ΕΘΝΙΚΟ **Μ**ΕΤΣΟΒΙΟ **Π**ΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ **Ε**ΦΑΡΜΟΣΜΕΝΩΝ **Μ**ΑΘΗΜΑΤΙΚΩΝ ΚΑΙ **Φ**ΥΣΙΚΩΝ **Ε**ΠΙΣΤΗΜΩΝ

ΤΟΜΕΑΣ ΜΗΧΑΝΙΚΗΣ

«Ανάπτυξη υπολογιστικών μεθόδων βασισμένων στην εκμάθηση μηχανών και πολλαπλοτήτων για την πρόβλεψη κόστους διαδρομών δικτύου από την ανάκτηση διαδικτυακών δεδομένων»

Διπλωματική εργασία

του

ΧΡΗΣΤΟΥ ΧΑΤΖΗΑΠΟΣΤΟΛΑΚΗ

Τριμελής Επιτροπή

Επιβλέπων: Κωνσταντίνος Σιέττος , *Αναπληρωτής Καθηγητής Ε.Μ.Π.*

Γεώργιος Ματσόπουλος, *Αναπληρωτής Καθηγητής Ε.Μ.Π.*

Ιωάννης Κομίνης, *Επίκουρος Καθηγητής Ε.Μ.Π.*

ΑΘΗΝΑ 2017

……..
Χρήστος Χατζηαποστολάκης
Διπλωματούχος Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών, Ε.Μ.Π.

Περίληψη

Από τις τελευταίες δεκαετίες του 20ου αιώνα μέχρι και σήμερα ο τρόπος με τον οποίο κοστολογείται ένα αεροπορικό εισιτήριο αποτελεί μυστήριο για τους περισσότερους καταναλωτές. Αντίθετα με τα υπόλοιπα μέσα μεταφοράς, η τιμή μιας πτήσης δεν είναι πάντα ανάλογη της απόστασης που καλύπτεται και μπορεί να αλλάξει οποιαδήποτε στιγμή. Για να αυξήσουν όσο το δυνατόν περισσότερο τα κέρδη τους οι αεροπορικές εταιρίες έχουν δημιουργήσει ένα πολύπλοκο σύστημα τιμολόγησης των αεροπορικών θέσεων, το οποίο καθιστά σχεδόν αδύνατο το να αγοράσει κάποιος το βέλτιστο δυνατό εισιτήριο. Σε αυτή την εργασία εξετάζεται το κατά πόσο είναι δυνατό να κατασκευαστεί ένα μοντέλο το οποίο να περιγράφει ικανοποιητικά το δίκτυο των αεροπορικών ναύλων. Για την κατασκευή του μοντέλου χρησιμοποιήθηκαν γραμμικές και μη-γραμμικές μεθόδοι μείωσης διαστάσεων και αναγνώρισης προτύπων από τις οποίες προκύπτουν οι κατάλληλες μεταβλητές και η τελική προσαρμογή.

Abstract

Since the last decades of the 20th century the way that an airplane ticket is priced seems like a mystery to most consumers. Unlike most transportation options, the price of a flight is not proportional to its distance and can also change at any moment. In order to maximize their profits, airlines have developed complex pricing methods that makes almost impossible for a traveler to buy at an optimal price. This paper examines if it is possible to construct a valid model which will successfully describe the European airfare network, using just a handful of variables. In order to create the model, various linear and non-linear feature extraction methods have been used accompanied by machine learning methods.

Ευχαριστίες

Με την ολοκλήρωση αυτής της διπλωματικής εργασίας, θα ήθελα να εκφράσω την ευγνωμοσύνη μου στον επιβλέποντα καθηγητή μου, Δρ. Κωνσταντίνο Σιέττο για τις πολύτιμες γνώσεις που μου μετέδωσε, την άπλετη βοήθεια και στήριξή του, καθώς και για την άψογη συνεργασία μας.

Επίσης οφείλω ένα μεγάλο ευχάριστώ στους συνεργάτες μου Ανδρέα Καρούτζο και Δημήτρη Σχίζα για τις αμέτρητες ώρες που δουλέψαμε μαζί και την συνεισφορά τους στην διατύπωση του προβλήματος.

Τέλος να ευχαριστήσω την οικογένειά μου για τη συνεχή ηθική και υλική στήριξή της και την Ιφιγένεια Μαυραγάνη για τη διαρκή του συμπαράσταση και αγάπη.

Περιεχόμενα

Κεφάλαιο 1 Εισαγωγή

1.1 Το πρόβλημα

Καθώς οι τιμές των προϊόντων είναι διαθέσιμες πλέον στον διαδίκτυο, οι καταναλωτές προσπαθούν να κατανοήσουν πώς οι εταιρείες μεταβάλουν τις τιμές τους. Παρόλαυτα οι εταιρείες καθορίζουν τις τιμές των προϊόντων τους με βάση τους δικούς τους αλγόριθμους, οι οποίοι λαμβάνουν υπόψη παραμέτρους οι οποίες δεν είναι διαθέσιμες στο κοινό, όπως ο αριθμός διαθέσιμων θέσεων μιας πτήσης. Πολλοί έχουν προσπαθήσει να προσεγγίσουν αυτές τις παραμέτρους [30, 31, 32] εκπαιδεύοντας αλγορίθμους οι οποίοι θα μπορούσαν να εξοικονομήσουν μέχρι και 27.1% στους επιβάτες εάν τους συμβούλευε πότε είναι η κατάλληλη στιγμή να αγοράσουν το εισιτήριο τους. Ένας άλλος σημαντικός παράγοντας που επηρεάζει τη διακύμανση των αεροπορικών εισιτηρίων είναι οι συνεργασίες μεταξύ των αεροπορικών εταιριών. Έχει φανεί [33] πως τέτοιες συνεργασίες μπορούν να ρίξουν τις τιμές μέχρι και 25% σε σχέση με τις αυτόνομες εταιρίες.

Σύμφωνα με τα στελέχη των αεροπορικών εταιριών, αυτό που κάνει την τιμολόγηση των αεροπορικών ταξιδιών τόσο περίπλοκη, είναι ότι δεν υπάρχει μία τιμή που μπορούν να χρεώνουν για μια θέση ώστε να έχουν ένα κέρδος. Είτε η τιμή θα είναι πολύ υψηλή και η αεροπορική εταιρεία δεν θα προσελκύσει αρκετούς επιβάτες για να γεμίσει το αεροπλάνο, ή το αεροπλάνο θα είναι πλήρες, αλλά η εταιρεία δεν θα καλύψει το κόστος της. Ως αποτέλεσμα, οι αεροπορικές εταιρείες έχουν αναπτύξει ένα σύστημα στο οποίο όσοι ταξιδεύουν για επαγγελματικούς λόγους πληρώνουν περισσότερα για ευέλικτα εισιτήρια και βολικές ώρες, και όσοι επιθυμούν να πληρώσουν λιγότερα θα πρέπει να κλείσουν νωρίτερα τα εισιτήρια τους ή να δεχθούν ότι θα πετάξουν πιο δύσκολες ώρες.

Η βασική μονάδα της τιμολόγησης, ένα «εισιτήριο», ορίζεται ως η τιμή ενός ταξιδιού μεταξύ των δύο πόλεων, ανεξάρτητα από τον αριθμό των πτήσεων που εμπλέκονται. Οι κατάλογοι των ναύλων ενημερώνονται από τις αεροπορικές εταιρείες δέκα φορές την ημέρα. Με κάθε εισιτήριο έρχεται ένα σύνολο κανόνων για τη χρήση του. Χαμηλές τιμές, για παράδειγμα, συχνά έχουν απαιτήσεις όπως η αγορά των προτέρων

δύο εβδομάδων, μια σύνδεση, ή τα ταξίδια σε ακατάλληλη ώρα της ημέρας. Κανόνες άλλου τύπου περιορίζουν τον τρόπο με τον οποίο ένα εισιτήριο μπορεί να συνδυαστεί με άλλες τιμές.

Ως αποτέλεσμα του πολύπλοκου συστήματος κανόνων, ο σχεδιασμός αεροπορικών ταξιδιών είναι πολύ πιο περίπλοκος από ό, τι μια αναζήτηση συντομότερης διαδρομής. Ένα φθηνό αεροπορικό εισιτήριο από την πόλη Α στην πόλη Γ δεν μπορεί κατ 'ανάγκη να δημιουργηθεί από ένα συνδυασμό φθηνών εισιτηρίων από το Α στο Β και από το Β στο Γ.

Η υπολογιστική πολυπλοκότητα του προβλήματος αναζήτησης αεροπορικών εισιτηρίων είναι αποτέλεσμα πολύπλοκων συστημάτων κανόνων των αεροπορικών εταιρειών. Κανόνες που έχουν σχέση με ένα εισιτήριο που χρησιμοποιείται για να πληρώσει για μια ενιαία πτήση μπορεί να περιορίσουν κάθε άλλο ναύλο και πτήση που συνδέονται με το ίδιο εισιτήριο. Περιορισμοί αυτού του είδους βάζουν το πρόβλημα των αεροπορικών ταξιδιών στην κατηγορία των NP-hard προβλημάτων. Έτσι η ύπαρξη ενός αποδοτικού αλγορίθμου για την επίλυση αυτή συνεπάγεται την ύπαρξη αποδοτικών αλγορίθμων για μία ολόκληρη κατηγορία φαινομενικά δυσεπίλυτων προβλημάτων.

Στην εργασία αυτή γίνεται ανάλυση του δικτύου των αεροπορικών ναύλων με τη χρήση τεχνικών μείωσης διαστάσεων για να εξεταστεί εάν τελικά υπάρχουν κάποιες κρυφές μεταβλητές, οι οποίες έχουν προκύψει μέσα από αυτό το πολύπλοκο σύστημα τιμολόγησης. Η ύπαρξη τέτοιων μεταβλητών θα βοηθήσει στην κατανόηση του τρόπου λειτουργίας των αερομεταφορών που παίζει ένα πολύ σημαντικό ρόλο στο παγκόσμιο εμπόριο και την οικονομία.

Κεφάλαιο 2 Μείωση διαστάσεωνδεδομένων

2.1 Εισαγωγή

Τα δεδομένα που συλλέγονται από πρακτικά προβλήματα όπως η ανάλυση της ανθρώπινης ομιλίας, οι ψηφιακές φωτογραφίες ή η λειτουργική απεικόνιση μαγνητικού συντονισμού, συνήθως έχουν πολύ μεγάλο αριθμό διαστάσεων. Για να καταφέρουμε να χρησιμοποιήσουμε τα δεδομένα αυτά αποτελεσματικά πολλές φορές χρειάζεται να μειώσουμε τον όγκο τους, και κατά συνέπεια έχουν αναπτυχθεί πολλές τεχνικές μείωσης διαστάσεων για την επίλυση αυτού του προβλήματος. Μείωση διαστάσεων είναι ο μετασχηματισμός των πολυδιάστατων δεδομένων σε μία αναπαράσταση σημαντικά λιγότερων διαστάσεων, με τη λιγότερη δυνατή απώλεια πληροφορίας. Ιδανικά οι διαστάσεις αυτής τη νέας αναπαράστασης θα είναι και ο ελάχιστος αριθμός μεταβλητών που χρειάζεται για να περιγράψουμε τα δεδομένα [1].

Η μείωση διαστάσεων είναι σημαντική σε πολλούς τομείς αφού μας βοηθάει στο να αντιμετωπίσουμε πολυδιάστατα και σύνθετα προβλήματα. Σαν αποτέλεσμα η μείωση διαστάσεων χρησιμοποιείται για την ταξινόμηση, την απεικόνιση και την συμπίεση πολυδιάστατων δεδομένων. Παραδοσιακά η μείωση των διαστάσεων ενός συνόλου δεδομένων επιτυγχάνονταν με χρήση γραμμικών μεθόδων όπως η μέθοδος της Εύρεσης των Κύριων Συνιστωσών (PCA) [2] και η ανάλυση κατά παράγοντες. Ωστόσο αυτές οι γραμμικές τεχνικές αποτυγχάνουν στη διαχείρισημη γραμμικών προβλημάτων.

Για αυτό το λόγο την τελευταία δεκαετία έχουν προταθεί πολλές μη γραμμικές μέθοδοι για την μείωση διαστάσεων βασισμένες κυρίως στην εκμάθηση πολλαπλοτήτων (manifoldlearning). Σε αντίθεση με τις κλασσικές γραμμικές τεχνικές, αυτές οι μη γραμμικές μέθοδοι μπορούν να αντιμετωπίσουν σύνθετα προβλήματα με μη γραμμικά δεδομένα. Για την ακρίβεια επειδή τα περισσότερα πρακτικά προβλήματα σπάνια παρουσιάζουν κάποια γραμμικότητα, αυτές οι μη γραμμικές τεχνικές προσφέρουν ένα σημαντικό πλεονέκτημα. Προηγούμενες μελέτες έχουν δείξει πως μη γραμμικές τεχνικές έχουν πολύ καλύτερα αποτελέσματα από αντίστοιχες γραμμικές μεθόδους στην επίλυση σύνθετων θεωρητικών προβλημάτων.

Πέρα από αυτά τα τεχνητά προβλήματα όμως δεν είναι ξεκάθαρο κατά πόσο οι μη γραμμικές τεχνικές επιλύουν καλύτερα αληθινά, πρακτικά προβλήματα.

Σε αυτή την εργασία θα συγκρίνουμε τα αποτελέσματαδιαφόρων τεχνικών μείωσης διαστάσεων, σε ένα πρακτικό ζήτημα όπως αυτό της τιμολόγησης των αεροπορικών εισιτηρίων. Αναλυτικότερα στο Κεφάλαιο 3 θα δούμε την σημαντικότερη ίσως μέθοδο μείωσης διαστάσεων, αυτή της Εύρεσης των Κύριων Συνιστωσών (PCA). Οι υπόλοιπες τεχνικές που θα δούμε είναι η μέθοδος Isomap [19] (Κεφάλαιο 4) η μέθοδος Diffusionmaps [22,23] (Κεφάλαιο 5) που ανήκουν στους αλγορίθμους εκμάθησης πολλαπλοτήτων και η μέθοδος των Autoencoders (Κεφάλαιο 6) που ανήκει στην κατηγορία της εκμάθησης μηχανών.

2.2 Το πρόβλημα της μείωσης διαστάσεων

Το πρόβλημα της μείωσης διαστάσεων μπορεί να οριστεί ως εξής. Έστω ένα σύνολο δεδομένων το οποίο αναπαριστάται από ένα $n \times D$ πίνακα X ο οποίος αποτελείται από n διανύσματα x_i ($i \in \{1, 2, \dots, n\}$) με Dδιαστάσεις. Ας υποθέσουμε ακόμα ότι ο πραγματικός αριθμός διαστάσεων που χρειάζεται για να περιγράψουμε τα δεδομένα είναι d, όπου $d \ll D$. Όταν λέμε ότι ο αληθινός αριθμός των διαστάσεων είναι d ,εννοούμε πως τα δεδομένα ανήκουν σε μια d-διάσταση πολλαπλότητα εντός του D-διάστατου χώρου.

Οι τεχνικές μείωσης διαστάσεων μετασχηματίζουν το D-διάστατο σύνολο X σε ένα νέο σύνολο Y το οποίο έχει τελικά dδιαστάσεις, διατηρώντας ταυτόχρονα όσο το δυνατόν καλύτερα τη γεωμετρία του αρχικού συνόλου. Τις περισσότερες φορές δεν γνωρίζουμε εξ αρχής τη γεωμετρία της d-διάστατης πολλαπλότητας, ούτε τον αριθμό των πραγματικών διαστάσεων d. Συνεπώς η μείωση διαστάσεων είναι ένα ασθενώς-ορισμένο πρόβλημα το οποίο μπορούμε να λύσουμε μόνο εάν κάνουμε συγκεκριμένες υποθέσεις για τα δεδομένα, όπως ο αριθμός d.

2.3 Η μέθοδος εύρεσης των κύριων συνιστωσών (PCA)

Οι γραμμικές μέθοδοι καταφέρνουν την μείωση των διαστάσεων απεικονίζοντας τα δεδομένα σε ένα υπόχωρο του αρχικού με πολύ λιγότερες διαστάσεις. Η πιο γνωστή

από τις γραμμικές μεθόδους είναι ο αλγόριθμος εύρεσης κύριων συνιστωσών (PrincipalComponentAnalysis). Η μέθοδος αυτή κατασκευάζει μια αναπαράσταση των δεδομένων με λιγότερες διαστάσεις διατηρώντας όσο το δυνατόν περισσότερη από τη διασπορά του αρχικού συνόλου. Αυτό επιτυγχάνεται βρίσκοντας μια γραμμική βάση λιγότερων διαστάσεων για τον πίνακα των δεδομένων, όπου το σύνολο της διασποράς είναι μέγιστο.

Πιο συγκεκριμένα η μέθοδος PCA προσπαθεί να βρει μια απεικόνισηM η οποία μεγιστοποιεί την ποσότητα $M^T cov(X) M$ όπου $cov(X)$ είναι ο πίνακας συνδιασποράς του συνόλου δεδομένων X.Αποδεικνύεται πως αυτή η απεικόνιση κατασκευάζεται από τα d κύρια ιδιοδιανύσματα (κύριες συνιστώσες) του πίνακα συνδιασποράς του κανονικοποιημένου συνόλου δεδομένων. Συνεπώςη μέθοδος PCA λύνει το παρακάτω πρόβλημα ιδιοτιμών:

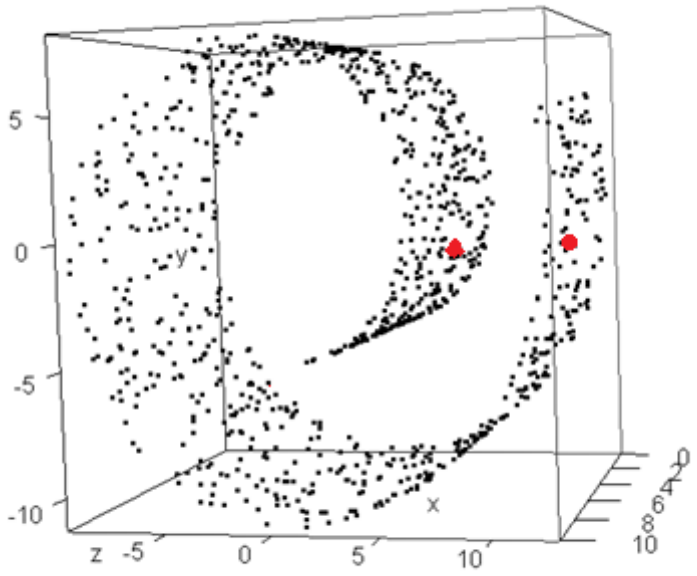
$$cov(X)M = \lambda M$$

Τελικά το ζητούμενο σύνολο δεδομένωνY κατασκευάζεται προβάλλοντας το αρχικό σύνολο πάνω στην γραμμική βάση M.Η μέθοδος PCA έχει εφαρμοσθεί με επιτυχία σε μια πληθώρα προβλημάτων όπως η αναγνώριση ανθρώπινων προσώπων [4], η ταξινόμηση νομισμάτων [5], και η ανάλυση χρονοσειρών σεισμικών δονήσεων [6]. Το κύριο μειονέκτημα αυτής της μεθόδου είναι ότι ο υπολογισμός των ιδιοδιανυσμάτων μπορεί να μην πραγματοποιείται για πολυδιάστατα δεδομένα.

2.4 Η μέθοδος Isomap

Η μη γραμμική μέθοδος Isomap μας βοηθάει να εντοπίσουμε την αληθινή πολλαπλότητα στην οποία ανήκουν τα δεδομένα μας. Πολλές άλλες μη γραμμικές τεχνικές όπως η Πολυδιάστατη Κλιμακοποίηση (MDS) [3] αποτυγχάνουν να αποτυπώσουν την αληθινή φύση των δεδομένων επειδή η ανάλυση τους στηρίζεται στη χρήση της ευκλείδειας απόστασης μεταξύ των παρατηρήσεων. Όταν όμως τα δεδομένα προέρχονται από μία μη γραμμική πολλαπλότητα, όπως το σύνολο δεδομένων Swissroll [7] η ευκλείδεια απόσταση δεν αποτελεί ιδανικό μέτρο για την αληθινή απόσταση μεταξύ δύο σημείων.

Όπως φαίνεται στην Εικόνα 1 κάνοντας χρήση της ευκλείδειας απόστασης στον τρισδιάστατο χώρο οδηγούμαστε στο εσφαλμένο συμπέρασμα πως δύο σημεία είναι κοντά μεταξύ τους, ενώ στην πραγματικότητα απέχουν κατά πολύ πάνω στην πολλαπλότητα στην οποία ανήκουν.



Εικόνα 1: Τοσύνολο δεδομένων των 3-διαστάσεων δεδομένων του επονομαζόμενου Swiss- roll

Ο αλγόριθμος Isomap [7] προσπαθεί να αντιμετωπίσει αυτό το πρόβλημα προσεγγίζοντας την πραγματική απόσταση πάνω στην πολλαπλότητα με τη χρήση της γεωδαιτικής απόστασης.

Ειδικότερα, οι γεωδαιτικές αποστάσεις [7]πάνω στην πολλαπλότητα υπολογίζονται κατασκευάζοντας τον γράφο G των παρατηρήσεων x_i ($i \in \{1, 2, \dots, n\}$), στον οποίο κάθε παρατήρηση x_i είναι συνδεδεμένη με τα k πλησιέστερα σημεία x_{ij} ($j \in \{1, 2, \dots, n\}$). Το συντομότερο μονοπάτι μεταξύ δύο σημείων πάνω στο γράφο είναι μία καλήπροσέγγιση της γεωδαιτικής απόστασης και μπορεί να υπολογιστεί χρησιμοποιώντας τον αλγόριθμο του Dijkstra ή του Floyd [8, 9]. Με αυτό τον τρόπο κατασκευάζουμε τον πίνακα των γεωδαιτικών αποστάσεων. Το τελικό d-διάστατο σύνολο Yπροκύπτει εφαρμόζοντας τον αλγόριθμο της πολυδιάστατης κλιμακοποίησης πάνω στον πίνακα αυτών των αποστάσεων.

2.5 Η μέθοδος DiffusionMap

Η μέθοδος αυτή [10, 11] προέρχεται από τον χώρο των δυναμικών συστημάτων και αποσκοπεί πάλι στην εκμάθηση της χαμηλής διάστασης των πολλαπλοτήτων που χαρακτηρίζουν τα δεδομένα. Η τεχνική αυτή βασίζεται σε έναν Μαρκοβιανό τυχαίο περίπατο πάνω στον γράφο των δεδομένων. Μετά από κάποια στάδια του τυχαίου περίπατου μπορούμε να ορίσουμε ένα μέτρο που αφορά την εγγύτητα μεταξύ των σημείων. Το μέτρο αυτό καλείται απόσταση διάχυσης. Στη μετέπειτα *d*-διάστατη αναπαράσταση των δεδομένων στόχος είναι η διατήρηση της απόστασης διάχυσης όσο το δυνατόν καλύτερα.

