Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ

# ΤΟΠΙΚΗ ΓΡΑΜΜΙΚΗ ΕΝΣΩΜΑΤΩΣΗ ΣΕ ΕΦΑΡΜΟΓΕΣ ΑΝΑΓΝΩΡΙΣΗΣ ΠΡΟΤΥΠΩΝ

Εκπόνηση:
Πέτρος Κατσιλέρος

Επίβλεψη:
Νικόλαος Πιτσιάνης

Με την εκπόνηση της εν λόγω διπλωματικής εργασίας ολοκληρώνεται ο κύκλος των προπτυχιακών μου σπουδών αποκτώντας δίπλωμα Ηλεκτρολόγου Μηχανικού και Μηχανικού Η/Υ απο το Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης.

# Περίληψη

Στα πλαίσια της εργασίας αυτής διερευνήθηκε η συμπεριφορά και η απόδοση του αλγορίθμου "Τοπική Γραμμική Ενσωμάτωση"[1] στο πεδίο της Αναγνώρισης Προτύπων. Ο αλγόριθμος ανήκει στην ευρύτερη κατηγορία "Αλγόριθμοι Μείωσης των Διαστάσεων" με τους οποίους μπορούμε να επιτύχουμε μείωση των παραμέτρων οι οποίες προσδιορίζουν κάποιο συγκεκριμένο πρόβλημα. Οι βασικοί μας στόχοι μέσω αυτής της διαδικασίας είναι αρχικά η συμπίεση της πληροφορίας, δηλαδή η δυνατότητα να εκφράσουμε την πληροφορία των αρχικών μας δεδομένων με ένα υποσύνολο της, με τις ελάχιστες δυνατές απώλειες. Επίσης μπορούμε να επιτύχουμε τεράστια μείωση της υπολογιστικής πολυπλοκότητας αλλά και της διαθέσιμης μνήμης που απαιτούνται για την προσπέλαση,αποθήκευση και μετέπειτα επεξεργασία των δεδομένων. Τέλος υπάρχουν περιπτώσεις στις οποίες θέλουμε να απομακρύνουμε απο τα δεδομένα μας, στοιχεία τα οποία αποτελούν θόρυβο και επιδρούν αρνητικά στην εξαγωγή ορθού συμπεράσματος ταξινόμησης.

Πιο συγκεκριμένα έγινε εφαρμογή της διαδικασίας μείωσης των διαστάσεων μέσω του αλγορίθμου "Τοπική Γραμμική Ενσωμάτωση"[1] σε τρία σετ δεδομένων. Τα δύο πρώτα περιέχουν εικόνες με ψηφία-αριθμούς και είναι τα MNIST[;] και Google Streen View House Numbers[2]. Το τρίτο είναι το Arcene[3] και περιέχει δεδομένα απο τον χώρο της Ιατρικής και συγκεκριμένα πρόκειται για δεδομένα απο ασθενείς με σκοπό την πρόβλεψη εμφάνισης κάποιας μορφής καρκίνου. Στα πρώτα δύο ο στόχος μας είναι να γίνει σωστή αναγνώριση κάθε ψηφίου.

Ο τελικός σκοπός είναι να προσδιορίσουμε με ακρίβεια κατά πόσο μπορούμε να επιτύχουμε συμπίεση της πληροφορίας και τι επιδράσεις θα έχει αυτό στην διαδικασία της ταξινόμησης των δεδομένων σε κλάσεις. Μέσα απο τα πειράματα λοιπόν έγινε προσπάθεια να διερευνηθεί τόσο η αποτελεσματικότητα του αλγορίθμου στα διαφορετικά σετ δεδομέναν αλλά και την επίδραση που έχουν οι παράμετροί του στην επίλυση κάθε προβλήματος χωριστά. Ως μετρική αξιολόγησης της αποτελεσματικότητας του αλγορίθμου χρησιμοποιήθηκε η σύγκριση μεταξύ του σφάλματος ταξινόμησης πρίν και μετά την διαδικασία μείωσης των διαστάσεων.

Πολύ σημαντικό εύρημα της εν λόγω δουλειάς πέραν των πολύ ικανοποιητικών αποτελεσμάτων μετά την μείωση των διαστάσεων είναι η παρουσίαση δύο νέων μεθόδων, οι οποίες αποτελούν παραλλαγές του αλγορίθμου "Τοπική Γραμμική Ενσωμάτωση"[1]. Με την πρώτη μέθοδο γίνεται εφικτή η χρήση του αλγορίθμου σε προβλήματα ταξινόμησης όπου τα αποτελέσματα θα πρέπει να δίνονται σε "πραγματικό χρόνο", μειώνοντας παράλληλα και την πολυπλοκότητα εκτέλεσης του αλγορίθμου. Με την δεύτερη μέθοδο μειώνεται δραματικά το πολύ μεγάλο υπολογιστικό κόστος

που απαιτεί ο αλγόριθμος κατά την εκτέλεσή του. Τέλος, πολύ σημαντικό στοιχείο αποτελεί το γεγονός ότι οι δύο αυτές μέθοδοι μπορούν να συνδιαστούν μεταξύ τους έχοντας έτσι πολλαπλή μείωση της πολυπλοκότητας και άμεση εξαγωγή των αποτελεσμάτων ταξινόμησης.

# Ευχαριστίες

Με την ολοκλήρωση αυτής της διπλωματικής εργασίας θα ήθελα καταρχήν να ευχαριστήσω τον επίκουρο καθηγητή του τμήματός Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης κ.Νικόλαο Πιτσιάνη ο οποίος μου έδωσε το ερέθισμα, απαραίτητες χρήσιμες συμβουλές αλλά και πόρους ώστε να μπορέσω να ολοκληρώσω την έρευνα για το συγκεκριμένο θέμα. Επίσης ένα μεγάλο ευχαριστώ στον υποψήφιο διδάκτορα του τμήματος Νίκο Σισμάνη για την καθοδήγηση του καθ΄όλη την διάρκεια εκπόνησης της εργασίας μου αυτής.

Τέλος, ένα πολύ θερμό και μεγάλο ευχαριστώ στους γονείς μου οι οποίοι με στήριξαν τόσο οικονομικά όσο και ψυχολογικά όλα αυτά τα χρόνια ώστε να μπορέσω να αποκτήσω το δίπλωμά μου. Στο σημείο αυτό δεν θα μπορούσα να παραλείψω τον σκύλο μου,την κοπέλα και τους φίλους μου διότι ο καθένας ξεχωριστά και με τον τρόπο του βοήθησαν στην αντιμετώπιση των δυσκολιών που συνάντησα καθόλη την διάρκεια των σπουδών μου.

Κατσιλέρος Πέτρος

Θεσσαλονίκη, Μάϊος 2016

# Αφιέρωση

Αφιερώνω την διπλωματική αυτή εργασία πρωτίστως στον εαυτό μου για τον κόπο μου όλα αυτά τα χρόνια ώστε να μπορέσω να αποκτήσω το δίπλωμα Ηλεκτρολόγου Μηχανικού και Μηχανικού Η/Υ και κατά δεύτερον στους γονείς μου οι οποίοι με στήριξαν ανελλιπώς και με κάθε τρόπο σε όλη αυτή την πορεία.

Κατσιλέρος Πέτρος

Θεσσαλονίκη, Μάϊος 2016

# Περιεχόμενα

# Κατάλογος Πινάκων

# Κατάλογος Σχημάτων

# Κεφάλαιο 1

# Εισαγωγή

Η τεχνητή νοημοσύνη μέσω της μηχανικής μάθησης είναι αναμφισβήτητα ένας επιστημονικός κλάδος ο οποίος επικεντρώνει το ενδιαφέρον ολοένα και περισσότερων μηχανικών-ερευνητών. Το γεγονός αυτό οφείλεται στην επιτυχία τέτοιου είδους εφαρμογών σε διάφορους κλάδους της καθημερινότητάς μας όπως αυτός της ρομποτικής, της υγείας, της εξόρυξης γνώσης κλπ. Επίσης οι σημερινοί υπολογιστές λόγω της ραγδαίας ανάπτυξης της τεχνολογίας παρέχουν τους απαραίτητους πόρους ώστε να μπορέσουν να αναπτυχθούν και να διερευνηθούν τέτοιου είδους προβλήματα. Παρ΄ όλα αυτά, όσους πόρους και αν διαθέσουμε δεν μπορούμε σε καμιά περίπτωση να δημιουργήσουμε κάτι αντίστοιχο με τον ανθρώπινο εγκέφαλο.

Γνωρίζουμε οτι ο ανθρώπινος εγκέφαλος είναι ένα τρομερά περίπλοκο σύστημα εκατομμυρίων νευρώνων συνδεδεμένων μεταξύ τους οι οποίοι είναι σε θέση να εκτελούν σε κλάσματα του δευτερολέπτου έναν τεράστιο αριθμό λογικών πράξεων. Το μοντέλο αυτό είναι αδύνατον να προσομοιωθεί με οποιοδήποτε υπολογιστικό σύστημα διαθέτει ο άνθρωπος σήμερα. Στην προσπάθεια των Μηχανικών να μοντελοποιήσουν τις λειτουργίες του λαμβάνοντας φυσικά υπόψιν ευρήματα και αποτελέσματα των επιστημόνων της Ιατρικής, σημαντικές λύσεις και βελτιστοποιήσεις έρχονται να δώσουν αλγόριθμοι οι οποίοι έχουν ως στόχο να μειώσουν τις παραμέτρους τις οποίες πρέπει να εκτιμηθούν για την επίλυση κάποιου προβλήματος.

## 1.1 Αναγνώριση προτύπων και μηχανική μάθηση

Αναγνώριση προτύπων καλείται η επιστημονική περιοχή που έχει στόχο την ταξινόμηση αντικειμένων σε κατηγορίες ή κλάσεις. Ανάλογα με την κάθε εφαρμογή τα δεδομένα μπορεί να είναι είτε εικόνες, είτε σήματα είτε οποιοδήποτε άλλο σετ δεδομένων χρειάζεται για κάποιο λόγο να ταξινομηθεί. Στις μέρες μας η ανάγκη διαχείρισης αλλά και ανάκτησης πληροφοριών μέσω ηλεκτρονικών υπολογιστών αποκτά τεράστια σπουδαιότητα. Αυτό διότι ο όγκος των πληροφοριών αυξάνεται ραγδαία με ρυθμό αδύνατο να τις διαχειριστεί ο άνθρωπος. Επίσης η ανάπτυξη της τεχνολογίας μας παρέχει πολύ ισχυρά υπολογιστικά συστήματα με τη χρήση των οποίων μπορούμε να δημιουργήσουμε πολύπλοκα μοντέλα εξόρυξης γνώσης.

Επιστημονικοί κλάδοι στους οποίους έχει τεράστια σημασία η αναγνώριση προτύπων είναι αυτοί της Ιατρικής, της Βιολογίας, ο χώρος των αγορών και των επιχειρήσεων και τέλος η διαχείριση και η εξόρυξη γνώσης απο τον τεράστιο όγκο της πληροφορίας που είναι διαθέσιμος στο διαδίκτυο. Φυσικά η αναγνώριση προτύπων είναι ένα πολύ σημαντικό μέρος του κλάδου της Μηχανικής μάθησης σε ρομποτικά/υπολογιστικά συστήματα.

Η υπολογιστική όραση για παράδειγμα είναι αντικείμενο ιδιαίτερα χρήσιμο τόσο στον χώρο της Ρομποτικής όσο σε αυτόν της Ιατρικής αλλά προφανώς και της Βιομηχανίας. Τέτοιου είδους εφαρμογές έχουν εισέλθει πολύ δυναμικά στην καθημερινότητά μας τα τελευταία χρόνια. Συγκεκριμένα στον χώρο της βιομηχανίας υπάρχουν συστήματα τα οποία επιβλέπουν μέσω μια κάμερας την γραμμή παραγωγής καθώς και ρομπότ τα οποία μεταφέρουν και συναρμολογούν αντικείμενα. Επίσης υπάρχουν εφαρμογές οι οποίες αναγνωρίζουν για παράδειγμα πρόσωπα τραβώντας μια εικόνα με το κινητό μας τηλέφωνο. Τέλος στον χώρο της αυτοκινητοβιομηχανίας δεν είναι λίγες αντίστοιχες εφαρμογές οι οποίες έχουν συμβάλει δυναμικά στην αυτόνομη οδήγηση αλλά και στην προειδοποίηση για εμπόδια κλπ.

Ιδιαίτερη έμφαση αξίζει να δοθεί στην εξόρυξη γνώσης σε κλάδους όπως στη Βιολογία αλλά και στην Ιατρική. Για παράδειγμα η πρόβλεψη εμφάνισης ασθενειών όπως ο καρκίνος μέσω αναγνώρισης συγκεκριμένων μοτίβων σε εικόνες απο μαγνητικό τομογράφο, η μελέτη της αλύσίδας του γεννετικού υλικού αλλά και εγχειρίσεις υψηλής ακρίβειας με τη χρήση ρομποτικού βραχίονα.

## 1.2 Ερεθίσματα απο τον τρόπο λειτουργίας του ανθρώπινου εγκεφάλου

Απο μελέτες που έχουν γίνει για την λειτουργία του ανθρώπινου εγκεφάλου γνωρίζουμε ότι για οποιοδήποτε σύνολο μετρήσεων προέρχεται για παράδειγμα είτε απο την όραση μας είτε απο την ακοή μας ο εγκέφαλός μας μετασχηματίζει το σύνολο των δεδομένων αυτών σε ένα νέο σύνολο χαρακτηριστικών. Με τον τρόπο αυτό, επιλέγοντας προφανώς κάθε φορά τα κατάλληλα χαρακτηριστικά, επιτυγχάνεται τεράστια συμπίεση του όγκου της πληροφορίας σε σύγκριση με τα αρχικά δεδομένα εισόδου. Αυτο σημαίνει λοιπόν ότι το μεγαλύτερο μέρος της πληροφορίας, για παράδειγμα μια σκηνής που βλέπουμε και στην οποία θέλουμε να αναγνωρίσουμε τα αντικείμενα που περιέχονται, συμπιέζεται σε έναν πολύ μικρό αριθμό χαρακτηριστικών. Η παραπάνω διαδικασία χαρακτηρίζεται ως τεχνική μείωσης διάστασης γνωστή στην βιβλιογραφία με τον όρο Dimensionality Reduction.

Ας πάρουμε για παράγειγμα τον κλάδο της υπολογιστικής όρασης ο οποίος αποτελεί και αντικείμενο μελέτης της εν λόγω εργασίας και ας αναρωτηθούμε το εξής: Πόσο δύσκολο είναι για κάποιον απο εμάς να αναγνωρίσει κάποιο νούμερο αποτυπωμένο σε μια εικόνα· Η προφανής απάντηση είναι καθόλου. Και αυτή είναι μια πολύ σωστή απάντηση, διότι για τον ανθρώπινο εγκέφαλο το να καταλάβει οτι το ψηφίο το οποίο βρίσκεται στην εικόνα είναι για παράδειγμα το 1 και όχι το 9 είναι ένα πολύ απλό πρόβλημα.

Πιο συγκεκριμένα βλέποντας μια οποιαδήποτε σκηνή ο ανθρώπινος εγκέφαλος προσπαθεί να εντοπίσει σημεία ενδιαφέροντος τα οποία αποτελούν χαρακτηριστικά σημεία της. Τέτοια μπορεί να είναι πολύ έντονες αλλαγές στην φωτεινότητα όπως για παράδειγμα γωνίες, κενά ή τρύπες. Στην συνέχεια εντοπίζει πιο σύνθετες γεωμετρίες όπως ευθείες ή καμπύλες γραμμές και τέλος προσδιορίζει πιο ολοκληρωμένες δομές τρισδιάστατων αντικειμένων. Το ίδιο ακριβώς γίνεται και στην παραπάνω περίπτωση με το ψηφίο. Εντοπίζουμε αρχικά οτι το μοτίβο του ψηφίου 1 είναι πολύ κοντά σε αυτά των ψηφίων εφτά και τέσσερα αλλά σε καμιά περίπτωση δεν θα λέγαμε οτι έχει τρομερές ομοιότητες με αυτό του δύο ή του οχτώ για παράδειγμα.

Το παραπάνω παράδειγμα είναι ένα πολύ απλό δείγμα του τρόπου με τον οποίο ο ανθρώπινος εγκέφαλος προσπαθεί με κάθε τρόπο να ελαχιστοποιήσει τις παραμέτρους που πρέπει να εκτιμήσει.

Φυσικά για ένα ρεαλιστικό περίπλοκο πρόβλημα της καθημερινότητάς μας θα δούμε οτι απαιτούνται πολύ πιο σύνθετοι υπολογισμοί και θα πρέπει να συνδιάσουμε ένα πλήθος απο παραμέτρους ώστε τελικά να καταλήξουμε στο τελικό συμπέρασμα για κάποια απόφαση. Σε κάθε περίπτωση όμως γίνεται τεράστια συμπίεση της αρχικής πληροφορίας μέσω τεχνικών μείωσης διαστάσεων ώστε να ελαχιστοποιηθούν οι παράμετροι που πρέπει να υπολογιστούν και να επιταχυνθεί η διαδικασία εξαγωγής της τελικής μας απόφασης.

Το γεγονός αυτό και δεδομένου οτι το όραμα της επιστημονικής κοινότητας των Μηχανικών που ασχολούνται με την Μηχανική μάθηση και την Εξόρυξη Γνώσης είναι να δημιουργηθεί ένα μοντέλο αντίστοιχο με αυτό του ανθρώπινου εγκεφάλου δεν θα μπορούσε να τους αφήσει αδιάφορους ώστε να μελετήσουν και να αναπτύξουν αντίστοιχους αλγορίθμους.

### 1.2.1 Μάθηση με επίβλεψη - χωρίς επίβλεψη - με ημιεπίβλεψη

Ένα πολύ εύλογο ερώτημα το οποίο προκύπτει απο την παραπάνω ανάλυση είναι, πως ο ανθρώπινος εγκέφαλος έχει μάθει και τελικώς έχει αποθηκεύσει το σύνολο αυτών των μοντέλων για το κάθε ψηφίο ή για οποιοδήποτε άλλο αντικείμενο ή μοτίβο μπορεί να αναγνωρίσει με τόσο μεγάλη ταχύτητα και ευκολία. Η απάντηση είναι προφανώς η συνεχής εκπαίδευση και η διαρκής υπενθύμιση των συγκεκριμένων προτύπων.

Πιο συγκεκριμένα ο άνθρωπος απο την μέρα που αρχίζει να αλληλεπιδρά με το περιβάλλον παίρνει διάφορα ερεθίσματα τα οποία καιρό με τον καιρό μαθαίνει να τα ταξινομεί κατάλληλα και να τα χρησιμοποιεί σε περίπτωση που εμφανιστούν μπροστά του. Τα ερεθίσματα αυτά είναι είτε εικόνες, είτε ήχοι είτε ερεθίσματα τα οποία μπορεί να προέρχονται απο τις υπόλοιπες αισθήσεις του.

Ο τρόπος με τον οποίο καταφέρνουμε να συγκρατούμε και να μπορούμε να διαχειριστούμε ανα πάσα στιγμή τον τεράστιο όγκο πληροφοριών που βρίσκονται καταχωρημένες στον εγκέφαλό μας είναι ένας συνδιασμός τεχνικών μάθησης και συνεχούς εκπαίδευσης. Οι τεχνικές αυτές στον χώρο της τεχνητής νοημοσύνης αναφέρονται ως τεχνικές μάθησης με επίβλεψη, χωρίς επίβλεψη και με ημιεπίβλεψη. Θα μπορούσε κάποιος αρχικά να υποστηρίξει ότι ο ανθρώπινος εκγέφαλος χρησιμοποιεί κατεξοχήν τεχνικές μάθησης χωρίς επίβλεψη διότι μπορεί να μαθαίνει μόνος του νέα πράγματα.

Είναι όμως πραγματικά αυτό το οποίο συμβαίνει· Η απάντηση είναι όχι, και αυτό διότι απο την πολύ νεαρή του ηλικία ο καθένας μας έχει γύρω του ανθρώπους οι οποίοι προσπαθούν συνεχώς να μας μεταφέρουν γνώση και να μας μάθουν τι βρίσκεται γύρω μας και πως να αλληλεπιδρούμε μαζί του. Παρ΄όλα αυτά μετά απο κάποιο σημείο ο ανθρώπινος εγκέφαλος αποκτά δυνατότητες με τις οποίες μπορεί να αξιολογεί και να μαθαίνει μόνος του πολύ σύνθετα προβλήματα. Αυτό το επιτυγχάνει αναλύοντάς τα σε απλούστερα τα οποία γνωρίζει ήδη πως να τα διαχειριστεί. Επίσης είναι στην φύση του ανθρώπου να εξερευνεί συνεχώς άγνωστα μονοπάτια και να αναζητεί απαντήσεις σε άγνωστα προβλήματα επιτυγχάνοντας αξιοθαύμαστα αποτελέσματα.

Απο τα παραπάνω καταλήγουμε στο συμπέρασμα ότι ο άνθρωπος χρησιμοποιεί τεχνικές ημιεπίβλεψης για την εκπαίδευση του εγκεφάλου του γεγονός το οποίο του δίνει την δυνατότητα να μπορεί να διαχειριστεί αλλά και να επεξεργασθεί πολύ σύνθετα μοντέλα. Μέσα απο αυτή την διαδικασία είναι σε θέση με το πέρασμα του χρόνου να δημιουργήσει ένα τεράστιο και πανίσχυρο δίκτυο πληροφοριών, ταξινομημένο με τρόπο τον οποίο δεν μπορούμε ακόμα να εξηγήσουμε και να κατανοήσουμε. Με αυτό το μοντέλο είναι σε θέση ταχύτατα να αποφασίζει που βρίσκεται ο ευρύτερος χώρος της πληροφορίας που θέλει να αντλήσει και στην συνέχεια να λαμβάνει με τεράστια ακρίβεια και ταχύτητα την τελική του απόφαση.

Το μοντέλο αυτό είναι αν μη τι άλλο αξιοθαύμαστο και μέχρι στιγμής ανεξήγητο. Παρ΄ όλα είναι πολύ δύσκολο να εφαρμοστεί στον τομέα της τεχνητής νοημοσύνης και αυτό διότι ακόμα δεν είμαστε σε θέση να δώσουμε εξηγήσεις για τον ακριβή τρόπο λειτουργίας του. Το συνηθέστερο και πιο αποτελεσματικό μέχρι στιγμής μοντέλο το οποίο χρησιμοποιείται στην εξόρυξη γνώσης μέσω ηλεκτρονικών υπολογιστών είναι αυτό της μάθησης με επίβλεψη. Σύμφωνα με το μοντέλο αυτό θα πρέπει αν συλλέξουμε ένα μεγάλο συνήθως όγκο δεδομένων, τον οποίο να τροφοδοτήσουμε στην συνέχεια ως είσοδο στο σύστημά μας και με την κατάλληλη μεθοδολογία να το καθοδηγήσουμε ώστε τελικά να μάθει συγκεκριμένα μοντέλα τα οποία να μπορεί να χρησιμοποιήσει στην συνέχεια με σκοπό την εξαγωγή κάποιου συμπεράσματος.

## 1.3 Μείωση της διάστασης των δεδομένων

Στην παραπάνω διαδικασία δεδομένου ότι στις περισσότερες περιπτώσεις έχουμε να αντιμετωπίσουμε πολύ σύνθετα υπολογιστικά προβλήματα ο αριθμός των παραμέτρων που πρέπει να υπολογιστούν είναι σε συγκεκριμένες εφαρμογές απαγορευτικά μεγάλος. Σε κάποιες εφαρμογές το πρόβλημα είναι θέμα χρόνου όπου πρέπει να γίνει μείωση των παραμέτρων ώστε να ελαχιστοποιηθεί ο χρόνος εξαγωγής του συμπεράσματος. Σε άλλες είναι θέμα χώρου διότι ένας μεγάλος αριθμός πολυδιάστατων δεδομένων μπορεί να αποτελεί πρόβλημα σε συγκεκριμένες εφαρμογές. Τέλος υπάρχουν περιπτώσεις στις οποίες χρειαζόμαστε την μείωση των διαστάσεων ώστε να διώξουμε εντελώς παραμέτρους οι οποίες επιδρούν σαν θόρυβος και επηρεάζουν αρνητικά την εξαγωγή ορθού συμπεράσματος ταξινόμησης. Προφανώς σε πολλές πρακτικές εφαρμογές επικρατεί ένας συνδυασμός των παραπάνω αναγκών.

Αντικείμενο λοιπόν της εν λόγω διπλωματικής εργασίας είναι η διερεύνηση και η χρήση του αλγορίθμου "Τοπική Γραμμική Ενσωμάτωση"[1] για την μείωση των διαστάσεων σε πρακτικά προβλήματα όπως η αναγνώριση ψηφίων αλλά και η ταξινόμηση ασθενών με βάση το αν πρόκειται να εμφανίσουν κάποιας μορφής καρκίνο ή όχι. Τα αποτελέσματα των πειραμάτων είναι ιδιαίτερα ενθαρυντικά και δείχνουν σε όλες τις περιπτώσεις ότι η μείωση των διαστάσεων επιδρά δραματικά στην μείωση του κόστους των υπολογισμών αλλά και στην αύξηση της σωστής πρόβλεψης λόγω απομάκρυνσης του θορύβου. Επίσης παρουσιάζονται δύο πρακτικές και ρεαλιστικές μέθοδοι εφαρμογής του αλγορίθμου σε πραγματικά προβλήματα απο τις οποίες η πρώτη παρέχει την δυνατότητα για την ταξινόμηση των αποτελεσμάτων και την εξαγωγή συμπεράσματος σε πραγματικό χρόνο και η δεύτερη έρχεται να αντιμετωπίσει το πρόβλημα της πολύ μεγάλης υπολογιστικής πολυπλοκότητας που απαιτεί η εκτέλεση του τελευταίο βήματος του αλγορίθμου.

