Advanced Analysis and Visualisation Techniques for Atmospheric Data

Richard W. Hyde

School of Computing and Communications
Lancaster University

This dissertation is submitted for the degree of
Doctor of Philosophy

Lancaster University September 2017

This thesis is dedicated to the memory of my parents, they knew I could do it and now, finally, I have.

Declaration

I hereby declare that, except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains approximately 65,000 words including appendices, bibliography, footnotes, tables and equations and has approximately 50 figures.

Richard W. Hyde
September 2017

Acknowledgements

This thesis would not have been possible without the help of many people. I would like to thank the Natural Environmental Research Council (NERC) whose funding allowed me to pay the bills during my research. Professor Plamen Angelov, my supervisor, for allowing me the freedom to explore, yet always being available to help and guide, and without whose knowledge of clustering techniques and their applications I may still be wandering. The insights provided by Professor Rob MacKenzie as co-supervisor proved invaluable in ensuring that the application of the work to atmospheric science has significance and meaning.

This thesis has added relevance for the application of the work to real atmospheric data. I would like to acknowledge the help of Dr James Levine for pointing me towards the SAMBBA data that demonstrate suitable results. I would also like to thank Professor Steven Wofsy for his knowledge of the Hiaper Pole to Pole Observation (HIPPO) data and Dr Gary Fuller for his help with the London Air Quality data. Finally I'd like to thank Professor Neil Harris, principle investigator of the Coordinated Airborne Studies in the Tropics (CAST) of which this work is a part for his understanding and support of the role of computer science in the analysis of atmospheric science data.

On a personal note I have to acknowledge my wonderful partner Sofia whose love and support throughout my studies has been invaluable, I could not have done this without her. My step-daughter, Anna, has also been a source of inspiration achieving great results at school and reminding me how hard work and dedication pays off. Finally, a big thank you to the National Chopper Club, a part of my life for over 20 years, for their continued friendship and support through good times and bad.

Abstract

Atmospheric science is the study of a large, complex system which is becoming increasingly important to understand. There are many climate models which aim to contribute to that understanding by computational simulation of the atmosphere. To generate these models, and to confirm the accuracy of their outputs, requires the collection of large amounts of data. These data are typically gathered during campaigns lasting a few weeks, during which various sources of measurements are used. Some are ground based, others airborne sondes, but one of the primary sources is from measurement instruments on board aircraft. Flight planning for the numerous sorties is based on pre-determined goals with unpredictable influences, such as weather patterns, and the results of some limited analyses of data from previous sorties. There is little scope for adjusting the flight parameters during the sortie based on the data received due to the large volumes of data and difficulty in processing the data online. The introduction of unmanned aircraft with extended flight durations also requires a team of mission scientists with the added complications of disseminating observations between shifts.

Earth's atmosphere is a non-linear system, whereas the data gathered is sampled at discrete temporal and spatial intervals introducing a source of variance. Clustering data provides a convenient way of grouping similar data while also acknowledging that, for each discrete sample, a minor shift in time and/ or space could produce a range of values which lie within its cluster region. This thesis puts forward a set of requirements to enable the presentation of cluster analyses to the mission scientist in a convenient and functional manner. This will enable in-flight decision making as well as rapid feedback for future flight planning.

Current state of the art clustering algorithms are analysed and a solution to all of the proposed requirements is not found. New clustering algorithms are developed to achieve these goals. These novel clustering algorithms are brought together, along with other visualization techniques, into a software package which is used to demonstrate how the analyses can provide information to mission scientists in flight. The ability to carry out offline analyses on historical data, whether to reproduce the online analyses of the current sortie, or to provide comparative analyses from previous missions, is also

demonstrated. Methods for offline analyses of historical data prior to continuing the analyses in an online manner are also considered.

The original contributions in this thesis are the development of five new clustering algorithms which address key challenges: speed and accuracy for typical hyper-elliptical offline clustering; speed and accuracy for offline arbitrarily shaped clusters; online dynamic and evolving clustering for arbitrary shaped clusters; transitions between offline and online techniques and also the application of these techniques to atmospheric science data analysis.

Table of contents

List of figures

List of tables

List of Algorithms

Nomenclature

δ_{in} Smoothed density drop: a term used in DDCAR, the mean density drop over 'n ' data samples prior to data sample i

ε a cluster parameter typically used to define a local radius

μ_0 Global Mean: a term used in RDE calculations, the mean of all data samples in a data set

μ_l the local mean of data samples

$\overline{\delta}$ Mean density drop: a term used in DDCAR, the mean density drop between adjacent data samples

$\{C_j\}$ The set of data assigned to cluster j

$\{Data\}$ The set of data

$alphahulls$ a method for drawing around data points that allows for concavity where convex hulls do not

$C_\mu(i)$ CODAS mean of the data assigned to micro-cluster i, used as the centre

$C_i(Centre)$ CEDAS, the centre of micro-cluster i

$C_i(Count)$ CEDAS, the number of data assigned to micro-cluster i

$C_i(Edge)$ CEDAS, the graph edges of micro-cluster i

$C_i(Macro)$ CEDAS, the macro-cluster assignment of micro-cluster i

$C_m(i)$ CODAS macro-cluster assignment of micro-cluster i

$C_n(i)$ CODAS number of data samples assigned to micro-cluster i

$C_r(i)$ CODAS radius of micro-cluster i. C_r has the same value for all micro-clusters, but is included for possible future use with variable radii.

D_i Global Density: a term used in RDE calculations

d_{min} the minimum distance between data and micro-cluster centres

i generic index

k used in various clustering techniques to indicate the number of clusters

MC macro-cluster - an agglomeration of micro-clusters which form the final clusters of data the clustering technique has defined as similar.

mC micro-cluster - small clusters of data limited to a local data space region, agglomerations of which form macro-clusters

N Numeric count, specified locally in the text

n generic count specified locally in the text

$natural\,cluster$ the clusters that are naturally present in the original data, which may differ from those found by a clustering technique.

$O(D)$ Complexity notation based on the number of data dimensions D

$O(n)$ Complexity notation based on the number of samples n

r_0 initial radius for forming clusters, or micro-clusters

$subspace$ where the full data space contains D dimensions, then a subspace is any part of that data space with n dimensions where $1 \leq n < D$

T_{min} minimum threshold for a micro-cluster to be considered part of a cluster rather than a group of outlier or noise data

X_0 Scalar Product: a term used in RDE calculations

x_{ij} clustered data sample i in cluster j

x_i un-clustered data sample i

X_l the local scalar product, similar to X_0 but for local data samples

Chapter 1

Aims, Objectives and Structure of the Thesis

1.1 Aims and Objectives

The role of atmospheric science, the data, its analysis and models has achieved great importance in recent times. Climate change and the future effect on Earth and its future has focussed much attention on research in this area. There is an on-going, increasing level of data gathering missions, both airborne and ground based which is resulting in an ever increasing amount of data to be analysed. Research in this area to date has generated numerous climate models which are extremely good at predicting typical climate changes. However, many have weaknesses when anomalous climate behaviour occurs. This thesis considers that one of the reasons behind this may be the data collection methods used. Typically these involve working with atmospheric sensors, whether mobile of stationary, in specific regions where it is predicted that data gathered will be of most use and add value to the models. The limited flight times, and pre-determined flight plans may result in fewer anomalous data being collected and limiting the input of anomalies to the models.

The aim of the research presented here is to investigate data analysis techniques to aid the collection, and analysis, of data gathered during these missions. The work comprises two key objectives:

1. Online analysis of data as it is gathered. This will allow targeting of specific data of interest, e.g. if the mission is to investigate pollution from biomass burning, then data can be analysed to ensure it relates to biomass burning, rather than general background data. While data pertaining to the background state is useful for comparisons, the ratio of target data to background data can be improved.

2. Offline analysis of the data gathered. If online analysis is used to improve the data gathered, then it follows that this analysis should be reproducible offline, post-mission. In this way decisions made can be verified and information of value can be reproduced in more detail when more time is available.

This research focusses on cluster analysis due to its ability to group data, identify anomalies and adapt to different situations in ways that other analysis techniques cannot. Requirements for clustering analysis are proposed and current techniques investigated for suitability. The specifications are not easily matched, if at all, by current techniques and so new clustering algorithms are required. These are developed in both online and offline compatible modes. The algorithms are demonstrated and shown to have the potential to add value to data gathering missions, offline analysis and ongoing online analysis of atmospheric data streams.

1.2 Thesis Structure

The thesis is presented in separate Chapters, beginning with the Aims, Objectives and Thesis Structure provided in Chapter 1. Chapter 2 which presents an overview of Atmospheric Science and the source of the challenges to be addressed. There follows a review of current clustering and cluster evaluation techniques in Chapter 3 which considers how clustering could aid in addressing these challenges and where a summary of the key types of clustering that may be applicable is presented. Also discussed are the benefits and limitations of the generic types, and specific clustering techniques, summarising why they are not suitable for the final solution envisaged here. The offline techniques developed for this thesis are presented in Chapter 4 where the development of the preferred algorithms are detailed. The aim of the offline techniques is to develop suitable offline clustering techniques to allow compatibility in visualization and operation between offline and online clustering. Chapter 5 develops the online clustering algorithms which are key to the final online software solution for use by atmospheric scientists, RASCAL, which is presented in Chapter 6. RASCAL demonstrates how the techniques can be used in the field to enhance data gathering campaigns. Chapter 6 also briefly discusses an offline version of RASCAL which allows rapid reproduction of similar results to the online version. A number of improvements to the online version, based on feedback from the Atmospheric Science community, have been included in the offline version.

Chapter 7 presents a summary of the work and its application to the challenges addressed by this thesis, overall conclusions and suggestions for developing this work in the future. This is divided into Subsections where 7.1 summarises the direct application

of the research to the primary goals of the thesis. Section 7.2.1 details the conclusions of the clustering algorithms developed. The future work discussed in Chapter 7.3 makes a number of proposals for continuing the research further. Suggestions include future development of the algorithms themselves and the software developed in this thesis as well as future research questions outside of the atmospheric science field in which this thesis is based.

Chapter 2

Introduction

This chapter provides a brief overview of the physics and chemistry of the atmosphere including the layers typically associated with atmospheric science in Section 2.1. We outline the basic physics of climate change, the resulting economic and health costs, and outline the role of atmospheric science and its funding in the scientific analysis of climate change in Section 2.2. Section 2.3 describes the characteristics of a typical atmospheric science data gathering campaign and the type and size of data sets such campaigns produce. Finally Section 2.4 outlines the challenges to be addressed during this research.

2.1 The Atmosphere of Earth

As the world has come to accept the reality of Climate Change and its current and future impact on the world the importance of the scientific study of the changes and their effects has become ever more significant. The atmosphere surrounding the Earth has significant impact on the global temperatures and the composition of the air that we breathe. The complex interaction of the chemistry in the various atmospheric layers determine the radiation reaching and leaving the Earth's surface, the temperatures of the atmospheric layers (Global Warming) and the dispersion of pollutants.

The Earth's atmosphere comprises a mixture of gasses surrounding the planet with most common being Nitrogen (N_2) at 78%, Oxygen (O_2) at 21% and various others at less than 1% including Carbon Dioxide (CO_2). Water vapour is also present at varying levels and also small particles known as 'aerosol particles' from various sources both natural and anthropogenic [152].

In general the atmosphere is composed of:

1. Exosphere, the very outer layer of the Earth's atmosphere consisting mainly of very low density hydrogen, helium and nitrogen, oxygen and carbon dioxide at lower levels. The Exosphere merges with outer space where there is no atmosphere.

2. Thermosphere (90 to between 500 and 1,000km), similar in composition to the Exosphere this region has a temperature inversion, i.e. has a gradual increase in temperature with height due to the absorption of solar radiation.

3. Mesosphere (50-85km) is difficult to study as it lies above the heights reachable by aircraft or balloon, but below that of orbital satellites.

4. Stratosphere (10-50km) A temperature inversion appears again here due to the absorption of solar radiation by Ozone (O_3) which is relatively abundant (i.e., several to a few tens of molecules of ozone per million molecules of air - parts per million (ppm)). The lack of vertical convection within the stratosphere, and between the stratosphere and Troposphere, means that air that reaches the stratosphere may remain years or even decades [113, 146]. Air moving into the stratosphere carries with it chemicals such as CFCs (chlorofluorocarbons) which have a significant impact on the ozone levels.

5. Troposphere (0-10km) contains most of the mass of the Earth's atmosphere. Characteristically the temperature drops as the height increases but layers with constant or increasing temperature with height are not uncommon. Most cloud formations appear in the troposphere and nearly all weather phenomenon occur here.

6. The Planetary Boundary Layer is the lowest part of the troposphere and is in contact with the Earth's surface. This contact allows the surface to affect the atmosphere - through exchange of heat, momentum, gases, and particles - and so some atmospheric behaviour is directly influenced by that contact.

In addition to the primary layers there are boundary regions between them. These boundaries play an important role, affecting the movement of air between the layers. They are a focus of study for atmospheric scientists, particularly the chemistry transport across these boundaries and the effects on the primary layers. These boundaries are:

1. Thermopause, the boundary between the thermosphere and the exosphere

2. Mesopause, the boundary between mesosphere and thermosphere.

3. Stratopause, the boundary between the stratosphere and the mesosphere.

4. Tropopause, the boundary between the troposphere and the stratosphere.

These boundary layers are not fixed and the altitudes vary under different atmospheric conditions. The tropopause is of particular interest to atmospheric scientist due to the differing effects of chemistry in the troposphere and stratosphere, e.g. CO_2 has a heating effect in the troposphere, yet a cooling effect in the stratosphere. Similarly ozone, O_3 in the troposphere is considered a pollutant, whereas in the stratosphere it protects Earth from the harmful effects of ultra-violet rays. The transport of these and other chemicals across the tropopause is a significant area of study.

The atmospheric layers being considered in this thesis are primarily the Boundary Layer, Troposphere and Stratosphere together with the tropopause. The Coordinated Airborne Studies in the Tropics (CAST) project, of which this research is a part, was specifically targeted to investigate the interactions between these layers in a coordinated effort to study the atmospheric chemistry and transport between these layers [75].

2.2 Climate Change

Climate change is now considered an accepted theory with far reaching effects. These effects will vary globally with a range of economic, social and health impacts. The Projection of Economic impacts of climate change in Sectors of the European Union based on bottom-up Analysis (PESETA) project report [34] considers many aspects of possible climate change scenarios across the EEA including he effects on agriculture, coastal systems, river floods, tourism and health. The findings indicate and annual cost to the EEA economy to be €20 to €65 billion between scenarios of 2.5^oC and 5.4^oC temperature rises.

2.2.1 Economic Costs

The World Development Report 2010 [149] reports that a global temperature change of 5^oC experienced since the last ice age is expected to occur within a century under current conditions. The long term rise allowed time for adaptation to change, whereas the rapid change expected in the near future does not and is likely to cause considerable cost, financially, socially and environmentally. The report indicates that the most vulnerable, developing countries will bear 75-80% of the costs of climate change and their report 'Economical Adaption to Climate Change' [160] places the cost to these countries at around $70- 100 billion per year. In the USA The Council of Economic Advisors [40] in their report 'The Cost of Delaying Action to Stem Climate Change' places the cost of to the economy of the USA at 0.9% of GDP, or $150 billion dollars for every 1^oC increase in temperature.

2.2.2 Health Impacts

'Climate Change: The Cost of Inaction and The Cost of Adaption' [156] considers the health costs to the EEA. Although much of the health impacts for the future are unknown it suggests that 35,000 excess deaths occurred during the 2003 heatwave compared with previous similar events. However, overall, and discounting heat waves, the net impact is predicted to be an increase in heat related mortality balanced by a decrease in cold mortalities. The PESETA report also indicates a potential cost of several Billion Euro associated with climate sensitive diseases, allergens and air quality.

2.2.3 The Role of Atmospheric Science

Atmospheric Science has a key role to play in the study of climate change. The Earth's atmosphere is a significant part of the global climate and the uptake, transport and deposition of chemicals and pollutants all have an impact, generating an ever growing body of research e.g. [24, 31, 151, 73, 60, 38, 76]. The models used to predict the various scenarios rely on accurate input data and the accuracy of their predictions can be tested with subsequent monitoring. Such is the importance of these climate models that comparing the models are constantly compared, evaluated, modified and updated [94, 91, 21, 11, 79, 12].

With the wide range of potential costs and impact of the modelled scenarios the importance of atmospheric science in the role of climate science is apparent. The uncertainties involved in modelling such complex dynamical systems and the sensitivity to small changes, 'The Butterfly Effect', underline the importance of accurate data. Data from anomalous events can be of particular interest to adjust models to predict such unusual, or extreme, events, or to be ignored as genuine 'anomalies' which may be considered unpredictable.

2.2.4 Climate Science Funding

For the reasons discussed in the previous Subsection, climate research and atmospheric science research has achieved ever more attention and importance. Since 1992 the European Union (EU) LIFE programme has contributed more than €3.1 billion and supported over 4,000 projects. Funding for LIFE is set to rise to €864 million for the period 2014-2020 [55]. In the USA, the US Global Change Research Project (USGCRP) budget has increased from $1,816m in 2008 to $2,658m in 2014 [95] and the National Science Foundation (NSF) increased the Division of Atmospheric and GeoSpace Sciences (AGS) funding to to $122m for research in 2014. The UK Natural Environmental Research Council (NERC) currently has 8142 grants, fellowships and training grants totalling over

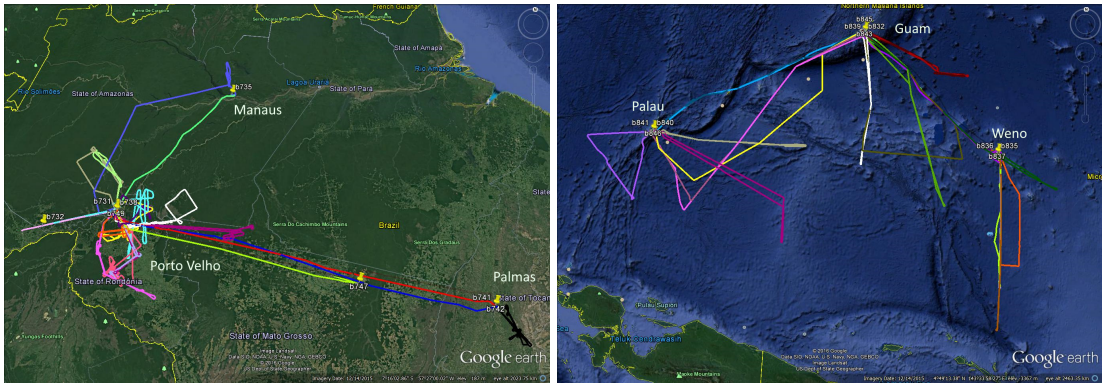
£1.6 billion [122]. Of these NERC funded programmes 1282 are specifically classified as being atmospheric science with a value of nearly £300 million [121].

2.3 Atmospheric Science Campaigns and Data

The fundamental building blocks of atmospheric science are observations and process-based models. Observations define past chemical and physical climatologies (e.g. the IPPC report on climate change [147]). Process-based models fill-in data-gaps (e.g. the ERA re-analysis project [54]), diagnose probable causes of observed events (e.g. extreme events like the Boscastle floods [69], or tropical hurricanes [112]), and provide forecasts of future weather, climate, and atmospheric composition. Most atmospheric observations are made operationally at a network of ground stations, or using satellites. However, to diagnose atmospheric processes in more detail than is possible using operational observations, intensive field campaigns are used. Atmospheric science campaigns typically run over days, weeks or longer and involve a number of possible sources of data. The include instruments on board aircraft, ground based instruments, rising sondes and drop-sondes. Additionally, there are continuous data streams from the standard aircraft instrumentation providing non-chemistry data such as aircraft position and status. Here we will describe two example atmospheric science campaigns.

