



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Προσαρμοστική Εκμάθηση και Προσαρμοστικοί Αλγόριθμοι

Μιχαήλ Ν. Χερουβείμ

Επιβλέπων: Σέργιος Θεοδωρίδης, Καθηγητής ΕΚΠΑ

ΑΘΗΝΑ

ΙΟΥΛΙΟΣ 2015

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Προσαρμοστική Εκμάθηση και Προσαρμοστικοί Αλγόριθμοι

Μιχαήλ Ν. Χερουβείμ

A.M.: 1115201000083

Επιβλέπων: **Σέργιος Θεοδωρίδης, Καθηγητής ΕΚΠΑ**

ΠΕΡΙΛΗΨΗ

Στην παρούσα διατριβή μελετούνται το πρόβλημα της προσαρμοστικής εκμάθησης στον τομέα της Μηχανικής Εκμάθησης και οι σχετικοί προσαρμοστικοί αλγόριθμοι. Στην προσαρμοστική εκμάθηση, σκοπός είναι η μελέτη και η ανάπτυξη αλγορίθμων, οι οποίοι έχουν την δυνατότητα να εκπαιδεύονται και να μαθαίνουν, μέσα από μετρήσεις εισόδου – εξόδου, και να προσαρμόζονται στις αλλαγές του περιβάλλοντος, με στόχο την εκτίμηση αγνώστων παραμέτρων κατά τη διάρκεια του χρόνου. Το αντικείμενο αυτό εμφανίζεται σε πολλές εφαρμογές της καθημερινής ζωής, όπως η κινητή τηλεφωνία, και χαρακτηρίζει συστήματα που λειτουργούν σε πραγματικό χρόνο. Έτσι, με την πάροδο του χρόνου, αναπτύχθηκαν προσαρμοστικοί αλγόριθμοι με τους οποίους επιτυγχάνεται ικανοποιητική εκτίμηση των αγνώστων παραμέτρων. Κατά τη μελέτη προσαρμοστικών αλγορίθμων, γίνεται εύκολα αντιληπτό το γεγονός της ύπαρξης σημαντικών και εκτενών θεωρητικών υποβάθρων για την δημιουργία τους και τη λειτουργία τους. Υπάρχουν αλγόριθμοι που χρησιμοποιούνται για γραμμικά αλλά και για μη γραμμικά συστήματα, εφαρμόζοντας έξυπνες μαθηματικές τεχνικές. Στα μη γραμμικά συστήματα η διαδικασία υλοποίησης ενός αλγορίθμου είναι περισσότερο πολύπλοκη, αλλά, με την κατάλληλη προσαρμογή στα μαθηματικά μοντέλα, επιφέρει πολύ καλά αποτελέσματα στην εκτέλεσή του. Στην παρούσα διατριβή θα μελετήσουμε την φιλοσοφία των αλγορίθμων αυτών, το θεωρητικό τους υπόβαθρο, όπου θα γνωρίσουμε και θα εμβαθύνουμε σε έννοιες, όπως συνάρτηση κόστους, πυρήνες, κ.α., και φυσικά στο τέλος την πειραματική τους εκτέλεση, συγκρίνοντας τα διάφορα πειράματα των αλγορίθμων και εξαγοντας συμπεράσματα.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Μηχανική Εκμάθηση

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: προσαρμοστική εκμάθηση, προσαρμοστικοί αλγόριθμοι, γραμμικά και μη γραμμικά συστήματα, κυρτά σύνολα, πυρήνες

ABSTRACT

In the current study, we consider the problem of adaptive learning in the field of Machine Learning together the respective algorithms. In adaptive learning, the goal is to study and develop algorithms, which are capable of learning through input/output measurements, and adapt in changes of the environment, in order to estimate unknown parameters at each time instant. Such problems are encountered in a lot of practical applications, such as mobile communications, and it characterizes systems that work in real time. Thus, in the respective literature, several adaptive algorithms have been studied, that achieve a good estimation of unknown parameters. At the study of adaptive algorithms, it becomes easily perceptible the need for the existence of important and extensive theoretical backgrounds for their creation and their operation. There are many algorithms that are used for linear but also for non linear systems, applying intelligent mathematic techniques. Concerning the non linear systems, the development of adaptive algorithms is more complicated, but, with a suitable adaptation in the mathematic models, this may lead to very good results. In the current manuscript, we will study the philosophy of adaptive algorithms, their theoretical background, where we will discuss them in detail, referring concepts such as cost function, kernels, and, of course, in the end we will see their experimental implementation, comparing the various experiments of algorithms and exporting conclusions.

SUBJECT AREA: Machine Learning

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: adaptive learning, adaptive algorithms, linear and non linear systems, convex sets, kernels

ΕΥΧΑΡΙΣΤΙΕΣ

Πρωτίστως, θα ήθελα να ευχαριστήσω τον επιβλέποντά μου κ. Σέργιο Θεοδωρίδη, Καθ. του Τμήματος Πληροφορικής και Τηλεπικοινωνιών του ΕΚΠΑ, ο οποίος μου πρότεινε την ενασχόλησή μου με τον χώρο της Μηχανικής Εκμάθησης και συγκεκριμένα με το θέμα της Προσαρμοστικής Εκμάθησης. Επίσης, θα ήθελα να ευχαριστήσω, τον κ. Παντελή Μπουμπούλη, Διδ. του Τμήματος Πληροφορικής και Τηλεπικοινωνιών του ΕΚΠΑ, ο οποίος με βοήθησε στην κατανόηση των αλγορίθμων προσαρμοστικής εκμάθησης και ο συμβουλευτικός του ρόλος στην υλοποίηση των αλγορίθμων από μέρους μου ήταν πολύ σημαντικός. Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου για την διαχρονική υποστήριξή τους και τη συνεχή συμπαράστασή τους σε όλες τις βαθμίδες εκπαίδευσης μου.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ.....	11
1. ΕΙΣΑΓΩΓΗ	12
1.1 Παραμετρική Μοντελοποίηση και Εκτίμηση Παραμέτρων (Parametric Modeling and Parameter Estimation).....	12
1.2 Γραμμική Παλινδρόμηση (Linear Regression)	13
1.3 Προσαρμοστική Εκμάθηση (Online Adaptive Learning)	14
1.3.1 Υπερπροσαρμογή, Τακτοποίηση, Αραίωση (Overfitting, Regularization, Sparsification)	15
1.3.2 Γραμμική εκτίμηση μέσου τετραγωνικού σφάλματος (Mean-Square Error Linear Estimation)	16
1.3.2.1 Ο αλγόριθμος Ελαχίστων Μέσων Τετραγώνων (Least Mean Square, LMS)	17
2. ΠΡΟΣΑΡΜΟΣΤΙΚΗ ΕΚΜΑΘΗΣΗ ΣΕ ΚΥΡΤΑ ΣΥΝΟΛΑ.....	19
2.1 Τοποθέτηση Προβλήματος.....	19
2.2 Κυρτή Ανάλυση (Convex Analysis)	19
2.2.1 Κυρτά Σύνολα	19
2.2.2 Κυρτές Συναρτήσεις	20
2.3 Προβολές πάνω σε κυρτά σύνολα (Projections Onto Convex Sets, POCS).....	21
2.3.1 Προβολή πάνω σε κλειστή μπάλα (Closed ball)	22
2.4 Η θεμελιώδης θεωρία των προβολών σε κυρτά σύνολα (POCS).....	23
2.5 Η παράλληλη έκδοση της θεωρίας των προβολών σε κυρτά σύνολα (POCS)	24
2.6 Εκτίμηση παραμέτρων και εκμάθηση μηχανής με κυρτά σύνολα	25
2.6.1 Ο αλγόριθμος Προσαρμοστικής Προβαλλόμενης Υποπαραγώγων Μεθόδου (Adaptive Projected Subgradient Method, APSM)	26
2.6.2 Σύγκλιση του APSM αλγορίθμου	27
3. ΠΡΟΣΑΡΜΟΣΤΙΚΗ ΕΚΜΑΘΗΣΗ ΣΕ ΜΗ ΓΡΑΜΜΙΚΑ ΣΥΣΤΗΜΑΤΑ.....	29
3.1 Τοποθέτηση Προβλήματος.....	29
3.2 Η διαδικασία της παλινδρόμησης (regression) σε μη γραμμικά συστήματα	29
3.3 Αναπαραγωγή Πυρήνα σε Hilbert χώρους (Reproducing Kernel Hilbert Spaces, RKHS)	31
3.3.1 Παραδείγματα Συναρτήσεων Πυρήνα	32
3.3.2 Representer Θεώρημα	34

3.4 Εκμάθηση σε Απευθείας Σύνδεση σε RKHS (Online Learning in RKHS)	34
3.4.1 Ο αλγόριθμος Kernel Least Mean Squares (KLMS)	35
3.4.2 Ο αλγόριθμος Kernel Adaptive Projected Subgradient Method (KAPSM)	36
3.4.3 Αραίωση της λύσης (Sparsifying the solution)	37
3.4.3.1 Αραίωση στον KLMS	37
3.4.3.2 Αραίωση στον KAPSM.....	38
4. ΠΕΙΡΑΜΑΤΙΚΗ ΕΚΜΑΘΗΣΗ ΠΡΟΣΑΡΜΟΣΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ	40
4.1 Τοποθέτηση Προβλήματος	40
4.2 Ταύτιση Συστήματος (System Identification).....	40
4.3 Ισοστάθμιση Καναλιού (Channel Equalization).....	41
4.4 Ταύτιση Καναλιού στην προσαρμοστική εκμάθηση και πειραματική μελέτη των προσαρμοστικών αλγορίθμων NLMS, APSM.....	42
4.5 Ισοστάθμιση Καναλιού στην προσαρμοστική εκμάθηση και πειραματική μελέτη των προσαρμοστικών αλγορίθμων KNLMS, KAPSM	49
ΣΥΜΠΕΡΑΣΜΑΤΑ.....	54
ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ	55
ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ.....	56
ΠΑΡΑΡΤΗΜΑ Ι.....	57
ΑΝΑΦΟΡΕΣ	65

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1.1: Η συνάρτηση κόστους μέσου τετραγωνικού σφάλματος έχει παραβολική μορφή.....	17
Σχήμα 2.1: α) Η l_2 -ball ακτίνας 1 περιλαμβάνει όλα τα σημεία με Ευκλείδια νόρμα μικρότερη ή ίσης του 1. β) Η l_1 -ball περιλαμβάνει όλα τα σημεία με νόρμα μικρότερη ή ίση του 1.	20
Σχήμα 2.2: Ο γραμμικός συνδυασμός των σημείων $(x_1, f(x_1))$, $(x_2, f(x_2))$	20
Σχήμα 2.3: Η προβολή x^* του x πάνω στο επίπεδο	21
Σχήμα 2.4: Η γεωμετρική αναπαράσταση της χαλαρής προβολής.....	22
Σχήμα 2.5: Η προβολή πάνω σε κλειστή μπάλα ακτίνας δ και κέντρου 0.....	22
Σχήμα 2.6: Η γεωμετρική αναπαράσταση του θεωρήματος.....	24
Σχήμα 2.7: Η γεωμετρική αναπαράσταση της παράλληλης εκδοσης των POCS	24
Σχήμα 2.8: Η υπερπλάκα που ορίζεται από τα σημεία εκπαίδευσης (y_n, x_n)	25
Σχήμα 2.9: Τη χρονική στιγμή n , $q=2$ υπερπλάκες έχουν σχηματιστεί, $S_{\varepsilon, n}$, $S_{\varepsilon, n-1}$, το θ_{n-1} είναι ταυτόχρονα προβεβλημένο και στις δυο και οι προβολές είναι κυρτά συνδεδεμένες. Ο επόμενος υπολογισμός είναι το θ_n και έπειτα σχηματίζεται η υπερπλάκα $S_{\varepsilon, n+1}$ κ.ο.κ. Κάθε στιγμή ο υπολογισμός πλησιάζει την τομή.....	27
Σχήμα 3.1: Προσαρμογή του συνόλου των δεδομένων εκπαίδευσης, (y_n, x_n) , $n=1, 2, \dots, N-1$ από μια γραμμική και μια τετραγωνική συνάρτηση αντίστοιχα.....	30
Σχήμα 3.2: Η μη γραμμική διαδικασία στον αυθεντικό μικρής διάστασης χώρο απεικονίζεται σε γραμμική σε ένα υψηλότερης διάστασης χώρο RKHS H	32
Σχήμα 4.1: Στην ταύτιση συστήματος, η απόκριση του μοντέλου εκτιμάται βέλτιστα, ώστε η έξοδος να είναι κοντά, όσον αφορά το MSE, στο αντίστοιχο του αγνώστου έργου. Η κόκκινη γραμμή δείχνει ότι το σφάλμα χρησιμοποιείται για τη βέλτιστη εκτίμηση των αγνώστων παραμέτρων του φίλτρου	40
Σχήμα 4.2: Ένας ισοσταθμιστής, ο οποίος ανακτά τη μεταδιδόμενη πληροφορία.....	42
Σχήμα 4.3: MSE απόδοση για το πρώτο πείραμα, με $q=30$ στον APSM.....	43
Σχήμα 4.4: MSE απόδοση για το δεύτερο πείραμα, με $q=5$ στον APSM.....	44
Σχήμα 4.5: MSE απόδοση για το δεύτερο πείραμα, με $q=50$ στον APSM	44

Σχήμα 4.6: MSE απόδοση για το δεύτερο πείραμα, με $q=100$ στον APSM.....	45
Σχήμα 4.7: MSE απόδοση για το τρίτο πείραμα με $\mu_n=0.2M_n$	45
Σχήμα 4.8: MSE απόδοση για το τρίτο πείραμα με $\mu_n=1.2M_n$	46
Σχήμα 4.9: MSE απόδοση για το τρίτο πείραμα με $\mu_n=1.9M_n$	46
Σχήμα 4.10: Όσο μικρότερη είναι η παράμετρος μ στον NLMS, τόσο πιο αργή είναι η σύγκλιση, με το μέσο τετραγωνικό σφάλμα να είναι μικρότερο. α) $\mu=0.5$, β) $\mu=0.8$	47
Σχήμα 4.11: α) MSE σε dB ($\sigma=0.1$), β) MSE σε dB ($\sigma=0.5$). Με την αύξηση της συνεισφοράς του θορύβου μεγαλώνει αρκετά η τιμή του σφάλματος σε σχέση με την αρχική μέτρηση.....	47
Σχήμα 4.12: MSE απόδοση για α) $N_w=200, q=20, N=10000$, β) $N_w=300, q=30, N=10000$, γ) $N_w=400, q=40, N=10000$, δ) $N_w=500, q=50, N=10000$, ε) $N_w=1000, q=100, N=20000$. Με τη αύξηση της διάστασης και την αντίστοιχη αύξηση στην παράμετρο q διαπιστώνουμε ότι η σύγκλιση γίνεται πιο αργή αλλά και μεγαλώνει η τιμή του σφάλματος και στις δυο μεθόδους	48
Σχήμα 4.13: MSE απόδοση για το πρώτο πείραμα με προβολή πάνω σε κλειστή σφαίρα ακτίνας $\delta=10$	50
Σχήμα 4.14: MSE απόδοση για το πρώτο πείραμα με novelty criterion	50
Σχήμα 4.15: MSE απόδοση για το δεύτερο πείραμα με προβολή πάνω σε κλειστή σφαίρα ακτίνας $\delta=10$ για α) $q=2$, β) $q=4$, γ) $q=8$ και δ) $q=16$	51
Σχήμα 4.16: MSE απόδοση για το δεύτερο πείραμα με novelty criterion για α) $q=2$, β) $q=4$, γ) $q=8$ και δ) $q=16$	51
Σχήμα 4.17: MSE απόδοση για το τρίτο πείραμα με προβολή σε κλειστή μπάλα ακτίνας $\delta=10$ για α) $\mu_n = 0.3M_n$, β) $\mu_n = 1M_n$, γ) $\mu_n = 1.9M_n$	52
Σχήμα 4.18: MSE απόδοση για το τρίτο πείραμα με novelty criterion για α) $\mu_n = 0.3M_n$, β) $\mu_n = 1M_n$, γ) $\mu_n = 1.9M_n$	53

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 2.1: Ο APSM αλγόριθμος.....	26
------------------------------------	----

ΠΡΟΛΟΓΟΣ

Η παρούσα εργασία αποτελεί υποχρεωτικό, για τη λήψη του Πτυχίου, κομμάτι των σπουδών μου στο τμήμα Πληροφορικής και Τηλεπικοινωνιών του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών. Η εργασία εκπονήθηκε κατά το εαρινό εξάμηνο (Φεβρουάριος 2015 – Ιούνιος 2015) υπό την επίβλεψη του Καθ. Σέργιου Θεοδωρίδη.

1. Εισαγωγή

Η Μηχανική Εκμάθηση (Machine Learning) είναι ένα αντικείμενο της επιστήμης των υπολογιστών που παρουσιάζει συνεχή ανάπτυξη με την πάροδο των ετών. Εμπεριέχεται στη μελέτη της αναγνώρισης προτύπων και ασχολείται με την κατασκευή αλγορίθμων, οι οποίοι υποβάλλονται σε εκμάθηση με βάση κάποια δεδομένα και αποκτούν την ικανότητα να εξάγουν προβλέψεις, αποτελέσματα και συμπεράσματα κατά την εκτέλεσή τους.

Αντικείμενο του κλάδου της μηχανικής εκμάθησης αποτελεί η παραμετρική μοντελοποίηση (Parametric modeling), η οποία αποτελείται από ένα ευρύ φάσμα μελέτης που σχετίζεται με την εργασία της εκμάθησης, όταν παραμετρικά μοντέλα καλούνται να περιγράψουν διαθέσιμα δεδομένα. Ένα μεγάλο τμήμα προβλημάτων παραμετρικής μοντελοποίησης καταλήγει σε μια λειτουργία ύστερα από κάποια προσεγγιστική εργασία. Κατά τη λειτουργία αυτή συλλέγονται πληροφορίες από διαθέσιμα δεδομένα κατά την φάση της εκπαίδευσης, με αποτέλεσμα να γίνεται εφικτή η πρόβλεψη των τιμών εξόδου με βάση τις μετρήσεις που γίνονται από τις τιμές εισόδου. Έτσι, μετέπειτα μπορεί να πραγματοποιηθεί η λήψη αποφάσεων. Δύο είναι οι κύριες εργασίες της μηχανικής εκμάθησης. Η παλινδρόμηση (Regression), στην οποία βασίζεται η εκπόνηση της εργασίας, και η ταξινόμηση (Classification). Επιπλέον, είναι πολύ σημαντική η μελέτη των διάφορων αλγορίθμων μοντελοποίησης, της εξαγωγής συμπερασμάτων, έπειτα από την εκτέλεσή τους, καθώς και τα αποτελέσματα των διάφορων πειραμάτων.

1.1 Παραμετρική Μοντελοποίηση και Εκτίμηση Παραμέτρων (Parametric Modeling and Parameter Estimation)

Η παραμετρική μοντελοποίηση αναφέρεται στη διαδικασία κατά την οποία γίνεται εκτίμηση μιας συναρτησιακής εξάρτησης μεταξύ κάποιων μεταβλητών εισόδου και μεταβλητών εξόδου. Συχνά, αυτή η συνάρτηση έχει τη μορφή συγκεκριμένου τύπου, όπως μια γραμμική συνάρτηση ή μια τετραγωνική, και περιγράφεται από ένα σύνολο αγνώστων παραμέτρων. Η εργασία εκτίμησης της τιμής ενός αγνώστου διανύσματος παραμέτρων, θ , αποτελεί μια ενδιαφέρουσα διαδικασία σε πολλές εφαρμογές.

Δοθέντος ενός συνόλου δεδομένων, πρέπει να βρεθεί μία επιφάνεια που “ικανοποιεί” τα δεδομένα. Για να συμβεί αυτό, υιοθετείται μια μορφή συνάρτησης, μια γραμμική ή μια τετραγωνική όπως προαναφέρθηκε, με σκοπό να υπολογιστούν οι άγνωστοι συντελεστές, ώστε το γράφημα να ικανοποιεί τα δεδομένα και την κατανομή τους στο χώρο όσο πιο κοντά γίνεται. Τα δεδομένα βρίσκονται στον δισδιάστατο χώρο των πραγματικών αριθμών $\mathbb{R}^2$ και έχουν τη μορφή (y_n, x_n) , $n=1,2,\dots,N$. Για την περίπτωση της γραμμικής συνάρτησης, έχουμε

$$y = f_{\theta}(x) = \theta_0 + \theta_1 x,$$

και για την τετραγωνική

$$y = f_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2.$$

Τα άγνωστα διανύσματα παραμέτρων είναι $\theta = [\theta_0, \theta_1]^T$ και $\theta = [\theta_0, \theta_1, \theta_2]^T$, αντίστοιχα. Και στις δυο περιπτώσεις, η εργασία αποτελείται από δυο βήματα: α) ορίζεται η μορφή της συνάρτησης και β) εκτιμούνται οι τιμές των αγνώστων παραμέτρων, ώστε να επιτευχθεί μια καλή κατανομή στο χώρο. Στην πιο γενική μορφή, δοθέντος ενός συνόλου δεδομένων από σημεία (y_n, x_n) , $y_n \in \mathbb{R}$, $x_n \in \mathbb{R}^l$, $n=1,2,\dots,N$ και ενός παραμετρικού συνόλου συναρτήσεων,

$$F := \{f_\theta(\cdot) : \theta \in A \subseteq \mathbb{R}^K\},$$

βρίσκουμε μια συνάρτηση στο F , η οποία υποδηλώνεται σαν $f(\cdot) := f_{\theta_*}(\cdot)$, όπου δοθέντος μιας τιμή $x \in \mathbb{R}^l$, η $f(x)$ προσεγγίζει καλύτερα την αντίστοιχη τιμή $y \in \mathbb{R}$. Η τιμή θ_* είναι το αποτέλεσμα από την υπολογιστική διαδικασία. Καθώς έχει οριστεί ένα παραμετρικό σύνολο συναρτήσεων, F , χρειάζεται να γίνει η εκτίμηση των αγνώστων παραμέτρων. Μια μέθοδος σχετικά με αυτή τη διαδικασία είναι ο ορισμός μιας συνάρτησης κόστους, η οποία προσδιορίζει το σφάλμα ανάμεσα στην υπολογισμένη τιμή του y και στην πρόβλεψη που γίνεται, με βάση την αντίστοιχη μέτρηση του x . Ορίζουμε λοιπόν μια μη αρνητική συνάρτηση κόστους

$$L(\cdot, \cdot) : \mathbb{R} \times \mathbb{R} \rightarrow [0, \infty),$$

και υπολογίζουμε το θ_* , ώστε να ελαχιστοποιήσουμε το συνολικό σφάλμα. Μια χαρακτηριστική συνάρτηση κόστους είναι αυτή των ελαχίστων τετραγώνων (least-squares loss function, LS)

$$L(y, f_\theta(x)) = (y - f_\theta(x))^2.$$

Η γραμμικότητα που παρουσιάζει η συνάρτηση κόστους ελαχίστων τετραγώνων απλοποιεί τους αλγεβρικούς υπολογισμούς και αποτελεί το βασικό παράδειγμα με το οποίο μπορεί να γίνει πιο εύκολα κατανοητή η ιδέα της εκτίμησης παραμέτρων. Φυσικά, υπάρχουν και μη γραμμικές διαδικασίες εκτίμησης, οι οποίες αντιμετωπίζονται με μετατροπή του μη γραμμικού προβλήματος σε γραμμικό. Στο παρών κεφάλαιο θα ασχοληθούμε μόνο με γραμμικές διαδικασίες και γραμμικά συστήματα εκτίμησης παραμέτρων.

