Ατομική Διπλωματική Εργασία

ΧΡΗΣΗ ΤΩΝ ΜΗΧΑΝΩΝ ΔΙΑΝΥΣΜΑΤΩΝ ΥΠΟΣΤΗΡΙΞΗΣ ΣΤΗΝ ΕΚΤΙΜΗΣΗ ΤΙΜΩΝ ΑΚΙΝΗΤΩΝ

Ειρήνη Γεωργίου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάρτιος 2009

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΧΡΗΣΗ ΤΩΝ ΜΗΧΑΝΩΝ ΔΙΑΝΥΣΜΑΤΩΝ ΥΠΟΣΤΗΡΙΞΗΣ ΣΤΗΝ ΕΚΤΙΜΗΣΗ ΤΙΜΩΝ ΑΚΙΝΗΤΩΝ

Ειρήνη Γεωργίου

Επιβλέπων Καθηγητής

Δρ. Χρίστος Χριστοδούλου

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάρτιος 2009

Ευχαριστίες

Θα ήθελα να ευχαριστήσω περισσότερο από όλους τον επιβλέποντα καθηγητή μου Δρ. Χρίστο Χριστοδούλου, για την βοήθεια, στήριξη και καθοδήγηση που μου παρείχε καθόλη τη διάρκεια της έρευνας και ολοκλήρωσης αυτής της διπλωματικής εργάσιας.

Ακόμη θα ήθελα να ευχαριστήσω την Μάριαμ Αποστολίδου και Ειρήνη Πέτρου, απόφοιτοι του Πανεπιστημίου Κύπρου του Τμήματος Πληροφορικής, οι οποίες ασχολήθηκαν με αυτό το θέμα και ολοκλήρωσαν τη θέση τους τον Ιούνιο του 2007 και 2008 αντίστοιχα. Η μελέτη και κατανόηση της μελέτης καθώς και η χρήση των δεδομένων που είχαν οι ίδιες επεξεργαστεί, φάνηκε ιδιαίτερα χρήσιμη στην ανάπτυξη και ολοκλήρωση της δικής μου ατομικής διπλωματικής εργασίας.

Περίληψη

Η εκτίμηση των τιμών ακίνητης περιουσίας είναι η διαδικασία εκτίμησης μιας τιμής σε ευρώ, αντίστοιχη στην αξία της ακίνητης περιουσίας. Δεδομένων κάποιων συνθηκών της αγοράς η τιμή της ακίνητης περιουσίας αλλάζει, επομένως η διαδικασία αυτή πρέπει να επαναλαμβάνεται περιοδικά έτσι ώστε να αντικατοπτρίζει τις αλλαγές της αγοράς.

Ο παραδοσιακός τρόπος εκτίμησης, που βασίζεται στη σύγκριση κόστους και τιμής πώλησης, στερείται τα αποδεκτά πρότυπα και μια διαδικασία ταυτοποίησης. Επομένως, η παροχή ενός μοντέλου εκτίμησης τιμών ακινήτων μπορεί να βοηθήσει σημαντικά στην εκπλήρωση ενός σημαντικού χάσματος έλλειψης πληροφοριών και να βελτιώσει την αποδοτικότητα στην αγορά ακίνητης περιουσίας.

Τα δεδομένα που θα χρησιμοποιηθούν σε αυτή τη διπλωματική εργασία έχουν παρθεί από το Κτηματολόγιο της Κυπριακής Δημοκρατίας και αφορούν κυρίως διαμερίσματα της Λευκωσίας. Τα δεδομένα αυτά έχουν υποστεί επεξεργασία από την Ειρήνη Πέτρου που ολοκλήρωσε τη θέση της τον Ιούνιο του 2008 [5] και τη Μάριαμ Αποστολίδου τον Ιούνιο του 2007 [4]. Για την εκτίμηση των τιμών ακινήτων θα χρησιμοποιήσω Μηχανές Διανυσμάτων Υποστήριξης ή αλλιώς Support Vector Machines (SVMs) τις οποίες μπορούμε να δούμε ως ένα καινούργιο τρόπο εκπαίδευσης των feedforward νευρωνικών δικτύων.

Τα αποτελέσματα αυτής της διπλωματικής εργασίας ήταν αρκετά ικανοποιητικά και αυτό αποδεικνύει ότι τα Νευρωνικά Δίκτυα μπορούν όντως να βοηθήσουν σε τέτοιου είδους προβλήματα, όπως η εκτίμηση τιμών ακινήτων. Το καλύτερο αποτέλεσμα που λήφθηκε ανάλογα με συγκεκριμένο συνδυασμό παραμέτρων ανέρχεται στο 97.1%.

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΚΕΦΑΛΑΙΟ 1 6

ΕΙΣΑΓΩΓΗ 6

2. ΚΕΦΑΛΑΙΟ 2 8

ΠΕΡΙΓΡΑΦΗ ΠΡΟΒΛΗΜΑΤΟΣ 8
2.1 ΕΚΤΙΜΗΣΗ ΤΙΜΩΝ ΑΚΙΝΗΤΩΝ 8
2.2 ΠΑΡΑΓΟΝΤΕΣ ΠΟΥ ΕΠΗΡΕΑΖΟΥΝ ΤΙΣ ΤΙΜΕΣ ΤΩΝ ΑΚΙΝΗΤΩΝ 8
2.2.1 Κατασκευαστικό κόστος οικίας 9
2.2.2 Αυξομειώσεις στη προσφορά και ζήτηση των ακινήτων 9
2.2.3 Οικογενειακό εσόδημα και ανάγκες στέγασης της οικογένειας 10
2.2.4 Επιτόκια για στεγαστικό δάνειο 10
2.2.5 Δημογραφικές εξελίξεις και ιδιοκατοίκηση 10
2.2.6 Διεθνής αγορά επενδύσεων ακινήτων 11
2.2.7 Η κατοικία ως επένδυση 11
2.2.8 Εξωτερική ζήτηση 11
2.2.9 Συμπέρασμα 11

3. ΚΕΦΑΛΑΙΟ 3 13

ΕΠΙΣΚΟΠΗΣΗ ΒΙΒΛΙΟΓΡΑΦΙΑΣ 13

4. ΚΕΦΑΛΑΙΟ 4 15

ΠΡΟΤΕΙΝΟΜΕΝΗ ΛΥΣΗ 15
4.1 ΕΚΤΙΜΗΣΗ ΤΙΜΩΝ ΑΚΙΝΗΤΩΝ 15
4.2 SUPPORT VECTOR MACHINES 16
4.2.1 Βέλτιστη υπερεπιφάνεια για γραμμικά διαχωρίσιμα πρότυπα 17
4.2.2 Βέλτιστη υπερεπιφάνεια για μη-γραμμικά διαχωρίσιμα πρότυπα 27
4.2.3 Πώς να δημιουργήσεις ένα support vector machine για αναγνώριση προτύπου 32
4.2.4 Θεώρημα του Mercer 36
4.2.5 Βέλτιστος σχεδιασμός ενός support vector machine 37
4.2.6 Παραδείγματα των support vector machines 38

5. ΚΕΦΑΛΑΙΟ 5 42

ΑΝΑΛΥΣΗ ΔΕΔΟΜΕΝΩΝ 42
5.1 ΠΕΡΙΓΡΑΦΗ ΔΕΔΟΜΕΝΩΝ 42
5.2 ΣΥΝΟΛΑ ΔΕΔΟΜΕΝΩΝ 44

6. ΚΕΦΑΛΑΙΟ 6 46

ΥΛΟΠΟΙΗΣΗ 46
6.1 ΠΡΟΣΟΜΟΙΩΤΗΣ SVM CLASSIFIER 46

6.1.1 Μορφή αρχείου εισόδου *46*
6.1.2 Πλαίσιο κατηγοριοποίησης (Classification frame) *47*
6.1.3 Πλαίσιο πρόβλεψης (Prediction frame) *48*
6.1.4 Εισαγωγή Δεδομένων *48*
6.1.5 Kernel Types *50*
6.1.6 SVM Types *51*
6.1.7 Cross Validation *54*
6.1.8 Shrinking *55*
6.1.9 Caching *55*

7. ΚΕΦΑΛΑΙΟ 7 56

ΑΠΟΤΕΛΕΣΜΑΤΑ 56
7.1 ΑΝΑΛΥΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΕΚΤΙΜΗΣΗΣ ΑΚΙΝΗΤΩΝ 56
7.1.1 Διάφορα σύνολα δεδομένων χωρίς χρήση ηλικίας *56*
7.1.2 Αποτελέσματα με δεδομένα τη χρήση ηλικίας *57*
7.1.3 Τύπος του SVM (SVM Type) *58*
7.1.4 Τύπος του Kernel (Kernel Type) *59*
7.2 ΑΠΟΔΟΣΗ ΔΙΚΤΥΟΥ ΜΕ ΤΙΣ ΚΑΛΥΤΕΡΕΣ ΤΙΜΕΣ 63

8. ΚΕΦΑΛΑΙΟ 8 66

ΣΥΖΗΤΗΣΗ 66
8.1 ΣΥΓΚΡΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΜΕ ΠΡΟΗΓΟΥΜΕΝΕΣ ΜΕΛΕΤΕΣ 66
8.2 ΠΡΟΒΛΗΜΑΤΑ ΠΟΥ ΠΑΡΟΥΣΙΑΣΤΗΚΑΝ 67

9. ΚΕΦΑΛΑΙΟ 9 68

ΣΥΜΠΕΡΑΣΜΑΤΑ 68

10. ΒΙΒΛΙΟΓΡΑΦΙΑ 69

11. APPENDIX 70

11.1 ΠΡΟΣΟΜΟΙΩΤΗΣ SVM CLASSIFIER 70
11.1.1 Κατηγοριοποίηση (Classification) *71*
11.1.2 Πρόβλεψη (Prediction) *75*
11.2 ΜΕΤΑΤΡΟΠΗ ΑΠΟ DELIMITED ΣΕ LABELED 78
11.3 SOURCE CODE 0
11.3.1 Labeled.java *0*
11.3.2 Line.java *8*
11.3.3 Type.java *11*
11.3.4 Comparer.java *12*

1. ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

Η ακίνητη περιουσία, μπορεί να εκτιμηθεί μέχρι και το ένα τέταρτο στα περιουσιακά στοιχεία κάποιου κράτους: επομένως, η εκτίμηση των τιμών των ακινήτων είναι μια συχνή και σημαντική δραστηριότητα. Η εκτίμηση των τιμών ακίνητης περιουσίας είναι η διαδικασία εκτίμησης μιας τιμής σε ευρώ, αντίστοιχη στην αξία της ακίνητης περιουσίας, δεδομένων κάποιων συνθηκών της αγοράς. Η τιμή της ακίνητης περιουσίας αλλάζει ανάλογα με τις συνθήκες αγοράς, επομένως η διαδικασία αυτή πρέπει να επαναλαμβάνεται περιοδικά έτσι ώστε να αντικατοπτρίζει τις αλλαγές της αγοράς.

Μια εκτίμηση της τιμής ακίνητης περιουσίας είναι σημαντική στους: πιθανούς αγοραστές ακίνητης περιουσίας, developers, επενδυτές, αποτιμητές και άλλους συμμετέχοντες στην αγορά ακίνητης περιουσίας. Ο παραδοσιακός τρόπος εκτίμησης, που βασίζεται στη σύγκριση κόστους και τιμής πώλησης, στερείται τα αποδεκτά πρότυπα και μια διαδικασία ταυτοποίησης. Επομένως, η παροχή ενός μοντέλου εκτίμησης τιμών ακινήτων μπορεί να βοηθήσει σημαντικά στην εκπλήρωση ενός σημαντικού χάσματος έλλειψης πληροφοριών και να βελτιώσει την αποδοτικότητα στην αγορά ακίνητης περιουσίας.

Τα μοντέλα Νευρωνικών Δικτύων, έχουν αποδειχθεί κατάλληλα για μια αποδοτικότερη και γρηγορότερη διαδικασία εκτίμησης των τιμών ακινήτων. Σε επόμενο κεφάλαιο, θα αναλύσουμε το πώς ορισμένα μοντέλα Νευρωνικών Δικτύων μπορούν να δώσουν καλύτερα και εγκυρότερα αποτέλεσματα έναντι κάποιων άλλων μοντέλων. Σημαντικός παράγοντας που επηρεάζει αισθητά τα αποτέλεσματα της εκτίμησης, είναι τα δεδομένα και το μέγεθος τους.

Σε αυτή τη μελέτη, τα δεδομένα που έχουν χρησιμοποιηθεί είναι τα ίδια δεδομένα που έχουν χρησιμοποιηθεί από τις Μάριαμ Αποστολίδου και Ειρήνη Πέτρου τον Ιούνιο του

2007 [4], και Ιούνιο 2008 [5] αντίστοιχα. Τα δεδομένα αυτά θα προσαρμοστούν σε ένα διαφορετικό μοντέλο ΝΔ που θεωρητικά είναι καλύτερο από αυτά που έχουν χρησιμοποιηθεί παλιά, έτσι ώστε να γίνει μια σύγκριση ανάμεσα στα διαφορετικά μοντέλα και να δούμε ποιό παράγει τα καλύτερα αποτελέσματα. Για την εκτίμηση των τιμών ακινήτων θα χρησιμοποιήσω Μηχανές Διανυσμάτων Υποστήριξης ή αλλιώς Support Vector Machines (SVMs) τις οποίες μπορούμε να δούμε ως ένα καινούργιο τρόπο εκπαίδευσης των feedforward νευρωνικών δικτύων. Στα δεδομένα έχει εφαρμοστεί περαιτέρω επεξεργασία για την καλύτερη λειτουργία των SVMs που θα οδηγήσει σε καλύτερα αποτελέσματα.

2. ΚΕΦΑΛΑΙΟ 2

Περιγραφή Προβλήματος

2.1 Εκτίμηση τιμών ακινήτων.....8
2.1 Παράγοντες που επηρεάζουν τις τιμές των ακινήτων.....8

2.1 Εκτίμηση τιμών ακινήτων

Η εκτίμηση των τιμών ακίνητης περιουσίας είναι η διαδικασία εκτίμησης μιας τιμής σε ευρώ, αντίστοιχη στην αξία της ακίνητης περιουσίας, δεδομένων κάποιων συνθηκών της αγοράς. Η τιμή ενός ακινήτου εξαρτάται όχι μόνο από τα δικά του χαρακτηριστικά, αλλά και από άλλους εξωτερικούς παράγοντες και αλλάζει ανάλογα με τις συνθήκες αγοράς, επομένως η διαδικασία αυτή πρέπει να επαναλαμβάνεται περιοδικά έτσι ώστε να αντικατοπτρίζει τις αλλαγές της αγοράς. Μια σωστή εκτίμηση πρέπει να λαμβάνει υπόψην της όλους αυτούς τους παράγοντες, για να μπορέσει να εξαχθεί μια ακριβέστερη εκτίμηση.

2.2 Παράγοντες που επηρεάζουν τις τιμές των ακινήτων

Πιο κάτω περιγράφονται οι πιό κρίσιμοι παράγοντες που επηρεάζουν την μεταβολή των ακινήτων, όπως έχει περιγραφεί στη μελέτη του Στέλιου Πλατή και Στέλιου Ορφανίδη [8]. Αξίζει να σημειωθεί ότι κάθε ένας από τους παράγοντες αυτούς επηρεάζουν σε διαφορετικό βαθμό τις τιμές των ακινήτων.

2.2.1 Κατασκευαστικό κόστος οικίας

Το κατασκευαστικό κόστος μιας οικίας καθορίζεται από τα αρχιτεκτονικά σχέδια, το κόστος υλικών οικοδομής, τα εργατικά χέρια, τα καύσιμα, τα επιτοκία, τη γη και τους πολεοδομικούς κανονισμούς και περιορισμούς. Το κόστος μια οικίας καθορίζει την ισορροπία μεταξύ του οικιστικού αποθέματος, δηλαδή της προσφοράς και των τιμών κατοικίας από την άλλη, δηλαδή της ζήτησης. Το ψηλό κόστος μιας οικίας είναι καθοριστικό στην αγορά καινούργιας οικίας, οδηγώντας στη μείωση της ζήτησης. Ή ακόμα και στη μείωση της προσφοράς λόγω της μείωσης του κέρδους από τους κατασκευαστές. Επομένως, το κόστος οικίας φαίνεται να ασκεί καθοριστική επίδραση στη διαμόρφωση των τιμών κατοικίας.

2.2.2 Αυξομειώσεις στη προσφορά και ζήτηση των ακινήτων

Η προσφορά και ζήτηση είναι ίσως από τις βασικότερες αρχές στην οικονομία. Η ζήτηση έχει να κάνει με το πόση ζήτηση έχει ένα προϊόν από τους υποψήφιους αγοραστές. Ενώ η προσφορά έχει να κάνει με το πόση από την ποσότητα που ζητείται μπορεί να προσφερθεί. Η αυξημένη ζήτηση ακινήτων, με την προσφορά να παραμένει σταθερή, είναι λογικό να οδηγεί βραχυπρόθεσμα στην αύξηση των τιμών των ακινήτων. Ενώ αντίθετα, η αύξηση στην προσφορά οικιστικών μονάδων οδηγεί σε βραχυπρόθεσμη σταθεροποίηση των τιμών. Επομένως, η τιμή ενός προϊόντος αντικατοπτρίζεται από τη ζήτηση και προσφορά. Στην Κύπρο, και ιδιαίτερα στη Λευκωσία που είναι και η πρωτεύουσα, η προσφορά και ζήτηση παίζουν καθοριστικό ρόλο στον καθορισμό τιμής ακινήτων. Οι πόλεις μεταξύ τους παρουσιάζουν μια διαφορετικότητα όσο αφορά την τιμή πώλησης, για παράδειγμα, από τα παράλια όπου υπάρχει μεγαλύτερη ζήτηση από ξένους.

2.2.3 Οικογενειακό εσόδημα και ανάγκες στέγασης της οικογένειας

Η γενική αύξηση του οικογενειακού εισοδήματος δημιουργεί τις προϋποθέσεις που επιτρέπουν την αύξηση του ποσοστού ιδιοκατοίκησης και άρα αυξημένης ζήτησης κατοικίας. Η αύξηση στο εισόδημα των οικογενειών της Κύπρου αυξάνει και τις απαιτήσεις για καλύτερη κατοικία, επομένως αύξηση της ζήτησης αγοράς κατοικίας.

2.2.4 Επιτόκια για στεγαστικό δάνειο

Οι τιμές των επιτοκίων για αγορά κατοικίας φαίνεται να επηρεάζουν την τιμή ακινήτων. Πιο συγκεκριμένα, η αύξηση των επιτοκίων προκαλεί μείωση στις τιμές των ακινήτων και αντίθετα. Αυτό οφείλεται στο γεγονός ότι όταν τα επιτόκια είναι ψηλά οι αγοραστές είναι λιγότερο πρόθυμοι να κάνουν στεγαστικό δάνειο για αγορά οικίας, ενώ αντίθετα όταν τα επιτόκια είναι χαμηλά οι αγοραστές τείνουν να κάνουν πιο εύκολα δάνειο για αγορά οικίας. Τα επιτόκια καθορίζονται από την Ευρωπαϊκή Τράπεζα ανάλογα με τις οικονομικές αλλαγές που συμβαίνουν στην Κύπρο.

2.2.5 Δημογραφικές εξελίξεις και ιδιοκατοίκηση

Οι δημογραφικές εξελίξεις που συνέβαλαν στην Κύπρο κατά την Τουρκική εισβολή του 1974, οδήγησαν στη μεταβολή και σχηματισμό του πληθυσμού και την ανάγκη για γρήγορη ανάπτυξη διαφόρων περιοχών της Λευκωσίας από τους πρόσφυγες. Αυτό είχε σημαντική επίδραση στη διαμόρφωση των τιμών των ακινήτων. Όπως επίσης η δημιουργία Πανεπιστημίων που αύξησε τη ροή των ξένων φοιτητών στην Κύπρο, και η ανάγκη για ανεύρεση εργασίας από τους μετανάστες είναι συντελεστές που αυξάνουν τη ζήτηση για στέγαση. Επομένως αυξάνεται και η κατασκευαστική διαδικασία, όχι τόσο για σκοπούς ιδοκατοίκησης αλλά περισσότερο ως επένδυση για μελλοντική ενοικίαση.

2.2.6 Διεθνής αγορά επενδύσεων ακινήτων

Οι Κύπριοι επενδυτές λόγω των αυξημένων τιμών των ακινήτων στα εδάφη της Κύπρου, έχουν στρέψει τις βλέψεις τους προς άλλες χώρες πιο συμφέρουσες για επένδυση. Η διεθνής αγορά ακινήτων για επενδύσεις επηρεάζει τις τιμές των ακινήτων στην Κυπριακή αγορά.

2.2.7 Η κατοικία ως επένδυση

Μια κατοικία δε θεωρείται μόνο προϊόν για στέγαση αλλά ταυτόχρονα θεωρείται και ως επενδυτικό προϊόν. Σημαντική αύξηση της ζήτησης ακινήτων για επένδυση παρουσιάστηκε στην Κύπρο το 1999 μετά την πτώση της αξίας των μετοχών στο Χρηματιστήριο Αξιών. Η αυξημένη ζήτηση οδήγησε στην αύξηση των τιμών των ακινήτων όπως επίσης και στην αύξηση της κατασκευαστικής δραστηριότητας για να καλύψει τις ανάγκες της ζήτησης.

2.2.8 Εξωτερική ζήτηση

Η εξωτερική ζήτηση ή αλλιώς, ζήτηση δεύτερης κατοικίας, είναι ακόμη ένας σημαντικός παράγοντας που επηρεάζει τη μεταβολή στις τιμές των ακινήτων. Κυρίως επαγγελματικοί λόγοι ή για σκοπούς αναψυχής οδηγούν στην αγορά δεύτερης οικίας με κύριο σκοπό την επένδυση.

2.2.9 Συμπέρασμα

Οι κύριοι παράγοντες που επηρεάζουν τις τιμές της ακίνητης περιουσίας έχουν σχέση με χρηματοοικονομικές παραμέτρους, όπως δηλαδή το διαθέσιμο εισόδημα, το κόστος και τους όρους χρηματοδότησης στο τραπεζικό σύστημα, την απελευθέρωση των επιτοκίων και την μείωση του κόστους δανεισμού, το κόστος κατασκευής, την αξία της γης, την

επιβολή φόρου προστιθέμενης αξίας στην αγορά κατοικίας/ακινήτου αλλά και την πορεία του χρηματιστηρίου. Επίσης, άλλοι παράγοντες που μπορούν να διαμορφώσουν τη ζήτηση και την προσφορά είναι αυτοί που σχετίζονται με τη δημογραφική σύνθεση, τα πολιτιστικά χαρακτηριστικά, αλλά και τις δυναμικές που δημιουργούν οι πολιτικές εξελίξεις. Η αύξηση της εξωγενούς ζήτησης για οικιστικές μονάδες και κυρίως για εξοχικά, καθώς επίσης και η στροφή των επενδυτών στην κτηματαγορά, λόγω της κακής πορείας του Χρηματιστηρίου, συνέβαλαν επίσης, αυξητικά στη διαμόρφωση των τιμών κατοικίας και των ακινήτων γενικότερα.

3. ΚΕΦΑΛΑΙΟ 3

Επισκόπηση Βιβλιογραφίας

Τα τελευταία περίπου είκοσι χρόνια, έχει παρουσιαστεί μια εξάπλωση στις εμπειρικές μελέτες που αναλύουν τις τιμές στην ακίνητη περιουσία. Η εκτίμηση τιμών ακινήτων συμπίπτει με την ανάπτυξη πληροφοριακών συστημάτων με επακόλουθο διάφορες στατιστικές τεχνικές να ενσωματωθούν στην επεξεργασία των δεδομένων αγοράς. Στις μέρες μας, η χρήση συστημάτων Τεχνητής Νοημοσύνης (AI-Artificial Intelligence) έχει γίνει καλή εναλλακτική λύση που χρησιμοποιείται ευρύτατα και πιο πρακτικά.

