ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Σχολη Χρηματοοικονομικης και Στατιστικης

Τμημα Στατιστικης και Ασφαλιστικης Επιστημης

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ
ΣΤΗΝ **ΕΦΑΡΜΟΣΜΕΝΗ ΣΤΑΤΙΣΤΙΚΗ**

ΟΜΑΔΟΠΟΙΗΣΗ ΜΕΓΑΛΗΣ ΚΛΙΜΑΚΑΣ ΔΕΔΟΜΕΝΩΝ ΣΤΗΝ ΠΛΑΤΦΟΡΜΑ SPARK

Στέφανος Αυδάλας

Διπλωματική Εργασία
που υποβλήθηκε στο Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς ως μέρος των απαιτήσεων για την απόκτηση του Μεταπτυχιακού Διπλώματος Ειδίκευσης στην Εφαρμοσμένη Στατιστική

Πειραιάς
Ιούνιος 2018

Η παρούσα Διπλωματική Εργασία εγκρίθηκε ομόφωνα από την Τριμελή Εξεταστική Επιτροπή που ορίσθηκε από τη ΓΣΕΣ του Τμήματος Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς στην υπ' αριθμ. …….. συνεδρίασή του σύμφωνα με τον Εσωτερικό Κανονισμό Λειτουργίας του Προγράμματος Μεταπτυχιακών Σπουδών στην Εφαρμοσμένη Στατιστική.

Τα μέλη της Επιτροπής ήταν:
- Ν. Πελέκης - Επίκ. Καθηγητής. (Επιβλέπων)
- Ι. Θεοδωρίδης - Καθηγητής.
- Μ. Κούτρας - Καθηγητής.

Η έγκριση της Διπλωματική Εργασίας από το Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς δεν υποδηλώνει αποδοχή των γνωμών του συγγραφέα.

UNIVERSITY OF PIRAEUS

SCHOOL OF FINANCE AND STATISTICS

DEPARTMENT OF STATISTICS AND INSURANCE SCIENCE

POSTGRADUATE PROGRAM IN
APPLIED STATISTICS

LARGE-SCALE DATA CLUSTERING IN SPARK

Stefanos Avdalas

MSc Dissertation
submitted to the Department of Statistics and Insurance Science of the University of Piraeus in partial fulfilment of the requirements for the degree of Master of Science in Applied Statistics

Piraeus
June 2018

Στους γονείς μου,
Ιωάννη και Ξανθή.

Ευχαριστίες

Η παρούσα Διπλωματική Εργασία αποτελεί το τελικό κομμάτι του Μεταπτυχιακού Προγράμματος στην Εφαρμοσμένη Στατιστική του Πανεπιστημίου Πειραιώς. Ένα κομμάτι που χρειάστηκε χρόνο, προσπάθεια αλλά και αρκετή βοήθεια, στήριξη και καθοδήγηση από κάποιους ανθρώπους.

Ακριβώς για αυτό το λόγο, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της διπλωματικής εργασίας, Επίκουρο Καθηγητή Πελέκη Νικόλαο, αρχικά για την ανάθεση ενός θέματος με ιδιαίτερο ενδιαφέρον αλλά και για τη γενικότερη καθοδήγηση καθ' όλη τη χρονική περίοδο που χρειάστηκε για την περάτωση της.

Στη συνέχεια θα ήθελα να ευχαριστήσω τον υποψήφιο διδάκτορα Ταμπάκη Παναγιώτη και τους εργαστηριακούς βοηθούς Πέτρου Πέτρο και Σιδερίδη Στυλιανό για την καταλυτική τους βοήθεια στο κομμάτι της ανάλυσης που υπάρχει στην εργασία.

Τις ευχαριστίες μου εκφράζω και στους καθηγητές Θεοδωρίδη Ιωάννη και Κούτρα Μάρκο για τις πολύτιμες παρεμβάσεις τους.

Ιδιαίτερες ευχαριστίες απευθύνω στη Μαριλένα για τη σημαντική της στήριξη αλλά και για τη βοήθεια της όλον αυτό τον καιρό.

Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου, Ιωάννη και Ξανθή, όχι μόνο για τη βοήθεια τους στο χρονικό πλαίσιο της εργασίας αλλά για όλα όσα έχουν κάνει τόσα χρόνια.

Περιεχόμενα

Κατάλογος πινάκων

Κατάλογος σχημάτων

Περίληψη

Η σύγχρονη εποχή την οποία διανύουμε χαρακτηρίζεται ως εποχή των ¨Μεγάλων Δεδομένων¨ εξαιτίας της αύξησης των δεδομένων που παράγονται καθημερινά. Τα δεδομένα αυτά πλέον αποτελούν τη βασική πηγή εξόρυξης γνώσης. Πρόσφατες εκτιμήσεις αναφέρουν ότι ο όγκος των δεδομένων που παράγονται κάθε δύο μέρες είναι ίσος με το πλήθος των δεδομένων που έχουν δημιουργηθεί από την αρχή της ανθρωπότητας ως το 2003.

Για την ανάλυση των δεδομένων αυτών, τα παραδοσιακά εργαλεία ανάλυσης δεδομένων δεν αρκούν για τέτοιου είδους διεργασίες. Έτσι συνεχώς δημιουργούνται νέα εργαλεία για την ανάλυση των δεδομένων μεγάλης κλίμακας.

Βασισμένοι σε αυτές τις ανάγκες, η παρούσα εργασία έχει θέμα την Ομαδοποίηση Μεγάλης Κλίμακας Δεδομένων στην Πλατφόρμα Spark. Στο πρώτο κεφάλαιο ο αναγνώστης εισάγεται στην έννοια των Δεδομένων Μεγάλης Κλίμακας. Πιο συγκεκριμένα παρουσιάζονται η εξέλιξη και οι προκλήσεις αυτών, καθώς και οι τρόποι παραγωγής και απόκτησής τους.

Στο δεύτερο κεφάλαιο εισάγεται η έννοια της Ομαδοποίησης. Παρουσιάζονται αναλυτικά τα μέτρα απόστασης και ομοιότητας για κάθε μορφής δεδομένα, όπως και αυτά που χρησιμοποιούμε για την περίπτωση των συστάδων. Στη συνέχεια αναφέρονται οι κατηγορίες αλγορίθμων ομαδοποίησης, όπως και οι κατηγορίες που εφαρμόζονται για την ομαδοποίηση δεδομένων μεγάλης κλίμακας.

Στο επόμενο κεφάλαιο παρουσιάζεται η πλατφόρμα Spark, η οποία χρησιμοποιείται ευρέως για ανάλυση δεδομένων μεγάλης κλίμακας. Ειδικότερα αναφέρονται αναλυτικά οι συνιστώσες της πλατφόρμας καθώς και οι διάφορες βιβλιοθήκες της, μεταξύ των οποίων και οι MLlib και PySpark, οι οποίες χρησιμοποιούνται για την ανάλυση που γίνεται στην παρούσα εργασία.

Στο τελευταίο κεφάλαιο περιγράφονται και συγκρίνονται τα αποτελέσματα που έδωσαν οι αλγόριθμοι ομαδοποίησης που βρίσκονται στη βιβλιοθήκη MLlib μέσα από διάφορα μέτρα αξιολόγησης που υπολογίστηκαν για κάθε περίπτωση.

Abstract

The modern age we live in is characterized as a ”Big Data” era due to the increase in daily produced data. These data are now the basic source of mining knowledge. Recent estimates indicate that the volume of data produced every two days is equal to the number of data created since the beginning of mankind until 2003.

For the analysis of these data, traditional data analysis tools are not sufficient for such processes. New tools for analyzing large-scale data are thus constantly being developed.

Based on these needs, the present dissertation deals with the Large Data Clustering on the Spark Platform. In the first chapter the reader is introduced into the concept of Large-Scale Data. More specifically, we present their development and challenges, as well as the ways that they can be produced and acquired.

The second chapter introduces the concept of Clustering. We present the distance and similarity measures for each type of data, as well as the measures used for clusters. The categories of clustering algorithms are listed below, as well as the categories used for clustering large scale data.

In the next chapter we present the Spark platform, which is widely used for large-scale data analysis. More specifically, the components of the platform as well as its various libraries are presented, including MLlib and PySpark, which are used for the analysis in this dissertation.

The last chapter describes and compares the results of the clustering algorithms found in the MLlib library through various evaluation measures calculated for each case.

Κεφάλαιο 1

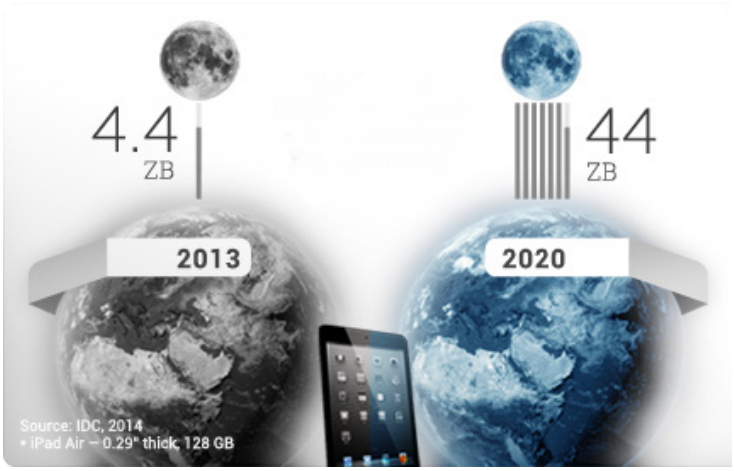
Δεδομένα Μεγάλης Κλίμακας

1.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει μια παρουσίαση των Δεδομένων Μεγάλης Κλίμακας. Θα δούμε τις βασικές ιδιότητες που τα χαρακτηρίζουν καθώς και την εξέλιξή τους. Θα παρουσιάσουμε τις προκλήσεις που καλούμαστε να αντιμετωπίσουμε στην αλληλεπίδραση μας με τα δεδομένα μεγάλης κλίμακας. Τέλος, θα δούμε πως παράγονται αλλά και πως αποκτώνται.

1.2 Τι είναι τα Δεδομένα Μεγάλης Κλίμακας

Τα τελευταία 20 χρόνια τα δεδομένα που έχουμε στην διάθεσή μας έχουν αυξηθεί σε μια πολύ μεγάλη κλίμακα στους περισσότερους τομείς. Σύμφωνα με το IDC (*International Data Comporation*) το 2011 ο συνολικός όγκος δεδομένων που δημιουργήθηκαν παγκοσμίως ήταν 1,8ZB ($\approx 10^{21}$B) ενώ το 2020 αναμένεται να είναι 44ZB. (βλέπε [11]). Στο Σχήμα 1.1 που ακολουθεί, βλέπουμε πως εάν τοποθετούσαμε όλα τα δεδομένα που είχαμε στην διάθεση μας το 2013, σε tablet, θα δημιουργούσαμε μια στοίβα η οποία θα κάλυπτε τα δύο τρίτα της απόστασης Γη-Σελήνη. Το 2020 όμως θα δημιουργηθούν 6,6 στοίβες από την Γη στη Σελήνη.



Σχήμα 1.1: Διαφορά του μεγέθους των δεδομένων από το 2013 στο 2020.

Ο όρος Δεδομένα Μεγάλης Κλίμακας (*Big Data* ή *Large-Scale Data*) δημιουργήθηκε μετά τη μαζική αύξηση δεδομένων παγκοσμίως. Κυρίως χρησιμοποιείται για την περιγραφή τεράστιων συνόλων από δεδομένα, στα οποία δεν μπορούμε να χρησιμοποιήσουμε τα παραδοσιακά εργαλεία πληροφορικής και λογισμικού για την απόκτηση, την επεξεργασία και την ανάλυσή τους, μέσα σε ένα ανεκτό πλαίσιο χρόνου. Οι Balzdek και Hathaway πρότειναν την κατηγοριοποίηση των δεδομένων μεγάλης κλίμακας όπως φαίνεται στον Πίνακα 1.1 (βλέπε [12]). Επιπρόσθετα, μέσω των δεδομένων μεγάλης κλίμακας μπορούμε να αποκτήσουμε μια βαθύτερη γνώση σε σχέση με κρυφές συσχετίσεις μεταξύ των μεταβλητών μας.

Κατηγορίες Δεδομένων Μεγάλης Κλίμακας					
Μέγεθος σε Bytes	10^6	10^8	10^{10}	10^{12}	$10^{>12}$
Κατηγορία	Medium	Large	Huge	Monster	Very Large

Πίνακας 1.1: Η κατηγοριοποίηση των Balzdek και Hathaway για τα δεδομένα μεγάλης κλίμακας.

Μέχρι σήμερα, τα δεδομένα μεγάλης κλίμακας έχουν εγείρει το ενδιαφέρον των επιχειρήσεων, της ακαδημαϊκής κοινότητας αλλά και των κυβερνητικών αρχών. Πλέον μπορούμε να πούμε πως η εποχή των δεδομένων μεγάλης κλίμακας έχει έρθει, πέρα από κάθε αμφιβολία.

Καθώς τα δεδομένα μεγάλης κλίμακας απασχολούν τόσους διαφορετικούς τομείς, οι ορισμοί που μπορεί να αποδώσει ο κάθε ένας μπορεί να διαφέρουν σε κάποια σημεία. Η κοινή τους τομή όμως είναι:

1. η **Ταχύτητα**: με τον όρο αυτό εννοούμε την ταχύτητα με την οποία συσσωρεύονται τα δεδομένα,
2. ο **Όγκος**: με τον όρο αυτό εννοούμε την κλίμακα των δεδομένων όπως και το πλήθος των δεδομένων που αποθηκεύονται,
3. η **Ποικιλία**: με τον όρο αυτό εννοούμε την διαφορετικότητα που υπάρχει στα δεδομένα και
4. η **Εγκυρότητα**: με τον όρο αυτό εννοούμε το κατά πόσο τα δεδομένα μας είναι αληθή ή όχι.

Αυτά αποτελούν τα «4V» των δεδομένων μεγάλης κλίμακας. Τα «4V» προέρχονται από τις λέξεις *velocity, volume, variety,* και *veracity*.

Βέβαια, ένα ακόμα «V» που έρχεται να προστεθεί είναι η **Αξία** (*value*), δηλαδή το πως θα μετατραπούν τα δεδομένα σε αξία. Η αξία αυτή δεν είναι μόνο το οικονομικό κέρδος αλλά και ιατρικό ή κοινωνικό κέρδος. Ένα παράδειγμα είναι μια έρευνα της *Google* το 2009 για την καταπολέμηση μιας πανδημίας γρίπης χρησιμοποιώντας ανάλυση δεδομένων μεγάλης κλίμακας. (βλέπε [9])

1.2.1 Η Εξέλιξη των Δεδομένων Μεγάλης Κλίμακας

Στα τέλη της δεκαετίας του 1970 άρχισε να εμφανίζεται η ιδέα των «μηχανών βάσεων δεδομένων», μια τεχνολογία η οποία θα ειδικευόταν στην αποθήκευση αλλά και στην ανάλυση δεδομένων. Με την αύξηση του πλήθους των δεδομένων, ο χώρος αλλά και η υπολογιστική ισχύς ενός μηχανήματος δεν αρκούσε. Έτσι τη δεκαετία του 1980 προτάθηκε το «share nothing», ένα παράλληλο σύστημα για βάσεις δεδομένων. Βέβαια, η πρώτη επιτυχημένη προσπάθεια για τη δημιουργία ενός παραλλήλου συστήματος για βάσεις δεδομένων υλοποιήθηκε από την *Teradata* το 1986, όπου το σύστημα είχε χωρητικότητα 1TB. Από την δεκαετία του 1990 μέχρι και σήμερα η εξέλιξη στον συγκεκριμένο τομέα είναι δραματική.

Η ανάπτυξη των υπηρεσιών μέσω του διαδικτύου και η συνεχώς αναπτυσσόμενη τεράστια αποθηκευτική ικανότητα που υπάρχει έχει διεγείρει το ενδιαφέρον σχεδόν όλων των μεγάλων εταιριών. Οι *EMC, Oracle, IBM, Microsoft, Google, Amazon, Facebook* και άλλες, έχουν αρχίσει να επενδύουν σημαντικά στα δεδομένα μεγάλης κλίμακας. Ένα παράδειγμα είναι η επένδυση της IBM από το 2005 σε τεχνολογίες σχετικές με δεδομένα μεγάλης κλίμακας, ύψους 16 δισεκατομμυρίων δολαρίων.

Βέβαια και η ακαδημαϊκή κοινότητα δεν έμεινε ασυγκίνητη από την ραγδαία ανάπτυξη των δεδομένων μεγάλης κλίμακας. Πλέον, πανεπιστήμια σε όλο τον κόσμο έχουν εντάξει την ανάλυση μεγάλης κλίμακας δεδομένων στο πρόγραμμα σπουδών τους.

Τέλος σε επίπεδο κρατών, κυβερνήσεις όπως αυτές των Η.Π.Α. και της Ιαπωνίας έχουν επενδύσει στην έρευνα και την ανάλυση δεδομένων μεγάλης κλίμακας. Τα Υπουργεία Εσωτερικών τους πιστεύουν πως μέσω της ανάλυσης δεδομένων μεγάλης κλίμακας θα παρέχουν περισσότερα στους πολίτες και ταυτόχρονα θα τους προστατεύουν.

1.2.2 Προκλήσεις των Δεδομένων Μεγάλης Κλίμακας

Η ταχύτατη πλημμύρα από δεδομένα που μας κατακλύζει καθημερινά, εκτός από τα θετικά που μπορεί να φέρει, δημιουργεί και ένα πλήθος από προκλήσεις. Προκλήσεις όσον αφορά την απόκτηση, την αποθήκευση, την διαχείριση και την ανάλυση των δεδομένων μεγάλης κλίμακας. Μερικές από αυτές τις προκλήσεις είναι:

- Αναπαράσταση δεδομένων: πολλές βάσεις δεδομένων παρουσιάζουν μια ανομοιογένεια στην δομή τους. Στόχος της αναπαράστασης δεδομένων είναι να κάνει τα δεδομένα καλύτερα για την ανάλυσή τους αλλά και περισσότερο κατανοητά στον χρήστη. Παρόλα αυτά, υπάρχει ο κίνδυνος να χαθεί κάποια σημαντική πληροφορία από τα δεδομένα μας.

- Διαχείριση κύκλου ζωής δεδομένων: παρά την συνεχώς αυξανόμενη δυνατότητα αποθήκευσης δεδομένων, δεδομένα δημιουργούνται από διάφορες πήγες σε απίστευτα μεγάλες ποσότητες. Κατά αυτό τον τρόπο πολλές φορές πρέπει να παρθεί η απόφαση κατά πόσο θα δημιουργηθούν νέοι αποθηκευτικοί χώροι ή κάποια δεδομένα θα πρέπει να διαγραφούν.

- Εμπιστευτικότητα δεδομένων: πολλές εταιρίες ή χρήστες δεν έχουν την δυνατότητα να αποκτήσουν και να επεξεργαστούν τεράστιες βάσεις δεδομένων. Έτσι αναγκάζονται να απευθυνθούν σε μεγάλες εταιρίες ή οργανισμούς. Όπως είναι κατανοητό κάτι τέτοιο μπορεί να περιέχει πιθανά ρίσκα, καθώς κάποιες ευαίσθητες πληροφορίες μπορεί να γίνουν γνωστές σε κάποιον τρίτο. Άρα σε περιπτώσεις όπως αυτές θα πρέπει πρώτα να γίνονται διεργασίες για την διαφύλαξη των προσωπικών δεδομένων.

- Διαχείριση ενέργειας: για την διατήρηση τόσο μεγάλων βάσεων δεδομένων καθημερινά καταναλώνονται τεράστια ποσά ενέργειας. Αυτομάτως δημιουργούνται δύο ζητήματα, ένα από οικονομικής πλευράς και ένα από οικολογικής. Πλέον ευνοούνται μέθοδοι για την καταπολέμηση και των δύο ζητημάτων.

- Ευελιξία και εξελικτικότητα: τα σημερινά συστήματα ανάλυσης δεδομένων θα πρέπει να είναι ικανά να υποστηρίξουν παρόντα αλλά και μελλοντικά δεδομένα. Έτσι ένας αλγόριθμος θα πρέπει να είναι ικανός να αντεπεξέλθει σε πιο μεγάλες και περίπλοκες βάσεις δεδομένων από αυτές που έχουμε διαθέσιμες σήμερα.

1.3 Παραγωγή Δεδομένων Μεγάλης Κλίμακας

Η παραγωγή δεδομένων είναι το πρώτο βήμα για τα δεδομένα μεγάλης κλίμακας. Συγκεκριμένα, ο μεγάλος όγκος, η διαφορετικότητα και η πολυπλοκότητα των δεδομένων παράγονται από πηγές που διαφέρουν σημαντικά η μια από την άλλη. Στις μέρες μας, οι κύριες πηγές παραγωγής δεδομένων είναι λειτουργικές και συναλλακτικές πληροφορίες εταιριών, υλικοτεχνικές και ανιχνευτικές πληροφορίες από το IoT (*Internet of Things*), πληροφορίες από την αλληλεπιδραση του ανθρώπου με το διαδίκτυο, δεδομένα από έρευνες κ.α.

1.3.1 Δεδομένα από Επιχειρήσεις

Το 2013, η IBM με άρθρο της (βλέπε [9]) υποστήριξε πως τα δεδομένα από επιχειρήσεις είναι η βασική πηγή δεδομένων μεγάλης κλίμακας.

Μέσα στο πέρασμα των τελευταίων δεκαετιών, η επιστήμη της πληροφορικής και της ανάλυσης δεδομένων έχουν αυξήσει τα κέρδη των επιχειρήσεων. Προβλέπεται πως τα δεδομένα που θα έχει η κάθε επιχείρηση στη διάθεσή της θα διπλασιάζονται κάθε 1,2 χρόνια. Έτσι λοιπόν, καθώς τα δεδομένα των επιχειρήσεων αυξάνονται, δημιουργείται η ανάγκη της άμεσης ανάλυσης των δεδομένων με σκοπό τη γρηγορότερη ανάκτηση γνώσης από αυτά. Για παράδειγμα, η *Walmart* δέχεται ένα εκατομμύριο συναλλαγές από πελάτες κάθε ώρα και τα δεδομένα αυτά τοποθετούνται σε μια βάση δεδομένων χωρητικότητας 2,5PB.

1.3.2 Δεδομένα από το IoT

Το IoT (*Internet of Things*) είναι μία έννοια που αφορά τα αντικείμενα της καθημερινότητας μας, από βιομηχανικές μηχανές μέχρι wearable συσκευές που χρησιμοποιούν ενσωματωμένους αισθητήρες για τη συλλογή δεδομένων και την ανάληψη κάποιας δράσης σε αυτά μέσα σε ένα δίκτυο. Κάπως έτσι λειτουργεί ένα κτίριο που χρησιμοποιεί αισθητήρες για την αυτόματη ρύθμιση της θέρμανσης ή του φωτισμού.

Αμέσως καταλαβαίνουμε πως το IoT είναι μια κύρια πηγή δεδομένων μεγάλης κλίμακας. Καθώς συλλέγονται δεδομένα μεγάλης κλίμακας από τη βιομηχανία, τη γεωργία, τα μέσα μεταφοράς, τους δημόσιους φορείς, τις κατοικίες κ.ο.κ.

Από τα δεδομένα τα οποία συλλέγονται από το IoT μπορούμε να ξεχωρίσουμε κάποια χαρακτηριστικά:

- Μεγάλος Όγκος: τα δεδομένα τα οποία θα έχουμε στην διάθεσή μας μπορεί να είναι απλά αριθμητικά δεδομένα, όπως μια τοποθεσία, ή να είναι κάτι αρκετά περίπλοκο, όπως ένα βίντεο. Για να μπορέσει να εφαρμοστεί μια μέθοδος ανάλυσης των δεδομένων αυτών, θα χρειαστούν και άλλα ιστορικά δεδομένα μέσα από ένα χρονικό πλαίσιο. Έτσι δημιουργείται και ο μεγάλος όγκος.

- Ανομοιογένεια: εξαιτίας τις πολυπλοκότητας των συσκευών από τα οποία γίνέται η συλλογή των δεδομένων.

- Ισχυρή σχέση Χρόνου και Χώρου: κάθε αντικείμενο το οποίο συλλέγει πληροφορίες βρίσκεται σε ένα συγκεκριμένο γεωγραφικό σημείο, μια συγκεκριμένη χρονική στιγμή.

- Μικρή Μερίδα των Δεδομένων Μεγάλης Κλίμακας έχουν Αξία: ένα φαινόμενο που είναι πάρα πολύ συνηθισμένο στο IoT. Μέσα στα δεδομένα μας θα συναντήσουμε αρκετά συχνά τιμές που δεν θα έχουν καμία αξία στην ανάλυσή μας. Για παράδειγμα, κατά την διάρκεια ενός βίντεο για την κυκλοφορία σε ένα δρόμο θα μας ενδιαφέρουν μόνο μερικά δευτερόλεπτα από ένα ατύχημα και όχι οι ατελείωτες ώρες κανονικής κυκλοφορίας.

1.3.3 Δεδομένα από το Διαδίκτυο

Τα δεδομένα από το διαδίκτυο, μεταξύ άλλων, αποτελούνται από καταχωρίσεις αναζήτησης, δημοσιεύσεις σε φόρουμ, συνομιλίες κ.α. Τα δεδομένα τέτοιου τύπου έχουν κάποια κοινά χαρακτηριστικά, όπως υψηλή αξία και χαμηλή πυκνότητα. Αυτή η κατηγορία δεδομένων δεν θα ήταν και τόσο σημαντική σε περίπτωση που αναλύαμε κάθε ποσότητα ξεχωριστά. Αλλά σε περίπτωση που τις συνδυάσουμε, μπορούμε να βρούμε σημαντικές πληροφορίες για συνήθειες και χόμπι χρηστών, καθώς και να προβλέψουμε την πιθανή τους συμπεριφορά και συναισθηματική κατάσταση.

1.3.4 Βιοϊατρικα Δεδομένα

Μετά από μια σειρά από υψηλής απόδοσης βιο-ιατρικές τεχνολογίες, που ξεκίνησαν και αναπτύχθηκαν στις αρχές του αιώνα, οι κλάδοι της βιολογίας και της ιατρικής έχουν ενταχθεί για τα καλά στην παραγωγή δεδομένων μεγάλης κλίμακας. Αυτό γίνεται περισσότερο αντιληπτό εάν αναλογιστούμε ότι το μέσο νοσοκομείο στις Η.Π.Α. έχει στην διάθεσή του όγκο δεδομένων περίπου ίσο με 600TB, σύμφωνα με εκτιμήσεις.

1.3.5 Άλλες Πηγές Δεδομένων Μεγάλης Κλίμακας

Καθώς οι επιστημονικές εφαρμογές αυξάνονται, ο όγκος των βάσεων δεδομένων συνεχώς μεγαλώνει. Αυτή η αύξηση, των επιστημονικών εφαρμογών, είναι άμεσα συνδεδεμένη με την ανάλυση από μεγάλης ποσότητας δεδομένα. Στη συνέχεια θα παρουσιάσουμε μερικά παραδείγματα πάνω σε αυτό το κομμάτι. Το πρώτο παράδειγμα έρχεται από την υπολογιστική βιολογία. Η GenBank είναι μια βάση δεδομένων για αλληλουχία νουκλεοτιδίων, που έχει στην κατοχή του το U.S. National Bio-Technology Innovation Center. Από το 2009 στη βάση αυτή υπάρχουν εγγραφές για πάνω από 150000 διαφορετικούς οργανισμούς. Το δεύτερο παράδειγμα σχετίζεται με την αστρονομία. Η SDSS (*Sloan Digital Sky Survey*) είναι η μεγαλύτερη έρευνα στην αστρονομία. Για την περίοδο 1998 - 2008 είχε στην κατοχή της 25TB δεδομένα. Από την αναβάθμιση που έκανε στο τηλεσκόπιο της, παρήγαγε δεδομένα που ξεπερνούσαν τα 20TB κάθε βράδυ. Το τελευταίο παράδειγμα είναι από τον τομέα της φυσικής υψηλών ενεργειών (*high-energy physics*). Από τις αρχές του 2008, το πείραμα ATLAS που πραγματοποιείται στο CERN παράγει 2PB ανά δευτερόλεπτο ακατέργαστα δεδομένα.

1.4 Απόκτηση Δεδομένων Μεγάλης Κλίμακας

Η απόκτηση των δεδομένων μεγάλης κλίμακας περιέχει τη συλλογή, τη μεταφορά και την προεπεξεργασία των δεδομένων. Κατά τη διάρκεια της απόκτησης των δεδομένων, όταν δηλαδή έχουν συλλεχθεί τα ακατέργαστα δεδομένα, ένας αποτελεσματικός μηχανισμός θα πρέπει να εφαρμόζεται έτσι ώστε τα δεδομένα αυτά να μπορούν να χρησιμοποιηθούν από διάφορες αναλυτικές εφαρμογές. Αρκετές φορές τα συλλεχθέντα δεδομένα μας περιέχουν και πληροφορία που μπορεί να μην μας φανεί χρήσιμη, με αποτέλεσμα την μείωση της αποθηκευτικής μας διαθεσιμότητας. Οι τεχνικές συμπίεσης δεδομένων μπορούν να εφαρμοστούν για να μειωθεί ο πλεονασμός. Συνεπώς, οι εργασίες προεπεξεργασίας των δεδομένων είναι απαραίτητες για την εξασφάλιση της αποθήκευσης και εκμετάλλευσης των δεδομένων.

1.4.1 Συλλογή Δεδομένων

Η συλλογή δεδομένων χρησιμοποιεί ειδικές τεχνικές για την απόκτηση ακατέργαστων δεδομένων από διάφορες πηγές δεδομένων. Στη συνέχεια, θα δούμε δύο από αυτές τις τεχνικές.

- Log Files: Είναι μια από τις πιο διαδεδομένες μεθόδους συλλογής δεδομένων. Τα Log Files είναι αρχεία καταγραφής που παράγονται αυτόματα από μια πηγή δεδομένων, έτσι ώστε να καταγράφονται οι διάφορες δραστηριότητες, σε ανάλογη μορφή, με στόχο τη μεταγενέστερη ανάλυση. Τα Log Files χρησιμοποιούνται

σχεδόν από όλες τις ηλεκτρονικές συσκευές. Για παράδειγμα, σε Log Files καταγράφεται ο αριθμός των «κλικ» ή των επισκεπτών σε μια ιστοσελίδα.

- Αισθητήρες: οι αισθητήρες μετατρέπουν μια φυσική ποσότητα σε ένα αναγνώσιμο σήμα έτσι ώστε να μπορεί να αποθηκευτεί και να επεξεργαστεί. Δεδομένα που προέρχονται από αισθητήρες μπορεί να είναι ήχος, εικόνα, πίεση του αέρα, θερμοκρασία κ.α. Οι πληροφορίες από τους αισθητήρες μεταφέρονται σε μια συλλογή δεδομένων με ενσύρματα ή ασύρματα δίκτυα. Οι εφαρμογές στις οποίες χρησιμοποιούνται οι αισθητήρες είναι εύκολα αντιληπτό πως διαφέρουν κατά πολύ μεταξύ τους. Για παράδειγμα, μπορούμε να έχουμε ένα απλό, ενσύρματο δίκτυο καμερών παρακολούθησης μέχρι ένα ασύρματο δίκτυο αισθητήρων που βοηθά την περιβαλλοντολογική έρευνα ή την ρύθμιση της ποιότητας υδάτων.

1.4.2 Μεταφορά Δεδομένων

Καθώς έχει τελειώσει η συλλογή των ακατέργαστων δεδομένων, τα δεδομένα μεταφέρονται σε ένα Κέντρο Δεδομένων (*Data Center*) για επεξεργασία και ανάλυση. Η διάταξη των δεδομένων θα πρέπει να διαμορφωθεί έτσι ώστε να βελτιστοποιείται η υπολογιστική αποτελεσματικότητα. Δηλαδή, θα πρέπει να γίνουν μεταδόσεις των δεδομένων (*data transmission*) εντός του Κέντρου Δεδομένων. Οι μεταδόσεις αυτές αποτελούνται από δύο φάσεις: τις μεταδόσεις Inter-DCN και τις μεταδόσεις Intra-DCN. (βλέπε [9])

Οι μεταδόσεις Inter-DCN γίνονται από την πηγή δεδομένων στο Κέντρο Δεδομένων και πραγματοποιούνται από μια υποδομή φυσικού δικτύου, όπως για παράδειγμα ένα σύστημα μετάδοσης οπτικών ινών. Οι μεταδόσεις Intra-DCN αφορούν τις επικοινωνιακές ροές δεδομένων μέσα στα Κέντρα Δεδομένων, όπως για παράδειγμα οι εσωτερικές μνήμες των servers.

1.4.3 Προεπεξεργασία Δεδομένων

Εξαιτίας της μεγάλης ποικιλίας των πηγών δεδομένων, οι βάσεις δεδομένων που αποκτώνται ποικίλουν ανάλογα με το θόρυβο, τον πλεονασμό, τη συνέπεια κ.α. και η αποθήκευση τέτοιων βάσεων δεδομένων θεωρείται αναμφισβήτητα σπατάλη. Επιπλέον, μερικές αναλυτικές μέθοδοι έχουν αυστηρές απαιτήσεις στην ποιότητα των δεδομένων. Επομένως, τα δεδομένα πρέπει να προεπεξεργάζονται με σκοπό να επιτύχουμε μια αποτελεσματική ανάλυση των δεδομένων μας. Η προεπεξεργασία των δεδομένων μειώνει το κόστος αποθήκευσης και βελτιώνει την ακρίβεια της ανάλυσης. Κάποιες τεχνικές που σχετίζονται με την προεπεξεργασία των δεδομένων θα αναλυθούν ακολούθως.

Ολοκλήρωση Δεδομένων

Η ολοκλήρωση δεδομένων αποτελεί τον ακρογωνιαίο λίθο της μοντέρνας πληροφορικής, η οποία περιλαμβάνει το συνδυασμό δεδομένων από διαφορετικές πηγές και παρέχει στους χρήστες τη δυνατότητα να βλέπουν τα δεδομένα ομοιόμορφα. Αυτό είναι ένα ώριμο ερευνητικό πεδίο για τις παραδοσιακές βάσεις δεδομένων. Ιστορικά, δύο μέθοδοι είναι ευρέως αποδεκτές: οι αποθήκες δεδομένων (*data warehouse*) και η εικονικοποίηση δεδομένων (*data federation* ή *data virtualisation*).

Οι αποθήκες δεδομένων περιέχουν τη διαδικασία που ονομάζεται ETL (*Extract, Transform, Load*). Η Εξαγωγή (*Extraction*) περιλαμβάνει τη σύνδεση πηγαίων συστημάτων, την επιλογή, τη συλλογή, την ανάλυση και την επεξεργασία των απαραίτητων δεδομένων. Ο Μετασχηματισμός (*Transfromation*) είναι η εκτέλεση μιας σειράς κανόνων για το μετασχηματισμό των δεδομένων που έχουν εξαχθεί σε συγκεκριμένες μορφές. Η Φόρτωση (*Loading*) αφορά την εισαγωγή των εξαχθέντων και μετασχηματισμένων δεδομένων στην υποδομή αποθήκευσης που επιθυμούμε.

Μια εικονική βάση δεδομένων δομείται με σκοπό να συγκεντρώσει δεδομένα από διαφορετικές πηγές, αλλά αυτή η βάση δεν περιέχει δεδομένα. Αντίθετα, περιέχει πληροφορίες ή μεταδεδομένα που σχετίζονται με τα πραγματικά δεδομένα.

Καθαρισμός Δεδομένων

Ο καθαρισμός δεδομένων είναι μια διαδικασία με σκοπό την αναγνώριση δεδομένων που θεωρούνται ανακριβή, ελλιπή ή μη ρεαλιστικά και τη μετέπειτα τροποποίηση ή διαγραφή αυτών για τη βελτίωση της ποιότητας των δεδομένων. Γενικά, ο καθαρισμός δεδομένων περιλαμβάνει πέντε συμπληρωματικές διεργασίες:

- τον καθορισμό και τον προσδιορισμό των σφαλμάτων,

- την αναζήτηση και την αναγνώριση των σφαλμάτων,
- τη διόρθωση των σφαλμάτων,
- την αναφορά παραδειγμάτων και τύπων των σφαλμάτων και
- την τροποποίηση των διαδικασιών εισαγωγής δεδομένων με σκοπό τη μείωση μελλοντικών σφαλμάτων.

Κατά τη διάρκεια του καθαρισμού, πρέπει να ελεγχθούν οι τύποι των δεδομένων, η πληρότητα, η ρεαλιστικότητα και οι περιορισμοί τους. Ο καθαρισμός είναι ζωτικής σημασίας για τη διατήρηση της συνέπειας, κάτι που είναι ευρέως εφαρμόσιμο σε πολλούς τομείς όπως η τραπεζική, η ασφάλιση, το εμπόριο, οι τηλεπικοινωνίες κ.α.

Εξάλειψη Πλεονασμού Δεδομένων

Ο πλεονασμός των δεδομένων αναφέρεται στην επανάληψη αυτών, κάτι που εμφανίζεται συχνά σε πολλές βάσεις δεδομένων. Ο πλεονασμός αυξάνει το κόστος της μετάδοσης δεδομένων που μπορεί να αποφευχθεί, κάτι που προκαλεί προβλήματα στα συστήματα αποθήκευσης. Επομένως, διάφορες μέθοδοι εξάλειψης πλεονασμού έχουν προταθεί, όπως η αναγνώριση πλεονασμού (*redundancy detection*), το φιλτράρισμα δεδομένων (*data filtering*), και η συμπίεση δεδομένων (*data compression*). Αυτές οι μέθοδοι μπορούν να εφαρμοστούν σε διάφορες βάσεις δεδομένων. Παρόλα αυτά, η αναγνώριση πλεονασμού μπορεί να προκαλέσει διάφορα αρνητικά αποτελέσματα. Για παράδειγμα, η συμπίεση και η αποσυμπίεση των δεδομένων μπορεί να εμφανίσουν επιπρόσθετο υπολογιστικό φορτίο. Επομένως, τα πλεονεκτήματα και τα μειονεκτήματα της εξάλειψης πλεονασμού πρέπει να εξισορροπηθούν προσεκτικά.

Κεφάλαιο 2

Ομαδοποίηση

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει μια εισαγωγή στην Ομαδοποίηση (*Clustering*). Αρχικά, θα παρουσιάσουμε τι είναι η Ομαδοποίηση, ή αλλιώς συσταδοποίηση, και θα δώσουμε παραδείγματα εφαρμογής της σε διάφορους τομείς. Στη συνέχεια θα αναφέρουμε μέτρα απόστασης και ομοιότητας με τα οποία διαχωρίζουμε κάποια αντικείμενα με βάση το πόσο απέχουν ή διαφέρουν. Ένα επίσης κύριο κομμάτι αυτού του κεφαλαίου είναι η παρουσίαση των διάφορων κατηγοριών, στις οποίες χωρίζονται οι μέθοδοι ομαδοποίησης. Στο τελευταίο μέρος του κεφαλαίου θα παρουσιάσουμε τις τεχνικές ομαδοποίησης δεδομένων μεγάλης κλίμακας.

2.2 Τι είναι Ομαδοποίηση

Η Ομαδοποίηση (*Clustering*), ή αλλιώς συσταδοποίηση ή ταξιδόμηση, είναι η μέθοδος της δημιουργίας ομάδων, ή αλλιώς συστάδων, από διάφορα αντικείμενα έτσι ώστε κάθε αντικείμενο μέσα σε κάθε ομάδα να είναι όμοιο με τα υπόλοιπα και τα αντικείμενα από διαφορετικές ομάδες να έχουν σημαντική διαφορά. Η έννοια της Ομαδοποίησης συχνά συγχέεται λανθασμένα με αυτή της Κατηγοριοποίησης (*Classification*), όπου τα αντικείμενα αντιστοιχίζονται σε ήδη προκαθορισμένες ομάδες. Για την ακρίβεια αποτελούν τις δύο βασικές κατηγορίες της Μηχανικής Μάθησης (*Machine Learning*) δηλαδή, την Εποπτευόμενη Μάθηση (Κατηγοριοποίηση) όπου το σύστημα καλείται να «μάθει» μια έννοια ή συνάρτηση από ένα σύνολο δεδομένων/αντικειμένων, η οποία αποτελεί περιγραφή ενός μοντέλου. Αντίθετα, την Μη Εποπτευόμενη Μάθηση (Ομαδοποίηση) το σύστημα πρέπει μόνο του να ανακαλύψει συσχετίσεις ή ομάδες σε ένα σύνολο δεδομένων, δημιουργώντας πρότυπα, χωρίς να είναι γνωστό αν υπάρχουν, πόσα και ποια είναι. (βλέπε [1])

Η έννοια της Ομαδοποίησης άρχισε να χρησιμοποιείται στην δεκαετία του 1960, εξαιτίας όμως της έλλειψης της τεχνολογίας των υπολογιστών της εποχής, η συσταδοποίηση δεν γνώρισε την άνθηση που γνωρίζει τις τελευταίες δεκαετίες. Βασισμένοι στην υπολογιστική ικανότητα που έχουν ακόμα και οι προσωπικοί υπολογιστές, πλέον μπορούμε να έχουμε πολύ καλά και αξιόπιστα αποτελέσματα. Ως άμεσο αποτέλεσμα αυτής της τεχνολογικής εξέλιξης, η Συσταδοποίηση βρίσκει εφαρμογή σχεδόν σε όλους τους επιστημονικούς κλάδους.

Για να καταλάβουμε όμως καλύτερα την έννοια της ομαδοποίησης μπορούμε να δούμε μερικούς από τους τομείς στους οποίους έχει εφαρμογή:

- Marketing: εύρεση ομάδων από πελάτες με παρόμοια συμπεριφορά, δοθέντος μιας βάσης δεδομένων από πελάτες που θα περιέχει παρελθοντικές προτιμήσεις σε αγορές.
- Ασφάλιση: ταυτοποίηση ομάδων από ασφαλισμένους με υψηλές απαιτήσεις για αναγνώριση πιθανών εξαπατιών.
- Πολεοδομία: εύρεση συστάδων από σπίτια βάσει της τιμής τους ή του τύπου τους.
- Βιολογία: συσταδοποίηση φυτών ή ζώων βάσει κάποιων χαρακτηριστικών τους.
- Αστρονομία: ταξιδόμηση αστεριών ή γαλαξιών όσον αφορά την απόστασή τους.

2.3 Μέτρα Απόστασης και Μέτρα Ομοιότητας

Για να μπορέσουμε όμως να δημιουργήσουμε διάφορες ομάδες αντικειμένων, βασισμένοι σε κάποια ομοιότητα τους, θα πρέπει να έχουμε κάποια μέτρα τα οποία θα μας υποδεικνύουν κατά πόσο κάθε αντικείμενο είναι κοντά ή μοιάζει με κάποιο άλλο. Έτσι, τα μέτρα απόστασης είναι οι θεμέλιοι λίθοι στην Κατηγοριοποίηση, στην Αναγνώριση Προτύπων και φυσικά στην Ομαδοποίηση. Για χιλιάδες χρόνια, διάφοροι πολιτισμοί, όριζαν ως συντομότερη απόσταση δύο αντικειμένων την Ευκλείδεια Απόσταση. Από τον προηγούμενο όμως αιώνα αυτή η αντίληψη άρχισε να αλλάζει. Παρατηρήθηκε πως για διάφορες εφαρμογές, θα έπρεπε να χρησιμοποιηθούν διαφορετικά μέτρα απόστασης.

Το κοινό χαρακτηριστικό των μέτρων απόστασης είναι πως παρατηρήσεις που μοιάζουν μεταξύ τους, θα πρέπει να δίνουν μικρή τιμή στο μέτρο της απόστασης.

Ένα εναλλακτικό μέτρο που χρησιμοποιείται για να δούμε κατά πόσο δύο αντικείμενα μοιάζουν, είναι τα μέτρα ομοιότητας. Η διαφορά τους με τα μέτρα απόστασης είναι πως οι παρατηρήσεις που μοιάζουν μεταξύ τους, θα πρέπει να δίνουν μεγάλη τιμή στο μέτρο της ομοιότητας.

Μια συνάρτηση d, θα θεωρείται μέτρο απόστασης όταν πληρεί τις παρακάτω υποθέσεις:

- **(D1)** $d(\mathbf{x}, \mathbf{y}) \geq 0$
- **(D2)** ανακλαστικότητα: $d(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}$
- **(D3)** συμμετρική ιδιότητα: $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$
- **(D4)** τριγωνική ανισότητα: $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{y}, \mathbf{z})$

και μια συνάρτηση s, θα θεωρείται μέτρο ομοιότητας όταν πληροί τις παρακάτω υποθέσεις:

- **(S1)** $0 \leq s(\mathbf{x}, \mathbf{y}) \leq 1$
- **(S2)** $s(\mathbf{x}, \mathbf{x}) = 1$
- **(S3)** συμμετρική ιδιότητα: $s(\mathbf{x}, \mathbf{y}) = s(\mathbf{y}, \mathbf{x})$

όπου τα $\mathbf{x}, \mathbf{y}$ και $\mathbf{z}$ τυχαία διανύσματα.

Έχοντας ορίσει μια απόσταση d μπορούμε να δημιουργήσουμε ένα αντίστοιχο μέτρο ομοιότητας s από τον παρακάτω τύπο:

$$s(\mathbf{x}, \mathbf{y}) = \frac{1}{1 + d(\mathbf{x}, \mathbf{y})}$$

Για την αντίθετη ενέργεια, δηλαδή έχοντας ορίσει ένα μέτρο ομοιότητας s μπορούμε να δημιουργήσουμε την αντίστοιχη απόσταση από τον παρακάτω τύπο:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{2(1 - s(\mathbf{x}, \mathbf{y}))}$$

Η τελευταία σχέση, δεν ικανοποιεί όμως την τριγωνική ιδιότητα **(D4)**. Ο Gower (βλέπε [6]) απέδειξε ότι, αν ο πίνακας $[s(\mathbf{x}, \mathbf{y})]_{nxn}$ με στοιχεία τις τιμές του μέτρου ομοιότητας για n αντικείμενα, είναι μη αρνητικά ορισμένος τότε η παραπάνω συνάρτηση d ικανοποιεί την τριγωνική ιδιότητα **(D4)**.

Για την επιλογή όμως του καταλληλότερου μέτρου απαιτείται εμπειρία, γνώση αλλά και τύχη. Στις παραγράφους που ακολουθούν θα παρουσιάσουμε τα βασικότερα και περισσότερο διαδεδομένα μέτρα απόστασης και μέτρα ομοιότητας, για διαφορετικούς τύπους δεδομένων.

2.3.1 Μέτρα για Συνεχή Δεδομένα

Τα μέτρα που θα αναφερθούν σε αυτή τη παράγραφο, χρησιμοποιούνται για συνεχή δεδομένα.

Ευκλείδεια Απόσταση

Η Ευκλείδεια απόσταση είναι πιθανότατα η πιο γνωστή και περισσότερο χρησιμοποιήσιμη απόσταση για συνεχή δεδομένα. Για δύο διανύσματα $\mathbf{x}$ και $\mathbf{y}$ στον d-διάστατο χώρο, η Ευκλείδεια απόσταση ορίζεται ως εξής:

$$d_{euc}(\mathbf{x}, \mathbf{y}) = \left(\sum_{j=1}^{d} (x_j - y_j)^2 \right)^{1/2} \tag{2.1}$$

όπου x_j και y_j οι τιμές των j-οστών στοιχείων από τα $\mathbf{x}$ και $\mathbf{y}$ αντίστοιχα.

Απόσταση Manhattan

Η απόσταση Manhattan ή απόσταση city block, ορίζεται ως το άθροισμα των απόλυτων αποκλίσεων, σε αντίθεση με την Ευκλείδεια που χρησιμοποιεί τις τετραγωνικές αποκλίσεις. Για δύο διανύσματα **x** και **y** στον d-διάστατο χώρο, η απόσταση Manhattan ορίζεται ως εξής:

$$d_{man}(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{d} |x_j - y_j| \tag{2.2}$$

όπου x_j και y_j οι τιμές των j-οστών στοιχείων από τα **x** και **y** αντίστοιχα.

Η απόσταση Manhattan δίνει παρόμοια αποτελέσματα με την Ευκλείδεια, δίνει όμως ανθεκτικότερα αποτελέσματα σε περίπτωση που έχουμε ακραίες τιμές, καθώς δεν τετραγωνίζει τις αποκλίσεις σαν την Ευκλείδεια απόσταση.

Απόσταση Maximum

Η απόσταση Maximum ή απόσταση Chebyshev, ορίζεται ως η μέγιστη απόκλιση των δύο σημείων. Για δύο διανύσματα **x** και **y** στον d-διάστατο χώρο, η απόσταση Maximum ορίζεται ως εξής:

$$d_{max}(\mathbf{x}, \mathbf{y}) = \max_{1 \leq j \leq d} |x_j - y_j| \tag{2.3}$$

όπου x_j και y_j οι τιμές των j-οστών στοιχείων από τα **x** και **y** αντίστοιχα.

Απόσταση Minkowski

Η Ευκλείδεια απόσταση, η απόσταση Manhattan και η απόσταση Maximum αποτελούν ειδικές περιπτώσεις της απόστασης Minkowski η οποία ορίζεται ως εξής:

$$d_{min}(\mathbf{x}, \mathbf{y}) = \left(\sum_{j=1}^{d} |x_j - y_j|^r \right)^{1/r} \tag{2.4}$$

όπου x_j και y_j οι τιμές των j-οστών στοιχείων από τα **x** και **y** αντίστοιχα, $r \geq 1$ δεδομένη παράμετρος, η οποία ονομάζεται τάξη της απόστασης Minkowski.

Μπορούμε εύκολα να παρατηρήσουμε πως για $r = 2, 1$ και ∞ η απόσταση Minkowski ισούται με την Ευκλείδεια, τη Manhattan και τη Maximum αντίστοιχα.

Απόσταση Mahalanobis

Όπως προαναφέραμε, οι παραπάνω αποστάσεις αποτελούν ειδικές περιπτώσεις της απόστασης Minkowski. Έτσι όλες αυτές δεν λαμβάνουν υπόψη τους τις συνδιακυμάνσεις ανάμεσα στις μεταβλητές, κάτι που κάνει η απόσταση Mahalanobis, η οποία ορίζεται ως εξής:

$$d_{mah}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})\Sigma^{-1}(\mathbf{x} - \mathbf{y})'} \tag{2.5}$$

όπου ο Σ είναι ο διανυσματικός πίνακας διακύμανσης - συνδιακύμανσης που αντιστοιχεί στα διανύσματα $\mathbf{x} = (x_1, x_2, ..., x_d)$ και $\mathbf{y} = (y_1, y_2, ..., y_d)$.

Άλλες Αποστάσεις

Στη διεθνή βιβλιογραφία (βλέπε [3], [4], [5]), μπορούμε να βρούμε και άλλες αποστάσεις για συνεχή δεδομένα όπως:

- Η Μέση Απόσταση:

$$d_{ave}(\mathbf{x}, \mathbf{y}) = \left(\frac{1}{d} \sum_{j=1}^{d} (x_j - y_j)^2 \right)^{1/2} \tag{2.6}$$

- Η Απόσταση Chord:

$$d_{chord}(\mathbf{x}, \mathbf{y}) = \left(2 - 2\frac{\sum_{d}^{k=1} x_j y_j}{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2}\right)^{\frac{1}{2}} \tag{2.7}$$

- Η Απόσταση Caberra.

$$d_{canb}(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{d} \frac{|x_j - y_j|}{(x_j + y_j)} \tag{2.8}$$

2.3.2 Μέτρα για Κατηγορικά Δεδομένα

Σε αυτή την παράγραφο θα αναφερθούμε στα μέτρα για Κατηγορικά δεδομένα. Κατηγορικά δεδομένα ονομάζονται τα δεδομένα που προέρχονται από μεταβλητές οι τιμές των οποίων εκφράζουν τάξεις ή κατηγορίες.

Απόσταση Simple Matching

Η απόσταση Simple Matching ίσως αποτελεί την πιο γνωστή και εύχρηστη απόσταση για κατηγορικά δεδομένα (βλέπε [6]).

Ας υποθέσουμε τα x και y ως δύο κατηγορικές τιμές. Τότε η απόσταση Simple Matching μεταξύ τους θα είναι:

$$\delta(x, y) = \begin{cases} 0 & , x = y \\ 1 & , x \neq y \end{cases} \tag{2.9}$$

Ας υποθέσουμε τα **x** και **y** ως δύο διανύσματα που περιγράφονται από d κατηγορικές ιδιότητες. Τότε η απόσταση Simple Matching θα είναι:

$$d_{sim}(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{d} \delta(x_j, y_j) \tag{2.10}$$

Άλλα Μέτρα για Κατηγορικά Δεδομένα

Εκτός από τη συνηθισμένη απόσταση Simple Matching, υπάρχουν και άλλα μέτρα που βρίσκουν εφαρμογή σε διάφορα ζητήματα. Πριν τα ορίσουμε όμως, θα υποθέσουμε τα **x** και **y** ως δύο διανύσματα που περιγράφονται από d κατηγορικές ιδιότητες, και:

- N_{a+d} θα είναι ο αριθμός των ιδιοτήτων των δύο διανυσμάτων στα οποία οι εγγραφές ταιριάζουν
- N_d θα είναι ο αριθμός των ιδιοτήτων των δύο διανυσμάτων στα οποία οι εγγραφές ταιριάζουν σε μια μη εφαρμόσιμη κατηγορία
- N_{b+c} θα είναι ο αριθμός των ιδιοτήτων των δύο διανυσμάτων στα οποία οι εγγραφές δεν ταιριάζουν

και ορίζονται ως εξής:

$$N_{a+d} = \sum_{j=1}^{d} [1 - \delta(x_j, y_j)] \tag{2.11}$$

$$N_d = \sum_{j=1}^{d} [\delta(x_j, ?) + \delta(?, y_j) - \delta(x_j, ?)\delta(?, y_j)] \tag{2.12}$$

$$N_{b+c} = \sum_{j=1}^{d} \delta(x_j, y_j) \tag{2.13}$$

όπου το «?» συμβολίζει τις ελλείπουσες παρατηρήσεις, για παράδειγμα όταν το $x_j = ?$, τότε το **x** διάνυσμα έχει ελλείπουσα τιμή στη j-οστή παρατήρηση. Έτσι μπορούμε να ορίσουμε μερικά από τα σημαντικότερα μέτρα για κατηγορικά δεδομένα:

Απόσταση	***d(i,j)***	**Περιγραφή**
Russel και Rao	$\frac{N_{a+d} - N_d}{N_{a+d} + N_{b+ac}}$	Ίσα βάρη.
Jaccard	$\frac{N_{a+d} - N_d}{N_{a+d} - N_d + N_{b+c}}$	Ίσα βάρη.
Dice	$\frac{2N_{a+d} - 2N_d}{2N_{a+d} - N_d + 2N_{b+c}}$	Διπλάσιο βάρος στις συμφωνίες.
Rogers και Tanimoto	$\frac{N_{a+d}}{N_{a+d} + 2N_{b+c}}$	Διπλάσιο βάρος στις ασυμφωνίες.