Όσον αφορά τον αλγόριθμο, πρώτα κατασκευάζεται ο γράφος των δεδομένων. Τα βάρη στις ακμές υπολογίζονται από τον γκαουσιανό kernel. Κατασκευάζεται δηλαδή ένας πίνακας *W*με στοιχεία:

$$w_{ij} = e^{\frac{\|x_i - x_j\|^2}{2\sigma^2}}$$

Όπου σ^2 είναι η διασπορά της γκαουσιανής συνάρτησης. Στη συνέχεια εφαρμόζεται κανονικοποίηση του πίνακα *W* κατασκευάζοντας τον πίνακα $P^{(1)}$ με στοιχεία:

$$p_{ij}^{(1)} = \frac{w_{ij}}{\sum_k w_{ik}}$$

Ό πίνακας $P^{(1)}$ μπορεί να θεωρηθεί ένας πίνακας Markov που περιγράφει την πιθανότητα μετάβασης μιας δυναμικής διαδικασίας. Δηλαδή ο πίνακας $P^{(1)}$ αναπαριστά την πιθανότητα μετάβασης από μια παρατήρηση του συνόλου δεδομένων σε μία άλλη σε ένα βήμα. Ο πίνακας πιθανοτήτων μετάβασης μετά από t βήματα $P^{(t)}$ είναι $(P^{(1)})^t$. Χρησιμοποιώντας τις πιθανότητες μετάβασης του τυχαίου περιπάτου $p_{ij}^{(t)}$ ορίζουμε την απόσταση διάχυσης ως εξής:

$$D^{(t)}(x_i, x_j) = \sqrt{\sum_k \frac{(p_{ik}^{(t)} - p_{jk}^{(t)})^2}{\psi(x_k)^{(0)}}}$$

Ο όρος $\psi(x_k)^{(0)}$ δίνει περισσότερο βάρος στις πυκνότερες περιοχές του γράφου και ορίζεται ως: $\psi(x_k)^{(0)} = \frac{m_i}{\sum_j m_j}$, όπου m_i είναι ο βαθμός του κόμβου x_iκαι

υπολογίζεται $m_i = \sum_j p_{ij}$. Από την τελευταία εξίσωση φαίνεται πως τα ζεύγη παρατηρήσεων με μεγάλη πιθανότητα μετάβασης έχουν μικρή απόσταση διάχυσης. Η κύρια ιδέα είναι ότι στηρίζεται σε πολλά μονοπάτια του γράφου, αυτό κάνει την απόσταση διάχυσης λιγότερο επιρρεπή στο θόρυβο από ότι για παράδειγμα η γεωδαιτική απόσταση. Στην τελική αναπαράσταση των δεδομένων *Y,* ο αλγόριθμος diffusionmap προσπαθεί να διατηρήσει τις αποστάσεις διάχυσης. Αυτή η αναπαράσταση προκύπτειλύνοντας το παρακάτω πρόβλημα ιδιοτιμών

$$P^{(t)}\upsilon = \lambda\upsilon$$

Η πρώτη ιδιοτιμή είναιτετριμμένη ($\lambda_1 = 1$) και το ιδιοδιάνυσμά της υ_1απορρίπτεται. Το τελικό *d*-διάστατο σύνολο Y δίνεται από τα υπόλοιπα d ιδιοδιανύσματα, αφού κανονικοποιηθούν από τις αντίστοιχες ιδιοτιμές, δηλαδή:

$$Y = \{\lambda_2\upsilon_2,\ \lambda_3\upsilon_3, \ldots, \lambda_{2d+1}\upsilon_{d+1}\}$$

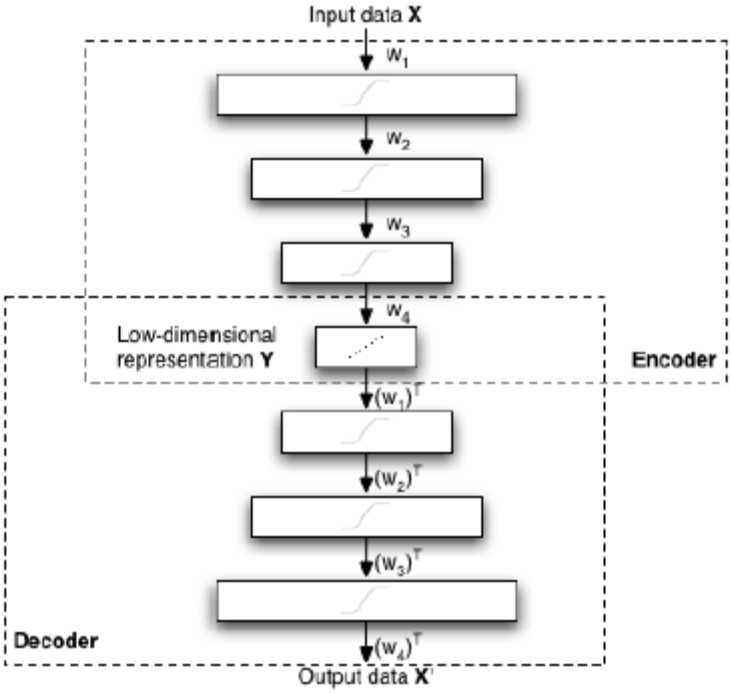
2.6 MultilayerAutoencoders

Πρόκειται ουσιαστικά για τεχνητά νευρωνικά δίκτυα με αρκετά κρυφά στρώματα νευρώνων [12, 13]. Ο μεσαίος κρυμμένος νευρώνας αποτελείται από d κόμβους, ενώ οι νευρώνες εισόδου και εξόδου αποτελούνται από *D*κόμβους. Το δίκτυο (βλ. Εικόνα 2) εκπαιδεύεται ώστε να ελαχιστοποιεί το σφάλμα μεταξύ των δεδομένων εισόδου και εξόδου. Με τη διαδικασία της εκπαίδευσης ο μεσαίος κρυμμένος νευρώνας υπολογίζει μια *d*-διάστατη αναπαράσταση του αρχικού συνόλου *X,* διατηρώντας όση περισσότερη πληροφορία από το αρχικό σύνολο γίνεται. Χρησιμοποιώντας γραμμική συνάρτηση ενεργοποίησης μπορούμε να πετύχουμε γραμμική μείωση διαστάσεων ενώ χρησιμοποιώντας τη σιγμοειδή συνάρτηση ενεργοποίησης μπορούμε να βρούμε μη γραμμικές απεικονίσεις.

Συνήθως ο αριθμός των ακμών μεταξύ του κόμβων του δικτύου είναι πολύ μεγάλος και αυτό οδηγεί σε πολύ αργή σύγκλιση του αλγόριθμου εκμάθησης. Επίσης είναι πολύ πιθανό ο αλγόριθμος να κολλήσει σε τοπικά ελάχιστα. Το πρόβλημα αυτό

λύνεται με τη χρήση μηχανών Boltzmann, δηλαδή τεχνητά νευρωνικά δίκτυα με δύο νευρώνες.

Η μέθοδος multilayerAutoencoders έχει εφαρμοσθεί με επιτυχία σε προβλήματα όπως η δημιουργία χαμένων τιμών [34]και η ανάλυση του ιού HIV[35].



Εικόνα 2: Περιγραφή της αρχιτεκτονικής ενός autoencoder. Πηγή: [39]

Κεφάλαιο 3 Ανάλυση Κύριων Συνιστωσών (PCA)

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο ορίζουμε όλες εκείνες τις στοιχειώδεις μαθηματικές έννοιες που συνθέτουν τη μέθοδο της ανάλυσης κυρίων συνιστωσών. Θα εξετάσουμε την κάθε μια ξεχωριστά και με παραδείγματα. Είναι σημαντικότερο να κατανοήσουμε γιατί μια τέτοια μέθοδος χρησιμοποιείται και τι μας λέει για τα δεδομένα το αποτέλεσμα, πάρα να θυμόμαστε τον ακριβή μηχανισμό της μεθόδου.

Ο τομέας της στατιστικής ανάλυσης δεδομένων βασίζεται στην εξής ιδέα, έστω ότι έχουμε ένα μεγάλο σύνολο δεδομένων και θέλουμε να το αναλύσουμε ως προς τις σχέσεις μεταξύ των διαφόρων ξεχωριστών χαρακτηριστικών (μεταβλητών) του. Θα δούμε κάποια από τα μέτρα που μας δίνουν μια πρώτη εικόνα για τα δεδομένα.

3.2 Μαθηματικό υπόβαθρο

3.2.1 Τυπική απόκλιση

Για να κατανοήσουμε την τυπικήαπόκλιση αρχικά χρειαζόμαστε ένα σύνολο δεδομένων ή αλλιώς ένα *τυχαίο δείγμα* από ένα *πληθυσμό*. Μπορούμε να χρησιμοποιήσουμε ως παράδειγμα τις εκλογές σε μια χώρα όπου πληθυσμός είναι ολόκληρος ο πληθυσμός της χώρας και το σύνολο δεδομένων μας ένα υποσύνολο του πληθυσμού από το οποίο μπορούμε να πάρουμε μετρήσεις (π.χ. exitpolls). Με τη βοήθεια της στατιστικής μπορούμε, εξετάζοντας μόνο το τυχαίο αυτό δείγμα να προβλέψουμε με σχετική ακρίβεια ποίο θα ήταν το αποτέλεσμα των εκλογών εάν είχαμε τη δυνατότητα να εξετάσουμε ολόκληρο τον πληθυσμό. Σε όλα τα παραδείγματα που ακολουθούν θεωρούμε ότι τα δείγματα που μελετάμε είναι υποσύνολα κάποιου μεγαλύτερου πληθυσμού.

Έστω το σύνολο δεδομένων:

$$X = [\,1\ 2\ 4\ 6\ 12\ 15\ 25\ 45\ 68\ 67\ 65\ 98\,]$$

Θα αναφερόμαστε στο παραπάνω σύνολο με το σύμβολο X, και με X_i στην παρατήρηση που αντιστοιχεί στο δείκτη *i*όπου *i*ϵ {1,2,3,…,12}. Για παράδειγμα $X_3 = 4$. Ακόμα συμβολίζουμε με *n*τον αριθμό των στοιχείων του συνόλου X. Με βάση τα παραπάνω μπορούμε να υπολογίσουμε κάποιες ποσότητες που περιγράφουν το εκάστοτε σύνολο δεδομένων. Για παράδειγμα μπορούμε να υπολογίσουμε το *δειγματικό μέσο* του συνόλου:

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}$$

Συμβολίζουμε με $\bar{X}$ το δειγματικό μέσο του συνόλου X, ο οποίος μας δίνει τη μέση τιμή του τυχαίου δείγματος. Πέρα από αυτή τη μέση τιμή ο δειγματικός μέσος δεν μας δίνει κάποια άλλη πληροφορία για τη φύση του δείγματος. Για παράδειγμα τα σύνολα δεδομένων που ακολουθούν έχουν ίδιο δειγματικό μέσο (10) αλλά είναι πολύ διαφορετικά μεταξύ τους:

[0 8 12 20] *και* [8 9 11 12]

Τι είναι όμως αυτό που κάνει τα δύο αυτά σύνολα διαφορετικά μεταξύ τους; Είναι το *εύρος* των δεδομένων που κάνει τα δείγματα να διαφέρουν. Η τυπική απόκλιση ενός συνόλου δεδομένων μετρά ακριβώς αυτή την έκταση των δεδομένων. Πιο συγκεκριμένα η τυπική απόκλιση είναι «Η μέση απόσταση των δεδομένων του δείγματος από το δειγματικό μέσο» και υπολογίζεται ως εξής:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(X_i - X)^2}{(n-1)}}$$

Συμβολίζουμε την τυπική απόκλιση με *s*. Χρησιμοποιούμε *(n - 1)* αντί για *n* στον παρονομαστή του παραπάνω κλάσματος είναι ότι όταν υπολογίζουμε την τυπική απόκλιση σε ενός τυχαίου δείγματος αυτός ο τύπος μας δίνει καλύτερη εκτίμηση της πραγματικής τυπικής απόκλισης του πληθυσμού. Αντίθετα αν υπολογίζαμε την τυπική απόκλιση του πληθυσμού απευθείας θα χρησιμοποιούσαμε *n* στον παρονομαστή. Συνεπώς οι τυπικές αποκλίσεις των δύο προηγούμενων συνόλων είναι 8.3266 και 1.8257 αντίστοιχα, φαίνεται δηλαδή ξεκάθαρα η μεγάλη διαφορά μεταξύ των δειγμάτων.

3.2.2 Διασπορά

Η διασπορά είναι ένα ακόμα μέτρο του «εύρους» των δεδομένων ενός τυχαίου δείγματος. Στην πραγματικότητα δεν είναι τίποτα άλλο από την τυπική απόκλιση υψωμένη στο τετράγωνο. Η διασπορά ενός τυχαίου δείγματος συμβολίζεται με s^2 και υπολογίζεται ως εξής:

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - X)^2}{(n-1)}$$

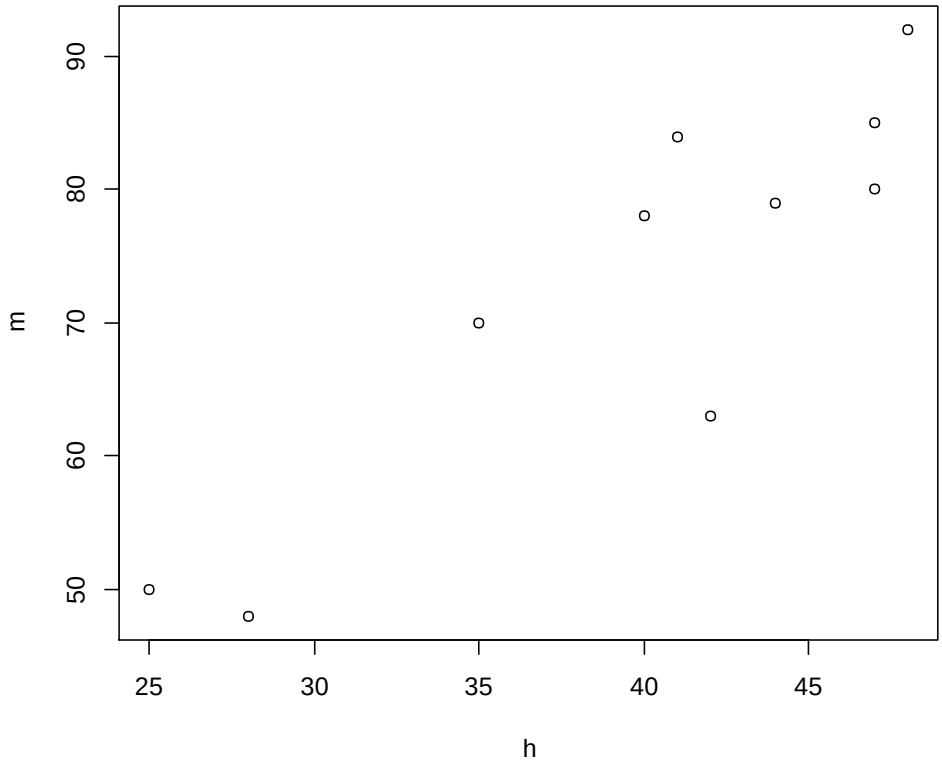
3.2.3 Συνδιασπορά

Η τυπική απόκλιση και η διασπορά είναι και οι δύο μονοδιάστατα μέτρα. Είναι χρήσιμα για να εξερευνήσουμε το εύρος δειγμάτων που αφορούν για παράδειγμα το ύψος των μαθητών ενός σχολείου ή βαθμολογία τους σε ένα διαγώνισμα. Πολλές φορές όμως θέλουμε να εξετάσουμε τη σχέση μεταξύ δύο ξεχωριστών χαρακτηριστικών του πληθυσμού. Για παράδειγμα μπορεί να θέλουμε να εξετάσουμε εάν επηρεάζει το ύψος των μαθητών την απόδοση τους στο συγκεκριμένο διαγώνισμα.
Η συνδιασπορά είναι ένα τέτοιο μέγεθος και υπολογίζεται πάντα μεταξύ δυο διαστάσεων ως εξής:

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - X)(Y_i - Y)}{(n-1)}$$

Πως όμως δουλεύει η συνδιασπορά; Ας υποθέσουμε πως έχουμε συλλέξει ένα δυσδιάστατο σύνολο δεδομένων από κάποιους μαθητές ενός σχολείου, τους οποίους ρωτήσαμε πόσες ώρες ξόδεψαν μελετώντας για ένα μάθημα *(H)* και τον τελικό βαθμό της εξέτασης στο μάθημα αυτό *(M)*.

Όπου *H* = [40 42 47 47 25 44 41 48 35 28] και *M* = [78 63 80 85 50 79 84 92 70 48]



Εικόνα 3: Οι ώρες μελέτης και ο τελικός βαθμός του μαθήματος

Στην εικόνα 3 βλέπουμε όπως ήταν αναμενόμενο ότι όσο περισσότερες ώρες μελέτης είχε αφιερώσει ένας μαθητής τόσο υψηλότερη βαθμολογία πέτυχε στο τελικό διαγώνισμα. Αυτή ακριβώς τη σχέση μας δίνει η συνδιασπορά. Για την ακρίβεια η συνδιασπορά προκύπτει 106.63. Το νούμερο αυτό δεν μας λέει τίποτα από μόνο του, αυτό που μας ενδιαφέρει είναι το πρόσημο του.

Αν είναι αρνητικό όσο αυξάνεται το ένα από τα δύο χαρακτηριστικά, το άλλο μειώνεται. Αντίθετα αν το πρόσημο της συνδιασποράς είναι θετικό, όσο αυξάνεται το ένα από τα δύο χαρακτηριστικά τότε και το άλλο αυξάνεται.

3.2.4 Πίνακας συνδιασποράς

Είδαμε ότι η συνδιασπορά υπολογίζεται πάντα μεταξύ 2 διαστάσεων. Αν το σύνολο δεδομένων μας έχει παραπάνω από 2 διαστάσεις υπάρχει ένα μέγεθος που μπορούμε

να υπολογίσουμε. Για παράδειγμα από ένα τρισδιάστατο σύνολο δεδομένων με διαστάσεις *x,y,z*μπορούμε να υπολογίσουμε τα *cov(x,y), cov(x,z)* και *cov(y,z).* Για την ακρίβεια για ένα *n*-διάστατο σύνολο δεδομένων μπορούμε να υπολογίσουμε $\frac{n!}{2(n-2)!}$ διαφορετικές διασπορές.

Ένας πρακτικός τρόπος να πάρουμε όλες τις πιθανές συνδιασπορές είναι να τις υπολογίσουμε όλες και να τις βάλουμε σε ένα πίνακα. Συνεπώς ο πίνακας συνδιασποράς σε ένα *n*-διάστατο σύνολο δεδομένων ορίζεται ως εξής:

$$C^{nxn} = (c_{ij}, c_{ij})$$

Για παράδειγμα ο πίνακας συνδιασποράς για ένα τρισδιάστατο σύνολο δεδομένων με διαστάσεις *x,y,z*είναι :

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,y) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

Η διαγώνιος μας δίνει τη διασπορά της κάθε διάστασης και αφού γενικά ισχύει *cov(x,y) = cov(y,x)*, βλέπουμε ακόμα ότι ο πίνακας είναι συμμετρικός ως προς την διαγώνιο του.

3.3 Η μέθοδος ΑνάλυσηςΚύριων Συνιστωσών (PCA)

Η μέθοδος PCAβρίσκει πολλές εφαρμογές στην αναγνώριση προτύπων καθώς μας βοηθάει να εκφράσουμε τα δεδομένα που μελετάμε, με τέτοιο τρόπο ώστε να γίνουν εμφανείς κρυφές ομοιότητες ή σημαντικές διαφοροποιήσεις που μπορεί να υπάρχουν στα δεδομένα.

Η πιο δημοφιλής εφαρμογή αυτής της μεθόδου είναι η συμπίεση δεδομένων, π.χ. φωτογραφιών, μειώνοντας τις διαστάσεις τους και η μετέπειτα επανάκτηση τους με ελάχιστη απώλεια πληροφορίας.

3.3.1 Περιγραφή της μεθόδου

Βήμα 1: Δεδομένα

Σε αυτό το απλό παράδειγμα θα χρησιμοποιήσουμε το 2-διάστατο σύνολο δεδομένων του προηγούμενου παραδείγματος, ώστε να είναι δυνατή η γραφική αναπαράσταση των αποτελεσμάτων κάθε βήματος της μεθόδου.

Βήμα 2: Κανονικοποίηση

Για να πετύχουμε τη βέλτιστη απόδοση της μεθόδου θα πρέπει όλες οι διαστάσεις να ανήκουν στην ίδια τάξη μεγέθους. Αυτό το πετυχαίνουμε κανονικοποιώντας τα δεδομένα, διαιρώντας τη κάθε διάσταση με το δειγματικό της μέσο. Αυτή η τακτική παράγει ένα σύνολο δεδομένων με μέση τιμή μηδέν.

Βήμα 3: Υπολογισμός του πίνακα συνδιασποράς

Υπολογίζουμε τον πίνακα συνδιασποράς όπως συζητήσαμε στην παράγραφο 2.1.4. Αφού τα δεδομένα μας είναι δισδιάστατα ο πίνακας έχει διαστάσεις 2 × 2. Από τα δεδομένα προκύπτει ο εξής πίνακας συνδιασποράς:

$$cov = \begin{pmatrix} 0.61655 & 0.61544 \\ 0.61544 & 0.71655 \end{pmatrix}$$

Επειδή τα στοιχεία στη διαγώνιο του πίνακα συνδιασποράς είναι θετικά, ξέρουμε πως όσο αυξάνεται η μία συνιστώσα του δείγματος αυξάνει και η δεύτερη.

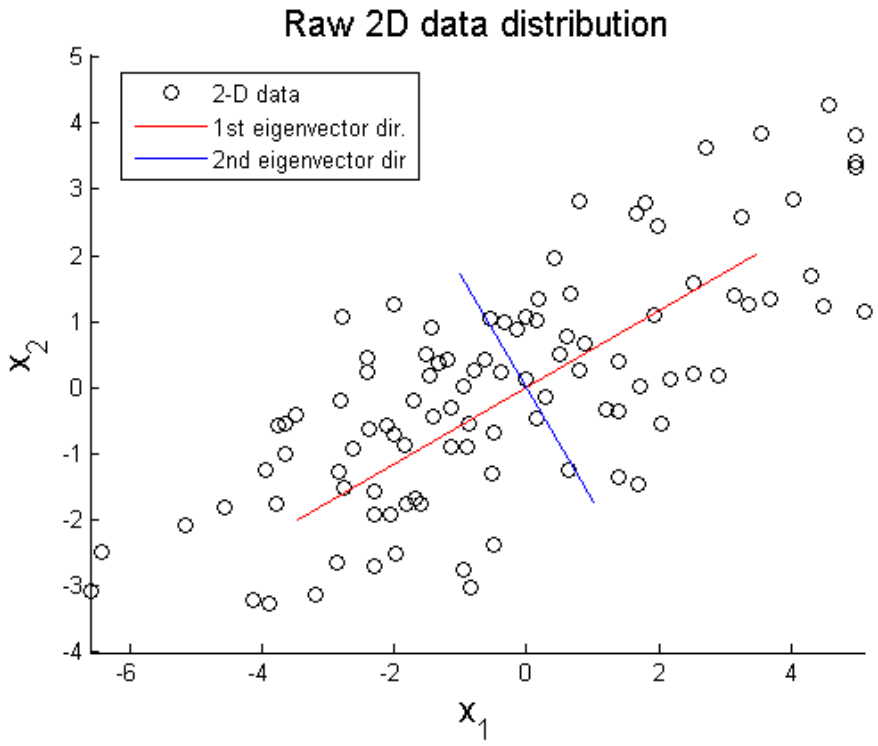
Βήμα 4: Υπολογισμός ιδιοδιανυσμάτων και ιδιοτιμών του πίνακα συνδιασποράς

Ο πίνακας συνδιασποράς είναι τετραγωνικός, επομένως μπορούμε να υπολογίσουμε τις ιδιοτιμές λ και τα ιδιοδιανύσματαv

$$\lambda = \begin{pmatrix} 0.04908 \\ 1.28402 \end{pmatrix}$$

$$v = \begin{pmatrix} -0.73518 & -0.67787 \\ 0.67787 & -0.73518 \end{pmatrix}$$

Είναι σημαντικό να τονίσουμε ότι για τη μέθοδο PCA είναι απαραίτητος ο υπολογισμός των μοναδιαίων ιδιοδιανυσμάτων. Για αυτό το λόγο στο συγκεκριμένο παράδειγμα τα ιδιοδιανύσματα που υπολογίσαμε έχουν μήκος 1.



Εικόνα 4: Τα ιδιοδιανύσματα τα οποία δείχνουν τις διευθύνσεις με τη μεγαλύτερη διασπορά.

Στην εικόνα 4 μπορούμε να δούμε τη φυσική σημασία των ιδιοδιανυσμάτων. Αρχικά τα δεδομένα μας όπως περιμέναμε και από τον πίνακα συνδιασποράς αυξάνουν μαζί και στις δύο διαστάσεις. Φαίνεται από το σχήμα ότι τα ιδιοδιανύσματα μπορούν και περιγράφουν πολύ ικανοποιητικά τα δεδομένα μας. Αυτές οι δυο ευθείες, κάθετες η μία στην άλλη έχουν διευθύνσεις τις δύο διευθύνσεις στις οποίες υπάρχει η μεγαλύτερη διασπορά των δεδομένων.

Με αυτή τη μέθοδο καταφέραμε να δημιουργήσουμε δύο ευθείες που είναι αρκετές για να περιγράψουμε τα δεδομένα. Τα επόμενα βήματα περιγράφουν το πως θα εκφράσουμε τα δεδομένα με βάση αυτές τις ευθείες.

Βήμα 5: Διαλέγοντας τις κύριες συνιστώσες

Σε αυτό το βήμα εισάγουμε την έννοια της συμπίεσης των δεδομένων και της μείωσης διαστάσεων. Αν παρατηρήσουμε τις ιδιοτιμές από το προηγούμενο βήμα βλέπουμε ότι έχουν πολύ διαφορετικές τιμές. Θα λέμε ότι το ιδιοδιάνυσμα με την *μεγαλύτερη* ιδιοτιμή είναι η *κύρια συνιστώσα* του συνόλου δεδομένων. Στηνεικόνα 4 η κύρια συνιστώσα είναι το ιδιοδιάνυσμα στη διεύθυνση με το μεγαλύτερο βαθμό διασποράς των δεδομένων.

Όποτε μόλις υπολογίσουμε τα ιδιοδιανύσματα τα κατατάσσουμε με βάση τις αντίστοιχες ιδιοτιμές από τη μικρότερη στη μεγαλύτερη. Με αυτό τον τρόπο έχουμε τις συνιστώσες με σειρά σημαντικότητας. Από αυτές κρατάμε όσες χρειαζόμαστε με βάση το πόση πληροφορία θέλουμε να διατηρήσουμε στοκαινούριο συμπιεσμένο σύνολο δεδομένων. Δηλαδή αν έχουμε έναn-διάστατο σύνολο δεδομένων, από τα nιδιοδιανύσματα επιλέγουμε τα pόπου $p<n$για να κατασκευάσουμε ένα καινούριο p-διάστατο σύνολο δεδομένων. Στο παράδειγμα μας είναι φανερό η συνιστώσα με τη μεγαλύτερη ιδιοτιμή $\begin{pmatrix}-0.67787\\-0.73518\end{pmatrix}$ είναι αρκετή για να περιγράψουμε τα δεδομένα.Ο πίνακας με τις σημαντικότερες ιδιοτιμές ονομάζεται *χαρακτηριστικό διάνυσμα,* το οποίο είναι ένας πίνακας $n \times p$.