Πιο κάτω αναφέρω μερικές από τις εφαρμογές που έχουν γίνει τα τελυταία χρόνια.

a. *1997*. Ο Rossini με δεδομένα για ακίνητα του Λονδίνου, χρησιμοποιώντας στη μελέτη του το μοντέλο Multiple Analysis Regression (MRA) απέδειξε ότι αυτό το μοντέλο πετυγχαίνει καλύτερα αποτελέσματα για κατηγοριοποίηση με απόκλιση λιγότερη απο 5% της πραμγματικής τιμής σε σχέση με τα νευρωνικά δίκτυα.
b. *2001*. Ο Francois Pelissa εκπαίδευσε ένα νευρωνικό δίκτυο χρησιμοποιώντας συνδυασμό Multilayer Perceptron (MLP) και Self Organizing Map (SOM) με δεδομένα από διαφημίσεις από real estates agencies του Λονδίνου. Η προσπάθεια του όμως δεν είχε και τόσο καλά αποτελέσματα με σφάλμα γύρω στο 22%, αλλά έθιξε αρκετά σημεία για περαιτέρω έρευνα τα οποία θα βοηθούσαν σε καλύτερη κατηγοριοποίηση.
c. *2004*. Η προσπάθεια της Goble για κατηγοριοποίηση δεν έφερε ικανοποιητικά απτέλεσματα. Η σωστή κατηγοριοποίηση ανερχόταν μόνο στο 19% των δεδομένων ελέγχου και με καλύτερο αποτέλεσμα 35,2% σωστής κατηγοριοποίηση όταν χρησιμοποίησε δεδομένα από ένα συγκεκριμένο

ταχυδρομικό κώδικα και ένα συγκεκριμένο πεδίο τιμών ακινήτων (200,000 – 299,000).

d. *2007*. Η Μάριαμ Αποστολίδου έχει χρησιμοποιήσει δίκτυο Multilayer Perceptron, με δεδομένα από το Κτηματολόγιο της Κυπριακής Δημοκρατίας που αφορούν διαμερίσματα της Λευκωσίας, πετυγχαίνοντας 85% σωστή κατηγοριοποίηση.

e. *2008*. Η Ειρήνη Πέτρου έχει αναπτύξει και εκπαιδεύσει ένα Radial Basis Function Network (RBF) που είναι μια υποκατηγορία των Support Vector Machines (SVMs), αλλά και ένα υβριδικό Νευρωνικό Δίκτυο που είναι ένας συνδυασμός RBF με Χάρτη Αυτοδιοργάνωσης (Self Organizing Map). Τα αποτελέσματα του υβριδκού δικτύου έχουν δώσει τις αρχικές παραμέτρους (θέση κέντρων, διασπορές) που χρησιμοποιήθηκαν στο RBF δίκτυο. Το RBF δίκτυο με τυχαία κέντρα πέτυχε ποσοστό επιτυχίας 91%, δηλαδή 6% αύξηση στο ποσοστό επιτυχίας που πέτυχε η Μάριαμ Απόστολίδου. Ο συνδυασμός δικτύων RBF με SOM, πέτυχε μεγαλύτερο ποσοστό δίνοντας 96% επιτυχία με απόκλιση λάθους 5%. Σαφώς καλύτερα αποτελέσματα με χρήση RBF δικτύων σε σχέση με τα MLP.

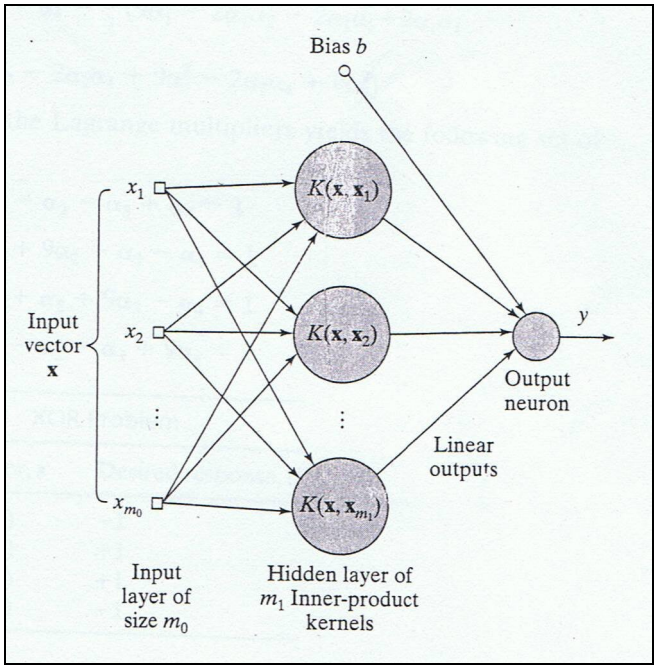
4. ΚΕΦΑΛΑΙΟ 4

Προτεινόμενη Λύση

4.1 Εκτίμηση τιμών ακινήτων.....15
4.2 Support Vector Machines……16

4.1 Εκτίμηση τιμών ακινήτων

Σε αυτή τη διπλωματική εργασία, ο τρόπος με τον οποίο θα αυτοματοποιηθεί η διαδικασία εκτίμησης των τιμών των ακινήτων είναι με τις Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines).



Σχήμα (1) *Αρχιτεκτονική ενός support vector machine*

Η δομή ενός δικτύου SVM φαίνεται στο σχήμα. Στο πιο πάνω SVM δίκτυο δίνεται ως είσοδος ένα διάνυσμα (x_1, x_2,..., x_{m0}) το οποίο αντιστοιχεί σε διάφορα χαρακτηριστικά

από δεδομένα προηγούμενων πωλήσεων σε ακίνητα της Λευκωσίας. Αφού εκπαιδευτεί το δίκτυο με προηγούμενα δεδομένα, θα είναι σε θέση να παράξει στην έξοδο μια τιμή (σε ευρώ) που αντιπροσωπεύει την τιμή που «εκτιμήθηκε» για κάποιο ακίνητο που δεν έχει «ξαναδεί».

4.2 Support Vector Machines

Τις Μηχανές Διανυσμάτων Υποστήριξης ή αλλιώς τα SVMs, που πρωτάρχισαν από τον Vapnik και τους συνεργάτες του το 1963 [6], [7] μπορούμε να τα δούμε ως ένα καινούργιο τρόπο εκπαίδευσης των feedforward νευρωνικών δικτύων των οποίων οι γνωστοί αλγόριθμοι εκπαίδευσης (όπως Radial Basis Functions) απορρέουν ως ειδικές περιπτώσεις. Συγκεκριμένα, ένα Support Vector Machine χρησιμοποιεί μη-γραμμικό μετασχηματισμό του χώρου εισόδου σε ένα πολυδιάστατο χώρο χαρακτηριστικών (feature space), στον οποίο κατασκευάζεται ένα βέλτιστο υπερεπίπεδο (hyperplane) που επιχειρεί να διαχωρίσει δύο κλάσεις. Εφόσον το βέλτιστο υπερεπίπεδο ανταποκρίνεται σε μια μη-γραμμική επιφάνεια στο χώρο εισόδου, έπεται ότι η μέθοδος, άμεσα, κατασκευάζει μια βέλτιστη διαχωριστική επιφάνεια στο χώρο εισόδου που επιχειρεί να διαχωρίσει δύο κλάσεις.

Ένα SVM είναι μια γραμμική μηχανή με μερικές πολύ καλές ιδιότητες. Όπως έχω αναφέρει πιο πάνω, η κύρια ιδέα ενός SVM είναι να δημιουργήσει ένα υπερεπίπεδο (hyperplane) ως την επιφάνεια απόφασης με τέτοιο τρόπο ώστε το περιθώριο ανάμεσα στα θετικά και αρνητικά παραδείγματα να είναι το μέγιστο. Η μηχανή επιτυγχάνει αυτό την επιθυμητή ιδιότητα, ακολουθώντας μια αρχή που έχει τις βάσεις της στην θεωρία στατιστικής μάθησης (statistical learning theory). Πιό συγκεκριμένα, τα SVMs είναι μια κατά προσέγγιση υλοποίηση της μεθόδου structural risk minimization. Αυτή η συνεπαγωγή βασίζεται στο γεγονός ότι το ποσοστό λάθους μιας μηχανής εκμάθησης στα δεδομένα ελέγχου (δηλαδή το generalization error rate) περικλείεται από το άθροισμα

του ποσοστού λάθους εκπαίδευσης και ενός όρου που βασίζεται στο Vapnik-Chervonenkis (VC) dimension. Στην περίπτωση όπου έχουμε διαχωρίσιμα πρότυπα, το SVM παράγει μηδενική τιμή για τον πρώτο όρο και ελαχιστοποιεί το δεύτερο όρο. Επομένως, ένα SVM μπορεί να παράξει καλά αποτελέσματα γενίκευσης σε προβλήματα pattern classification παρά το γεγονός ότι δεν έχει ενσωματωμένη γνώση για το πρόβλημα. Αυτό το χαρακτηριστικό είναι μοναδικό στα support vector machines.

Η κεντρική ιδέα στην κατασκευή ενός αλγορίθμου SVM είναι το εσωτερικό γινόμενο μεταξύ του του "support vector" x_i και του vector x από την είσοδο. Τα support vectors αποτελούνται από ένα μικρό υποσύνολο των δεδομένων εκπαίδευσης. Εξαρτώμενοι από το πώς αυτό το εσωτερικό γινόμενο παράχθηκε, μπορούμε να κατασκευάσουμε διαφορετικές μηχανές εκμάθησης που χαρακτηρίζονται από δικές τους μη-γραμμικές επιφάνειες απόφασης. Πιο συγκεκριμένα, μπορούμε να χρησιμοποιήσουμε τον αλγόριθμο εκμάθησης των support vectors για να δημιουργήσουμε τους ακόλουθους τρεις τύπους μηχανών εκμάθησης (μεταξύ άλλων) : Polynomial learning machines, Radial-basis function networks, Two-layer perceptrons (δηλαδή με ένα hidden layer). Δηλαδή, για κάθε ένα από αυτά τα feedforward δίκτυα μπορούμε να χρησιμοποιήσουμε τον αλγόριθμο εκμάθησης των support vectors για να υλοποιήσουμε μια διαδικασία εκμάθησης χρησιμοποιώντας ένα δεδομένα σύνολο από δεδομένα, αποφασίζοντας αυτόματα τον απαιτούμενο αριθμό των hidden units. Με άλλα λόγια: Ενώ ο αλγόριθμος back-propagation είναι σχεδιασμένος ειδικά για να εκπαιδεύσει ένα multilayer perceptron, ο αλγόριθμος support vector είναι πιο γενικής φύσης επειδή έχει πιο ευρεία εφαρμοσιμότητα.

4.2.1 Βέλτιστη υπερεπιφάνεια για γραμμικά διαχωρίσιμα πρότυπα

Θεωρούμε δεδομένο εκπαίδευσης το $\{(x_i, d_i)\}_{i=1..n}$, όπου το x_i είναι το πρότυπο εισόδου για το i-οστό παράδειγμα και d_i είναι το επιθυμητό αποτέλεσμα (target output). Για αρχή, θεωρούμε ότι το πρότυπο (κλάση) που αναπαρίσταται από το υποσύνολο $d_i = +1$ και το πρότυπο που αναπαρίσταται από το υποσύνολο $d_i = -1$ είναι «γραμμικά διαχωρίσιμα». Η

εξίσωση για την επιφάνεια απόφασης στη μορφή μιας υπερεπιφάνειας που κάνει το διαχωρισμό είναι:

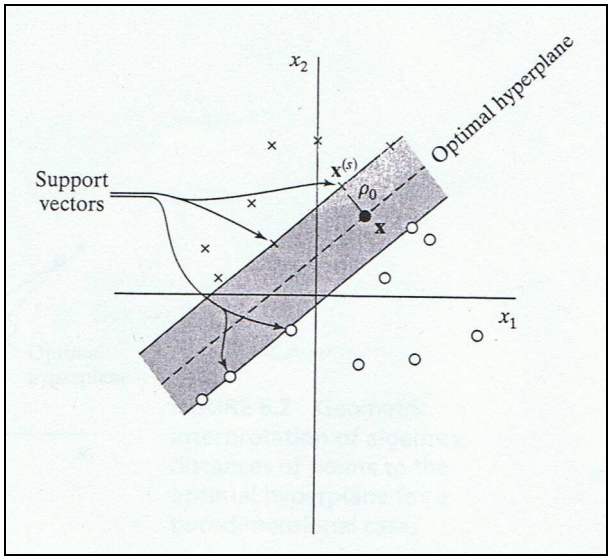
$$(1)\ w^T x + b = 0$$

όπου x είναι ένα διάνυσμα εισόδου (input vector), το w είναι ένα ρυθμιζόμενο διάνυσμα βαρών, και το b είναι το κατώφλι. Μπορούμε επομένως να γράψουμε:

$$(2)\ w^T x_i + b \geq 0 \quad \text{για } d_i = +1,$$
$$w^T x_i + b < 0 \quad \text{για } d_i = -1$$

Η υπόθεση για γραμμικά διαχωρίσιμα πρότυπα γίνεται εδώ για να εξηγήσουμε τη βασική ιδέα πίσω από ένα support vector machine υπό ένα πιο απλό σκηνικό.

Για δεδομένο διάνυσμα βαρών w και κατώφλι b, ο διαχωρισμός μεταξύ της υπερεπιφάνειας που ορίστηκε στην εξίσωση (1) και του κοντινότερου data point (σημείο δεδομένων) λέγεται margin of separation ή αλλιώς περιθώριο διαχωρισμού, που δηλώνεται από το ρ. Ο στόχος ενός support vector machine είναι να βρεί την συγκεκριμένη υπερεπιφάνεια για την οποία το περιθώριο διαχωρισμού ρ να είναι το μέγιστο. Υπό αυτές τις συνθήκες η επιφάνεια διαχωρισμού αναφέρεται ως η *βέλτιστη υπερεπιφάνεια (optimal hyperplane)*. Το σχήμα (2) απεικονίζει τη γεωμετρική κατασκευή μιας βέλτιστης υπερεπιφάνειας για ένα δισδιάστατο χώρο εισόδου.



Σχήμα (2) *Απεικόνιση μιας βέλτιστης υπερεπιφάνειας για γραμμικά διαχωρίσιμα πρότυπα*

Έστω ότι τα w_0 και b_0 δηλώνουν τις βέλτιστες τιμές του διανύσματος βαρών και του κατωφλίου, αντίστοιχα. Παρομοίως, η βέλτιστη υπερεπιφάνεια, που αναπαριστά μια πολυδιάστατη γραμμική επιφάνεια απόφασης στο χώρο είσοδου, ορίζεται ως

$$(3)\ w^T_0 x + b_0 = 0$$

η οποία είναι επαναδιατύπωση της εξίσωσης (1). Η συνάρτηση

$$(4)\ g_x = w^T_0 x + b_0$$

δίνει την αλγεβρική ποσότητα της απόστασης μεταξύ του x και της βέλτιστης υπερεπιφάνειας. Ίσως ο πιό εύκολος τρόπος για να το δούμε αυτό είναι να εκφράσουμε το x ως:

$$x = x_p + r\,(w_0 / \,||w_0||)$$

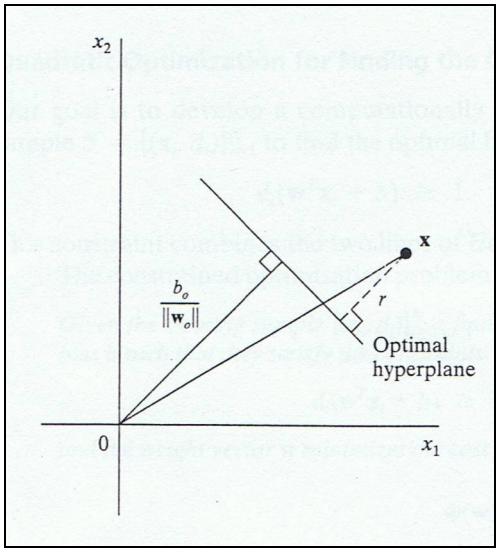
όπου το x_p είναι η κανονική προβολή του x πάνω στη βέλτιστη υπερεπιφάνεια, και το r είναι η επιθυμητή αλγεβρική απόσταση (το r είναι θετικό αν το x βρίσκεται στη θετική πλευρά της βέλτιστης υπερεπιφάνειας και αρνητικό αν το x βρίσκεται στην αρνητική πλευρά). Εφόσον, εξ'ορισμού το $g(x_p) = 0$, τότε ακολούθως

$$(5)\ g(x) = w^T_0\, x + b_0 = r\ ||w_0||$$

ή

$$r = g(x) / \ ||w_0||$$

Πιο συγκεκριμένα, η απόσταση από την αρχή συντεταγμένων (δηλαδή, x = 0) μέχρι τη βέλτιστη υπερεπιφάνεια δίνεται από το $b_0 / \|w_0\|$. Αν $b_0 > 0$, η αρχή συντεταγμένων βρίσκεται στη θετική πλευρά της βέλτιστης υπερεπιφάνειας, και αν $b_0 < 0$, βρίσκεται στην αρνητική πλευρά. Αν $b_0 = 0$, τότε η βέλτιστη υπερεπιφάνεια περνά διαμέσου της αρχής συντεταγμένων. Μια γεωμετρική αναπαράσταση αυτών των αλγεβρικών αποτελεσμάτων βρίσκεται στο σχήμα (3).



Σχήμα (3) *Γεωμετρική αναπαράσταση των αλγεβρικών αποστάσεων μεταξύ των σημείων και της βέλτιστης υπερεπιφάνειας για δισδιάστατο χώρο.*

Το ζητούμενο είναι να βρούμε τις παραμέτρους w_0 και b_0 της βέλτιστης υπερεπιφάνειας, δεδομένου ενός συνόλου εκπαίδευσης $J = \{(x_i, d_i)\}$. Λαμβάνοντας υπ'όψη τα αποτέλεσματα του σχήματος (3), βλέπουμε ότι το ζεύγος (w_0, b_0) πρέπει να τηρεί τους περιορισμούς:

$$(6)\ w^T_0\, x_i + b_0 \geq 1 \qquad \text{για } d_i = +1$$
$$w^T_0\, x_i + b_0 \leq -1 \qquad \text{για } d_i = -1$$

Να σημειώσουμε ότι, αν η εξίσωση (2) ισχύει, δηλαδή τα πρότυπα είναι γραμμικά διαχωρίσιμα, μπορούμε πάντα να ξαναφτιάξουμε τα w_0 και b_0 σε μικρότερη κλίμακα έτσι ώστε να ισχύει και η εξίσωση (6) (αυτό αφήνει την εξίσωση (3) ανεπηρέαστη).

Τα ειδικά σημεία (x_i, d_i) για τα οποία η πρώτη ή δεύτερη γραμμή της εξίσωσης (6) ικανοποιείται με το σήμα ισότητας λέγονται διανύσματα υποστήριξης (support vectors), εξ'ού και το όνομα "support vector machine". Αυτά τα διανύσματα (vectors) παίζουν αξιοπρόσεκτο ρόλο στη λειτουργία αυτού του είδους αλγορίθμων εκμάθησης. Τα support vectors είναι εκείνα τα σημεία τα οποία βρίσκονται κοντινότερα στην επιφάνεια απόφασης και επομένως είναι και τα πιο δύσκολα για να κατηγοριοποιηθούν.

Θεωρούμε ένα support vector $x^{(s)}$ για το οποίο $d^{(s)} = +1$. Τότε εξ'ορισμού, έχουμε

$$(7)\ g(x^{(s)}) = w^T_0 x^{(s)} \pm b_0 = \pm 1 \quad \gamma\iota\alpha \quad d^{(s)} = \pm 1$$

Από την εξίσωση (5) η αλγεβρική απόσταση μεταξύ του support vector $x^{(s)}$ και της βέλτιστης υπερεπιφάνειας είναι

$$(8)\ r = g(x^{(s)}) / \|w_0\|$$

$$= 1 / \|w_0\| \quad \alpha\nu \quad d^{(s)} = +1$$
$$= -1 / \|w_0\| \quad \alpha\nu \quad d^{(s)} = -1$$

όπου το θετικό πρόσημο υποδηλώνει ότι το $x^{(s)}$ βρίσκεται στη θετική πλευρά της βέλτιστης υπερεπιφάνειας και το αρνητικό πρόσημο υποδηλώνει ότι βρίσκεται στην αρνητική πλευρά. Έστω ότι το ρ δηλώνει τη μέγιστη τιμή του περιθωρίου διαχωρισμού (margin of separation) μεταξύ δύο κλάσεων που αποτελούνται από το σύνολο εκπαίδευσης J. Τότε από την εξίσωση (8) προκύπτει ότι

$$(9)\ \rho = 2r = 2 / \|w_0\|$$

Η εξίσωση (9) δηλώνει ότι μεγιστοποιώντας το περιθώριο διαχωρισμού μεταξύ των κλάσεων είναι ισοδύναμο με την ελαχιστοποίηση του Ευκλείδιου μέτρου (Euclidean norm) του διανύσματος βαρών w.

Περιληπτικά, η βέλτιστη υπερεπιφάνεια που ορίζεται από την εξίσωση (3) είναι μοναδική υπό την έννοια ότι το βέλτιστο διάνυσμα βαρών w_0 παρέχει το μέγιστο δυνατό διαχωρισμό μεταξύ των θετικών και αρνητικών παραδειγμάτων. Αυτή η βέλτιστη κατάσταση επιτυγχάνεται με την ελαχιστοποίηση του Ευκλίδειου μέτρου του διανύσματος βαρών w.

4.2.1.1 Τετραγωνική βελτιστοποίηση για εύρεση της βέλτιστης υπερεπιφάνειας

Ο στόχος μας είναι να αναπτύξουμε μια υπολογιστικά αποδοτική διαδικασία χρησιμοποιώντας τα δεδομένα εκπαίδευσης $J = \{(x_i, d_i)\}_{i=1...n}$ για την εύρεση της βέλτιστης επιφάνειας υπό τον περιορισμό

$$(10) \quad d_i(w^T x_i + b) \geq 1 \qquad \text{για } i = 1,2,\ldots,N$$

Αυτός ο περιορισμός συνδυάζει τις δύο γραμμές της εξίσωσης (6) με το w να χρησιμοποιείται στη θέση του w_0. Το περιορισμένο πρόβλημα βελτιστοποίησης (constrained optimization problem) που έχουμε να λύσουμε μπορεί να εκφραστεί ως:

Δεδομένου του δείγματος εκπαίδευσης $\{(x_i, d_i)\}_{i=1...n}$, πρέπει να βρεθούν οι βέλτιστες τιμές του διανύσματος βαρών w και κατωφλίου b έτσι ώστε να ικανοποιείται ο περιορισμός $d_i(w^T x_i + b) \geq 1$ για $i = 1,2,\ldots,N$ και το διάνυσμα βαρών w να ελαχιστοποιεί τη συνάρτηση κόστους $\Phi(w) = \frac{1}{2} w^T w$.

Ο παράγοντας ½ συμπεριλαμβάνεται εδώ για ευκολία στην παρουσίαση. Αυτό το περιορισμένο πρόβλημα βελτιστοποίησης λέγεται βασικό πρόβλημα (primal problem).

Μπορεί να χαρακτηριστεί ως ακολούθως:

- Η συνάρτηση κόστους Φ(w) είναι μια κυρτή (convex) συνάρτηση του w
- Οι περιορισμοί στο w είναι γραμμικοί

Επομένως, μπορούμε να λύσουμε το περιορισμένο πρόβλημα βελτιστοποίησης χρησιμοποιώντας τη μέθοδο των πολλαπλασιαστών Lagrange.

Πρώτα κατασκευάζουμε τη συνάρτηση Lagrange:

$$(11) \quad J(w,b,\alpha) = \tfrac{1}{2}\, w^T w - \Sigma_{i=1..N}\, \alpha_i \,[d_i\,(w^T x_i + b) - 1]$$

Όπου οι βοηθητικές μη-αρνητικές τιμές α_i ονομάζονται *πολλαπλασιαστές Lagrange*. Η λύση στο περιορισμένο πρόβλημα βελτιστοποίησης καθορίζεται από το σημείο J(w,b,α) της Lagrangian συνάρτησης, που πρέπει να ελαχιστοποιηθεί σε σχέση με το w και b, επίσης πρεπέι να μεγιστοποιηθεί σε σχέση με το α. Συνεπώς, διακρίνωντας το J(w,b,α) σε σχέση με τα w και b και θέτωντας τα αποτελέσματα ίσα με το μηδέν, παίρνουμε τις εξής δύο συνθήκες βελτιστοποίησης:

Συνθήκη 1: $\partial\, J(w,b,a) / \partial \mathbf{w} = 0$

Συνθήκη 2: $\partial\, J(w,b,a) / \partial \mathbf{b} = 0$

Η εφαρμογή της πρώτης συνθήκης στη συνάρτηση Lagrange της εξίσωσης (11) παράγει:

$$(12) \quad w = \Sigma_{i=1..N}\, \alpha_i\, d_i\, x_i$$

Η εφαρμογή της δεύτερης συνθήκης στη συνάρτηση Lagrange της εξίσωσης (11) παράγει:

$$(13) \quad \Sigma_{i=1..N}\, \alpha_i\, d_i = 0$$

Αξίζει να σημειώσουμε ότι στο saddle point, για κάθε πολλαπλασιαστή Lagrange α_i, το γινόμενο αυτού του πολλαπλασιαστή με τον αντίστοιχο περιορισμό του εξαφανίζεται, όπως φαίνεται και πιο κάτω:

$$(14) \quad \alpha_i\,[d_i\,(w^T x_i + b) - 1] = 0 \qquad \text{για } i = 1,2,\ldots,N$$

Επομένως, μπορούμε να υποθέσουμε μη-αρνητικές τιμές μόνο στους πολλαπλασιαστές εκείνους οι οποίοι ικανοποιούν ακριβώς την εξίσωση (14). Αυτή η ιδιότητα ακολουθεί τη θεωρία βελτιστοποίησης Kuhn-Tucker.