Πίνακας 2.1: Μέτρα για Κατηγορικά Δεδομένα

2.3.3 Μέτρα για Δίτιμες Μεταβλητές

Σε αυτή την παράγραφο θα αναφερθούμε στα μέτρα τα οποία χρησιμοποιούμε για δίτιμες μεταβλητές, μεταβλητές δηλαδή που μπορούν να πάρουν μόνο δύο τιμές, έστω 0 και 1. Συνήθως, όταν έχουμε δίτιμες μεταβλητές, ο αριθμός 0 υποδηλώνει την έλλειψη ή την απουσία κάποιου χαρακτηριστικού/ιδιότητας, ενώ ο αριθμός 1 υποδηλώνει την παρουσία κάποιου χαρακτηριστικού/ιδιότητας.

Για να ορίσουμε τα μέτρα απόστασης για τις δίτιμες μεταβλητές, θα πρέπει να κατασκευάσουμε πρώτα τον παρακάτω 2x2 πίνακα συνάφειας:

		αντικείμενο j		
		1	0	
αντικείμενο i	1	a	b	$a+b$
	0	c	d	$c+d$
		$a+c$	$b+d$	p

Στον παραπάνω πίνακα, ο αριθμός a αντιπροσωπεύει το πλήθος του συνδυασμού (1,1), δηλαδή την ύπαρξη κάποιου χαρακτηριστικού/ιδιότητας και στα δύο αντικείμενα. Αντίστοιχα, ο b αντιπροσωπεύει το πλήθος του συνδυασμού (1,0), δηλαδή την ύπαρξη κάποιου χαρακτηριστικού/ιδιότητας στο i αντικείμενο και την απουσία του στο j, κοκ. Είναι προφανές πως $a + b + c + d = p$. Στηριζόμενοι σε αυτούς τους συμβολισμούς, θα εκφράσουμε τις σημαντικότερες αποστάσεις για δίτιμες μεταβλητές. (βλέπε [6])

Απόσταση Simple Matching

Όπως και στα κατηγορικά δεδομένα, έτσι και στα δίτιμα, η γνωστότερη και περισσότερο χρησιμοποιήσιμη απόσταση είναι η Simple Matching,η οποία ορίζεται ως εξής:

$$d_{sim}(i,j) = \frac{b+c}{a+b+c+d} \tag{2.14}$$

Η απόσταση Simple Matching αποτελεί πρακτικά το ποσοστό των ασυμφωνιών. Συχνά η απόσταση Simple Matching αναφέρεται και σαν *M-coefficient* ή δείκτης εγγύτητας (*affinity index*).

Άλλες Αποστάσεις για Δίτιμα Δεδομένα

Πέρα από τη χρήση της απόστασης Simple Matching υπάρχουν και άλλες αποστάσεις που περιγράφονται στον παρακάτω πίνακα:

Απόσταση	***d(i,j)***	**Περιγραφή**
Rogers και Tanimoto	$\frac{a+d}{(a+d)+2(b+c)}$	Ποσοστό ασυμφωνιών δίνοντας διπλάσιο βάρος στις ασυμφωνίες.
Sokal και Sneath	$\frac{2(a+d)}{2(a+d)+(b+c)}$	Ποσοστό ασυμφωνιών δίνοντας διπλάσιο βάρος στις συμφωνίες.
Jaccard	$\frac{b+c}{a+b+c}$	Ποσοστό ασυμφωνιών αγνοώντας την απουσία και των δύο χαρακτηριστικών.
Dice και Sorensen	$\frac{b+c}{2a+b+c}$	Ποσοστό ασυμφωνιών αγνοώντας την απουσία και των δύο χαρακτηριστικών.

Πίνακας 2.2: Αποστάσεις για Δίτιμα Δεδομένα

2.3.4 Μέτρα για Μεικτού τύπου Δεδομένα

Σε πολλές εφαρμογές, τα δεδομένα τα οποία καλούμαστε να διαχειριστούμε δεν έχουν μόνο ένα τύπο δεδομένων. Ακριβώς για αυτό το λόγο σε αυτή την παράγραφο θα αναφερθούμε σε μέτρα τα οποία βρίσκουν εφαρμογή σε μεικτού τύπου δεδομένα.

Γενικός Συντελεστής Ομοιότητας

Ο Γενικός Συντελεστής Ομοιότητας (*General Similarity Coefficient*) , που εισήχθηκε από τον Gower το 1971 (βλέπε [3]), χρησιμοποιείται για να μετρήσει την ομοιότητα μεταξύ δύο μεταβλητών με μεικτού τύπου τιμές. Ο Γενικός Συντελεστής Ομοιότητας, βρίσκει εφαρμογή και για δεδομένα με ελλείπουσες τιμές.

Για δύο διανύσματα **x** και **y** στον d-διάστατο χώρο, ο Γενικός Συντελεστής Ομοιότητας ορίζεται ως εξής:

$$s_{gower}(\mathbf{x},\mathbf{y}) = \frac{1}{\sum_{k=1}^{d} w(x_k,y_k)} \sum_{k=1}^{d} w(x_k,y_k)s(x_k,y_k) \tag{2.15}$$

όπου, x_k και y_k οι τιμές των k-οστών στοιχείων από τα **x** και **y** αντίστοιχα. Τα $w(x_k,y_k)$ και $s(x_k,y_k)$ ορίζονται ως εξής:

- Για *συνεχείς* τιμές των x_k και y_k:

  $$s(x_k,y_k) = 1 - \frac{|x_k - y_k|}{R_k}$$

  όπου R_k είναι το εύρος των k-οστών στοιχείων, και $w(x_k,y_k) = 0$ όταν τα **x** ή **y** έχουν ελλείπουσες τιμές στην k-οστή παρατήρηση και $w(x_k,y_k) = 1$ αντίθετα.

- Για *κατηγορικές* τιμές των x_k και y_k,$s(x_k,y_k) = 1$ όταν $x_k = y_k$, αλλιώς $s(x_k,y_k) = 0$ και $w(x_k,y_k) = 0$ όταν τα **x** ή **y** έχουν ελλείπουσες τιμές στην k-οστή παρατήρηση και $w(x_k,y_k) = 1$ αντίθετα.

- Για *δίτιμες* τιμές των x_k και y_k, $s(x_k,y_k) = 1$ όταν και τα δύο **x** , **y** στην k-οστή παρατήρηση έχουν την ιδιότητα "παρών", αλλιώς $s(x_k,y_k) = 0$ και $w(x_k,y_k) = 0$ όταν και τα δύο **x** , **y** στην k-οστή παρατήρηση έχουν την ιδιότητα "απών", αλλιώς $s(x_k,y_k) = 1$

Γενικός Συντελεστής Απόστασης

Ο Γενικός Συντελεστής Απόστασης (*General Distance Coefficient*) , που εισήχθηκε από τον Gower το 1971 (βλέπε [3]), χρησιμοποιείται για να μετρήσει την απόσταση μεταξύ δύο μεταβλητών με μεικτού τύπου τιμές.

Για δύο διανύσματα **x** και **y** στον d-διάστατο χώρο, ο Γενικός Συντελεστής Απόστασης ορίζεται ως εξής:

$$d_{gower}(\mathbf{x},\mathbf{y}) = \left(\frac{1}{\sum_{k=1}^{d} w(x_k, y_k)} \sum_{k=1}^{d} w(x_k, y_k) d^2(x_k, y_k) \right)^{\frac{1}{2}} \tag{2.16}$$

όπου, x_k και y_k οι τιμές των k-οστών στοιχείων από τα **x** και **y** αντίστοιχα. Το $w(x_k, y_k)$ ορίζεται ακριβώς όπως και στην εξίσωση (2.15). Το $d^2(x_k, y_k)$ ορίζεται ως εξής:

- Για *συνεχείς* τιμές των x_k και y_k:

  $$d(x_k, y_k) = \frac{|x_k - y_k|}{R_k}$$

  όπου R_k είναι το εύρος των k-οστών στοιχείων.

- Για *κατηγορικές* τιμές των x_k και y_k, $d(x_k, y_k) = 0$ όταν $x_k = y_k$, αλλιώς $d(x_k, y_k) = 1$

- Για *δίτιμες* τιμές των x_k και y_k, $d(x_k, y_k) = 0$ όταν και τα δύο **x** , **y** στην k-οστή παρατήρηση έχουν την ιδιότητα "παρόν" ή "απών", αλλιώς $d(x_k, y_k) = 1$

2.4 Μέτρα Απόστασης και Ομοιότητας Μεταξύ των Συστάδων

Στην επόμενη παράγραφο 2.5, θα εξηγήσουμε πως κάθε διαφορετικός αλγόριθμος ομαδοποίησης δημιουργεί ομάδες/συστάδες. Πρώτα όμως, θα πρέπει να υπολογίσουμε την απόσταση μεταξύ ενός σημείου και μιας συστάδας και την απόσταση μεταξύ δύο συστάδων.

Στις παραγράφους που ακολουθούν θα χρησιμοποιήσουμε τους ακόλουθους συμβολισμούς: $C_1 = \{x_1, x_2, ..., x_r\}$ και $C_2 = \{y_1, y_2, ..., y_s\}$, όπου τα C_1 και C_2 υποδηλώνουν δύο συστάδες r και s μεγέθους αντίστοιχα.

2.4.1 Απόσταση Βασισμένη στο Μέσο

Η απόσταση βασισμένη στο Μέσο (*Mean-based Distance*) είναι ένα συνηθισμένο μέτρο για να υπολογίζουμε την απόσταση μεταξύ δύο συστάδων με αριθμητικά δεδομένα. Υποθέτουμε δύο συστάδες C_1 και C_2 με αριθμητικά δεδομένα, η απόσταση βασισμένη στο Μέσο ορίζεται ως:

$$D_{mean}(C_1, C_2) = d(\mu(C_1), \mu(C_2)) \tag{2.17}$$

όπου τα $\mu(C_1)$ και $\mu(C_2)$ είναι οι μέσοι των συστάδων C_1 και C_2 αντίστοιχα.

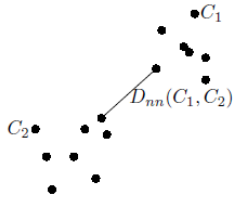
$$\mu(C_j) = \frac{1}{|C_j|} \sum_{x \epsilon C_j} \mathbf{x}, \quad j = 1, 2.$$

2.4.2 Απόσταση Πλησιέστερου Γείτονα

Δοθείσας μιας συνάρτησης απόστασης d, η απόσταση πλησιέστερου γείτονα (*Nearest Neighbor Distance*) μεταξύ δύο συστάδων C_1 και C_2 ορίζεται ως:

$$D_{nn}(C_1, C_2) = \min_{1 \le i \le r, 1 \le j \le s} d(x_i, y_j) \tag{2.18}$$

Στο σχήμα 2.1 που ακολουθεί, μπορούμε να δούμε ένα παράδειγμα της απόστασης πλησιέστερου γείτονα στον δισδιάστατο χώρο.



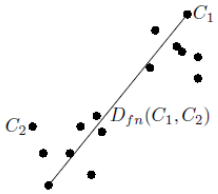
Σχήμα 2.1: Η απόσταση πλησιέστερου γείτονα μεταξύ δύο συστάδων

2.4.3 Απόσταση Μακρινότερου Γείτονα

Δοθείσας μιας συνάρτησης απόστασης d, η απόσταση μακρινότερου γείτονα (*Farthest Neighbor Distance*) μεταξύ δύο συστάδων C_1 και C_2 ορίζεται ως:

$$D_{fn}(C_1, C_2) = \max_{1 \leq i \leq r, 1 \leq j \leq s} d(x_i, y_j) \tag{2.19}$$

Στο σχήμα 2.2 που ακολουθεί, μπορούμε να δούμε ένα παράδειγμα της απόστασης μακρινότερου γείτονα στον δισδιάστατο χώρο.



Σχήμα 2.2: Η απόσταση μακρινότερου γείτονα μεταξύ δύο συστάδων

2.5 Κατηγορίες Αλγορίθμων Ομαδοποίησης

Η δημιουργία ομάδων από ένα σύνολο αντικειμένων μπορεί να αποτελέσει από μια πολύ απλή διαδικασία έως κάτι που θα απαιτεί γνώση, χρόνο και εμπειρία. Έτσι λοιπόν η προσέγγιση την οποία θα ακολουθήσουμε, για να πραγματοποιήσουμε μια ομαδοποίηση, διαδραματίζει το σημαντικότερο ρόλο στην ανάλυση μας.

Συνεχώς αναπτύσσονται νέες μέθοδοι ομαδοποίησης, κάθε μία όμως έχει τη δική της προσέγγιση, καθώς ο στόχος και τα δεδομένα διαφέρουν στις περισσότερες των περιπτώσεων. Αρχικά, οι δύο βασικότερες κατηγορίες αλγορίθμων Ομαδοποίησης, που προτάθηκαν, είναι οι Ιεραρχικοί Αλγόριθμοι (*Hierarchical Algorithms*) και οι Διαμεριστικοί Αλγόριθμοι (*Partitioning Algorithms*). Στη συνέχεια όμως προστέθηκαν άλλες τρεις κατηγορίες, οι Μέθοδοι βασισμένοι στη Πυκνότητα (*density-based methods*), οι Μέθοδοι βασισμένοι στο Μοντέλο (*model-based methods*) και οι Μέθοδοι βασισμένοι στο Πλέγμα (*grid-based methods*).

Σχήμα 2.3: Κατηγορίες Αλγορίθμων Ομαδοποίησης

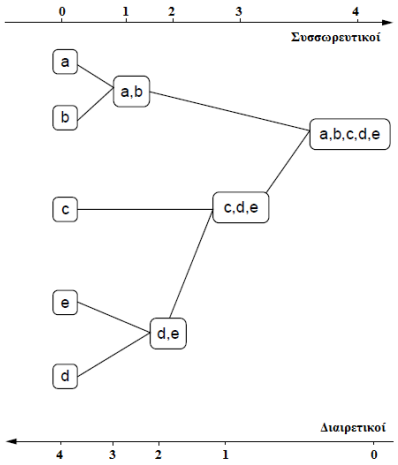
Οι Ιεραρχικοί Αλγόριθμοι χωρίζουν τα δεδομένα με δύο διαφορετικούς τρόπους, είτε ξεκινώντας αντιστοιχίζοντας κάθε αντικείμενο σε μία ξεχωριστή ομάδα και σε κάθε βήμα να ενώνουν τις ομάδες αυτές, είτε αρχίζουν θεωρώντας όλα τα δεδομένα ως μια μεγάλη ομάδα και σε κάθε βήμα να διαχωρίζουν αυτή την

ομάδα. Οι πρώτες μέθοδοι ονομάζονται συσσωρευτικές (*agglomerative*) ενώ οι δεύτερες διαιρετικές (*divisive*). Από την άλλη πλευρά στους Διαμεριστικούς Αλγορίθμους ο αριθμός των ομάδων είναι γνωστός από πριν. Στην περίπτωση αυτή, η εκάστοτε παρατήρηση αντιστοιχίζεται στην ομάδα που είναι πιο κοντά σε αυτή. Οι μέθοδοι βασισμένοι στην πυκνότητα, υποθέτουν πως κάθε αντικείμενο μέσα στην εκάστοτε συστάδα προέρχεται από μια κατανομή πιθανότητας. Οι μέθοδοι βασισμένοι στο μοντέλο, προσπαθούν να βελτιστοποιήσουν την εφαρμογή των δεδομένων μας σε κάποια μαθηματικά μοντέλα.Τέλος στις μεθόδους βασισμένες στο πλέγμα ο χώρος χωρίζεται σε ένα πεπερασμένο αριθμό κελιών, σε μορφή πλέγματος, όπου στο κάθε ένα εφαρμόζεται η ομαδοποίηση.

Κάθε μέθοδος Ομαδοποίησης έχει τα θετικά αλλά και τα αρνητικά της. Έτσι λοιπόν δεν είναι πάντα ξεκάθαρο πότε μπορούμε να χρησιμοποιήσουμε την εκάστοτε μέθοδο.

2.5.1 Ιεραρχικοί Αλγόριθμοι

Βασικό χαρακτηριστικό των ιεραρχικών αλγορίθμων (*Hierarchical Algorithms*) ομαδοποίησης είναι πως παράγουν μια ιεραρχία στη μορφή δενδροδιαγράμματος, όπου στα στάδια, το πλήθος k των ομάδων, στις οποίες θέλουμε να χωρίσουμε τα n αντικείμενα, παίρνει όλες τις δυνατές τιμές από το 1 έως το n. Στο ένα άκρο αυτής της ιεραρχίας υπάρχει μόνο μια ομάδα που παίρνει n αντικείμενα και στο άλλο υπάρχουν n ομάδες όπου η κάθε μια περιέχει μόνο ένα αντικείμενο.



Σχήμα 2.4: Παράδειγμα για την μορφή δενδροδιαγράμματος, που έχουν οι ιεραρχικοί αλγόριθμοι (βλέπε [6]).

Η παραπάνω περιγραφή, πρακτικά αναφέρεται στις δύο βασικές κατηγορίες ιεραρχικών αλγορίθμων, τους συσσωρευτικούς (*agglomerative*) και τους διαιρετικούς (*divisive*) αλγορίθμους.

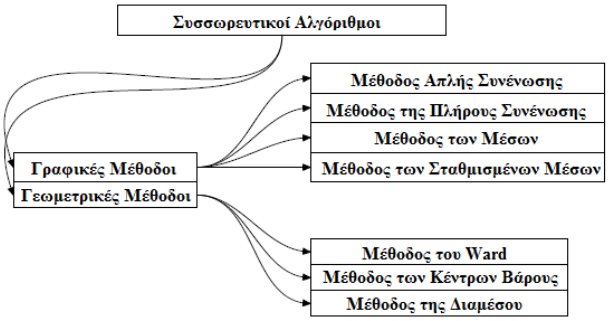
Στις παραγράφους που ακολουθούν θα θεωρήσουμε πως διαθέτουμε παρατηρήσεις για n αντικείμενα $x_1, x_2, ..., x_n$ και για τα οποία έχουμε υπολογίσει τις αποστάσεις μεταξύ τους $d_{ij} = d(x_i, x_j), i, j = 1, 2, ..., n$ και τις αποστάσεις αυτές τις έχουμε τοποθετήσει σε έναν πίνακα D με n γραμμές και n στήλες.

Συσσωρευτικοί Αλγόριθμοι

Οι συσσωρευτικοί (*agglomerative*) αλγόριθμοι αποτελούν κατηγορία των ιεραρχικών μεθόδων ομαδοποίησης. Οι αλγόριθμοι αυτοί ξεκινούν δημιουργώντας n το πλήθος ομάδες, όπου στην κάθε ομάδα τοποθετούν ένα αντικείμενο από τα δεδομένα μας. Στη συνέχεια, μέσω του πίνακα D, εντοπίζουν τα αντικείμενα με την μικρότερη απόσταση και τα τοποθετούν σε μια ομάδα. Έτσι λοιπόν, έχουμε πλέον $n - 1$ ομάδες, όπου μια ομάδα έχει 2 αντικείμενα και οι υπόλοιπες $n - 2$ ομάδες έχουν από ένα αντικείμενο. Η διαδικασία αυτή συνεχίζεται κατά αυτόν τον τρόπο όπου μετά από k βήματα, έχει δημιουργηθεί μία και μόνο ομάδα, η οποία περιέχει όλα τα αντικείμενα από τα δεδομένα μας.

Ανάλογα με τα διάφορα μέτρα απόστασης μεταξύ των ομάδων, οι συσσωρευτικοί αλγόριθμοι μπορούν να χωριστούν σε υποκατηγορίες, όπως η μέθοδος απλής συνένωσης, η μέθοδος της πλήρους συνένωσης κ.α.

Όπως βλέπουμε και στο παρακάτω σχήμα 2.5, οι συσσωρευτικοί αλγόριθμοι χωρίζονται σε δύο υποκατηγορίες, τις γραφικές μεθόδους και τις γεωμετρικές μεθόδους.



Σχήμα 2.5: Οι συνηθέστεροι συσσωρευτικοί αλγόριθμοι.

Στις γραφικές μεθόδους ανήκουν:

1. **Μέθοδος της απλής συνένωσης** (*Single Linkage Method*)

   Η μέθοδος της απλής συνένωσης είναι μια από τις απλούστερες μεθόδους ιεραρχικής ομαδοποίησης. Επίσης, είναι γνωστή και ως μέθοδος του πλησιέστερου γείτονα. Αυτή η ονομασία είναι απόλυτα λογική καθώς η συγκεκριμένη μέθοδος χρησιμοποιεί την απόσταση του πλησιέστερου γείτονα για την απόσταση μεταξύ δύο συστάδων.

2. **Μέθοδος της πλήρους συνένωσης** (*Complete Linkage Method*)

   Σε αντίθεση με την μέθοδο της απλής συνένωσης, η μέθοδος της πλήρους συνένωσης χρησιμοποιεί την απόσταση του μακρινότερου γείτονα για την απόσταση μεταξύ δύο συστάδων. Ακριβώς για αυτό το λόγο ονομάζεται και μέθοδος του μακρινότερου γείτονα.

3. **Μέθοδος των μέσων** (*Group Average Method*)

   Η μέθοδος των μέσων, ή αλλιώς UPGMA (*unweighted pair group method using arithmetic averages*), για να ορίσει την απόσταση μεταξύ δύο συστάδων χρησιμοποιεί τη μέση απόσταση μεταξύ όλων των δυνατών αντικειμένων εάν ενώσουμε τις δύο συστάδες.

4. **Μέθοδος των σταθμισμένων μέσων** (*Weighted Group Average Method*)

   Σε αυτή τη μέθοδο, η απόσταση μεταξύ δύο συστάδων είναι ο μέσος των αποστάσεων όλων των αντικειμένων της μίας συστάδας με τα αντικείμενα της άλλης.

Στις γεωμετρικές μεθόδους ανήκουν:

1. **Μέθοδος του Ward** (*Ward's Method*)

   Η συγκεκριμένη μέθοδος είναι σχεδιασμένη να ελαχιστοποιεί τη διακύμανση μέσα στην κάθε συστάδα. Ακόμη, υπολογίζει την απόσταση μεταξύ κάθε αντικειμένου και του κέντρου της συστάδας. Η απόσταση η οποία χρησιμοποιείται είναι η ευκλείδεια, για αυτό τον λόγο η μέθοδος του Ward εφαρμόζεται μόνο σε ποσοτικά δεδομένα.

2. **Μέθοδος των κέντρων βάρους** (*Centroid Method*)

   Η απόσταση που εφαρμόζει η μέθοδος των κέντρων βάρους είναι η απόσταση των κέντρων των συστάδων. Έτσι η εναλλακτική της ονομασία είναι «Μέθοδος μη-σταθμισμένων ζευγαριών χρησιμοποιώντας τα κέντρα βάρους».

3. **Μέθοδος της διαμέσου** (*Median Method*)

   Η μέθοδος της διαμέσου ή μέθοδος του Gower, μπορεί να εφαρμοστεί μόνο σε ποσοτικά δεδομένα καθώς χρησιμοποιεί την ευκλείδεια απόσταση. Η μέθοδος αυτή δεν είναι ιδιαίτερα διαδεδομένη και για περισσότερες λεπτομέρειες παραπέμπουμε στο [3].

Διαιρετικοί Αλγόριθμοι

Οι διαιρετικοί ιεραρχικοί αλγόριθμοι ακολουθούν την ακριβώς αντίθετη διαδικασία από αυτή που περιγράψαμε στους συσσωρευτικούς αλγόριθμους. Δηλαδή, οι διαιρετικοί αλγόριθμοι ξεκινούν θεωρώντας όλα τα n αντικείμενα που έχουμε στην διάθεση μας ως μια συστάδα. Η διαδικασία συνεχίζεται δημιουργώντας συστάδες με λιγότερα αντικείμενα στην κάθε μια. Η διαδικασία φτάνει στο τέλος της όταν σε κάθε συστάδα έχει μείνει μόνο ένα αντικείμενο.

Πρακτικά, οι διαιρετικοί αλγόριθμοι δεν είναι διαδεδομένοι. Ο κύριος λόγος που συμβαίνει αυτό είναι οι πολλές υπολογιστικές απαιτήσεις που χρειάζονται σε σχέση με τους συσσωρευτικούς αλγορίθμους. Αυτό γίνεται εύκολα αντιληπτό, καθώς στο πρώτο βήμα του αλγορίθμου οι πιθανοί διαμερισμοί των n αντικειμένων από μια συστάδα σε δύο είναι $2^{n-1} - 1$.

Ωστόσο, ο συνηθέστερος διαιρετικός αλγόριθμος είναι ο DIANA (DIvisive ANAlysis) (βλέπε [6]), ο οποίος έχει προταθεί από τους Kaufman & Rousseeuw (1990). Ένας άλλος δημοφιλής διαιρετικός αλγόριθμός είναι ο αλγόριθμος των Edwards & Cavalli-Sforza (1965), όπου η λογική του αλγορίθμου είναι κοινή με αυτή της μεθόδου του Ward στους συσσωρευτικούς αλγορίθμους.

2.5.2 Διαμεριστικοί Αλγόριθμοι

Ο στόχος των διαμεριστικών αλγορίθμων (*Partitioning Algorithms*) είναι να ομαδοποιήσουν τα n αντικείμενα που έχουμε στην διάθεση μας σε k συστάδες. Η διαφορά τους με τους ιεραρχικούς αλγορίθμους είναι πως στους διαμεριστικούς ο αριθμός k των συστάδων θα πρέπει να είναι καθορισμένος από πριν. Αυτόματα δημιουργείται ένας περιορισμός, διότι είτε με κάποιο τρόπο πρέπει να γνωρίζουμε τον βέλτιστο αριθμό για το k, είτε να επαναλάβουμε αρκετές φορές τον αλγόριθμο με διάφορες τιμές για το k. Δύο βασικές υποθέσεις που πρέπει να πληρούνται στους διαμεριστικούς αλγορίθμους είναι:

1. Κάθε συστάδα πρέπει να περιέχει τουλάχιστον ένα αντικείμενο.

2. Κάθε αντικείμενο πρέπει να ανήκει σε ακριβώς μια συστάδα.

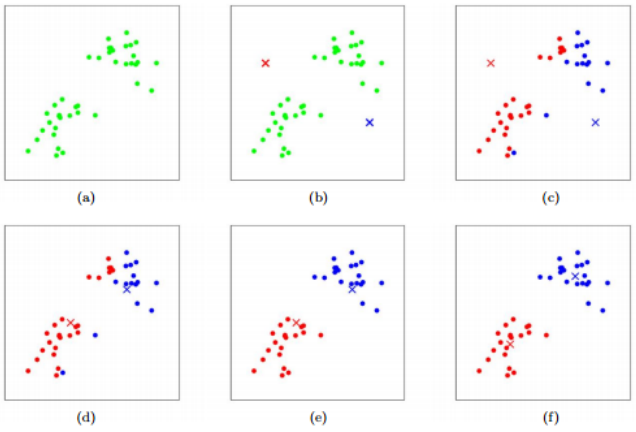
Οι διαμεριστικοί αλγόριθμοι χωρίζονται σε δύο κατηγορίες, στους αλγορίθμους ελαχιστοποίησης σφάλματος (*Error Minimization Algorithms*) και στις μεθόδους Graph-Theoretic.

Αλγόριθμοι Ελαχιστοποίησης Σφάλματος

Οι συγκεκριμένοι αλγόριθμοι, οι οποίοι τείνουν να λειτουργούν καλύτερα με απομονωμένες και συμπαγείς συστάδες, είναι οι πιο ευκολονόητοι και συνηθισμένοι αλγόριθμοι. Η βασική ιδέα κάτω από την οποία λειτουργούν είναι να κατασκευάζουν ομάδες στις οποίες κάποιο κριτήριο για το σφάλμα, το οποίο μετρά την απόσταση δύο αντικειμένων, θα ελαχιστοποιείται. Το γνωστότερο κριτήριο είναι το άθροισμα των τετραγώνων των σφαλμάτων (*Sum of Squared Error, SSE*).

Ο συνηθέστερος και πιο γνωστός αλγόριθμος που χρησιμοποιεί το κριτήριο για το άθροισμα των τετραγώνων των σφαλμάτων είναι αλγόριθμος *K-means*. Ο αλγόριθμος χωρίζει τα δεδομένα μας σε K συστάδες $(C_1, C_2, ..., C_K)$, των οποίων τα κέντρα υπολογίζονται ως οι μέσοι από τα αντικείμενα που συμπεριλαμβάνονται σε κάθε συστάδα.

Ο αλγόριθμος ξεκινά έχοντας K κέντρα συστάδων, επιλεγμένα τυχαία ή μέσω κάποιας ευρετικής διαδικασίας. Σε κάθε επανάληψη, κάθε αντικείμενο τοποθετείται στη συστάδα με το κοντινότερο κέντρο, βάση της Ευκλείδειας απόστασης και το κέντρο ξανά υπολογίζεται. Το κέντρο κάθε συστάδας ορίζεται ως ο μέσος όλων των αντικειμένων που βρίσκονται στην συστάδα αυτή. Η διαδικασία ολοκληρώνεται μόλις τα κέντρα των ομάδων παραμείνουν τα ίδια. Ένα μειονέκτημα του *K-means* είναι πως παρουσιάζει αδυναμία στη συσταδοποίηση δεδομένων με outliers. Η διαδικασία φαίνεται και στο σχήμα 2.6.



Σχήμα 2.6: Ο Αλγόριθμος K-means.

Από το παραπάνω σχήμα, τα αντικείμενα που έχουμε στη διάθεση μας αντιπροσωπεύονται από τις τελείες και τα κέντρα των συστάδων από τους σταυρούς. Στο στάδιο (a), έχουμε τα αρχικά μας δεδομένα. Στο στάδιο (b), διαλέγουμε δύο τυχαία κέντρα συστάδων. Από το στάδιο (c) έως το στάδιο (f) βλέπουμε την διαδικασία δημιουργίας των δύο ομάδων.

Ένας άλλος διαμεριστικός αλγόριθμος, που προσπαθεί να ελαχιστοποιήσει το SSE είναι ο *K-medoids* ή PAM (*partition around medoids*). Αυτός ο αλγόριθμος είναι αρκετά παραπλήσιος με τον *K-means*. Η διαφορά τους είναι πως ο *K-medoids* ως κέντρα των συστάδων χρησιμοποιεί πιο κεντραρισμένα σημεία, τα οποία αποτελούν και στοιχεία που βρίσκονται εντός της κάθε συστάδας. Σε σύγκριση με τον *K-means* ο *K-medoids* ανταποκρίνεται καλύτερα σε δεδομένα με θόρυβο ή ακραίες τιμές, καθώς αυτά επηρεάζουν λιγότερο τα medoids από ότι τον μέσο.

Μέθοδοι Graph-Theoretic

Οι μέθοδοι Graph-Theoretic είναι μέθοδοι που δημιουργούν ομάδες βάσει γραφημάτων. Οι ακμές των γραφημάτων συνδέουν τα αντικείμενα που έχουμε στη διάθεση μας ως κόμβους. Η γνωστότερη Graph-Theoretic μέθοδος βασίζεται στο Minimum Spanning Tree - MST. Σκοπός της μεθόδου είναι να ενώσει όλα τα αντικείμενα έτσι ώστε να έχουν την ελάχιστη δυνατή απόσταση. Έτσι τα αντικείμενα που απέχουν περισσότερο από άλλα αποτελούν ξεχωριστές ομάδες.

2.5.3 Μέθοδοι βασισμένοι στην Πυκνότητα

Οι μέθοδοι βασισμένοι στην πυκνότητα, υποθέτουν πως κάθε αντικείμενο μέσα στην εκάστοτε συστάδα προέρχεται από μια κατανομή πιθανότητας. Επίσης, υποθέτουν πως η κατανομή όλων των δεδομένων είναι μια μίξη των εκάστοτε κατανομών. Στόχος των μεθόδων αυτών είναι να αναγνωρίσουν τις συστάδες, καθώς και τις παραμέτρους των κατανομών τους. Οι μέθοδοι αυτοί είναι σχεδιασμένοι να αναγνωρίζουν συστάδες τυχαίου σχήματος, χωρίς απαραίτητα να είναι κυρτές.

Στο πεδίο αυτό, ο μεγαλύτερος όγκος της έρευνας έχει βασιστεί στην υπόθεση πως οι κατανομές των συστάδων είναι πολυμεταβλητές Κανονικές (στην περίπτωση αριθμητικών δεδομένων) ή Πολυωνυμικές (στην περίπτωση των κατηγορικών δεδομένων)

Στις μεθόδους βασισμένες στην πυκνότητα ανήκουν και οι μέθοδοι mode-seeking. Η βασική ιδέα στους αλγόριθμους αυτούς είναι να μεγαλώνουν τον όγκο των συστάδων μέχρι ενός δεδομένου κατωφλιού πυκνότητας. Δηλαδή, εντός μιας δοσμένης ακτίνας να υπάρχει τουλάχιστον ένας αριθμός αντικειμένων.

Άλλοι αλγόριθμοι βασισμένοι στην πυκνότητα είναι:

- Ο DBSCAN (*density-dased sparital clustering of applications with noise*), ο οποίος ανακαλύπτει ομάδες τυχαίου σχήματος και είναι αρκετά αποτελεσματικός σε δεδομένα με θόρυβο και απομακρυσμένες παρατηρήσεις.

- Ο AUTOCLASS, ένας αρκετά χρησιμοποιήσιμος αλγόριθμος που χρησιμοποιεί διάφορες κατανομές, όπως την Κανονική, τη Bernoulli, την Poisson και τη Λογαριθμοκανονική.

2.5.4 Μέθοδοι βασισμένοι στο Μοντέλο

Οι μέθοδοι βασισμένοι στο μοντέλο προσπαθούν να βελτιστοποιήσουν την εφαρμογή των δεδομένων μας σε κάποια μαθηματικά μοντέλα. Αντίθετα από τις υπόλοιπες μεθόδους ομαδοποίησης, που εντοπίζουν ομάδες αντικειμένων, οι μέθοδοι βασισμένοι στο μοντέλο βρίσκουν μια ιδιότητα μέσα στην ομάδα, η οποία αντιπροσωπεύεται από μια κλάση. Οι συνηθέστερες κατηγορίες μεθόδων βασισμένες στο μοντέλο είναι τα δέντρα απόφασης και τα νευρωνικά δίκτυα.

Δέντρα Απόφασης

Στα δέντρα απόφασης (*decision trees*), τα δεδομένα παρουσιάζονται από ένα ιεραρχικό δέντρο, όπου κάθε φύλλο του αναφέρεται σε μια συνθήκη και περιέχει μια πιθανολογική περιγραφή για αυτή τη συνθήκη.

Γνωστοί αλγόριθμοι για δέντρα απόφασης είναι:

- Ο αλγόριθμος COBWEB ο οποίος θεωρεί πως όλες οι μεταβλητές είναι ανεξάρτητες, μια αφελής τις περισσότερες φορές υπόθεση. Στόχος του αλγορίθμου είναι να πετυχαίνει υψηλή προβλεψιμότητα στις ονομαστικές μεταβλητές σε μια συστάδα. Ο αλγόριθμος δεν είναι κατάλληλος για μεγάλες βάσεις δεδομένων.
- Ο αλγόριθμος CLASSIT, αποτελεί μια επέκταση του αλγορίθμου COBWEB για συνεχή δεδομένα, αλλά και αυτός έχει τα ίδια αρνητικά.

Νευρωνικά Δίκτυα

Ένα νευρωνικό δίκτυο (*neural trees*) είναι ένα δίκτυο από απλούς υπολογιστικούς κόμβους (νευρώνες, νευρώνια), διασυνδεδεμένους μεταξύ τους.

Ένας πολύ γνωστός νευρωνικός αλγόριθμος για ομαδοποίηση είναι ο SOM (*self-organized map*), ο οποίος χρησιμοποιείται για αναγνώριση ομιλίας και για οπτικοποίηση πολυδιάστατων δεδομένων σε δύο ή τρεις διαστάσεις.

2.5.5 Μέθοδοι βασισμένοι στο Πλέγμα

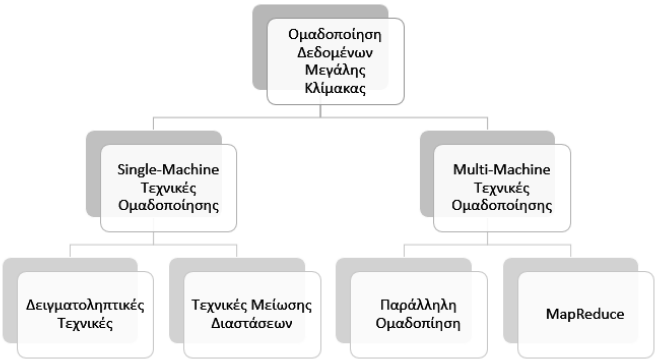
Στις μεθόδους βασισμένες στο πλέγμα ο χώρος χωρίζεται σε ένα πεπερασμένο αριθμό κελιών, σε μορφή πλέγματος, όπου στο κάθε ένα εφαρμόζεται η ομαδοποίηση. Το βασικό τους πλεονέκτημα είναι ο λιγοστός χρόνος ο οποίος χρειάζονται για την ανάλυση. Οι συνηθέστεροι αλγόριθμοι είναι ο Wave-Cluster και ο STING.

2.6 Ομαδοποίηση Δεδομένων Μεγάλης Κλίμακας

Όπως είδαμε και προηγούμενο κεφάλαιο, τα δεδομένα μεγάλης κλίμακας, εξαιτίας του όγκου τους, δεν μπορούν να αναλυθούν με τους παραδοσιακούς τρόπους ανάλυσης. Όπως είναι λογικό, το ίδιο ισχύει και στην ομαδοποίηση. Δηλαδή, αν προσπαθήσουμε να αναλύσουμε κατά συστάδες τα δεδομένα μεγάλης κλίμακας, που θα έχουμε στην διάθεσή μας, με τις παραδοσιακές μεθόδους ομαδοποίησης δεν αναμένουμε και τα καλύτερα δυνατά αποτελέσματα. Ακριβώς για αυτό τον λόγο έχουν προταθεί διάφορες άλλες τεχνικές. Οι τεχνικές αυτές χωρίζονται σε δύο κύριες κατηγορίες: single-machine τεχνικές ομαδοποίησης και multiple-machine τεχνικές ομαδοποίησης. Τα τελευταία χρόνια,την προσοχή έχουν τραβήξει οι multiple-machine τεχνικές ομαδοποίησης, καθώς προσφέρουν περισσότερες δυνατότητες και λιγότερο χρόνο επεξεργασίας των δεδομένων. Στο Σχήμα 2.7 φαίνονται οι τεχνικές που συμπεριλαμβάνονται σε κάθε κατηγορία.

2.6.1 Single-Machine Τεχνικές Ομαδοποίησης

Οι τεχνικές που βρίσκονται σε αυτή την κατηγορία εφαρμόζονται σε έναν μόνο υπολογιστή. Στη συνέχεια θα δούμε δύο από τις βασικότερες single-machine τεχνικές ομαδοποίησης, τις δειγματοληπτικές τεχνικές και τις τεχνικές μείωσης διαστάσεων.



Σχήμα 2.7: Τεχνικές Ομαδοποίησης Δεδομένων Μεγάλης Κλίμακας.

Δειγματοληπτικές Τεχνικές

Οι τεχνικές αυτές ήταν η πρώτη προσπάθεια για τη βελτίωση της ταχύτητας και των δυνατοτήτων που μπορούσε να προσφέρει η ανάλυση δεδομένων μεγάλης κλίμακας. Ο λόγος για τον οποίο έχουν ονομαστεί ως δειγματοληπτικές τεχνικές είναι εύκολα αντιληπτός, καθώς αντί να εφαρμοστεί η ομαδοποίηση σε όλα τα δεδομένα, εφαρμόζεται σε ένα δείγμα αυτών. Αφού έχει γίνει η ανάλυση σε ένα δείγμα των δεδομένων, τότε το αποτέλεσμα γενικεύεται για όλα τα δεδομένα. Η διαδικασία αυτή επιταχύνει κατά πολύ την όλη ανάλυση καθώς χρειάζεται λιγότερη υπολογιστική ισχύς και λιγότερος αποθηκευτικός χώρος. Οι αλγόριθμοι που ανήκουν σε αυτές τις τεχνικές είναι:

- Ο αλγόριθμος CLARANS (*Clustering Large Applications based on Randomized Sampling*) (βλέπε [12]), αποτελεί εξέλιξη του αλγορίθμου CLARA (*Clustering Large Applications*) που είναι αντίστοιχος του PAM (*Partition Around Medoids*) για δεδομένα μεγάλης κλίμακας. Στον αλγόριθμο PAM έχουμε αναφερθεί σε προηγούμενο κεφάλαιο.
- Ο αλγόριθμος BIRCH (*Balanced Iterative Reducing and Clustering using Hierarchies*) (βλέπε [14]), χρησιμοποιεί μια δική του δομή που ονομάζεται clustering feature (CF) όπως επίσης και το CF-δέντρο. Η CF είναι μια περιεκτική σύνοψη για την κάθε συστάδα. Πρόκειται για μια τριάδα (N, LS, SS) που περιέχει τον αριθμό των αντικειμένων σε μια συστάδα, το άθροισμα των αντικειμένων στη συστάδα και το άθροισμα τετραγώνων των αντικειμένων στη συστάδα.
- Ο αλγόριθμος CURE (*Clustering Using REpresentatives*), χρησιμοποιεί αντιπροσωπευτικά αντικείμενα μέσα από κάθε συστάδα. Με το να χρησιμοποιεί πάνω από ένα αντιπροσωπευτικό σημείο είναι αρκετά αποτελεσματικός στο να εντοπίζει μη σφαιρικές συστάδες, καθώς τα αντιπροσωπευτικά αντικείμενα δεν είναι μόνο απομακρυσμένα από το κέντρο της συστάδας αλλά και μεταξύ τους.

Τεχνικές Μείωσης Διαστάσεων

Παρόλο που η πολυπλοκότητα και ταχύτητα των αλγορίθμων συσταδοποίησης εξαρτώνται από τον αριθμό των παρατηρήσεων στη βάση δεδομένων, ένας άλλος καταλυτικός παράγοντας είναι οι διαστάσεις που μπορεί να έχουν τα δεδομένα μας. Για την ακρίβεια, όσο περισσότερες διαστάσεις έχουν τα δεδομένα μας τόσο πιο περίπλοκη γίνεται η ανάλυση μας, επηρεάζοντας και τον τελικό χρόνο εκτέλεσης. Οι δειγματοληπτικές τεχνικές, που είδαμε στην προηγούμενη παράγραφο, μπορούν να μειώσουν τον όγκο των δεδομένων μας αλλά δεν προσφέρουν κάποια λύση στη περίπτωση των πολλών διαστάσεων των δεδομένων. Στη συνέχεια θα δούμε δύο τεχνικές μείωσης διαστάσεων.

- *Random Projection*. Στη τεχνική αυτή, τα d-διάστατα δεδομένα προβάλλονται σε ένα k-διάστατο υποσύνολο, όπου $k << d$. Οι περισσότεροι αλγόριθμοι ομαδοποίησης βασίζονται στην απόσταση, έτσι περιμένουμε πως το αποτέλεσμα στον k-διάστατο χώρο θα μας δώσει παρόμοια αποτελέσματα με αυτά που θα είχαμε στον αρχικό χώρο. Η τεχνική *Random Projection* μπορεί να πραγματοποιηθεί μέσω ενός γραμμικού μετασχηματισμού του πίνακα A των αρχικών δεδομένων. Εάν θεωρήσουμε τον πίνακα περιστροφής R, $d \times k$ διαστάσεων και τα στοιχεία του $R(i, j)$ τα οποία είναι ανεξάρτητες τυχαίες μεταβλητές, τότε ο

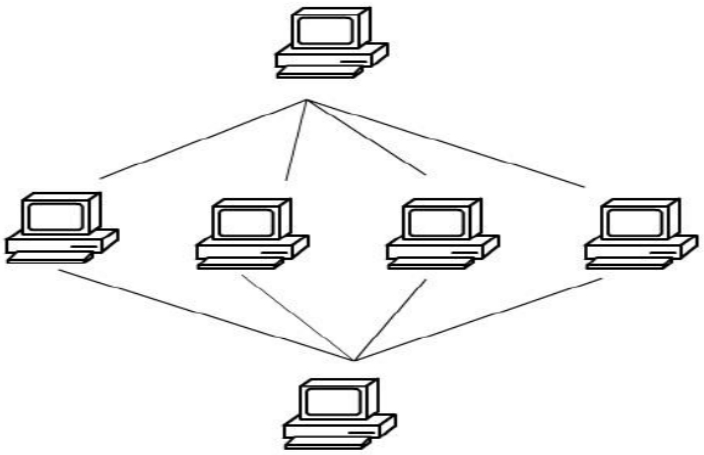
$A' = A.R$ είναι ο πίνακας προβολής του A στον k-διάστατο χώρο. Τότε κάθε σειρά του A' έχει k διαστάσεις. Τέλος, πρέπει να αναφέρουμε πως ο πίνακας περιστροφής R είναι διαφορετικός σε κάθε διαφορετικό random projection αλγόριθμο.

- *Global Projection*. Στην τεχνική αυτή, το ζητούμενο είναι κάθε αντικείμενο που προβάλλεται να είναι όσο το δυνατόν κοντύτερα στο αρχικό αντικείμενο. Αν θεωρήσουμε πως ο πίνακας A περιέχει τα αρχικά δεδομένα και ο πίνακας A' είναι η προβολή του, τότε ο στόχος της τεχνικής global projection είναι να ελαχιστοποιήσει την ποσότητα $\|A' - A\|$. Υπάρχουν διάφορες τεχνικές για την κατασκευή του πίνακα A', μερικές από αυτές είναι οι SVD (singular value decomposition), CX/CUR, CMD και Colibri. (βλέπε [12])

2.6.2 Multi-Machine Τεχνικές Ομαδοποίησης

Παρόλο που οι δειγματοληπτικές τεχνικές και οι τεχνικές μείωσης διαστάσεων που χρησιμοποιούνται στις single-machine τεχνικές ομαδοποίησης, που είδαμε στην προηγούμενη παράγραφο, βελτιώνουν την απόδοση και την ταχύτητα της ομαδοποίησης, πλέον στις μέρες μας η αύξηση του όγκου των δεδομένων είναι κατά πολύ μεγαλύτερη από τη βελτίωση των δυνατοτήτων των υπολογιστών όσον αφορά θέματα μνήμης και επεξεργαστή. Επομένως, ένας υπολογιστής με έναν επεξεργαστή και μια μνήμη δε μπορεί να διαχειριστεί terabytes και petabytes δεδομένων. Για το λόγο αυτό δημιουργείται η ανάγκη για αλγόριθμους που μπορούν να τρέξουν σε πολλούς υπολογιστές. Όπως φαίνεται στο Σχήμα 2.8, αυτή η τεχνική μας επιτρέπει να χωρίσουμε το μεγάλο όγκο δεδομένων σε μικρότερα κομμάτια, τα οποία μπορούν να φορτωθούν σε διάφορους υπολογιστές και στη συνέχεια να χρησιμοποιηθεί η επεξεργαστική ισχύς αυτών των υπολογιστών για να ολοκληρωθεί η ανάλυση. Οι multi-machine τεχνικές ομαδοποίησης χωρίζονται σε δύο κατηγορίες:

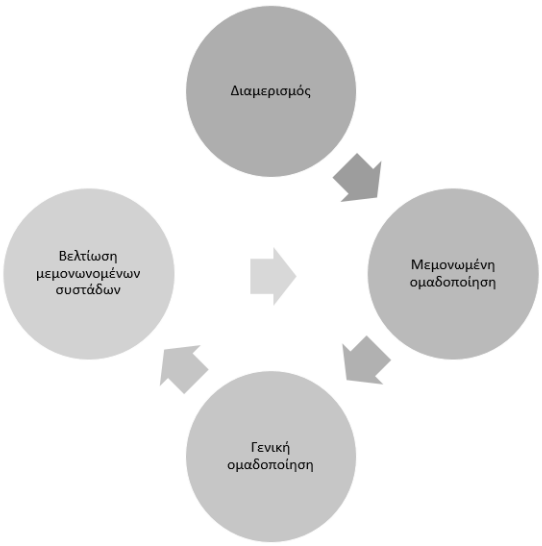
- Παράλληλη ομαδοποίηση
- MapReduce.



Σχήμα 2.8: Γενική ιδέα των multi-machine τεχνικών ομαδοποίησης.

Στην παράλληλη ομαδοποίηση οι αναλυτές ασχολούνται και με προκλήσεις που αφορούν τις λεπτομέρειες σχετικές με την διαδικασία κατανομής των δεδομένων στους διάφορους υπολογιστές, κάτι που κάνει τη διαδικασία αυτή περισσότερο περίπλοκη και χρονοβόρα. Οι διαφορές μεταξύ των αλγορίθμων παράλληλης ομαδοποίησης και στο MapReduce είναι πως το δεύτερο απαλλάσσει τους αναλυτές από προβλήματα που αφορούν τη φόρτωση των δεδομένων, την κατανομή τους καθώς και την ανοχή σε σφάλματα, αφού τα διαχειρίζεται αυτόματα. Αυτή η ιδιότητα επιτρέπει την ευκολότερη και γρηγορότερη επεκτασιμότητα στο παράλληλο αυτό σύστημα. Αυτοί οι αλγόριθμοι ακολουθούν έναν γενικό κύκλο, όπως φαίνεται στο Σχήμα 2.9.

Στο πρώτο στάδιο, τα δεδομένα διαχωρίζονται και στη συνέχεια κατανέμονται στους διάφορους υπολογιστές. Έπειτα, κάθε υπολογιστής πραγματοποιεί ομαδοποίηση μεμονωμένα στο αντίστοιχο μέρος των δεδομένων. Δύο βασικές προκλήσεις που έχουν να αντιμετωπίσουν οι multi-machine αλγόριθμοι είναι η ελαχιστοποίηση της κίνησης των δεδομένων και η χαμηλή ακρίβεια. Η χαμηλή ακρίβεια στους multi-machine αλγορίθμους μπορεί να προκληθεί για δύο κύριους λόγους. Πρώτον, είναι πιθανόν διαφορετικοί αλγόριθμοι ομαδοποίησης



Σχήμα 2.9: Γενικός κύκλος για τους multi-machine αλγορίθμους ομαδοποίησης.

να εφαρμοστούν στους διάφορους υπολογιστές. Δεύτερον, ακόμη και αν ο ίδιος αλγόριθμος χρησιμοποιηθεί σε όλους τους υπολογιστές, σε κάποιες περιπτώσεις τα διαχωρισμένα δεδομένα μπορεί να προκαλέσουν αλλαγή στο τελικό αποτέλεσμα της ομαδοποίησης. Στη συνέχεια θα δούμε τους αλγορίθμους για παράλληλη ομαδοποίηση και MapReduce.

Παράλληλη Ομαδοποίηση

Παρόλο που η παράλληλη ομαδοποίηση δημιουργεί δυσκολίες στους αναλυτές, έχει ιδιαίτερη αξία καθώς βελτιώνει τους αλγορίθμους ομαδοποίησης, όσον αφορά την επεκτασιμότητα και την ταχύτητα. Μερικοί αλγόριθμοι παράλληλης ομαδοποίησης που αξίζουν αναφοράς είναι ο DBDC και ο ParMETIS (βλέπε [12]).

Ο parallel k-means, που προτάθηκε από τους Dhillon και Modha εφαρμόστηκε σε 16 παράλληλους υπολογιστές. Επίσης, οι Stoffel και Belkoniene εφάρμοσαν μια επέκταση του parallel k-means χρησιμοποιώντας 32 υπολογιστές σε ένα Ethernet δίκτυο και απέδειξαν μια σχεδόν γραμμική επιτάχυνση για δεδομένα μεγάλης κλίμακας. Για περισσότερες πληροφορίες παραπέμπουμε στο [13].

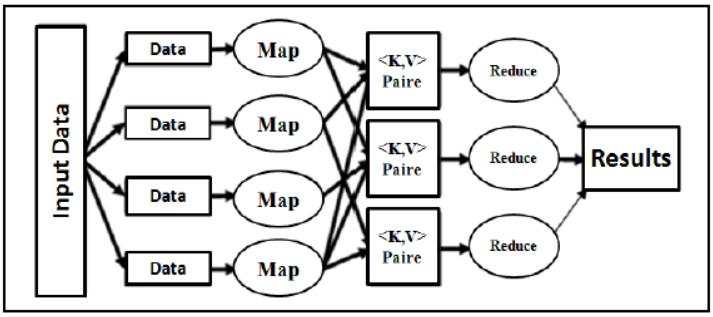
MapReduce

Παρόλο που η παράλληλη ομαδοποίηση βελτίωσε την επεκτασιμότητα και την ταχύτητα των αλγορίθμων ομαδοποίησης, η πολυπλοκότητα της διαχείρισης της κατανομής της μνήμης και του επεξεργαστή των υπολογιστών,αποτελούσε ακόμη μια πρόκληση. Το MapReduce είναι ένας μηχανισμός που διαχωρίζει τις εργασίες (με μεγάλους όγκους δεδομένων) με σκοπό την κατανεμημένη εκτέλεση αυτών σε πολλούς servers. Αρχικά οι εργασίες διαχωρίζονται σε μικρότερες διεργασίες (Map). Στη συνέχεια, οι διεργασίες αυτές αποστέλλονται σε διαφορετικούς servers και τα αποτελέσματα συλλέγονται και αποθηκεύονται (Reduce). Το προαναφερθέν πλαίσιο, όπως φαίνεται στο Σχήμα 2.10, παρουσιάστηκε αρχικά από τη Google και τη Hadoop.

Στο στάδιο Map τα εισαχθέντα δεδομένα αναλύονται, χωρίζονται σε υποκατηγορίες και αποστέλλονται σε άλλους κόμβους (οι οποίο κάνουν την ίδια διαδικασία αναδρομικά). Αυτό μπορεί να εκτελεστεί αργότερα χρησιμοποιώντας τη συνάρτηση Map η οποία έχει ένα ζεύγος (κλειδί,τιμή) το οποίο συσχετίζεται με νέα ζεύγη (κλειδί,τιμή).