The South AMerican Biomass Burning Analysis (SAMBBA) [110] had specific objectives to study the pollution and effects of biomass burning in the South American rainforests. The main SAMBBA findings were reported in a special issue of Atmospheric Chemistry and Physics [7, 22, 92, 106, 126]. As with all such data sets research and publication is on-going. The SAMBBA campaign (Sept. - Oct. 2012) made use of the Facility for Airborne Atmospheric Measurements (FAAM) BAe-146 aircraft configured with 52 instruments on board [23]. The flight paths are shown in Figure 2.1a with each shown in a different colour. We will use the data from this campaign later in the thesis demonstrating specific example of how the results of this research can add value to these missions.

The Coordinated Airborne Studies in the Tropics (CAST) [120] project involved 3 aircraft together with additional sondes to gather data. The aircraft come from the National Aeronautics and Space Administration (NASA) Airborne Tropical TRopopause EXperiment (ATTREX) [116] project, the National Centre for Atmospheric Research (NCAR) as part of their CONvective TRansport of Active Species in the Tropics (CONTRAST) [119] project and FAAM Co-Ordinated Airborne Studies in the Tropics (CAST) [120]. In a unique set of missions designed to sample atmospheric air chemistry throughout the



(a) SAMBBA Flight paths. (b) CAST Flight paths.

Fig. 2.1 Maps of the flights paths for 2.1a SAMBBA campaign and 2.1b the CAST campaign. The SAMBBA campaign focussed on higher altitudes up to approximately 8,000m, with some time spent at 1,000m-2,000m. CAST focussed on lower altitudes around 500m-1,500m with some time spent up to approximately 6,000m

.

full atmospheric range from ground to stratosphere three aircraft with complementing configurations of instruments on board flew at a range of heights.

- Low altitude sampling was carried out by the FAAM BAe-146 at heights between 0 - 6,000m
- Mid altitude flights were carried out by the NCAR GulfstreamV (GV) flying between 6,000m - 14,000m
- High altitude flights carried out by NASA's Northrop Grumman Global Hawk UAV at altitudes up to 14,000m - 20,000m

These are the target flight altitudes, in practice there was some overlap between the aircraft flight profiles. The campaign took place in a 6 week period in January and February 2014 from bases around Guam in the Western Pacific.

The instruments carried by the FAAM BAe-146 aircraft [120] are primarily self-contained and capture and store the data for download post-flight. Some instruments currently gather air samples for later analysis so the data is not available in flight. Final data is not provided until much later after instrument calibration and necessary adjustments have been made, although 'quicklook' data is often available in real time. In total there were 28 instruments, 35 including the standard core instrumentation and basic flight instrumentation.

The NCAR GV aircraft payload [118] consists of 16 scientific instruments. Some instruments measure multiple chemical species, others measure particle size in a range

Table 2.1 Examples of typical data rate and dataset sizes for a single, full duration flight for each aircraft in the CAST campaign.

Platform	Instruments	Data Streams	Samples/flight ($\times 10^6$)	Dataset Size
Global Hawk	12	391	42 (30 hrs)[84]	950 MB
GulfStream V	16	352	12.7 (10 hrs)	47 MB
BAe146	28	40	0.72 (5 hrs)	1.6 MB
Totals	56	774	55.32	1 GB

of statistical bins. The total number of potential data streams is 352 in the instrument configuration deployed during the CAST project.

The NASA GH had a payload of 12 instruments [115] plus standard flight information. Some of the instruments provide chemistry information on multiple chemical species. Other instruments measure particle size or concentrations in different ranges. Taken as separate data sources and not including error flags there were a total of 391 data streams. As an unmanned vehicle the data from the GH is sent to the ground as data streams already although it is not processed online.

Typical examples of the volume of data gathered during campaign flights are given in table 2.1. These figures are based on a 1Hz sampling rate for a typical full flight duration. In practice flight durations may change and sample rates vary per instrument.

2.4 Atmospheric Science Data Challenges

With most data analysis occurring post-campaign, data pertaining to anomalies or other data of specific interest may be limited. Processing the data online and in-flight could improve the data collected by helping focus the data gathering on specific regions of interest. As discussed in Section 2.3 it can be seen from Table 2.1 that the volume of data captured and streamed to mission scientist is vast. Faced with this torrent of data it may often be the case that mission scientists focus on a small subset of the available data. New techniques and analysis are required to aid mission scientists by providing online analysis of the data to simplify the decision making process during campaigns. Additionally, these analysis techniques should be reproducible offline for post-campaign analysis and re-analysis of historical data.

Here we outline some of the key challenges facing mission scientists and data gathering campaigns together with the limitations they impose.

2.4.1 Live Monitoring of Multiple Data Streams

Monitoring of single data streams is most frequently achieved with line plots, often overlaying multiple lines on the same graph window. Scatter plots have similar issues to line plots with regard to scaling, but add the difficulties of choosing a suitable marker size. This has many drawbacks such as:

1. Differing y-axis scales for multiple plots. Providing a single plot window for multiple data streams may result in a range of y-axis maximum values. The resulting plot can be confusing with either multiple y-axis scales, or, in extreme cases, featureless horizontal plot lines at the top and bottom of the graph.
2. Sizes of plot markers. Plot marker size can have a noticeable affect on data visualization. Large plot marker sizes may hide new data points or give the appearance of merged regions of data. Small markers may be hard to see and to distinguish between colours in colour plots.
3. Visualization of historical data. Line plots may be of limited use for extended time scales. As the data gets compressed into the plot window fine details of anomalies or 'spikes' may be hard to see, and may even not be present. Scatter plots suffer from the marker size problem mentioned above.
4. Distinguishing colours or markers. There are limits to the numbers of colours it is possible to distinguish by the human eye, particularly with many lines, or markers overlapping.
5. Over plotting of similar data. Both line plots and scatter plots suffer from overwriting of old data such that the latest line to be drawn, or data to be plotted, overwrites and obscure underlying information.

2.4.2 Discrete Data Sampling

Although data may be continuous, discrete sampling rates result in point data rather than data space regions of similar values. This produces line or scatter plots as described in Section 2.4.1 and datasets of discrete samples. The reality is continuously variable data such that a small change in sample time and, therefore spatial location, would produce a similar, but slightly different, reading.

2.4.3 Online Chemistry Analysis

As shown above in Table 2.1 the volume of data produced during a data mission can be large. With this volume of data analysis becomes time consuming beyond what could

reasonably described as 'online'. With some grouping, or clustering, of similar data the volume of data to be analysed may be significantly reduced while still providing some reasonable approximation.

2.4.4 Online Anomaly Identification

Anomalies can fall into two main categories, although these overlap somewhat, 'local' and 'global' anomalies. Local anomalies refer to data that is anomalous over a specific range, typically a temporal range, although the data may not be unusual in general, e.g. a temperature of 18^oC is rather anomalous for December in the UK, but is quite normal for summer. To continue that example, a temperature of 35^oC would be considered a 'global' anomaly as it is an unusually high temperature for the UK at any time of year. It is therefore necessary to differentiate data drift, gradual expected changes, from data anomalies. Such temporal separation may be important for the understanding of pollution which may have both diurnal and seasonal variation [134].

2.4.5 Online Identification of Drift

As mentioned in the previous Section, 2.4.4, differentiating between drift and anomalies may be important. This is not only true for the identification of anomalies, but also for identifying general drift and in readings which may be important in themselves. To continue the example from the previous Subsection it may be significant to notice general drift in the UK temperature for a given month over several years and to notice that the drift is independent of any anomalies. However this can equally apply to spatial drift during flight operations, e.g. moving over different terrain types, or from land to ocean areas.

2.4.6 Historical and Temporal Separation of Chemistry

Temporal separation of data with similar values is important particularly during airborne operations. Chemical sources such as short lived halogens from ocean regions [27] rise and drift through the atmosphere. Tracking these plumes helps understanding of the decay, transport and dispersion of the chemical species. By identifying similar data at different points in the flight the plumes can be tracked [143]. Knowing where these data appear online and in flight could allow flight paths to be altered to improve coverage of the relevant chemistry.

2.4.7 Historical Analysis and Presentation of Data to New Operatives

The lengths of flights on data missions are ever increasing such as those of the Global Hawk at 30 hours, or potential future solar powered aircraft like the Solar Impulse which offer unlimited flight hours. [1, 144]. These missions require multiple mission scientists on shifts to cover the full duration of the flight. When new operatives start their shift presenting them with historical analysis of the flight up to that time is useful to aid understanding of the data captured. With historical analysis available it is also possible to compare previous flight data from the campaign, or even data from previous campaigns. The same reasoning can be applied to data streams gathered over many years and across many data missions.

2.4.8 In Flight Analysis for Flight Path Adaptation

During many data science campaigns the data found is broadly consistent with that expected and predicted by weather, climate, or operational pollution forecast models. In some cases, however, it may be found that significant anomalies are discovered during post campaign analysis, whether they be anomalous to typical data gathered so far, variations from predicted model output values or the data specific to the main investigation of the campaign, such as evidence of biomass burning [143]. Identifying such data in-flight would enable potential changes to the flight path to capture more of the anomalous data. This additional data may be of significance in either improving climate models, providing additional data for the primary investigation or identifying unexpected chemistry.

2.4.9 Rapid Future Flight Planning

Similar to the way that online analysis can aid in flight route changes, the same can be said for future flight plans. The on-line information available during a flight, together with the ability to reproduce the analysis post-flight could provide a useful insight into the geographical locations of the data of most interest. This information can feed in to the future flight plans.

2.4.10 Reproducible Analysis Post Sortie

If valuable insight can be gained by rapid, online analysis during data missions then it reasonable to expect that such rapid data exploration may also be of value offline, post-campaign. Not only will this allow for ease of reproducing results, but it could also

facilitate purely exploratory analysis. Pure exploratory analysis has a long history in all scientific fields but is also well known for providing serendipitous discoveries of great importance [58, 137].

2.4.11 Summary of Atmospheric Science Challenges

The challenges listed above can be broadly grouped into 4 key areas:

1. Data Collection Improvement by online analysis and identification of data of interest allowing targeting specific spatial locations providing the most appropriate data.

2. Online Data Analysis challenges to provide the insight necessary to help improve the data collection.

3. Post Flight Analysis to reproduce the online analysis when applied to the full flight dataset.

4. Post Campaign Analysis should reproduce the same results and allow for fast, detailed analysis of the full campaign datasets.

Chapter 3

Literature Review of Relevant Clustering Methods and Cluster Quality Measures

This chapter provides a summary of how clustering techniques could aid in responding to the Environment Science data challenges proposed in Chapter 2.4. Each challenge is introduced in Section 3.2 with an overview of how clustering may help and what type of clustering may be best suited to address each challenge. Current clustering techniques are discussed in Chapter 3.3 with Sections devoted to: the advantages of hyper-elliptical versus arbitrarily-shaped clusters, 3.3.1; offline techniques, 3.3.2; online techniques, 3.3.3; and the compatibility between the two, 3.3.5.

The objective of this research was to provide clustering algorithms to address the specific needs of atmospheric science data gathering missions and post mission analysis. Section 3.2 summarises the requirements expected of these algorithms and how they may satisfy the overall objectives. Various review papers [162, 124, 8, 141] summarise the current state of the art in offline and online clustering and the book 'Data Clustering, Algorithms and Applications' [2] draws together many sources of techniques and applications. In general there are a limited number of clustering methods, each of which have numerous methods of implementation. It is also the case that many algorithms have been improved upon since first publication, or have multiple variants for specific situations, yet retain the same underlying principles. It was found that it is these underlying principles that prevent a total solution to the requirements for our proposal. For example, there may be algorithms which satisfy the online requirements, and others that satisfy the offline requirements. However, the two techniques may not be fully compatible, or may even produce significantly different results in some circumstances. Thus, this section does not propose to review and analyse each of the many different clustering techniques

individually, but rather to consider the underlying principles and consider their suitability in this basis.

3.1 Clustering in General

Clustering is an unsupervised machine learning method for grouping data. Early clustering algorithms grouped data by a simple similarity measure, i.e. data within a cluster should be more similar to data within its cluster than to data in other clusters. Many early techniques required a priori knowledge of the number of clusters to be found. Later developments found other methods of generating the clusters with different user inputs, e.g. the typical expected size of a cluster in the data space. These early techniques, and this cluster definition has significant limitations as to they type of data groups that could be found, very early techniques discovering hyper-spherical clusters, later methods extending this to hyper-elliptical clusters. Yet, groups recognisable by humans vary considerably from these simple shapes and later developments allowed the discovery of data groups of arbitrary shapes. Another development path for clustering algorithms moved the techniques from the offline realm, where all data is available, to online techniques where data streams constantly update the available data leading to 'online' and 'evolving' techniques. These different techniques are discussed in the following Subsections.

It should be noted that no technique claims to be the panacea of clustering algorithms and different techniques should be applied under different circumstances. A recent summary of many clustering techniques, together with a range of applications can be found in [2]. Applications include data streams, document, biological and multimedia clustering. Within atmospheric science clustering is little used and in the majority of cases k-means clustering is the most common [33, 87, 28, 70, 56, 103], although other methods are not unknown such as fuzzy C-means[154] and agglomerative hierarchical clustering [41, 139]. In many cases in atmospheric science, the aim of the clustering algorithm is to group the data into a pre-determined number of clusters. As these clusters may be relative classification techniques are unsuitable and, with k-means being the widest known clustering algorithm as well as one of the fastest, and eminently suitable where the number of clusters is known, this could explain its widespread use. However, such a technique may not be suitable where data exploration and knowledge discovery are the primary aims of the work and alternatives are explored in the following sections.

3.2 Clustering as a Response to the Atmospheric Science Challenges

This section summarises the proposed responses to the atmospheric science challenges outlined in the introduction. The challenges are summarised together with a brief outline of how various clustering technique principles could be employed to meet the challenges. These form the requirements of the techniques and they are summarised in Table 3.1.

1. **Improving the Visualization of Online Data Streams.** Clustering techniques can be used to display the data space regions in which the data has appeared, not just the exact data points measured. This can be achieved by clustering the data and/ or using alpha hulls or similar to outline the data region. This requires extra processing beyond the original clustering technique and this additional process needs to be repeated with every data sample.

   When a single data stream is chosen for visualization from the start then the technique can be independent of offline techniques. However, if it is expected that a change of data streams may be warranted for any reason then the clustering technique should be compatible with an offline clustering technique. In this way the historical data can be clustered offline before the online method takes over and updates continuously from that time.

2. **Online Anomaly Detection** requires an online clustering technique that allows for drift in the data, yet still displays local anomaly changes. Anomalous data could expand hyper-elliptical cluster shapes, yet still leave much of the data space within that cluster empty of data samples. It is possible that new anomalies may go unnoticed should they appear within that empty cluster space. To avoid this the cluster shape should accurately cover the recorded data only, i.e. must generate arbitrarily shaped clusters. Again, as outlined in 1, the techniques should be compatible with an offline technique.

3. **Detection of Data Stream Drift.** The drift in collected data over temporal or spatial variances should be distinguishable from anomalous data. It should also ensure that data which is locally anomalous, yet may still be within the bounds of previously collected data, is identified as such. This indicates that a fully evolving clustering technique should be capable of providing diminishing importance to ageing data such that, at any given time, the data displayed is relevant to the recent situation and not influenced by past data, e.g. if data is being collected over tropical forestation before moving over towns, or sea, the effect of the forestation data should have diminishing effects on the clustering results.

4. **Temporal and Spatial Separation of Historical Data.** To achieve this a clustering technique should be capable of including spatial or temporal data such that data can be separated by either. Data that is similar in value, but separated by time or space should be clearly distinguishable. This is particularly important when considering such events as pollution sources and their corresponding plumes.

5. **Visualization of Historical Data.** Offline clustering of historical data should be fast, as the historical datasets may be large. When applying the offline clustering techniques it is important to ensure that the clusters, and data space regions in which they lie, are consistent with any online technique. This ensures that the visualizations created are consistent.

6. **In Flight Analysis for Flight Path Modification.** In flight cluster analysis requires an online clustering technique. As outlined before it is important that the technique is capable of producing arbitrary shapes and responding promptly to anomaly detection.

7. **Rapid Future Flight Planning.** Future flight planning can benefit from the results of any online, in-flight clustering, or from reproducing the analysis offline with more detailed investigations. The clustering results may add information regarding the locations of data that is of most interest.

8. **Reproducible Analysis Offline, Post Sortie and Post Campaign.** Reproducing any online in-flight analysis offline will allow for more detailed examination of data. In flight the main user of the analysis is likely to be the mission scientists, however offline any experts can be easily involved in examining the data, analysing and identifying data of interest.

3.3 Current Clustering Techniques

This section will discuss the techniques, and resulting clustering, of current clustering techniques. Based on the responses to atmospheric science challenges many alternative clustering techniques will be considered and their suitability assessed. Visualizations are included as an aid to understanding the text. The clustering results they show are not intended to indicate the best, or worst, possible result of any technique, but merely to give clarity to the text.

The first Subsection, 3.3.1 will consider the differences and applicability of distance based hyper-elliptical clusters relative to density-linked arbitrarily shaped clusters. Subsection 3.3.2 considers offline techniques of these two types. Following this is Subsection

Table 3.1 Summary of clustering techniques required to meet the defined atmospheric science challenges

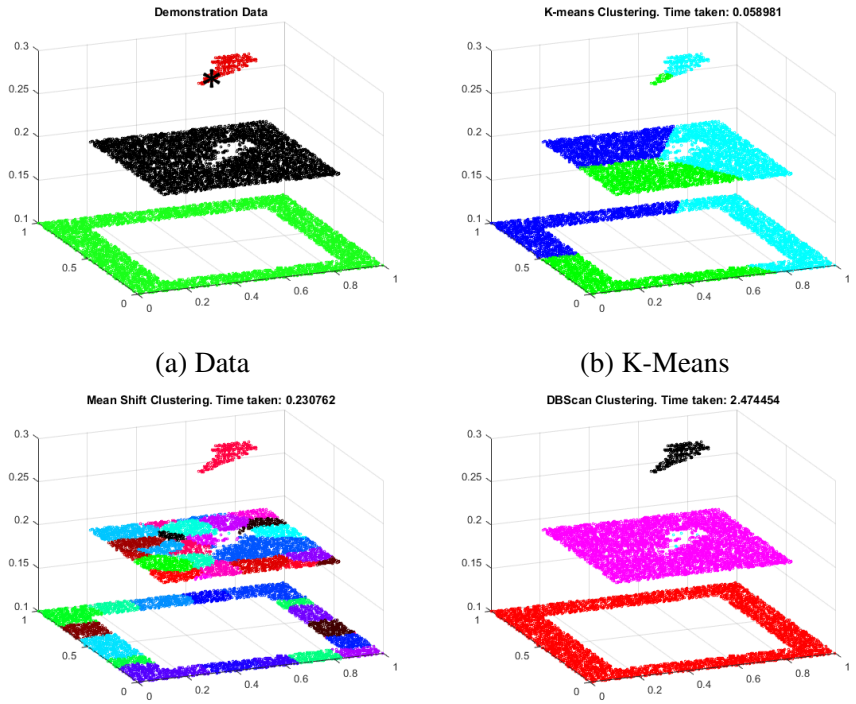
Challenge	Online	Offline	On \Offline	Reproducible Offline	Arbitrary Shapes
1	Y	Y		Y	Y
2	Y		Y	Y	Y
3	Y		Y	Y	Y
4		Y		Y	Y
5		Y		Y	Y
6	Y		Y	Y	Y
7		Y		Y	Y
8		Y	Y	Y	Y

3.3.3, an examination of hyper-elliptical and arbitrarily shaped clusters in online, dynamic and evolving clustering techniques where different challenges arise from those of offline techniques. Subsection 3.3.5 reviews the typical results obtained from the offline techniques and discusses their compatibility with online techniques. Finally we summarise the techniques and consider a path for applying clustering to the atmospheric science challenges in Subsection 3.3.6.