Όπως αναφέραμε, το βασικό πρόβλημα (primal problem) έχει να κάνει με μια συνάρτηση κόστους και με γραμμικούς περιορισμούς. Δεδομένου ενός τέτοιου περιορισμένου προβλήματος βελτιστοποίησης, είναι πιθανό να κατασκευάσουμε ακόμα ένα πρόβλημα το οποίο λέγεται δυαδικό πρόβλημα (dual problem). Αυτό το δεύτερο πρόβλημα έχει την ίδια βέλτιστη τιμή όπως το primal problem, αλλά με τους πολλαπλασιαστές Lagrange να παρέχουν τη βέλτιστη λύση. Πιο συγκεκριμένα, μπορούμε να αναφέρουμε το ακόλουθο θεώρημα δυισμού (duality theorem):

a. Αν το primal problem έχει βέλτιστη λύση, το dual problem έχει επίσης μια βέλτιστη λύση, και οι αντίστοιχες βέλτιστες τιμές είναι ίσες.

b. Για να είναι το w_0 βέλτιστη primal λύση και το α_0 να είναι βέλτιστη dual λύση, είναι απαραίτητο και αρκετό το w_0 να είναι εφικτό για το primal problem και

$$\Phi(w_0) = J(w_0, b_0, \alpha_0) = \min_w J(w, b_0, \alpha_0)$$

Επεκτείνουμε την εξίσωση (11) ως ακολούθως

$$(15) \quad J(w, b, \alpha) = \tfrac{1}{2} w^T w - \Sigma_{i=1..N} \alpha_i d_i w^T x_i - b \Sigma_{i=1..N} \alpha_i d_i + \Sigma_{i=1..N} \alpha_i$$

Ο τρίτος όρος στα δεξιά της εξίσωσης (15) είναι 0 από τη συνθήκη βελτιστοποίησης της εξίσωσης (13). Ακόμη, από την εξίσωση (12) έχουμε

$$w^T w = \Sigma_{i=1..N} \alpha_i d_i w^T x_i = \Sigma_{i=1..N} \Sigma_{j=1..N} \alpha_i \alpha_j d_i d_j x_i^T x_j$$

Ανάλογα, θέτωντας την αντικειμενική συνάρτηση $J(w, b, \alpha) = Q(\alpha)$, μπορούμε να ανασχηματίσουμε την εξίσωση (15) ως

$$(16) \quad Q(\alpha) = \Sigma_{i=1..N} \alpha_i - \tfrac{1}{2} \Sigma_{i=1..N} \Sigma_{j=1..N} \alpha_i \alpha_j d_i d_j x_i^T x_j$$

όπου τα α_i παίρνουν μη αρνητικές τιμές.

Μπορούμε τώρα να ορίσουμε το δυαδικό πρόβλημα:

Δεδομένου του δείγματος εκπαίδευσης $\{(x_i, d_i)\}_{i=1...N}$, *πρέπει να βρεθούν οι πολλαπλασιαστές Lagrange* $\{\alpha_i\}i=1..N$ *οι οποίοι μεγιστοποιούν την αντικειμενική συνάρτηση* $Q(\alpha) = \Sigma_{i=1..N}\ \alpha_i - ½\ \Sigma_{i=1..N}\Sigma_{j=1..N}\ \alpha_i\ \alpha_j\ d_i\ d_j\ x_i^T x_j$ *υπό τους περιορισμούς*

1) $\Sigma_{i=1..N}\ \alpha_i\ d_i = 0$

2) $\alpha_i \geq 0$ *για* $i = 1,2,..,N$

Το dual problem αποτιμάται πλήρως από τα δεδομένα εκπαίδευσης. Ακόμη, η συνάρτηση Q(α) για να μεγιστοποιηθεί εξαρτάται μόνο από τα πρότυπα εισόδου υπό τη μορφή ενός συνόλου από εσωτερικά γίνομενα, $\{x^T_i, x_j\}_{(i,j)=1...N}$.

Έχωντας αποφασίσει τους βέλτιστους πολλαπλασιαστές Lagrange, που υποδηλώνονται από το $\alpha_{0,i}$, μπορούμε να υπολογίσουμε το βέλτιστο διάνυσμα βαρών w_0 χρησιμοποιώντας τη συνάρτηση (12) και επομένως να γράψουμε

$$(17) \qquad w_0 = \Sigma_{i=1...N}\ \alpha_{0,i}\ d_i\ x_i$$

Για να υπολογίσουμε το βέλτιστο κατώφλι b_0, μπορούμε να χρησιμοποιήσουμε το w_0 που βρήκαμε και να εκμεταλλευτούμε την εξίσωση (7) που αναφέρεται σε ένα θετικό support vector, και επομένως να γράφουμε

$$(18) \qquad b_0 = 1 - w^T_0\ x^{(s)} \quad \text{για } d^{(s)} = 1$$

4.2.1.2 Στατιστικές ιδιότητες της βέλτιστης υπερεπιφάνειας

Για να εφαρμόσουμε τη μέθοδο structural risk minimization χρειάζεται να κατασκευάσουμε ένα σύνολο από διαχωριστικές υπερεπιφάνειες με διαφοροποιημένο το VC dimension τέτοιο ώστε το empirical risk (δηλαδή το λάθος κατηγοριοποίησης της

εκπαίδευσης) και το VC dimension να είναι ελάχιστα την ίδια στιγμή. Σε ένα support vector machine η δομή επιβάλλεται από το σύνολο των διαχωριστικών υπερεπιφανειών με το να περιορίσουμε το Ευκλείδιο μέτρο του διανύσματος των βαρών w. Μπορούμε να ορίσουμε το ακόλουθο θεώρημα:

Έστω D η διάμετρος της μικρότερης σφαίρας που περιέχει όλα τα διανύσματα εισόδου x_1, x_2, ..., x_N. Το σύνολο από βέλτιστες υπερεπιφάνειες που περίγραφεται από την εξίσωση $w^T_0 x + b_0 = 0$ έχει ένα VC dimension h με ανώτατο όριο ως:

$$(19) \quad h \leq \min \{\lceil D2/\rho \rceil, m0\} + 1$$

όπου το σύμβολο οροφής ⌈.⌉ σημαίνει ο μικρότερος ακέραιος αριθμός μεγαλύτερος ή ίσος με τον αριθμό που εγκλείται μέσα, ρ είναι το περιθώριο διαχωρισμού ίσο με $2/||w_0||$, και m_0 είναι η διάσταση του χώρου εισόδου.

Αυτό το θεώρημα μας λέει ότι μπορούμε να ασκήσουμε έλεγχο πάνω στο VC dimension (δηλαδή στην πολυπλοκότητα) της βέλτιστης υπερεπιφάνειας, ανεξάρτητα από την διάσταση m_0 του χώρου εισόδου, με την κατάλληλη επιλογή του περιθωρίου διαχωρισμού ρ.

Υποθέτωντας ότι έχουμε μια ένθετη δομή που περιγράφεται σε σχέση με τις διαχωριστικές υπερεπιφάνειες ως ακουλούθως:

$$(20) \quad S_k = \{w^T x + b: ||w||^2 \leq c_k\}, k = 1,2,..$$

Λόγω του άνω ορίου της VC dimension h που ορίστηκε στην εξίσωση (19), η ένθετη δομή που περιγράφηκε στην εξίσωση (20) μπορεί να αναδιαμορφωθεί σε σχέση με το περιθώριο διαχωρισμού με την μορφή:

$$(21) \quad Sk = \{\lceil r2/\rho 2 \rceil + 1 : \rho 2 \geq \alpha k\}, k = 1,2,..$$

Τα α_k και c_k είναι σταθερές.

Από τη θεωρία στατιστικής μάθησης (statistical learning theory), για να πετύχουμε καλή ικανότητα γενίκευσης, θα έπρεπε να επιλέξουμε τη συγκεκριμένη δομή με τη μικρότερη VC dimension και μικρότερο λάθος εκπαίδευσης, σύμφωνα με την αρχή του structural risk minimization. Από τις εξισώσεις (19) και (21) βλέπουμε ότι αυτή η απαίτηση μπορεί να ικανοποιηθεί χρησιμοποιώντας τη βέλτιστη υπερεπιφάνεια (δηλαδή, τη διαχωριστική υπερεπιφάνεια με το μεγαλύτερο περιθώριο διαχωρισμού ρ). Ισοδύναμα, από την εξίσωση (9) θα μπορούσαμε να χρησιμοποιήσουμε το βέλτιστο διάνυσμα βαρών w_0 που έχει το κατώτατο Ευκλίδειο μέτρο. Επομένως, η επιλογή της βέλτιστης υπερεπιφάνειας για ένα σύνολο από γραμμικά διαχωρίσιμα πρότυπα δεν είναι μόνο διαισθητικά ικανοποιητική αλλά επίσης εκπληρώνει πλήρως την αρχή του structural risk minimization για ένα support vector machine.

4.2.2 Βέλτιστη υπερεπιφάνεια για μη-γραμμικά διαχωρίσιμα πρότυπα

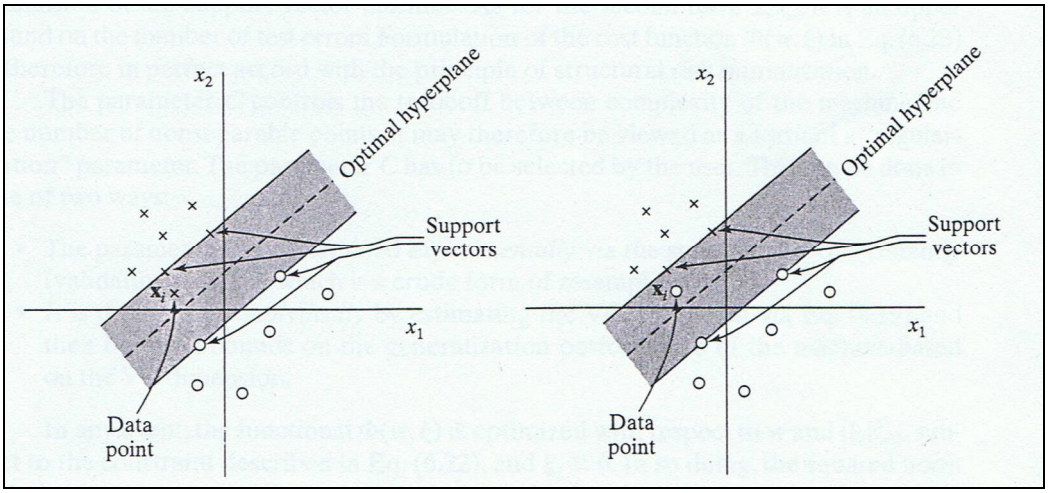
Μέχρι τώρα έχουμε αναφερθεί στα γραμμικά διαχωρίσιμα πρότυπα, τώρα θα ασχοληθούμε με την πιό δύσκολη περίπτωση όπου έχουμε μη γραμμικά διαχωρίσιμα πρότυπα. Δεδομένου ενός τέτοιου συνόλου εκπαίδευσης, είναι δύσκολο να κατασκευάσουμε μια διαχωριστική υπερεπιφάνεια χωρίς να αντιμετωπίσουμε λάθη κατηγοριοποίησης. Παρ'όλα αυτά, θέλουμε να βρούμε μια βέλτιστη υπερεπιφάνεια η οποία ελαχιστοποιεί την πιθάνοτητα λαθών στην κατηγοριοποίηση.

Το περιθώριο διαχωρισμού μεταξύ των κλάσεων λέγεται soft δηλαδή μαλακό, στην περίπτωση όπου ένα δεδομένο (x_i, d_i) παραβιάζει την εξής συνθήκη (εξίσωση (10)) :

$$d_i(w^T x_i + b) \geq +1 \qquad \text{για } i = 1,2,\ldots,N$$

Αυτή η παραβίαση μπορεί να προκύψει για δύο λόγους:

- Το δεδομένο (x_i, d_i) πέφτει εντός της περιοχής διαχωρισμού αλλά στη σωστή πλευρά της επιφάνειας απόφασης (σχήμα 4.α)
- Το δεδομένο (x_i, d_i) πέφτει στη λάθος πλευρά της επιφάνειας απόφασης (σχήμα 4.β)



Σχήμα (4) *α. Το σημείο xi (που ανήκει στη κλάση β_1) πέφτει εντός της περιοχής διαχωρισμού αλλά στη σωστή πλευρά της επιφάνειας απόφασης. β. Το σημείο x_i (που ανήκει στην κλάση β_2) πέφτει στη λάθος πλευρά της επιφάνειας απόφασης.*

Παρατηρούμε ότι στην πρώτη περίπτωση έχουμε σωστή κατηγοριοποίηση, ενώ στη δεύτερη περίπτωση έχουμε λανθασμένη κατηγοριοποίηση.

Για το χειρισμό των μη γραμμικά διαχωρίσιμων δεδομένων εισάγουμε ένα καινούργιο σύνολο από μη αρνητικές βαθμωτές μεταβλητές $\{\xi_i\}_{i=1..N}$ στον ορισμό της υπερεπιφάνειας διαχωρισμού (δηλαδή επιφάνειας απόφασης) όπως φαίνεται πιο κάτω:

$$(22) \quad di\,(w^T xi + b) \geq 1 - \xi i, \quad \text{για } i = 1,2,\ldots,N$$

Η μεταβλητή ξ_i λέγεται *slack variable*, και μετρά την απόκλιση ενός δεδομένου από την ιδανική κατάσταση στο διαχωρισμό προτύπων. Για $0 \leq \xi_i \leq 1$, το δεδομένο πέφτει εντός της περιοχής διαχωρισμού αλλά στη σωστή πλευρά της επιφάνειας απόφασης, όπως φαίνεται και στο σχήμα 4.α. Για $\xi_i > 1$, πέφτει στη λάθος πλευρά της υπερεπιφάνειας διαχωρισμού, όπως φαίνεται και στο σχήμα 4.β. Τα support vectors είναι εκείνα τα σημεία δεδομένων τα οποία ικανοποιούν ακριβώς την εξίσωση (22), ακόμα και αν $\xi_i > 0$. Σημειώνουμε ότι εάν ένα σημείο με $\xi_i > 0$ δεν συμπεριληφθεί στο σύνολο εκπαίδευσης, τότε η επιφάνεια απόφασης θα αλλάξει. Επομένως, τα support vectors ορίζονται με τον ίδιο ακριβώς τρόπο για τα διαχωρίσιμα και για τα μη γραμμικά διαχωρίσιμα δεδομένα.

Ο στόχος μας είναι να βρούμε μια διαχωριστική υπερεπιφάνεια για την οποία το λάθος της λαθασμένης κατηγοριοποίησης να είναι το ελάχιστο. Μπορούμε να το κάνουμε αυτό με την ελαχιστοποίηση της συνάρτησης:

$$\Phi(\xi) = \Sigma_{i=1..N} I(\xi_i - 1)$$

σε σχέση με το διάνυσμα βαρών w, υπό τους περιορισμούς που περιγράφηκαν στην εξίσωση (22) και τον περιορισμό στο $||w^2||$. Η συνάρτηση I(ξ) είναι μια συνάρτηση δείκτη (*indicator function*) που ορίζεται από το

$$I(\xi) = 0 \quad \text{αν } \xi \leq 0$$
$$= 1 \quad \text{αν } \xi > 0$$

Δυστυχώς, η ελαχιστοποίηση του Φ(ξ) σε σχέση με το w είναι ένα nonconvex πρόβλημα βελτιστοποίησης που είναι NP-complete.

Για να κάνουμε το πρόβλημα βελτιστοποίησης μαθηματικώς υπάκουο, προσεγγίζουμε τη συνάρτηση Φ(ξ) γράφωντας

$$\Phi(\xi) = \Sigma_{i=1..N} \xi_i$$

Επιπλέον, απλοποιούμε τον υπολογισμό διατυπώνοντας τη συνάρτηση που θα ελαχιστοποιηθεί σε σχέση με το διάνυσμα βαρών w ως ακολούθως:

$$(23) \quad \Phi(w, \xi) = \tfrac{1}{2} w^T w + C \Sigma_{i=1..N} \xi_i$$

Όπως και πριν, η ελαχιστοποίηση του πρώτου όρου της εξίσωσης (23) σχετίζεται με την ελαχιστοποίηση της VC dimension του support vector machine. Όσο για τον δεύτερο όρο $\Sigma_i \xi_i$, είναι ένα άνω όριο στο πλήθος των test errors. Επομένως η μορφοποίηση της συνάρτησης κόστους Φ(w, ξ) στη συνάρτηση (23) συμφωνεί απόλυτα με την αρχή του structural risk minimization.

Η παράμετρος C ελέγχει την ανταλλαγή ανάμεσα στην πολυπλοκότητα της μηχανής και στο πλήθος των μη διαχωριζόμενων σημείων, μπορούμε επομένως να τη δούμε ως μια

μορφή παραμέτρου κανονικοποίησης. Η παράμετρος C πρέπει να επιλεγχθεί από το χρήστη. Αυτό μπορεί να γίνει με δύο τρόπους:

- Η παράμετρος C επιλέγεται πειραματικά μέσω του κοινού τρόπου εκπαίδευσης/επαλήθευσης.
- Επιλέγεται αναλυτικά υπολογίζοντας το VC dimension μέσω της εξίσωσης (19) και χρησιμοποιώντας όρια πάνω στην απόδοση γενίκευσης της μηχανής βάση του VC dimension.

Σε οποιαδήποτε περίπτωση, η συνάρτηση Φ(w, ξ) βελτιστοποιείται σε σχέση με το w και το $\{\xi_i\}_{i=1..N}$, υπό τον περιορισμό που περιγράφηκε στην εξίσωση (22), και $\xi_i \geq 0$. Έτσι το τετραγωνικό μέτρο του w χρησιμοποιείται ως μια ποσότητα που θα ελαχιστοποιηθεί από κοινού σε σχέση με τα μη γραμμικά διαχωρίσιμα σημεία παρά ως περιορισμός που επιβάλεται στην ελαχιστοποίηση του πλήθους των μη γραμμικά διαχωρίσιμων σημείων.

Το πρόβλημα βελτιστοποίησης για τα μη γραμμικά διαχωρίσιμα πρότυπα που μόλις δηλώσαμε, περιλαμβάνει το πρόβλημα βελτιστοποίησης για γραμμικά διαχωρίσιμα πρότυπα ως ειδική περίπτωση. Πιο συγκεκριμένα, θέτωντας το $\xi_i = 0$ για όλα τα i στις εξισώσεις (22) και (23) τις απλοποιεί στις αντίστοιχες μορφές για την περίπτωση όπου έχουμε γραμμικά διαχωρίσιμα πρότυπα.

Μπορούμε επίσημα να δηλώσουμε το primal πρόβλημα για την περίπτωση των μη γραμμικά διαχωρίσιμων:

Δεδομένου του συνόλου εκπαίδευσης $\{(x_i, d_i)\}_{i=1..N}$*, πρέπει να βρεθούν οι βέλτιστες τιμές του διανύσματος βαρών w και κατωφλίου b τέτοιες ώστε να ικανοποιούν τον περιορισμό*

$$d_i (w^T x_i + b) \geq 1 - \xi_i \quad \text{για } i = 1,2,..N$$

$$\xi_i \geq 0 \quad \text{για όλα τα } i$$

και τέτοια ώστε το διάνυσμα βαρών w και οι μεταβλητές slack variables ξ_i *να ελαχιστοποιούν τη συνάρτηση κόστους*

$$\Phi(w, \xi) = \tfrac{1}{2} w^T w + C \Sigma_{i=1..N} \xi_i$$

όπου το C είναι μια θετική παράμετρος που καθορίζεται από το χρήστη.

Χρησιμοποιώντας τη μέθοδο των πολλαπλασιαστών Lagrange μπορούμε να διαμορφώσουμε το dual problem για μη γραμμικά διαχωρίσιμα πρότυπα ως:

Δεδομένου του συνόλου εκπαίδευσης $\{(x_i, d_i)\}_{i=1..N}$*, πρέπει να βρεθούν οι πολλαπλασιαστές Lagrange* $\{\alpha_i\}_{i=1..N}$ *οι οποίοι μεγιστοποιούν την αντικειμενική συνάρτηση*

$$Q(a) = \Sigma_{i=1..N} a_i - \frac{1}{2} \Sigma_{i=1..N} \Sigma_{j=1..N} a_i a_j di\ dj\ x^T_i x_j$$

Υπο τους περιορισμούς

1) $\Sigma_{i=1..N} \alpha_i d_i = 0<$

2) $0 \leq \alpha i \leq C$ *για* $I = 1,2,..N$

όπου το C είναι μια θετική παράμετρος που καθορίζεται από το χρήστη.

Σημειώνουμε ότι ούτε τα slack variables ξ_i ούτε οι πολλαπλασιαστές Lagrange εμφανίζονται στο dual problem. Το dual problem για μη γραμμικά διαχωρίσιμα πρότυπα είναι επομένως παρόμοιο με αυτό για διαχωρίσιμα πρότυπα εκτός μιας μικρής αλλά σημαντικής διαφοράς. Η αντικειμενική συνάρτηση Q(α) η οποία πρέπει να μεγιστοποιηθεί είναι η ίδια και στις δύο περιπτώσεις. Η περίπτωση μη γραμμικά διαχωρίσιμων διαφέρει από την περίπτωση διαχωρήσιμων στο ότι ο περιορισμός $\alpha_i \geq 0$ αντικαθιστάτε με τον πιό αυστηρό περιορισμό $0 \leq \alpha_i \leq C$. Εκτός από αυτή την τροποποίηση, η υπόλοιπη διαδικασία για υπολογισμό των βέλτιστων τιμών του διανύσματος βαρών και του κατωφλίου είναι η ίδια για τις δύο περιπτώσεις. Όπως επίσης και τα support vectors ορίζονται με τον ίδιο τρόπο όπως προηγουμένως.

Η βέλτιστη λύση για το διάνυσμα βαρών w δίνεται από το

$$(24) \qquad w_0 = \Sigma_{i=1..Ns} \alpha_{0,i}\ d_i\ x_i$$

όπου το N_s είναι το πλήθος των support vectors.

Για τον καθορισμό των βέλτιστων τιμών του κατωφλίου ακολουθείται μια διαδικασία παρόμοια με αυτή που περιγράφηκε προηγουμένως. Πιο συγκεκριμένα οι συνθήκες Kuhn-Tucker τώρα μπορούν να οριστούν ως

$$(25) \quad \alpha_i [d_i (w^T x_i + b) - 1 + \xi_i] = 0, \quad i = 1,2,..N$$

και

$$(26) \quad \mu_i \xi_i = 0, \quad i = 1,2,..N$$

Η εξίσωση (25) είναι επαναδιατύπωση της εξίσωσης (14) εκτός από την αντικατάσταση του όρου μονάδα (1) με τον όρο (1 - ξ_i). Όσο για την εξίσωση (26), το μ_i είναι πολλαπλασιαστές Lagrange που εισάχθηκαν για να ενδυναμώσουν την μη-αρνητικότητα των μεταβλητών slack variables ξ_i για όλα τα i. Στο σημείο saddle point το παράγωγο της συνάρτηση Lagrange για το primal problem σε σχέση με την μεταβλητή ξ_i είναι μηδέν, η αποτίμηση του οποίου αποδίδεται ως

$$(27) \quad \alpha_i + \mu_i = C$$

Συνδυάζοντας τις εξισώσεις (26) και (27) βλέπουμε ότι

$$(28) \quad \xi_i = 0 \ \text{αν} \ \alpha_i < C$$

Μπορούμε να προσδιορίσουμε το βέλτιστο κατώφλι b_0 παίρνοντας οποιοδήποτε σημείο (x_i, d_i) από το σύνολο εκπαίδευσης για το οποίο ισχύει $0< \alpha_{0,i} < C$ και επομένως $\xi_i = 0$, και χρησιμοποιώντας εκείνο το σημείο στη εξίσωση (25). Από πλευράς αριθμητικής είναι καλύτερα να πάρουμε τη μέση τιμή του b_0 που απορρέει από όλα τα σημεία στο σύνολο εκπαίδευσης.

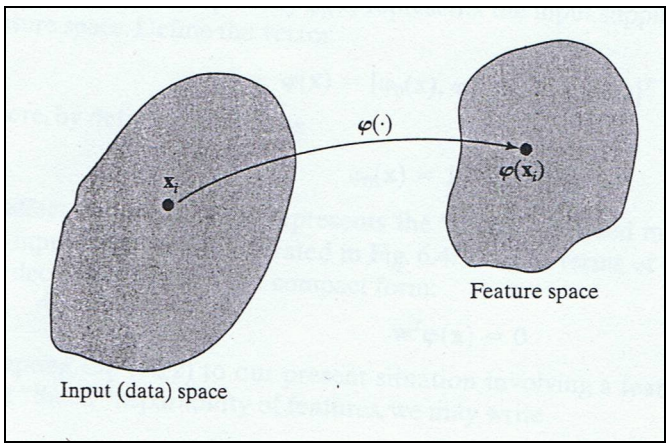
4.2.3 Πώς να δημιουργήσεις ένα support vector machine για αναγνώριση προτύπου

Σε αυτό το σημείο θα περιγράψουμε την κατασκευή ενός support vector machine για αναγνώριση προτύπου.

Γενικά η ιδέα ενός support vector machine βασίζεται σε δύο μαθηματικές λειτουργίες οι οποίες περιγράφονται περιληπτικά εδώ και απεικονίζονται στο σχήμα (5):

1. Μη γραμμική συσχέτιση ενός διανύσματος εισόδου σε ένα χώρο χαρακτηριστικών (feature space) ψηλότερης διάστασης που είναι κρυφός (hidden) από την είσοδο και έξοδο.
2. Δημιουργία μιας βέλτιστης υπερεπιφάνειας για διαχωρισμό των χαρακτηριστικών που βρέθηκαν στο βήμα 1.