Βήμα 6: Κατασκευάζοντας το νέο σύνολο δεδομένων

Το τελευταίο βήμα της μεθόδου είναι και το πιο εύκολο. Ουσιαστικά προβάλουμε το αρχικό σύνολο δεδομένων πάνω στο χαρακτηριστικό διάνυσμα. Έστω **Y** το καινούριο p-διάστατο σύνολο δεδομένων, **C**το χαρακτηριστικό διάνυσμα, και **X** το αρχικό σύνολο δεδομένων. Το καινούριο σύνολο προκύπτει ως εξής:

$$\mathbf{Y} = \mathbf{C}^{\boldsymbol{T}}\mathbf{X}^{\boldsymbol{T}}$$

3.3.2 Ανάκτηση των αρχικών δεδομένων

Αφού καταφέραμε να συμπιέσουμε το αρχικό n-διάστατο σύνολο δεδομένων σε ένα p-διάστατο, πολλές φορές χρειάζεται να τα ανακτήσουμε.

Είδαμε ότι

$$\mathbf{Y} = \mathbf{C}^{\boldsymbol{T}}\mathbf{X}^{\boldsymbol{T}}$$

Επομένως

$$\mathbf{X}^{\boldsymbol{T}} = [\mathbf{C}^{\boldsymbol{T}}]^{-\mathbf{1}}\mathbf{Y}$$

Όμως

$$[\mathbf{C}^{\boldsymbol{T}}]^{-\mathbf{1}} = \mathbf{C}$$

Οπότε τελικά

$$\mathbf{X}^{\boldsymbol{T}} = \mathbf{CY}$$

Η τελευταία εξίσωση μας δίνει το αρχικό σύνολο δεδομένων κανονικοποιημένο επειδή τα ιδιοδιανύσματα που χρησιμοποιήσαμε είναι κανονικοποιημένα. Για να πάρουμε πίσω τα δεδομένα στην σωστή κλίμακα πρέπει να προσθέσουμε την μέση τιμή με την οποία τα διαιρέσαμε στα προηγούμενα βήματα έστω **M.** Το αρχικό σύνολο δεδομένων δίνεται από τον τύπο:

$$\mathbf{X}^{\boldsymbol{T}} = \mathbf{CY} + \mathbf{M}$$

Από τον τελευταίο τύπο φαίνεται πως όσα περισσότερα ιδιοδιανύσματα περιέχει το χαρακτηριστικό διάνυσμα **C**, τόσο περισσότερη ακρίβεια θα έχουμε στην ανάκτηση των αρχικών δεδομένων.

Κεφάλαιο 4 Isomap

4.1 Εισαγωγή

Σε πολλά προβλήματα με μεγάλο όγκο πολυδιάστατων δεδομένων όπως η μελέτη του κλίματος του πλανήτη, τα αστρικά φάσματα, ή οι κατανομές του ανθρώπινου γονιδιώματος, συχνά χρειαζόμαστε τεχνικές που μας επιτρέπουν τη μείωση των διαστάσεων των δεδομένων σε ουσιαστικές και κρυφές δομές λιγότερων διαστάσεων. Σε αντίθεση με την τεχνική PCA, η Isomap είναι μια μέθοδος η οποία εντοπίζει τους βασικούς μη γραμμικούς βαθμούς ελευθερίας πίσω από πολύπλοκα δεδομένα όπως ανθρώπινη γραφή ή αναγνώριση ανθρώπινων προσώπων υπό γωνία [7]. Σε αντίθεση με άλλες τεχνικές μείωσης διαστάσεων, η μέθοδος Isomap υπολογίζει αποτελεσματικά μια ολικά βέλτιστη λύση, και, για μια σημαντική κλάση πολλαπλοτήτων συγκλίνει ασυμπτωτικά στηνδομή χαμηλότερης διάστασης πολλαπλοτήτων που ψάχνουμε.

4.2 Μαθηματικό Υπόβαθρο

4.2.1 Μη γραμμικές τεχνικές μείωσης διαστάσεων

Οι κλασσικές τεχνικές μείωσης διαστάσεων, όπως η PCA, είναι απλές στην υλοποίηση, δεν έχουν μεγάλο υπολογιστικό κόστος και εντοπίζουν την πραγματική δομή των δεδομένων πάνω σε ένα γραμμικό υπόχωρο του αρχικού πολυδιάστατου χώρου. Η PCA προβάλει τα δεδομένα σε μία δομή με λιγότερες διαστάσεις από το αρχικό σύνολο η οποία διατηρεί όσο το δυνατόν καλύτερα την αρχική διασπορά των δεδομένων. Παρολαυτά, πολλά σύνολα δεδομένων κρύβουν μη γραμμικές δομές τις οποίες κλασσικές γραμμικές όπως η PCA δεν μπορούν να εντοπίσουν [14, 15, 16, 17]. Για παράδειγμα η PCA αδυνατεί να εντοπίσει τη δομή του συνόλου δεδομένων Swiss-Roll.

Η μέθοδος που περιγράφεται στο παρόν κεφάλαιο η οποία συνδυάζει όλα εκείνα τα στοιχεία που κάνουν την PCA αποτελεσματική - μικρό υπολογιστικό κόστος, ολική βελτιστοποίηση, ασυμπτωτική σύγκλιση – με την επιπλέον ευελιξία μίας μη

γραμμικής μεθόδου. Αν κάνουμε χρήση μιας γραμμικής μεθόδου όπως η PCA, που στηρίζεται στην ευκλείδεια απόσταση μεταξύ των σημείων του συνόλου, πάνω στην δομή ενός συνόλου όπως το Swiss-Roll, θα οδηγηθούμε σε λάθος συμπεράσματα. Αντίθετα η χρήση της γεωδαιτικής απόστασης μεταξύ των δύο σημείων περιγράφει πολύ καλύτερα τη μεταξύ τους ομοιότητα.

4.2.2 Πολυδιάστατη Κλιμακοποίηση (multidimensional scaling ή MDS)

Πριν προχωρήσουμε στην ανάλυση της μεθόδου Isomap, είναι απαραίτητη για την κατανόηση της, μία εισαγωγή στη μέθοδο της πολυδιάστατης κλιμακοποίησης. Η μέθοδος της πολυδιάστατης κλιμακοποίησης (MDS) [18] αναπαριστά μια συλλογήαπό μη γραμμικές τεχνικές οι οποίες απεικονίζουν τα πολυδιάστατα δεδομένα σε μια λιγότερων διαστάσεων αναπαράσταση, ενώ παράλληλα διατηρούν τις αποστάσεις μεταξύ των σημείων όσο το δυνατόν περισσότερο. Η ποιότητα της τελικής απεικόνισης υπολογίζεται από μια συνάρτηση κόστους η οποία μετράει τη διαφορά μεταξύ των αποστάσεων ανά ζεύγη στο αρχικό σύνολο και στο τελικό. Δύο δημοφιλείς συναρτήσεις κόστους είναι η συνήθης συνάρτηση κόστους και η συνάρτηση Sammon [36]. Η συνήθης συνάρτηση κόστους δίνεται από την εξίσωση:

$$\Phi(Y) = \sum_{ij} (\|x_i - x_j\| - \|y_i - y_j\|)^2$$

όπου $\|x_i - x_j\|$ είναι η ευκλείδια απόσταση μεταξύ των σημείων x_i και x_j στο αρχικό σύνολο και $\|y_i - y_j\|$ είναι η ευκλείδια απόσταση μεταξύ των σημείων y_i και y_j στο τελικό. Η συνάρτηση Sammon δίνεται από την εξίσωση:

$$\Phi(Y) = \frac{1}{\sum_{ij}\|x_i - x_j\|} \sum_{i \neq j} \frac{(\|x_i - x_j\| - \|y_i - y_j\|)^2}{\|x_i - x_j\|}$$

Η διαφορά τους έγκειται στο ότι η συνάρτηση Sammon δίνει περισσότερη έμφαση στη διατήρηση αποστάσεων που ήταν εξ' αρχής μικρές. Η ελαχιστοποίηση της συνάρτησης κόστους επιτυγχάνεται με διάφορες μεθόδους, όπως η εύρεση των ιδιοτιμών και των ιδιοδιανυσμάτων του πίνακα των αποστάσεων μεταξύ των σημείων του αρχικού συνόλου ή άλλες αριθμητικές μέθοδοι [18].

4.2.3 Isomap

Η τεχνική εκμάθησης πολλαπλοτήτων Isomap [19] προσπαθεί να βρει την εσωτερική γεωμετρία του εκάστοτε προβλήματος, όπως αυτή περιγράφεται από τις γεωδαιτικές αποστάσεις κάθε ζεύγους του συνόλου δεδομένων. Η δυσκολία εδώ βρίσκεται στον υπολογισμό της γεωδαιτικής απόστασης μεταξύ σημείων που βρίσκονται μακριά μεταξύ τους. Όταν δύο σημεία είναι πολύ κοντά το ένα στο άλλο μπορούμε εύκολα να προσεγγίσουμε την γεωδαιτική απόσταση χρησιμοποιώντας την ευκλείδεια απόσταση. Σε αντίθετη περίπτωση προσεγγίζουμε την γεωδαιτική απόσταση ως έναν «περίπατο» μεταξύ κοντινών σημείων. Αυτές οι προσεγγίσεις επιτυγχάνονται υπολογίζοντας το συντομότερο μονοπάτι στο γράφημα που σχηματίζουν τα δεδομένα.

Ο αλγόριθμος Isomap αποτελείται από τρία. Το πρώτο βήμα καθορίζει ποια σημεία του γραφήματος γειτονεύουν στην πολλαπλότητα έστω *M*, με βάση τις αποστάσεις $d_x(i,j)$ μεταξύ του ζεύγους σημείων *i,j*. Το δεύτερο βήμα είναι να κατασκευάσουμε το γράφημα *G*, το οποίο προκύπτει από το αρχικό σύνολο δεδομένων. Κατασκευάζουμε το γράφημα, συνδέοντας κάθε σημείο με τους *K* πλησιέστερους γείτονες, λαμβάνοντας υπόψη μια σταθερή ακτίνα ϵ. Αυτές οι συνδέσεις αναπαριστώνται σαν ένα γράφημα *G*με βάρη, πάνω στα σημεία του αρχικού συνόλου δεδομένων, όπου τα βάρη παίρνουν τις τιμές $d_x(i,j)$. Σε αυτό το βήμα ο αλγόριθμος προσεγγίζει τις τιμές των γεωδεσιακών αποστάσεων $d_M(i,j)$ μεταξύ κάθε σημείου της πολλαπλότητας *M* υπολογίζοντας το συντομότερο μονοπάτι για την απόσταση $d_G(i,j)$πάνω στο γράφημα *G*.

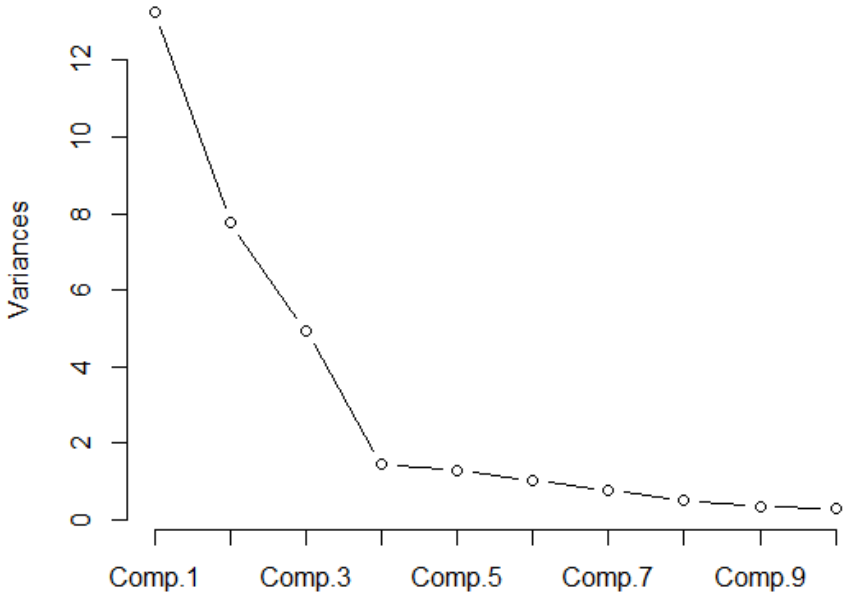
Το τελευταίο βήμα είναι να εφαρμόσουμε την μέθοδο MDS στον πίνακα με τις αποστάσεις$D_G = \{d_G(i,j)\}$, των σημείων πάνω στο γράφημα *G*. Με αυτό τον τρόπο πετυχαίνουμε μια αποτύπωση των δεδομένων σε ένα χώρο *Y*μικρότερων διαστάσεων που διατηρεί όσο τον δυνατόν καλύτερα το την εσωτερική γεωμετρία του χώρου των αρχικών δεδομένων. Ο χώρος *Y*επιλέγεται έτσι ώστε να ελαχιστοποιείται η συνάρτηση κόστους:

$$E = \|\tau(D_G) - \tau(D_Y)\|_{L^2}$$

Όπου D_Yο πίνακας των ευκλείδειων αποστάσεων $\{d_Y(i,j) = \|y_i - y_j\|\}$ και $\|A\|_{L^2}$η L^2 νόρμα πινάκων $\sqrt{\sum_{i,j} A_{ij}^2}$. Ο τελεστής *τ* μετατρέπει τις αποστάσεις σε

εσωτερικά γινόμενα που χαρακτηρίζουν με μοναδικό τρόπο τη γεωμετρία των δεδομένων. Για την ακρίβεια ο τελεστής τ ορίζεται ως $\tau(D) = -HSH/2$, όπου S είναι ο πίνακας των αποστάσεων στο τετράγωνο $\{S_{ij} = D_{ij}^2\}$ και H είναι ο πίνακας «κεντραρίσματος»(centeringmatrix να το αλλάξω αυτό) $\{H_{ij} = \delta_{ij} - 1/N\}$.

Γνωρίζουμε ότι έχουμε εντοπίσει την αληθινή δομή του χώρου των δεδομένων όταν μειωθεί ικανοποιητικά το σφάλμα ενώ αυξάνουμε τις διαστάσεις του χώρου Y(elbowrule), όπως φαίνεται στην εικόνα 5.

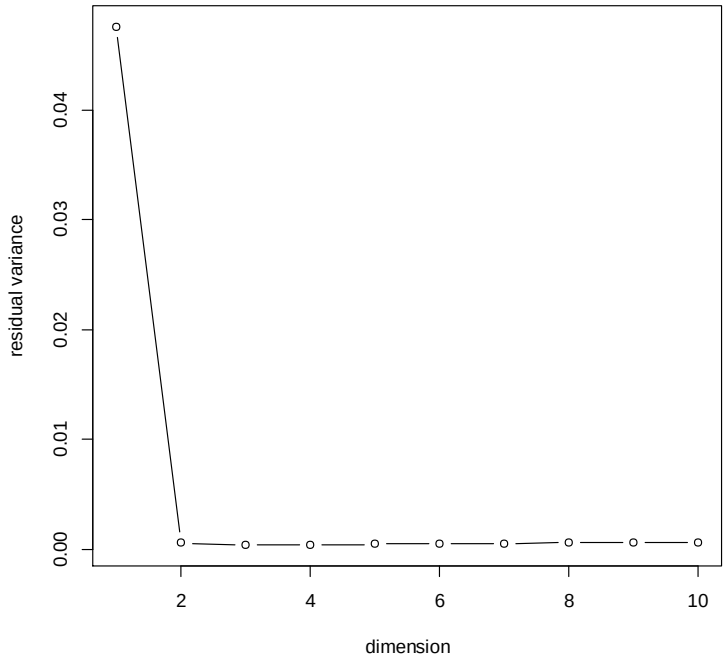


Εικόνα 5: Με βάση το κανόνα elbow-rule επιλέγονται τέσσερεις συνιστώσες για την περιγραφή του συνόλου δεδομένων. Τα δεδομένα που χρησιμοποιήθηκαν είναι ένα τυχαίο δείγμα από το σύνολο δεδομένων που εξετάστηκε στο κεφάλαιο 8.

4.2.4 Εφαρμογή

Σε αυτή την εφαρμογή εφαρμόζουμε τη μέθοδο Isomap για να προσδιορίσουμε την αληθινή γεωμετρία του συνόλου δεδομένων swiss-roll.

Από την εικόνα 6 είναι φανερό πως αρκεί μια μόνο συνιστώσα για περιγράψουμε τα δεδομένα, ενώ το σύνολο μπορεί να περιγραφεί πλήρως από 2 νέες διαστάσεις.



Εικόνα 6: 2 μόνο διαστάσεις μπορούν να περιγράψουν το σύνολο δεδομένωνswiss-rollμε τη χρήση της μεθόδου Isomap.

Κεφάλαιο 5 Diffusionmaps

5.1 Εισαγωγή

Όπως έχουμε δει μέχρι τώρα, σκοπός των τεχνικών μείωσης διαστάσεων είναι η αλλαγή της αναπαράστασης ενός αρχικού συνόλου δεδομένων που αποτελείται από ένα μεγάλο πλήθος μεταβλητών σε ένα νέο σύνολο που αποτελείται από πολύ λιγότερες μεταβλητές. Το νέο αυτό σύνολο δεδομένων πρέπει φυσικά να διατηρεί όση περισσότερη πληροφορία γίνεται από το αρχικό σύνολο. Ένα ανάλογο πρόβλημα με αυτό της μείωσης διαστάσεων είναι το πρόβλημα της εύρεσης της υποκείμενης δομής των δεδομένων δηλαδή του χώρου των πολλαπλοτήτων. Ο σκοπός εδώ είναι η εξαγωγή σημαντικών χαρακτηριστικών του συνόλου δεδομένων, με τελικό στόχο την κατανόηση του αρχικού φαινομένου που ευθύνεται για τη δημιουργία τους.

Για τη λύση των παραπάνω δύο προβλημάτων έχουν δημιουργηθεί πολλές μέθοδοι εξόρυξης δεδομένων και εκμάθησης μηχανών και πολλαπλοτήτων, οι οποίες στηρίζονται στο γράφο του δεδομένων. Τα γραφήματα με βάρη είναι ένας πολύ εύκολος τρόπος για αναπαραστήσουμε την γεωμετρία των δεδομένων με βάση τις αποστάσεις ή τις αλληλεπιδράσεις μεταξύ των παρατηρήσεων. Όταν συνδυάζονται με τεχνικές αλυσίδων Markov, οι μέθοδοι γραφημάτων μπορούν να δώσουν πολύ καλά αποτελέσματα. Σε προβλήματα ταξινόμησης και ομαδοποίησης, μέθοδοι που κάνουν χρήση ενός τυχαίου περίπατου πάνω στον γράφο των δεδομένων, έχουν ανακαλύψει με επιτυχία την δομή σύνθετης γεωμετρίας.

Σε αυτό το κεφάλαιο θα αναλύσουμε την μέθοδο diffusionmaps [21], μια μη-γραμμική τεχνική εύρεσης της δομής των δεδομένων μέσω του υπολογισμού ενός χαμηλής διάστασης χώρου πολλαπλοτήτων, ο οποίος διατηρεί ικανοποιητικά την πληροφορία του υψηλής διάστασης χώρου των δεδομένων. Ο αλγόριθμος αυτός είναι σχετικά απλός και με μικρό υπολογιστικό κόστος. Περιλαμβάνει ένα πρόβλημα ιδιοδιανυσμάτων και κάποιους τοπικούς υπολογισμούς. Ένα άλλο σημαντικό πλεονέκτημα της μεθόδου είναι το ότι δεν επηρεάζεται από τυχών θόρυβο στα δεδομένα.

Τέλος, η μέθοδος diffusionmaps έχει πολλές εφαρμογές στην αντιστοίχηση σχημάτων, την ανάλυση γονιδιώματος, αναγνώριση και απομόνωση ήχων, 3Dμοντελοποίηση, φασματική ανάλυση κ.α.

5.2 Ο αλγόριθμος

Όπως είδαμε όλες οι τεχνικές μείωσης διαστάσεων έχουν σαν στόχο την αναπαράσταση ενός συνόλου δεδομένων $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(k)}\} \in \mathbb{R}^{\boldsymbol{n}}$όπου $\mathbf{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)})$και i = 1, 2, …,k από ένα $\mathbf{Y} = \{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(k)}\} \in \mathbb{R}^{\boldsymbol{m}}$όπου $\boldsymbol{y}^{(j)} = (y_1^{(j)}, y_2^{(j)}, \dots, y_m^{(j)})$και j = 1, 2, …,kέτσι ώστε $m \ll n$.

Έστω τώρα, ότι στην πραγματικότητα ότι τα $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(k)} \in M$ όπου το Mείναι μια πολλαπλότητα του $\mathbb{R}^{\boldsymbol{n}}$. Μία πολλαπλότητα είναι ένας αφηρημένος μαθηματικός χώρος του οποίου κάθε σημείο έχει μια περιοχή γύρω από αυτό με ιδιότητες ενός ευκλείδιου χώρου. Δωθέντων τώρα των kπαρατηρήσεων $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(k)}$, κατασκευάζουμε το γράφο τους, χρησιμοποιώντας τις παρατηρήσεις σαν κόμβους και συνδέοντας τους με τα αντίστοιχα βάρη. Ο πίνακας που αναπαριστά αυτό το γράφο είναι γνωστός ως Λαπλασιανός πίνακας. Τα ιδιοδιανύσματα του Λαπλασιανού πίνακα είναι τελικά οι συντεταγμένες της πολλαπλότητας που αναζητάμε [20]. Ο αλγόριθμος μπορεί να αναλυθεί στα παρακάτω βήματα:

Βήμα 1. Κατασκευή του πίνακα γειτνίασης, **W**, από τον γράφο των δεδομένων. Οι τιμές του **W**είναι τα βάρη των ακμών που ενώνουν τις κορυφές του γράφου και ορίζονται από την παρακάτω συνάρτηση kernel:

$$W_{ij} = e^{\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\varepsilon}\right)}$$

Όπου $\|.\|$ η Ευκλείδια νόρμα και ε μία κατάλληλα επιλεγμένη παράμετρος διασποράς.

Βήμα 2. Κατασκευή του $k \times k$ διαγώνιου πίνακα κανονικοποίησης **D** και του Λαπλασιανού πίνακα **L**, οι οποίοι ορίζονται ως εξής:

$$D_{ij} = \sum_{j=1}^{k} W_{ji} \qquad \text{και} \qquad \mathbf{L} = \mathbf{D} - \mathbf{W}.$$

Φαίνεται πως ο πίνακας **L**είναι συμμετρικός.

Βήμα 3. Υπολογισμός των ιδιοδιανυσμάτων και των ιδιοτιμών με επίλυση της εξίσωσης:

$$\mathbf{Ly} = \lambda \mathbf{Dy}$$

Όπου $\mathbf{y} \in \mathbb{R}^{k}$είναι οι στήλες διανύσματα του ζητούμενου πίνακα **Y**. Τα ιδιοδιανύσματα $\mathbf{y_0}, \mathbf{y_1}, \ldots, \mathbf{y_{k-1}}$ που αντιστοιχούν στις διατεταγμένες ιδιοτιμές $0 = \lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_{\kappa-1}$ είναι οι λύσεις της παραπάνω εξίσωσης.

Βήμα 4. Το ιδιοδιάνυσμα $\mathbf{y_0}$ που αντιστοιχεί στην ιδιοτιμή $\lambda_0 = 0$ απορείπτεται και χρησιμοποιούμε τα υπόλοιπα *m* ιδιοδιανύσματα για την αναπαράσταση του αρχικού *n*-διάστατου χώρου από το ακόλουθο *diffusionmap*:

$$\boldsymbol{\Psi_m} : \mathbf{x}^{(i)} \rightarrow (y_1^{(j)}, y_2^{(j)}, \quad \ldots, y_m^{(j)})$$

Όπου $\boldsymbol{\Psi_m}$: $\mathbb{R}^{n} \rightarrow \mathbb{R}^{m}$δηλαδή $\boldsymbol{\Psi_m}$: **X** → **Y**όπως θα έπρεπε [20, 22].

5.3 Επιλογή της παραμέτρου *ε*

Στην προηγούμενη παράγραφο είδαμε πως τα βάρη ορίζονται ως $W_{ij} = e^{\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\varepsilon}\right)}$, επομένως η επιλογή της παραμέτρου *ε* είναι πολύ σημαντική για κατασκευή τους. Η παράμετρος *ε* εξαρτάται από τα αρχικά δεδομένα και θα μπορούσαμε να την δούμε σαν τη παράμετρο κλίμακας του παραπάνω kernel. Μη σωστή επιλογή αυτής της παραμέτρου μπορεί να οδηγήσει τον αλγόριθμο σε μη σύγκλιση. Πρέπει να τονιστεί ότι αναζητάμε ένα εύρος τιμών και όχι μία μοναδική τιμή.