1.2 Γραμμική παλινδρόμηση (Linear Regression)

Ο όρος Παλινδρόμηση (Regression) επινοήθηκε για να ορίσει την εργασία της μοντελοποίησης της σχέσης μια τυχαίας εξαρτημένης μεταβλητής, y , η οποία θεωρείται ότι είναι η απόκριση ενός συστήματος, όταν αυτό ενεργοποιείται από ένα σύνολο τυχαίων μεταβλητών, $x_1, x_2, \dots, x_l$, οι οποίες παρουσιάζονται ως τα στοιχεία ενός ισοδύναμου τυχαίου διανύσματος x [1:αναφορά chapter3 στο linear regression]. Μοντελοποιείται, λοιπόν, μια σχέση στην οποία προστίθεται και ο θόρυβος, η , που είναι μια τυχαία μεταβλητή. Στόχος της παλινδρόμησης είναι να εκτιμηθεί το διάνυσμα παραμέτρων, θ , δοθέντος ενός συνόλου μετρήσεων, (y_n, x_n) , $n=1,2,\dots,N$. Η εξαρτημένη μεταβλητή είναι γνωστή ως μεταβλητή εξόδου και το διάνυσμα x ως διάνυσμα εισόδου ή regressor. Το μοντέλο της παλινδρόμησης, επομένως, γράφεται ως,

$$y = \theta^T x + \eta.$$

Στη γραμμική παλινδρόμηση ορίζουμε το ακόλουθο μοντέλο πρόβλεψης

$$\hat{y} = \hat{\theta}_0 + \hat{\theta}_1 x_1 + \dots + \hat{\theta}_l x_l := \hat{\theta}^T x \quad \text{ή} \quad \hat{y} = \hat{\theta}_0 + \sum_{l=1}^l \hat{\theta}_l x_l \quad (1.1)$$

Χρησιμοποιώντας την συνάρτηση κόστους ελαχίστων τετραγώνων (LS loss function), υπολογίζεται το $\hat{\theta}$, που ελαχιστοποιεί τη τετραγωνική διαφορά του $\hat{y}_n$ με το y_n . Έτσι, για να ελαχιστοποιήσουμε, σε σχέση με το θ , τη συνάρτηση κόστους,

$$J(\theta) = \sum_{n=1}^N \psi_n - \theta^T x_n \quad (1.2)$$

παίρνουμε την μερική παράγωγο της (1.2) ως προς θ , και την εξισώνουμε με το διάνυσμα 0. Τελικά, καταλήγουμε στην παρακάτω εξίσωση

$$\sum_{n=1}^N x_n x_n^T \hat{\theta} = \sum_{n=1}^N x_n y_n \quad (1.3)$$

Ένας άλλος τρόπος να αναπαραστήσουμε την παραπάνω εξίσωση είναι μέσω πινάκων. Έστω ένας πίνακας εισόδου, X , με διαστάσεις $N \times (l+1)$, ο οποίος έχει σαν γραμμές τα διανύσματα εισόδου (regressor vectors), x_n^T , $n=1,2,\dots,N$, και αναπαριστάται με τον παρακάτω τρόπο

$$X := \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1l} & 1 \\ x_{21} & \cdots & x_{2l} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ x_{N1} & \cdots & x_{Nl} & 1 \end{bmatrix}$$

Τότε η (1.3) μπορεί επίσης να γραφεί

$$X^T X \hat{\theta} = X^T y, \quad \text{όπου } y := [y_1, y_2, \dots, y_N]^T$$

και η εκτίμηση της συνάρτησης κόστους LS είναι

$$\hat{\theta} = (X^T X)^{-1} X^T y, \quad \text{αρκεί να υπάρχει ο } (X^T X)^{-1}.$$

Συνεπώς, η εκτίμηση του παραμετρικού διανύσματος δίνεται από ένα γραμμικό σύνολο εξισώσεων. Η λύση που μόλις δόθηκε είναι μοναδική, γεγονός που επεξηγεί μια παραβολική μορφή στο γράφημα της συνάρτησης κόστους ελαχίστων τετραγώνων.

1.3 Προσαρμοστική εκμάθηση (Online Adaptive Learning)

Για τη διαδικασία υπολογισμού παραμέτρων υπάρχουν αρκετοί τρόποι και μέθοδοι για την επίλυση ενός προβλήματος μηχανικής εκμάθησης. Οι πιο αποδοτικές μέθοδοι είναι αυτές που παρουσιάζουν τη βελτιστοποίηση του προβλήματος αναδρομικά. Το κύριο χαρακτηριστικό αυτής της μεθόδου είναι ότι ενώ ανανεώνεται ο υπάρχων υπολογισμός, κάθε φορά λαμβάνεται και μια καινούρια μέτρηση. Έτσι, γίνεται αντικείμενο μελέτης ο νέος υπολογισμός των παραμέτρων και τα προηγούμενα διαθέσιμα δεδομένα δεν χρειάζονται πια. Όλη η πληροφορία βρίσκεται πλέον στον καινούριο υπολογισμό, που στη συνέχεια θα χρησιμοποιηθεί σαν δεδομένο, με σκοπό έναν άλλο υπολογισμό που θα οδηγήσει σε βελτιστοποίηση. Αυτό το μοντέλο εκτίμησης παραμέτρων καλείται προσαρμοστικό (adaptive). Ο όρος προσαρμοστική εκμάθηση χρησιμοποιείται στην επεξεργασία σήματος και τα μοντέλα αυτά έχουν εφαρμοσθεί ευρέως στις Επικοινωνίες (Communications), στη Θεωρία συστημάτων (System

Theory) και στον Αυτόματο Έλεγχο (Automatic Control) με τη μορφή αλγορίθμων όπως ο LMS, RLS, κλπ [2-4].

Η ανάγκη που οδήγησε στη δημιουργία των μοντέλων αυτών ήταν ότι οι μηχανισμοί εκπαίδευσης έπρεπε να είναι ικανοί να δέχονται τις αλλαγές του εκπαιδευτικού περιβάλλοντος και να ξεχνούν τα δεδομένα του παρελθόντος. Έτσι, με αυτόν τον τρόπο ήρθε εις πέρας η δημιουργία μοντέλων προσαρμοστική εκμάθησης που προσαρμόζονται συνεχώς σε καινούριες καταστάσεις. Οι μηχανισμοί αυτοί είναι δημοφιλείς σε πολλές εφαρμογές και ιδιαίτερα σε περιπτώσεις όπου υπάρχουν μεγάλες βάσεις δεδομένων με τεράστιο αριθμό δεδομένων εκπαίδευσης, τα οποία δεν μπορούν να αποθηκευτούν και πρέπει να μελετώνται εκείνη την στιγμή. Συνεπώς, για την εφαρμογή των μοντέλων αυτών, έχουν προταθεί αρκετοί αλγόριθμοι που ικανοποιούν συγκεκριμένες συνθήκες μοντελοποίησης, των οποίων τα αποτελέσματα δείχνουν ότι οδηγούν στη βελτιστοποίηση του προβλήματος. Παρακάτω θα μεετήσουμε αλγορίθμους που εφαρμόζονται σε γραμμικά συστήματα.

1.3.1 Υπερπροσαρμογή, Τακτοποίηση, Αραίωση (Overfitting, Regularization, Sparsification)

Πριν ξεκινήσουμε να εισχωρούμε περισσότερο στην έννοια της προσαρμοστικής εκμάθησης και σε κάποια πραδείγματα προσαρμοστικών αλγορίθμων που λειτουργούν αποτελεσματικά, είναι αναγκαίο να αναφερθεί το πρόβλημα της υπερπροσαρμογής (overfitting). Το πρόβλημα αυτό χρίζει ιδιαίτερης προσοχής και είναι αρκετά σημαντικό σε οποιοδήποτε πρόβλημα μηχανικής εκμάθησης, διότι σχετίζεται με την πολυπλοκότητα του υιοθετημένου μοντέλου προσαρμοστικής εκμάθησης. Ο όρος πολυπλοκότητα σχετίζεται συχνά με τον αριθμό των αγνώστων παραμέτρων. Υπάρχουν όμως και περιπτώσεις όπου η πολυπλοκότητα είναι μια πιο γενική ιδιότητα, καθώς μπορεί ένα μοντέλο να έχει εκπαιδευτεί σε χώρους άπειρης διάστασης με πεπερασμένο αριθμό δεδομένων εκπαίδευσης, τα οποία όμως μπορεί να τα υποστηρίξει πολύ καλά.

Συγκεκριμένα, αν το μοντέλο είναι αρκετά πολύπλοκο, σε σχέση με τον αριθμό των δεδομένων εκπαίδευσης, N , τότε τείνει να μάθει πολλά περισσότερα από ό,τι γνωρίζει, καθώς εκπαιδεύεται στο συγκεκριμένο σύνολο δεδομένων εκπαίδευσης. Για παράδειγμα, στην διαδικασία της παλινδρόμησης (regression), τείνει να μάθει όχι μόνο τις χρήσιμες πληροφορίες από τα δεδομένα εκπαίδευσης αλλά και επηρεάζεται αρκετά και από τη συνεισφορά του θορύβου. Επομένως, αν το μοντέλο είναι πολύ πολύπλοκο, τείνει να κάνει πολύ καλή προσαρμογή στα δεδομένα του συνόλου εκπαίδευσης, αλλά μειονεκτεί στο γεγονός πως δεν μπορεί να αντιμετωπίσει τα δεδομένα που δεν βρίσκονται στα σύνολα εκπαίδευσης.

Αναφέρθηκε, λοιπόν, τι συμβαίνει στην περίπτωση όπου το μοντέλο έχει πολύ μεγάλη πολυπλοκότητα. Αν όμως έχει μικρή πολυπλοκότητα, είναι δηλαδή απλό, τι συμβαίνει; Είναι φανερό ότι αν είναι απλό, τότε δεν έχει την ικανότητα να απορροφά τις χρήσιμες πληροφορίες από τα δεδομένα εκπαίδευσης, με αποτέλεσμα να οδηγεί σε πολύ φτωχές προβλέψεις.

Ένας τρόπος να αντιμετωπιστεί το πρόβλημα της υπερπροσαρμογής είναι η μέθοδος της τακτοποίησης (regularization). Είναι ένα μαθηματικό εργαλείο, το οποίο φροντίζει ταυτόχρονα α) να διατηρεί τη συνεισφορά της συνάρτησης κόστους όσο πιο μικρή γίνεται, γεγονός που καθορίζει την ακρίβεια του υπολογισμού, και β) να διατηρεί το μοντέλο όσο λιγότερο πολύπλοκο γίνεται. Αυτό επιτυγχάνεται προσθέτοντας έναν ακόμη όρο στη συνάρτηση κόστους, με στόχο να διατηρήσει τη νόρμα των παραμέτρων όσο πιο μικρή γίνεται, πχ.,

$$J(\theta) := \sum_{n=1}^N L(y_n, f_{\theta}(x_n)) + \lambda \Omega(\|\theta\|) \quad (1.4)$$

όπου $\Omega(\cdot)$ είναι μια γνησίως αύξουσα μη αρνητική συνάρτηση. Η παράμετρος λ καλείται παράμετρος τακτοποίησης (regularization parameter) και ελέγχει τη σχέση των δυο όρων στη συνάρτηση κόστους. Πολύ συχνά το Ω επιλέγεται ως,

$$\Omega(\|\theta\|) = \sum_{k=0}^{K-1} |\theta_k|^2, \text{ ή } \Omega(\|\theta\|) = \sum_{k=0}^{K-1} |\theta_k|.$$

Χρησιμοποιώντας τη σχέση (1.1) και προσθέτοντας τον σχετικό όρο στη συνάρτηση κόστους, μετά την ελαχιστοποίηση και εξαιτίας της παρουσίας της νόρμας, μερικά στοιχεία του διανύσματος θ ωθούνται προς το μηδέν. Οι όροι που ωθούνται προς το μηδέν είναι αυτοί των οποίων η εξαλειψή επηρεάζει λιγότερο τον όρο της συνάρτησης κόστους. Αυτό το γεγονός οδηγεί σε λιγότερο πολύπλοκα μοντέλα.

Η αραίωση (sparsification) είναι ένας άλλος όρος που δείχνει ότι η μέθοδος εκτίμησης που υιοθετήθηκε, έχει τη δυνατότητα να ωθήσει τα λιγότερο σημαντικά στοιχεία, από θέμα ακρίβειας, στο μηδέν. Έτσι, αυτός είναι ένας ακόμη τρόπος να αντιμετωπιστεί η υπερπροσαρμογή.

1.3.2 Γραμμική Εκτίμηση μέσου τετραγωνικού σφάλματος (Mean-Square Error Linear Estimation)

Η γραμμική εκτίμηση μέσου τετραγωνικού σφάλματος αποτελεί μια θεματική ενότητα υψηλής σημασίας για την εκτίμηση παραμέτρων. Επίσης, αποτελεί ένα θεμέλιο λίθο στην κατανόηση των μοντέλων της προσαρμοστικής μάθησης και είναι απαραίτητη η γνώση του. Σε συνέχεια της εργασίας εκτίμησης που αναπτύχθηκε στο παρόν κεφάλαιο, η βέλτιστη MSE λύση του y , δοθέντος της τιμής $x = x$ είναι

$$\hat{y} = E[y|x],$$

όπου $E[\cdot]$ δηλώνεται η αναμενόμενη τιμή (expected value). Η συγκεκριμένη συνάρτηση είναι μη γραμμική, οπότε θα επικεντρωθούμε σε μια γραμμική συνάρτηση. Έστω $(y, x) \in \mathbb{R}^1 \times \mathbb{R}^l$ είναι δυο τυχαίες οντότητες μηδενικής μέσης τιμής. Στόχος είναι να υπολογιστεί ένα $\theta \in \mathbb{R}^l$ στο γραμμικό μοντέλο

$$\hat{y} = \theta^T x, \quad (1.5)$$

ώστε η συνάρτηση κόστους,

$$J(\theta) = E[(y - \hat{y})^2], \quad (1.6)$$

να ελαχιστοποιείται. Συνεπώς, ο βέλτιστος παράγοντας επιλέγεται έτσι ώστε να ελαχιστοποιείται η διαφορά του σφάλματος των τυχαίων μεταβλητών

$$e = y - \hat{y} \quad (1.7)$$

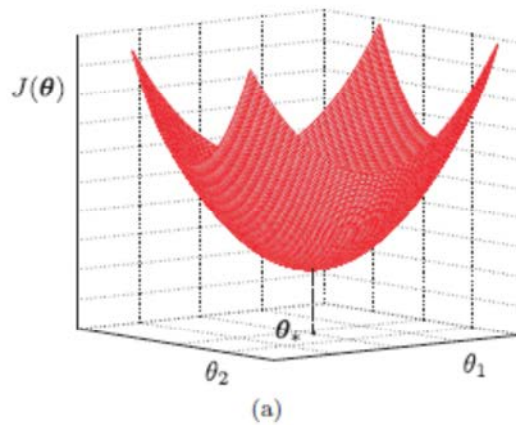
Ελαχιστοποιούμε τη συνάρτηση κόστους $J(\theta)$, παίρνοντας την μερική παράγωγο ως προς θ και εξισώνοντας με το 0,

$$\begin{aligned} \nabla J(\theta) &= \nabla E[(y - \theta^T x)(y - x^T \theta)] \\ &= \nabla \{E[y^2] - 2\theta^T E[xy] + \theta^T E[xx^T] \theta\} \\ &= -2p + 2\Sigma_x \theta = 0 \end{aligned}$$

ή

$$\Sigma_x \theta^* = p, \quad (1.8)$$

όπου $p = E[xy]$ το διάνυσμα ετεροσυσχέτισης μεταξύ εισόδου και επιθυμητής εξόδου και ο πίνακας $\Sigma = E[xx^T]$ το μητρώο αυτοσυσχέτισης.



Σχήμα 1.1: Η συνάρτηση κόστους μέσου τετραγωνικού σφάλματος έχει παραβολική μορφή

1.3.2.1 Ο αλγόριθμος Ελαχίστων Μέσων Τετραγώνων (Least Mean Square, LMS)

Ο αλγόριθμος Least Mean Square (LMS) βασίζεται στο κριτήριο ελαχίστων τετραγώνων και αποτελεί μέρος μιας επιτυχημένης γκάμας αλγορίθμων που ασχολούνται με προσαρμοστική μηχανική εκμάθηση σε κάθε χρονική στιγμή (online adaptive learning). Η εκτίμησή του βασίζεται αποκλειστικά στις μετρήσεις οι οποίες θα λαμβάνονται ακολουθιακά, δηλαδή ένα ζεύγος μετρήσεων κάθε χρονική στιγμή. Έστω ένα σύνολο δεδομένων (x_n, y_n) , όπου $x_n \in \mathbb{R}^l$ και $y_n \in \mathbb{R}$, $n=1,2,\dots,N$ και μια παραμετρική κλάση συναρτήσεων $F = \{f_\theta(\cdot) : \theta \in A \subseteq \mathbb{R}^k\}$. Ο στόχος της τυπικής διαδικασίας της προσαρμοστικής εκμάθησης είναι να συμπεράνουμε μια σχέση εισόδου - εξόδου f_θ από τη κλάση F , βασιζόμενη στα διαθέσιμα δεδομένα, ώστε να ελαχιστοποιηθεί μια συνάρτηση κόστους $L(\theta)$, που υπολογίζει το σφάλμα μεταξύ της εξόδου y με την έξοδο που υπολογίσθηκε $\hat{y}$, τη χρονική στιγμή n . Στη βασική του μορφή ο αλγόριθμος LMS ορίζει το σύνολο των γραμμικών συναρτήσεων,

$$F = \{f_\theta(x) = \theta^T x : \theta \in \mathbb{R}^l\},$$

και το μέσο τετραγωνικό σφάλμα,

$$L(\theta) = E[|y_n - \theta^T x|^2], \text{ ως συνάρτηση κόστους.}$$

Παίρνοντας την παράγωγο της συνάρτησης κόστους και για κάθε στιγμή $n=1,2,\dots,N$, έχουμε

$$\begin{aligned} \nabla L(\theta) &= -2E[(y_n - \theta^T x_n)x_n] \\ &\approx -2(y_n - \theta^T x_n)x_n = \nabla L_n(\theta), \text{ όπου} \end{aligned}$$

$$L_n(\theta) := (y_n - \theta^T x_n)^2, n=1,2,\dots,N.$$

Αν ορίσουμε το σφάλμα $e_n = y_n - \theta_{n-1}^T x_n$, όπου θ_{n-1} είναι ο παρών υπολογισμός, τότε η ανανέωση στο επόμενο βήμα θα ισούται με

$$\begin{aligned}\theta_n &= \theta_{n-1} - \mu \nabla L_n(\theta_{n-1}) \\ &= \theta_{n-1} + \mu e_n x_n, n=1,2,\dots\end{aligned}\quad (1.9)$$

Υποθέτοντας ότι $\theta_0=0$, τότε η συνεχόμενη εφαρμογή της (1.9) μας δίνει τον παρακάτω τύπο

$$\theta_n = \mu \sum_{i=1}^n e_i x_i, n=1,2,\dots$$

και η τιμή εξόδου συστήματος εκμάθησης σε κάθε στιγμή n γίνεται

$$\hat{y}_n = f_{\theta_{n-1}}(x_n) = \mu \sum_{i=1}^{n-1} e_i x_i^T x_n, n=1,2,\dots$$

Υπάρχουν πολλά κριτήρια σύγκλισης για τον αλγόριθμο LMS [3], από τα οποία τα πιο δημοφιλή είναι τα εξής:

- Η σύγκλιση κατά μέση τιμή στη λύση Wiener[5]: $E[e_n] \rightarrow 0$, καθώς $n \rightarrow \infty$, όπου $e_n = \theta_n - \theta^*$ και θ^* είναι η Wiener λύση.
- Η σύγκλιση κατά το μέσο τετραγωνικό σφάλμα: Αν η σχέση εισόδου – εξόδου είναι $y_n = \theta^{*T} x_n + \eta_n$, τότε $L[\theta_{n-1}] = E[|e_n|^2] \rightarrow$ σταθερή τιμή, καθώς $n \rightarrow \infty$, με το μ να ικανοποιεί τη συνθήκη $0 < \mu < \frac{2}{\sigma_{\max}}$, όπου $\sigma_{\max}$ είναι η μεγαλύτερη ιδιοτιμή του πίνακα αυτοσυσχέτισης $\Sigma = E[x_n x_n^T]$.

Ένα πρόβλημα που προκύπτει με τη χρήση του αλγορίθμου LMS είναι με ποια στρατηγική επιλέγουμε τον συντελεστή μ . Δεν είναι εύκολο να γνωρίζει κανείς το διάστημα μέσα στο οποίο θα πρέπει να βρίσκεται ο συντελεστής μ , διότι α) η επιλογή του περιορίζεται από την στατιστική των ποσοτήτων που λαμβάνουν μέρος στη διαδικασία της εκμάθησης και β) είναι επιθυμητό να γίνει κατάλληλη επιλογή του μ , ώστε να εξασφαλίζεται μια εφικτή και γρήγορη σύγκλιση. Μια λύση είναι ο Κανονικοποιημένος Αλγόριθμος Ελαχίστων Μέσων Τετραγώνων (Normalized Least Mean Square, NLMS), ο οποίος αποτελεί μια παραλλαγή του LMS και κατά τη μεθοδολογία του κανονικοποιεί την εξίσωση βήματος με τη δύναμη της εισόδου. Η γενικευμένη μορφή του NLMS είναι:

$$\theta_n = \theta_{n-1} + \frac{\mu e_n}{\|x_n\|^2} x_n, n=1,2,\dots,$$

όπου $\mu \in (0,2)$.

Παρατηρήσεις:

Παραλλαγές του αλγορίθμου LMS προτείνονται στη βιβλιογραφία[4]. Η διαφορά από τον κλασσικό LMS, βρίσκεται στο πώς εισέρχεται το σφάλμα e_n ή η είσοδος x_n στην αναδρομή. Όπως είδαμε στον κανονικοποιημένο αλγόριθμο NLMS, η είσοδος κανονικοποιείται, αφού εισάγουμε στην εξίσωση ενημέρωσης τον όρο $\frac{x_n}{\|x_n\|^2}$.

2. Προσαρμοστική εκμάθηση σε κυρτά σύνολα

2.1 Τοποθέτηση προβλήματος

Με την πάροδο των ετών, οι κυρτές συναρτήσεις (convex functions) και η διαδικασία βελτιστοποίησης ενός προβλήματος αναπτύχθηκαν αρκετά μέσω της χρήσης των υπολογιστών. Η κυρτή βελτιστοποίηση εμπεριέχει μεθόδους και τεχνικές που επικεντρώνονται σε προσαρμοστικούς αλγορίθμους. Μια οικογένεια αλγορίθμων είναι αυτή που ασχολείται με τις προβολές πάνω σε κυρτά σύνολα, που αποτελεί αντικείμενο της προσαρμοστικής εκμάθησης, το οποίο έχει αναπτυχθεί και έχει προωθηθεί αρκετά τα τελευταία χρόνια.

2.2 Κυρτή Ανάλυση (Convex Analysis)

Το εύρος του αντικειμένου των κυρτών συνόλων και συναρτήσεων είναι αρκετά μεγάλο και οι διάφοροι ορισμοί και θεωρήματα αναφέρονται σε γενικές περιπτώσεις των χώρων Hilbert (Hilbert spaces). Η μαθηματική έννοια του Hilbert χώρου γενικεύεται στην έννοια του Ευκλείδειου χώρου (Euclidean space). Περιλαμβάνει μεθόδους της γραμμικής άλγεβρας από έναν δισδιάστατο ή ένα τρισδιάστατο χώρο σε χώρους με πεπερασμένο ή άπειρο αριθμό διαστάσεων. Ένας Hilbert χώρος είναι ένας διανυσματικός χώρος του οποίου βασικό χαρακτηριστικό αποτελεί το εσωτερικό γινόμενο και έχει ανοίξει ορίζοντες στους υπολογισμούς διανυσματικών συντελεστών, συμπράττοντας στην ανάπτυξη αποτελεσματικών μεθόδων προσαρμοστικής εκμάθησης.