Η λογική των πιο πάνω λειτουργιών εξηγείται πιο κάτω:



Σχήμα (5) *Μη γραμμική συσχέτιση φ(.) από το χώρο εισόδου (input space) σε ένα χώρο χαρακτηριστικών (feature space)*

Η πρώτη λειτουργία εκτελείται σύμφωνα με το θεώρημα του Cover για τη γραμμική διαχωρισιμότητα των προτύπων. Θεωρούμε ένα χώρο εισόδου που αποτελείται από μη γραμμικά διαχωρίσιμα πρότυπα. Το θεώρημα του Cover δηλώνει ότι ένας τέτοιος πολυδιάστατος χώρος μπορεί να μετασχηματιστεί σε ένα καινούργιο χώρο χαρακτηριστικών (feature space) όπου τα πρότυπα είναι γραμμικά διαχωρίσιμα με μεγάλη πιθανότητα όταν ικανοποιούνται δύο συνθήκες. Πρώτη, ο μετασχηματισμός είναι μη γραμμικός. Δεύτερο, η διάσταση του χώρου χαρακτηριστικών είναι αρκετά ψηλή. Αυτές οι δύο συνθήκες είναι ενσωματομένες στη λειτουργία 1. Να σημειωθεί όμως ότι το θεώρημα του Cover δεν αναφέρεται στη βελτιστοποίηση της υπερεπιφάνειας διαχωρισμού. Μόνο με τη χρήση μιας βέλτιστης διαχωριστικής υπερεπιφάνειας η VC

dimension ελαχιστοποιείται και επιτυγχάνεται γενίκευση. Και εδώ είναι που μπαίνει η δεύτερη λειτουργία. Πιο συγκεκριμένα, η λειτουργία 2 εκμεταλεύεται την ιδέα όπου κατασκευάζεται μια βέλτιστη διαχωριστική υπερεπιφάνεια σε σχέση με τη θεωρία που εξηγήθηκε στο 4.2.2 για μη γραμμικά διαχωρίσιμα πρότυπα, αλλά με μια βασική διαφορά: Η διαχωριστική υπερεπιφάνεια ορίζεται τώρα ως μια γραμμική συνάρτηση από διανύσματα που εξάγεται από το χώρο χαρακτηριστικών (feature space) παρά από τον αρχικό χώρο εισόδου. Πολύ σημαντικό το γεγονός ότι η κατασκευή αυτής της υπερεπιφάνειας εκτελείται σε σχέση με την αρχή του structural risk minimization που έχει τις ρίζες του στη θεωρία του VC dimension. Η κατασκευή εξαρτάται από τον υπολογισμό τον πυρήνα εσωτερικού γινομένου (inner product kernel).

4.2.3.1 Πυρήνας εσωτερικού γινομένου (inner product kernel)

Έστω ότι το x δηλώνει ένα διάνυσμα που εξάγεται από το χώρο εισόδου, υποθέτωντας ότι είναι της διάστασης m_0. Έστω ότι το $\{\varphi_j(x)\}_{j=1..m1}$ δηλώνει ένα σύνολο από μη γραμμικούς μετασχηματισμούς από το χώρο εισόδου στο χώρο χαρακτηριστικών (m_1 είναι η διάσταση του χώρου χαρακτηριστικών). Υποθέτουμε ότι το $\varphi_j(x)$ ορίζεται ως το priori για όλα τα j. Δεδομένου ενός τέτοιου συνόλου από μη γραμμικούς σχηματισμούς, μπορούμε να ορίσουμε την υπερεπιφάνεια ως την επιφάνεια απόφασης ως ακολούθως:

$$(29) \quad \Sigma_{j=1..m1} w_j \varphi_j(x) + b = 0$$

Όπου το $\{w_j\}_{j=1..m1}$ δηλώνει ένα σύνολο από γραμμικά βάρη που συνδέουν το χώρο χαρακτηριστικών με το χώρο εξόδου, και b είναι το κατώφλι. Μπορούμε να το απλοποιήσουμε γράφωντας:

$$(30) \quad \Sigma_{j=0..m1} w_j \varphi_j(x) = 0$$

Όπου υποθέτουμε ότι το $\varphi_0(x) = 1$ για όλα τα x, έτσι ώστε το w_0 να δηλώνει το κατώφλι b. Η εξίσωση (30) ορίζει την επιφάνεια απόφασης που υπολογίζεται στο feature space σε σχέση με τα γραμμικά βάρη της μηχανής. Η ποσότητα $\varphi_j(x)$ απεικονίζει την είσοδο που προμηθεύτηκε στο βάρος w_j διαμέσου του feature space. Ορίζουμε το διάνυσμα

$$(31) \quad \varphi(x) = [\varphi_0(x), \varphi_1(x), .., \varphi_{m1}(x)]^T$$

όπου εξ'ορισμού έχουμε

$$(32) \quad \varphi_0(x) = 1 \quad \text{για όλα τα } x$$

Στην ουσία, το διάνυσμα φ(x) αναπαριστά την «εικόνα» που προκλήθηκε στο χώρο χαρακτηριστικών λόγω του διανύσματος εισόδου x, όπως απεικονίζεται στο σχήμα (4). Επομένως, όσο αφορά αυτή την εικόνα μπορούμε να ορίσουμε την επιφάνεια απόφασης στη μορφή:

$$(33) \quad w^T \varphi(x) = 0$$

Προσαρμόζωντας την εξίσωση (12) στην παρούσα κατάσταση που περιλαμβάνει ένα feature space στο οποίο ψάχνουμε γραμμικό διαχωρισμό των χαρακτηριστικών, μπορούμε να γράψουμε:

$$(34) \quad w = \Sigma_{i=1..N} \alpha_i d_i \varphi(x_i)$$

όπου το διάνυσμα χαρακτηριστικών $\varphi(x_i)$ αντιστοιχεί στο πρότυπο εισόδου x_i για το i–οστο παράδειγμα. Επομένως, αντικαθιστώντας την εξίσωση (34) στην (33), μπορούμε να ορίσουμε την επιφάνεια απόφασης που υπολογίζεται στο χώρο χαρακτηριστικών ως:

$$(35) \quad \Sigma_{i=1..N} \alpha_i d_i \varphi^T(x_i) \varphi(x) = 0$$

Ο όρος $\varphi^T(x_i)\varphi(x)$ αναπαριστά το εσωτερικό γινόμενο δύο διανυσμάτων που προκλήθηκαν στο χώρο χαρακτηριστικών από το διάνυσμα εισόδου x και το πρότυπο εισόδου x_i που αναφέρεται στο i-οστό παράδειγμα. Μπορούμε επομένως να εισάξουμε τον πυρήνα εσωτερικού γινομένου που υποδηλώνεται από το $K(x, x_i)$ και ορίζεται από

$$(36) \quad K(x, x_i) = \varphi^T(x) \varphi(x_i)$$
$$= \Sigma_{j=0..m1} \varphi_j(x) \varphi(x_i) \quad \text{για} \quad i = 1,2,..N$$

Από αυτό τον ορίσμο βλέπουμε ότι ο πυρήνας εσωτερικού γινομένου είναι μια συμμετρική συνάρτηση των παραμέτρων του, όπως φαίνεται και από:

$$(37) \quad K(x, x_i) = K(x_i, x) \quad \text{για όλα τα } i$$

Το πιο σημαντικό, είναι ότι μπορούμε να χρησιμοποιήσουμε τον πυρήνα εσωτερικού γινομένου $K(x, x_i)$ για να κατασκευάσουμε τη βέλτιστη υπερεπιφάνεια στο χώρο

χαρακτηριστικών χωρίς να χρειαστεί να μελετήσουμε άμεσα τον ίδιο το χώρο χαρακτηριστικών. Αυτό εύκολα μπορούμε να το δούμε χρησιμοποώντας της εξίσωση (36) στην (35), όπου η βέλτιστη υπερεπιφάνεια ορίζεται από

$$(38) \quad \Sigma_{i=1..N} \alpha_i d_i K(x, x_i) = 0$$

4.2.4 Θεώρημα του Mercer

Η επέκταση της εξίσωσης (36) για τον πυρήνα του εσωτερικού γινομένου $K(x, x_i)$ είναι μια ειδική περίπτωση του θεωρήματος του Mercer. Αυτό το θεώρημα μπορούμε να το ορίσουμε ως:

Έστω ότι το $K(x, x_i)$ είναι ένας συνεχές συμμετρικός πυρήνας που ορίζεται στο κλειστό διάστημα $a <= x <= b$ και παρόμοια για το x'. Ο πυρήνας $K(x, x')$ μπορεί να επεκταθεί

$$(39) \quad K(x, x') = \Sigma_{i=1..oo} \lambda_i \varphi_i(x) \varphi_i(x')$$

Με θετικούς συντελεστές, $\lambda_i > 0$ για όλα τα i. Για να είναι αυτή η επέκταση έγκυρη και για να συγκλίνει απόλυτα και σταθερά, είναι απαραίτητο και επαρκές να ισχύει η συνθήκη

$$\int_a^b \int_a^b K(x, x')\psi(x)\psi(x')dxdx' \geq 0$$

Για όλα τα ψ(.) για τα οποία

$$\int_a^b \psi^2(x)dx < \infty$$

Οι συναρτήσεις $\varphi_i(x)$ λέγονται ιδιοσυναρτήσεις ή χαρακτηριστικές συναρτήσεις (eigenfunctions) της επέκτασης και οι μεταβλητές λ_i λέγονται χαρακτηριστικές ρίζες (eigenvalues). Το γεγονός ότι όλες οι χαρακτηριστικές ρίζες είναι θετικές σημαίνει ότι και ο πυρήνας $K(x, x')$ είναι σίγουρα θετικός.

Υπο το θεώρημα του Mercer, μπορούμε να κάνουμε κάποιες παρατηρήσεις:

- Για $\lambda_i \neq 1$, η i-οστή εικόνα $\sqrt{\lambda_i}\, \varphi_i(x)$ που παρακινείται στο χώρο χαρακτηριστικών από το χώρο εισόδου x είναι μια χαρακτηριστική συνάρτηση της επέκτασης.

- Στη θεωρία, η διάσταση του χώρου χαρακτηριστικών (δηλαδή το πλήθος των eigenvalues/eigenfunctions) μπορεί να είναι απείρως μεγάλη.

Το μόνο που μας λέει το θεώρημα του Mercer είναι αν ένας υποψήφιος πυρήνας είναι ή δεν είναι ένας πυρήνας εσωτερικού γινομένου σε κάποιο χώρο και επομένως παραδεκτός για χρήση σε ένα support vector machine. Όμως, δεν λέει τίποτα για το πώς να κατασκευάσουμε τις συναρτήσεις $\varphi_i(x)$, αυτό πρέπει να το κάνουμε μόνοι μας.

Από την εξίσωση (23) βλέπουμε ότι ένα support vector machine περιλαμβάνει ένα είδος ομαλοποίησης με μια υπονοούμενη αίσθηση.

4.2.5 Βέλτιστος σχεδιασμός ενός support vector machine

Η επέκταση του πυρήνα εσωτερικού γινομένου $K(x, x_i)$ στην εξίσωση (36) μας επιτρέπει να κατασκευάσουμε μια επιφάνεια απόφασης που είναι μη γραμμική στο χώρο εισόδου, αλλά η εικόνα του στο χώρο χαρακτηριστικών είναι γραμμική. Με αυτή την επέκταση, μπορούμε να ορίσουμε την dual μορφή για την περιορισμένη βελτιστοποίηση ενός support vector machine ως ακολούθως:

Δεδομένου ενός δείγματος εκπαίδευσης $\{(x_i, d_i)\}_{i=1..N}$*, πρέπει να βρουμε τους πολλαπλασιαστές Lagrange* $\{\alpha_i\}_{i=1..N}$ *που μεγιστοποιούν την αντικειμενική συνάρτηση*

$$(40) \quad Q(a) = \Sigma_{i=1..N} a_i - \tfrac{1}{2} \Sigma_{i=1..N} \Sigma_{j=1..N} a_i a_j d_i d_j K(x_i, x_j)$$

Υπό τους περιορισμούς:

1. $\Sigma_{i=1..N} \alpha_i d_i = 0$
2. $0 \le \alpha_i \le C \quad για\ i = 1,2,..N$

Όπου το C είναι μια θετική παράμετρος που καθορίζεται από το χρήστη.

Να σημειώσουμε ότι ο περιορισμός (1) προκύπτει από τη βελτιστοποίηση της Lagrangian $Q(\alpha)$ σε σχέση με το κατώφλι $b = w_0$ για $\varphi_0(x) = 1$. Το dual πρόβλημα που μόλις αναφέραμε είναι της ίδιας μορφής οπώς εκείνο για τα μη γραμμικά διαχωρίσιμα

πρότυπα, εκτός από το γεγονός ότι το εσωτερικό γινόμενο $x_i^T x_j$ που χρησιμοποιείται έχει αντικατασταθεί με τον πυρήνα εσωτερικού γινομένου $K(x_i, x_j)$. Μπορούμε να δούμε το $K(x_i, x_j)$ ως το ij-οστο στοιχείο ενός συμμετρικού N-επί-N πίνακα K, όπως φαίνεται από

$$(41) \quad K = \{K(x_i, x_j)\}_{(i,j) = 1 .. N}$$

Έχωντας βρει τις βέλτιστες τιμές των πολλαπλασιαστών Lagrange, που υποδηλώνονται από το $\alpha_{0,i}$, μπορούμε να αποφασίσουμε τις αντίστοιχες βέλτιστες τιμές του γραμμικού διανύσματος βαρών, w_0, που συνδέει το χώρο χαρακτηριστικών με το χώρο εξόδου εφαρμόζοντας τον τύπο της εξίσωσης (17) στην καινούργια κατάσταση. Πιο συγκεκριμένα, αναγνωρίζοντας ότι η εικόνα $\varphi(x_i)$ παίζει το ρόλο της εισόδου στο διάνυσμα βαρών w, μπορούμε να ορίσουμε το w_0 ως

$$(42) \quad w_0 = \Sigma_{i=1..N} \alpha_{o,i} d_i \varphi(x_i)$$

όπου το $\varphi(x_i)$ είναι η εικόνα που προκαλείται στο χώρο χαρακτηριστικών λόγω του x_i. Σημειώνουμε ότι η πρώτη συνιστώσα του w_0 αναπαριστά το βέλτιστο κατώφλι b_0.

4.2.6 Παραδείγματα των support vector machines

Η ανάγκη για τον πυρήνα $K(x, x_i)$ είναι για να ικανοποιήσουμε το θεώρημα του Mercer. Σε αυτή την απαίτηση υπάρχει μια ελευθερία για το πώς θα επιλεγχθεί. Στον πίνακα πιο κάτω συνοψίζουμε τους πυρήνες εσωτερικού γινομένου για τρεις κοινούς τύπους support vector machines: Polynomial, Learning Machine, Radial-Basis Function Network, και Two-Layer Perceptron. Αξίζει να σημειώσουμε τα πιο κάτω σημεία:

1. Οι πυρήνες εσωτερικού γινομένου για polynomial και radial-basis function types των support vector machines πάντα ικανοποιούν το θεώρημα του Mercer. Σε αντίθεση, οι πυρήνες εσωτερικού γινομένου για το two-layer perceptron type του support vector machine είναι κατά κάποιο τρόπο περιορισμένο, όπως φαίνεται στην τελευταία γραμμή του πίνακα. Αυτό είναι μια χειροπιαστή απόδειξη στο γεγονός ότι η απόφαση για το αν ένας πυρήνας ικανοποιεί ή όχι το θεώρημα του Mercer μπορεί όντως να είναι δύσκολο ζήτημα.

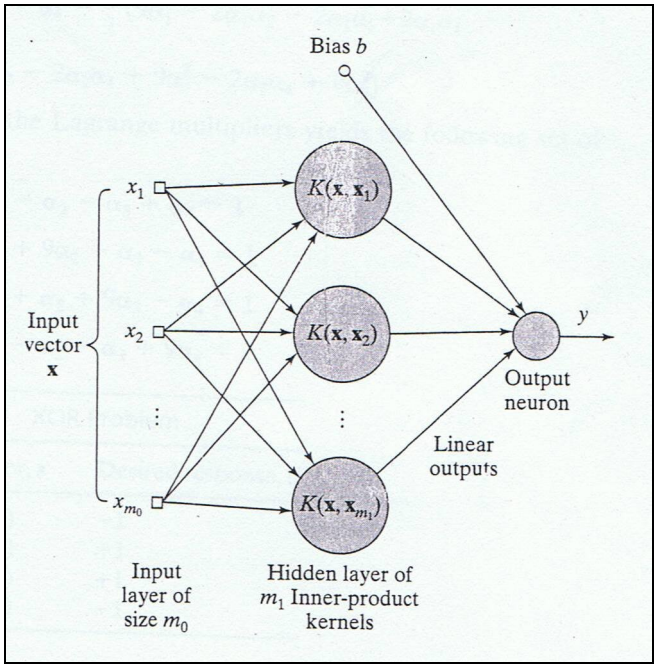
2. Και για τα τρία είδη μηχανών, η διάσταση του χώρου χαρακτηριστικών καθορίζεται από το πλήθος των support vectors που εξάγχθηκαν από τα δεδομένα εκπαίδευσης μέσω της λύσης στο περιορισμένο πρόβλημα βελτιστοποίησης.
3. Η θεμελιώδης θεωρία ενός support vector machine αποφεύγει την ανάγκη για heuristics που συχνά χρησιμοποιούνται στο σχεδιασμό των συνηθισμένων radial-basis function networks και των multilayer perceptrons:
    a. Στον radial basis function τυπό ενός support vector machine, το πλήθος των radial basis functions και τα κέντρα τους καθορίζονται αυτόματα από το πλήθος των support vector machines και των τιμών τους, αντίστοιχα.
    b. Στον two-layer perceptron τύπο ενός support vector machine, το πλήθος των κρυφών νευρώνων και τα διανύσματα των βαρών τους καθορίζονται αυτόματα από το πλήθος των support vectors και των τιμών τους, αντίστοιχα.

Περίληψη των πυρήνων εσωτερικού γινομένου

Τύπος του support vector machine	**Πυρήνας εσωτερικού γινομένου $K(x, x_i)$, i=1,2,..,N**	**Σχόλια**
Polynomial Learning Machine	$(x^T x_i + 1)^p$	Η δύναμη p ορίζεται ως priori από το χρήστη
Radial-Basis function network	$\exp((- 1/(2\sigma^2)) * \|x - x_i\|^2)$	Το πλάτος σ^2, κοινό σε όλους τους πυρήνες, καθορίζεται ως priori από το χρήστη
Two-layer perceptron	$\tanch(\beta_0 x^T x_i + \beta_i)$	Το θεώρημα του Mercer ικανοποιείται μόνο για μερικές τιμές β_0 και β_i

Πίνακας (1) Πίνακας όπου περιληπτικά αναφέρονται οι πυρήνες εσωτερικού γινομένου

Στο σχήμα (6) παρουσιάζεται η αρχιτεκτονική ενός support vector machine.



Σχήμα (6) *Αρχιτεκτονική ενός support vector machine*

Άσχετα από το πώς θα υλοποιηθεί ένα support vector machine, διαφέρει από την συνηθισμένη προσέγγιση στο σχεδιασμό ενός multilayer perceptron με θεμελιώδη διαφορά. Στη συνηθισμένη προσέγγιση, η πολυπλοκότητα του μοντέλου ελέγχεται κρατώντας το πλήθος των χαρακτηριστικών (δηλαδή των κρυφών νευρώνων) μικρό. Από την άλλη, το support vector machine προσφέρει μια λύση στο σχεδιασμό μιας μηχανής εκμάθησης ελέγχοντας την πολυπλοκότητα του μοντέλου ανεξάρτητα από τη διάσταση, όπως συνοψίζεται εδώ

- **Conceptual problem.** Η διάσταση του χώρου χαρακτηριστικών (hidden) είναι σκόπιμα φτιαγμένη πολύ μεγάλη για να επιτρέψει την κατασκευή μιας επιφάνειας απόφασης στη μορφή υπερεπιφάνειας σε αυτό τον χώρο. Για καλή απόδοση στη γενίκευση, η πολυπλοκότητα του μοντέλου ελέγχεται με την επιβολή κάποιων περιορισμών στην κατασκευή της διαχωριστικής υπερεπιφάνειας, που έχει ως αποτέλεσμα την εξαγωγή ενός κλάσματος (fraction) των δεδομένων εκπαίδευσης ως support vectors.

- **Computational Problem.** Αυτό το υπολογιστικό πρόβλημα αποφεύγεται χρησιμοποιώντας την ιδέα ενός πυρήνα εσωτερικού γινομένου (που ορίζεται σύμφωνα με το θεώρημα του Mercer) και επιλύοντας τη διπλή μορφή του περιορισμένου προβλήματος βελτιστοποίησης που διαμορφώθηκε στο χώρο εισόδου (δεδομένων).

5. ΚΕΦΑΛΑΙΟ 5

Ανάλυση Δεδομένων

5.1 Περιγραφή Δεδομένων.....41
5.2 Σύνολα Δεδομένων.....43

Τα δεδομένα που χρησιμοποιήθηκαν έχουν παρθεί από προηγούμενη διπλωματική εργασία η οποία έγινε από τη Μάριαμ Αποστολίδου [4]. Για τη σωστή διεξαγωγή των αποτελεσμάτων τα δεδομένα έπρεπε να υποστούν κάποια διαδικασία επεξεργασίας. Η επεξεργασία αυτή έχει γίνει από την Ειρήνη Πέτρου [5] στη διπλωματική της εργασία το 2008.

5.1 Περιγραφή Δεδομένων

Τα δεδομένα που δόθηκαν από το Κτηματολόγιο περιλάμβαναν διάφορα χαρακτηριστικά των ακινήτων εκ των οποίων επιλέχθηκαν τα καταλληλότερα και αφαιρέθηκαν εκείνα τα οποία θα οδήγουσαν σε λιγότερο αξιόπιστα αποτελέσματα. Παρατηρήθηκε ότι οι τιμές κυμαίνονταν μεταξύ των 11,000 ΛΚ μέχρι 290,000 ΛΚ. Τα ακίνητα των οποίων η τιμή υπερβαίνει τις 120,000 ΛΚ είναι ελάχιστα, έτσι είναι αναμενόμενο ότι η μηχανή ίσως να μη δίνει μια ακριβής πρόβλεψη για τέτοιου είδους ακίνητα. Αντίθετα για τα ακίνητα που κυμαίνονται από τις 30,000 Λ.Κ. μέχρι τις 100,000 ΛΚ αναμένεται ότι το δίκτυο θα είναι σε θέση να δώσει πιό ακριβή πρόβλεψη.

Τα χαρακτηριστικά που χρησιμοποιήθηκαν από τα αρχικά δεδομένα είναι τα εξής: Town/Village, Quarter, Accepted Price, Sale Acceptance Date, Enclosed, Main Sbp. Kind (Property Type), Covered, Uncovered και General Features. Οι διάφορες τιμές που μπορούν να πάρουν τα διάφορα αυτά χαρακτηριστικά φαίνονται στον πίνακα (2).

Στήλη/Χαρακτηριστικό	**Πιθανές τιμές**
Town/Village	Geri Dimos Aglantzias Dimos Egkomis Dimos Idaliou Dimos Latsion Dimos Lakatamias Dimos Lefkosias Dimos Strovolou
Quarter	Agia Paraskevi, Agioi Kostantinos Kai Eleni Agioi Omologites Agios Andreas Agios Antonios Agios Dimitrios Agios Eleftherios Agios Georgios Agios Vasilios Aglantzia Ap. Varnavas & Ag. Makarios Arxagellos/Anthoupoli Arxaggelos Mixail Egkomi Geri Kaimakli P Lakatamia - Agios Nikolaos Panagia Panagias Evaggelistrias Stavros Tripiotis Xriseleousa
Accepted Price	Τιμή σε Λ.Κ.
Sale Acceptance Date	Ημερομηνία πώλησης ακινήτου
Enclosed (m^2)	Αριθμός τετραγωνικών μέτρων εσωτερικού χώρου

Main Sbp. Kind (Property Type)	Office Apartment Shop Two floor apartment Shop with middle floor Group of Offices
Covered (m2)	Αριθμός τετραγωνικών μέτρων καλυμμένου χώρου
Uncovered (m2)	Αριθμός τετραγωνικών μέτρων ακάλυπτου χώρου
Special Features	Akaliptos Xoros Apothiki Apoxoritirio Avli Boithitikos Xoros Domatio Eksoterika Ktiria Fournos Grafio Κλπ.

Πίνακας (2) Πίνακας με τα χαρακτηριστικά που επιλέχθηκαν για να αντιπροσωπεύουν ένα ακίνητομ και οι τιμές που μπορεί να πάρει το κάθε χαρακτηριστικό

5.2 Σύνολα Δεδομένων

Στα πιο πάνω χαρακτηριστικά έχουν εφαρμοσθεί δύο μεθόδοι κανονικοποίησης από την Ειρήνη Πέτρου [5]. Οι δύο αυτές μεθόδοι είναι η Min-Max Method και 1-C Method. Μέτα από την κατάλληλη επεξεργασία έχουν δημιουργηθεί τα πιο κάτω σύνολα.