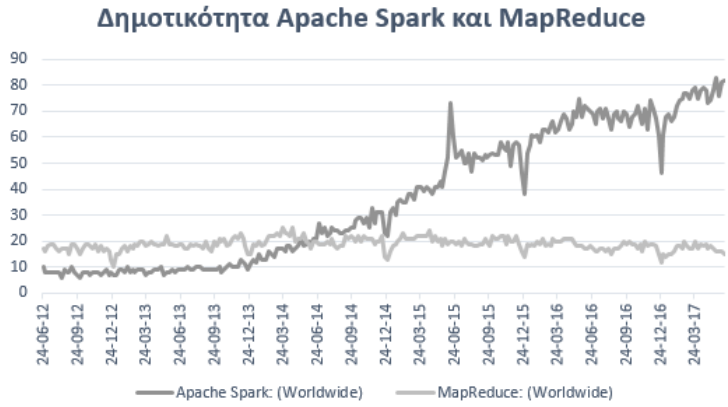
Στη συνέχεια ακολουθεί το στάδιο Reduce, όπου οι κατώτεροι κόμβοι επιστρέφουν τα αποτελέσματά τους στους γονείς κόμβους, που τους τα έχουν ζητήσει. Στο στάδιο αυτό υπολογίζεται ένα μερικό αποτέλεσμα χρησιμοποιώντας τη συνάρτηση Reduce που περιλαμβάνει όλες τις αντίστοιχες τιμές για το ίδιο κλειδί σε ένα μοναδικό ζεύγος (κλειδί,τιμή). Έπειτα, επιστρέφει πληροφορία με τη σειρά του.

Διάφοροι προσεγγιστικοί μέθοδοι που έχουν προταθεί και χρησιμοποιήθηκαν στο MapReduce με σκοπό να βελτιώσουν τους υπάρχοντες αλγορίθμους συσταδοποίησης, βρίσκονται στα [13] και [12].



Σχήμα 2.10: Το πλαίσιο του MapReduce.

Η συσταδοποίηση είναι μία από τις πιο σημαντικές διεργασίες στην εξόρυξη δεδομένων και στη σημερινή εποχή απαιτείται συνεχής βελτίωση με σκοπό οι αναλυτές δεδομένων να επιτύχουν τη βέλτιστη εξόρυξη γνώσης από terabytes και petabytes δεδομένων. Δεδομένα τέτοιας κλίμακας απαιτούν μεγάλη διάρκεια εκτέλεσης στο MapReduce, παρόλο που τα δεδομένα διαχωρίζονται και αναλύονται ξεχωριστά. Για να επιλυθεί αυτό το πρόβλημα, δημιουργήθηκε η πλατφόρμα Apache Spark, που αποτελεί μια επέκταση του MapReduce και η διαφορά της έγκειται στην in-memory επεξεργασία των δεδομένων. Αυτή η in-memory διαδικασία είναι γρηγορότερη, αφού δε χάνεται χρόνος στη μεταφορά των δεδομένων από και προς το δίσκο, κάτι που ισχύει για το MapReduce. Η Apache Spark, για την οποία θα μιλήσουμε αναλυτικά στο επόμενο κεφάλαιο, γίνεται όλο και περισσότερο δημοφιλής με το πέρασμα του χρόνου συγκριτικά με το MapReduce. Αυτό είναι εμφανές και από το Σχήμα 2.11, όπου παρουσιάζεται η σύγκριση της δημοτικότητας των όρων αναζήτησης Apache Spark και MapReduce στο Google τα τελευταία πέντε έτη.



Σχήμα 2.11: Σύγκριση δημοτικότητας των Apache Spark και MapReduce.

Κεφάλαιο 3

Πλατφόρμα Spark

3.1 Εισαγωγή

Στο κεφάλαιο αυτό θα κάνουμε μια εισαγωγή στην πλατφόρμα Apache Spark. Αρχικά θα παρουσιάσουμε τι είναι και από ποιές συνιστώσες αποτελείται. Ιδιαίτερη έμφαση θα δοθεί στην βιβλιοθήκη MLlib, που αποτελεί την βιβλιοθήκη της Spark για μηχανική μάθηση, καθώς στη βιβλιοθήκη αυτή περιέχονται και οι αλγόριθμοι που αποτελούν τον κεντρικό κορμό της παρούσας εργασίας.

3.2 Τι είναι η Apache Spark

Η Apache Spark είναι μια open source υπολογιστική πλατφόρμα για την ανάλυση μεγάλης κλίμακας δεδομένων, σχεδιασμένη να είναι γενικού σκοπού, εύκολη και γρήγορη (βλέπε [15]). Η Apache Spark έχει γίνει δωρεά από το 2013 στο Apache Software Foundation και πλέον αποτελεί ένα από τα βασικότερα προγράμματα του ιδρύματος.

Η πλατφόρμα Spark μπορεί να θεωρηθεί γενικού σκοπού καθώς πρακτικά αποτελεί μια επέκταση του πολύ διαδεδομένου μέχρι σήμερα MapReduce αφού υποστηρίζει ευρύτερο φάσμα διεργασιών. Οι σημαντικότερες διεργασίες τις οποίες πραγματοποιεί είναι: περισσότεροι τύποι υπολογισμών, μαζικές εφαρμογές, επαναληπτικοί αλγόριθμοι, διαδραστικές εντολές SQL και επεξεργασία ροών. Υποστηρίζοντας όλες αυτές τις διεργασίες σε έναν μόνο υπολογιστή, η Spark μπορεί να αποτελέσει την εναλλακτική λύση για πολλές εφαρμογές όπου έχουν την δυνατότητα να πραγματοποιήσουν μόνο μία από της παραπάνω διεργασίες.

Το επόμενο πλεονέκτημα της Spark είναι πως είναι πολύ προσιτή στην χρήση. Προσφέροντας απλά APIs για τις γλώσσες προγραμματισμού Python, Java, Scala και την R όπως επίσης και πολλές built-in βιβλιοθήκες. Επίσης, μπορεί να ενσωματώσει και άλλα εργαλεία για την ανάλυση μεγάλης κλίμακας δεδομένων. Τέλος η Spark μπορεί να λειτουργήσει σε μια συστοιχία του Hadoop, όπως είναι το Hadoop YARN, και το Apache Mesos, βέβαια μπορεί να λειτουργήσει και ως αυτόνομη πλατφόρμα με τον δικό της scheduler.

Στην σημερινή εποχή όπου ο χρόνος επεξεργασίας των δεδομένων ίσως να θεωρείται από τους βασικότερους παράγοντες της ανάλυσης, η Spark μπορεί να προσφέρει μια από τις κυριότερες ικανότητες της, την ικανότητα να πραγματοποιεί in-memory υπολογισμούς. Κάτι που μας παρέχει γρηγορότερη ανάλυση των δεδομένων αλλά και ανταπόκριση. Βέβαια υπερτερεί του MapReduce και όσον αφορά περίπλοκες εφαρμογές που πραγματοποιούνται στον δίσκο του υπολογιστή.

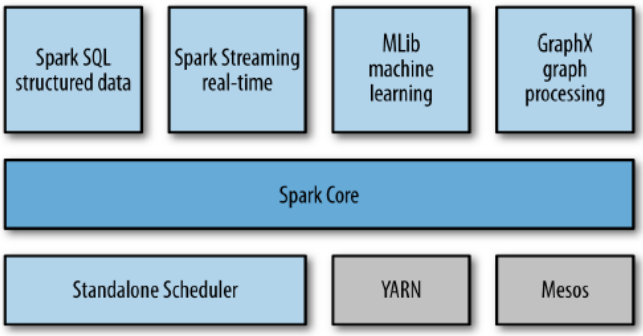
3.3 Οι συνιστώσες της Spark

Η Spark περιέχει πολλαπλές συνιστώσες. Αρχικά, ο πυρήνας της Spark είναι μια «υπολογιστική μηχανή» η οποία είναι υπεύθυνη για τον σχεδιασμό και την κατανομή των διεργασιών. Επειδή ο πυρήνας της Spark είναι σχεδιασμένος να είναι γρήγορος και γενικού σκοπού, μπορεί να υποστηρίξει πολλαπλές συνιστώσες για διαφορετικές διεργασίες, όπως επεξεργασία ροών ή μηχανική μάθηση. Αυτές οι συνιστώσες είναι σχεδιασμένες να λειτουργούν παράλληλα, επιτρέποντας έτσι στον χρήστη να τις συνδυάζει σαν βιβλιοθήκες λογισμικού.

Η ιδέα της στενής σχέσης μεταξύ των συνιστωσών της Spark έχει πολλά οφέλη. Αρχικά, όταν υπάρχει μια βελτιστοποίηση στον πυρήνα αυτόματα βελτιώνονται και οι βιβλιοθήκες της επεξεργασίας ροών ή της μηχανικής μάθησης για παράδειγμα. Ένα ακόμη θετικό αυτής της στενής σχέσης είναι πως ο χρόνος εκτέλεσης των διεργασιών μειώνεται. Αυτό συμβαίνει διότι αντί να λειτουργούν ταυτόχρονα διάφορα λειτουργικά συστήματα, ο

χρήστης μπορεί να λειτουργεί μόνο ένα. Κάτι που ταυτόχρονα σημαίνει πως εάν μια νέα συνιστώσα προστεθεί στη Spark, ο χρήστης μπορεί αυτόματα να έχει πρόσβαση σε αυτή, χωρίς να χρειάζεται να αγοράσει ή να κατεβάσει ένα νέο λογισμικό. Ένα τελευταίο πλεονέκτημα είναι πως μπορούν να πραγματοποιηθούν διάφορες διεργασίες συνδυάζοντας διαφορετικά μοντέλα επεξεργασίας. Για παράδειγμα, στη Spark μπορεί να δημιουργηθεί μια εφαρμογή που χρησιμοποιεί τεχνική μάθηση.

Στο Σχήμα 3.1 που ακολουθεί μπορούμε να δούμε συνοπτικά τις συνιστώσες της Spark.



Σχήμα 3.1: Οι συνιστώσες της Apache Spark.

3.3.1 Ο πυρήνας της Spark

Ο πυρήνας της Spark περιέχει την βασική λειτουργία της Spark, συμπεριλαμβάνοντας συνιστώσες για προγραμματισμό διεργασιών, διαχείριση μνήμης, αλληλεπίδραση με τα συστήματα αποθήκευσης κ.α. Στον Πυρήνα επίσης βρίσκονται και τα APIs τα οποία ονομάζονται RDDs (*Resilient Distributed Datasets*). Τα RDDs αντιπροσωπεύουν μια συλλογή αντικειμένων, κατανεμημένα σε πολλούς υπολογιστικούς κόμβους έτσι ώστε να χειρίζονται παράλληλα. Ο πυρήνας της Spark παρέχει πολλά APIs για την κατασκευή και την διαχείριση αυτών των συλλογών.

3.3.2 Spark SQL

Η Spark SQL είναι ένα πακέτο της Spark για εργασίες με δομημένα δεδομένα. Επιτρέπει στον χρήστη να χρησιμοποιεί queries για να επεξεργάζεται τα δεδομένα μέσω της SQL αλλά και μέσω της HQL (*Hive Query Language*), η οποία υποστηρίζει πολλούς τύπος δεδομένων όπως πίνακες Hive, Parquet και JSON. Εκτός από το να προσφέρει ένα περιβάλλον της SQL στην Spark, η Spark SQL επιτρέπει στους αναλυτές να συνενώσουν, σε μια εφαρμογή, queries της SQL με προγραμματισμό στις γλώσσες προγραμματισμού Python, Java, Scala και R μέσω συγκεκριμένων RDDs.

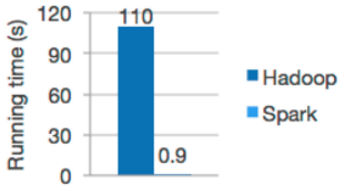
3.3.3 Spark Streaming

Η Spark Streaming είναι μια συνιστώσα της Spark που πραγματοποιεί επεξεργασία για ροές δεδομένων. Ένα παραδείγματα για ροές δεδομένων είναι Log Files που παράγονται από servers σε μια γραμμή παραγωγής ενός εργοστασίου. Η Spark Streaming παρέχει ένα API για διαχείριση ροών δεδομένων αντίστοιχο με αυτό του πυρήνα της Spark, κάτι που διευκολύνει τους αναλυτές καθώς μπορούν να διαχειρίζονται δεδομένα τα οποία είναι αποθηκευμένα στη μνήμη του υπολογιστή είτε να καταφθάνουν σε πραγματικό χρόνο.

3.3.4 MLlib

Η Spark περιέχει μια βιβλιοθήκη στην οποία βρίσκονται οι λειτουργίες της μηχανικής μάθησης, που ονομάζεται MLlib. Η MLlib παρέχει πολλούς τύπους αλγορίθμων μηχανικής μάθησης, μεταξύ των οποίων αλγορίθμους για Παλινδρόμηση, Κατηγοριοποίηση, Ομαδοποίηση, Συνεργατική Διήθηση (*Collaborative Filtering*) κ.α. Η βιβλιοθήκη αυτή είναι σχεδιασμένη να παρέχει γρήγορη και γενικού σκοπού ανάλυση δεδομένων μεγάλης κλίμακας. Συγκρίνοντας την με το Hadoop MapReduce, μπορεί να εκτελέσει διεργασίες εκατό φορές πιο γρήγορα στη μνήμη ή δέκα φορές πιο γρήγορα στον δίσκο του υπολογιστή. Στο Σχήμα 3.2 που ακολουθεί βλέπουμε μια

σύγκριση του χρόνου εκτέλεσης (σε δευτερόλεπτα) Λογιστικής Παλινδρόμησης στο Hadoop MapReduce και στη βιβλιοθήκη MLlib της Spark.



Σχήμα 3.2: Χρόνος εκτέλεσης Λογιστικής Παλινδρόμησης στο Hadoop και στη Spark.

Στη συνέχεια θα παρουσιάσουμε συνοπτικά τους αλγορίθμους που περιλαμβάνονται την βιβλιοθήκη MLlib. Αρχικά, μέσω της βιβλιοθήκης MLlib μπορούμε να χρησιμοποιήσουμε βασικές στατιστικές μεθόδους, όπως συσχέτιση μεταβλητών και ελέγχους υποθέσεων. Οι αλγόριθμοι για συσχέτιση μεταβλητών είναι ο συντελεστής συσχέτισης του Pearson και ο συντελεστής συσχέτησης του Spearman. Όσον αφορά τους ελέγχους υποθέσεων, στη παρούσα φάση, διαθέσιμος είναι μόνο ο έλεγχος χ^2 του Pearson.

Επίσης, υπάρχει μια πληθώρα αλγορίθμων για εξαγωγή, μετασχηματισμό και επιλογή δεδομένων. Η βιβλιοθήκη MLlib περιέχει τις ακόλουθες τρεις μεθόδους για εξαγωγή ακατέργαστων δεδομένων: TF-IDF, Word2Vec και CountVectorizer. Για τον μετασχηματισμό των δεδομένων, ο χρήστης μπορεί να βρει πάνω από είκοσι διαφορετικούς μετασχηματισμούς, μεταξύ των οποίων είναι η Τυποποίηση, οι Κανονικοποίησεις MinMax και MaxAbs, η Ανάλυση Κυρίων Συνιστωσών (PCA) κ.α. Στο κομμάτι της επιλογής δεδομένων υπάρχουν οι τρεις ακόλουθοι αλγόριθμοι: VectorSlicer, RFormula και ChiSqSelector.

Ακολούθως, θα αναφέρουμε τις κατηγορίες αλγορίθμων για Κατηγοριοποίηση και για Παλινδρόμηση στη βιβλιοθήκη MLlib. Έτσι λοιπόν μπορούμε να βρούμε αλγόριθμους για Γραμμική και Λογιστική Παλινδρόμηση (Διωνυμική και Πολυωνυμική), για Κατηγοριοποίηση Naive Bayes, για Κατηγοριοποίηση με δέντρα απόφασης κ.α.

Ένα επιπλέον μέρος της μηχανικής μάθησης που καλύπτεται από την βιβλιοθήκη MLlib της Spark είναι η Συνεργατική Διήθηση (*Collaborative Filtering*). Ο αλγόριθμος που χρησιμοποιείται κατά κύριο λόγο για αυτού του είδους τις διεργασίες είναι ο αλγόριθμος ALS (*Alternating Least Squares*), όπως και κάποιες βελτιώσεις του.

Όπως είναι φυσικό και οι αλγόριθμοι για την Ομαδοποίηση βρίσκονται σε αυτή τη βιβλιοθήκη και συνοπτικά είναι οι: K-means, Gaussian Mixture Model, Power Iteration Clustering (PIC), Latent Diririchlet Allocation (LDA), Bisecting K-means και Streaming K-means.

Στη συνέχεια θα παρουσιάσουμε κάποιες γενικές πληροφορίες για τους αλγορίθμους αυτούς.

K-means

Ο K-means είναι ένας ευρέως χρησιμοποιούμενος αλγόριθμος ομαδοποίησης, ο οποίος ομαδοποιεί τα διάφορα σημεία του συνόλου δεδομένων σε έναν προκαθορισμένο αριθμό από ομάδες. Η εφαρμογή στην MLlib περιλαμβάνει μια παραλληλισμένη παραλλαγή της μεθόδου k-means++, η οποία ονομάζεται k-means|| (ή διαφορετικά κλιμακωτός k-means++).

Αλγόριθμος 1 *k-means++(k)* αρχικοποίηση.	
1:	$C \leftarrow$ επίλεξε ομοιόμορφα ένα σημείο στη τύχη από την X
2:	**όσο** $\lvert C\rvert < k$ **εκτέλεσε**
3:	Επίλεξε $x \in X$ με πιθανότητα $p_x = \frac{d^2(x,C)}{\phi_x(C)}$
4:	$C \leftarrow C \cup x$
5:	**τέλος**

Πίνακας 3.1: Ο ψευτοκώδικας για τον αλγόριθμο k-means++.

Αλγόριθμος 2 *k-means||(k,l)* αρχικοποίηση.

1: $C \leftarrow$ επίλεξε ομοιόμορφα ένα σημείο στη τύχη από την X
2: $\psi \leftarrow \phi x(C)$
3: **για** $O(\log \psi)$ φορές **εκτέλεσε**
4: $C' \leftarrow$ επίλεξε $x \in X$ με πιθανότητα $p_x = \frac{l \cdot d^2(x,C)}{\phi_x(C)}$
5: $C \leftarrow C \cup C'$
6: **τέλος**
7: Για $x \in C$,θέσε w_x να είναι ο αριθμός των σημείων στο X πλησιέστερα στο x από κάθε άλλο σημείο στο C
8: Επαναομαδοποίησε τα επιβαρυμένα σημεία στο C σε k ομάδες

Πίνακας 3.2: Ο ψευτοκώδικας για τον αλγόριθμο k-means||.

Για περισσότερες πληροφορίες σχετικά με τους k-means++ και k-means|| βλέπε [19]

Gaussian Mixture Model

Ο αλγόριθμος αυτός αποτελεί μια προσέγγιση βασισμένη σε μοντέλο, καθώς βασίζεται στη χρήση συγκεκριμένων μοντέλων για τις ομάδες και στην προσπάθεια να βελτιστοποιήσει την προσαρμογή μεταξύ των δεδομένων και των μοντέλων. Ένα μοντέλο Gaussian Mixture παρουσιάζει μια σύνθετη κατανομή της οποίας τα σημεία προέρχονται από k Gaussian υπο-κατανομές. Το πακέτο MLlib χρησιμοποιεί τον αλγόριθμο *Expectation-Maximization* (EM) για να δημιουργήσει το μοντέλο μέγιστης πιθανοφάνειας, δοθέντος ενός συνόλου δειγμάτων.

Ειδικότερα, κάθε ομάδα μπορεί να αναπαρασταθεί μαθηματικά από μια παραμετρική Gaussian κατανομή με κέντρο τα βαρύκεντρα τους και ολόκληρο το σύνολο δεδομένων από μια μίξη από αυτές τις κατανομές. Ο αλγόριθμος δουλεύει ως εξής (βλέπε [20]):

- Διαλέγει τη συνιστώσα (Gaussian) τυχαία με πιθανότητα $P(w_i)$
- Δοκιμάζει ένα σημείο $N(\mu_i, \sigma_i^2)$.

Υποθέτουμε ότι έχουμε

- $x_1, x_2, ..., x_N$
- $P(w_1), ..., P(w_K), \sigma$

Μπορούμε να υπολογίσουμε την πιθανότητα $P(x|w_i, \mu_1, \mu_2, ..., \mu_K)$. Αυτό που θέλουμε είναι να μεγιστοποιήσουμε την $P(x|w_i, \mu_1, \mu_2, ..., \mu_K)$. Δεδομένου ότι $P(x|\mu_i) = \sum_i (w_i)(x|w_i, \mu_1, \mu_2, ..., \mu_K)$, μπορούμε να πούμε ότι P(σύνολο δεδομένων$|\mu_i) = \prod_1^N \sum_i P(w_i)P(x|w_i, \mu_1, \mu_2, ..., \mu_K)$ Αυτό που μένει είναι να μεγιστοποιήσουμε τη συνάρτηση πιθανοφάνειας με χρήση κάθε φορά της εξίσωσης $\frac{\partial L}{\partial \mu_i} = 0$. Ο υπολογισμός αυτός είναι πολύ δύσκολος και για αυτό χρησιμοποιείται ο απλοποιημένος αλγόριθμος EM.

Για πληροφορίες σχετικά με τον EM αλγόριθμο βλέπε [21].

Power Iteration Clustering (PIC)

Ο αλγόριθμος PIC ομαδοποιεί κορυφές ενός γραφήματος το οποίο έχει ως ιδιότητες ακμής όλα τα δυνατά ζεύγη ομοιότητας που προκύπτουν από το σύνολο των δεδομένων μας. Υπολογίζει ένα ψευδο-ιδιοδιάνυσμα του κανονικοποιημένου πίνακα συγγένειας (*affinity matrix*) μέσω της επανάληψης ισχύος και το χρησιμοποιεί για να ομαδοποιήσει κορυφές. Ο αλγόριθμος προϋποθέτει ότι οι ομοιότητες είναι μη αρνητικές και ότι το μέτρο ομοιότητας είναι συμμετρικό. Επίσης, ένα ζεύγος, ανεξαρτήτου διάταξης, πρέπει να εμφανίζεται το πολύ μια φορά στα δεδομένα εισόδου, ενώ αν ένα ζεύγος λείπει από τα δεδομένα εισόδου τότε το μέτρο ομοιότητάς τους θεωρείται ίσο με μηδέν. Πρόκειται για έναν κλιμακωτό και αποτελεσματικό αλγόριθμο. Για περισσότερες πληροφορίες σχετικά με τον αλγόριθμο βλέπε [22]

Latent Dirichlet Allocation (LDA)

Ο LDA αλγόριθμος είναι ένα μοντέλο θέματος, το οποίο καταλήγει σε ένα θέμα από μια συλλογή εγγράφων κειμένου. Ο LDA μπορεί να θεωρηθεί ως αλγόριθμος ομαδοποίησης με τις εξής παραδοχές:

- Τα θέματα αντιστοιχούν στα κέντρα των ομάδων και τα έγγραφα αντιστοιχούν σε παραδείγματα (*rows*) σε ένα σύνολο δεδομένων.
- Τα θέματα και τα έγγραφα υπάρχουν σε έναν χώρο, στον οποίο τα διανύσματα είναι διανύσματα από αριθμούς λέξεων.
- Αντί να φτιάχνει μια ομαδοποίηση χρησιμοποιώντας μια παραδοσιακή απόσταση, χρησιμοποιεί μια λειτουργία που βασίζεται σε ένα στατιστικό μοντέλο για το πώς παράγονται τα κείμενα.

Σημειώνουμε ότι ο αλγόριθμος αυτός βρίσκεται ακόμη σε πειραματικό στάδιο και αναπτύσσεται ακόμη.

Bisecting K-means

Ο αλγόριθμος αυτός συχνά είναι ταχύτερος από τον κλασικό K-means και γενικά παράγει διαφορετικά ομαδοποίηση. Πρόκειται για ένα είδος ιεραρχικής ομαδοποίησης (η οποία είναι μια από τις πιο ευρέως χρησιμοποιούμενες μεθόδους ομαδοποίησης), που επιδιώκει να χτίσει μια ιεραρχία ομάδων χρησιμοποιώντας την προσέγγιση από κάτω προς τα πάνω (διαχωριστικός αλγόριθμος). Αυτό σημαίνει πως όλες οι παρατηρήσεις ξεκινούν από το ίδιο κέντρο και εκτελούνται αναδρομικά διαχωρισμοί καθώς κινείται προς τα κάτω στην ιεραρχία. Ο ψευδοκώδικας του αλγορίθμου αυτού είναι:

1. Διάλεξε μια ομάδα για να διαχωρίσεις
2. Βρες 2 υπο-ομάδες χρησιμοποιώντας τον κλασικό K-means αλγόριθμο (βήμα διχοτόμησης - bisecting)
3. Επανάλαβε το προηγούμενο βήμα, το βήμα διχοτόμησης, για X φορές και διάλεξε τον διαχωρισμό που παράγει την ομαδοποίηση με τη μέγιστη συνολική ομοιότητα (όπου X είναι ο μέγιστος αριθμός επαναλήψεων)
4. Επανάλαβε τα βήματα 1,2 και 3 μέχρι να επιτευχθεί ο επιθυμητός αριθμός ομάδων.

Για περισσότερες πληροφορίες σχετικά με τον αλγόριθμο αυτό βλέπε [23]

Streaming K-means

Όταν τα δεδομένα φτάνουν μέσω ενός ρεύματος (*stream*), είναι πιθανό να επιθυμούμε να υπολογίσουμε δυναμικά τα κέντρα, ενημερώνοντάς τα καθώς φθάνουν νέα δεδομένα. Ο αλγόριθμος χρησιμοποιεί μια γενίκευση του κανόνα ενημέρωσης K-means mini-batch, σύμφωνα με την οποία, για κάθε νέα παρτίδα δεδομένων, αναθέτουμε όλα τα σημεία στο πλησιέστερο κέντρο και υπολογίζουμε εκ νέου τα κέντρα των ομάδων.

3.3.5 GraphX

Η GraphX είναι μια βιβλιοθήκη της Spark για την επεξεργασία γραφημάτων. Όπως η Spark SQL και η Spark Streaming, έτσι και η GraphX επεκτείνει τις βασικές λειτουργίες του πυρήνα της Spark επιτρέποντας στον χρήστη παράλληλη επεξεργασία γραφημάτων με κατάλληλες ιδιότητες για κάθε ακμή και κόμβο. Επίσης η GraphX περιέχει μερικούς από τους βασικότερους αλγορίθμους για ανάλυση γραφημάτων, όπως:

- τον αλγόριθμο PageRank, ο οποίος μετρά την σημαντικότητα κάθε κόμβου στο γράφημα,
- τον αλγόριθμο Connected Components, που υποδεικνύει κάθε συνδεδεμένο στοιχείο του γραφήματος με το ID του κόμβου με τις χαμηλότερες αριθμήσεις
- και τον αλγόριθμο Triangle Counting, ο οποίος αριθμεί τα τρίγωνα τα οποία σχηματίζει ο κάθε κόμβος.

3.3.6 Cluster Managers

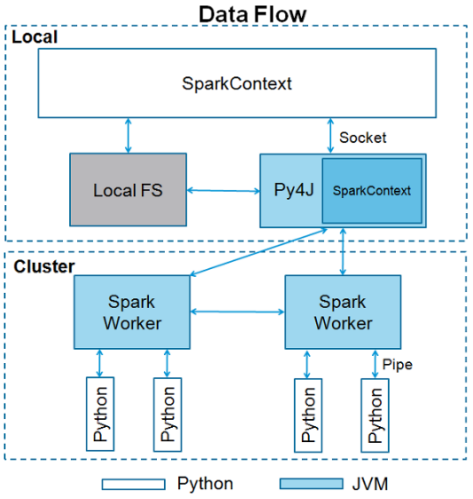
Η πλατφόρμα Spark είναι σχεδιασμένη να λειτουργεί εξίσου αποτελεσματικά τόσο σε έναν όσο και σε πολλές χιλιάδες υπολογιστικούς κόμβους. Οι εφαρμογές στη Spark μπορούν να λειτουργήσουν ανεξάρτητα σε κάθε κομμάτι από μια συστοιχία υπολογιστών, καθοδηγημένες από το SparkContext. Το SparkContext είναι ένα αντικείμενο στο βασικό μας πρόγραμμα (*driver program*). Για την ευκολία της διαχείρισης των εφαρμογών σε μια συστοιχία υπολογιστών η Spark είναι συμβατή με αρκετούς cluster managers, με τους οποίος συνδέεται το SparkContext. Ο βασικότερος cluster manager είναι αυτός που συμπεριλαμβάνεται στην πλατφόρμα και ονομάζεται Standalone Scheduler. Οι άλλοι δύο πιο σημαντικοί cluster managers είναι ο Hadoop YARN και ο Apache Mesos.

3.4 PySpark

Η πλατφόρμα Spark έχει γραφτεί στη γλώσσα προγραμματισμού Scala. Για την υποστήριξη της Python με Spark, η Apache Spark Community δημοσίευσε το API PySpark. Χρησιμοποιώντας το PySpark, μπορούμε να δουλέψουμε με τα RDDs στην Python, χάρη στη βιβλιοθήκη Py4j. Το PySpark προσφέρει ακόμα το PySpark Shell το οποίο συνδέει το Python API με τον πυρήνα της Spark και ενεργοποιεί το SparkContext.

Το SparkContext είναι το πρωταρχικό σημείο σε οποιαδήποτε εφαρμογή της Spark. Όταν τρέχουμε μια τέτοια εφαρμογή, το SparkContext ενεργοποιείται στο driver program, το οποίο έχει τη βασική λειτουργία. Το driver program τότε τρέχει τις διεργασίες μέσα στους επεξεργαστές των υπολογιστικών κόμβων.Το SparkContext χρησιμοποιεί τη βιβλιοθήκη Py4j για την εκκίνηση ενός JVM (*Java Virtual Machine*) και δημιουργεί ένα JavaSparkContext.

Η παραπάνω περιγραφή απεικονίζεται στην ακόλουθη εικόνα.



Σχήμα 3.3: Η λειτουργία του SparkContext.

Κεφάλαιο 4

Εφαρμογές και Συμπεράσματα

4.1 Περιγραφή της Ανάλυσης

4.1.1 Σκοπός της Ανάλυσης

Στο κεφάλαιο αυτό θα γίνει η παρουσίαση της ανάλυσης που εφαρμόστηκε με σκοπό την ομαδοποίηση ενός συνόλου τροχιών, οι οποίες είναι πτήσεις που πραγματοποιήθηκαν μεταξύ Βαρκελώνης και Μαδρίτης. Η ομαδοποίηση αυτή πραγματοποιήθηκε με χρήση των αλγορίθμων ομαδοποίησης K-means, Gaussian Mixture Model, Power Iteration Clustering και Bisecting K-means, οι οποίοι υπάρχουν στη βιβλιοθήκη MLlib της πλατφόρμας Spark. Σημαντικό κομμάτι της ανάλυσης υπήρξε η αξιολόγησή των αλγορίθμων, όπως αυτή προέκυψε από τη σύγκριση των αποτελεσμάτων τους για τις διάφορες μεθόδους διανυσματοποίησης (*vectorization*) που εφαρμόστηκαν σε δύο σύνολα δεδομένων.

Το σύνολο δεδομένων περιέχει όλες τις πτήσεις που έγιναν μεταξύ Βαρκελώνης και Μαδρίτης σε διάστημα ενός μήνα. Τα δεδομένα προέρχονται από τα ραντάρ IFS που παρέχει η CRIDA. Αναλυτικότερα, περιλαμβάνονται 909.644 θέσεις πρωταρχικών σημείων από αεροσκάφη που πετούν μεταξύ των αεροδρομίων της Μαδρίτης (Barajas) και της Βαρκελώνης κατά τη διάρκεια 1-30 Απριλίου 2016.

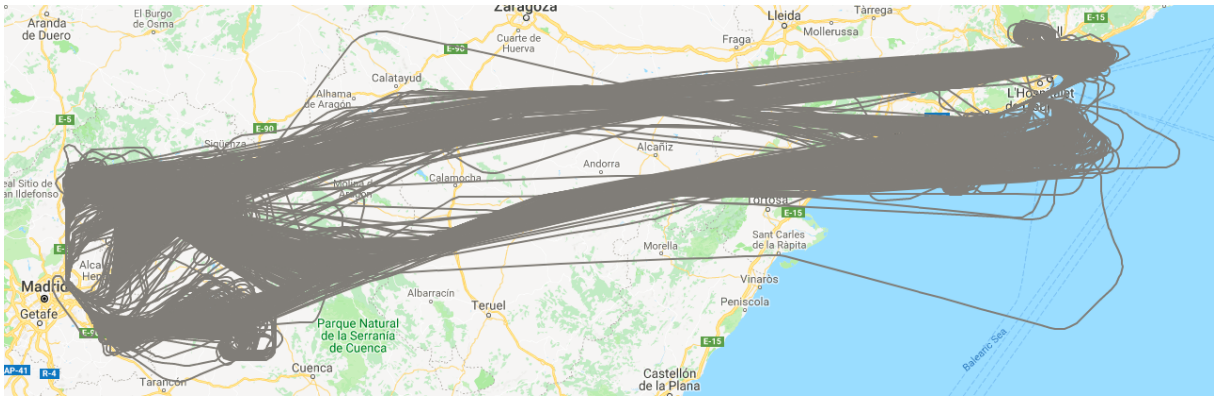
Εκτός από το παραπάνω σύνολο δεδομένων, χρησιμοποιήθηκε και μια σύνοψή του. Για την ανίχνευση των κρίσιμων σημείων, η οποία πραγματοποιήθηκε σύμφωνα με τις προδιαγραφές DatAron Task 2.1 για τη συνοπτική περιγραφή των τροχιών, χρησιμοποιήθηκαν εφαρμογές λογισμικού που υλοποιήθηκαν στη Scala μέσω του Apache Flink 0.10.2 και οι οποίες προσομοιώνουν την εισαγωγή του συνόλου δεδομένων σε χρονολογική σειρά (βλέπε [17]). Συγκεκριμένα ανιχνεύτηκαν 183.372 κρίσιμα σημεία.

Συνοψίζοντας, τα δεδομένα μας είναι ακολουθιακά, καθώς αποτελούν τροχιές κινούμενων αντικειμένων. Όσον αφορά τις συνόψεις, αποτελούν τη μετατροπή των αρχικών ακατέργαστων δεδομένων σε σημασιολογικές τροχιές (βλέπε [18]).

Οι μεταβλητές που δίνονται στα δύο σύνολα δεδομένων είναι:

- id: το κλειδί του συνόλου δεδομένων που υποδεικνύει τις τροχιές
- timestamp: η χρονική στιγμή κατά την οποία πάρθηκαν οι μετρήσεις (epochs δηλαδή το χρονικό διάστημα από 1/1/1970 μέχρι τη στιγμή των μετρήσεων σε milliseconds)
- longitude: γεωγραφικό μήκος
- latitude: γεωγραφικό πλάτος
- altitude: ύψος
- speed: ταχύτητα

Στο Σχήμα 4.1 που ακολουθεί έχουμε απεικονίσει στο χάρτη όλες τις 2D τροχιές των πτήσεων που υπάρχουν στα σύνολα δεδομένων.



Σχήμα 4.1: 2D τροχιές σε χαρτογραφικό υπόβαθρο.

4.1.2 Τεχνικές Μετατροπής των Ακολουθιακών Δεδομένων σε Διανύσματα

Λόγω της φύσης των συνόλων δεδομένων εφαρμόστηκαν τέσσερις διαφορετικές τεχνικές που είχαν σκοπό τη μετατροπή των ακολουθιακών δεδομένων σε διανύσματα και στα δύο σύνολα δεδομένων:

- Διανυσματοποίηση με χρήση παρεμβολής λαμβάνοντας υπόψη το γεωγραφικό μήκος και γεωγραφικό πλάτος των πτήσεων (2D τροχιά). Το μέγεθος διανύσματος ήταν 5, 10, 25 και 50 παρατηρήσεις για καθεμιά από τις δύο προαναφερθείσες μεταβλητές. Στη συνέχεια παρουσιάζουμε τον ψευδοκώδικα για τη δημιουργία των διανυσμάτων από τα ακολουθιακά δεδομένα.

Ψευδοκώδικας διανυσματοποίησης με χρήση παρεμβολής στη 2D τροχιά	
Είσοδος	**1.** Σύνολο ακατέργαστων δεδομένων ή συνοπτικό σύνολο δεδομένων **2.** Μέγεθος διανύσματος: 5, 10, 25, 50
Διαδικασία	**1.** Δημιουργία ενός κενού πίνακα με 1396 γραμμές και 10, 20, 50 ή 100 στήλες **2.** Για κάθε μία από τις τροχιές του συνόλου δεδομένων: **α.** Επιλέγουμε τις γραμμές του συνόλου δεδομένων που αντιστοιχούν στις παρατηρήσεις της τροχιάς αυτής **β.** Εφαρμόζουμε κυβική παρεμβολή μεγέθους 5, 10, 25, 50 στις στήλες 'Γεωγραφικό Μήκος' και 'Γεωγραφικό Πλάτος', με ανεξάρτητη μεταβλητή το timestamp, δημιουργώντας 2 διανύσματα διάστασης ίσης με το μέγεθος του διανύσματος **γ.** Σύνδεση των δύο παραπάνω διανυσμάτων, με σκοπό τη δημιουργία ενός διανύσματος διάστασης 10, 20, 50, 100 αντίστοιχα **δ.** Αποθήκευση του συγχωνευμένου διανύσματος στην αντίστοι χηγραμμή του κενού πίνακα
Έξοδος	Πίνακας με 1396 γραμμές (όσες και οι τροχιές) και στήλες όσες ο διπλάσιος αριθμός του μεγέθους διανύσματος που επιλέξαμε

- Διανυσματοποίηση με χρήση παρεμβολής λαμβάνοντας υπόψη το γεωγραφικό μήκος, το γεωγραφικό πλάτος και το ύψος των πτήσεων (3D τροχιά). Το μέγεθος διανύσματος ήταν 5, 10, 25 και 50 παρατηρήσεις για καθεμιά από τις τρεις προαναφερθείσες μεταβλητές. Στη συνέχεια παρουσιάζουμε τον ψευδοκώδικα για τη δημιουργία των διανυσμάτων από τα ακολουθιακά δεδομένα.

Ψευδοκώδικας διανυσματοποίησης με χρήση παρεμβολής στη 3D τροχιά	
Είσοδος	**1.** Σύνολο ακατέργαστων δεδομένων ή συνοπτικό σύνολο δεδομένων **2.** Μέγεθος διανύσματος (τιμές 5, 10, 25, 50)
Διαδικασία	**1.** Δημιουργία ενός κενού πίνακα με 1396 γραμμές και 15, 30, 75 ή 150 στήλες **2.** Για κάθε μία από τις τροχιές του συνόλου δεδομένων: **α.** Επιλέγουμε τις γραμμές του συνόλου δεδομένων που αντιστοιχούν στις παρατηρήσεις της τροχιάς αυτής **β.** Εφαρμόζουμε κυβική παρεμβολή μεγέθους 5, 10, 25, 50 στις στήλες 'Γεωγραφικό Μήκος' και 'Γεωγραφικό Πλάτος' και 'Υψος', με ανεξάρτητη μεταβλητή το timestamp, δημιουργώντας 3 διανύσματα διάστασης ίσης με το μέγεθος του διανύσματος **γ.** Σύνδεση των δύο παραπάνω διανυσμάτων, με σκοπό τη δημιουργία ενός διανύσματος διάστασης 15, 30, 75, 150 αντίστοιχα **δ.** Αποθήκευση του συγχωνευμένου διανύσματος στην αντίστοιχη γραμμή του κενού πίνακα
Έξοδος	Πίνακας με 1396 γραμμές (όσες και οι τροχιές) και στήλες όσες ο τριπλάσιος αριθμός του μεγέθους διανύσματος που επιλέξαμε

- Διανυσματοποίηση η οποία έλαβε υπόψιν της το γεωγραφικό μήκος, το γεωγραφικό πλάτος και το ύψος, αφού αυτά ευθυγραμμίστηκαν και κανονικοποιήθηκαν. Πιο συγκεκριμένα, αφού κάναμε όλες τις τροχιές να ξεκινούν από τη χρονική στιγμή $t = 0$ (*aligned τροχιές*), χρησιμοποιήσαμε τη διάρκεια της μέγιστης πτήσης με σκοπό να κανονικοποιήσουμε τη μεταβλητή timestamp του συνόλου δεδομένων. Με τον τρόπο αυτό η πτήση με τη μεγαλύτερη διάρκεια έπαιρνε τιμές στο *t* από 0 μέχρι 1, ενώ για όλες τις άλλες πτήσεις η μέγιστη τιμή του *t* ήταν μικρότερη του 1. Το μέγεθος διανύσματος είναι 5, 10, 25 και 50 παρατηρήσεις για καθεμιά από τις μεταβλητές γεωγραφικό μήκος, γεωγραφικό πλάτος και ύψος.

  Για παράδειγμα για μέγεθος διανύσματος ίσο με 5, η διανυσματοποίηση έγινε με βάση το διάνυσμα [0, 0.2, 0.4, 0.6, 0.8, 1]. Για μια τροχιά της οποίας το πρώτο σημείο ήταν το 0 και το τελευταίο το 0.7, κρατήσαμε τις τιμές του γεωγραφικού μήκους, γεωγραφικού πλάτους και ύψους που αντιστοιχούν στις χρονικές στιγμές 0, 0.2, 0.4 και 0.6, ενώ για τις χρονικές στιγμές 0.8 και 1 κρατήσαμε τις τιμές του γεωγραφικού μήκους, γεωγραφικού πλάτους και ύψους που αντιστοιχούν στην τελευταία παρατήρηση της τροχιάς αυτής.

  Ειδικότερα, αν συμβολίσουμε με $t_{i,j}$ το timestamp που αντιστοιχει την $i-$οστή παρατήρηση της τροχιάς j τότε το $t_{i,j}$ μετασχηματίζεται σύμφωνα με τον τύπο:

$$t'_{i,j} = \frac{t_{i,j} - min_j\{t_{i,j}\}}{max\{\Delta t_j\}}$$

  όπου

$$\Delta t_j = max_j\{t_{i,j}\} - min_j\{t_{i,j}\}$$

  Στη συνέχεια παρουσιάζουμε τον ψευδοκώδικα για τη δημιουργία των διανυσμάτων από τα ακολουθιακά δεδομένα.

Ψευδοκώδικας διανυσματοποίησης κανονικοποιημένης και ευθυγραμμισμένης 3D τροχιάς	
Είσοδος	**1.** Σύνολο ακατέργαστων δεδομένων ή συνοπτικό σύνολο δεδομένων **2.** Μέγεθος διανύσματος (τιμές 5, 10, 25, 50)
Διαδικασία	**1.** Δημιουργία ενός κενού πίνακα με 1396 γραμμές και 15 στήλες **2.** Βρίσκουμε τη διάρκεια της μέγιστης πτήσης και τη συμβολίζουμε με MaxDur **3.** Δημιουργία ενός κενού συνόδου δεδομένων (df2) **4.** Για κάθε μία από τις τροχιές του συνόλου δεδομένων: **α.** Επιλέγουμε τις γραμμές του συνόλου δεδομένων που αντιστοιχούν στις παρατηρήσεις της τροχιάς αυτής και τις αποθηκεύουμε σε ένα νέο σύνολο δεδομένων (flight11) **β.** Βρίσκουμε το ελάχιστο timestamp του flight11 και το συμβολίζουμε με mintimestamp0 **γ.** Δημιουργούμε μία νέα στήλη στο flight11 με το όνομα timestamp2 που προκύπτει από το μετασχηματισμό της στήλης timestamp σύμφωνα με τον τύπο (timestamp-mintimestamp0)/MaxDur **δ.** Συγχωνεύουμε το df2 με το flight11 **5.** Δημιουργία τριών κενών πινάκων διαστάσεων 1396*5, με όνομα lon, lat και alt **6.** Για κάθε μία από τις τροχιές του συνόλου δεδομένων df2: **α.** Επιλέγουμε τις γραμμές του συνόλου δεδομένων που αντιστοιχούν στις παρατηρήσεις της τροχιάς αυτής **β.** Βρίσκουμε το μέγιστο timestamp2 και το συμβολίζουμε με maxtimestamp0 **γ.** Δημιουργία τριών κενών διανυσμάτων μεγέθους 5 με ονόματα longit2, latit2 και altit2 **δ.** Για i από 1 έως 5: **i.** Υπολογίζουμε το x=i/5 **ii.** Αν x $\leqslant$ maxtimestamp0: **a.** Βρίσκουμε την τιμή του 'Γεωγραφικού Μήκους', του 'Γεωγραφικού Πλάτους' και του 'Ύψους' που αντιστοιχούν στην παρατήρηση που βρίσκεται πιο κοντά στη χρονική στιγμή timestamp2= i/5 **b.** Εκχώρηση των τιμών αυτών στις i-οστές θέσεις των διανυσμάτων longit2, latit2 και altit2 αντίστοιχα Αλλιώς: **a.** Βρίσκουμε την τιμή του 'Γεωγραφικού Μήκους', του 'Γεωγραφικού Πλάτους' και του 'Ύψους' που αντιστοιχούν στην παρατήρηση με το μέγιστο timestamp2 **b.** Εκχώρηση των τιμών αυτών στις i-οστές θέσεις των διανυσμάτων longit2, latit2 και altit2 αντίστοιχα **ε.** Αποθήκευση των longit2, latit2 και altit2 στην αντίστοιχη γραμμή των κενών πινάκων lon, lat και alt **7.** Συγχώνευση κατά στήλες των πινάκων lon, lat και alt
Έξοδος	Πίνακας με 1396 γραμμές (όσες και οι τροχιές) και στήλες όσες ο τριπλάσιος αριθμός του μεγέθους διανύσματος που επιλέξαμε

- Διανυσματοποίηση που προέκυψε από το συνδυασμό της παραπάνω διανυσματοποίησης και της Ανάλυσης Κυρίων Συνιστωσών. Πιο συγκεκριμένα, χρησιμοποιώντας το σύνολο δεδομένων που προέκυψε από την προηγούμενη διανυσματοποίηση για μέγεθος διανύσματος ίσο με 50, εφαρμόσαμε τη μέθοδο της Ανάλυσης Κυρίων Συνιστωσών με σκοπό να καταλήξουμε από τις 50 παρατηρήσεις για κάθε μία από τις μεταβλητές γεωγραφικό μήκος, γεωγραφικό πλάτος και ύψος, σε 5 και 10. Στη συνέχεια παρουσιάζουμε τον ψευδοκώδικα για τη δημιουργία των διανυσμάτων βιασμένοι στη προηγούμενη διανυσματοποίηση.

Ψευδοκώδικας διανυσματοποίησης με χρήση Ανάλυσης Κυρίων Συνιστωσών στη 3D τροχιά	
Είσοδος	**1.** Πίνακας που προέκυψε από την τελευταία διανυσματοποίηση στο σύνολο ακατέργαστων δεδομένων για μέγεθος διανύσματος 50 ή Πίνακας που προέκυψε από την τελευταία διανυσματοποίηση στο συνοπτικό σύνολο δεδομένων για μέγεθος διανύσματος 50 **2.** Μέγεθος διανύσματος (τιμές 5, 10)
Διαδικασία	**1.** Διάσπαση του πίνακα εισόδου (στον οποίο οι πρώτες 50 στήλες αποτελούν το 'Γεωγραφικό Μήκος', οι 50 επόμενες το 'Γεωγραφικό Πλάτος' και οι 50 τελευταίες το 'Υψος') σε τρία σύνολα δεδομένων, τα Lon, Lat και Alt **2.** Εφαρμογή της Ανάλυσης Κυρίων Συνιστωσών σε καθένα από τα Lon, Lat και Alt με πλήθος 5 και 10 κυρίων συνιστωσών και αποθήκευση των νέων δεδομένων στους πίνακες Lon2, Lat2 και Alt2 **3.** Συγχώνευση κατά στήλες των Lon2, Lat2 και Alt2
Έξοδος	Πίνακας με 1396 γραμμές (όσες και οι τροχιές) και στήλες όσες ο τριπλάσιος αριθμός του πλήθους κυρίων συνιστωσών που επιλέξαμε

Πριν συνεχίσουμε με την υπόλοιπη ανάλυση κρίνεται σκόπιμο να αναφέρουμε αναλυτικότερα τι είναι η παρεμβολή και η Ανάλυση Κυρίων Συνιστωσών.

Παρεμβολή

Παρεμβολή απλά σημαίνει προσαρμογή κάποιας συνάρτησης σε δεδομένα, έτσι ώστε η συνάρτηση να έχει τις ίδιες τιμές με τις τιμές των δεδομένων.

Γενικά το απλούστερο πρόβλημα παρεμβολής σε μία διάσταση μπορεί να διατυπωθεί ως εξής:

Για δοθέντα δεδομένα $(x_i, f_i), i = 1, 2, ..., n$, με $x_1 < x_2 < ... < x_n$, ζητάμε να υπολογίσουμε μία συνάρτηση $y(x)$ έτσι ώστε: $y(x_i) = f_i, i = 1, 2, ..., n$, και θα ονομάζουμε την $y(x)$, παρεμβολική συνάρτηση για τα δοθέντα δεδομένα. Για μεγαλύτερη ακρίβεια επιλέξαμε το πολυώνυμο παρεμβολής να είναι μέχρι τρίτου βαθμού. Για περισσότερες πληροφορίες σχετικά με την παρεμβολή βλέπε [2]

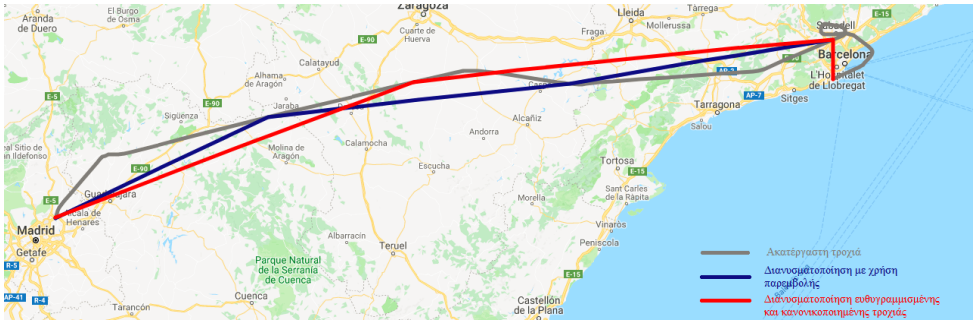
Για την εφαρμογή της παρεμβολής στα σύνολα των δεδομένων χρησιμοποιήθηκε το πακέτο *interpolate* της open-source βιβλιοθήκης *SciPy* της Python.

Ανάλυση Κυρίων Συνιστωσών

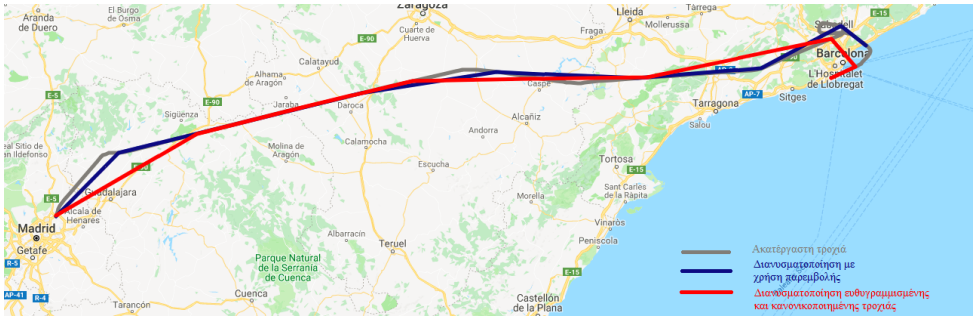
Η ανάλυση κύριων συνιστωσών είναι μία στατιστική διαδικασία η οποία μετατρέπει μία ομάδα τιμών (παρατηρήσεων) δυνητικά συσχετιζόμενων μεταβλητών σε μία ομάδα νέων τιμών μη γραμμικά συσχετιζόμενων μεταβλητών οι οποίες καλούνται κύριες συνιστώσες. Ο αριθμός των νέων μεταβλητών που προκύπτει είναι ίσος ή και συχνότερα πολύ μικρότερος από τον αριθμό των αρχικών μεταβλητών. Η μετάβαση αυτή πραγματοποιείται με τέτοιο τρόπο ώστε, η πρώτη συνιστώσα να εξηγεί τη μέγιστη δυνατή διακύμανση που αναπτύσσεται μεταξύ των αρχικών μεταβλητών, η δεύτερη, μη συσχετιζόμενη με την πρώτη, να εξηγεί ένα σημαντικό μέρος αυτής αλλά πάντα μικρότερο της πρώτης κοκ (βλέπε [24]).

Για την εφαρμογή της Ανάλυσης Κυρίων Συνιστωσών στα σύνολα των δεδομένων χρησιμοποιήθηκε η βιβλιοθήκη MLlib.

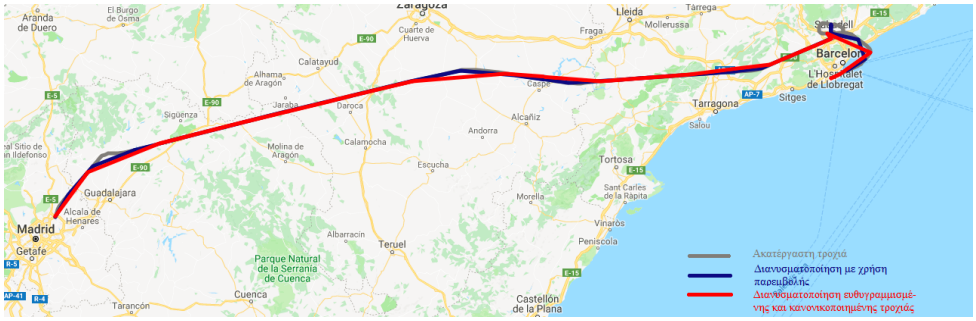
Για την καλύτερη κατανόηση, στη συνέχεια παρουσιάζουμε οπτικά τις διανυσματοποιήσεις μιας τυχαίας τροχιάς από το σύνολο των δεδομένων μαζί με την αντίστοιχη ακατέργαστη τροχιά. Ειδικότερα, απεικονίζουμε σε χαρτογραφικό υπόβαθρο την τροχιά μιας συγκεκριμένης πτήσης και τη συγκρίνουμε με την αντίστοιχη διανυσματοποιημένη τροχιά, όπως αυτή προέκυψε για μεγέθη διανύσματος 5, 10, 25 και 50.