3.3.1 Elliptical Versus Arbitrarily Shaped Clusters

Clustering can be grouped into two main types, those that use a purely distance based measure and produce hyper-elliptical clusters (in terms of the distance measure used), or those that use density-linked type groupings that produce arbitrarily shaped clusters. It is acknowledged that where hyper-elliptical cluster encroach on each other's data space they may form centroidal voronoi tessellations but throughout this thesis they shall be referred to as hyper-elliptical clusters and techniques in reference to their cluster membership functions.

If arbitrarily shaped clusters can, by definition, produce clusters of any shape, then it follows that hyper-elliptical clusters must be a subset of these arbitrarily shaped clusters. As a broad mathematical definition of these clusters we have equation 3.1 defining hyper-elliptical clusters such that each data sample in a cluster x_{i0}, should be closer to its cluster centre x_i than to any other cluster centre x_j. Whereas in the case of the arbitrary shaped cluster we define that each data sample in a cluster, x_{i0}, must be closer to any other data sample within its cluster, x_{ik}, than to any data sample in another cluster, x_{jk} as



(a) Data (b) K-Means

(c) Mean Shift (d) DBScan

Fig. 3.1 Plot (a) Shows a simulated pollution release, green is surrounding countryside, black urban area and red is pollution release from the location identified by the asterisk. Pollution levels are 0.1, 0.2 and 0.3 respectively. Typical results from clustering techniques are shown in the remaining images. K-Means (b) is unable to create meaningful results when asked to find 3 clusters. We also see that distance based techniques such as Mean Shift (c), resulting in hyper-elliptical clusters, may identify the pollution but the larger regions are broken up and disjointed making it hard to visualise. This is a result of these types of techniques filling non-hyper-elliptical shapes with partial hyper-ellipses. In simple cases such as this it is possible to tune the radii along each axis such that each hyper-ellipse could be 'flat' to improve the results, however this requires a priori knowledge of the resultant cluster shapes. Arbitrarily shaped cluster resulting from techniques such as DBScan (d) produce arbitrarily shaped clusters which identify the regions with much greater accuracy.

given in equation 3.2.

$$||x_i - x_{i0}|| < ||x_j - x_{i0}|| \tag{3.1}$$

$$||x_{ik} - x_0|| < ||x_{jk} - x_0|| \tag{3.2}$$

Thus, we see that the hyper-elliptical cluster is that subset which defines the data point to which the sample must be closest.

In practice clusters such as those shown in Figure 3.1 result. Artificial data comprising of 10,000 data samples is used to represent a typical pollution based scenario as shown in Figure 3.1a. 'Countryside' is represented by the green data, with a pollution level of 0.1. The black region represents an 'urban' region with pollution levels of 0.2, while the red data represents a specific 'pollution' release with pollution levels of 0.3. The data represents a time after the start of the pollution release where the pollution has drifted and spread somewhat. We can see from these plots that, despite its popularity in atmospheric science, k-means is unable to create meaningful clusters from the data and k-medoids produces similar results, Figure 3.1b. Mean shift is capable of separating out the small pollution region, but the urban and countryside regions are broken up into multiple clusters, Figure 3.1c. This is an improvement over analysing the raw data, as we have fewer groups of data, which we have identified as similar. However, when using arbitrarily shaped cluster techniques such as DBScan, we can see that the each of the different regions can be clearly identified, Figure 3.1d.

It is worth clarifying that in many cases density based clustering techniques fall in to the group of distance based techniques and, so, tend to produce clusters of hyper-elliptical shapes. This is not necessarily the case however, it is just that many density calculations are themselves distance based. There are, of course, situations where hyper-elliptical techniques may be more appropriate to use. If the data is known to be, or can be approximated to, hyper-ellipses then the speed of the hyper-elliptical techniques may be of considerable advantage. Even on the simple example, shown above, of 10,000 samples of 3 dimensional data we see an approximately 10 fold reduction in clustering time between DBScan at 2.47s and Mean Shift at 0.23s.

Both distance-based and density-linked arbitrarily shaped clustering techniques are considered throughout this thesis to compare the differences and trade-off between accurate data representation and speed.

3.3.2 Offline Clustering Techniques

When considering offline clustering techniques we must first understand both the reasons and the nature of offline clustering. The primary focus of offline techniques is to form clusters from a known and finite number of data samples according to some rule of

Table 3.2 Examples of typical cluster definitions by algorithm basis.

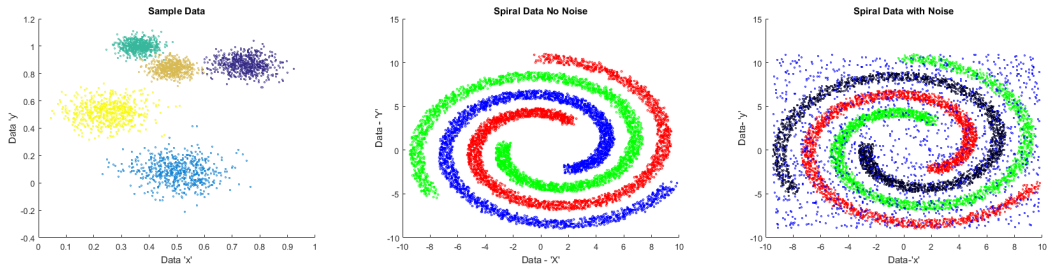
Algorithm Basis	Cluster Definition
Connectivity	Data within a cluster should be more closely related to nearby data within the cluster than it is to data in other clusters
Centroid	Data within a cluster should be closer to the centroid of that cluster than to the centroid of another cluster.
Distribution	Data belonging to a cluster should most closely form a pre-defined statistical distribution
Density	Data belonging to a cluster should be connected to all other data in the cluster by regions of a defined density range.
Sub-Space	Data may belong to multiple clusters in multiple sub-spaces. The number of subspaces is 2^d for orthogonal sub-spaces and infinite for non-axis-parallel. Any clustering algorithm is possible within the subspaces.

similarity. Clusters do not have a single definition and different clustering techniques may use different definitions to suit their algorithms. Examples of cluster definitions are given in Table 3.2.

The aim of offline clustering is typically to produce a list of cluster assignations, or cluster membership likelihoods, for each data sample. In this way the definition of each cluster may be a large and unwieldy vector list. Cluster results consisting of a full data sample list are particularly problematic when interfacing with online techniques.

One of the key drivers of online techniques is memory and computational efficiency, especially for on-going, endless data streams. Storage of the full data stream will require enormous memory capacity. Additionally, comparing future incoming data to that already stored will require ever increasing processing time. The next section will discuss on-line clustering techniques in detail, however it is clear that the results produced by traditional offline techniques are not directly compatible. Offline clustering results could, possibly, be converted to a compatible format by grouping and analysing the results with an interface algorithm. However, this is generally an offline, compatible version of an online algorithm and will also require additional processing time.

Throughout this subsection examples of typical clustering results will be shown for synthetic data sets. These datasets are shown in Figure 3.2.



(a) Gaussian Data (b) Spiral Data without Noise (c) Spiral Data with Noise

Fig. 3.2 Example datasets used to illustrate clustering techniques. (3.2a) consists of 5 clusters of Gaussian distributed data. (3.2b) consists of 3 spirals of data, with no noise. (3.2c) consists of the same spirals but with random noise across the data space.

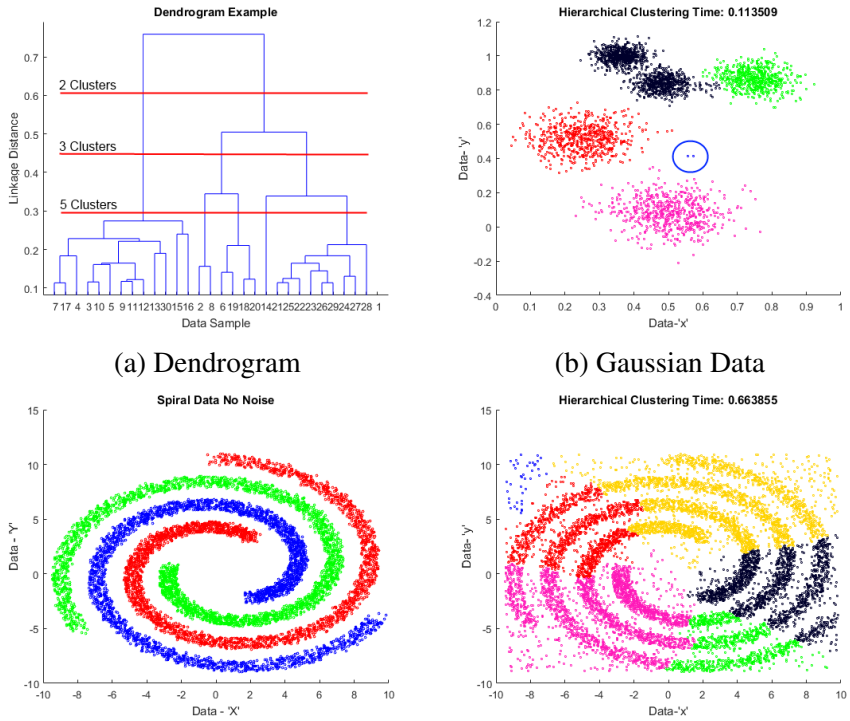
Connectivity Clustering

Connectivity based cluster analysis refers to techniques such as Hierarchical clustering [109] and developments of this technique. It has two forms, 'agglomerative' and 'divisive'. In agglomerative analysis all the data samples are considered separate clusters initially and are merged, based on an appropriate distance measure, until only one cluster remains. The results are often presented in a dendrogram to allow the user to decide the appropriate cluster split, however, this can be automated to suit a user input for the number of clusters, or a pre-determined linkage height. Divisive analysis functions in reverse with the initial state being a single cluster, which is then divided until all data samples are separate.

Typical results for Hierarchical clustering are shown in Figure 3.3 which show the limitations of the technique. Under certain circumstance Hierarchical clustering is capable of producing some arbitrarily shaped clusters, e.g. where cluster are well separated with little or no noise as shown in Figures 3.3b and 3.3c. However, noisy data can create linkages between natural clusters as seen in Figure 3.3d. Hierarchical clustering has complexity of $O(n^3)$ for agglomerative and $O(2^n)$ for divisive clustering with exhaustive search. This high complexity, combined with high memory requirements, makes it unsuitable for large datasets even with optimised algorithms such as SLINK [140] or CLINK [44].

Centroid Clustering

Centroid clustering refers to cluster analysis that groups similar data around a cluster centre. This cluster centre is a vector in the data space and is not necessarily coincident with an available data sample. The centre is typically calculated from the data during the analysis although some techniques can be seeded with initial centres to speed up calculation. Cluster membership is often 'greedy' with each data sample being assigned



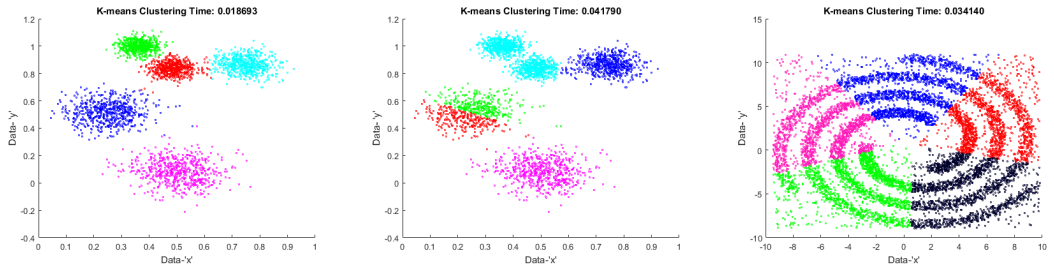
(a) Dendrogram (b) Gaussian Data

(c) Spiral Data Set without Noise (d) Spiral Data

Fig. 3.3 Typical agglomerative analysis dendrogram (3.3a) based on the Gaussian sample data indicating linkage heights for 2, 3 and 5 clusters. (3.3b) shows the results on the Gaussian data. Outliers on the magenta cluster (circled) have distorted the results causing 2 clusters to merge (black) and preventing the successful identification of the 5 clusters. Figure 3.3c shows successful clustering of arbitrary shapes where no noise is present. Figure (3.3d) shows the results on the Spiral dataset where noise has produced linkages 'across the gap' resulting in voronoi tessellations where the clusters meet

to a single cluster. However, fuzzy techniques are also available where each data sample is a assigned a membership likelihood for each available cluster.

The k-means algorithm [105], and the related k-medoids, is perhaps the most widely known, and widely used in Atmospheric Science. K-means requires the user to define the number of clusters to place the data into. Without a priori knowledge of the data it is not possible to define this number. A number of techniques for estimating the number of clusters, 'k', have been proposed. However they typically rely on repeated iterations of the algorithm with differing 'k' combined with a measure of the 'best' results, e.g. the Elbow method, Information Criterion [18], Gap Statistic [150], v-fold Cross Validation [145] or Silhouette Analysis [62] and many others as described by Arbelaitz [13]. The iterative nature of these evaluation techniques may be time consuming, particularly with large data sets and/ or many clusters. There are many variations on the basic k-means algorithm which try to address these issues, but the need to specify 'k' remains a common



(a) Gaussian Data Clustered Well

(b) Gaussian Data with Erroneous Clustering

(c) Spiral Data is unsuitable for K-Means

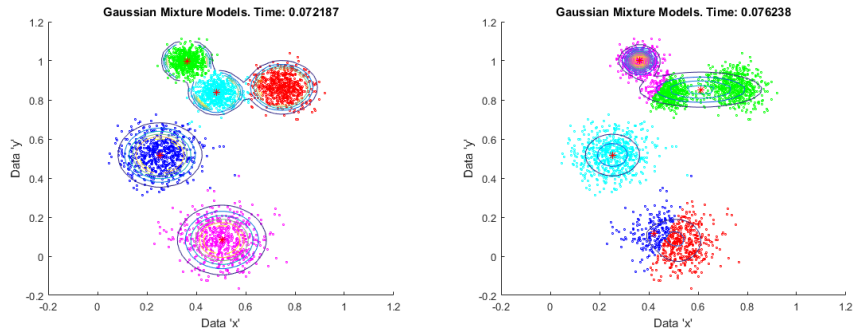
Fig. 3.4 (3.4a) shows the results of K-Means successfully clustering the Gaussian data. (3.4b) shows how the random seeding of the K-Means algorithm can produce erroneous results. (3.4c) shows the results on the Spiral dataset demonstrating the limitations of hyper-elliptical, distance based clustering. The clusters actually form straight edges as they butt up against each other.

drawback. Fuzzy-C-means [19] is similar in principle to k-means, however, each data sample is given a 'membership' value for each cluster, rather than being assigned to a single cluster. Figure 3.4 illustrates some of the short comings of k-means, and other typical centroid clustering algorithms.

Subtractive clustering, a technique based on mountain clustering [163], does not require the number of clusters to be predetermined. Rather it uses a radius, within which to include data samples in the cluster. The centroid of the cluster is calculated by determining which data sample has the minimum distance to all other data samples. This data sample and all those within the user-defined radius are considered to be a cluster and are removed from the data set. The process is repeated until all the data is clustered. With appropriate radii the technique can discover the correct number of clusters, however it is susceptible to incorrect radii, i.e. too small a radii and natural clusters will be broken into smaller clusters, too large a radii and natural clusters may merge.

Distribution Clustering

Distribution based clustering techniques are based on statistical distributions and assume that the data falls in to such distributions. Gaussian Mixture Models (GMM) [107] are an implementation of an Expectation Maximization [45] whereby it seeks to maximise the likelihood of the membership of a data sample to a Gaussian distribution model. It is an iterative process that adds data, sample by sample to a pre-defined number of Gaussian distributions. The parameters of the distribution are modified to encompass all of the assigned data. GMMs suffer from a few sources of potential errors such as random seeding, data order dependence and incorrect number of distributions defined. As with



(a) Gaussian Data Clustered Well (b) Gaussian Data with Errors

Fig. 3.5 (3.5a) shows the results of Gaussian Mixed Models (GMM) successfully clustering the Gaussian data. (3.5b) shows how the GMMs algorithm can produce erroneous results using the same parameters and data set. Gaussian Mixture Models are unable to cluster arbitrary, non-gaussian shaped clusters such as the spiral data set.

k-means the a priori knowledge required to define the number of Gaussian distributions is a limiting factor. Examples of the limitations of GMMs are shown in Figure 3.5.

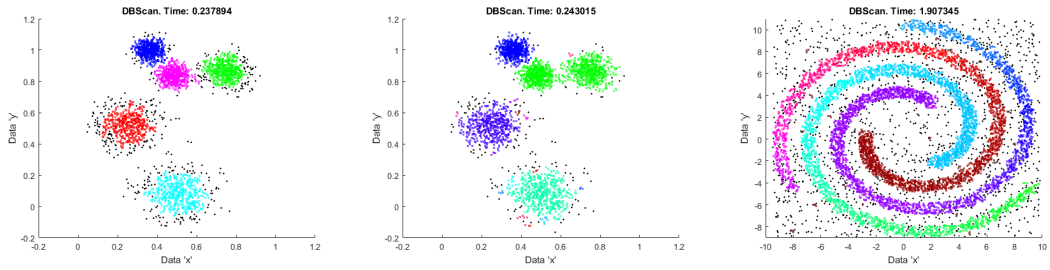
Density Clustering

Density based clustering refers to techniques that cluster data based on similarity of data density. The most well known of these techniques is DBScan [53]. DBScan has limitations in terms of complexity $(O(n^2))$ and its inability to distinguish clusters of varying density. Many variations have been proposed to overcome these limitations such as hybrid techniques by Yasser et al. [49] which first uses CLARANS [123] to partition the data space. VDBScan [97] improves the speed of DBScan by first ordering the data and only visiting data samples outside of the radius defined by ε, the density radius. A critical analysis of most variants on DBScan can be found in Ali et al [6].

The demonstration data Gaussian distributions can be clustered as shown in Figure 3.6a, however, lower density regions of a cluster may be labelled as outliers. If the required density is reduced then lower density regions between clusters may cause them to merge as shown in Figure 3.6b. Density based clustering techniques are able to find cluster of arbitrary shapes as illustrated in Figure 3.6c. Note also that the time for clustering, as shown in the plot title, shows a significant increase over alternative methods illustrated in this chapter.

Subspace Clustering

Subspace clustering divides the data space into multiple sub-spaces and detects the clusters in these subspaces. Thus a data sample may be a member of multiple clusters in multiple sub-spaces. This is particularly useful for high dimensional data where



(a) Gaussian Data with well separated clusters, but many outliers.