1) **All**, το οποίο περιείχε όλες τις στήλες του αρχικού συνόλου από την Matlab.
2) **With Apartment, Square Meters and Date Only**, όπου από το αρχικό σύνολο χρησιμοποιήθηκαν μόνο οι στήλες Price, Diamerisma, Date, Enclosed Ext., Covered Ext.και Uncovered Ext.
3) **With Square Meters (S.M.) and Date Only**, όπου από το αρχικό σύνολο χρησιμοποιήθηκαν μόνο οι στήλες Price, Date, Enclosed Ext., Covered Ext.και Uncovered Ext.

4) **All with Economic Factors**, το οποίο χρησιμοποιεί όλα τα χαρακτηριστικά από το αρχικό σύνολο και επιπρόσθετα τον CPI και MLFR.

5) **All without Apartment Type (AT) and Quarter**, όπου από το αρχικό σύνολο χρησιμοποιήθηκαν μόνο οι στήλες Price, Town, Date, Enclosed Ext., Covered Ext. και Uncovered Ext. και Combined.

Στην συνέχεια αφού υπολογίσθηκε η εικονική ηλικία δημιουργήθηκαν ακόμη 2 σύνολα δεδομένων:

6) **With Age (min-max) and Square Meters Only**, όπου από το αρχικό σύνολο χρησιμοποιήθηκαν μόνο οι στήλες Price, Enclosed Ext., Covered Ext.και Uncovered Ext και η επιπλέον στήλη Age.

7) **With Age (1-C method) and Square Meters Only**, όπου από το αρχικό σύνολο χρησιμοποιήθηκαν μόνο οι στήλες Price, Enclosed Ext., Covered Ext.και Uncovered Ext και οι 7 επιπλέον στήλες για την ηλικία 2000, 2001, 2002, 2003,2004, 2005, 2006, 2007.

6. ΚΕΦΑΛΑΙΟ 6

Υλοποίηση

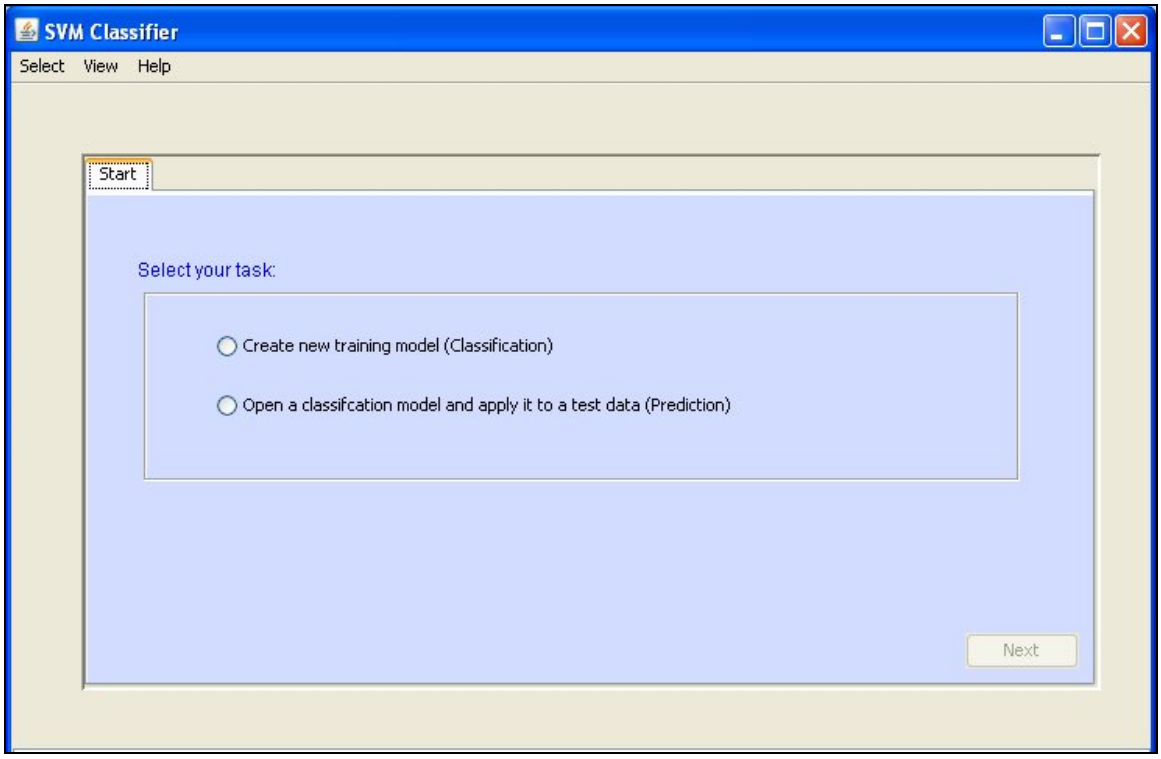
6.1 Προσομοιωτής SVM Classifier……45

6.1 Προσομοιωτής SVM Classifier

Για την διπλωματική αυτή και για το πιο πάνω πρόβλημα έχει χρησιμοποιηθεί ένας προσομοιωτής (simulator). Ο προσομοιωτής που χρησιμοποιήθηκε λέγεται SVM Classifer και έχει εκδοθεί το Δεκέμβριο του 2006, από τους Mehdi Pirooznia και Yooping Deng του University of Southern Mississippi και βρίσκεται στο http://mcbc.usm.edu/svm/. Ο SVM Classifer είναι ένα java interface για κατηγοριοποίηση δεδομένων με χρήση μηχανών διανυσμάτων υποστήριξης ή αλλιώς support vector machines. Είναι μια γραφική εφαρμογή που μπορεί να χειριστεί πολύ καλά μεγάλο πλήθος δεδομένων και επιτρέπει στους χρήστες να εκτελέσουν εκπαίδευση, κατηγοριοποίηση και πρόβλεψη με χρήση των SVMs.

6.1.1 Μορφή αρχείου εισόδου

Η εφαρμογή αυτή χωρίζεται σε δύο πλαίσια (frames), το πλαίσιο κατηγοριοποίησης και το πλαίσιο πρόβλεψης όπως φαίνεται στο σχήμα (7).



Σχήμα (7) *Πλαίσια (frames), το πλαίσιο κατηγοριοποίησης και το πλαίσιο πρόβλεψης.*

6.1.2 Πλαίσιο κατηγοριοποίησης (Classification frame)

Στο πλαίσιο κατηγοριοποίησης ο χρήστης θα δημιουργήσει ένα μοντέλο model.svm από τα δεδομένα εκπαίδευσης: για κατηγοριοποίηση (C-SVC, nu-SVC), για regression (epsilon-SVR, nu-SVR) ή για distribution estimation. Στο δικό μας πρόβλημα χρησιμοποιούμε το πρώτο, δηλαδή για κατηγοριοποίηση.

Σε αυτό το πλαίσιο, ο χρήστης μπορεί να εισάξει τα δεδομένα εκπαίδευσης στην εφαρμογή, να επιλέξει το μονοπάτι στο οποίο θα αποθηκευθεί το αρχείο μοντέλου (model file), να επιλέξει το κατάλληλο SVM και kernel type και να δημιουργήσει ένα μοντέλο για τα συγκεκριμένα δεδομένα εκπαίδευσης. Το model file μπορεί να χρησιμοποιηθεί πιο μετά για σκοπούς πρόβλεψης. Υπάρχει επίσης επιλογή για cross validation. Η τεχνική του cross validation (CV) χρησιμοποιείται για να υπολογιστεί η ακρίβεια του κάθε συνδιασμού παραμέτρων στην καθορισμένη εμβέλεια (range) και

βοηθά να αποφασίσουμε ποιές είναι οι καλύτερες παραμέτροι στο πρόβλημα κατηγοριοποίησης.

6.1.3 Πλαίσιο πρόβλεψης (Prediction frame)

Σε αυτό το πλαίσιο θα δοθεί το μοντέλο (model file) που βρέθηκε στην κατηγοριοποίηση το οποίο θα εφαρμοστεί στα δεδομένα ελέγχου για να προβλεφθεί η κατηγοριοποίηση άγνωστων δεδομένων.

6.1.4 Εισαγωγή Δεδομένων

Και στα δύο πλαίσια η μορφή του αρχείου δεδομένων μπορεί να είναι "Labeled" ή "Delimited". Τα αρχεία με τα δεδομένα είναι της μορφής Delimited και για τη μετατροπή τους έχω δημιουργήσει ένα μικρό πρόγραμμα σε Java το οποίο δέχεται ως είσοδο το αρχείο και το μετατρέπει στη μορφή "Labeled".

6.1.4.1 Labeled μορφή του αρχείου δεδομένων.

Η μορφή των αρχείων training και testing είναι:

<label> <index1>:<value1> <index2>:<value2>…

Το <label> είναι η επιθυμητή τιμή των δεδομένων εκπαίδευσης. Για κατηγοριοποίηση, μπορεί να είναι μια ακέραια τιμή η οποία αναπαριστά μια κλάση (υποστηρίζεται και multi-class κατηγοριοποίηση). Το <index> είναι μια ακέραια τιμή που αρχίζει από 1, <value> είναι ένας πραγματικός (real) αριθμός.

```
1       1:0.05286       2:0.0127586     3:0.07929512    4:0.044         5:0.0172
1       1:0.0611        2:0.05724       3:0.056751      4:0.0218        5:0.0088
1       1:0.0580939     2:0.008         3:0.08292199    4:0.06639       5:0.024
1       1:0.04453656    2:0.01960784    3:0.1090688     4:0.04857       5:0.0235
-1      1:0.0570191     2:0.03508756    3:0.0307649     4:0.01311       5:0.0131
-1      1:0.0355556     2:0.0431031     3:0.084         4:0.04          5:0.0086
-1      1:0.0588247     2:0.0375        3:0.100834454   4:0.0294        5:0.0166
-1      1:0.06779915    2:0.04098377    3:0.076678      4:0.02925       5:0.0122
-1      1:0.0673023     2:0.018264      3:0.07630769    4:0.04308       5:0.0136
-1      1:0.07692769    2:0.0084384     3:0.0895897     4:0.02          5:0.0084
-1      1:0.036194      2:0.038         3:0.11301       4:0.02783       5:0.0042
```

Σχήμα (8) *Labeled μορφή του αρχείου δεδομένων.*

6.1.4.2 Delimited μορφή του αρχείου δεδομένων.

Η μορφή των αρχείων training και testing είναι:

<label><label><label>…<value1> <value2><value3>…

Το <label> είναι η επιθυμητή τιμή των δεδομένων εκπαίδευσης. Για κατηγοριοποίηση, μπορεί να είναι μια ακέραια τιμή η οποία αναπαριστά μια κλάση (υποστηρίζεται και multi-class κατηγοριοποίηση). Το <value> είναι ένας πραγματικός (real) αριθμός. Τα labels στο αρχείο επαλήθευσης χρησομοποιούνται μόνο για υπολογισμό της ακρίβειας ή του λάθους. Αν η τιμή τους δεν είναι γνωστή, απλά συμπληρώνεται αυτή η στήλη με ένα αριθμό.

```
1       1       1       1       1       1       2       2       2       2       3
0.862   1.032   0.789   0.885   0.966   1.132   0.0     0.719   0.737   1.107   0.6
1.185   1.477   1.149   0.0     1.166   0.866   1.236   1.355   1.456   1.032   0.6
0.73    0.834   0.888   0.0     0.748   0.887   0.798   0.0     0.0     0.933   0.9
0.803   0.661   0.826   0.788   1.066   0.982   0.745   0.634   0.868   0.88    0.9
0.877   0.0     1.02    0.0     1.033   0.958   1.091   0.0     0.0     0.998   0.8
0.727   0.795   0.857   0.875   0.797   0.855   0.892   0.942   1.085   0.929   0.8
0.84    0.0     1.291   0.972   0.74    0.673   1.043   0.641   0.805   0.919   0.8
0.0     0.0     1.107   0.0     0.903   0.804   0.881   1.368   1.264   0.969   1.0
1.007   0.0     1.108   0.0     1.027   0.758   1.409   0.0     1.197   0.954   0.7
```

Σχήμα (9) *Delimited μορφή του αρχείου δεδομένων*

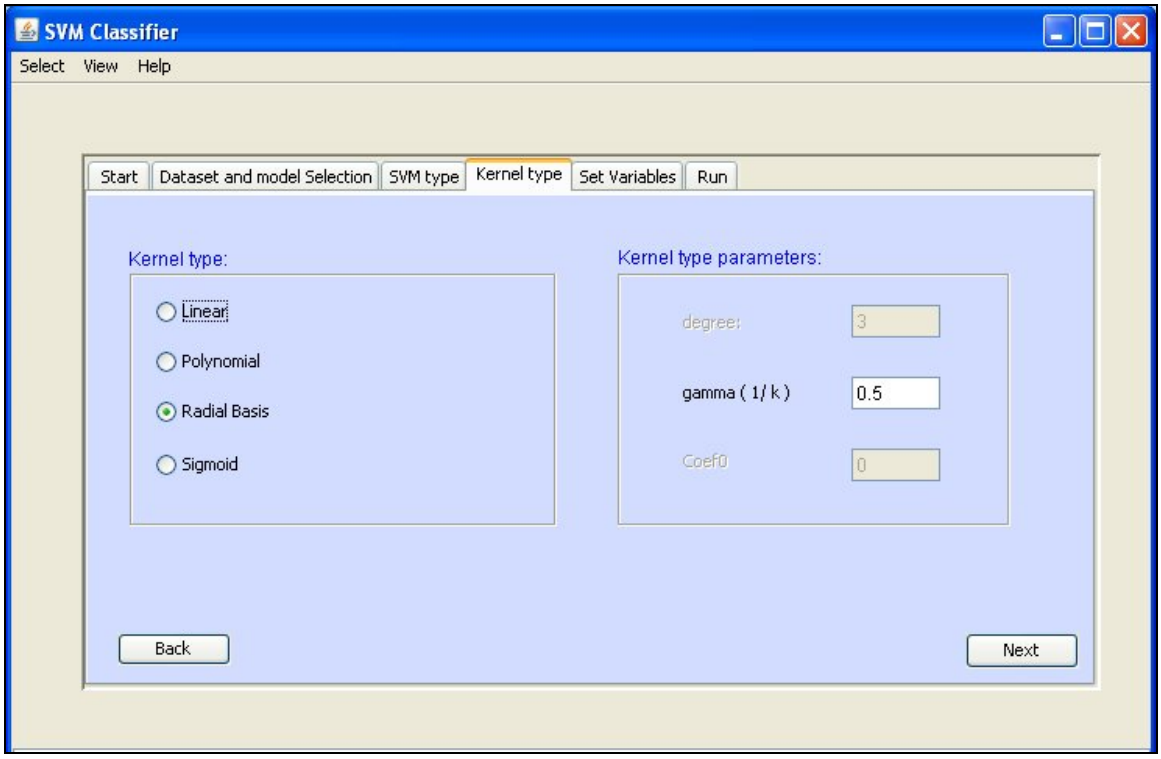
6.1.5 Kernel Types

Ο χρήστης έχει την επιλογή μέσω ενός παραθύρου, να διαλέξει τον τύπο του Kernel που θέλει να χρησιμοποιήσει. Οι τέσσερις πιο βασικοί πυρήνες (kernels) είναι: linear, polynomial, radial basic function (RBF) και sigmoid.

1. **Linear**: $K(x, x_i) = x^T x_i$
2. **Polynomial**: Ο πολυωνυμικός πυρήνας δύναμης d είναι της μορφής

   $K(x, x_i) = (x, x_i)^d$
3. **RBF**: Ο γκαουσιανός πυρήνας (Gaussian) επίσης γνωστός και ως radial basis function, είναι της μορφής $K(x, x_i) = \exp(- (||x_i - x_j||^2) / (2\sigma^2))$
4. **Sigmoid**: $K(x, x_i) = \tanh(k(x, x_i) + \theta)$

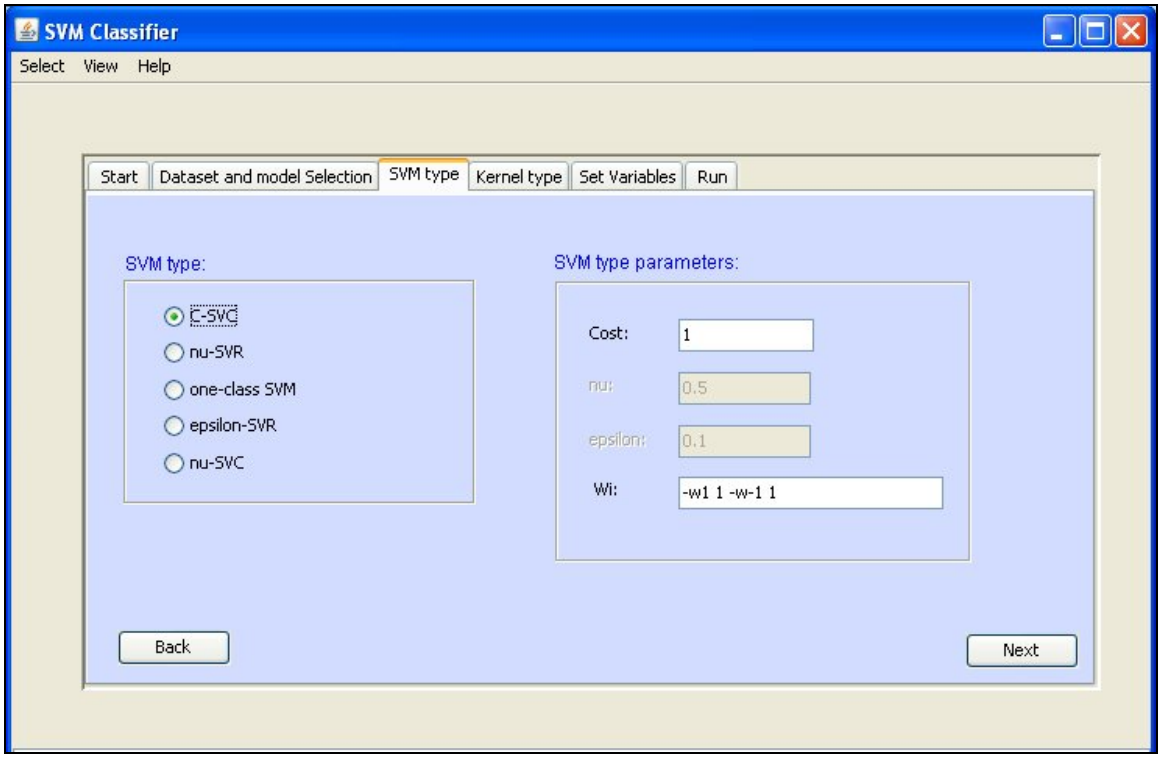
   Όταν χρησιμοποιείται sigmoid kernel με το SVM μπορεί να θεωρηθεί ως two-layer network.



Σχήμα (10) *Ο χρήστης έχει την επιλογή μέσω ενός παραθύρου, να διαλέξει τον τύπο του Kernel που θέλει να χρησιμοποιήσει.*

6.1.6 SVM Types

Ο χρήστης έχει την επιλογή μέσω ενός παραθύρου, να διαλέξει τον τύπο του SVM που θέλει να χρησιμοποιήσει . Τα SVM Types που παρέχονται από τον προσομοιωτή είναι τα εξής: C-SVC, nu-SVR, one-class SVM, epsilon-SVR, nu-SVC.



Σχήμα (11) *Ο χρήστης έχει την επιλογή μέσω ενός παραθύρου, να διαλέξει τον τύπο του SVM που θέλει να χρησιμοποιήσει.*

6.1.6.1 C-SVC: C-Support Vector Classification (Binary Case)

Θεωρούμε το δεδομένο εκπαίδευσης $\{(x_i, d_i)\}_{i=1..n}$, όπου το x_i είναι το πρότυπο εισόδου για το i-οστό παράδειγμα και d_i είναι το επιθυμητό αποτέλεσμα (target output) με τιμές $\{+1, -1\}$.

Το support vector machine απαιτεί λύση στο ακόλουθο πρόβλημα βελτιστοποίησης:

Δεδομένου του συνόλου εκπαίδευσης $\{(x_i, d_i)\}_{i=1..N}$, πρέπει να βρεθούν οι βέλτιστες τιμές του διανύσματος βαρών w και κατωφλίου b τέτοιες ώστε να ικανοποιούν τον περιορισμό

$$d_i (w^T x_i + b) \geq 1 - \xi_i \quad \text{για } i = 1,2,..N$$

$$\xi_i \geq 0 \quad \text{για όλα τα } i$$

και τέτοια ώστε το διάνυσμα βαρών w και οι μεταβλητές slack variables ξ_i να ελαχιστοποιούν τη συνάρτηση κόστους

$$\Phi(w, \xi) = ½ w^T w + C \Sigma_{i=1..N} \xi_i$$

όπου το C είναι μια θετική παράμετρος που καθορίζεται από το χρήστη.

Η συνάρτηση απόφασης είναι sgn($\Sigma_{i=1..N} d_i$ αi $K(x_i,x) + b$).

6.1.6.2 nu-SVC: v-Support Vector Classification (Binary Case)

Η παράμετρος ν Є (0,1) είναι άνω όριο στο κλάσμα των λαθών εκπαίδευσης και κάτω όριο στο κλάσμα των support vectors. Δεδομένου του συνόλου εκπαίδευσης $\{(x_i, d_i)\}_{i=1..N}$, πρέπει να βρεθούν οι βέλτιστες τιμές του διανύσματος βαρών w και κατωφλίου b τέτοιες ώστε να ικανοποιούν τον περιορισμό:

$$d_i (w^T x_i + b) \geq \rho - \xi_i \quad \text{για } i = 1,2,..N$$

$$\xi_i \geq 0, \rho \geq 0 \quad \text{για όλα τα } i$$

και τέτοια ώστε το διάνυσμα βαρών w και οι μεταβλητές slack variables ξ_i να ελαχιστοποιούν τη συνάρτηση κόστους

$$\Phi(w, \xi) = ½ w^T w - \theta\rho + (1/N) \Sigma_{i=1..N} \xi_i$$

Η συνάρτηση απόφασης είναι sgn($\Sigma_{i=1..N} d_i$ αi $K(x_i,x) + b$).

6.1.6.3 One-class SVM: distribution estimation

Η διαφορά στο one-class SVM σε σχέση με τα συνηθισμένα είναι ότι τα δεδομένα εκπαίδευσης δεν είναι πανομοιότυπα κατανεμημένα με τα δεδομένα ελέγχου. Τα δεδομένα περιέχουν δύο κλάσεις: μια απο αυτές, η κλάση που στοχεύουμε (target class), ελέγχεται καλά, ενώ η άλλη κλάση είναι απούσα ή ελέγχεται αραιά λόγω των λίγων

δειγμάτων. Η προσέγγιση του Scholkopf et al. [9] προτείνει να γίνεται διαχωρισμός μεταξύ της target class και του origin μέσω μιας υπερεπιφάνειας. Δεδομένου του συνόλου εκπαίδευσης $\{(x_i, d_i)\}_{i=1..N}$, πρέπει να βρεθούν οι βέλτιστες τιμές του διανύσματος βαρών w και κατωφλίου b τέτοιες ώστε να ικανοποιούν τον περιορισμό:

$$w^T x_i + b \geq \rho - \xi_i \quad \text{για } i = 1,2,..N$$

$$\xi_i \geq 0 \quad \text{για όλα τα } i$$

και τέτοια ώστε το διάνυσμα βαρών w και οι μεταβλητές slack variables ξ_i να ελαχιστοποιούν τη συνάρτηση κόστους

$$\Phi(w, \xi) = \tfrac{1}{2} w^T w - \rho + (1/N) \Sigma_{i=1..N} \xi_i$$

Η συνάρτηση απόφασης είναι $\text{sgn}(\Sigma_{i=1..N} d_i \alpha i K(x_i,x) + b)$.

Για τους πιο κάτω τύπους epsilon-SVR: ε-Support Vector Regression (ε-SVR), nu-SVR: ν-Support Vector Regression (ν-SVR) δεν έχει γίνει περιγραφή εφόσον χρησιμοποιούνται για regression, ενώ στη δική μας περίπτωση χρησιμοποιούμε classification.

6.1.7 Cross Validation

Ο λόγος που χρησιμοποιείται cross validation είναι για να βρεθούν οι καλές παραμέτροι έτσι ώστε ο classifier να μπορεί να προβλέψει με ακρίβεια τα άγνωστα δεδομένα [10].

Ένας κοινός τρόπος είναι να χωρίσεις τα δεδομένα εκπαίδευσης σε δύο μέρη, εκ των οποίων το ένα θεωρείται άγνωστο για την εκπαιδεύση του classifier. Έτσι η ακρίβεια της πρόβλεψης σε αυτό το σύνολο μπορεί καλύτερα να αντικατοπτρίσει την απόδοση της κατηγοριοποίησης άγνωστων δεδομένων. Μια βελτιωμένη έκδοση αυτής της διαδικασίας είναι το cross-validation.

Σε μια v-fold cross-validation, το σύνολο δεδομένων χωρίζεται σε v υποσύνολα του ίδιου μεγέθους. Σειριακά, ένα υποσύνολο ελέγχεται χρησιμοποιώντας ένα classifier που είχε

εκπαιδευτεί στα υπόλοιπα (v – 1) υποσύνολα. Επομένως, κάθε δείγμα (instance) από ολόκληρο το σύνολο εκπαίδευσης προβλέπεται μόνο μια φορά επομένως η ακρίβεια του cross-validation είναι το ποσοστό των δεδομένων που είχαν σωστή κατηγοριοποίηση. Η διαδικασία του cross-validation μπορεί να εμποδίσει το πρόβλημα όπου γίνεται overfitting [10].