2.2.1 Κυρτά σύνολα

Ορισμός 2.1. Ένα μη-κενό υποσύνολο C του χώρου Hilbert H , $C \subseteq H$, καλείται κυρτό, αν $\forall x_1, x_2 \in C$ και $\forall \lambda \in [0,1]$, ισχύει η παρακάτω σχέση:

$$x := \lambda x_1 + (1-\lambda)x_2 \in C. \quad (2.1)$$

Αν $\lambda=1$, τότε $x=x_1$ και αν $\lambda=0$, τότε $x=x_2$. Για οποιαδήποτε άλλη τιμή του λ στο διάστημα $[0,1]$, το x είναι ο γραμμικός συνδυασμός του x_1, x_2 . Η (2.1) μπορεί να γραφτεί και ως

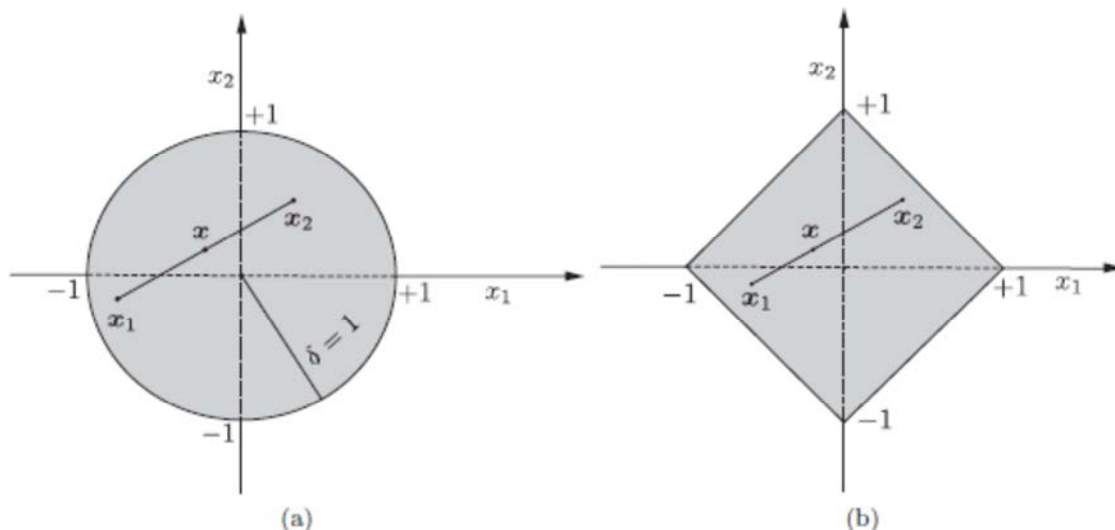
$$x - x_2 = \lambda(x_1 - x_2), \lambda \in [0,1]$$

Παρακάτω παρατίθενται δυο παραδείγματα κυρτών συνόλων στον δισδιάστατο Ευκλείδειο χώρο, $\mathbb{R}^2$. Στο Σχήμα 2.1α το σύνολο περιλαμβάνει όλα τα σημεία των οποίων η Ευκλείδεια (l_2) νόρμα είναι μικρότερη ή ίση με το 1, δηλαδή όλα τα σημεία που είναι πάνω και μέσα στον κύκλο,

$$C_2 = \{x : \sqrt{x_1^2 + x_2^2} \leq 1\}.$$

Το σύνολο C_2 μπορούμε να το ονομάσουμε και l_2 -ball ακτίνας ίσης με 1. Το σύνολο στο Σχήμα 2.1b περιλαμβάνει όλα τα σημεία που βρίσκονται πάνω και μέσα στον κώνο και ορίζεται ως,

$$C_1 = \{x : |x_1| + |x_2| \leq 1\}.$$



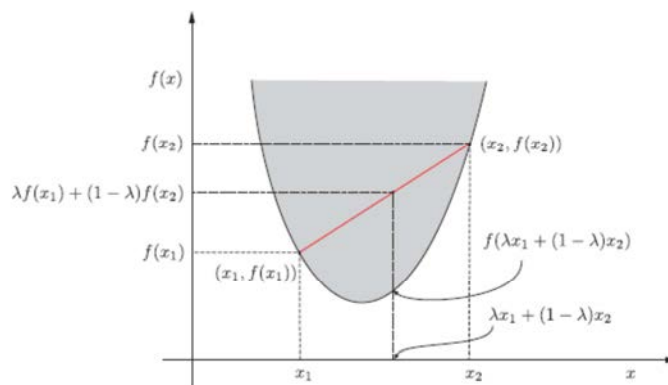
Σχήμα 2.1: a) Η l_2 -ball ακτίνας 1 περιλαμβάνει όλα τα σημεία με Ευκλείδεια νόρμα μικρότερη ή ίσης του 1. b) Η l_1 -ball περιλαμβάνει όλα τα σημεία με νόρμα μικρότερη ή ίση του 1.

2.2.2 Κυρτές συναρτήσεις

Ορισμός 2.2. Μια συνάρτηση $f : X \subseteq \mathbb{R}^I \rightarrow \mathbb{R}$, καλείται κυρτή αν το X είναι κυρτό και $\forall x_1, x_2 \in X$ ισχύει η παρακάτω σχέση:

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2), \lambda \in [0,1]. \quad (2.2)$$

Η συνάρτηση καλείται αυστηρά κυρτή, αν ισχύει αυστηρά η ανισότητα, όταν $\lambda \in (0,1)$, $x_1 \neq x_2$. Στο Σχήμα 2.2 παρατηρούμε το γραμμικό συνδυασμό των σημείων $(x_1, f(x_1))$, $(x_2, f(x_2))$. Επίσης, μια συνάρτηση καλείται κοίλη, αν η αρνητική συνάρτηση $-f(\cdot)$ είναι κυρτή.



Σχήμα 2.2: Ο γραμμικός συνδυασμός των σημείων $(x_1, f(x_1))$, $(x_2, f(x_2))$.

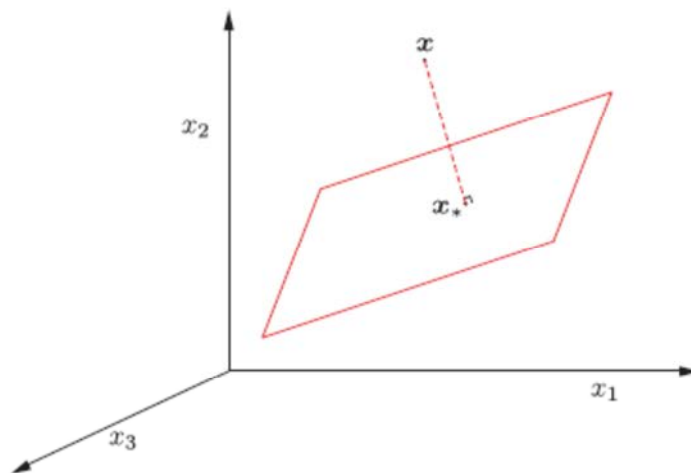
2.3 Προβολές πάνω σε κυρτά σύνολα (Projections Onto Convex Sets, POCS)

Η έννοια της προβολής είναι ένα θεμελιώδες αντικείμενο μελέτης των μαθηματικών με πολλές εφαρμογές στη βασική γεωμετρία και όχι μόνο. Παρατηρώντας την μέθοδο των προβολών, γίνεται αντιληπτό το γεγονός πως κατά την παρουσία μιας προβολής, π.χ. σχεδιάζοντας ένα τμήμα ελαχίστου μήκους από ένα σημείο σε μια γραμμή ή ένα επίπεδο, λύνεται ένα πρόβλημα βελτιστοποίησης. Ένα χαρακτηριστικό παράδειγμα προβολής βρίσκεται, π.χ. στον τρισδιάστατο χώρο, όταν υπάρχει ένα σημείο x^* , το οποίο είναι η προβολή του x σ' ένα επίπεδο, H , συμπεριλαμβανομένων όλων των σημείων που βρίσκονται στο επίπεδο και η απόστασή τους από το $x = [x_1, x_2, x_3]$ είναι ελάχιστη,

$$x^* = \min ((x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2), y \in H. \quad (2.3)$$

Θεώρημα 2.1. Έστω C ένα μη κενό κλειστό κυρτό σύνολο που βρίσκεται στο Hilbert χώρο, H , και $x \in H$. Τότε υπάρχει ένα μοναδικό σημείο, που συμβολίζεται με $P_C(x) \in C$, όπου,

$$\|x - P_C(x)\| = \min \|x - y\| : \text{Προβολή του } x \text{ στο } C.$$

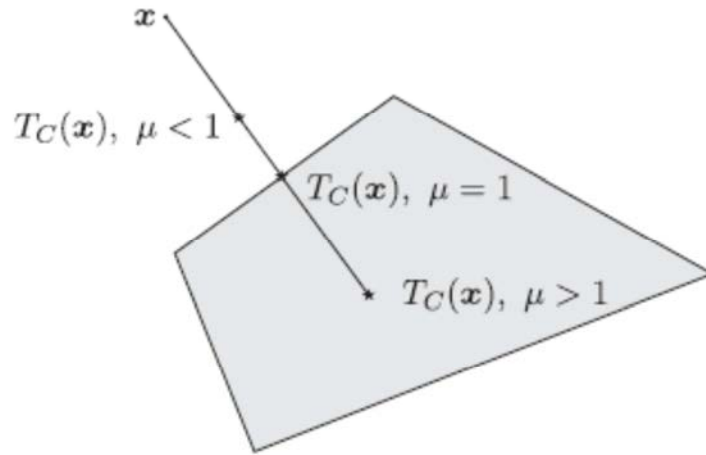


Σχήμα 2.3: Η προβολή x^* του x πάνω στο επίπεδο

Η μεταβλητή $P_C(x)$ καλείται προβολή του x στο C . Αν $x \in C$, τότε $P_C(x) = x$, αφού η νόρμα $\|x - P_C(x)\| = 0$. Η απόδειξη του θεωρήματος όπως και οι ιδιότητες των προβολών αναφέρονται στο [1, chapter 8]. Στο σημείο αυτό αξίζει να αναφέρουμε τον ορισμό της χαλαρής προβολής (relaxed projection), καθώς διαδραματίζει σημαντικό ρόλο στη βελτίωση της σύγκλισης αλγορίθμων που σχετίζονται με την περίπτωση της προβολής P_C .

Ορισμός 2.3. Έστω ένα κλειστό κυρτό σύνολο C στο Hilbert χώρο. Ένας παράγοντας (operator) $T_C : H \rightarrow C$ καλείται χαλαρή προβολή αν, $\forall x \in H$

$$T_C(x) = x + \mu(P_C(x) - x), \mu \in (0, 2): \text{Χαλαρή προβολή του } x \text{ στο } C.$$



Σχήμα 2.4: Η γεωμετρική αναπαράσταση της χαλαρής προβολής

2.3.1 Προβολή πάνω σε κλειστή μπάλα (closed ball)

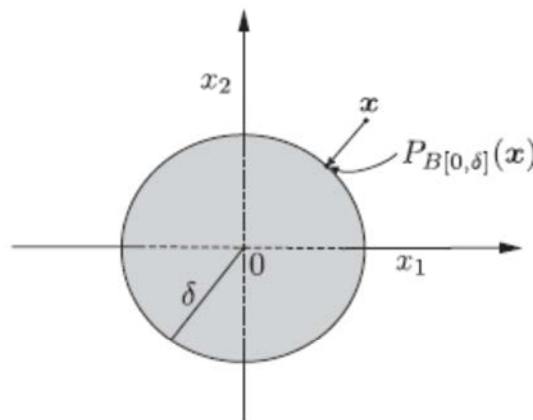
Έστω ένα σημείο $x \in H$, όπου H είναι Hilbert χώρος και ύπαρξη προβολής του σημείου x πάνω σε μια κλειστή μπάλα ακτίνας δ . Η κλειστή μπάλα, που έχει κέντρο το 0 και ακτίνα δ , συμβολίζεται $B[0, \delta]$ και ορίζεται ως

$$B[0, \delta] = \{y : \|y\| \leq \delta\}.$$

Η προβολή του $x \notin B[0, \delta]$, πάνω στο $B[0, \delta]$, δίνεται από τον παρακάτω τύπο,

$$P_{B[0, \delta]}(x) = \begin{cases} x, & \text{if } \|x\| \leq \delta \\ \delta \frac{x}{\|x\|}, & \text{if } \|x\| > \delta \end{cases} : \text{Προβολή πάνω σε κλειστή μπάλα,}$$

και η γεωμετρική αναπαράσταση εμφανίζεται στο Σχήμα 2.5. Περισσότερα παραδείγματα σχετικά με τις προβολές, όπως η προβολή πάνω σε ένα υπερεπίπεδο, ή σε ένα κώνο αναφέρονται στο [6].



Σχήμα 2.5: Η προβολή πάνω σε κλειστή μπάλα ακτίνας δ και κέντρου 0

2.4 Η θεμελιώδης θεωρία των προβολών σε κυρτά σύνολα (POCS)

Έστω C_k , $k=1,2,\dots,K$, είναι ένας πεπερασμένος αριθμός κλειστών κυρτών συνόλων στον Hilbert χώρο H , τα οποία υποθέτουμε ότι μοιράζονται μια μη κενή τομή,

$$C = \bigcap_{k=1}^K C_k \neq \emptyset.$$

Έστω T_{C_k} , $k=1,2,\dots,K$, είναι οι αντίστοιχες απεικονίσεις των χαλαρών προβολών

$$T_{C_k} = I + \mu_k(P_{C_k} - I), \mu_k \in (0,2), k=1,2,\dots,K.$$

Η συνένωση των χαλαρών προβολών συμβολίζεται ως

$$T := T_{C_K} T_{C_{K-1}} \dots T_{C_1}.$$

Συνεπώς, το T περιλαμβάνει μια ακολουθία χαλαρών προβολών, αρχίζοντας από το C_1 .

Θεώρημα 2.2. Έστω C_k , $k=1,2,\dots,K$, κλειστά κυρτά σύνολα στο Hilbert χώρο, H , με μη κενή τομή. Τότε, για οποιοδήποτε $x_0 \in H$, η ακολουθία $(T^n(x_0))$, $n=1,2,\dots$ συγκλίνει ασθενώς σ' ένα σημείο στο $C = \bigcap_{k=1}^K C_k$.

Το θεώρημα εισάγει την έννοια της ασθενούς σύγκλισης [7]. Όταν H είναι ένας Ευκλείδειος χώρος, η έννοια της ασθενούς σύγκλισης συμπίπτει με την κύρια και γνωστή σύγκλιση. Η ασθενής σύγκλιση συναντάται όταν αναφερόμαστε σε χώρους άπειρης διάστασης. Μια ακολουθία, $x_n \in H$, λέμε ότι συγκλίνει ασθενώς σε ένα σημείο x^* , αν $\forall y \in H$,

$$\langle x_n, y \rangle \rightarrow \langle x^*, y \rangle$$

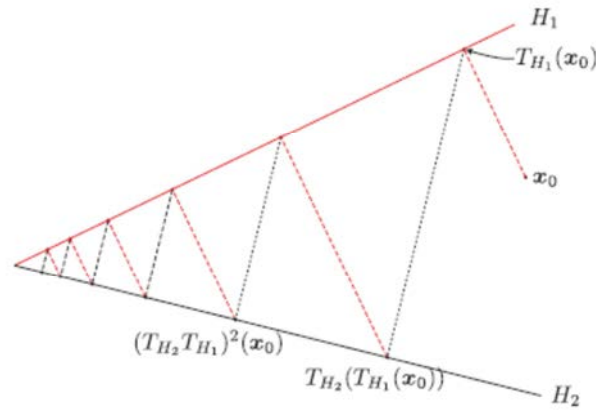
και γράφουμε

$$x_n \rightarrow x^*.$$

Το Σχήμα 2.6 δείχνει τη γεωμετρική αναπαράσταση του θεωρήματος.

Παρατηρήσεις:

- Παρατηρώντας το Σχήμα 2.6 γίνεται αντιληπτό ότι τα κλειστά κυρτά σύνολα είναι 2 ευθείες στον $\mathbb{R}^2$ και η ακολουθία των προβολών τείνει στην τομή των H_1, H_2 .
- Στην ειδική περίπτωση όπου όλα τα C_k , $k=1,2,\dots,K$, είναι κλειστά υποσύνολα στον H , τότε $T^n(x_0) \rightarrow P_C(x_0)$, δηλαδή η ακολουθία των χαλαρών προβολών συγκλίνει ισχυρά στην προβολή του x_0 στο C . Υπενθυμίζεται ότι αν κάθε C_k , $k=1,2,\dots,K$, είναι ένα κλειστό υποσύνολο, αποδεικνύεται ότι και η τομή τους είναι και αυτή κλειστό υποσύνολο στον H .



Σχήμα 2.6: Η γεωμετρική αναπαράσταση του θεωρήματος

2.5 Η παράλληλη έκδοση της θεωρίας των προβολών σε κυρτά σύνολα (POCS)

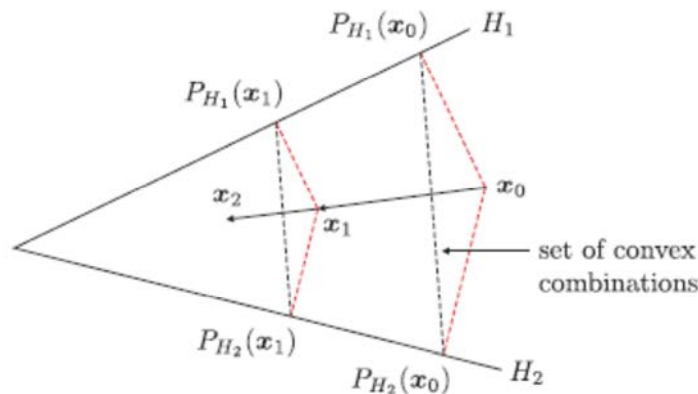
Θεώρημα 2.3. Έστω $C_k, k=1,2,\dots,K$, κλειστά κυρτά σύνολα στο Hilbert χώρο, H . Τότε για οποιοδήποτε $x_0 \in H$, η ακολουθία x_n , ορίζεται ως

$$x_n = x_{n-1} + \mu_n \left(\sum_{k=1}^K \omega_k P_{C_k}(x_{n-1}) - x_{n-1} \right), \quad (2.4)$$

με $0 < \mu_n \leq M_n$ και $M_n := \sum_{k=1}^K \frac{\omega_k \|P_{C_k}(x_{n-1}) - x_{n-1}\|^2}{\left\| \sum_{k=1}^K \omega_k P_{C_k}(x_{n-1}) - x_{n-1} \right\|^2}$, όπου $\omega_k > 0, k=1,2,\dots,K$,

έτσι ώστε $\sum_{k=1}^K \omega_k = 1$.

Η (2.4) με την αναδρομή της δείχνει ότι σε κάθε επανάληψη, όλες οι προβολές πάνω στα κυρτά σύνολα ενεργούν ταυτόχρονα και τότε είναι κυρτά συνδυασμένες. Η παράμετρος μ επιλέγεται στο διάστημα $(0, M_n]$, όπου το M_n υπολογίζεται αναδρομικά, με αποτέλεσμα να εγγυάται την σύγκλιση. Η απόδειξη του παραπάνω θεωρήματος αναφέρεται στο [8].



Σχήμα 2.7: Η γεωμετρική αναπαράσταση της παράλληλης έκδοσης των POCS

2.6 Εκτίμηση παραμέτρων και εκμάθηση μηχανής με κυρτά σύνολα

Μελετώντας το μοντέλο παλινδρόμησης (regression), η σχέση εισόδου – εξόδου δίνεται από τον παρακάτω τύπο,

$$y_n = \theta_0^T x_n + \eta_n, (y_n, x_n) \in \mathbb{R} \times \mathbb{R}^l, n=1,2,\dots,N, \quad (2.5)$$

όπου το διάνυσμα θ_0 είναι η άγνωστη παράμετρος. Υποθέτοντας ότι η παράμετρος η_n είναι μια οριοθετημένη ακολουθία θορύβου, π.χ.

$$|\eta_n| \leq \epsilon,$$

τότε σε συνδυασμό με τη (2.5), ισχύει ότι

$$|y_n - x_n^T \theta_0| \leq \epsilon. \quad (2.6)$$

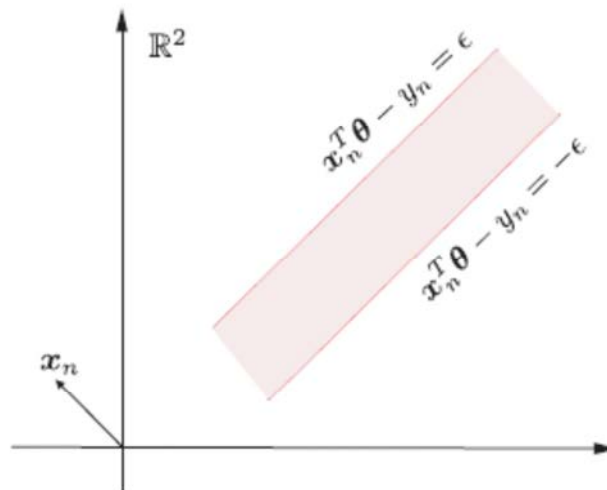
Ας προσπαθήσουμε να μελετήσουμε τώρα το παρακάτω σύνολο σημείων,

$$S_\epsilon = \{ \theta : |y_n - x_n^T \theta| \leq \epsilon \}. \quad (2.7)$$

Το σύνολο αυτό καλείται υπερπλάκα (hyperslab) και περιλαμβάνει όλα τα σημεία που βρίσκονται στην περιοχή που σχηματίζεται από τα δυο υπερεπίπεδα, όπως φαίνεται και στο Σχήμα 2.8,

$$x_n^T \theta - y_n = \epsilon, \quad x_n^T \theta - y_n = -\epsilon.$$

Η περιοχή αυτή είναι ένα κλειστό κυρτό σύνολο, όπου κάθε ζευγάρι των σημείων εκπαίδευσης (y_n, x_n) , $n=1,2,\dots,N$, ορίζει μια υπερπλάκα με διαφορετικό προσανατολισμό, που εξαρτάται από το x_n , και διαφορετική θέση στο χώρο, η οποία εξαρτάται από το y_n . Το άγνωστο θ_0 βρίσκεται μέσα σ' αυτές τις υπερπλάκες, οπότε βρίσκεται στην τομή τους. Επομένως, αν οριστεί ο φορέας προβολής στις υπερπλάκες και χρησιμοποιηθεί κάποιο από τα σχήματα του θεωρήματος POCS, θα βρεθεί ένα σημείο στην τομή. Με αυτόν τον τρόπο, αν η τομή είναι αρκετά "μικρή", τότε οποιοδήποτε σημείο θα είναι κοντά στο θ_0 .



Σχήμα 2.8: Η υπερπλάκα που ορίζεται από τα σημεία εκπαίδευσης (y_n, x_n)

Όσον αφορά τον υπολογισμό του φορέα προβολής μιας υπερπλάκας, S_ϵ , έχει αποδειχθεί ότι, δοθέντος ενός θ , η προβολή του πάνω στο S_ϵ δίνεται από τη σχέση,

$$P_{S_\epsilon} = \theta + \beta_\theta \frac{y_n - x_n^T \theta}{\|x_n\|^2} x_n, \quad (2.8)$$

όπου,

$$\beta_{\theta}(\mathbf{y}_n, \mathbf{x}_n) = \begin{cases} \frac{\mathbf{y}_n - \langle \mathbf{x}_n, \theta \rangle - \varepsilon}{\|\mathbf{x}_n\|^2}, & \text{if } \langle \mathbf{x}_n, \theta \rangle - \mathbf{y}_n < -\varepsilon \\ 0, & \text{if } |\langle \mathbf{x}_n, \theta \rangle - \mathbf{y}_n| \leq \varepsilon \\ \frac{\mathbf{y}_n - \langle \mathbf{x}_n, \theta \rangle + \varepsilon}{\|\mathbf{x}_n\|^2}, & \text{if } \langle \mathbf{x}_n, \theta \rangle - \mathbf{y}_n > \varepsilon. \end{cases} \quad (2.9)$$

Αυτό σημαίνει ότι αν το σημείο βρίσκεται μέσα στην υπερπλάκα, συμπίπτει με την προβολή του, αλλιώς η προβολή βρίσκεται πάνω σε ένα από τα δυο υπερεπίπεδα, ανάλογα με τον ορισμό του S_ε .

