζουμε ωστόσο τη συνάρτηση αυτή. Μάλιστα η νέα αναπαράσταση των διανυσμάτων δεν είναι σε μεγαλύτερη διάσταση. Τώρα θα ορίσουμε τη συνάρτηση αυτή για κάθε πυρήνα όπου $\mathbf{\Phi}_i\mathbf{(d)}$ και $\mathbf{d}_i$ είναι η συνιστώσα *i* του νέου διανύσματος και του αρχικού αντίστοιχα του μηνύματος *d*. Στους ορισμούς που παρουσιάζονται στον πίνακα 6.3 το *D* αντιστοιχεί στο σύνολο εκπαίδευσης, το $\#_D(\mathbf{d}_i)$ στο πλήθος μηνυμάτων που περιέχουν τον όρο *i* και στην περίπτωση του πυρήνα Fisher το διπλό άθροισμα στον παρανομαστή αντιστοιχεί στον αριθμό εμφανίσεων όλων των όρων σε όλα τα μηνύματα.

Όνομα πυρήνα	Μαθηματικός τύπος
Γραμμικός	$\mathbf{\Phi}_i(\mathbf{d}) = \mathbf{d}_i$
L_0 κανονικοποίηση	$\mathbf{\Phi}_i(\mathbf{d}) = \frac{\mathbf{d}_i}{\|\mathbf{d}\|_0}$ και $\|\mathbf{d}\|_0 = \sum_i \mathbf{d}_i^0$ ανν $0^0 \equiv 0$
L_1 κανονικοποίηση	$\mathbf{\Phi}_i(\mathbf{d}) = \frac{\mathbf{d}_i}{\|\mathbf{d}\|_1}$ και $\|\mathbf{d}\|_1 = \sum_i \mathbf{d}_i$
L_2 κανονικοποίηση	$\mathbf{\Phi}_i(\mathbf{d}) = \frac{\mathbf{d}_i}{\|\mathbf{d}\|_2}$ και $\|\mathbf{d}\|_2 = \sqrt{\sum_i \mathbf{d}_i^2}$
TFIDF	$\mathbf{\Phi}_i(\mathbf{d}) = \mathbf{d}_i \cdot \log \frac{\|D\|}{\#_D(\mathbf{d}_i)}$
Log-TFIDF	$\mathbf{\Phi}_i(\mathbf{d}) = \log(\mathbf{d}_i + 1) \cdot \log \frac{\|D\|}{\#_D(\mathbf{d}_i)}$
Straight-TFIDF	$\mathbf{\Phi}_i(\mathbf{d}) = \mathbf{d}_i \cdot \frac{\|D\|}{\#_D(\mathbf{d}_i)}$
Bhattacharyya	$\mathbf{\Phi}_i(\mathbf{d}) = \sqrt{\frac{\mathbf{d}_i}{\|\mathbf{d}\|_1}}$ και $\|\mathbf{d}\|_1 = \sum_i \mathbf{d}_i$
Fisher πολυωνυμικής κατανομής	$\mathbf{\Phi}_i(\mathbf{d}) = \frac{\mathbf{d}_i}{\hat{\theta}_i}$ και $\hat{\theta}_i = \frac{\sum_{\mathbf{d} \in D} \mathbf{d}_i}{\sum_{\mathbf{d} \in D} \sum_i \mathbf{d}_i}$

Πίνακας 6.3: Ορισμοί πυρήνων.

Στη συνέχεια παραθέτουμε τα πειραματικά αποτελέσματα για τους πυρήνες αυτούς όπου πέρα από την ακρίβεια παρουσιάζεται και ο μέσος αριθμός support vectors για τις δέκα επαναλήψεις του 10-fold cross validation. Παρατηρήστε πως κανένας δεν διαθέτει παραμέτρους. Έτσι στην εκπαίδευση η μόνη παράμετρος που υπάρχει είναι το *C*.

	$C=2^0$		$C=2^6$		$C=2^{10}$	
	Ακρίβεια	**SV**	**Ακρίβεια**	**SV**	**Ακρίβεια**	**SV**
Γραμμικός	98.84%	460	98.84%	460	98.84%	460
L_0 κανονικοποίηση	95.27%	1880	99.04%	640	99.19%	590
L_1 κανονικοποίηση	93.84%	2690	98.27%	775	99.1%	580
L_2 κανονικοποίηση	98.04%	770	99.07%	545	99.07%	545
TFIDF	99.09%	965	99.35%	895	99.35%	900
Log-TFIDF	99.34%	975	99.37%	945	99.37%	945
Straight-TFIDF	98.72%	2210	98.59%	2060	98.64%	2055
Bhattacharyya	99.09%	695	99.32%	570	99.32%	570
Fisher	97.99%	1515	97.99%	1515	97.99%	1515

Πίνακας 6.4: Σύγκριση πυρήνων με χρήση ολόκληρων των μηνυμάτων στη συλλογή SpamAssassin.

	$C=2^0$		$C=2^3$		$C=2^6$	
	Ακρίβεια	**SV**	**Ακρίβεια**	**SV**	**Ακρίβεια**	**SV**
Γραμμικός	89.65%	1930	89.29%	1530	89.22%	1450
L_0 κανονικοποίηση	86.63%	3000	90.47%	2020	89.72%	1700
L_1 κανονικοποίηση	86.53%	3000	90.44%	2030	89.7%	1710
L_2 κανονικοποίηση	89.52%	2320	89.69%	1840	89.34%	1500
TFIDF	90.07%	2200	89.77%	1770	89.45%	1450
Log-TFIDF	90.04%	2200	89.72%	1770	89.52%	1450
Straight-TFIDF	88.62%	2410	88.72%	2020	88.94%	1580
Bhattacharyya	89.57%	2320	89.69%	1840	89.36%	1500
Fisher	89.64%	1980	89.19%	1630	89.27%	1460

Πίνακας 6.5: Σύγκριση πυρήνων με χρήση μόνο του θέματος στη συλλογή SpamAssassin.

Από τα αποτελέσματα του πίνακα 6.4 προκύπτει ότι στη περίπτωση που χρησιμοποιούμε ολόκληρα τα μηνύματα οι πυρήνες TFIDF, Log-TFIDF και Bhattacharyya εμφανίζουν την μεγαλύτερη ακρίβεια και μάλιστα χωρίς να απαιτούν πολλά support vectors όπως για παράδειγμα ο πυρήνας Fisher. Οι $L_{0,1,2}$ κανονικοποιήσεις έχουν μειωμένη ακρίβεια για $C=2^0$ ωστόσο αυτή βελτιώνεται αισθητά για μεγαλύτερα C.