6.1.8 Shrinking

Οι Chang και Lin [10] ανάφεραν ότι εφόσον για πολλά προβλήματα το πλήθος των ελεύθερων support vectors (δηλαδή $0 \leq \alpha_i \leq C$) είναι μικρό, η τεχνική αυτή μειώνει το μέγεθος του εργασιακού προβλήματος χωρίς να λαμβάνει υπόψην του κάποιες περιορισμένες (bounded) μεταβλητές.

6.1.9 Caching

Αυτή είναι ακόμη μια τεχνική για να μειώσουμε τον υπολογιστικό χρόνο. Εφόσον ο πίνακας Q (ο Q είναι ένας NxN θετικός semi definite πίνακας, $Q_{ij} = d_i d_j K(x_i, x_j)$) είναι πλήρως πυκνός και μπορεί να μην αποθηκευθεί στη μνήμη του υπολογιστή, τα στοιχεία Q_{ij} υπολογίζονται όπως χρειάζεται. Συνηθώς μια ειδική μνήμη που χρησιμοποιεί την ιδέα της cache χρησιμοποιείται για να φυλάγονται τα Q_{ij} που χρησιμοποιήθηκαν πιο πρόσφατα [10].

7. ΚΕΦΑΛΑΙΟ 7

Αποτελέσματα

7.1 Ανάλυση αποτελεσμάτων εκτίμησης ακνήτων.....55
7.2 Απόδοση δικτύου με τις καλύτερες τιμές.....62

Σε αυτό το κεφάλαιο περιγράφονται όλες οι δοκιμές που έγιναν σε κάθε ένα από τα δίκτυα και γίνεται αναφορά στις παραμέτρους, οι οποίες επηρεάζουν την απόδοση, καθώς και η ανάλυση και η παρουσίαση των αποτελεσμάτων που πάρθηκαν σε κάθε δοκιμή.

7.1 Ανάλυση αποτελέσματων εκτίμησης ακινήτων

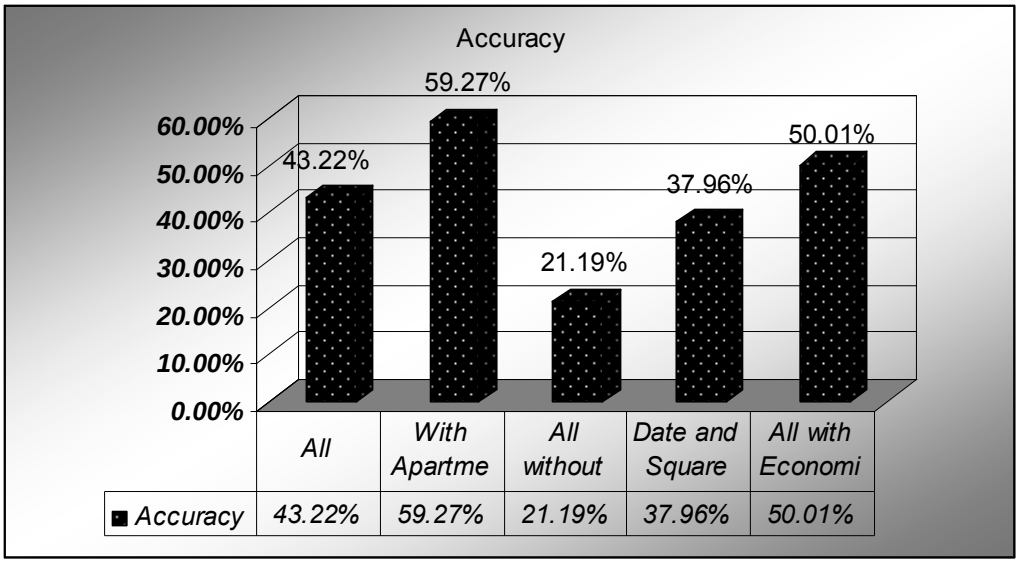
Για την εκτίμηση της τιμής των ακινήτων χρησιμοποιήθηκε το java interface SVM Classifier που αναλύθηκε προηγουμένως. Όπως έχω αναφέρει, η απόδοση των μηχανών διανυσμάτων υποστήριξης εξαρτάτε από πολλές παραμέτρους. Οι μηχανές αυτές εκπαιδεύτηκαν με διαφορετικούς παραμέτρους μέχρι να βρεθούν οι καλύτερες τιμές. Πιο κάτω παρουσιάζονται τα αποτελέσματα για τα δίκτυα αυτά.

7.1.1 Διάφορα σύνολα δεδομένων χωρίς χρήση ηλικίας

Για την εκπαίδευση του δικτύου έχουν χρησιμοποιηθεί διαφορετικά είδη συνόλων δεδομένων τα οποία έχουν επεξηγηθεί με σαφήνεια στο κεφάλαιο 5. Για να βρεθεί ο καλύτερος συνδυασμός των χαρακτηριστικών που θα μας δώσει τα καλύτερα αποτελέσματα έχω κρατήσει σταθερά τα SVM Type, Kernel Type και τις παραμέτρους τους και κάθε φορά η μηχανή εκπαιδευόταν με διαφορετικό σύνολο δεδομένων. Τα δεδομένα που χρησιμοποιήθηκαν για την εκπαίδευση του δικτύου είναι SVM Type: C-SVC, με cost = 10, w_i = -w1 1 -w-1 1, Kernel Type: Radial Basis, gamma(1/k) : 0.5.

	All	With Apartment, Square Meters and Date only	All without Apartment Type and Quarter	Date and Square Meters	All with Economic Factors
Accuracy	43.22 %	59.27 %	21.19 %	37.96 %	50.01 %

Πίνακας (3) Πίνακας με τα ποσοστά επιτυχίας για διάφορα σύνολα δεδομένων



Παρατηρούμε σε γενικές γραμμές ότι τα ποσοστά είναι αρκετά χαμηλά. Με τα δεδομένα With Apartment, Square Meters and Date only φτάνουμε το ποσοστό 59.27 % που είναι αρκετά καλύτερο από τα υπόλοιπα αλλά δεν παύει να είναι χαμηλό ποσοστό. Γενικά δε μπορούμε να πάρουμε ψηλό ποσοστό ακρίβειας λόγω απουσίας της ηλικίας από τα δεδομένα.

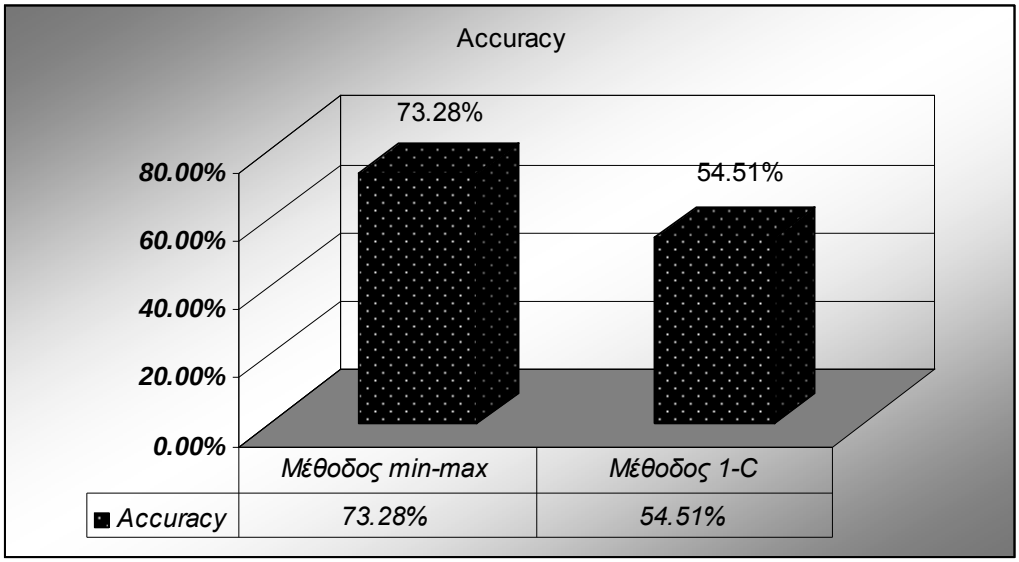
7.1.2 Αποτελέσματα με δεδομένα τη χρήση ηλικίας

Από τα προηγούμενα αποτελέσματα παρατηρούμε ότι τα ποσοστά δεν είναι πολύ ψηλά, γεγονός που το πιο πιθανό ευθύνεται στην απουσία ηλικίας. Για να εξακριβωθεί κατά πόσο ήταν εφικτή η βελτιστοποίηση της απόδοσης της μηχανής, η Ειρήνη Πέτρου [5] βρήκε ένα τρόπο να δημιουργεί μια εικονική ηλικία για τα ακίνητα. Έτσι εκπαιδεύσαμε το SVM προσθέτοντας στα δεδομένα μας την ηλικία. Η ηλικία είχε κανονικοποιηθεί με δύο διαφορετικούς τρόπους, τη μέθοδο min-max και τη μέθοδο 1-C. Εξετάστηκαν και οι

δύο για να δούμε ποιά από τις δύο μεθόδους μας δίνει τα καλύτερα αποτελέσματα. Τα αποτέλεσματα φαίνονται στον πίνακα (3) πιο κάτω. Παρατηρούμε ότι με τη μέθοδο 1-C το ποσοστό ανέρχεται μόλις στο 54.51 %, ενώ με τη μέθοδο min-max το ποσοστό ακρίβειας ανέρχεται στο 73.28 % που είναι αισθητά καλύτερο. Τα δεδομένα που χρησιμοποιήθηκαν είναι τα ίδια όπως αναφέρθηκαν προηγουμένως.

	Μέθοδος min-max	**Μέθοδος 1-C**
Accuracy	73.28 %	54.51 %

Πίνακας (4) Πίνακας με τα ποσοστά επιτυχίας για τα σύνολα δεδομένων που έχουν ηλικία με τη μέθοδο min-max και τη μέθοδο 1-C



7.1.3 Τύπος του SVM (SVM Type)

Οι τύποι των SVMs είναι οι: C-SVC, nu-SVC, epsilon-SVR, nu-SVR, One-Class SVM και έχουν εξηγηθεί στο κεφάλαιο 6.1.5. Για το δικό μας πρόβλημα το οποίο αποτελεί πρόβλημα κατηγοριοποίησης ο κατάλληλος από τους πέντε πιο πάνω τύπους είναι ο πρώτος, δηλαδή ο C-SVC. Οι δύο τελευταίοι epsilon-SVR, nu-SVR εξαιρούνται εφόσον αυτοί αναφέρονται σε regression και όχι σε classification που δεν μας ενδιαφέρει στην

περίπτωση μας. Όπως επίσης και οι τύποι nu-SVC και One-class SVM δε μπορούν να χρησιμοποιηθούν λόγω του ότι αναφέρονται σε διαφορετικό είδος προβλήματος από αυτό που μας απασχολεί.

Τα δεδομένα που χρησιμοποιήθηκαν είναι με χρήση ηλικίας και τετραγωνικά μέτρα με τη μέθοδο min-max. Όπως αναφέρθηκε ο τύπος που χρησιμοποιήθηκε για το δικό μας πρόβλημα είναι ο C-SVC που είναι ο καταλληλότερος τύπος SVM για να χρησιμοποιηθεί σε προβλήματα κατηγοριοποίησης (classification). Η αποδοτικότητα αυτού του τύπου επηρεάζεται από την παράμετρο cost. Για να ελέγξουμε αυτή τη παράμετρο έχουμε κρατήσει σταθερά τα δεδομένα εισόδου, το kernel type και τις παραμέτρους του. Πιο κάτω, ο πίνακας (5) απεικονίζει τις διάφορες τιμές που δοκιμάστηκαν για την παράμετρο cost κρατώντας τα υπόλοιπα σταθερά. Παρατηρούμε ότι το κόστος στον τύπο του SVM επηρεάζει αισθητά την απόδοση του δικτύου. Δοκίμαζοντας διάφορες τιμές του cost σε διαφορετικούς kernel types κάθε φορά έχω παρατηρήσει ότι δεν υπάρχει μια βέλτιστη τιμή του cost που να παράγει τα καλύτερα αποτέλεσματα σε κάθε kernel type. Ο κάθε kernel type επηρεάζεται διαφορετικά με την τιμή του κόστους και για κάθε ένα υπάρχει διαφορετική βέλτιστη τιμή του cost.

	Cost = 1.5	**Cost = 7.5**	**Cost = 12**	**Cost = 20**
Accuracy	93.58%	96.06%	96.28%	96.32%

Πίνακας (5) Πίνακας με τα ποσοστά επιτυχίας για διάφορες τιμές της παραμέτρου cost για τον τύπο των SVM C-SVC

7.1.4 Τύπος του Kernel (Kernel Type)

Για να βρεθεί ο καλύτερος τύπος του πυρήνα που θα μας δώσει τα καλύτερα αποτελέσματα έχω κρατήσει σταθερά τα SVM Type και τις παραμέτρους του και κάθε φορά η μηχανή εκπαιδευόταν με διαφορετικό τύπο πυρήνα. Οι τύποι πυρήνα είναι οι εξής: Linear, Polynomial, Radial Basis, Sigmoid.

7.1.4.1 Linear

Σε αυτό τον τύπο δε χρειάστηκε να κοθορίσουμε παραμέτρους, εφόσον δεν χρησιμοποιείται καμία παράμετρος που να χρειάζεται να οριστεί από το χρήστη. Εκπαιδεύοντας τη μηχανή με σταθερές όλες τις υπόλοιπες παραμέτρους και τα δεδομένα δοκίμασαμε διάφορες τιμές για το κριτήριο τερματισμού όπως φαίνεται στον πίνακα (6). Όσο πιο μικρό το κριτήριο τερματισμού παρατηρούμε ότι τόσο καλύτερα είναι και τα αποτελέσματα. Το ψηλότερο ποσοστό ακρίβειας που επιτυγχάνεται είναι το 83.84 %. Παρατηρούμε ότι αυξάνωντας το κριτήριο τερματισμού με τιμή πέραν του 0,00001 το ποσοστό ακριβείας δε βελτιώνεται επομένως κρατούμε ως καλύτερη τιμή για το termination criteria το 0,00001.

	Termination criteria = 0.1	**Termination criteria = 0.001**	**Termination criteria = 0.00001**	**Termination criteria = 0.0000001**
Accuracy	83.74 %	83.82%	83.84%	83.84%

Πίνακας (6) Πίνακας με τα ποσοστά επιτυχίας για διάφορες τιμές του κριτήριο τερματισμού για τον τύπο των SVM C-SVC και Linear kernel type

7.1.4.2 Polynomial

Ο πολυωνυμικός τύπος πυρήνα εξαρτάται από τρεις παραμέτρους: degree, gamma και coefficient. Το degree δε φαίνεται να παρουσιάζει καμία αλλαγή στο ποσοστό ακρίβειας γι'αυτό παρουσιάζουμε τις άλλες δύο παραμέτρους που φαίνεται να επηρεάζουν την απόδοση της μηχανής. Για να βρούμε τις καλύτερες τιμές κρατάμε σταθερά τα δύο εκ των τριών και ελέχγουμε μια παράμετρο κάθε φορά.

7.1.4.2.1 ΠΑΡΑΜΕΤΡΟΣ GAMMA(1/K)

Παρατηρούμε ότι για μικρές τιμές της παραμέτρου gamma η απόδοση του δικτύου βρίσκεται κοντά στο 91%. Παρατηρούμε ότι όσο μεγαλώνει η τιμή της παραμέτρου τα αποτελέσματα είναι καλύτερα αλλά η βέλτιστη τιμή βρίσκεται κάπου στο μέσο. Δηλαδή με τιμή 1,2 για τη παράμετρο gamma τα ποσοστά ανέρχονται στο 94,92%.

	gamma=0.5	Gamma = 1.2	gamma = 1.9	gamma = 2.5	gamma = 10
Accuracy	91.60 %	94.92%	94.86%	94.64%	94.52%

Πίνακας (7) Πίνακας με τα ποσοστά επιτυχίας για διάφορες τιμές της παραμέτρου gamma για το kernel type polynomial

7.1.4.2.2 ΠΑΡΑΜΕΤΡΟΣ COEFFICIENT

Η παράμετρος coefficient φαίνεται να επηρεάζει αισθητα το δίκτυο. Έχουμε κρατήσει σταθερή την τιμή του gamma που βρέθηκε προηγουμένως, δηλαδή στο 1,2 και ελέγχαμε το δίκτυο για διαφορετικές τιμές του coefficient. Ξεκινώντας από μικρές τιμές, παρατηρούμε ότι το δίκτυο δεν έχει και τόσο καλά αποτελέσματα. Αυξάνωντας το μέγεθος της μεταβλητής αυτής παρατηρούμε ότι αυξάνεται και η απόδοση. Η καλύτερη απόδοση που επιτυγχάνεται φθάνει το 95,5% με τιμή για τη μεταβλητή coefficient ίση με 1. Παρατηρούμε ότι για τιμές μεγαλύτερες από 1 η απόδοση του δικτύου μειώνεται.

	coefficient = 0.5	coefficient = 1	coefficient = 1.2	coefficient = 1.6
Accuracy	95.42%	95.5%	95.34%	95.08%

Πίνακας (8) Πίνακας με τα ποσοστά επιτυχίας για διάφορες τιμές της παραμέτρου coefficient για το kernel type polynomial

7.1.4.2.3 ΠΑΡΑΜΕΤΡΟΣ DEGREE

Κρατώντας σταθερά το coefficient και gamma στις τιμές 1 και 1,2 αντίστοιχα, ελέγξαμε για τις διάφορες τιμές της παραμέτρου degree. Παρατηρούμε ότι ο συνδυασμός με gamma = 1,2 και coefficient = 1 και degree = 3 και cost του SVM Type ίσο με 8 παίρνουμε τα καλύτερα αποτέλεσματα για τον πολυωνυμικό τύπο πυρήνα που ανέρχεται στο 95,52%.

	degree = 1	degree = 2	degree = 3	degree = 5	degree = 8
Accuracy	85.06 %	94.69 %	95.52 %	94.76 %	92.72 %

Πίνακας (9) Πίνακας με τα ποσοστά επιτυχίας για διάφορες τιμές της παραμέτρου degree για το kernel type polynomial

7.1.4.3 Radial Basis

Στον πιο κάτω πινάκα ελέξαμε τον τύπο radial basis για έξι διαφορετικές τιμές τις παραμέτρου gamma και για SVM type cost = 1. Παρατηρούμε ότι όταν το gamma πάρει μικρή ή μεγάλη τιμή δεν αποδίδει και τόσο καλά. Τα καλύτερα αποτελέσματα τα παίρνουμε με ένα μέσο gamma = 5 όπου επτυγχάνεται ποσσοτό 96,52 %.

7.1.4.3.1 ΠΑΡΑΜΕΤΡΟΣ GAMMA(1/K)

	gamma = 0.2	gamma = 0.5	gamma = 1.0	gamma = 2.5	gamma = 5	gamma = 10
Accuracy	86.96 %	91.98 %	94.62%	96.2%	96.52%	94.3%

Πίνακας (10) Πίνακας με τα ποσοστά επιτυχίας για διάφορες τιμές της παραμέτρου gamma για το kernel type radial basis

7.1.4.4 Sigmoid

Ο πολυωνυμικός τύπος πυρήνα εξαρτάται από δύο παραμέτρους: gamma και coefficient. Για να βρούμε τις καλύτερες τιμές κρατάμε σταθερό το ένα εκ των δύο και ελέχγουμε μια παράμετρο κάθε φορά. Για την εύρεση του καλύτερου gamma κρατάμε την τιμή του coefficient ίση με μηδέν. Ενώ για την εύρεση του καλύτερου coefficient χρησιμοποιούμε την τιμή του gamma που βρέθηκε ως η καλύτερη δηλαδή . Παρατηρούμε ότι για gamma = 10 και coefficient = 50 το SVM πετυγχαίνει καλύτερα αποτελέσματα που ανέρχονται στο 59,62% και 56,78% αντιστοίχα.

7.1.4.4.1 ΠΑΡΑΜΕΤΡΟΣ GAMMA(1/K)

	gamma = 0.5	gamma = 1.5	gamma = 0.9	Gamma = 10	gamma = 30
Accuracy	14.52%	33.22 %	57.29 %	59.62 %	52.41 %

Πίνακας (11) Πίνακας με τα ποσοστά επιτυχίας για διάφορες τιμές της παραμέτρου gamma για το kernel type sigmoid

7.1.4.4.2 ΠΑΡΑΜΕΤΡΟΣ COEFFICIENT

	coefficient =5	coefficient = 15	coefficient = 50	coefficient = 100
Accuracy	43.90 %	47.62 %	56.78 %	49.50 %

Πίνακας (12) Πίνακας με τα ποσοστά επιτυχίας για διάφορες τιμές της παραμέτρου coefficient για το kernel type sigmoid

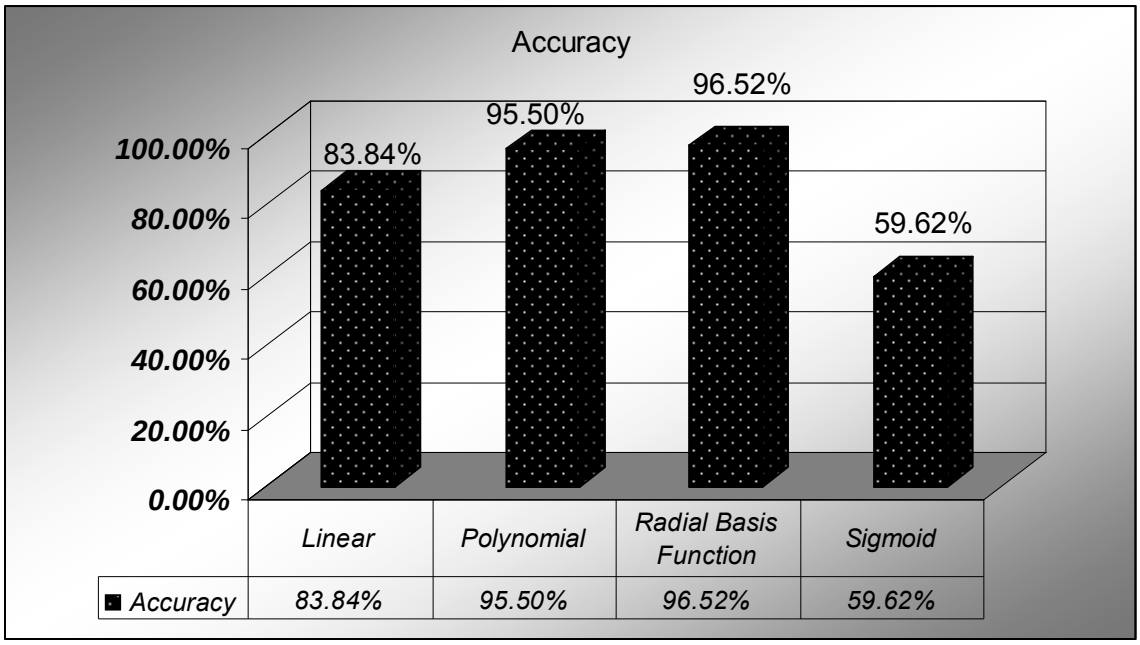
7.2 Απόδοση δικτύου με τις καλύτερες τιμές

Από τα πιο πάνω αποτελέσματα παρατηρούμε ότι το δίκτυο δεν αποδίδει και τόσο καλά με τον kernel type Sigmoid όπου τα καλύτερα ποσοστά ανέρχονται κάπου μεταξύ 56-59%. Με τη χρήση του linear kernel type παίρνουμε αποτελέσματα κοντά στο 83% που είναι αρκετά καλύτερα σε σχέση με αυτά που πήραμε με το sigmoid kernel type. Οι δύο

καλύτεροι kernel types φαίνεται να είναι ο polynomial και ο RBF. Με αρκετό πειρματισμό στις παραμέτρους gamma, coefficient και degree που φαίνεται να επηρεάζουν αισθητά την απόδοση του polynomial, επιτυγχάνουμε ποσοστό που ανέρχεται στο 95.5%. Ενώ με το kernel type RBF το ποσοστό ανέρχεται στο 96.2%.