2.6.1 Ο αλγόριθμος της Προσαρμοστικής Μεθόδου της Προβεβλημένης Υποπαραγώγου (Adaptive Projected Subgradient Method, APSM)

Η κεντρική ιδέα του αλγορίθμου APSM έγκειται στην εξής φιλοσοφία: σε κάθε χρονική στιγμή, n , ένα ζευγάρι δεδομένων εκπαίδευσης εισόδου – εξόδου λαμβάνεται και δημιουργείται ένα κλειστό κυρτό σύνολο, C_n και λαμβάνονται υπόψη τα q τελευταία σύνολα. Η παράμετρος q ορίζεται από τον χρήστη και ουσιαστικά ορίζει ένα κινούμενο παράθυρο στο χρόνο, μέσα στο οποίο εμφανίζονται οι προβολές πάνω στα q τελευταία σύνολα. Ο αλγόριθμος APSM στηρίζεται στην παράλληλη έκδοση της θεωρίας των POCS και έχει ως στόχο να βρεθεί ένα $\theta \in \mathbb{R}^l$, το οποίο βρίσκεται στην τομή όλων εκείνων των συνόλων, C_n , ανάλογα με το q , καθώς γίνονται διαθέσιμα τα δεδομένα $(\mathbf{y}_n, \mathbf{x}_n) \in \mathbb{R} \times \mathbb{R}^l$ και σχηματίζονται οι αντίστοιχες υπερπλάκες, $S_{\varepsilon, n}$, $n=1,2,\dots$, σε κάθε χρονική στιγμή. Στην Εικόνα 2.1 παρατίθεται ο αλγόριθμος APSM.

The APSM Algorithm

- Initialization
 - * Choose $\theta_0 \in \mathbb{R}^l$.
 - * Choose q ; The number of property sets to be processed at each time instant.
- **For** $n = 1, 2, \dots, q - 1$, **Do**; Initial period, that is, $n < q$.
 - * Choose $\omega_1, \dots, \omega_n$: $\sum_{k=1}^n \omega_k = 1, \omega_k \geq 0$
 - * Select μ_n

$$\theta_n = \theta_{n-1} + \mu_n \left(\sum_{k=1}^n \omega_k P_{S_{\varepsilon, k}}(\theta_{n-1}) - \theta_{n-1} \right)$$

- **End For**
- **For** $n = q, q + 1, \dots$, **Do**
 - * Choose $\omega_n, \dots, \omega_{n-q+1}$; usually $\omega_k = \frac{1}{q}, k = n - q + 1, \dots, n$.
 - * Select μ_n

$$\theta_n = \theta_{n-1} + \mu_n \left(\sum_{k=n-q+1}^n \omega_k P_{S_{\varepsilon, k}}(\theta_{n-1}) - \theta_{n-1} \right)$$

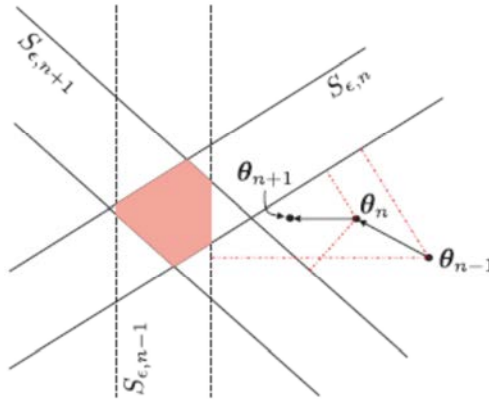
- **End For**

Εικόνα 2.1: Ο APSM αλγόριθμος

Η παράμετρος μ_n παίρνει τιμές στο διάστημα $(0, 2M_n)$, όπου,

$$M_n := \sum_{k=n-q+1}^n \frac{\omega_k \|P_{S_{\varepsilon,k}} \theta_{n-1} - \theta_{n-1}\|^2}{\left\| \sum_{k=n-q+1}^n \omega_k P_{S_{\varepsilon,k}} \theta_{n-1} - \theta_{n-1} \right\|^2}, \quad (2.10)$$

ώστε να επιτευχθεί σύγκλιση. Για τις πρώτες q επαναλήψεις η σχέση (2.10) ισχύει για $k=1$ αντί για $k=n-q+1$. Ο υπολογισμός των προβολών $P_{S_{\varepsilon,k}}(\cdot)$ δίνεται από τις σχέσεις (2.8), (2.9). Το Σχήμα 2.9 δείχνει τη γεωμετρική αναπαράσταση του APSM αλγορίθμου, όπου ο αριθμός των υπερπλακών που μελετούνται για την προβολή σε κάθε χρονική στιγμή λαμβάνεται ως $q=2$. Κάθε επανάληψη περιλαμβάνει: α) q προβολές, που διενεργούν παράλληλα, β) τον κυρτό συνδυασμό τους και γ) το βήμα ανανέωσης.



Σχήμα 2.9: Τη χρονική στιγμή n , $q=2$ υπερπλάκες έχουν σχηματιστεί, $S_{\varepsilon,n}$, $S_{\varepsilon,n-1}$, το θ_{n-1} είναι ταυτόχρονα προβεβλημένο και στις δυο και οι προβολές είναι κυρτά συνδεδεμένες. Ο επόμενος υπολογισμός είναι το θ_n και έπειτα σχηματίζεται η υπερπλάκα $S_{\varepsilon,n+1}$ κ.ο.κ. Κάθε στιγμή ο υπολογισμός πλησιάζει την τομή.

2.6.2 Σύγκλιση του APSM αλγορίθμου

Υποθέτοντας ότι ο θόρυβος είναι οριοθετημένος ($|\eta_n| \leq \varepsilon$) και ότι υπάρχει μια πραγματική τιμή θ_0 , που δημιουργεί τα δεδομένα π.χ. $y_n = x_n^T \theta_0 + \eta_n$, ισχύει

$$|x_n^T \theta_0 - y_n| \leq \varepsilon.$$

Επομένως, η τιμή θ_0 βρίσκεται στην τομή όλων των υπερπλακών της μορφής,

$$|x_n^T \theta - y_n| \leq \varepsilon.$$

Γίνεται λοιπόν φανερό το ερώτημα που γεννιέται σχετικά με το πόσο κοντά ασυμπτωτικά στην θ_0 μπορούμε να φτάσουμε, καθώς το $n \rightarrow \infty$. Αν το μέγεθος της τομής είναι μεγάλο, ακόμη κι αν συγκλίνει σε κάποιο σημείο ο αλγόριθμος, δεν γνωρίζουμε πόσο κοντά είναι στην θ_0 . Έχει όμως αποδειχθεί, [9], ότι ο υπολογισμός φτάνει οριακά κοντά στην πραγματική τιμή θ_0 υπό κάποιες προϋποθέσεις. Ο APSM αλγόριθμος απαιτεί την επιλογή τριών μεταβλητών, των ε , μ_n και q . Για την επιλογή του μ_n λαμβάνεται υπόψη το γεγονός ότι όσο μεγαλύτερο είναι το μ_n , τόσο γρηγορότερη είναι η ταχύτητα σύγκλισης, με κόστος όμως την ύπαρξη υψηλότερου τελικού επιπέδου

θορύβου. Σχετικά με το ε μια τυπική επιλογή είναι $\varepsilon = \sqrt{2}\sigma$, όπου σ είναι ο συντελεστής θορύβου. Τέλος, όσον αφορά το q , λαμβάνεται υπόψη ότι όσο μεγαλύτερο είναι τόσο πιο γρήγορα επιτυγχάνεται η σύγκλιση και αυξάνεται το κατώφλι του σφάλματος, με κόστος βέβαια στην πολυπλοκότητα του αλγορίθμου καθώς απαιτούνται περισσότερες επαναλήψεις.

Παρατηρήσεις:

- Αν οι υπερπλάκες ταυτίζονται με τα υπερεπίπεδα ($\varepsilon=0$) και $q=1$ τότε ο αλγόριθμος συμπεριφέρεται όπως ο κανονικοποιημένος LMS(NLMS) και έχει ακριβώς τα ίδια αποτελέσματα.
- Ο APSM αλγόριθμος μπορεί να γενικευτεί και στην περίπτωση όπου το σύστημα, το οποίο συνδέει είσοδο – έξοδο, είναι μη γραμμικό και επομένως ανήκει στον απειροδιάστατο χώρο Hilbert.

3. Προσαρμοστική εκμάθηση σε μη γραμμικά συστήματα

3.1 Τοποθέτηση προβλήματος

Στα προηγούμενα κεφάλαια έχει παρουσιαστεί η προσαρμοστική εκμάθηση σε γραμμικά συστήματα. Επίσης, έχουν αναπτυχθεί και αλγόριθμοι, όπως ο LMS και ο APSM, οι οποίοι επιφέρουν πολύ καλά αποτελέσματα κατά την εκτέλεση τους και χρησιμοποιούνται σε πολλές εφαρμογές. Στο κεφάλαιο αυτό θα ασχοληθούμε με την προσαρμοστική εκμάθηση στη γενική περίπτωση των μη γραμμικών συστημάτων, όπου και σ' αυτή την κατηγορία υπάρχουν αλγόριθμοι, και συγκεκριμένα ο Kernel LMS και ο Kernel APSM, οι οποίοι, ακολουθώντας τη θεωρία της μη γραμμικότητας, εκτελούνται και χρησιμοποιούνται σε εφαρμογές της προσαρμοστικής εκμάθησης σε μη γραμμικά συστήματα.

Η προσέγγιση στον τομέα της προσαρμοστικής εκμάθησης σε μη γραμμικά συστήματα θα αναφέρεται σε μια μεγάλη κατηγορία της θεωρίας της μη γραμμικότητας, η οποία συναντάται σε εφαρμογές, και αφορά το ισχυρό εργαλείο της Αναπαραγωγής Πυρήνα σε Hilbert χώρους (Reproducing Kernel Hilbert Spaces, RKHS). Η μεθοδολογία αυτή επιτρέπει την περιγραφή μιας ευρείας κατηγορίας μη γραμμικότητας με μοναδικό τρόπο, επιλύοντας ένα ισοδύναμο γραμμικό πρόβλημα σε διαφορετικό χώρο από τον προηγούμενο, όπου βρίσκονται οι αρχικές μετρήσεις. Αυτό επιτυγχάνεται με μια απεικόνιση από το χώρο που βρίσκονται τα αρχικά δεδομένα σε έναν άλλο χαρακτηριστικό χώρο (feature space). Στο σημείο αυτό αξίζει να επισημανθεί ότι σ' ένα παραμετρικό πρόβλημα αυτής της κατηγορίας, ο σχεδιαστής δεν χρειάζεται να ανησυχεί για τη φύση και τη διάσταση αυτού του μελλοντικού χώρου, καθώς του δίνεται η δυνατότητα να δοκιμάζει διαφορετικές απεικονίσεις, που επαληθεύουν διαφορετικές γραμμικότητες, και να καταλήγει σε αυτή που ταιριάζει καλύτερα στις ανάγκες του.

Σε οποιαδήποτε περίπτωση, τα μοντέλα που ασχολούνται με το αντικείμενο της μη γραμμικότητας απαιτούν δραστική αραίωση (sparsification), ώστε να μην υποφέρουν από το πρόβλημα της υπερπροσαρμογής, που αναπτύχθηκε στο Κεφάλαιο 1. Επιπλέον, εκτός από την υπερπροσαρμογή, όσο ο αριθμός των δεδομένων εκπαίδευσης, N , αυξάνεται, τόσο ένα μοντέλο αυτής της κατηγορίας γίνεται μη διαχειρίσιμο. Στον παρόν κεφάλαιο θα αναφερθούν κάποιες τεχνικές αραίωσης, διότι έχουν μεγάλη σημασία για την αποτελεσματικότητα τέτοιων μοντέλων.

3.2 Η διαδικασία της παλινδρόμησης (regression) σε μη γραμμικά συστήματα

Έστω ένα σύνολο από δεδομένα εκπαίδευσης, (y_n, x_n) , $n=1,2,\dots$, $y_n \in \mathbb{R}$, $x_n \in \mathbb{R}^l$. Στόχος είναι να ορίσουμε μια σχέση εισόδου – εξόδου μέσω του μοντέλου της μορφής

$$y_n = f(x_n) + \eta_n, n=1,2,\dots \quad (3.1)$$

όπου η_n είναι μια ακολουθία θορύβου και η μη γραμμική συνάρτηση περιγράφεται από τη σχέση

$$f(x) = \theta_0 + \sum_{k=1}^{K-1} \theta_k \phi_k(x), \quad (3.2)$$

όπου θ_k , $k=1,2,\dots,K-1$, περιλαμβάνει το σύνολο από αγνώστους παραμέτρους και $\phi_k(x)$ είναι προεπιλεγμένες μη γραμμικές συναρτήσεις

$$\phi_k(\cdot) : \mathbb{R}^l \rightarrow \mathbb{R}, k=1,2,\dots,K-1.$$

Τα διανύσματα εισόδου είναι γνωστά ως regressors. Λαμβάνοντας υπόψη τις σχέσεις (3.1), (3.2), υπολογίζουμε

$$y_n = \theta_0 + \sum_{k=1}^{K-1} \theta_k \phi_k(x_n) + \eta_n := \theta^T \phi(x_n) + \eta_n,$$

όπου

$$\phi(\cdot) = [\phi_1(\cdot), \phi_2(\cdot), \dots, \phi_{K-1}(\cdot), 1]^T \text{ και } \theta = [\theta_1, \theta_2, \dots, \theta_{K-1}, \theta_0]^T.$$

Η τιμή θ_0 είναι γνωστή ως bias και έχει συμπεριληφθεί στο διάνυσμα θ με την ταυτόχρονη προσθήκη της μονάδας σαν τελευταίο στοιχείο στο διάνυσμα ϕ . Ο στόχος αυτού του παραμετρικού προβλήματος είναι ο υπολογισμός του διανύσματος $\hat{\theta}$ από το διάνυσμα θ χρησιμοποιώντας τα δεδομένα εκπαίδευσης. Όταν το $\hat{\theta}$ υπολογιστεί και δοθέντος μιας τιμής του x , η πρόβλεψη της αντίστοιχης τιμής εξόδου υπολογίζεται σύμφωνα με το παρακάτω μοντέλο

$$\hat{y} = \hat{\theta}^T \phi(x) \quad (3.3)$$

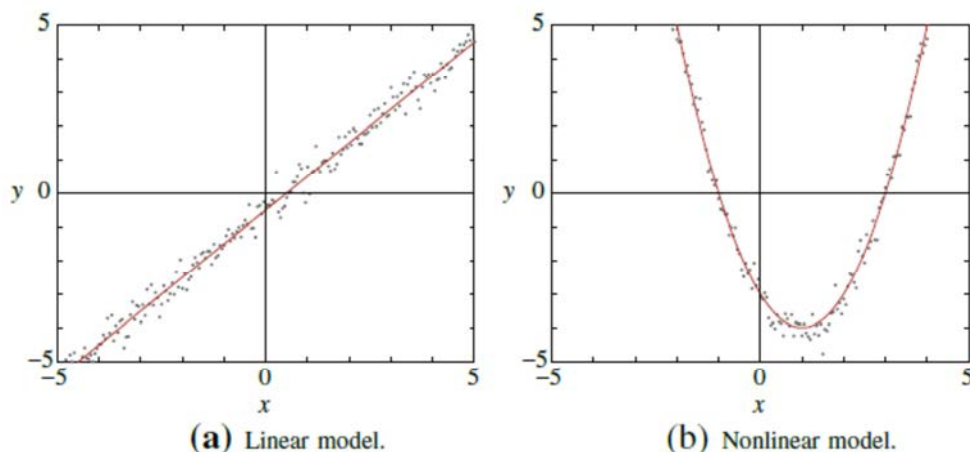
Το Σχήμα 3.1 δείχνει το γράφημα της (3.3) για δυο διαφορετικές περιπτώσεις δεδομένων εκπαίδευσης, (y_n, x_n) . Στο Σχήμα 3.1a το μοντέλο είναι, π.χ.,

$$\hat{y} = -0.5 + x,$$

και το αντίστοιχο μοντέλο στο Σχήμα 3.1b είναι

$$\hat{y} = -3 - 2\phi_1(x) + \phi_2(x),$$

όπου $\phi_1(x) = x$ και $\phi_2(x) = x^2$.



Σχήμα 3.1: Προσαρμογή του συνόλου των δεδομένων εκπαίδευσης, (y_n, x_n) , $n=1,2,\dots,N-1$ από μια γραμμική και μια τετραγωνική συνάρτηση αντίστοιχα.

Επιπλέον, το σχήμα φανερώνει το κύριο στόχο της παλινδρόμησης (regression), που είναι να εξηγήσει το μηχανισμό δημιουργίας των δεδομένων καθώς και να ορίσει το γράφημα της κατάλληλης καμπύλης ώστε τα δεδομένα στον (y, x) χώρο να είναι όσο κοντά σ' αυτή την καμπύλη γίνεται.

3.3 Αναπαραγωγή Πυρήνα σε Hilbert χώρους (Reproducing Kernel Hilbert Spaces, RKHS)

Στην παρούσα θεματική ενότητα θα μελετήσουμε ένα γραμμικό χώρο Hilbert, H , που ανήκουν πραγματικές συναρτήσεις ορισμένες σε ένα σύνολο $X \subseteq \mathbb{R}^l$. Ο χώρος αυτός περιέχει την λειτουργία του εσωτερικού γινομένου, $\langle \cdot \rangle_H$, που ορίζει μια αντίστοιχη νόρμα $\|\cdot\|_H$.

Ορισμός 3.1. Ένας Hilbert χώρος, H , καλείται Reproducing Kernel Hilbert Space (RKHS), αν υπάρχει συνάρτηση

$$k: X \times X \rightarrow \mathbb{R},$$

με τις παρακάτω ιδιότητες:

α) Για κάθε $x \in X$, $k(\cdot, x)$ ανήκει στο H .

β) η $k(\cdot, \cdot)$ έχει τη λεγόμενη reproducing ιδιότητα:

$$f(x) = \langle f, k(\cdot, x) \rangle, \quad \forall f \in H, \forall x \in X. \quad (3.4)$$

Μια περίπτωση της παραπάνω ιδιότητας είναι πως, αν θέσουμε $f(\cdot) = k(\cdot, y)$, $y \in X$, τότε

$$\langle k(\cdot, y), k(\cdot, x) \rangle = k(x, y) = k(y, x). \quad (3.5)$$

Ορισμός 3.2. Έστω H ένας Reproducing Kernel Hilbert Space (RKHS), που σχετίζεται με μια συνάρτηση πυρήνα (kernel function) $k(\cdot, \cdot)$, και X ένα σύνολο στοιχείων. Τότε η απεικόνιση (mapping)

$$X \ni x \rightarrow \phi(x) := k(\cdot, x) \in H,$$

είναι γνωστή ως χαρακτηριστική απεικόνιση (feature map) και ο χώρος, H , χαρακτηριστικός χώρος (feature space).

Συνεπώς, αν το X είναι το σύνολο των διανυσμάτων παρατήρησης, τότε η χαρακτηριστική απεικόνιση απεικονίζει κάθε διάνυσμα στον RKHS. Επιπλέον, εφόσον ένας Hilbert χώρος μπορεί να είναι άπειρης διάστασης και τα στοιχεία του να γίνονται συναρτήσεις, τότε κάθε δεδομένο εκπαίδευσης μπορεί να απεικονιστεί και σαν συνάρτηση. Επομένως, οι απεικονίσεις μπορούν να συμβολιστούν σαν συναρτήσεις, ως $\phi(\cdot)$.

Έχοντας, λοιπόν, καταφέρει να κάνουμε την απεικόνιση από έναν αυθεντικό χώρο, όπου βρίσκονται τα πραγματικά μας δεδομένα, σ' έναν μεγαλύτερης διάστασης χώρο (RKHS), μπορούμε να κάνουμε πιο εύκολους και εφικτούς υπολογισμούς με το εσωτερικό γινόμενο των συναρτήσεων απεικόνισης. Επομένως, αν $x, y \in X \subseteq \mathbb{R}^l$, τότε το εσωτερικό γινόμενο των συναρτήσεων απεικόνισης γράφεται

$$\langle \phi(x), \phi(y) \rangle = \langle k(\cdot, x), k(\cdot, y) \rangle,$$

$$\text{ή} \quad \langle \phi(x), \phi(y) \rangle = k(x, y) \quad (3.6)$$

Η παραπάνω ιδιότητα (3.6) ονομάζεται τέχνασμα πυρήνα (Kernel Trick) και διευκολύνει σημαντικά τους υπολογισμούς. Επίσης, επιτρέπει την ακόλουθη προσέγγιση, η οποία χρησιμοποιείται ευρέως στην μηχανική εκμάθηση:

- Σχεδιασμός ενός αλγορίθμου, ο οποίος απεικονίζει τα δεδομένα εκπαίδευσης στην είσοδο σε έναν RKHS

$$x_n \rightarrow \phi(x_n) \in H, n=1,2,\dots,N.$$

- Επίλυση ενός γραμμικού προβλήματος εκτίμησης στον H , εμπεριέχοντας τις εικόνες $\phi(x_n)$, $n=1,2,\dots,N$.
- Πρωταγωνιστές αυτού του αλγορίθμου είναι οι υπολογισμοί του εσωτερικού γινομένου

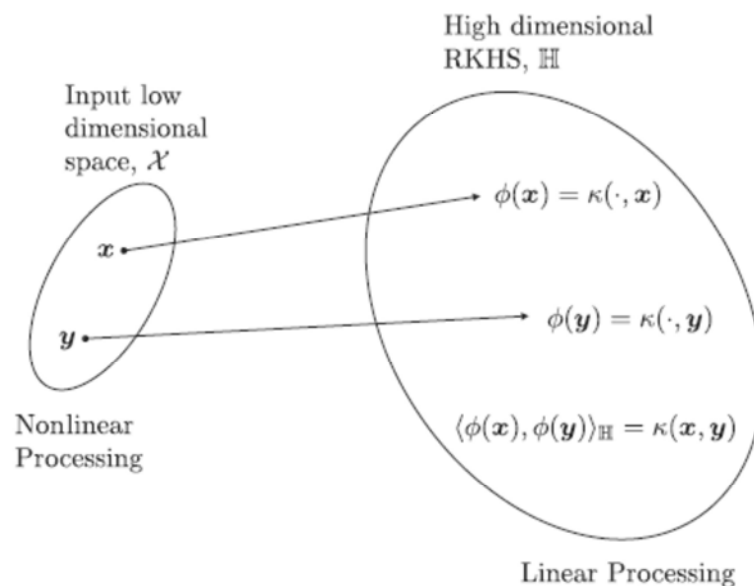
$$\langle \phi(x_i), \phi(x_j) \rangle, i, j = 1, 2, \dots, N.$$

- Αντικατάσταση κάθε εσωτερικού γινομένου με μια λειτουργία πυρήνα, π.χ.

$$\langle \phi(x_i), \phi(x_j) \rangle = \kappa(x_i, x_j).$$

Το αποτέλεσμα του αλγορίθμου θα είναι ισοδύναμο με την επίλυση μιας γραμμικής μοντελοποίησης σε έναν RKHS, που έχει οριστεί από τον επιλεγμένο πυρήνα. Κάθε εσωτερικό γινόμενο στον RKHS είναι μια μη γραμμική λειτουργία συνάρτησης στον χώρο εισόδου. Το kernel trick χρησιμοποιήθηκε αρχικά στα [10,11].

Το σχήμα 3.2 εξηγεί το θεωρητικό υπόβαθρο της διαδικασίας. Πρακτικά, τα παραπάνω βήματα είναι ισοδύναμα με: α) την εργασία στον αυθεντικό χώρο (μικρής διάστασης Ευκλείδειος χώρος) και έκφραση όλων των διαδικασιών σε όρους εσωτερικού γινομένου και β) την αντικατάσταση στο τελευταίο βήμα των εσωτερικών γινομένων με μια συνάρτηση πυρήνα.



Σχήμα 3.2: Η μη γραμμική διαδικασία στον αυθεντικό μικρής διάστασης χώρο απεικονίζεται σε γραμμική σε ένα υψηλότερης διάστασης χώρο RKHS H .

3.3.1 Παραδείγματα Συναρτήσεων Πυρήνα

Υπάρχουν μερικές συναρτήσεις πυρήνα που είναι αρκετά δημοφιλείς και χρησιμοποιούνται σε πληθώρα εφαρμογών:

1. Ο Gaussian πυρήνας (Gaussian Kernel) αποτελεί το πιο δημοφιλές παράδειγμα και ορίζεται ως

$$\kappa(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right), \text{ με } \sigma > 0.$$

Η διάσταση του RKHS που δημιουργείται από τον Gaussian πυρήνα είναι άπειρη. Η απόδειξη ότι ο Gaussian πυρήνας ικανοποιεί τις απαιτούμενες ιδιότητες δίνεται στο [12].