(b) Gaussian Data with merged clusters and fewer outliers.

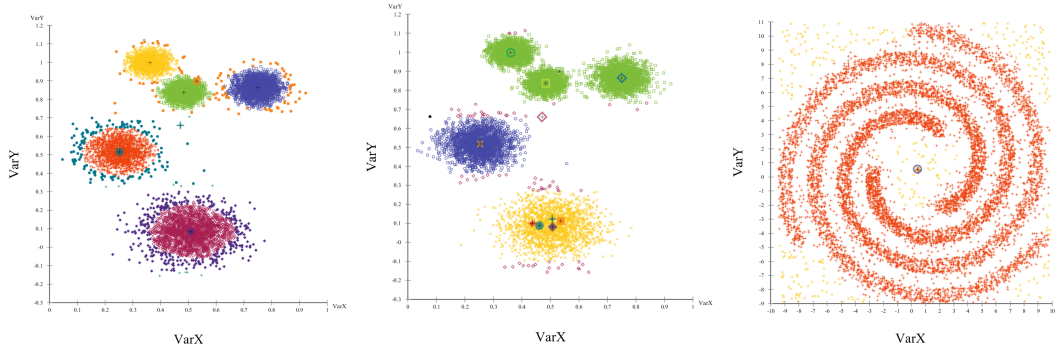
(c) Spiral Data with good clusters and outliers identified.

Fig. 3.6 (3.6a) shows the results of DBScan successfully clustering the Gaussian data, however, to separate the clusters low density portions of each cluster are identified as outliers. (3.6b) shows the results of reducing the minimum density to reduce the number of outliers has also merged two clusters. (3.6c) shows DBScan successfully clustering the spiral data set. All data not within a spiral is identified as an outlier.

the clusters in the full data space may not be of as much interest as those in certain sub-spaces, e.g. in gene expression mapping a certain gene may be associated with disease A in combination with one set of genes and disease B in combination with others. Or where such a dataset contains a value for 'age' this attribute may disperse the data so they do not form clusters if 'age' is not relevant to the diseases then clusters will only be present in a subspace if it does not include 'age'.

Subspace clustering falls into two broad groups: Top Down and Bottom up. Top down approaches such as PROCLUS [4] and FINDIT [159] find clusters in the full data space, then evaluate the subspaces of each cluster. Bottom up approaches such as Mafia [68], Clique [5], SubClu [88] and others find clusters in low dimensional subspaces, the lowest being each data axis, and combining them in higher dimensional subspaces to form higher dimensional clusters. A review of subspace clustering techniques can be found in [165, 128].

Subspace clustering is primarily aimed at finding clusters in low-dimensional data subspaces that may otherwise be too sparse in higher dimensional sub-spaces. Further, because the data may belong to more than one cluster in different subspaces this introduces additional sources of uncertainty and possible confusion we wish to avoid in this work. However, in future work subspace clustering may have particular relevance in rapidly moving between the clustering of different subsets of data. Some techniques may function well in typical full data space clustering situations. An example of such results from subspace clustering can be found in Figure 3.7. These examples have been generated using SubClu and they illustrate the general technique and limitations.



(a) Gaussian Data with well separated clusters, but many outliers.

(b) Gaussian Data with merged clusters and fewer outliers.

(c) Spiral Data cannot be clustered.

Fig. 3.7 (3.7a) shows the results of SubClu successfully clustering the Gaussian data, however, to separate the clusters low density portions of each cluster are identified as outliers. (3.7b) shows the results of reducing the minimum density to reduce the number of outliers results in the merging of nearby clusters. (3.7c) illustrates a limitation of SubClu as each subspace consists of only a single cluster despite clearly separate groups.

3.3.3 Online, Dynamic and Evolving Clustering Terminology

Recent technological advances in many disciplines have seen an increase in the amount of data being provided in continuous streams of data, i.e. 'on-line data'. These data streams range from machine condition monitoring and atmospheric science data to social media analysis. The analysis and clustering of data streams has become increasingly important [14]. However, condition monitoring can suffer from sensor drift due to ageing, temperature fluctuations, modifications or upgrades to machine components, changes in load or type of use. Environmental monitoring will also be affected by sensor drift, but also seasonal variations and secular trends due to technological, socio-economic or climate change. While seasonality and other cyclic periodicities can be moved relatively easily off-line, any attempt to do this online renders the analysis vulnerable to aliasing changing seasonal cycles into secular changes. Other problem datasets are short-term but high-dimension and rapidly changing: chemical batch processors [50], environmental mesocosms [161], or ecological manipulation experiments [125], for instance. Social media analysis will be affected by the inevitable changes in peoples' taste, population changes and many other influences. In examples such as these the assumption of a stationary data environment is invalid and techniques for data analysis need to be capable of coping with evolving data streams. It is often the case in such data, particularly that incorporating spatial or relational information, that clusters of related data will not be hyper-elliptical and will fall into arbitrarily shaped groupings. The cases for arbitrary shaped clusters are well established and found in many sources [30, 131, 129]

. Specifically a case such as that shown in [129] demonstrates the need for evolving clusters of arbitrary shapes - as the nature of the landscape changes over time, so must the clusters.

The ability to adapt our analytic to these secular (non-periodic) changes requires not only a method of reducing the importance of old data but also a way to divide previously singular clusters of data into multiple clusters. With many previously available techniques discussed in this section this is achieved, not by dividing the clusters in an online manner, but rather by re-clustering using an offline clustering technique on demand. With ever-increasing data sets, i.e. 'Big Data', the need to discard or archive the data after processing once becomes necessary for both computational and memory efficiency.

Online clustering differs considerably from offline clustering. The aim of online clustering is to group data into clusters, as defined by Table 3.2, from streams of data. These streams of data may be open ended resulting in data sets that would be too large to remain in memory. The data can be discarded completely and / or archived for later use.

Data can be assigned to a cluster as they arrive, and these results stored along with the data. This, however, has inherent dangers. As clusters change, move and evolve over time the cluster assignment originally assigned to the data may no longer be correct at a later point in time. It is easily conceivable that two similar data samples could be assigned to separate clusters simply due to the temporal difference. In such a case it should be a matter of record that the cluster assignment was only correct at the time it was made.

As an alternative, the data space regions covered by the clusters could be evolving and, when the cluster assignation is required for any data they are checked with the current cluster status and the assignment made. In this way the cluster assignation is current and correct with the latest cluster state. This however has the complications in finding out the historical cluster assignation at the time the data sample was taken and it is possible that two data samples of similar values find themselves in the same cluster, when we would prefer them to be separate due to the temporal separation. It is important, therefore, to select the correct technique for the intended use of the data and clusters.

Online clustering techniques can be broadly categorised by the type and nature of the data streams they are operating on. There is no formal agreement on the terminology so this thesis uses perhaps the most common descriptive names for these different types: 'Dynamic Clustering' and 'Evolving Clustering'.

Dynamic Clustering

Where data will arrive into clusters which we wish to keep then we shall refer to this as "Dynamic Clustering". In this case the clusters, once formed, may move or adjust

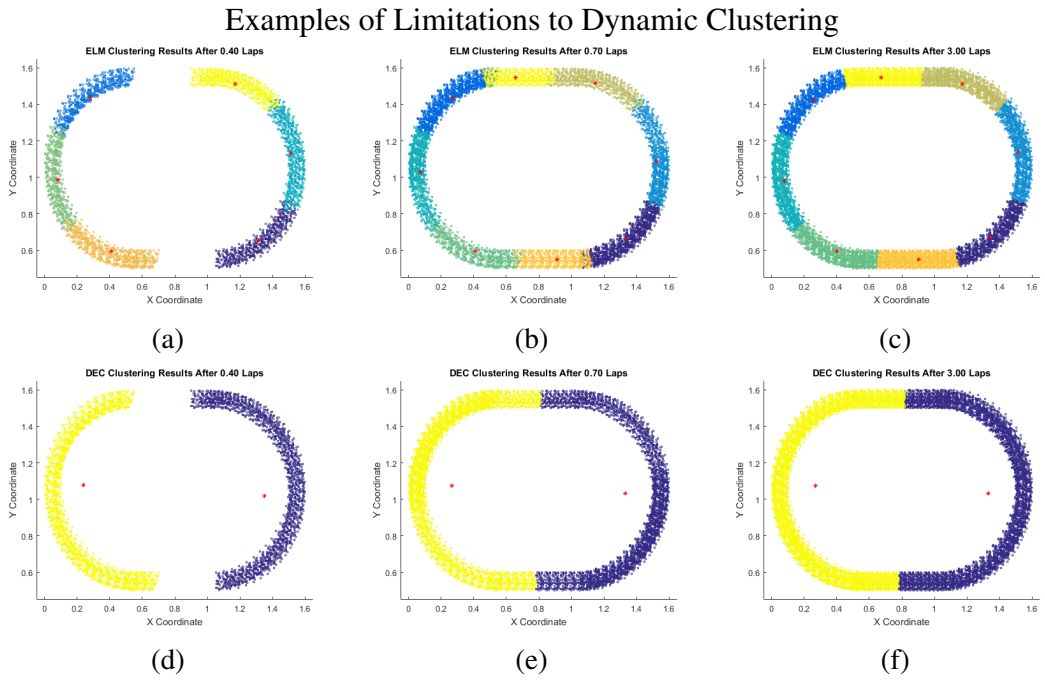


Fig. 3.8 Plots of clustering results for two dynamic clustering techniques, the top row is ELM, bottom row DEC, showing different techniques for dealing with unrestrained cluster growth. The plots show clustering of 2 groups of people running three laps of an oval track. ELM places a user-defined hard limit on the cluster radius resulting in multiple clusters. When people arrive at a location where data has previously been clustered, they join that cluster. DEC does not have a limit for the cluster radius and so they continue to grow until they meet. At this time the people from one cluster move into the other cluster.

their size and position to better group the data. In some cases clusters may merge as data arrives in the spaces between them. Dynamic clustering has no 'ageing' parameter and so clusters, once formed, remain indefinitely. In this thesis we will refer to data streams with such natural clusters as 'Dynamic Data Streams', i.e. as the natural clusters may move or change size, but are not analogous to biological evolution in that the clusters do not die and are not born into new generations. It doesn't require much imagination to conceive of a situation in which the data space become completely full with both data and their respective clusters. This is particularly true of dynamic systems in which the data can vary across the full range of their respective limits.

Consider an example such as two groups of students arriving at a sports running track. They form groups at opposing side of the track. As more people arrive the groups get larger and the cluster centre and radii (or equivalent) are dynamically adjusted. However, if the students then start to run around the track and the cluster parameters continue their dynamic adjustments then, without some appropriate limitations, the cluster centres and radii (or equivalent) are adjusted until the clusters all overlap and merge into one. With

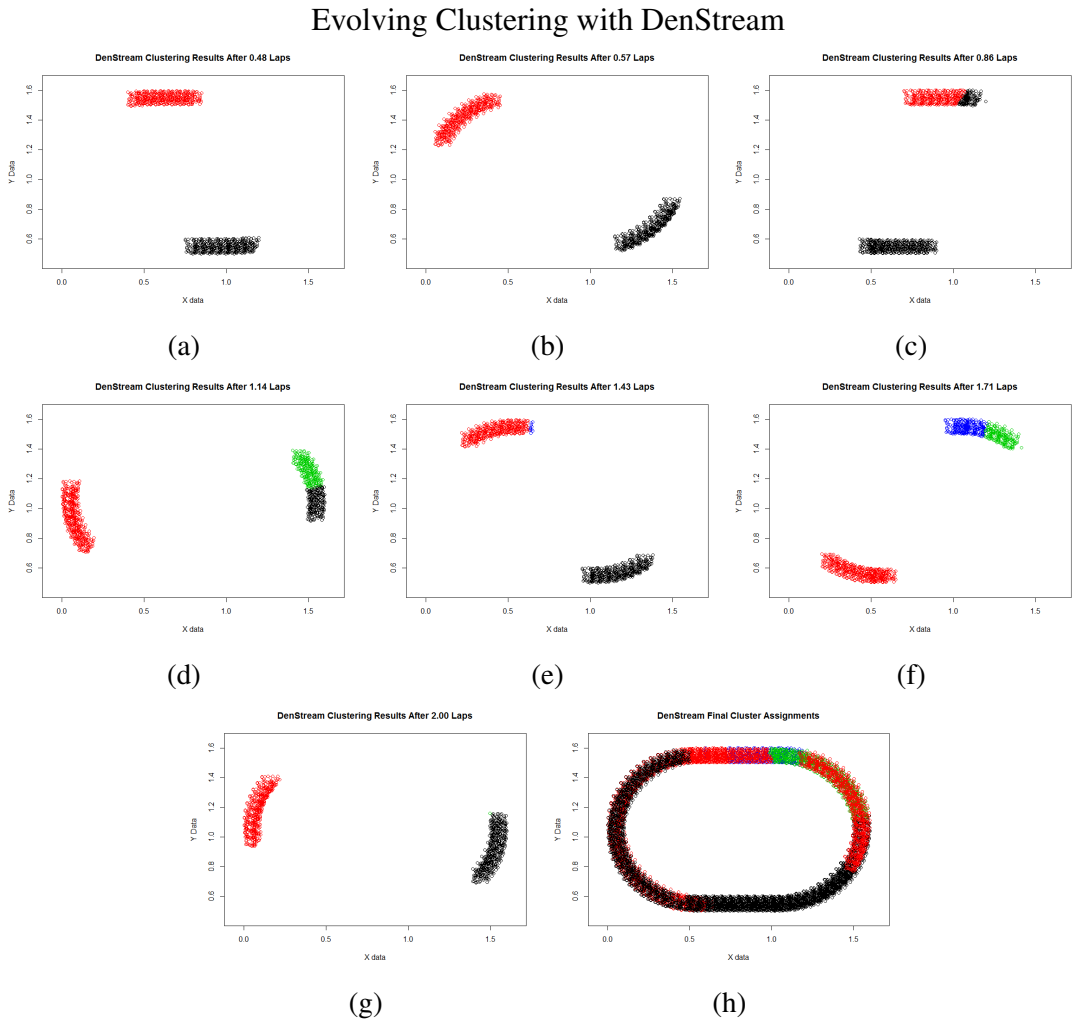


Fig. 3.9 Plots of DenStream clustering of the two clusters moving around the race track. With a value of $\varepsilon = 0.1$ at some times the clusters are divided, 3.9c, 3.9d, 3.9f giving rise to the overall assignments shown in 3.9h. While not perfect, the clustering creates a more temporally accurate result than the Dynamic clustering.

limitations in place to prevent this then either multiple clusters occur, Figures 3.8a-3.8c, or the clusters grow until they meet, Figures 3.8d-3.8f.

Evolving Clustering

An alternative type of data stream is one where data, and their respective natural clusters, can 'arrive and leave', whether by data becoming old and no longer relevant or objects associated with the data no longer existing. In this thesis we will call these 'Evolving Data Streams' as the clusters may evolve, move, die or be born - analogous to biological evolution. To continue the analogy above, if the students gathered into groups on the running track and then started to run around it then a dynamic clustering technique

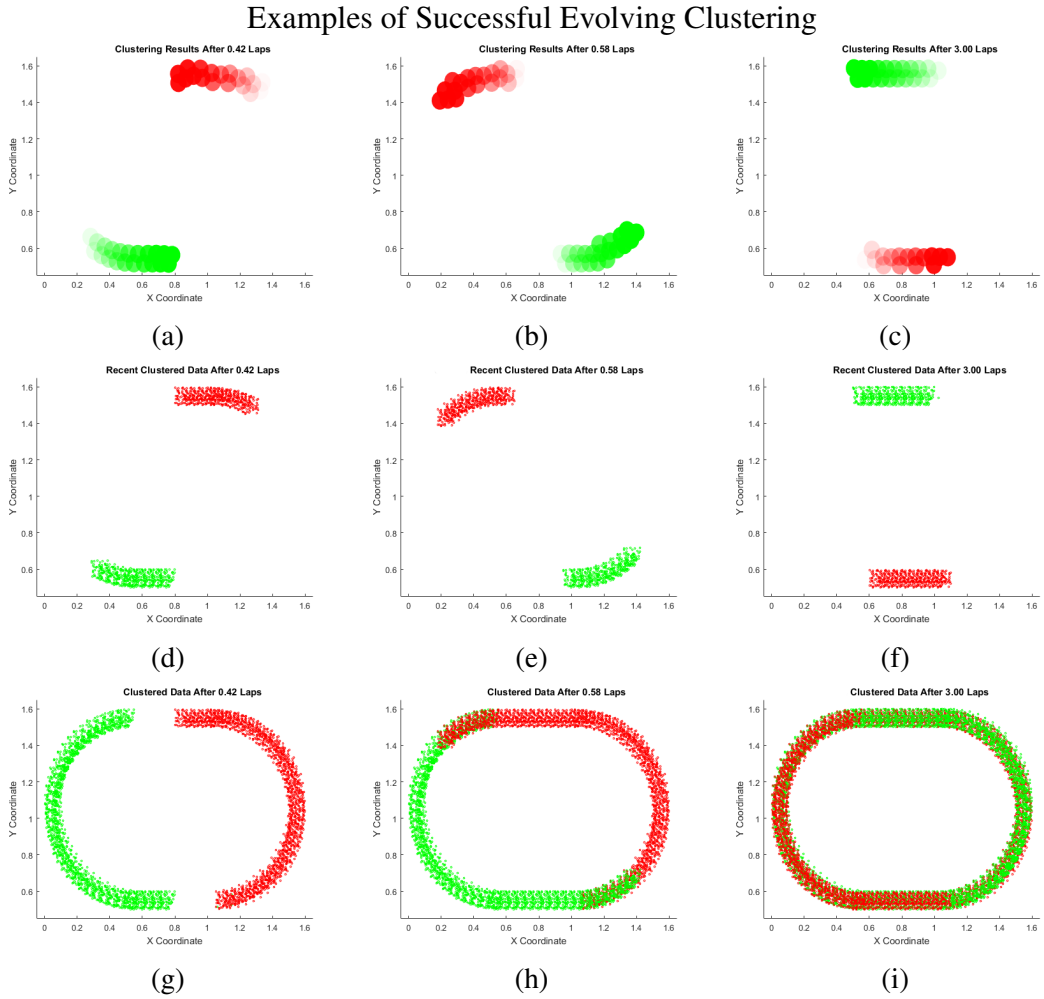


Fig. 3.10 Plots of various types of clustering results for fully evolving clustering. The top row shows the data space occupied by the cluster definitions with the transparency proportional to the age of the data. The second row shows the recent data, within the decay period. Row three shows the final cluster assignment of the data. The red and green clusters are inter-mingled showing how the separate clusters have occupied the same data space at different times. Fully evolving clustering can ensure that the data is correctly assigned to the same cluster over time.

produces the type of results in Figure 3.8. However if the technique has some time parameter that allows for a decay in old data then the clusters can fully evolve such that at any given time the clusters have evolved from their original state into a new, updated state. This would allow the clusters to follow the groups of students around the track and keep them separate. DenStream [26], as shown in Figure 3.9 can achieve this to some degree with appropriate tuning of parameters. The ideal solution is that shown in Figure 3.10 where we see that the red and green clusters are traced as separate clusters, even when they occupy similar data space, due to the temporal difference of when they occupy that data space, i.e. all the green data is always associated with all other green data and never with the red data and vice versa.

Evolving data streams are particularly relevant in situations where historical data has a reduced effect on recent events. Such situations occur in financial transactions, machine condition monitoring and, of particular relevance here, environmental monitoring as discussed at the start of this chapter.