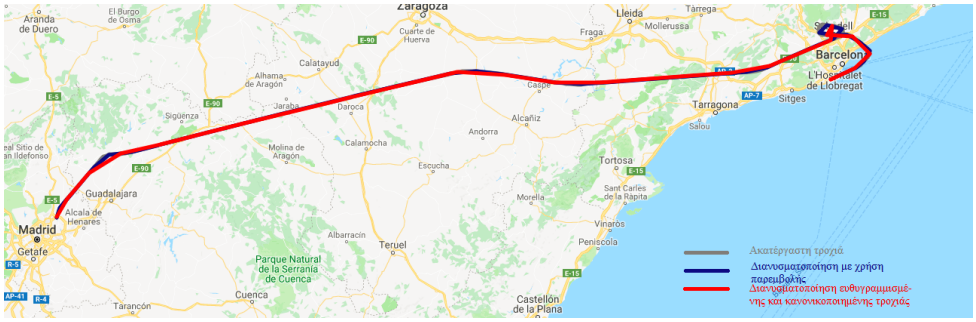
Σχήμα 4.2: Διανυσματοποίηση τροχιάς μεγέθους 5.



Σχήμα 4.3: Διανυσματοποίηση τροχιάς μεγέθους 10.



Σχήμα 4.4: Διανυσματοποίηση τροχιάς μεγέθους 25.



Σχήμα 4.5: Διανυσματοποίηση τροχιάς μεγέθους 50.

Στα παραπάνω σχήματα, παρατηρούμε πως καθώς αυξάνεται το μέγεθος του διανύσματος, οι διανυσματοποιημένες τροχιές τείνουν να συμπέσουν στην ακατέργαστη τροχιά. Αξίζει να σημειώσουμε ότι

οι διανυσματοποιημένες τροχιές που προέκυψαν από την εφαρμογή της παρεμβολής στη 2D και 3D τροχιά ταυτίζονται στη συγκεκριμένη απεικόνιση.

4.1.3 Αλγόριθμοι Ομαδοποίησης

Οι αλγόριθμοι K-means, Gaussian Mixture Model, Power Iteration Clustering και Bisecting K-means εκτελέστηκαν για όλες τις τεχνικές διανυσματοποίησης που περιγράψαμε στην προηγούμενη παράγραφο και για κάθε μέγεθος διανύσματος όπως αναφέρθηκε στην εκάστοτε περιγραφή. Οι αλγόριθμοι εκτελέστηκαν για διάφορες τιμές των παραμέτρων τους μέσω της πλατφόρμας Spark με χρήση της βιβλιοθήκης MLlib μέσω του PySpark. Οι παράμετροι των αλγορίθμων που αξιολογήθηκαν είναι:

1. Το πλήθος των ομάδων (k). Σημειώνουμε ότι είναι πιθανό να επιστραφούν λιγότερες ομάδες από τον επιθυμητό αυτό αριθμό.

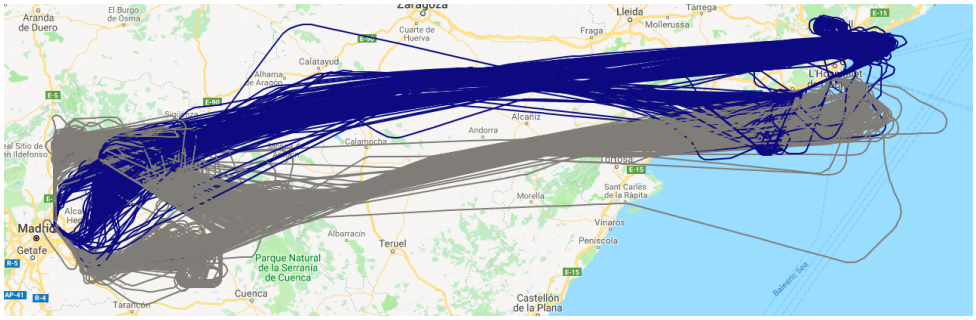
2. Το μέγιστο πλήθος των επαναλήψεων (*MaxIterations*) είναι το μέγιστο πλήθος επαναλήψεων που μπορεί ο αλγόριθμος να τρέξει.

Ειδικότερα, κάθε αλγόριθμος εκτελέστηκε για 2,3,4 και 5 ομάδες και 10,50 και 100 επαναλήψεις.

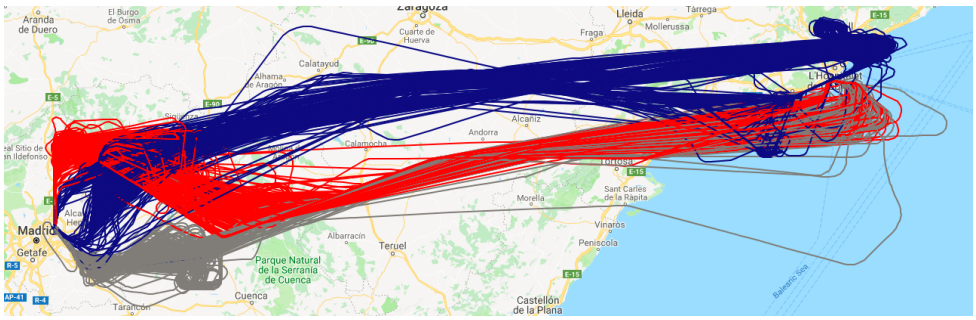
4.1.4 Αξιολόγηση Αλγορίθμων Ομαδοποίησης

Επόμενο βήμα μετά την εφαρμογή των αλγορίθμων ήταν η αξιολόγησή τους, η οποία έγινε με χρήση κάποιων μέτρων απόδοσης. Τα μέτρα που χρησιμοποιήθηκαν είναι τα Accuracy, Precision και F1 Score και σκοπός τους ήταν να αξιολογήσουν την απόδοση του κάθε μοντέλου και την ακρίβεια των προβλέψεων.

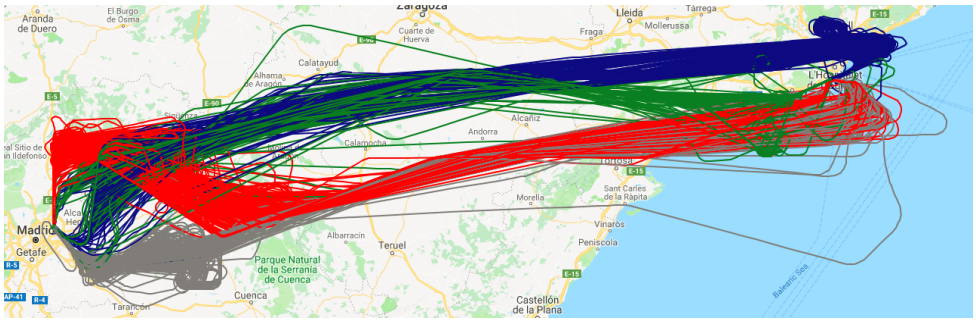
Για την εφαρμογή της αξιολόγησης, έπρεπε να γνωρίζουμε την ομάδα στην οποία ανήκει η κάθε πτήση στην πραγματικότητα, αν αυτές χωρίζονταν σε 2,3,4 και 5 ομάδες. Για το λόγο αυτό, απεικονίσαμε τη 2D τροχιά κάθε μιας πτήσης (γεωγραφικό μήκος και γεωγραφικό πλάτος) και τις χωρίσαμε σε 2,3,4 και 5 ομάδες, δημιουργώντας έτσι το ground truth. Στο σχήματα που ακολουθούν έχουμε απεικονίσει στο χάρτη όλες τις 2D τροχιές των πτήσεων που υπάρχουν στα σύνολα δεδομένων, στα οποία διακρίνεται η ομαδοποίηση των τροχιών βάση του ground truth για κάθε δυνατό αριθμό ομάδων.



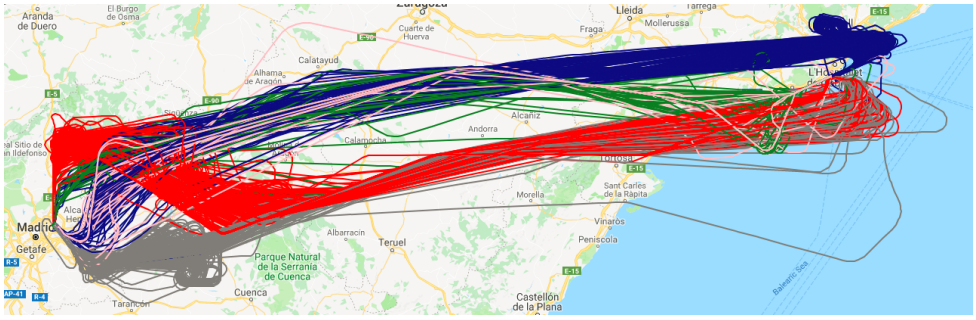
Σχήμα 4.6: Ground truth για 2 ομάδες.



Σχήμα 4.7: Ground truth για 3 ομάδες.



Σχήμα 4.8: Ground truth για 4 ομάδες.



Σχήμα 4.9: Ground truth για 5 ομάδες.

Για την περιγραφή των μέτρων απόδοσης που αναφέρθηκαν παραπάνω, θεωρούμε έναν πίνακα σύγχυσης (*confusion matrix*), ο οποίος χρησιμοποιείται συχνά για να περιγράψει την απόδοση ενός μοντέλου ομαδοποίησης για δύο ομάδες, σε ένα σύνολο δεδομένων για τα οποία οι πραγματικές τιμές είναι γνωστές. Η μορφή του πίνακα αυτού φαίνεται στον Πίνακα 4.1

		Προβλεπόμενη Τιμή	
Πραγματική Τιμή		Τιμή = ΝΑΙ	Τιμή = ΟΧΙ
	Τιμή = ΝΑΙ	Σωστό Θετικό	Λανθασμένο Αρνητικό
	Τιμή = ΟΧΙ	Λανθασμένο Θετικό	Σωστό Αρνητικό

Πίνακας 4.1: Πίνακας Σύγχυσης.

Οι τιμές Σωστό Θετικό και Σωστό Αρνητικό είναι οι παρατηρήσεις που προβλέφθηκαν σωστά, ενώ οι τιμές Λανθασμένο Θετικό και Λανθασμένο Αρνητικό είναι οι παρατηρήσεις που προβλέφθηκαν λάθος, για αυτό κι επιθυμούμε να είναι οι ελάχιστες δυνατές. Πιο ειδικά, θεωρώντας ότι η Πραγματική Τιμή = ΝΑΙ υποδεικνύει ότι μια πτήση ανήκει στην ομάδα 0, και η Πραγματική Τιμή = ΟΧΙ υποδεικνύει ότι μια πτήση ανήκει στην ομάδα 1, τότε:

- Σωστό Θετικό (True Positive - TP): είναι οι πτήσεις που η πρόβλεψη τις κατατάσσει στην ομάδα 0, στην οποία ανήκουν στην πραγματικότητα.
- Σωστό Αρνητικό (True Negative - TN): είναι οι πτήσεις που η πρόβλεψη τις κατατάσσει στην ομάδα 1, στην οποία ανήκουν στην πραγματικότητα.
- Λανθασμένο Θετικό (False Positive - FP): είναι οι πτήσεις που η πρόβλεψη τις κατατάσσει στην ομάδα 0, αλλά στην πραγματικότητα ανήκουν στην ομάδα 1.
- Λανθασμένο Αρνητικό (False Negative - FN): είναι οι πτήσεις που η πρόβλεψη τις κατατάσσει στην ομάδα 1, αλλά στην πραγματικότητα ανήκουν στην ομάδα 0.

Η τιμή Accuracy δίνεται από τον τύπο:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{4.1}$$

Αποτελεί το πιο διαισθητικό μέτρο απόδοσης αφού ορίζεται απλώς ως το πηλίκο των σωστά προβλεφθέντων παρατηρήσεων προς το σύνολο των παρατηρήσεων. Η ερώτηση στην οποία απαντά αυτό το μέτρο είναι: "από όλες τις πτήσεις, πόσες προβλέφθηκαν να είναι στην ομάδα 0;".

Η τιμή Precision δίνεται από τον τύπο:

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

Πρόκειται για το πηλίκο των σωστά προβλεφθέντων παρατηρήσεων με τιμή = NAI προς το σύνολο των σωστά προβλεφθέντων παρατηρήσεων. Η ερώτηση στην οποία απαντά αυτό το μέτρο είναι: "από όλες τις πτήσεις που προβλέφθηκαν να είναι στην ομάδα 0, πόσες στην πραγματικότητα είναι;".

Η τιμή F1 Score δίνεται από τον τύπο:

$$F1Score = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{4.3}$$

όπου το Recall ορίζεται ως:

$$Recall = \frac{TP}{TP + FN} \tag{4.4}$$

Το Recall υποδεικνύει από όλες τις πτήσεις που στην πραγματικότητα ανήκουν στην ομάδα 0, πόσες προβλέφθηκαν να είναι στην ομάδα αυτή. Το F1 Score είναι ένας σταθμισμένος μέσος των μέτρων Recall και Precision και για αυτό δεν είναι εύκολο να ερμηνευτεί. Η σημασία του όμως είναι μεγάλη, ιδιαίτερα εάν η κατανομή των παρατηρήσεων μας είναι άνιση.

Εκτός από τα μέτρα απόδοσης που περιγράφηκαν παραπάνω, υπολογίστηκε επίσης η τετραγωνική ρίζα της μέσης τετραγωνικής απόκλισης, RMSE, κάθε κεντροειδούς από τα μέλη της αντίστοιχης ομάδας. Τέλος, για όλες τις εκτελέσεις των αλγορίθμων σημειώθηκε ο χρόνος εκτέλεσης.

4.2 Αποτελέσματα Ανάλυσης

Στην παράγραφο αυτή θα αναφερθούμε αναλυτικά στα αποτελέσματα της κάθε ανάλυσης όπως αυτά προέκυψαν από τις τέσσερις τεχνικές διανυσματοποίησης που περιγράψαμε προηγουμένως.

4.2.1 Διανυσματοποίηση σε 2D Τροχιά με Χρήση Παρεμβολής

Όπως αναφέραμε παραπάνω, η διανυσματοποίηση αυτή έγινε στα δύο σύνολα δεδομένων με χρήση της παρεμβολής. Για κάθε μία από τις μεταβλητές γεωγραφικό μήκος και γεωγραφικό πλάτος εφαρμόσαμε παρεμβολή 5, 10, 25 και 50 σημείων.

Η είσοδος που δέχτηκαν οι αλγόριθμοι K-means, Gaussian Mixture Model και Bisecting K-means, είναι ένας πίνακας με τόσες γραμμές όσες και οι πτήσεις του συνόλου δεδομένων και τόσες στήλες όσες ο διπλάσιος αριθμός του μεγέθους του διανύσματος που επιλέξαμε. Για παράδειγμα στην περίπτωση που επιλέξαμε 5 σημεία παρεμβολής, τότε για την κάθε πτήση κρατήσαμε 10 σημεία, όπου τα 5 πρώτα αποτελούν το γεωγραφικό μήκος και τα 5 τελευταία το γεωγραφικό πλάτος (όπως αυτά προέκυψαν από την εφαρμογή της παρεμβολής).

Η είσοδος που δέχτηκε ο αλγόριθμος Power Iteration Clustering είναι ένας πίνακας με 3 στήλες, όπου η πρώτη στήλη υποδεικνύει την πτήση i, η δεύτερη στήλη υποδεικνύει την πτήση j και η τρίτη στήλη υποδεικνύει την απόσταση των προαναφερθέντων πτήσεων (με $i \leq j$). Σημειώνουμε ότι η απόσταση αυτή υπολογίστηκε με χρήση της συνάρτησης $gpxpy.geo.haversine\_distance$ στην Python, η οποία δέχεται το γεωγραφικό μήκος και γεωγραφικό πλάτος δύο σημείων στο χάρτη και δίνει τη χιλιομετρική τους απόσταση.

Για τους κώδικες που χρησιμοποιήθηκαν για τη δημιουργία των συνόλων δεδομένων που δέχονται σαν είσοδο οι αλγόριθμοι ομαδοποίησης, παραπέμπουμε στο Παράρτημα Α'.

Στη συνέχεια τρέξαμε τους αλγορίθμους ομαδοποίησης για όλες τις περιπτώσεις του μεγέθους του διανύσματος. Για τους κώδικες των αλγορίθμων παραπέμπουμε στο Παράρτημα Ε'.

Τα αποτελέσματα της αξιολόγησης των αλγορίθμων για καθεμία από τις παραμέτρους για τις οποίες εξετάστηκαν (πλήθος ομάδων *k* και μέγιστο πλήθος επαναλήψεων *MaxIterations*) είναι συγκεντρωμένα στους πίνακες που ακολουθούν.

Οι πίνακες που δίνονται στη συνέχεια περιλαμβάνουν τις τιμές των Accuracy, Precision και F1 Score που αξιολογούν το αποτέλεσμα του κάθε αλγορίθμου που έτρεξε σε ολόκληρο το σύνολο δεδομένων, σε σχέση με το ground truth.

Sampling = 5 longitudes, latitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.999	0.999	0.999	0.998	0.996	0.998	0.999	0.999	0.999	0.999	0.999	0.999
	Precision	0.997	0.997	0.997	0.996	0.997	0.996	0.997	0.997	0.997	0.997	0.997	0.997
	F1-Score	0.999	0.999	0.999	0.998	0.996	0.998	0.999	0.999	0.999	0.999	0.999	0.999
k=3	Accuracy	0.547	0.531	0.531	0.722	0.718	0.718	0.771	0.771	0.771	0.529	0.531	0.531
	Precision	0.518	0.518	0.518	0.524	0.526	0.526	0.671	0.672	0.671	0.518	0.518	0.518
	F1-Score	0.471	0.462	0.462	0.553	0.552	0.552	0.642	0.643	0.642	0.461	0.462	0.462
k=4	Accuracy	0.600	0.784	0.723	0.693	0.606	0.787	0.638	0.637	0.638	0.785	0.784	0.784
	Precision	0.570	0.786	0.614	0.605	0.418	0.731	0.598	0.597	0.598	0.787	0.786	0.786
	F1-Score	0.552	0.777	0.611	0.592	0.423	0.707	0.548	0.547	0.548	0.779	0.777	0.777
k=5	Accuracy	0.547	0.684	0.576	0.666	0.791	0.532	0.638	0.637	0.638	0.686	0.684	0.684
	Precision	0.430	0.603	0.473	0.531	0.623	0.330	0.551	0.550	0.551	0.601	0.600	0.600
	F1-Score	0.419	0.572	0.462	0.511	0.591	0.339	0.544	0.543	0.544	0.572	0.571	0.571

Sampling = 10 longitudes, latitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.999	0.999	0.999	0.998	0.994	0.994	0.999	0.999	0.999	0.999	0.999	0.999
	Precision	0.997	0.997	0.997	0.996	0.997	0.997	0.997	0.997	0.997	0.997	0.997	0.997
	F1-Score	0.999	0.999	0.999	0.998	0.994	0.994	0.999	0.999	0.999	0.999	0.999	0.999
k=3	Accuracy	0.949	0.949	0.949	0.760	0.700	0.696	0.768	0.675	0.768	0.546	0.546	0.508
	Precision	0.931	0.931	0.931	0.564	0.523	0.523	0.664	0.518	0.664	0.518	0.518	0.518
	F1-Score	0.931	0.931	0.931	0.573	0.545	0.543	0.638	0.532	0.638	0.471	0.471	0.448
k=4	Accuracy	0.735	0.735	0.812	0.630	0.638	0.623	0.640	0.662	0.640	0.812	0.812	0.812
	Precision	0.640	0.663	0.805	0.575	0.565	0.489	0.602	0.627	0.602	0.805	0.805	0.805
	F1-Score	0.629	0.625	0.804	0.491	0.556	0.410	0.539	0.587	0.539	0.804	0.804	0.804
k=5	Accuracy	0.559	0.579	0.766	0.562	0.676	0.527	0.633	0.648	0.633	0.701	0.701	0.701
	Precision	0.460	0.539	0.700	0.557	0.532	0.387	0.492	0.560	0.492	0.626	0.626	0.626
	F1-Score	0.442	0.498	0.672	0.452	0.501	0.346	0.460	0.548	0.460	0.589	0.589	0.589

Sampling = 25 longitudes, latitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.999	0.999	0.999	0.706	0.537	0.757	0.999	0.999	0.999	0.999	0.999	0.999
	Precision	0.997	0.997	0.997	0.801	0.559	0.689	0.997	0.997	0.997	0.997	0.997	0.997
	F1-Score	0.999	0.999	0.999	0.646	0.391	0.791	0.999	0.999	0.999	0.999	0.999	0.999
k=3	Accuracy	0.529	0.544	0.953	0.754	0.590	0.710	0.698	0.698	0.698	0.546	0.546	0.546
	Precision	0.518	0.518	0.953	0.649	0.637	0.601	0.518	0.518	0.518	0.518	0.518	0.518
	F1-Score	0.461	0.470	0.937	0.630	0.572	0.598	0.542	0.542	0.542	0.471	0.471	0.471
k=4	Accuracy	0.628	0.817	0.814	0.626	0.735	0.621	0.640	0.646	0.640	0.816	0.816	0.816
	Precision	0.624	0.810	0.808	0.606	0.669	0.524	0.603	0.617	0.603	0.809	0.809	0.809
	F1-Score	0.594	0.808	0.806	0.567	0.652	0.463	0.541	0.563	0.541	0.808	0.808	0.808
k=5	Accuracy	0.670	0.562	0.718	0.718	0.590	0.640	0.633	0.633	0.633	0.704	0.704	0.704
	Precision	0.534	0.458	0.642	0.553	0.433	0.445	0.495	0.494	0.495	0.631	0.631	0.631
	F1-Score	0.473	0.443	0.605	0.534	0.411	0.422	0.462	0.461	0.462	0.593	0.593	0.593

Sampling = 50 longitudes, latitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.999	0.999	0.999	0.594	0.699	0.563	0.999	0.999	0.999	0.999	0.999	0.999
	Precision	0.997	0.997	0.997	0.746	0.827	0.635	0.997	0.997	0.997	0.997	0.997	0.997
	F1-Score	0.999	0.999	0.999	0.399	0.620	0.384	0.999	0.999	0.999	0.999	0.999	0.999
k=3	Accuracy	0.527	0.953	0.546	0.648	0.552	0.535	0.697	0.697	0.697	0.544	0.544	0.544
	Precision	0.518	0.937	0.518	0.517	0.513	0.445	0.518	0.518	0.518	0.518	0.518	0.518
	F1-Score	0.460	0.937	0.471	0.548	0.480	0.444	0.541	0.541	0.541	0.469	0.469	0.469
k=4	Accuracy	0.736	0.816	0.817	0.462	0.508	0.376	0.642	0.648	0.642	0.815	0.815	0.815
	Precision	0.653	0.810	0.810	0.417	0.439	0.355	0.605	0.618	0.605	0.809	0.809	0.809
	F1-Score	0.634	0.808	0.808	0.389	0.432	0.292	0.545	0.567	0.545	0.807	0.807	0.807
k=5	Accuracy	0.772	0.761	0.715	0.367	0.323	0.706	0.635	0.635	0.635	0.703	0.703	0.703
	Precision	0.691	0.693	0.640	0.303	0.298	0.642	0.498	0.498	0.498	0.631	0.631	0.631
	F1-Score	0.671	0.666	0.602	0.236	0.198	0.620	0.466	0.465	0.466	0.592	0.592	0.592

Πίνακας 4.2: Accuracy, Precision και F1 Score(%) για Διανυσματοποίηση σε 2D Τροχιά με Χρήση Παρεμβολής.

Οι ίδιοι πίνακες, που δίνουν τα αποτελέσματα αξιολόγησης των αλγορίθμων όπως αυτά έτρεξαν για το συνοπτικό σύνολο δεδομένων, δίνονται στη συνέχεια.

Sampling = 5 longitudes, latitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.999	0.999	0.999	0.999	0.996	0.999	0.999	0.999	0.999	0.999	0.999	0.999
	Precision	0.997	0.997	0.997	0.997	0.997	0.997	0.997	0.997	0.997	0.997	0.997	0.997
	F1-Score	0.999	0.999	0.999	0.999	0.996	0.999	0.999	0.999	0.999	0.999	0.999	0.999
k=3	Accuracy	0.779	0.534	0.534	0.762	0.719	0.719	0.772	0.775	0.772	0.532	0.532	0.532
	Precision	0.852	0.518	0.518	0.541	0.524	0.524	0.674	0.685	0.674	0.518	0.518	0.518
	F1-Score	0.573	0.464	0.464	0.567	0.552	0.552	0.643	0.670	0.643	0.462	0.462	0.462
k=4	Accuracy	0.605	0.795	0.783	0.614	0.612	0.667	0.635	0.638	0.635	0.783	0.783	0.783
	Precision	0.573	0.620	0.786	0.448	0.459	0.593	0.589	0.597	0.589	0.786	0.786	0.786
	F1-Score	0.555	0.651	0.777	0.447	0.462	0.569	0.526	0.545	0.526	0.777	0.777	0.777
k=5	Accuracy	0.739	0.670	0.681	0.595	0.570	0.583	0.626	0.626	0.626	0.703	0.681	0.681
	Precision	0.666	0.595	0.530	0.424	0.343	0.439	0.487	0.487	0.487	0.631	0.600	0.600
	F1-Score	0.641	0.562	0.479	0.411	0.359	0.416	0.464	0.463	0.464	0.592	0.569	0.569

Sampling = 10 longitudes, latitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.999	0.999	0.999	0.776	0.994	0.994	0.999	0.999	0.999	0.999	0.999	0.999
	Precision	0.997	0.997	0.997	0.807	0.997	0.997	0.997	0.997	0.997	0.997	0.997	0.997
	F1-Score	0.999	0.999	0.999	0.761	0.994	0.994	0.999	0.999	0.999	0.999	0.999	0.999
k=3	Accuracy	0.951	0.951	0.951	0.722	0.713	0.708	0.696	0.659	0.696	0.545	0.545	0.545
	Precision	0.934	0.934	0.934	0.578	0.561	0.547	0.518	0.518	0.518	0.518	0.518	0.518
	F1-Score	0.934	0.934	0.934	0.580	0.569	0.560	0.541	0.526	0.541	0.470	0.470	0.470
k=4	Accuracy	0.807	0.623	0.736	0.601	0.594	0.643	0.640	0.661	0.640	0.814	0.814	0.814
	Precision	0.803	0.619	0.640	0.451	0.452	0.601	0.603	0.632	0.603	0.807	0.807	0.807
	F1-Score	0.800	0.588	0.630	0.439	0.436	0.518	0.541	0.602	0.541	0.805	0.805	0.805
k=5	Accuracy	0.763	0.557	0.766	0.615	0.792	0.667	0.633	0.643	0.633	0.706	0.706	0.706
	Precision	0.695	0.460	0.696	0.456	0.593	0.553	0.491	0.558	0.491	0.620	0.620	0.620
	F1-Score	0.668	0.441	0.671	0.368	0.591	0.460	0.460	0.552	0.460	0.590	0.590	0.590

Sampling = 25 longitudes, latitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.999	0.999	0.999	0.991	0.691	0.839	0.999	0.999	0.999	0.999	0.999	0.999
	Precision	0.997	0.997	0.997	0.997	0.790	0.915	0.997	0.997	0.997	0.997	0.997	0.997
	F1-Score	0.999	0.999	0.999	0.991	0.622	0.820	0.999	0.999	0.999	0.999	0.999	0.999
k=3	Accuracy	0.532	0.543	0.954	0.724	0.736	0.633	0.696	0.696	0.696	0.545	0.545	0.545
	Precision	0.518	0.518	0.938	0.578	0.619	0.603	0.518	0.518	0.518	0.518	0.518	0.518
	F1-Score	0.463	0.469	0.938	0.580	0.610	0.587	0.541	0.541	0.541	0.470	0.470	0.470
k=4	Accuracy	0.815	0.815	0.814	0.475	0.633	0.793	0.645	0.644	0.645	0.814	0.814	0.814
	Precision	0.808	0.809	0.807	0.467	0.512	0.722	0.616	0.614	0.616	0.807	0.807	0.807
	F1-Score	0.807	0.807	0.805	0.431	0.491	0.709	0.565	0.563	0.565	0.805	0.805	0.805
k=5	Accuracy	0.586	0.764	0.673	0.646	0.620	0.606	0.633	0.634	0.633	0.703	0.703	0.703
	Precision	0.535	0.694	0.534	0.485	0.483	0.405	0.493	0.493	0.493	0.628	0.628	0.628
	F1-Score	0.505	0.668	0.475	0.394	0.448	0.371	0.462	0.463	0.462	0.591	0.591	0.591

Sampling = 50 longitudes, latitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.999	0.999	0.999	0.543	0.548	0.650	0.999	0.999	0.999	0.999	0.999	0.999
	Precision	0.997	0.997	0.997	0.663	0.528	0.747	0.997	0.997	0.997	0.997	0.997	0.997
	F1-Score	0.999	0.999	0.999	0.253	0.642	0.557	0.999	0.999	0.999	0.999	0.999	0.999
k=3	Accuracy	0.952	0.952	0.546	0.446	0.431	0.574	0.696	0.696	0.696	0.543	0.543	0.543
	Precision	0.935	0.935	0.518	0.360	0.339	0.526	0.518	0.518	0.518	0.518	0.518	0.518
	F1-Score	0.935	0.935	0.471	0.334	0.329	0.482	0.541	0.541	0.541	0.469	0.469	0.469
k=4	Accuracy	0.786	0.737	0.622	0.540	0.558	0.337	0.646	0.645	0.646	0.814	0.814	0.814
	Precision	0.794	0.653	0.617	0.460	0.437	0.294	0.617	0.615	0.617	0.808	0.808	0.808
	F1-Score	0.781	0.635	0.587	0.410	0.433	0.249	0.566	0.564	0.566	0.806	0.806	0.806
k=5	Accuracy	0.673	0.764	0.671	0.307	0.513	0.382	0.635	0.636	0.635	0.703	0.703	0.703
	Precision	0.534	0.693	0.533	0.248	0.336	0.296	0.498	0.499	0.498	0.628	0.628	0.628
	F1-Score	0.476	0.667	0.473	0.221	0.337	0.239	0.465	0.464	0.465	0.591	0.591	0.591

Πίνακας 4.3: Accuracy, Precision και F1 Score(%) για Διανυσματοποίηση σε 2D Τροχιά με Χρήση Παρεμβολής, του συνοπτικού συνόλου δεδομένων.

Παρατηρούμε ότι και για τα δύο σύνολα δεδομένων, η καλύτερη ομαδοποίηση γίνεται στην περίπτωση των 2 ομάδων. Συγκεκριμένα, καθώς το πλήθος των ομάδων αυξάνεται οι τιμές των Accuracy, Precision και F1-Score μειώνονται αισθητά. Ακόμη οι αλγόριθμοι που φαίνεται να έχουν την καλύτερη απόδοση για την περίπτωση των 2 ομάδων είναι οι K-means, Power Itration Clustering και Bisecting K-means, αφού η απόδοση του Gaussian Mixture Model φαίνεται να μειώνεται καθώς αυξάνεται το μέγεθος του διανύσματος.

Στη συνέχεια δίνονται οι πίνακες που περιλαμβάνουν τη μέση απόσταση του κεντροειδούς από τα μέλη της ομάδας, όπως αυτή υπολογίστηκε για κάθε αλγόριθμο που έτρεξε στο συνολικό σύνολο δεδομένων.

Sampling = 5 longitudes, latitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	14.289	14.289	14.289	14.609	14.609	14.609	14.289	14.289	14.289	14.289	14.289	14.289
	Cluster 2	16.033	16.033	16.033	16.008	16.008	16.008	16.033	16.033	16.033	16.033	16.033	16.033
k=3	Cluster 1	16.614	17.036	17.036	13.874	13.829	13.829	341.206	341.224	341.206	17.115	17.036	17.036
	Cluster 2	15.118	15.118	15.118	15.172	15.314	15.314	340.013	340.013	340.013	15.118	15.118	15.118
	Cluster 3	349.976	349.586	349.586	300.476	261.412	261.412	17.290	17.283	17.290	349.520	349.586	349.586
k=4	Cluster 1	11.362	17.819	11.886	10.981	11.759	39.620	344.737	344.699	344.737	17.951	17.819	17.819
	Cluster 2	15.170	27.645	11.057	336.485	15.345	10.701	338.872	338.872	338.872	10.858	10.858	10.858
	Cluster 3	352.157	26.624	11.143	336.524	119.503	344.371	344.491	344.491	344.491	349.648	349.713	349.713
	Cluster 4	19.007	31.898	328.072	328.804	38.279	331.124	328.465	328.479	328.465	331.313	331.313	331.313
k=5	Cluster 1	12.006	17.819	24.364	191.446	342.429	11.551	10.795	10.795	10.795	342.873	342.873	342.873
	Cluster 2	15.170	16.885	348.500	23.240	204.539	15.394	16.991	16.933	16.991	10.722	10.722	10.722
	Cluster 3	356.152	11.055	345.176	21.986	11.197	100.988	344.286	344.286	344.286	349.648	349.713	349.713
	Cluster 4	20.586	32.808	25.246	30.560	30.713	24.939	22.510	22.510	22.510	21.799	21.846	21.846
	Cluster 5	29.598	334.546	336.043	21.063	21.788	34.395	335.680	335.694	335.680	338.471	338.471	338.471

Sampling = 10 longitudes, latitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	17.419	17.419	17.419	17.787	17.787	17.787	17.419	17.419	17.419	17.419	17.419	17.419
	Cluster 2	20.292	20.292	20.292	20.280	20.280	20.280	20.292	20.292	20.292	20.292	20.292	20.292
k=3	Cluster 1	16.982	16.982	16.982	16.949	17.059	62.014	323.838	16.637	323.838	21.225	21.225	21.225
	Cluster 2	13.861	13.861	13.861	19.202	19.255	322.771	322.731	19.304	322.731	327.946	327.946	327.946
	Cluster 3	12.983	12.983	12.983	187.989	304.349	21.418	21.537	320.350	21.537	21.787	21.787	21.787
k=4	Cluster 1	331.479	329.751	14.136	18.495	12.394	327.459	18.434	327.657	18.434	22.578	22.578	22.578
	Cluster 2	34.871	322.971	13.733	96.116	19.018	289.650	19.091	324.191	19.091	13.733	13.733	13.733
	Cluster 3	28.970	34.668	12.967	22.810	239.078	97.451	22.449	320.416	22.449	331.365	331.365	331.365
	Cluster 4	28.421	310.007	22.846	27.682	24.761	28.585	28.832	308.496	28.832	309.994	309.994	309.994
k=5	Cluster 1	29.011	13.347	28.756	14.952	326.142	117.138	327.657	18.891	327.657	331.669	331.669	331.669
	Cluster 2	334.127	19.278	34.008	17.959	20.680	101.550	316.169	20.905	316.169	12.709	12.709	12.709
	Cluster 3	21.646	334.840	33.719	323.471	325.045	326.188	328.065	323.129	328.065	331.365	331.365	331.365
	Cluster 4	30.592	23.271	31.006	39.688	307.811	28.445	30.381	32.461	30.381	24.099	24.099	24.099
	Cluster 5	35.034	16.480	39.912	291.547	31.947	316.057	315.672	315.672	315.672	316.975	316.975	316.975

Sampling = 25 longitudes, latitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	18.765	18.765	18.765	109.530	148.398	36.279	18.765	18.765	18.765	18.765	18.765	18.765
	Cluster 2	21.896	21.896	21.896	64.103	137.571	98.696	21.896	21.896	21.896	21.896	21.896	21.896
k=3	Cluster 1	22.196	20.435	18.334	184.972	266.089	18.377	20.445	20.445	20.445	22.901	22.901	22.901
	Cluster 2	20.664	20.664	14.754	312.067	280.021	19.672	313.392	313.392	313.392	317.563	317.563	317.563
	Cluster 3	321.036	308.548	13.948	25.106	117.074	191.909	23.345	23.345	23.345	23.582	23.582	23.582
k=4	Cluster 1	14.453	15.349	15.349	18.773	15.971	15.822	19.880	19.880	19.880	24.370	24.370	24.370
	Cluster 2	20.742	14.623	14.623	17.767	36.581	20.018	20.565	20.094	20.565	14.646	14.646	14.646
	Cluster 3	323.003	13.929	13.929	33.085	16.086	24.594	24.595	23.142	24.595	320.734	320.734	320.734
	Cluster 4	20.518	23.981	23.981	36.527	69.923	54.304	30.484	30.484	30.484	298.673	298.673	298.673
k=5	Cluster 1	15.714	15.165	15.349	16.138	295.694	57.941	317.515	317.351	317.515	322.106	322.106	322.106
	Cluster 2	13.703	323.813	13.721	14.906	163.576	18.276	305.833	305.833	305.833	13.566	13.566	13.566
	Cluster 3	14.894	303.878	13.547	13.426	314.221	314.641	316.179	316.179	316.179	320.734	320.734	320.734
	Cluster 4	300.632	295.304	25.039	123.534	32.931	294.967	31.795	31.795	31.795	25.039	25.039	25.039
	Cluster 5	305.194	38.341	310.696	123.248	304.627	200.152	304.799	304.844	304.799	305.633	305.633	305.633

Sampling = 50 longitudes , latitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	19.142	19.142	19.142	154.631	111.632	142.904	19.142	19.142	19.142	19.142	19.142	19.142
	Cluster 2	22.346	22.346	22.346	157.525	57.468	114.645	22.346	22.346	22.346	22.346	22.346	22.346
k=3	Cluster 1	21.329	18.711	21.007	89.039	121.034	122.867	20.922	20.922	20.922	23.321	23.321	23.321
	Cluster 2	21.083	15.037	21.166	222.121	107.306	161.428	309.938	309.938	309.938	314.172	314.172	314.172
	Cluster 3	305.447	14.219	305.012	19.824	75.286	64.335	23.827	23.827	23.827	24.066	24.066	24.066
k=4	Cluster 1	16.031	15.672	313.532	90.594	62.934	158.764	20.345	20.345	20.345	24.791	24.791	24.791
	Cluster 2	14.006	14.906	303.189	58.357	146.957	156.208	20.984	20.503	20.984	14.930	14.930	14.930
	Cluster 3	13.829	14.202	14.202	84.130	206.444	159.280	25.096	23.612	25.096	317.430	317.430	317.430
	Cluster 4	300.012	24.331	39.139	135.161	290.284	144.797	30.973	30.973	30.973	295.017	295.017	295.017
k=5	Cluster 1	14.367	32.078	310.850	158.770	1158.770	14.733	314.115	313.949	314.115	318.779	318.779	318.779
	Cluster 2	298.839	14.930	314.290	156.209	156.208	149.101	302.070	302.070	302.070	13.830	13.830	13.830
	Cluster 3	14.212	319.813	47.516	159.281	159.283	301.844	312.727	312.727	312.727	317.430	317.430	317.430
	Cluster 4	46.536	32.597	25.349	144.133	144.132	293.232	32.353	32.353	32.353	25.349	25.349	25.349
	Cluster 5	301.428	301.923	302.372	146.712	146.714	299.689	301.057	301.102	301.057	301.923	301.923	301.923

Πίνακας 4.4: RMSE απόσταση του κεντροειδούς από τα μέλη της ομάδας (km) για Διανυσματοποίηση σε 2D Τροχιά με Χρήση Παρεμβολής, του συνόλου δεδομένων.

Οι ίδιοι πίνακες, που δίνουν τη μέση απόσταση του κεντροειδούς από τα μέλη της ομάδας, όπως αυτά έτρεξαν για το συνοπτικό σύνολο δεδομένων, δίνονται στη συνέχεια.

Sampling = 5 longitudes, latitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	14.403	14.403	14.403	14.637	14.637	14.637	14.403	14.403	14.403	14.403	14.403	14.403
	Cluster 2	16.219	16.219	16.219	16.195	16.195	16.195	16.219	16.219	16.219	16.219	16.219	16.219
k=3	Cluster 1	13.932	16.844	16.844	13.927	51.588	13.878	340.402	341.208	340.402	16.946	16.901	16.901
	Cluster 2	15.310	15.335	15.335	15.287	339.158	15.371	339.272	339.272	339.272	15.335	15.335	15.335
	Cluster 3	56.745	349.021	349.021	173.871	17.274	297.657	17.568	17.776	17.568	348.963	348.991	348.991
k=4	Cluster 1	23.746	342.218	12.715	12.950	12.890	11.893	13.720	13.720	13.720	18.013	17.936	17.936
	Cluster 2	15.335	69.029	28.049	15.422	13.631	275.152	15.075	15.075	15.075	10.944	10.944	10.944
	Cluster 3	278.485	11.131	338.847	343.565	17.344	21.733	17.030	17.030	17.030	349.024	349.052	349.052
	Cluster 4	37.303	24.689	326.558	276.462	24.753	328.340	24.671	24.671	24.671	330.440	330.440	330.440
k=5	Cluster 1	11.616	12.783	11.982	345.194	106.273	344.718	344.030	344.013	344.030	342.228	342.228	342.228
	Cluster 2	10.944	16.873	16.467	344.353	187.782	18.871	336.749	336.749	336.749	10.828	10.828	10.828
	Cluster 3	351.157	22.506	15.563	343.533	17.290	343.619	346.740	346.740	346.740	349.024	349.052	349.052
	Cluster 4	296.848	21.982	328.453	325.315	25.593	26.561	26.073	26.073	26.073	21.900	21.926	21.926
	Cluster 5	13.943	338.133	332.909	16.486	22.336	280.356	335.028	335.036	335.028	337.616	337.616	337.616

Sampling = 10 longitudes, latitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	17.605	17.605	17.605	80.754	17.918	17.918	17.605	17.605	17.605	17.605	17.605	17.605
	Cluster 2	20.516	20.516	20.516	65.900	20.474	20.474	20.516	20.516	20.516	20.516	20.516	20.516
k=3	Cluster 1	17.153	17.153	17.153	80.648	17.200	17.162	19.187	16.770	19.187	21.210	21.210	21.210
	Cluster 2	13.785	13.785	13.785	321.496	19.161	19.459	323.208	19.483	323.208	327.320	327.320	327.320
	Cluster 3	13.073	13.073	13.073	22.695	271.330	253.774	21.815	320.178	21.815	21.815	21.815	21.815
k=4	Cluster 1	326.381	13.546	14.724	326.666	14.979	15.220	18.642	327.825	18.642	22.809	22.809	22.809
	Cluster 2	327.520	19.483	13.195	321.817	141.454	17.649	19.309	323.667	19.309	13.785	13.785	13.785
	Cluster 3	319.476	332.841	29.297	148.458	328.144	37.436	22.894	319.989	22.894	330.823	330.823	330.823
	Cluster 4	309.383	19.910	313.272	81.338	308.204	73.917	29.069	307.748	29.069	309.383	309.383	309.383
k=5	Cluster 1	13.593	14.400	13.531	14.926	16.971	14.932	327.281	18.941	327.281	330.794	330.794	330.794
	Cluster 2	312.200	19.483	329.027	102.516	21.888	17.120	315.701	21.112	315.701	13.075	13.075	13.075
	Cluster 3	34.189	338.291	34.189	326.625	324.672	19.365	327.313	322.144	327.313	330.823	330.823	330.823
	Cluster 4	306.965	57.848	21.274	189.540	304.861	37.448	30.508	32.748	30.508	24.242	24.242	24.242
	Cluster 5	40.483	35.378	316.413	315.418	317.665	139.325	315.218	314.983	315.218	316.413	316.413	316.413

Sampling = 25 longitudes, latitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	18.943	18.943	18.943	20.016	112.096	68.670	18.943	18.943	18.943	18.943	18.943	18.943
	Cluster 2	22.092	22.092	22.092	22.020	67.684	34.313	22.092	22.092	22.092	22.092	22.092	22.092
k=3	Cluster 1	21.562	20.857	18.506	312.155	133.800	18.971	20.627	20.627	20.627	22.850	22.850	22.850
	Cluster 2	306.840	20.927	14.704	196.747	310.841	86.412	312.644	312.644	312.644	316.899	316.899	316.899
	Cluster 3	23.555	308.026	14.105	313.610	24.774	44.926	23.555	23.555	23.555	23.555	23.555	23.555
k=4	Cluster 1	15.576	316.140	15.576	20.157	163.665	16.288	20.113	20.113	20.113	24.557	24.557	24.557
	Cluster 2	14.718	316.940	14.772	221.833	28.184	14.068	20.369	20.440	20.369	14.785	14.785	14.785
	Cluster 3	14.089	308.085	14.092	21.001	26.666	14.381	23.594	23.809	23.594	320.094	320.094	320.094
	Cluster 4	24.163	298.071	24.163	48.179	29.871	47.391	30.699	30.699	30.699	297.960	297.960	297.960
k=5	Cluster 1	20.521	14.712	15.894	316.200	316.279	110.908	316.695	316.592	316.695	321.195	321.195	321.195
	Cluster 2	20.927	14.704	21.871	150.835	311.913	305.462	305.000	305.000	305.000	13.726	13.726	13.726
	Cluster 3	316.399	322.440	32.539	217.792	97.586	164.917	315.986	315.986	315.986	320.094	320.094	320.094
	Cluster 4	25.654	295.677	297.323	266.922	29.886	294.596	31.961	31.961	31.961	25.191	25.191	25.191
	Cluster 5	22.093	17.872	304.499	28.066	196.952	28.009	304.106	304.131	304.106	304.966	304.966	304.966

Sampling = 50 longitudes, latitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	19.320	19.320	19.320	154.284	154.284	122.149	19.320	19.320	19.320	19.320	19.320	19.320
	Cluster 2	22.537	22.537	22.537	157.156	157.157	80.515	22.537	22.537	22.537	22.537	22.537	22.537
k=3	Cluster 1	18.888	18.888	21.237	150.013	154.263	118.510	21.086	21.086	21.086	23.268	23.268	23.268
	Cluster 2	15.055	15.047	21.345	46.588	155.790	76.175	309.116	309.116	309.116	313.490	313.490	313.490
	Cluster 3	14.413	14.428	304.339	233.790	158.909	78.723	24.033	24.033	24.033	24.033	24.033	24.033
k=4	Cluster 1	16.848	310.092	31.809	62.403	16.644	164.827	20.604	20.604	20.604	24.974	24.974	24.974
	Cluster 2	37.636	38.226	21.345	224.632	21.141	151.246	20.774	20.847	20.774	15.047	15.047	15.047
	Cluster 3	305.598	310.739	312.724	18.016	134.089	156.510	24.064	24.283	24.064	316.770	316.770	316.770
	Cluster 4	293.594	299.151	44.837	85.332	94.767	124.058	31.191	31.191	31.191	294.333	294.333	294.333
k=5	Cluster 1	16.213	15.002	314.070	158.596	233.550	158.425	313.173	313.147	313.173	317.845	317.845	317.845
	Cluster 2	14.078	15.000	308.035	155.639	87.113	155.787	301.263	301.263	301.263	13.964	13.964	13.964
	Cluster 3	15.139	14.364	15.411	158.811	310.414	158.974	311.918	311.918	311.918	316.770	316.770	316.770
	Cluster 4	291.918	22.372	288.210	143.903	98.097	143.749	32.424	32.424	32.424	25.492	25.492	25.492
	Cluster 5	305.681	44.658	305.659	146.554	299.849	146.407	300.387	300.397	300.387	301.286	301.286	301.286

Πίνακας 4.5: RMSE απόσταση του κεντροειδούς από τα μέλη της ομάδας (km) για Διανυσματοποίηση σε 2D Τροχιά με Χρήση Παρεμβολής, του συνοπτικού συνόλου δεδομένων.

Παρατηρούμε και για τα δύο σύνολα δεδομένων ότι η μέση απόσταση του κεντροειδούς από τα μέλη της ομάδας αυξάνεται καθώς αυξάνεται το μέγεθος το διανύσματος (για τον ίδιο αριθμό ομάδων). Επίσης οι μικρότερες αποστάσεις παρατηρούνται στην περίπτωση των δύο ομάδων, γεγονός που υποδεικνύει ότι οι αλγόριθμοι δημιουργούν ομάδες πιο ομογενείς στην περίπτωση των 2 ομάδων. Οι αλγόριθμοι που δημιουργούν ομαδοποίηση με μικρές RMSE αποστάσεις είναι οι K-means, Power Iteration Clustering και Bisecting K-means. Ο Gaussian Mixture Model φαίνεται πως καθώς αυξάνεται το μέγεθος του διανύσματος δε δημιουργεί τόσο ομογενείς ομάδες.

Τέλος παραθέτουμε έναν πίνακα με τους χρόνους εκτέλεσης καθενός αλγορίθμου για τις διάφορες τιμές των παραμέτρων που μελετήσαμε, όπως έτρεξαν για το συνολικό σύνολο δεδομένων.

Sampling = 5 longitudes, latitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.026	0.026	0.028	0.034	0.031	0.031	0.027	0.037	0.051	0.025	0.031	0.043
k=3	0.032	0.034	0.036	0.029	0.041	0.040	0.030	0.037	0.046	0.027	0.044	0.063
k=4	0.030	0.037	0.036	0.043	0.038	0.035	0.027	0.038	0.056	0.026	0.044	0.063
k=5	0.033	0.034	0.033	0.045	0.044	0.048	0.028	0.037	0.051	0.027	0.055	0.088

Sampling = 10 longitudes, latitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.027	0.028	0.025	0.032	0.039	0.033	0.028	0.036	0.050	0.022	0.031	0.041
k=3	0.031	0.034	0.034	0.039	0.047	0.042	0.027	0.036	0.052	0.026	0.042	0.065
k=4	0.034	0.033	0.038	0.044	0.047	0.060	0.028	0.035	0.052	0.027	0.047	0.065
k=5	0.036	0.036	0.036	0.048	0.056	0.057	0.029	0.038	0.046	0.028	0.055	0.083

Sampling = 25 longitudes, latitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.030	0.029	0.029	0.049	0.054	0.051	0.027	0.037	0.048	0.024	0.033	0.046
k=3	0.034	0.036	0.034	0.077	0.071	0.075	0.029	0.035	0.049	0.026	0.045	0.065
k=4	0.036	0.033	0.040	0.124	0.097	0.100	0.029	0.038	0.052	0.028	0.046	0.064
k=5	0.040	0.037	0.035	0.136	0.129	0.131	0.026	0.036	0.050	0.027	0.056	0.088

Sampling = 50 longitudes, latitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.032	0.028	0.032	0.108	0.120	0.104	0.028	0.038	0.052	0.026	0.031	0.048
k=3	0.036	0.034	0.038	0.183	0.180	0.179	0.029	0.035	0.046	0.028	0.049	0.067
k=4	0.034	0.046	0.040	0.285	0.281	0.281	0.033	0.038	0.050	0.030	0.048	0.067
k=5	0.039	0.049	0.045	0.406	0.582	0.402	0.031	0.037	0.045	0.034	0.060	0.088

Πίνακας 4.6: Χρόνος εκτέλεσης (sec) για Διανυσματοποίηση σε 2D Τροχιά με Χρήση Παρεμβολής, του συνόλου δεδομένων.

Οι ίδιοι πίνακες, που δίνουν το χρόνο εκτέλεσης των αλγορίθμων, όπως αυτοί έτρεξαν για το συνοπτικό σύνολο δεδομένων, δίνονται στη συνέχεια.