2. Ο ομοιογενής πολυωνυμικός πυρήνας έχει τη μορφή

$$\kappa(x,y) = (x^T y)^r, \text{ όπου } r \text{ είναι παράμετρος.}$$

3. Ο ανομοιογενής πολυωνυμικός πυρήνας έχει τη μορφή

$$\kappa(x,y) = (x^T y + c)^r, \text{ όπου } c \geq 0 \text{ και } r \text{ οι παράμετροι.}$$

Η διάσταση RKHS που σχετίζεται με πολυωνυμικούς πυρήνες είναι πεπερασμένη.

4. Ο Laplacian πυρήνας ορίζεται ως

$$\kappa(x,y) = \exp(-t\|x-y\|), \text{ όπου } t > 0 \text{ είναι παράμετρος.}$$

Η διάσταση RKHS που σχετίζεται με τον Laplacian πυρήνα είναι άπειρη.

5. Η συνάρτηση δειγματοληψίας ή sinc πυρήνας έχει ιδιαίτερο ενδιαφέρον από την όψη της επεξεργασίας σήματος και ορίζεται ως

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}.$$

6. Άλλη μια κατηγορία πυρήνων ιδιαίτερης σημασίας καλούνται πυρήνες συμβολοσειράς (string kernels) και χρησιμοποιούνται σε εφαρμογές όπου τα δεδομένα παρουσιάζονται σαν σύμβολα συμβολοσειρών.

Επιπλέον, είναι εφικτό να κατασκευαστούν πυρήνες βασιζόμενοι στις παρακάτω ιδιότητες:

1. Αν $\kappa_1(x,y) : X \times X \rightarrow \mathbb{R}$, $\kappa_2(x,y) : X \times X \rightarrow \mathbb{R}$, είναι πυρήνες, τότε

$$\kappa(x,y) = \kappa_1(x,y) + \kappa_2(x,y),$$

και

$$\kappa(x,y) = \alpha \kappa_1(x,y), \alpha > 0,$$

και

$$\kappa(x,y) = \kappa_1(x,y) \kappa_2(x,y),$$

είναι επίσης πυρήνες.

2. Έστω $f : X \rightarrow \mathbb{R}$, τότε

$$\kappa(x,y) = f(x)f(y),$$

είναι επίσης ένας πυρήνας.

3. Έστω η συνάρτηση $g : X \rightarrow \mathbb{R}$, και μια συνάρτηση πυρήνα

$$\kappa_1(\cdot, \cdot) : \mathbb{R}^l \times \mathbb{R}^l \rightarrow \mathbb{R}.$$

Τότε,

$$\kappa(x,y) = \kappa_1(g(x),g(y)),$$

είναι επίσης ένας πυρήνας.

4. Έστω A ένας θετικά ορισμένος $I \times I$ πίνακας. Τότε

$$\kappa(x,y) = x^T A y,$$

είναι ένας πυρήνας.

5. Αν $\kappa_1(x,y) : X \times X \rightarrow \mathbb{R}$, τότε

$$\kappa(x,y) = \exp(\kappa_1(x,y)),$$

είναι επίσης ένας πυρήνας, και αν $\rho(\cdot)$ είναι ένα πολυώνυμο με μη αρνητικούς συντελεστές,

$$\kappa(x,y) = \rho(\kappa_1(x,y)),$$

είναι επίσης ένας πυρήνας.

3.3.2 Representer Θεώρημα

Το Representer θεώρημα είναι ιδιαίτερα σημαντικό, από πρακτική άποψη, διότι επιτρέπει την εμπειρική βελτιστοποίηση της συνάρτησης κόστους, βασισμένη σε ένα πεπερασμένο σύνολο δεδομένων εκπαίδευσης, ακόμη και αν η συνάρτηση που πρέπει να εκτιμηθεί βρίσκεται σ' έναν μεγάλης διάστασης (πιθανόν και άπειρης) χώρο, H .

Θεώρημα 3.1. Έστω $\Omega : [0, +\infty) \rightarrow \mathbb{R}$, μια αυθαίρετη γνησίως αύξουσα συνάρτηση. Επίσης, έστω $L : \mathbb{R}^2 \rightarrow \mathbb{R} \cup \{\infty\}$ μια αυθαίρετη συνάρτηση κόστους. Τότε κάθε ελαχιστοποιητής, $f \in H$, της

$$J(\theta) := \sum_{n=1}^N L(y_n, f_{\theta}(x_n)) + \lambda \Omega(\|\theta\|)$$

έχει τη μορφή

$$f(\cdot) = \sum_{n=1}^N \theta_n \kappa(\cdot, x_n), \text{ όπου } \theta_n \in \mathbb{R}, n=1,2,\dots,N. \quad (3.7)$$

Το θεώρημα αποδείχθηκε πρώτη φορά στο [13]. Η σημασία αυτού του θεωρήματος είναι ότι για να γίνει βελτιστοποίηση της $J(\theta)$ σε σχέση με το f , πρέπει να χρησιμοποιηθεί η σχέση (3.7) και η ελαχιστοποίηση επιτυγχάνεται σε σχέση με το πεπερασμένο σύνολο παραμέτρων, θ_n , $n=1,2,\dots,N$. Είναι, επίσης, αξιοσημείωτο το γεγονός ότι όταν δουλεύουμε σε μεγάλης διάστασης χώρους, η παρουσία της διαδικασίας της τακτοποίησης είναι απαραίτητη και αναπόφευκτη, διότι αν δεν υπάρχει η βελτιστοποιημένη λύση θα υποφέρει από το πρόβλημα της υπερπροσαρμογής, που συζητήθηκε στο Κεφάλαιο 1. Η επιρροή της τακτοποίησης στην γενικευμένη εμφάνιση και σταθερότητα της λύσης έχει μελετηθεί σε αρκετές δημοσιεύσεις [14,15,16,17].

3.4 Εκμάθηση σε Απευθείας Σύνδεση σε RKHS (Online Learning in RKHS)

Στην παραμετρική μοντελοποίηση σε ένα Ευκλείδιο χώρο $\mathbb{R}^l$, ο αριθμός των αγνώστων παραμέτρων παραμένει ίδιος για όλες τις μονάδες του χρόνου. Η μοντελοποίηση, όμως, μιας άγνωστης συνάρτησης που βρίσκεται στον RKHS, επιφέρει αύξηση του αριθμού των αγνώστων παραμέτρων γραμμικά με τον χρόνο, με

αποτέλεσμα να αυξάνεται η πολυπλοκότητα με τις χρονικές επαναλήψεις και να γίνεται μη διαχειρίσιμοι οι υπολογισμοί. Για το λόγο αυτό, υπάρχουν κάποιοι αλγόριθμοι που επιλύουν το πρόβλημα αυτό, με τους πιο δημοφιλείς να είναι ο Kernel LMS και ο Kernel APSM.

3.4.1 Ο αλγόριθμος Kernel Least Mean Square (KLMS)

Έστω $x \in \mathbb{R}^l$ και ο χαρακτηριστική απεικόνιση (feature map)

$$x \rightarrow \phi(x) = \kappa(\cdot, x) \in H.$$

Για να κάνουμε εκτίμηση της $f \in H$, ελαχιστοποιούμε τη συνάρτηση κόστους

$$J(f) = \frac{1}{2} E[|y - \langle f, \phi(x) \rangle|^2], \quad (3.8)$$

όπου $\langle \cdot, \cdot \rangle$ συμβολίζει το εσωτερικό γινόμενο στον H . Παίρνοντας τη μερική παράγωγο της (3.8) ως προς θ , έχουμε

$$\nabla_f J(f) = -E[\phi(x)(y - \langle f, \phi(x) \rangle)].$$

Επομένως, λαμβάνοντας υπόψη την έννοια της στοχαστικής παραγώγου και αντικαθιστώντας τις τυχαίες μεταβλητές με τις παρατηρήσεις, έχουμε την ακόλουθη αναδρομική σχέση που ορίζει το βήμα ανανέωσης σε κάθε χρονική στιγμή του αλγορίθμου KLMS

$$\begin{aligned} f_n &= f_{n-1} + \mu_n e_n \phi(x_n), \\ &= f_{n-1} + \mu_n e_n \kappa(\cdot, x_n), \end{aligned} \quad (3.9)$$

όπου $e_n = y_n - \langle f_{n-1}, \phi(x_n) \rangle$, με $n=1,2,\dots$

Ξεκινώντας τις επαναλήψεις από την αρχική τιμή $f_0 = 0$ και θέτοντας $\mu_n = \mu$, υπολογίζουμε

$$f_n = \mu \sum_{i=1}^n e_i \kappa(\cdot, x_i), \quad (3.10)$$

και η πρόβλεψη εξόδου, βασιζόμενη στα δείγματα εκπαίδευσης δίνεται ως

$$\begin{aligned} \hat{y}_n &= \langle f_{n-1}, \kappa(\cdot, \phi(x_n)) \rangle \\ &= \mu \sum_{i=1}^{n-1} e_i \kappa(x_n, x_i). \end{aligned}$$

Παρατηρούμε ότι η σχέση (3.10) συμφωνεί με το Representer θεώρημα. Έτσι, αντικαθιστώντας τον όρο μe_i με θ_i και συνδέοντας τις (3.9), (3.10), σε κάθε χρονική στιγμή ο αλγόριθμος KLMS έχει τα ακόλουθα βήματα:

$$\begin{aligned} e_n &= y_n - \hat{y}_n = y_n - \sum_{i=1}^{n-1} \theta_i \kappa(x_n, x_i), \\ \theta_n &= \mu e_n. \end{aligned} \quad (3.11)$$

Η εξίσωση βήματος του κανονικοποιημένου KLMS (normalized KLMS) γίνεται

$$\theta_n = \mu \frac{e_n}{\kappa(x_n, x_n)}.$$

3.4.2 Ο αλγόριθμος Kernel Adaptive Projected Subgradient Method (KAPSM)

Ο αλγόριθμος APSM στηρίζεται στη θεωρία των προβολών πάνω σε κυρτά σύνολα (POCS theory) και η βασική φιλοσοφία του επεκτείνεται στο χώρο των συναρτήσεων, με αποτέλεσμα την ύπαρξη του αλγορίθμου KAPSM. Για τον KAPSM, ισχύει

$$f_n = f_{n-1} + \mu_n \left(\frac{1}{q} \sum_{k=n-q+1}^n P_k(f_{n-1}) - f_{n-1} \right), \quad (3.12)$$

όπου τα βάρη για τον κυρτό συνδυασμό των προβολών είναι ίσα με $\omega_k = \frac{1}{q}$, $k \in [n-q+1, n]$. Με P_k συμβολίζεται ο φορέας προβολής στο αντίστοιχο κυρτό σύνολο.

Στη διαδικασία της παλινδρόμησης (regression), επιλέγουμε να πάρουμε τις προβολές στις υπερπλάκες (hyperslabs), που είναι ορισμένες από τα σημεία $(y_n, \kappa(\cdot, x_n))$ και την παράμετρο ε , η οποία ελέγχει το πλάτος της υπερπλάκας. Έτσι, λοιπόν, έχουμε

$$P_k(f_{n-1}) = f_{n-1} + \beta_k \kappa(\cdot, x_k), \quad (3.13)$$

με

$$\beta_k = \begin{cases} \frac{y_k - \langle f_{n-1}, \kappa(\cdot, x_k) \rangle - \varepsilon}{\kappa(x_k, x_k)}, & \text{if } \langle f_{n-1}, \kappa(\cdot, x_k) \rangle - y_k < -\varepsilon \\ 0, & \text{if } |\langle f_{n-1}, \kappa(\cdot, x_k) \rangle - y_k| \leq \varepsilon \\ \frac{y_k - \langle f_{n-1}, \kappa(\cdot, x_k) \rangle + \varepsilon}{\kappa(x_k, x_k)}, & \text{if } \langle f_{n-1}, \kappa(\cdot, x_k) \rangle - y_k > \varepsilon. \end{cases} \quad (3.14)$$

Παρατηρείται ότι από τις σχέσεις (3.13), (3.14), η (3.12) μπορεί να πάρει τη μορφή

$$f_n = f_{n-1} + \mu_n \left(\sum_{k=n-q+1}^n \beta_k \kappa(\cdot, x_k) \right),$$

όπου ο συντελεστής $1/q$ συμπεριλαμβάνεται στο β_k .

Τα βήματα του αλγορίθμου KAPSM είναι τα ίδια με τα βήματα του αλγορίθμου APSM (Εικόνα 2.1), με τη διαφορά ότι αναφερόμαστε σε συναρτήσεις πυρήνα και ως σχέση ανανέωσης χρησιμοποιούμε την (3.12), που έχει ως αποτέλεσμα τον υπολογισμό των σχέσεων (3.13), (3.14). Η παράμετρος μ_n μπορεί να πάρει τιμές στο διάστημα $(0, 2M_n)$, όπου

$$M_n := \sum_{k=n-q+1}^n \frac{\beta_k^2 \kappa(x_k, x_k)}{\sum_{i=n-q+1}^k \beta_i \beta_k \kappa(x_i, x_i)}, \quad \text{αν } \sum_{i=n-q+1}^k \beta_i \beta_k \kappa(x_i, x_i) \neq 0, \text{ αλλιώς } M_n = 1.$$

Για τις πρώτες q επαναλήψεις ισχύει η παραπάνω σχέση αλλά με $k=1$ και $i=1$.

Παρατηρήσεις:

- Στην περίπτωση των υπερπλάκων, αν $\varepsilon=0$ και $q=1$, ο KAPSM κατά την εκτέλεσή του λειτουργεί σαν τον κανονικοποιημένο KLMS.

3.4.3 Αραίωση της λύσης (Sparsifying the solution)

Όπως είδαμε στο Κεφάλαιο 1, η αραίωση (sparsification) της λύσης είναι ένας τρόπος με τον οποίο μπορεί να αντιμετωπιστεί το πρόβλημα της υπερπροσαρμογής (overfitting). Εκτός όμως από τους Ευκλείδειους χώρους, η αραίωση μπορεί να εφαρμοστεί και στους RKH χώρους με απόλυτη επιτυχία και την μέθοδο αυτή την χρησιμοποιούν και οι δυο αλγόριθμοι που μελετάμε ο KLMS και ο KAPSM.

Θα μελετήσουμε τις διάφορες τεχνικές αραίωσης που μπορούν να εφαρμοστούν και στους δυο αλγορίθμους αλλά και τεχνικές που εφαρμόζουμε σε καθένα ξεχωριστά.

3.4.3.1 Αραίωση στον KLMS

Ένα μειονέκτημα του KLMS είναι ότι ένας αυξανόμενος αριθμός σημείων εκπαίδευσης, x_n , εμπεριέχεται στην εκτίμηση της εξόδου του συστήματος. Το σύνολο αυτών των σημείων μπορεί να θεωρηθεί ως ένα “λεξικό” (“dictionary”), D , το οποίο αποθηκεύεται στη μνήμη. Αυτή η ιδέα περιλαμβάνεται σε οποιοδήποτε μοντέλο μηχανικής εκμάθησης βασιζόμενη στη θεωρία kernel. Επομένως, γίνεται αντιληπτό ότι η παραπάνω ιδέα απαιτεί μεγαλύτερη μνήμη και περισσότερες υπολογιστικές πηγές, καθώς ο χρόνος περνά. Ιδιαίτερα, αν ο KLMS χρησιμοποιηθεί σε εφαρμογές σε συστήματα πραγματικού χρόνου, το λεξικό μεγαλώνει απεριόριστα, με αποτέλεσμα να γεμίσει πλήρως τη μνήμη της μηχανής και να απαιτεί περισσότερο χρόνο να γίνουν οι υπολογισμοί. Για να αποφευχθεί αυτή η δυσάρεστη έκταση που μπορεί να πάρει ο αλγόριθμος, υπάρχουν τεχνικές, όπου το λεξικό των σημείων δημιουργείται στην έναρξη της εκτέλεσης του αλγορίθμου και τα νέα σημεία εισάγονται μέσα σ’ αυτό μόνο υπό συγκεκριμένες προϋποθέσεις. Η πολυπλοκότητα του KLMS σε κάθε χρονική στιγμή n είναι $O(n)$ αλλά αν εφαρμοσθεί κάποια απ’ αυτές τις τεχνικές η πολυπλοκότητα γίνεται $O(M_n)$, όπου M_n είναι το μέγεθος του λεξικού τη χρονική στιγμή n . Ας δούμε μερικές δημοφιλείς τεχνικές:

- Κριτήριο Καινοτομίας (Novelty Criterion)[18]:

Έστω D_{n-1} το υπάρχον λεξικό, το μέγεθος του είναι M_{n-1} και τα στοιχεία του συμβολίζονται ως u_k , $k=1,2,\dots,M_{n-1}$. Όταν ένα νέο σημείο παρατήρησης έρχεται, η απόστασή του από όλα τα σημεία του D_{n-1} υπολογίζεται ως

$$d(x_n, D_{n-1}) = \min_{u_k \in D_{n-1}} \{\|x_n - u_k\|\}.$$

- Αν αυτή η απόσταση είναι μικρότερη από ένα κατώφλι, δ_1 , τότε το σημείο αυτό αγνοείται και το λεξικό παραμένει το ίδιο, δηλαδή $D_n = D_{n-1}$.
- Αλλιώς, υπολογίζεται το σφάλμα

$$e_n = y_n - \hat{y}_n = y_n - \sum_{k=1}^{M_{n-1}} \theta_k \kappa(x_n, u_k).$$

Αν $|e_n| < \delta_2$, όπου δ_2 ένα κατώφλι, τότε το σημείο απορρίπτεται. Αν όχι, το x_n εισέρχεται στο λεξικό και $D_n = D_{n-1} \cup \{x_n\}$.

- Κριτήριο Συνοχής (Coherence Criterion):

Σύμφωνα μ’ αυτό το κριτήριο, το σημείο x_n εισέρχεται στο λεξικό, αν η συνοχή του είναι πάνω από κάποιο κατώφλι, ϵ_0 , π.χ.

$$\max_{u_k \in D_{n-1}} \{\|\kappa(x_n, u_k)\|\} > \epsilon_0.$$

Έχει αποδειχθεί[19] ότι με αυτό το κριτήριο το μέγεθος του λεξικού παραμένει πεπερασμένο, καθώς $n \rightarrow \infty$.

- Κριτήριο Αιφνιδιασμού (Surprise Criterion)[20]:

Ο αιφνιδιασμός(surprise) ενός νέου ζευγαριού (y_n, x_n) σε ένα σύστημα εκμάθησης T , ορίζεται ως η λογαριθμική πιθανότητα του (y_n, x_n) ,

$$S_T(y_n, x_n) = -\ln p((y_n, x_n)|T).$$

Σύμφωνα μ' αυτό το κριτήριο, ένα ζευγάρι δεδομένων εντάσσεται σε μια από τις παρακάτω κατηγορίες:

- Ασυνήθης (Abnormal): $S_T(y_n, x_n) > \delta_1$,
- Επικήτητη (Learnable): $\delta_1 \geq S_T(y_n, x_n) \geq \delta_2$,
- Πλεονάζων (Redundant): $S_T(y_n, x_n) < \delta_2$, όπου δ_1, δ_2 , τιμές κατωφλίου. Για την περίπτωση του LMS με Gaussian εισόδους, υπολογίζεται η παρακάτω πιθανότητα

$$S_T(y_n, x_n) = \frac{1}{2} \ln(r_n) + \frac{e_n^2}{2r_n},$$

$$\text{όπου } r_n = \lambda + \kappa(x_n, x_n) - \max_{u_k \in D_{n-1}} \left\{ \frac{\kappa^2(x_n, u_k)}{\kappa(u_k, u_k)} \right\},$$

με λ : παράμετρος τακτοποίησης ορισμένη από το χρήστη.

Μια άλλη τεχνική, η οποία δίνει τη δυνατότητα αλλαγής των αντίστοιχων βαρών των σημείων του λεξικού είναι η τεχνική της κβάντισης (quantization technique), που περιγράφεται από τον αλγόριθμο quantized KLMS (QKLMS)[21].

3.4.3.2 Αραίωση στον KAPSM

Η αραίωση στον KAPSM μπορεί να επιτευχθεί είτε μέσω της χρήσης λεξικού, με τις τεχνικές που μελετήθηκαν και για την αραίωση στον KLMS, αλλά μέσω τακτοποίησης (regularization). Για την τακτοποίηση, επιβάλλεται ο παρακάτω περιορισμός

$$\|f_n\| \leq \delta.$$

Αν ελαχιστοποιηθεί η συνάρτηση κόστους υπό τον παραπάνω περιορισμό, τότε συμβαίνει ελαχιστοποίηση του κόστους. Υπό αυτές τις συνθήκες η ανανέωση στον KAPSM (3.12) γίνεται

$$f_n = P_{B[0, \delta]}(f_{n-1} + \mu_n \left(\frac{1}{q} \sum_{k=n-q+1}^n P_k(f_{n-1}) - f_{n-1} \right)), \quad (3.15)$$

όπου $P_{B[0, \delta]}$ είναι η προβολή πάνω σε μια κλειστή μπάλα, ορισμένη ως $B[0, \delta]$. (Κεφάλαιο 2.3.1),

$$B[0, \delta] = \{f \in H : \|f\| \leq \delta\},$$

και η προβολή υπολογίζεται

$$P_{B[0,\delta]}(\mathbf{f}) = \begin{cases} \mathbf{f}, & \text{if } \|\mathbf{f}\| \leq \delta \\ \delta \frac{\mathbf{f}}{\|\mathbf{f}\|}, & \text{if } \|\mathbf{f}\| > \delta \end{cases}.$$

Επομένως, στην ανανέωση της εκτίμησης της (3.12) λαμβάνεται υπόψη και η παραπάνω προβολή που ικανοποιεί τον περιορισμό που τέθηκε.

Όπως και στον KLMS, έτσι και στον KAPSM υπάρχει ο αλγόριθμος του quantized KAPSM, ο οποίος περιγράφεται στο [22].

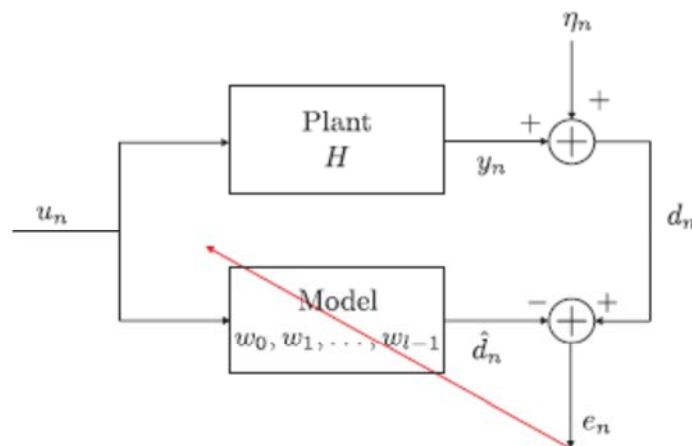
4. Πειραματική μελέτη προσαρμοστικών αλγορίθμων

4.1 Τοποθέτηση Προβλήματος

Στο παρόν κεφάλαιο θα μελετήσουμε τα αποτελέσματα της εκτέλεσης των αλγορίθμων για γραμμικά συστήματα και συγκεκριμένα των APSM, NLMS, καθώς και για μη γραμμικά συστήματα των KAPSM, KNLMS. Όλοι αυτοί οι αλγόριθμοι και το θεωρητικό τους υπόβαθρο μελετήθηκε στα προηγούμενα κεφάλαια. Για τα γραμμικά συστήματα θα χρησιμοποιήσουμε ένα γραμμικό πρόβλημα μοντελοποίησης, που καλείται ταύτιση καναλιού (Channel Identification) και για τα μη γραμμικά συστήματα θα μελετήσουμε την ισοστάθμιση καναλιού (Channel Equalization).

4.2 Ταύτιση Συστήματος (System Identification)

Στην ταύτιση συστήματος, σκοπός είναι να μοντελοποιηθεί η απόκριση ενός αγνώστου έργου (plant), H , ώστε να είναι εφικτή η πρόσβαση στο σήμα εισόδου του, όπως επίσης και σε μια θορυβώδη έκδοση της εξόδου του. Η εργασία αυτή ασχολείται, δηλαδή, με το σχεδιασμό ενός μοντέλου, του οποίου η απόκριση προσεγγίζει την αντίστοιχη ενός αγνώστου έργου. Για το λόγο αυτό, σχεδιάζεται με βέλτιστο τρόπο ένα γραμμικό φίλτρο, του οποίου η είσοδος είναι ίδια με την αντίστοιχη που υπάγεται στο έργο και η επιθυμητή απόκρισή του είναι η θορυβώδης έξοδος του έργου, όπως φαίνεται και στο Σχήμα 4.1.