3.3.4 Online Clustering Techniques

Online clustering techniques fall into two main classes, 'single-stage' and 'multi-stage'. Single-stage techniques complete the cluster updates in a single pass as the data arrives whereas multi-stage techniques use (typically) a two step algorithm whereby 'micro-clusters' (*mC*) are updated online as the data arrives and 'macro-clusters' (*MC*) are created from agglomerations of these micro-clusters.

Single stage techniques are invariably distance based, producing hyper-spherical, or hyper-elliptical, clusters in the data space. ELM [47] and DEC [15] are both single stage techniques however they differ in their approach to limiting cluster growth. ELM uses a bandwidth parameter which limits the cluster radius to a maximum value resulting in the clusters shown in Figure 3.8a-3.8c. The data in each cluster is indeed similar to each other, however the arbitrary boundary may result in the division of natural clusters. DEC does have an ageing parameter which reduces the importance of old data and old clusters. However the clusters are not removed completely such that, if a recent data moves into the data space occupied by an old cluster, then the new data becomes part of the older cluster. Thus DEC is somewhat of a halfway stage between dynamic and evolving clustering.

Most popular, and successful, online clustering techniques for evolving data streams are two stage hybrid processes, i.e. some form of Micro-Cluster (*mC*) is maintained online and a separate, offline, technique is used to group the *mC* together to form Macro-Clusters (*MC*). The details of these *mC* and the methods for creating them are well summarized in [141] with each technique claiming improvements in one area or another.

Another variation is that of the grid based techniques such as Clique [5], DStream [32], MR-Stream [155]. These use a grid based technique to divide the data space up into an ever finer set of hyper-cubic segments and place the incoming data into these.

It is, however, the second, offline, step that presents the greatest variations between, and limitations of, these techniques. Techniques such as Birch [167], CluStream [3] and variants [135] and [98], DGClust [61], use k-means in their processing. The result is that all the *MC* are hyper-spherical in nature and they also require a priori information regarding the number of expected clusters. ClusTree [93], DStream [32] and Denstream [26] make use of DBScan [53], while DSClu [114] uses a very similar approach, and so are capable of finding *MC*s of arbitrary shape.

The secondary, offline, stage is somewhat of a bottleneck for these techniques. While the online maintenance of the micro-clusters may be extremely fast the time penalty of the offline stage may limit the final *MC* stage to being either periodic, or on demand.

3.3.5 Compatibility Between Offline and Online Techniques

For the proposed solutions of the Atmospheric Science challenges, in particular that outlined in Section 1 it is a requirement to find compatible offline and online clustering techniques. This will allow a fast offline technique to rapidly cluster historical data in such a way that allows an online technique to take the clustering results and continue. The inherent difference between offline and online techniques, i.e. offline results provide all the data with cluster assignment, whereas online techniques provide data space regions, means that offline and online techniques are not directly compatible. It would therefore be required to develop an interface between the methods, or to develop new techniques which are compatible.

The rest of this thesis presents the work carried out to create a suite of clustering techniques that satisfy all the criteria required of the Atmospheric Science challenges, together with the development software used to demonstrate the validity of the approaches in a data gathering mission environment.

3.3.6 Summary of Current Clustering Techniques

A brief summary of the range of clustering techniques is given in Tables 3.3 (Offline) and 3.4 (Online). The example techniques are by no means exhaustive but represent the most popular and commonly cited techniques in each category. Many incremental improvements have been made, and continue to be made, for each technique. In particular some changes to the underlying techniques may improve the performances in particular environments, however the general principles remain.

Table 3.3 Summary of Offline Cluster Algorithm Types.

Type	Examples	Results	Cluster Shape
Connectivity	Hierarchical, SLINK, CLINK	Data List	Hyper-Ellipse
Centroid	K-means, K-medoids, Fuzzy C-Means, Subtractive	Data List	Hyper-Ellipse
Distribution	GMM	Data List	Hyper-Ellipse
Density	DBScan (and variants), CLARANS	Data List	Arbitrary
Subspace	SubClu, Clique, Proclus, Findit, Mafia	Data List in various formats	Hyper-Elliptical

Table 3.4 Summary of Online Cluster Algorithm Types.

Stages	*MC* Stage	Examples	Results	Cluster Shape
1	Centroid	Elm, DEC	Data List, Centroid + Radii	Hyper-Ellipse
2	Centroid	Birch, Clustream, Clustree, DGClust	$mC + MC$ or memory limited gridded data space clusters	Hyper-Ellipse (or gridded approximates of)
2	Density	Clustree, Denstream, D-Stream, DSClu	gridded data space clusters	Arbitrary (or gridded approximates of)

The ability of density linked clustering techniques to produce clusters of arbitrary shape suggest that the aim of the work here should be towards developing techniques of this type. However, those online techniques that use this form of clustering are not, in fact, fully online but rather incremental, i.e. offline second stages working on a window of micro-clusters that are maintained online.

One of the key pieces of information is that the distinct differences in the goals of Offline versus Online clustering have resulted in fundamental differences in the underlying methods and in the way the base clustering techniques report their results. These differences suggest that there are currently no compatible technique that allow instantaneous switching between Offline and Online clustering in the way we desire.

The goal of this thesis is to present suitable clustering techniques for the range of Atmospheric Science Challenges presented in Section 2.4. Section 3.3 outlines the requirements which are summarised in Table 3.1. While there are techniques available which can address many of these requirements, none are capable of solving them all. Perhaps the most difficult challenge is that of the compatibility between online and offline techniques and their cluster descriptions. This is, perhaps, most significant when considering the possibility of switching between offline and online clustering during a mission. Allowing clustering of recent historical data from the streams, followed by on-going online clustering, provides a more suitable environments for clustering of a high number of data streams. The alternative to such switching would be to continuously cluster every possible combination of data streams, something that is impractical where there are hundreds of data sources. It is the desire for compatible online and offline clustering that has driven the development of the new algorithms contained in the following chapters.

3.3.7 Applications of Clustering

Clustering algorithms have been applied across a range of different applications. Generally, the cluster analysis tends to be applied to offline datasets, or online data streams, but not any form of mixing between the two. Within atmospheric science offline techniques have been applied to atmospheric circulation, [33], analysis of ozone observations [103], hazard assessment, [28] and emissions characterization [41]. On-line, or real-time, methods have been applied to natural hazard monitoring [66, 142] and pollution threshold data for public health alerts [111, 72, 17]. Due to recent technological advances the analysis and clustering of data streams has become increasingly important [14]. Many of these data streams are sensitive to drift, e.g. sensor ageing in condition monitoring, climate change, societal changes in population, peoples taste etc. in social media analysis. These

temporal variations demonstrate the case for online analysis to be evolving, rather than simply dynamic.

High dimensional data streams such as chemical batch processors [50], environmental mesocosms [161], or ecological manipulation experiments [125] also indicate that it may not always be practical to apply cluster analysis to every available combination of data sources within a data stream. In this case it may be advantageous to analyse a few, indicative, data sources until a trigger event occurs. At this time analysis of alternative data sources becomes necessary. As online cluster analysis operates on data samples as they arrive there is a delay between starting to analyse data and meaningful results becoming available. In such a case it would be beneficial to be able to analyse 'recent 'historical data offline, before continuing to update the results in an online manner.

In many data streams the assumption of a stationary data environment cannot be justified and analytical techniques should be able to adapt the results to cope with the evolution of the data. Frequently, in cases with evolving data streams, in particular those with a spatial or temporal factors, clusters will not be found to be hyper-elliptical and may inhabit the data space such that any hyper-ellipses enclosing the data would overlap. Distance based cluster membership techniques are unable to cope with such cases and algorithms that provide arbitrarily shaped clusters are needed. The case for arbitrary shaped clusters are well established and found in many sources [30, 131, 129].

There are no currently available clustering solutions that address all of the requirements set out in this thesis. Previous applications of clustering have been applied to online analysis or offline analysis, but not combined, and many of these techniques are limited, e.g. to hyper-elliptical shapes etc. This thesis presents novel clustering algorithms which satisfy:

1. Offline clustering requiring minimal, intuitive parameters
2. Offline clustering requiring no user input
3. Online clustering for dynamic data streams, not limited to hyper-elliptical clusters
4. Online clustering for evolving data streams, not limited to hyper-elliptical clusters
5. Online and offline clustering algorithms providing compatible arbitrarily shaped clusters, allowing the reproduction of online analysis in an offline mode for subsets of a data stream.
6. Online and offline clustering algorithms whereby the compatibility of the arbitrarily shaped clusters allows for offline clustering of historical data, prior to on-going updating of the clusters on-line.

These new algorithms allow for the development of clustering based analysis software for use with atmospheric science data, whether online or offline.

3.4 Cluster Quality Measures

Cluster quality measures fall in to three main categories: internal; external and relative. A number of review papers summarise the various common methods [100, 13, 16, 136] and this section provides an overview of the most popular methods and the selection of the methods used throughout this thesis. The internal and external measures discussed are summarised in Table 3.5. Relative measures are not discussed separately as they consist of repeated applications of internal or external measures and comparing the result in order to provide a comparison between results rather than an absolute measure.

3.4.1 Internal Measures

Internal measures of cluster quality such as Dunn Index [46], Calinski Harabasz [25], Davies-Bouldin Index [43] and Silhouette [138] use some metric of the resultant clusters partitioned by the algorithm. Typically these metric are some way of measuring how similar data is within a cluster and how different it is from data in other clusters. While these measures work well with compact, well-separated clusters they are less clear when used on clusters from noisy data or where there are no clear distinction between natural clusters, e.g. the type of data resulting form the mixing of fluids. Generally, these are unsuitable for arbitrarily shaped clusters, especially where any other clusters intersect the hyper-ellipse generated by any distance based similarity measure.

3.4.2 External Measures

External measures compare the results of a clustering algorithm with an external 'ground truth'. The ground truth is an accepted result considered to be correct, whether by experimentation, known facts or expert opinion. This is especially useful when dealing with synthetic data sets where the clusters have been pre-determined. When dealing with real data opinions may differ as to the what constitutes the ground truth, resulting in differing results. Common external measures include Rand Index [133], Adjusted Rand Index [153] (Rand index corrected for chance grouping), Jaccard Index [83] and the Fowlkes-Mallows Index [59].

Table 3.5 Common Cluster Quality Analysis Techniques

Name	Type	Equation	Variables
Dunn [46]	Internal	$DI_m = \frac{\min\limits_{1\leq i<j\leq m} \delta(C_i,C_j)}{\max\limits_{1\leq k\leq m} \Delta_k}$	$\delta(C_i,C_j)$ inter-cluster distance metric, m the number of clusters, δ_k mean distance between data in cluster k
Davies-Bouldin [43]	Internal	$DB \equiv \frac{1}{N}\sum\limits_{i=1}^{N} \max\limits_{j\neq i}(\frac{S_i+S_j}{M_{i,j}})$	N number of clusters, S in cluster separation, M between cluster separation
Silhouette [138]	Internal	$s(i) = \frac{b(i)-a(i)}{max\{a(i),b(i)\}}$	a in cluster dissimilarity, b dissimilarity to nearest neighbouring cluster
Rand Index [133]	External	$R = \frac{a+b}{\frac{n}{2}}$	a the number of data in the same cluster, b the number of data in different clusters, when comparing algorithm results and ground truth
Jaccard [83]	External	$J(A,B) = \frac{\lvert A\bigcap B\rvert}{\lvert A\bigcup B\rvert}$	A,B are clusters being compared
Fowlkes-Mallows [59]	External	$FM = \sqrt{\frac{TP}{TP+FP}\cdot\frac{TP}{TP+FN}}$	TP true positives, FP false positives, FN false negatives

3.4.3 Quality Measures Used in this Thesis

Throughout this thesis cluster quality is evaluated using 2 techniques, Purity and Accuracy. Purity is used primarily for comparison to other algorithms where they have used this measure. It is an inherently unsatisfactory method as it is easily fooled, e.g. a high purity score is obtained by placing each data sample into its own cluster. Given that one of the primary aims of clustering is to place all similar data into the same group, it could be argued that a good purity score may be indicative of *poor* clustering in some cases.

The accuracy measure used in this thesis is an external measure comparing the cluster results to the ground truth of the data set and the overall equation is shown in Equation 3.3.

$$accuracy = \frac{\sum_{i=1}^{n} \frac{N_i^d}{N_i}}{N_s} \tag{3.3}$$

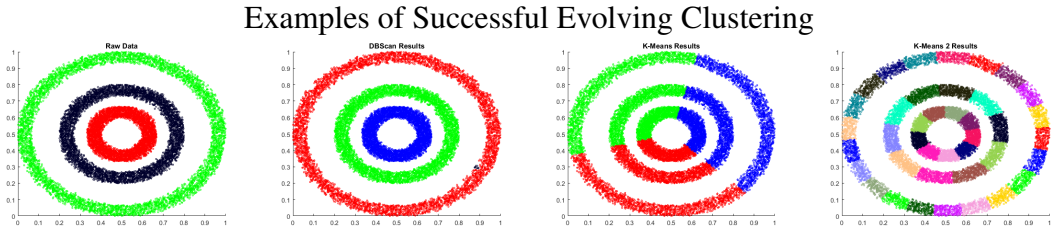
where N_i^d is the number of samples in dominant natural cluster d, N_i is the number of samples in cluster i and N_s is the total number of samples. The 'dominant natural cluster 'is the natural cluster into which the majority of the algorithm derived cluster data falls, i.e. the natural cluster that the algorithm cluster is closest to matching.

The basis for this calculation is a modification of the Jaccard index such that it provides an overall value for all the resulting clusters. To compare with the Jaccard equation in Table 3.5 we see that:

$$\frac{N_i^d}{N_i} \text{ is equivalent to } |A \bigcap B|$$

We divide this value by the number of data in the algorithm generated cluster only. This calculates the fraction of assigned data in the algorithm cluster that have been correctly assigned to the equivalent natural cluster. By summing these values for all the clusters, including the 'cluster' of outliers where appropriate, we have an overall value for the number of data that have been accurately placed into appropriate clusters. Dividing this result by the total number of data samples gives the overall value for the fraction of the data that have been accurately placed in appropriate clusters where $0 < Accuracy \leq 1$ with $Accuracy \to 0$ being few data correctly assigned and $Accuracy = 1$ being perfectly matching clusters and outliers. It is this ability to allow for a cluster of outliers and arbitrarily shaped clusters which is the primary reason for adopting this external measure. In such cases data that are correctly labelled as a single cluster may not be more similar to each other than to data in other clusters, as required by internal measures.

To illustrate the utility of this measure, let us consider the raw data shown in Figure 3.11a. The cluster results are shown for DBScan, 3.11b, K-Means with k=3, 3.11c



(a) (b) (c) (d)

Fig. 3.11 Example of clustering results on arbitrarily shaped natural clusters. DBScan (3.11b) most closely matches the natural clusters, K-Means with k=3 (3.11c) is very poor, while K-Means with k=40 (3.11d) divides the natural clusters excessively.

Table 3.6 Examples of cluster validity measures for the cluster results shown in Figure 3.11

	Calinski Harabasz	Davies Bouldin	Silhouette	Purity	Modified Jaccard
Better is:	Larger	Smaller	Closer to 1	Closer to 1	Closer to 1
DBScan	6.2	583.5	-0.142	1	0.999
K-Means (k=3)	9461.0	0.93	0.501	0.336	0.336
K-Means (k=40)	20168.4	0.61	0.625	1	0.093

and K-Means with k=40, 3.11d. It can be clearly seen that DBScan provides cluster results that more closely match the natural clusters. K-Means with k=3, fails to provide meaningful clusters. K-Means with k=40 sub-divides each natural cluster in to many clusters, which may also be considered a poor result. Internal measures will typically score K-Means with k=40 as the best of the results, yet it is the DBScan results that most closely match the natural clusters. The results of some common measures are given in Table 3.6. Purity identifies both DBScan and K-Means with k=40 as being good results. However, the purity measure makes no allowance for the number of divisions of a natural cluster, only whether the data in each cluster belongs to a single natural cluster. The Modified Jaccard index used here considers only the cluster with the closest match to a natural cluster and, so, correctly identifies the cluster quality as a reflection of how well they match the natural clusters.

Chapter 4

Development and Application of Offline Clustering Techniques

4.1 Overview of Clustering Requirements

This chapter presents new offline clustering techniques developed to solve the atmospheric science Data Challenges outlined in chapter 2. The combination of techniques aims to provide a suite of compatible clustering algorithms which meet these challenges and overcome the difficulties associated with current techniques described in chapter 3. During the journey to the final suite of algorithms clustering solutions suitable for other cluster analysis scenarios have also been developed. In particular, DDC (chapter 4.2) is a precursor to the DDCAS (chapter 4.4) technique. DDC is, however, an offline clustering algorithm in its own right, demonstrating high speed and accuracy, and was also extended into DDCAR (chapter 4.3) as a parameter free clustering algorithm. Similarly, CODAS (chapter 5.2) is also the basic first step from which CEDAS (chapter 5.3) was developed, although both techniques may be utilized in the proposed RASCAL software (chapter 6) depending on the circumstances and type of analysis required.

This chapter describes the offline techniques, the reasons behind their development and a summary of the benefits of each technique together with how they fit into the overall goals of the thesis. The online techniques are described in chapter 5.

4.1.1 Implementation and Testing of the Developed Algorithms

All of the following algorithms where created and tested using Matlab®, starting from version 2012a. They have all been subsequently tested under subsequent versions up to 2016b. All testing was run under Windows 7 on a Dell® Optiplex®9010 with an Intel® Core™ i7-3770 CPU at 3.40GHz and 8GB of memory.

The code has all been written for clarity, rather than optimised for speed, and did not require background tasks to be stopped. Where the tests are made for comparisons to alternative technique the aim was to demonstrate that these new techniques have comparable accuracy and speed and are, therefore, valid methods with no significant penalties or restrictions except where mentioned.

The algorithms are tested in such a way as to demonstrate their abilities as well as their limitations. The algorithms have been implemented using the Euclidean distance measure. They will function with any distance measure suitable for specific instances, however, the Euclidean measure proves adequate for providing easy visualization and demonstrating the principles of the techniques. When working with data of different scales we normalise the data, based on a priori knowledge of the range, or utilising expert knowledge to assess the relevant importance of the distances between data samples.

4.2 A Fast, Offline Data Density Based Clustering Technique (DDC)

This section is based on the paper "Data Density Based Clustering" presented at the Computational Intelligence UKCI 2014 conference. [80]. The work is primarily that of the author with the aid of comments from the co-authors.

4.2.1 Reasons for Developing DDC

Clustering algorithms have long been considered useful methods of extracting information from large datasets, especially those with high dimensionality that are hard to visualize. As we enter the world of 'big data' the speed, efficiency, accuracy and autonomy of the methods becomes ever more important. As the size of data sets, in terms of both number of samples and data dimensions, grow small differences in algorithm efficiency start to become more significant. By minimizing the number of calculations required to calculate the density Data Density Based Clustering (DDC) [80] is seen to be computationally efficient and, therefore, of particular use in the realm of 'big data'. It can be reasoned that in such large datasets it is impractical to expect the user to know the number of clusters expected in the data. Indeed, it can be argued that one of the main reasons behind the use of clustering in data mining is the discovery of such hidden information as the number of natural clusters.