Sampling = 5 longitudes, latitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.028	0.026	0.027	0.029	0.031	0.029	0.028	0.037	0.050	0.024	0.032	0.043
k=3	0.027	0.035	0.030	0.032	0.033	0.035	0.029	0.036	0.052	0.027	0.042	0.065
k=4	0.036	0.032	0.031	0.033	0.036	0.039	0.028	0.036	0.055	0.024	0.042	0.066
k=5	0.033	0.034	0.035	0.037	0.048	0.041	0.026	0.037	0.052	0.027	0.054	0.086

Sampling = 10 longitudes, latitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.026	0.027	0.027	0.033	0.033	0.035	0.027	0.038	0.052	0.025	0.031	0.045
k=3	0.031	0.032	0.032	0.040	0.045	0.043	0.028	0.037	0.048	0.025	0.043	0.062
k=4	0.034	0.034	0.034	0.042	0.048	0.050	0.027	0.040	0.049	0.025	0.043	0.067
k=5	0.033	0.035	0.033	0.048	0.049	0.058	0.026	0.035	0.049	0.030	0.055	0.083

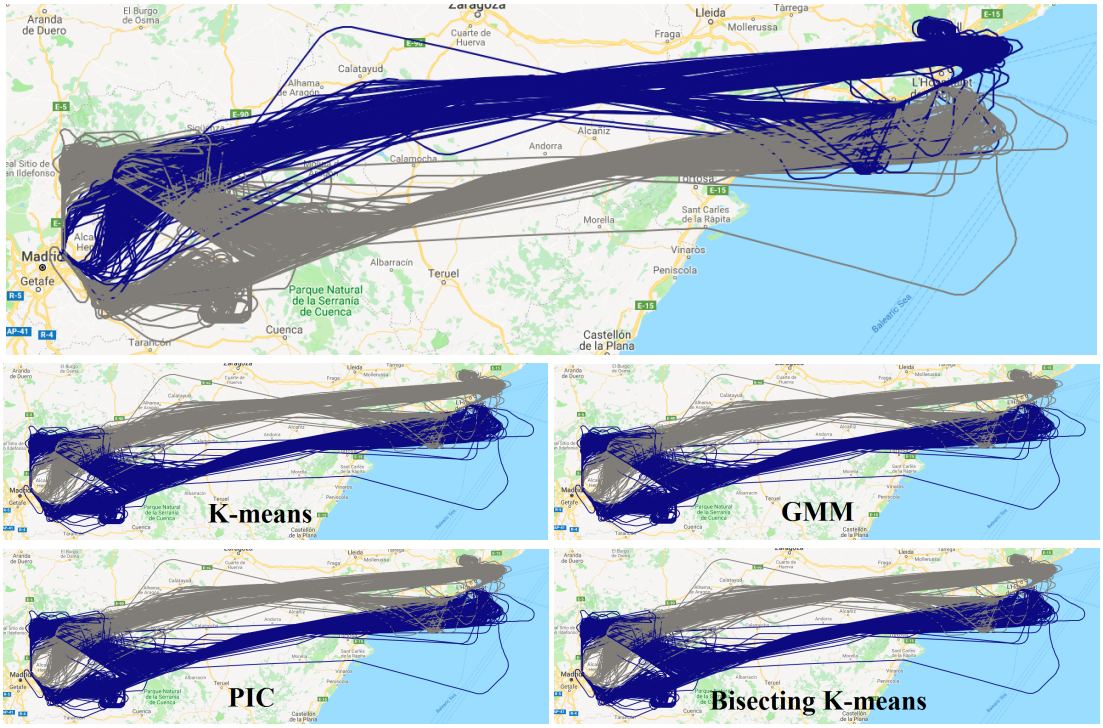
Sampling = 25 longitudes, latitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.032	0.026	0.027	0.047	0.052	0.056	0.027	0.040	0.047	0.027	0.036	0.046
k=3	0.034	0.036	0.034	0.089	0.078	0.076	0.029	0.036	0.050	0.026	0.047	0.065
k=4	0.035	0.041	0.037	0.098	0.099	0.107	0.027	0.037	0.054	0.027	0.048	0.069
k=5	0.041	0.039	0.036	0.126	0.137	0.162	0.027	0.040	0.052	0.028	0.058	0.057

Sampling = 50 longitudes, latitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.034	0.029	0.029	0.110	0.107	0.108	0.029	0.037	0.047	0.027	0.034	0.046
k=3	0.033	0.032	0.038	0.179	0.176	0.175	0.027	0.041	0.050	0.033	0.051	0.068
k=4	0.039	0.039	0.038	0.287	0.276	0.283	0.030	0.038	0.051	0.032	0.051	0.075
k=5	0.048	0.046	0.041	0.411	0.423	0.409	0.026	0.037	0.051	0.033	0.063	0.096

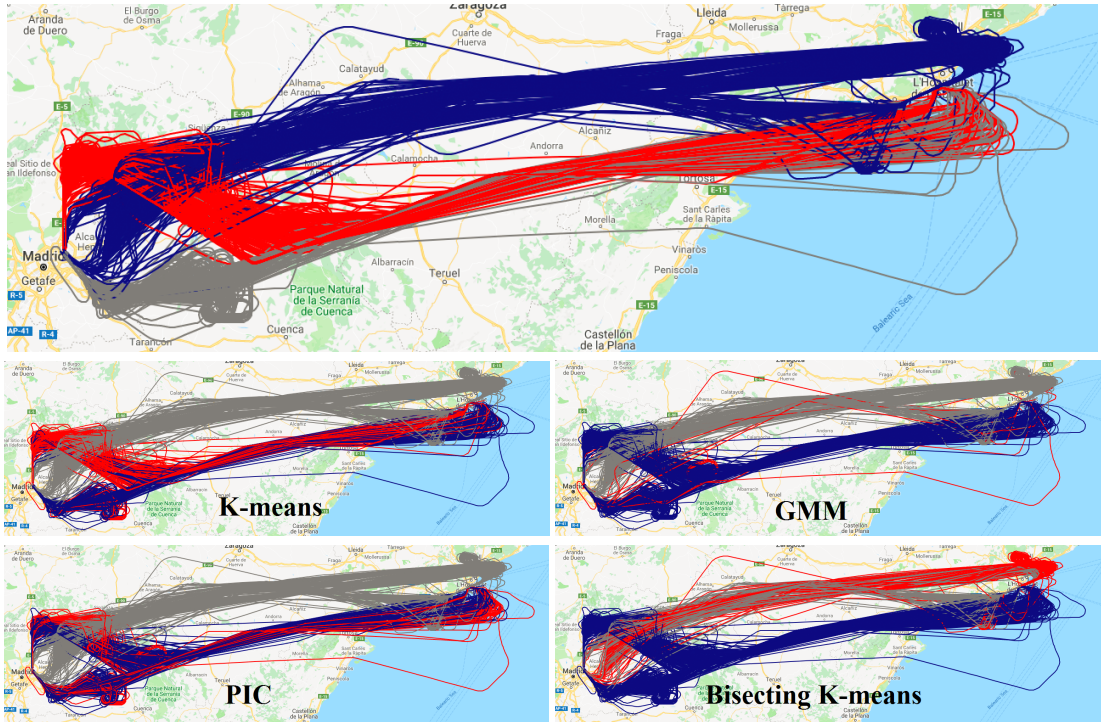
Πίνακας 4.7: Χρόνος εκτέλεσης (sec) για Διανυσματοποίηση σε 2D Τροχιά με Χρήση Παρεμβολής, του συνοπτικού συνόλου δεδομένων.

Οι χρόνοι εκτέλεσης δε φαίνεται να διαφέρουν ιδιαίτερα από αλγόριθμο σε αλγόριθμο. Παρατηρείται μία μικρή αύξηση του χρόνου εκτέλεσης όλων των αλγορίθμων καθώς αυξάνεται το πλήθος των ομάδων που δημιουργούν οι αλγόριθμοι. Επίσης η αύξηση του αριθμού των επαναλήψεων αυξάνει το χρόνο εκτέλεσης για την περίπτωση των αλγορίθμων Power Iteration Clustering και Bisecting K-means, ενώ οι αλγόριθμοι K-means και Gaussian Mixture Model δε φαίνεται να επηρεάζονται με κάποιο συστηματικό τρόπο.

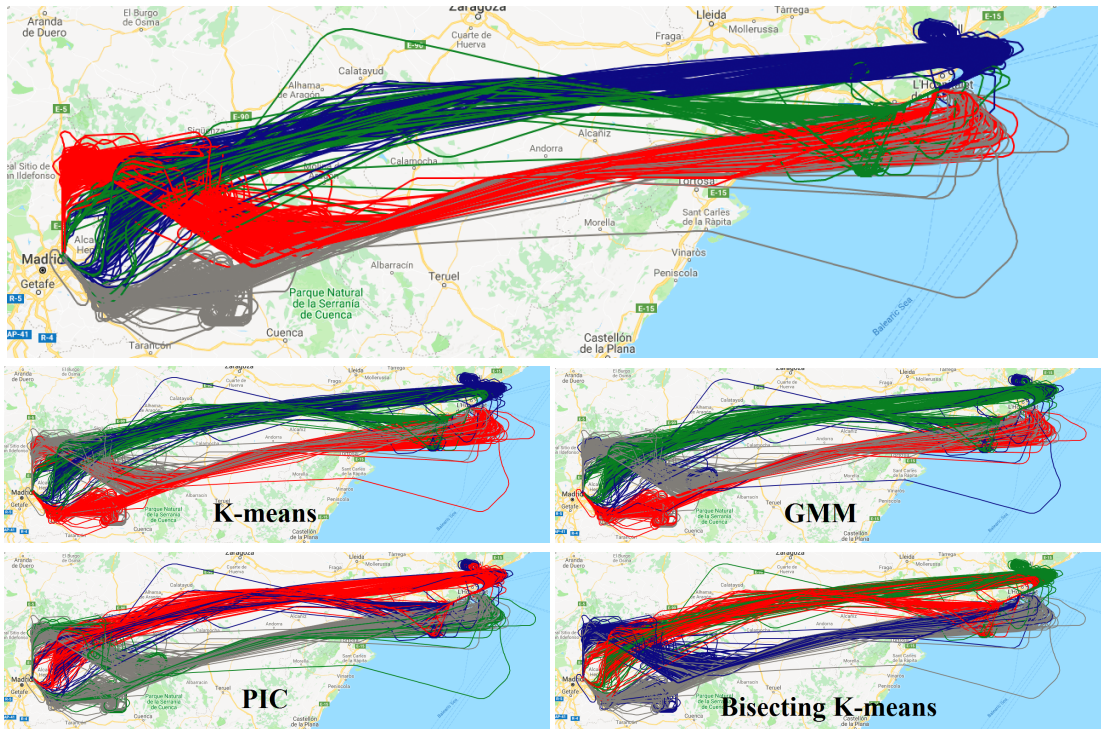
Στη συνέχεια απεικονίζουμε τις καλύτερες ομαδοποιήσεις ανά αριθμό ομάδων για καθέναν από τους αλγορίθμους σε σύγκριση με το ground truth.



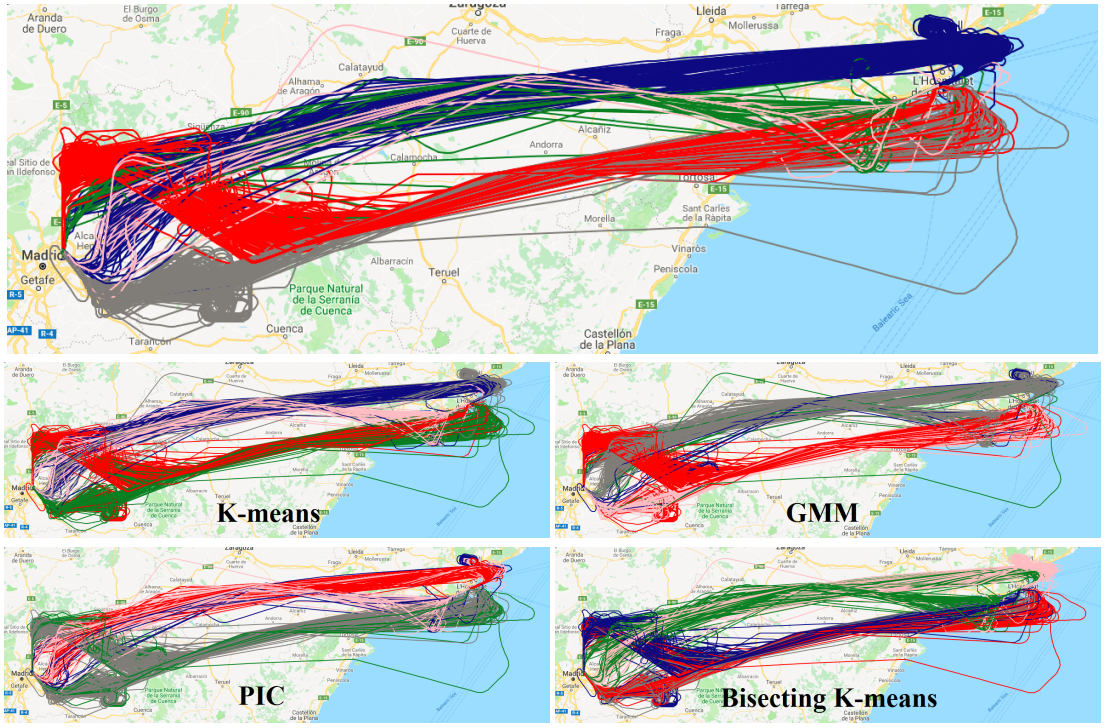
Πίνακας 4.8: Οπτική σύγκριση ομαδοποίησης αλγορίθμων με ground truth για 2 ομάδες.



Πίνακας 4.9: Οπτική σύγκριση ομαδοποίησης αλγορίθμων με ground truth για 3 ομάδες.



Πίνακας 4.10: Οπτική σύγκριση ομαδοποίησης αλγορίθμων με ground truth για 4 ομάδες.



Πίνακας 4.11: Οπτική σύγκριση ομαδοποίησης αλγορίθμων με ground truth για 5 ομάδες.

Κάτι που επιβεβαιώνεται και οπτικά είναι πως οι αλγόριθμοι δίνουν καλύτερα αποτελέσματα στην περίπτωση των δύο ομάδων. Όσο ο αριθμός των ομάδων αυξάνεται, παρατηρούμε σημαντική διαφοροποίηση σε σχέση με το ground truth. Επίσης, η ομαδοποίηση που επιτυγχάνουν και οι τέσσερις αλγόριθμοι για δύο ομάδες είναι σχεδόν η ίδια.

4.2.2 Διανυσματοποίηση σε 3D Τροχιά με Χρήση Παρεμβολής

Όπως αναφέραμε παραπάνω, η διανυσματοποίηση αυτή έγινε στα δύο σύνολα δεδομένων με χρήση της παρεμβολής. Για κάθε μία από τις μεταβλητές γεωγραφικό μήκος, γεωγραφικό πλάτος και ύψος εφαρμόσαμε παρεμβολή 5, 10, 25 και 50 σημείων.

Η είσοδος που δέχτηκαν οι αλγόριθμοι K-means, Gaussian Mixture Model και Bisecting K-means, είναι ένας πίνακας με τόσες γραμμές όσες και οι πτήσεις του συνόλου δεδομένων και τόσες στήλες όσες ο τριπλάσιος αριθμός του μεγέθους του διανύσματος που επιλέξαμε. Για παράδειγμα στην περίπτωση που επιλέξαμε 5 σημεία παρεμβολής, τότε για την κάθε πτήση κρατήσαμε 15 σημεία, όπου τα 5 πρώτα αποτελούν το γεωγραφικό μήκος, τα 5 επόμενα το γεωγραφικό πλάτος και τα 5 τελευταία το ύψος (όπως αυτά προέκυψαν από την εφαρμογή της παρεμβολής).

Η είσοδος που δέχτηκε ο αλγόριθμος Power Iteration Clustering είναι ένας πίνακας με 3 στήλες, όπου η πρώτη στήλη υποδεικνύει την πτήση i, η δεύτερη στήλη υποδεικνύει την πτήση j και η τρίτη στήλη υποδεικνύει την απόσταση των προαναφερθέντων πτήσεων (με $i \leq j$). Σημειώνουμε ότι η απόσταση αυτή υπολογίστηκε κάνοντας μετατροπή των πολικών συντεταγμένων σε καρτεσιανές και εφαρμόζοντας την ευκλείδεια απόσταση.

Για τους κώδικες που χρησιμοποιήθηκαν για τη δημιουργία των συνόλων δεδομένων που δέχονται σαν είσοδο οι αλγόριθμοι ομαδοποίησης, παραπέμπουμε στο Παράρτημα Β'.

Στη συνέχεια τρέξαμε τους αλγορίθμους ομαδοποίησης για όλες τις περιπτώσεις του μεγέθους του διανύσματος (οι κώδικες των αλγορίθμων ομαδοποίησης παραμένουν ίδιοι).

Τα αποτελέσματα της αξιολόγησης των αλγορίθμων για καθεμία από τις παραμέτρους για τις οποίες εξετάστηκαν (πλήθος ομάδων k και μέγιστο πλήθος επαναλήψεων *MaxIterations*) είναι συγκεντρωμένα στους πίνακες που ακολουθούν.

Οι πίνακες που δίνονται στη συνέχεια περιλαμβάνουν τις τιμές των Accuracy, Precision και F1 Score που αξιολογούν το αποτέλεσμα του κάθε αλγορίθμου που έτρεξε σε ολόκληρο το σύνολο δεδομένων, σε σχέση με το *ground truth*.

Sampling = 5 longitudes, latitudes, altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.526	0.509	0.509	0.908	0.778	0.991	0.998	0.998	0.998	0.519	0.509	0.509
	Precision	0.541	0.508	0.508	0.869	0.908	0.996	0.996	0.996	0.996	0.510	0.508	0.508
	F1-Score	0.368	0.351	0.351	0.912	0.732	0.991	0.998	0.998	0.998	0.600	0.351	0.351
k=3	Accuracy	0.477	0.614	0.622	0.753	0.817	0.841	0.724	0.724	0.724	0.463	0.478	0.478
	Precision	0.510	0.607	0.609	0.527	0.763	0.822	0.518	0.518	0.518	0.508	0.528	0.528
	F1-Score	0.464	0.581	0.586	0.556	0.765	0.786	0.552	0.552	0.552	0.453	0.467	0.467
k=4	Accuracy	0.511	0.521	0.519	0.552	0.618	0.667	0.635	0.635	0.635	0.412	0.400	0.400
	Precision	0.487	0.491	0.490	0.496	0.429	0.593	0.488	0.488	0.488	0.431	0.396	0.396
	F1-Score	0.480	0.486	0.484	0.416	0.412	0.535	0.483	0.483	0.483	0.388	0.351	0.351
k=5	Accuracy	0.475	0.479	0.460	0.560	-	0.575	0.589	0.589	0.589	0.391	0.404	0.404
	Precision	0.380	0.389	0.380	0.457	-	0.468	0.430	0.430	0.430	0.373	0.373	0.373
	F1-Score	0.375	0.379	0.367	0.382	-	0.341	0.409	0.409	0.409	0.339	0.335	0.335

Sampling = 10 longitudes, latitudes, altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.504	0.503	0.503	-	-	-	0.999	0.999	0.999	0.501	0.501	0.501
	Precision	0.499	0.497	0.497	-	-	-	0.997	0.997	0.997	0.494	0.494	0.494
	F1-Score	0.430	0.429	0.429	-	-	-	0.999	0.999	0.999	0.435	0.435	0.435
k=3	Accuracy	0.458	0.599	0.605	-	-	-	0.734	0.734	0.734	0.547	0.548	0.548
	Precision	0.448	0.602	0.634	-	-	-	0.518	0.518	0.518	0.572	0.575	0.575
	F1-Score	0.403	0.586	0.603	-	-	-	0.555	0.555	0.555	0.544	0.545	0.545
k=4	Accuracy	0.447	0.496	0.494	-	-	-	0.612	0.612	0.612	0.524	0.525	0.525
	Precision	0.423	0.500	0.502	-	-	-	0.547	0.547	0.547	0.520	0.521	0.521
	F1-Score	0.415	0.484	0.484	-	-	-	0.495	0.495	0.495	0.511	0.512	0.512
k=5	Accuracy	0.418	0.478	0.443	-	-	-	0.577	0.577	0.577	0.437	0.430	0.430
	Precision	0.337	0.425	0.401	-	-	-	0.442	0.442	0.442	0.419	0.441	0.441
	F1-Score	0.324	0.406	0.364	-	-	-	0.428	0.428	0.428	0.383	0.382	0.382

Sampling = 25 longitudes, latitudes, altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.505	0.504	0.505	-	-	-	0.999	0.999	0.999	0.505	0.505	0.505
	Precision	0.500	0.499	0.500	-	-	-	0.997	0.997	0.997	0.500	0.500	0.500
	F1-Score	0.554	0.430	0.554	-	-	-	0.999	0.999	0.999	0.554	0.554	0.554
k=3	Accuracy	0.595	0.596	0.606	-	-	-	0.723	0.723	0.723	0.532	0.532	0.532
	Precision	0.595	0.614	0.616	-	-	-	0.518	0.518	0.518	0.557	0.557	0.557
	F1-Score	0.583	0.591	0.596	-	-	-	0.551	0.551	0.551	0.529	0.529	0.529
k=4	Accuracy	0.527	0.462	0.509	-	-	-	0.615	0.615	0.615	0.499	0.499	0.499
	Precision	0.510	0.436	0.497	-	-	-	0.460	0.460	0.460	0.487	0.487	0.487
	F1-Score	0.505	0.430	0.487	-	-	-	0.464	0.464	0.464	0.482	0.482	0.482
k=5	Accuracy	0.443	0.395	0.458	-	-	-	0.603	0.603	0.603	0.420	0.419	0.419
	Precision	0.378	0.386	0.402	-	-	-	0.428	0.428	0.428	0.377	0.373	0.373
	F1-Score	0.359	0.350	0.372	-	-	-	0.405	0.405	0.405	0.357	0.353	0.353

Sampling = 50 longitudes, latitudes, altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.504	0.503	0.504	-	-	-	0.999	0.999	0.999	0.504	0.504	0.504
	Precision	0.499	0.497	0.499	-	-	-	0.997	0.997	0.997	0.499	0.499	0.499
	F1-Score	0.553	0.427	0.553	-	-	-	0.999	0.999	0.999	0.554	0.553	0.553
k=3	Accuracy	0.587	0.597	0.605	-	-	-	0.723	0.723	0.723	0.530	0.531	0.531
	Precision	0.660	0.614	0.609	-	-	-	0.518	0.518	0.518	0.552	0.553	0.553
	F1-Score	0.603	0.591	0.594	-	-	-	0.551	0.551	0.551	0.528	0.528	0.528
k=4	Accuracy	0.508	0.448	0.509	-	-	-	0.615	0.615	0.615	0.497	0.498	0.498
	Precision	0.491	0.440	0.497	-	-	-	0.460	0.460	0.460	0.484	0.485	0.485
	F1-Score	0.486	0.428	0.488	-	-	-	0.464	0.464	0.464	0.480	0.481	0.481
k=5	Accuracy	0.426	0.427	0.432	-	-	-	0.602	0.602	0.602	0.409	0.408	0.408
	Precision	0.433	0.394	0.380	-	-	-	0.427	0.427	0.427	0.364	0.361	0.361
	F1-Score	0.381	0.367	0.359	-	-	-	0.404	0.404	0.404	0.344	0.342	0.342

Πίνακας 4.12: Accuracy, Precision και F1 Score(%) για Διανυσματοποίηση σε 3D Τροχιά με Χρήση Παρεμβολής, του συνόλου δεδομένων.

Οι ίδιοι πίνακες, που δίνουν τα αποτελέσματα αξιολόγησης των αλγορίθμων όπως αυτά έτρεξαν για το συνοπτικό σύνολο δεδομένων, δίνονται στη συνέχεια.

Sampling = 5 longitudes, latitudes, altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.569	0.602	0.513	0.994	0.991	0.880	0.998	0.998	0.998	0.501	0.501	0.501
	Precision	0.554	0.569	0.514	0.987	0.996	0.831	0.996	0.996	0.996	0.494	0.494	0.494
	F1-Score	0.604	0.668	0.374	0.994	0.991	0.887	0.998	0.998	0.998	0.372	0.365	0.365
k=3	Accuracy	0.481	0.481	0.481	0.761	0.746	0.787	0.725	0.725	0.725	0.492	0.494	0.494
	Precision	0.517	0.518	0.518	0.544	0.535	0.739	0.518	0.518	0.518	0.534	0.535	0.535
	F1-Score	0.470	0.467	0.469	0.562	0.563	0.657	0.552	0.552	0.552	0.483	0.485	0.485
k=4	Accuracy	0.479	0.504	0.510	0.615	0.596	0.615	0.640	0.640	0.640	0.413	0.413	0.413
	Precision	0.457	0.481	0.490	0.453	0.461	0.419	0.497	0.497	0.497	0.410	0.407	0.407
	F1-Score	0.455	0.476	0.480	0.389	0.411	0.410	0.485	0.485	0.485	0.371	0.368	0.368
k=5	Accuracy	0.432	0.449	0.458	0.661	0.615	0.586	0.597	0.597	0.597	0.365	0.367	0.367
	Precision	0.366	0.376	0.378	0.536	0.423	0.483	0.423	0.423	0.423	0.372	0.368	0.368
	F1-Score	0.348	0.362	0.365	0.439	0.327	0.301	0.405	0.405	0.405	0.322	0.321	0.321

Sampling = 10 longitudes, latitudes, altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.521	0.527	0.528	-	-	-	0.999	0.999	0.999	0.527	0.527	0.527
	Precision	0.522	0.529	0.530	-	-	-	0.997	0.997	0.997	0.529	0.529	0.529
	F1-Score	0.443	0.460	0.461	-	-	-	0.999	0.999	0.999	0.459	0.460	0.460
k=3	Accuracy	0.607	0.608	0.613	-	-	-	0.723	0.723	0.723	0.568	0.567	0.567
	Precision	0.644	0.639	0.616	-	-	-	0.518	0.518	0.518	0.595	0.594	0.594
	F1-Score	0.611	0.610	0.600	-	-	-	0.551	0.551	0.551	0.565	0.564	0.564
k=4	Accuracy	0.501	0.552	0.491	-	-	-	0.620	0.620	0.620	0.534	0.533	0.533
	Precision	0.472	0.515	0.488	-	-	-	0.564	0.564	0.564	0.522	0.517	0.517
	F1-Score	0.472	0.515	0.474	-	-	-	0.516	0.516	0.516	0.515	0.512	0.512
k=5	Accuracy	0.460	0.483	0.402	-	-	-	0.567	0.567	0.567	0.450	0.450	0.450
	Precision	0.430	0.424	0.406	-	-	-	0.467	0.467	0.467	0.421	0.416	0.416
	F1-Score	0.398	0.406	0.362	-	-	-	0.459	0.459	0.459	0.390	0.388	0.388

Sampling = 25 longitudes, latitudes, altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.525	0.527	0.527	-	-	-	0.999	0.999	0.999	0.527	0.527	0.527
	Precision	0.528	0.531	0.531	-	-	-	0.997	0.997	0.997	0.530	0.531	0.531
	F1-Score	0.441	0.446	0.446	-	-	-	0.999	0.999	0.999	0.455	0.446	0.446
k=3	Accuracy	0.602	0.606	0.604	-	-	-	0.723	0.723	0.723	0.556	0.563	0.563
	Precision	0.634	0.636	0.636	-	-	-	0.518	0.518	0.518	0.581	0.583	0.583
	F1-Score	0.604	0.608	0.606	-	-	-	0.551	0.551	0.551	0.553	0.559	0.559
k=4	Accuracy	0.494	0.510	0.509	-	-	-	0.622	0.622	0.622	0.524	0.531	0.531
	Precision	0.458	0.485	0.500	-	-	-	0.566	0.566	0.566	0.513	0.518	0.518
	F1-Score	0.455	0.481	0.489	-	-	-	0.519	0.519	0.519	0.506	0.510	0.510
k=5	Accuracy	0.413	0.413	0.442	-	-	-	0.565	0.565	0.565	0.443	0.446	0.446
	Precision	0.343	0.340	0.372	-	-	-	0.465	0.465	0.465	0.406	0.412	0.412
	F1-Score	0.327	0.326	0.354	-	-	-	0.457	0.457	0.457	0.383	0.386	0.386

Sampling = 50 longitudes, latitudes, altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.529	0.529	0.529	-	-	-	0.999	0.999	0.999	0.524	0.529	0.529
	Precision	0.533	0.533	0.533	-	-	-	0.997	0.997	0.997	0.526	0.533	0.533
	F1-Score	0.451	0.451	0.451	-	-	-	0.999	0.999	0.999	0.456	0.451	0.451
k=3	Accuracy	0.608	0.606	0.614	-	-	-	0.723	0.723	0.723	0.549	0.564	0.564
	Precision	0.616	0.607	0.620	-	-	-	0.518	0.518	0.518	0.575	0.584	0.584
	F1-Score	0.599	0.593	0.604	-	-	-	0.551	0.551	0.551	0.546	0.560	0.560
k=4	Accuracy	0.524	0.509	0.438	-	-	-	0.615	0.615	0.615	0.519	0.529	0.529
	Precision	0.506	0.498	0.434	-	-	-	0.503	0.503	0.503	0.509	0.516	0.516
	F1-Score	0.503	0.488	0.421	-	-	-	0.496	0.496	0.496	0.502	0.509	0.509
k=5	Accuracy	0.408	0.435	0.458	-	-	-	0.602	0.602	0.602	0.440	0.446	0.446
	Precision	0.399	0.356	0.384	-	-	-	0.465	0.465	0.465	0.397	0.411	0.411
	F1-Score	0.361	0.338	0.368	-	-	-	0.431	0.431	0.431	0.375	0.386	0.386

Πίνακας 4.13: Accuracy, Precision και F1 Score(%) για Διανυσματοποίηση σε 3D Τροχιά με Χρήση Παρεμβολής, του συνοπτικού συνόλου δεδομένων.

Παρατηρούμε αρχικά ότι ο αλγόριθμος Gaussian Mixture Model αποτυγχάνει στην περίπτωση για τα μεγέθη διανύσματος μεγαλύτερα του 5, παρόλο που η ομαδοποίηση που επιτυγχάνει στην περίπτωση του μεγέθους διανύσματος ίσο με 5 δίνει αρκετά υψηλές τιμές στα μέτρα απόδοσης. Σημειώνουμε επίσης ότι οι αλγόριθμοι K-means και Bisecting K-means έχουν χαμηλές τιμές στα μέτρα απόδοσης για όλες τις περιπτώσεις του πλήθους ομάδων αλλά και για όλες τις περιπτώσεις του μεγέθους του διανύσματος. Αντίθετα, τα μέτρα απόδοσης του αλγορίθμου Power Iteration Clustering είναι σχεδόν ίσα με τη μονάδα για την περίπτωση των δύο ομάδων και μειώνονται καθώς το πλήθος των ομάδων αυξάνεται και η εικόνα αυτή παρατηρείται για όλα τα μεγέθη διανύσματος που έχουμε εφαρμόσει.

Στη συνέχεια δίνονται οι πίνακες που περιλαμβάνουν τη μέση απόσταση του κεντροειδούς από τα μέλη της ομάδας, όπως αυτή υπολογίστηκε για κάθε αλγόριθμο που έτρεξε στο συνολικό σύνολο δεδομένων.

Sampling = 5 longitudes, latitudes, altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	15.282	15.595	15.595	15.518	15.517	3.307	3.218	3.218	3.218	13.970	15.595	15.595
	Cluster 2	14.912	15.754	15.754	16.348	16.347	4.010	3.925	3.925	3.925	16.032	15.754	15.754
k=3	Cluster 1	7.590	4.110	16.908	17.314	15.514	15.513	3.180	3.180	3.180	18.822	18.410	18.410
	Cluster 2	11.462	8.911	21.231	19.313	15.928	15.927	3.795	3.795	3.795	16.667	15.445	15.445
	Cluster 3	17.911	14.342	9.231	13.462	16.878	16.878	22.475	22.475	22.475	21.046	21.306	21.306
k=4	Cluster 1	5.084	18.417	7.038	15.743	2.873	15.742	2.804	2.804	2.804	7.949	6.923	6.923
	Cluster 2	20.916	20.623	5.774	15.929	3.830	15.927	3.795	3.795	3.795	10.700	13.102	13.102
	Cluster 3	11.763	21.641	14.253	16.889	18.927	16.889	21.657	21.657	21.657	13.460	10.408	10.408
	Cluster 4	20.614	20.830	7.901	14.986	12.280	14.985	4.272	4.272	4.272	9.056	11.615	11.615
k=5	Cluster 1	8.356	21.656	7.738	15.741	-	15.742	4.010	4.010	4.010	7.949	6.923	6.923
	Cluster 2	5.130	20.929	6.451	15.927	-	15.928	3.684	3.684	3.684	4.494	4.610	4.610
	Cluster 3	14.783	13.839	21.767	16.890	-	16.890	3.961	3.961	3.961	20.252	21.194	21.194
	Cluster 4	4.069	7.129	16.351	14.845	-	14.844	4.192	4.192	4.192	18.749	18.758	18.758
	Cluster 5	19.080	19.244	21.274	15.399	-	15.399	4.832	4.832	4.832	9.562	12.093	12.093

Sampling = 10 longitudes, latitudes, altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	18.014	18.058	18.058	-	-	-	3.662	3.662	3.662	18.096	18.110	18.110
	Cluster 2	18.881	18.943	18.943	-	-	-	4.586	4.586	4.586	18.973	18.994	18.994
k=3	Cluster 1	17.310	19.169	5.836	-	-	-	3.627	3.627	3.627	22.833	22.827	22.827
	Cluster 2	19.224	23.006	15.332	-	-	-	4.452	4.452	4.452	18.821	18.841	18.841
	Cluster 3	15.427	12.239	12.732	-	-	-	22.484	22.484	22.484	23.208	23.220	23.220
k=4	Cluster 1	5.220	22.425	5.227	-	-	-	3.799	3.799	3.799	22.867	22.861	22.861
	Cluster 2	7.209	12.323	10.802	-	-	-	23.138	23.138	23.138	22.957	22.978	22.978
	Cluster 3	17.420	21.937	10.892	-	-	-	5.429	5.429	5.429	22.947	22.953	22.953
	Cluster 4	17.744	6.227	12.551	-	-	-	22.733	22.733	22.733	22.898	22.884	22.884
k=5	Cluster 1	4.961	23.126	23.312	-	-	-	3.347	3.347	3.347	23.291	22.152	22.152
	Cluster 2	12.026	12.545	19.381	-	-	-	23.165	23.165	23.165	22.957	22.978	22.978
	Cluster 3	16.327	19.546	23.030	-	-	-	5.429	5.429	5.429	22.947	22.953	22.953
	Cluster 4	22.918	6.431	21.401	-	-	-	22.509	22.509	22.509	22.690	22.905	22.905
	Cluster 5	17.784	8.089	14.677	-	-	-	7.785	7.785	7.785	20.708	23.491	23.491

Sampling = 25 longitudes, latitudes, altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	18.483	19.318	18.483	-	-	-	3.912	3.912	3.912	18.434	18.483	18.483
	Cluster 2	20.347	20.270	20.347	-	-	-	4.888	4.888	4.888	20.317	20.347	20.347
k=3	Cluster 1	7.584	7.335	21.626	-	-	-	3.872	3.872	3.872	24.309	24.309	24.309
	Cluster 2	12.919	16.078	24.677	-	-	-	4.735	4.735	4.735	20.460	20.419	20.419
	Cluster 3	18.031	15.020	14.944	-	-	-	23.954	23.954	23.954	24.663	24.663	24.663
k=4	Cluster 1	24.579	6.119	6.809	-	-	-	24.628	24.628	24.628	24.351	24.351	24.351
	Cluster 2	24.474	19.044	14.374	-	-	-	5.470	5.470	5.470	24.489	24.489	24.489
	Cluster 3	24.669	7.846	12.436	-	-	-	24.430	24.430	24.430	24.380	24.388	24.388
	Cluster 4	23.704	19.485	15.446	-	-	-	24.677	24.677	24.677	24.107	24.104	24.104
k=5	Cluster 1	25.153	24.788	7.072	-	-	-	24.964	24.964	24.964	24.426	24.215	24.215
	Cluster 2	15.017	16.332	8.607	-	-	-	24.782	24.782	24.782	24.489	24.489	24.489
	Cluster 3	13.897	20.072	13.353	-	-	-	5.214	5.214	5.214	24.380	24.388	24.388
	Cluster 4	7.671	6.216	13.028	-	-	-	24.107	24.107	24.107	23.878	23.875	23.875
	Cluster 5	11.543	8.701	15.987	-	-	-	6.976	6.976	6.976	24.340	24.839	24.839

Sampling = 50 longitudes, latitudes, altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	18.923	19.751	18.923	-	-	-	3.996	3.996	3.996	18.874	18.923	18.923
	Cluster 2	20.787	20.735	20.787	-	-	-	4.989	4.989	4.989	20.762	20.787	20.787
k=3	Cluster 1	19.572	7.479	21.958	-	-	-	3.955	3.955	3.955	24.813	24.813	24.813
	Cluster 2	8.133	16.316	25.146	-	-	-	4.835	4.835	4.835	20.822	20.789	20.789
	Cluster 3	24.837	15.390	14.857	-	-	-	24.433	24.433	24.433	25.091	25.093	25.093
k=4	Cluster 1	9.226	24.814	24.062	-	-	-	25.126	25.126	25.126	24.859	24.859	24.859
	Cluster 2	13.384	25.179	22.559	-	-	-	5.585	5.585	5.585	24.883	24.886	24.886
	Cluster 3	14.419	20.043	25.126	-	-	-	24.919	24.919	24.919	24.872	24.873	24.873
	Cluster 4	13.511	21.977	23.828	-	-	-	25.165	25.165	25.165	24.593	24.593	24.593
k=5	Cluster 1	25.010	6.184	10.551	-	-	-	25.126	25.126	25.126	24.433	24.273	24.273
	Cluster 2	24.912	15.927	18.167	-	-	-	5.530	5.530	5.530	24.883	24.886	24.886
	Cluster 3	25.050	5.135	25.025	-	-	-	24.980	24.980	24.980	24.872	24.873	24.873
	Cluster 4	22.162	6.885	24.721	-	-	-	6.266	6.266	6.266	24.355	24.355	24.355
	Cluster 5	22.710	21.858	23.317	-	-	-	25.864	25.864	25.864	25.742	25.874	25.874

Πίνακας 4.14: RMSE απόσταση του κεντροειδούς από τα μέλη της ομάδας για Διανυσματοποίηση σε 3D Τροχιά με Χρήση Παρεμβολής, του συνόλου δεδομένων.

Οι ίδιοι πίνακες, που δίνουν τη μέση απόσταση του κεντροειδούς από τα μέλη της ομάδας, όπως αυτά έτρεξαν για το συνοπτικό σύνολο δεδομένων, δίνονται στη συνέχεια.

Sampling = 5 longitudes, latitudes, altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	12.387	10.948	15.606	3.367	3.367	15.601	3.271	3.271	3.271	15.888	15.810	15.810
	Cluster 2	14.560	14.746	15.751	4.100	4.100	16.455	3.999	3.999	3.999	16.358	16.210	16.210
k=3	Cluster 1	7.724	7.765	14.536	12.361	12.943	15.597	3.233	3.233	3.233	19.118	19.086	19.086
	Cluster 2	12.712	12.800	19.980	13.432	21.827	16.029	3.857	3.857	3.857	16.026	15.886	15.886
	Cluster 3	17.042	17.187	13.639	16.288	4.217	16.994	22.657	22.657	22.657	21.458	21.426	21.426
k=4	Cluster 1	12.786	18.932	8.424	15.825	15.826	22.438	2.856	2.856	2.856	6.785	6.901	6.901
	Cluster 2	5.443	7.485	7.072	16.029	16.029	21.822	3.857	3.857	3.857	12.316	12.364	12.364
	Cluster 3	14.714	21.657	21.707	17.006	17.005	19.276	21.798	21.798	21.798	11.581	11.644	11.644
	Cluster 4	4.468	8.293	17.312	15.069	15.068	12.416	4.208	4.208	4.208	12.074	11.993	11.993
k=5	Cluster 1	21.590	20.567	8.724	15.826	15.826	15.826	4.669	2.856	2.856	6.785	6.901	6.901
	Cluster 2	20.768	7.894	7.796	16.029	16.023	16.033	3.746	3.976	3.976	18.152	18.570	18.570
	Cluster 3	5.382	21.109	13.951	17.005	17.009	17.004	4.053	4.892	4.892	11.581	11.644	11.644
	Cluster 4	8.458	4.015	4.045	14.927	14.926	14.927	4.064	4.064	4.064	21.067	21.052	21.052
	Cluster 5	15.127	17.487	20.092	15.479	15.477	15.480	4.900	7.222	7.222	12.538	12.459	12.459

Sampling = 10 longitudes, latitudes, altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	17.836	17.669	17.652	-	-	-	3.742	3.742	3.742	17.758	17.669	17.669
	Cluster 2	18.582	18.375	18.350	-	-	-	4.677	4.677	4.677	18.482	18.375	18.375
k=3	Cluster 1	5.742	5.990	22.816	-	-	-	3.705	3.705	3.705	22.768	22.770	22.770
	Cluster 2	16.860	16.361	16.668	-	-	-	4.539	4.539	4.539	18.345	18.241	18.241
	Cluster 3	10.163	11.577	23.326	-	-	-	22.825	22.825	22.825	23.368	23.373	23.373
k=4	Cluster 1	8.224	7.509	15.809	-	-	-	3.324	3.324	3.324	22.809	22.810	22.810
	Cluster 2	15.681	23.211	12.422	-	-	-	4.482	4.482	4.482	23.202	23.219	23.219
	Cluster 3	6.291	5.529	13.710	-	-	-	5.044	5.044	5.044	23.080	23.063	23.063
	Cluster 4	11.544	21.454	6.174	-	-	-	5.984	5.984	5.984	23.189	23.209	23.209
k=5	Cluster 1	22.871	23.335	20.240	-	-	-	3.423	3.423	3.423	23.496	23.462	23.462
	Cluster 2	22.871	23.246	13.325	-	-	-	22.749	22.749	22.749	23.202	23.219	23.219
	Cluster 3	23.743	22.934	22.820	-	-	-	5.044	5.044	5.044	23.080	23.063	23.063
	Cluster 4	23.079	21.884	5.584	-	-	-	4.833	4.833	4.833	22.973	22.993	22.993
	Cluster 5	16.368	18.395	23.950	-	-	-	23.632	23.632	23.632	20.013	20.533	20.533

Sampling = 25 longitudes, latitudes, altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	18.968	18.921	18.921	-	-	-	4.001	4.001	4.001	19.014	18.921	18.921
	Cluster 2	19.705	19.627	19.627	-	-	-	4.988	4.988	4.988	19.811	19.627	19.627
k=3	Cluster 1	7.207	7.028	6.932	-	-	-	3.961	3.961	3.961	24.137	24.257	24.257
	Cluster 2	17.889	17.801	17.826	-	-	-	4.834	4.834	4.834	19.678	19.504	19.504
	Cluster 3	11.376	11.405	11.433	-	-	-	24.171	24.171	24.171	24.894	24.922	24.922
k=4	Cluster 1	17.622	16.245	7.563	-	-	-	3.557	3.557	3.557	24.195	24.310	24.310
	Cluster 2	10.284	16.075	13.791	-	-	-	4.746	4.746	4.746	24.790	24.833	24.833
	Cluster 3	17.217	13.792	12.182	-	-	-	5.315	5.315	5.315	24.553	24.508	24.508
	Cluster 4	6.344	7.199	12.094	-	-	-	6.456	6.456	6.456	24.741	24.743	24.743
k=5	Cluster 1	22.035	20.754	7.294	-	-	-	3.650	3.650	3.650	24.782	24.914	24.914
	Cluster 2	18.924	24.892	7.658	-	-	-	24.304	24.304	24.304	24.790	24.833	24.833
	Cluster 3	24.854	8.305	23.962	-	-	-	5.315	5.315	5.315	24.553	24.508	24.508
	Cluster 4	20.167	19.053	22.988	-	-	-	5.150	5.150	5.150	24.499	24.503	24.503
	Cluster 5	25.461	22.654	24.909	-	-	-	25.277	25.277	25.277	22.837	22.844	22.844

Sampling = 50 longitudes, latitudes, altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	19.278	19.278	19.278	-	-	-	4.085	4.085	4.085	19.441	19.278	19.278
	Cluster 2	19.983	19.983	19.983	-	-	-	5.091	5.091	5.091	20.266	19.983	19.983
k=3	Cluster 1	24.105	21.799	24.430	-	-	-	4.045	4.045	4.045	24.579	24.754	24.754
	Cluster 2	16.796	25.383	17.268	-	-	-	4.934	4.934	4.934	20.131	19.857	19.857
	Cluster 3	25.356	14.209	25.363	-	-	-	24.647	24.647	24.647	25.385	25.420	25.420
k=4	Cluster 1	6.145	7.934	21.848	-	-	-	25.398	25.398	25.398	24.642	24.809	24.809
	Cluster 2	8.274	14.026	25.385	-	-	-	5.814	5.814	5.814	25.273	25.328	25.328
	Cluster 3	15.392	12.648	12.238	-	-	-	25.157	25.157	25.157	25.081	25.024	25.024
	Cluster 4	7.871	12.423	20.161	-	-	-	25.289	25.289	25.289	25.232	25.244	25.244
k=5	Cluster 1	5.937	6.092	8.719	-	-	-	25.398	25.398	25.398	24.835	25.457	25.457
	Cluster 2	25.052	7.502	7.716	-	-	-	5.820	5.820	5.820	25.273	25.328	25.328
	Cluster 3	12.173	20.950	15.563	-	-	-	25.216	25.216	25.216	25.081	25.024	25.024
	Cluster 4	23.052	20.420	8.969	-	-	-	6.370	6.370	6.370	24.984	24.996	24.996
	Cluster 5	25.565	20.198	24.847	-	-	-	26.004	26.004	26.004	24.426	23.162	23.162

Πίνακας 4.15: RMSE απόσταση του κεντροειδούς από τα μέλη της ομάδας για Διανυσματοποίηση σε 3D Τροχιά με Χρήση Παρεμβολής, του συνοπτικού συνόλου δεδομένων.

Παρατηρούμε και για τα δύο σύνολα δεδομένων ότι η μέση απόσταση του κεντροειδούς από τα μέλη της ομάδας αυξάνεται καθώς αυξάνεται το μέγεθος του διανύσματος (για τον ίδιο αριθμό ομάδων). Επίσης οι

μικρότερες αποστάσεις παρατηρούνται στην περίπτωση του Power Iteration Clustering αλγορίθμου (και μάλιστα στην περίπτωση του συνοπτικού συνόλου δεδομένων), γεγονός που υποδεικνύει ότι ο αλγόριθμος αυτός δημιουργεί ομάδες πιο ομογενείς. Οι τιμές του RMSE όπως προκύπτουν από την ομαδοποίηση των αλγορίθμων K-means και Bisecting K-means είναι αρκετά υψηλές, επομένως οι αλγόριθμοι αυτοί δε δημιουργούν ομογενείς ομάδες. Ο αλγόριθμος Gaussian Mixture Model αποτυγχάνει για την περίπτωση του μεγέθους διανύσματος μεγαλύτερου του 5, ενώ για μέγεθος ίσο με 5 οι ομάδες που δημιουργεί είναι πιο ομογενείς στο συνοπτικό σύνολο δεδομένων συγκριτικά με το συνολικό.

Τέλος παραθέτουμε έναν πίνακα με τους χρόνους εκτέλεσης καθενός αλγορίθμου για τις διάφορες τιμές των παραμέτρων που μελετήσαμε, όπως έτρεξαν για το συνολικό σύνολο δεδομένων.

Sampling = 5 longitudes, latitudes, altitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.031	0.035	0.036	0.032	0.038	0.032	0.028	0.049	0.050	0.020	0.030	0.041
k=3	0.033	0.036	0.035	0.032	0.031	0.028	0.030	0.046	0.050	0.023	0.043	0.062
k=4	0.040	0.038	0.043	0.037	0.045	0.039	0.026	0.056	0.051	0.026	0.044	0.069
k=5	0.039	0.043	0.042	0.046	-	0.046	0.025	0.051	0.047	0.028	0.055	0.084

Sampling = 10 longitudes, latitudes, altitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.034	0.035	0.029	-	-	-	0.026	0.049	0.050	0.026	0.030	0.044
k=3	0.039	0.042	0.039	-	-	-	0.025	0.050	0.048	0.023	0.043	0.063
k=4	0.040	0.041	0.035	-	-	-	0.027	0.050	0.051	0.029	0.041	0.069
k=5	0.038	0.039	0.042	-	-	-	0.031	0.048	0.047	0.033	0.055	0.092

Sampling = 25 longitudes, latitudes, altitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.035	0.036	0.033	-	-	-	0.025	0.052	0.046	0.029	0.031	0.045
k=3	0.039	0.041	0.038	-	-	-	0.029	0.052	0.049	0.026	0.047	0.073
k=4	0.034	0.048	0.046	-	-	-	0.025	0.046	0.050	0.027	0.048	0.068
k=5	0.035	0.043	0.045	-	-	-	0.028	0.047	0.046	0.036	0.060	0.092

Sampling = 50 longitudes, latitudes, altitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.036	0.037	0.036	-	-	-	0.026	0.047	0.047	0.027	0.034	0.056
k=3	0.039	0.047	0.053	-	-	-	0.027	0.049	0.049	0.027	0.052	0.072
k=4	0.054	0.057	0.054	-	-	-	0.027	0.047	0.047	0.031	0.049	0.074
k=5	0.054	0.054	0.049	-	-	-	0.028	0.048	0.048	0.040	0.063	0.096

Πίνακας 4.16: Χρόνος εκτέλεσης (sec) για Διανυσματοποίηση σε 3D Τροχιά με Χρήση Παρεμβολής, του συνόλου δεδομένων.

Οι ίδιοι πίνακες, που δίνουν το χρόνο εκτέλεσης των αλγορίθμων, όπως αυτοί έτρεξαν για το συνοπτικό σύνολο δεδομένων, δίνονται στη συνέχεια.

Sampling = 5 longitudes, latitudes, altitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.029	0.032	0.029	0.034	0.029	0.025	0.027	0.052	0.048	0.026	0.031	0.044
k=3	0.032	0.029	0.030	0.035	0.048	0.030	0.024	0.054	0.050	0.028	0.043	0.064
k=4	0.039	0.034	0.035	0.038	0.040	0.040	0.025	0.051	0.050	0.025	0.041	0.065
k=5	0.032	0.037	0.034	0.043	0.053	0.040	0.026	0.049	0.048	0.028	0.054	0.088

Sampling = 10 longitudes, latitudes, altitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.029	0.027	0.031	-	-	-	0.029	0.048	0.044	0.029	0.029	0.042
k=3	0.034	0.032	0.033	-	-	-	0.028	0.044	0.046	0.024	0.041	0.067
k=4	0.033	0.036	0.038	-	-	-	0.029	0.046	0.046	0.024	0.043	0.062
k=5	0.041	0.040	0.034	-	-	-	0.026	0.049	0.043	0.027	0.055	0.088

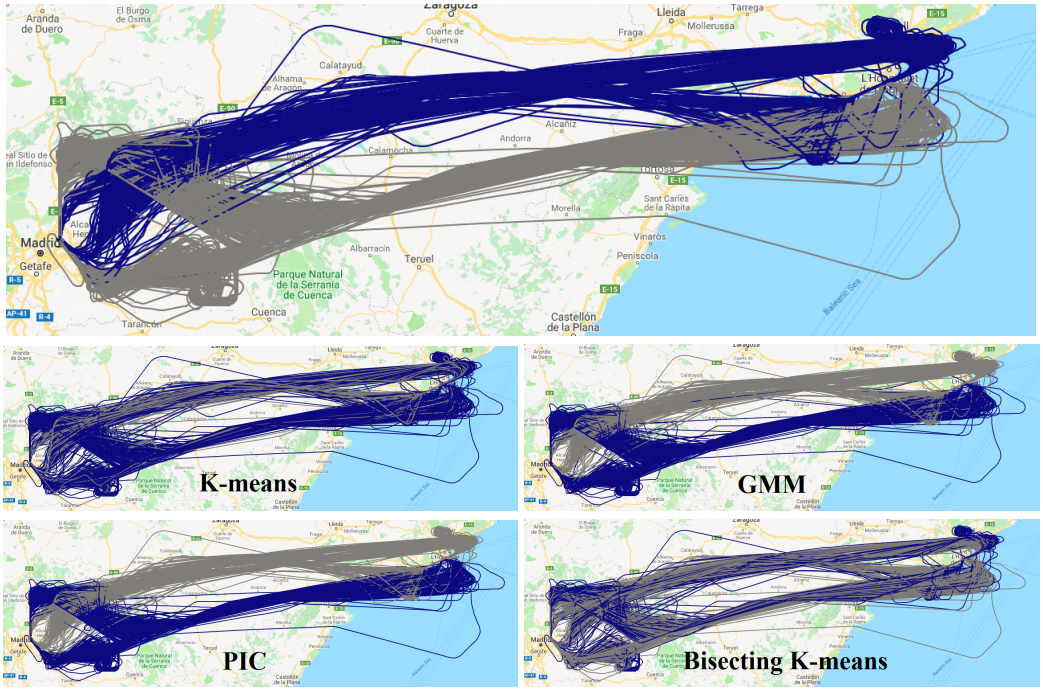
Sampling = 25 longitudes, latitudes, altitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.033	0.035	0.034	-	-	-	0.024	0.047	0.048	0.025	0.030	0.041
k=3	0.032	0.040	0.032	-	-	-	0.030	0.045	0.047	0.025	0.047	0.066
k=4	0.041	0.043	0.040	-	-	-	0.026	0.048	0.046	0.026	0.048	0.070
k=5	0.038	0.046	0.043	-	-	-	0.026	0.049	0.052	0.032	0.062	0.092

Sampling = 50 longitudes, latitudes, altitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.033	0.034	0.039	-	-	-	0.027	0.047	0.050	0.026	0.034	0.050
k=3	0.046	0.037	0.040	-	-	-	0.028	0.048	0.046	0.028	0.056	0.078
k=4	0.043	0.046	0.043	-	-	-	0.029	0.047	0.049	0.029	0.052	0.074
k=5	0.045	0.053	0.044	-	-	-	0.026	0.050	0.047	0.035	0.064	0.094

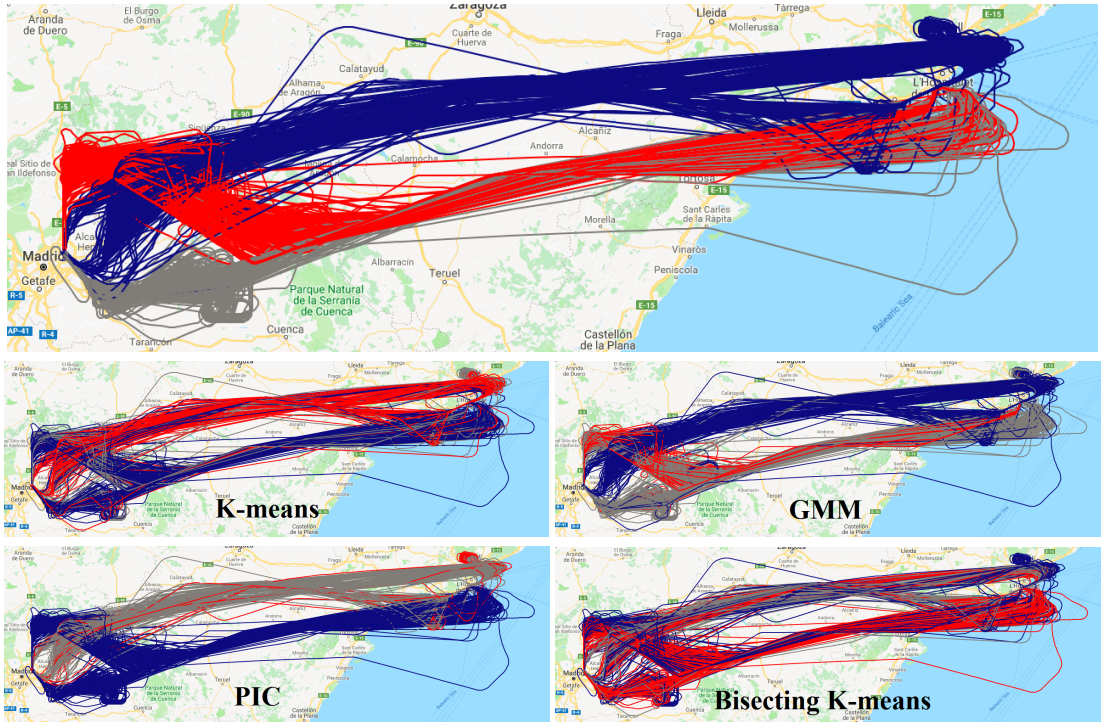
Πίνακας 4.17: Χρόνος εκτέλεσης (sec) για Διανυσματοποίηση σε 3D Τροχιά με Χρήση Παρεμβολής, του συνοπτικού συνόλου δεδομένων.

Οι χρόνοι εκτέλεσης δε φαίνεται να διαφέρουν ιδιαίτερα από αλγόριθμο σε αλγόριθμο. Παρατηρείται ότι η αύξηση του αριθμού των επαναλήψεων αυξάνει το χρόνο εκτέλεσης για την περίπτωση των αλγορίθμων Power Iteration Clustering και Bisecting K-means, ενώ οι αλγόριθμοι K-means και Gaussian Mixture Model δε φαίνεται να επηρεάζονται με κάποιο συστηματικό τρόπο. Ακόμη η αύξηση του πλήθους των ομάδων που δημιουργούν οι αλγόριθμοι δε φαίνεται να επηρεάζει με κάποιο συστηματικό τρόπο το χρόνο εκτέλεσης των αλγορίθμων.