Στην περίπτωση που χρησιμοποιούμε μόνο το θέμα των μηνυμάτων παρατηρούμε πως σχεδόν για όλους τους πυρήνες η ακρίβεια μειώνεται για $C=2^6$ σε σχέση με αυτή που είχαμε για $C=2^3$ το οποίο σημαίνει ότι δεν πρέπει να είμαστε πολύ αυστηροί με τα λάθη στην ταξινόμηση του συνόλου εκπαίδευσης. Οι $L_{0,1}$ κανονικοποιήσεις εμφανίζουν και εδώ μειωμένη ακρίβεια για $C=2^0$ η οποία ωστόσο αυξάνεται για μεγαλύτερες τιμές της παραμέτρου. Τέλος αν εξαιρέσουμε τον Straight-TFIDF πυρήνα όλοι οι άλλοι πετυχαίνουν ακρίβεια 89-90.5%.

Γενικά τόσο στην περίπτωση που χρησιμοποιούμε ολόκληρα τα μηνύματα όσο και σε αυτή μόνο με το θέμα για $C\neq2^0$ όλοι οι πυρήνες πετυχαίνουν μεγάλη ακρίβεια χωρίς να μπορούμε με σαφήνεια να πούμε ποιος είναι καλύτερος.

6.4.3 Αξιολόγηση αλγορίθμου simple margin

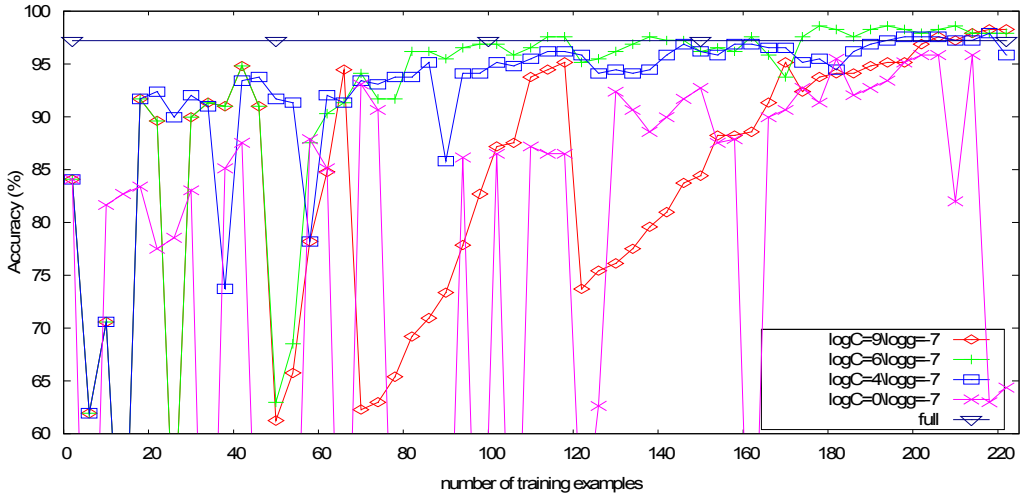
Στην ενότητα αυτή παρουσιάζουμε τα πειραματικά αποτελέσματα για τη μία εκ των δύο μεθόδων ενεργητικής μάθησης. Στα πλαίσια της μελέτης αυτής για τον ταξινομητή SVM χρησιμοποιήσαμε τον RBF πυρήνα και τα πειράματα έγιναν τόσο με ολόκληρο το μήνυμα όσο και μόνο με το θέμα. Τα βάρη στα διανύσματα αντιστοιχούν στη συχνότητα εμφάνισης κάθε όρου. Για να αξιολογηθεί η μέθοδος ως αρχικό σύνολο εκπαίδευσης στον

SVM ταξινομητή δίναμε ένα spam μήνυμα και ένα μη-spam και στη συνέχεια προσθέταμε σε αυτό μηνύματα που επιλέγει ο αλγόριθμος simple margin. Αφού ο αλγόριθμος επιλέξει έναν αριθμό μηνυμάτων ελέγχεται η ακρίβεια του ταξινομητή σε ένα σύνολο ελέγχου. Στα πειράματα αυτά χρησιμοποιήθηκε η συλλογή Ling-Spam και καθώς αυτή είναι ήδη χωρισμένη σε δέκα μέρη χρησιμοποιήσαμε τον κατάλογο part8 ως σύνολο ελέγχου και τους υπόλοιπους ως σύνολο μηνυμάτων που δεν γνωρίζουμε την κατηγορία και από αυτά επιλέγει ο αλγόριθμος για να ρωτήσει. Στον πίνακα που ακολουθεί συνοψίζουμε την προεπεξεργασία της συλλογής Ling-Spam που είναι ακριβώς ίδια με αυτή για τα πειράματα της ενότητας 6.4.1 και οι τιμές που αναφέρονται είναι αυτές που λάβαμε όταν το σύνολο ελέγχου ήταν ο κατάλογος part8 στο 10-fold cross validation της ενότητας 6.4.1. Επίσης αναφέρονται οι βέλτιστες τιμές για τη μέθοδο SVM, η ακρίβεια που πετύχαμε με χρήση ολόκληρου του συνόλου εκπαίδευσης και το μέγεθος αυτού.

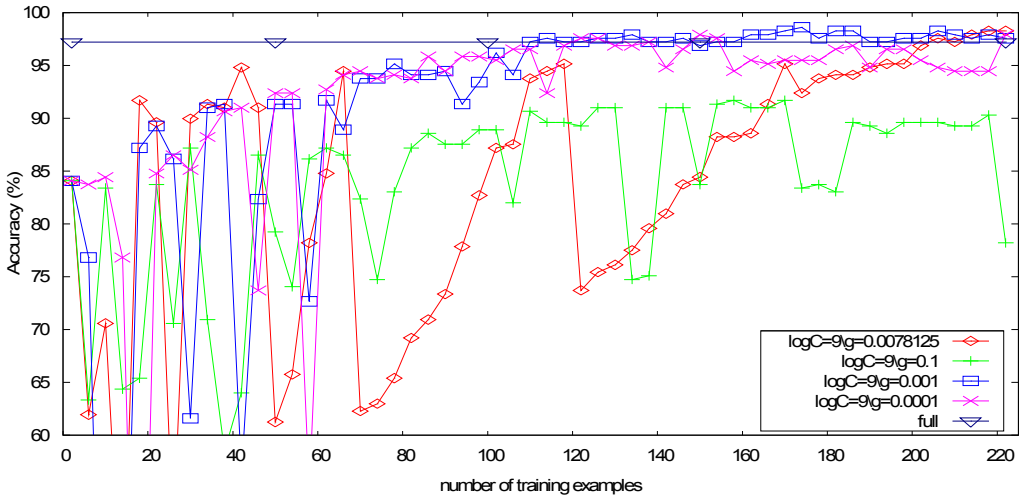
Τμήμα μηνύματος	**Σπάνιες λέξεις**	**IG**	**Λεξικό**	***C***	***γ***	**Ακρίβεια**	**Σύνολο εκπαίδευσης**
Ολόκληρο	2	Όχι	17054	2^{10}	2^{-7}	97.23%	2604
Θέμα	Όχι	Όχι	2641	$2^{3.5}$	2^{-4}	89.99%	2604

Πίνακας 6.6: Προεπεξεργασία και αποτελέσματα με χρήση ολόκληρου του συνόλου εκπαίδευσης.