	Linear	Polynomial	Radial Basis Function	Sigmoid
Accuracy	83.84%	95.50%	96.52%	59.62%

Πίνακας (13) Πίνακας με τα ποσοστά επιτυχίας για του διαφορετικούς kernel types

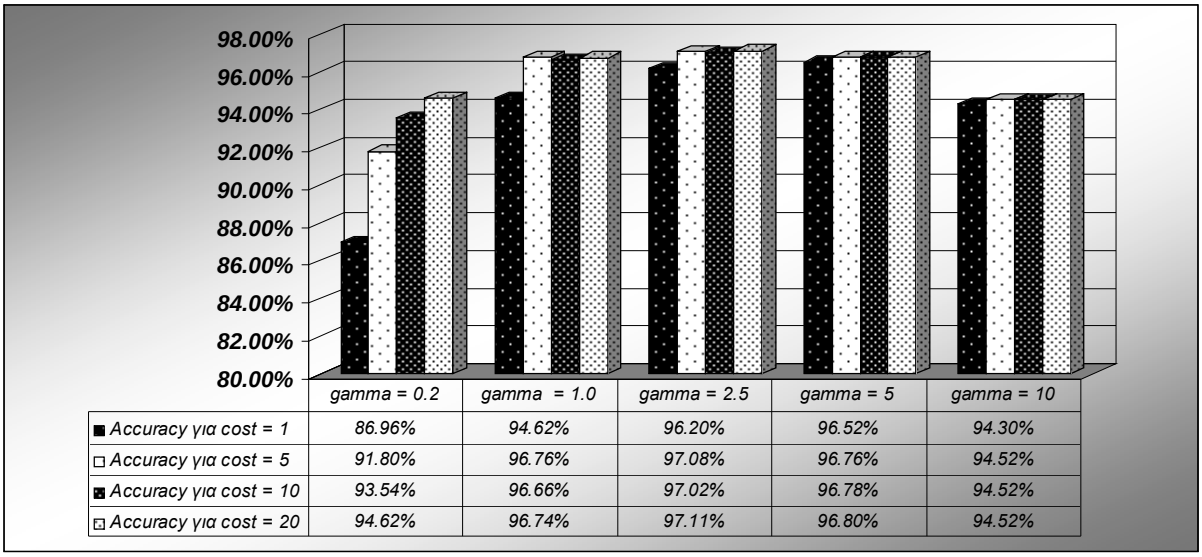


Όπως έχω αναφέρει στο 7.1.3 το cost του SVM Type δεν έχει μια σταθερή βέλτιστη τιμή για όλα τα kernel types και τις παραμέτρους τους. Επομένως αφού καταλήξαμε ότι με το RBF επιτυγχάνονται τα καλύτερα αποτελέσματα θα ελέγξουμε για διάφορες τιμές να βρούμε τον καλύτερο συνδιασμό τιμών που μας δίνει το καλύτερο ποσοστό. Στον πίνακα

(14) παρουσιάζω τις διάφορες τιμές του gamma και cost και τα ποσοστά που επιτυγχάνονται για κάθε συνδυασμό τιμών.

	gamma = 0.2	gamma = 1.0	gamma = 2.5	gamma = 5	gamma = 10
Accuracy για cost = 1	86.96 %	94.62%	96.2%	96.52%	94.3%
Accuracy για cost = 5	91.8 %	96.76 %	97.08 %	96.76%	94.52 %
Accuracy για cost = 10	93.54 %	96.66 %	97.02 %	96.78 %	94.52 %
Accuracy για cost = 20	94.62%	96.74 %	97.11 %	96.8%	94.52 %

Πίνακας (14) Πίνακας με τα ποσοστά επιτυχίας για διάφορες τιμές της παραμέτρου gamma και του cost για το kernel type radial basis



Από τον πιο πάνω πίνακα παρατηρούμε ότι τα ποσοστά επηρεάζονται ανάλογα με τον συνδυασμό παραμέτρων. Ο καλύτερος συνδυασμός επιτυγχάνεται με gamma ίσο με 2.5 και cost ίσο με 20 όπου το ποσοστό ανέρχεται στο 97.11%.

8. ΚΕΦΑΛΑΙΟ 8

Συζήτηση

8.1 Σύγκριση αποτελεσμάτων με προηγούμενες μελέτες.....65
8.2 Προβλήματα που παρουσιάστηκαν.....66

8.1 Σύγκριση αποτελεσμάτων με προηγούμενες μελέτες

Στη μελέτη της Μάριαμ Αποστολίδου [4] χρησιμοποιώντας τα ίδια δεδομένα και (δηλαδή τα δεδομένα με την ηλικία και τα τετραγωνικά μέτρα ενός ακινήτου) πέτυχε ποσοστό 85% χρησιμοποιώντας Multilayer Perceptron με χρήση του backpropagation algorithm. Η Ειρήνη Πέτρου [5], με τη χρήση Radial Basis Function δικτύου και με τα ίδια δεδομένα απέδειξε ότι μπορεί να πετύχει καλύτερα αποτέλεσματα. Στη διπλωματική εργασία της Ειρήνης Πέτρου, όπου χρησιμοποιήθηκε δίκτυο Radial Basis Function με τυχαία κέντρα το ποσοστό επιτυχίας ανήλθε στο 91%, δηλαδή με 6% αύξηση από το ποσοστό της Μάριαμ Αποστολίδου. Το καλύτερο ποσοστό όμως επιτεύχθηκε με το υβριδικό δίκτυο Χάρτη Αυτοοργάνωσης σε συνδυασμό με το Radial Basis Function που ανήλθε στο 96%. Σαφώς το υβριδικό δίκτυο παρέχει καλύτερα αποτέλεσματα από ένα Multilayer Perceptron. Σε αυτή τη διπλωματική εργασία, με τα ίδια δεδομένα που χρησιμοποιήθηκαν στις προηγούμενες περιπτώσεις και με εκπαίδευση ενός SVM το ποσοστό που επίτευχθηκε ανέρχεται στο 97%. Το αποτελέσμα είναι ελαφρώς καλύτερο από αυτό που επιτεύχθηκε από το υβριδικό δίκτυο της Ειρήνης Πέτρου. Τα αποτέλεσματα είναι τα αναμενώμενα εφόσον από τη θεωρία τα SVMs παρουσιάζουν καλύτερα αποτελέσματα από τα προηγούμενα δίκτυα.

8.2 Προβλήματα που παρουσιάστηκαν

Ένα σημαντικό πρόβλημα ήταν η εύρεση του κατάληλου προσομοιωτή ο οποίος θα μπορούσε να εφαρμοστεί ορθά στις δικές μας ανάγκες. Αρκετές εφαρμογές είχαν λάθη στη μετάφραση ή στο runtime, και χρειάστηκε να επέμβω σε κώδικα πράγμα που αποδείχτηκε εξαιρετικά δύσκολο εφόσον δεν υπήρχε το κατάλληλο documentation. Αυτό ξεπεράστηκε με την εύρεση του προσομοιωτή SVM Classifier που είναι υλοποιημένος σε ένα φιλικό περιβάλλον java graphical user interface και κάνει χρήση των βιβλιοθηκών LIBSVM κατάλληλες για μηχανές SVMs.

Ακόμη ένα σημαντικό πρόβλημα που παρουσιάστηκε ήταν να μετατρέψουμε τα δεδομένα στην κατάλληλη μορφή ώστε να μπορεί να γίνει ορθή εκπαίδευση και γενίκευση από τη μηχανή. Ως λύση αυτού, έχω δημιουργήσει ένα μικρό πρόγραμμα το οποίο δέχεται ως είσοδο το αρχείο και το μετατρέπει στην κατάλληλη μορφή.

Ο προσομοιωτής φαίνεται να είναι αρκετά ευαίσθητος στη μορφή των δεδομένων. Με μικρές αλλαγές στο αρχείο δεδομένων ο προσομοιωτής δούλευε με καλύτερα ποσοστά ή χειρότερα ανάλογα με την αλλαγή.

9. ΚΕΦΑΛΑΙΟ 9

Συμπεράσματα

Ο παραδοσιακός τρόπος εκτίμησης, που βασίζεται στη σύγκριση κόστους και τιμής πώλησης, στερείται τα αποδεκτά πρότυπα και μια διαδικασία ταυτοποίησης. Επομένως, η παροχή ενός μοντέλου εκτίμησης τιμών ακινήτων μπορεί να βοηθήσει σημαντικά στην εκπλήρωση ενός σημαντικού χάσματος έλλειψης πληροφοριών και να βελτιώσει την αποδοτικότητα στην αγορά ακίνητης περιουσίας. Τα μοντέλα Νευρωνικών Δικτύων, έχουν αποδειχθεί κατάλληλα για μια αποδοτικότερη και γρηγορότερη διαδικασία εκτίμησης των τιμών ακινήτων. Αυτός ήταν και ο στόχος της διπλωματικής αυτής εργασίας. Η ανεύρεση και ανάπτυξη μεθόδων Νευρωνικών Δικτύων που θα βοηθήσουν στην αυτοματοποίηση της εκτίμησης τιμής των ακινήτων. Σημαντικό είναι να σημειώσουμε ότι τέτοιες μεθόδοι μπορούν να χρησιμοποιηθούν για να βοηθήσουν στη διεργασία αυτή και όχι για να αντικαταστήσουν τον παραδοσιακό τρόπο εκτίμησης.

Αυτή η διπλωματική εργασία επικεντρώθηκε στη μέθοδο των Support Vector Machines με την προϋπόθεση ότι μπορούν να επιφέρουν καλύτερα αποτελέσματα έναντι άλλων τεχνικών. Αυτή η μέθοδος στηρίζεται σε πολλές παραμέτρους ανάμεσα τους και η δομή των δεδομένων. Τα SVMs μπορούν να επιφέρουν καλύτερα αποτελέσματα σε σύνολα δεδομένων που αποτελούνται από μεγάλο πλήθος δεδομένων και χαρακτηριστικών. Σε αυτή τη διπλωματική εργασία, τα ποσοστά ακρίβειας και η σωστή κατηγοριοποίηση πέτυχαν ικανοποιητικά ποσοστά.

Αρκετοί παράγοντες δεν έχουν ληφθεί υπόψη στα παρόν δεδομένα. Όπως για παράδειγμα, η οικονομική κατάσταση που θεωρήθηκε σταθερή με το πέρασαμα του χρόνου αλλά και άλλοι οικονομικοί παράγοντες που δεν έχουν συμπεριληφθεί στα δεδομένα. Αυτά και άλλα χαρακτηριστικά θα μπορούσαν να βοηθήσουν την καλύτερη λειτουργία ενός support vector machine.

10. ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Rossini, P.A., (1997), "Application of Artificial Neural Networks to the Valuation of Residential Property", 3rd Pacific Rim Real Estate Society Conference, New Zealand 1997.

[2] Goble J., A Study of the Use of Neural Networks to Predict Residential Property Prices, BSc (Information Systems and Management) Thesis, Birkbeck, University of London, 2004.

[3] Pelissa Francois, Using Neural Network schemes as a prediction method for the property prices, MSc Computing Science Project Report, Birkbeck College, University of London, 2001.

[4] Apostolidou Mariam, Valuation of apartments in the city of Nicosia using Neural Networks, MSc (Computing Science) Thesis, University of Cyprus, 2007.

[5] Ειρήνη Πέτρου, Χρήση των Νευρωνικών Δικτύων στην Εκτίμηση τιμών ακινήτων, BSc (Computing Science), University of Cyprus, 2008.

[6] Vapnik, V. (1979). Estimation of Dependences Based on Empirical Data [in Russian], Nauka: Moscow. (1982). English Translation: Springer Verlag: New York.

[7] Vapnik, V. (1995). The Nature of Statistical Learning Theory. Springer Verlag: New York.

[8] Platis, Stelios and Orphanides, Stelios "*Cyprus Housing Market*", 2005 http://www.mapsplatis.com/research/samplereseacrh.html. Last looked: December 2008

[9] Schölkopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC: Estimating the support of a high-dimensional distribution. *Neural Computation* 2001, 13(**7**)**:**1443-1471.

[10] Chih-Chung Chang, Chih-Jen Lin: LIBSVM, a library for support vector machines. 2001 http://www.csie.ntu.edu.tw/~cjlin/libsvm]. Last looked: May 2009

11. APPENDIX

11.1 Προσομοιωτής SVM Classifier.....69

11.2 Μετατροπή από Delimited σε Labelled……..77

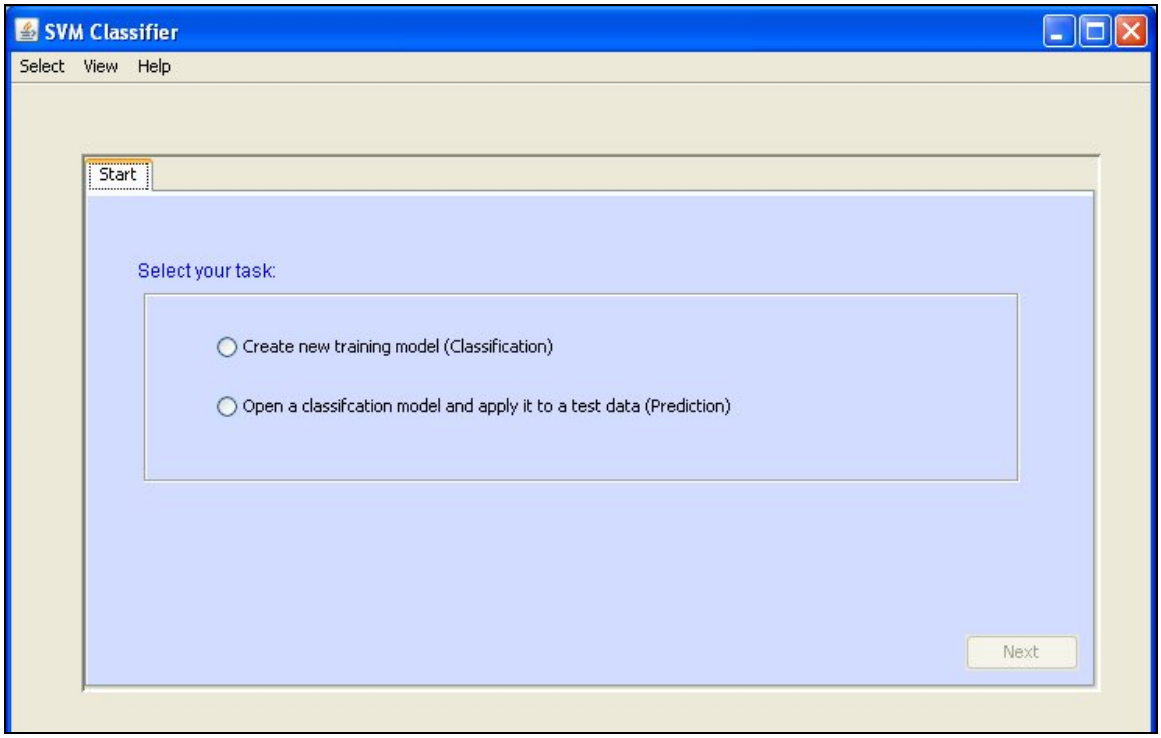
11.3 Source Code……...78

11.1 Προσομοιωτής SVM Classifier

Το πιο κάτω σχήμα εμφανίζεται με την έναρξη του προσομοιωτή. Στον προσομοιωτή υπάρχουν δύο επιλογές για κατηγοριοποίηση και επαλήθευση. Οι δύο επιλογές είναι:

1. Create new training model (Classification) και
2. Open a classification model and apply it to a test data (Prerdiction).

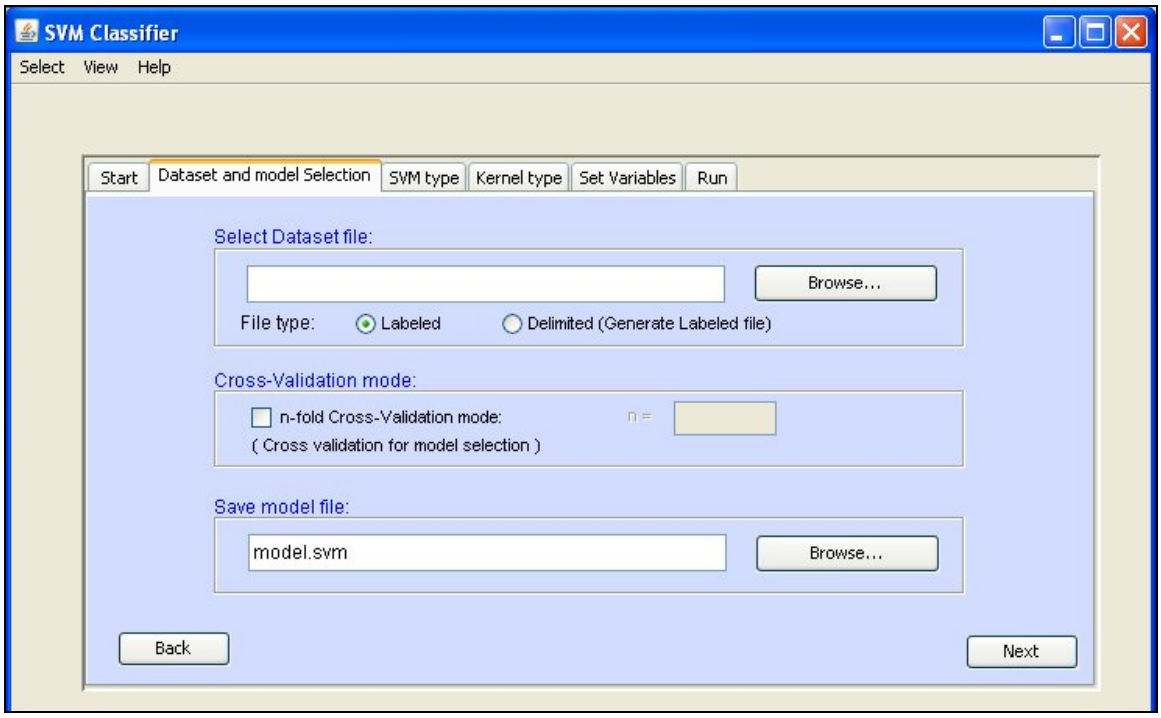
Πιο κάτω περιγράφω τα βήματα τα οποία πρέπει να ακολουθηθούν.



Σχήμα (1) Παράθυρο με τα πλαίσια επιλογής

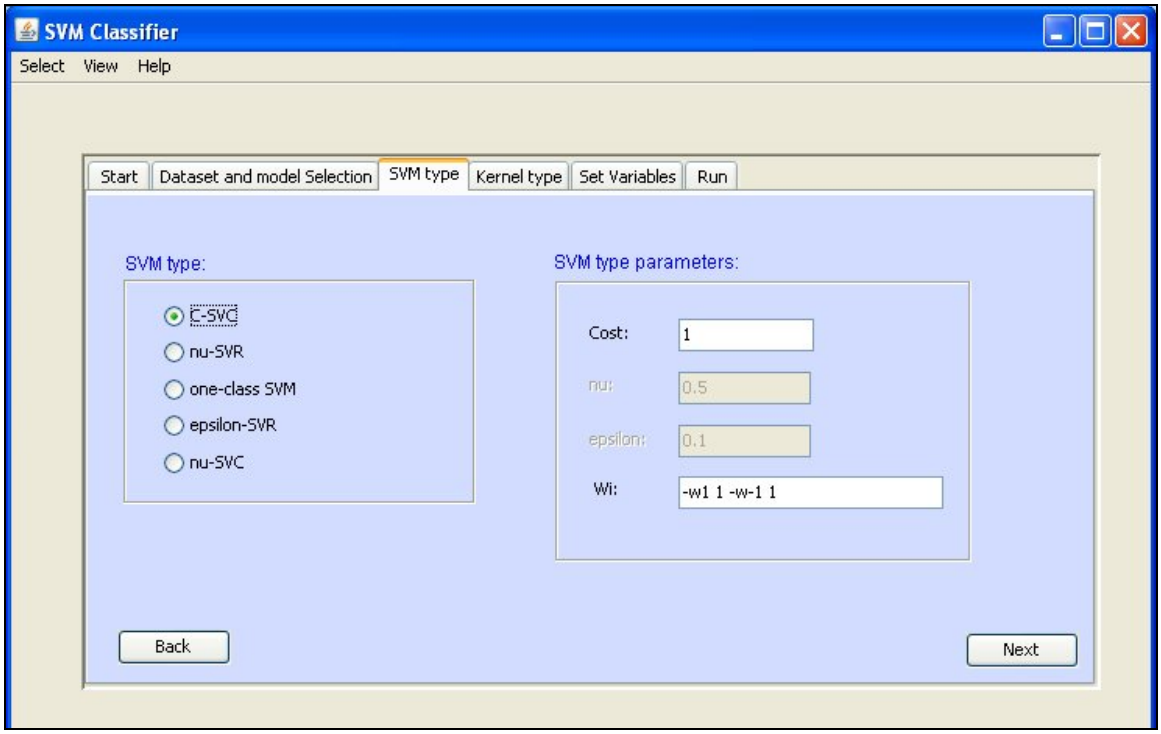
11.1.1 Κατηγοριοποίηση (Classification)

Απο τις πιο πάνω επιλογές πρώτα πρέπει να γίνει κατηγοριοποίηση και μετά πρόβλεψη. Επιλέγουμε το πρώτο bullet point και πατάμε next. Εμφανίζεται το tab Dataset and model selection όπως φαίνεται στο σχήμα (2). Σε αυτό το παράθυρο στο σημείο Select dataset file επιλέγεις το αρχείο με τα δεδομένα εκπαίδευσης. Το αρχείο μπορεί να έχει δύο μορφές είτε Labeled είτε Delimited. Λόγω προβλημάτων που παρουσιάστηκαν έχω υλοποιήσει ένα πρόγραμμα σε Java το οποίο δέχεται τα δύο αρχεία εκπαίδευσης και επαλήθευσης στη μορφή delimited και τα μετρατρέπει στη μορφή labeled. Αφού γίνει επιλογή του σωστού αρχείου εκπαίδευσης ο χρήστης πρέπει να επιλέξει που θα αποθηκεύσει το model file. Το model file είναι ένα μοντέλο που παράγεται κατά την κατηγοροποίηση και χρησιμοποιείται μετά στην πρόβλεψη.



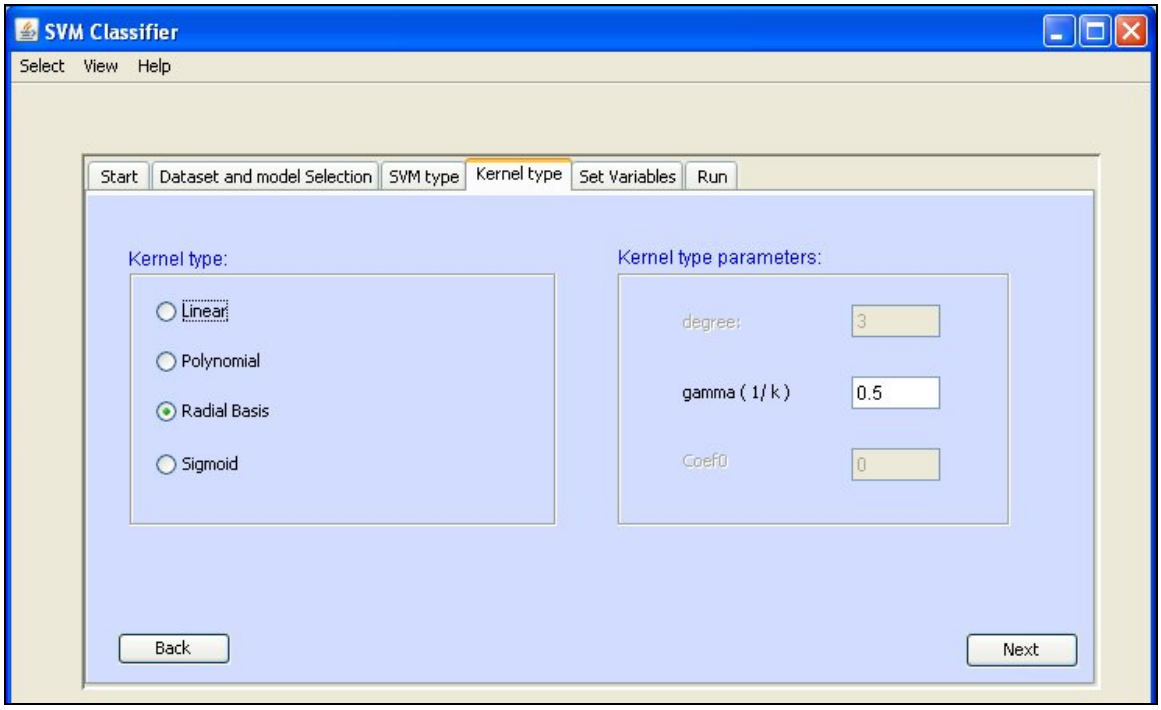
Σχήμα (2) Παράθυρο για επιλογή του αρχείου εκπαίδευσης και αποθήκευση του model file

Συνεχίζουμε στο επόμενο tab όπου πρέπει να επιλέξουμε το κατάλληλο SVM Type όπως φαίνεται στο σχήμα (3). Οι τύποι έχουν εξηγηθεί στο κεφάλαιο 6. Επιλέγεται ο κατάλληλος τύπος και αλλάζουμε τις παραμέτρους ανάλογα.