4.2.2 Principles of the Algorithm

Returning to first principles of some of the earliest clustering techniques we consider K-means [105], Subtractive clustering (based on the Mountain Method [164]), Hierarchical and, generally, grid based methods. It is known that k-means requires the number of clusters to be pre-determined and that Subtractive clustering requires repeated visits to every data sample in the data set, although it can discover the number of clusters. Thus these techniques and others based on them may be discounted. Hierarchical clustering [109] and other linkage techniques [140, 44] have a tendency to high memory or storage requirements while grid based methods also require a secondary process.

In Subtractive clustering the value of the 'potential' for a data sample is the sum of the distances from this data sample to all other data. In subtractive clustering this is re-calculated after each iteration. However, if a technique similar in operation to subtractive clustering could be achieved without the need to repeatedly re-visit every data sample and calculate the potential then the required number of clusters could be generated without the same time penalty. To this end Recursive Density Estimation (RDE) [9] is employed to recursively update the data sample with the highest potential. By combining this data sample with a user defined radius and learning parameter we can accurately find an appropriate number of clusters.

This clustering technique, being based on the data density, was termed Data Density based Clustering (DDC) [80]. The basic algorithm, calculating the potential of each data sample to be the cluster centre, and clustering the data within a user specified radius, functions in a similar manner to Subtractive clustering and suffers from the same sensitivity to the user defined initial radius and likelihood of divided natural clusters. To overcome this, additional steps are included to refine the clusters and improve the cluster quality and accuracy. These steps allow a larger than expected initial radius which is then adapted to better match the data within the cluster.

4.2.3 DDC Algorithm

The full mathematical steps for the algorithm are given in Appendix A and a descriptive overview is provided here.

To initiate the clustering process a user defined parameter, the initial radius r_0 is required. The initial radius can be a vector of radii for each data dimension or a scaler, in which case the radii are equal. In the case of a vector the initial cluster definition may be hyper-elliptical. In the case of a scalar the initial cluster will be a hyper-sphere, however, when the algorithm adjusts the radii to match the data these may alter to becoming hyper-elliptical. Then the Global Mean, μ_0, Global Scalar Product, X_0, and Global Density, D_i,

Algorithm 1: DDC Algorithm

Input: {Data}, r_0
while $\{Data\} \neq \emptyset$ **do**
  Find global densest sample from data set and assign as cluster centre
  Assign data to cluster centre
  Remove outliers
  Find local densest sample and assign as cluster centre
  Assign data to new centre
  Remove outliers
  Adjust radii to match data
  Remove assigned data from {Data}
end
Merge clusters whose centre lies within another cluster

are calculated using equations 4.1, 4.2 and 4.3 respectively, where N is the number of data samples and x_0 is the data sample for which we are calculating the density.

$$\mu_0 = \frac{1}{N}\sum_{i=1}^{N}(x_i \in \{Data\}) \tag{4.1}$$

$$X_0 = \frac{1}{N}\sum_{i=1}^{N}(x_i \in \{Data\})^2 \tag{4.2}$$

$$D_i = \frac{1}{1 + \|x_i - \mu_0\|^2 + X_0 - \|x_0\|^2} \tag{4.3}$$

The Global Density of each data sample is, therefore, a measure of the combined distances to every other data sample. The data sample with the highest Global Density is temporarily assigned as the cluster centre and all data within the cluster ellipse are temporarily assigned to the cluster, $\{C_j\}$ as defined by Equation 4.4.

$$\{C_j\} \ni \sum_{i=1}^{N}\sum_{k=1}^{d}\frac{[(x_{ik} \in \{Data\}) - \mu_{ik}]^2}{r_0^2} \leq 1 \tag{4.4}$$

Any data samples whose distance from the cluster centre $> 3\sigma$ are considered outliers and removed from the cluster.

It is often the case that the globally densest data sample may be on, or near, the edge of a natural cluster. The positioning of the cluster centre is refined by finding the Locally Densest Sample and moving the cluster centre to that data sample using equations 4.5, 4.6 and 4.7.

$$\mu_l = \frac{1}{N}\sum_{i=1}^{N}(x_i \in \{C_j\}) \tag{4.5}$$

$$X_l = \frac{1}{N}\sum_{i=1}^{N} ||x_i \in C_j||^2 \tag{4.6}$$

$$D_i = \frac{1}{1 + \left\|(x_i \in \{C_j\}) - \mu_0\right\|^2 + X_0 - \left\|(x_i \in \{C_j\})\right\|^2} \tag{4.7}$$

Again data samples which lie outside, $D_i > 3\sigma$, distance from the cluster centre are considered outliers and removed from the cluster. The clustering of the data is now considered to be complete and we adjust the radii of the cluster to match the included data. It is possible to further refine the cluster by repeated application of these steps, however, the additional processing time for rapidly diminishing returns indicated that a single iteration is sufficient in all the cases tested here.

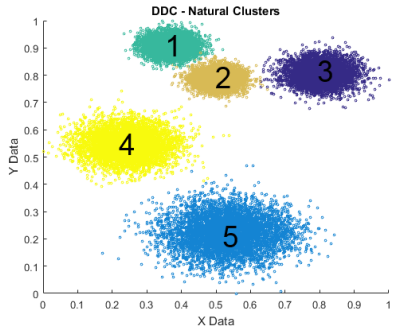
This is the core algorithm of DDC and will accurately cluster data which forms natural clusters of a similar size and shape. However, where natural cluster sizes vary, and in particular where smaller clusters are in close proximity, choosing initial radii big enough to cover the larger clusters will merge the smaller clusters. The alternative, selecting small enough radii to separate the small clusters will divide the larger natural cluster. However, it is a feature of DDC that, should a larger natural cluster become divided into smaller DDC clusters, then these cluster centres may lie within the hyper-ellipse of others. Exploiting this feature to create a 'merge' function combines these sub-clusters towards the full natural cluster.

This is visualised in Figure 4.1. We show an example data set in Figure 4.1a with natural clusters numbered. Selecting radii appropriate for the largest cluster, 5, provides reasonable results for that cluster. However, where there are smaller natural clusters in close proximity, 1, 2 and 3, it is seen that cluster 2 overlaps into the others producing the poor results shown in Figures 4.1b and 4.1c. Selecting radii more appropriate for these smaller natural clusters may result in the larger natural clusters becoming divided, again with poor results, Figure 4.1d. However, merging clusters whose centres lie within nearby clusters produces the superior results shown in Figure 4.1e.

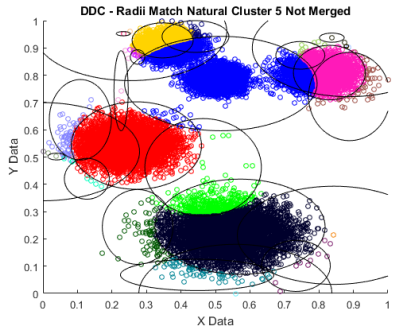
4.2.4 Testing DDC by Clustering of Synthetic Data

The first step in testing the DDC algorithm is clustering on synthetic data, created along the same lines as the algorithm, i.e. Gaussian distribution around a central point distributed along varied hyper-elliptical, axis-orthogonal axes. Such a dataset is that generated to produce the examples shown in Figure 4.1 and discussed in Section 4.2.3 above. The dataset presents particular difficulties in the varied sizes of the natural clusters, but especially in the proximity of natural clusters 1, 2 and 3.

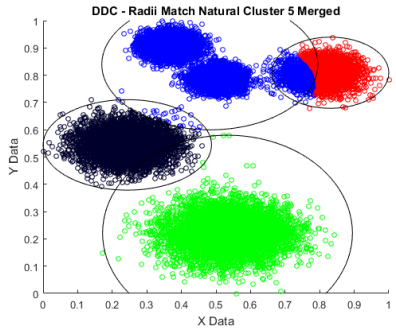
To test the algorithm further and to demonstrate its limits the dataset DS2, shown in Figure 4.2 is used. The raw data is shown in the first image, 4.2a and is particularly



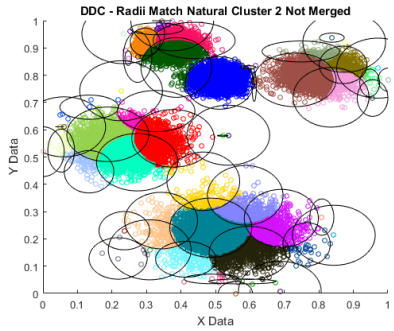
(a) Data



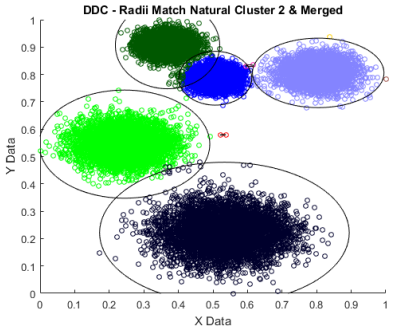
(b) Large radii, un-merged



(c) Large radii, merged

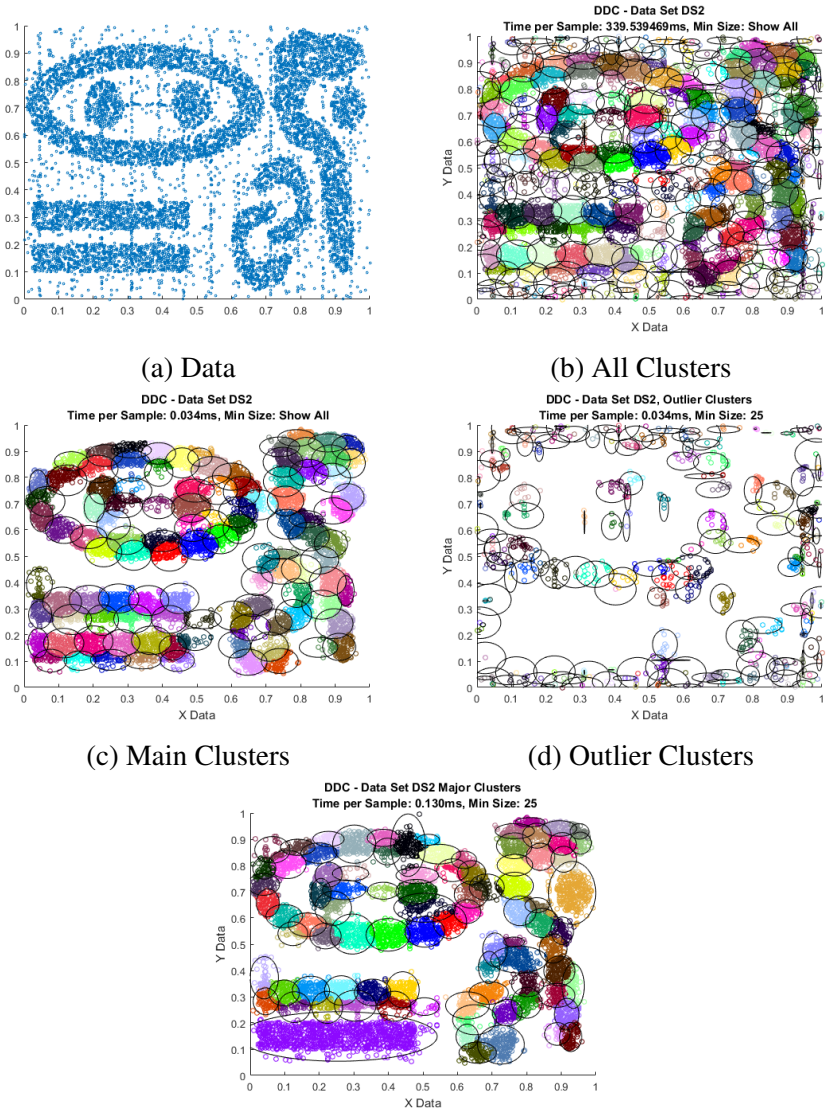


(d) Small radii, un-merged



(e) Small radii, merged

Fig. 4.1 Visualizations of the discussion in Subsection 4.2.3. Figure 4.1a shows the data with the natural cluster numbered. Figure 4.1b shows the results of un-merged clustering with radii suitable for natural cluster 5. Figure 4.1c shows the clusters merged both illustrating how the clusters overlap nearby natural clusters if the radii are too large. Figure 4.1d shows how small radii divide larger natural clusters. However, merging the clusters produced by smaller radii produces superior results as shown in Figure 4.1e



(a) Data

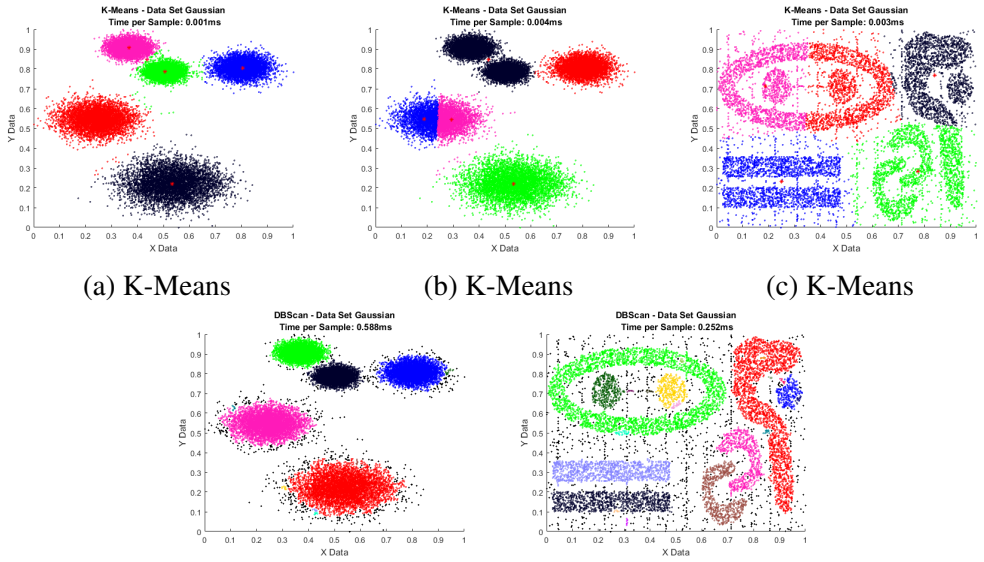
(b) All Clusters

(c) Main Clusters

(d) Outlier Clusters

(e) Main Clusters Merged

Fig. 4.2 Visualizations of the discussion in Subsection 4.2.4. Figure 4.2a shows the raw data set. Figure 4.2b shows the results of un-merged clustering with all data and clusters shown. Figure 4.2c shows only the clusters with >25 members. Figure 4.1d shows how DDC is capable of separating and identifying small groups of outlier data, where other techniques simply discard outliers. Merging the main clusters shows that even a simple merging routine can start to produce more meaningful results from even the most awkward shapes, Figure 4.2e



(a) K-Means (b) K-Means (c) K-Means

(d) DBScan (e) DBScan

Fig. 4.3 Visualizations of the discussion in Subsection 4.2.4. Figure 4.3a shows K-Means successfully clustering the Gaussian data, however this proved unreliable and frequently gave results as seen in 4.3b. Figure 4.3c shows K-Means is unable to make any sense of arbitrarily shaped clusters. Figure 4.3d shows DBScan successfully finding the Gaussian natural clusters, however a large number of outliers results form the high density required to prevent merging. DBScan excels at clustering arbitrary shaped data, Figure 4.3e, where the cluster quality more than outweighs the time penalty.

unsuitable for a technique intended to discover hyper-elliptical, axis-orthogonal clusters only. Indeed, the results shown in Figure 4.2c seem messy and not particularly useful. It must be remembered though that DDC clusters *all* the data and so this figure shows every data sample, even if it is a lone member of a cluster. Figure 4.2d shows the clusters and data with those clusters with less than 25 members removed. It can now be seen that the main natural cluster shapes are well represented, even if they are divided as would be expected. Additionally, because DDC clusters all the data, where other techniques, e.g. even DBScan [53], discard outliers, it is possible to interrogate the results and show only the outliers, defined as 'small clusters' (<25 members in this example), we can in fact identify the outliers in their respective clusters as shown in Figure 4.2d.

Figure 4.2e shows these same results, but with merging implemented. It can be seen that the merging reduces the number of clusters and, in some cases, groups entire natural clusters successfully. It is a property of the merging technique used that it is designed to still create hyper-elliptical clusters and it is this that perhaps most limits its usefulness for the intended atmospheric science procedures. It is, however, the first indication that DDC is a step along the path to the goal of arbitrary shaped clusters. To quantify the results we measured the cluster purity according to Equation 4.8 and cluster accuracy

according to Equation 4.9. The purity gives an indication of the similarity of the data in each cluster and whether it belongs together, or should have been assigned to a different cluster. The purity value is easily skewed, as with any mean calculation, by a having a large number of small clusters with high purity and a low number of large clusters with low purity. So an accuracy measure is also provided which gives an overall picture of the number of data samples correctly assigned to a proper and pure cluster.

$$purity = \frac{\sum_{i=1}^{n} \frac{N_i^d}{N_i}}{n} \tag{4.8}$$

Where N_i is the number of samples in cluster i, N_i^d is the number of samples in cluster i in dominant natural cluster d and n is the number of clusters.

$$accuracy = \frac{\sum_{i=1}^{n} \frac{N_i^d}{N_i}}{N_s} \tag{4.9}$$

where N_i^d is the number of samples in dominant natural cluster d, N_i is the number of samples in cluster i and N_s is the total number of samples.

The results are given in Table 4.1 and we see that DDC compares well, approaching the speed of K-Means, with the cluster purity and accuracy of both K-Means and DBScan. However, the number of clusters is a significant factor and the ability of DBScan to produce arbitrarily shaped clusters means it excels on the DS2 data set. The ability of DDC to cluster all the data, without forcing it into inappropriate clusters like K-Means, results in the ability to identify and distinguish outlier groups, as shown in Figure 4.2d, and is a significant advantage.

Neither the Gaussian cloud nor DS2 dataset are especially large so these datasets may not be particularly good indicators of the algorithm speed, however, with a mean clustering speed of around $0.004ms$, un-merged, and $0.021ms$, merged, per sample for the Gaussian data and $0.077ms$, merged, and $0.034ms$, merged, per sample for the DS2 data set it demonstrates a speed comparable to k-means at $0.001ms/sample$ and far better than DBScan at $0.45ms/sample$. However k-means, with the original technique of random seeding, was unable to reliably form the correct natural clusters for the Gaussian data as shown in Figures 4.3a, 4.3b and failed completely on the DS2 data, Figure 4.3c. DBScan also proves difficult to tune the parameters and find reasonable results. The close proximity of the natural clusters resulting in similar local densities between and within the clusters resulting in a high number of samples labelled as 'outliers' as shown in Figure 4.3d, however, when tuned the results were repeatable and reproducible. DBScan is considerably better when used on data set DS2, the type of arbitrarily shaped natural clusters it is designed to work on, this is shown in Figure 4.3e. Where faster

Table 4.1 Purity, Speed and Accuracy comparisons between DDC and alternative techniques.