Παρακάτω παρουσιάζονται τα αποτελέσματα για την περίπτωση που χρησιμοποιούμε ολόκληρο το μήνυμα.



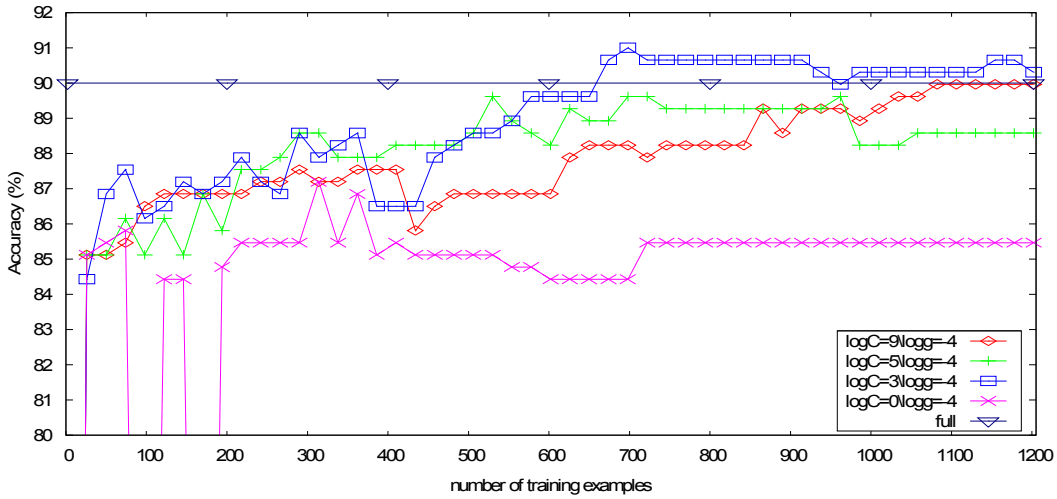
Σχήμα 6.1: Simple margin με ολόκληρο το μήνυμα για διάφορες τιμές του C στη συλλογή Ling-Spam.



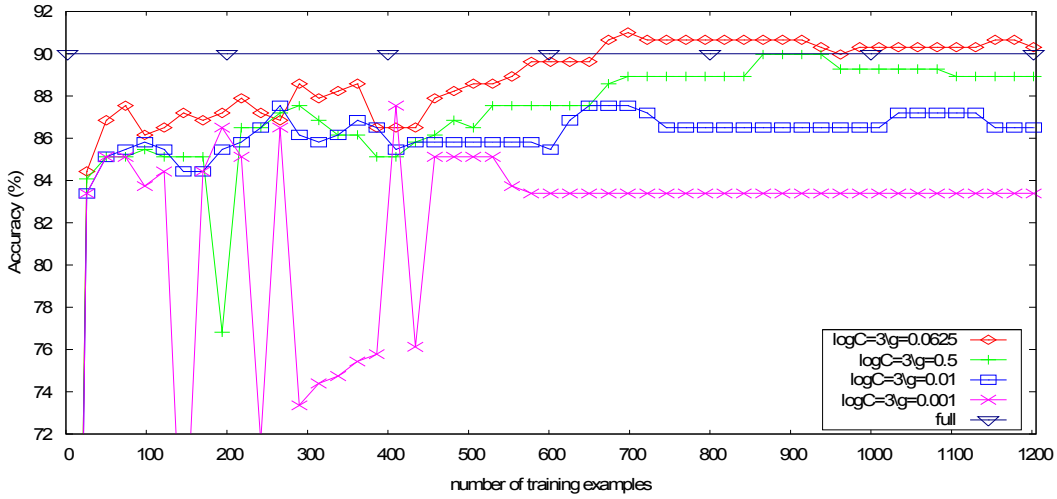
Σχήμα 6.2: Simple margin με ολόκληρο το μήνυμα για διάφορες τιμές του γ στη συλλογή Ling-Spam.

Οι γραφικές παραστάσεις παρουσιάζουν την ακρίβεια συναρτήσει του μεγέθους του συνόλου εκπαίδευσης για διάφορες τιμές των παραμέτρων. Η καμπύλη "full" αντιστοιχεί στην ακρίβεια που έχουμε εάν εκπαιδεύσουμε τον ταξινομητή με όλα τα μηνύματα με τις παραμέτρους που αναφέραμε στον πίνακα 6.6. Παρατηρούμε στο σχήμα 6.1 ότι στην αρχή που το σύνολο εκπαίδευσης είναι μικρό οι ταξινομητές έχουν χαμηλή και ασταθή απόδοση. Ωστόσο έπειτα από 220 ερωτήσεις η ακρίβεια πλησιάζει αυτή της "full" περίπτωσης με μοναδική εξαίρεση το $C=2^0$ το οποίο είναι συνεχώς ασταθές. Μάλιστα για $C=2^4$ και $C=2^6$ η "full" περίπτωση προσεγγίζεται ακόμη και από τις 120 ερωτήσεις. Αυτά μας δείχνουν ότι ο αλγόριθμος simple margin είναι πολύ αποδοτικός όταν επιλέξουμε σωστές παραμέτρους για τον ταξινομητή. Στο σχήμα 6.2 διατηρήσαμε την παράμετρο C σταθερή και μεταβάλαμε το γ. Το $\gamma=2^{-7}$ για την πρώτη καμπύλη που είναι ίδια με αυτή του σχήματος 6.1. Και εδώ παρατηρούμε αστάθεια στην αρχή ωστόσο μετά από 220 ερωτήσεις κατορθώνουμε να προσεγγίσουμε τη "full" περίπτωση. Είναι φανερό ότι μεγάλη τιμή στην παράμετρο γ μειώνει την ακρίβεια ενώ για $\gamma=0.001$ κατορθώνουμε να πετύχουμε υψηλότερη ακρίβεια από τη "full" περίπτωση. Γενικά πετύχαμε το πολύ με 220 επιλεγμένα από τον ταξινομητή μηνύματα να έχουμε ίδια ακρίβεια με αυτή που δίνει ένα σύνολο εκπαίδευσης αποτελούμενο από 2604 μηνύματα μειώνοντας έτσι σημαντικά τον κόπο για την ανάθεση της κατηγορίας στα μηνύματα του συνόλου εκπαίδευσης.