Σχήμα 4.1: Στην ταύτιση συστήματος, η απόκριση του μοντέλου εκτιμάται βέλτιστα, ώστε η έξοδος να είναι κοντά, όσον αφορά το MSE, στο αντίστοιχο του αγνώστου έργου. Η κόκκινη γραμμή δείχνει ότι το σφάλμα χρησιμοποιείται για τη βέλτιστη εκτίμηση των αγνώστων παραμέτρων του φίλτρου.

Έστω λοιπόν u_n η είσοδος του συστήματος, y_n η έξοδος του έργου χωρίς θόρυβο και d_n η τελική έξοδος μαζί με την παρουσία του θορύβου. Έχουμε, επομένως, τις παρακάτω ισότητες

$$\sum_u w^* = E[u_n, d_n] = E[u_n, y_n] + 0,$$

υποθέτοντας πως ο θόρυβος η_n είναι στατιστικά ανεξάρτητος της u_n και Σ ο πίνακας συνδυασποράς. Επομένως, οι εξισώσεις παραμένουν ίδιες ακόμα και αν υφίσταται το μοντέλο όπου η επιθυμητή απόκριση είναι ίση με την έξοδο χωρίς θόρυβο του αγνώστου έργου, π.χ., $d_n = y_n$. Η ταύτιση συστήματος είναι ιδιαίτερα σημαντική σε πολλές εφαρμογές. Σε συστήματα ελέγχου χρησιμοποιείται για να καθοδηγεί τους ελεγκτές. Στην επικοινωνία δεδομένων, με τη χρήση της εκτιμάται το κανάλι μετάδοσης για να δημιουργήσει μεγίστης πιθανοφάνειας εκτιμητές των δεδομένων μετάδοσης. Τέλος, σε πολλά συστήματα εφαρμογών, υλοποιούνται προσαρμοστικές εκδόσεις του σχήματος ταύτισης συστήματος.

Παράδειγμα 4.1. Ταύτιση Καναλιού (Channel Identification):

Η εργασία και η δομή που χαρακτηρίζει την ταύτιση καναλιού απεικονίζεται στο Σχήμα 4.1. Υποθέτουμε ότι έχουμε πρόσβαση σε ένα σύνολο παρατηρήσεων εισόδου – εξόδου, u_n και d_n αντίστοιχα, $n=0,1,2,\dots,N-1$. Επιπλέον, η απόκριση του συστήματος είναι μηδενικής μέσης τιμής, ο πίνακας συνδυασποράς της είναι Σ_w και περιλαμβάνει I συνδέσεις (taps). Ο θόρυβος είναι και αυτός μηδενικής μέσης τιμής και ο πίνακας συνδυασποράς του συμβολίζεται ως Σ_η . Συνεπώς, έχουμε το παρακάτω μοντέλο

$$\mathbf{d} := \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_{I-1} \\ \vdots \\ d_{N-1} \end{bmatrix} = \mathbf{U} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{I-1} \end{bmatrix} + \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_{I-1} \\ \vdots \\ \eta_{N-1} \end{bmatrix},$$

όπου,

$$\mathbf{U} := \begin{bmatrix} u_0 & 0 & 0 & \dots & 0 \\ u_1 & u_0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ u_{I-1} & u_{I-2} & \dots & \dots & u_0 \\ \dots & \dots & \dots & \dots & \dots \\ u_{N-1} & \dots & \dots & \dots & u_{N-I+1} \end{bmatrix}.$$

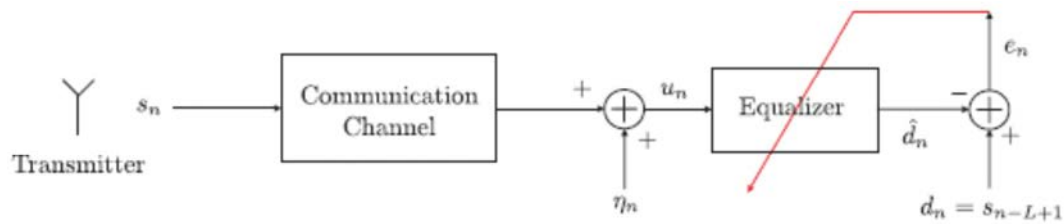
Στη συνέχεια θα αναφερθεί το συγκεκριμένο παράδειγμα της ταύτισης συστήματος που χρησιμοποιήθηκε στη πειραματική μελέτη των αλγορίθμων NLMS, APSM.

4.3 Ισοστάθμιση Καναλιού (Channel Equalization)

Στην ισοστάθμιση καναλιού, η έμφαση δίνεται στην είσοδο του αγνώστου συστήματος. Στόχος είναι να ανακτηθεί, στην βελτιστοποίηση του MSE, το (καθυστερημένο) σήμα εισόδου, u_{n-L} , όπου L είναι η καθυστέρηση στην περίοδο δειγματοληψίας, T . Η διαδικασία αυτή καλείται και ως αντίστροφη ταύτιση συστήματος (inverse system identification). Ο όρος ισοστάθμιση καναλιού χρησιμοποιείται στις επικοινωνίες.

Ο στόχος ενός ισοσταθμιστή (equalizer) είναι να ανακτήσει τα μεταδιδόμενα σύμβολα πληροφορίας, μετρίζοντας τη λεγόμενη παρεμβολή μεταξύ συμβόλων (inter-

symbol interference (ISI)), όπου οποιοδήποτε κανάλι επικοινωνίας συμμετέχει στο μεταδιδόμενο σήμα. Το Σχήμα 4.2 αναπαριστά ένα ισοσταθμιστή.



Σχήμα 4.2: Ένας ισοσταθμιστής, ο οποίος ανακτά τη μεταδιδόμενη πληροφορία

Ο ισοσταθμιστής εκπαιδεύεται, ώστε η έξοδος του να είναι αρκετά κοντά στα μεταδιδόμενα δεδομένα που έχουν καθυστερήσει για λίγο χρόνο, L . Αυτή η καθυστέρηση χρησιμοποιείται για να ληφθεί υπόψη η συνολική καθυστέρηση που επιβάλλεται από το σύστημα καναλιού – ισοσταθμιστή. Η ισοστάθμιση καναλιού βρίσκεται στην καρδιά μεγάλου αριθμού εφαρμογών στις επικοινωνίες, στην ακουστική, στην οπτική, στην επεξεργασία σεισμικού σήματος, στα συστήματα ελέγχου.

Παράδειγμα 4.2. Ισοστάθμιση Καναλιού (Channel Equalization)

Μελετώντας το Σχήμα 4.2 της ισοστάθμισης καναλιού και χρησιμοποιώντας ένα αριθμητικό παράδειγμα, η έξοδος του καναλιού δίνεται από τη σχέση

$$u_n = 0.5s_n + s_{n-1} + \eta_n.$$

Ο στόχος είναι να σχεδιασεί ισοσταθμιστής με τρεις συνδέσεις (taps), π.χ. $w = [w_0, w_1, w_2]^T$, έτσι ώστε

$$\hat{d}_n = w^T u_n,$$

και να εκτιμηθούν οι άγνωστες συνδέσεις χρησιμοποιώντας μια επιθυμητή ακολουθία απόκρισης $d_n = s_{n-1}$. Επίσης, ισχύει $E[s_n] = E[\eta_n] = 0$ και $\Sigma_s = \sigma_s^2 I$, $\Sigma_\eta = \sigma_\eta^2 I$. Για την επιθυμητή απόκριση, χρησιμοποιούμε καθυστέρηση, π.χ., $L=1$. Για να γίνει περισσότερο κατανοητός ο λόγος που χρησιμοποιείται η καθυστέρηση, αρκεί να παρατηρήσουμε ότι σε κάθε χρονική στιγμή n , η περισσότερη συνεισφορά στο u_n προέρχεται από το σύμβολο s_{n-1} , το οποίο έχει συντελεστή 1, ενώ το s_n συντελεστή 0.5. Παρατηρώντας λοιπόν αυτή τη συμπεριφορά, καταλήγουμε στο ότι η προσπάθεια στρέφεται στον υπολογισμό της εκτίμησης για το s_{n-1} . Έτσι, αιτιολογείται η χρήση της καθυστέρησης.

Συμπερασματικά, ο στόχος του ισοσταθμιστή είναι να αποτρέψει κατά βέλτιστο τρόπο α) την παρεμβολή μεταξύ των συμβόλων της πληροφορίας (ISI) και β) την παρουσία του θορύβου στο σύστημα. Στη συνέχεια θα αναφερθεί το συγκεκριμένο παράδειγμα της ισοστάθμισης καναλιού που χρησιμοποιήθηκε στη πειραματική μελέτη των αλγορίθμων KNLMS, KAPSM.

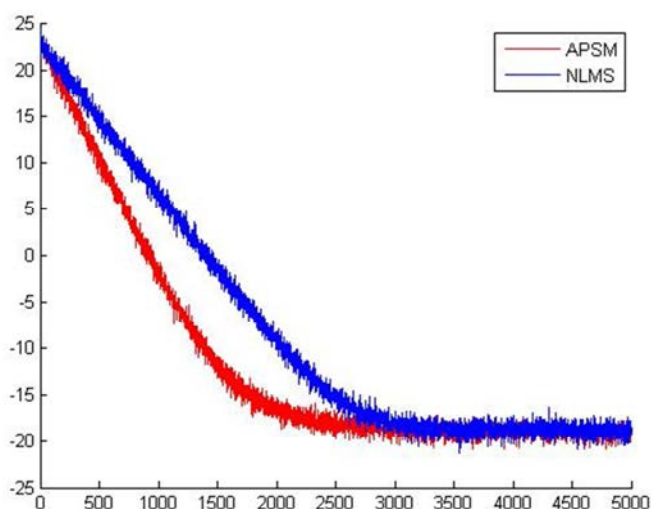
4.4 Τάυτιση καναλιού στην προσαρμοστική εκμάθηση και πειραματική μελέτη των προσαρμοστικών αλγορίθμων NLMS, APSM

Έστω κάθε ζευγάρι δεδομένων εκπαίδευσης $(y_n, x_n) \in \mathbb{R} \times \mathbb{R}^{N_w}$ και ο θόρυβος η_n . Τα δεδομένα δημιουργούνται σύμφωνα με το γνωστό μοντέλο

$$y_n = w^T x_n + \eta_n.$$

Τα πειράματά μας θα εκτελεστούν για $N=5.000$ δείγματα. Η παράμετρος, N_w , στο παράδειγμα της εφαρμογής ορίζεται ως $N_w=200$. Οι παράμετροι – συντελεστές των

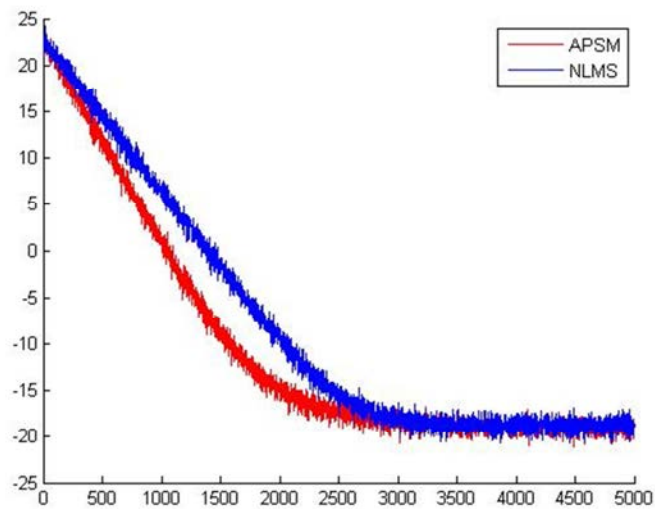
διανυσμάτων εισόδου, $w \in \mathbb{R}^{200}$, επιλέγονται τυχαία και ακολουθούν την Gaussian κατανομή με μέση τιμή ίση με το μηδέν και διακύμανση ίση με το 1, και κατά τη διάρκεια της εκτέλεσης της εφαρμογής μεταβάλλονται. Τα διανύσματα εισόδου συνοδεύονται από μια ακολουθία λευκού Gaussian θορύβου με μέση τιμή ίση με το μηδέν και διακύμανση $\sigma = 0.1$. Επίσης, θα παρουσιαστεί η μετρική απόδοσης του μέσου τετραγωνικού σφάλματος, $MSE_n = (y_n - x_n^T w_n)^2$. Τέλος, οι καμπύλες, οι οποίες θα παρουσιαστούν στη συνέχεια, προκύπτουν από 100 ανεξάρτητα πειράματα. Σκοπός της πειραματικής μελέτης είναι η παρατήρηση στη σύγκλιση των μεθόδων NLMS και APSM, των μεταξύ τους διαφορών στις καμπύλες και στη σύγκλιση αλλά και η συμπεριφορά του καθενός όταν αλλάζουν οι παράμετροι που τους χαρακτηρίζουν. Όλα τα πειράματα γίνονται με ταυτόχρονη εκτέλεση του NLMS και του APSM και την παράλληλη εμφάνιση των καμπυλών τους. Ο κώδικας του αλγορίθμου APSM αναφέρεται αναλυτικά στο Παράρτημα 1. Ας ξεκινήσουμε, λοιπόν, ορίζοντας αριθμητικά τις παραμέτρους του κάθε αλγορίθμου. Έστω $\mu = 0.5$ για τον NLMS και $\varepsilon = \sqrt{2}\sigma$, $q=30$, $\mu_n = 0.5M_n$ για τον APSM. Υπενθυμίζεται ότι η παράμετρος μ_n στον APSM παίρνει τιμές κατά τη διάρκεια εκτέλεσης του αλγορίθμου, αφού εξαρτάται από τη συνεχόμενα μεταβαλλόμενη σε κάθε επανάληψη παράμετρο M_n και θα πρέπει να ανήκει στο διάστημα $(0, 2M_n)$.



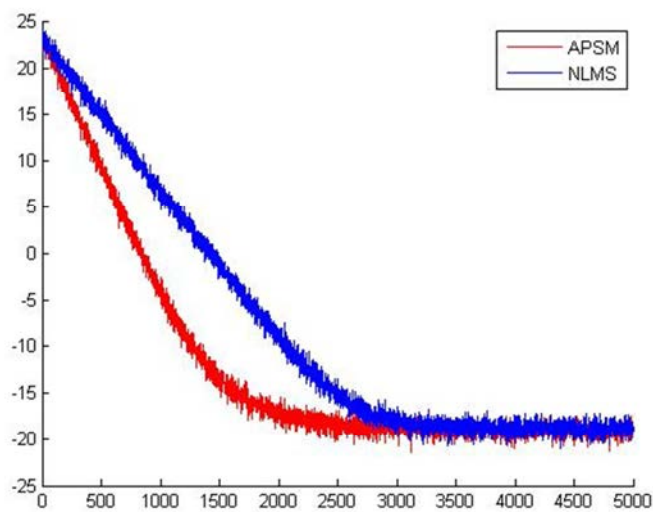
Σχήμα 4.3: MSE απόδοση για το πρώτο πείραμα, με $q=30$ στον APSM

Η εκτέλεση του πρώτου πειράματος έχει σκοπό να φανερώσει τη διαφορά στην καμπύλη της MSE απόδοσης μεταξύ των δυο αλγορίθμων. Στο Σχήμα 4.3, όπου απεικονίζεται το γράφημα του πρώτου πειράματος, παρατηρούμε ότι η εκτέλεση του APSM φέρει καλύτερη σύγκλιση σε σχέση με την αντίστοιχη του NLMS. Η ταχύτητα σύγκλισης του APSM δηλαδή φέρει σημαντική βελτίωση, ενώ και το μέσο τετραγωνικό σφάλμα βρίσκεται στο ίδιο επίπεδο με το αντίστοιχο του NLMS. Στα προηγούμενα κεφάλαια έγινε συζήτηση σχετικά με το πόσο επηρεάζουν οι παράμετροι του κάθε αλγορίθμου τη σύγκλισή τους. Επομένως, αφού κατανοήσαμε ότι ο APSM παράγει καλύτερα αποτελέσματα στην καμπύλη από τον NLMS, στα επόμενα πειράματα θα μελετήσουμε την σύγκλιση, αλλάζοντας τις παραμέτρους των δυο αλγορίθμων. Στο δεύτερο πείραμα, θα εξετάσουμε τι συμβαίνει στην περίπτωση που μεταβάλλουμε μόνο τη παράμετρο q στον APSM. Ο NLMS έχει ακριβώς τις ίδιες παραμέτρους με το πρώτο πείραμα. Γνωρίζουμε ότι ο APSM συμπεριφέρεται σαν τον NLMS, όταν $q=1$ και $\varepsilon=0$. Στα

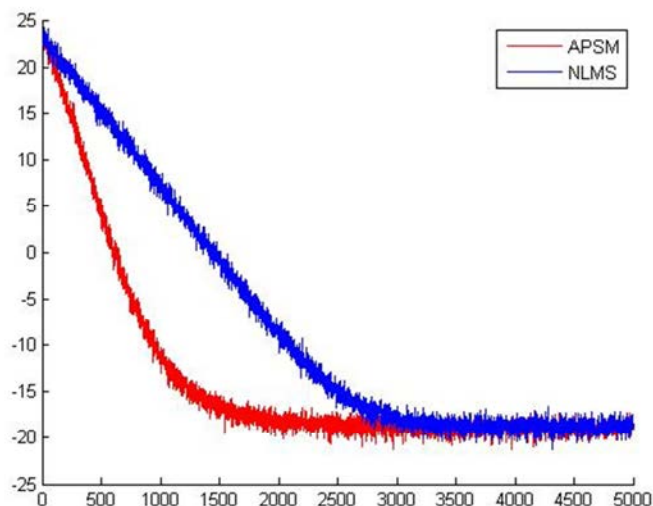
Σχήματα 4.3, 4.4, 4.5 και 4.6 έχουμε την εκτέλεση του πειράματος για τις τιμές $q=30$, $q=5$, $q=50$, $q=100$ αντίστοιχα. Παρατηρούμε, λοιπόν, ότι



Σχήμα 4.4: MSE απόδοση για το δεύτερο πείραμα, με $q=5$ στον APSM



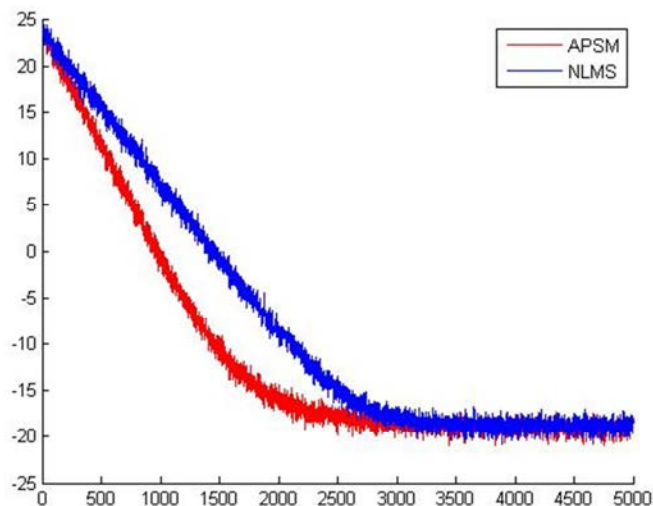
Σχήμα 4.5: MSE απόδοση για το δεύτερο πείραμα, με $q=50$ στον APSM



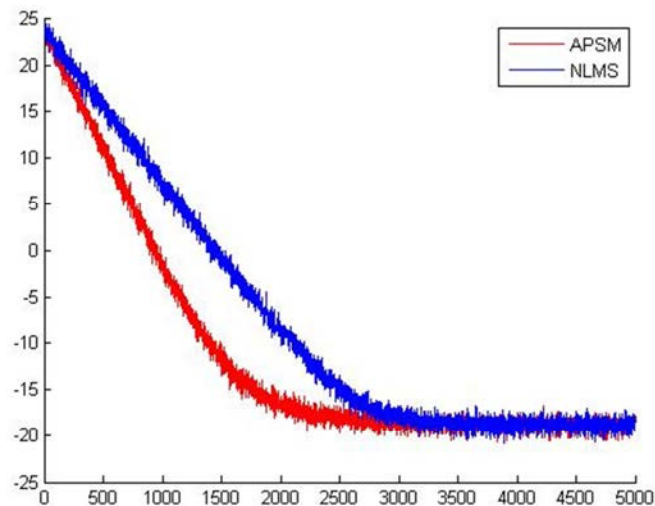
Σχήμα 4.6: MSE απόδοση για το δεύτερο πείραμα, με $q=100$ στον APSM

όσο μεγαλώνει ο αριθμός των κυρτών συνόλων, τόσο πιο γρήγορα επιτυγχάνεται σύγκλιση, με μικρή αύξηση του σφάλματος στη σταθερή κατάσταση. Θα πρέπει όμως να σημειωθεί ότι όσο μεγαλώνει η παράμετρος q , τόσο αυξάνεται η πολυπλοκότητα του αλγορίθμου, ο οποίος είναι της τάξης $O(qN)$.

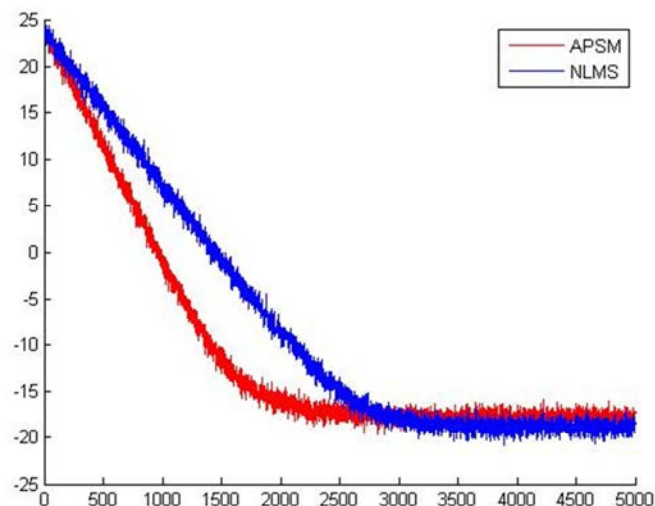
Στο τρίτο πείραμα θα εξετάσουμε την παράμετρο μ_n του APSM. Θα κρατήσουμε την τιμή $q=30$ και θα μεταβάλλουμε το μ_n , το οποίο θα λαμβάνει τις τιμές $0.2M_n$, $1.2M_n$, $1.9M_n$. Στα Σχήματα 4.7, 4.8, 4.9 έχουμε τις καμπύλες για τις παραπάνω τιμές αντίστοιχα.



Σχήμα 4.7: MSE απόδοση για το τρίτο πείραμα με $\mu_n=0.2M_n$

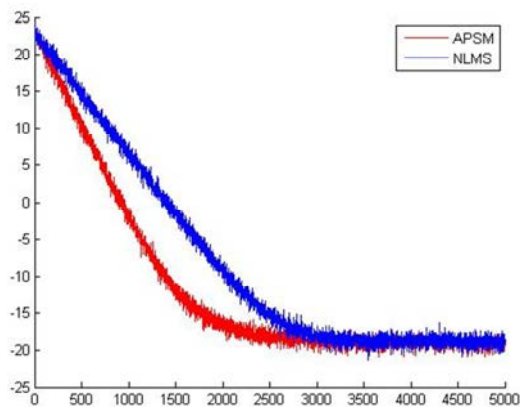


Σχήμα 4.8: MSE απόδοση για το τρίτο πείραμα με $\mu_n=1.2M_n$

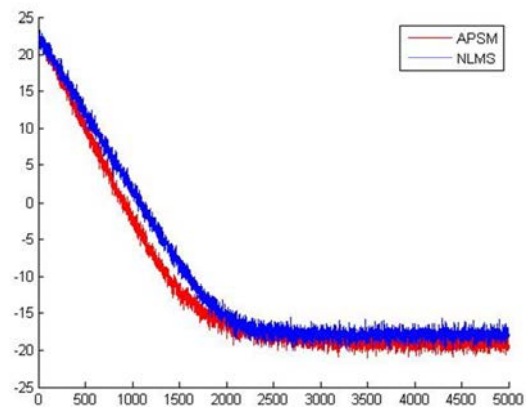


Σχήμα 4.9: MSE απόδοση για το τρίτο πείραμα με $\mu_n=1.9M_n$

Παρατηρούμε ότι ο APSM συμπεριφέρεται με τη λογική ότι όσο μικρότερη είναι η παράμετρος μ_n , τόσο πιο αργή είναι η σύγκλιση, αλλά το τελικό επίπεδο θορύβου είναι χαμηλότερο. Αυτή η συμπεριφορά παρουσιάζεται και στον NLMS όταν μεταβάλλεται το μ και για του λόγου του αληθές αρκεί να παρατηρήσουμε το Σχήμα 4.10, όπου η παράμετρος μ στο Σχήμα 4.10α είναι $\mu=0.5$ και στο 4.10β $\mu=0.8$.