# Κεφάλαιο 2

# Μαθηματικό και θεωρητικό υπόβαθρο

## 2.1 Διανύσματα βάσης

Έστω ότι έχουμε ένα σύνολο δειγμάτων εισόδου με αντίστοιχο διάνυσμα $\mathbf{x}$ διάστασης $N \times 1$,

$$\mathbf{x}^T = \Big[x(0), \ldots, x(N-1)\Big]$$

Έστω επίσης ορθοκανονικό μητρώο A, τάξης $N \times N$. Τότε ορίζουμε το μετασχηματισμένο διάνυσμα $\mathbf{y}$ του $\mathbf{x}$ ώς

$$\mathbf{y} = \mathbf{A}^H \mathbf{x} \equiv \begin{bmatrix} \mathbf{a}_0^H \\ \vdots \\ \mathbf{a}_{N-1}^H \end{bmatrix} \mathbf{x} \tag{2.1.1}$$

Το H δηλώνει τον Hermitian τελεστή, δηλαδή τον μιγαδικό συζηγή του ανάστροφου. Απο τον ορισμό των ορθοκανονικών μητρώων έχουμε

$$\mathbf{x} = \mathbf{A}\mathbf{y} = \sum_{i=0}^{N-1} y(i)\mathbf{a}_i \tag{2.1.2}$$

Οι στήλες του $\mathbf{A}$, $\mathbf{a}_i = 0, 1, \ldots, N-1$ καλούνται *διανύσματα βάσης* του μετασχηματισμού. Τα στοιχεία $y(i)$ του $\mathbf{y}$ είναι οι προβολές του διανύσματος $\mathbf{x}$ σε αυτά τα διανύσματα βάσης. Λαμβάνοντας υπόψιν την ιδιότητα της ορθοκανονικότητας μπορούμε να επαληθεύσουμε την παραπάνω διατύπωση υπολογίζοντας το εσωτερικό γινόμενο του $\mathbf{x}$ με το $\mathbf{a}_j$. Έχουμε:

$$\langle \mathbf{a}_j, \mathbf{x} \rangle \equiv \mathbf{a}_j^H \mathbf{x} = \sum_{i=0}^{N-1} y(i) \langle \mathbf{a}_j, \mathbf{a}_i \rangle = \sum_{i=0}^{N-1} y(i) \delta_{ij} = y(j) \tag{2.1.3}$$

### 2.1.1 Διάνυσμα εικόνας

Άν πάρουμε για παράδειγμα μια εικόνα, το σύνολο των δειγμάτων εισόδου είναι μια δυδιάστατη ακολουθία $X(i,j), i,j = 0, 1, \ldots, N-1$, η οποία ορίζει ένα μητρώο $Q$, τάξεως $N \times N$. Σε αυτή την περίπτωση μπορούμε να μετατρέψουμε την είσοδο αυτή σε ένα διάνυσμα $\mathbf{x}$ διάστασης $N^2$ διατάσσοντας για παράδειγμα τις γραμμές του μητρώου την μία μετά την άλλη έχοντας τελικά

$$\mathbf{x}^T = \Big[ X(0,0), \ldots, X(0,N-1), \ldots, X(N-1,0), \ldots, X(N-1,N-1) \Big] \tag{2.1.4}$$

Με αυτό τον μετασχηματισμό όμως ο αριθμός των πράξεων που απαιτούνται για τον πολλαπλασιασμό ενός τετραγωνικού μητρώου τάξεως $N \times N$ με ένα διάνυσμα $\mathbf{x}$ διαστάσεων $N^2 \times 1$, είναι της τάξης $\mathcal{O}(N^4)$ μέγεθος απαγορευτικό για τις περισσότερες ρεαλιστικές εφαρμογές.

### 2.1.2 Ορθοκανονικά ιδιοδιανύσματα

Το παραπάνω εμπόδιο μπορεί να ξεπεραστεί αν μετασχηματίσουμε το μητρώο $Q$ μέσω ενός συνόλου *μητρώων βάσης*. Έστω λοιπόν $U$ και $V$ ορθοκανονικά μητρώα διάστασης $N \times N$. Ορίζουμε τότε

το μετασχηματισμένο μητρώο $Y$ του $X$ ως

$$Y = U^H X V \tag{2.1.5}$$

ή

$$X = U Y V^H \tag{2.1.6}$$

Μέσω αυτού του μετασχηματισμού ο αριθμός των πράξεων μειώνεται σε $\mathcal{O}(N^3)$. Πιο αναλυτικά η παραπάνω εξίσωση θα μπορούσε να γραφεί ως

$$Q = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} Y(i,j) \mathbf{u}_i \mathbf{v}_j^H \tag{2.1.7}$$

όπου $\mathbf{u}_i$ είναι τα διανύσματα στήλης του U και $\mathbf{v}_j$ τα διανύσματα στήλης του V. Η παραπάνω εξίσωση είναι ένα ανάπτυγμα του μητρώου X ως προς τις $N \times 2$ εικόνες βάσης. Τέλος κάθε ένα απο τα γινόμενα $\mathbf{u}_i \mathbf{v}_j$ είναι ένα μητρώο $N \times N$

$$\mathbf{u}_i \mathbf{v}_j = \begin{bmatrix} u_{i0} v_{j0}^* & \dots & u_{i0} v_{jN-1}^* \\ \vdots & \vdots & \vdots \\ u_{iN-1} v_{j0}^* & \dots & u_{iN-1} v_{jN-1}^* \end{bmatrix} \tag{2.1.8}$$

Στην περίπτωση κατα την οποία το Y είναι διαγώνιο τότε έχουμε

$$Q = \sum_{i=0}^{N-1} Y(i,i)\mathbf{u}_i\mathbf{v}_i^H \tag{2.1.9}$$

με αποτέλεσμα το πλήθος των μητρώων-εικόνων βάσης να μειώνεται σε N. Τέλος έπειτα απο μερικές πράξεις και τροποποιήσεις μπορούμε να ορίσουμε κάθε στοιχείο $(i, j)$ του μετασχηματισμένου μητρώου ως τον πολλαπλασιασμό κάθε στοιχείου του X με τον συζυγή του αντίστοιχου στοιχείου του $A_{ij}$ και αθροίζοντας όλα τα γινόμενα. Δηλαδή

$$\langle A, B\rangle = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} A(m,n)^* B(m,n) \tag{2.1.10}$$

και τελικά

$$Y(i,j) = \langle A_{i,j}, X\rangle \tag{2.1.11}$$

## 2.2 Ο μετασχηματισμός Karhunen-Loeve - PCA

Ο μετασχηματισμός Karhunen-Loeve[4] αξιοποιεί την στατιστική πληροφορία που περιγράφει τα δεδομένα και ο υπολογισμός του μητρώου γίνεται χωρίς επίβλεψη. Ας υποθέσουμε και πάλι ένα διάνυσμα $\mathbf{x}$ το οποίο αποτελείται απο τα δείγματα μια εικόνας τα οποία έχουν διαταχθεί λεξικογραφικά όπως περιγράφτηκε παραπάνω. Πρέπει να επισημανθεί στο σημείο αυτό η επιθυμητή ιδιότητα των εξαχθέντων χαρακτηριστικών να είναι αμοιβαίως ασυσχέτιστα και αυτό για την αποφυγή πλεονάζουσας πληροφορίας. Η πιο συνηθισμένη συνθήκη για την γέννηση τέτοιου είδους χαρακτηριστικών είναι η μέση τιμή των δεδομένων να έχει μηδενική τιμή. Δηλαδή θέλουμε την ιδιότητα

$$E[y(i)y(j)] = 0, i \neq j \tag{2.2.1}$$

Έστω

$$\mathbf{y} = A^T\mathbf{x} \tag{2.2.2}$$

Εφόσον έχουμε υποθέσει ότι $E[x] = 0$ αμέσως βλέπουμε ότι $E[y] = 0$ και

$$R_y = E[\mathbf{y}\mathbf{y}^T] = E[A^T\mathbf{x}\mathbf{x}^T A] = A^T R_x A \tag{2.2.3}$$

Πρακτικά το $R_x$ αντιπροσωπεύει μια μέση τιμή πάνω στο δοθέν σύνολο διανυσμάτων εκπαίδευσης. Επίσης είναι συμμετρικό μητρώο και επομένως τα ιδιοδιανύσματά του είναι αμοιβαίως ορθογώνια. Άρα έστω ότι επιλέγεται ένα μητρώο Α με στήλες τα ορθοκανονικά ιδιοδιανύσματα $\mathbf{a}_i, i = 0, 1, \ldots, N-1$ του $R_x$ τότε το $R_y$ είναι διαγώνιο.

$$R_y = A^T R_x A = \Lambda \tag{2.2.4}$$

Το $\Lambda$ είναι διαγώνιο μητρώο με διαγώνια στοιχεία τις αντίστοιχες ιδιοτιμές $\lambda_i, i = 0, 1, \ldots, N-1$ του $R_x$. Αποτέλεσμα της παραπάνω διαδικασίας είναι ένας μετασχηματισμός, ο μετασχηματισμός Karhunen-Loeve, ο οποίος επιτυγχάνει τον αρχικό μας στόχο, δηλαδή την δημιουργία χαρακτηριστικών τα οποία είναι στατιστικώς ανεξάρτητα.

### 2.2.1 Προσέγγιση μέσου τετραγωνικού σφάλματος - MSE

Σε αυτή την υποενότητα θα αναλυθεί η διαδικασία με την οποία μπορούμε να οδηγηθούμε στην επιλογή κάποιων, έστω m το πλήθος, κυρίαρχων χαρακτηριστικών μέσω της προσέγγισης μέσου τετραγνικού σφάλματος. Ας πάρουμε ξανά τις εξισώσεις (2.1.1) και (2.1.2) τότε έχουμε

$$\mathbf{x} = \sum_{i=0}^{N-1} y(i)\mathbf{a}_i \quad \text{και} \quad y(i) = \mathbf{a}_i^T \mathbf{x} \tag{2.2.5}$$

Ορίζουμε λοιπόν τώρα ένα νέο διάνυσμα στον m-διάστατο υποχώρο

$$\widehat{\mathbf{x}} = \sum_{i=0}^{m-1} y(i)\mathbf{a}_i \tag{2.2.6}$$

στο οποίο προφανώς εμπλέκονται μόνο m απο τα διανύσματα βάσης. Με τον παραπάνω τρόπο δηλαδή ορίζεται η προβολή του $\mathbf{x}$ στον υποχώρο που ορίζουν τα ορθοκανονικά διανύσματα m τα οποία εμπλέκονται στην παραπάνω άθροιση.

Σκοπός μας λοιπόν στο σημείο αυτό είναι να προσεγγίσουμε με όσο το δυνατόν μικρότερο σφάλμα το διάνυσμα $\mathbf{x}$. Η προσέγγισή μας είναι το διάνυσμα $\widehat{\mathbf{x}}$ και θα προκύψει χρησιμοποιώντας την εξίσωση ελαχιστοποίησης μέσου τετραγωνικού σφάλματος. Έχουμε λοιπόν την εξίσωση

$$E[\|\mathbf{x} - \widehat{\mathbf{x}}\|^2] = E\left[\| \textstyle\sum_{i=m}^{N-1} y(i)\mathbf{a}_i\|^2\right] \tag{2.2.7}$$

Απο την παραπάνω εξίσωση στόχος μας τώρα είναι να επιλέξουμε τα ιδιοδιανύσματα τα οποία οδηγούν στο ελάχιστο μέσο τετραγωνικό σφάλμα. Λαμβάνοντας υπόψιν την ορθοκανονικότητα των ιδιοδιανυσμάτων και την παραπάνω εξίσωση καταλήγουμε ότι

$$E\left[\| \textstyle\sum_{i=m}^{N-1} y(i)\mathbf{a}_i\|^2\right] = E\left[\textstyle\sum_i \sum_j (y(i)\mathbf{a}_i^T)(y(j)\mathbf{a}_j)\right] = \tag{2.2.8}$$

$$= \sum_{i=m}^{N-1} E[y^2(i)] = \sum_{i=m}^{N-1} \mathbf{a}_i^T E[\mathbf{x}\mathbf{x}^T]\mathbf{a}_i \tag{2.2.9}$$

και λαμβάνοντας υπόψιν τον ορισμό των ιδιοδιανυσμάτων προκύπτει τελικά ότι

$$E[\|\mathbf{x}-\widehat{\mathbf{x}}\|^2] = \sum_{i=m}^{N-1} \mathbf{a}_i^T \lambda_i \mathbf{a}_i = \sum_{i=m}^{N-1} \lambda_i \tag{2.2.10}$$

Αν επομένως στην παραπάνω εξίσωση επιλέξουμε τα ιδιοδιανύσματα που αντιστοιχούν στις m ιδιοτιμές του μητρώου συσχέτισης τότε το σφάλμα της εξίσωσης ελαχιστοποιείται και μάλιστα ισούται με το άθροισμα των N-m μικρότερων ιδιοτιμών. Επιπλέον έχει αποδειχθεί ότι αυτό είναι το ελάχιστο μέσο τετραγωνικό σφάλμα σε σύγκριση με οποιαδήποτε άλλη προσέγγιση του $\mathbf{x}$ απο ένα m-διάστατο διάνυσμα. Για τον λόγο αυτό ο μετασχηματισμός Karhunen-Loeve είναι επίσης γνωστός ως *Ανάλυση κυρίων συνιστωσών* (Principal component analysis-PCA).

### 2.2.2 Συνολική Διασπορά

Έστω $\mathbf{y}$ το μετασχηματισμένο κατα **KL** διάνυσμα του $\mathbf{x}$ και $E[x] = 0$. Τότε απο τον αντίστοιχο ορισμό της διασποράς έχουμε ότι $\sigma^2_{y(i)} \equiv E[y^2(i)] = \lambda_i$. Δηλαδή έχουμε ότι οι διασπορές του μητρώου συσχέτισης εισόδου είναι ίσες με τις διασπορές των μετασχηματισμένων χαρακτηριστικών. Επομένως επιλεγοντας εκείνα τα χαρακτηριστικά $y(i) = \mathbf{a}_i^T\mathbf{x}$ που αντιστοιχούν στις m μεγαλύτερες ιδιοτιμές οδηγούμαστε σε μεγιστοποίηση της αθροιστικής διασποράς $\sum_i \lambda_i$. Συμπεραίνουμε λοιπόν ότι με αυτή την μεθολογία που ακολουθήσαμε, τα m χαρακτηριστικά που έχουν επιλεχθεί διατηρούν το μεγαλύτερο μέρος απο την συνολική διασπορά που σχετίζεται με τις αρχικές τυχαίες ματαβλητές $x(i)$.

### 2.2.3 Μείωση της διάστασης μέσω PCA

Απο την παραπάνω ανάλυση είναι φανερό ότι η μέθοδος PCA[4] επιτυγχάνει τον *γραμμικό μετασχηματισμό* ενός χώρου υψηλής διάστασης σε έναν χαμηλής διάστασης του οποίου μάλιστα τα στοιχεία είναι στατιστικώς ασυσχέτιστα. Έχοντας υποθέσει ότι $E[x] = 0$ και επίσης ότι οι N-m μικρότερες ιδιοτιμές του μητρώου συσχέτισης είναι μηδέν τότε απο την εξίσωση (2.2.10) συνεπάγεται ότι $\mathbf{x} = \widehat{\mathbf{x}}$. Δηλαδή έχουμε ότι το διάνυσμα $\mathbf{x}$ του αρχικού χώρου διάστασης N βρίσκεται σε έναν m-διάστατο υποχώρο του αρχικού και μάλιστα μπορούμε να το προσδιορίσουμε μέσω του

διανύσματος $\widehat{\mathbf{x}}$ με πολύ καλή προσέγγιση. Το γεγονός αυτό εισάγει την έννοια της *εγγενούς διάστασης* (intrinsic dimensionality). Τέλος στην περίπτωση της εγγενούς διάστασης μπορούμε να πούμε ό,τι το X μπορεί να περιγραφεί απο m ελεύθερες παραμέτρους.

## 2.3 Μετρική πολυδιάστατης κλιμάκωσης (Metric multidimensional scaling - MDS)

Ένας ακόμα πολύ διαδεδομένος αλγόριθμος μείωσης διάστασης είναι ο αλγόριθμος *Μετρική πολυδιάστατης κλιμάκωσης* (Metric multidimensional scaling - MDS)[5]. Ο αλγόριθμος αυτός δοθέντος ενός συνόλου $Q \subset \Re^N$, έχει ως στόχο να γίνει προβολή σε χώρο χαμηλότερης διάστασης, $Y \subset \Re^m$, έτσι ώστε τα εσωτερικά γινόμενα να διατηρηθούν κατά βέλτιστο τρόπο. Πρέπει δηλαδή να γίνει η ελαχιστοποίηση της εξίσωσης

$$E = \sum_i \sum_j (\mathbf{x}_i^T \mathbf{x}_j - \mathbf{y}_i^T \mathbf{y}_j)^2 \tag{2.3.1}$$

όπου $\mathbf{y}_i$ είναι η εικόνα του $\mathbf{x}_i$ και το άθροισμα υπολογίζεται ως προς όλα τα σημεία εκπαίδευσης του $X$. Το πρόβλημα δηλαδή, και σε αυτή την περίπτωση είναι όμοιο με αυτό της μεθόδου PCA, και μπορεί να αποδειχθεί ότι η λύση δίνεται απο την ανάλυση σε ιδιοτιμές-ιδιοδιανύσματα του μητρώου Gram, τα στοιχεία του οποίου ορίζονται ως

$$K(i,j) = \mathbf{x}_i^T \mathbf{x}_j \tag{2.3.2}$$

Ένας εναλλακτικός τρόπος επίλυσης του προβλήματος είναι η απαίτηση να διατηρηθούν, κατά βέλτιστο τρόπο, οι Ευκλείδιες αποστάσεις αντί των εσωτερικών γινομένων. Μπορούμε έτσι, να δημιουργήσουμε ένα μητρώο Gram συμβατό με τις τετραγωνικές Ευκλείδιες αποστάσεις, το οποίο μας οδηγεί στην ίδια λύση όπως και στην προηγούμενη περίπτωση. Προκύπτει μάλιστα, ότι οι

λύσεις που προκύπτουν απο τις μεθόδους PCA[4] και MDS[5] είναι ισοδύναμες.

Μια σύντομη απόδειξη της παραπάνω διατύπωσης είναι η εξής. Η μέθοδος PCA εκτελεί την ανάλυση ιδιοτιμών του μητρώου συσχέτισης $R_x$, το οποίο προσεγγίζεται απο τη σχέση

$$R_x = E[\mathbf{x}\mathbf{x}^T] \approx \frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_k \mathbf{x}_k^T = \frac{1}{n} X^T X \tag{2.3.3}$$

όπου

$$X^T = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n,] \tag{2.3.4}$$

Απο την άλλη το μητρώο Gram μπορεί επίσης να γραφεί ως

$$K = XX^T \tag{2.3.5}$$

Τέλος αποδεικνύεται ότι τα δύο μητρώα $X^T X$ καί $XX^T$ είναι ίδου βαθμού και έχουν τις ίδιες ιδιοτιμές με ιδιοδιανύσματα τα οποία ναι μεν είναι διαφορετικά μεταξύ τους αλλά παρόλα αυτά σχετίζονται.

## 2.4 Ανάλυση στην βάση των ιδιαζουσών τιμών (Singular Value Decomposition - SVD)

Η ανάλυση ενός μητρώου με βάση τις ιδιάζουσες τιμές είναι μια απο τις πιο κομψές και ισχυρές μεθόδους γραμμικής άλγεβρας η οποία έχει χρησιμοποιηθεί εκτενώς για την μείωση του βαθμού και της διάστασης σε προβλήματα αναγνώρισης προτύπων και σε εφαρμογές ανάκτησης πληροφορίας.

Δοθέντως ενός μητρώου X, τάξης $l \times n$, βαθμού r με $r \leq \min\{l,n\}$ υπάρχουν ορθοκανονικά μητρώα U και V, τάξης $l \times l$ και $n \times n$ αντίστοιχα ώστε

$$X = U \begin{bmatrix} \Lambda^{\frac{1}{2}} & \mathcal{O} \\ \mathcal{O} & 0 \end{bmatrix} V^H \quad \acute{\eta} \quad Y = \begin{bmatrix} \Lambda^{\frac{1}{2}} & \mathcal{O} \\ \mathcal{O} & 0 \end{bmatrix} = U^H X V \tag{2.4.1}$$

όπου $\Lambda^{\frac{1}{2}}$ είναι το $r \times r$ διαγώνιο μητρώο με στοιχεία $\sqrt{\lambda_i}$ με $\lambda_i$ οι μη μηδενικές ιδιοτιμές που σχετίζονται με το μητρώο $X^H X$. Με $\mathcal{O}$ συμβολίζουμε το μητρώο μηδενικών τιμών. Απο τα παραπάνω γίνεται φανερό ότι υπάρχουν μητρώα U και V που μετασχηματίζουν το X στην διαγώνια δομή του Y. Αν $\mathbf{u}_i$,$\mathbf{v}_i$ είναι τα διανύσματα στήλης των μητρώων U και V αντίστοιχα τότε η παραπάνω εξίσωση μπορεί να γραφεί στην μορφή

$$X = [u_0, u_1, \ldots, u_{r-1},] \begin{bmatrix} \sqrt{\lambda_0} & & & \\ & \sqrt{\lambda_1} & & \\ & & \ddots & \\ & & & \sqrt{\lambda_{r-1}} \end{bmatrix} \begin{bmatrix} \mathbf{v}_0^H \\ \mathbf{v}_1^H \\ \vdots \\ \mathbf{v}_{r-1}^H \end{bmatrix} \tag{2.4.2}$$

ή

$$X = \sum_{i=0}^{r-1} \sqrt{\lambda_i} \mathbf{u}_i \mathbf{v}_i^H = U_r \Lambda^{\frac{1}{2}} V_r^H \tag{2.4.3}$$

όπου $U_r$ δηλώνει το $l \times r$ μητρώο που αποτελείται απο τις r πρώτες στήλες του U και $V_r$ το $r \times n$ μητρώο που σχηματίζεται χρησιμοποιώντας τις πρώτες r στήλες του V. Επίσης $\mathbf{u}_i$,$\mathbf{v}_i$ είναι τα ιδιοδιανύσματα που αντιστοιχούν στις μη μηδενικές ιδιοτιμές των μητρώων $XX^H$ και $X^H X$ αντίστοιχα. Οι ιδιοτιμές $\lambda_i$ είναι γνωστές ως *ιδιάζουσες τιμές* (singular values) του $X$ και το ανάπτυγμα της παραπάνω εξίσωσης ως *ανάλυση με βάση τις ιδιάζουσες τιμές* (singular value decomposition - SVD)[6][7] του $X$.