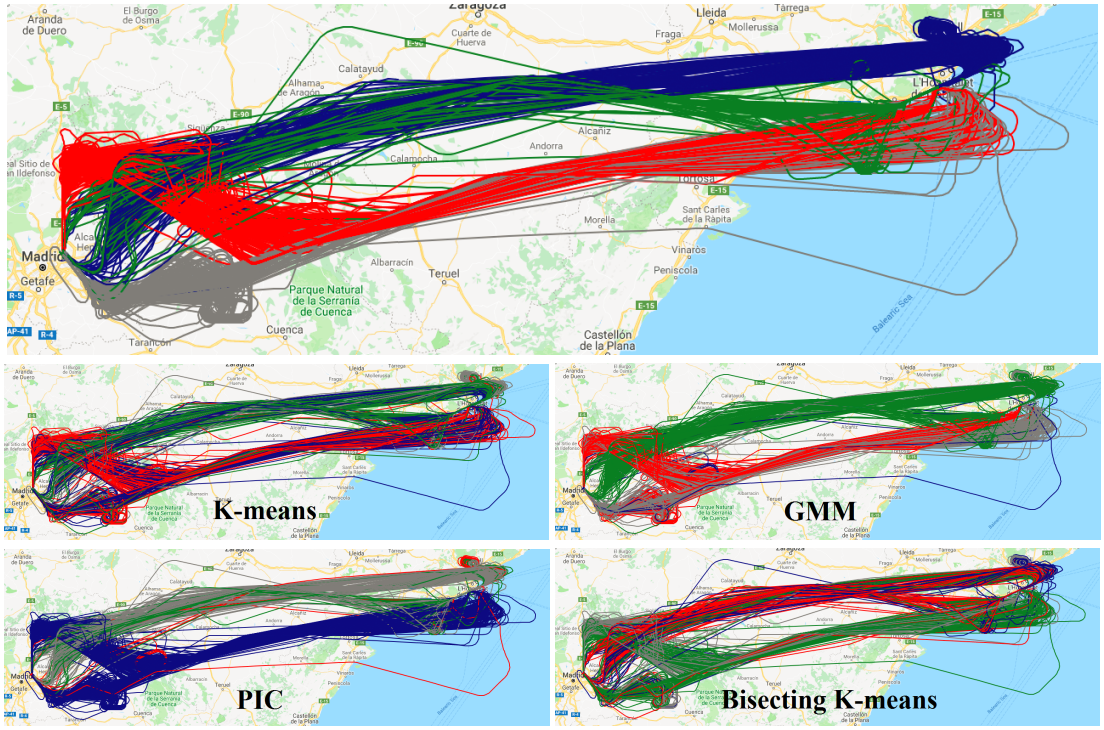
Στη συνέχεια απεικονίζουμε τις καλύτερες ομαδοποιήσεις ανά αριθμό ομάδων για καθέναν από τους αλγορίθμους σε σύγκριση με το ground truth.



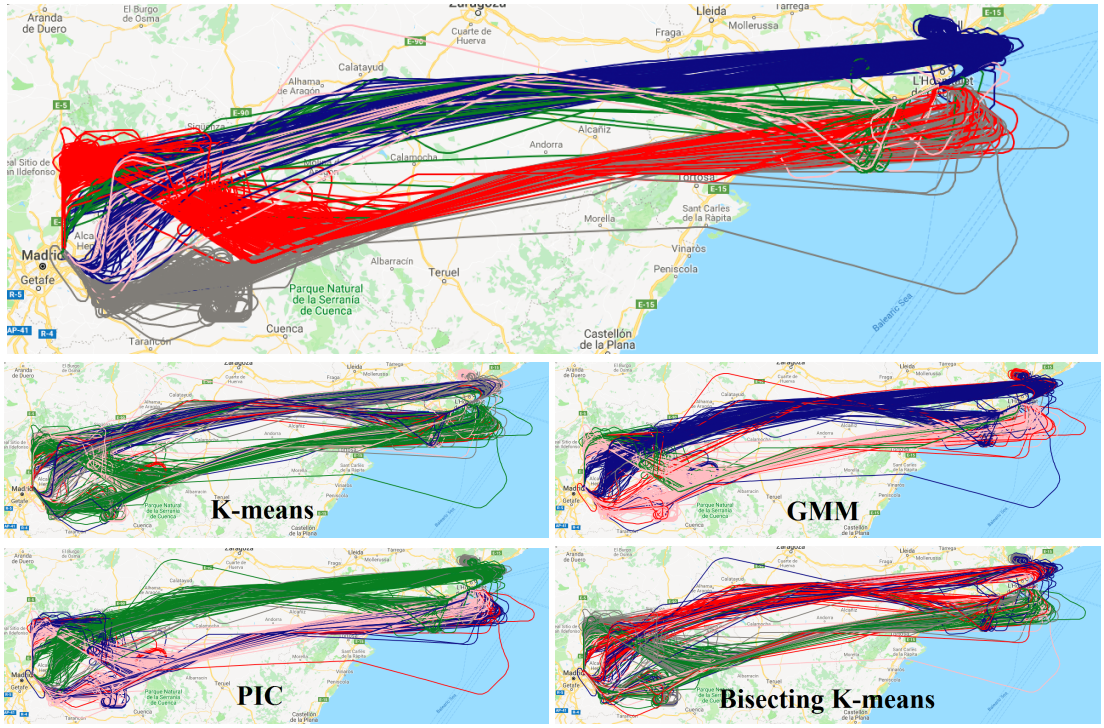
Πίνακας 4.18: Οπτική σύγκριση ομαδοποίησης αλγορίθμων με ground truth για 2 ομάδες.



Πίνακας 4.19: Οπτική σύγκριση ομαδοποίησης αλγορίθμων με ground truth για 3 ομάδες.



Πίνακας 4.20: Οπτική σύγκριση ομαδοποίησης αλγορίθμων με ground truth για 4 ομάδες.



Πίνακας 4.21: Οπτική σύγκριση ομαδοποίησης αλγορίθμων με ground truth για 5 ομάδες.

Κάτι που επιβεβαιώνεται και οπτικά είναι πως οι αλγόριθμοι δίνουν καλύτερα αποτελέσματα στη περίπτωση των δύο ομάδων. Όσο ο αριθμός των ομάδων αυξάνεται, παρατηρούμε σημαντική διαφοροποίηση σε σχέση με το ground truth. Επίσης, η ομαδοποίηση που επιτυγχάνουν οι αλγόριθμοι Power Iteration Clustering και Gaussian Mixture Model είναι καλύτερη συγκριτικά με αυτή των αλγορίθμων K-means και Bisecting K-means για την περίπτωση των δύο ομάδων.

4.2.3 Διανυσματοποίηση σε 3D Κανονικοποιημένη και Ευθυγραμμισμένη Τροχιά

Όπως αναφέραμε παραπάνω, η διανυσματοποίηση αυτή έγινε στα δύο σύνολα δεδομένων αφού όλες οι τροχιές ευθυγραμμίστηκαν ώστε να ξεκινούν από τη χρονική στιγμή 0 και κανονικοποιήθηκαν σύμφωνα με τη διάρκεια της μεγαλύτερης πτήσης. Για κάθε μία από τις μεταβλητές γεωγραφικό μήκος, γεωγραφικό πλάτος και ύψος κρατήσαμε 5, 10, 25 και 50 σημεία, όπως περιγράψαμε αναλυτικά στην Παράγραφο 4.1.2.

Η είσοδος που δέχτηκαν οι αλγόριθμοι K-means, Gaussian Mixture Model και Bisecting K-means, είναι ένας πίνακας με τόσες γραμμές όσες και οι πτήσεις του συνόλου δεδομένων και τόσες στήλες όσες ο τριπλάσιος αριθμός του μεγέθους του διανύσματος που επιλέξαμε. Για παράδειγμα στην περίπτωση που επιλέξαμε 5 σημεία, τότε για την κάθε πτήση κρατήσαμε 15 σημεία, όπου τα 5 πρώτα αποτελούν το γεωγραφικό μήκος, τα 5 επόμενα το γεωγραφικό πλάτος και τα 5 τελευταία το ύψος (όπως αυτά προέκυψαν από την εφαρμογή της παρεμβολής).

Η είσοδος που δέχτηκε ο αλγόριθμος Power Iteration Clustering είναι ένας πίνακας με 3 στήλες, όπου η πρώτη στήλη υποδεικνύει την πτήση i, η δεύτερη στήλη υποδεικνύει την πτήση j και η τρίτη στήλη υποδεικνύει την απόσταση των προαναφερθέντων πτήσεων (με $i \leq j$). Σημειώνουμε ότι η απόσταση αυτή υπολογίστηκε κάνοντας μετατροπή των πολικών συντεταγμένων σε καρτεσιανές και εφαρμόζοντας την ευκλείδεια απόσταση.

Για τους κώδικες που χρησιμοποιήθηκαν για τη δημιουργία των συνόλων δεδομένων που δέχονται σαν είσοδο οι αλγόριθμοι ομαδοποίησης, παραπέμπουμε στο Παράρτημα Γ'.

Στη συνέχεια τρέξαμε τους αλγορίθμους ομαδοποίησης για όλες τις περιπτώσεις του μεγέθους διανύσματος (οι κώδικες των αλγορίθμων ομαδοποίησης παραμένουν ίδιοι).

Τα αποτελέσματα της αξιολόγησης των αλγορίθμων για καθεμία από τις παραμέτρους για τις οποίες εξετάστηκαν (πλήθος ομάδων k και μέγιστο πλήθος επαναλήψεων *MaxIterations*) και για τα δύο σύνολα δεδομένων είναι συγκεντρωμένα στους πίνακες που ακολουθούν.

Οι πίνακες που δίνονται στη συνέχεια περιλαμβάνουν τις τιμές των Accuracy, Precision και F1 Score που αξιολογούν τους αλγορίθμους που έτρεξαν σε ολόκληρο το σύνολο δεδομένων, σε σχέση με το *ground truth*.

Sampling = 5 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.822	0.822	0.822	0.529	0.989	0.519	0.996	0.996	0.996	0.822	0.822	0.822
	Precision	0.914	0.914	0.914	0.513	0.997	0.667	0.991	0.991	0.991	0.914	0.914	0.914
	F1-Score	0.797	0.797	0.797	0.669	0.997	0.102	0.996	0.996	0.996	0.797	0.797	0.797
k=3	Accuracy	0.702	0.730	0.730	0.724	0.431	0.759	0.752	0.752	0.752	0.684	0.684	0.684
	Precision	0.674	0.706	0.706	0.627	0.377	0.628	0.534	0.534	0.534	0.679	0.679	0.679
	F1-Score	0.654	0.706	0.706	0.578	0.347	0.576	0.565	0.565	0.565	0.665	0.665	0.665
k=4	Accuracy	0.577	0.634	0.634	0.469	0.468	-	0.648	0.648	0.648	0.503	0.503	0.503
	Precision	0.559	0.560	0.560	0.456	0.417	-	0.636	0.636	0.636	0.430	0.430	0.430
	F1-Score	0.561	0.530	0.530	0.320	0.310	-	0.607	0.607	0.607	0.437	0.437	0.437
k=5	Accuracy	0.548	0.562	0.504	0.568	0.442	-	0.565	0.565	0.565	0.479	0.479	0.479
	Precision	0.482	0.469	0.364	0.424	0.347	-	0.505	0.505	0.505	0.422	0.422	0.422
	F1-Score	0.470	0.442	0.361	0.347	0.249	-	0.476	0.476	0.476	0.399	0.399	0.399

Sampling = 10 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.832	0.832	0.832	-	-	-	0.995	0.995	0.995	0.832	0.832	0.832
	Precision	0.900	0.900	0.900	-	-	-	0.991	0.991	0.991	0.900	0.900	0.900
	F1-Score	0.814	0.815	0.815	-	-	-	0.995	0.995	0.995	0.815	0.815	0.815
k=3	Accuracy	0.680	0.688	0.688	-	-	-	0.769	0.769	0.769	0.569	0.569	0.569
	Precision	0.671	0.672	0.672	-	-	-	0.587	0.587	0.587	0.597	0.597	0.597
	F1-Score	0.659	0.659	0.659	-	-	-	0.576	0.576	0.576	0.560	0.560	0.560
k=4	Accuracy	0.476	0.468	0.565	-	-	-	0.604	0.604	0.604	0.465	0.465	0.465
	Precision	0.405	0.499	0.524	-	-	-	0.585	0.585	0.585	0.497	0.497	0.497
	F1-Score	0.413	0.467	0.476	-	-	-	0.523	0.523	0.523	0.465	0.465	0.465
k=5	Accuracy	0.448	0.521	0.486	-	-	-	0.591	0.591	0.591	0.459	0.459	0.459
	Precision	0.413	0.484	0.427	-	-	-	0.480	0.480	0.480	0.478	0.478	0.478
	F1-Score	0.388	0.438	0.397	-	-	-	0.439	0.439	0.439	0.394	0.394	0.394

Sampling = 25 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.835	0.835	0.835	-	-	-	0.995	0.995	0.995	0.835	0.835	0.835
	Precision	0.905	0.905	0.905	-	-	-	0.991	0.991	0.991	0.905	0.905	0.905
	F1-Score	0.817	0.817	0.817	-	-	-	0.995	0.995	0.995	0.817	0.817	0.817
k=3	Accuracy	0.659	0.693	0.686	-	-	-	0.767	0.767	0.767	0.539	0.539	0.539
	Precision	0.482	0.680	0.676	-	-	-	0.581	0.581	0.581	0.579	0.579	0.579
	F1-Score	0.505	0.673	0.671	-	-	-	0.575	0.575	0.575	0.534	0.534	0.534
k=4	Accuracy	0.549	0.521	0.493	-	-	-	0.637	0.637	0.637	0.489	0.489	0.489
	Precision	0.543	0.518	0.513	-	-	-	0.619	0.619	0.619	0.512	0.512	0.512
	F1-Score	0.534	0.493	0.488	-	-	-	0.499	0.499	0.499	0.485	0.485	0.485
k=5	Accuracy	0.525	0.538	0.535	-	-	-	0.593	0.593	0.593	0.473	0.473	0.473
	Precision	0.486	0.448	0.447	-	-	-	0.478	0.478	0.478	0.446	0.446	0.446
	F1-Score	0.440	0.436	0.434	-	-	-	0.434	0.434	0.434	0.400	0.400	0.400

Sampling = 50 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.834	0.834	0.834	-	-	-	0.995	0.995	0.995	0.834	0.834	0.834
	Precision	0.905	0.905	0.905	-	-	-	0.991	0.991	0.991	0.905	0.905	0.905
	F1-Score	0.816	0.816	0.816	-	-	-	0.995	0.995	0.995	0.816	0.816	0.816
k=3	Accuracy	0.678	0.694	0.682	-	-	-	0.767	0.767	0.767	0.544	0.538	0.538
	Precision	0.677	0.679	0.674	-	-	-	0.581	0.581	0.581	0.577	0.579	0.579
	F1-Score	0.671	0.667	0.669	-	-	-	0.575	0.575	0.575	0.538	0.534	0.534
k=4	Accuracy	0.504	0.501	0.491	-	-	-	0.636	0.636	0.636	0.485	0.486	0.486
	Precision	0.521	0.519	0.514	-	-	-	0.618	0.618	0.618	0.510	0.509	0.509
	F1-Score	0.499	0.494	0.484	-	-	-	0.498	0.498	0.498	0.481	0.481	0.481
k=5	Accuracy	0.563	0.499	0.505	-	-	-	0.593	0.593	0.593	0.470	0.474	0.474
	Precision	0.474	0.465	0.444	-	-	-	0.480	0.480	0.480	0.480	0.477	0.477
	F1-Score	0.459	0.412	0.411	-	-	-	0.435	0.435	0.435	0.400	0.403	0.403

Πίνακας 4.22: Accuracy, Precision και F1 Score(%) για Διανυσματοποίηση σε 3D Κανονικοποιημένη και Ευθυγραμμισμένη Τροχιά με Χρήση Παρεμβολής, του συνόλου δεδομένων.

Οι ίδιοι πίνακες, που δίνουν τα αποτελέσματα αξιολόγησης των αλγορίθμων όπως αυτά έτρεξαν για το συνοπτικό σύνολο δεδομένων, δίνονται στη συνέχεια.

Sampling = 5 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.509	0.500	0.500	0.720	0.533	0.777	0.996	0.996	0.996	0.813	0.813	0.813
	Precision	0.508	0.491	0.491	0.963	0.516	0.987	0.991	0.991	0.991	0.904	0.904	0.904
	F1-Score	0.366	0.362	0.362	0.615	0.659	0.712	0.996	0.996	0.996	0.787	0.787	0.787
k=3	Accuracy	0.481	0.592	0.481	0.759	0.434	0.509	0.630	0.719	0.719	0.688	0.688	0.688
	Precision	0.517	0.595	0.518	0.609	0.320	0.438	0.518	0.522	0.522	0.683	0.683	0.683
	F1-Score	0.469	0.567	0.469	0.579	0.250	0.276	0.513	0.551	0.551	0.670	0.670	0.670
k=4	Accuracy	0.488	0.516	0.515	0.367	0.455	0.467	0.559	0.559	0.559	0.511	0.511	0.511
	Precision	0.465	0.493	0.491	0.534	0.484	0.484	0.428	0.428	0.428	0.437	0.437	0.437
	F1-Score	0.462	0.485	0.483	0.252	0.295	0.312	0.445	0.445	0.445	0.444	0.444	0.444
k=5	Accuracy	0.431	0.438	0.456	-	0.489	0.458	0.502	0.504	0.504	0.487	0.487	0.487
	Precision	0.363	0.411	0.376	-	0.398	0.303	0.451	0.453	0.453	0.427	0.427	0.427
	F1-Score	0.345	0.382	0.362	-	0.287	0.303	0.426	0.428	0.428	0.405	0.405	0.405

Sampling = 10 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.528	0.527	0.528	-	-	-	0.996	0.996	0.996	0.831	0.831	0.831
	Precision	0.530	0.529	0.530	-	-	-	0.993	0.993	0.993	0.855	0.855	0.855
	F1-Score	0.461	0.460	0.461	-	-	-	0.996	0.996	0.996	0.823	0.823	0.823
k=3	Accuracy	0.619	0.612	0.612	-	-	-	0.769	0.769	0.769	0.536	0.536	0.536
	Precision	0.623	0.617	0.617	-	-	-	0.581	0.581	0.581	0.569	0.569	0.569
	F1-Score	0.607	0.599	0.599	-	-	-	0.575	0.575	0.575	0.520	0.520	0.520
k=4	Accuracy	0.554	0.508	0.486	-	-	-	0.621	0.621	0.621	0.473	0.473	0.473
	Precision	0.525	0.497	0.481	-	-	-	0.622	0.622	0.622	0.493	0.493	0.493
	F1-Score	0.525	0.495	0.468	-	-	-	0.474	0.474	0.474	0.475	0.475	0.475
k=5	Accuracy	0.468	0.464	0.451	-	-	-	0.547	0.547	0.547	0.407	0.410	0.410
	Precision	0.410	0.397	0.394	-	-	-	0.470	0.470	0.470	0.379	0.380	0.380
	F1-Score	0.389	0.377	0.370	-	-	-	0.409	0.409	0.409	0.350	0.352	0.352

Sampling = 25 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.528	0.527	0.527	-	-	-	0.995	0.995	0.995	0.838	0.838	0.838
	Precision	0.531	0.531	0.531	-	-	-	0.991	0.991	0.991	0.893	0.893	0.893
	F1-Score	0.456	0.446	0.446	-	-	-	0.995	0.995	0.995	0.824	0.824	0.824
k=3	Accuracy	0.610	0.605	0.604	-	-	-	0.769	0.769	0.769	0.541	0.541	0.541
	Precision	0.616	0.607	0.636	-	-	-	0.555	0.555	0.555	0.579	0.579	0.579
	F1-Score	0.601	0.592	0.606	-	-	-	0.571	0.571	0.571	0.533	0.533	0.533
k=4	Accuracy	0.475	0.440	0.448	-	-	-	0.619	0.619	0.619	0.491	0.491	0.491
	Precision	0.445	0.436	0.427	-	-	-	0.558	0.558	0.558	0.515	0.515	0.515
	F1-Score	0.440	0.422	0.418	-	-	-	0.417	0.417	0.417	0.488	0.488	0.488
k=5	Accuracy	0.447	0.433	0.457	-	-	-	0.638	0.638	0.638	0.423	0.423	0.423
	Precision	0.411	0.421	0.400	-	-	-	0.515	0.515	0.515	0.419	0.419	0.419
	F1-Score	0.383	0.380	0.380	-	-	-	0.406	0.406	0.406	0.377	0.377	0.377

Sampling = 50 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.529	0.529	0.529	-	-	-	0.995	0.995	0.995	0.835	0.835	0.835
	Precision	0.533	0.533	0.533	-	-	-	0.991	0.991	0.991	0.898	0.898	0.898
	F1-Score	0.451	0.451	0.451	-	-	-	0.995	0.995	0.995	0.818	0.818	0.818
k=3	Accuracy	0.565	0.606	0.606	-	-	-	0.769	0.769	0.769	0.545	0.545	0.545
	Precision	0.574	0.607	0.607	-	-	-	0.555	0.555	0.555	0.577	0.577	0.577
	F1-Score	0.556	0.593	0.593	-	-	-	0.571	0.571	0.571	0.538	0.538	0.538
k=4	Accuracy	0.518	0.442	0.441	-	-	-	0.595	0.595	0.595	0.487	0.487	0.487
	Precision	0.492	0.438	0.438	-	-	-	0.587	0.587	0.587	0.512	0.512	0.512
	F1-Score	0.489	0.424	0.423	-	-	-	0.507	0.507	0.507	0.485	0.485	0.485
k=5	Accuracy	0.441	0.418	0.435	-	-	-	0.573	0.573	0.573	0.467	0.467	0.467
	Precision	0.390	0.368	0.418	-	-	-	0.482	0.482	0.482	0.446	0.446	0.446
	F1-Score	0.371	0.346	0.380	-	-	-	0.427	0.427	0.427	0.396	0.396	0.396

Πίνακας 4.23: Accuracy, Precision και F1 Score(%) για Διανυσματοποίηση σε 3D Κανονικοποιημένη και Ευθυγραμμισμένη Τροχιά με Χρήση Παρεμβολής, του συνοπτικού συνόλου δεδομένων.

Παρατηρούμε αρχικά ότι ο αλγόριθμος Gaussian Mixture Model αποτυγχάνει στην περίπτωση για τα μεγέθη των διανυσμάτων μεγαλύτερα του 5, και η ομαδοποίηση που επιτυγχάνει στην περίπτωση του μεγέθους ίσο με 5 δε δίνει υψηλές τιμές στα μέτρα απόδοσης. Σημειώνουμε επίσης ότι οι αλγόριθμοι K-means και Bisecting K-means έχουν παρόμοιες τιμές στα μέτρα απόδοσης στην περίπτωση του συνολικού συνόλου δεδομένων, ενώ ο Bisecting K-means φαίνεται να επιτυγχάνει καλύτερη ομαδοποίηση στην περίπτωση του συνοπτικού συνόλου δεδομένων για τις διάφορες τιμές του πλήθους των ομάδων. Τα μέτρα απόδοσης του αλγορίθμου Power Iteration Clustering είναι τα υψηλότερα συγκριτικά με τους υπόλοιπους αλγορίθμους, ενώ ο αλγόριθμος αυτός επιτυγχάνει την καλύτερη ομαδοποίηση στην περίπτωση των 2 ομάδων όπου τα μέτρα απόδοσης είναι σχεδόν ίσα με τη μονάδα και μειώνονται καθώς το πλήθος των ομάδων αυξάνεται.

Στη συνέχεια δίνονται οι πίνακες που περιλαμβάνουν τη μέση απόσταση του κεντροειδούς από τα μέλη της ομάδας, όπως αυτή υπολογίστηκε για κάθε αλγόριθμο που έτρεξε στο συνολικό σύνολο δεδομένων.

Sampling = 5 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	3.129	3.129	3.129	5.092	1.857	5.022	1.734	1.734	1.734	3.129	3.129	3.129
	Cluster 2	3.465	3.465	3.465	6.866	2.644	7.531	2.625	2.625	2.625	3.465	3.465	3.465
k=3	Cluster 1	2.457	7.211	5.771	5.083	5.083	5.083	1.705	1.705	1.705	1.785	1.785	1.785
	Cluster 2	4.277	6.070	4.128	6.686	6.686	6.686	2.850	2.850	2.850	6.358	6.358	6.358
	Cluster 3	3.489	5.191	6.492	7.105	7.105	7.105	8.707	8.707	8.707	2.973	2.973	2.973
k=4	Cluster 1	6.942	5.500	5.500	5.025	5.025	-	1.485	1.485	1.485	1.595	1.595	1.595
	Cluster 2	4.254	3.768	6.062	6.686	6.686	-	2.956	2.956	2.956	6.363	6.363	6.363
	Cluster 3	6.561	6.486	10.555	7.118	7.118	-	2.216	2.216	2.216	2.797	2.797	2.797
	Cluster 4	2.559	0.275	7.199	5.242	5.242	-	2.377	2.377	2.377	8.245	8.245	8.245
k=5	Cluster 1	2.386	1.620	4.999	5.025	7.596	-	1.481	1.481	1.481	6.225	6.225	6.225
	Cluster 2	6.625	3.652	5.216	6.686	6.053	-	6.080	6.080	6.080	6.173	6.173	6.173
	Cluster 3	3.103	5.045	6.498	7.118	7.604	-	2.213	2.213	2.213	6.595	6.595	6.595
	Cluster 4	5.993	2.591	6.666	5.193	7.738	-	3.011	3.011	3.011	5.187	5.187	5.187
	Cluster 5	3.025	5.769	5.890	5.383	2.337	-	6.987	6.987	6.987	8.408	8.408	8.408

Sampling = 10 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	4.685	4.669	4.669	-	-	-	1.893	1.893	1.893	4.669	4.669	4.669
	Cluster 2	4.160	4.155	4.155	-	-	-	2.858	2.858	2.858	4.155	4.155	4.155
k=3	Cluster 1	2.652	2.952	2.952	-	-	-	1.859	1.859	1.859	2.295	2.295	2.295
	Cluster 2	8.621	7.573	7.573	-	-	-	3.087	3.087	3.087	12.056	12.056	12.056
	Cluster 3	3.428	3.652	3.652	-	-	-	12.939	12.939	12.939	3.781	3.781	3.781
k=4	Cluster 1	2.180	7.096	2.611	-	-	-	13.155	13.155	13.155	14.013	14.013	14.013
	Cluster 2	11.238	5.297	9.275	-	-	-	12.288	12.288	12.288	12.027	12.027	12.027
	Cluster 3	3.046	12.852	3.035	-	-	-	13.074	13.074	13.074	12.784	12.784	12.784
	Cluster 4	14.112	12.462	13.740	-	-	-	12.399	12.399	12.399	12.439	12.439	12.439
k=5	Cluster 1	13.895	1.819	6.423	-	-	-	12.927	12.927	12.927	12.561	12.561	12.561
	Cluster 2	13.004	2.953	4.446	-	-	-	12.306	12.306	12.306	12.164	12.164	12.164
	Cluster 3	2.583	3.806	3.678	-	-	-	13.026	13.026	13.026	13.479	13.479	13.479
	Cluster 4	6.135	2.674	2.665	-	-	-	12.312	12.312	12.312	11.563	11.563	11.563
	Cluster 5	3.213	14.303	14.096	-	-	-	10.874	10.874	10.874	14.154	14.154	14.154

Sampling = 25 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	4.844	4.844	4.844	-	-	-	1.932	1.932	1.932	4.844	4.844	4.844
	Cluster 2	4.146	4.146	4.146	-	-	-	2.870	2.870	2.870	4.146	4.146	4.146
k=3	Cluster 1	4.605	14.219	3.094	-	-	-	1.900	1.900	1.900	2.613	2.613	2.613
	Cluster 2	9.693	12.836	9.227	-	-	-	3.097	3.097	3.097	12.781	12.781	12.781
	Cluster 3	3.957	8.913	3.379	-	-	-	12.765	12.765	12.765	3.741	3.741	3.741
k=4	Cluster 1	2.690	2.556	2.524	-	-	-	13.518	13.518	13.518	14.571	14.571	14.571
	Cluster 2	4.855	12.855	5.156	-	-	-	3.041	3.041	3.041	12.662	12.661	12.661
	Cluster 3	4.973	4.080	3.854	-	-	-	13.829	13.829	13.829	13.637	13.641	13.641
	Cluster 4	4.616	13.047	6.537	-	-	-	10.697	10.697	10.697	13.070	13.070	13.070
k=5	Cluster 1	1.844	2.545	4.854	-	-	-	11.478	11.478	11.478	13.306	13.306	13.306
	Cluster 2	3.002	13.040	5.192	-	-	-	3.036	3.036	3.036	12.662	12.661	12.661
	Cluster 3	3.861	3.305	13.542	-	-	-	13.851	13.851	13.851	4.131	4.131	4.131
	Cluster 4	2.764	12.591	13.828	-	-	-	2.548	2.548	2.548	2.726	2.726	2.726
	Cluster 5	14.806	14.832	3.388	-	-	-	13.690	13.690	13.690	12.667	12.667	12.667

Sampling = 50 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	4.867	4.867	4.867	-	-	-	1.932	1.932	1.932	4.867	4.867	4.867
	Cluster 2	4.164	4.164	4.164	-	-	-	2.874	2.874	2.874	4.164	4.164	4.164
k=3	Cluster 1	2.679	3.162	3.043	-	-	-	1.899	1.899	1.899	2.675	2.601	2.601
	Cluster 2	10.594	7.901	9.512	-	-	-	3.100	3.100	3.100	12.796	12.802	12.802
	Cluster 3	3.215	3.651	3.342	-	-	-	12.809	12.809	12.809	3.760	3.760	3.760
k=4	Cluster 1	6.529	6.546	13.338	-	-	-	13.545	13.545	13.545	14.610	14.613	14.613
	Cluster 2	4.320	4.333	12.855	-	-	-	3.044	3.044	3.044	12.690	12.672	12.672
	Cluster 3	4.070	4.192	13.647	-	-	-	13.860	13.860	13.860	13.652	13.661	13.661
	Cluster 4	3.036	3.022	14.401	-	-	-	10.704	10.704	10.704	13.106	13.108	13.108
k=5	Cluster 1	15.197	13.272	6.291	-	-	-	11.482	11.482	11.482	13.343	13.345	13.345
	Cluster 2	4.163	12.895	12.665	-	-	-	3.039	3.039	3.039	12.690	12.672	12.672
	Cluster 3	13.634	3.628	3.911	-	-	-	13.892	13.892	13.892	14.069	14.069	14.069
	Cluster 4	13.449	12.325	14.113	-	-	-	2.552	2.552	2.552	12.152	12.144	12.144
	Cluster 5	2.480	3.342	14.365	-	-	-	13.691	13.691	13.691	14.753	14.756	14.756

Πίνακας 4.24: RMSE απόσταση του κεντροειδούς από τα μέλη της ομάδας για Διανυσματοποίηση σε 3D Κανονικοποιημένη και Ευθυγραμμισμένη Τροχιά με Χρήση Παρεμβολής, του συνόλου δεδομένων.

Οι ίδιοι πίνακες, που δίνουν τη μέση απόσταση του κεντροειδούς από τα μέλη της ομάδας, όπως αυτά έτρεξαν για το συνοπτικό σύνολο δεδομένων, δίνονται στη συνέχεια.

Sampling = 5 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	12.865	12.842	12.842	5.676	5.676	5.676	1.860	1.860	1.860	3.549	3.549	3.549
	Cluster 2	11.951	12.189	12.189	7.512	7.512	7.512	2.728	2.728	2.728	3.790	3.790	3.790
k=3	Cluster 1	15.118	16.025	15.133	5.664	5.664	7.824	1.816	1.832	1.832	1.924	1.924	1.924
	Cluster 2	13.330	14.039	13.328	7.261	7.261	7.611	2.863	2.845	2.845	6.726	6.726	6.726
	Cluster 3	12.826	11.808	12.826	7.846	7.846	7.813	6.537	6.743	6.743	3.471	3.471	3.471
k=4	Cluster 1	15.621	15.013	15.320	5.673	5.082	5.673	1.665	1.665	1.665	1.801	1.801	1.801
	Cluster 2	11.239	14.976	16.177	7.261	3.696	7.261	2.805	2.805	2.805	6.726	6.726	6.726
	Cluster 3	15.563	11.464	17.330	7.853	8.455	7.853	9.060	9.060	9.060	3.356	3.356	3.356
	Cluster 4	13.752	14.979	11.017	5.672	7.872	5.672	2.382	2.382	2.382	8.284	8.284	8.284
k=5	Cluster 1	14.534	13.979	16.690	-	5.673	5.673	2.040	2.098	2.098	6.571	6.571	6.571
	Cluster 2	15.161	11.099	13.156	-	7.261	7.261	2.684	2.684	2.684	6.370	6.370	6.370
	Cluster 3	12.261	18.275	15.088	-	7.853	7.853	3.322	3.368	3.368	6.839	6.839	6.839
	Cluster 4	16.009	15.354	10.867	-	5.579	5.579	2.287	2.284	2.284	5.558	5.558	5.558
	Cluster 5	15.031	17.393	14.107	-	5.944	5.944	2.118	2.120	2.120	8.664	8.664	8.664

Sampling = 10 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	18.206	18.206	18.207	-	-	-	2.143	2.143	2.143	4.377	4.377	4.377
	Cluster 2	17.452	17.465	17.457	-	-	-	3.058	3.058	3.058	5.177	5.177	5.177
k=3	Cluster 1	18.391	19.522	18.347	-	-	-	2.107	2.107	2.107	2.501	2.501	2.501
	Cluster 2	18.320	16.834	18.298	-	-	-	3.212	3.212	3.212	12.526	12.526	12.526
	Cluster 3	17.109	20.391	17.165	-	-	-	13.908	13.908	13.908	4.998	4.998	4.998
k=4	Cluster 1	18.384	18.123	18.052	-	-	-	11.390	11.390	11.390	14.047	14.047	14.047
	Cluster 2	18.437	18.391	18.107	-	-	-	3.125	3.125	3.125	12.187	12.187	12.187
	Cluster 3	17.019	17.270	16.452	-	-	-	4.082	4.082	4.082	13.322	13.322	13.322
	Cluster 4	16.619	16.942	18.173	-	-	-	2.631	2.631	2.631	12.423	12.423	12.423
k=5	Cluster 1	19.390	19.098	19.149	-	-	-	11.699	11.699	11.699	12.710	12.693	12.693
	Cluster 2	19.755	19.660	16.044	-	-	-	3.125	3.125	3.125	12.187	12.187	12.187
	Cluster 3	17.967	16.173	20.723	-	-	-	13.366	13.366	13.366	4.149	4.149	4.149
	Cluster 4	16.187	18.232	17.663	-	-	-	2.636	2.636	2.636	6.355	6.355	6.355
	Cluster 5	18.285	19.700	19.900	-	-	-	13.365	13.365	13.365	11.637	11.575	11.575

Sampling = 25 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	19.753	19.733	19.733	-	-	-	2.567	2.567	2.567	5.014	5.014	5.014
	Cluster 2	19.095	19.055	19.055	-	-	-	3.459	3.459	3.459	4.864	4.864	4.864
k=3	Cluster 1	20.498	19.802	19.238	-	-	-	2.536	2.536	2.536	3.006	3.006	3.006
	Cluster 2	18.265	20.011	20.066	-	-	-	3.635	3.635	3.635	12.926	12.926	12.926
	Cluster 3	21.227	18.900	18.161	-	-	-	14.939	14.939	14.939	4.554	4.554	4.554
k=4	Cluster 1	20.560	20.199	20.286	-	-	-	2.385	2.385	2.385	14.732	14.732	14.732
	Cluster 2	20.644	19.522	19.363	-	-	-	3.577	3.577	3.577	12.669	12.669	12.669
	Cluster 3	20.699	21.178	21.173	-	-	-	5.816	5.816	5.816	13.900	13.900	13.900
	Cluster 4	18.059	18.507	18.187	-	-	-	8.671	8.671	8.671	13.192	13.192	13.192
k=5	Cluster 1	19.756	18.651	19.532	-	-	-	2.385	2.385	2.385	13.420	13.420	13.420
	Cluster 2	19.835	19.627	20.400	-	-	-	14.327	14.327	14.327	12.669	12.669	12.669
	Cluster 3	20.728	20.776	21.203	-	-	-	3.680	3.680	3.680	4.581	4.581	4.581
	Cluster 4	18.087	19.659	20.422	-	-	-	13.124	13.124	13.124	5.615	5.615	5.615
	Cluster 5	20.277	20.264	18.195	-	-	-	15.323	15.323	15.323	9.292	9.292	9.292

Sampling = 50 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	20.121	20.121	20.121	-	-	-	2.456	2.456	2.456	5.074	5.074	5.074
	Cluster 2	19.433	19.433	19.433	-	-	-	3.372	3.372	3.372	4.742	4.742	4.742
k=3	Cluster 1	19.031	20.956	20.241	-	-	-	2.424	2.424	2.424	3.059	3.059	3.059
	Cluster 2	21.057	18.866	20.507	-	-	-	3.530	3.530	3.530	12.864	12.864	12.864
	Cluster 3	21.011	21.540	19.275	-	-	-	14.826	14.826	14.826	4.434	4.434	4.434
k=4	Cluster 1	20.024	20.266	19.081	-	-	-	14.122	14.122	14.122	14.642	14.642	14.642
	Cluster 2	20.569	19.936	20.780	-	-	-	12.945	12.945	12.945	12.707	12.707	12.707
	Cluster 3	18.777	19.091	21.453	-	-	-	13.960	13.960	13.960	13.733	13.733	13.733
	Cluster 4	19.726	20.266	20.769	-	-	-	13.074	13.074	13.074	13.089	13.089	13.089
k=5	Cluster 1	20.929	20.768	19.146	-	-	-	13.668	13.668	13.668	13.330	13.330	13.330
	Cluster 2	20.946	20.211	20.221	-	-	-	12.940	12.940	12.940	12.707	12.707	12.707
	Cluster 3	19.063	18.880	21.137	-	-	-	13.948	13.948	13.948	4.561	4.561	4.561
	Cluster 4	19.740	20.931	20.002	-	-	-	12.959	12.959	12.959	3.151	3.151	3.151
	Cluster 5	20.979	19.647	20.477	-	-	-	11.272	11.272	11.272	12.773	12.773	12.773

Πίνακας 4.25: RMSE απόσταση του κεντροειδούς από τα μέλη της ομάδας για Διανυσματοποίηση σε 3D Κανονικοποιημένη και Ευθυγραμμισμένη Τροχιά με Χρήση Παρεμβολής, του συνοπτικού συνόλου δεδομένων.

Παρατηρούμε και για τα δύο σύνολα δεδομένων ότι η μέση απόσταση του κεντροειδούς από τα μέλη της ομάδας αυξάνεται καθώς αυξάνεται το μέγεθος του διανύσματος (για τον ίδιο αριθμό ομάδων). Επίσης οι μικρότερες αποστάσεις παρατηρούνται στην περίπτωση του Power Iteration Clustering αλγορίθμου (και για τα δύο σύνολα δεδομένων), γεγονός που υποδεικνύει ότι ο αλγόριθμος αυτός δημιουργεί ομάδες πιο ομογενείς. Οι τιμές του RMSE όπως προκύπτουν από την ομαδοποίηση του αλγορίθμου K-means είναι αισθητά μικρότερες στην περίπτωση του συνολικού συνόλου δεδομένων σε σύγκριση με το συνοπτικό. Αρκετά ομογενείς ομάδες φαίνεται να δημιουργεί και ο αλγόριθμος Bisecting K-means, του οποίου οι τιμές για το RMSE είναι χαμηλές στην περίπτωση των δύο ομάδων και για τα δύο σύνολα δεδομένων. Ο αλγόριθμος Gaussian Mixture Model αποτυγχάνει για την περίπτωση του μεγέθους διανύσματος μεγαλύτερου του 5, ενώ για μέγεθος ίσο με 5 οι ομάδες που δημιουργεί δεν είναι τόσο ομογενείς όσο αυτές που δημιουργούνται από την ομαδοποίηση των αλγορίθμων Power Iteration Clustering και Bisecting K-means.

Τέλος παραθέτουμε έναν πίνακα με τους χρόνους εκτέλεσης καθενός αλγορίθμου για τις διάφορες τιμές των παραμέτρων που μελετήσαμε, όπως έτρεξαν για το συνολικό σύνολο δεδομένων.

Sampling = 5 longitudes / latitudes / altitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.029	0.041	0.028	0.027	0.036	0.033	0.027	0.033	0.046	0.025	0.031	0.044
k=3	0.028	0.033	0.033	0.031	0.030	0.031	0.029	0.037	0.037	0.024	0.043	0.068
k=4	0.028	0.029	0.035	0.036	0.035	-	0.024	0.033	0.036	0.026	0.044	0.068
k=5	0.033	0.029	0.035	0.039	0.045	-	0.028	0.033	0.036	0.028	0.057	0.091

Sampling = 10 longitudes / latitudes / altitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.034	0.031	0.031	-	-	-	0.025	0.044	0.053	0.022	0.031	0.042
k=3	0.032	0.032	0.032	-	-	-	0.025	0.042	0.052	0.025	0.047	0.070
k=4	0.030	0.034	0.036	-	-	-	0.027	0.044	0.044	0.024	0.045	0.067
k=5	0.038	0.036	0.041	-	-	-	0.029	0.048	0.045	0.025	0.057	0.090

Sampling = 25 longitudes / latitudes / altitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.027	0.034	0.034	-	-	-	0.026	0.049	0.054	0.026	0.033	0.047
k=3	0.033	0.034	0.036	-	-	-	0.027	0.048	0.049	0.026	0.049	0.068
k=4	0.038	0.039	0.036	-	-	-	0.025	0.052	0.049	0.029	0.051	0.065
k=5	0.036	0.047	0.039	-	-	-	0.028	0.050	0.050	0.034	0.062	0.093

Sampling = 50 longitudes / latitudes / altitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.039	0.036	0.034	-	-	-	0.024	0.048	0.051	0.026	0.035	0.046
k=3	0.042	0.040	0.046	-	-	-	0.027	0.050	0.049	0.030	0.049	0.080
k=4	0.048	0.041	0.047	-	-	-	0.026	0.049	0.048	0.041	0.054	0.077
k=5	0.045	0.044	0.048	-	-	-	0.029	0.054	0.051	0.039	0.066	0.102

Πίνακας 4.26: Χρόνος εκτέλεσης (sec) για Διανυσματοποίηση σε 3D Κανονικοποιημένη και Ευθυγραμμισμένη Τροχιά με Χρήση Παρεμβολής, του συνόλου δεδομένων.

Οι ίδιοι πίνακες, που δίνουν το χρόνο εκτέλεσης των αλγορίθμων, όπως αυτοί έτρεξαν για το συνοπτικό σύνολο δεδομένων, δίνονται στη συνέχεια.

Sampling = 5 longitudes / latitudes / altitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.033	0.029	0.030	0.027	0.027	0.030	0.028	0.036	0.042	0.020	0.030	0.039
k=3	0.033	0.041	0.032	0.032	0.032	0.045	0.032	0.043	0.038	0.025	0.043	0.067
k=4	0.032	0.033	0.033	0.039	0.052	0.039	0.026	0.040	0.035	0.023	0.042	0.065
k=5	0.031	0.034	0.037	-	0.052	0.043	0.027	0.033	0.031	0.027	0.053	0.088

Sampling = 10 longitudes / latitudes / altitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.027	0.029	0.029	-	-	-	0.026	0.040	0.041	0.024	0.031	0.045
k=3	0.035	0.032	0.035	-	-	-	0.025	0.041	0.041	0.023	0.043	0.063
k=4	0.035	0.036	0.039	-	-	-	0.025	0.041	0.043	0.027	0.042	0.064
k=5	0.032	0.039	0.037	-	-	-	0.025	0.042	0.040	0.028	0.053	0.087

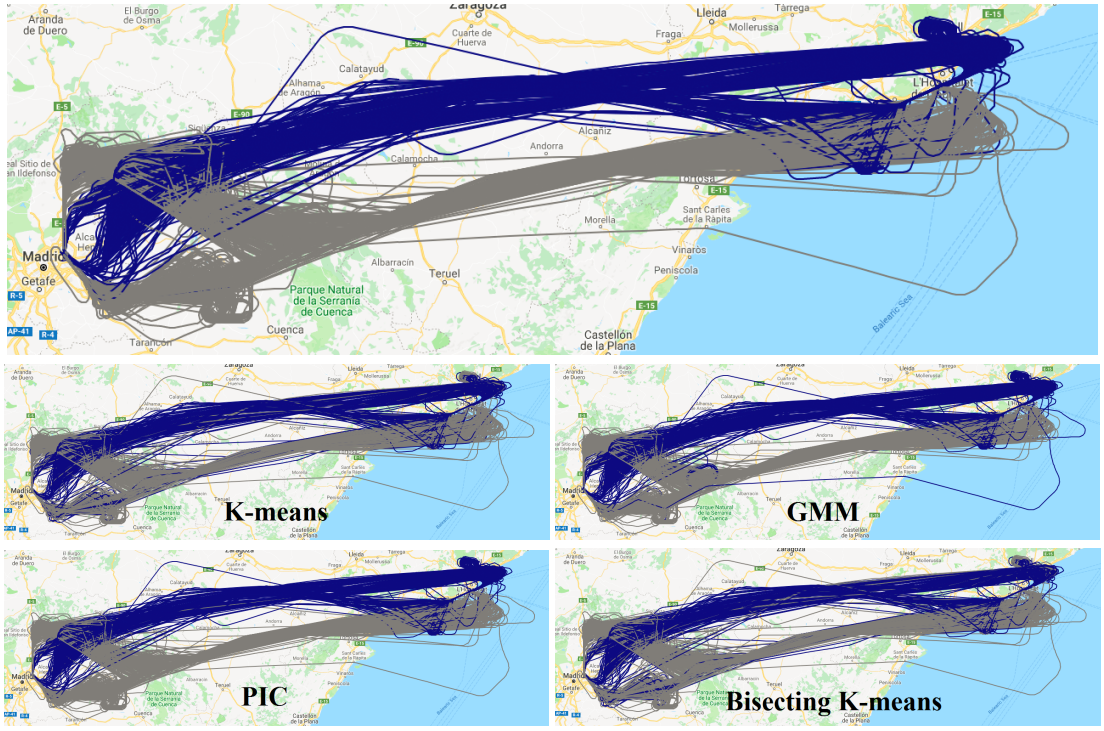
Sampling = 25 longitudes / latitudes / altitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.029	0.031	0.035	-	-	-	0.024	0.038	0.038	0.023	0.033	0.044
k=3	0.033	0.035	0.040	-	-	-	0.024	0.039	0.039	0.026	0.047	0.066
k=4	0.038	0.037	0.038	-	-	-	0.024	0.038	0.039	0.031	0.045	0.064
k=5	0.039	0.042	0.045	-	-	-	0.026	0.037	0.037	0.033	0.056	0.087

Sampling = 50 longitudes / latitudes / altitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.037	0.035	0.039	-	-	-	0.027	0.041	0.041	0.025	0.036	0.047
k=3	0.042	0.038	0.040	-	-	-	0.025	0.040	0.041	0.032	0.048	0.075
k=4	0.052	0.058	0.044	-	-	-	0.024	0.057	0.042	0.032	0.054	0.073
k=5	0.054	0.043	0.043	-	-	-	0.026	0.041	0.044	0.037	0.059	0.098

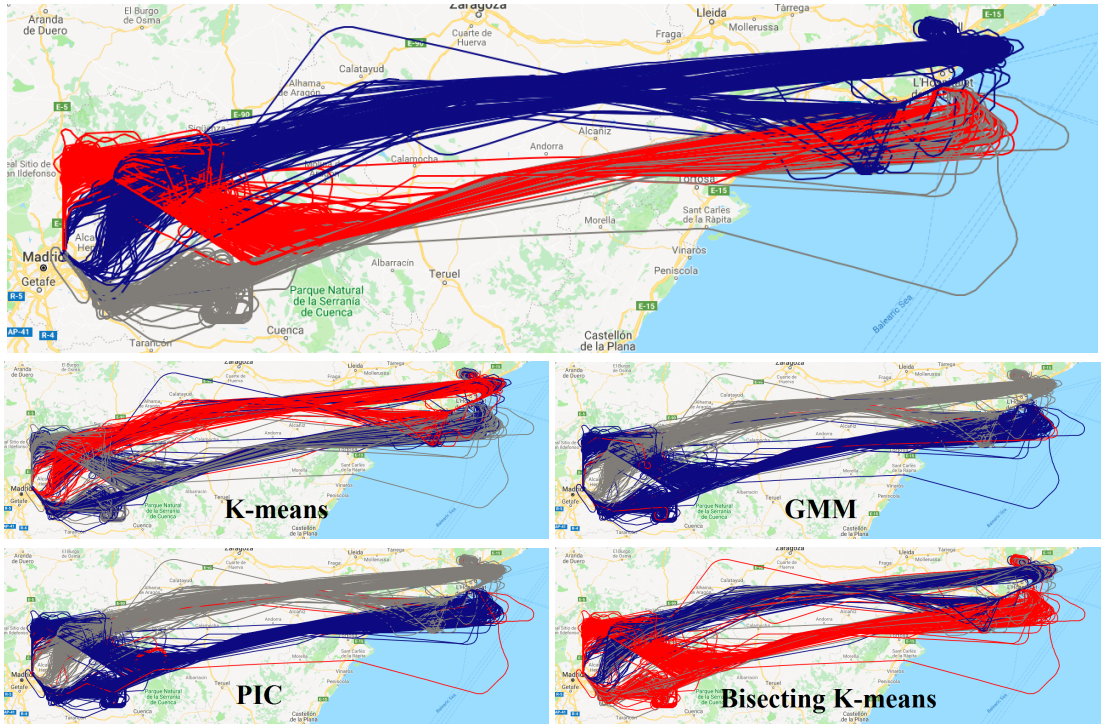
Πίνακας 4.27: Χρόνος εκτέλεσης (sec) για Διανυσματοποίηση σε 3D Κανονικοποιημένη και Ευθυγραμμισμένη Τροχιά με Χρήση Παρεμβολής, του συνοπτικού συνόλου δεδομένων.

Οι χρόνοι εκτέλεσης δε φαίνεται να διαφέρουν ιδιαίτερα από αλγόριθμο σε αλγόριθμο. Παρατηρείται ότι η αύξηση του αριθμού των επαναλήψεων αυξάνει το χρόνο εκτέλεσης για την περίπτωση των αλγορίθμων Power Iteration Clustering και Bisecting K-means, ενώ οι αλγόριθμοι K-means και Gaussian Mixture Model δε φαίνεται να επηρεάζονται με κάποιο συστηματικό τρόπο. Ακόμη η αύξηση του πλήθους των ομάδων που δημιουργούν οι αλγόριθμοι δε φαίνεται να επηρεάζει με κάποιο συστηματικό τρόπο το χρόνο εκτέλεσης των αλγορίθμων.

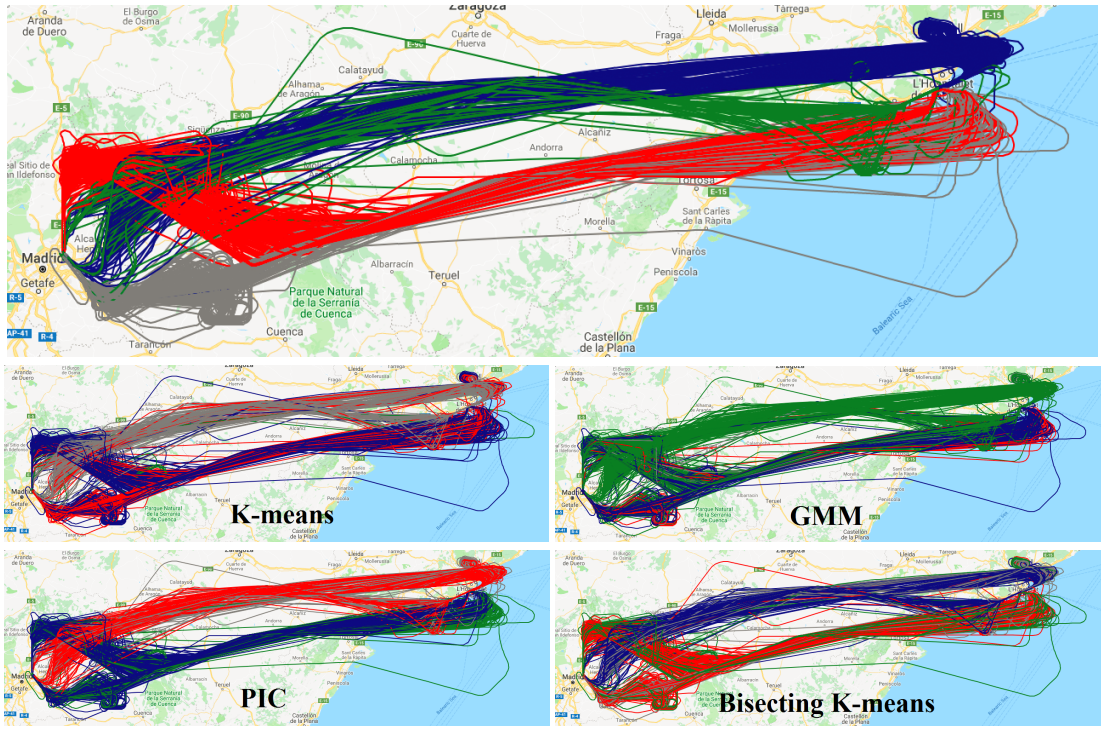
Στη συνέχεια απεικονίζουμε τις καλύτερες ομαδοποιήσεις ανά αριθμό ομάδων για καθέναν από τους αλγορίθμους σε σύγκριση με το ground truth.