Στη συνέχεια παραθέτουμε τα αποτελέσματα για τη περίπτωση που χρησιμοποιούμε μόνο το θέμα κάθε μηνύματος όπου και πάλι μία φορά κρατάμε σταθερό το C και μία το γ.



Σχήμα 6.3: Simple margin μόνο με το θέμα για διάφορες τιμές του C στη συλλογή Ling-Spam.



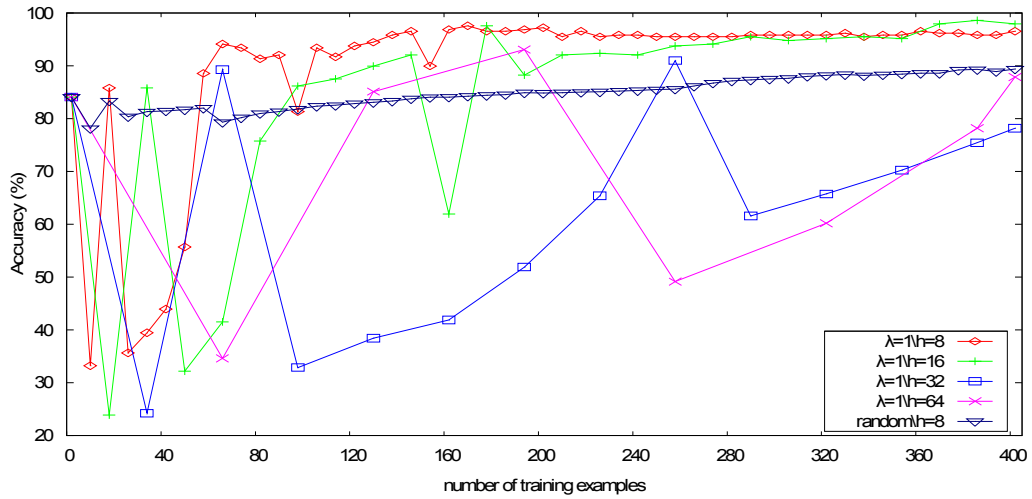
Σχήμα 6.4: Simple margin μόνο με το θέμα για διάφορες τιμές του γ στη συλλογή Ling-Spam.

Από τις δύο παραπάνω γραφικές παραστάσεις είναι φανερό ότι για να προσεγγίσουμε τη "full" περίπτωση απαιτούνται περισσότερα μηνύματα απ' όταν κάναμε εκπαίδευση με ολόκληρα τα μηνύματα. Αυτό είναι λογικό καθώς μόνο με το θέμα η πληροφορία κάθε μηνύματος μειώνεται σημαντικά και άρα ο ταξινομητής απαιτεί περισσότερα για να μάθει να διαχωρίζει τις κατηγορίες. Στο σχήμα 6.3 παρατηρούμε αστάθεια στην αρχή αλλά με μόνη εξαίρεση το $C=2^0$ μετά από 800 ερωτήσεις η ακρίβεια είναι πολύ κοντά σε αυτή της "full" περίπτωσης και αρχίζει να σταθεροποιείται γεγονός που δείχνει ότι δεν κερδίζουμε από την προσθήκη νέων μηνυμάτων στο σύνολο εκπαίδευσης. Η μόνη καμπύλη που βελτιώνεται είναι η $C=2^9$ και γι' αυτή αξίζει η προσθήκη επιπλέον μηνυμάτων. Μάλιστα για $C=2^3$ μετά από 700 ερωτήσεις ξεπερνούμε την ακρίβεια του

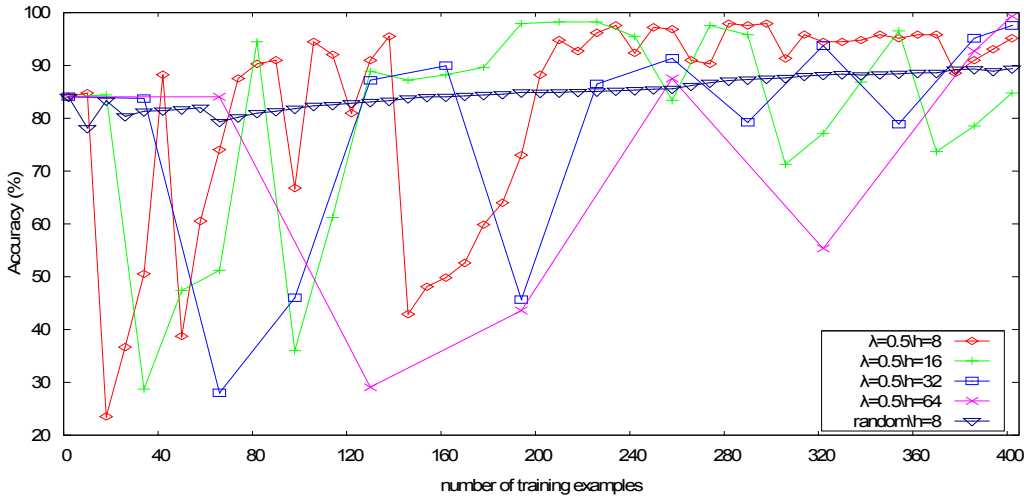
"full". Στο σχήμα 6.4 κρατήσαμε σταθερό το $C=2^3$ και μεταβάλλαμε το γ. Στην πρώτη καμπύλη $\gamma=2^{-4}$ και είναι ίδια με αυτή του σχήματος 6.3. Είναι φανερό πως η ακρίβεια είναι υψηλή όταν το γ λαμβάνει μεγάλες τιμές. Μάλιστα για $\gamma=2^{-4}$ και $\gamma=0.5$ στις 700 ερωτήσεις η πρώτη ξεπερνά και η δεύτερη προσεγγίζει τη "full" περίπτωση. Γενικά θα λέγαμε ότι με τις σωστές παραμέτρους μπορούμε χωρίς να θυσιάσουμε τίποτα σε ακρίβεια με τον αλγόριθμο simple margin να μειώσουμε σημαντικά το σύνολο εκπαίδευσης.