### 2.4.1 Μείωση της διάστασης μέσω SVD

Η μέθοδος SVD[6][7] έχει χρησιμοποιηθεί εκτενώς για την μείωση της διάστασης του χώρου χαρακτηριστικών σε ένα μεγάλο εύρος εφαρμογών αναγνώρισης προτύπων. Έστω οτι έχουμε την

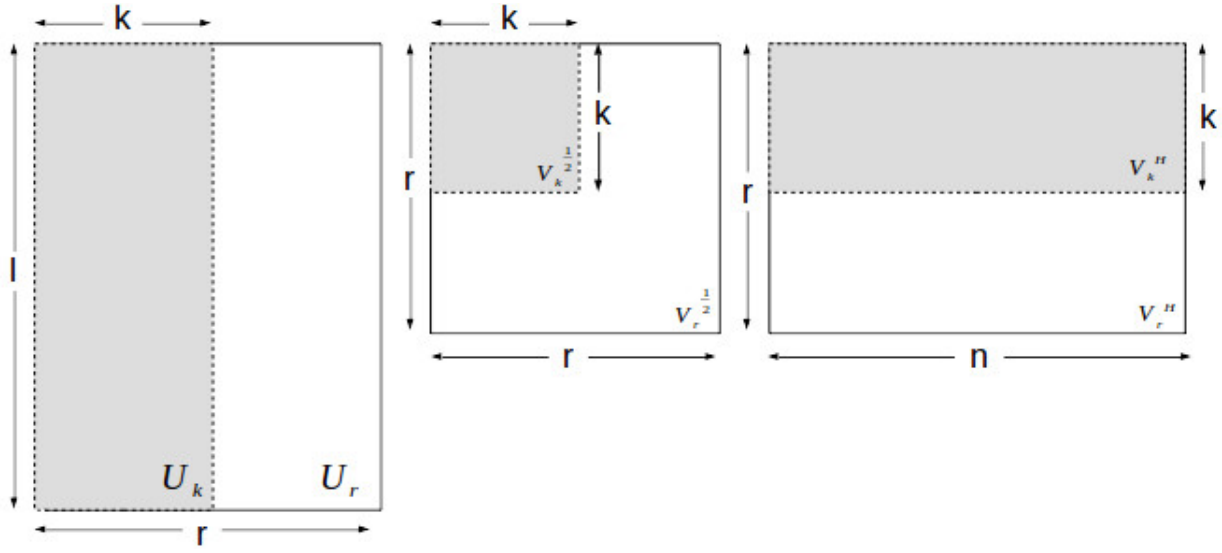
προσέγγιση χαμηλού βαθμού (low rank approximation) $\hat{X}$ του X. Αποδεικνύεται μέσω ελαχιστοποίησης του μέσου τετραγωνικού σφάλματος ότι αν η παραπάνω προσέγγιση σχηματίζεται απο την άθροιση των k μεγαλύτερων ιδιοτιμών τότε το μέσο τετραγωνικό σφάλμα της προσέγγισης είναι το ελάχιστο. Μπορούμε να καταλήξουμε στο συμπέρασμα ότι η μέθοδος SVD[6][7] οδηγεί στο ελάχιστο τετραγωνικό σφάλμα και επομένως το $\hat{X}$ είναι η καλύτερη *προσέγγιση βαθμού* $k$ του X. Η προσέγγιση αυτή δίνεται απο τον τύπο

$$X \simeq \hat{X} = \sum_{i=0}^{k-1} \sqrt{\lambda_i} \mathbf{u}_i \mathbf{v}_i^H, \quad k \leq r$$

$$= [\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{k-1}] \begin{bmatrix} \sqrt{\lambda_0}\mathbf{v}_0^H \\ \sqrt{\lambda_1}\mathbf{v}_1^H \\ \vdots \\ \sqrt{\lambda_{k-1}}\mathbf{v}_{k-1}^H \end{bmatrix} = U_k[\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{n-1},] \tag{2.4.4}$$

όπου ο μητρώο $U_k$ αποτελείται απο τις k πρώτες στήλες του $U$ και τα k-διάστατα διανύσματα $\mathbf{a}_i, i = 0, 1, \ldots, n-1$ είναι τα διανύσματα στήλες της $k \times n$ μήτρας του γινομένου $\Lambda^{\frac{1}{2}} V_k^H$. Το μητρώο $V_k^H$ αποτελείται απο τις $k$ πρώτες γραμμές του $V^H$ και $\Lambda^{\frac{1}{2}}$ είναι διαγώνιο μητρώο με στοιχεία τις τατραγωνικές ρίζες των αντίστοιχων $k$ ιδιαζουσών τιμών.

Στο παρακάτω σχήμα παρουσιάζεται γραφικά ώστε να γίνει καλύτερα κατανοητή η παραπάνω διαδικασία.




Σχήμα 2.1: Μείωση της διάστασης με SVD

Απο την παραπάνω ανάλυση καταλήγουμε στο συμπέρασμα ότι το l-διάστατο διάνυσμα $\mathbf{x}_i$ προσεγγίζεται απο το k-διάστατο διάνυσμα $\mathbf{a}_i$ που βρίσκεται στον υποχώρο που ορίζουν τα $\mathbf{u}_i, i = 0, 1, \ldots, k-1$ (το $\mathbf{a}_i$ είναι στην ουσία η προβολή του $\mathbf{x}_i$ στον υποχώρο αυτόν). Επίσης, λόγω της ορθοκανονικότητας των στηλών $\mathbf{u}_i, i = 0, 1, \ldots, k-1$ του $U_k$ βλέπουμε ότι

$$\|\mathbf{x}_i - \mathbf{x}_j\| \simeq \|U_k(\mathbf{a}_i - \mathbf{a}_j)\| = \|\sum_{m=0}^{k-1} \mathbf{u}_m(a_i(m) - a_j(m))\| = \|\mathbf{a}_i - \mathbf{a}_j\|, \quad i, j = 0, 1, \ldots, n-1 \quad (2.4.5)$$

Αντιλαμβανόμαστε λοιπόν ότι χρησιμοποιώντας την προηγούμενη προβολή και υποθέτωντας ότι η προσέγγιση είναι ικανοποιητική, η Ευκλείδια απόσταση μεταξύ $\mathbf{x}_i$ και $\mathbf{x}_j$ στον υψηλής διάστασης $l$-διάστατο χώρο διατηρείται (κατά προσέγγιση) κατά την προβολή στον χαμηλότερης διάστασης $k$-διάστατο χώρο.

## 2.5 Πρακτική εφαρμογή

Στο σημείο αυτό αξίζει να αναφερθεί ένα απλό παράδειγμα μέσω του οποίου μπορεί να γίνει αντιληπτή η πρακτική εφαρμογή των παραπάνω. Ας θεωρήσουμε λοιπόν ένα σύνολο $n$ προτύπων, όπου το καθένα αναπαρίσταται απο ένα $l$-διάστατο διάνυσμα χαρακτηριστικών. Τότε, δοθέντως ενός άγνωστου προτύπου στόχος μας είναι να αναζητήσουμε στο σύνολο των γνωστών προτύπων που έχουμε ώστε να βρούμε αυτό το οποίο παρουσιάζει την μεγαλύτερη ομοιότητα με το άγνωστο για το οποίο θέλουμε να καταλήξουμε σε κάποιο συγκεκριμένο συμπέρασμα. Η διαδικασία αυτή είναι εφικτή υπολογίζοντας την Ευκλείδια απόσταση μεταξύ του άγνωστου προτύπου με όλα τα γνωστά και επιλέγοντας τελικά το ζευγάρι με την μικρότερη απόσταση, δηλαδή αυτό με την μεγαλύτερη ομοιότητα.

Σε περιπτώσεις όπου τόσο ο αριθμός των διαστάσεων όσο και ο αριθμός των δειγμάτων είναι μεγάλος τότε η παραπάνω διαδικασία μπορεί να είναι ιδιαίτερα χρονοβόρα. Προκειμένου λοιπόν να απλοποιήσουμε τους υπολογισμούς μπορούμε να ακολουθήσουμε την παραπάνω διαδικασία που αναλύσαμε ώστε να μειώσουμε τις διαστάσεις του προβλήματός μας. Η διαδικασία έχει ως εξής: Αρχικά σχηματίζουμε το μητρώο δεδομένων $X$, διάστασης $l \times n$ με στήλες τα $n$ διανύσματα χαρακτηριστικών. Εκτελούμε την μεθοδολογία SVD[6][7] στο $X$ και αναπαριστούμε κάθε διάνυσμα χαρακτηριστικών $\mathbf{x}_i$ με την χαμηλότερης διάστασης προβολή του, $\mathbf{a}_i$. Το άγνωστο διάνυσμα προβάλλεται στον υποχώρο που ορίζουν οι στήλες του $U_k$ και εκτελούνται οι υπολογισμοί των Ευκλείδιων αποστάσεων στον $k$-διάστατο χώρο. Επειδή οι Ευκλείδιες αποστάσεις διατηρούνται κατά προσέγγιση, είναι εφικτό να αποφασίσουμε τους κοντινότερους γείτονες των διανυσμάτων εργαζόμενοι στον χώρο χαμηλότερης διάστασης. Σε περιπτώσεις για τις οποίες έχουμε $k \ll l$ επιτυγχάνεται σημαντική εξοικονόμηση στους υπολογισμούς.

Τέλος, αξίζει να αναφερθεί ότι η μεθοδολογία SVD[6][7] είναι πολύ αποτελεσματική τεχνική μείωσης της διάστασης σε περιπτώσεις όπου τα δεδομένα μπορούν να περιγραφούν επαρκώς μέσω του μητρώου συνδιασποράς, για παράδείγμα περιπτώσεις όταν ακολουθούν κατανομές παρόμοιες με την Gaussian κατανομή.

# Κεφάλαιο 3

# Αλγόριθμοι μείωσης διαστάσεων

## 3.1 Γραμμική μείωση διαστάσεων

Όλες οι τεχνικές μείωσης διαστάσεων στις οποίες έχουμε αναφερθεί μέχρι στιγμής είναι κατεξοχήν τεχνικές μείωσης της διάστασης του χώρου των χαρακτηριστικών. Μάλιστα το ιδιαίτερο χαρακτηριστικό τους είναι ότι αποτελούν μεθόδους οι οποίες σέβονται την γραμμικότητα. Η μέθοδος PCA[4] για παράδειγμα η οποία αποτελεί μια απο τις γνωστότερες αλλά και πιο ισχυρές μεθόδους γραμμικής μείωσης διαστάσεων λειτουργεί καλά αν τα σημεία των δεδομένων είναι κατανεμημένα σε ένα υπερεπίπεδο. Όπως αναλύθηκε στην ενότητα (2.2) η μέθοδος PCA[4] προβάλλει στις διευθύνσεις μέγιστης διασποράς. Επίσης όπως εξηγήσαμε στο προηγούμενο κεφάλαιο η ανάλυση ιδιοτιμών-ιδιοδιανυσμάτων του μητρώου συσχέτισης αποκαλύπτει την διάσταση του υπερεπιπέδου στο οποίο τα δεδομένα είναι διεσπαρμένα.

Με άλλα λόγια δηλαδή η διάσταση είναι ένα μέτρο του πλήθους των ελεύθερων μεταβλητών που είναι υπεύθυνες για τον τρόπο με τον οποίο μεταβάλλεται ένα σήμα, δηλαδή για την πραγματική πληροφορία την οποία κωδικοποιούν τα δεδομένα.

Παρότι ο αλγόριθμος PCA[4] αποτελεί μία πολύ ισχυρή και ευρέως χρησιμοποιούμενη μέθοδο μείωσης της διάστασης υπάρχουν περιπτώσεις στις οποίες η μέθοδος αποτυγχάνει. Τέτοιες είναι περιπτώσεις κατα τις οποίες ο μηχανισμός παραγωγής των δεδομένων είναι έντονα μη γραμμικός με αποτέλεσμα τα δεδομένα να κείτονται σε πιο περίπλοκες πολλαπλότητες. Ας πάρουμε για πα-

ράδειγμα τις εξισώσεις

$$x_1 = rcos\theta, \quad x_2 = rsin\theta$$

Προφανώς απο τις παραπάνω εξισώσεις είναι φανερό ότι το $x$ βρίσκεται στην περιφέρεια κύκλου ακτίνας $r$. Πρόκειται δηλαδή για πρόβλημα μονοδιάστατης πολλαπλότητας αφού αρκεί μια μόνο μεταβλητή για την περιγραφή των δεδομένων. Η παράμετρος αυτή είναι η απόσταση κατα μήκος της περιφέρειας απο ένα σημείο(αφετηρία) πάνω στην περίμετρο του κύκλου. Αν λοιπόν εφαρμόσουμε την μέθοδο PCA[4] στο παραπάνω σύνολο δεδομένων τότε η απάντηση που θα μας δώσει για την διάσταση των δεδομένων θα είναι, λανθασμένα προφανώς, ίση με δύο.

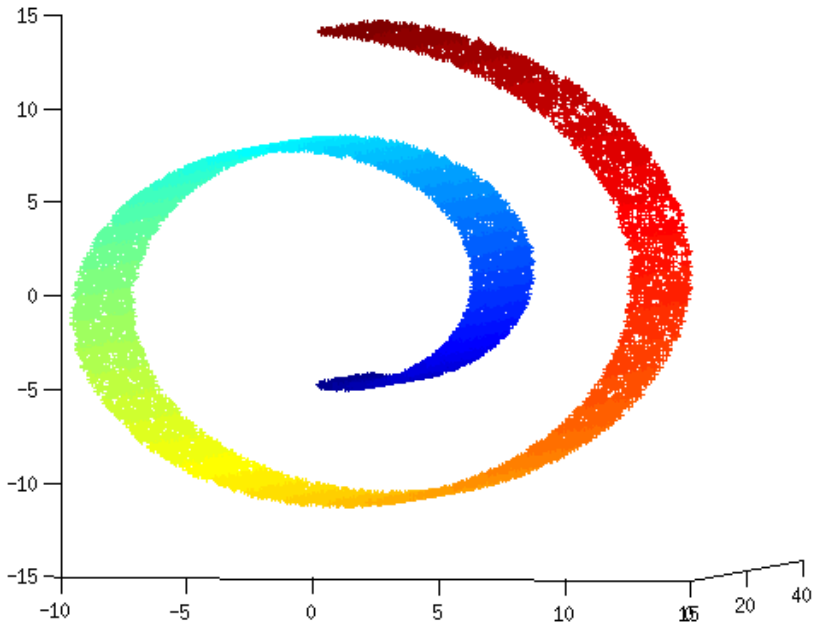
Περιπτώσεις όπως οι παραπάνω απαιτούν αλγορίθμους μείωσης διάστασης και εξαγωγής χαρακτηριστικών οι οποίοι να λαμβάνουν υπόψιν την γεωμετρία του προβλήματος ώστε να μπορούν να εξάγουν ασφαλή συμπεράσματα για την διάσταση των δεδομένων. Στον τομέα της υπολογιστικής όρασης για παράδειγμα, ο οποίος όπως αναφέραμε και παραπάνω αποτελεί βασικό κομμάτι της εν λόγω διατριβής, απαιτούνται κατεξοχήν αλγόριθμοι μη γραμμικής μείωσης διαστάσεων αφού οι εικόνες ή τα χαρακτηριστκά των εικόνων τα οποία αποτελούν τα δεδομένα μας είναι κατα κύριο λόγο μη γραμμικά.

## 3.2 Μη γραμμική μείωση διαστάσεων

Υπάρχει λοιπόν μια ευρεία γκάμα εφαρμογών οι οποίες απαιτούν αλγορίθμους μη γραμμικής μείωσης διαστάσεων. Αυτό συμβαίνει διότι στις συγκεκριμένες εφαρμογές η γεωμετρική αναπαράσταση των δεδομένων είναι τέτοια ώστε απαιτείται να βρεθεί μια ενσωμάτωση μικρότερης διάστασης η οποία βρίσκεται "κρυμμένη" στον χώρο των αρχικών διαστάσεων. Θα πρέπει μάλιστα κατά την διαδικασία αυτή να ληφθούν υπόψιν τα γεωμετρικά χαρακτηριστικά του χώρου των δεδομένων.

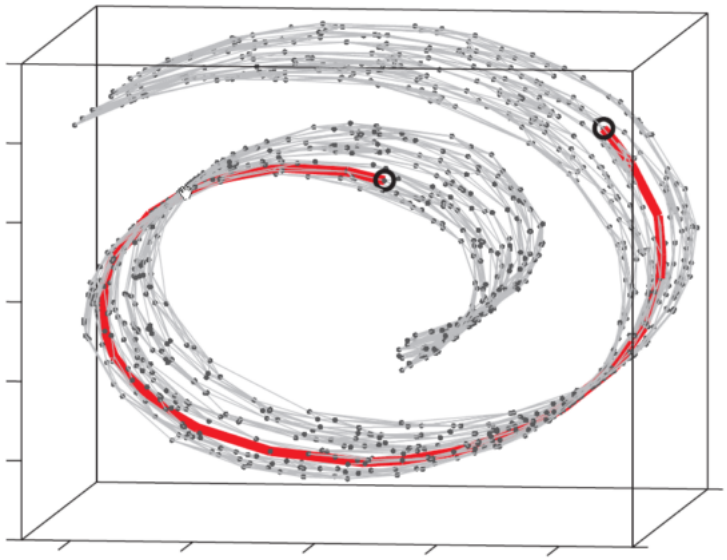
Έχει πολύ μεγάλη σημασία στο σημείο αυτό να κατανοήσουμε τι εννοούμε όταν αναφερόμαστε στα γεωμετρικά χαρακτηριστικά του προβλήματος. Το πιο χαρακτηριστικό και ευρέως χρησιμοποιούμενο παράδειγμα για τον σκοπό αυτό είναι ένα τεχνητό σετ δεδομένων, με την όνομασία Swiss

Roll[8] το οποίο φαίνεται στην παρακάτω εικόνα.



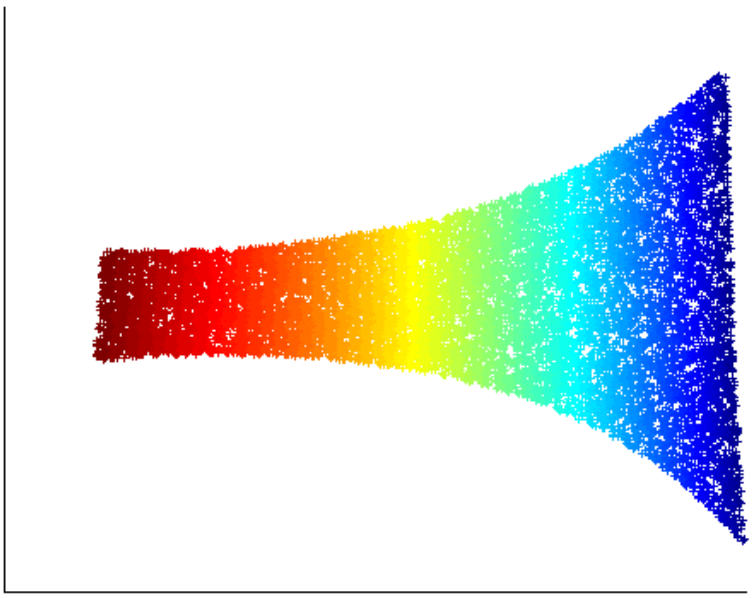

Σχήμα 3.1: Τρισδιάστατη αναπαράσταση του συνθετικού σετ δεδομένων - Swiss Roll.

Αυτό που αξίζει να παρατηρηθεί λοιπόν στο παραπάνω σετ δεδομένων είναι ότι αν για παράδειγμα διαλέξουμε κάποιο οποιοδήποτε σημείο του απο την κόκκινη περιοχή και προσπαθήσουμε να βρούμε ποιά δεδομένα αποτελούν κοντινότερους γείτονες του σημείου αυτού πιθανότατα θα πέφταμε στην παγίδα, όπως και οι τεχνικές γραμμικής μείωσης διαστάσεων, να πούμε ότι κάποια σημεία απο την μπλέ περιοχή βρίσκονται και αυτά στην γειτονιά του σημείου που διαλέξαμε. Αυτό προφανώς είναι λάθος αφού απο τον χρωματισμό των παραπάνω δεδομένων αντιλαμβανόμαστε ότι στην πραγματικότητα τα μπλέ δεδομένα βρίσκονται πολύ μακριά απο τα κόκκινα. Ο παραπάνω εσφαλμένος συλλογισμός αναπαρίσταται στο παρακάτω γράφημα.



Σχήμα 3.2: Διάσχιση της γεωμετρίας - Swiss Roll.

Αντιλαμβανόμαστε λοιπόν, μέσω της παραπάνω απεικόνισης ότι θα πρέπει να ληφθεί υπόψιν η γεωμετρία του προβλήματος ώστε σε καμιά περίπτωση υπολογίζοντας κοντινότερες αποστάσεις να συμπεριλάβουμε το αρχικό και το τελικό σημείο ως κοντινούς γείτονες, ενώντάς τα απευθείας μεταξύ τους. Αυτή είναι και η διαφορά των αλγορίθμων μη γραμμικής μείωσης διαστάσεων με αυτούς της γραμμικής. Για να γίνει πλήρως κατανοητός ο τρόπος μείωσης των διαστάσεων του παραπάνω σετ δεδομένων, δίνεται η απεικόνιση των δεδομένων σε χώρο χαμηλής διάστασης μετά απο την εφαρμογή αλγορίθμου μη γραμμικής μείωσης διαστάσεων.



Σχήμα 3.3: Μείωση της διάστασης του Swiss Roll απο τον τρισδιάστατο στον δυσδιάστατο χώρο.

Απο την παραπάνω απεικόνιση μπορούμε να συμπεράνουμε ότι κάνοντας μείωση των διαστάσεων στην πραγματικότητα "ξετυλίξαμε" το Swiss Roll[8] και έτσι απο τον αρχικό χώρο των τριών διαστάσεων στην πραγματικότητα η εγγενής διάσταση των δεδομένων είναι ίση με δύο. Στις επόμενες ενότητες θα γίνει παρουσίαση των πιο γνωστών μεθόδων μη γραμμικής μείωσης διαστάσεων καθώς επίσης θα γίνει και η μαθηματική τους ανάλυση.

### 3.2.1 ISOMAP

Ένας βασικός αλγόριθμος μη γραμμικής μείωσης διαστάσεων είναι ο αλγόριθμος Ισομετρική απεικόνιση (Isometric Mapping - ISOMAP)[9]. Ο αλγόριθμος αυτός υιοθετεί την άποψη ότι μόνο οι γεωδαιτικές αποστάσεις μεταξύ όλων των ζευγών των σημείων των δεδομένων μπορούν να αντικατοπτρίσουν την πραγματική δομή της πολλαπλότητας του προβλήματος. Η παραπάνω διατύπωση αντικατοπτρίζει το παράδειγμα που δόθηκε στο γράφημα (3.2), και τονίζει το γεγονός ότι οι Ευ-

κλείδιες αποστάσεις μεταξύ σημείων μιας πολλαπλότητας δεν μπορούν να την αναπαραστήσουν ικανοποιητικά. Αυτό διότι σημεία (στο γράφημα τα δύο σημεία που έχουν επισυμανθεί με μαύρους κύκλους) που είναι απομακρυσμένα μεταξύ τους σύμφωνα με την γεωδαιτική απόσταση, μπορεί να θεωρηθούν λανθασμένα, κοντικά ως προς την Ευκλείδια απόστασή τους.

Ουσιαστικά η μέθοδος ISOMAP[9] είναι μια παραλλαγή του αλγορίθμου Multi Dimensional Scaling - MDS[5], με την διαφορά ότι οι Ευκλείδιες αποστάσεις αντικαθίστανται απο τις αντίστοιχες γεωδαιτικές κατά μήκος της πολλαπλότητας των δεδομένων. Η ουσία του αλγορίθμου είναι να εκτιμηθούν σωστά οι γεωδαιτικές αποστάσεις μεταξύ σημείων τα οποία είναι απομακρυσμένα μεταξύ τους. Ο αλγόριθμος μπορεί να χωριστεί σε δύο βασικά βήματα:

**Βήμα-1**:

Για κάθε σημείο $x_i, i = 1, 1 \ldots, n$, υπολόγισε τους πλησιέστερους γείτονες και κατασκέυασε έναν γράφο $G(V, E)$ του οποίου οι κορυφές αναπαριστούν πρότυπα εισόδου και οι ακμές συνδέουν τους πλησιέστερους γείτονες. Οι παράμετροι $k$ (αριθμός των κοντινών γειτόνων κάθε σημείου) ή $\epsilon$ (ακτίνα σφαίρας στην οποία ανήκουν γειτονικά σημεία) είναι παράμετροι που καθορίζονται απο τον χρήστη. Στις ακμές ανατίθενται βάρη σύμφωνα με τις αντίστοιχες Ευκλείδιες αποστάσεις (για τους πλησιέστερους γείτονες αυτή είναι μια καλή προσέγγιση της γεωδαιτικής απόστασης).

**Βήμα-2**:

Υπολόγισε ανα ζεύγος την γεωδαιτική απόσταση για όλα τα ζεύγη κατα μήκος των συντομότερων διαδρομών μέσα στον γράφο. Το πιο σημαντικό σημείο, είναι ότι η γεωδαιτική απόσταση μεταξύ δύο οποιονδήποτε σημείων της πολλαπλότητας μπορεί να προσεγγιστεί μέσω της συντομότερης διαδρομής που ενώνει τα δύο σημείο στο γράφο $G(V, E)$. Ο πιο γνωστός αλγόριθμος υλοποίησης της παραπάνω διαδικασίας είναι ο αλγόριθμος Djikstar με πολυπλοκότητα $\mathcal{O}(n^2 \ln n + n^2 k)$, μέγεθος απαγορευτικό για τις περισσότερες πρακτικές εφαρμογές.

Εφόσον έχουν εκτελεστεί τα δύο αυτά βήματα είμαστε πλέον σε θέση νε εφαρμόσουμε την κλασική μέθοδο MDS[5]. Το πρόβλημα λοιπόν απο εδώ και στο εξής γίνεται ισοδύναμο με την εφαρμογή της ανάλυσης ιδιοδιανυσμάτων του αντίστοιχου μητρώου Gram και την επιλογή των m περισσότερο σημαντικών ιδιοδιανυσμάτων για την αναπαράσταση του χώρου χαμηλής διάστασης. Μετά απο αυτή την αναπαράσταση, οι Ευκλείδιες αποστάσεις μεταξύ των σημείων του χώρου χαμηλής

διάστασης ταιριάζουν με τις αντίστοιχες γεωδαιτικές αποστάσεις στην πολλαπλότητα του αρχικού χώρου υψηλής διάστασης. Όπως και στις μεθόδους PCA[4] και MDS[5]. η διάσταση m εκτιμάται απο το πλήθος των m περισσότερο σημαντικών ιδιοτιμών. Αποδεικνύεται τέλος ότι η μέθοδος ISOMAP ασυμπτωτικά ($n \to \inf$) θα ανακτήσει την αληθινή διάσταση για ένα σύνολο δεδομένων μη γραμμικής πολλαπλότητας.