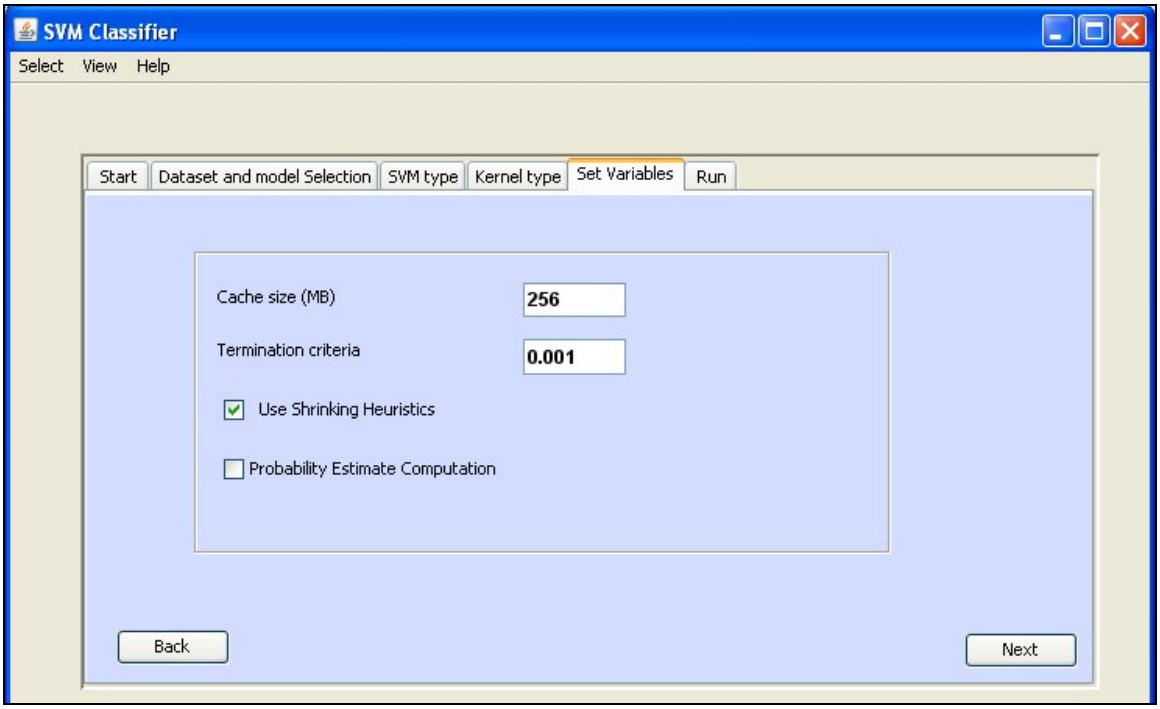
Σχήμα (3) Επιλογή του κατάλληλου SVM Type και των κατάλληλων παραμέτρων

Αφού επιλέξουμε το SVM Type και τις κατάλληλες παραμέτρους πατάμε next και προχωράμε στο tab Kernel Type, όπως φαίνεται στο σχήμα (4). Εδώ μας δίνεται η ευκαιρία να επιλέξουμε το kernel type και να αλλάξουμε τις παραμέτρους του. Για κάθε διαφορετικό τύπο πυρήνα υπάρχουν διαφορετικές παραμέτροι που μπορεί κάποιος να αλλάξει.



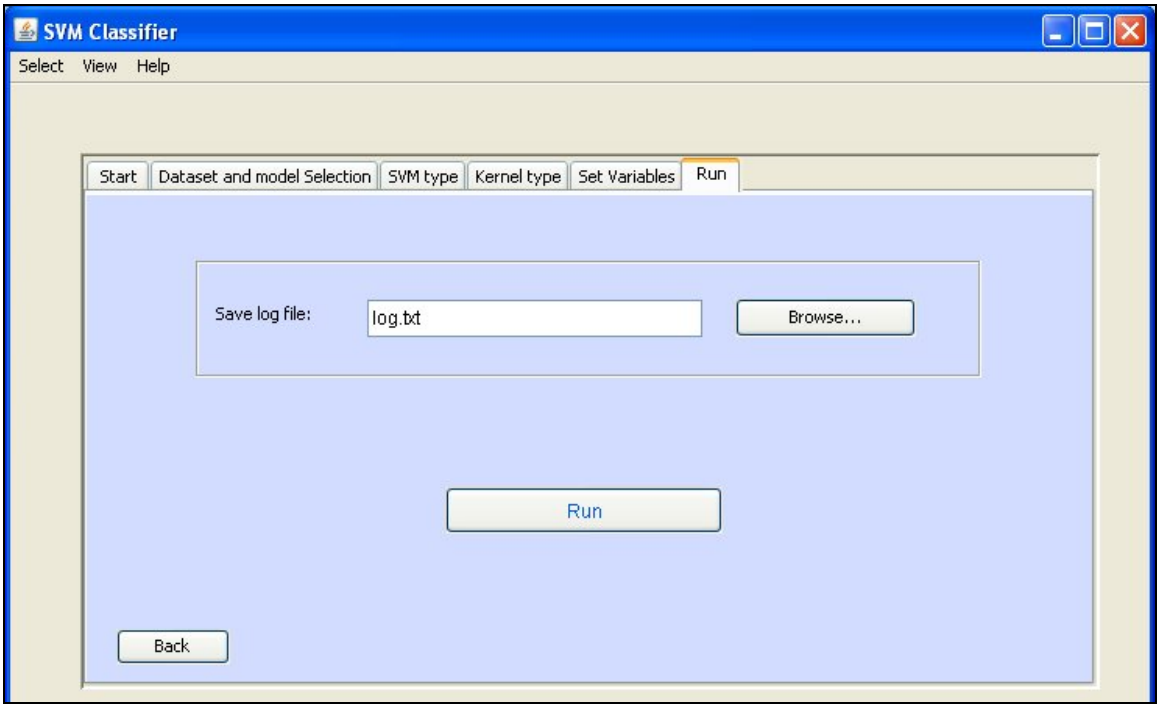
Σχήμα (4) Επιλογή του κατάλληλου kernel type και αλλαγή των παραμέτρων του

Μετά την επιλογή του kernel type προχωρούμε στο tab Set variables όπου μπορούμε να επιλέξουμε παραμέτρους που αφορούν γενικά το πρόγραμμα. Το Cache size είναι η μνήμη που δηλώνεις ότι χρειάζεται το πρόγραμμα και είναι σε MB. Η προκαθορισμένη τιμή είναι 256MB, ενώ ανάλογα με το μέγεθος του προβλήματος ίσως χρειαστεί να αυξηθεί. Η μεταβλητή termination criteria ελέγχει κατά κάποιο τρόπο τις επαναλήψεις που χρειάζεται να τρέξει το πρόγραμμα. Όσο πιο μικρή η τιμή τόσο περισσότερες επαναλήψεις θα εκτελεστούν.



Σχήμα (5) Το tab Set variables σου επιτρέπει να καθορίσεις κάποιες παραμέτρους γενικές με το πρόγραμμα όπως απαιτούμενη μνήμη και κριτήριο τερματισμού

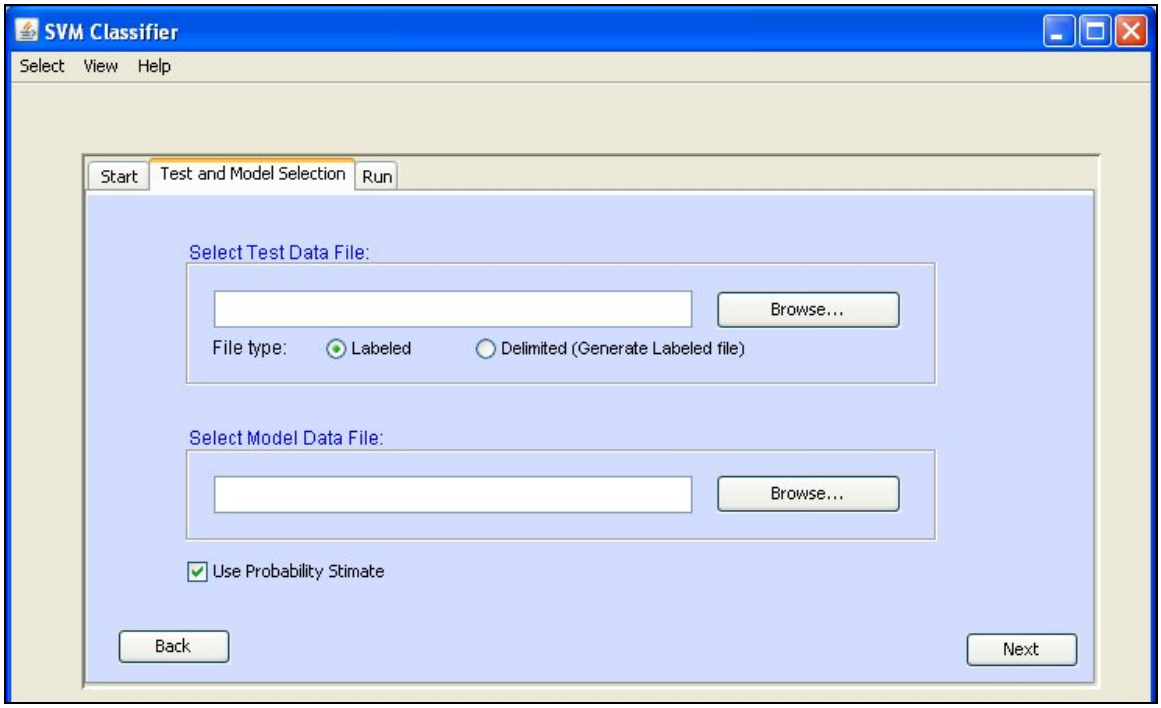
Στο τελευταίο βήμα για κατηγοριοποίηση επιλέγεις πού θέλεις να αποθηκεύσεις ένα log file στο οποίο καταγράφονται διάφορες λεπτομέρειες που παράγονται κατά την εκτέλεση του προγράμματος και επιλέγεις το κουμπί Run για να αρχίσει η διαδικασία κατηγοριοποίησης.



Σχήμα (6) Στο τελευταίο βήμα επιλέγεις πού θέλεις να αποθηκευθεί το log file και επιλέγεις "Run"

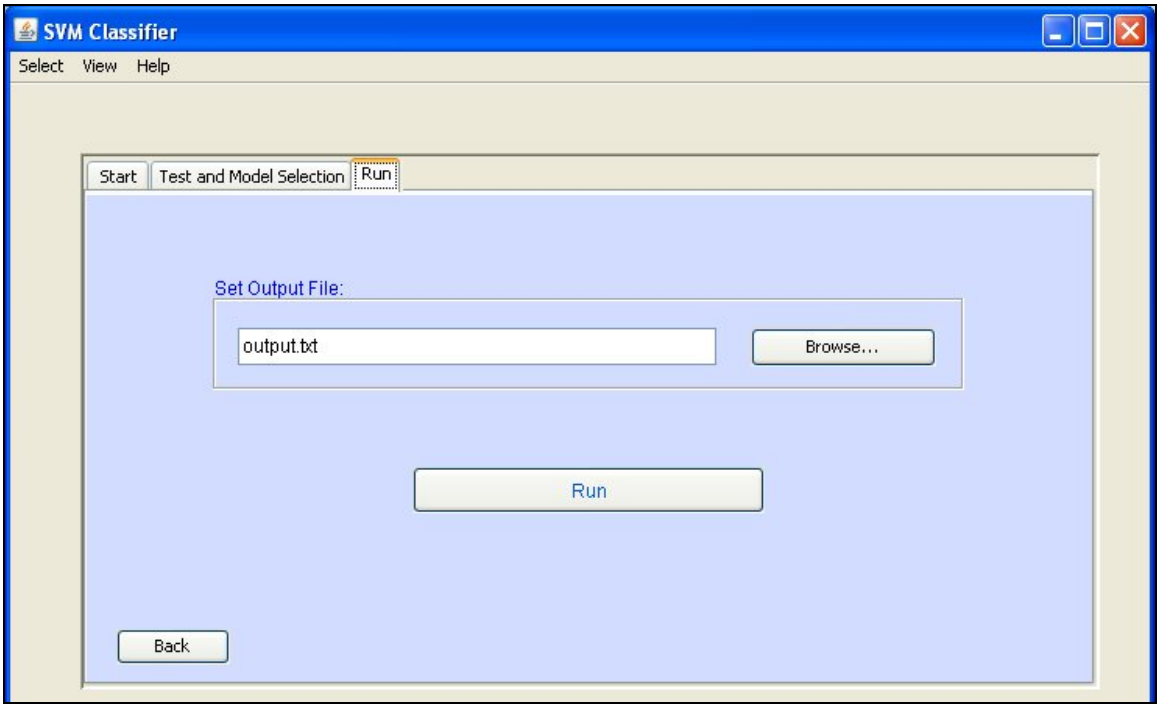
11.1.2 Πρόβλεψη (Prediction)

Αφού γίνει η κατηγοριοποίηση τότε επιστρέφεις στο πρώτο tab όπως φαίνεται στο σχήμα (1) όπου σου δίνεται η επιλογή για κατηγοριοποίηση και πρόβλεψη, και επιλέγεις το δεύτερο bullet point δηλαδή για πρόβλεψη. Τότε προχωράς στο δεύτερο tab όπως φαίνεται στο σχήμα (7) όπου πρέπει να επιλέξεις το αρχείο επαλήθευσης (και εδώ ισχύει ότι το αρχείο πρέπει να είναι της μορφής labeled όπως και με το αρχείο εκπαίδευσης). Επίσης πρέπει να επιλέξεις το model file από το σημείο όπου το είχες αποθηκεύσει προηγουμένως στο πλαίσιο κατηγοριοποίησης.



Σχήμα (7) Επιλογή του αρχείου επαλήθευσης και του model file

Αφού ολοκληρωθεί αυτή η διαδικασία προχωράς στο επόμενο tab όπου επιλέγεις το χώρο στον οποίο θα αποθεκεύσεις το output file και επιλέγεις Run για να ξεκινήσει η διαδικασία πρόβλεψης.



Σχήμα (8) Αποθηκεύεις το αρχείο εξόδου και επιλέγει "Run"

Τα αποτέλεσματα εμφανίζονται στην οθόνη που βρίσκεται από κάτω. Το ποσοστό ακρίβειας αντιστοιχεί στο ποσοστό των ακινήτων που είχαν ορθή κατηγοριοποίηση.

11.2 Μετατροπή από Delimited σε Labeled

Για την μετατροπή έχω γράψει ένα πρόγραμμα σε Java το οποίο δέχεται στην είσοδο του ένα αρχείο στη μορφή:

```
Price Enclosed Ext.     Covered Ext.      Uncovered Ext.    Age
0.125 0.071129707 0.115384615 0     0.214480874
0.214285714 0.150627615 0.153846154 0     0.194309601
0.117857143 0.071129707 0.115384615 0     0.203096539
0.096428571 0.10251046  0.134615385 0     0.123041311
0.114285714 0.190376569 0.230769231 0     0.070512821
0.155357143 0.194560669 0.144230769 0.039800995 0.100137829
0.148214286 0.20292887  0.221153846 0     0.090792181
```

Και παράγει ένα file στη μορφή:

```
0.125 1:0.071129707 2:0.115384615 4:0.214480874
0.214285714 1:0.150627615 2:0.153846154 4:0.194309601
0.117857143 1:0.071129707 2:0.115384615 4:0.203096539
0.096428571 1:0.10251046 2:0.134615385 4:0.123041311
0.114285714 1:0.190376569 2:0.230769231 4:0.070512821
0.155357143 1:0.194560669 2:0.144230769 3:0.039800995 4:0.100137829
0.148214286 1:0.20292887 2:0.221153846 4:0.090792181
0.049285714 1:0.09623431 2:0.038461538 4:0.085470085
```

Το πρόγραμμα για να τρέξει χρειάζεται το πρόγραμμα eclipse και το jdk java interface εγκατεστημένο στον υπολογιστή. Πρέπει να γίνει import από το πρόγραμμα eclipse και τα αρχεία που εισάγονται να βρίσκονται στο φάκελο όπου βρίσκεται υλοποιημένο το πρόγραμμα.

Ζητά από τον χρήστη να δώσει το όνομα του αρχείου εκπαίδευσης (το αρχείο πρέπει να είναι στο φάκελο όπου βρίσκεται υλοποιημένο το πρόγραμμα). Και αφού δώσει έγκυρο αρχείο ζητά από το χρήστη αν θέλει να θεωρήσει την κάθε τιμή ως μια ξεχωριστή κλάση ή αν θέλει να δώσει όρια τιμών (πχ κλάση Α για τιμές 10,000 – 15,000, κλάση Β για τιμές 15,000-20,000 κοκ). Τέλος αφού κάνει την επιλόγη του, το πρόγραμμα ζητά να δώσει το αρχείο επαλήθευσης. Τα αρχεία παράγονται στο φάκελο όπου βρίσκεται υλοποιημένο το πρόγραμμα με το όνομα train.txt και test.txt ανάλογα.

11.3 Source code

11.3.1 Labeled.java

```
import java.io.*;
import java.util.*;
import stdinputPack.StdInput;
import java.lang.Math;

public class Labelled {

	/*************************************************************
	* pairnei eisodo ton vector me tis grammes tou arxeiou kai
	* ektiponei tis grammes se ena arxeio "labeled.txt"
	*************************************************************/
	public static void saveLabelled(Vector<Line> text, String filename) throws IOException{

		Writer output = null;

		File file = new File(filename+"_labeled.txt");
		output = new BufferedWriter(new FileWriter(file));
		for(int i=0; i<text.size(); i++){
			// an vriskete sto teleuteo stoixeio den prosthetei "\n" sto telos
```

```
                //if(i==text.size()-1) output.write(text.elementAt(i).printLabelled());
                output.write(text.elementAt(i).printLabelled()+"\n");
            }
            output.close();
            System.out.println(" Your file has been written");
        }
        /*************************************************************
         * dinei ston xristi tin epilogi an to arxeio eisodou einai gia
         * training h testing. An einai gia training rwta to plithos
         * twn klasewn pou thelei na dimiourgithoun.
         *************************************************************/
        public static void main(String[] args)  throws IOException{


            System.out.println("\n\n Convert a Delimited file to a Labelled file\n by Irene
Georgiou, 2009 \n");

            //============================================================
            // anoigoume to file me ta dedomena
            String filename;
            System.out.println(" Give the input file: ");
            filename = StdInput.readString();
            FileReader file = new FileReader(filename);
            BufferedReader in = new BufferedReader(file);
```

```
            //=================================================================

            String[] sline;
            //int i=0;
            Vector<Line>  new_line  =  new  Vector<Line>();  //  apothikevei  to  arxeio  pou
diavastike
            Vector<String> types = new Vector<String>(); // apothikevei tis klaseis vasi tis
timis

            System.out.println(" Reading file please wait... ");

            //=================================================================
            // diavazei tin proti grammi kai tin prosperna
            // efoson den periexei xrisimes plirofories
            String line = in.readLine();

            // reads a line from the file
            while(in.ready()){

                  line = in.readLine();
                  sline =  line.split("\\s");

                  new_line.add(new Line(sline));
                  if(!types.contains(sline[0]))  types.add(sline[0]);
                  //i++;
```

```
}
//=============================================================
System.out.println(" File successfully read! \n");
System.out.println(" Number of data in file: "+new_line.size());
System.out.println(" Number of classes found: "+types.size()+"\n");

Collections.sort(types, new Comparer());
System.out.println(" Classes found: "+types+"\n");

int sel;
System.out.println(" 1. Each price is a different class");
System.out.println(" 2. Specify number of classes");
System.out.println(" Make your selection: ");

// epilogi tou xristi apo to menu
sel = StdInput.readInt();

// arithmos klasewn pou dinonte apo to xristi an i epilogi einai i 2i
int types_num=0;

//=============================================================
if(sel == 1){

     // metatrepei ta dedomena stin katallili morfi kai
```

```
        // ta ektiponei stin othoni
        for(int i=0; i<new_line.size(); i++){

              new_line.elementAt(i).toLabelled(types);
              new_line.elementAt(i).printLabelled();
        }
}
//=============================================================
else if(sel == 2){

        System.out.println(" Give number of classes: ");

        // arithmos klasewn pou dinonte apo to xristi
        types_num = StdInput.readInt(); types_num--;

        // vriskei to plithos twn antikeimenwn pou perikleiontai se mia klasi
        int items_per_type = Math.round(types.size()/(float)types_num);
        System.out.println(" Items per type: "+items_per_type);

        // metatrepei ta dedomena stin katallili morfi kai
        // ta ektiponei stin othoni
        for(int i=0; i<new_line.size(); i++){

              new_line.elementAt(i).toLabelled(types,items_per_type);
```

```
            new_line.elementAt(i).printLabelled();
        }
}
//=================================================================
// apothikevei to file me ti morfi pou theloume
saveLabelled(new_line, "train");
//=================================================================

//=================================================================
// anoigoume to file me ta dedomena

System.out.println("\n Give the testing file: ");
filename = StdInput.readString();
file = new FileReader(filename);
in = new BufferedReader(file);
//=================================================================

new_line.removeAllElements();
System.out.println(" Reading file please wait... ");

//=================================================================
// diavazei tin proti grammi kai tin prosperna
// efoson den periexei xrisimes plirofories
line = in.readLine();
```

```
// reads a line from the file
while(in.ready()){

      line = in.readLine();
      sline =  line.split("\\s");

      new_line.add(new Line(sline));
}
System.out.println(" Number of data in file: "+new_line.size());
//=============================================================
if(sel == 1){

      // metatrepei ta dedomena stin katallili morfi kai
      // ta ektiponei stin othoni
      for(int i=0; i<new_line.size(); i++){

            new_line.elementAt(i).toLabelled(types);
            new_line.elementAt(i).printLabelled();
      }
}
//=============================================================
else if(sel == 2){
```

```
                // arithmos klasewn pou dinonte apo to xristi
                //int types_num = StdInput.readInt();

                // vriskei to plithos twn antikeimenwn pou perikleiontai se mia klasi
                int items_per_type = Math.round(types.size()/(float)types_num);
                System.out.println(" Items per type: "+items_per_type);

                // metatrepei ta dedomena stin katallili morfi kai
                // ta ektiponei stin othoni
                for(int i=0; i<new_line.size(); i++){

                    new_line.elementAt(i).toLabelled(types,items_per_type);
                    new_line.elementAt(i).printLabelled();
                }
            }
            //=============================================================
            // apothikevei to file me ti morfi pou theloume
            saveLabelled(new_line, "test");
            //=============================================================
        }

}
```

11.3.2 Line.java

```
import java.util.Vector;
import java.lang.Math;

public class Line {

	String[] data;
	String[] labelled;

	Line(String[] data){
		this.data = data;
		this.labelled = new String[data.length];
	}
	public void toLabelled(Vector<String> types, int items_per_type){

		Comparer c = new Comparer();
		for(int i=0; i<data.length; i++){
			// vriskei ton tipo ston opoio anikei ena spiti analoga me tin timi
			if(i==0) {

				for(int j=0; j<types.size(); j++){
```

```
                        // an to dedomeno pou elegxoume vrisktete sti thesi j tou pinaka
me tous tipous
                        // tote vriskei ton arithmo tou tipou ston opoio anikei kai ton
apothikevei
                        if(c.compare(data[i], types.elementAt(j)) < 1){
                              //System.out.println(" position on types: "+j+" price found:
"+types.elementAt(j)+" price searching for: "+data[i]);
                              labelled[i] =
Integer.toString(Math.round(j/(float)items_per_type)+1);
                              break;
                        }
                  }
            }
            else
                  if(!data[i].equals("0")) labelled[i] = i+":"+data[i];
                  else labelled[i] = "";
                  /*if(data[i].equals("0"))
                        labelled[i] = i+":"+"0.0";
                  else labelled[i] = i+":"+data[i];*/
      }
   }
   public void toLabelled(Vector<String> types){

      for(int i=0; i<data.length; i++){
            // vriskei ton tipo ston opoio anikei ena spiti analoga me tin timi
```

```
                if(i==0) {
                      /*if(types.indexOf(data[i]) == -1) labelled[i] = "-1";
                      else labelled[i] = Integer.toString(types.indexOf(data[i]) + 1);*/
                      labelled[i] = data[i];
                }
                else
                      /*if(!data[i].equals("0")) labelled[i] = i+":"+data[i];
                      else labelled[i] = i+":"+(Math.random());*/
                      if(!data[i].equals("0")) labelled[i] = i+":"+data[i];
                      else labelled[i] = "";
                      /*
                      if(data[i].equals("0"))
                            labelled[i] = i+":"+"0.0";
                      else labelled[i] = i+":"+data[i];*/
          }
     }
     // ektiponei stin othoni mia grammi stin morfi labelled
     // kai epistrefei ti grammi pou metatrepse
     public String printLabelled(){
          String line="";
          for(int i=0; i<labelled.length; i++){
                // an vriskete sto teleuteo stoixeio den prosthetei whitespace sto telos
                if(i==labelled.length-1 || labelled[i].equals("")) line = line +
labelled[i];
```

```
                else line = line + labelled[i] + " ";
            }
            //System.out.println(line);
            return line;
      }
      public String toString(){
            String line="";
            for(int i=0; i<data.length; i++){
                  line = line + data[i] + " ";
            }
            return line;
      }
}
```

11.3.3 Type.java

```
public class Type{

      double price;
      int type;
      static int counter = 0;

      // dimioutgeitai kainourgios tipos/klasi vasi tis timis
      // i kathe klasi antistoixei se ena akeraio arithmo
```

```
	Type(double price){
		this.price = price;
		this.type = counter++;
	}
	public String toString(){
		return this.price+" -> "+this.type;
	}
	public boolean equals(Object o){
		if(this.price == (Double)o) return true;
		else return false;
	}
}
```

11.3.4 Comparer.java

```
import java.util.*;

class Comparer implements Comparator {
	public int compare(Object obj1, Object obj2)
	{
		double i1 = Double.parseDouble((String)obj1);
		double i2 = Double.parseDouble((String)obj2);
		if(i1 < i2) return -1;
		else if(i1 > i2) return 1;
		else return 0;
```

```
        }
}
```