6.4.4 Αξιολόγηση αλγορίθμου για ομάδες ενεργά επιλεγμένων μηνυμάτων

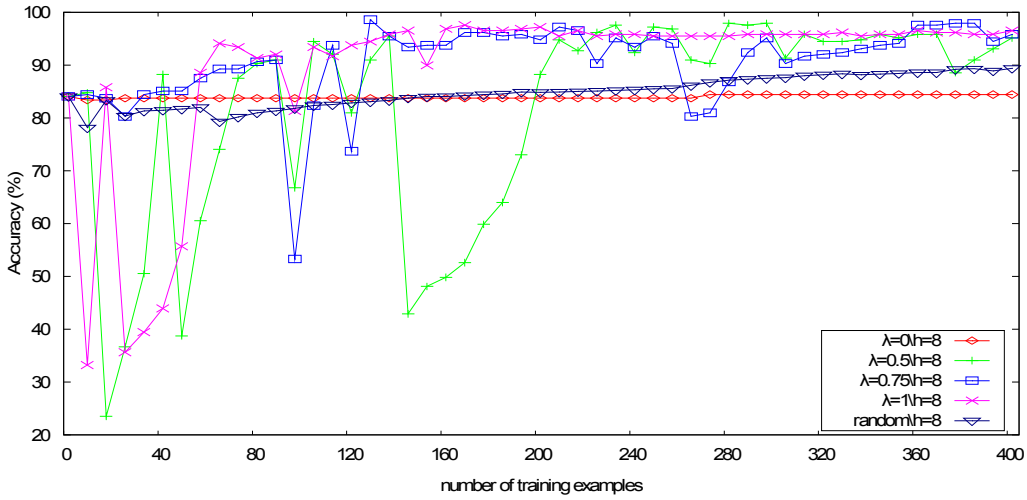
Στο σημείο αυτό παρουσιάζουμε τα πειραματικά αποτελέσματα για τη μέθοδο ενεργητικής μάθησης που επιλέγει μία ομάδα (batch) μηνυμάτων σε κάθε βήμα και παρουσιάστηκε στην ενότητα 4.3.3. Στόχος μας είναι να μελετήσουμε τόσο την επίδραση του μεγέθους της ομάδας h όσο και της παραμέτρου λ που καθορίζει την έμφαση που δίνουμε στις γωνίες και την απόσταση. Η προεπεξεργασία και η αξιολόγηση στη συλλογή Ling-Spam έγινε με τον ίδιο τρόπο που περιγράψαμε για τη μέθοδο simple margin (βλέπε §6.4.3) μόνο που εδώ σε κάθε βήμα επιλέγουμε περισσότερα από ένα μηνύματα για προσθήκη στο σύνολο εκπαίδευσης. Για να φανεί η αξία της μεθόδου τη συγκρίνουμε με τη τυχαία (random) επιλογή νέων μηνυμάτων σε κάθε βήμα που σημαίνει απλά ότι σε κάθε βήμα επιλέγουμε στην τύχη h μηνύματα. Το πείραμα με την τυχαία επιλογή έγινε δέκα φορές ώστε το αποτέλεσμα να είναι έγκυρο και η ακρίβεια που παρουσιάζεται είναι ο μέσος όρος των δέκα επαναλήψεων. Παρακάτω παρατίθενται τα αποτελέσματα για την περίπτωση που χρησιμοποιούμε ολόκληρα τα μηνύματα.



Σχήμα 6.5: Επίδραση του μεγέθους της ομάδας για λ=1 στη συλλογή Ling-Spam.



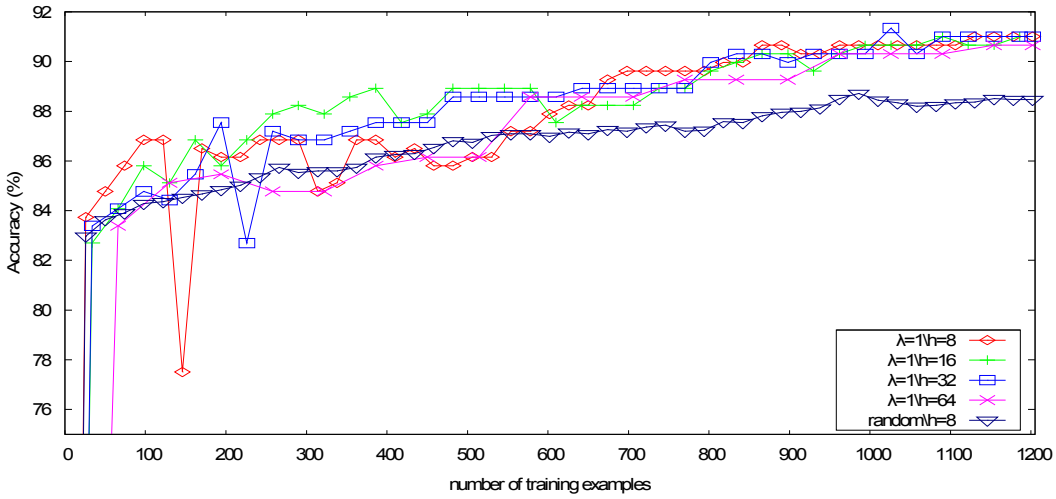
Σχήμα 6.6: Επίδραση του μεγέθους της ομάδας για λ=0.5 στη συλλογή Ling-Spam.



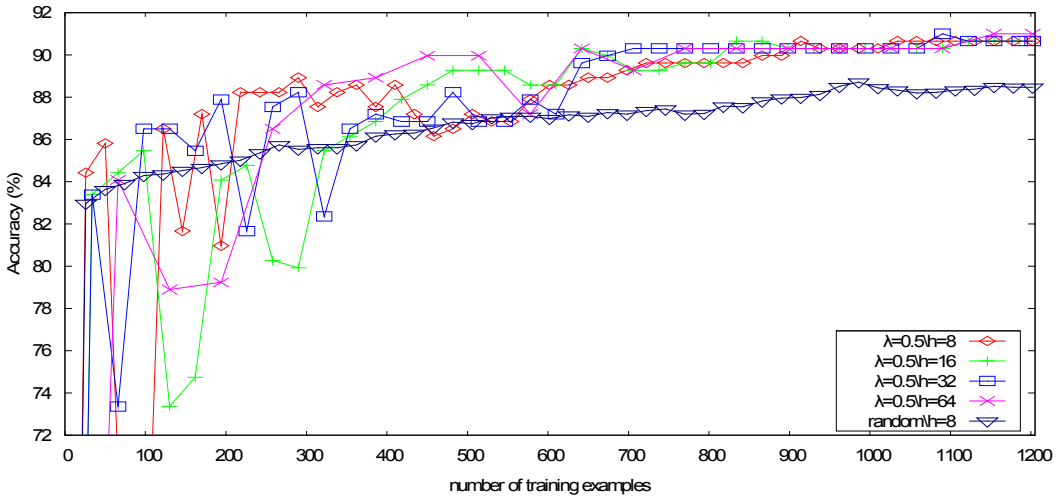
Σχήμα 6.7: Επίδραση της παραμέτρου λ για h=8 στη συλλογή Ling-Spam.