### 3.2.2 Laplassian Eigenmaps

Η μέθοδος Laplassian Eigenmaps[10] στηρίζεται στην υπόθεση ότι τα σημεία των δεδομένων βρίσκονται σε μια λεία πολλαπλότητα $M \supset Q$, της οποίας η εγγενής διάσταση είναι ίση με $m < N$ και είναι ενσωματωμένη στον $\Re^N$, δηλαδή $M \supset \Re^N$. Η διάσταση $m$ δίνεται ως παράμετρος απο τον χρήστη και εξαρτάται απο το σύνολο των δεδομένων για κάθε εφαρμογή. Η κύρια φιλοσοφία πίσω απο την μέθοδο είναι να υπολογιστεί η αναπαράσταση των δεδομένων σε χώρο χαμηλής διάστασης, έτσι ώστε η τοπική πληροφορία γειτνίασης στον χώρο $Q \supset M$ να διατηρείται κατά βέλτιστο τρόπο. Με τον τρόπο αυτό προσπαθούμε να βρούμε μια λύση που ανταναικλά τη γεωμετρική δομή της πολλαπλότητας. Για την επίτευξη αυτού απαιτούνται τα παρακάτω βήματα:

**Βήμα-1**: Κατασκευή ενός γράφου $G = (V, E)$, όπου $V = v_i, i = 1, 2, \ldots, n$ είναι ένα σύνολο κορυφών και $E = \epsilon_{ij}$ το σύνολο των ακμών που συνδέουν κορυφές $(v_i, v_j)$. Κάθε κόμβος $v_i$ του γράφου αντιστοιχεί σε ένα σημείο $\mathbf{x}_i$ του συνόλου των δεδομένων $X$. Συνδέουμε τις $v_i$,$v_j$, δηλαδή εισάγουμε την ακμή $\epsilon_{ij}$ μεταξύ των αντίστοιχων κόμβων, αν τα σημεία $\mathbf{x}_i, \mathbf{x}_j$ είναι μεταξύ τους κοντινά. Η μέθοδος ορίζει την εγγύτητα αυτή με δύο τρόπους:
1. $\|\mathbf{x}_i - \mathbf{x}_j\|^2 < \epsilon$, για κάποια παράμετρο $\epsilon$ η οποία ορίζεται απο τον χρήστη. Με $\|\cdot\|$ ορίζουμε την πράξη της Ευκλείδιας νόρμας στον χώρο $\Re^N$.
2. Το $\mathbf{x}_j$ είναι μεταξύ των $k$ πλησιέστερων γειτόνων του $\mathbf{x}_i$ ή και αντίστροφα, με το $k$ να είναι και σε αυτή την περίπτωση είσοδος η οποία καθορίζεται απο τον χρήστη. Επίσης οι γείτονες επιλέγονται χρησιμοποιώντας την μετρική της Ευκλείδιας απόστασης στον χώρο $\Re^N$. Η χρήση της Ευκλείδιας απόστασης αιτιολογείται απο την υπόθεση ότι η πολλαπλότητα είναι λεία, γεγονός που μας επιτρέπει να προσεγγίσουμε, τοπικά, τη γεωδαισία της πολλαπλότητας με Ευκλείδιες αποστάσεις.

Για να αποσαφηνιστεί πλήρως η παραπάνω διατύπωση δίνεται το χαρακτηριστικό παράδειγμα όπου

θεωρούμε μια σφαίρα ενσωματωμένη στον τρισδιάστατο χώρο, και έστω κάποιος περιορίζεται να ζεί πάνω στην επιφάνεια της σφαίρας. Τότε η συντομότερη διαδρομή απο ένα σημείο της σφαίρας σε ένα άλλο είναι η γεωδαιτική διαδρομή μεταξύ των δύο σημείων. Προφανώς αυτή δεν θα είναι ευθεία γραμμή, αλλά ένα τόξο στην επιφάνεια της σφαίρας. Παρ΄όλα αυτά όμως, αν τα δύο σημεία είναι πολύ κοντά μεταξύ τους, η γεωδαιτική απόσταση μπορεί να προσεγγιστεί απο την Ευκλείδια απόσταση, υπολογισμένη στον τρισδιάστατο χώρο.

**Βήμα-2**: Κάθε ακμή $\epsilon_{ij}$ συσχετίζεται με ένα βάρος $W(i,j)$. Για κόμβους που δεν συνδέονται μεταξύ τους, τα αντίστοιχα βάρη είναι μηδέν. Κάθε βάρος $W(i,j)$ είναι ένα μέτρο της εγγύτητας των αντίστοιχων γειτόνων $\mathbf{x}_i, \mathbf{x}_j$. Μια τυπική επιλογή είναι

$$W(i,j) = \begin{cases} \exp(\|\frac{\mathbf{x}_i - \mathbf{x}_j}{\sigma^2}\|) & , if \quad x_i, x_j \quad neighbors \\ 0 & , not \quad neighbors \end{cases}$$

με $\sigma^2$, παράμετρος η οποία ορίζεται και αυτή απο τον χρήστη. Σχηματίζουμε το μητρώο βαρών $W$, μεγέθους $(n \times n)$, το οποίο έχει για στοιχεία τα βάρη $W(i,j)$. Σημειώνουμε ότι το $W$ είναι συμμετρικό και αραιό αφού στην πράξη προκύπτει ότι πολλά απο τα στοιχεία του είναι μηδενικά.

**Βήμα-3**: Ορίζεται το διαγώνιο μητρώο $D$ με στοιχεία $D_{ij} = \sum_j W(i,j), i = 1, 2, \ldots, n$, καθώς και το μητρώο $L = D - W$. Το τελευταίο είναι γνωστό ως το μητρώο Laplace του γράφου $G = (V, E)$. Εφαρμόζεται η γενικευμένη ανάλυση σε ιδιοτιμές και ιδιοδιανύσματα

$$\Lambda \mathbf{v} = \lambda D \mathbf{v}$$

Έστω $0 = \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_m$, οι $m+1$ μικρότερες ιδιοτιμές. Αγνοείται η ιδιοτιμή $\mathbf{v}_0$ που αντιστοιχεί στην ιδιοτιμή $\lambda_0 = 0$ και επιλέγονται τα υπόλοιπα $m$ ιδιοδιανύσματα $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m$,. Στην συνέχεια εκτελείται η απεικόνιση

$$\mathbf{x}_i \in \Re^N \mapsto \mathbf{y}_i \in \Re^m, i = 1, 2, \ldots, n$$

όπου

$$\mathbf{y}_i^T = [\mathbf{v}_1(i), \mathbf{v}_2(i), \ldots, \mathbf{v}_m(i)], i = 1, 2, \ldots, m$$

Όπως έχουμε αναλύσει στην αντίστοιχη εντότητα η πολυπλοκότητα υπολογισμού ιδιοτιμών και ιδιοδιανυσμάτων είναι, γενικά, της τάξης $\mathcal{O}(n^3)$. Ωστόσο για αραιά μητρώα, όπως στην συγκεκριμένη περίπτωση το $L$, μπορούν να εφαρμοστούν αποτελεσματικές τεχνικές με αποτέλεσμα την μείωση της πολυπλοκότητας σε τάξη κάποιο πολλαπλάσιο του $\mathcal{O}(n^2)$. Η πιο γνωστή και αποτελεσματική τεχνική για τον σκοπό αυτό είναι ο αλγόριθμος Lanczos[11].

Ο αλγόριθμος Laplassian Eigenmaps[10] ο οποίος αναλύθηκε παραπάνω, ανήκει στην ίδια κατηγορία (*μέθοδοι μείωσης διαστάσεων που βασίζονται σε γράφους*) με τον αλγόριθμο **Locally Linear Embeddings - LLE**[1] ο οποίος αποτελεί και βασικό αντικείμενο της εν λόγω διατριβής. Οι δύο αλγόριθμοι έχουν πολύ κοινή λογική και μεθοδολογία και γι αυτό στο επόμενο κεφάλαιο στο οποίο γίνεται αναλυτική μαθηματική ανάλυση του LLE[1] θα αποσαφηνιστούν και τα παραπάνω βήματα του Laplassian Eigenmaps[10] καθώς τα βήματα τους είναι πανομοιότυπα.

# Κεφάλαιο 4

# Τοπική Γραμμική Ενσωμάτωση (LLE)

Ο αλγόριθμος Locally Linear Embeddings (LLE)[1] ανήκει στην κατηγορία αλγορίθμων μη γραμμικής μείωσης διαστάσεων με την χρήση γράφων και αποτελεί μια απο τις αποτελεσματικότερες αλλά και γρηγορότερες τεχνικές αυτού του είδους. Όπως αναφέραμε και στο προηγούμενο κεφάλαιο, βασική υπόθεση της μεθόδου είναι ότι τα δεδομένα μας βρίσκονται σε μια αρκετά λεία πολλαπλότητα, διάστασης $m$, και η οποία είναι ενσωματωμένη στον υποχώρο του $\Re^N$, με $m < N$. Η υπόθεση για το λείο της πολλαπλότητας μας επιτρέπει να υποθέσουμε επιπλέον ότι, με δεδομένη την ύπαρξη αρκετών δεδομένων και ότι η πολλαπλότητα είναι "καλά" δειγματοληπτημένη, τα κοντινά σημεία βρίσκονται πάνω (ή κοντά) σε ένα "τοπικό" γραμμικό τμήμα της πολλαπλότητας.

## 4.1 Ο αλγόριθμος ως τεχνική μη γραμμικής μείωσης διαστάσεων

Δεδομένης της αποτελεσματικότητας του αλγορίθμου να ανακαλύπτει τον χώρο μειωμένης διάστασης στον οποίο βρίσκεται ενσωματωμένη η πληροφορία ενός προβλήματος, ο αλγόριθμος έχει χρησιμοποιηθεί με επιτυχία σε αρκετές πρακτικές εφαρμογές. Ιδιαίτερο ενδιαφέρον παρουσιάζουν νέες μελέτες κυρίως απο τον χώρο της Ιατρικής [12] [13]. Απο τις αναφορές αυτές είναι φανερό ότι ο ρόλος της μείωσης των διαστάσεων μπορεί να καθορίσει σε μεγάλο βαθμό την βελτίωση του αποτελέσματος ταξινόμησης. Στις συγκεκριμένες περιπτώσεις στόχος είναι να γίνει σωστή

πρόβλεψη για το αν κάποιος ασθενής πάσχει απο μια συγκεκριμένη ασθένεια ή βρίσκεται στην ευπαθή ομάδα με μεγάλη πιθανότητα να του παρουσιαστεί στο μέλλον. Φαίνεται οτι ο αλγόριθμος LLE[1] είναι ένα πολύ ισχυρό εργαλείο το οποίο μπορεί να υλοποιήσει την μείωση των διαστάσεων σε τέτοιου είδους εφαρμογές και μάλιστα επιφέροντας σημαντικά και ουσιαστικά αποτελέσματα. Ο αλγόριθμος επίσης, έχει χρησιμοποιηθεί και σε εφαρμογές ταξινόμησης με σετ δεδομένων ευρέως διαδεδομένα στον χώρο της αναγνώρισης προτύπων [14] [15] [16], όπως το σετ δεδομένων με χειρόγραφα ψηφία MNIST[17].

## 4.2 Μαθηματική ανάλυση και υλοποίηση του αλγορίθμου Locally Linear Embeddings

Ο αλγόριθμος LLE[1] αποτελεί το κύριο κομμάτι της εν λόγω διατριβής και η υλοποίηση του έχει στηριχθεί στον αλγόριθμο της παραπάνω αναφοράς. Ο ψευδοκώδικας είναι διαθέσιμος στην παρακάτω τοποθεσία LLE Algorithm Pseudocode. Παρ΄ όλα αυτά στην συγκεκριμένη υλοποίηση έχουν γίνει συγκεκριμένες βελτιστοποιήσεις σε κάποια βήματα του αλγορίθμου, όπως για παράδειγμα η χρήση του αλγορίθμου κοντινότερων γειτόνων, υλοποιημένο σε CUDA, με σκοπό την μείωση του χρόνου εκτέλεσης του συγκεκριμένου βήματος.

### 4.2.1 Βήμα-1: Εύρεση του πίνακα γειτνίασης

Κατά το πρώτο βήμα του αλγορίθμου γίνεται ο υπολογισμός των κοντινότερων γειτόνων για κάθε σημείο $X_i$ του συνόλου των δεδομένων. Στο βήμα αυτό ο χρήστης επιλέγει ανάλογα με την κάθε εφαρμογή έναν αριθμό γειτόνων K και χρησιμοποιεί κάποιον αλγόριθμο υπολογισμού κοντινότερων γειτόνων για κάθε ένα απο τα σημεία του δείγματος. Με τον τρόπο αυτό έχει υπολογιστεί ο τετραγωνικός πίνακας $N \times N$, ο οποίος δίνει πληροφορία για κάθε σημείο του δείγματος ως προς τους k κοντινότερους γείτονές του.

Ο τρόπος υπολογισμού του πίνακα αυτού στην συγκεκριμένη υλοποίηση γίνεται μέσω της συνάρτησης knnsearch του MATLAB για την σειριακή υλοποίηση και με την συνάρτηση gpu_knn για την

παράλληλη υλοποίηση. Η συνάρτηση αυτή αντιπροσωπεύει την κύρια συνάρτηση gpuknnHeap του πακέτου knn-toolbox, η οποία με την σειρά της αποτελεί το πέρασμα απο τον κώδικα MATLAB στην συνάρτηση πυρήνα υπολογισμού κοντινότερων γειτόνων με τη χρήση παράλληλης υλοποίησης σε CUDA γραμμένη στην γλώσσα προγραμματισμού C. Η υλοποίηση αυτή χρησιμοποιεί την Ευκλείδια απόσταση ως μέθοδο προσδιορισμού των κοντινότερων γειτόνων. Παρ΄ όλα αυτά στο βήμα αυτό μπορούν να χρησιμοποιηθούν και άλλες μετρικές γειτνίασης όπως για παράδειγμα ο προσδιορισμός κοντινών γειτόνων με τη χρήση σφαίρας ακτίνας $\epsilon$. Επίσης, μια άλλη γνωστή μέθοδος επίλυσης του βήματος αυτού η οποία βελτιώνει τον χρόνο εκτέλεσης είναι η χρήση KD-trees.

### 4.2.2 Βήμα-2: Εύρεση του πίνακα βαρών W

Στο δεύτερο αυτό βήμα του αλγορίθμου στόχος είναι να υπολογιστεί ο πίνακας βαρών $W(i,j), i,j = 1,2,\ldots,n$, μέσω των οποίων είναι εφικτή η ανακατασκευή του κάθε δείγματος $X_i$ μέσω των βαρών που αντιστοιχούν στους κοντινούτερους γείτονές του. Πιο απλά στο βήμα αυτό θέλουμε να προσδιορίζουμε κάθε σημείο $X_i$ του δείγματός μας, ελαχιστοποιώντας την συνάρτηση κόστους

$$\arg\min_{w} E_w = \sum_{i=1}^{n} \|\mathbf{X_i} - \sum_{j=1}^{n} W(i,j)\mathbf{X}_j\|^2 \tag{4.2.1}$$

η οποία στην πραγματικότητα αυτό που προσπαθεί να ανακαλύψει είναι οι κοντινότεροι γείτονες $j$ του σημείου $X_i$ οι οποίοι ασκούν την σημαντικότερη επιρροή πάνω του ως προς την ανακατασκευή του. Η ελαχιστοποίηση της συνάρτησης αυτής γίνεται εφαρμόζοντας τον αλγόριθμο ελαχίστων τετραγώνω Least squares εξασφαλίζοντας παράλληλα κάποιες απαραίτητες ιδιότητες για τον πίνακα βαρών $W$. Καταρχήν πρέπει να ισχύει ότι το κάθε δείγμα $X_i$ θα πρέπει να μπορεί να ανακατασκευαστεί μόνο απο τους κοντινότερους γείτονές του γεγονός που θέτει τον περιορισμό $W(i,j) = 0$ στην περίπτωση κατά την οποία το $j$ στοιχείο δεν είναι γείτονας του $i$. Επίσης θα πρέπει τα στοιχεία κάθε γραμμής του μητρώου βαρών $W$ να αθροίζονται στην μονάδα, δηλαδή $\sum_{j=1}^{n} W(i,j) = 1$, ώστε να εξασφαλιστεί η αμεταβλητότητα κατά την μεταφορά. Με τους περιορισμούς αυτούς λοιπόν εξασφαλίζεται ότι τα βάρη τα οποία ελαχιστοποιούν την παραπάνω συνάρτηση κόστους είναι αμετάβλητα κατά την περιστροφή, την μεταφορά και την κλιμάκωση.

Για να γίνει κατανοητή η παραπάνω διαδικασία ακολουθούμε τον εξής συλλογισμό. Ας πάρουμε για παράδειγμα ένα σημείο $x$ το οποίο έχει $K$ κοντινούς γείτονες $n_j$ και βάρη ανακατασκευής $w_j$ για τα οποία ισχύει η συνθήκη $\sum_j w_j = 1$. Τότε μπορούμε να γράψουμε την συνάρτηση κόστους ως

$$\epsilon = \mid \overrightarrow{x} - \sum_j w_j \overrightarrow{n}_j \mid^2 = \mid \sum_j w_j (\overrightarrow{x} - \overrightarrow{n}_j) \mid^2 = \sum_{jk} w_j w_k C_{jk} \tag{4.2.2}$$

Στην παραπάνω σχέση χρησιμοποιήσαμε το μητρώο Gram το οποίο ορίζεται ως

$$C_{jk} = (\overrightarrow{x} - \overrightarrow{n}_j) \cdot (\overrightarrow{x} - \overrightarrow{n}_k) \tag{4.2.3}$$

Εκ κατασκευής για τον πίνακα Gram έχουμε ότι είναι συμμετρικός και θετικά ημιορισμένος. Σύμφωνα με τα παραπάνω λοιπόν τα βέλτιστα βάρη ανακατασκευής $w_j$ της συνάρτησης κόστους μπορούν να υπολογιστούν, αφού μέσω του πολλαπλασιαστή Lagrange εξασφαλίσουμε τη συνθήκη $\sum_j w_j = 1$, μέσω της επίλυσης του συστήματος

$$w_j = \frac{\sum_k C_{jk}^{-1}}{\sum_{lm} G_{lm}^{-1}} \tag{4.2.4}$$

Όπως είχαμε αναφέρει στην παράγραφο (2.3) οι πίνακες $X^T X$(πίνακας συνδιασποράς) και $XX^T$(πίνακας Gram) έχουν τις ίδιες ιδιοτιμές και ιδιοδιανύσματα τα οποία σχετίζονται μεταξύ τους. Για τον λόγο αυτό μπορεί να παραληφθεί η αντιστροφή του πίνακα Gram, όπως φαίνεται και στην υλοποίηση που παρατέθηκε παραπάνω, λύνοντας το σύστημα $\sum_j C_{jk} w_k = 1$ και έπειτα απαιτώντας τον περιορισμό $\sum_j w_j = 1$ ο οποίος υλοποιείται με την τελευταία γραμμή του παραπάνω κώδικα. Επίσης βλέπουμε ότι στην υλοποίηση του κώδικα δεν υπολογίζεται ο πίνακας Gram αλλά αυτός της συνδιασποράς και στην συνέχεια ακολουθείται η παραπάνω διαδικασία. Τελευταία διευκρίνηση για τη γραμμή 5 του κώδικα, στην οποία γίνεται κανονικοποίηση του πίνακα συνδιασποράς. Αυτό απαιτείται στην περίπτωση για την οποία ο πίνακας συνδιασποράς προκύπτει μοναδιαίος ή πολύ κοντά σε αυτόν, οπότε και δεν υπάρχει μοναδική λύση του συστήματος.

Καταλήγουμε λοιπόν μέσω της παραπάνω διαδικασίας στον υπολογισμό του μητρώου βαρών W για το σύνολο των δεδομένων. Ο τρόπος μάλιστα με τον οποίο έγινε ο υπολογισμός αυτός εξασφαλίζει το γεγονός ότι η εσωτερική ενσωματωμένη γεωμετρία η οποία υπάρχει στην γειτονιά ενός σημείου $X_i$ του συνόλου των δεδομένων θα εξακολουθεί να υπάρχει και στον χώρο της μειωμένης διάστασης. Το γεγονός αυτό εξασφαλίζεται απο την ανεξαρτησία των βαρών ως προς την περιστροφή, την μεταφορά και την κλιμάκωση αλλά και απο το γεγονός ότι οι γείτονες του σημείου $X_i$ στον χώρο αρχικών διαστάσεων $D$ θα εξακολουθούν να αποτελούν γείτονες του σημείου $Y_i$ (προβολή του $X_i$ απο τον χώρο υψηλής διάστασης στο σημείο $Y_i$ χαμηλής διάστασης). Αυτό συμβαίνει επίσης, διότι όπως θα δούμε παρακάτω τα βάρη με τα οποία γίνεται ανακατασκεύη του $X_i$ τα ίδια θα χρησιμοποιηθούν και για την κατασκευή του $Y_i$ στον χώρο μειωμένης διάστασης. Συνέπεια λοιπόν των παραπάνω είναι ότι τα βάρη $w_j$ που υπολογίστηκαν δεν εξαρτώνται απο το εκάστοτε σημείο αλλά κωδικοποιούν πληροφορία σχετική με τα εγγενή χαρακτηριστικά κάθε γειτονιάς τα οποία και διατηρούνται κατά την ενσωμάτωση των δεδομένων στον χώρο χαμηλότερης διάστασης.

### 4.2.3 Βήμα-3: Επιλογή των τελικών διαστάσεων με τη χρήση του πίνακα W

Στο τελευταίο βήμα του αλγορίθμου πραγματοποιείται η μείωση των διαστάσεων των δειγμάτων απο τον χώρο υψηλής διάστασης $D$ σε έναν χαμηλότερης $d$. Η διαδικασία αυτή πραγματοποιείται όπως αναφέραμε και παραπάνω χρησιμοποιώντας τον πίνακα των βαρών $W$ και τα οποία έχουν την ιδιότητα ότι αντανακλούν τις εγγενείς ιδιότητες της τοπικής γεωμετρίας στην οποία υπόκεινται τα δεδομένα. Η λύση λοιπόν προκύπτει επιλύοντας και πάλι ένα πρόβλημα ελαχιστοποίησης το οποίο ορίζεται ως

$$\arg\min_{w} E_y = \sum_{i=1}^{n} \|\mathbf{Y_i} - \sum_{j=1}^{n} W(i,j)\mathbf{Y}_j\|^2 \tag{4.2.5}$$

Και σε αυτή την περίπτωση απαιτούμε την διατήρηση των συνθηκών, $\sum_i Y_i = 0$ ώστε να εξασφαλιστεί η αμεταβλητότητα ως προς την μεταφορά, και $\frac{1}{N}\sum_i Y_i Y i^T = I$ η οποία εξασφαλίζει ότι οι διαστάσεις $d$ θα είναι δευτέρου βαθμού ασυσχέτιστες , ότι τα βάρη ανακατασκευής για τις δια-

στάσεις $d$ θα υπολογιστούν σε κοινή κλίμακα και ότι αυτή η κλίμακα θα είναι μοναδιαίου βαθμού. Ο πίνακας $I$ συμβολίζει τον μοναδιαίο πίνακα διάστασης $d \times d$.

Η λύση της εξίσωσης (4.2.5) για τα άγνωστα στοιχεία $y_i, i = 1, 2, \ldots, n$, είναι ισοδύναμη με την εύρεση των $d+1$ μικρότερων ιδιοτιμών του τετραγωνικού πίνακα $M$ ο οποίος προκύπτει απο την σχέση

$$E_y = \sum_{i=1}^{n} \|\mathbf{Y_i} - \sum_{j=1}^{n} W(i,j)\mathbf{Y}_j\|^2 = \mid (I-W)Y \mid^2 = Y_T M Y \tag{4.2.6}$$

και ισούτε με

$$M = (I-W)^T(I-W) \tag{4.2.7}$$

Ο πίνακας αυτός έχει διαστάσεις $N \times N$, όπου $N$ το πλήθος των δεδομένων εισόδων. Παρ΄όλα αυτά ο πίνακας αυτός στην πράξη προκύπτει αραιός (sparce matrix) γεγονός το οποίο απλοποιεί σημαντικά τους υπολογισμούς, ιδιαίτερα για μεγάλες τιμές του $N$. Στην συνέχεια χρησιμοποιώντας τον πολλαπλασιαστή Lagrange καταλήγουμε στην επίλυση της εξίσωσης

$$(M - \Lambda)Y^T = 0 \tag{4.2.8}$$

όπου $\Lambda$ είναι ο διαγώνιος πίνακας των πολλαπλασιαστών Lagrange.

Η παραπάνω επίλυση του προβλήματος ανάλυσης ιδιοτιμών μας οδηγεί στον προσδιορισμό των ιδιοδιανυσμάτων τα οποία αποτελούν και λύσεις του πίνακα $M$. Επίσης τα ιδιοδιανύσματα με τις μεγαλύτερες ιδιοτιμές είναι αυτά τα οποία ελαχιστοποιούν την συνάρτηση κόστους την οποία και θέλαμε να επιλύσουμε. Στο σημείο αυτό λοιπόν είμαστε σε θέση να προσδιορίσουμε τις τελικές διαστάσεις οι οποίες αντιπροσωπεύουν τα δεδομένα μας στον νέο χώρο μειωμένης διάστασης. Οι τελικές αυτές διαστάσεις $d$ λαμβάνουν υπόψιν τους περιορισμούς οι οποίοι έχουν τεθεί και έτσι με τον τρόπο αυτό εξασφαλίζεται η διατήρηση των γεωμετρικών χαρακτηριστικών της κάθε γειτονιάς για όλα τα σημεία $X_i$ του αρχικού συνόλου δεδομένων μεγέθους $N$.