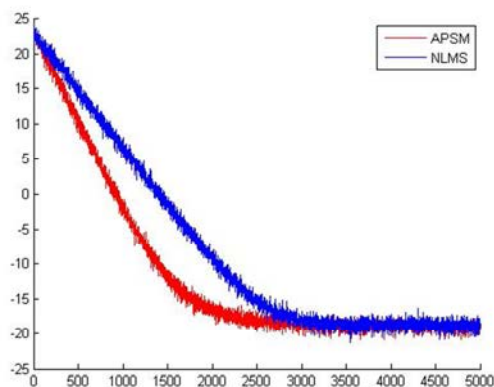
α)



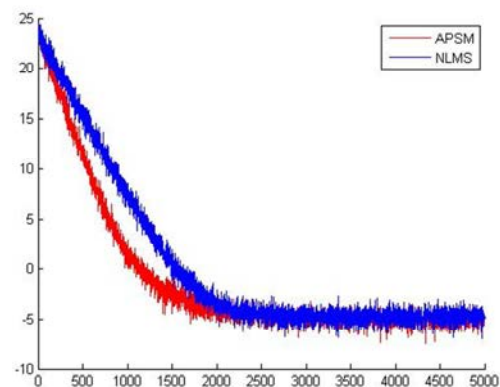
β)

Σχήμα 4.10: Όσο μικρότερη είναι η παράμετρος μ στον NLMS, τόσο πιο αργή είναι η σύγκλιση, με το μέσο τετραγωνικό σφάλμα να είναι μικρότερο. α) $\mu=0.5$, β) $\mu=0.8$.

Στο επόμενο πείραμα, μεταβάλλουμε την παράμετρο ϵ στον APSM, που καθορίζει το πλάτος των υπερπλάκων, μεταβάλλοντας τη συνεισφορά του θορύβου και θέτουμε $\sigma=0.5$. Η παράμετρος $\mu_n = 0.5M_n$ και $q=30$. Στον NLMS η παράμετρος $\mu=0.5$. Με βάση, λοιπόν, τα δεδομένα, συγκρίνουμε το Σχήμα 4.11α, που είναι παρόμοιο με το Σχήμα 4.3, με το Σχήμα 4.11β, όπου το ϵ έχει μεταβληθεί, αφού το σ έχει πάρει άλλη τιμή.



α)



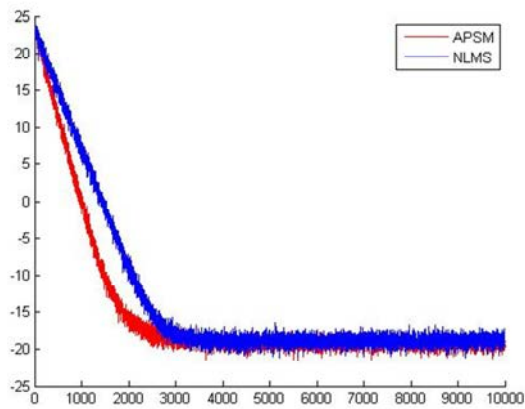
β)

Σχήμα 4.11: α) MSE σε dB ($\sigma=0.1$), β) MSE σε dB ($\sigma=0.5$). Με την αύξηση της συνεισφοράς του θορύβου μεγαλώνει η τιμή του σφάλματος σε σχέση με την αρχική μέτρηση.

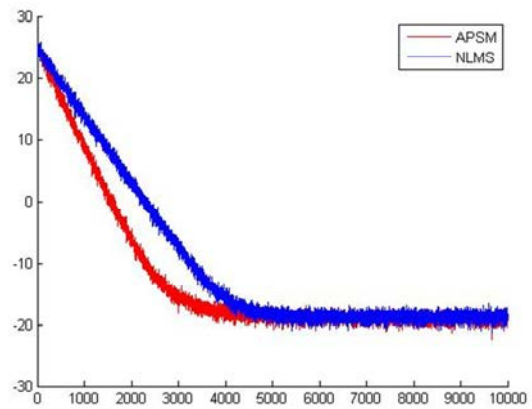
Παρατηρώντας το Σχήμα 4.11, συμπεραίνουμε ότι με την αύξηση της συνεισφοράς του θορύβου, παρατηρείται αύξηση της τιμής του σφάλματος, γεγονός που δείχνει ότι θέλουμε όσο το δυνατόν μικρό σ , δηλαδή η συνεισφορά να μην επηρεάζει το πρόβλημα μοντελοποίησης, σε βαθμό όπου το μέσο τετραγωνικό σφάλμα να παραμένει μικρό.

Στο τελευταίο πείραμα, μεταβάλλουμε τις διαστάσεις των δεδομένων εισόδου και χρησιμοποιούμε την παράμετρο q σε αναλογία με την παράμετρο Nw . Εκτελούμε για $N=10000$ δείγματα και η παράμετρος Nw ξεκινάει από 200 μέχρι και 1000, δίνοντας

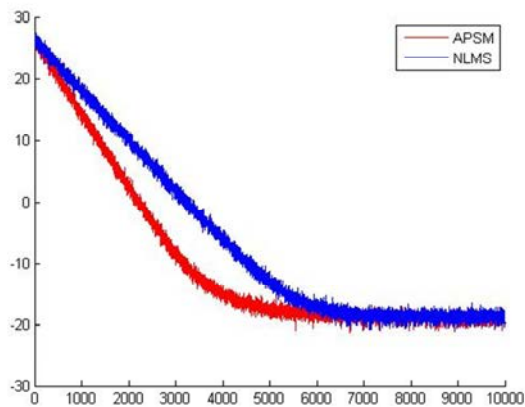
αντίστοιχες τιμές στην παράμετρο q από 20 μέχρι και 100. Έτσι, όπως βλέπουμε και στο σχήμα 4.12, όσο αυξάνεται η διάσταση των δεδομένων εισόδου και συγκεκριμένα η παράμετρος Nw , τόσο πιο αργή γίνεται η σύγκλιση με παράλληλη αύξηση στην τιμή του σφάλματος. Αξίζει να σημειωθεί ότι το πείραμα του Σχήματος 4.12ε εκτελείται για $N=20000$ δείγματα, διότι η σύγκλιση επιτυγχάνεται μετά τα 10000 δείγματα.



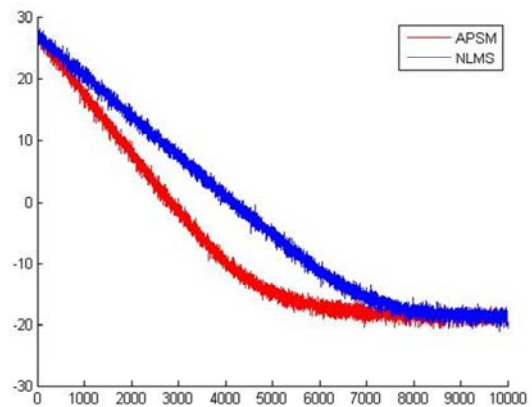
α)



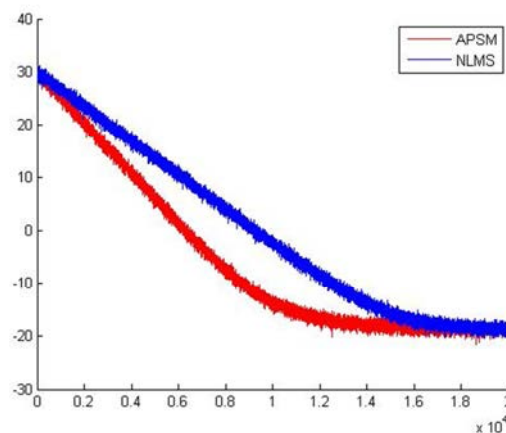
β)



γ)



δ)



ε)

Σχήμα 4.12: MSE απόδοση για α) $Nw=200, q=20, N=10000$, β) $Nw=300, q=30, N=10000$, γ) $Nw=400, q=40, N=10000$, δ) $Nw=500, q=50, N=10000$, ε) $Nw=1000, q=100, N=20000$. Με τη αύξηση της διάστασης και την αντίστοιχη αύξηση στην παράμετρο q

διαπιστώνουμε ότι η σύγκλιση γίνεται πιο αργή αλλά και μεγαλώνει η τιμή του σφάλματος και στις δυο μεθόδους.

4.5 Ισοστάθμιση καναλιού στην προσαρμοστική μάθηση και πειραματική μελέτη των προσαρμοστικών αλγορίθμων KNLMS, KAPSM

Αφού μελετήσαμε τα πειράματα των αλγορίθμων NLMS και APSM για γραμμικά συστήματα, έφτασε η στιγμή να μελετήσουμε και τους αλγορίθμους KNLMS και KAPSM που αναφέρονται σε μη γραμμικά συστήματα. Έστω, λοιπόν, μια μη γραμμική ισοστάθμιση καναλιού. Το μη γραμμικό κανάλι αποτελείται από ένα γραμμικό φίλτρο

$$t_n = -0.8y_n + 0.7y_{n-1} - 0.6y_{n-2} + 0.4y_{n-3}$$

και μια μη γραμμικότητα χωρίς μνήμη

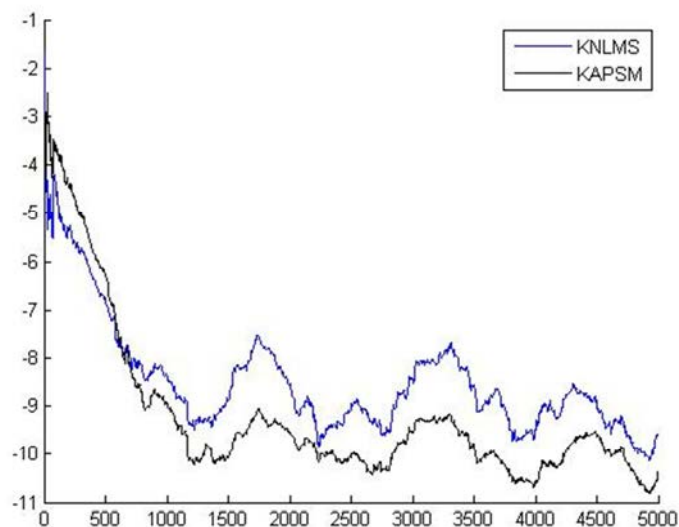
$$q_n = t_n + 0.08t_n^2.$$

Στο σήμα προστίθεται και ο λευκός Gaussian θόρυβος και μετά αυτό συμβολίζεται x_n . Ο σκοπός της ισοστάθμισης καναλιού είναι να μοντελοποιήσει ένα αντίστροφο φίλτρο, το οποίο επιδρά στην έξοδο του καναλιού, x_n , και αναπαράγει το αυθεντικό σήμα εισόδου, y_n . Στο τέλος, εφαρμόζουμε τον KNLMS και τον KAPSM στο σύνολο των δειγμάτων

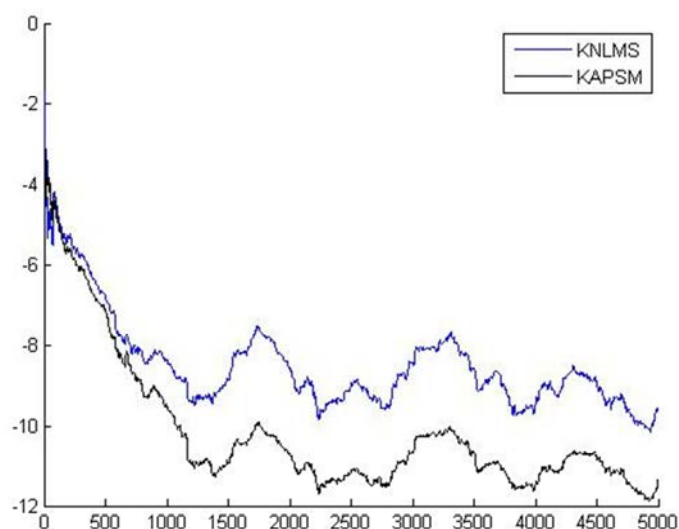
$$(x_n, y_{n-D}) := (x_n, x_{n-1}, \dots, x_{n-L+1}), y_{n-D},$$

όπου η παράμετρος $L > 0$ αποτελεί το μήκος του ισοσταθμιστή και D είναι η καθυστέρηση ισοσταθμίσης. Η έξοδος του ισοσταθμιστή κάθε χρονική στιγμή n , παρέχει την εκτίμηση του y_{n-D} . Όλα τα πειράματα εκτελούνται για $N=5000$ δείγματα και το μέσο τετραγωνικό σφάλμα προκύπτει από την εκτέλεση 100 πειραμάτων. Το σήμα εισόδου είναι μια Gaussian τυχαία μεταβλητή μηδενικής τιμής και για την συνάρτηση πυρήνα $k(\cdot)$ χρησιμοποιείται Gaussian πυρήνας με $\sigma=5$. Το μήκος του φίλτρου L , έχει την τιμή $L=5$ και καθυστέρηση $D=2$. Κάθε πείραμα εκτελείται δυο φορές, την πρώτη για την περίπτωση όπου η αραιώση της λύσης (sparsification) γίνεται στην περίπτωση της προβολής πάνω σε κλειστή μπάλα, και τη δεύτερη όπου χρησιμοποιείται το κριτήριο καινοτομίας (novelty criterion). Για την πρώτη περίπτωση η ακτίνα $\delta=10$ και για τη δεύτερη, τα κατώφλια $\delta_1 = 0.15$ και $\delta_2 = 0.2$. Στα πειράματα αυτά θα παρατηρήσουμε τις διαφορές στη συμπεριφορά των καμπυλών των αλγορίθμων KNLMS και KAPSM και θα μελετήσουμε τη σύγκλισή τους. Η λύση στον APSM πάντα εκχωρείται στη διαδικασία της αραιώσης (sparsification) και μάλιστα θα δούμε δυο τεχνικές αραιώσης, την τεχνική της προβολής πάνω σε κλειστή μπάλα και το κριτήριο καινοτομίας (novelty criterion).

Στο πρώτο πείραμα θα ασχοληθούμε με την ταυτόχρονη εκτέλεση των αλγορίθμων KNLMS και KAPSM, θέτοντας τιμές στις παραμέτρους τους και θα διαπιστώσουμε τις διαφορές των καμπυλών τους στον τρόπο σύγκλισης. Ο κώδικας του αλγορίθμου KAPSM αναφέρεται αναλυτικά στο Παράρτημα 1. Έτσι, για τον KNLMS ορίζουμε $\mu=1$ και για τον KAPSM ορίζουμε $\mu_n=M_n$, $\varepsilon=10^{-5}$, $q=4$. Στο Σχήμα 4.12 απεικονίζονται οι καμπύλες και των δυο αλγορίθμων με την τεχνική της προβολής πάνω σε κλειστή μπάλα. Στο Σχήμα 4.13 οι αντίστοιχες καμπύλες όπου εφαρμόζεται το novelty criterion. Παρατηρώντας τα Σχήματα 4.13 και 4.14, γίνονται αντιληπτά τα εξής δυο συμπεράσματα: α) ο αλγόριθμος KAPSM συγκλίνει καλύτερα από τον KNLMS και β) με το novelty criterion ο KAPSM παρουσιάζει μικρότερο μέσο τετραγωνικό σφάλμα από ότι η προβολή πάνω σε κλειστή μπάλα.

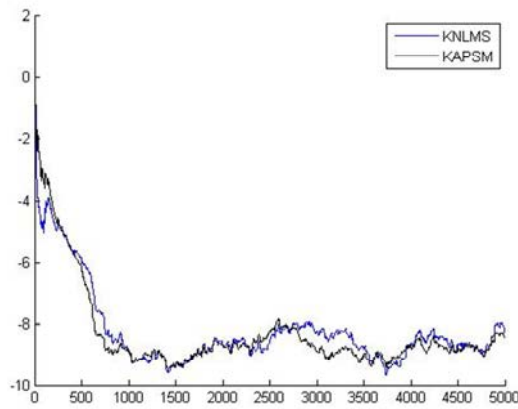


Σχήμα 4.13: MSE απόδοση για το πρώτο πείραμα με προβολή πάνω σε κλειστή σφαίρα ακτίνας $\delta=10$

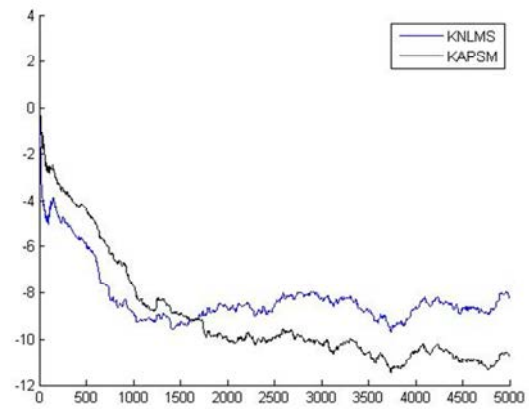


Σχήμα 4.14: MSE απόδοση για το πρώτο πείραμα με novelty criterion

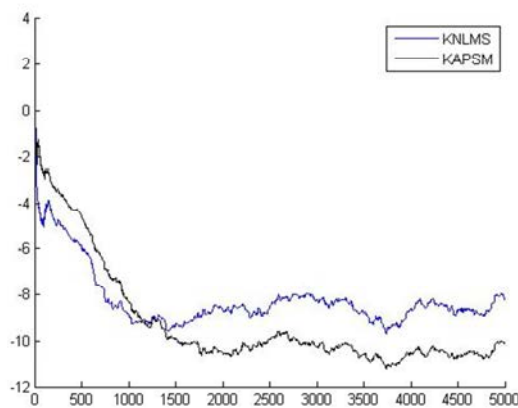
Στο δεύτερο πείραμα μελετάμε τη συμπεριφορά του KAPSM ως προς τις διάφορες τιμές που μπορεί να πάρει η παράμετρος q . Οι τιμές που θα πάρει η παράμετρος q , είναι $q=2$, $q=4$, $q=8$ και $q=16$. Παρατηρώντας το Σχήμα 4.15, και αρχικά το 4.15α, διαπιστώνουμε ότι για $q=2$ η διαφορά στη σύγκλιση με τον KNLMS δεν είναι μεγάλη, όμως για τις άλλες περιπτώσεις η σύγκλιση στον KAPSM είναι πολύ καλύτερη. Επιπλέον, παρατηρούμε ότι όσο αυξάνεται η παράμετρος q , τόσο πιο γρήγορη γίνεται η σύγκλιση, με παράλληλη όμως μικρή αύξηση του μέσου τετραγωνικού σφάλματος. Άλλο ένα μειονέκτημα έγκειται στο ότι μεγαλώνει και η πολυπλοκότητα του αλγορίθμου.



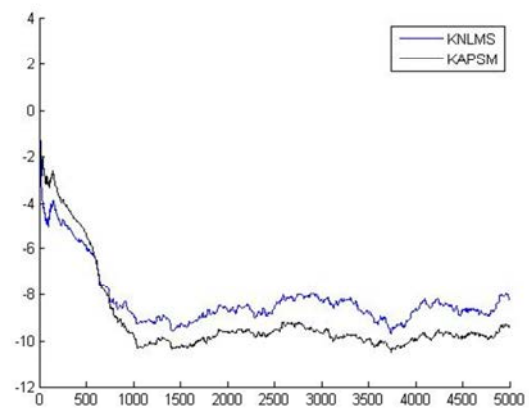
α)



β)



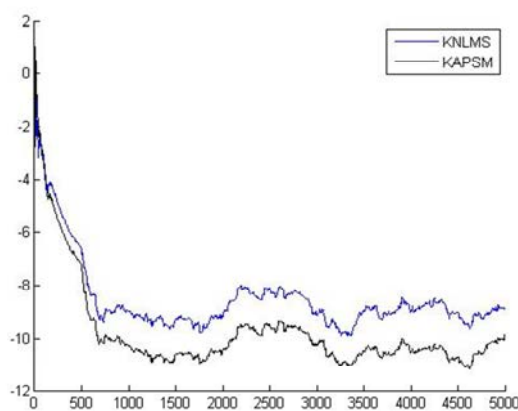
γ)



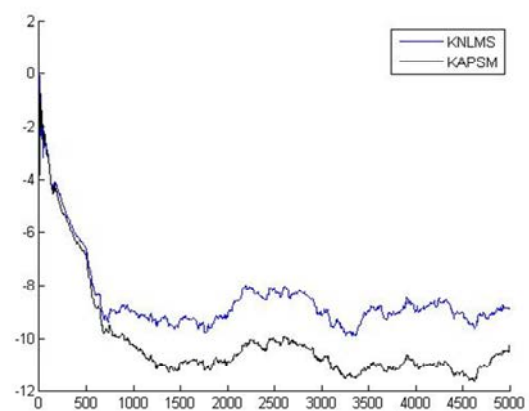
δ)

Σχήμα 4.15: MSE απόδοση για το δεύτερο πείραμα με προβολή πάνω σε κλειστή σφαίρα ακτίνας $\delta=10$ για α) $q=2$, β) $q=4$, γ) $q=8$ και δ) $q=16$.

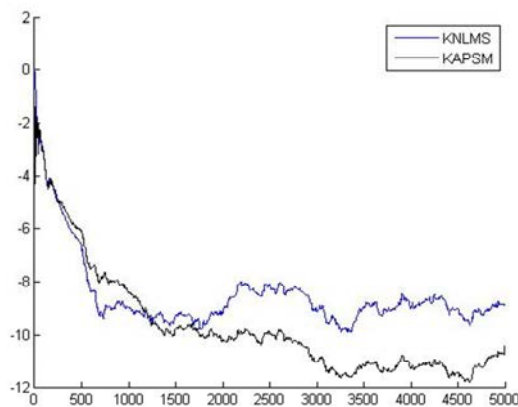
Αφού μελετήσαμε την παράμετρο q στην περίπτωση της προβολής πάνω σε κλειστή μπάλα, θα δούμε τι συμβαίνει και στην περίπτωση του novelty criterion. Παρατηρώντας,



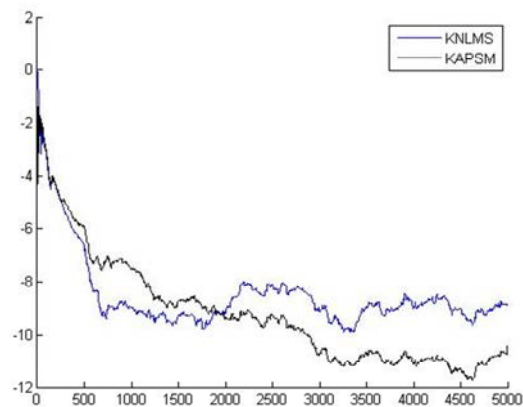
α)



β)



γ)

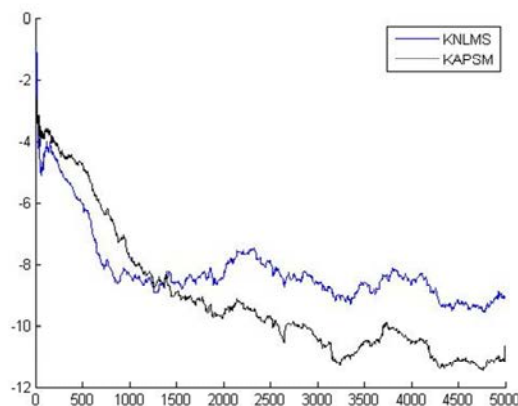


δ)

Σχήμα 4.16: MSE απόδοση για το δεύτερο πείραμα με novelty criterion για α) $q=2$, β) $q=4$, γ) $q=8$ και δ) $q=16$.

το Σχήμα 4.16 και μελετώντας τις καμπύλες για τις τιμές της παραμέτρου q , διαπιστώνουμε ότι όσο αυξάνεται το q , η σύγκλιση επιτυγχάνεται πιο αργά, αλλά το μέσο τετραγωνικό σφάλμα μειώνεται.