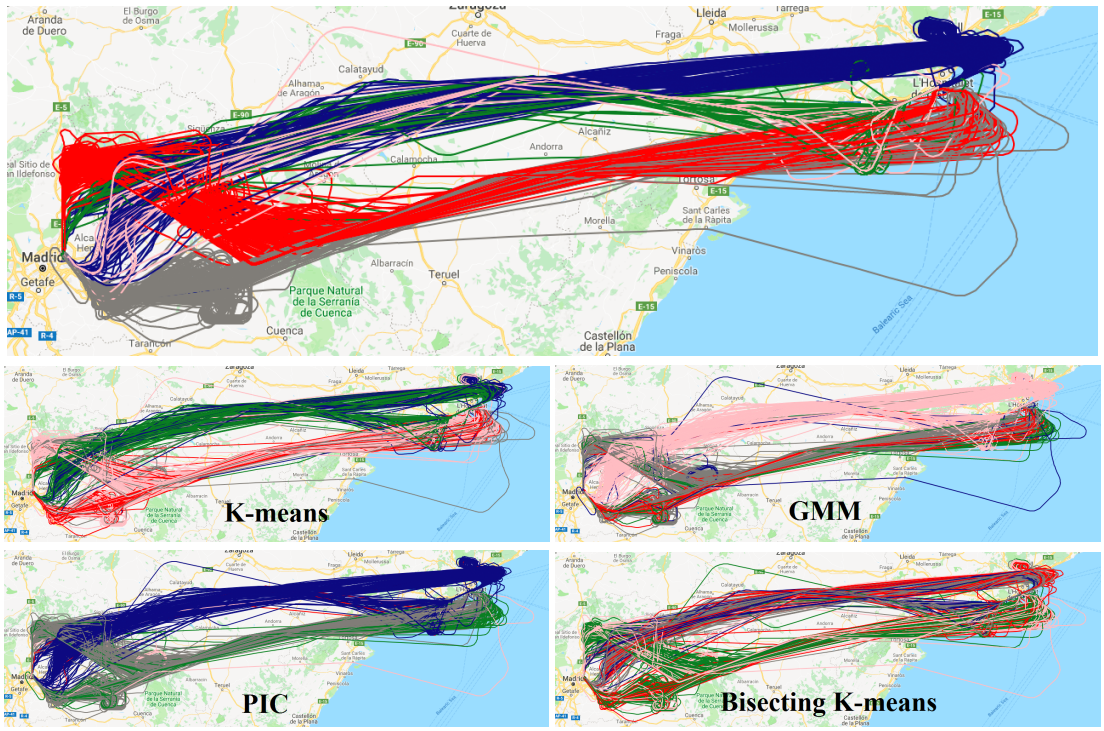
Πίνακας 4.28: Οπτική σύγκριση ομαδοποίησης αλγορίθμων με ground truth για 2 ομάδες.



Πίνακας 4.29: Οπτική σύγκριση ομαδοποίησης αλγορίθμων με ground truth για 3 ομάδες.



Πίνακας 4.30: Οπτική σύγκριση ομαδοποίησης αλγορίθμων με ground truth για 4 ομάδες.



Πίνακας 4.31: Οπτική σύγκριση ομαδοποίησης αλγορίθμων με ground truth για 5 ομάδες.

Κάτι που επιβεβαιώνεται και οπτικά είναι πως οι αλγόριθμοι δίνουν καλύτερα αποτελέσματα στη περίπτωση των δύο ομάδων. Όσο ο αριθμός των ομάδων αυξάνεται, παρατηρούμε σημαντική διαφοροποίηση σε σχέση με το ground truth. Επίσης, η ομαδοποίηση των αλγορίθμων για δύο ομάδες είναι σχεδόν η ίδια.

4.2.4 Διανυσματοποίηση σε 3D Τροχιά με Χρήση της Ανάλυσης Κυρίων Συνιστωσών

Η διανυσματοποίηση αυτή έγινε στα σύνολα δεδομένων που προέκυψαν από την προηγούμενη διανυσματοποίηση, για μέγεθος ίσο με 50 (για το συνολικό και το συνοπτικό σύνολο δεδομένων) και εφαρμόζοντας σε αυτά την μέθοδο της Ανάλυσης Κυρίων Συνιστωσών.

Η Ανάλυση Κυρίων Συνιστωσών (PCA) είναι μια στατιστική μέθοδος η οποία χρησιμοποιείται ευρέως σε προβλήματα μείωσης διαστάσεων. Η PCA βρίσκει μια περιστροφή τέτοια ώστε η πρώτη συνιστώσα να έχει τη μεγαλύτερη δυνατή διακύμανση και κάθε επόμενη συνιστώσα να έχει με τη σειρά της τη μεγαλύτερη δυνατή διακύμανση. Οι στήλες του πίνακα περιστροφής ονομάζονται κύριες συνιστώσες.

Η μέθοδος αυτή εφαρμόστηκε μέσω της Spark με χρήση της βιβλιοθήκης MLlib, για να καταλήξουμε σε 5 και 10 κύριες συνιστώσες, αντί για 50 που είχαμε αρχικά. Για παράδειγμα, στο σύνολο δεδομένων που προέκυψε από την προηγούμενη διανυσματοποίηση με μέγεθος ίσο με 50 (αυτό το σύνολο δεδομένων αποτελούνταν από 150 στήλες, οι 50 πρώτες αφορούσαν το γεωγραφικό μήκος, οι 50 επόμενες το γεωγραφικό πλάτος και οι 50 τελευταίες το ύψος) εφαρμόσαμε PCA μία φορά για τις στήλες που αφορούν το γεωγραφικό μήκος, μία φορά για τις στήλες που αφορούν το γεωγραφικό πλάτος και μία φορά για τις στήλες που αφορούν το ύψος, κρατώντας κάθε φορά από τις 50 στήλες μόνο τις 5 και 10 αντίστοιχα. Με τη διαδικασία αυτή δημιουργήσαμε τα σύνολα δεδομένων που δέχτηκαν σαν είσοδο οι αλγόριθμοι K-means, Gaussian Mixture Model και Bisecting K-means.

Ο κώδικας για την Ανάλυση Κυρίων Συνιστωσών δίνεται στο Παράρτημα Δ'

Η είσοδος που δέχτηκε ο αλγόριθμος Power Iteration Clustering είναι ένας πίνακας με 3 στήλες, όπου η πρώτη στήλη υποδεικνύει την πτήση i, η δεύτερη στήλη υποδεικνύει την πτήση j και η τρίτη στήλη υποδεικνύει την απόσταση των προαναφερθέντων πτήσεων (με $i \leq j$). Σημειώνουμε ότι η απόσταση αυτή υπολογίστηκε κάνοντας μετατροπή των πολικών συντεταγμένων σε καρτεσιανές και εφαρμόζοντας την ευκλείδεια απόσταση. Ο κώδικας που χρησιμοποιήθηκε για τη δημιουργία των συνόλων δεδομένων που δέχεται σαν είσοδο ο PIC αλγόριθμος δίνεται στο Παράρτημα Δ'.

Στη συνέχεια τρέξαμε τους αλγορίθμους ομαδοποίησης για όλες τις περιπτώσεις του μεγέθους διανύσματος (οι κώδικες των αλγορίθμων ομαδοποίησης παραμένουν ίδιοι).

Τα αποτελέσματα της αξιολόγησης των αλγορίθμων για καθεμία από τις παραμέτρους για τις οποίες εξετάστηκαν (πλήθος ομάδων k και μέγιστο πλήθος επαναλήψεων *MaxIterations*) είναι συγκεντρωμένα στους πίνακες που ακολουθούν.

Οι πίνακες που δίνονται στη συνέχεια περιλαμβάνουν τις τιμές των Accuracy, Precision και F1 Score που αξιολογούν το αποτέλεσμα του κάθε αλγορίθμου που έτρεξε σε ολόκληρο το σύνολο δεδομένων, σε σχέση με το *ground truth*.

Sampling = 5 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.833	0.833	0.833	0.834	0.527	0.552	0.612	0.612	0.612	0.833	0.833	0.833
	Precision	0.905	0.905	0.905	0.905	0.535	0.569	0.608	0.608	0.608	0.905	0.905	0.905
	F1-Score	0.815	0.815	0.815	0.816	0.417	0.466	0.609	0.610	0.610	0.815	0.815	0.815
k=3	Accuracy	0.698	0.681	0.688	0.506	0.502	0.431	0.511	0.520	0.520	0.549	0.549	0.549
	Precision	0.680	0.673	0.678	0.406	0.404	0.354	0.505	0.515	0.515	0.583	0.583	0.583
	F1-Score	0.644	0.668	0.671	0.334	0.361	0.328	0.496	0.504	0.504	0.543	0.543	0.543
k=4	Accuracy	0.465	0.505	0.505	0.393	0.331	0.415	0.455	0.483	0.483	0.486	0.486	0.486
	Precision	0.497	0.522	0.522	0.375	0.279	0.373	0.474	0.508	0.508	0.510	0.510	0.510
	F1-Score	0.465	0.498	0.498	0.295	0.239	0.325	0.439	0.453	0.453	0.483	0.483	0.483
k=5	Accuracy	0.542	0.536	0.536	0.352	0.352	0.307	0.367	0.413	0.413	0.467	0.467	0.467
	Precision	0.441	0.446	0.446	0.282	0.258	0.258	0.412	0.424	0.424	0.477	0.477	0.477
	F1-Score	0.439	0.435	0.435	0.250	0.185	0.251	0.310	0.362	0.362	0.399	0.399	0.399

Sampling = 10 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.834	0.834	0.834	-	-	-	0.607	0.612	0.612	0.834	0.834	0.834
	Precision	0.905	0.905	0.905	-	-	-	0.604	0.608	0.608	0.905	0.905	0.905
	F1-Score	0.816	0.816	0.816	-	-	-	0.603	0.609	0.609	0.816	0.816	0.816
k=3	Accuracy	0.684	0.693	0.681	-	-	-	0.506	0.516	0.516	0.537	0.537	0.537
	Precision	0.677	0.679	0.673	-	-	-	0.500	0.510	0.510	0.579	0.579	0.579
	F1-Score	0.672	0.668	0.668	-	-	-	0.491	0.499	0.499	0.534	0.534	0.534
k=4	Accuracy	0.555	0.501	0.492	-	-	-	0.466	0.464	0.464	0.486	0.487	0.487
	Precision	0.549	0.518	0.514	-	-	-	0.453	0.474	0.474	0.510	0.510	0.510
	F1-Score	0.542	0.494	0.485	-	-	-	0.435	0.446	0.446	0.482	0.481	0.481
k=5	Accuracy	0.480	0.501	0.501	-	-	-	0.405	0.410	0.410	0.473	0.476	0.476
	Precision	0.408	0.442	0.442	-	-	-	0.416	0.419	0.419	0.481	0.477	0.477
	F1-Score	0.398	0.408	0.408	-	-	-	0.357	0.359	0.359	0.403	0.403	0.403

Πίνακας 4.32: Accuracy, Precision και F1 Score(%) για Διανυσματοποίηση σε 3D Τροχιά με Χρήση της Ανάλυσης Κυρίων Συνιστωσών, του συνόλου δεδομένων.

Οι ίδιοι πίνακες, που δίνουν τα αποτελέσματα αξιολόγησης των αλγορίθμων όπως αυτά έτρεξαν για το συνοπτικό σύνολο δεδομένων, δίνονται στη συνέχεια.

Sampling = 5 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.832	0.833	0.832	0.592	0.610	0.592	0.523	0.500	0.500	0.832	0.832	0.832
	Precision	0.896	0.899	0.896	0.605	0.579	0.630	0.515	0.493	0.493	0.896	0.896	0.896
	F1-Score	0.815	0.816	0.815	0.550	0.662	0.507	0.570	0.430	0.430	0.815	0.815	0.815
k=3	Accuracy	0.628	0.674	0.676	0.415	0.467	0.395	0.491	0.534	0.534	0.532	0.532	0.532
	Precision	0.646	0.666	0.668	0.395	0.341	0.336	0.484	0.538	0.538	0.563	0.563	0.563
	F1-Score	0.631	0.662	0.665	0.396	0.295	0.306	0.490	0.537	0.537	0.525	0.525	0.525
k=4	Accuracy	0.544	0.491	0.475	0.315	0.373	0.378	0.467	0.474	0.474	0.483	0.483	0.483
	Precision	0.544	0.511	0.500	0.211	0.308	0.290	0.446	0.464	0.464	0.506	0.507	0.507
	F1-Score	0.536	0.485	0.467	0.215	0.300	0.297	0.443	0.451	0.451	0.479	0.479	0.479
k=5	Accuracy	0.469	0.535	0.512	0.340	0.320	0.350	0.411	0.421	0.421	0.464	0.464	0.464
	Precision	0.416	0.444	0.438	0.257	0.217	0.242	0.389	0.401	0.401	0.441	0.442	0.442
	F1-Score	0.388	0.436	0.420	0.241	0.162	0.244	0.351	0.359	0.359	0.393	0.393	0.393

Sampling = 10 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Accuracy	0.834	0.834	0.834	-	-	-	0.515	0.504	0.504	0.834	0.834	0.834
	Precision	0.896	0.896	0.896	-	-	-	0.508	0.499	0.499	0.899	0.899	0.899
	F1-Score	0.817	0.817	0.817	-	-	-	0.566	0.556	0.556	0.817	0.817	0.817
k=3	Accuracy	0.658	0.682	0.686	-	-	-	0.491	0.513	0.513	0.547	0.547	0.547
	Precision	0.666	0.673	0.677	-	-	-	0.486	0.511	0.511	0.579	0.579	0.579
	F1-Score	0.657	0.669	0.671	-	-	-	0.491	0.513	0.513	0.539	0.539	0.539
k=4	Accuracy	0.529	0.491	0.491	-	-	-	0.468	0.477	0.477	0.484	0.487	0.487
	Precision	0.539	0.514	0.515	-	-	-	0.451	0.457	0.457	0.510	0.512	0.512
	F1-Score	0.521	0.487	0.488	-	-	-	0.447	0.451	0.451	0.482	0.485	0.485
k=5	Accuracy	0.404	0.508	0.508	-	-	-	0.416	0.418	0.418	0.463	0.467	0.467
	Precision	0.414	0.438	0.438	-	-	-	0.389	0.403	0.403	0.446	0.448	0.448
	F1-Score	0.365	0.419	0.419	-	-	-	0.353	0.359	0.359	0.395	0.398	0.398

Πίνακας 4.33: Accuracy, Precision και F1 Score(%) για Διανυσματοποίηση σε 3D Τροχιά με Χρήση της Ανάλυσης Κυρίων Συνιστωσών, του συνοπτικού συνόλου δεδομένων.

Παρατηρούμε αρχικά ότι ο αλγόριθμος Gaussian Mixture Model αποτυγχάνει στην περίπτωση για μέγεθος

διανύσματος ίσο με 10, και η ομαδοποίηση που επιτυγχάνει στην περίπτωση του μεγέθους διανύσματος ίσο με 5 δε δίνει υψηλές τιμές στα μέτρα απόδοσης. Σημειώνουμε επίσης ότι οι αλγόριθμοι K-means και Bisecting K-means έχουν παρόμοιες τιμές στα μέτρα απόδοσης, με τον K-means να εφαρμόζει καλύτερη ομαδοποίηση στην περίπτωση των 3 ή περισσότερων ομάδων. Τέλος, αντίθετα με τις άλλες μεθόδους διανυσματοποίησης, τα μέτρα απόδοσης του αλγορίθμου Power Iteration Clustering δεν είναι τα υψηλότερα από όλους τους αλγορίθμους.

Στη συνέχεια δίνονται οι πίνακες που περιλαμβάνουν τη μέση απόσταση του κεντροειδούς από τα μέλη της ομάδας, όπως αυτή υπολογίστηκε για κάθε αλγόριθμο που έτρεξε στο συνολικό σύνολο δεδομένων.

Sampling = 5 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	17.943	17.943	17.943	22.423	22.423	22.423	21.471	21.433	21.433	17.943	17.943	17.943
	Cluster 2	33.106	33.106	33.106	29.590	29.590	29.590	28.199	28.297	28.297	33.106	33.106	33.106
k=3	Cluster 1	16.363	16.496	16.543	22.385	22.385	22.385	22.018	26.319	26.319	15.806	15.806	15.806
	Cluster 2	25.385	26.521	26.804	28.858	28.858	28.858	24.950	29.377	29.377	24.536	24.536	24.536
	Cluster 3	39.723	34.204	34.803	30.569	30.569	30.569	29.633	31.944	31.944	35.200	35.200	35.200
k=4	Cluster 1	17.277	22.165	22.165	21.737	21.737	21.737	22.299	26.643	26.643	27.018	27.018	27.018
	Cluster 2	22.838	32.681	28.838	28.858	28.858	28.858	29.089	29.519	29.519	22.451	22.451	22.451
	Cluster 3	36.202	39.058	39.058	30.542	30.542	30.542	27.155	32.032	32.032	29.804	29.804	29.804
	Cluster 4	21.634	17.955	17.955	24.105	24.105	24.105	21.381	22.578	22.578	26.005	26.005	26.005
k=5	Cluster 1	17.363	14.929	37.300	21.737	21.737	21.737	28.307	28.622	28.622	21.339	21.339	21.339
	Cluster 2	24.603	32.537	32.537	28.858	28.858	28.858	29.330	27.267	27.267	22.451	22.451	22.451
	Cluster 3	25.941	26.747	26.747	30.542	30.542	30.542	30.894	32.194	32.194	26.520	26.520	26.520
	Cluster 4	16.891	18.080	17.312	23.749	23.749	23.749	21.048	22.996	22.996	17.837	17.837	17.837
	Cluster 5	32.522	34.252	18.394	25.138	25.138	25.138	29.019	26.588	26.588	34.239	34.239	34.239

Sampling = 10 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	9.588	9.588	9.588	-	-	-	11.472	11.372	11.372	9.588	9.588	9.588
	Cluster 2	17.264	17.264	17.264	-	-	-	14.907	14.854	14.854	17.264	17.264	17.264
k=3	Cluster 1	12.909	8.936	8.891	-	-	-	11.670	13.793	13.793	8.577	8.577	8.577
	Cluster 2	17.237	14.918	13.900	-	-	-	13.082	15.422	15.422	13.098	13.098	13.098
	Cluster 3	15.903	19.023	17.934	-	-	-	15.822	16.072	16.072	18.424	18.424	18.424
k=4	Cluster 1	11.669	8.150	14.292	-	-	-	14.935	14.744	14.744	14.248	14.248	14.248
	Cluster 2	11.280	19.429	15.233	-	-	-	15.882	15.239	15.239	11.826	11.918	11.918
	Cluster 3	15.750	18.046	15.752	-	-	-	17.140	16.138	16.138	15.771	16.332	16.332
	Cluster 4	16.652	11.758	11.735	-	-	-	12.613	13.215	13.215	13.854	13.854	13.854
k=5	Cluster 1	8.454	8.184	11.678	-	-	-	11.907	15.032	15.032	11.277	11.277	11.277
	Cluster 2	19.772	24.116	14.775	-	-	-	15.364	14.338	14.338	11.826	11.918	11.918
	Cluster 3	16.046	18.632	18.632	-	-	-	14.637	17.081	17.081	14.173	14.195	14.195
	Cluster 4	12.307	11.782	9.531	-	-	-	12.382	11.767	11.767	9.394	9.730	9.730
	Cluster 5	14.797	13.712	22.326	-	-	-	10.569	13.999	13.999	17.763	17.763	17.763

Πίνακας 4.34: RMSE απόσταση του κεντροειδούς από τα μέλη της ομάδας για Διανυσματοποίηση σε 3D Τροχιά με Χρήση της Ανάλυσης Κυρίων Συνιστωσών, του συνόλου δεδομένων.

Οι ίδιοι πίνακες, που δίνουν τη μέση απόσταση του κεντροειδούς από τα μέλη της ομάδας, όπως αυτά έτρεξαν για το συνοπτικό σύνολο δεδομένων, δίνονται στη συνέχεια.

Sampling = 5 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	19.077	19.071	19.077	23.580	23.580	23.580	28.046	24.014	24.014	19.077	19.077	19.077
	Cluster 2	34.328	34.459	34.328	30.241	30.241	30.241	29.926	31.103	31.103	34.328	34.328	34.328
k=3	Cluster 1	17.215	26.287	26.324	23.549	23.549	23.549	24.703	23.343	23.343	17.376	17.376	17.376
	Cluster 2	31.500	36.453	36.275	32.017	32.017	32.017	30.642	30.554	30.554	29.432	29.432	29.432
	Cluster 3	30.041	28.589	28.651	28.076	28.076	28.076	25.898	26.168	26.168	32.879	32.879	32.879
k=4	Cluster 1	17.736	22.152	26.864	22.155	22.155	22.155	23.299	23.436	23.436	26.237	26.356	26.356
	Cluster 2	28.690	28.072	28.041	32.017	32.017	32.017	30.662	30.976	30.976	28.062	28.062	28.062
	Cluster 3	31.459	32.464	31.937	28.074	28.074	28.074	25.306	27.251	27.251	31.346	31.346	31.346
	Cluster 4	21.948	35.970	28.372	26.924	26.924	26.924	30.555	28.691	28.691	27.649	27.690	27.690
k=5	Cluster 1	22.315	15.296	20.683	22.155	22.155	22.155	23.050	23.492	23.492	21.875	21.857	21.857
	Cluster 2	41.578	28.675	35.242	32.017	32.017	32.017	32.218	30.292	30.292	28.062	28.062	28.062
	Cluster 3	30.152	26.163	35.364	28.074	28.074	28.074	32.303	25.409	25.409	34.878	34.994	34.994
	Cluster 4	24.449	36.843	21.286	26.190	26.190	26.190	29.783	27.590	27.590	19.691	19.691	19.691
	Cluster 5	28.318	21.371	27.326	29.056	29.056	29.056	23.520	23.790	23.790	26.171	22.502	22.502

Sampling = 10 longitudes / latitudes / altitudes													
		K-Means			GMM			PIC			Bisecting K-Means		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	11.067	11.067	11.067	-	-	-	15.695	15.776	15.776	11.081	11.081	11.081
	Cluster 2	18.617	18.617	18.617	-	-	-	16.615	16.949	16.949	18.685	18.685	18.685
k=3	Cluster 1	10.130	10.194	10.233	-	-	-	13.929	13.474	13.474	10.115	10.115	10.115
	Cluster 2	16.622	16.857	16.768	-	-	-	16.649	16.670	16.670	15.638	15.638	15.638
	Cluster 3	16.446	16.901	16.929	-	-	-	14.081	14.591	14.591	18.023	18.023	18.023
k=4	Cluster 1	14.265	12.962	14.713	-	-	-	12.078	11.513	11.513	14.756	14.756	14.756
	Cluster 2	15.555	18.004	16.205	-	-	-	16.816	16.847	16.847	15.460	15.362	15.362
	Cluster 3	15.592	18.702	17.816	-	-	-	14.299	14.363	14.363	16.244	16.009	16.009
	Cluster 4	15.102	12.696	12.744	-	-	-	14.603	15.255	15.255	15.522	15.522	15.522
k=5	Cluster 1	14.222	11.874	11.874	-	-	-	13.129	13.351	13.351	12.523	12.523	12.523
	Cluster 2	15.085	16.990	16.990	-	-	-	16.655	16.981	16.981	15.460	15.362	15.362
	Cluster 3	15.329	20.139	20.139	-	-	-	14.738	17.310	17.310	18.819	18.819	18.819
	Cluster 4	15.333	18.175	18.175	-	-	-	16.441	16.634	16.634	11.505	11.500	11.500
	Cluster 5	21.144	14.215	14.215	-	-	-	13.539	13.522	13.522	12.652	12.640	12.640

Πίνακας 4.35: RMSE απόσταση του κεντροειδούς από τα μέλη της ομάδας για Διανυσματοποίηση σε 3D Τροχιά με Χρήση της Ανάλυσης Κυρίων Συνιστωσών, του συνοπτικού συνόλου δεδομένων.

Αντίθετα με τις άλλες μεθόδους διανυσματοποίησης, παρατηρούμε ότι και για τα δύο σύνολα δεδομένων ότι η μέση απόσταση του κεντροειδούς από τα μέλη της ομάδας μειώνεται καθώς αυξάνεται το μέγεθος διανύσματος (για τον ίδιο αριθμό ομάδων). Επίσης οι μικρότερες αποστάσεις παρατηρούνται στην περίπτωση των K-means και Bisecting K-means αλγορίθμων για την περίπτωση των δύο ομάδων. Οι δύο παραπάνω αλγόριθμοι δίνουν παρόμοιες τιμές για το RMSE με τον K-means να δημιουργεί πιο ομογενείς ομάδες στην περίπτωση των 4 ή περισσότερων ομάδων. Ο αλγόριθμος Gaussian Mixture Model αποτυγχάνει για την περίπτωση του μεγέθους διανύσματος μεγαλύτερου του 5, ενώ για μέγεθος διανύσματος ίσο με 5 οι ομάδες που δημιουργεί δεν είναι ομογενείς αφού δίνουν αρκετά υψηλές τιμές στο RMSE του κεντροειδούς από τα μέλη των ομάδων. Τέλος, αντίθετα με τις προηγούμενες τεχνικές διανυσματοποίησης που εφαρμόστηκαν, ο αλγόριθμος Power Iteration Clustering δε δημιουργεί ιδιαίτερα ομογενείς ομάδες, ενώ η ομογένεια αυτή φαίνεται να βελτιώνεται καθώς το μέγεθος διανύσματος αυξάνεται.

Τέλος παραθέτουμε έναν πίνακα με τους χρόνους εκτέλεσης καθενός αλγορίθμου για τις διάφορες τιμές των παραμέτρων που μελετήσαμε, όπως έτρεξαν για το συνολικό σύνολο δεδομένων.

Sampling = 5 longitudes / latitudes / altitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.026	0.030	0.028	0.029	0.028	0.027	0.025	0.028	0.032	0.026	0.032	0.047
k=3	0.027	0.029	0.038	0.034	0.030	0.033	0.025	0.030	0.029	0.026	0.043	0.063
k=4	0.033	0.037	0.041	0.039	0.042	0.035	0.028	0.029	0.030	0.028	0.044	0.067
k=5	0.032	0.043	0.041	0.045	0.043	0.044	0.028	0.031	0.030	0.035	0.054	0.096

Sampling = 10 longitudes / latitudes / altitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.036	0.030	0.027	-	-	-	0.027	0.028	0.028	0.024	0.031	0.046
k=3	0.035	0.041	0.035	-	-	-	0.027	0.030	0.033	0.028	0.043	0.069
k=4	0.032	0.040	0.035	-	-	-	0.027	0.032	0.032	0.029	0.050	0.064
k=5	0.037	0.036	0.042	-	-	-	0.029	0.029	0.031	0.032	0.059	0.093

Πίνακας 4.36: Χρόνος εκτέλεσης (sec) για Διανυσματοποίηση σε 3D Τροχιά με Χρήση της Ανάλυσης Κυρίων Συνιστωσών, του συνόλου δεδομένων.

Οι ίδιοι πίνακες, που δίνουν το χρόνο εκτέλεσης των αλγορίθμων, όπως αυτοί έτρεξαν για το συνοπτικό σύνολο δεδομένων, δίνονται στη συνέχεια.

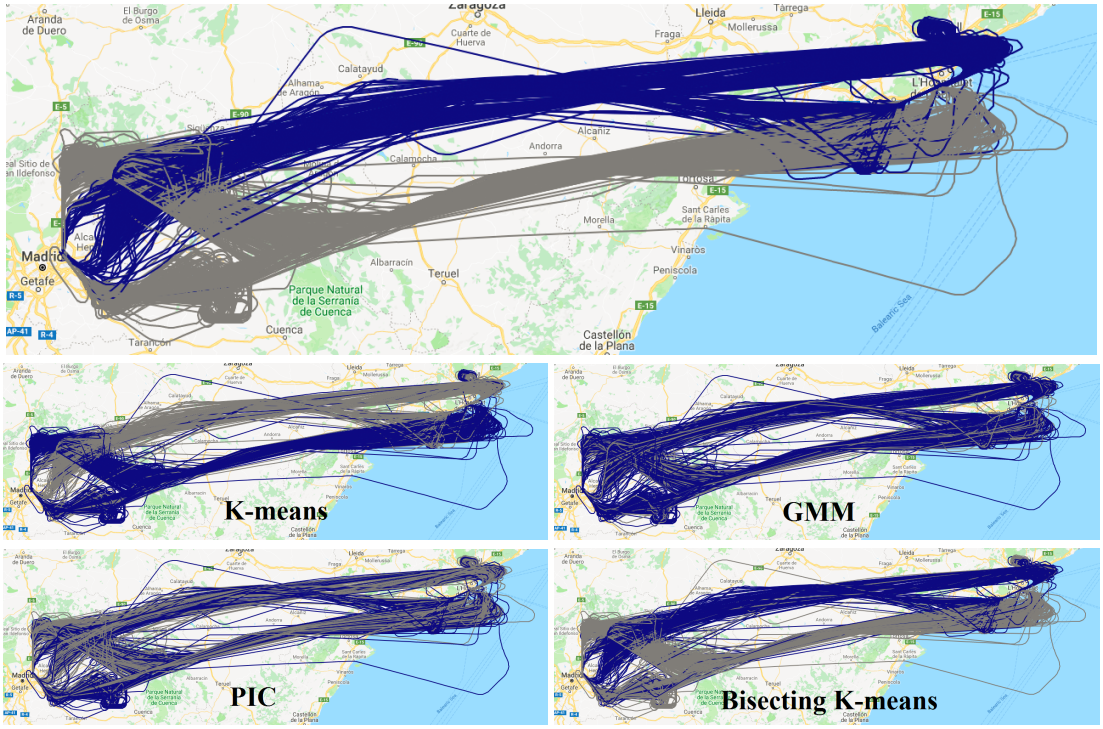
Sampling = 5 longitudes / latitudes / altitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.030	0.030	0.030	0.027	0.028	0.032	0.027	0.030	0.041	0.026	0.031	0.044
k=3	0.031	0.032	0.039	0.036	0.033	0.030	0.027	0.045	0.042	0.026	0.040	0.067
k=4	0.035	0.044	0.036	0.040	0.036	0.041	0.028	0.029	0.029	0.036	0.041	0.067
k=5	0.030	0.066	0.045	0.046	0.048	0.045	0.026	0.032	0.029	0.032	0.054	0.085

Sampling = 10 longitudes / latitudes / altitudes												
	K-Means			GMM			PIC			Bisecting K-means		
	MaxIterations			MaxIterations			MaxIterations			MaxIterations		
	10	50	100	10	50	100	10	50	100	10	50	100
k=2	0.029	0.029	0.028	-	-	-	0.027	0.027	0.026	0.026	0.030	0.043
k=3	0.035	0.036	0.035	-	-	-	0.029	0.033	0.035	0.027	0.044	0.064
k=4	0.039	0.045	0.044	-	-	-	0.027	0.027	0.030	0.027	0.043	0.067
k=5	0.035	0.037	0.039	-	-	-	0.026	0.029	0.026	0.030	0.056	0.084

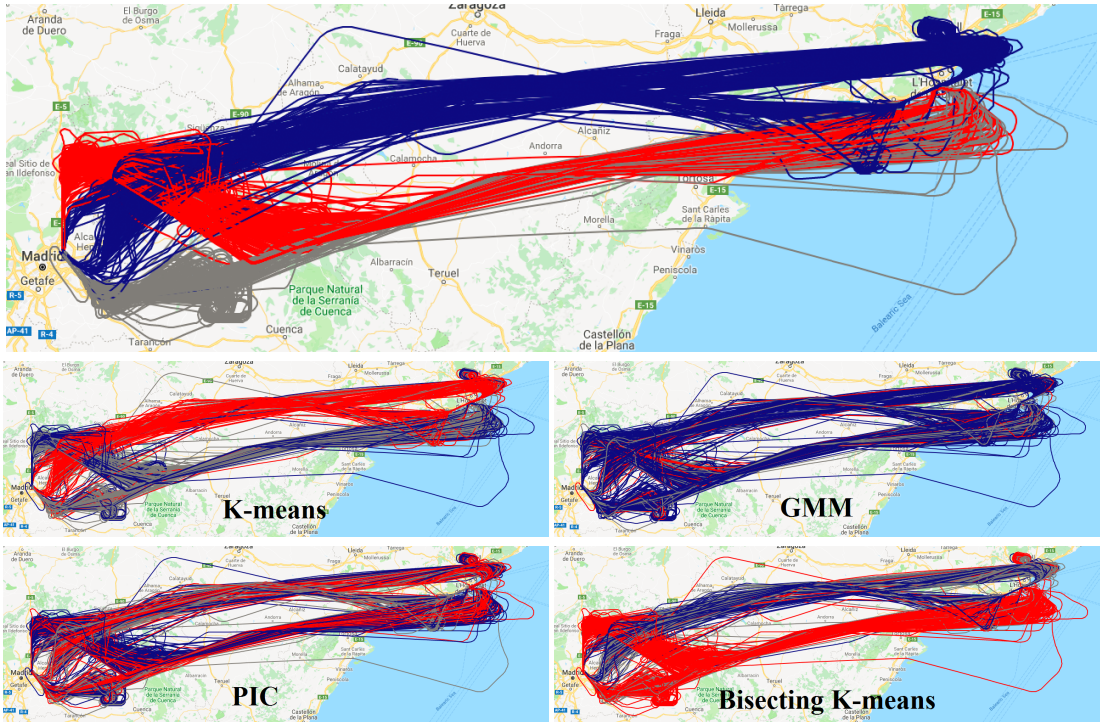
Πίνακας 4.37: Χρόνος εκτέλεσης (sec) για Διανυσματοποίηση σε 3D Τροχιά με Χρήση της Ανάλυσης Κυρίων Συνιστωσών, του συνοπτικού συνόλου δεδομένων.

Οι χρόνοι εκτέλεσης δε φαίνεται να διαφέρουν ιδιαίτερα από αλγόριθμο σε αλγόριθμο. Παρατηρείται ότι η αύξηση του αριθμού των επαναλήψεων αυξάνει το χρόνο εκτέλεσης για την περίπτωση των αλγορίθμων Power Iteration Clustering και Bisecting K-means, ενώ οι αλγόριθμοι K-means και Gaussian Mixture Model δε φαίνεται να επηρεάζονται με κάποιο συστηματικό τρόπο. Ακόμη η αύξηση του πλήθους των ομάδων που δημιουργούν οι αλγόριθμοι δε φαίνεται να επηρεάζει με κάποιο συστηματικό τρόπο το χρόνο εκτέλεσης των αλγορίθμων.

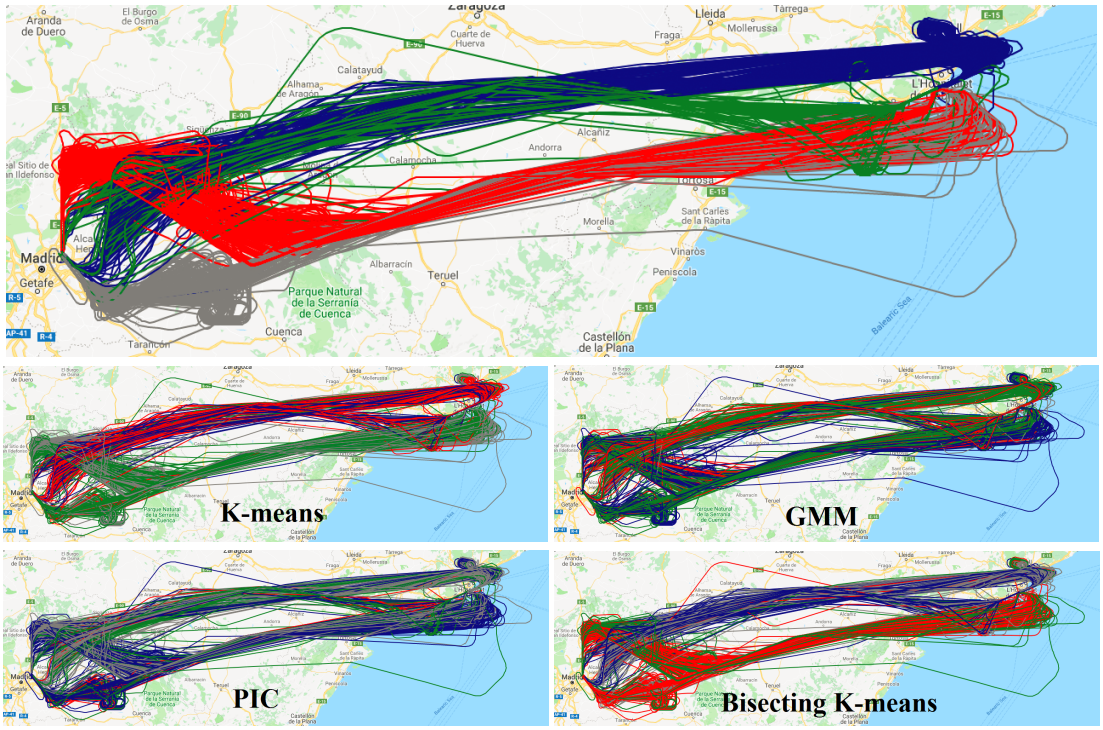
Στη συνέχεια απεικονίζουμε τις καλύτερες ομαδοποιήσεις ανά αριθμό ομάδων για καθέναν από τους αλγορίθμους σε σύγκριση με το ground truth.



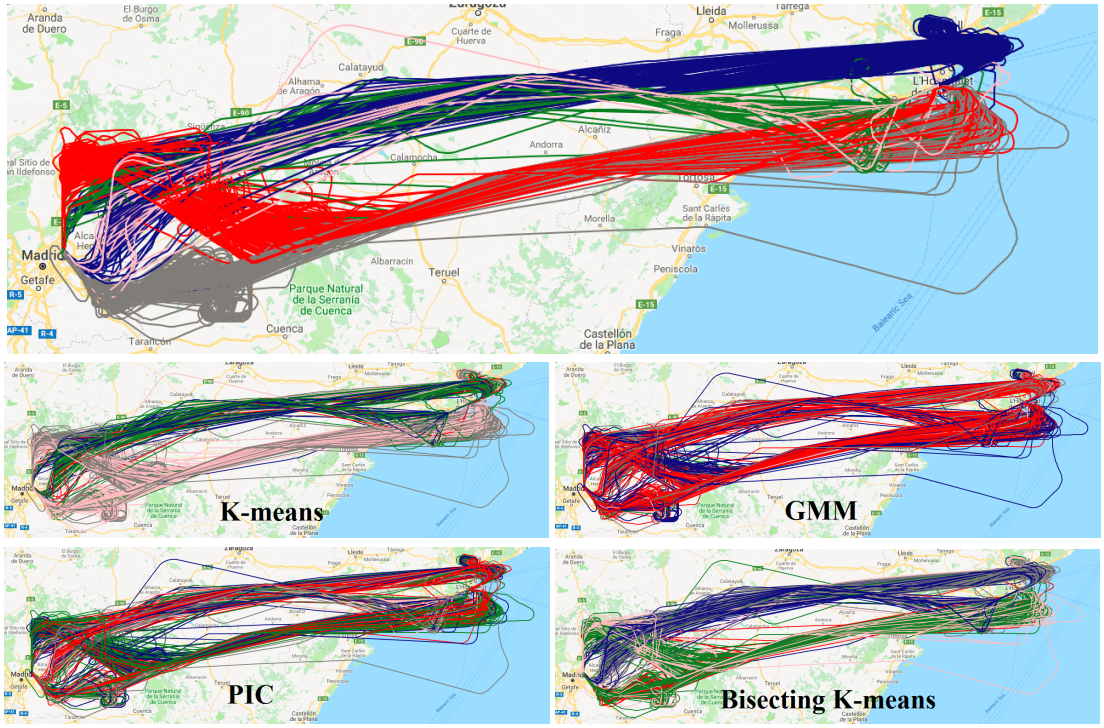
Πίνακας 4.38: Οπτική σύγκριση ομαδοποίησης αλγορίθμων με ground truth για 2 ομάδες.



Πίνακας 4.39: Οπτική σύγκριση ομαδοποίησης αλγορίθμων με ground truth για 3 ομάδες.



Πίνακας 4.40: Οπτική σύγκριση ομαδοποίησης αλγορίθμων με ground truth για 4 ομάδες.



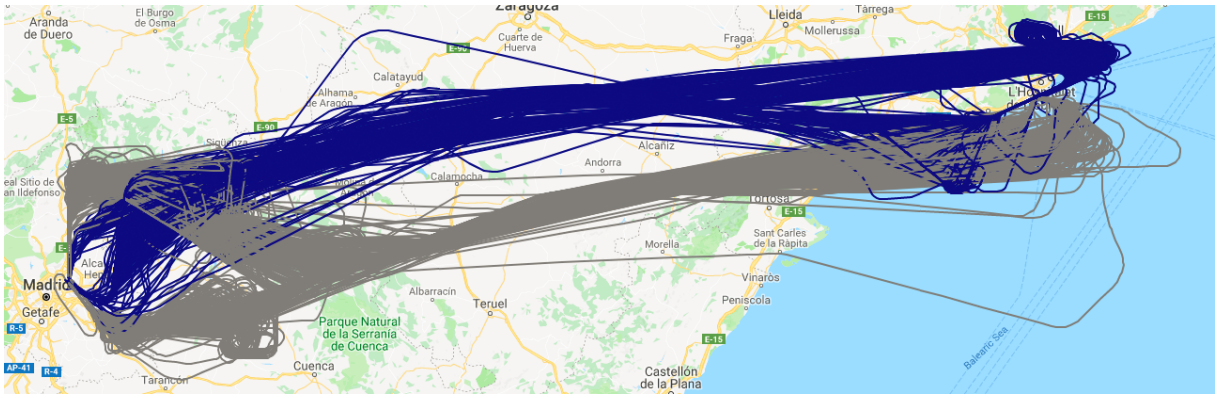
Πίνακας 4.41: Οπτική σύγκριση ομαδοποίησης αλγορίθμων με ground truth για 5 ομάδες.

Κάτι που επιβεβαιώνεται και οπτικά είναι πως οι αλγόριθμοι δίνουν καλύτερα αποτελέσματα στη περίπτωση των δύο ομάδων. Όσο ο αριθμός των ομάδων αυξάνεται, παρατηρούμε σημαντική διαφοροποίηση σε σχέση με το ground truth. Επίσης, η ομαδοποίηση που επιτυγχάνουν οι αλγόριθμοι K-means και Bisecting K-means είναι καλύτερη συγκριτικά με αυτή των αλγορίθμων Power Iteration Clusterin και Gaussian Mixture Model για την περίπτωση των δύο ομάδων.

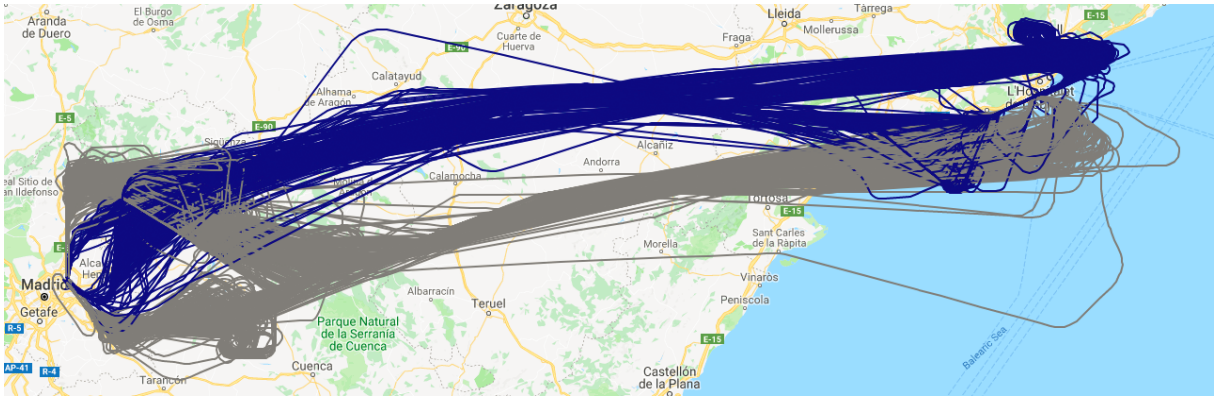
4.3 Τελικά Συμπεράσματα

Από την ανάλυση που πραγματοποιήθηκε παραπάνω, είναι σαφές πως σε όλες τις περιπτώσεις η καλύτερη ομαδοποίηση γίνεται για την περίπτωση των 2 ομάδων. Ο αλγόριθμος Gaussian Mixture Model φαίνεται να αποτυγχάνει σε αρκετές περιπτώσεις όπου το πλήθος της διανυσματοποίησης αυξάνεται. Ακόμη και για τις περιπτώσεις που επιτυγχάνει η ομαδοποίηση που προκύπτει δεν έχει ιδιαίτερα υψηλά μέτρα απόδοσης. Ακόμη οι αλγόριθμοι K-means και Bisecting K-means φαίνεται να έχουν αρκετά υψηλή απόδοση στην περίπτωση της διανυσματοποίησης με χρήση της Ανάλυσης Κυρίων Συνιστωσών, ενώ ο αλγόριθμος Power Iteration Clustering έχει πολύ υψηλές επιδόσεις για όλες τις υπόλοιπες τεχνικές διανυσματοποίησης.

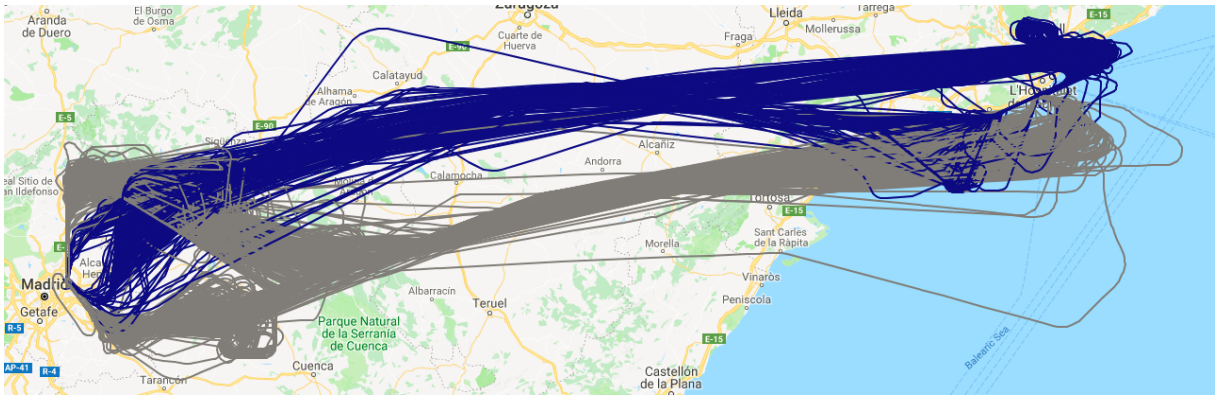
Στη συνέχεια ακολουθούν τέσσερις εικόνες, μια για κάθε διανυσματοποίηση που εφαρμόστηκε, στις οποίες απεικονίζονται οι 2D τροχιές σε χαρτογραφικό υπόβαθρο. Σε κάθε μια από αυτές φαίνεται ο διαχωρισμός των τροχιών σε ομάδες. Σε κάθε περίπτωση διανυσματοποίησης επιλέχθηκε η ομαδοποίηση με την καλύτερη αξιολόγηση.



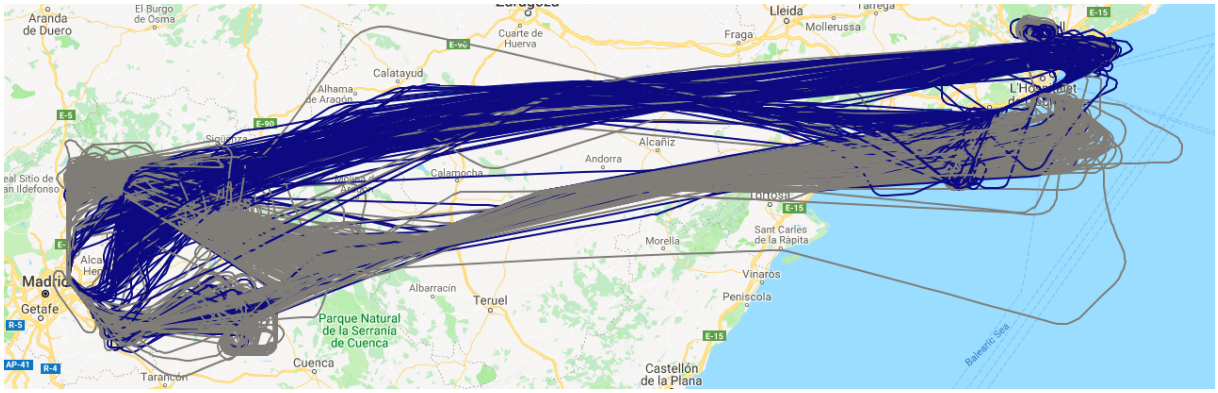
Σχήμα 4.10: Βέλτιστη Ομαδοποίηση για Διανυσματοποίηση 2D Τροχιών με χρήση παρεμβολής.



Σχήμα 4.11: Βέλτιστη Ομαδοποίηση για Διανυσματοποίηση 3D Τροχιών με χρήση παρεμβολής.



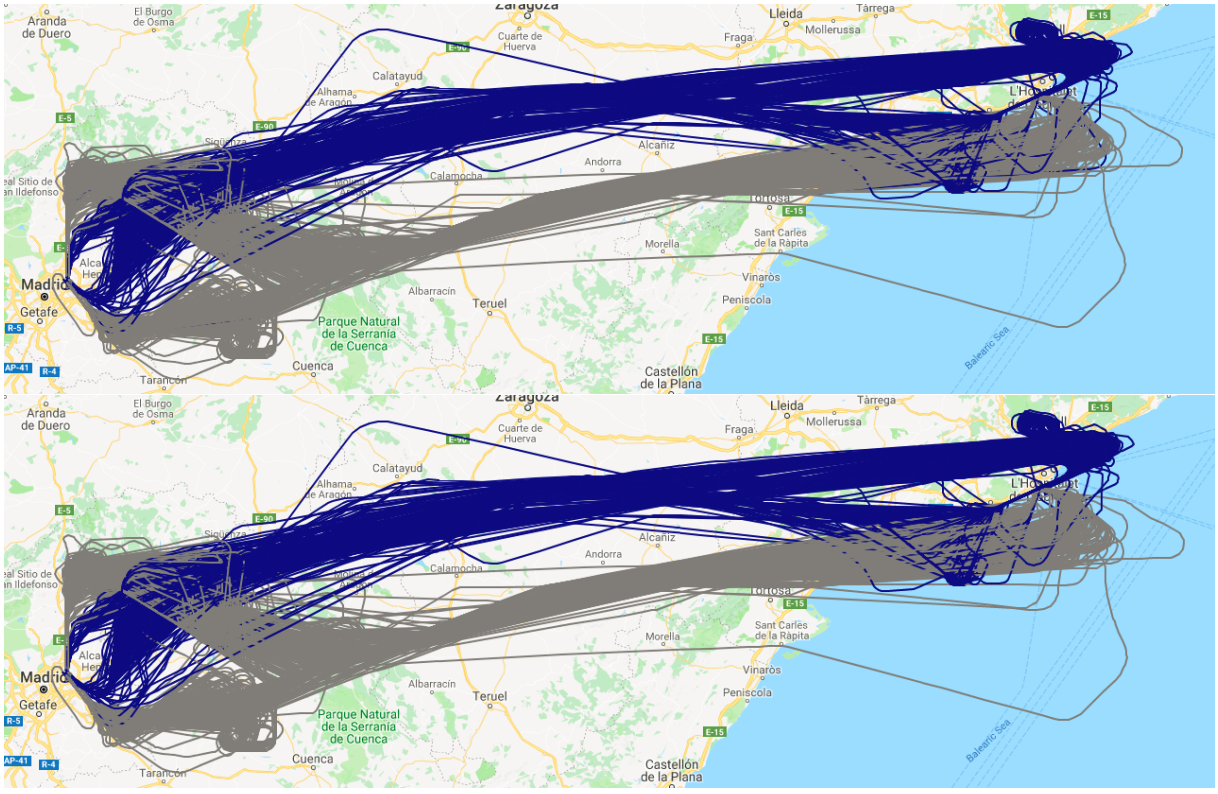
Σχήμα 4.12: Βέλτιστη Ομαδοποίηση για Διανυσματοποίηση 3D Κανονικοποιημένων και Ευθυγραμμισμένων Τροχιών.



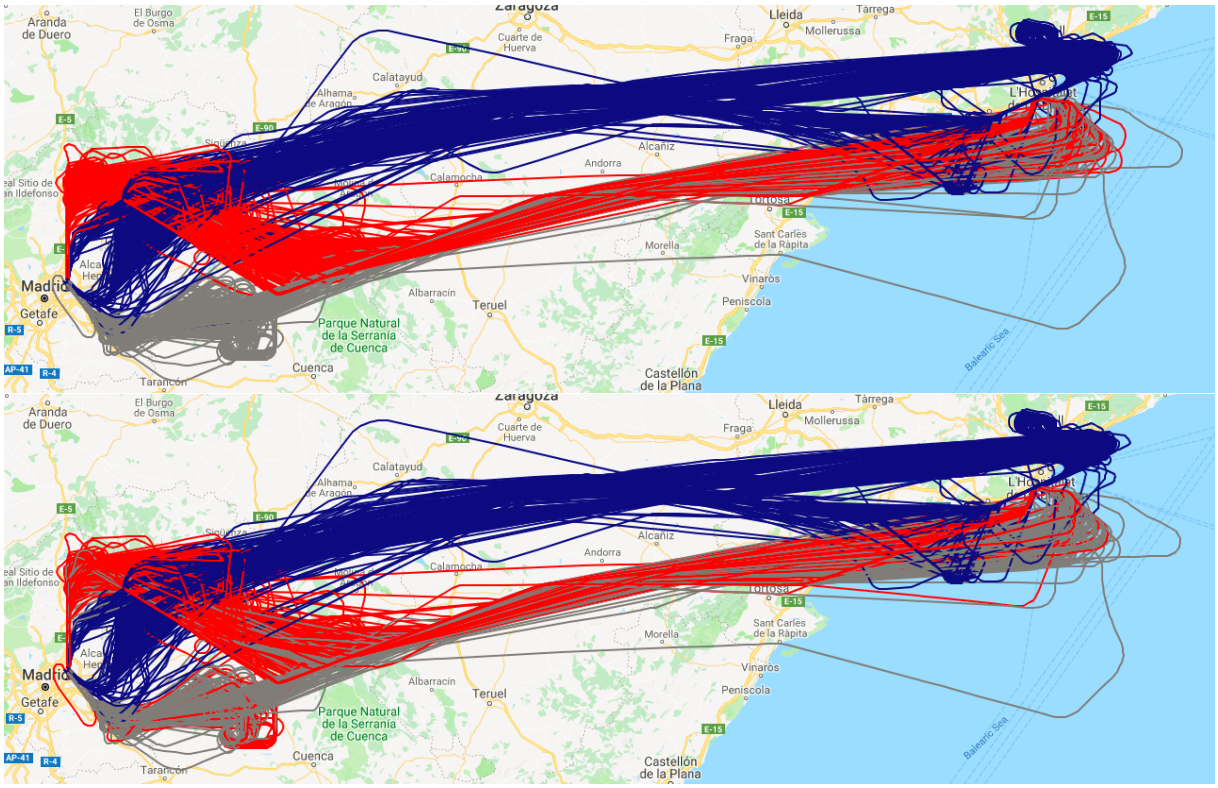
Σχήμα 4.13: Βέλτιστη Ομαδοποίηση για Διανυσματοποίηση 3D Τροχιών με Χρήση της Ανάλυσης Κυρίων Συνιστωσών.

Από τις παραπάνω εικόνες παρατηρούμε πως για κάθε διαφορετική διανυσματοποίηση τα αποτελέσματα είναι σχεδόν ίδια.