Σημαντικό σημείο στην παραπάνω διαδικασία αποτελεί το γεγονός ότι δεν λαμβάνουμε υπόψιν την μικρότερη ιδιοτιμή της λύσης του παραπάνω συστήματος και αυτό διότι ισούται με το μηδέν. Η ιδιοτιμή αυτή αντιπροσωπεύει το μοναδιαίο διάνυσμα το οποίο εξασφαλίζει ότι το σύνολο των δεδομένων έχει μηδενική μέση τιμή, εξασφαλίζοντας έτσι τον περιορισμό ως προς την αμεταβλητότητα κατά την μεταφορά.

Η εύρεση των ιδιοτιμών-ιδιοδιανυσμάτων γίνεται με τον αλγόριθμο Lanczos[11] ο οποίος βελτιστοποιεί σε μεγάλο βαθμό την επίλυση του προβλήματος, απο την στιγμή που ο τετραγωνικός πίνακας $M$ του συστήματος είναι αραιός και θετικά ημιορισμένος.

# Κεφάλαιο 5

# Τεχνικές μείωσης της πολυπλοκότητας του αλγορίθμου LLE

Το σημαντικότερο αποτέλεσμα της εργασίας αυτής είναι οι δύο νέες συνδυαστικές μέθοδοι που προτείνονται με τις οποίες μπορεί κάποιος να αποφύγει το τεράστιο υπολογιστικό κόστος που απαιτείται για την εκτέλεση του αλγορίθμου. Συγκεκριμένα το τελευταίο βήμα του αλγορίθμου το οποίο είναι και το πιο απαιτητικό έχει πολυπλοκότητα $\mathcal{O}(N^3)$ στην γενική περίπτωση ενώ στην συγκεκριμένη περίπτωση λόγω του αραιού μητρώου $M$ είναι της τάξης $\mathcal{O}(N^2)$. Αντιλαμβανόμαστε λοιπόν ότι ακόμα και για ένα σχετικά μικρό σετ δεδομένων, για τα δεδομένα του κλάδου της μηχανικής μάθησης, το πρόβλημα που έχουμε να αντιμετωπίσουμε έχει απαγορευτικές διαστάσεις.

Ένα άλλο πρόβλημα που συναντάει κανείς κατά την εφαρμογή του αλγορίθμου σε κάποιο σετ δεδομένων είναι ο εξής περιορισμός. Ας υποθέσουμε ότι έχουμε ένα σύνολο δεδομένων μεγέθους $N$, απο τα οποία για κάποιον αριθμό $N1$ απο αυτά γνωρίζουμε την ετικέτα τους. Με τον όρο ετικέτα εννοούμε την τελική κλάση στην οποία ανήκει το κάθε δείγμα. Για τα υπόλοιπα δείγματα, έστω μεγέθους $N2$ δεν γνωρίζουμε την ετικέτα τους και είναι αυτά τα δείγματα για τα οποία θέλουμε να εξάγουμε το συμπέρασμα. Το συμπέρασμα αυτό είναι φυσικά η τελική απόφαση ως προς σε ποιά κλάση θα πρέπει να ταξινομηθεί το καθένα απο αυτά. Προφανώς η παραπάνω απόφαση προκύπτει λαμβάνοντας υπόψιν την πληροφορία την οποία μας δίνει το σύνολο των δεδομένων $N1$ τα οποία στον χώρο της μηχανικής μάθησης αναφέρονται ως το σύνολο των δεδομένων εκπάιδευσης (train

data). Τα υπόλοιπα δείγματα $N2$ αναφέρονται ως το σύνολο των δεδομένων αξιολόγησης (test data).

Στο συγκεκριμένο λοιπόν έστω οτι τα δείγματα του αρχικού χώρου έχουν αρχική διάσταση μεγέθους $D$ και μέσω του αλγόριθμου μείωσης των διαστάσεων θέλουμε να βρεθούμε σε έναν νέο χώρο διάστασης $d$, προφανώς με $d < D$. Στην περίπτωση αυτή λοιπόν ο πιο απλός συλλογισμός που θα μπορούσε να κάνει κάποιος είναι να εφαρμόσει τον αλγόριθμο LLE πάνω στο σετ δεδομένων εκπαίδευσης ώστε να έχει ένα σύνολο δεδομένων μεγέθους $N1$, διάστασης $d$. Με τον ίδιο ακριβώς τρόπο θα μπορούσε να έχει και το δεύτερο σετ δεδομένων, τα δεδομένα αξιολόγησης, μεγέθους $N2$ και αυτά διάστασης $d$. Έπειτα για την ταξινόμηση των αποτελεσμάτων θα μπορούσε να εφαρμοστεί ο αλγόριθμος ταξινόμησης κοντινότερων γειτόνων ανάμεσα στο σετ αξιολόγησης με το σετ εκπαίδευσης. Στη συνέχεια ανάλογα με την κλάση στην οποία ανήκει ο κοντινότερος γείτονας απο το σετ εκπαίδευσης για κάθε ένα στοιχείο των δεδομένων αξιολόγησης, θα καταλήγαμε στην τελική απόφαση για την κλάση στην οποία ανήκει κάθε ένα απο τα δεδομένα του σετ $N2$. Προφανώς ο αλγόριθμος κοντινότερων γειτόνων θα εφασρμοσθεί στον χώρο μειωμένης διάστασης μεγέθους $d$ χρησιμοποιώντας για παράδειγμα την μετρική της Ευκλείδιας απόστασης μεταξύ των σημείων.

Αν λοιπόν εφαρμόσουμε την παραπάνω διαδικασία για κάποιο σετ δεδομένων, θα παρατηρήσουμε ότι το τελικό αποτέλεσμα της ταξινόμησής μας έχει πολύ μικρή επιτυχία. Αυτό συμβαίνει διότι, οι δύο υποχώροι οι οποίοι προέκυψαν απο το τελικό βήμα του αλγορίθμου LLE, κατά το οποίο υπολογίστηκε ο νέος χώρος μειωμένης διάστασης για κάθε ένα απο τα δύο σύνολα δεδομένων, έχουν διαφορετική διανυσματική βάση και δεν μπορούν σε καμιά περίπτωση να συσχετιστούν μεταξύ τους ώστε να μπορέσουμε απο τα δεδομένα του ενός να καταλήξουμε σε κάποιο ορθό συμπέρασμα για τα δεδομένα του άλλου. Ο παραπάνω λοιπόν περιορισμός μας αναγκάζει να εφαρμόσουμε τον αλγόριθμο μείωσης των διαστάσεων στο σύνολο των δεδομένων, δηλαδή δίνοντας σαν είσοδο στον αλγόριθμο το σύνολο των δεδομένων μεγέθους $N = N1+N2$. Με τον τρόπο αυτό θα καταλήγαμε σε ένα νέο σετ δεδομένων μεγέθους $N$ αλλά διάστασης $d < D$. Τέλος σε αυτό το σετ δεδομένων μπορούμε τώρα να εφαρμόσουμε τον αλγόριθμο εύρεσης κοντινότερων γειτόνων για κάθε ένα απο τα δεδομένα αξιολόγησης ως πρός τα δεδομένα εκπαίδευσης, φυσικά στον χώρο $d$ διστάσεων, και έτσι να καταλήξουμε στην ορθή ταξινόμηση των δειγμάτων $N2$ ως προς την κλάση στην οποία

ανήκουν.

## 5.1 Μέθοδοι αντιμετώπισης της πολυπλοκότητας του προβλήματος

Όπως αντιλαμβανόμαστε απο την παραπάνω ανάλυση, η διαδικασία αυτή δεν είναι καθόλου πρακτική και μάλιστα δεν δίνει την δυνατότητα για λήψη αποφάσεων και ταξινόμησης δειγμάτων σε πραγματικό χρόνο. Αυτό διότι, για κάθε δείγμα αξιολόγησης που μας έρχεται ως είσοδος κάποια συγκεκριμένη χρονική στιγμή, και για το οποίο θέλουμε να το ταξινομήσουμε σε κάποια κλάση, θα πρέπει να το ενσωματώνουμε στο σετ των δεδομένων εκπαίδευσης και στην συνέχεια να εκτελούμε τον αλγόριθμο LLE[1]. Η συγκεκριμένη διαδικασία δεν προσφέρεται σε καμιά περίπτωση για πρακτικές εφαρμογές κατα τις οποίες μάλιστα ο στόχος μας είναι να γίνει μείωση των διαστάσεων ώστε να μπορούμε να λαμβάνουμε ταχύτερα και ακριβέστερα αποτελέσματα. Το γεγονός αυτό μάλιστα αντιτίθεται στην συνολική φιλοσοφία της μείωσης των διαστάσεων κατά την οποία μέσω της εφαρμογής της μπορεί να επιταχυνθεί σε πολύ μεγάλο βαθμό η διαδικασία της ταξινόμησης.

### 5.1.1 Μέθοδος-1: Προβολή στον χώρο των δεδομένων εκπαίδευσης

Για τους λόγους λοιπόν οι οποίοι αναλύθηκαν παραπάνω, προκύπτει η ανάγκη να βρεθεί κάποιος τρόπος με τον οποίο να μπορούμε οποιαδήποτε στιγμή να ταξινομήσουμε κάποιο δεδομένο, χρησιμοποιώντας βέβαια την πληροφορία που μπορούν να μας δώσουν τα δεδομένα του σετ εκπαίδευσης. Την λύση στο πρόβλημα αυτό λοιπόν έρχεται να δώσει η λογική με την οποία λειτουργεί ο αλγόριθμος LLE[1]. Πιο συγκεκριμένα όπως αναφέραμε παραπάνω, η μέθοδος αυτή έχει ως στόχο να διατηρήσει τα γεωμετρικά τοπικά χαρακτηριστικά για κάθε ένα απο τα δείγματα του συνόλου εκπαίδευσης τόσο στον χώρο των αρχικών διαστάσεων όσο και στον τελικό χώρο μειωμένης διάστασης. Επίσης, λόγω του ότι η ενσωμάτωση των δεδομένων στον χώρο μειωμένης διάστασης γίνεται με τη χρήση του πίνακα βαρών $W$, ο οποίος προσδιορίζει για κάθε δείγμα του αρχικού

χώρου την ανακατασκευή του μέσω των κοντινών του γειτόνων προκύπτει και η ιδέα της μεθόδου αυτής. Ο αντίστοιχος ψευδοκώδικας είναι ο παρακάτω

**Algorithm 1** Projection Method

Let $Xtrain$ be $[D \times N1]$ Train_dataset matrix and $Xtest$ be $[D \times N2]$ Test_dataset matrix ▷ N1,N2 declare the number of data and D the number of dimensions

Let matrix $Y$ be $[d \times N1]$ Train data, after dimensionality reduction ▷ d < D

Let matrix nn_graph with size $[N1 \times N2]$ and all elements equal to zero

**for** $i = 1$ to $N_2$ **do**
    Find K-Nearest Neighbors from $Xtrain$
**end for**

Keep the results to matrix IDX with size $[N2 \times K]$ ▷ K is the number of nearest neighbors
**for** $i = 1$ to $N_2$ **do**
    Set IDX(i,1:K) cells of nn_graph matrix equal to ones
    Make the matrix multiplication $Y \times$ nn_graph$(1 : N1, i)$ and store the result to
$testY(1 : d, i)$ ▷ testY(:,i) is the result of dimensionality reduced $Xtest_i$
**end for**

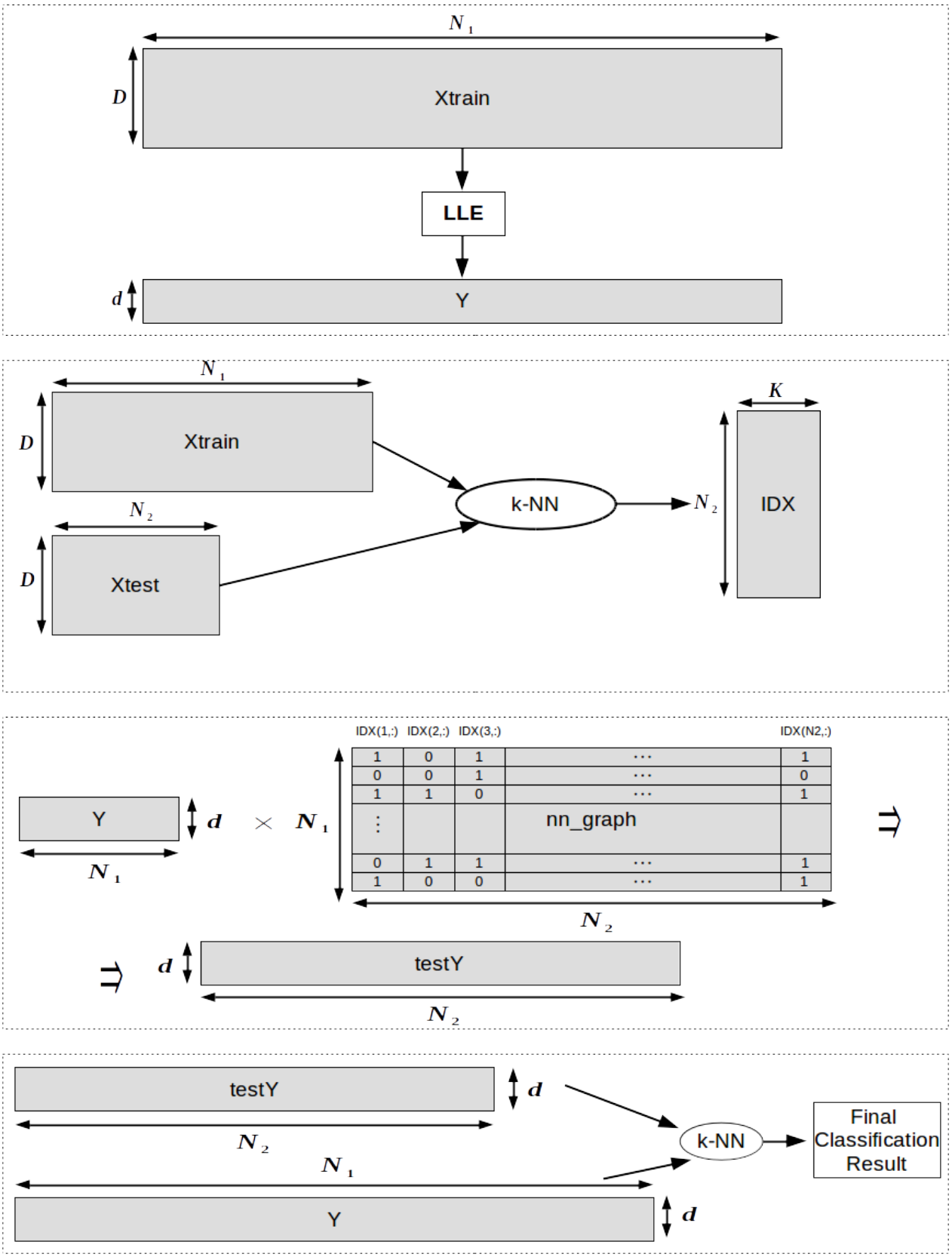
Final matrix testY has size $[d \times N2]$ and represents the projection of Xtest D-dimensional data into the d-dimensional emndedding subspace.

Now execute K-NN Classification between testY and Y datasets, to the d-dimensional space

Απο την παραπάνω ανάλυση της μεθόδου γίνεται φανερό ότι μπορούμε να χρησιμοποιήσουμε την πληροφορία των δεδομένων Y, τα οποία είναι τα δεδομένα εκπαίδευσης Xtrain στον χώρο μειωμένης διάστασης, ώστε να ταξινομήσουμε οποιοδήποτε δείγμα απο το σετ δεδομένων αξιολόγησης. Το γεγονός αυτό, σε συνδυασμό με την ελάχιστη αύξηση του σφάλματος ταξινόμησης όπως θα γίνει φανερό στα αποτελέσματα των πειραμάτων κάνει την μέθοδο αυτή πολύ ελκυστική για πρακτικές εφαρμογές. Τέτοιες εφαρμογές απαιτούν αποτελέσματα σε πραγματικό χρόνο και μάλιστα σε πολύ μεγάλες ταχύτητες γεγονός το οποίο εξασφαλίζεται απο τους υπολογισμούς στον χώρο των μειωμένων διαστάσεων $d$. Όπως μπορούμε να παρατηρήσουμε στο πρώτο βήμα της μεθόδου εφαρμόζουμε τον αλγόριθμο κοντινότερων γειτόνων στον χώρο των αρχικών διαστάσεων μεγέθους $D$, για το σετ δεδομένων αξιολόγησης ως προς το σετ δεδομένων εκπαίδευσης. Επομένως θα μπορούσαμε να ισχυριστούμε ότι το υπόλοιπο της διαδικασίας είναι περιττό εκτώς και αν το τελικό αποτέλεσμα ταξινόμησης είναι καλύτερο απο αυτό στον αρχικό χώρο. Παρ΄όλα αυτά αν αναλογιστο-

ύμε ένα πολύ μεγάλο σετ δεδομένων με έναν αρκετά μεγάλο αριθμό διαστάσεων, όπως πρόκειται άλλωστε για τα περισσότερα πραγματικά σετ στον χώρο της αναγνώρισης προτύπων, τότε το γεγονός οτι απο το σημείο αυτό και μετά μπορούμε την πληροφορία των $D$ διαστάσεων να την πάρουμε απο τις πολύ λιγότερες όπως θα δούμε $d$ τελικές διαστάσεις αποτελεί τεράστιο κέρδος. Το κέρδος είναι τόσο σε κόστος υπολογισμού πχ σε κάποιον υπολογιστικά ακριβό αλγόριθμο στην συνέχεια της ροής του προγράμματος μας όσο και στους απαραίτητους πόρους μνήμης που απαιτούνται για την διαχείριση των δεδομένων.

Στην παραπάνω διαδικασία θεωρήσαμε ότι έχουμε ήδη εφαρμόσει τον αλγόριθμο LLE στο σετ δεδομένων εκπαίδευσης και έτσι έχουμε το αποτέλεσμα, δηλαδή τα δεδομένα στον χώρο μειωμένης διάστασης στον πίνακα Y. Το βήμα αυτό, παρ΄ότι είναι σαφώς οικονομικότερο απο την περίπτωση στην οποία θα εφαρμόζαμε στον αλγόριθμο στο σύνολο των δεδομένων εκπαίδευσης αλλά και των δεδομένων αξιολόγησης, στις περισσότερες εφαρμογές απαιτεί πολύ μεγάλο μέγεθος μνήμης γεγονός που καθιστά τις περισσότερες φορές αδύνατη την εκτέλεση του αλγορίθμου. Για να διευκρινιστεί η διαδικασία εφαρμογής της μεθόδου αλλά και οι διαστάσεις των πινάκων κάθε βήματος δίνεται το παρακάτω διάγραμμα

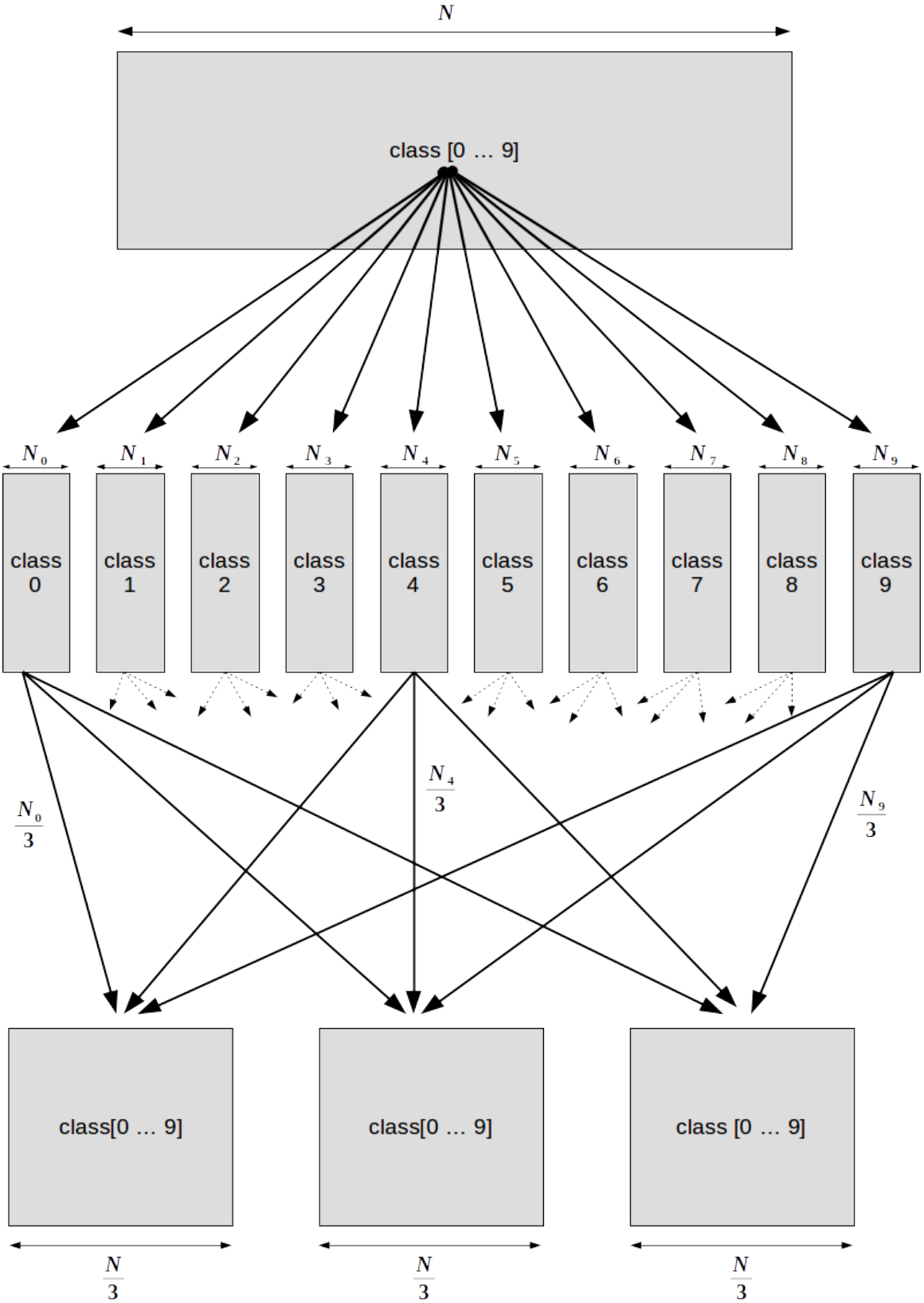



Σχήμα 5.1: Μέθοδος-1: Προβολή των δεδομένων αξιολόγησης στον χώρο των δεδομένων εκπαίδευσης.

### 5.1.2 Μέθοδος-2: Δημιουργία υποχώρων και πλειοψηφική απόφαση ταξινόμησης

Για την αντιμετώπιση λοιπόν του παραπάνω προβλήματος μπορούμε να εφαρμόσουμε την Μέθοδο-2. Η βασική ιδέα της μεθόδου αυτής, όπως θα δούμε αναλυτικά και στον ψευδοκώδικα παρακάτω, είναι να διασπάσει το αρχικό σετ δεδομένων σε υποσύνολα απο τα οποία στην συνέχεια συνδυάζει την πληροφορία που δίνει το καθένα και εξάγει το τελικό αποτέλεσμα ταξινόμησης. Σημαντικό σημείο στην διαδικασία αυτή είναι η κατασκευή των υποσυνόλων να γίνει με τρόπο τέτοιο ώστε το καθένα απο αυτά να περιέχει την ίδια ποσότητα πληροφορίας, με την έννοια ότι θα πρέπει ο διαμοιρασμός των δειγμάτων κάθε κλάσης να γίνει ομοιόμορφα σε όλα τα υποσύνολα. Με τον τρόπο αυτό στην πραγματικότητα επιλύονται πολλά μικρά υποπροβλήματα όμοια με το αρχικό. Υποπροβλήματα δηλαδή τα οποία περιέχουν την ίδια πληροφορία με το αρχικό σετ δεδομένων εκπαίδευσης αλλά σε μικρότερη ποσότητα. Αν προσέξουμε ώστε το κάθε υποσύνολο να περιέχει αρκετά δείγματα ώστε να μπορέσει να διατηρηθεί το λείο της πολλαπλότητας το οποίο είναι απαίτηση του αλγορίθμου LLE[1] τότε το αποτέλεσμα της λύσης κάποιου υποχώρου θα είναι πολύ κοντά σε αυτό του αρχικού προβλήματος. Συνδυάζοντας την πληροφορία των υποχώρων στη συνέχεια, και καταλήγοντας στο αποτέλεσμα της ταξινόμησης ανάλογα με την πλειοψηφία των αποτελεσμάτων όλων των υποχώρων το κέρδος είναι διπλό. Μειώνεται καταρχήν δραματικά το κόστος υπολογισμού του αλγορίθμου LLE[1] λόγω της μείωσης κατά μεγάλο βαθμό των δειγμάτων στα οποία εφαρμόζεται. Επίσης με την διαδικασία του ψηφίσματος και της πλειοψηφικής τελικής επιλογής βελτιώνεται κατά πολύ το αποτέλεσμα της ταξινόμησης σε σχέση με αυτό του κάθε υποχώρου ξεχωριστά.