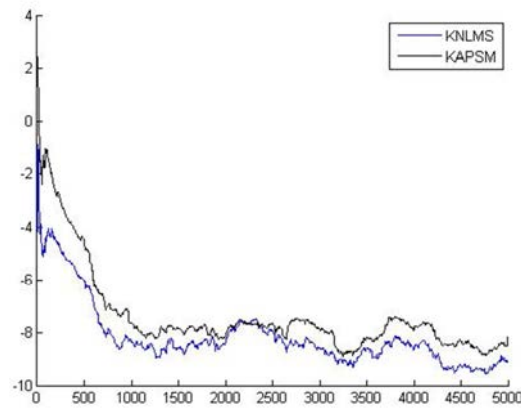
Στο τρίτο και τελευταίο πείραμα, θα μελετήσουμε τη παράμετρο μ_n του KAPSM και για τις δυο περιπτώσεις της αραιώσης (sparsification). Η παράμετρος q έχει την τιμή $q=4$, οι υπόλοιποι παράμετροι παραμένουν ίδιοι όπως το πρώτο πείραμα, και θα εξετάσουμε τι συμβαίνει στις καμπύλες για $\mu_n=0.3M_n$, $\mu_n=1M_n$ και $\mu_n=1.9M_n$. Αρχικά, παρατηρώντας το Σχήμα 4.17, που αναφέρεται στην προβολή πάνω σε κλειστή μπάλα, διαπιστώνουμε ότι όσο η παράμετρος μ_n αυξάνεται, τόσο πιο γρήγορη είναι η σύγκλιση, αλλά με αύξηση του σφάλματος. Επιπλέον, παρατηρούμε στο Σχήμα 4.17γ ότι ο KNLMS επιτυγχάνει σύγκλιση με πιο μικρό σφάλμα από τον KAPSM, για την ακτίνα $\delta=10$ της κλειστής μπάλας που έχει χρησιμοποιηθεί, για $\mu_n=1.9M_n$. Τέλος, παρατηρώντας την περίπτωση που αναφέρεται στο novelty criterion και το Σχήμα 4.18 διαπιστώνουμε ότι όσο αυξάνεται το μ_n , τόσο πιο γρήγορη σύγκλιση επιτυγχάνεται, αλλά με μικρή αύξηση στο μέσο τετραγωνικό σφάλμα.



α)

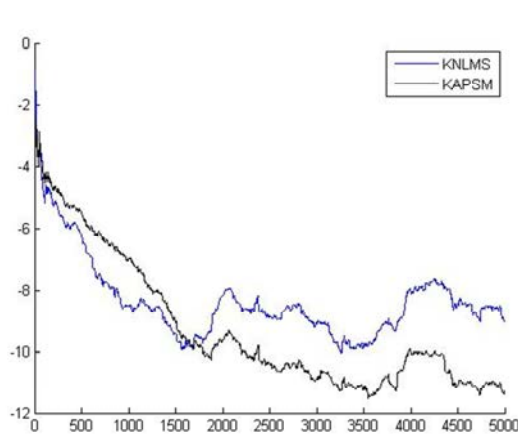


β)

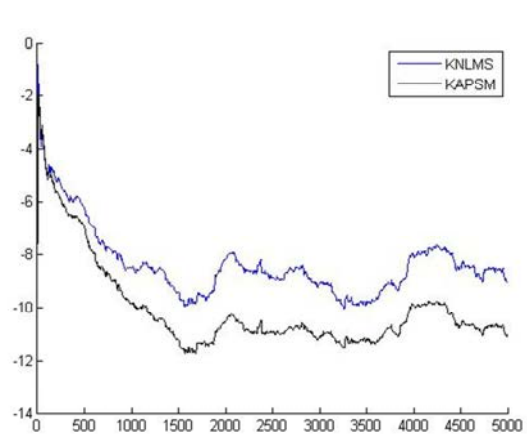


γ)

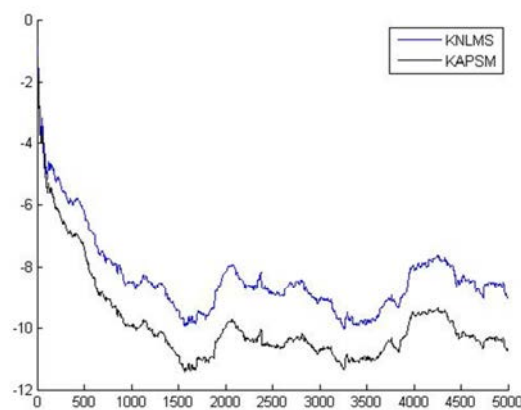
Σχήμα 4.17: MSE απόδοση για το τρίτο πείραμα με προβολή σε κλειστή μπάλα ακτίνας $\delta=10$ για α) $\mu_n = 0.3M_n$, β) $\mu_n = 1M_n$, γ) $\mu_n = 1.9M_n$



α)



β)



γ)

Σχήμα 4.18: MSE απόδοση για το τρίτο πείραμα με novelty criterion για α) $\mu_n = 0.3M_n$, β) $\mu_n = 1M_n$, γ) $\mu_n = 1.9M_n$

ΣΥΜΠΕΡΑΣΜΑΤΑ

Συμπερασματικά, η μηχανική εκμάθηση:

1. διαδραματίζει σημαντικό ρόλο στην επιστήμη των υπολογιστών και σε πολλές εφαρμογές της καθημερινής ζωής, όπως στην επικοινωνία.
2. περιλαμβάνει μεθόδους και αλγορίθμους, οι οποίοι όπως είδαμε χρησιμοποιούνται ευρέως και αποτελεσματικά.
3. περιέχει τη μελέτη των καμπυλών και του μέσου τετραγωνικού σφάλματος αρκετών αλγορίθμων, όπως ο APSM και ο NLMS, που οδηγεί στην κατανόηση της συμβολής τους στην προσαρμοστική μάθηση τόσο σε γραμμικά συστήματα όσο και σε μη γραμμικά.
4. γνωρίζει μεγάλη ανάπτυξη και συνεχώς βελτιώνεται με τη χρήση μεθόδων που εξάγουν όλο και καλύτερα αποτελέσματα στην επίλυση προβλημάτων μοντελοποίησης.

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	Ελληνικός Όρος
abnormal	ασυνήθης
adaptive	προσαρμοστικός
classification	ταξινόμηση
coherence	συνοχή
convex	κυρτός
feature	χαρακτηριστικό
hyperslab	υπερπλάκα
kernel	πυρήνας
learnable	επίκτητος
novelty	καινοτομία
operator	παράγοντας
overfitting	υπερπροσαρμογή
plant	έργο
quantization	κβάντιση
redundant	πλεονάζων
regression	παλινδρόμηση
regularization	τακτοποίηση
relaxed	χαλαρή
reproducing	αναπαραγωγή
sparsification	αραίωση
surprise	αιφνιδιασμός
tap	σύνδεση

ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

APSM	Adaptive Projected Subgradient Method
KAPSM	Kernel Adaptive Projected Subgradient Method
KNLMS	Kernel Normalized Least Mean Square
LMS	Least Mean Square
LS	Loss Function
MSE	Mean Square Error
RKHS	Reproducing Kernel Hilbert Space
RLS	Recursive Least Square
ΕΚΠΑ	Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

ΠΑΡΑΡΤΗΜΑ Ι

Όλοι οι αλγόριθμοι που χρησιμοποιήθηκαν για την πειραματική μελέτη εκτελέστηκαν στο μαθηματικό προγραμματιστικό περιβάλλον του MATLAB. Οι κώδικες των αλγορίθμων NLMS και KNLMS σε MATLAB προέρχονται από το υλικό του Παντελή Μπουμπούλη, το οποίο είναι αναρτημένο στην ιστοσελίδα <http://bouboulis.mysch.gr/>. Οι αλγόριθμοι APSM και KAPSM σε MATLAB υλοποιήθηκαν από μέρους μου. Λόγω της πολυπλοκότητας του αλγορίθμου KAPSM, υλοποιήθηκε ο κώδικάς του και στη προγραμματιστική γλώσσα C#, με σκοπό την ταχύτερη εκτέλεσή του. Παρακάτω παρατίθενται οι κώδικες των APSM, KAPSM σε MATLAB αλλά και ο KAPSM σε C#.

1) Adaptive Projected Subgradient Method (APSM) σε MATLAB

```
function [ e_n ] = APSM( x, y, N, Nu, mu, eps, q, delta)

err=1e-10;
e_n=zeros(N,1);
u=zeros(Nu,1);
omega=zeros(N,1);
P=zeros(Nu,N);
for n=1:N
    sum=0;
    Mn=0;
    y_hat= u*x(:,n);
    e_n(n)=y(n)-y_hat;
    if(n<=q-1)
        for i=1:n
            omega(i)=1/n;
            dif= u*x(:,i) - y(i);
            if (dif < -eps)
                b= (-dif-eps) / norm(x(:,i)).^2;
            elseif (dif > eps)
                b= (-dif+eps) / norm(x(:,i)).^2;
            else
                b=0;
            end
            P(:,i)=u+ b*x(:,i);
            sum= sum + omega(i) * P(:,i);
        end
        if norm(sum-u)<=err
            Mn = 1;
        else
            for k=1:n
                Mn=Mn+omega(k)*(norm(P(:,k)-u)).^2/norm(sum-u).^2;
            end
        end
    else
        for i=n-q+1:n
            omega(i)=1/q;
            dif= u*x(:,i) - y(i);
```

```

        if (dif < -eps)
            b= (-dif-eps) / norm(x(:,i)).^2;
        elseif (dif > eps)
            b= (-dif+eps) / norm(x(:,i)).^2;
        else
            b=0;
        end
        P(:,i) =u+ b*x(:,i);
        sum= sum + omega(i) * P(:,i);
    end
    if norm(sum-u)<=err
        Mn = 1;
    else
        for k=n-q+1:n
            Mn=Mn+omega(k)*(norm(P(:,k)-u).^2)/norm(sum-u).^2;
        end
    end
    end
    mu2=mu*Mn;
    u= (1-mu2)*u + mu2 * sum;
    if( norm(u)>delta)
        u=(delta/norm(u))*u;
    end

end
end

```

2) Kernel Adaptive Projected Subgradient Method (KAPSM) σε MATLAB

```

function [e_n] = KAPSM( x, y, N, mu, eps, q, delta,sigma,spars,thres1,thres2)

centers=[];
Nc=0;
u=zeros(1,N);

for n=1:N
    Mn=0;
    sum=0;
    sum1=0;
    summary=0;
    for i=1:Nc
        sum=sum+u(i)*Kappa(centers(i,:)',x(n,:)',sigma);
    end
    y_hat=sum;
    e_n(n)=y(n)-y_hat;
    if(n<q)
        for i=1:n
            sum=0;
            for j=1:n-1
                sum=sum+u(j)*Kappa(centers(j,:)',x(i,:)',sigma);
            end
            s=Kappa(x(i,:)',x(i,:)',sigma);

```

```

    dif= sum- y(i);
    if (dif < -eps)
        b(i)= (-dif-eps) / s;
    elseif (dif > eps)
        b(i)= (-dif+eps) / s;
    elseif( dif>=-eps && dif<=eps)
        b(i)=0;
    end

    omega(i)=1/n;
    P(i)=u(i)+b(i)*Kappa(x(i,:),x(n,:),sigma);
    summary=summary+omega(i)*P(i);
    for k=1:i
        sum1=sum1+ b(i)*b(k) * s;
    end
end
for k=1:n
    if(sum1==0)
        Mn=1;
    else
        Mn=Mn+b(k).^2 *s/sum1;
    end
end
else
    for i=n-q+1:n
        sum=0;
        for j=1:Nc
            sum=sum+u(j)*Kappa(centers(j,:),x(i,:),sigma);
        end
        s=Kappa(x(i,:),x(i,:),sigma);
        dif= sum- y(i);
        if (dif < -eps)
            b(i)= (-dif-eps) / s;
        elseif (dif > eps)
            b(i)= (-dif+eps) / s;
        elseif( dif>=-eps && dif<=eps)
            b(i)=0;
        end
        omega(i)=1/q;
        P(i)=u(i)+b(i)*Kappa(x(i,:),x(n,:),sigma);
        summary=summary+omega(i)*P(i);
        for k=n-q+1:i
            sum1=sum1+ b(i)*b(k) * s;
        end
    end
    for k=n-q+1:n
        if(sum1==0)
            Mn=1;
        else
            Mn=Mn+b(k).^2 *s/sum1;
        end
    end
end

```

```

end
if(spars==1) % closed ball
    if( norm(u)>delta)
        u=(delta/norm(u))*u;
    end
    Nc=Nc+1;
    centers(Nc,:)=x(n,:);
    mu=1*Mn;
    u(Nc)=(1-mu)*u(Nc)+mu*summary;
elseif(spars==2) % novelty criterion
    min=1000000000;
    for k=1:Nc
        dist=norm(((x(n,:)-centers(k,:))));
        if (dist<min)
            min=dist;
        end;
    end
    if (min<thres1) %do not add this center
        add_this_center=0;
    else
        if abs(e_n(n))>thres2
            add_this_center=1;
        else
            add_this_center=0;
        end;
    end;
    if (add_this_center==1)
        Nc=Nc+1;
        centers(Nc,:)=x(n,:);
        mu=1*Mn;
        u(Nc)=(1-mu)*u(Nc)+mu*summary;
    else
        u(Nc+1)=0;
    end
else % no sparsification
    Nc=Nc+1;
    centers(Nc,:)=x(n,:);
    mu=1*Mn;
    u(Nc)=(1-mu)*u(Nc)+mu*summary;
end
end
end

```

3) Kernel Adaptive Projected Subgradient Method (KAPSM) σε C#

```

#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "matrix.h"
#include "mex.h"

```

```

double Kappa(double *x,double *y,int K,double sigma);

void mexFunction( int nlhs, mxArray *plhs[],
                  int nrhs, const mxArray *prhs[] )
{
    double *x, *y, N, eps, q, delta,sigma,spars,thres1,thres2;
    double *e_n, *u, *P, *omega, *b, **centers;
    int n,xcol,xrow,i,j,Nc,k,add_this_center;
    double sum, summary, y_hat, *temp, dif, mu, s, norm,Mn,sum1,distance,min;
    if (nrhs!=11)
        mexErrMsgTxt(" 11 inputs required.");
    if (nlhs!=1)
        mexErrMsgTxt(" 1 output required.");

    /*1st input */
    x=mxGetPr(prhs[0]);
    /*2nd input */
    y=mxGetPr(prhs[1]);
    /*3rd input */
    N = mxGetScalar(prhs[2]);
    /*4th input*/
    mu = mxGetScalar(prhs[3]);
    /*5th input */
    eps=mxGetScalar(prhs[4]);
    /*6th input */
    q=mxGetScalar(prhs[5]);
    /*7th input */
    delta=mxGetScalar(prhs[6]);
    /*8th input */
    sigma=mxGetScalar(prhs[7]);
    /*9th input */
    spars=mxGetScalar(prhs[8]);
    /*10th input */
    thres1=mxGetScalar(prhs[9]);
    /*11th input */
    thres2=mxGetScalar(prhs[10]);

    xcol=mxGetN(prhs[0]);
    xrow=mxGetM(prhs[0]);

    e_n= malloc(N*sizeof(double));
    u=malloc(N*sizeof(double));
    b=malloc(N*sizeof(double));
    P=malloc(N*sizeof(double));
    centers=malloc(xrow*sizeof(double *));
    omega=malloc(N*sizeof(double));

    if(centers==NULL)
        return -1;

    for (i=0;i<xrow;i++){
        *(centers+i)=malloc(xcol*sizeof(double));
        if(*(centers+i)==NULL)
            return -1;
    }
    for (i=0;i<N;i++){
        e_n[i]=0;
        omega[i]=0;
        u[i]=0;
        P[i]=0;
        b[i]=0;
        for (j=0;j<xcol;j++){
            centers[i][j]=0;
        }
    }
}

```

```

}

Nc=0;
for(n=0;n<N;n++){
    Mn=0;
    sum1=0;
    y_hat=0;
    summary=0;
    for(i=0;i<Nc;i++){
        y_hat=y_hat + u[i]* Kappa(centers[i],&(x[n]),xcol,sigma);
    }
    e_n[n]= y[n]-y_hat;

    if(n<q-1){
        for(i=0;i<=n;i++){
            omega[i]=1/(n+1);
            sum=0;
            for(j=0;j<=n-1;j++){
                sum=sum + u[j]* Kappa(centers[j],&(x[i]),xcol,sigma);
            }
            s= Kappa(&(x[i]),&(x[i]),xcol,sigma);
            dif=sum-y[i];

            if (dif<-eps)
                b[i]=(-dif-eps)/s;
            else if(dif>eps)
                b[i]=(-dif+eps)/s;
            else if(dif>=-eps && dif<=eps)
                b[i]=0;
            P[i]=u[i]+b[i]* Kappa(&(x[i]),&(x[n]),xcol,sigma);
            summary=summary+omega[i]*P[i];
            for(k=0;k<=i;k++)
                sum1=sum1+ b[i]*b[k]*s;
        }
        if(sum1==0)
            Mn=1;
        else
            for (k=0;k<=n;k++)
                Mn=Mn+b[k]*b[k]*s/sum1;
    }
    else{
        for (i=n-q+1;i<=n;i++){
            omega[i]=1/q;
            sum=0;
            for(j=0;j<Nc;j++){
                sum=sum + u[j]* Kappa(centers[j],&(x[i]),xcol,sigma);
            }
            s= Kappa(&(x[i]),&(x[i]),xcol,sigma);
            dif=sum-y[i];

            if (dif<-eps)
                b[i]=(-dif-eps)/s;
            else if(dif>eps)
                b[i]=(-dif+eps)/s;
            else if(dif>=-eps && dif<=eps)
                b[i]=0;
            P[i]=u[i]+b[i]* Kappa(&(x[i]),&(x[n]),xcol,sigma);
            summary=summary+omega[i]*P[i];
            for(k=n-q+1;k<=i;k++)
                sum1=sum1+ b[i]*b[k] * s;
        }
        if(sum1==0)
            Mn=1;
        else

```

```

        for (k=n-q+1;k<=n;k++)
            Mn=Mn+b[k]*b[k]*s/sum1;
    }
    if(spars==1){ //closed ball
        norm=0;
        for (i=0;i<N;i++){
            norm=norm+u[i]*u[i];
        }
        norm=sqrt(norm);
        if(norm>delta)
            for(i=0;i<N;i++)
                u[i]=(delta/norm)*u[i];
        mu=1*Mn;
        u[Nc]=(1-mu)*u[Nc]+mu*summary;
        for(i=0;i<xcol;i++)
            centers[Nc][i]= (*(&(x[n+i*xrow])));
        Nc=Nc+1;
    }
    else if(spars==2){// novelty criterion
        min=1000000000;

        for (k=0;k<Nc;k++){
            distance=0;
            for(j=0;j<xcol;j++){
                distance=distance+pow((( *(&(x[n+j*xrow])))-
centers[k][j])),2);
            }
            distance=sqrt(distance);
            if (distance<min)
                min=distance;
        }
        if (min<thres1) //not add this center
            add_this_center=0;
        else{
            if (fabs(e_n[n])>thres2)
                add_this_center=1;
            else
                add_this_center=0;
        }
        if (add_this_center==1){
            mu=1*Mn;
            u[Nc]=(1-mu)*u[Nc]+mu*summary;
            for(i=0;i<xcol;i++)
                centers[Nc][i]= (*(&(x[n+i*xrow])));
            Nc=Nc+1;
        }
        else
            u[Nc]=0;
    }
    else if(spars==0){// no sparsification
        mu=1*Mn;
        u[Nc]=(1-mu)*u[Nc]+mu*summary;
        for(i=0;i<xcol;i++)
            centers[Nc][i]= (*(&(x[n+i*xrow])));
        Nc=Nc+1;
    }
    else{
        printf("Wrong string for sparsification\n");
        return -1;
    }
}

/*Output*/

```

```

    plhs[0]=mxCreateDoubleMatrix(N,1,mxREAL);
    temp = mxGetPr(plhs[0]);
    for(i=0;i<N;i++)
        temp[i]=e_n[i];
    //free(temp);

}

double Kappa(double *x,double *y,int K,double sigma){

    //printf("%f\n",*(x+4));
    double norm,val;
    int i;

    norm=0;
    for (i=0;i<K;i++){
        norm = norm+(x[i]-y[i])*(x[i]-y[i]);
    }

    val=exp(-norm/(sigma*sigma));
    return ( val );

}

```


ΑΝΑΦΟΡΕΣ

- [1] Theodoridis S., *Machine Learning: A Bayesian and Optimization Perspective*, Academic Press, 2015.
- [2] P.S.R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, third ed., Springer, 2008.
- [3] S. Haykin, *Adaptive Filter Theory*, third ed., Prentice-Hall, New Jersey, 1996.
- [4] A.H. Sayed, *Fundamentals of Adaptive Filtering*, John Wiley & Sons, New Jersey, 2003.
- [5] Levinson N. "The Wiener error criterion in filter design and prediction," *Journal of Mathematical Physics*, Vol. 25, pp. 261-278, 1947
- [6] Theodoridis S., Slavakis K., and Yamada I., Adaptive Learning in a World of Projections, *A unifying framework for linear and nonlinear classification and regression tasks*, IEEE Signal Processing Magazine, pp. 101-114, Jan 2011.
- [7] Bregman L.M. "The method of successive projections for finding a common point of convex sets," *Soviet Math. Dokl.*, Vol. 6, pp. 688-692, 1965.
- [8] Pierra G. "Decomposition through formalization in a product space," *Mathematical Programming*, Vol. 8, pp. 96-115, 1984
- [9] Chouvardas S., Slavakis K., Theodoridis S., Yamada I. "Stochastic Analysis of Hyperslab-based adaptive projected subgradient method under boundary noise," *IEEE Signal Processing Letters*, pp. 729-732, 2013.
- [10] M.A. Aizerman, E.M. Braverman, L.I. Rozonoer, Theoretical foundations of the potential function method in pattern recognition learning, *Autom. Remote Control* 25, pp 821-837, 1964.
- [11] B. Boser, I. Guyon, V. Vapnik, A training algorithm for optimal margin classifiers, *Fifth Annual Workshop on Computational Learning Theory*, Pittsburgh, pp. 144-152, 1992.
- [12] Slavakis K., Bouboulis P., Theodoridis S. "Online learning in reproducing kernel Hilbert spaces," *Academic Press Library in Signal Processing: Volume 1 Signal processing Theory and Machine Learning*, Vol. 1, pp. 883-987, Elsevier, 2013.
- [13] Kimeldorf G., Wahba G. "Some results on Tchebychffian spline functions," *Journal of Mathematical Analysis and Applications*, Vol. 33, pp. 82-95, 1971.
- [14] Bousquet O., Elisseeff A. "Stability and generalization," *Journal of Machine Learning Research*, Vol. 2, pp 599-526, 2002.
- [15] Poggio T., Smole S. "The mathematics of learning: Dealing with data," *American Mathematical Society Notice*, Vol. 50(5), pp. 537-544, 2003.
- [16] Evgeniou T., Pontil M., Poggio T. "Regularization networks and support vector machines," *Advances in Computational Mathematics*, Vol. 13, pp 1-50, 2000.
- [17] Miccheli C.A., Pontil M. "Learning the kernel function via regularization," *Journal of Machine Learning Research*, Vol. 6, pp. 1099-1125, 2005.
- [18] Liu W., Pokharel P., Principe J. "The kernel least mean squares algorithm," *IEEE Transactions on Signal Processing*, Vol. 56(2), pp. 543-554, 2008.
- [19] Richard C., Bermudez J., Honeine P. "Online prediction of time series data with kernels," *IEEE Transactions on Signal Processing*, Vol. 57(3), pp. 1058-1067, 2009.
- [20] Liu W., Principe J.C., Haykin S. *Kernel Adaptive Filtering: A Comprehensive Introduction*, John Wiley, 2010.
- [21] Chen B., Zhao S., Zhu P., Principe J.C. "Quantized kernel least mean square algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 23(1), pp. 22-32, 2012.
- [22] Slavakis K., Bouboulis P., Theodoridis S. "Online learning in reproducing kernel Hilbert spaces," *E-reference for Signal Processing*, S. Theodoridis, R. Chellapa (Eds), Academic Press, 2013.