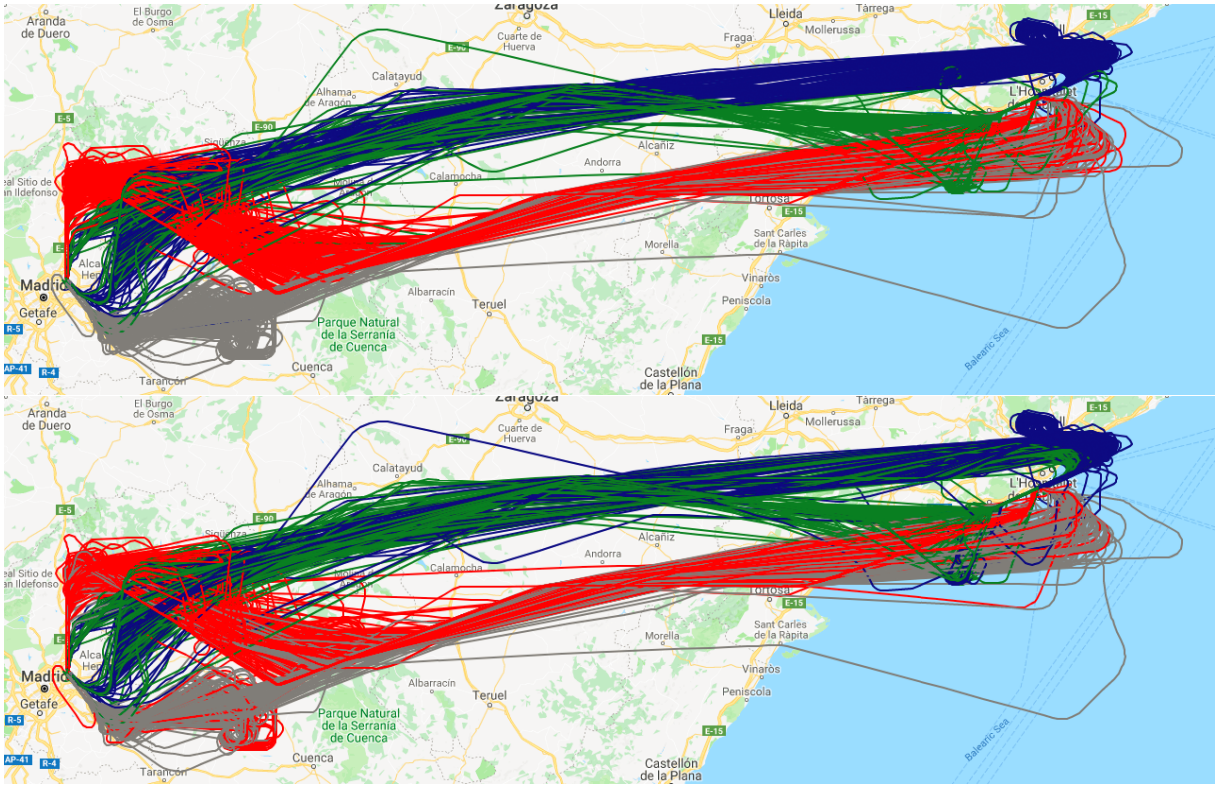
Στη συνέχεια ακολουθούν οι καλύτερες ομαδοποιήσεις ανεξαρτήτως αλγορίθμου και διανυσματοποίησης για 2, 3, 4 και 5 ομάδες.



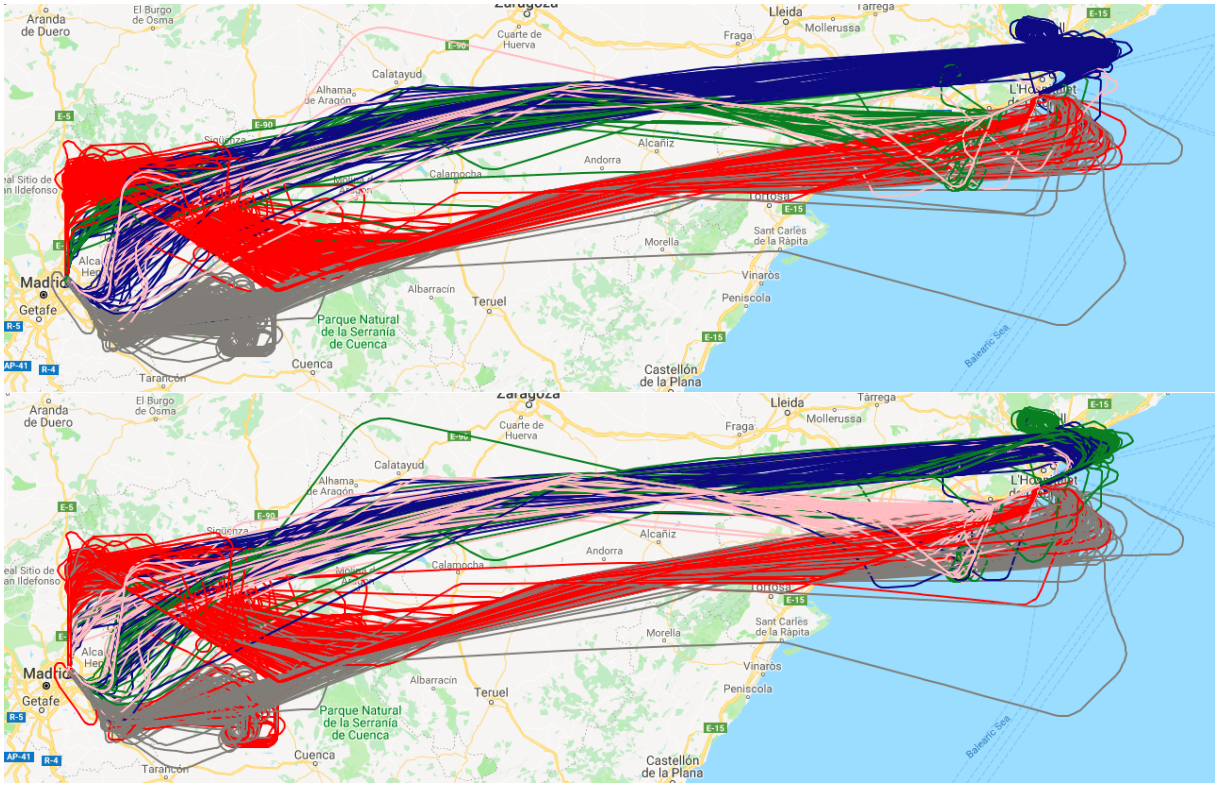
Σχήμα 4.14: Βέλτιστη Ομαδοποίηση για 2 ομάδες (κάτω) σε σχέση με το ground truth (πάνω).



Σχήμα 4.15: Βέλτιστη Ομαδοποίηση για 3 ομάδες (κάτω) σε σχέση με το ground truth (πάνω).



Σχήμα 4.16: Βέλτιστη Ομαδοποίηση για 4 ομάδες (κάτω) σε σχέση με το ground truth (πάνω).



Σχήμα 4.17: Βέλτιστη Ομαδοποίηση για 5 ομάδες (κάτω) σε σχέση με το ground truth (πάνω).

Αξίζει να σημειώσουμε ότι η βέλτιστη ομαδοποίηση στην περίπτωση των δύο ομάδων προέκυψε από όλους τους αλγορίθμους καθώς επιτύχανε την ίδια ομαδοποίηση και μάλιστα για την περίπτωση της διανυσματοποίησης σε 2D τροχιά. Όμως, για την περίπτωση των τριών ή περισσότερων ομάδων, η βέλτιστη ομαδοποίηση προήλθε μονάχα από τον αλγόριθμο K-means στην περίπτωση της διανυσματοποίησης σε 2D τροχιά και πάλι.

Σαν μια επιπλέον παρατήρηση θα μπορούσαμε να πούμε πως όσο οι τροχιές βρίσκονται στο μέσο της απόστασης Μαδρίτης-Βαρκελώνης είναι ξεκάθαρο πως εντοπίζονται δύο ομάδες. Βέβαια, όσο η πτήση βρίσκεται κοντά είτε στη Μαδρίτη, είτε στη Βαρκελώνη, παρατηρούνται διάφορες ομάδες για την προσγείωση και απογείωση αντίστοιχα.

Παρατηρώντας τα αποτελέσματα συγκεντρωτικά, αξίζει να εξετάσουμε κάποιες ενδιαφέρουσες περιπτώσεις όσον αφορά την ομαδοποίηση σε δύο ομάδες, η οποία κρίθηκε η αποτελεσματικότερη σύμφωνα με την παραπάνω ανάλυση.

Αρχικά θα εξετάσουμε κατά πόσο η αύξηση του μεγέθους διανυσματοποίησης βελτιώνει την αποτελεσματικότητα της ομαδοποίησης. Για το σκοπό αυτό, έχει νόημα να υπολογίσουμε το μέσο όρο που προκύπτει από τα τρία μέτρα απόδοσης Accuracy, Precision και F1-Score και από τους τρεις αριθμούς επαναλήψεων του κάθε αλγορίθμου και για καθένα από τα δύο σύνολα δεδομένων. Αξίζει να σημειώσουμε, πως ο μέσος όρος των μέτρων απόδοσης έχει νόημα καθώς αυτά είναι ποσοστά. Οι υπολογισμοί αυτοί παρουσιάζονται στους Πίνακες 4.42, 4.43, 4.44, 4.45. Παρατηρούμε λοιπόν πως καθώς αυξάνεται το μέγεθος της διανυσματοποίησης δεν παρατηρείται κάποια συστηματική βελτίωση στην αποτελεσματικότητα των αλγορίθμων. Αυτό ισχύει και για τις 4 τεχνικές διανυσματοποίησης, αλλά και για τους 4 αλγορίθμους που εξετάσαμε. Συγκεκριμένα, η μεγαλύτερη αύξηση που διαπιστώνεται, είναι της τάξης του 5% και εμφανίζεται στην περίπτωση της διανυσματοποίησης 3D κανονικοποιημένων και ευθυγραμμισμένων τροχιών.

Στη συνέχεια θα εξετάσουμε κατά πόσο τα συνοπτικά σύνολα δεδομένων επηρεάζουν την αποτελεσματικότητα των αλγορίθμων ομαδοποίησης. Βασισμένοι στους ίδιους πίνακες, παρατηρούμε πως στις περισσότερες περιπτώσεις η διαφορά των μέσων όρων ανάμεσα στα δύο σύνολα δεδομένων για όλες τις περιπτώσεις των αλγορίθμων και του μεγέθους διανυσματοποίησης δεν είναι σημαντική. Η μοναδική διαφοροποίηση ανάμεσα στα δύο σύνολα δεδομένων παρατηρείται στην περίπτωση του K-Means αλγορίθμου για τη διανυσματοποίηση των 3D κανονικοποιημένων και ευθυγραμμισμένων τροχιών. Η διαφοροποίηση αυτή είναι σημαντική καθώς η αποτελεσματικότητα της ομαδοποίησης όπως αυτή προκύπτει μέσα από τα μέτρα απόδοσης, φαίνεται σχεδόν να

διπλασιάζεται όταν ο αλγόριθμος εφαρμόζεται στο σύνολο των ακατέργαστων δεδομένων.

Αξίζει να σημειώσουμε πως οι παραπάνω συγκρίσεις δε λαμβάνουν υπόψην τα αποτελέσματα του GMM αλγορίθμου, καθώς αυτός στις περισσότερες περιπτώσεις αποτυγχάνει.

Σύνολο δεδομένων	Μέγεθος διανυσματοποίησης	Αλγόριθμοι			
		K-Means	GMM	PIC	Bisecting K-Means
Σύνολο ακατέργαστων δεδομένων	5	0.998	0.997	0.998	0.998
	10	0.998	0.996	0.998	0.998
	25	0.998	0.653	0.998	0.998
	50	0.998	0.607	0.998	0.998
Συνοπτικό σύνολο δεδομένων	5	0.998	0.997	0.998	0.998
	10	0.998	0.924	0.998	0.998
	25	0.998	0.851	0.998	0.998
	50	0.998	0.570	0.998	0.998

Πίνακας 4.42: Μέσοι όροι μέτρων απόδοσης για διανυσματοποίηση 2D τροχιών με χρήση παρεμβολής.

Σύνολο δεδομένων	Μέγεθος διανυσματοποίησης	Αλγόριθμοι			
		K-Means	GMM	PIC	Bisecting K-Means
Σύνολο ακατέργαστων δεδομένων	5	0.463	0.898	0.997	0.485
	10	0.477	-	0.998	0.477
	25	0.506	-	0.998	0.520
	50	0.504	-	0.998	0.519
Συνοπτικό Σύνολο δεδομένων	5	0.552	0.950	0.997	0.454
	10	0.502	-	0.998	0.505
	25	0.500	-	0.998	0.502
	50	0.504	-	0.998	0.503

Πίνακας 4.43: Μέσοι όροι μέτρων απόδοσης για διανυσματοποίηση 3D τροχιών με χρήση παρεμβολής.

Σύνολο δεδομένων	Μέγεθος διανυσματοποίησης	Αλγόριθμοι			
		K-Means	GMM	PIC	Bisecting K-Means
Σύνολο ακατέργαστων δεδομένων	5	0.844	0.664	0.994	0.844
	10	0.849	-	0.994	0.849
	25	0.853	-	0.994	0.853
	50	0.851	-	0.994	0.851
Συνοπτικό Σύνολο δεδομένων	5	0.454	0.720	0.994	0.835
	10	0.506	-	0.995	0.836
	25	0.503	-	0.994	0.852
	50	0.504	-	0.994	0.850

Πίνακας 4.44: Μέσοι όροι μέτρων απόδοσης για διανυσματοποίηση 3D κανονικοποιημένων και ευθυγραμμισμένων τροχιών.

Σύνολο δεδομένων	Μέγεθος διανυσματοποίησης	Αλγόριθμοι			
		K-Means	GMM	PIC	Bisecting K-Means
Σύνολο ακατέργαστων δεδομένων	5	0.851	0.625	0.610	0.851
	10	0.851	-	0.608	0.851
Συνοπτικό Σύνολο δεδομένων	5	0.848	0.592	0.495	0.847
	10	0.849	-	0.523	0.850

Πίνακας 4.45: Μέσοι όροι μέτρων απόδοσης για διανυσματοποίηση 3D τροχιών με χρήση Ανάλυσης Κυρίων Συνιστωσών.

Τέλος, ιδιαίτερο ενδιαφέρον παρουσιάζει η εξέταση τυχούσας διαφοροποίησης των αποτελεσμάτων της ομαδοποίησης στην περίπτωση της διανυσματοποίησης 3D τροχιών με χρήση Ανάλυσης Κυρίων Συνιστωσών για μεγέθη διανύσματος 5 και 10. Από τους πίνακες που δόθηκαν παραπάνω, παρατηρούμε ότι οι μέσοι όροι σχεδόν ταυτίζονται. Από τους πίνακες που δίνονται παρακάτω, οι οποίοι παρουσιάζουν τον αριθμό των τροχιών της κάθε ομάδας όπως αυτές προέκυψαν από τους αλγορίθμους ομαδοποίησης, βλέπουμε ότι καθώς αυξάνεται το μέγεθος

της διανυσματοποίησης δεν παρατηρείται κάποια σημαντική διαφοροποίηση στον αριθμό αυτό. Συγκεκριμένα η μεγαλύτερη διαφοροποίηση εμφανίζεται στον αλγόριθμο PIC στο συνοπτικό σύνολο δεδομένων και πρόκειται για 9 πτήσεις οι οποίες με την αύξηση του μεγέθους διανυσματοποίησης μεταφέρθηκαν από την πρώτη ομάδα στη δεύτερη.

Sampling = 5 longitudes / latitudes / altitudes													
Πλήθος ομάδων	Ομάδα	**K-means**			**GMM**			**PIC**			**Bisecting K-means**		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	830	830	830	829	955	916	702	700	700	830	830	830
	Cluster 2	566	566	566	567	441	480	694	696	696	566	566	566
Sampling = 10 longitudes / latitudes / altitudes													
k=2	Cluster 1	829	829	829	-	-	-	707	704	704	829	829	829
	Cluster 2	567	567	567	-	-	-	689	692	692	567	567	567

Πίνακας 4.46: Αριθμός τροχιών ανά ομάδα για το σύνολο των ακατέργαστων δεδομένων.

Sampling = 5 longitudes / latitudes / altitudes													
Πλήθος ομάδων	Ομάδα	**K-means**			**GMM**			**PIC**			**Bisecting K-means**		
		MaxIterations			MaxIterations			MaxIterations			MaxIterations		
		10	50	100	10	50	100	10	50	100	10	50	100
k=2	Cluster 1	820	822	820	821	476	931	537	533	533	820	820	820
	Cluster 2	576	574	576	575	920	465	859	863	863	576	576	576
Sampling = 10 longitudes / latitudes / altitudes													
k=2	Cluster 1	817	817	817	-	-	-	528	526	526	821	821	821
	Cluster 2	579	579	579	-	-	-	868	870	870	575	575	575

Πίνακας 4.47: Αριθμός τροχιών ανά ομάδα για το σύνολο των ακατέργαστων δεδομένων.

Βιβλιογραφία

[1] Ι. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης, Φ. Κόκκορας, Η. Σακελλαρίου (2011). *Τεχνητή Νοημοσύνη*. Εκδόσεις Πανεπιστημίου Μακεδονίας.

[2] Γ.Σ. Παπαγεωργίου, Χ.Γ. Τσίτουρας (2011). *Αριθμητική Ανάλυση με εφαρμογές σε Matlab και Mathematica*. Εκδόσεις Συμεών.

[3] Gan, Guojun, Chaoqun Ma, Jianhong Wu (2007). *Data Clustering: Theory, Algorithms, and Applications*. ASA-SIAM Series on Statistics and Applied Probability, SIAM, Philadelphia, ASA, Alexandria, VA.

[4] Sung-Hyuk Cha (2007). *Comprehensive Serey on Distanc/Similarity Measures between Probability Density Functions*. International Journal of Mathematical Models and Methods in Applied Sciences,Issue 4, Volume 1, 300-307.

[5] Rui Xu, Donald C. Wunsch II (2005). *Survey of Clustering Algorithms*. IEEE Trasactions on Neural Networks, Volume 16, No. 3, 645-678.

[6] Leonard Kaufman, Peter J. Rousseeuw (1990). *Finding Groups in Data: An Intoduction to Cluster Analysis*. John Wiley & Sons, Inc., Hoboken, New Jersey.

[7] Oded Maimon, Lior Rokach (2010). *Data Mining and Knowledge Discovery Handbook*. Springer New York Dordrecht Heidelberg London.

[8] Adil Fahad, Najlaa Alshatri, Zahir Tari, Abdullah Alamri, Ibrahim Khalil, Albert Y. Zomaya, Sebti Foufou, Abdelaziz Bouras (2004). *A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis*. IEEE Transactions on Emerging Topics in Computing, Volume 2, No3.

[9] Min Chen, Shiwen Mao, Yin Zhang, Victor C.M. Leung (2014). *Big Data, Related Technologies, Challenges and Future Prospects*. Springer Cham Heidelberg New York Dordrecht London.

[10] Kenneth Cukier (2010). *Data, data everywhere: A special report on managing information*. Economist Newspaper.

[11] John Gantz, David Reinsel (2011). *Extracting Value from Chaos*. IDC iView, 1–12.

[12] Ali S. Shirkhorshidi, Saeed Aghabozorgi, Teh Y. Wah, Tutut Herawan (2014). *Big Data Clustering: A Review*. ICCSA 2014, Guimarães, Portugal, Volume: 8583.

[13] Btissam Zerhari, Ayoub A. Lahcen, Salma Mouline (2015). *Big Data Clustering: Algorithms and Challenges*. International Conference on Big Data, Cloud and Applications BDCA'15, At Tetuan, Morocco.

[14] Tian Zhang, Raghu Ramakrishnan, Miron Livny (1996). *BIRCH: An Efficient Data Clustering Method for Very Large Databases*. SIGMOD '96, Montreal, Canada.

[15] Holdem Karau, Andy Konwinski, Patric Wendell, Matei Zaharia (2015). *Learning Spark Lightning-Fast Data Analysis*. O'Reilly Media, Inc., Sebastopol, CA.

[16] The Apache Software Foundation, Apache Spark (2017). https://spark.apache.org

[17] George A. Vouros, Akrivi Vlachou, Georgios M. Santipantakis, Christos Doulkeridis, Nikos Pelekis, Harris V. Georgiou, Yannis Theodoridis, Kostas Patroumpas, Elias Alevizos, Alexander Artikis, Christophe Claramunt, Cyril Ray, David Scarlatti, Georg Fuchs, Gennady L. Andrienko, Natalia V. Andrienko, Michael Mock, Elena Camossi, Anne-Laure Jousselme, Jose Manuel Cordero Garcia (2018). *Big Data Analytics for Time Critical Mobility Forecasting: Recent Progress and Research Challenges.* EDBT: 612-623

[18] Stylianos Sideridis, Nikos Pelekis, Yannis Theodoridis (2016). *On querying and mining semantic-aware mobility timelines*. I. J. Data Science and Analytics 2(1-2): 29-44

[19] Bahman Bahmani, Benjamin Moseley, Andrea Vattan, Ravi Kumar, Sergei Vassilvitskii (2012). *Scalable k-means++, Proceedings of the VLDB Endowment* v.5 n.7, p.622-633.

[20] Jia Li. *Data Mining - Clustering by Mixture Models*. http://personal.psu.edu/jol2/course/stat597e/notes/mix.pdf

[21] A.P. Dempster, N.M. Laird, and D.B. Rubin (1977). *Maximum Likelihood from Incomplete Data via the EM algorithm*, Journal of the Royal Statistical Society, Series B, vol. 39, 1:1-38

[22] F. Lin and W. W. Cohen (2012). *Power iteration clustering*. In ICML.

[23] G Karypis, V Kumar, M Steinbach (2009). *A comparison of document clustering techniques*. In KDD Workshop on Text Mining.

[24] Jolliffe I.T. (1990). *Principal Component Analysis: A Beginner's Guide- I. Introduction and application*. Weather, 45, 375–382.

Παράρτημα Α΄

Κώδικας για Διανυσματοποίηση σε 2D Τροχιά με Χρήση Παρεμβολής

Κώδικας για τη δημιουργία συνόλου δεδομένων που δέχονται σαν είσοδο οι αλγόριθμοι K-means, Gaussian Mixture Model και Bisecting K-means

```
sampling=5, 10, 25 or 50

#find unique values of flight ids
flightids=df.id.unique()

lat = [[0] * sampling] * len(flightids)
lon = [[0] * sampling] * len(flightids)

for i in range(len(flightids)):
flight1 = df.loc[df['id'] == flightids[i]]
flight11 = flight1.iloc[1:] #to remove row with dublicates
#flight11=flight1 # in case there are no dublicates
x1 = flight11['timestamp']

lon1 = flight11['longitude']
points = zip(x1, lon1)
points = sorted(points, key=lambda point: point[0])
x1, lon1 = zip(*points)
xf1 = np.array(x1)
lonf1 = np.array(lon1)
new_length = sampling
new_xf1 = np.linspace(xf1.min(), xf1.max(), new_length)
new_lonf1 = sp.interpolate.interp1d(xf1, lonf1, kind='cubic',fill_value="extrapolate")(new_xf1)

x1 = flight11['timestamp']
lat1 = flight11['latitude']
points = zip(x1, lat1)
points = sorted(points, key=lambda point: point[0])
x1, lat1 = zip(*points)
xf1 = np.array(x1)
latf1 = np.array(lat1)
new_latf1 = sp.interpolate.interp1d(xf1, latf1, kind='cubic',fill_value="extrapolate")(new_xf1)

lon[i]=new_lonf1
lat[i]=new_latf1
lon_lat=np.concatenate((lon, lat), axis=1)

np.savetxt('LonLat_BigDataSynops100', lon_lat, delimiter=' ')
```

Κώδικας για τη δημιουργία συνόλου δεδομένων που δέχεται σαν είσοδο ο αλγόριθμος Power Iteration Clustering

```
sampling=5, 10, 25 or 50

#find unique values of flight ids
flightids=df.id.unique()

lat = [[0] * sampling] * len(flightids)
lon = [[0] * sampling] * len(flightids)

for i in range(len(flightids)):
flight1 = df.loc[df['id'] == flightids[i]]
flight11 = flight1.iloc[1:] #to remove row with dublicates
#flight11=flight1 # in case there are no dublicates

x1 = flight11['timestamp']
lon1 = flight11['longitude']
points = zip(x1, lon1)
points = sorted(points, key=lambda point: point[0])
x1, lon1 = zip(*points)
xf1 = np.array(x1)
lonf1 = np.array(lon1)
new_length = sampling
new_xf1 = np.linspace(xf1.min(), xf1.max(), new_length)
new_lonf1 = sp.interpolate.interp1d(xf1, lonf1, kind='cubic',fill_value="extrapolate")(new_xf1)

x1 = flight11['timestamp']
lat1 = flight11['latitude']
points = zip(x1, lat1)
points = sorted(points, key=lambda point: point[0])
x1, lat1 = zip(*points)
xf1 = np.array(x1)
latf1 = np.array(lat1)
new_latf1 = sp.interpolate.interp1d(xf1, latf1, kind='cubic',fill_value="extrapolate")(new_xf1)

lon[i]=new_lonf1
lat[i]=new_latf1
lon_lat=np.concatenate((lon, lat), axis=1)

Distance=np.zeros((len(flightids),len(flightids)))

for i in range(len(flightids)):
for j in range(len(flightids)):
SEc0=0
for z in range(sampling):
SEc0 = SEc0 + gpxpy.geo.haversine_distance(lon_lat[i, z + sampling], lon_lat[i, z],
       lon_lat[j, z + sampling], lon_lat[j, z])
Distance[i,j]=(SEc0/sampling)/1000

print(Distance)

x=0
comb=[]
for i in range(len(flightids)):
for j in range(len(flightids)):
if i<=j:
x=x+1

DistanceMat = [[0] * 3]* x
print(DistanceMat)
y=0
for i in range(len(flightids)):
for j in range(len(flightids)):
if i<=j:
DistanceMat[y] = [i,j,Distance[i,j]]
y = y + 1

np.savetxt('DistanceMat_BigDataSynops10', DistanceMat, delimiter=' ')
```

Παράρτημα Β΄

Κώδικας για Διανυσματοποίηση σε 3D Τροχιά με Χρήση Παρεμβολής

Κώδικας για τη δημιουργία συνόλου δεδομένων που δέχονται σαν είσοδο οι αλγόριθμοι K-means, Gaussian Mixture Model και Bisecting K-means

```
Sampling=5, 10 ,25 or 50


#find unique values of flight ids
flightids=df.id.unique()

lat = [[0] * Sampling] * len(flightids)
lon = [[0] * Sampling] * len(flightids)
alt = [[0] * Sampling] * len(flightids)

for i in range(len(flightids)):
flight1 = df.loc[df['id'] == flightids[i]]
flight11 = flight1.iloc[1:] #to remove dublicates   #comment for whole dataset
#flight11=flight1                #comment for synopsis dataset

x1 = flight11['timestamp']
lon1 = flight11['longitude']
points = zip(x1, lon1)
points = sorted(points, key=lambda point: point[0])
x1, lon1 = zip(*points)
xf1 = np.array(x1)
lonf1 = np.array(lon1)
new_length = Sampling
new_xf1 = np.linspace(xf1.min(), xf1.max(), new_length)
new_lonf1 = sp.interpolate.interp1d(xf1, lonf1, kind='cubic',fill_value="extrapolate")(new_xf1)

x1 = flight11['timestamp']
lat1 = flight11['latitude']
points = zip(x1, lat1)
points = sorted(points, key=lambda point: point[0])
x1, lat1 = zip(*points)
xf1 = np.array(x1)
latf1 = np.array(lat1)
new_latf1 = sp.interpolate.interp1d(xf1, latf1, kind='cubic',fill_value="extrapolate")(new_xf1)

x1 = flight11['timestamp']
alt1 = flight11['altitude']
points = zip(x1, alt1)
points = sorted(points, key=lambda point: point[0])
x1, alt1 = zip(*points)
xf1 = np.array(x1)
altf1 = np.array(alt1)
new_altf1 = sp.interpolate.interp1d(xf1, altf1 ,kind='cubic',fill_value="extrapolate")(new_xf1)
```

```
lon[i]=new_lonf1
lat[i]=new_latf1
alt[i]=new_altf1
lon_lat_alt=np.concatenate((lon, lat,alt), axis=1)

np.savetxt('LonLatAlt_BigDataSynops5', lon_lat_alt, delimiter=' ')
```

Κώδικας για τη δημιουργία συνόλου δεδομένων που δέχεται σαν είσοδο ο αλγόριθμος Power Iteration Clustering

```
sampling=5,10,25 or 50

#find unique values of flight ids
flightids=df.id.unique()
print(flightids)

lat = [[0] * sampling] * len(flightids)
lon = [[0] * sampling] * len(flightids)
alt = [[0] * sampling] * len(flightids)

for i in range(len(flightids)):
flight1 = df.loc[df['id'] == flightids[i]]
flight11 = flight1.iloc[1:] #to remove dublicates       #comment for whole dataset
#flight11=flight1                          #comment for synopsis dataset

x1 = flight11['timestamp']
lon1 = flight11['longitude']
points = zip(x1, lon1)
points = sorted(points, key=lambda point: point[0])
x1, lon1 = zip(*points)
xf1 = np.array(x1)
lonf1 = np.array(lon1)
new_length = sampling
new_xf1 = np.linspace(xf1.min(), xf1.max(), new_length)
new_lonf1 = sp.interpolate.interp1d(xf1, lonf1, kind='cubic',fill_value="extrapolate")(new_xf1)

x1 = flight11['timestamp']
lat1 = flight11['latitude']
points = zip(x1, lat1)
points = sorted(points, key=lambda point: point[0])
x1, lat1 = zip(*points)
xf1 = np.array(x1)
latf1 = np.array(lat1)
new_latf1 = sp.interpolate.interp1d(xf1, latf1, kind='cubic',fill_value="extrapolate")(new_xf1)

x1 = flight11['timestamp']
alt1 = flight11['altitude']
points = zip(x1, alt1)
points = sorted(points, key=lambda point: point[0])
x1, alt1 = zip(*points)
xf1 = np.array(x1)
altf1 = np.array(alt1)
new_altf1 = sp.interpolate.interp1d(xf1, altf1, kind='cubic', fill_value="extrapolate")(new_xf1)

lon[i]=new_lonf1
lat[i]=new_latf1
alt[i]=new_altf1
lon_lat_alt=np.concatenate((lon, lat,alt), axis=1)

Distance=np.zeros((len(flightids),len(flightids)))
for i in range(len(flightids)):
for j in range(len(flightids)):
SEc0=0
for z in range(sampling):
dist = sqrt(((lon_lat_alt[j,z + 2*sampling]*cos(lon_lat_alt[j, z + sampling])
*sin(lon_lat_alt[j,z])) - (lon_lat_alt[i,z + 2*sampling]
*cos(lon_lat_alt[i, z + sampling])*sin(lon_lat_alt[i,z]))) ** 2
```

```
+ ((lon_lat_alt[j,z + 2*sampling]*sin(lon_lat_alt[j, z + sampling]))
- (lon_lat_alt[i,z + 2*sampling]*sin(lon_lat_alt[i, z + sampling]))) ** 2
+ ((lon_lat_alt[j,z + 2*sampling]*cos(lon_lat_alt[j, z + sampling])*cos(lon_lat_alt[j,z]))
- (lon_lat_alt[i,z + 2*sampling]*cos(lon_lat_alt[i, z + sampling])*cos(lon_lat_alt[i,z]))) ** 2)
SEc0 = SEc0 + dist
Distance[i,j]=(SEc0/sampling)/1000

x=0
comb=[]
for i in range(len(flightids)):
for j in range(len(flightids)):
if i<=j:
x=x+1

DistanceMat = [[0] * 3]* x
print(DistanceMat)
y=0
for i in range(len(flightids)):
for j in range(len(flightids)):
if i<=j:
DistanceMat[y] = [i,j,Distance[i,j]]
y = y + 1
np.savetxt('DistanceMat_BigData5', DistanceMat, delimiter=' ')
```

Παράρτημα Γ΄

Κώδικας για Διανυσματοποίηση σε 3D Κανονικοποιημένη και Ευθυγραμμισμένη Τροχιά

Κώδικας για τη δημιουργία συνόλου δεδομένων που δέχονται σαν είσοδο οι αλγόριθμοι K-means, Gaussian Mixture Model και Bisecting K-means

```
sampling=5, 10, 25 or 20

#find unique values of flight ids
flightids=df.id.unique()

MaxDif=0
for i in range(len(flightids)):
flight1 = df.loc[df['id'] == flightids[i]]
flight11 = flight1.iloc[1:] #to remove dublicates        #comment for whole dataset
#flight11=flight1                          #comment for synopsis dataset
mintimestamp0=flight11['timestamp'].min()
maxtimestamp0=flight11['timestamp'].max()
Diff=maxtimestamp0-mintimestamp0
if MaxDif<Diff:
MaxDif=Diff

df2 = pd.DataFrame()
for i in range(len(flightids)):
flight1 = df.loc[df['id'] == flightids[i]]
flight11 = flight1.iloc[1:] #to remove dublicates        #comment for whole dataset
#flight11=flight1                          #comment for synopsis dataset
mintimestamp0 = flight11['timestamp'].min()
maxtimestamp0 = flight11['timestamp'].max()
transf=(flight11['timestamp'] - mintimestamp0) / MaxDif
flight11=pd.concat([flight11, transf.rename("timestamp2")], axis=1)
df2=pd.concat([df2, flight11])

lon = [[0] * sampling]* len(flightids)
lat = [[0] * sampling]* len(flightids)
alt = [[0] * sampling]* len(flightids)

for j in range(len(flightids)):
flight1 = df2.loc[df2['id'] == flightids[j]]
flight11 = flight1.iloc[1:] #to remove dublicates        #comment for whole dataset
#flight11=flight1                          #comment for synopsis dataset
maxtimestamp0 = flight11['timestamp2'].max()
longit2=[0] * sampling
latit2=[0] * sampling
altit2=[0] * sampling
for i in range(sampling):
```

```
x = i / float(sampling)
if(x<=maxtimestamp0):
idx = (np.abs(flight11['timestamp2'] - x)).idxmin()
longit=flight11.loc[idx,'longitude']
latit=flight11.loc[idx,'latitude']
altit = flight11.loc[idx, 'altitude']
ts = flight11.loc[idx, 'timestamp2']
longit2[i]=longit
latit2[i]=latit
altit2[i]=altit
else:
idx = (np.abs(flight11['timestamp2'])).idxmax()
longit=flight11.loc[idx,'longitude']
latit=flight11.loc[idx,'latitude']
altit = flight11.loc[idx, 'altitude']
longit2[i]=longit
latit2[i]=latit
altit2[i]=altit
lon[j]=longit2
lat[j]=latit2
alt[j]=altit2
lon_lat_alt=np.concatenate((lon, lat,alt), axis=1)

np.savetxt('LonLatAlt_BigData5.txt', lon_lat_alt, delimiter=' ')
```

Κώδικας για τη δημιουργία συνόλου δεδομένων που δέχεται σαν είσοδο ο αλγόριθμος Power Iteration Clustering

```
sampling=5, 10, 25 ή 50

#find unique values of flight ids
flightids=df.id.unique()

MaxDif=0
for i in range(len(flightids)):
flight1 = df.loc[df['id'] == flightids[i]]
flight11 = flight1.iloc[1:] #to remove dublicates        #comment for whole dataset
#flight11=flight1                          #comment for synopsis dataset
mintimestamp0=flight11['timestamp'].min()
maxtimestamp0=flight11['timestamp'].max()
Diff=maxtimestamp0-mintimestamp0
if MaxDif<Diff:
MaxDif=Diff


df2 = pd.DataFrame()
for i in range(len(flightids)):
flight1 = df.loc[df['id'] == flightids[i]]
flight11 = flight1.iloc[1:] #to remove dublicates        #comment for whole dataset
#flight11=flight1                          #comment for synopsis dataset
mintimestamp0 = flight11['timestamp'].min()
maxtimestamp0 = flight11['timestamp'].max()
transf=(flight11['timestamp'] - mintimestamp0) / MaxDif
flight11=pd.concat([flight11, transf.rename("timestamp2")], axis=1)
df2=pd.concat([df2, flight11])

lon = [[0] * sampling]* len(flightids)
lat = [[0] * sampling]* len(flightids)
alt = [[0] * sampling]* len(flightids)

for j in range(len(flightids)):
flight1 = df2.loc[df2['id'] == flightids[j]]
flight11 = flight1.iloc[1:] #to remove dublicates        #comment for whole dataset
#flight11=flight1                          #comment for synopsis dataset
maxtimestamp0 = flight11['timestamp2'].max()
longit2=[0] * sampling
latit2=[0] * sampling
altit2=[0] * sampling
```

```
for i in range(sampling):
x = i / float(sampling)
if(x<=maxtimestamp0):
idx = (np.abs(flight11['timestamp2'] - x)).idxmin()
longit=flight11.loc[idx,'longitude']
latit=flight11.loc[idx,'latitude']
altit = flight11.loc[idx, 'altitude']
ts = flight11.loc[idx, 'timestamp2']
longit2[i]=longit
latit2[i]=latit
altit2[i]=altit
else:
idx = (np.abs(flight11['timestamp2'])).idxmax()
longit=flight11.loc[idx,'longitude']
latit=flight11.loc[idx,'latitude']
altit = flight11.loc[idx, 'altitude']
longit2[i]=longit
latit2[i]=latit
altit2[i]=altit
lon[j]=longit2
lat[j]=latit2
alt[j]=altit2
lon_lat_alt=np.concatenate((lon, lat,alt), axis=1)

Distance=np.zeros((len(flightids),len(flightids)))
for i in range(len(flightids)):
for j in range(len(flightids)):
SEc0=0
for z in range(sampling):
dist = sqrt(((lon_lat_alt[j,z + 2*sampling]*cos(lon_lat_alt[j, z + sampling])
*sin(lon_lat_alt[j,z])) - (lon_lat_alt[i,z + 2*sampling]*cos(lon_lat_alt[i, z +
 sampling])*sin(lon_lat_alt[i,z]))) ** 2
+ ((lon_lat_alt[j,z + 2*sampling]*sin(lon_lat_alt[j, z + sampling])) -
 (lon_lat_alt[i,z + 2*sampling]*sin(lon_lat_alt[i, z + sampling]))) ** 2
+ ((lon_lat_alt[j,z + 2*sampling]*cos(lon_lat_alt[j, z + sampling])
*cos(lon_lat_alt[j,z])) - (lon_lat_alt[i,z + 2*sampling]
*cos(lon_lat_alt[i, z + sampling])*cos(lon_lat_alt[i,z]))) ** 2)
SEc0 = SEc0 + dist
Distance[i,j]=(SEc0/sampling)/1000

x=0
comb=[]
for i in range(len(flightids)):
for j in range(len(flightids)):
if i<=j:
x=x+1

DistanceMat = [[0] * 3]* x
y=0
for i in range(len(flightids)):
for j in range(len(flightids)):
if i<=j:
DistanceMat[y] = [i,j,Distance[i,j]]
y = y + 1
np.savetxt('DistanceMat_BigData5', DistanceMat, delimiter=' ')
```

Παράρτημα Δ΄

Κώδικας για Διανυσματοποίηση σε 3D Τροχιά με Χρήση της Ανάλυσης Κυρίων Συνιστωσών

Αλγόριθμος Ανάλυσης Κυρίων Συνιστωσών

```
from pyspark.context import SparkContext
from pyspark.sql.session import SparkSession
sc = SparkContext('local')
spark = SparkSession(sc)
from pyspark.sql import SQLContext
sqlContext = SQLContext(sc)

from pyspark.mllib.linalg
import Vectors
from pyspark.mllib.linalg.distributed
import RowMatrix

data = sc.textFile('/home/stefanos/Desktop/PCA/wholedataset/Lat_BigData50.txt')
rows = data.map(lambda line: array([float(x) for x in line.strip().split(' ')]))

mat = RowMatrix(rows)

# Compute the top 5 or 10 principal components.
# Principal components are stored in a local dense matrix.
pc = mat.computePrincipalComponents(5 or 10)

# Project the rows to the linear space spanned by the top 5 or 10 principal components.
projected = mat.multiply(pc)
collected = projected.rows.collect()

np.savetxt('PCALat10.txt', collected, delimiter=' ')
```

Κώδικας για τη δημιουργία συνόλου δεδομένων που δέχεται σαν είσοδο ο αλγόριθμος PIC

```
lon_lat_alt2= pd.read_csv('Path for dataframe came up after PCA',
              delimiter=' ', header=None)
lon_lat_alt = lon_lat_alt2.as_matrix()

sampling=5 or 10


Distance=np.zeros((len(lon_lat_alt),len(lon_lat_alt)))
for i in range(len(lon_lat_alt)):
```

```
for j in range(len(lon_lat_alt)):
SEc0=0
for z in range(sampling):
dist = sqrt(((lon_lat_alt[j,z + 2*sampling]*cos(lon_lat_alt[j, z + sampling])
*sin(lon_lat_alt[j,z])) - (lon_lat_alt[i,z + 2*sampling]
*cos(lon_lat_alt[i, z + sampling])*sin(lon_lat_alt[i,z]))) ** 2
+ ((lon_lat_alt[j,z + 2*sampling]*sin(lon_lat_alt[j, z + sampling])) -
(lon_lat_alt[i,z + 2*sampling]*sin(lon_lat_alt[i, z + sampling]))) ** 2
+ ((lon_lat_alt[j,z + 2□sampling]*cos(lon_lat_alt[j, z + sampling])
*cos(lon_lat_alt[j,z])) - (lon_lat_alt[i,z + 2*sampling]
*cos(lon_lat_alt[i, z + sampling])*cos(lon_lat_alt[i,z]))) ** 2)
SEc0 = SEc0 + dist
Distance[i,j]=(SEc0/sampling)/1000

x=0
comb=[]
for i in range(len(lon_lat_alt)):
for j in range(len(lon_lat_alt)):
if i<=j:
x=x+1


DistanceMat = [[0] * 3]* x
y=0
for i in range(len(lon_lat_alt)):
for j in range(len(lon_lat_alt)):
if i<=j:
DistanceMat[y] = [i,j,Distance[i,j]]
y = y + 1

np.savetxt('PCADistanceMat_BigData5', DistanceMat, delimiter=' ')
```

Σημειώνουμε ότι το σύνολο δεδομένων που διαβάζει ο παρακάτω κώδικας είναι αυτό που προέκυψε μετά την PCA, δηλαδή αυτό που δέχτηκαν σαν είσοδο οι αλγόριθμοι K-means, Gaussian Mixture Model και Bisecting K-means.

Παράρτημα Ε΄

Αλγόριθμοι Ομαδοποίησης

Αλγόριθμος K-means

```
import time
from pyspark.mllib.clustering import KMeans, KMeansModel
from pyspark.context import SparkContext
from pyspark.sql.session import SparkSession
sc = SparkContext('local')
spark = SparkSession(sc)
from pyspark.sql import SQLContext
sqlContext = SQLContext(sc)

start_time = time.clock()

sampling=2, 3, 4 or 5
Iterations=10, 50 or 100

data = sc.textFile('/home/stefanos/Desktop/thesis/NewData/LonLat_BigData50.txt')
parsedData = data.map(lambda line: array([float(x) for x in line.split(' ')]))
clusters = KMeans.train(parsedData, sampling, maxIterations= Iterations,
initializationMode="random")

def error(point):
center = clusters.centers[clusters.predict(point)]
return sqrt(sum([x ** 2 for x in (point - center)]))

WSSSE = parsedData.map(lambda point: error(point)).reduce(lambda x, y: x + y)
print("Within Set Sum of Squared Error = " + str(WSSSE))

#cluster centers, represented as a list of NumPy arrays
print(clusters.clusterCenters)

#Total number of clusters
print("The Total Number of Clusters is: " + str(clusters.k))

#Return the K-means cost (sum of squared distances of points to their nearest center)
print("The Cost of K-means is: " +str(clusters.computeCost(parsedData)))

#Find the cluster that each of the points belongs to in this model.
prediction = clusters.predict(parsedData)
cluster_number_predicted = prediction.collect()
print(cluster_number_predicted)

print time.clock() - start_time, "seconds"
```

Αλγόριθμος Gaussian Mixture Model

```
import time
from pyspark.mllib.clustering import GaussianMixture, GaussianMixtureModel
from pyspark.context import SparkContext
from pyspark.sql.session import SparkSession
sc = SparkContext('local')
spark = SparkSession(sc)
from pyspark.sql import SQLContext
sqlContext = SQLContext(sc)

start_time = time.clock()

sampling=2, 3, 4 or 5
Iterations=10, 50 or 100

# Load and parse the data
data = sc.textFile('/home/stefanos/Desktop/thesis/NewData/LonLat_BigData50.txt')
parsedData = data.map(lambda line: array([float(x) for x in line.strip().split(' ')]))

# Build the model (cluster the data)
gmm = GaussianMixture.train(parsedData, sampling, maxIterations= Iterations)

# output parameters of model
for i in range(sampling):
print("weight = ", gmm.weights[i], "mu = ", gmm.gaussians[i].mu)

prediction = gmm.predict(parsedData)
cluster_number_predicted = prediction.collect()
print(cluster_number_predicted)

print time.clock() - start_time, "seconds"
```

Αλγόριθμος Power Iteration Clustering

```
import time
from pyspark.mllib.clustering import PowerIterationClustering, PowerIterationClusteringModel
from pyspark.context import SparkContext
from pyspark.sql.session import SparkSession
sc = SparkContext('local')
spark = SparkSession(sc)
from pyspark.sql import SQLContext
sqlContext = SQLContext(sc)

start_time = time.clock()

sampling=2, 3, 4 or 5
Iterations=10, 50 or 100

# Load and parse the data
data = sc.textFile('/home/stefanos/Desktop/thesis/NewData/DistanceMat_BigData50')
similarities = data.map(lambda line: tuple([float(x) for x in line.split(' ')]))

# Cluster the data into two classes using PowerIterationClustering
model = PowerIterationClustering.train(similarities, sampling, Iterations)
print(model.k)
result = sorted(model.assignments().collect(), key=lambda x: x.id)
print(result)

print(len(result))
x=[0]□len(result)
for i in range(len(result)):
x[i]=result[i].cluster
print(x)

print time.clock() - start_time, "seconds"
```

Αλγόριθμος Bisecting K-means

```
import time
from pyspark.mllib.clustering import BisectingKMeans, BisectingKMeansModel
from pyspark.context import SparkContext
from pyspark.sql.session import SparkSession
sc = SparkContext('local')
spark = SparkSession(sc)
from pyspark.sql import SQLContext
sqlContext = SQLContext(sc)

start_time = time.clock()

sampling=2, 3, 4 or 5
Iterations=10, 50 or 100

# Load and parse the data
data = sc.textFile('/home/stefanos/Desktop/thesis/NewData/LonLat_BigData50.txt')
parsedData = data.map(lambda line: array([float(x) for x in line.strip().split(' ')]))

# Build the model (cluster the data)
model = BisectingKMeans.train(parsedData, sampling, maxIterations= Iterations)

# Evaluate clustering
cost = model.computeCost(parsedData)
print("Bisecting K–means Cost = " + str(cost))

#cluster centers, represented as a list of NumPy arrays
print(model.clusterCenters)

#Total number of clusters
print("The Total Number of Clusters is: " + str(model.k))

prediction = model.predict(parsedData)
cluster_number_predicted = prediction.collect()
print(cluster_number_predicted)

print time.clock() – start_time, "seconds"
```

Παράρτημα Ϛ΄

Κώδικας για Μέτρα Αξιολόγησης των Αλγορίθμων Ομαδοποίησης

Κώδικας που υπολογίζει τα μέτρα απόδοσης Accuracy, Precision, F1 Score και RMSE του κεντροειδούς από τα μέλη της ομάδας, για την περίπτωση των 2 ομάδων.

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import f1_score
from math import sqrt
from math import sin, cos, sqrt, atan2, radians
from scipy.stats import itemfreq

Interp=5,10,25 or 50 #sampling

df= pd.read_csv('Path for dataframe',delimiter=',')

#find unique values of flight ids
flightids=df.id.unique()

#Cluster Centre 0
c0=[Coordinates of cluster 0, as given by the clustering algorithms]

#Cluster Centre 1
c1=[Coordinates of cluster 1, as given by the clustering algorithms]

#Classification according to ground truth
classif=[Vector with the clustering of flights based on ground truth]
print("Classification of flights according to Ground Truth: " + str(itemfreq( classif )))

#prediction of algorithm
pred2=[Vector with the clustering of flights based on the algorithm]

pred22 = [0] * len(flightids)
for i in range(len(pred2)):
if pred2[i]==0:
pred22[i]=0
if pred2[i]==1:
pred22[i] = 1

acc=accuracy_score(classif, pred22)
prec=precision_score(classif, pred22,average='binary')
f1=f1_score(classif, pred22,average='binary')
predf=pred22

pred22 = [0] * len(flightids)
for i in range(len(pred2)):
if pred2[i]==0:
pred22[i]=1
```

```
if pred2[i]==1:
pred22[i] = 0
if accuracy_score(classif, pred22)>acc:
acc = accuracy_score(classif, pred22)
prec = precision_score(classif, pred22, average='binary')
f1 = f1_score(classif, pred22, average='binary')
temp=c0
c0=c1
c1=temp
predf=pred22

print("Classification of flights according to Algorithm: " + str(itemfreq( predf )))
print("Accuracy: " + str(acc))
print("Precision: " + str(prec))
print("F1 Score: " + str(f1))

lat = [[0] * Interp] * len(flightids)
lon = [[0] * Interp] * len(flightids)
for i in range(len(flightids)):
flight1 = df.loc[df['id'] == flightids[i]]
flight11 = flight1.iloc[1:] #to remove dublicates   #comment this for whole dataset
#flight11 = flight1             #comment this for synopsis dataset

x1 = flight11['timestamp']
lon1 = flight11['longitude']
points = zip(x1, lon1)
points = sorted(points, key=lambda point: point[0])
x1, lon1 = zip(*points)
xf1 = np.array(x1)
lonf1 = np.array(lon1)
new_length = Interp
new_xf1 = np.linspace(xf1.min(), xf1.max(), new_length)
new_lonf1 = sp.interpolate.interp1d(xf1, lonf1, kind='cubic',fill_value="extrapolate")(new_xf1)

x1 = flight11['timestamp']
lat1 = flight11['latitude']
points = zip(x1, lat1)
points = sorted(points, key=lambda point: point[0])
x1, lat1 = zip(*points)
xf1 = np.array(x1)
latf1 = np.array(lat1)
new_latf1 = sp.interpolate.interp1d(xf1, latf1, kind='cubic',fill_value="extrapolate")(new_xf1)
lon[i]=new_lonf1
lat[i]=new_latf1
lon_lat=np.concatenate((lon, lat), axis=1)

c0_np=np.array(c0)
c1_np=np.array(c1)

SEc0=0
N0=0
SEc1=0
N1=0
SEc2=0
N2=0

for i in range(len(flightids)):
if classif[i]==0:
for j in range(Interp):
SEc0 = SEc0+gpxpy.geo.haversine_distance(lon_lat[i,j+Interp], lon_lat[i,j],
       c0_np[j+Interp], c0_np[j])
N0=N0+1
if classif[i]==1:
for j in range(Interp):
SEc1 =SEc1+ gpxpy.geo.haversine_distance(lon_lat[i, j + Interp], lon_lat[i, j],
      c1_np[j + Interp],c1_np[j])
N1=N1+1
RMSEc0=((SEc0/Interp)/N0)/1000
RMSEc1=((SEc1/Interp)/N1)/1000
```

```
print("RMSE Cluster 0: " + str(RMSEc0))
print("RMSE Cluster 1: " + str(RMSEc1))
```

Στην περίπτωση του PIC αλγορίθμου, ο οποίος δε δίνει σαν αποτέλεσμα τα κέντρα των ομάδων, τα c_0 και c_1 βρίσκονται όπως φαίνεται στον επόμενο κώδικα

```
Interp=5,10,25 or 50

df= pd.read_csv('Path for dataframe',delimiter=',')

flightids=df.id.unique()
pred2=[Vector with the clustering of flights based on the algorithm]

aList =list()
bList=list()
for i in range(len(pred2)):
if pred2[i]==0:
aList.append(i)
if pred2[i]==1:
bList.append(i)


lat = [[0] * Interp] * len(flightids)
lon = [[0] * Interp] * len(flightids)
for i in range(len(flightids)):
flight1 = df.loc[df['id'] == flightids[i]]
flight11 = flight1.iloc[1:]      #to remove dublicates #comment for whole dataset
#flight11 = flight1                    #comment for synopsis dataset

x1 = flight11['timestamp']
lon1 = flight11['longitude']
points = zip(x1, lon1)
points = sorted(points, key=lambda point: point[0])
x1, lon1 = zip(*points)
xf1 = np.array(x1)
lonf1 = np.array(lon1)
new_length = Interp
new_xf1 = np.linspace(xf1.min(), xf1.max(), new_length)
new_lonf1 = sp.interpolate.interp1d(xf1, lonf1, kind='cubic',fill_value="extrapolate")(new_xf1)

x1 = flight11['timestamp']
lat1 = flight11['latitude']
points = zip(x1, lat1)
points = sorted(points, key=lambda point: point[0])
x1, lat1 = zip(*points)
xf1 = np.array(x1)
latf1 = np.array(lat1)
new_latf1 = sp.interpolate.interp1d(xf1, latf1, kind='cubic',fill_value="extrapolate")(new_xf1)
lon[i]=new_lonf1
lat[i]=new_latf1
lon_lat=np.concatenate((lon, lat), axis=1)

lon_lat0 = [[0] * 2*Interp] * len(aList)
lon_lat1 = [[0] * 2*Interp] * len(bList)
a=0
b=0
for i in range(len(pred2)):
if pred2[i]==0:
lon_lat0[a]=lon_lat[i]
a=a+1
if pred2[i]==1:
lon_lat1[b] = lon_lat[i]
b=b+1

lon_lat00=np.array(lon_lat0)
centroid0=[0] * 2*Interp
for j in range(2*Interp):
x=0
```

```
for i in range(len(aList)):
x=x+lon_lat00[i,j]
centroid0[j]=x/len(aList)

lon_lat11=np.array(lon_lat1)
centroid1=[0] * 2*Interp
for j in range(2*Interp):
x=0
for i in range(len(bList)):
x=x+lon_lat11[i,j]
centroid1[j]=x/len(bList)

print("Centroid of Cluster 0: " + str(centroid0))
print("Centroid of Cluster 1: " + str(centroid1))
```