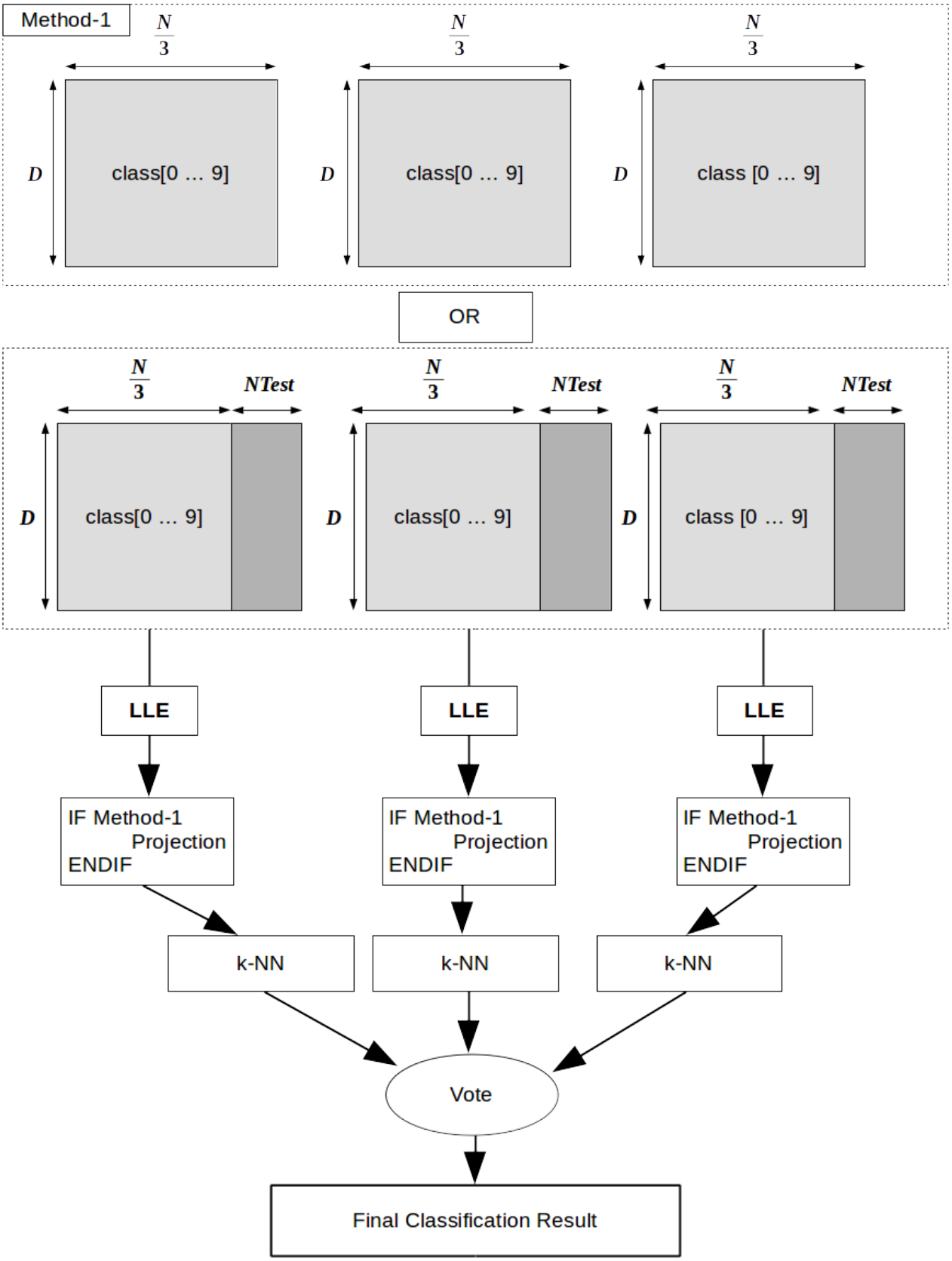
Για να γίνει κατανοητός ο τρόπος κατασκευής των υποσυνόλων αλλά και της συνολικής διαδικασίας της μεθόδου δίνεται ένα γράφημα το οποίο αναπαριστά τον διαμοιρασμό των δειγμάτων και έπειτα ο συνολικός ψευδοκώδικας της μεθόδου. Στο παρακάτω γράφημα έστω ότι το αρχικό μου σύνολο δεδομένων είναι το γνωστό σύνολο δεδομένων MNIST[17] με μέγεθος N και το οποίο περιέχει δεδομένα τα οποία ανήκουν σε δέκα κλάσεις (ψηφία απο το 0 εως το 9). Επίσης κάθε εικόνα έχει γίνει μετατροπή σε ένα διάνυσμα-στήλη μεγέθους $[Width \times Height]$, έστω $D$. Τέλος το συγκεκριμένο παράδειγμα επιλέγουμε να το χωρίσουμε σε 3 υποσύνολα. Η διαδικασία διαμοιρασμού των δειγμάτων φαίνεται γραφικά παρακάτω




Σχήμα 5.2: Μέθοδος-2.1: Δημιουργία των υποσυνόλων.

Όπως φαίνεται αναλυτικά στο παραπάνω σχήμα το σύνολο των αρχικών κλάσεων ομαδοποιείται και στην συνέχεια μοιράζονται ανάλογα τα δείγματα κάθε κλάσης, ομοιόμορφα, σε όσα υποσύνολα έχουμε επιλέξει. Απο το σημείο αυτό λοιπόν μπορούμε πλέον να εφαρμόσουμε τον αλγόριθμο μείωσης διστάσεων, LLE[1], σε κάθε ένα απο τα τελικά υποσύνολα δεδομένων καταλήγοντας σε τρείς νέους χώρους μειωμένης διάστασης d. Να διευκρινιστεί στο σημείο αυτό ότι η διαδικασία προβολής των δεδομένων αξιολόγησης μπορεί να γίνει είτε ενσωματώνοντάς τα σε κάθε ένα απο τα τρία τελικά σετ δεδομένων πρίν την εφαρμογή του αλγορίθμου ή να εφαρμοστεί η Μέθοδος-1. Σύμφωνα με την Μέθοδο-1, όπως εξηγήσαμε και παραπάνω, θα γίνει μείωση των διαστάσεων για κάθε υποσύνολο και στην συνέχεια για κάθε ένα χωριστά θα γίνει η προβολή των δεδομένων αξιολόγησης στον χώρο μειωμένης διάστασης του καθενός.

Αφού εφαρμοστεί μια απο τις παραπάνω μεθόδους, ανεξαρτήτως ποια, μπορούμε πλέον για κάθε ένα σύνολο δεδομένων (τελικά σετ εκπαίδευσης ένα εως τρία και σετ αξιολόγησης στον χώρο μειωμένης διάστασης $d$) να εφαρμόσουμε τον αλγόριθμο κοντινότερων γειτόνων (k-NN) και να κάνουμε την ταξινόμηση κάθε δείγματος του σετ αξιολόγησης, στην κλάση εκτίμησης για κάθε έναν απο τους τελικούς υποχώρους. Τέλος, πλειοψηφικά αποφασίζουμε σε ποιά κατηγορία ανήκει το κάθε δείγμα, λαμβάνοντας υπόψιν την ψήφο ως προς την κλάση ταξινόμησης του δείγματος απο τους τρείς υποχώρους. Με τον τρόπο αυτό, όπως θα φανεί και στα πειράματα παρακάτω, βελτιώνεται σε πολύ μεγάλο βαθμό το τελικό αποτέλεσμα της ταξινόμησης σε σχέση με αυτό των τριών υποχώρων. Η διαδικασία αυτή, παρουσιάζεται και γραφικά στο παρακάτω σχήμα




Σχήμα 5.3: Μέθοδος-2.2: Μείωση των διαστάσεων στα υποσύνολα και πλειοψηφική απόφαση της τελικής ταξινόμησης.

# Κεφάλαιο 6

# Εφαρμογή του αλγορίθμου LLE και των δύο μεθόδων σε πραγματικά σετ δεδομένων

## 6.1 Στόχος των πειραμάτων

Στην εργασία αυτή δόθηκε έμφαση στην μελέτη του αλγορίθμου με συγκεκριμένα σετ δεδομένων και παράλληλα διερευνήθηκε σε μεγάλο βαθμό το πως επηρεάζουν την συμπεριφορά του οι παράμετροι που δέχεται ως είσοδο απο τον χρήστη. Αυτές είναι ο αριθμός των κοντινότερων γειτόνων (k) για το πρώτο βήμα του αλγορίθμου και ο αριθμός των τελικών διστάσεων (d) για το τελικό βήμα του. Επίσης, γίνεται σύγκριση στην απόδοση του αλγορίθμου ταξινόμησης κοντινότερων γειτόνων (k-NN) για τον χώρο αρχικών διαστάσεων D και αυτού των τελικών d. Τελικός στόχος λοιπόν έπειτα απο την εκτέλεση των πειραμάτων είναι να καταλήξουμε στο συμπέρασμα κατά πόσο η μείωση των διαστάσεων με χρήση του αλγορίθμου LLE μπορεί να συμβάλει θετικά στην βελτίωση του ποσοστού ταξινόμησης σε εφαρμογές Μηχανιής Μάθησης.

## 6.2 Πειράματα και Αποτελέσματα

### 6.2.1 Πειράματα - MNIST

**MNIST:** Το πρώτο σετ δεδομένων το οποίο χρησιμοποιήθηκε είναι το πολύ γνωστό και ευρέως χρησιμοποιούμενο σετ δεδομένων στον χώρο της αναγνώρισης προτύπων, MNIST[17]. Το σετ αυτό αποτελείται απο 70.000 εικόνες, διάστασης $[28 \times 28]$ pixel, οι οποίες περιέχουν χειρόγραφα ψηφία. Οι 60.000 απο αυτές ανήκουν στο σετ εκπαίδευσης και οι 10.000 στο σετ αξιολόγησης. Για την είσοδο των δεδομένων στον αλγόριθμο, εφαρμόστηκε η λεξικογραφική διάταξη σε κάθε μια απο τις εικόνες, καταλήγοντας σε ένα διάνυσμα διάστασης $D = 784$. Τέλος, να διευκρινιστεί ότι το τελικό αποτέλεσμα ταξινόμησης, για τα σετ δεδομένων με τα ψηφία είναι το μέσο σφάλμα ταξινόμησης δηλαδή το άθροισμα του σφάλματος κάθε κλάσης προς τον συνολικό αριθμό των κλάσεων. Η επιλογή αυτή έγινε, διότι η μετρική αυτή δίνει έναν πολύ πιο ακριβές και γενικευμένο αποτέλεσμα ως προς την απόδοση του αλγορίθμου.

Στο πρώτο πείραμα με αυτό το σετ δεδομένων διερευνήθηκε αρχικά η συμπεριφορά του αλγορίθμου ως προς την απόδοση του αποτελέσματος ταξινόμησης μετά την μείωση των διαστάσεων. Οι παράμετροι οι οποίες αξιολογήθηκαν είναι ο αριθμός K των κοντινότερων γειτόνων, ο αριθμός των τελικών διαστάσεων d αλλά και ο αριθμός των υποσυνόλων της Μεθόδου-2. Συγκεκριμένα για τον αριθμό κοντινότερων γειτόνων του πρώτου βήματος του αλγορίθμου δόθηκαν οι τιμές $K = 6, 7, 8, 9, 10, 12, 16, 20, 24, 32, 64,$ για τον αριθμό των τελικών διαστάσεων του τελευταίου βήματος οι τιμές $d = 10, 16, 20, 24, 32, 40, 52, 64, 96, 128, 256$ και για τον αριθμό των δειγμάτων των υποσυνόλων οι τιμές $batch\_size = 10.000, 20.000, 60.000$ .Δηλαδή χωρίσαμε το σετ δεδομένων εκπαίδευσης των 60.000 εικόνων σε 6,3,1 υποσύνολα αντίστοιχα. Για το πείραμα αυτό δεν χρησιμοποιήσαμε την Μέθοδο-1, δηλαδή σε κάθε υποσύνολο κάθε φορά ενσωματώθηκαν τα δεδομένα αξιολόγησης στα δεδομένα εκπαίδευσης και στην συνέχεια έγινε εφαρμογή του αλγόριθμου για την μείωση των διαστάσεων στο σύνολο των δεδομένων αυτών. Για την εξαγωγή του τελικού αποτελέσματος εφαρμόστηκε ο αλγόριθμος k-NN με $k = 2$. Το μικρότερο ποσοστό σφάλματος ταξινόμησης δόθηκε για τις παραμέτρους $\mathbf{K = 12, d = 128, batch\_size = 60.000}$, με τιμή **3.06%** έναντι του **3.5%** το οποίο είναι το σφάλμα ταξινόμησης στον χώρο των αρχικών δια-

στάσεων D. Πολύ μεγάλο ενδιαφέρον παρουσιάζει το γεγονός ότι το σφάλμα για τις παραμέτρους $\mathbf{K = 8, d = 10, batch\_size = 60.000}$ ισούτε με $\mathbf{3.31\%}$ το οποίο είναι και αυτό μικρότερο απο την ταξινόμηση πρίν την μείωση διαστάσεων. Αξίζει να δοθεί έμφαση στην συγκεκριμένη αυτή περίπτωση διότι έχουμε καλύτερο ποσοστό ταξινόμησης έπειτα απο δραματική μείωση των διαστάσεων, αφού απο τις 784 αρχικές επιλέγουμε τελικά να κρατήσουμε 10 τελικές.

Παρατηρώντας τα αποτελέσματα των παρακάτω πινάκων μπορεί να παρατηρήσει κανείς ότι για την περίπτωση όπου έχουμε 64 ή περισσότερες τελικές διαστάσεις το τελικό ποσοστό σφάλματος είναι μικρότερο ή οριακά ίσο με αυτό του χώρου των αρχικών διαστάσεων. Το ίδιο ισχύει και για την περίπτωση στην οποία έχουμε $batch\_size = 20.000$. Το γεγονός αυτό μας δίνει ένα πολύ μεγάλο πλεονέκτημα διότι χωρίζοντας το σετ δεδομένων εκπαίδευσης των 60.000 δειγμάτων, σε 3 υποσύνολα έχουμε τεράστια μείωση στο κόστος των υπολογισμών αλλά και στην διαθέσιμη μνήμη η οποία απαιτείται για την εκτέλεση του αλγορίθμου. Ακόμα πιο ενδιαφέρον είναι το αποτέλεσμα του πειράματος με παραμέτρους $\mathbf{K = 10, d = 128, batch\_size = 10.000}$ για το οποίο έχουμε σφάλμα ταξινόμησης $\mathbf{3.31\%}$, αποτέλεσμα μικρότερο απο αυτό των αρχικών διαστάσεων και μάλιστα πολύ οικονομικότερο στον χρόνο υπολογισμού διότι στην περίπτωση αυτή έχουμε χωρίσει το σετ εκπαίδευσης σε **6** μικρούς σχετικά υποχώρους οι οποίοι μειώνουν την πολυπλοκότητα επίλυσης του προβλήματος κατά πολλές τάξεις μεγέθους. Οι παρακάτω πίνακες δείχνουν τα αποτέλεσμα του σφάλματος ταξινόμησης για όλους τους συνδυασμούς των παραμέτρων, και φανερώνουν σημαντικά στοιχεία για την αποτελεσματικότητα της μείωσης των διαστάσεων. Με έντονη γραμματοσειρά είναι ποσοστά σφάλματος μικρότερα απο το $\mathbf{3.5\%}$ των αρχικών D διαστάσεων.

Πίνακας 6.1: Μέσο (%) σφάλμα ταξινόμησης του σετ δεδομένων MNIST με τον αλγόριθμο κοντινότερων γειτόνων (Μέθοδος-2: 6 υποσύνολα)

| | K=6 | K=7 | K=8 | K=9 | K=10 | K=12 | K=16 | K=20 | K=24 | K=32 | K=64 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| d=10 | 4.07 | 4.18 | 4.20 | 4.48 | 4.60 | 4.69 | 5.82 | 7.35 | 9.87 | 13.2 | 22.83 |
| d=16 | 3.91 | 3.98 | 4.06 | 4.22 | 4.20 | 4.36 | 5.29 | 5.92 | 7.81 | 10.45 | 17.10 |
| d=20 | 4.03 | 3.99 | 4.16 | 4.15 | 4.14 | 4.44 | 4.96 | 5.69 | 6.95 | 9.31 | 15.55 |
| d=24 | 4.13 | 4.05 | 4.10 | 4.14 | 4.22 | 4.13 | 4.77 | 5.40 | 6.35 | 8.32 | 14.12 |
| d=32 | 3.92 | 3.91 | 4.03 | 3.98 | 4.07 | 4.18 | 4.27 | 5.07 | 5.78 | 7.37 | 12.36 |
| d=40 | 4.00 | 3.82 | 3.90 | 3.99 | 3.97 | 4.03 | 4.15 | 4.72 | 5.22 | 6.41 | 11.93 |
| d=52 | 3.93 | 3.78 | 3.97 | 3.83 | 3.93 | 3.97 | 4.17 | 4.37 | 4.89 | 6.30 | 10.74 |
| d=64 | 3.95 | 3.76 | 3.77 | 3.87 | 3.80 | 3.87 | 3.81 | 4.14 | 4.61 | 6.05 | 10.28 |
| d=96 | 4.02 | 3.6 | 3.7 | 3.68 | 3.68 | 3.72 | 3.63 | 3.90 | 4.22 | 5.51 | 10.06 |
| d=128 | 3.89 | 3.67 | 3.59 | **3.51** | **3.31** | 3.66 | 3.60 | 3.98 | 4.43 | 5.29 | 10.01 |
| d=256 | 3.77 | 3.64 | **3.51** | **3.33** | **3.47** | **3.34** | 3.56 | 4.06 | 4.61 | 5.40 | 9.43 |

Πίνακας 6.2: Μέσο (%) σφάλμα ταξινόμησης του σετ δεδομένων MNIST με τον αλγόριθμο κοντινότερων γειτόνων (Μέθοδος-2: 3 υποσύνολα)

| | K=6 | K=7 | K=8 | K=9 | K=10 | K=12 | K=16 | K=20 | K=24 | K=32 | K=64 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| d=10 | 3.90 | 3.75 | 3.92 | 4.07 | 3.97 | 4.60 | 5.49 | 6.64 | 10.66 | 14.99 | 31.58 |
| d=16 | 3.88 | 3.80 | 3.75 | 4.00 | 3.87 | 4.19 | 4.86 | 5.63 | 7.55 | 10.45 | 20.65 |
| d=20 | 3.81 | 3.82 | 3.84 | 3.89 | 3.95 | 4.01 | 4.61 | 5.40 | 6.92 | 9.88 | 17.65 |
| d=24 | 3.73 | 3.73 | 3.81 | 3.87 | 3.87 | 3.90 | 4.47 | 5.29 | 6.19 | 9.28 | 15.88 |
| d=32 | 3.78 | 3.63 | 3.72 | 3.65 | 3.82 | 3.79 | 4.26 | 4.94 | 5.53 | 8.23 | 13.69 |
| d=40 | 3.73 | 3.70 | 3.73 | 3.71 | 3.71 | 3.75 | 4.09 | 4.51 | 5.57 | 6.90 | 12.43 |
| d=52 | 3.77 | 3.73 | 3.61 | 3.66 | 3.61 | 3.68 | 4.05 | 4.30 | 4.85 | 6.36 | 11.09 |
| d=64 | 3.73 | 3.65 | 3.66 | 3.57 | **3.52** | 3.76 | 3.81 | 4.21 | 4.79 | 5.91 | 10.56 |
| d=96 | 3.65 | 3.64 | **3.51** | 3.56 | **3.49** | **3.41** | 3.63 | 3.92 | 4.23 | 5.25 | 10.02 |
| d=128 | 3.81 | **3.53** | **3.52** | **3.48** | **3.47** | **3.35** | 3.71 | 3.88 | 4.22 | 5.21 | 9.48 |
| d=256 | 3.57 | **3.46** | **3.39** | **3.30** | **3.25** | **3.37** | **3.27** | 3.66 | 4.32 | 5.47 | 8.73 |

Πίνακας 6.3: Μέσο (%) σφάλμα ταξινόμησης του σετ δεδομένων MNIST με τον αλγόριθμο κοντινότερων γειτόνων (Χωρίς υποσύνολα)

| | K=6 | K=7 | K=8 | K=9 | K=10 | K=12 | K=16 | K=20 | K=24 | K=32 | K=64 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| d=10 | **3.56** | **3.35** | **3.31** | **3.53** | 3.63 | 4.10 | 4.48 | 5.59 | 9.41 | 16.72 | 37.48 |
| d=16 | **3.48** | **3.35** | **3.44** | **3.42** | **3.40** | 3.67 | 4.10 | 5.00 | 7.07 | 11.39 | 25.14 |
| d=20 | **3.41** | **3.31** | **3.40** | **3.40** | **3.41** | **3.53** | 4.08 | 4.56 | 6.57 | 9.63 | 22.04 |
| d=32 | **3.45** | **3.39** | **3.42** | **3.51** | **3.35** | **3.30** | 3.90 | 4.55 | 6.18 | 9.35 | 19.53 |
| d=24 | 3.62 | **3.33** | **3.47** | **3.45** | **3.26** | **3.40** | **3.51** | 4.24 | 5.36 | 8.45 | 16.59 |
| d=40 | **3.52** | **3.34** | **3.54** | **3.46** | **3.37** | **3.37** | **3.48** | 4.18 | 5.14 | 7.91 | 14.58 |
| d=52 | **3.43** | **3.13** | **3.41** | **3.34** | **3.35** | **3.44** | **3.44** | **3.64** | 4.96 | 7.15 | 12.27 |
| d=64 | **3.52** | **3.20** | **3.36** | **3.35** | **3.34** | **3.43** | **3.46** | 3.68 | 4.89 | 6.52 | 11.06 |
| d=96 | **3.24** | **3.10** | **3.26** | **3.21** | **3.22** | **3.33** | **3.35** | 3.61 | 4.34 | 6.05 | 10.37 |
| d=128 | **3.19** | **3.25** | **3.11** | **3.30** | **3.12** | **3.06** | **3.34** | **3.55** | 4.06 | 5.84 | 10.18 |
| d=256 | **3.18** | **3.34** | **3.22** | **3.21** | **3.18** | **3.14** | **3.17** | 3.62 | 3.88 | 5.72 | 9.51 |

Απο την στιγμή που απο τα παραπάνω αποτελέσματα επιβεβαιώθηκε το γεγονός ότι μπορούμε να πάρουμε καλύτερο αποτέλεσμα ταξινόμησης εφαρμόζοντας την μέθοδο της διάσπασης του σετ εκπαίδευσης σε υποσύνολα, και μάλιστα με την διαδικασία της ταξινόμησης να είναι πολυ οικονομικότερη αλλά και γρηγορότερη, εστιάσαμε στον τρόπο με τον οποίο γίνεται η επιλογή των δεδομένων με σκοπό την δημιουργία των τελικών υποσυνόλων. Σκεφτήκαμε λοιπόν την περίπτωση για την οποία θα μπορούσε να δημιουργηθεί ένας υποχώρος, ο οποίος να περιέχει την χρήσιμη πληροφορία απο ολόκληρο το σετ εκπαίδευσης. Δηλαδή, σύμφωνα με την παραπάνω μέθοδο απο όλα τα τελικά υποσύνολα. Ο τρόπος με τον οποίο προσπαθήσαμε να οδηγηθούμε σε αυτό το αποτέλεσμα είναι η εφαρμογή αλγορίθμων ομαδοποίησης των δεδομένων. Συγκεκριμένα εφαρμόστηκε ο αλγόριθμος K-means[18] επιλέγοντας σαν τελικά αντιπροσωπευτικά σημεία για το τελικό σύνολο δεδομένων εκπαίδευσης, το αποτέλεσμα του αλγορίθμου το οποίο είναι τα κεντροειδή σημεία τα οποία αντιπροσωπεύουν τις επιμέρους ομάδες. Δοκιμάστηκαν διαφορετικές τιμές για το σύνολο των τελικών κεντροειδών, καταλήγοντας στο βέλτιστο αποτέλεσμα για την περίπτωση στην οποία έχουμε $\mathbf{K = 9, d = 128, clustSize = 20.000}$. Το σφάλμα της ταξινόμησης για την περίπτωση αυτή είναι $\mathbf{3.28\%}$, μικρότερο απο αυτό των αρχικών διαστάσεων $\mathbf{(3.5\%)}$. Συγκεκριμένες τιμές για τις παραμέτρους είναι $\mathbf{K = 8, 9, 10, 12, 16, 20}$, d όπως και στο προηγούμενο ερώτημα και $\mathbf{clustSize = 5.000, 10.000, 15.000, 20.000}$. Το σύνολο των αποτελεσμάτων φαίνεται στους παρακάτω πίνακες