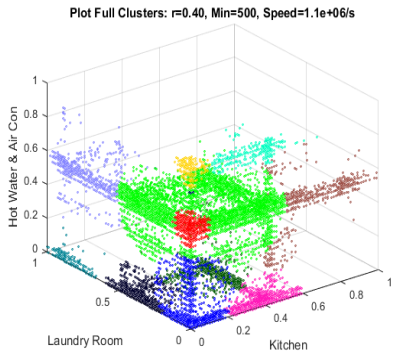
Data Set	Technique	Purity (%)			Accuracy	Time / sample
		Min	Max	Mean	(%)	(ms)
Gaussian	DDC Merged	99.75	100	99.9	99.9	0.017
	DDC Un-Merged	87.94	100	99.44	99.37	0.004
	K-Means Good	99.29	100	99.78	99.78	0.001
	K-Means Poor	50.03	100	89.89	79.95	0.001
	DBScan	99.91	100	99.97	99.97	0.46
	Mean Shift	99.15	99.98	99.79	99.79	0.008
	ELM	99.66	99.98	99.86	99.85	0.05
DS2	DDC Merged	66.67	100	99.07	99.17	0.077
	DDC Un-Merged	56.82	100	99.05	99.33	0.034
	K-Means	51.44	95.68	67.24	66.92	0.013
	DBScan	95.22	100	99.3	99.63	0.197
	Mean Shift	78.17	100	97.48	97.27	0.014
	ELM	82.87	100	97.03	96.92	0.067

techniques are unable to cluster the arbitrarily shaped groupings of the DS2 data set, DBScan produces extremely good results.

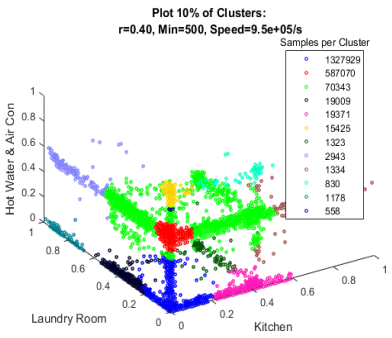
4.2.5 Grouping Users of Household Power by DDC

DDC is intended to be a fast, offline clustering technique capable of dealing with large datasets with a high number of samples and a high number of dimensions without significant time penalty. It should, therefore, be suitable of creating reasonable results from a large data set such as the Individual Household Electric Power Consumption dataset [63] available from the UCI Machine Learning Repository [86]. This dataset contains 2,075,259 measurements gathered between December 2006 and November 2010 (47 months) and has 9 attributes, each of which has been normalised to the range 0-1:

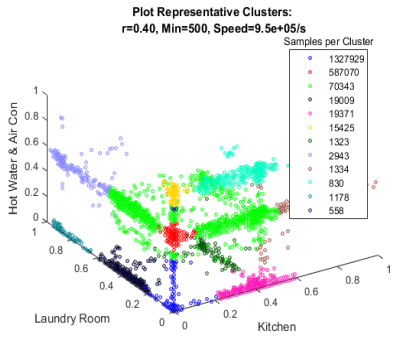
1. date: Date in format $dd/mm/yyyy$
2. time: time in format $hh:mm:ss$
3. global active power: household global minute-averaged active power (in kilowatt)
4. global reactive power: household global minute-averaged reactive power (in kilowatt)
5. voltage: minute-averaged voltage (in volt)
6. global intensity: household global minute-averaged current intensity (in ampere)



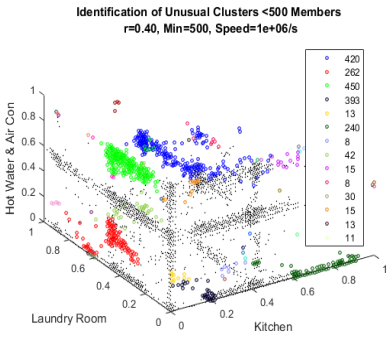
(a) Full Data



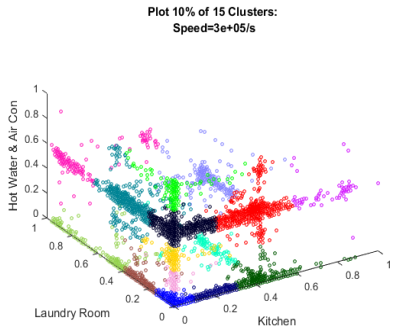
(b) 10% Data Showing Range



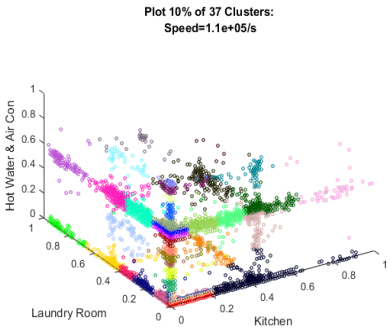
(c) Representative Data Showing Distribution



(d) Outlier Clusters



(e) K-Means, 12 Clusters



(f) K-Means, 37 clusters

Fig. 4.4 Visualizations of the Individual Household Electric Power Consumption dataset [63] clustering results. Figure 4.4a shows the full plot of all the clustered data (>2m samples). The memory requirements for the plot are such that the remaining plots use a randomly selected representative number of each cluster only. Figure 4.4b shows 10% of each cluster showing the typical range of data. Figure 4.4c also shows a random 10% but limited to a minimum of 100 and maximum of 5,000 samples to shows the typical distribution of the data. Figure 4.4d shows only the small, 'outlier' clusters with <500 members, which identifies where power usage is un-typical. Also shown are results for k-means clustering. Similar numbers of clusters were selected as shown in Table 4.3, however this did not isolate the outliers and divided natural cluster regions.

7. sub-metering 1: energy sub-metering No. 1 (in watt-hour of active energy). It corresponds to the kitchen, containing mainly a dishwasher, an oven and a microwave (hot plates are not electric but gas powered).
8. sub-metering 2: energy sub-metering No. 2 (in watt-hour of active energy). It corresponds to the laundry room, containing a washing-machine, a tumble-drier, a refrigerator and a light.
9. sub-metering 3: energy sub-metering No. 3 (in watt-hour of active energy). It corresponds to an electric water-heater and an air-conditioner.

To demonstrate the efficacy of DDC we shall consider the case of clustering the power usage data into 3 sets along each axis, representing 'High', 'Medium' and 'Low' usage in each household location, i.e. 'Laundry Room', 'Kitchen' and 'Hot Water/ Air Con'. We shall define 'unusual' clusters as being those with less than 500 occurrences over the 4 year period and the clusters will be left un-merged to indicate the results from the base algorithm. The results are shown in Figure 4.4. There is a total of 12 main clusters containing 2,047,313 data samples, approx 99.9% of the input data. The remaining data is contained in 15 'Outlier' clusters, i.e. those containing less than 500 samples.

It can be seen that the majority of household power usage data lies in the blue region, i.e. very low use in all 3 measures. When we consider that people spend, on average around 8 hours a day both asleep or out at work this is not surprising. The next largest cluster shows an increased use of 'Hot Water / Air Con' with low use of Kitchen and Laundry Room, the sort of data associated with typical home use. As would be expected the use of Kitchen and Laundry show fewer data samples which, considering these are typically short duration activities is again, not surprising.

Also of interest is the information that can be gleaned from the outlier clusters. The two largest show high use of Hot Water / Air Con while another larger cluster shows high energy use in the Kitchen. These could indicate readings of particular value as they may be due to unusual weather conditions and / or inefficient equipment.

For comparative purposes, K-Means and DBScan were tested across the same data. DBScan was unable to complete the process over a 24 hour period. K-means produces similar results to DDC shown in Figures 4.4e and 4.4f. DDC was tested with a range of different initial radii and the clustering rate and number of clusters generated recorded, as shown in Table 4.2. A range of 'k ' were used to match K-Means outputs to those of DDC both clusters and clusters including outlier clusters as shown in Table 4.3. K-Means consistently divided natural cluster regions and exposed, perhaps, its greatest weakness of forcing data into nearby, inappropriate clusters. So, although K-Means was similar in run time to DDC it fails to identify outliers in a meaningful way.

Table 4.2 DDC clustering results for various initial radii, on the Household Power dataset.

Initial Radius	Time / Sample ($\times 10^{-6}$s)	Sample Rate ($\times 10^3 s^{-1}$)	Total Clustered	Clusters	Total Outliers	Outlier Clusters
0.60	0.651	1536	2048786	6	494	17
0.55	0.652	1532	2048795	6	485	16
0.50	0.712	1403	2048361	7	919	24
0.45	0.763	1310	2048431	8	849	26
0.40	0.969	1031	2048583	15	697	39
0.35	1.072	932	2048407	18	873	52
0.30	1.155	865	2048408	19	872	64
0.25	1.416	706	2048436	32	844	82
0.20	1.856	539	2048158	37	1122	130

Table 4.3 K-Means clustering results, for similar numbers of clusters to DDC, on Household Power dataset.

Clusters (DDC Clusters)	Time per Sample ($\times 10^{-6}$s)	Samples / Second ($\times 10^3$)	Clusters (DDC + DDC Outlier Clusters)	Time / Sample ($\times 10^{-6}$s)	Samples / Second ($\times 10^3$)
6	1.099	909	23	4.859	205
6	1.554	643	22	4.341	230
7	1.371	729	31	7.102	140
8	2.087	479	34	8.330	120
15	4.908	204	54	2.841	35
18	3.899	256	70	2.802	35
19	4.815	208	83	3.688	27
32	8.580	117	114	6.575	15
37	1.145	873	167	120	8

4.2.6 Analysis of the DDC Method

DDC is a high speed and accurate clustering technique, primarily for hyper-elliptical clustering. It demonstrates useful characteristics such as not requiring a priori knowledge of the data and expected number of clusters. The cluster radii automatically adapt to suit the data distribution improving the cluster accuracy and reducing the amount of empty data space influenced by generated clusters. It isolates and clusters separate groups of outliers aiding the identification of data that is erroneous or anomalous for different reasons.

4.3 Fully Autonomous Clustering, Data Density Based Clustering with Automatic Radii (DDCAR)

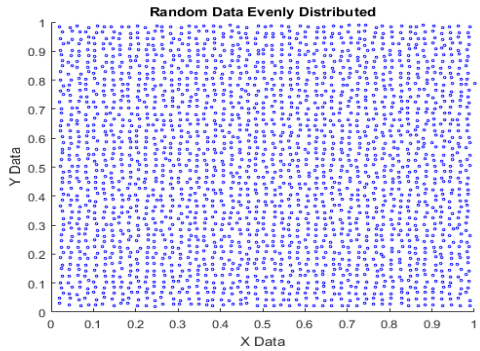
This section is based on the paper "A fully autonomous data density based clustering algorithm" presented at the 2014 IEEE symposium on Evolving and Autonomous Learning Systems at the Symposium Series on Computational Intelligence conference. [80]. The work is primarily that of the author with the aid of comments from the co-authors.

In Section 4.2 it was shown that data could be clustered using the RDE equations, 4.1, 4.2 and 4.3. It can be seen that the calculation for the density of each data sample includes a term that is inversely proportional to the distance from the global mean. If it is accepted that natural clusters must, by definition, be separated by regions of lower density, then it follows that the change in density between adjacent data samples must increase in these sparse regions. Using this principle the initial radius, r_0, required for DDC can be automatically estimated.

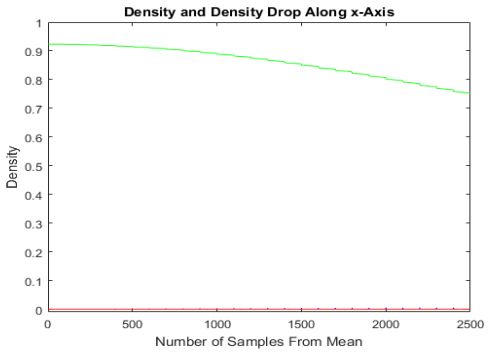
This section describes the principles and operation behind the proposed technique for this radius estimation. This technique has been termed Data Density based Clustering with Automated Radii, DDCAR [80].

4.3.1 Principles of Automatic Radius Estimation

This proposed extension to Data Density based Clustering (DDC) utilizes the clustering methodology of DDC but uses a calculation based on the data density to estimate the initial radii. Where a cluster can be defined as a group of data samples in close proximity the separation of the clusters can equally be defined as a region where the data samples are not in close proximity. The radii estimations are based on identifying the differences between in-cluster data (close proximity) and between-cluster data (sparse proximity). The density of any sample is a function of its distance from the global mean. By extension, it is also a function of its distance from the 'densest sample'. By ordering the data samples

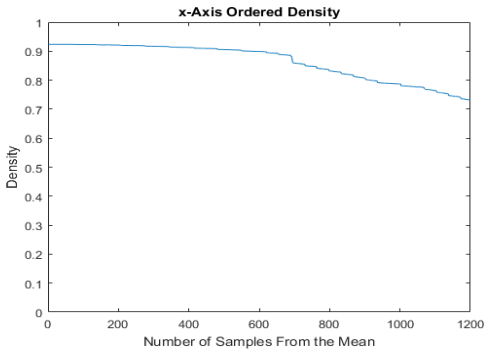


(a) Evenly Spread Data

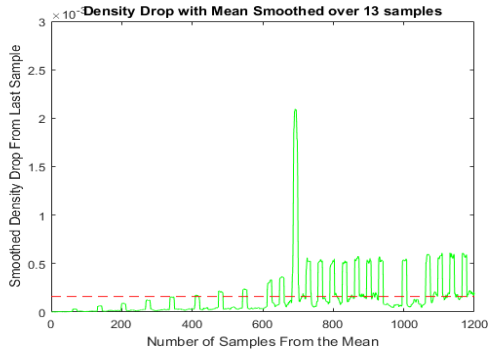


(b) Density, green line, and density drop, red line, between data

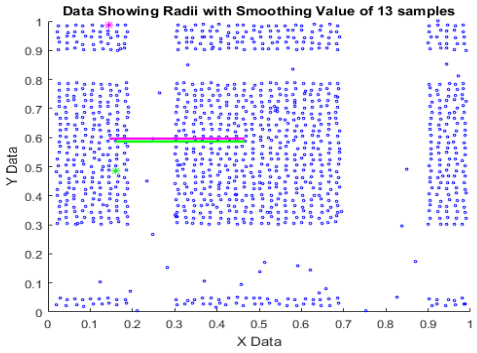
(c) Data with gaps introduced to form clusters



(d) Density of data in clusters



(e) Smoothed density drop



(f) Possible radii

Fig. 4.5 Visualizations of the DDCAR radii estimation process. Figure 4.5a shows random, even spread data and Figure 4.5b the data density and density drop between these data. Due to the approximately even spread of the data the density drop between data samples, the red line, is stable and of a low value. Figure 4.5c introduces gaps in the data to create natural clusters. The density of these data is shown in Figure 4.5d where we can see the larger drop as we leave the central cluster. To avoid the radii estimation being triggered by in-cluster variations we smooth the data density drops as shown in Figure 4.5e. Where the density drop crosses the mean we can choose either the data sample before, or after the crossing point to give two option of radii which are shown in Figure 4.5f.

in descending order of density they are, therefore, ordered by distance from the densest sample. It follows that samples within a cluster will have small density changes between neighbours and those in the regions between will have larger changes.

Consider an increasing radius from the densest sample. Each data point encountered at r_2 will have slightly lower density than those encountered at r_1 so long as we stay within the same cluster as the densest sample. As we leave the cluster the data becomes more sparse and the density drops increase as the distance between r_1 and r_2 increases. When the radius has increased to the size where it encounters samples from another cluster the density drops will decrease again.

The plots shown in Figure 4.5 will serve as a simple example to explain the process for estimating the radius of a single data dimension, the '*x-axis* ' in this case. This process has to be repeated for each data dimension to provide radii estimation of the full hyper-ellipse for DDC. Figure 4.5a shows some evenly spread data and the density and ordered density drops between the data are shown in Figure 4.5b. As expected there is a small and consistent drop in the densities. However, when gaps are introduced in the data to form natural clusters as shown in Figure 4.5c this introduces a larger, sudden drop between data as shown in Figure 4.5d. If the density drops are smoothed over n data samples then we have the smoothed graph of Figure 4.5e. The point at which the density drop crosses the mean density drop indicates two possible options to use for the initial radii in the *x-axis*, i.e. the data sample before, or after, this crossing. The radius to either data sample can be used, as shown in Figure 4.5f, or some combination of the two.

4.3.2 DDCAR Algorithm

The algorithm presented here is only concerned with the radius estimation. The radii that result are used to feed directly into the DDC algorithm described in Section 4.2. The detailed mathematical steps used for this algorithm are given in Appendix B while this Subsection provides a descriptive overview.

Algorithm 2: DDCAR Radius Estimation Algorithm

Input: {Data}, r_0
forall *Data Dimensions* **do**
  Calculate data densities using single dimension data only
  Order data by descending density
  Smooth density values
  Find first datum where density drop > mean density drop
  Calculate radius from data sample(s)
end

The radius estimation requires the full data set for the RDE [9] calculations. The RDE calculations used here are the same as Equations 4.1, 4.2 and 4.3 except that these are carried out on each dimensional component of the data individually. Thus the calculations for the mean, scalar product and density are along a single dimension only. The data samples are ordered by descending value of their densities and the mean density drop, $\overline{\delta}$, is calculated using Equation 4.10 where N is the number of data samples, D_i is the density of sample i.

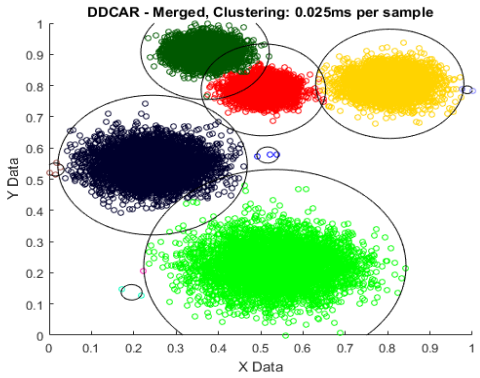
$$\overline{\delta} = \frac{1}{N}\sum_{i=2}^{N}(D_i - D_{i-1}) \tag{4.10}$$

$$\delta_{in} = \frac{1}{n}\sum_{i-n}^{i}(D_i - D_{i-1}) \qquad i > n \tag{4.11}$$

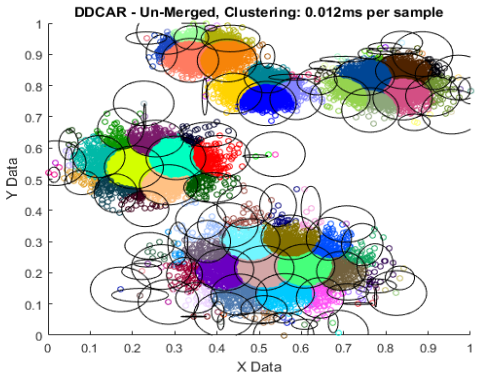
The effect of in-cluster density variations may create an early trigger for the radius estimation while the density-drop is still small. Included in the DDCAR radius estimation algorithm is a parameter n, called the 'smoothing factor', which defines the number of data density drops used to smooth out in-cluster density variations. It will be shown that the DDCAR algorithm is robust to variations in n such that it can be left unaltered and not considered a user parameter. The smoothing factor reduces the effect of small, in-cluster density drop variations by delaying any trigger until a series of larger, out-of-cluster density drops occur. The smoothed density drop value,δ_{in}, for each data sample is calculated using Equation 4.11. The first data sample for which the smoothed density drop is above the mean density drop is then found. This provides two potential values for calculating the radius, those of the distance to the data sample before or after this crossing point. Here the greater of the two distances was used. This process is repeated for each data dimension and the vector of radii produced is then used to initiate the DDC algorithm and cluster as normal.