Στις παραπάνω γραφικές παραστάσεις οι παράμετροι του SVM ταξινομητή είναι $C=2^{10}$ και $\gamma=2^{-7}$ για όλες τις περιπτώσεις. Στο σχήμα 6.5 μελετούμε τη περίπτωση όπου η επιλογή της νέας ομάδας μηνυμάτων γίνεται με κριτήριο μόνο την απόσταση. Παρατηρούμε πως για $h=32$ και $h=64$ η ακρίβεια είναι χαμηλότερη από την τυχαία επιλογή που σημαίνει ότι η ακρίβεια μεγάλων ομάδων δεν είναι ικανοποιητική. Αυτό συμβαίνει γιατί όσο περισσότερα μηνύματα προσθέτουμε σε ένα βήμα η αρχή ότι το version space πρέπει να μειώνεται στο μισό για κάθε μήνυμα παραβιάζεται όλο και πιο πολύ. Αντίθετα η ακρίβεια για τα μικρότερα h είναι υψηλότερη από αυτή της τυχαίας επιλογής και μάλιστα σταθεροποιείται μετά από 240 ερωτήσεις. Στο σχήμα 6.6 παρουσιάζεται η περίπτωση που δίνουμε ίση βαρύτητα στο κριτήριο της απόστασης και των γωνιών καθώς $\lambda=0.5$.

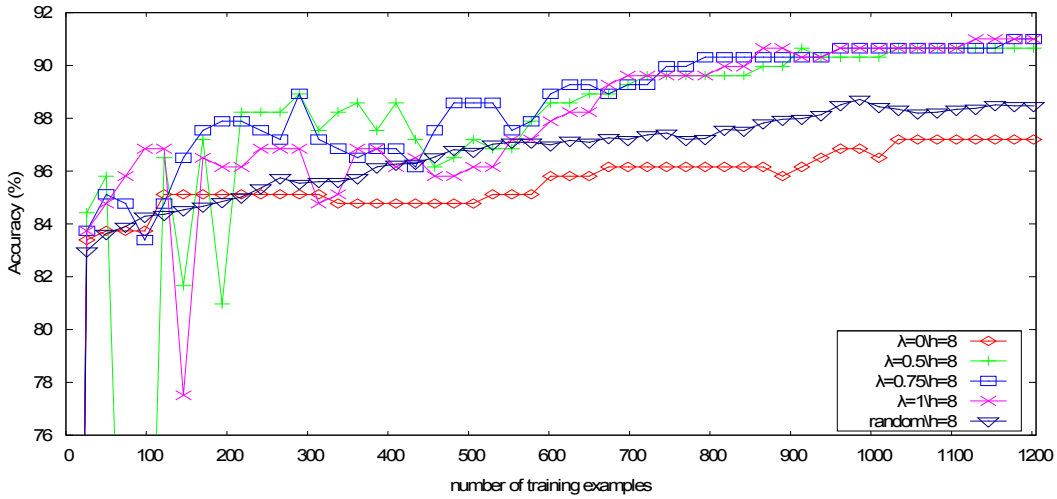
Παρατηρούμε πως όλες οι καμπύλες εμφανίζουν αστάθεια ακόμη και όταν έχουμε πολλά μηνύματα στο σύνολο εκπαίδευσης. Μόνο για h=8 μετά τις 200 ερωτήσεις η ακρίβεια είναι σταθερά μεγαλύτερη από την τυχαία επιλογή ενώ για τα άλλα μεγέθη ομάδας πότε είναι μικρότερη και πότε μεγαλύτερη. Αυτά μας δείχνουν ότι η επιλογή της τιμής του λ δεν είναι καλή. Στο σχήμα 6.7 συγκρίνουμε διάφορες τιμές του λ θέτοντας το h στη μικρότερη τιμή που είδαμε ότι έχει μεγαλύτερη ακρίβεια. Η περίπτωση που ως κριτήριο έχουμε μόνο τις γωνίες (λ=0) είναι χειρότερη από την τυχαία επιλογή και συνεπώς άχρηστη. Για τις άλλες τιμές της παραμέτρου παρατηρούμε μεγαλύτερη σταθερότητα και ακρίβεια όταν δίνουμε έμφαση στην απόσταση δηλαδή μεγάλα λ. Παρακάτω παραθέτουμε τα αντίστοιχα πειράματα όταν χρησιμοποιούμε μόνο το θέμα των μηνυμάτων.



Σχήμα 6.8: Επίδραση του μεγέθους της ομάδας για λ=1 στη συλλογή Ling-Spam.



Σχήμα 6.9: Επίδραση του μεγέθους της ομάδας για λ=0.5 στη συλλογή Ling-Spam.



Σχήμα 6.10: Επίδραση της παραμέτρου λ για h=8 στη συλλογή Ling-Spam.