Πίνακας 6.4: Μέσο (%) σφάλμα ταξινόμησης του σετ δεδομένων MNIST με τον αλγόριθμο κοντινότερων γειτόνων και δειγματοληψία με τον αλγόριθμο K-means με 5.000 κεντροειδή.

| | K=8 | K=9 | K=10 | K=12 | K=16 | K=20 |
|---|---|---|---|---|---|---|
| d=10 | 5.91 | 6.75 | 7.28 | 8.78 | 10.35 | 16.21 |
| d=16 | 5.31 | 6.15 | 6.44 | 7.54 | 9.25 | 11.56 |
| d=20 | 5.16 | 5.25 | 5.31 | 6.37 | 8.66 | 10.85 |
| d=24 | 5.11 | 5.08 | 5.13 | 6.09 | 7.97 | 10.13 |
| d=32 | 5.01 | 5.10 | 5.18 | 5.86 | 6.75 | 9.18 |
| d=40 | 4.90 | 5.00 | 4.81 | 5.35 | 6.50 | 8.32 |
| d=52 | 5.08 | 5.02 | 5.00 | 5.53 | 6.48 | 7.51 |
| d=64 | 4.90 | 4.99 | 5.02 | 5.61 | 6.08 | 6.97 |
| d=96 | 4.76 | 4.82 | 5.05 | 5.29 | 5.87 | 6.42 |
| d=128 | 4.79 | 4.96 | 5.02 | 5.22 | 5.70 | 5.77 |
| d=164 | 4.88 | 4.98 | 4.97 | 5.38 | 5.41 | 5.47 |
| d=196 | 5.08 | 5.09 | 5.11 | 5.42 | 5.39 | 5.32 |
| d=256 | 5.33 | 5.33 | 5.30 | 5.17 | 5.45 | 5.18 |

Πίνακας 6.5: Μέσο (%) σφάλμα ταξινόμησης του σετ δεδομένων MNIST με τον αλγόριθμο κοντινότερων γειτόνων και δειγματοληψία με τον αλγόριθμο K-means με 10.000 κεντροειδή.

| | K=8 | K=9 | K=10 | K=12 | K=16 | K=20 |
|---|---|---|---|---|---|---|
| d=10 | 4.98 | 5.87 | 6.18 | 7.32 | 9.04 | 13.17 |
| d=16 | 4.69 | 4.89 | 5.03 | 6.49 | 7.66 | 11.22 |
| d=20 | 4.37 | 4.70 | 4.83 | 5.88 | 6.82 | 9.74 |
| d=24 | 4.33 | 4.62 | 4.67 | 5.49 | 6.50 | 8.30 |
| d=32 | 4.38 | 4.53 | 4.72 | 5.20 | 6.08 | 7.55 |
| d=40 | 4.35 | 4.35 | 4.87 | 5.13 | 5.76 | 6.70 |
| d=52 | 4.31 | 4.18 | 4.52 | 5.26 | 5.78 | 6.61 |
| d=64 | 4.31 | 4.42 | 4.60 | 5.06 | 5.43 | 6.44 |
| d=96 | 4.13 | 4.23 | 4.30 | 4.88 | 5.14 | 5.67 |
| d=128 | 4.07 | 4.06 | 4.32 | 4.68 | 5.01 | 5.64 |
| d=164 | 3.98 | 4.13 | 4.37 | 4.67 | 5.08 | 5.09 |
| d=196 | 4.19 | 4.38 | 4.39 | 4.65 | 4.85 | 5.17 |
| d=256 | 4.37 | 4.48 | 4.54 | 4.65 | 4.90 | 4.82 |

Πίνακας 6.6: Μέσο (%) σφάλμα ταξινόμησης του σετ δεδομένων MNIST με τον αλγόριθμο κοντινότερων γειτόνων και δειγματοληψία με τον αλγόριθμο K-means με 15.000 κεντροειδή.

| | K=8 | K=9 | K=10 | K=12 | K=16 | K=20 |
|---|---|---|---|---|---|---|
| d=10 | 4.61 | 4.86 | 5.54 | 6.35 | 8.23 | 12.80 |
| d=16 | 4.09 | 4.27 | 4.72 | 5.18 | 7.38 | 9.52 |
| d=20 | 3.94 | 4.21 | 4.47 | 5.12 | 6.91 | 8.38 |
| d=24 | 3.87 | 4.05 | 4.39 | 4.86 | 6.48 | 7.91 |
| d=32 | 4.05 | 4.07 | 4.22 | 4.65 | 5.32 | 7.49 |
| d=40 | 3.72 | 3.88 | 4.11 | 4.43 | 4.94 | 6.75 |
| d=52 | 3.78 | 3.97 | 4.00 | 4.30 | 4.74 | 6.2 |
| d=64 | 3.79 | 3.94 | 4.07 | 4.38 | 4.73 | 5.81 |
| d=96 | 3.76 | 3.81 | 4.14 | 4.50 | 4.64 | 5.45 |
| d=128 | 3.74 | 3.96 | 4.25 | 4.34 | 4.67 | 5.08 |
| d=164 | 3.96 | 4.18 | 4.08 | 4.39 | 4.76 | 4.69 |
| d=196 | 4.03 | 4.18 | 4.04 | 4.30 | 4.73 | 4.72 |
| d=256 | 3.98 | 4.18 | 4.04 | 4.42 | 4.93 | 4.52 |

Πίνακας 6.7: Μέσο (%) σφάλμα ταξινόμησης του σετ δεδομένων MNIST με τον αλγόριθμο κοντινότερων γειτόνων και δειγματοληψία με τον αλγόριθμο K-means με 20.000 κεντροειδή.

| | K=8 | K=9 | K=10 | K=12 | K=16 | K=20 |
|---|---|---|---|---|---|---|
| d=10 | 4.29 | 4.52 | 4.83 | 5.65 | 6.97 | 10.78 |
| d=16 | 3.82 | 4.10 | 4.18 | 4.99 | 6.01 | 7.69 |
| d=20 | 3.80 | 4.00 | 4.02 | 4.70 | 5.59 | 7.58 |
| d=24 | 3.82 | 3.96 | 4.03 | 4.45 | 5.01 | 6.78 |
| d=32 | 3.91 | 3.92 | 4.14 | 4.34 | 4.62 | 6.02 |
| d=40 | 3.74 | 3.75 | 4.03 | 4.42 | 4.57 | 5.67 |
| d=52 | 3.73 | 3.64 | 3.88 | 4.23 | 4.54 | 5.21 |
| d=64 | 3.69 | 3.58 | 3.83 | 4.17 | 4.30 | 5.01 |
| d=96 | **3.54** | 3.60 | 3.58 | 3.92 | 4.31 | 4.85 |
| d=128 | **3.42** | **3.28** | **3.53** | 3.88 | 4.18 | 4.88 |
| d=164 | **3.44** | **3.32** | **3.41** | 3.79 | 4.57 | 4.74 |
| d=196 | **3.31** | **3.40** | **3.55** | 3.74 | 4.39 | 4.62 |
| d=256 | 3.60 | **3.47** | 3.58 | 3.71 | 4.20 | 4.43 |

Στον παραπάνω πίνακα ενδιαφέρον παρουσιάζει το γεγονός ότι έπειτα απο την δειγματοληψία μέσω του αλγορίθμου Kmeans[18] τα ελάχιστα σφάλματα ταξινόμησης είναι σε κάθε περίπτωση πολύ κοντά μεταξύ τους. Αυτό δηλαδή σημαίνει ότι απο το αρχικό σετ δεδομένων των 60.000 εικόνων, κάναμε σημαντική μείωση του αριθμού των δεδομένων σε κάθε περίπτωση αλλά παρ΄όλα αυτά είχαμε ελάχιστη εως και μηδενική απώλεια πληροφορίας. Η εξήγηση σε αυτό, δίνεται απο το γεγονός ότι

το σετ δεδομένων MNIST[17] είναι στην ουσία συνθετικό σετ. δηλαδή ένας σχετικά μικρός αριθμός του συνόλου των δεδομένων είναι μοναδικά και τα υπόλοιπα προκύπτουν απο ομογενείς μετασχηματισμούς ή παραμορφώσεις αυτών. Ακόμα, επιβεβαιώνει το γεγονός ότι ο αλγόριθμος LLE δεν απαιτεί έναν μεγάλο αριθμό δειγμάτων εκπαίδευσης όπως για παράδειγμα τα Νευρωνικά δίκτυα, αλλά αρκεί ένας ομοιόμορφα δειγματοληπτημένος χώρος των αρχικών δεδομένων ο οποίος να διατηρεί το λείο της πολλαπλότητας.

Στο επόμενο πείραμα με το συγκεκριμένο σετ εφαρμόστηκε η Μέθοδος-1, δηλαδή έγινε προβολή των δεδομένων αξιολόγησης στον χώρο μειωμένης διάστασης μέσω του πίνακα γειτνίασης στον χώρο των αρχικών διαστάσεων. Παρατηρώντας τους παρακάτω πίνακες βλέπουμε ότι το ελάχιστο σφάλμα ταξινόμησης στον χώρο των τελικών διαστάσεων ισούτε με **3.85% (K=8, d=256, batch_size=60.000)**, μεγαλύτερο δηλαδή απο αυτό των αρχικών διαστάσεων **(3.5%)** οπότε θα μπορούσαμε να καταλήξουμε στο συμπέρασμα ότι η διαδικασία της μεθόδου δεν μας βοηθά στην βελτίωση του αποτελέσματος. Παρ΄ όλα αυτά αποτελεί έναν πολύ πρακτικό και γρήγορο τρόπο, σε σχέση με την εφαρμογή του αλγορίθμου στο σύνολο των δειγμάτων εκπαίδευσης αλλά και αξιολόγησης, ώστε να καταφέρουμε να μειώσουμε τις διαστάσεις των τελευταίων. Αυτό είναι χρήσιμο σε περιπτώσεις στις οποίες μπορούμε νε ανεχθούμε την μικρή αυτή σχετικά διαφορά σφάλματος, και στις οποίες χρησιμοποιούμε τα δεδομένα αξιολόγησης σε πολλά επόμενα βήματα εκτελώντας υπολογιστικά απαιτητικούς αλγορίθμους. Το αποτέλεσμα εφαρμογής της μεθόδου για το σύνολο των παραμέτρων φαίνεται στους παρακάτω πίνακες

Πίνακας 6.8: Μέσο (%) σφάλμα ταξινόμησης του σετ δεδομένων MNIST με τον αλγόριθμο κοντινότερων γειτόνων (Χρήση της Μεθόδου-1 και της Μεθόδου-2 με 6 υποσύνολα)

| | K=8 | K=9 | K=10 | K=12 | K=16 | K=20 |
|---|---|---|---|---|---|---|
| d=10 | 15.63 | 15.88 | 16.84 | 18.19 | 21.49 | 21.99 |
| d=16 | 9.97 | 10.06 | 10.53 | 10.79 | 14.76 | 17.78 |
| d=20 | 8.14 | 8.29 | 9.04 | 9.86 | 11.68 | 14.75 |
| d=24 | 6.02 | 6.44 | 7.00 | 8.74 | 10.26 | 12.41 |
| d=32 | 5.99 | 6.09 | 6.40 | 6.81 | 7.95 | 9.98 |
| d=40 | 5.72 | 5.59 | 5.85 | 6.01 | 7.10 | 7.84 |
| d=52 | 5.60 | 5.52 | 5.54 | 5.81 | 5.99 | 7.34 |
| d=64 | 5.42 | 5.37 | 5.49 | 5.53 | 5.81 | 6.26 |
| d=96 | 5.41 | 5.35 | 5.52 | 5.58 | 5.53 | 5.57 |
| d=128 | 5.19 | 5.18 | 5.19 | 5.31 | 5.32 | 5.43 |
| d=164 | 5.12 | 5.16 | 5.19 | 5.17 | 5.17 | 5.24 |
| d=196 | 5.09 | 5.07 | 5.06 | 5.08 | 5.08 | 5.21 |
| d=256 | 4.98 | 5.01 | 5.03 | 5.15 | 5.07 | 5.01 |

Πίνακας 6.9: Μέσο (%) σφάλμα ταξινόμησης του σετ δεδομένων MNIST με τον αλγόριθμο κοντινότερων γειτόνων (Χρήση της Μεθόδου-1 και της Μεθόδου-2 με 3 υποσύνολα)

| | K=8 | K=9 | K=10 | K=12 | K=16 | K=20 |
|---|---|---|---|---|---|---|
| d=10 | 14.60 | 14.38 | 15.74 | 17.68 | 21.48 | 24.07 |
| d=16 | 8.88 | 9.33 | 9.67 | 9.53 | 14.00 | 17.01 |
| d=20 | 5.78 | 5.93 | 6.32 | 8.01 | 11.54 | 14.23 |
| d=24 | 5.39 | 5.37 | 5.50 | 6.40 | 9.03 | 11.07 |
| d=32 | 5.05 | 5.15 | 5.26 | 5.41 | 6.67 | 9.19 |
| d=40 | 4.97 | 5.03 | 5.04 | 5.24 | 5.62 | 7.14 |
| d=52 | 4.97 | 5.06 | 4.88 | 4.92 | 5.37 | 6.25 |
| d=64 | 4.96 | 5.02 | 4.73 | 4.84 | 5.13 | 5.46 |
| d=96 | 4.80 | 4.87 | 4.84 | 4.77 | 4.90 | 5.06 |
| d=128 | 4.63 | 4.61 | 4.75 | 4.74 | 4.81 | 4.81 |
| d=164 | 4.59 | 4.58 | 4.52 | 4.60 | 4.65 | 4.80 |
| d=196 | 4.54 | 4.54 | 4.55 | 4.55 | 4.55 | 4.75 |
| d=256 | 4.54 | 4.44 | 4.43 | 4.44 | 4.52 | 4.60 |

Πίνακας 6.10: Μέσο (%) σφάλμα ταξινόμησης του σετ δεδομένων MNIST με τον αλγόριθμο κοντινότερων γειτόνων (Χρήση της Μεθόδου-1 χωρίς υποσύνολα)

| | K=8 | K=9 | K=10 | K=12 | K=16 | K=20 |
|---|---|---|---|---|---|---|
| d=10 | 17.92 | 17.50 | 16.00 | 16.10 | 21.93 | 34.84 |
| d=16 | 7.09 | 7.52 | 8.43 | 8.57 | 12.95 | 19.03 |
| d=20 | 4.85 | 4.92 | 5.46 | 6.86 | 9.29 | 15.82 |
| d=24 | 4.70 | 4.67 | 4.61 | 4.74 | 8.49 | 12.86 |
| d=32 | 4.59 | 4.64 | 4.57 | 4.79 | 6.43 | 10.54 |
| d=40 | 4.52 | 4.47 | 4.47 | 4.58 | 4.78 | 8.90 |
| d=52 | 4.36 | 4.32 | 4.41 | 4.40 | 5.16 | 6.82 |
| d=64 | 4.36 | 4.32 | 4.32 | 4.46 | 5.05 | 5.12 |
| d=96 | 4.10 | 4.09 | 4.13 | 4.22 | 4.52 | 5.14 |
| d=128 | 4.08 | 4.08 | 4.12 | 4.05 | 4.32 | 4.52 |
| d=164 | 4.07 | 4.05 | 4.09 | 4.00 | 4.27 | 4.48 |
| d=196 | 3.92 | 4.00 | 4.07 | 4.02 | 4.24 | 4.39 |
| d=256 | 3.85 | 3.92 | 3.97 | 4.04 | 4.17 | 4.35 |

### 6.2.2 Πειράματα - SVHN

**SVHN:** Το δεύτερο σετ δεδομένων είναι το The Street View House Numbers (SVHN) Data-set[2] το οποίο περιέχει πραγματικές εικόνες απο αριθμούς σπιτιών, οι οποίες τραβήχτηκαν απο το αυτοκίνητο χαρτογράφησης της Google. Συγκεκριμένα χρησιμοποιήθηκε το Format-2 στο οποίο οι αριθμοί με περισσότερα απο ένα ψηφία έχουν διασπαστεί σε ψηφία απο το 0 εως το 9. Το σετ αυτό, είναι όμοιο με το MNIST[17] με την διαφορά ότι πρόκειται για πραγματικές εικόνες, το οποίο καθιστά την διαδικασία της αναγνώρισης κατά πολύ δυσκολότερη. Αυτο συμβαίνει λόγω φυσικών παραμορφώσεων, για παράδειγμα απο την αλλοίωση της φωτεινότητας, την παρουσία θορύβου κλπ. Το σετ περιέχει 73.257 εικόνες στο σετ εκπαίδευσης και 26032 εικόνες στο σετ αξιολόγησης, μεγέθους $[32 \times 32]$ pixel. Εφαρμόζοντας αντίστοιχα λεξικογραφική διάταξη στο σετ αυτό, καταλήγουμε να έχουμε για κάθε εικόνα ένα διάνυσμα μεγέθους $D = 1024$.

Αν για το συγκεκριμένο σετ δεδομένων εφαρμόσουμε την διαδικασία την οποία εφαρμόσαμε στο σετ δεδομένων MNIST[17] θα παρατηρήσουμε ότι τα αποτελέσματά μας έχουν τεράστιο σφάλμα ταξινόμησης. Αυτό συμβαίνει διότι, υπάρχει πολύ μεγάλο πρόβλημα απο το πρώτο κιόλας βήμα του αλγορίθμου στο οποίο εφαρμόζεται ο αλγόριθμος k-NN μεταξύ των δεδομένων αξιολόγησης και

των δεδομένων εκπαίδευσης στον χώρο των αρχικών διαστάσεων D. Το γεγονός αυτό προκύπτει απο την φύση των δεδομένων του σετ αυτού, και πιο συγκεκριμένα ο αλγόριθμος κοντινότερων γειτόνων αποτυγχάνει εξαιτίας του γεγονότος ότι πρόκειται για εικόνες απο το φυσικό περιβάλλον οι οποίες πάσχουν απο θόρυβο αλλά και παραμορφώσεις της φωτεινότητας.

**Histogram of Oriented Gradients (HoG)** [19]: Για την αντιμετώπιση της παραπάνω αδυναμίας εκτέλεσης του αλγορίθμου, θα πρέπει να του δώσουμε είσοδο η οποία να είναι μοναδική για κάθε δείγμα του σετ δεδομένων ώστε να μπορεί ο αλγόριθμος να ομαδοποιήσει δεδομένα τα οποία ανήκουν στην ίδια κλάση, αλλά και να τα ξεχωρίσει απο αυτά των υπολοίπων. Η διαδικασία αυτή είναι γνωστή στον χώρο της υπολογιστικής όρασης και της επεξεργασίας εικόνας με τον όρο εξαγωγή χαρακτηριστικών. Όπως δηλώνει και το όνομά της, μέσω της διακασίας αυτής σε κάθε εικόνα εντοπίζονται συγκεκριμένα χαρακτηριστικά σημεία τα οποία την αντιπροσωπεύουν μοναδικά. Τα πιο συνηθισμένα τέτοιου είδους χαρακτηριστικά είναι τα SIFT[20] και τα HoG[19] τα οποία και εφαρμόστηκαν στην συγκεκριμένη περίπτωση λόγω καλύτερων αποτελεσμάτων σε αυτό το σετ δεδομένων.

Η βασική αρχή λειτουργίας της μεθόδου αυτής είναι ότι ένα συγκεκριμένο αντικείμενο μέσα σε μια εικόνα μπορεί να προσδιοριστεί απο την κατανομή της φωτεινότητας πάνω στα σημεία του αλλά και απο τον εντοπισμό της κλίσης της δηλαδή τον εντοπισμό των ακμών του αντικειμένου. Ο αλγόριθμος χωρίζει την εικόνα σε μικρές περιοχές απο pixel και σε κάθε μια απο αυτές υπολογίζει την κλίση της φωτεινότητας δημιουργώντας ένα τελικό συνολικό ιστόγραμμα με τις τιμές αυτές. Το μέγεθος των περιοχών αυτών καθορίζεται απο το μέγεθος του πυρήνα ο οποίος αποτελεί παράμετρο του αλγορίθμου και καθορίζεται κατά την εκτέλεση του.

Κατά το πρώτο πείραμα λοιπόν με αυτό το σετ δεδομένων μελετήθηκε τόσο η συμπεριφορά του αλγορίθμου LLE αλλάζοντας τις παραμέτρους του όσο και η αποτελεσματικότητα των HoG[19] χαρακτηριστικών ανάλογα με το μέγεθος του πυρήνα. Πιο συγκεκριμένα για τις παραμέτρους του αλγορίθμου LLE εξετάστηκαν οι τιμές $\mathbf{K = 8, 10, 12}$ και $\mathbf{d = 16, 20, 32, 64, 96, 128, 164, 196, 256}$ και για το μέγεθος του πυρήνα του αλγορίθμου HoG[19] οι τιμές $\mathbf{kernels} = [\mathbf{2 \times 2}], [\mathbf{4 \times 4}], [\mathbf{8 \times 8}]$. Επίσης χρησιμοποιήθηκαν απο το σετ εκπαίδευσης **30.000** δείγματα, δηλαδή 3.000 εικόνες ψηφίων απο κάθε κλάση και το σύνολο των δεδομένων ταξινόμησης για την εξαγωγή του τελικού

αποτελέσματος. Απο το πείραμα αυτό λοιπόν προκύπτουν τα παρακάτω αποτελέσματα

Πίνακας 6.11: Μέσο (%) σφάλμα ταξινόμησης του σετ δεδομένων SVHN με τον αλγόριθμο κοντινότερων γειτόνων. Εξαγωγή HoG χαρακτηριστικών με μέγεθος πυρήνα $[2 \times 2]$ σε 30.000 δείγματα του συνόλου εκπαίδευσης.

| | K=8 | K=10 | K=12 |
|---|---|---|---|
| d=16 | 20.81 | 20.52 | 20.56 |
| d=20 | 20.50 | 20.12 | 19.57 |
| d=32 | 19.69 | 19.10 | 19.06 |
| d=64 | 20.23 | 19.94 | 19.79 |
| d=96 | 20.71 | 20.11 | 19.78 |
| d=128 | 21.11 | 20.56 | 20.50 |
| d=164 | 21.29 | 20.83 | 20.69 |
| d=196 | 21.81 | 20.98 | 20.66 |
| d=256 | 22.25 | 21.60 | 21.05 |

Πίνακας 6.12: Μέσο (%) σφάλμα ταξινόμησης του σετ δεδομένων SVHN με τον αλγόριθμο κοντινότερων γειτόνων. Εξαγωγή HoG χαρακτηριστικών με μέγεθος πυρήνα $[4 \times 4]$ σε 30.000 δείγματα του συνόλου εκπαίδευσης.

| | K=8 | K=10 | K=12 |
|---|---|---|---|
| d=16 | 19.24 | 19.57 | 19.79 |
| d=20 | 18.22 | 18.16 | 18.35 |
| d=32 | 17.91 | 17.63 | <u>17.42</u> |
| d=64 | 18.56 | 18.37 | 18.21 |
| d=96 | 18.90 | 18.72 | 18.73 |
| d=128 | 19.15 | 19.07 | 18.84 |
| d=164 | 19.68 | 19.33 | 19.22 |
| d=196 | 19.87 | 19.44 | 19.25 |
| d=256 | 20.30 | 19.79 | 19.58 |

Πίνακας 6.13: Μέσο (%) σφάλμα ταξινόμησης του σετ δεδομένων SVHN με τον αλγόριθμο κοντινότερων γειτόνων. Εξαγωγή HoG χαρακτηριστικών με μέγεθος πυρήνα $[8 \times 8]$ σε 30.000 δείγματα του συνόλου εκπαίδευσης.

| | K=8 | K=10 | K=12 |
|---|---|---|---|
| d=16 | 23.36 | 24.38 | 25.10 |
| d=20 | 22.71 | 23.16 | 23.80 |
| d=32 | 22.52 | 22.62 | 22.50 |
| d=64 | 22.19 | 22.30 | 22.15 |
| d=96 | 22.39 | 22.19 | 22.15 |
| d=128 | 22.42 | 22.32 | 22.18 |
| d=164 | 22.47 | 22.33 | 22.35 |
| d=196 | 22.60 | 22.58 | 22.58 |
| d=256 | 23.01 | 22.61 | 22.47 |

Απο τα παραπάνω αποτελέσματα βλέπουμε ότι το μικρότερο σφάλμα δίνεται για τις παραμέτρους $\mathbf{K = 12, d = 32}$ του LLE και $\mathbf{kernel} = [\mathbf{4 \times 4}]$ των HoG χαρακτηριστικών αντίστοιχα. Με αυτές τις βέλτιστες παραμέτρους λοιπόν περνάμε στην εκτέλεση του επόμενου πειράματος, στο οποίο θέλουμε να δούμε την αποτελεσματικότητα του αλγορίθμου LLE ως προς την ταξινόμηση των δεδομένων αξιολόγησης σχετικά με το αποτέλεσμα της ταξινόμησης χωρίς μείωση των διαστάσεων. Επίσης με το πείραμα αυτό έχουμε σκοπό να συγκρίνουμε το βέλτιστο αποτέλεσμα μας μετά απο την μείωση των διαστάσεων με τα αποτελέσματα της συγκεκριμένης δημοσίευσης Reading Digits in Natural Images with Unsupervised Feature Learning [2] η οποία αποτελεί και την πρωτότυπη δημοσίευση του συγκεκριμένου σετ δεδομένων και είναι μια συνεργασία της Google και του πανεπιστημίου του Stanford. Τρέχοντας το πείραμα λοιπόν για τις βέλτιστες παραμέτρους που βρήκαμε παραπάνω, δηλαδή $\mathbf{K = 12, d = 32}$ και $\mathbf{kernel} = [\mathbf{4 \times 4}]$, στο σύνολο των δεδομένων του SVHN τα αποτελέσματα είναι εντυπωσιακά. Πιο συγκεκριμένα, εφαρμόζοντας την εξαγωγή των χαρακτηριστικών HoG καταλήγουμε να έχουμε για κάθε εικόνα ένα διάνυσμα μεγέθους 1764 στοιχείων, απο τα οποία εφαρμόζοντας τον αλγόριθμο LLE καταλήγουμε να κρατήσουμε 32 απο αυτά. Για το μέσο τετραγωνικό σφάλμα ταξινόμησης με την χρήση του αλγορίθμου k-NN, με $k = 8$ έχουμε για τις αρχικές διαστάσεις D την τιμή $\mathbf{17.0\%}$ ενώ για τις τελικές διαστάσεις d την τιμή $\mathbf{16.67\%}$. Το γεγονός ότι μετά απο τεράστια μείωση των παραμέτρων έχουμε μια τόσο μικρή απόκλιση στο τελικό σφάλμα ταξινόμησης φανερώνει ότι η μείωση των διαστάσεων μπορεί να συμβάλει πολύ αποτελεσματικά στο κόστος των υπολογισμών κατά την διαδικασία της ταξινόμησης.