Διάφορες μέθοδοι για τον υπολογισμό της παραμέτρου έχουν προταθεί και ο Lafon[21] επιλέγει το *ε* να είναι η μέση μη μηδενική απόσταση μεταξύ οποιονδήποτε δύο παρατηρήσεων, $\mathbf{x_i}, \mathbf{x_j}$ του αρχικού συνόλου, δηλαδή:

$$\varepsilon = \frac{1}{k}\sum_{i=1}^{k} \min_{j:x_j \neq x_i} \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2.$$

Ο κύριος λόγος που επιλέγει το ε κατά αυτό τον τρόπο είναι να εξασφαλίσει πως θα υπάρχει διάχυση πληροφορίας μεταξύ των παρατηρήσεων και ταυτόχρονα αυτή να μην είναι ολική.

5.4 Τυχαίος περίπατος και εξίσωση διάχυσης

Ας υποθέσουμε τώρα ότι θέλουμε να μάθουμε την ακριβή περιγραφή μιας περιοχής. Ένας απλοϊκός τρόπος να το πετύχουμε είναι να αναθέσουμε σε μία ομάδα ανθρώπων και να τους αναθέσουμε να περιηγηθούν στην περιοχή αυτή ξεκινώντας από κοινή αφετηρία. Μετά από κάποιο χρονικό διάστημα το πιθανότερο είναι πως οι τοποθεσίες με μικρή απόσταση από την αφετηρία θα είναι και αυτές με τη μεγαλύτερη επισκεψιμότητα. Όμοια σε ένα σύνθετο πολυδιάστατο σύνολο δεδομένων θα μπορούσαμε να ξεκινήσουμε από ένα σημείο και εξερευνήσουμε τη δομή του ξεκινώντας έναν τυχαίο περίπατο πάνω στα δεδομένα. Ο τρόπος με τον οποίο μεταβαίνουμε από ένα σημείο στο επόμενο εξαρτάται από την πιθανότητα μετάβασης που ορίζει αυτός ο τυχαίος περίπατος και είναι ανάλογος με τον τρόπο πουδιαχέεται η θερμότητα σε κάποιο υλικό. Οδηγούμαστε λοιπόν σε μια καινούρια έννοια, αυτή της *απόστασης διάχυσης*. Η απόσταση διάχυσηςείναι μεγάλη σε σημεία που συνδέονται με μεγάλο βάρος, η εναλλακτικά έχουν υψηλή πιθανότητα μετάβασης, και μικρή σε σημεία που συνδέονται με μικρά βάρη. Συνεπώς η απόσταση διάχυσης αντανακλά την συνδεσιμότηταμεταξύ των σημείων του αρχικού συνόλου δεδομένων η οποία με τη σειρά της μας οδηγεί σε μία διαδικασία διάχυσης που αναδεικνύει την εσωτερική γεωμετρία του προβλήματος [21, 23].

5.5 Εφαρμογή

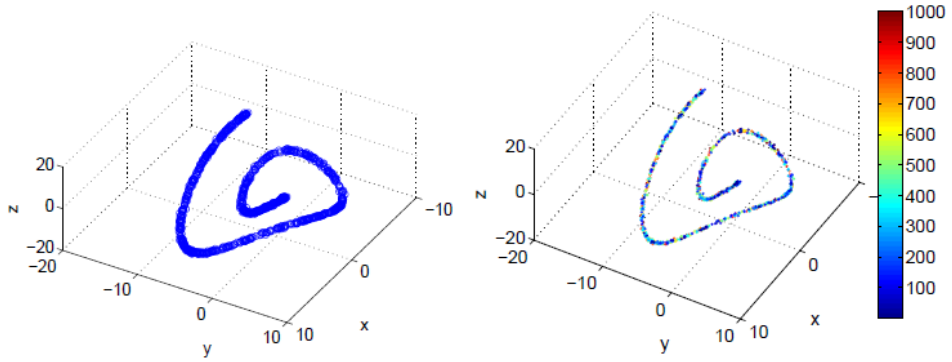
Τα δεδομένα για το παράδειγμα δημιουργούνται ως εξής:

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = (1 + a_i) \begin{pmatrix} sina_i \\ cosa_i \\ sin2a_i \end{pmatrix} + 0.1 \times w$$

Όπου $a_i = 10(1 - r_i)^{1.5}$, $i = 1, 2, \ldots, 1000$, και τα r_i παίρνουν τιμές από την ομοιόμορφη κατανομή, ενώ η μεταβλητή w ακολουθεί την κανονική κατανομή με μέση τιμή 0 και διασπορά 1.

Χρησιμοποιώντας τον αλγόριθμο diffusionmapsυπολογίστηκαν τα έξι μεγαλύτερα ιδιοδιανύσματα. Όμοια, χρησιμοποιώντας τον αλγόριθμο PCA υπολογίσαμε τις έξι κύριες συνιστώσες των δεδομένων.

Στην εικόνα 8βλέπουμε στο πρώτο γράφημα τα δεδομένα και στο δεύτερο τα ίδια δεδομένα χρωματισμένα ανάλογα με τη σειρά δημιουργίας τους που φαίνεται από το δείκτη i.



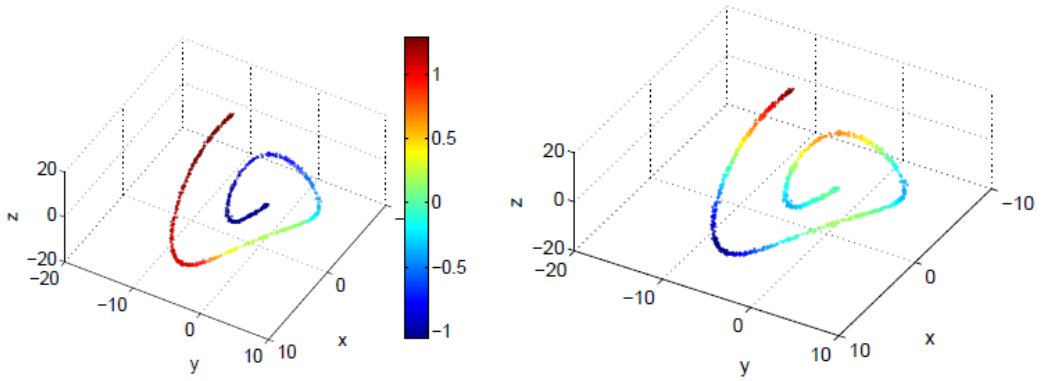
(A) (B)

Εικόνα 8 Στην εικόνα (Α) φαίνονται τα δεδομένα της εφαρμογής 5.4 στον τρισδιάστατο χώρο. Στην εικόνα (Β) φαίνονται τα δεδομένα της εφαρμογής 5.4 χρωματισμένα ανάλογα με τη σειρά της δημιουργίας τους.

Φαίνεται πως τα δεδομένα παρότι είναι σχεδιασμένα στον τρισδιάστατο χώρο αρκεί μία μόνο παράμετρος για τα να τα περιγράψουμε, η θέση τους πάνω στο σπειροειδές αυτό σχήμα.

Στην εικόνα 9 χρησιμοποιούμε το δεύτερο ιδιοδιάνυσμα του αλγόριθμου diffusionmap για να ταξινομήσουμε τα δεδομένα στο αριστερή εικόνα και την πρώτη κύρια συνιστώσα που προέκυψε από τον αλγόριθμο PCA για να ταξινομήσουμε τα

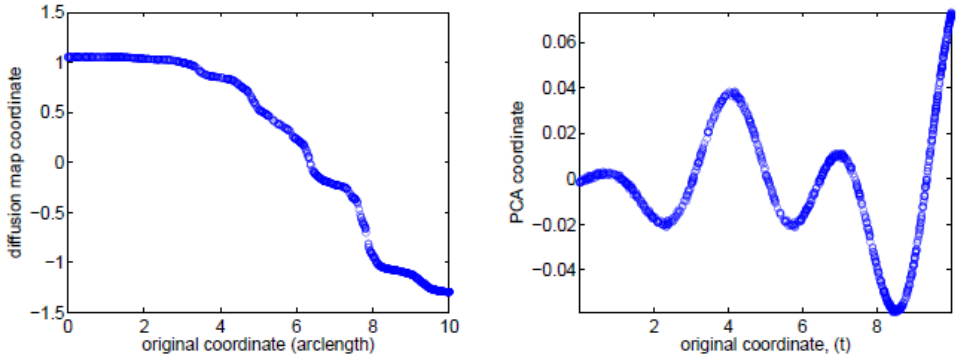
δεδομένα στο δεξί γράφημα. Βλέπουμε πως ο αλγόριθμος diffusionmap εντόπισε με επιτυχία την μοναδική παράμετρο που χρειαζόμαστε για να περιγράψουμε τα δεδομένα σε αντίθεση με τον αλγόριθμο PCA.



(A) (B)

Εικόνα 6. Στην εικόνα (A) φαίνεται πως ο αλγόριθμος diffusionmap κατάφερε να εντοπίσει την δομή των δεδομένων, χρωματίζοντας με κοντινές αποχρώσεις τις παρατηρήσεις οι οποίες βρίσκονται κοντά πάνω στο σύνολο. Αντίθετα στην εικόνα (B) βλέπουμε απέτυχε την σωστή ταξινόμηση των δεδομένων χρωματίζοντας με κοντινές αποχρώσεις σημεία που απέχουν κατά πολύ πάνω στη συγκεκριμένη πολλαπλότητα.

Τέλος στο αριστερό γράφημα στην εικόνα 10 έχει γίνει το γράφημα της μοναδικής παραμέτρου που χρειάζεται για την αναπαράσταση των δεδομένων, ουσιαστικά πρόκειται για το μήκος τόξου, σε σχέση με το δεύτερο ιδιοδιάνυσμα που προέκυψε από τον αλγόριθμο diffusionmaps. Όμοια έγινε η γραφική παράσταση του μήκους τόξου σε σχέση με την μεγαλύτερη κύρια συνιστώσα που προέκυψε από την μέθοδο PCA.



(A) (B)

Εικόνα 7. Στην εικόνα (Α) αναπαριστάται η σχέση του πρώτου ιδιοδιανύσματος που προέκυψε από τον αλγόριθμο diffusionmapως προς το μήκος τόξου του σχήματος της εικόνας 8.Α.Στην εικόνα (Β) αναπαριστάται η σχέση της πρώτης συνιστώσας που προέκυψε από τον αλγόριθμο PCA ως προς το μήκος τόξου του σχήματος της εικόνας 8.Α η οποία δεν είναι ένα προς ένα.

Το μεγαλύτερο ιδιοδιάνυσμα από την μέθοδο diffusionmap σχετίζεται με το μήκος τόξου με μια ένα προς ένα απεικόνιση, σε αντίθεση με την κύρια συνιστώσα της μεθόδου PCA. Τελικά φαίνεται πως η μέθοδος diffusionmap έχει πολύ καλύτερα αποτελέσματα όταν τα δεδομένα είναι μη γραμμικά.

Κεφάλαιο6 Autoencoders

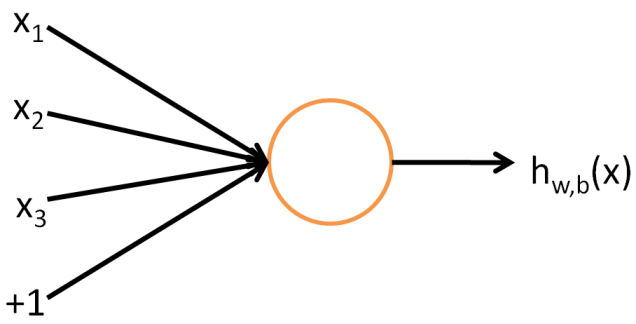
6.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα εξετάσουμε πως μπορούμε να μειώσουμε τις διαστάσεις πολυδιάστατων συνόλων δεδομένων εκπαιδεύοντας ένα τεχνητό νευρωνικό δίκτυο. Αρχικά θα δούμε αναλυτικά τι είναι ένα τεχνητό νευρωνικό δίκτυο και πως μπορεί να εφαρμοσθεί σε προβλήματα τεχνητής μάθησης (εκμάθηση μηχανών). Στη συνέχεια θα αναλύσουμε πως μπορούμε να πετύχουμε τη μείωση διαστάσεων των δεδομένων χρησιμοποιώντας ένα τεχνητό νευρωνικό δίκτυο με ένα μικρό κεντρικό νευρώνα που θα αποτελέσει τελικά την απεικόνιση των δεδομένων σε λιγότερες διαστάσεις [24]. Ο αλγόριθμος απότομης καθόδου, γνωστός και ως αλγόριθμος σύγκλισης με ελάττωση της παραγώγου (gradientdescent) μπορεί να χρησιμοποιηθεί για την βελτιστοποίηση των βαρών του δικτύου, αλλά συγκλίνει ικανοποιητικά μόνο εάν έχουμε επιλέξει τα αρχικά βάρη του δικτύου κοντά στην λύση. Θα περιγράψουμε ακόμα έναν τρόπο για την σωστή επιλογή των αρχικών βαρών, ώστε τελικά να επιτυγχάνεται η επιθυμητή μείωση διαστάσεων.

6.2 Τεχνητά νευρωνικά δίκτυα

Ας υποθέσουμε πως θέλουμε να βρούμε τη σχέση μεταξύ των μεταβλητών X,Y και έχουμε στη διάθεση μας ένα τυχαίο δείγμα παρατηρήσεων από αυτές τις μεταβλητές ($x^{(i)}$, $y^{(i)}$). Με την χρήση τεχνητών νευρωνικών δικτύων μπορούμε να προσεγγίσουμε οποιαδήποτε μη γραμμική απεικόνιση $h_{W,b}(x):X \rightarrow Y$.

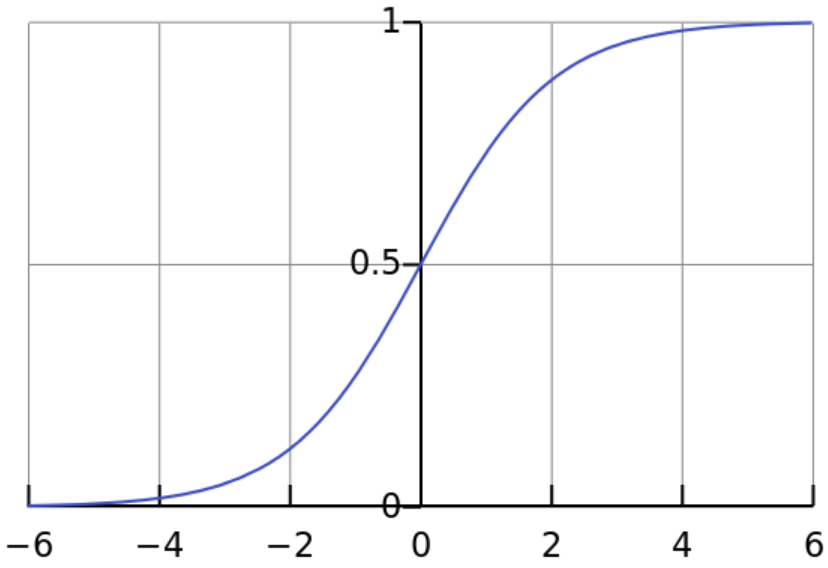
Θα ξεκινήσουμε την περιγραφή των νευρωνικών δικτύων [25] αναλύοντας το απλούστερο δυνατό. Αυτό το νευρωνικό δίκτυο αποτελείται από έναν μόνο νευρώνα και αναπαριστάτε στην Εικόνα 1.



Εικόνα 8Αναπαράσταση ενός τεχνητού νευρώνα.

Ο συγκεκριμένος νευρώνας είναι μια μονάδα υπολογισμού η οποία δέχεται σαν όρισμα τα x_1, x_2, x_3, και την μονάδα ως ένα σταθερό όρο, και επιστρέφει την απεικόνιση $h_{W,b}(x) = f(W^T x) = f(\sum_{i=1}^{3} W_i x_i + b)$ όπου η $f: \mathbb{R} \to \mathbb{R}$ καλείται συνάρτηση ενεργοποίησης. Συνήθως επιλέγουμε ως συνάρτηση ενεργοποίησης τη σιγμοειδή συνάρτηση:

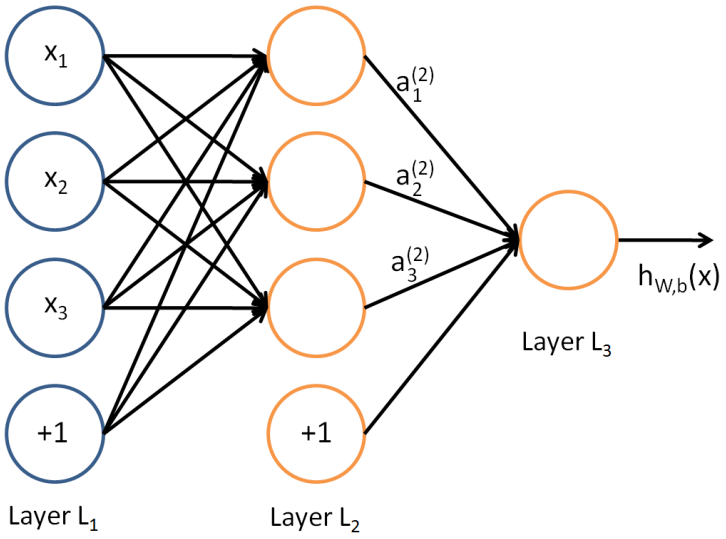
$$f(z) = \frac{1}{1 + e^{(-z)}}$$



Εικόνα 9Γραφική παράσταση της συνάρτησης ενεργοποίησης.

6.2.1 Μοντελοποίηση με τεχνητά νευρωνικά δίκτυα

Ένα νευρωνικό δίκτυο [25] δημιουργείται συνδέοντας πολλούς απλούς τεχνητούς νευρώνες, έτσι ώστε το αποτέλεσμα ενός νευρώνα να είναι το όρισμα για έναν άλλο. Μια τέτοια σύνδεση φαίνεται στην Εικόνα 3.



Εικόνα 10Αναπαράσταση ενός τεχνητού νευρωνικού δικτύου.

Στην Εικόνα 3, το στρώμα νευρώνων L_1 καλείται *στρώμα εισόδου* και το στρώμα νευρώνων L_3 καλείται *στρώμα εξόδου*. Το μεσαίο στρώμα L_2 ονομάζεται *κρυμμένο στρώμα*, καθώς οι τιμές του δεν υπάρχουν στο σύνολο παρατηρήσεων. Οι μονάδες με την τιμή +1 καλούνται *μονάδες μεροληψίας*. Τέλος το παραπάνω νευρωνικό δίκτυο έχει *τρεις μονάδες εισόδου*, *τρεις κρυφές μονάδες* και μια *μονάδα εξόδου*.

Έστω τώρα το n_l ο αριθμός των στρωμάτων στο δίκτυο μας, δηλαδή εδώ $n_l = 3$. Ακόμα συμβολίζουμε το στρώμα l ως L_l, οπότε το στρώμα εισόδου είναι το L_1και το στρώμα εξόδου είναι το L_{n_l}. Ακόμα όταν ορίσαμε τη συνάρτηση $h_{W,b}(x)$είδαμε ότι χρησιμοποιεί τις παραμέτρους $(\boldsymbol{W}, \boldsymbol{b}) = (\boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)}, \boldsymbol{W}^{(2)}, \boldsymbol{b}^{(2)})$ όπου $W_{ij}^{(l)}$ συμβολίζει την παράμετρο ή αλλιώς το βάρος της σύνδεσης μεταξύ της μονάδας i στο στρώμα l και της μονάδας j στο στρώμα $l + 1$. Επομένως στο παράδειγμα μας $\boldsymbol{W}^{(1)} \epsilon \mathbb{R}^{3\times3}$ και $\boldsymbol{W}^{(2)} \epsilon \mathbb{R}^{1\times3}$. Οι μονάδες μερολυψίας δεν έχουν συνδέσεις προς αυτές, αφού πάντα δίνουν αποτέλεσμα +1. Τέλος συμβολίζουμε με s_l τον αριθμό των μονάδων στο στρώμα l, χωρίς να λαμβάνουμε υπόψη τη μονάδα μεροληψίας.

Συμβολίζουμε με $a_i^{(l)}$ το αποτέλεσμα της μονάδας *i* στο στρώμα *l*. Για *l = 1* είναι προφανώς $a_i^{(l)} = x_i$. Αν έχουμε ορίσει τις παραμέτρους $\boldsymbol{W}, \boldsymbol{b}$τότε το τελικό αποτέλεσμα $h_{W,b}(x)$ είναι ένας πραγματικός αριθμός που υπολογίζεται ως εξής:

$$a_1^{(2)} = f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)})$$

$$a_2^{(2)} = f(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_1^{(1)})$$

$$a_3^{(2)} = f(W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3 + b_1^{(1)})$$

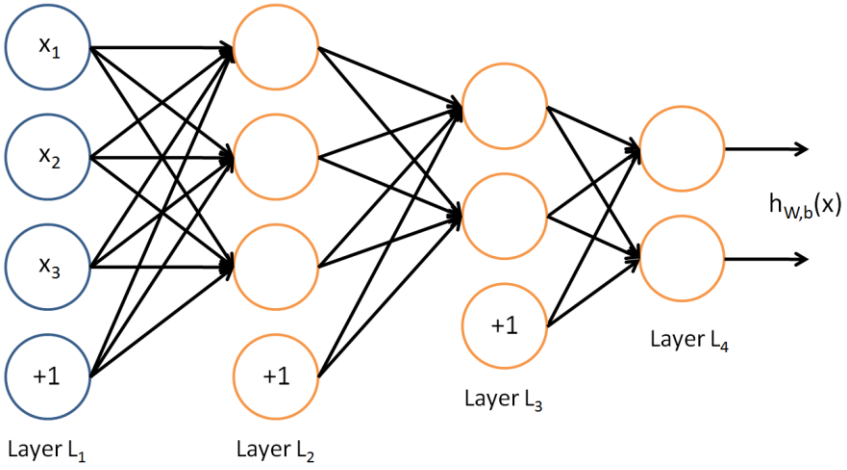
$$h_{W,b}(x) = a_1^{(3)} = f(W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)} + b_1^{(2)})$$

Στη συνέχεια για ευκολία, συμβολίζουμε $z_i^{(l)} = \sum_{j=1}^{n} W_{ij}^{(1)} x_j + b_i^{(1)}$ οπότε τελικά $a_i^{(l)} = f(z_i^{(l)})$.Και οι τελευταίες εξισώσεις γενικεύονται στις παρακάτω:

$$z^{(l+1)} = \boldsymbol{W}^{(l)}a^{(l)} + b^{(l)}$$

$$a^{(l+1)} = f(z^{(l+1)})$$

Οργανώνοντας με αυτό τον τρόπο τις παραμέτρους σε πίνακες μπορούμε να επιταχύνουμε δραματικά τους υπολογισμούς του αλγορίθμου με τη χρήση άλγεβρας πινάκων. Η παραπάνω διαδικασία υπολογισμού της $h_{W,b}(x)$ονομάζεται αλγόριθμος*πρόσθιας διάδοσης (forwardpropagation).* Φυσικά ο μπορούμε να δημιουργήσουμε μια μεγάλη γκάμα από αρχιτεκτονικές νευρωνικών δικτύων με περισσότερα κρυμμένα στρώματα αλλά και περισσότερες μονάδες εξόδου, όπως στην Εικόνα 4.



Εικόνα 11Αναπαράσταση ενός τεχνητού νευρωνικού δικτύου με δύο μεταβλητές εξόδου.

6.2.2 Εκπαίδευση τεχνητών νευρωνικών δικτύων

Έστω το σύνολο m παρατηρήσεων $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$. Για να καταφέρουμε να δημιουργήσουμε ένα νευρωνικό δίκτυο που προβλέπει σε ικανοποιητικό βαθμό τις μεταβλητές που μας ενδιαφέρουν πρέπει να εκπαιδεύσουμε το νευρωνικό δίκτυο χρησιμοποιώντας τις παρατηρήσεις για να προσδιορίσουμε τις τιμές των παραμέτρων $\boldsymbol{W}, \boldsymbol{b}$. Αρχικά ορίζουμε τη συνάρτηση κόστους για κάθε ζεύγος παρατηρήσεων

$$J(W, b; x, y) = \frac{1}{2}\left\|h_{W,b}(x) - y\right\|^2$$

Στη συνέχεια ορίζουμε την συνάρτηση ολικού κόστους ως εξής:

$$J(W, b) = \left[\frac{1}{m}\sum_{i=1}^{m} J(W, b; x^{(i)}, y^{(i)})\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}(W_{ij}^{(l)})^2$$

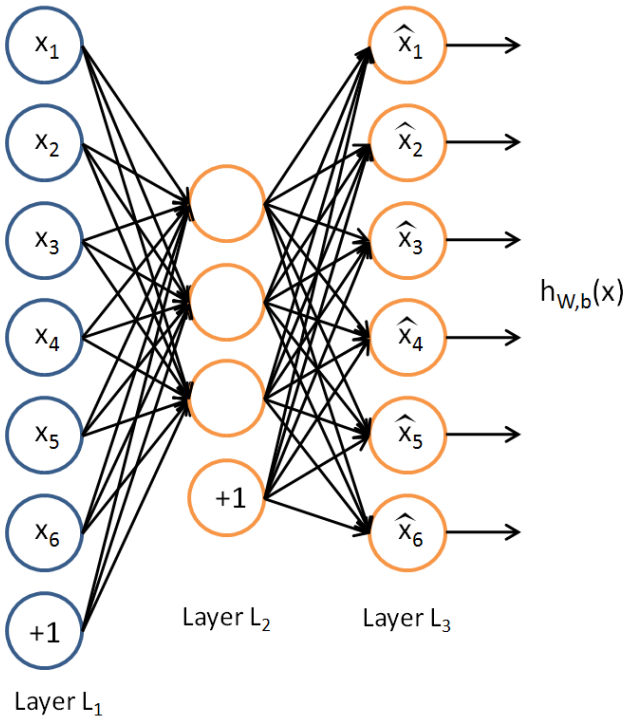
Ο πρώτος όρος είναι ουσιαστικά το μέσο σφάλμα που προκύπτει από την διαδικασία πρόσθιας διάδοσης. Ο δεύτερος όρος ονομάζεται όρος κοινωνικοποίησης και εξυπηρετεί στο να μην γίνεται overfitting κατά τη διαδικασία της εκπαίδευσης του νευρωνικού δικτύου.