Στις παραπάνω γραφικές παραστάσεις οι παράμετροι του SVM ταξινομητή είναι $C=2^{3.5}$ και $\gamma=2^{-4}$ για όλες τις περιπτώσεις. Καθώς η εκπαίδευση μόνο με το θέμα απαιτεί περισσότερα μηνύματα όπως αναφέραμε στην ενότητα 6.4.3 αφήσαμε τον ταξινομητή να ρωτήσει την κατηγορία πολλών μηνυμάτων. Στο σχήμα 6.8 όπου ως μόνο κριτήριο επιλογής έχουμε την απόσταση παρατηρούμε ότι μετά από 600 ερωτήσεις για όλα τα *h* έχουμε παρόμοια ακρίβεια και αρκετά υψηλότερη από αυτή της τυχαίας επιλογής. Όταν έχουν γίνει λίγες ερωτήσεις για $h=16$ και $h=32$ έχουμε μεγαλύτερη ακρίβεια ενώ για τις άλλες δύο τιμές συχνά η ακρίβεια είναι μικρότερη και από την τυχαία επιλογή. Ουσιαστικά στην περίπτωση που έχουμε μόνο το θέμα μπορούμε να χρησιμοποιήσουμε λίγο μεγαλύτερες ομάδες χωρίς να χάνουμε ακρίβεια. Στο σχήμα 6.9 όπου δίνουμε ίδια βαρύτητα σε απόσταση και γωνίες και πάλι βλέπουμε ότι μετά τις 700 ερωτήσεις για όλα τα *h* η ακρίβεια είναι παρεμφερής. Ωστόσο για λίγες ερωτήσεις δεν μπορούμε να πούμε με σαφήνεια ποιο μέγεθος ομάδας είναι καλύτερο. Στο σχήμα 6.10 συγκρίνουμε την ακρίβεια για ορισμένες τιμές του λ και παρατηρούμε πως όταν $\lambda=0$ η ακρίβεια είναι χαμηλότερη από αυτή της τυχαίας επιλογής. Για τις υπόλοιπες τιμές της παραμέτρου βλέπουμε πως η χρήση των γωνιών έχει θετική επίδραση στην ακρίβεια καθώς οι καμπύλες για $\lambda=0.5$ και $\lambda=0.75$ έχουν μεγαλύτερη ακρίβεια από την $\lambda=1$ όταν δεν έχουν γίνει πάρα πολλές ερωτήσεις (μεταξύ 200 και 700 ερωτήσεων). Όταν οι ερωτήσεις ξεπερνούν τις 700 και οι τρεις τιμές έχουν εφάμιλλη επίδοση. Άρα στην περίπτωση που έχουμε μόνο το θέμα καλό είναι να χρησιμοποιήσουμε και το κριτήριο των γωνιών.

Εν κατακλείδι μπορούμε να πούμε πως ο αλγόριθμος ενεργητικής μάθησης με ομάδες είναι αποδοτικός καθώς κατορθώνει να έχει υψηλότερη ακρίβεια από την τυχαία

επιλογή και πλησιάζει πολύ αυτή της εκπαίδευσης με ολόκληρη τη συλλογή μετά από έναν αριθμό ερωτήσεων (η ακρίβεια γι' αυτές τις περιπτώσεις φαίνεται στον πίνακα 6.6). Όσον αφορά την επιλογή παραμέτρων το μέγεθος της ομάδας καλό είναι να είναι μικρό καθώς το b=8 είδαμε ότι πετυχαίνει καλή ακρίβεια και στα πειράματα με ολόκληρο το μήνυμα και σε αυτά μόνο με το θέμα αν και μπορεί να μην είναι το καλύτερο πάντα. Για το λ είδαμε ότι στη μία περίπτωση η χρήση μόνο της απόστασης (λ=1) είναι καλύτερη ενώ στην άλλη ο συνδυασμός απόστασης και γωνιών ($0<\lambda<1$) αποδείχθηκε ανώτερος. Ωστόσο η αποκλειστική χρήση γωνιών (λ=0) είναι πάντα η χειρότερη επιλογή καθώς δεν ξεπερνά την τυχαία επιλογή. Άρα η τιμή της καλό είναι να καθορίζεται ανάλογα με την περίπτωση.

6.5 Συμπεράσματα και μελλοντική έρευνα

Από την πειραματική μελέτη που παρουσιάστηκε παραπάνω μπορούν να εξαχθούν χρήσιμα συμπεράσματα καθώς αυτή έγινε με μεγάλη προσοχή ώστε τα αποτελέσματα να αντικατοπτρίζουν την πραγματικότητα. Καταρχήν είδαμε ότι όλες οι μέθοδοι πέτυχαν πολύ καλά αποτελέσματα τόσο με τη χρήση ολόκληρου του μηνύματος όσο και μόνο με το θέμα. Όπως ήταν λογικό η χρήση μόνο του θέματος έχει μικρότερη ακρίβεια αλλά προσφέρει άλλα πλεονεκτήματα όπως είναι ο μειωμένος όγκος πληροφορίας που πρέπει να διαχειριστούμε, τα διανύσματα είναι μικρότερης διάστασης, η εκπαίδευση είναι ταχύτερη όπως και η ταξινόμηση νέων μηνυμάτων.

Η μέθοδος SVM πέτυχε πολύ μεγάλη ακρίβεια και στις δύο συλλογές με μόνη εξαίρεση την περίπτωση hard_ham της συλλογής SpamAssassin όπου όπως αναφέραμε η επιλογή μεγαλύτερου γ τη βελτιώνει σημαντικά. Αυτό επιβεβαιώνεται και από άλλες μελέτες [11,12] που κατατάσσουν την SVM μέθοδο ανάμεσα στις καλύτερες. Μάλιστα υψηλή απόδοση καταφέραμε να πετύχουμε με διάφορους πυρήνες. Όσον αφορά την ενεργητική μάθηση όντως κατορθώνει να περιορίσει σημαντικά το μέγεθος του συνόλου εκπαίδευσης μη χάνοντας σε ακρίβεια όταν συνδυάζεται με τη μέθοδο SVM. Η επιλογή σωστών παραμέτρων για τον SVM αλγόριθμο είναι καταλυτική για την απόδοση αυτών των μεθόδων. Στη περίπτωση που σε κάθε βήμα προσθέτουμε ομάδες μηνυμάτων στο σύνολο εκπαίδευσης μειώνουμε το υπολογιστικό κόστος καθώς αποφεύγουμε συχνές επανεκπαιδεύσεις του SVM σε σχέση με την περίπτωση simple margin όπου για κάθε νέο μήνυμα εκπαιδεύουμε εκ νέου τον ταξινομητή. Τα πειράματά μας έδειξαν ότι και η μέθοδος αυτή έχει μεγάλη ακρίβεια ξεπερνώντας την τυχαία επιλογή και προσεγγίζοντας και

ξεπερνώντας πολλές φορές την εκπαίδευση με όλα τα μηνύματα. Ωστόσο ένα ανοιχτό ζήτημα είναι η επιλογή του λ καθώς είδαμε πως ανάλογα με την περίπτωση συμφέρει να δίνουμε έμφαση στην απόσταση ή στην απόσταση και τις γωνίες.

Τα παραπάνω αποτελέσματα μας δίνουν κίνητρα για περαιτέρω έρευνα στο θέμα της ταξινόμησης κειμένων ηλεκτρονικού ταχυδρομείου. Ενδιαφέρον θα παρουσίαζε η μελέτη και άλλων ταξινομητών και η σύγκρισή τους με τη μέθοδο SVM ώστε να βρούμε ποιος έχει καλύτερη απόδοση. Μάλιστα ο συνδυασμός αυτών με ενεργητική μάθηση καθώς και η διερεύνηση άλλων μεθόδων ενεργητικής μάθησης πιστεύουμε πως θα είχε σημαντική συνεισφορά στην προσπάθεια για μείωση του συνόλου εκπαίδευσης και γενικότερα στην επίλυση του προβλήματος των spam μηνυμάτων.