Πίνακας 6.14: Μέσο (%) σφάλμα ταξινόμησης του σετ δεδομένων SVHN με τον αλγόριθμο κοντινότερων γειτόνων. Εξαγωγή HoG χαρακτηριστικών με μέγεθος πυρήνα $[4 \times 4]$ σε ολόκληρο το σύνολο των δεδομένων εκπαίδευσης

| | LLE dim-reduction | No dim-reduction |
|---|---|---|
| d=32<br>K=12<br>k-NN - k=8 | **16.67** | 17.0 |

Όπως έγινε φανερό και απο τα παραπάνω αποτελέσματα, ο αλγόριθμος LLE είναι σε θέση να μειώσει δραματικά το κόστος υπολογισμού του βήματος της τελικής ταξινόμησης μέσω της μείωσης των διαστάσεων κατά έναν πολύ μεγάλο αριθμό. Παράλληλα όπως φάνηκε το σφάλμα μπορεί κυμαίνεται σε αμελητέα όρια γεγονός το οποίο σε πολλές πρακτικές εφαρμογές είναι επιθυμητό με στόχο να κερδίσουμε στον χρόνο υπολογισμού του τελικού αποτελέσματος αποφεύγοντας την ταξινόμηση στον χώρο των αρχικών διαστάσεων αλλά εφαρμόζοντάς την στον τελικό μειωμένο χώρο, για τον οποίο $d \ll D$. Επίσης, για την περίπτωση στην οποία θέλουμε αποτελέσματα ταξινόμησης για το σετ δεδομένων αξιολόγησης σε πραγματικό χρόνο εφαρμόζουμε την Μέθοδο-1, η οποία προυποθέτει ότι έχουμε τα δεδομένα μειωμένων διαστάσεων του σετ εκπαίδευσης, και έπειτα μπορούμε ταχύτατα στον χώρο με διαστάσεις d να εφαρμόσουμε τον αλγόριθμο ταξινόμησης k-NN έχοντας τα αποτελέσματα σε πραγματικό χρόνο. Το συγκεκριμένο πείραμα μάλιστα χρησιμοποιώντας 42.000 απο τα δεδομένα εκπαίδευσης και όλο το σετ αξιολόγησης (με τις βέλτιστες παραμέτρους που αναφέραμε παραπάνω και εφαρμόζοντας την Μέθοδο-1) έδωσε σαν αποτελέσματα τις τιμές 18.00% για τις διαστάσεις D και 18.34% για τις διαστάσεις d, σφάλμα ανεκτό αν αναλογιστεί κανείς την διαφορά στο κόστος υπολογισμού ενός προβλήματος με πολυπλοκότητα $\mathcal{O}(N^2)$ με $N \simeq 70K$ (δεδομένα εκπαίδευσης) και $N \simeq 100K$ (δεδομένα εκπαίδευσης + δεδομένα αξιολόγησης) αντίστοιχα.

Ακόμα και αυτή η διαδικασία όμως της Μεθόδου-1, όπως αναφέραμε και σε προηγούμενη παραγραφο, σχεδόν σε όλες τις πρακτικές εφαρμογές λόγω της πολυπλοκότητας $\mathcal{O}(N^2)$ του τελευταίου βήματος του αλγορίθμου είναι αδύνατο να εκτελεσθεί. Για τον λόγο αυτό, μελετήσαμε την συμπεριφορά της Μεθόδου-2 για διάφορες τιμές ως προς τον αριθμό των υποχώρων. Τα αποτελέσματα του πειράματος παρουσιάζουν πολύ μεγάλο ενδιαφέρον διότι μπορούμε να μειώσουμε δραματικά το

κόστος υπολογισμού χωρίζοντας τον αρχικό χώρο σε μικρά υποσύνολα. Επίσης στο σημείο αυτό αξίζει να παρατηρηθεί η μεγάλη μείωση του σφάλματος στο τελικό αποτέλεσμα της ταξινόμησης μετά απο την διαδικασία της πλειοψηφικής επιλογής του τελικού αποτελέσματος. Συγκεκριμένα εκτελέσαμε το πείραμα αυτό χωρίζοντας το αρχικό σετ δεδομένων σε **3,5,10** και **20 υποχώρους**. Τα αποτελέσματα φαίνονται αναλυτικά στους παρακάτω πίνακες

Πίνακας 6.15: Μέσο (%) σφάλμα ταξινόμησης του σετ δεδομένων SVHN με τον αλγόριθμο κοντινότερων γειτόνων. Εξαγωγή HoG χαρακτηριστικών με μέγεθος πυρήνα $[4 \times 4]$ σε 30.000 δείγματα του συνόλου εκπαίδευσης. Παράμετροι του αλγορίθμου LLE: K=12, d=32 και χωρισμός σε 3 υποσύνολα μέσω της Μεθόδου-2.

| Subspaces | Subspace error | Method-2 | D-dimensions |
|---|---|---|---|
| 1 | 20.19 | | |
| 2 | 19.05 | **18.30** | 18.71 |
| 3 | 19.97 | | |

Πίνακας 6.16: Μέσο (%) σφάλμα ταξινόμησης του σετ δεδομένων SVHN με τον αλγόριθμο κοντινότερων γειτόνων. Εξαγωγή HoG χαρακτηριστικών με μέγεθος πυρήνα $[4 \times 4]$ σε 30.000 δείγματα του συνόλου εκπαίδευσης. Παράμετροι του αλγορίθμου LLE: K=12, d=32 και χωρισμός σε 5 υποσύνολα μέσω της Μεθόδου-2.

| Subspaces | Subspace error | Method-2 | D-dimensions |
|---|---|---|---|
| 1 | 20.28 | | |
| 3 | 21.11 | | |
| 2 | 20.93 | **18.37** | 18.61 |
| 4 | 20.92 | | |
| 5 | 21.11 | | |

Πίνακας 6.17: Μέσο (%) σφάλμα ταξινόμησης του σετ δεδομένων SVHN με τον αλγόριθμο κοντινότερων γειτόνων. Εξαγωγή HoG χαρακτηριστικών με μέγεθος πυρήνα $[4 \times 4]$ σε 30.000 δείγματα του συνόλου εκπαίδευσης. Παράμετροι του αλγορίθμου LLE: K=12, d=32 και χωρισμός σε 10 υποσύνολα μέσω της Μεθόδου-2.

| Subspaces | Subspace error | Method-2 | D-dimensions |
|---|---|---|---|
| 1 | 22.54 | | |
| 2 | 22.51 | | |
| 3 | 22.24 | | |
| 4 | 22.33 | | |
| 5 | 21.72 | **18.58** | 18.52 |
| 6 | 22.17 | | |
| 7 | 22.71 | | |
| 8 | 22.07 | | |
| 9 | 22.21 | | |
| 10 | 22.77 | | |

Πίνακας 6.18: Μέσο (%) σφάλμα ταξινόμησης του σετ δεδομένων SVHN με τον αλγόριθμο κοντινότερων γειτόνων. Εξαγωγή HoG χαρακτηριστικών με μέγεθος πυρήνα $[4 \times 4]$ σε 30.000 δείγματα του συνόλου εκπαίδευσης. Παράμετροι του αλγορίθμου LLE: K=12, d=32 και χωρισμός σε 20 υποσύνολα μέσω της Μεθόδου-2.

| Subspaces | Subspace error | Method-2 | D-dimensions |
|---|---|---|---|
| 1 | 23.72 | | |
| 2 | 23.40 | | |
| 3 | 23.70 | | |
| 4 | 24.93 | | |
| 5 | 24.06 | | |
| 6 | 23.26 | | |
| 7 | 23.54 | | |
| 8 | 24.87 | | |
| 9 | 23.01 | | |
| 10 | 23.39 | 19.13 | 18.52 |
| 11 | 23.75 | | |
| 12 | 24.52 | | |
| 13 | 24.77 | | |
| 14 | 24.89 | | |
| 15 | 24.11 | | |
| 16 | 23.86 | | |
| 17 | 23.24 | | |
| 18 | 24.26 | | |
| 19 | 23.62 | | |
| 20 | 25.07 | | |

### 6.2.3 Πειράματα - Arcene

**Arcene:**[3] Τελευταίο σετ δεδομένων πάνω στο οποίο εφαρμόσαμε τον αλγόριθμο μη γραμμικής μείωσης διαστάσεων LLE είναι το Arcene[3]. Πρόκειται για δεδομένα προερχόμενα απο τον χώρο της Ιατρικής και συγκεκριμένα στόχος είναι να γίνει σωστή ταξινόμηση των ασθενών ανάλογα με το αν πρόκειται να εμφανίζουν κάποιας μορφής καρκίνο ή όχι. Τα δεδομένα αυτά έχουν προέλθει εφαρμόζοντας την τεχνική φασματομετρία μάζας σε ασθενείς οι οποίοι είτε είναι υγιείς είτε έχουν παρουσιάσει ήδη κάποιας μορφής καρκίνο, προχωρημένου ή και πρώιμου σταδίου. Οι μετρήσεις έγιναν απο τα κέντρα National Cancer Institute (NCI) και Eastern Virginia Medical School (EVMS), και πρόκειται για ένα σετ δεδομένων απο 900 ασθενείς με 10000 παραμέτρους για τον καθέναν. Όπως αναφέραμε και παραπάνω, ο στόχος του σετ αυτού είναι να γίνει ο σωστός διαχωρισμός των ασθενών ως προς την πρόβλεψη για το αν βρίσκονται στην ευπαθή ομάδα ή όχι. Για το συγκεκριμένο σετ δεδομένων, χρησιμοποιήσαμε τα δεδομένα εκπαίδευσης και τα δεδομένα αξιολόγησης (training and validation sets) τα οποία περιέχουν 200 συνολικά δεδομένα. Απο αυτά χρησιμοποιήσαμε τα 150 ως δεδομένα εκπαίδευσης και τα 50 υπόλοιπα ως σετ αξιολόγησης. Τα αποτελέσματα για όλους τους συνδυασμούς των παραμέτρων φαίνονται στον παρακάτω πίνακα

Πίνακας 6.19: Μέσο (%) σφάλμα ταξινόμησης του σετ δεδομένων Arcene με τον αλγόριθμο κοντινότερων γειτόνων. Σφάλμα ταξινόμησης στον χώρο των αρχικών διαστάσεων (D=10.000) ίσο με 24%.

| | K=10 | K=12 | K=16 | K=20 | K=24 | K=32 | K=64 |
|---|---|---|---|---|---|---|---|
| d=10 | **10** | **12** | **18** | **18** | **20** | **18** | **18** |
| d=16 | **14** | **22** | **16** | **22** | **18** | **18** | **18** |
| d=20 | **16** | **18** | **16** | **16** | **22** | **16** | **16** |
| d=24 | **14** | **14** | **18** | **20** | **18** | **18** | **16** |
| d=32 | **14** | **20** | 24 | **18** | **20** | **18** | **18** |
| d=40 | **16** | **14** | **14** | **16** | **16** | **22** | **18** |
| d=52 | **10** | **16** | **14** | **14** | **16** | **16** | **14** |
| d=64 | **14** | **16** | **22** | **20** | **22** | **18** | **14** |
| d=96 | **22** | **22** | **22** | **18** | **12** | **16** | **22** |
| d=128 | 24 | **10** | **26** | **14** | 28 | 28 | 26 |

Όπως είναι φανερό απο τον παραπάνω πίνακα, στις περισσότερες απο τις περιπτώσεις μετά απο την μείωση των διαστάσεων εφαρμόζοντας τον αλγόριθμο LLE έχουμε πολύ μεγάλη αύξηση

του αποτελέσματος ορθής ταξινόμησης. Το βέλτιστο αποτέλεσμα δίνεται για τις παραμέτρους $(\mathbf{K = 10, d = 10})$, $(\mathbf{K = 10, d = 52})$ και $(\mathbf{K = 12, d = 128})$ και βλέπουμε ότι έχουμε μείωση του σφάλματος κατά $\mathbf{10\%}$ σε σχέση με αυτό του αρχικού χώρου των 10.000 διαστάσεων. Το γεγονός αυτό, σε συνδυασμό με την δραματική μείωση στο κόστος των υπολογισμών κατά την διαδικασία της ταξινόμησης, απο 10.000 παραμέτρους έχουμε κρατήσει μόνο **10, 52** και **128** για τις παραπάνω περιπτώσεις αντίστοιχα, έρχεται να αποδείξει για άλλη μια φορά ότι η διαδικασία της μείωσης των διαστάσεων μπορεί να έχει πολλαπλά οφέλη τόσο στην βελτίωση του αποτελέσματος όσο και στον χρόνο που απαιτείται για τον υπολογισμό του. Τέλος αποτελεί πολύ μεγάλο ενδιαφέρον στο συγκεκριμένο πείραμα το αποτέλεσμα του σφάλματος $\mathbf{10\%}$ για τελικό αριθμό διαστάσεων $\mathbf{d{=}10}$ απο τις αρχικές $\mathbf{D{=}10.000}$.

# Κεφάλαιο 7

# Συμπεράσματα

Λαμβάνοντας υπόψιν τα αποτελέσματα των πειραμάτων τα οποία πραγματοποιήθηκαν στα πλαίσια της εν λόγω διατριβής έχουμε πλέον τα απαραίτητα στοιχεία απο τα οποία γίνεται φανερό ότι η μείωση των διαστάσεων αποτελεί έναν πολύ σημαντικό παράγοντα τόσο στην επίτευξη καλύτερων αποτελεσμάτων όσο και στην δραματική μείωση της πολυπλοκότητας των υπολογισμών κατά την διαδικασία της εξαγωγής του αποτελέσματος ταξινόμησης. Επίσης, απο την διαδικασία αυτή προκύπτουν σημαντικά ευρήματα και για τον αλγόριθμο μη γραμμικής μείωσης διαστάσεων LLE. Πιο συγκεκριμένα στο πρώτο πείραμα με το σετ δεδομένων MNIST χρησιμοποιήθηκε ο αλγόριθμος LLE ουσιαστικά για την εξαγωγή συγκεκριμένου αριθμού χαρακτηριστικών πάνω στα pixel κάθε εικόνας. Το τελικό αποτέλεσμα είναι ένα διάνυσμα μήκους d (οι τελικές διαστάσεις του αλγορίθμου) για κάθε εικόνα. Όπως αποδείχθηκε απο τα πειράματα το διάνυσμα αυτό περιέχει το σύνολο της πληροφορίας την οποία χρειαζόμαστε για την ταξινόμηση των δεδομένων στην κατάλληλη κλάση. Σε συγκεκριμένες περιπτώσεις μάλιστα φάνηκε ότι η διαδικασία της μείωσης των διαστάσεων μπορεί να βελτιώσει το ποσοστό σφάλματος στο τελικό βήμα της ταξινόμησης.

Στο δεύτερο πείραμα εφαρμόστηκε ο αλγόριθμος όχι στα pixel της εικόνας αλλά στο διάνυσμα των χαρακτηριστικών το οποίο προέκυψε απο την εφαρμογή του αλγορίθμου εξαγωγής χαρακτηριστικών HoG. Και σε αυτή την περίπτωση φαίνεται ξεκάθαρα απο τα πειράματα ότι μπορούμε να μειώσουμε κατά έναν μεγάλο βαθμό τον όγκο της πληροφορίας την οποία θα πρέπει να επεξεργαστούμε ώστε να ταξινομήσουμε τα δεδομένα στις κατάλληλες κλάσεις. Επίσης σε συγκεκριμένες

περιπτώσεις, ακόμα και μετά απο δραματική μείωση της διάστασης του διανύσματος χαρακτηριστικών HoG το αποτέλεσμα είναι εξίσου καλό ή και καλύτερο απο αυτό του χώρου των αρχικών διαστάσεων. Αυτό το αποτέλεσμα φανερώνει την δυνατότητα του αλγορίθμου να εντοπίζει και να απομακρύνει τον θόρυβο που περιέχεται στην πληροφορία των αρχικών χαρακτηριστικών βελτιώνοντας έτσι και τον χρόνο των υπολογισμών αλλά και την απόδοση του τελικού αποτελέσματος. Απο το τελευταίο πείραμα στο οποίο επίσης ο αλγόριθμος εφαρμόζεται σε έναν πολύ μεγάλο αριθμό χαρακτηριστικών-παραμέτρων (10.000) φαίνεται ότι και σε αυτή την περίπτωση εντοπίζονται τα στοιχεία τα οποία δεν μπορούν να συνεισφέρουν θετικά στην εξαγωγή ορθού συμπεράσματος και αποβάλλοντάς τα επιτυγχάνεται πολύ μεγάλη αύξηση στο ποσοστό ορθής ταξινόμησης των δεδομένων.

Απο την εκτέλεση των παραπάνω πειραμάτων μπορούν επίσης να εξαγχθούν και συγκεκριμένα στοιχεία ως προς τον τρόπο λειτουργίας του αλγορίθμου LLE. Πιο συγκεκριμένα απο το πρώτο πείραμα μπορούμε να συμπεράνουμε ότι αντίθετα με αλγορίθμους μηχανικής μάθησης όπως τα Νευρωνικά δίκτυα, για την εκπαίδευση του αλγορίθμου δεν απαιτείται τεράστιος αριθμός δεδομένων αλλά αρκεί ένας σωστά δειγματοληπτημένος χώρος ο οποίος να διατηρεί το λείο της πολλαπλότητας. Επίσης ο χώρος των δεδομένων θα πρέπει να είναι ομοιόμορφα δειγματοληπτημένος ώστε να μην υπάρχουν "μεγάλες αποστάσεις" μεταξύ δεδομένων της ίδιας κλάσης διότι αυτό μπορεί να αποτελέσει αρνητικό παράγοντα στην διατήρηση των γεωμετρικών χαρακτηριστικών της γειτονιάς για κάποιο σημείο.

Απο τα δύο παρακάτω πειράματα φάνηκε ότι ο αλγόριθμος LLE είναι σε θέση να επιτύχει μείωση των παραμέτρων σε διανύσματα χαρακτηριστικών, γεγονός το οποίο αποτελεί σημαντική μείωση του κόστους των υπολογισμών. Και αυτό διότι η εξαγωγή χαρακτηριστικών σε εικόνες είναι απο μόνος του ένας τρόπος μείωσης κατά ένα μεγάλο ποσοστό του κόστους των υπολογισμών. Με την μείωση λοιπόν των παραμέτρων του διανύσματος των χαρακτηριστικών για μια εικόνα η συμπίεση της πληροφορίας πλέον είναι τεράστια και το αποτέλεσμα της ταξινόμησης μπορεί πλέον να εξαχθεί λαμβάνοντας υπόψιν έναν πολύ μικρό αριθμό παραμέτρων.

Όπως είχε αναλυθεί και στην εισαγωγή της εργασίας η διαδικασία αυτή, δηλαδή η δραματική μείωση των παραμέτρων που πρέπει να εκτιμηθούν για την εξαγωγή κάποιου αποτελέσματος, δόθηκε

σαν ερέθισμα στον χώρο της Τεχνητής νοημοσύνης απο τον τρόπο με τον οποίο λειτουργεί ο ανθρώπινος εγκέφαλος. Αδιαμφισβήτητα λοιπόν, και λαμβάνοντας υπόψιν τα θετικά αποτελέσματα των πειραμάτων της εν λόγω εργασίας σε εφαρμογές αναγνώρισης προτύπων, η μείωση των διαστάσεων θα πρέπει να αποτελεί βασικό κομμάτι προεπεξεργασίας της πληροφορίας σε όλες σχεδόν τις εφαρμογές Μηχανικής μάθησης. Όπως έγινε φανερό με τον τρόπο αυτό μπορεί να μειωθεί δραματικά το κόστος των υπολογισμών με αποτέλεσμα να είναι εφικτή η εξαγωγή συμπεράσματος για εφαρμογές αναγνώρισης προτύπων σε πραγματικό χρόνο. Επίσης μπορεί να χρησιμοποιηθεί η μείωση των διαστάσεων σε Ιατρικές εφαρμογές επιτυγχάνοντας όπως φάνηκε απο το συγκεκριμένο πείραμα πολύ μεγάλη βελτίωση στην πρόβλεψη εμφάνισης συγκεκριμένης μορφής ασθενειών.

Κλείνοντας λοιπόν την εν λόγω διατριβή είναι φανερή η αποτελεσματικότητα του συγκεκριμένου αλγορίθμου **(LLE)** μείωσης των διαστάσεων σε εφαρμογές αναγνώρισης προτύπων. Επίσης λαμβάνοντας υπόψιν ευρήματα ερευνών απο τον χώρο της Ιατρικής για τον τρόπο λειτουργίας του ανθρώπινου εγκεφάλου γίνεται άμεσα φανερό ότι θα πρέπει να μπορέσουμε, αφού κατανοήσουμε πλήρως τον τρόπο με τον οποίο λειτουργεί, στην συνέχεια να εφαρμόσουμε αντίστοιχες τεχνικές σε εφαρμογές Τεχνητής Νοημοσύνης. Οι τεχνικές αυτές σε συνδυασμό με την ραγδαία αύξηση της τεχνολογίας και των διαθέσιμων πόρων που μας προσφέρουν οι σημερινοί, πόσω μάλλον οι μελλοντικοί, ηλεκτρονικοί υπολογιστές μπορούν να συμβάλλουν σημαντικά στον χώρο της Ιατρικής προσφέροντας εξαιρετική βελτίωση στην αποτελεσματικότητα αντιμετώπισης σοβαρών ασθενειών. Επίσης μπορούν να χρησιμοποιηθούν σε ένα μεγάλο πλήθος τόσο καθημερινών όσο και βιομηχανικών εφαρμογών οι οποίες θα αλλάξουν κατά πολύ τον ρόλο της Ρομποτικής αλλά και της Τεχνητής Νοημοσύνης στην καθημερινότητα μας.

# Bibliography


[1] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. 2000.

[2] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[3] Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and Lofti Zadeh. *Feature Extraction, Foundations and Applications.* 2006.

[4] Jolliffe. Principal component analysis. 2002.

[5] Y. h. Taguchi and Y.Oono. Relational patterns of gene expression via non-metric multidimensional scaling analysis. 2004.

[6] G. W. Stewart. On the early history of the singular value decomposition. 1992.

[7] Frank Dellaert. Singular value and eigenvalue decompositions. 2008.

[8] Dinoj Surendran. Swiss roll dataset. 2004.

[9] V. de Silva J. B. Tenenbaum and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. 2000.

[10] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. 2000.

[11] Jane K. Cullum and Ralph A. Willoughby. *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Vol. 1.* 2002.

[12] Xin Liu, Duygu Tosun, Michael W. Weiner, and Norbert Schuff. Locally linear embedding (lle) for mri based alzheimer's disease classification. 2013.

[13] Hualei Shen, Dacheng Tao, and Dianfu Ma. Multiview locally linear embedding for effective medical image retrieval. 2013.

[14] L.J.P. van der Maaten, E. O. Postma, and H. J. van den Herik. Dimensionality reduction: A comparative review. 2008.

[15] Olga Kouropteva, Oleg Okun, and Matti Pietikäinen. Supervised locally linear embedding algorithm for pattern recognition. 2003.

[16] Dick de Ridder and Robert P.W. Duin. Locally linear embedding for classification. 2002.

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. 1998.

[18] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. 1967.

[19] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. 2005.

[20] David G.Lowe. Distinctive image features from scale-invariant keypoints. 2004.

[21] Yun Fu and Thomas S.Huang. Locally linear embedded eigenspace analysis. 2005.

[22] Lawrence K. Saul and Sam T. Roweis. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. 2003.

[23] Laszlo Lovasz. Eigenvalues of graphs. 2007.

[24] Olga Kouropteva, Oleg Okun, and Matti Pietikainen. Incremental locally linear embedding algorithm. 2005.

[25] S.Theodoridis and K.Koutroumbas. *Pattern Recognition.* 2008.