Σκοπός είναι να υπολογίσουμε τις παραμέτρους $\boldsymbol{W}, \boldsymbol{b}$ που ελαχιστοποιούν την συνάρτηση κόστους. Η ελαχιστοποίηση επιτυγχάνεται με τη χρήση υπολογιστικών μεθόδων όπως ο αλγόριθμος της κλήσης) [37] και ο αλγόριθμος οπισθοδρομικής διάδοσης [38].

Αφού έχουμε εκπαιδεύσει το νευρωνικό δίκτυο μπορούμε να το χρησιμοποιήσουμε για να προβλέψουμε νέες τιμές της μεταβλητής Y από νέες παρατηρήσεις της μεταβλητής X που δεν υπήρχαν προηγουμένως στο σύνολο παρατηρήσεων που χρησιμοποιήθηκε κατά την εκπαίδευση.

6.3 Η μέθοδος Autoencoders

Είδαμε πως μπορούμε να εκπαιδεύσουμε ένα τεχνητό νευρωνικό δίκτυο ώστε να προσεγγίσει οποιαδήποτε μη γραμμική συνάρτηση μεταξύ δύο μεταβλητών. Τώρα θα δούμε πως μπορούμε να εφαρμόσουμε τη λογική των τεχνητών δικτύων στο πρόβλημα της μείωσης διαστάσεων. Η μέθοδος *autoencoder*[24]είναι ένα τεχνητό νευρωνικό δίκτυο του οποίου οι τελικές τιμές που θέλει να προβλέψει είναι οι ίδιες με τις παρατηρήσεις εισόδου, όπως φαίνεται στην εικόνα 5.



Εικόνα 12Αναπαράσταση ενός autoencoder.

Δηλαδή ένας autoencoder προσπαθεί να προσεγγίσει τη συνάρτηση $h_{W,b}(x)$ έτσι ώστε $h_{W,b}(x) \approx x$. Θέλει δηλαδή να κατασκευάσει μία προσέγγιση της ταυτοτικής συνάρτησης έτσι ώστε το αποτέλεσμα της εξόδου $\hat{x}$ να είναι ίδιο με τη μεταβλητή εισόδου *x*. Προφανώς δεν χρειαζόμαστε ένα πολύπλοκο αλγόριθμο μάθησης όπως το νευρωνικό δίκτυο για να προσεγγίσουμε την ταυτοτική συνάρτηση. Αυτό που είναι διαφέρον είναι πως μπορούμε να μειώσουμε τον αριθμό των ενδιάμεσων κρυμμένων στρωμάτων για να ανακαλύψουμε μια απεικόνιση των δεδομένων με λιγότερες διαστάσεις.

Κεφάλαιο 7 Το πρόβλημα της τιμολόγησης αεροπορικών εισιτηρίων

7.1 Εισαγωγή

«Πείτε μας πόσα χρήματα μπορείτε να διαθέσετε και θα σας στείλουμε ένα αεροπορικό εισιτήριο». Για χρόνια οι αεροπορικές εταιρίες προσπαθούν να πουλήσουν το ίδιο εισιτήριο σε διαφορετική τιμή, ανάλογα με την οικονομική κατάσταση του πελάτη που το αγοράζει. Οι περισσότερες αεροπορικές εταιρίες προσφέρουν ένα μεγάλοεύρος τιμών για ναύλους ενός ταξιδιού μεταξύ δυο πόλεων, ξεκινώντας από πολύ χαμηλές τιμές για επιβάτες που επιλέγουν να ταξιδέψουν στην οικονομική θέση, μέχρι πολύ υψηλότερες τιμές για businessclass ή firstclasseισιτήρια. Αυτή η τακτική κοστολόγησης χρησιμοποιείται κυρίως από μεγάλες αεροπορικές εταιρίες που προσπαθούν να ανταγωνιστούν τις αεροπορικές εταιρίες χαμηλού κόστους που εμφανίζονται στην αγορά τις τελευταίες δεκαετίες. Προσφέροντας αεροπορικά εισιτήρια με μεγάλες εκπτώσεις οι καθιερωμένες αεροπορικές εταιρίες μπορούν τουλάχιστον να εμφανίζονται ανταγωνιστικές έναντι των χαμηλού κόστους εναλλακτικών εταιριών.

Ο αποθεματικός έλεγχος των θέσεων [26] είναι η πρακτική εξισορρόπησης του αριθμού των θέσεων με έκπτωση έναντι των υπόλοιπων θέσεων σε κανονική τιμή με στόχο τη μεγιστοποίηση των κερδών. Ωστόσο η πώληση πολλών εισιτηρίων με

έκπτωση είναι πολύ πιθανό να προκαλέσει μείωση των τελικών κερδών της εταιρίας καθώς οι οικονομικές αυτές θέσεις γίνονται διαθέσιμες και σε επιβάτες που μπορούν να διαθέσουν πολύ περισσότερα χρήματα. Η διαχείριση του παραπάνω προβλήματος ονομάζεται διοικητική απόδοση (yieldmanagement) και έχει να κάνει τόσο με τον καθορισμό των τιμών όσο και με τον αποθεματικό έλεγχο των θέσεων.

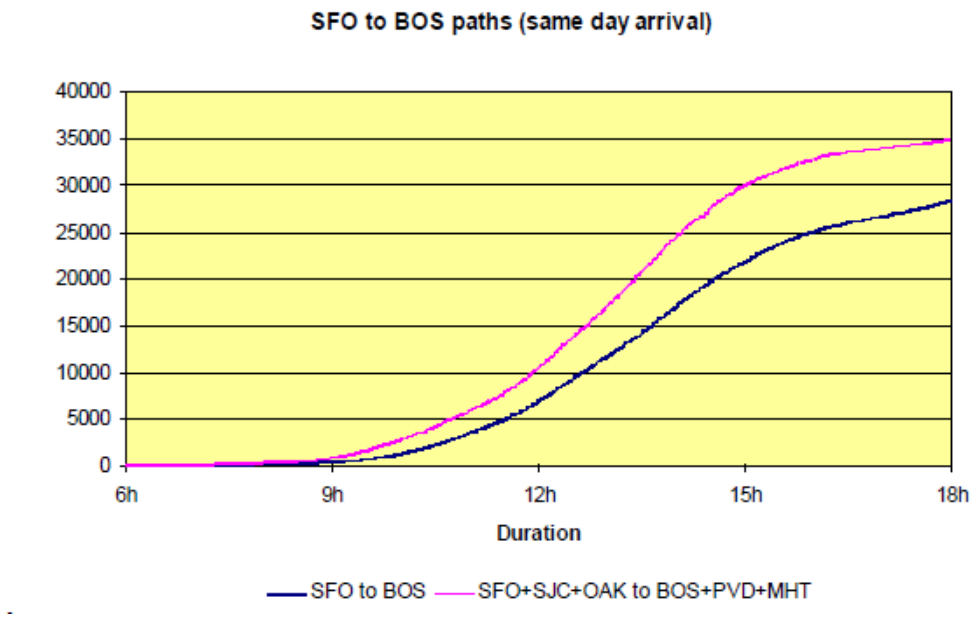
Αυτή η κατάσταση σε συνδυασμό με το πολύ μεγάλο πλήθος επιλογών για μια πτήση μεταξύ δύο πόλεων, κάνουν το πρόβλημα της κοστολόγησης των αεροπορικών εισιτηρίωνιδιαίτεραπολύπλοκο. Ακόμα και μετά τον καθορισμό των τιμών το πρόβλημα της επιλογής του βέλτιστου εισιτηρίου από όλους τους δυνατούς συνδυασμούς πτήσεων είναι υπολογιστικά δύσκολο.

7.2 Το δίκτυο των επιβατικών πτήσεων

Υπάρχουν περισσότερα από 4000 αεροδρόμια παγκοσμίως τα οποία εξυπηρετούν επιβάτες. Ο αριθμός των πτήσεων ανέρχεται στα 30.000.000 το χρόνο, δηλαδή μια πτήση κάθε δευτερόλεπτο [28]. Κατά μέσο όρο κάθε αεροδρόμιο έχει βαθμό 8, δηλαδή συνδέεται με 8 άλλα αεροδρόμια και εξυπηρετείται από 8 αεροπορικές εταιρείες. Παρόλα αυτά υπάρχουν κάποια μεγάλα αεροδρόμια τα οποία κυριαρχούν στο δίκτυο, εξυπηρετώντας 64 προορισμούς κατά μέσο όρο. Αυτό δείχνει το σύστημα κόμβου-ακτίνων που χρησιμοποιείται από πολλές αεροπορικές εταιρίες όπου για οι μισές τους αναχωρήσεις πραγματοποιούνται από ένα με τέσσερα αεροδρόμια. Παρά την υψηλή συνδεσιμότητα μεταξύ των κεντρικών αεροδρομίων το μέσο συντομότερο μονοπάτι είναι 5 πτήσεις. Υπάρχουν περιπτώσεις αεροδρομίων που η μετάβαση από το ένα στο άλλο χρειάζεται μέχρι και 20 πτήσεις, για παράδειγμα από ένα μικρό αεροδρόμιο στην Αλάσκα σε ένα άλλο μικρό αεροδρόμιο στην Ινδονησία.

Εικόνα 13Το πλήθος των δυνατών δρομολογίων από το Σαν Φρανσίσκο στη Νέα Υόρκη.Πηγή: [28].

Στην εικόνα 16φαίνονται τα 10.000 συντομότερα μονοπάτια μεταξύ της διαδρομής από τη Βοστόνη στο Σαν Φρανσίσκο. Όλες οι πτήσεις φτάνουν στο προορισμό τους την ίδια μέρα. Ακόμα, βλέπουμε πως καμία από τις διαδρομές δεν αφήνει την αμερικάνικη ήπειρο. Αυτές οι 10.000 διαδρομές μπορεί να είναι οι συντομότερες, όμως δεν εξαντλούν όλους τους πιθανούς τρόπους με τους οποίους μπορεί κανείς να μεταβεί από τη μία πόλη στην άλλη. Για την ακρίβεια ο αριθμός αυτός αυξάνεται εκθετικά όσο αυξάνεται και η επιθυμητή διάρκεια πτήσης (βλ. Εικόνα 17).



Εικόνα 17Το πλήθος των πτήσεων από το Σαν Φρανσίσκο στη Βοστόνη σε σχέση με τη διάρκεια πτήσης. Πηγή: [28]

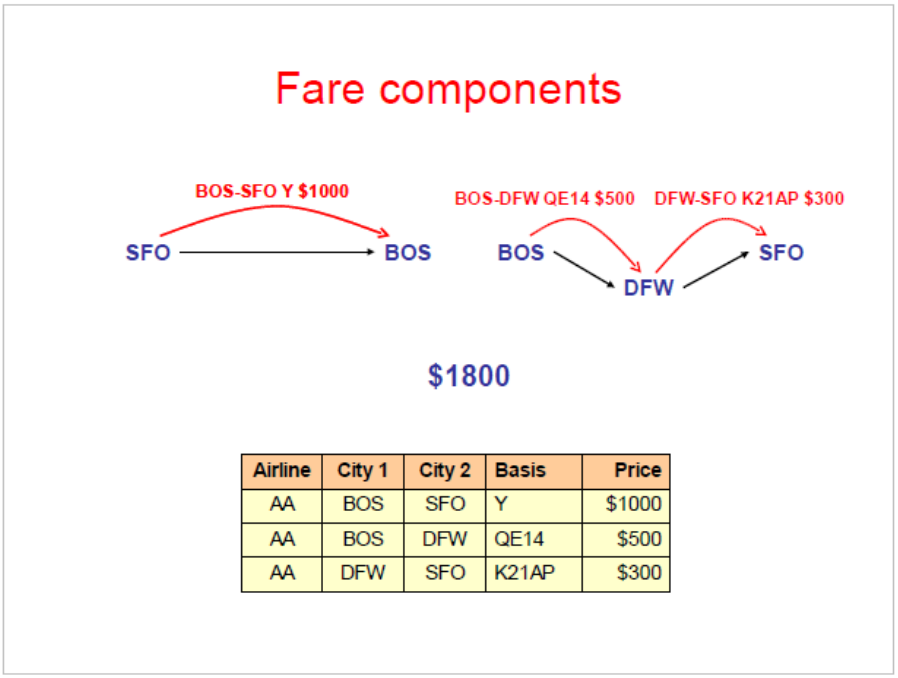
Από το τελευταίο παράδειγμα φαίνεται πως είναι αδύνατη η αναπαράσταση όλων των πιθανών πτήσεων από μία πόλη σε μία άλλη, πόσο μάλλον εάν θέλουμε να μια πτήση με επιστροφή. Ακόμα είναι πολύ δύσκολο για κάποιον ταξιδιώτη να επιλέξει την φθηνότερη πτήση λαμβάνοντας υπόψη όλες τις διαθέσιμες επιλογές. Αυτό το πρόβλημα κάνει την οργάνωση ενός αεροπορικού ταξιδιού πολύ διαφορετική από την οργάνωσηενός ταξιδιού με οποιοδήποτε άλλο μέσο.

Οι κλασσικοί αλγόριθμοι όπως για παράδειγμα ο αλγόριθμος του Dijkstra δεν μπορούν να χρησιμοποιηθούν για μειώσουν τον αριθμό των πιθανών πτήσεων, καθώς η τιμή κάθε πτήσης εξαρτάται άμεσα από το δίκτυο. Συνεπώς η δημιουργία ενός συστήματος που υπολογίζει σε ικανοποιητικό χρόνο τις βέλτιστες διαδρομές από άποψη κόστους αποτελεί μεγάλη πρόκληση [27].

7.3 Μηχανισμοί κοστολόγησης

Οι τιμές των αεροπορικών εισιτηρίων είναι πολύ πιο πολύπλοκες από τα περισσότερα προϊόντα. Ο τρόπος με τον οποίο καθορίζονται οι τιμές είναι αυτός που κάνει την οργάνωση ενός ταξιδίου ένα ιδιαίτερα δύσκολο πρόβλημα [28]. Η ατομική μονάδα κόστους στη συγκεκριμένη αγορά ονομάζεται *ναύλος*. Ένας ναύλος είναι η τιμή που ορίζει η αεροπορική εταιρεία για ένα ταξίδι μίας διαδρομής μεταξύ ενός ζεύγους πόλεων. Κάθε δύο τέτοιες πόλεις ονομάζονται *αγορά*. Σε κάθε ναύλο αντιστοιχεί ένα

μοναδικό αλφαριθμητικό όνομα, και του αντιστοιχούν συγκεκριμένοι κανόνες. Μια πτήση εξοφλείται από ακριβώς ένα ναύλο αλλά ένας ναύλος μπορεί να αντιστοιχεί σε παραπάνω από μια πτήσεις ενός ταξιδιού. Ο όρος που χρησιμοποιείται για να αναφερθούμε στις πτήσεις που αντιστοιχούν σε ένα συγκεκριμένο ναύλο είναι *συνιστώσες του ναύλου.*



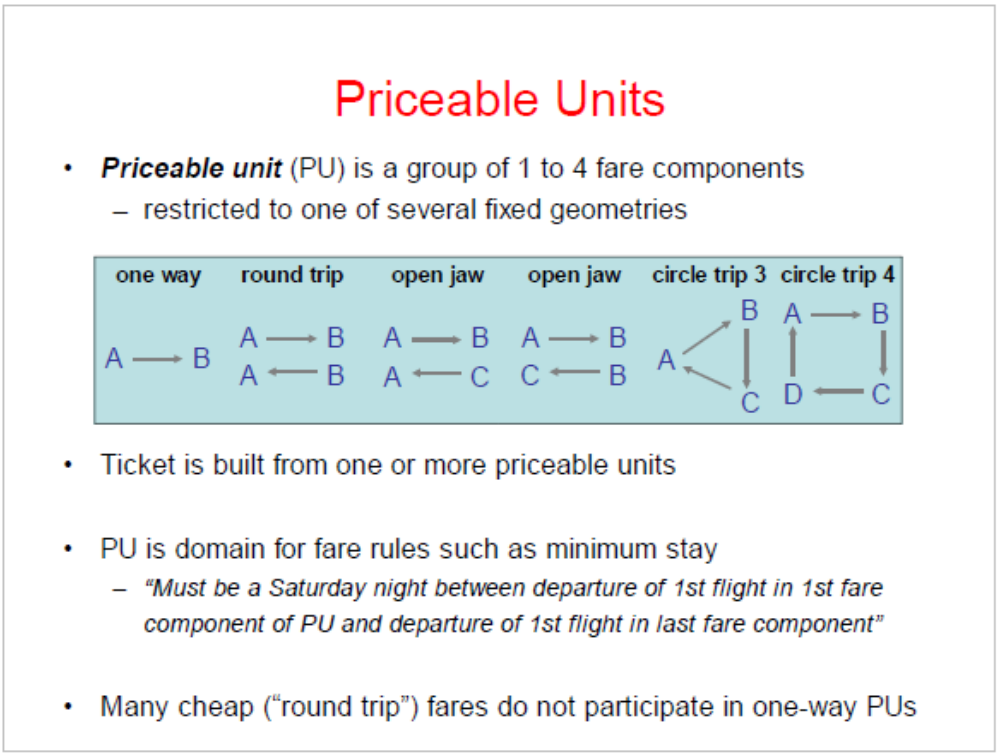
Airline	City 1	City 2	Basis	Price
AA	BOS	SFO	Y	$1000
AA	BOS	DFW	QE14	$500
AA	DFW	SFO	K21AP	$300

Εικόνα 18 Αναπαράσταση των συνιστωσών μιας πτήσης.Δύο διαφορετικοί τρόπο με τους οποίου μπορεί κανείς να ταξιδέψει από τη Βοστόνη στο Σαν Φρανσίσκο. Πηγή : [28]

Στην εικόνα 18 φαίνονται τρεις διαφορετικές επιλογές που έχει ένας ταξιδιώτης για να πραγματοποιήσει ένα ταξίδι από το Σαν Φρανσίσκο στη Βοστόνη. Βλέπουμε ότι μπορεί να ξοδέψει έως και 300 δολάρια λιγότερα εάν επιλέξει να πετάξει με τον συνδυασμόπτήσεωνQE14 και K21AP. Ο μόνος περιορισμός που θα είχε ένας ταξιδιώτης ώστε να μην μπορεί να επιλέξει τον συνδυασμό των παραπάνω πτήσεων θα ήταν να μην ικανοποιούνταν οι *κανόνες* που συνοδεύουν τους συγκεκριμένους ναύλους. Κάθε ναύλος έχει κάποιους κανόνες, για τους οποίους θα μιλήσουμε παρακάτω, των οποίων οι συνθήκες αν ικανοποιούνται τότε κάποιος μπορεί να τον αγοράσει.

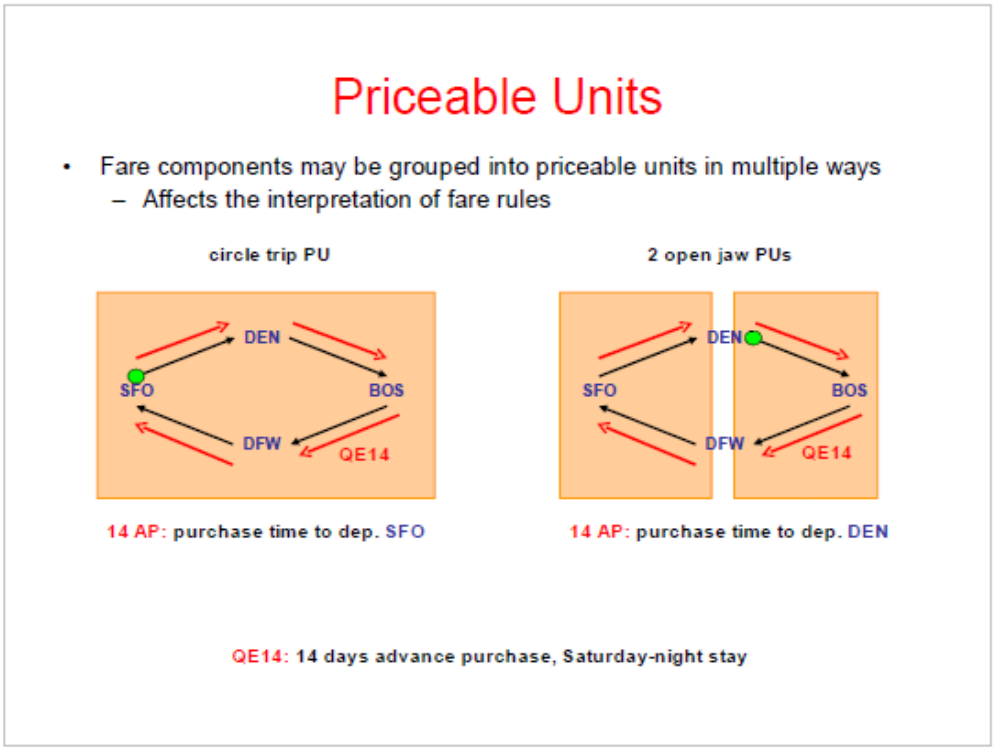
Υπάρχει ακόμα μια μονάδα αναπαράστασης μεταξύ των συνιστωσών ενός ναύλου και του τελικού εισιτηρίου. Αυτή η μονάδα ονομάζεται *Μονάδα Κοστολόγησης (PriceableUnit)*. Μια μονάδα κοστολόγησης μπορεί να αποτελείται από μία μέχρι

τέσσερεις συνιστώσες συνδυασμένα με ένα μικρό συγκεκριμένο αριθμό συνδυασμών όπως φαίνεται στην εικόνα 19. Αυτές οι μονάδες κοστολογήσεις είναι ο μικρότερος αριθμός πτήσεων που μπορούν να προμηθευτεί ένας ταξιδιώτης.



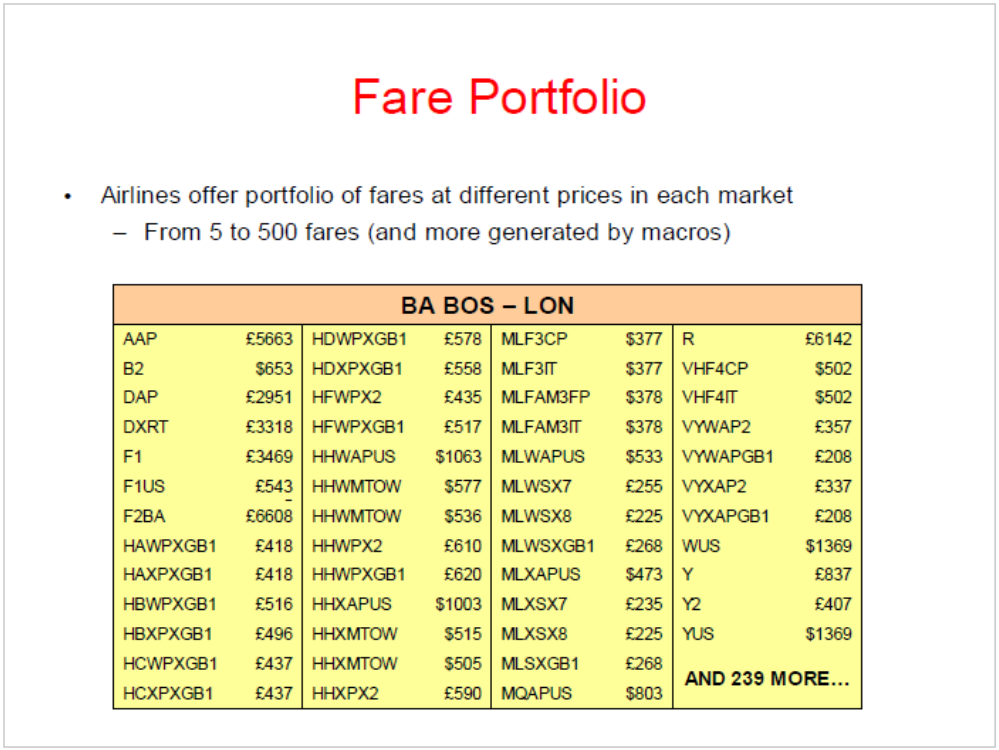
Εικόνα 19Μονάδες κοστολόγησης.Οι τέσσερεις τύποι μονάδων κοστολόγησης, απλή μετάβαση, ταξίδι με επιστροφή, ταξίδι ανοιχτού βρόγχου, κυκλικό ταξίδι. Πηγή: [28]

Η πιο απλή μορφή μια τέτοιας μονάδας είναι το ταξίδι μιας διαδρομής. Πιο περίπλοκες μορφές είναι τα ταξίδια με επιστροφή ή τα κυκλικά ταξίδια όπως φαίνονται στην εικόνα 19. Οι αεροπορικές χρησιμοποιούν αυτές τις δομές για να εφαρμόσουν πολλούς από τους κανόνες που συνοδεύουν τους αντίστοιχους ναύλους. Για παράδειγμα ο ναύλος μεταξύ δύο πόλεων που είναι μέρος ενός ταξιδιού με επιστροφή, τις περισσότερες φορές είναι φθηνότερος από τον αντίστοιχο ναύλο μεταξύ των ίδιων πόλεων που δεν είναι μέρος ενός ταξιδιού με επιστροφή.