Κατά τη μελέτη της απόδοσης των ταξινομητών ενδιαφέρον θα ήταν να δώσουμε διαφορετική βαρύτητα στη λανθασμένη ταξινόμηση spam και μη-spam μηνυμάτων και να εκπαιδεύσουμε τους ταξινομητές έτσι ώστε να μειώνονται όσο περισσότερο γίνεται τα λάθη για τα μη-spam. Κάτι τέτοιο θα μας οδηγούσε σε πιο ασφαλείς ταξινομητές με μειωμένη πιθανώς αποτελεσματικότητα στην εύρεση των spam μηνυμάτων. Για την ενεργητική μάθηση στοχεύοντας στη μείωση του κόστους από τις επανεκπαιδεύσεις μπορούμε να καταφύγουμε σε αυξητική μάθηση (incremental learning) ώστε κάθε φορά να μη χρησιμοποιούμε ολόκληρο το σύνολο εκπαίδευσης. Μάλιστα για τη μέθοδο SVM το γεγονός ότι το υπερεπίπεδο περιγράφεται μόνο από τα support vectors βοηθά προς τη κατεύθυνση αυτή.

Τέλος χρήσιμη μπορεί να αποδειχθεί και η χρήση αλγορίθμων για μάθηση με μερική επίβλεψη (partially supervised learning). Στους αλγόριθμους αυτούς παρέχεται ένα σύνολο από spam μηνύματα και ένα σύνολο από μηνύματα που δεν γνωρίζουμε την κατηγορία. Στόχος είναι να εξάγουν από το σύνολο των μηνυμάτων που δεν γνωρίζουμε τη κατηγορία μη-spam και spam μηνύματα και στη συνέχεια να γίνει η εκπαίδευση του ταξινομητή. Με τον τρόπο αυτό μπορούμε να δημιουργήσουμε ένα φίλτρο που θα συλλέγει μηνύματα που δεν γνωρίζουμε την κατηγορία τους καθώς αυτά φθάνουν στο γραμματοκιβώτιο του χρήστη και θα κατασκευάζει τον ταξινομητή σε συνδυασμό με ένα σύνολο spam μηνυμάτων.

Αναφορές

[1] Microsoft Outlook 2000, *Junk E-mail Filter and Adult Content Filter*, http://office.microsoft.com/en-us/assistance/HA010450051033.aspx

[2] MAPILab, *Microsoft Outlook 2003 Spam Filter: Under the hood*, http://www.mapilab.com

[3] S. Spitzer, *Junk Mail in Mozilla*, http://www.mozilla.org/mailnews/arch/spam/junknotes.html

[4] P. Graham, *A Plan for Spam*, http://www.paulgraham.com/spam.html

[5] SpamAssassin, *Tests Performed*, http://spamassassin.org/tests.html

[6] SORBS, http://www.nl.sorbs.net/using.shtml

[7] V. V. Prakash, *Vipul's Razor*, http://razor.sourceforge.net/

[8] F. Sebastiani, *Machine Learning in Automated Text Categorization*, ACM Computing Surveys, Vol. 34, No. 1, March 2002, pp. 1-47

[9] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, G. Paliouras, C. D. Spyropoulos, *An Evaluation of Naïve Bayesian Anti-Spam Filtering*, in Proceedings of the Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning (ECML) Barcelona, Spain, pp. 9-17, 2000

[10] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. D. Spyropoulos, P. Stamatopoulos, *A Memory-Based Approach to Anti-Spam Filtering for Mailing Lists*, Information Retrieval, Vol. 6, No. 1, pp. 49-73, 2003

[11] I. Androutsopoulos, G. Paliouras, E. Michelakis, *Learning to Filter Unsolicited Commercial E-Mail*, NCSR "Demokritos" Technical Report, No. 2004/2, March 2004

[12] L. Zhang, J. Zhu, T. Yao, *An Evaluation of Statistical Spam Filtering Techniques*, ACM Transactions on Asian Language Information Processing, Vol. 3, No. 4, December 2004, pp. 243-269

[13] K. Tretyakov, *Machine Learning Techniques in Spam Filtering*, Data Mining Problem-oriented Seminar, MTAT.03.177, May 2004, pp. 60-79

[14] J. Hovold, *Naïve Bayes Spam Filtering Using Word-Position-Based Attributes*, Second Conference on Email and Anti-Spam (CEAS), 2005

[15] K.-M. Schneider, *Learning to Filter Junk E-Mail from Positive and Unlabeled Examples*, IJCNPL 2004, LNAI 3248, pp. 426-435, 2005

[16] G. P. C. Fung, J. X. Yu, H. Lu, P. S. Yu, *Text Classification without Negative Examples Revisit*, IEEE Transactions on Knowledge and Data Engineering, Vol. 18, No. 1, January 2006

[17] S. Tong, D. Koller, *Support Vector Machine Active Learning with Applications to Text Classification*, Journal of Machine Learning Research, pp. 45-66, 2001

[18] K. Brinker, *Incorporating Diversity in Active Learning with Support Vector Machines*, in Proceedings of the 20th International Conference on Machine Learning (ICML), Washington DC, 2003

[19] C. Domeniconi, D. Gunopoulos, *Incremental Support Vector Machine Construction*, in Proceedings of the International Conference on Data Mining (ICDM), 2001

[20] C. Elkan, *Deriving TF-IDF as a Fisher Kernel*, SPIRE 2005, LNCS 3772, pp. 296-301, 2005

[21] M. F. Porter, *An algorithm for Suffix Stripping*, Program, Vol. 14, No. 3, pp. 130-137, 1980

[22] Y. Yang, J. O. Pedersen, *A Comparative Study on Feature Selection in Text Categorization*, in Proceedings of the 14th International Conference on Machine Learning (ICML), pp. 412-420, 1997

[23] P.-N. Tan, M. Steinbach, V. Kumar, *Introduction to Data Mining*, Addison-Wesley, 1st edition, 2006

[24] C. Burges, *A Tutorial on Support Vector Machines for Pattern Recognition*, Journal of Data Mining and Knowledge Discovery, Vol. 2, No. 2, pp. 121-167, 1998

[25] A. K. McCallum, *Bow: A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering*, http://www.cs.cmu.edu/~mccallum/bow/, 1996

[26] C.-C. Chang, C.-J. Lin, *LIBSVM: a Library for Support Vector Machines*, http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001

[27] Ling-Spam Corpus, http://www.iit.demokritos.gr/skel/i-config/

[28] SpamAssassin Corpus, http://spamassassin.org/publiccorpus/