Εικόνα 20Ο τρόπος με τον οποίο ομαδοποιούνται οι μονάδες κοστολόγησης επηρεάζει την δομή και το κόστος του τελικού ναύλου. Πηγή: [28]

Είναι προφανές πως ένας οποιοδήποτε συνδυασμός πτήσεων μπορεί να τμηματοποιηθεί με πολλούς τρόπους σε μονάδες κοστολόγησης. Στην εικόνα 20 φαίνονται 2 διαφορετικοί τρόποι με τους οποίους μπορεί να πραγματοποιηθεί ένα ταξίδι με επιστροφή από το Σαν Φρανσίσκο στη Βοστόνη με δύο διαφορετικούς τρόπους οργάνωσης των ναύλων σε μονάδες κοστολόγησης. Και οι δύο τρόποι κάνουν χρήση τεσσάρων ναύλων και τεσσάρων πτήσεων. Στην αριστερή εικόνα βλέπουμε οι ναύλοι αυτοί έχουν οργανωθεί σε μια μονάδα κοστολόγησης που αντιστοιχεί σε ένα κυκλικό ταξίδι, ενώ στη εικόνα έχουν οργανωθεί σε δύο μονάδες κοστολόγησης ανοιχτού βρόγχου.



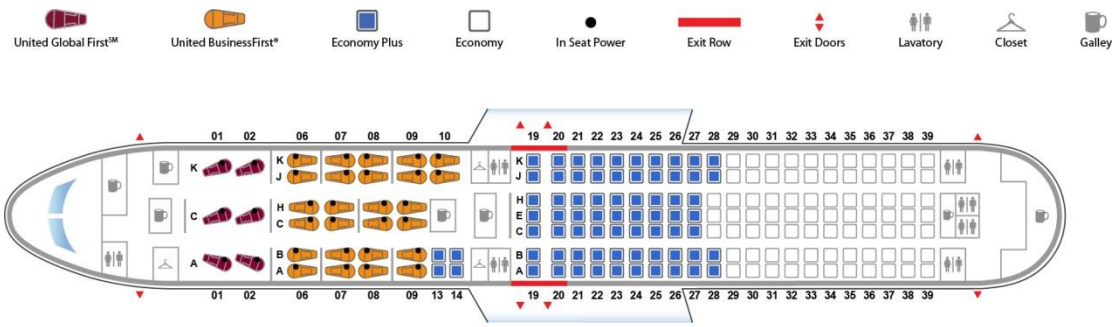
Εικόνα 21Διαφορετικοί ναύλοι για την ίδια διαδρομή.Φαίνονται διαφορετικοί κωδικοί εισιτηρίων που αφορούν την ίδια πτήση αλλά έχουν διαφορετικότελικό κόστος. Πηγή: [28]

Για κάθε αγορά, δηλαδή ένα δρομολόγιο μεταξύ δύο πόλεων, οι αεροπορικές εκδίδουν πολλούς διαφορετικούς ναύλους. Για παράδειγμα η BritishAirways προσφέρει παραπάνω από 280 διαφορετικούς ναύλους μεταξύ του Λονδίνου και της Βοστόνης. Άλλες αεροπορικές εκδίδουν μέχρι και 1000 διαφορετικούς ναύλους μεταξύ κάποιων διεθνών δρομολογίων. Κάθε ναύλος έχει διαφορετική τιμή και διαφορετικό κωδικό. Ένα ερώτημα που προκύπτει εδώ είναι, γιατί οι αεροπορικές εταιρίες εκδίδουν τόσους πολλούς διαφορετικούς ναύλους αφού ένας ταξιδιώτης θα ψάξει πάντα για το φθηνότερο. Η απάντηση είναι ότι, όπως είδαμε και προηγουμένως, κάθε ναύλοςσυνοδεύεταιαπό συγκεκριμένους κανόνες που επιτρέπουν την αγορά του μόνο υπό συγκεκριμένες συνθήκες.

Οι κανόνες αυτοί μπορεί να αφορούν σε οποιαδήποτε εμπλεκόμενη συνιστώσα του τελικού ταξιδιού. Υπάρχουν κανόνες που αφορούν την ηλικία ή την εθνικότητα του επιβάτη, την ημερομηνία του ταξιδιού, την τοποθεσία της αγοράς, τη διάρκεια, τις στάσεις κ.α.

7.4 Ρύθμιση αποθέματος σε αεροπορικές θέσεις

Είδαμε πως για ένα δρομολόγιο ανάμεσα σε δυο πόλεις οι αεροπορικές εταιρείες μπορεί να έχουν διαθέσιμες μέχρι και πάνω από 1000 διαφορετικές τιμές εισιτηρίων. Πως όμως καθορίζονται αυτές οι τιμές; Τι μηχανισμοί χρησιμοποιούνται για να αντιστοιχίσουν μια θέση σε μια συγκεκριμένη τιμή με συγκεκριμένους κανόνες; Για παράδειγμα η επιλογή ενός συγκεκριμένου τύπου αεροσκάφους για την πραγματοποίηση ενός δρομολογίου έχει άμεσο αντίκτυπο στην αριθμό των εισιτηρίων που μπορούν να είναι διαθέσιμα λόγω του μεγέθους του αεροσκάφους και επόμενος μπορεί να επηρεάσει άμεσα τη σχέση ζήτησης προσφοράς.

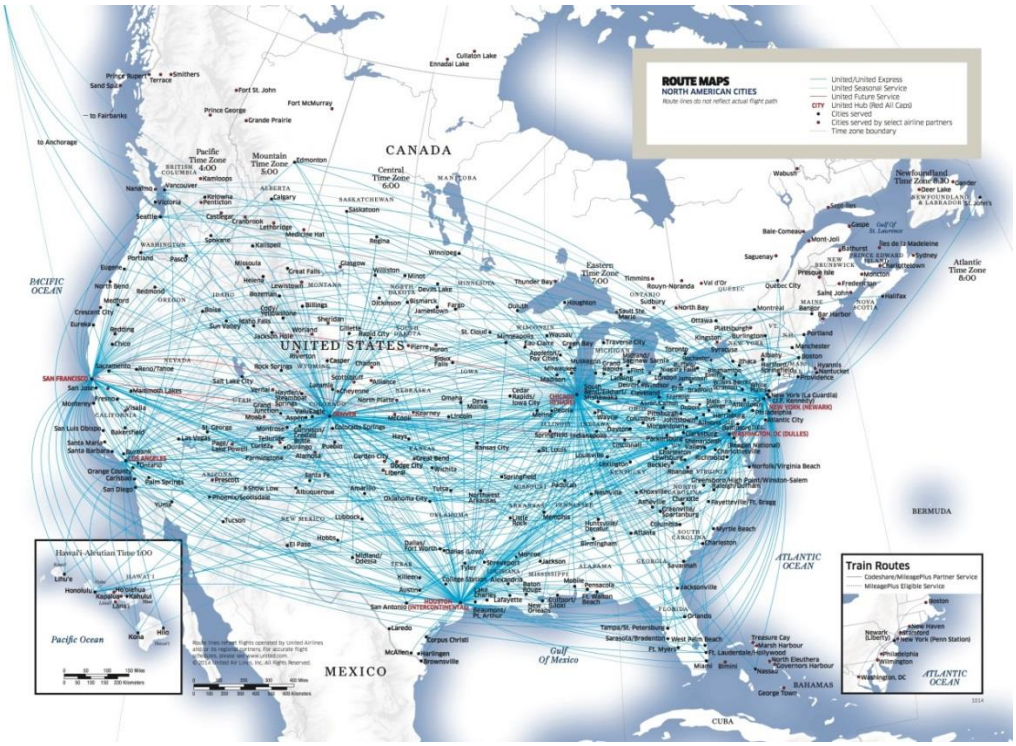


Εικόνα 2214:Κάτοψη ενός Boeing 767-300 (763) united . Πηγή: [28]

Οι αεροπορικές εταιρίες ακολουθούν την απλούστερη προσέγγιση στον αποθεματικό έλεγχο των αεροπορικών θέσεων είναι η αντιμετώπιση κάθε ξεχωριστής πτήσης του συνόλου των ταξιδιών ξεχωριστά, αντί να προσπαθούν να βελτιστοποιήσουν ολόκληρο το σύνολο των θέσεων σε ολόκληρο το δίκτυο των πτήσεων τους [26]. Όμως ακόμα και η εξέταση του προβλήματος για μια μόνο πτήση μπορεί να γίνει πολύ περίπλοκο αφού μια πτήση μεταξύ δύο πόλεων μπορεί να εξυπηρετεί επιβάτες από διαφορετικά σημεία αναχώρησης και διαφορετικούς τελικούς προορισμούς. Συνεπώς το πρόβλημα της ρύθμισης του αποθέματος των θέσεων ενός αεροσκάφους έχει να κάνει με τη κατάλληλη αντιστοίχηση επιβάτη-θέσης για τη μεγιστοποίηση των εσόδων της εκάστοτε εταιρείας.

Η πολυπλοκότητα της ρύθμισης του αποθέματος των θέσεων ενός αεροσκάφους ακόμα και για μία πτήση έχει αυξηθεί δραματικά με την χρήση του συστήματος

κόμβου-ακτίνων από πολλές μεγάλες αεροπορικές εταιρίες όσον αφορά τη χρήση των αεροδρομίων για την εξυπηρέτηση των δρομολογίων τους.



Εικόνα 23:Πλήθος πτήσεων από τα αεροδρόμια σεΣαν Φρανσίσκο, Νέα Υόρκη, Ουάσινγκτον και Χιούστον. Πηγή: [28]

Μια μεγάλη εταιρία μπορεί να έχει μέχρι και 1000 πτήσεις την ημέρα , να εξυπηρετεί χιλιάδες δρομολόγια και να έχει μέχρι 1000 διαφορετικές τιμές ναύλων για την ίδια πτήση. Για παράδειγμα η AmericanAirlines έχει πάνω από 2700 διαφορετικά δρομολόγια. Από το κεντρικό αεροδρόμιο που χρησιμοποιεί στο Ντάλας κάθε επιβάτης μπορεί να επιλέξει πάνω από 50 διαφορετικούς προορισμούς, σε πέντε διαφορετικές κλάσεις τιμών των ναύλων. Συνεπώς υπάρχουν πάνω από 250 δυνατοί συνδυασμοί τιμής/προορισμού για κάθε μια από τις διαθέσιμες θέσεις στα αεροσκάφη της. Κάθε πτήση έχει διαφορετικούς στόχους από την εταιρεία όσο αφορά τα έσοδα. Τέλος οι κρατήσεις για κάθε θέση ανοίγουν μέχρι και 11 μήνες πριν την πτήση πράγμα που σε συνδυασμό με τα παραπάνω κάνει πολύ δύσκολη τη διαχείριση του συγκεκριμένου προβλήματος.

7.5 Το πρόβλημα

Ο τρόπος με τον οποίο λειτουργεί η αγορά των επιβατικών πτήσεων, έχει πολλά χαρακτηριστικά που δημιουργούν μια μεγάλη πολυπλοκότητα [28]. Αρχικά ο τρόπος με τον οποίο δημιουργούνται οι τιμές από τις αεροπορικές εταιρίες είναι πολύ μακριά από μία βέλτιστη κατάσταση αφού η κοστολόγηση αυτή εξαρτάται από πάρα πολλούς εξωτερικούς αλλά και εσωτερικούς παράγοντες. Για αυτό το λόγο ακόμα και το πολύ απλό ερώτημα «πόσο κοστίζει ένα αεροπορικό εισιτήριοαπό την πόλη Α στην πόλη Β;» είναι δύσκολο να απαντηθεί και τις περισσότερες φορές απαιτείται η χρήση κάποιου ακριβού και πολύπλοκου λογισμικού.

Φανταστείτε ότι μπαίνετε σε ένα πολυκατάστημα και σε κανένα προϊόν δεν αναγράφεται η τιμή έως ότου το πάρετε από το ράφι. Σίγουρα κάτι τέτοιο δυσκολεύει τη διαδικασία της αγοράς σας. Παρόλα αυτά αυτή είναι μια συνήθης κατάσταση στην αγορητών αεροπορικών εισιτηρίων. Ακόμα και σήμερα δεν μπορούμε να απαντήσουμε άμεσα ερωτήσεις όπως «ποίες είναι οι τιμές όλων των πτήσεων από την Αθήνα προς όλο τον κόσμο σε τρεις μέρες από τώρα;». Φυσικά όταν λέμε ότι δεν μπορούμε εννοούμε ότι δεν μπορούμε σε ένα πολύ μικρό χρονικό διάστημα. Ακόμα και με τη χρήση εξειδικευμένου λογισμικού είναι πολύ δύσκολο και χρονοβόρο να απαντήσουμε στο παραπάνω ερώτημα. Αυτό το φαινόμενο είναι σχεδόν αποκλειστικό χαρακτηριστικό της συγκεκριμένης αγοράς.

Στο επόμενο κεφάλαιο προσπαθούμε να ανακαλύψουμε την κρυφή γεωμετρία πίσω από αυτό το περίπλοκο σύστημα κοστολόγησης χρησιμοποιώντας τεχνικές μείωσης διαστάσεων και εξόρυξης χαρακτηριστικών. Στόχος μας είναι να δούμεεάνείναι δυνατό να εντοπιστεί η αληθινή τιμή ενός αεροπορικού εισιτηρίου γνωρίζοντας την πολλαπλότητα πάνω στην οποία ανήκει τελικά το συγκεκριμένο πρόβλημα.

Κεφάλαιο 8 Έρευνα και αποτελέσματα

8.1 Εισαγωγή

Όπως είδαμε στο προηγούμενο κεφάλαιο, το πλήθος κανόνων για την τιμολόγηση των αεροπορικών εισιτηρίων έχει συμβάλει στη δημιουργία ενός πολύπλοκου δικτύου πτήσεων στο οποίο ακόμα και το, φαινομενικά πολύ απλό, πρόβλημα εύρεσης του φθηνότερου εισιτηρίου μεταξύ δύο πόλεων είναι μια ιδιαίτερα δύσκολη διαδικασία. Στο παρόν κεφάλαιο θα προσπαθήσουμε να ανακαλύψουμε την αληθινή γεωμετρία πίσω από το δίκτυο, αναλύοντας ένα μικρότερο δίκτυο ευρωπαϊκών πτήσεων. Η ανάλυση θα γίνει με τη χρήση των μεθόδων, PCA, Isomap, diffusionmaps και autoencoders (artificialneuralnetworks) που περιγράψαμε αναλυτικά στα κεφάλαια 3, 4, 5 και 6.

Πιο συγκεκριμένα, το σύνολο δεδομένων αποτελείται από τις τιμές των αεροπορικών εισιτηρίων μεταξύ της Αθήνας και άλλων τριάντα (30) ευρωπαϊκών πόλεων. Πάνω στο σύνολο αυτό θα εφαρμόσουμε τις μεθόδους μείωσης διαστάσεων που αναφέρθηκαν για να εντοπίσουμε πιθανές συνιστώσες που περιγράφουν ορθότερα την γεωμετρία του δικτύου. Για να ελέγξουμε την αξία των συνιστωσών που θα προκύψουν από την κάθε μέθοδο θα εκπαιδεύσουμε ένα τεχνητό νευρωνικό δίκτυο το οποίο θα δέχεται σαν όρισμα τις συνιστώσες που προέκυψαν από κάθε μέθοδο και θα έχει σαν μεταβλητές απόκρισης τις αληθινές τιμές των εισιτηρίων μεταξύ τις Αθήνας και της εκάστοτε πόλης.

8.2 Περιγραφή των δεδομένων

Το σύνολο δεδομένων ανακτήθηκε από το διαδίκτυο μέσω .....Αποτελείται από τριάντα (30) στήλες, όπου κάθε στήλη αντιστοιχεί σε μια ευρωπαϊκή πόλη. Κάθε στήλη αποτελείται από 35 παρατηρήσεις που περιγράφουν την διακύμανση των τιμών των αεροπορικών εισιτηρίων από την Αθήνα προς την εκάστοτε πόλη. Κάθε μια από τις 35 παρατηρήσεις κάθε στήλης αντιστοιχεί σε μια από τις 35 μέρες για το χρονικό

διάστημα 1/11/2013 έως 4/12/2013. Οι τιμές φαίνονται αναλυτικά στους πίνακες 1, 2, 3.

amsterdam	ancona	barcelona	berlin	bologna	brindisi	brussels	copenhagen	florence	geneva
177,1	293,2	197,1	130,9	183,1	311,2	79,98	152,4	311,8	86,98
142,7	257,2	124,2	109,9	164,1	257,2	60,98	163,1	244,1	108,6
161,1	301,2	117,1	219,5	164,1	293,2	60,98	156,6	238,8	88,98
162,8	215,2	124,2	130,9	164,1	215,2	60,98	195,1	224,9	148,6
162,8	197,2	157,1	189,9	164,1	197,2	69,98	195,1	271,4	148,6
146,8	380,2	117,1	156,4	164,1	309,9	60,98	194,4	244,1	86,98
162,8	197,2	164,2	121,9	141,1	197,2	60,98	153,3	244,1	128,6
115,8	197,2	139,4	110,6	132,1	215,2	60,98	115,6	238,8	71,98
135,8	257,2	124,2	134,9	164,1	257,2	60,98	163,1	244,1	108,6
161,1	301,2	102,1	182,4	141,1	244,8	60,98	156,6	224,9	93,98
150,4	197,2	147,1	156,4	156,1	197,2	60,98	152,4	238,8	88,65
125,8	197,2	177,1	148,9	132,1	197,2	60,98	156,6	238,8	148,6
83,89	197,2	175,4	102	141,1	197,2	60,98	139,1	203,4	71,98
115,8	197,2	159,2	119,9	141,1	197,2	60,98	201,4	238,8	128,6
115,8	374,1	211,8	114,9	121,6	294,5	60,98	178,8	238,8	61,98
105,8	197,2	122	102	121,6	197,2	60,98	167	224,9	88,65
120,2	275,2	102,1	117	121,1	257,2	60,98	174	210,3	102
115,8	341,2	124,2	163,9	121,6	309,9	60,98	235,6	224,9	108,6
83,89	197,2	102,1	102	121,6	197,2	60,98	167,9	191,8	117,1
83,89	197,2	117,1	77,01	121,6	197,2	60,98	152,4	190,9	93,98
105,8	197,2	124,2	110,9	121,6	197,2	60,98	138,8	238,8	125,3
146,8	278,9	137,1	123,9	192	278,9	60,98	134,6	205,9	103,3
162	197,2	117,1	77,01	155,1	197,2	60,98	90,14	224,9	102
162	275,2	158,1	117	140,1	257,2	60,98	160	274,8	196,2
191,3	197,2	124,2	143,9	159,1	197,2	91,98	184,1	236,8	243,8
142	197,2	122	102	155,1	197,2	60,98	156,6	335,8	102
122	197,2	157,1	97,98	177,1	197,2	60,98	98,65	221,4	102
150,4	197,2	190,1	97,98	149,1	197,2	60,98	157,5	340,8	125,3
150,4	197,2	190,1	97,98	149,1	197,2	60,98	157,5	340,8	125,3
95,89	418,8	92,48	82,98	148,1	314,9	60,98	98,65	163,9	102
95,89	197,2	117,1	82,98	138,1	197,2	60,98	152,4	241,8	151,3
142	328,7	106,1	135,9	149,1	337,9	60,98	180,5	190,9	142
83,89	438,8	118,4	135,9	181,9	293,9	124,9	212,9	188,9	129,6
83,89	197,2	105,6	97,98	129,1	197,2	91,98	138,6	238,8	97,14
83,89	278,9	137,1	60,98	109,1	278,9	79,98	80,65	204,8	102

Πίνακας 1Οι τιμές των αεροπορικώνεισιτηρίων από την Αθήνα σε 10 ευρωπαϊκές πόλεις.

genoa	istanbul	london	lyon	madrid	marseille	milan	munich	naples	nice
216,1	112,6	166,4	210	209,6	204,1	98	224,1	220,1	210
111,2	112,6	145,9	189	181,6	172,1	83	179,2	185,9	210
189,1	118,1	135,9	163	188,1	151,3	61	162	59,98	129
111,2	112,6	128,4	127	184,5	122,2	98	120,2	168,2	127
168,2	96,14	145,9	196	156,6	172,1	83	214,6	44,98	209
193,1	88,14	124,9	170	156,6	157,8	61	157,7	163,2	180
160,1	88,14	124,9	169	169,2	149,1	61	131,6	30,98	174
193,1	77,14	106,9	134	119,1	149,1	53	202,8	163,2	196
111,2	96,14	108,9	189	132	172,1	53	120,2	168,2	186
149,2	88,14	124,9	134	155,1	151,3	53	157,7	39,98	159
85,28	97,65	124,9	127	225,5	100,2	71	105,2	168,2	105
112,2	77,14	108,9	189	282,6	149,1	71	192	39,98	210
190,4	83,65	77,98	167	155,1	192,5	40	112	163,2	186
150,1	83,65	79,98	156	179,2	149,1	31	102,6	27,98	159
190,4	97,65	74,98	134	156,6	133,6	28	189,6	163,1	189
85,68	88,14	79,98	153	112	149,1	45	95,28	163,1	186
98,28	77,14	69,98	114	162,6	122,2	53	94,79	39,98	127
85,28	88,14	69,98	110	163,5	100,2	53	105,2	163,1	105
98,28	83,65	69,98	155	112	149,1	61	112	30,98	165
186,5	73,65	60,98	170	119,1	185,8	39	112	163,1	180
113,1	77,14	60,98	156	125,2	149,1	31	94,79	30,98	127
193,1	83,65	79,98	134	119,1	156,6	40	144,6	163,1	186
111,2	88,14	106,9	163	97,01	166,3	61	105,2	163,1	173
149,2	101,1	112	153	219,1	153,1	53	94,79	68,98	155
112,2	83,65	124,9	179	180,1	122,2	53	120,2	163,1	127
168,2	73,65	92,01	163	97,01	196,1	53	112	31,98	184
215,2	73,65	92,01	196	163,5	196,1	32	112	163,1	186
193,1	73,65	124,9	196	207,5	196,1	36	102,6	31,98	184
193,1	73,65	124,9	196	207,5	196,1	36	102,6	31,98	184
149,2	73,65	92,01	196	71,14	156,6	32	112	163,1	151
159,2	73,65	151,9	195	163,5	196,1	32	164,2	163,1	185
189,1	83,65	162	157	98,65	207,1	53	132	31,98	186
149,2	93,65	235,9	176	101,1	156,6	53	144,6	143,1	195
112,2	59,65	132	176	132	165,1	53	162,1	143,1	165
213,3	65,65	79,98	61	155,1	175,1	40	109,1	143,1	160

Πίνακας 2Οι τιμές των αεροπορικώνεισιτηρίων από την Αθήνα σε 10 ευρωπαϊκές πόλεις.

palermo	paris.	pisa.	rome.	stockholm	strasbourg	trieste	turin	venice	vienna
320,5	114	193	82	152,4	166,1	311,2	220	195,2	115,7
111,2	131	168	70	198,3	166,1	275,2	111	168,2	121,6
213,2	116	185	60	135,6	166,1	309,9	162	168,2	140,6
111,2	131	215	70	212,3	234,1	239,2	111	182,5	119,4
168,2	126	197	52	134,4	166,1	197,2	168	163,2	119,4
202,9	126	194	52	166,1	166,1	309,9	169	123,1	102,6
159,2	69	197	52	153,3	166,1	197,2	159	123,1	102,6
197,2	114	123	52	134,4	166,1	215,2	208	163,2	85,65
98,28	101	123	60	155,1	199,1	257,2	111	153,1	102,6
149,2	104	123	52	157,6	166,1	309,9	179	168,2	121,6
85,28	98	123	52	231,3	199,1	197,2	85,3	168,2	102,6
112,2	90,7	197	52	134,4	166,1	197,2	112	163,2	102,6
197,2	90,7	192	52	102	166,1	197,2	208	103,1	60,65
149,2	65	197	52	146,3	166,1	197,2	149	123,1	85,65
300,7	90,7	194	52	115,6	166,1	294,1	208	163,2	70,65
85,28	80,7	194	52	122	166,1	197,2	85,7	113,1	119,4
98,28	80	162	52	102	166,1	275,2	147	123,1	107
85,28	58	194	52	258	166,1	309,9	85,3	168,2	102,6
98,28	90,7	197	52	122	166,1	197,2	98,3	163,2	85,65
186,5	77	192	52	75,01	166,1	197,2	169	113,1	85,65
112,2	72	197	52	143,1	166,1	197,2	112	163,2	70,65
214,8	90,7	194	52	115,6	166,1	278,9	189	163,2	85,65
98,28	77	197	60	75,01	166,1	197,2	111	113,1	70,65
221,2	122	194	60	122	166,1	275,2	179	168,2	121,6
112,2	131	194	52	157,6	191,1	197,2	113	168,2	119,4
168,2	77	194	52	75,01	177,6	197,2	168	163,2	107
197,2	77	194	60	75,01	191,1	197,2	208	113,1	107
197,2	177	197	52	198,3	199,8	197,2	208	209,2	119,4
197,2	177	197	52	198,3	199,8	197,2	208	209,2	119,4
149,2	122	194	52	134,4	166,1	343,8	149	186,1	85,65
159,2	178	194	52	198,3	199,8	197,2	159	93,14	102,6
197,2	132	185	52	212,4	199,8	363,8	202	206,9	102,6
149,2	169	133	52	208,6	191,1	343,8	141	165,1	110,6
112,2	131	133	52	166,3	166,1	197,2	112	165,1	83,65
212,9	75,7	133	52	122	166,1	278,9	141	163,4	107

Πίνακας 3 Οι τιμές των αεροπορικώνεισιτηρίων από την Αθήνα σε 10 ευρωπαϊκές πόλεις.

8.3 Μεθοδολογία και αποτελέσματα

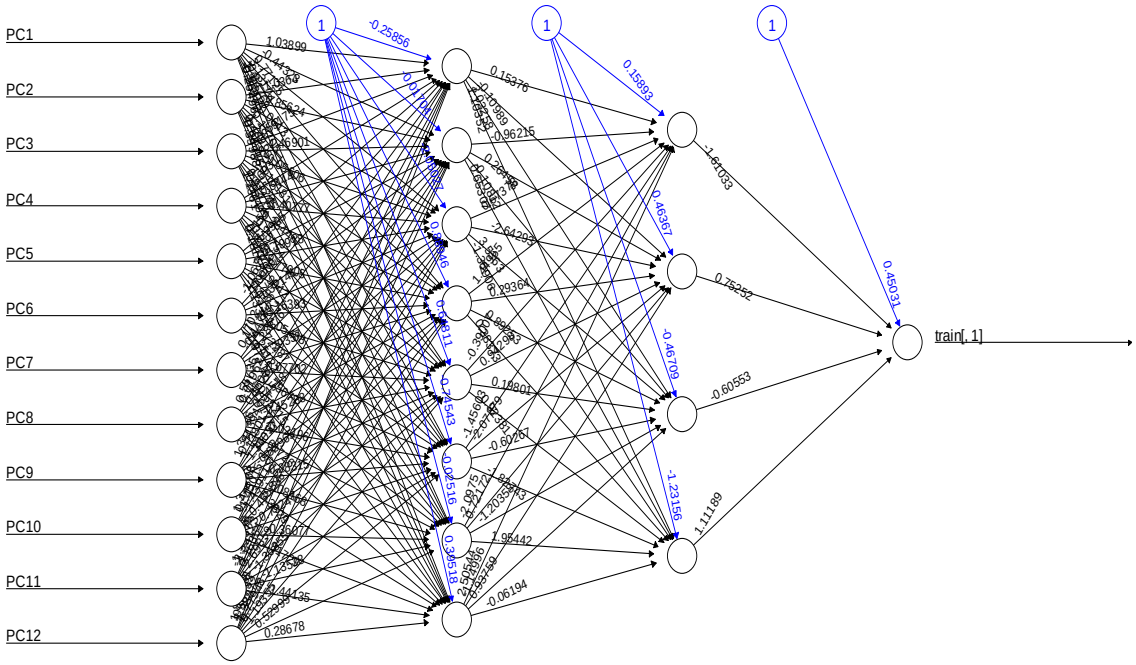
8.3.1 Μείωση διαστάσεων με χρήση της μεθόδου PCA

Η πρώτη μέθοδος που εφαρμόσαμε πάνω στο σύνολο δεδομένων είναι η μέθοδος ανάλυσης σε κύριες συνιστώσες, αφού πρώτα κανονικοποιήσουμε τα δεδομένα. Ο αριθμός των κύριων συνιστωσών που προκύπτει είναι φυσικά 30 όσες και οι στήλες των δεδομένων. Από την ανάλυση σημαντικότητας των συνιστωσών βλέπουμε ότι αρκούν 12 συνιστώσες για διατηρήσουμε το 90% της πληροφορίας του αρχικού συνόλου δεδομένων.

```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7
Standard deviation     2.5117 2.1951 1.9887 1.61562 1.48631 1.23216 1.18579
Proportion of Variance 0.2103 0.1606 0.1318 0.08701 0.07364 0.05061 0.04687
Cumulative Proportion  0.2103 0.3709 0.5027 0.58973 0.66337 0.71398 0.76085
                           PC8     PC9    PC10    PC11    PC12    PC13   PC14
Standard deviation     1.08769 0.97006 0.93165 0.81396 0.77219 0.75117 0.6505
Proportion of Variance 0.03944 0.03137 0.02893 0.02208 0.01988 0.01881 0.0141
Cumulative Proportion  0.80028 0.83165 0.86058 0.88267 0.90254 0.92135 0.9355
                          PC15    PC16    PC17   PC18   PC19   PC20    PC21
Standard deviation     0.59552 0.55491 0.53275 0.4899 0.4415 0.3796 0.34663
Proportion of Variance 0.01182 0.01026 0.00946 0.0080 0.0065 0.0048 0.00401
Cumulative Proportion  0.94728 0.95754 0.96700 0.9750 0.9815 0.9863 0.99031
                          PC22    PC23    PC24    PC25    PC26    PC27    PC28
Standard deviation     0.32758 0.24830 0.19924 0.17461 0.15663 0.11076 0.10358
Proportion of Variance 0.00358 0.00206 0.00132 0.00102 0.00082 0.00041 0.00036
Cumulative Proportion  0.99389 0.99594 0.99726 0.99828 0.99910 0.99951 0.99986
                          PC29    PC30
Standard deviation     0.05735 0.02804
Proportion of Variance 0.00011 0.00003
Cumulative Proportion  0.99997 1.00000
```

Χρησιμοποιώντας τις 12 πρώτες κύριες συνιστώσες εκπαιδεύσαμε ένα νευρωνικό δίκτυο το οποίο θα δέχεται ως όρισμα αυτές τις 12 μεταβλητές και έχει σαν μεταβλητή απόκρισης το κανονικοποιημένο σύνολο δεδομένων.

Στην εικόνα 24 φαίνεται το τεχνητό νευρωνικό δίκτυο που προέκυψε για την προσέγγιση της πρώτης στήλης του συνόλου δεδομένων.



Εικόνα 24Αναπαράσταση του νευρωνικού δικτύου όπως προέκυψε εφαρμόζοντας την μέθοδο PCA

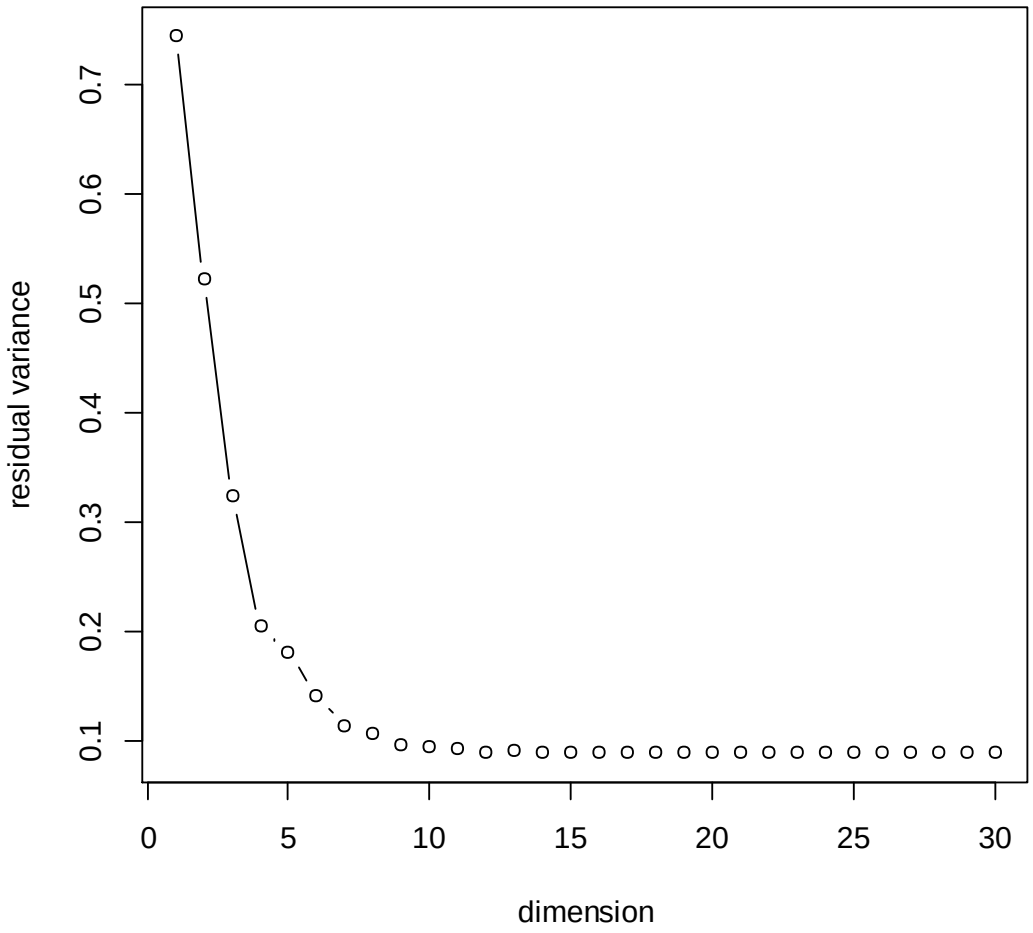
Στη συνέχεια χρησιμοποιήθηκε η μέθοδος 10-foldcrossvalidation ώστε η τελική τιμή του μέσου τετραγωνικού σφάλματος να είναι όσο το δυνατόν πιο αντιπροσωπευτική. Σύμφωνα με αυτή τη μέθοδο το σύνολο δεδομένων σε δύο υποσύνολα. Το πρώτο περιέχει το 10% των παρατηρήσεων, οι οποίες επιλέχθηκαν τυχαία. Το νευρωνικό δίκτυο εκπαιδεύεται με βάση το άλλο υποσύνολο το οποίο περιέχει τις υπόλοιπες παρατηρήσεις. Στη συνέχεια χρησιμοποιούμε το νευρωνικό δίκτυο που προκύπτει για να εκτιμήσουμε τις τιμές του υποσυνόλου που περιέχει το 10% των παρατηρήσεων και που δεν έπαιξε κανένα ρόλο στην εκπαίδευση του μοντέλου. Με αυτό τον τρόπο μπορούμε να συγκρίνουμε το μέσο τετραγωνικό σφάλμα των δύο συνόλων και να σιγουρευτούμε ότι έχουν αποφευχθεί φαινόμενα overfitting.

Το μέσο τετραγωνικό σφάλμα του νευρωνικού δικτύου που προέκυψε με βάση τις 12 κύριες συνιστώσες του συνόλου είναι 1.66929791.

8.3.2 Μείωση διαστάσεων με χρήση της μεθόδου Isomap

Η δεύτερη μέθοδος που χρησιμοποιήθηκε για την ανάλυση των δεδομένων είναι η μέθοδος Isomap. Από την εφαρμογή της μεθόδου πάνω στο κανονικοποιημένο

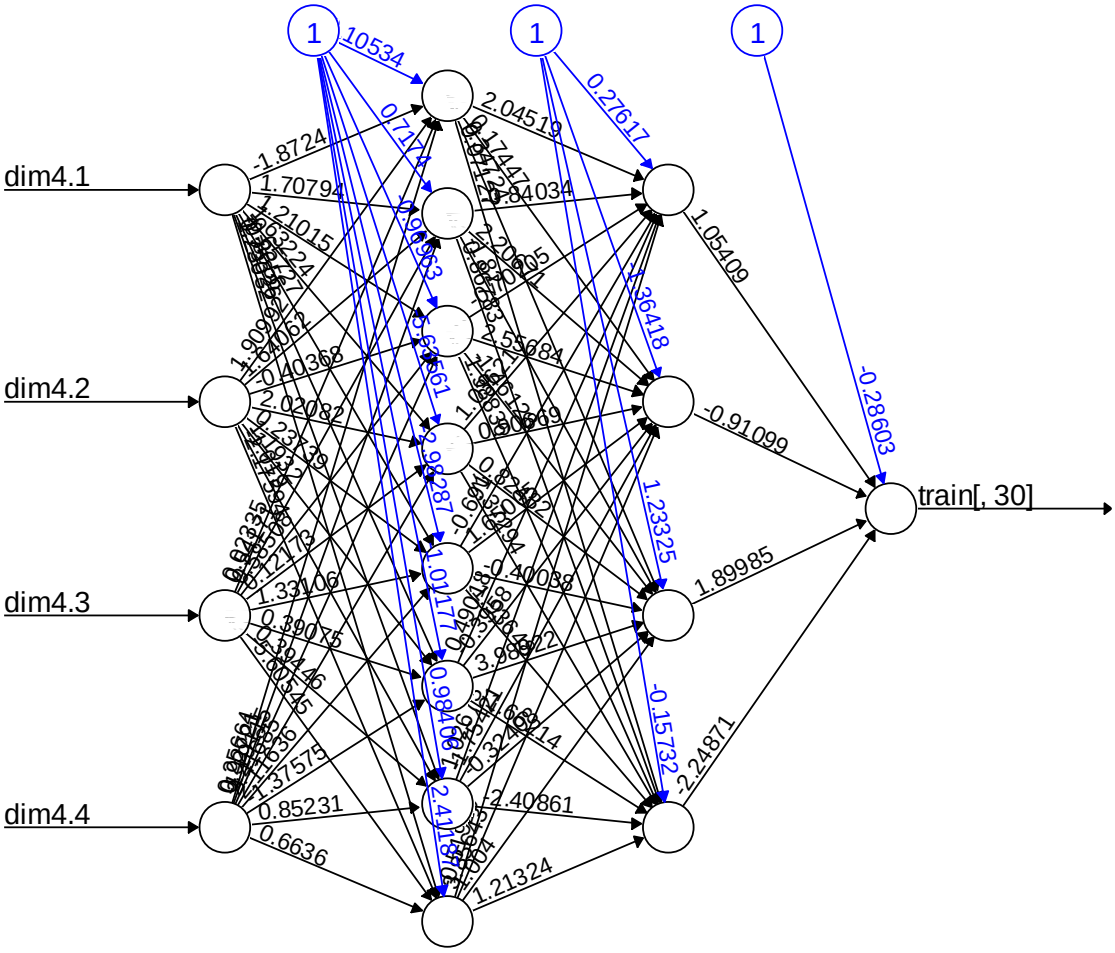
σύνολο δεδομένων φάνηκε πως τα δεδομένα μπορούν να περιγραφούν ικανοποιητικά από ένα σύνολο τεσσάρων έως δέκα διαστάσεων (βλ. εικόνα 25).



Εικόνα 25Με βάση το κανόνα elbow-rule επιλέγονται τέσσερεις συνιστώσες για την περιγραφή του συνόλου δεδομένων. Τα δεδομένα που χρησιμοποιήθηκαν είναι ένα τυχαίο δείγμα από το σύνολο δεδομένων που εξετάστηκε.

Ο τελικός αριθμός των απαραίτητων συνιστωσών για την ικανοποιητική περιγραφή των δεδομένων καθορίστηκε συγκρίνοντας δυο τεχνητά νευρωνικά δίκτυα. Κάθε ένα από τα νευρωνικά δίκτυα εκπαιδεύτηκε δέκα φορές με βάση το 90% των παρατηρήσεων, τυχαία επιλεγμένων κάθε φορά. Στην συνέχεια ελέγχθηκε το μέσο τετραγωνικό σφάλμα του μοντέλου προσπαθώντας να προβλέψει το υπόλοιπο 10% των παρατηρήσεων.

Το πρώτο έχει τέσσερεις μεταβλητές εισόδου, οι οποίες αντιστοιχούν στις δέκα συνιστώσες που προέκυψαν από τον αλγόριθμο Isomap.Το μέσο τετραγωνικό σφάλμα του μοντέλου είναι 3.899759187. Το δεύτερο νευρωνικό δίκτυο (βλ. Εικόνα 26) έχει ως μεταβλητές εισόδου τις τέσσερεις πρώτες συνιστώσες και εκπαιδεύτηκε ακριβώς με τον ίδιο τρόπο. Το μέσο τετραγωνικό σφάλμα πρόβλεψης είναι 3.432385595.

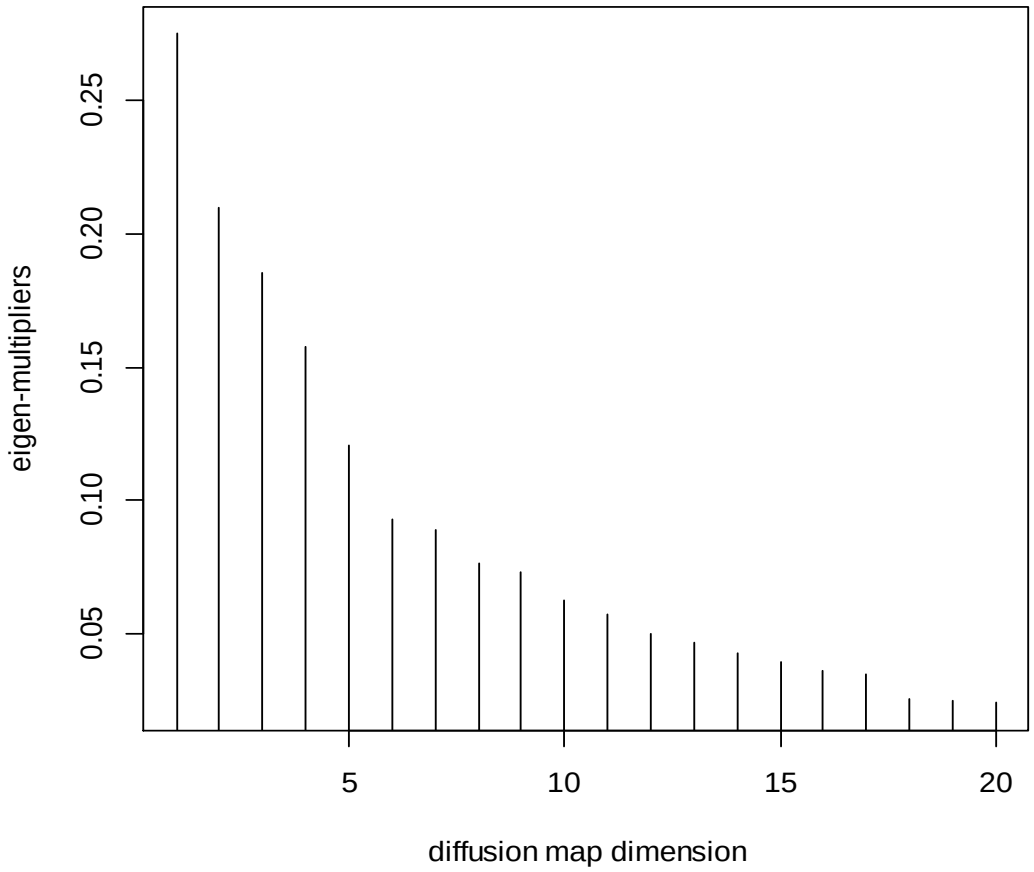


Εικόνα 26Αναπαράσταση του νευρωνικού δικτύου χρησιμοποιώντας τις 4 μεταβλητές που υπολογίζονται μέσω του αλγορίθμου ISOMAP**15**

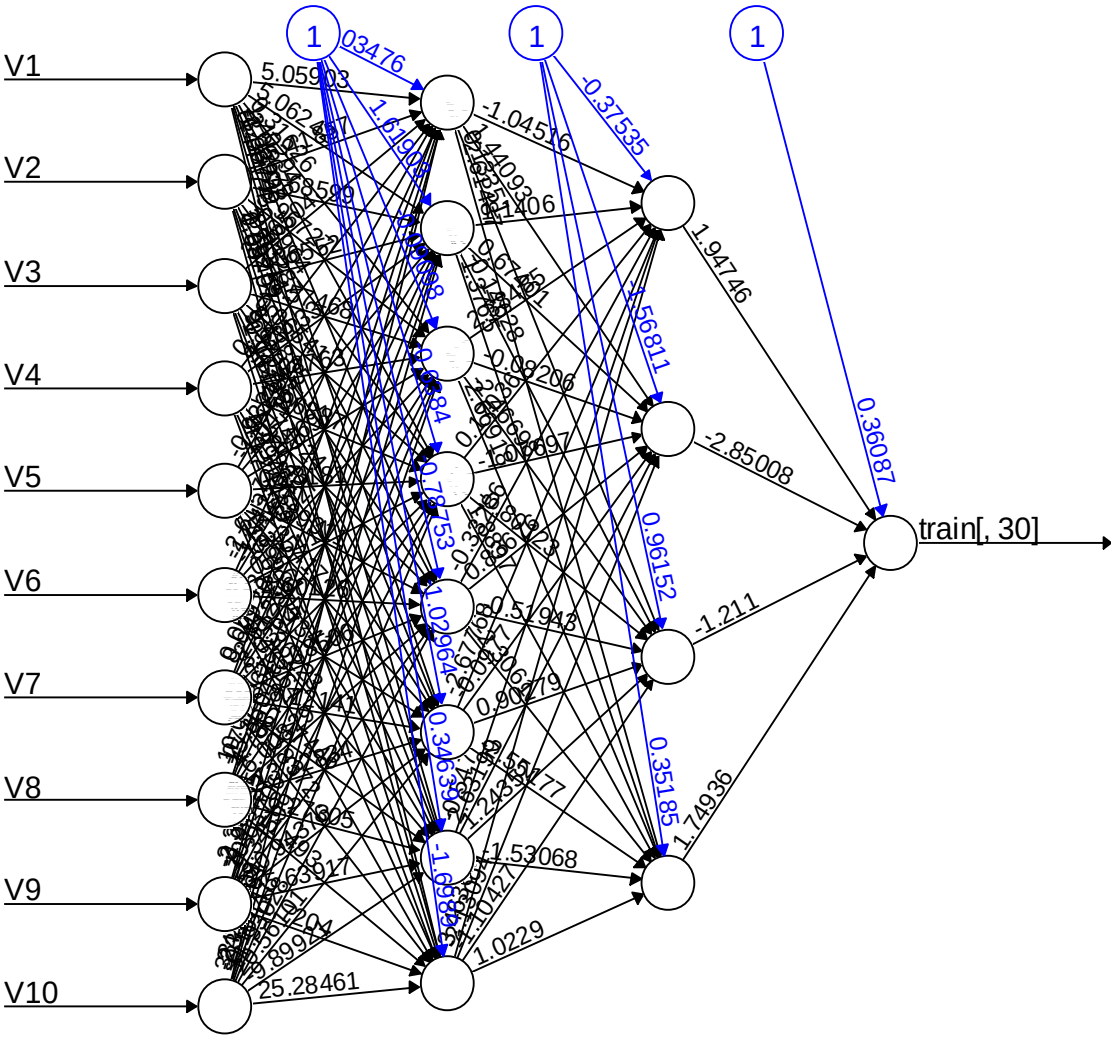
Συνεπώς είναι προφανές ότι μόλις τέσσερεις μη γραμμικές συνιστώσες είναι αρκετές για να περιγράψουν τα δεδομένα.

8.3.3 Μείωση διαστάσεων με χρήση της μεθόδου Diffusionmaps

Η δεύτερη μέθοδος που χρησιμοποιήθηκε για την ανάλυση των δεδομένων είναι η μέθοδος Diffusionmap. Χρησιμοποιώντας τα 10 πρώτα ιδιοδιανύσματα που προέκυψαν εκπαιδεύτηκε ένα νευρωνικό δίκτυο, του οποίου το μέσο τετραγωνικό σφάλμα έπειτα το cross-validation είναι 2.421940634 (βλ. Εικόνα 27).



Εικόνα 27Με βάση το κανόνα elbow-rule επιλέγονται δέκα συνιστώσες για την περιγραφή του συνόλου δεδομένων. Τα δεδομένα που χρησιμοποιήθηκαν είναι ένα τυχαίο δείγμα από το σύνολο δεδομένων που εξετάστηκε..

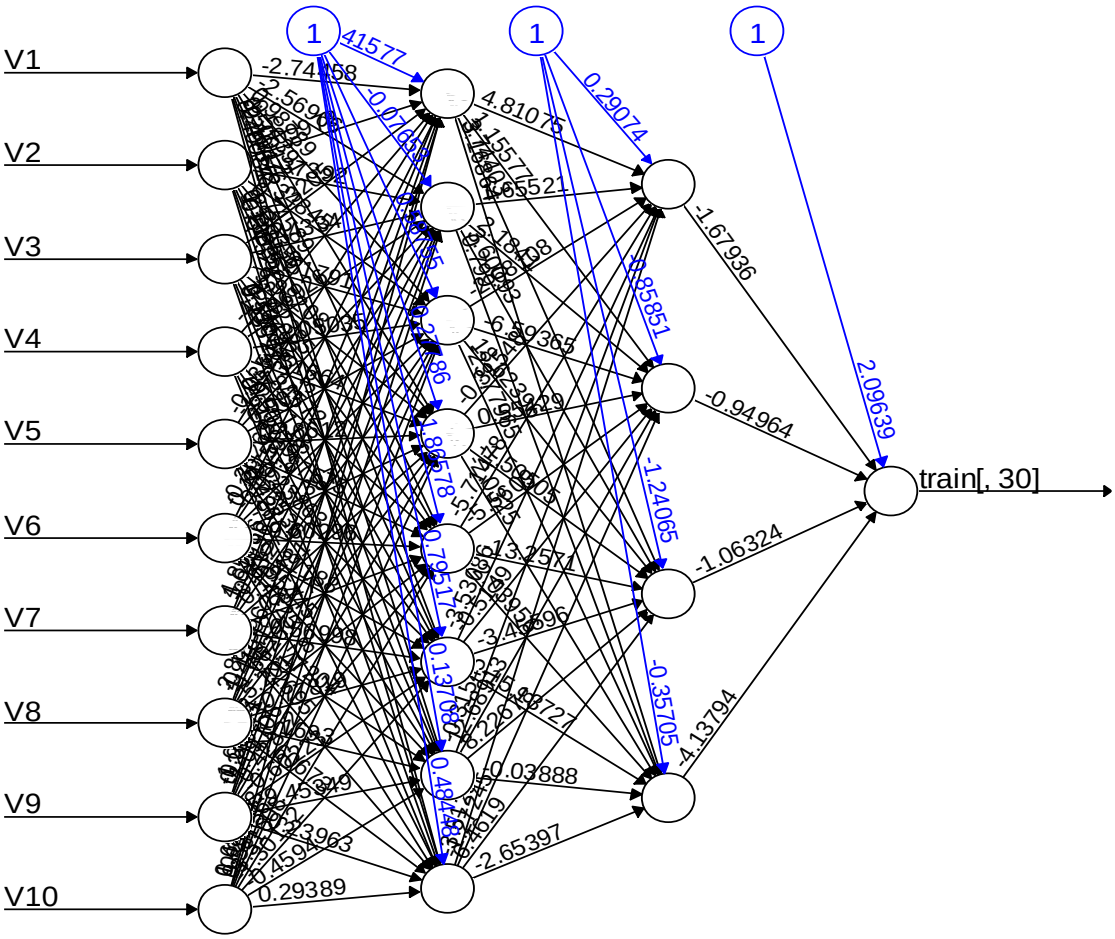


Εικόνα 28Αναπαράσταση του νευρωνικού δικτύου που προέκυψε με την χρησιμοποίηση των 10 μεταβλητών που υπολογίστηκαν με την μέθοδο των difusionmaps

8.3.4 Μείωση διαστάσεων με χρήση της μεθόδου Autoencoder

Η τελευταία μέθοδος που εφαρμόστηκε για την ανάλυση του δικτύου των τιμών των αεροπορικών εισιτηρίων η μέθοδος Autoencoder. Το νευρωνικό δίκτυο που εκπαιδεύτηκε για τον καθορισμό των διαστάσεων που περιγράφουν τα δεδομένα αποτελείται από 30 μεταβλητές εισόδου, 10 κρυφές μεταβλητές στο ενδιάμεσο στρώμα και 10 τελικές μεταβλητές εξόδου οι οποίες αντιστοιχούν στις διαστάσεις πάνω στις οποίες προβλήθηκαν τα δεδομένα.

Στη συνέχεια για να ελεγχθεί η αποτελεσματικότητα των διαστάσεων εφαρμόστηκε και πάλι η μέθοδος 10-foldcrossvalidation. Το τελικό μέσο τετραγωνικό σφάλμα που προέκυψε είναι 4.904846524.



Εικόνα 29Αναπαράσταση του νευρωνικού δικτύου που προέκυψε με την χρησιμοποίηση των 10 μεταβλητών που υπολογίστηκαν με την μέθοδο autoencoder.

Κεφάλαιο 9 Επίλογος - Συμπεράσματα

9.1 Συμπεράσματα και μελλοντικές κατευθύνσεις

Οι αμέτρητοι κανόνες κοστολόγησης ενός αεροπορικού εισιτηρίου έχουν εισαγάγει μια πολύ μεγάλη πολυπλοκότητα στον καθορισμό της τιμής οποιαδήποτε διαδρομής. Στην παρούσα εργασία εξετάστηκε ένα μικρό δείγμα αεροπορικών εισιτηρίων ως προς την ύπαρξη κάποιου απλούστερου χώρου ο οποίος περιγράφει ικανοποιητικά το δίκτυο των ναύλων.

Χρησιμοποιήθηκαν γραμμικές (PCA) και μη γραμμικές (Isomap, Diffusionmap, Autoencoder) μέθοδοι μείωσης διαστάσεων. Όλες οι τεχνικές έδειξαν πως είναι δυνατό να περιγράψουμε ικανοποιητικά το σύνολο δεδομένων. Η μέθοδος PCA έδωσε τα καλύτερα αποτελέσματα, αλλά απαιτεί τη χρήση 12 μεταβλητών. Από τις μη-γραμμικές μεθόδους ο αλγόριθμος Diffusionmap ήταν ο πιο αποτελεσματικός κατασκευάζοντας ένα νέο σύνολο δεδομένων αποτελούμενο από 10 μεταβλητές.

Μέθοδος	**Αριθμός Συνιστωσών**	**MSE**
PCA	12	1.67
Isomap	4	3.43
Diffusion map	10	2.42
Autoencoder	10	4.90
Isomap	12	3.40
Difussion map	12	1.99
Autoencoder	12	4.13

Η μεγαλύτερη μείωση διαστάσεων επιτεύχθηκε με τη χρήση της μεθόδου Isomap, σύμφωνα με την οποία αρκούν μόνο τέσσερεις μεταβλητές για περιγράψουν το σύνολο δεδομένων. Επιπλέον βλέπουμε πως ακόμα και αν επιλέξουμε 12 μεταβλητές εισόδου η απόδοση τους δεν βελτιώνεται σημαντικά. Επομένως η επιλογή της

καταλληλότερης μεθόδου είναι υποκειμενική και εξαρτάται από τη φύση του εκάστοτε προβλήματος.

Στο μέλλον θα είχε ενδιαφέρον η μελέτη ενός αντιπροσωπευτικότερου συνόλου δεδομένων και η προσπάθεια αντιστοίχησης των μεταβλητών που προέκυψαν με γραμμικούς συνδυασμούς γνωστών μεταβλητών όπως η τιμή πτήσεων προς μεγάλα αεροδρόμια, η εποχικότητα, η ημερομηνία κράτησης κ.α.

Βιβλιογραφία

1. K. Fukunaga. *Introduction to statistical pattern recognition*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.
2. N.P. Hughes and L. Tarassenko. *Novel signal shape descriptors through wavelet transforms and dimensionality reduction*. Wavelets: Applications in Signal and Image Processing X, 763–773, 2003.
3. T. Cox and M. Cox. *Multidimensional scaling*. Chapman & Hall, London, UK, 1994.
4. M.S. Venkatarajan and W. Braun. *New quantitative descriptors of amino acids based on multidimensional scaling of a large number of physical-chemical properties*. Journal of Molecular Modeling, 7(12):445–453, 2004.
5. O.C. Jenkins and M.J. Mataric. *Deriving action and behavior primitives from human motion data*. International Conference on Intelligent Robots and Systems, 3:2551–2556, 2002.
6. .B. Raytchev, I. Yoda, and K. Sakaue. *Head pose estimation by nonlinear manifold learning*. Proceedings of the 17th ICPR, 462–466, 2004.
7. L. Teng, H. Li, X. Fu, W. Chen, and I.-F. Shen. *Dimension reduction of microarray data based on local tangent space alignment*. Proceedings of the 4th IEEE International Conference on Cognitive Informatics, 2005.
8. E.W. Dijkstra. *A note on two problems in connection with graphs*. NumerischeMathematik, 1:269–271, 1959.
9. R.W. Floyd. *Algorithm 97: Shortest path*. Communications of the ACM, 5(6):345, 1962.
10. M.H. Law and A.K. Jain. *Incremental nonlinear dimensionality reduction by manifold learning*. IEEETransactions of Pattern Analysis and Machine Intelligence, 28(3):377–391, 2006.
11. A. Ng, M. Jordan, and Y. Weiss. *On spectral clustering: Analysis and an algorithm*. Advances in Neural Information Processing Systems, The MIT Press. Cambridge, MA, USA, 14:849–856, 2001.
12. D. DeMers and G. Cottrell. *Non-linear dimensionality reduction*. Advances in Neural Information Processing Systems, Morgan Kaufmann, San Mateo, CA, USA, 5:580–587, 1993.

13. H. Hoffmann. *Kernel PCA for novelty detection*. Pattern Recognition, 40(3):863–874, 2007.
14. H. Murase and S. K. Nayar. *Visual learning and recognition of 3-d objects from appearance*. International Journal of Computer Vision, 14(1):5-24, 1995.
15. J. W. McClurkin, L. M. Optican, B. J. Richmond and T. J. Gawne. *Concurrent processing and complexity of temporally encoded neuronal messages in visual perception*. Science, New Series, 253(5020):675-677, 1991.
16. C. A. L. Bailer-Jones, M. Irwin and T. von Hippel. *Physical parametrization of stellar spectra: the neural network approach*. Monthly Notices of the Royal Astronomical Society, 298(1):157-166, 1997.
17. A. H. Monahan. *Nonlinear principal component analysis by neural networks: theory and application to the Lorenz system*. Journal of Climate, 13:821-835, 2000.
18. T. Cox and M. Cox. *Multidimensional scaling*. Chapman & Hall, London, UK, 1994.
19. J.B. Tenenbaum, V. de Silva, and J.C. Langford. *A global geometric framework for nonlinear dimensionality reduction*. Science, 290(5500):2319–2323, 2000.
20. M. Belkin and P. Niyogi. *Laplacian eigenmaps for dimensionality reduction and data representation*. Neural Computation, 6(15):1373-1396, 2003.
21. S. Lafon and A.B. Lee. *Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(9):1393–1403, 2006.
22. R. Erban, T.A. Frewen, X. Wang, T.C. Elston, R. Coifman, B. Nadler and I.G. Kevrekidis. *Variable-free exploration of stochastic models: A gene regulatory network example*. The Journal of Chemical Physics, 126(15):155103, 2007.
23. R. Coifman and S. Lafon. *Diffusion maps.* Applied and Computational Harmonic Analysis, 21(1):5-30, 2006.
24. G.E. Hinton and R.R. Salakhutdinov. *Reducing the dimensionality of data with neural networks*. Science, 313(5786):504–507, 2006.
25. Θ. Αλεξόπουλος, Α. Τζαμαριουδάκη. *Στατιστική αναγνώριση προτύπων*. Σχολή Εφαρμοσμένων Μαθηματικών & Φυσικών Επιστημών, Ε.Μ.Π., 2005.
26. A.W. Donovan. *Yield Management in the Airline Industry*. Journal of Aviation/Aerospace Education & Research, 14(3), 2005.

27. O. Lordan. *Airline route networks: A complex network approach.* Doctoral thesis, UniversitatPolitècnica de Catalunya, 2014.
28. C. de Marcken. *Computational complexity of air travel planning.* Notes for MIT course 6.034, 2003.
29. R.R Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F. Warner and S.W. Zucker. *Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps*. Proceedings of the National Academy of Sciences of the United States of America, 102(21):7426-7431, 2005.
30. O. Etzioni, C. Knoblick, R. Tuchinda and A. Yates *To Buy or not to buy: Mining Airline Fare Data to Minimize Ticket Purchase Price*. Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining:119- 128, 2003.
31. A. Greenwald *The 2002 Trading Comoetition, An Overview of Trading Strategies* AI Magazine Volume 24(1): 83 – 91 2003.
32. M. Brons, E. Pels, P. Niijkamp and P. Rietveld *Price elasticities of demand for passenger air travel: a meta anlysis* Journal of Air Transport Management 8: 165-175, 2002.
33. J. K. Brueckner and W. T Whalen ThePriceEffects of International Airline Alliances. The Journal of Law & Economics 43(2): 503-546, 2002.
34. Nelwamondo F.V. and MarwalaT. *Fuzzy Artmap and Neural Network Approach to Online Processing of Inputs Missing Values* SAIEE Africa Research Journal 98(2) 45 – 51.
35. Betechuoh B. L., Tshilidzi M. and Thando T. *Autoencoder networks for HIV classification* Current Science, 91(11), 2006.
36. Sammon J.W. *A nonlinear mapping for data structure analysis* IEEE Transactions on Computers. 18 (401,402) 403–409, 1969.
37. Cauchy A. *Méthodegénérale pour la résolution des systèmesd'équationssimultanées*. Pp 536–538, 1847.
38. Linnainmaa S. Taylor *expansion of the accumulated rounding error* BIT Numerical Mathematics 16(2): 146–160, 1976.
39. Van Der Maaten, Laurens, Eric Postma, and Jaap Van den Herik*Dimensionality reduction: a comparative review* J Mach Learn Res 10 (2009): 66-71.

Παράρτημα: Κώδικας R

```
dests<-list.files(getwd())
########## ticketsTotable function ############
ticketsToTable<-function(x){

d<-rep(0,(length(x))*35)
d<-matrix(d,ncol=35)

for(i in 1:length(x)){
temp<-read.fwf(file=x[i],widths=c(5),col.names=c("price"),n=35)
d[i,]<-as.numeric(unlist(temp))
}
return(d)
}
################################################
destTable<-ticketsToTable(dests)
destTable<-t(destTable)

destTableframe<-as.data.frame(destTable)
colnames(destTableframe)<-dests
#normalize the features
ndat<-scale(destTableframe)
destTableframe<-ndat
ntab<- scale(destTable)
#run pca algorithm
pca<-prcomp(ndat,center=TRUE,scale=TRUE)
summary(pca)
v<-pca$x[,1:12]
```

```
pam<-pam(v,12)

#################### isomaps ####################
#install the package

library(RDRToolbox)


isodata = Isomap(data=destTable, dims=1:30, k=10,plotResiduals=TRUE)

################## diffusion maps ##################

#install to paketodiffusionMap
library(diffusionMap)

D = dist(destTable)
dmap = diffuse(D,neigen=20)
plot(1:20,dmap$eigenmult,typ='h',xlab="diffusion map dimension",ylab="eigen-multipliers")

####### example diffusion map ######################
## example with noisy spiral
n=2000
t=runif(n)^.7*10
al=.15;bet=.5;
x1=bet*exp(al*t)*cos(t)+rnorm(n,0,.1)
y1=bet*exp(al*t)*sin(t)+rnorm(n,0,.1)
plot(x1,y1,pch=20,main="Noisy spiral")
D = dist(cbind(x1,y1))
dmap = diffuse(D,neigen=10) # compute diffusion map
par(mfrow=c(2,1))
```

```
plot(t,dmap$X[,1],pch=20,axes=FALSE,xlab="spiral parameter",ylab="1st diffusion coefficient")

box()

plot(1:10,dmap$eigenmult,typ='h',xlab="diffusion map dimension",ylab="eigen-multipliers")

#########################################################


########## autoencoders ##############################


library(autoencoder)

nl=3 ## number of layers (default is 3: input, hidden, output)

unit.type = "logistic" ## specify the network unit type, i.e., the unit's

## activation function ("logistic" or "tanh")

Nx.patch=10 ## width of training image patches, in pixels

Ny.patch=10 ## height of training image patches, in pixels

N.input = Nx.patch*Ny.patch ## number of units (neurons) in the input layer (one unit per pixel)

N.hidden = 10 ## number of units in the hidden layer

lambda = 0.0002 ## weight decay parameter

beta = 6 ## weight of sparsity penalty term

rho = 0.01 ## desired sparsity parameter

epsilon<- 0.001 ## a small parameter for initialization of weights

## as small gaussian random numbers sampled from N(0,epsilon^2)

max.iterations = 1000 ## number of iterations in optimizer

## Train the autoencoder on training.matrix using BFGS optimization method

## (see help('optim') for details):

## WARNING: the training can take as long as 20 minutes for this dataset!

## Not run:

autoencoder.object<- autoencode(X.train=destTable,nl=nl,N.hidden=N.hidden,

unit.type=unit.type,lambda=lambda,beta=beta,rho=rho,epsilon=epsilon,

optim.method="BFGS",max.iterations=max.iterations,

rescale.flag=TRUE,rescaling.offset=0.001)
```

```
#### VALIDATING EACH METHODS' OUTPUT ################

#### Cross validation for each method ####
#### PCA #####



library(neuralnet)
final.dat<- cbind(destTableframe,v)
index<- sample(1:nrow(destTableframe),round(0.75*nrow(destTableframe)))
train<- final.dat[index,]
test<- final.dat[-index,]
f <- as.formula(paste("train[,1]~",paste(colnames(final.dat)[31:42],collapse="+")))
nn<- neuralnet(f , data = train ,hidden=c(8,4),linear.output = T)
pr.nn<- compute(nn,test[,31:42])

test.r<- test[,1]
MSE.nn<- sum((test.r - pr.nn$net.result)^2)/nrow(test)

set.seed(454)
er<-c(1:30)

for(j in 1:30){
        cv.error<- NULL
        k <- 10
        for(i in 1:k){
                index<- sample(1:nrow(destTableframe),round(0.9*nrow(destTableframe)))
                train.cv <- final.dat[index,]
                test.cv <- final.dat[-index,]
                f <-
as.formula(paste("train[,",j,"]~",paste(colnames(final.dat)[31:42],collapse="+")))
```

```
                    nn<- neuralnet(f , data = train ,hidden=c(8,4),linear.output = T)
                    pr.nn<- compute(nn,test[,31:42])
                test.cv.r<- test[,j]
                cv.error[i] <- sum((test.cv.r - pr.nn$net.result)^2)/nrow(test.cv)
                }
        print(mean(cv.error))
        er[j]<-mean(cv.error)
}
mean(er)


#ISOMAPS##
library(RDRToolbox)
isodata = Isomap(data=ntab, dims=1:30, k=10,plotResiduals=TRUE)
isomap = Isomap(data=ntab, dims=4, k=10)
final.dat<- cbind(destTableframe,as.data.frame(isomap))
index<- sample(1:nrow(destTableframe),round(0.75*nrow(destTableframe)))
train<- final.dat[index,]
test<- final.dat[-index,]
f <- as.formula(paste("train[,1]~",paste(colnames(final.dat)[31:34],collapse="+")))
nn<- neuralnet(f , data = train ,hidden=c(8,4),linear.output = T)
pr.nn<- compute(nn,test[,31:34])
test.r<- test[,1]
MSE.nn<- sum((test.r - pr.nn$net.result)^2)/nrow(test)

set.seed(343)
er<-c(1:30)
for(j in 1:30){
        cv.error<- NULL
        k <- 10
        for(i in 1:k){
```

```
                index<- sample(1:nrow(destTableframe),round(0.9*nrow(destTableframe)))
                   train.cv <- final.dat[index,]
                   test.cv <- final.dat[-index,]
                   f <-
as.formula(paste("train[,",j,"]~",paste(colnames(final.dat)[31:34],collapse="+")))
                         nn<- neuralnet(f , data = train ,hidden=c(8,4),linear.output = T)
                         pr.nn<- compute(nn,test[,31:34])
                test.cv.r<- test[,j]
                cv.error[i] <- sum((test.cv.r - pr.nn$net.result)^2)/nrow(test.cv)


                }
        print(mean(cv.error))
        er[j]<-mean(cv.error)
}
mean(er)


## Diffussion maps
library(diffusionMap)
D = dist(ntab)
dmap = diffuse(D,neigen=20)
plot(1:20,dmap$eigenmult,typ='h',xlab="diffusion map dimension",ylab="eigen-multipliers")
final.dat<- cbind(destTableframe,as.data.frame(dmap$X[,1:10])) #xrhsimopoiw 10 dimension
afouthelwna tis meiwsw. Me 20 eixeakraia results stokross validation.
index<- sample(1:nrow(destTableframe),round(0.75*nrow(destTableframe)))
train<- final.dat[index,]
test<- final.dat[-index,]
f <- as.formula(paste("train[,1]~",paste(colnames(final.dat)[31:40],collapse="+")))
nn<- neuralnet(f , data = train ,hidden=c(8,4),linear.output = T)
pr.nn<- compute(nn,test[,31:40])
test.r<- test[,1]
MSE.nn<- sum((test.r - pr.nn$net.result)^2)/nrow(test)
```

```
set.seed(558)
er<-c(1:30)
for(j in 1:30){
        cv.error<- NULL
        k <- 10
        for(i in 1:k){
                index<- sample(1:nrow(destTableframe),round(0.9*nrow(destTableframe)))
                   train.cv <- final.dat[index,]
                   test.cv <- final.dat[-index,]
                   f <-
as.formula(paste("train[,",j,"]~",paste(colnames(final.dat)[31:40],collapse="+")))
                        nn<- neuralnet(f , data = train ,hidden=c(8,4),linear.output = T)
                        pr.nn<- compute(nn,test[,31:40])
                test.cv.r<- test[,j]
                cv.error[i] <- sum((test.cv.r - pr.nn$net.result)^2)/nrow(test.cv)
                }
        print(mean(cv.error))
        er[j]<-mean(cv.error)
}
mean(er)


## AUTOENCODERS ####
library(autoencoder)
nl=3 ## number of layers (default is 3: input, hidden, output)
unit.type = "logistic" ## specify the network unit type, i.e., the unit's
## activation function ("logistic" or "tanh")
Nx.patch=10 ## width of training image patches, in pixels
Ny.patch=10 ## height of training image patches, in pixels
N.input = Nx.patch*Ny.patch ## number of units (neurons) in the input layer (one unit per
pixel)
N.hidden = 10 ## number of units in the hidden layer
lambda = 0.0002 ## weight decay parameter
```

```
beta = 6 ## weight of sparsity penalty term
rho = 0.01 ## desired sparsity parameter
epsilon<- 0.001 ## a small parameter for initialization of weights
## as small gaussian random numbers sampled from N(0,epsilon^2)
max.iterations = 1000 ## number of iterations in optimizer
## Train the autoencoder on training.matrix using BFGS optimization method
## (see help('optim') for details):
## WARNING: the training can take as long as 20 minutes for this dataset!
## Not run:
autoencoder.object<- autoencode(X.train=ntab,nl=nl,N.hidden=N.hidden,
unit.type=unit.type,lambda=lambda,beta=beta,rho=rho,epsilon=epsilon,
optim.method="BFGS",max.iterations=max.iterations,
rescale.flag=TRUE,rescaling.offset=0.001)


final.dat<- cbind(destTableframe,as.data.frame(autoencoder.object$W[[2]]))
index<- sample(1:nrow(destTableframe),round(0.75*nrow(destTableframe)))
train<- final.dat[index,]
test<- final.dat[-index,]
f <- as.formula(paste("train[,1]~",paste(colnames(final.dat)[31:40],collapse="+")))
nn<- neuralnet(f , data = train ,hidden=c(8,4),linear.output = T)
pr.nn<- compute(nn,test[,31:40])
test.r<- test[,1]
MSE.nn<- sum((test.r - pr.nn$net.result)^2)/nrow(test)

set.seed(558)
er<-c(1:30)
for(j in 1:30){
        cv.error<- NULL
        k <- 10
        for(i in 1:k){
```

```
                    index<- sample(1:nrow(destTableframe),round(0.9*nrow(destTableframe)))
                       train.cv <- final.dat[index,]
                       test.cv <- final.dat[-index,]
                       f <-
as.formula(paste("train[,",j,"]~",paste(colnames(final.dat)[31:40],collapse="+")))
                              nn<- neuralnet(f , data = train ,hidden=c(8,4),linear.output = T)
                              pr.nn<- compute(nn,test[,31:40])
                    test.cv.r<- test[,j]
                    cv.error[i] <- sum((test.cv.r - pr.nn$net.result)^2)/nrow(test.cv)
                    }
          print(mean(cv.error))
}
mean(er)



library(SAENET)
ae<- SAENET.train(ntab, n.nodes=c(8,10), lambda = 1e-5, beta = 1e-5, rho = 0.01, epsilon =
0.01)
reduced_data<- SAENET.predict(ae, ntab, all.layers = TRUE)
learned_representation = reduced_data[[2]]$X.output



final.dat<- cbind(destTableframe,as.data.frame(learned_representation))
index<- sample(1:nrow(destTableframe),round(0.75*nrow(destTableframe)))
train<- final.dat[index,]
test<- final.dat[-index,]
f <- as.formula(paste("train[,1]~",paste(colnames(final.dat)[31:40],collapse="+")))
nn<- neuralnet(f , data = train ,hidden=c(8,4),linear.output = T)
pr.nn<- compute(nn,test[,31:40])
test.r<- test[,1]
MSE.nn<- sum((test.r - pr.nn$net.result)^2)/nrow(test)
```

```
set.seed(524)
for(j in 1:30){
        cv.error<- NULL
        k <- 10
        for(i in 1:k){
                index<- sample(1:nrow(destTableframe),round(0.9*nrow(destTableframe)))
                   train.cv <- final.dat[index,]
                   test.cv <- final.dat[-index,]
                   f <-
as.formula(paste("train[,",j,"]~",paste(colnames(final.dat)[31:40],collapse="+")))
                        nn<- neuralnet(f , data = train ,hidden=c(8,4),linear.output = T)
                        pr.nn<- compute(nn,test[,31:40])
                test.cv.r<- test[,j]
                cv.error[i] <- sum((test.cv.r - pr.nn$net.result)^2)/nrow(test.cv)
                }
        print(mean(cv.error))
        er[j]<-mean(cv.error)
}### THE END ###
```