4.3.3 Clustering of Synthetic Data Sets Using DDCAR

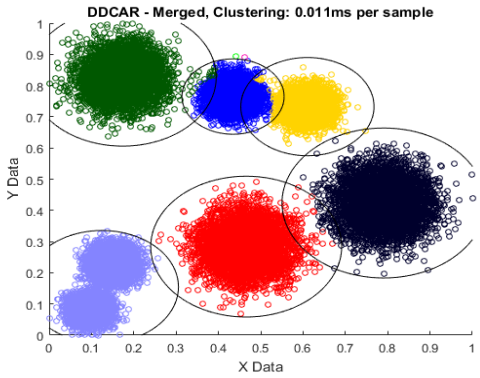
The prime focus of testing DDCAR is the comparison of clustering results with DDC supplied with manually selected radii. The results shown in Figure 4.6 indicate that the radii estimation are reasonable. The algorithm overall produces better results in the case of normally distributed data and the merge function is the root cause of most of the errors, merging natural clusters that are in extremely close proximity. Overall, the results shown in Table 4.4 show that the clusters produced without merging are of good quality. The main limitation is that of the number of clusters that an arbitrarily shaped natural cluster may be divided into, which is to be expected given the DDC algorithm basis.



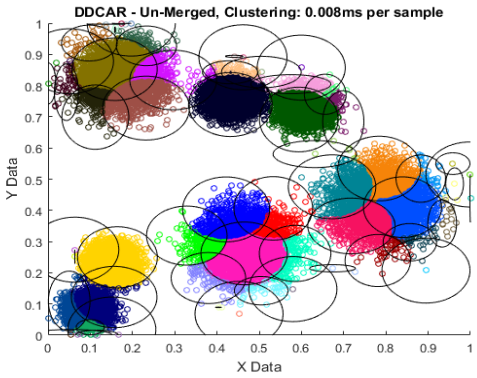
(a) Gaussian 1 Merged



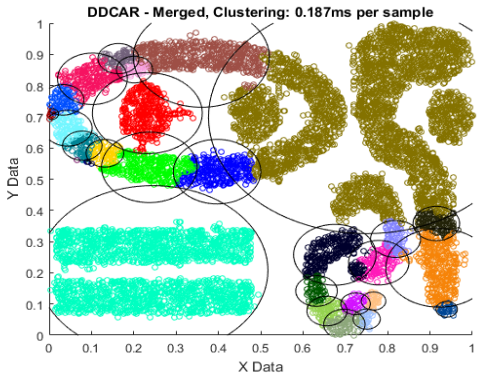
(b) Gaussian 1 Un-Merged



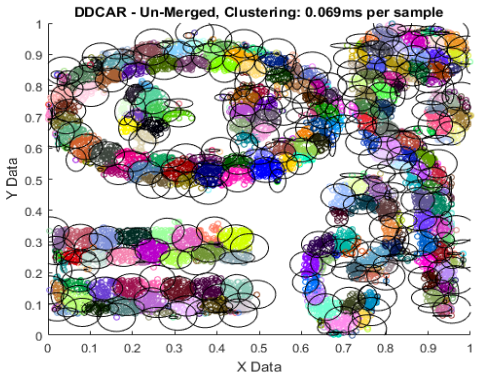
(c) Gaussian 2 Merged



(d) Gaussian 2 Un-Merged



(e) DS2 Merged

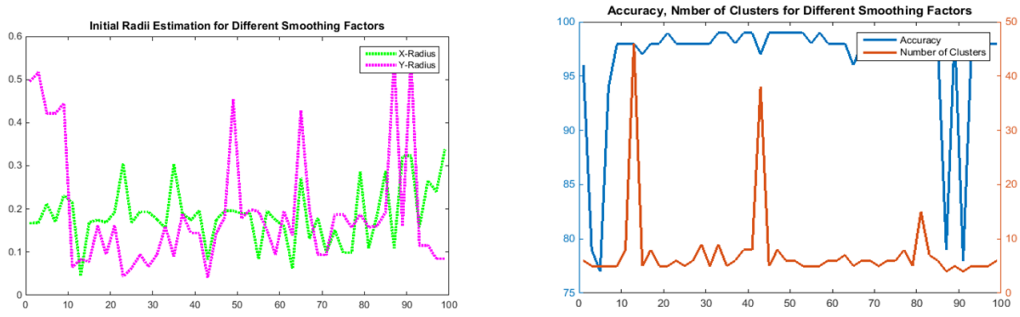


(f) DS2 Un-Merged

Fig. 4.6 Visualizations of the DDCAR test results shown in Table 4.4. Figures 4.6a and 4.6b show similar results to those of DDC with manual radii entry. Figure 4.6c is generally quite good, however the merging function has combined two natural clusters in close proximity. Figure 4.6d show the un-merged clusters have not combined the two natural cluster demonstrating that the merge function is the root cause of this error. Figure 4.6e again indicates the errors introduced by the merge function where 4.6f shows the un-merged clusters proving reasonable results, albeit with a high number of clusters.

Table 4.4 Purity, Speed and Accuracy for DDCAR.

Data Set	Technique	Purity (%)			Accuracy	Time /
		Min	Max	Mean	(%)	sample (ms)
Gaussian	DDC Merged	99.79	100	99.95	99.89	0.025
	DDC Un-Merged	50	100	99.08	99.89	0.012
Gaussian2	DDC Merged	50	100	93.64	85.59	0.011
	DDC Un-Merged	85.71	100	99.54	99.78	0.007
DS2	DDC Merged	80.1	100	95.49	69.37	0.187
	DDC Un-Merged	50	100	98.69	99.23	0.069



(a) Gaussian 1 Merged

(b) Gaussian 1 Un-Merged

Fig. 4.7 Testing the effects of the smoothing factor on DDCAR. Figure 4.7a shows how the estimated radii are stable across a wide range of smoothing factor. Figure 4.7b shows the resulting accuracy of the clustering.

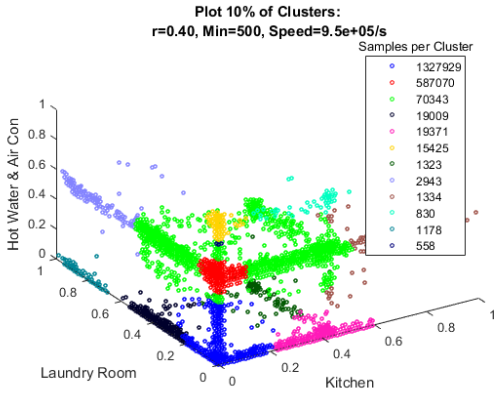
Increasing the value of n has a significant effect on the cluster accuracy up to values around 9 and from values of 13 to 90 is consistently over 97%. Although there may be a variation in the estimated radius the nature of the DDC algorithm, and its robustness to initial radii variations, is such that the final clustering results show minimal variation. This is illustrated in Figure 4.7 where the accuracy of the results remains constant. There is an unexplained peak in cluster numbers, as indicated however, overall, the technique is robust to variations in the smoothing factor indicating that this is not a parameter that needs to be altered between different data sets and so DDCAR can still be considered to be parameter free.

These results are indicative that such a route of automated radii estimation is worth pursuing, particularly in cases where hyper-elliptical clusters are suitable.

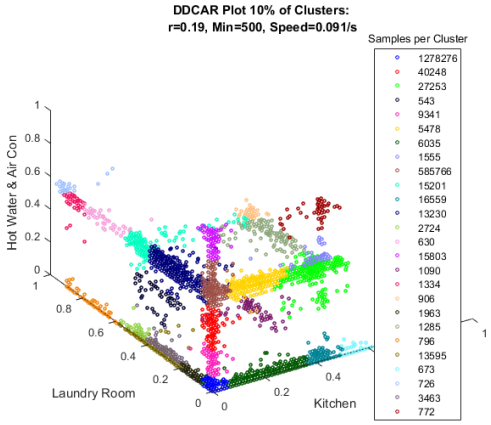
4.3.4 Comparison of DDCAR and DDC on Household Power Usage Data

To confirm the efficacy of the automated radii estimation of DDCAR the results of the clustering for the Household Power data set are compared with those for DDC. Plots of the clusters are shown in Figure 4.8. It is not expected that the results should be identical - DDC has user defined radii, whereas DDCAR generates data-based radii - however it serves to give an indication as to whether the clusters produced by DDCAR are reasonable.

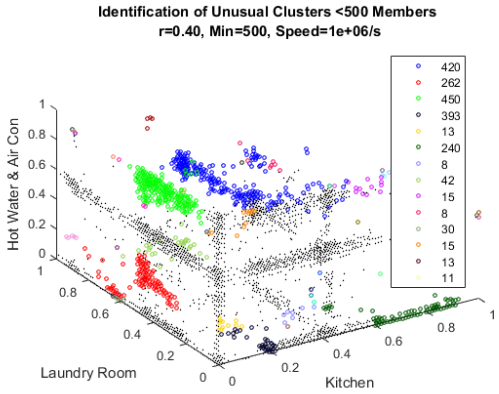
Plots of the clustering results for both DDC and DDCAR are shown in Figure 4.8 and initial visual examination suggests the results are reasonable and the clusters produced by DDCAR are subsets of each of the clusters produced by DDC. This would suggests that the density variations within the data are such that the natural clusters may actually be smaller than those originally produced by DDC. If this is the case, then the clusters produced by DDCAR should have high purity and accuracy when compared with the DDC results, i.e. if every DDCAR cluster is a genuine subset of a DDC cluster. The cluster purity details are given in Table 4.5, and, in summary, the average purity is 95.09% and the accuracy is 99.27%. Thus we can confidently state that the DDCAR results are valid. It could even be argued that the clusters produced by DDCAR may be more accurate and appropriate than those of DDC, primarily because the data-driven nature of DDCAR has divided the DDC clusters at regions of low density.



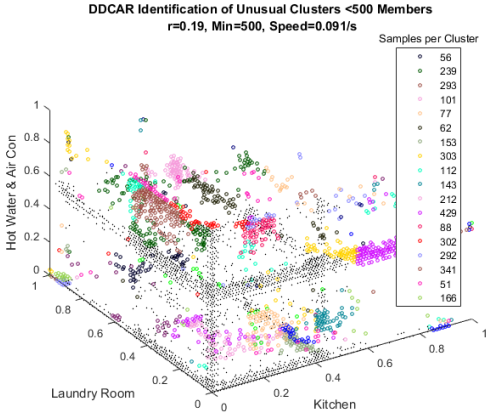
(a) DDC Results



(b) DDCAR Results



(c) DDC Outliers

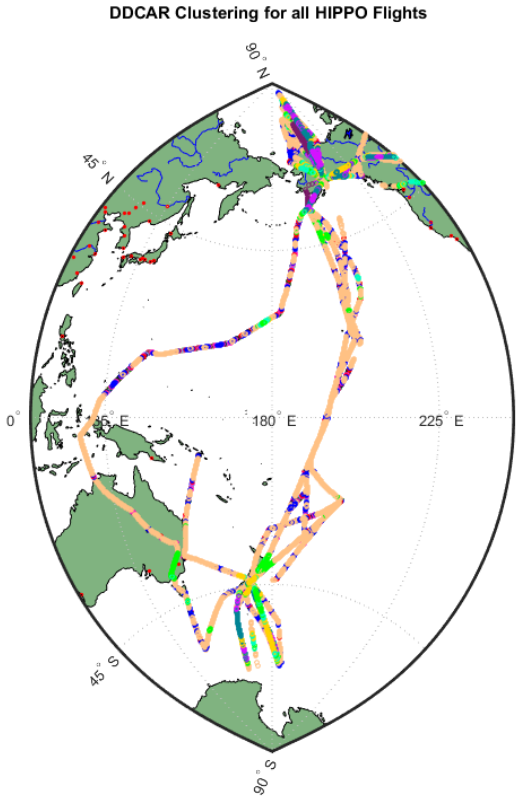


(d) DDCAR Outliers

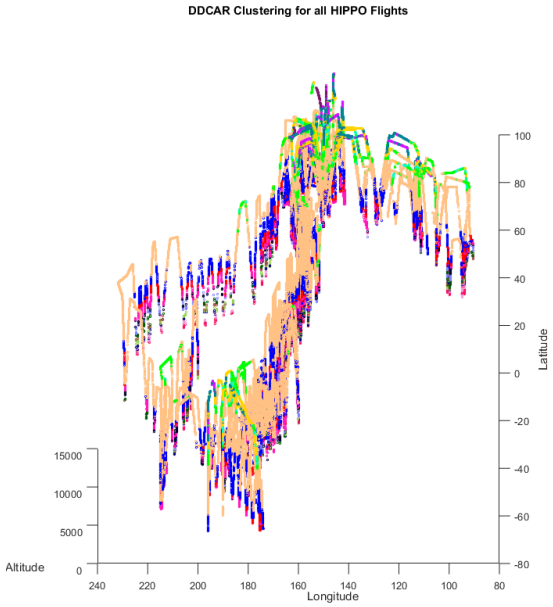
Fig. 4.8 Testing the effects of the smoothing factor on DDCAR. Figure 4.7a shows how the estimated radii are fairly stable across a wide range of smoothing factor. Figure 4.7b shows the resulting accuracy of the clustering.

Table 4.5 Comparison of DDCAR Results with DDC.

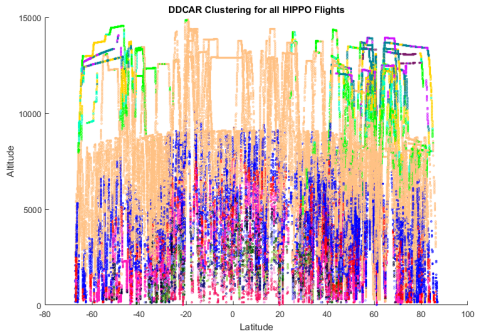
DDCAR Cluster	Number of Samples	DDC Dominant Cluster	Purity (%)
1	1278276	1	100.00
2	34016	1	84.52
3	27117	3	99.50
4	365	3	90.35
5	9341	1	100.00
6	4598	3	83.94
7	3455	1	61.16
8	1555	3	100.00
9	583841	2	99.67
10	15051	3	99.01
11	16559	5	100.00
12	11442	3	86.49
13	2708	4	100.00
14	630	8	100.00
15	15150	6	95.92
16	1090	7	100.00
17	1334	8	100.00
18	795	3	100.00
19	1963	1	100.00
20	1285	3	100.00
21	700	11	98.59
22	13354	4	98.23
23	513	5	100.00
24	726	8	100.00
25	2881	4	83.39
26	706	10	91.69



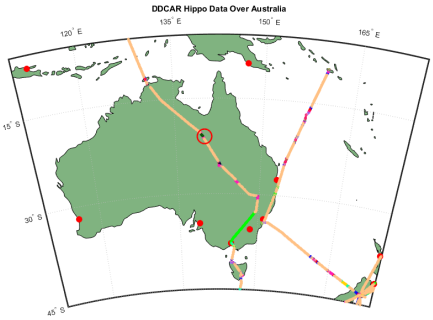
(a) HIPPO Flight Paths



(b) DDCAR Clustering across all flights.



(c) North South profile of DDCAR



(d) Close up of flight path crossing Australia

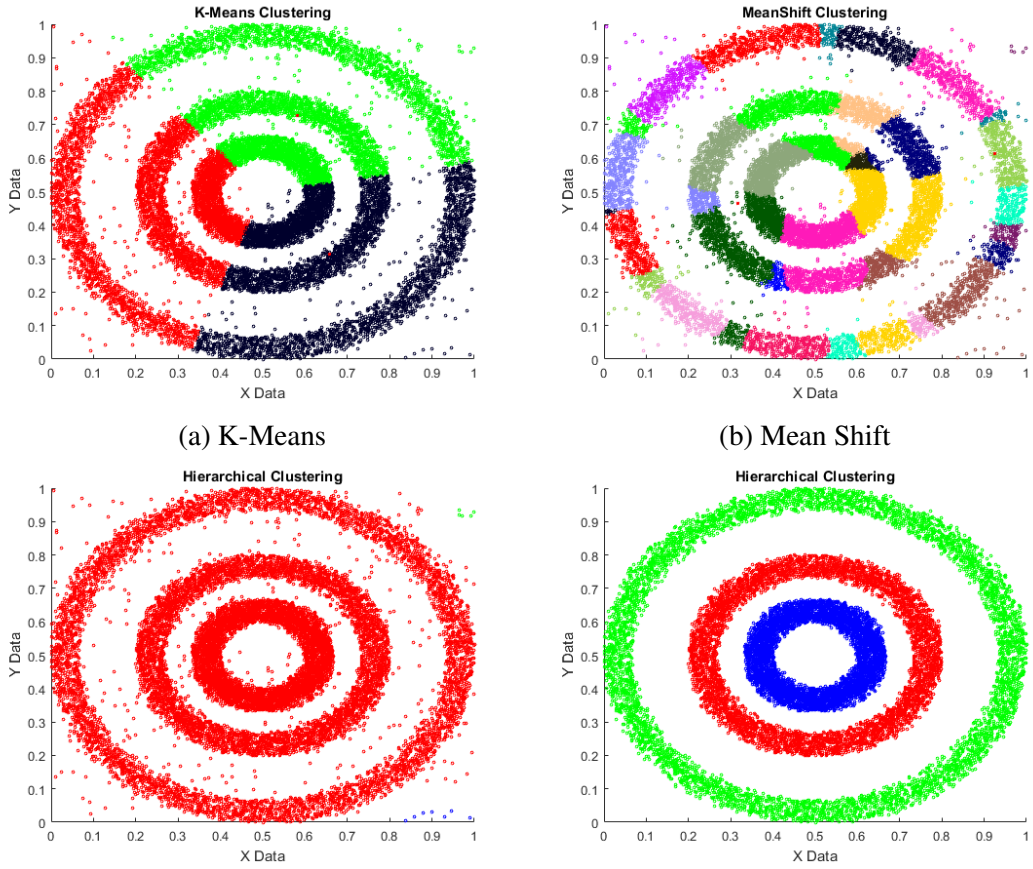
Fig. 4.9 Images relating to Subsection 4.3.5. Figure 4.9a shows the flight path overviews on a world map, pole to pole between 100^o and 230^o of longitude. Figure 4.9b is a 3D plot of the cluster results only. Figure 4.9c shows the cross sectional view of the cluster results looking east to west, i.e. the north, south profile with south to the left. This illustrates the variation in altitude at which the clustering identifies the different atmospheric regions. Figure 4.9d show a close up view of the flight over Australia where the cluster results identify pollution from urban conurbations and, circled, the large mining complex at Mount Isa.

4.3.5 DDCAR Analysis of HIPPO Data and Autonomous Identification of Australian Mining Complex

The HIAPER Pole to Pole Observation (HIPPO) was a project over 3 years, ending in 2011. the goal of the project was to monitor various atmospheric chemical species with a view to understanding the global carbon cycle, especially sources and sinks of CO_2, CH_4 and CO, and other carbon cycle related gasses. Datasets are freely available from [157] and it is a subset of the data from all flights merged as '*HIPPO Merged 10-second Meteorology, Atmospheric Chemistry, and Aerosol Data*' that is used here [158]. The full data set contains readings from up to 300 instruments and was chosen to test the speed of DDCAR across large datasets. However, a small subset was chosen, to satisfy the memory capacity of the host PC, using only ozone, O_3 and water vapour, H_2O to create the results presented here.

The flight paths of the HIAPER aircraft is shown in Figure 4.9a, the flight traversing the farthest to the west being of particular interest. It is a feature of Earth's troposphere that as the altitude increases O_3 increases and H_2O decreases. In this region, with the gradual variations in O_3 and H_2O combined with regular spaced sampling there is little change and we see single clusters covering a large range of altitude. The results do show consistency in high, medium and low altitude data values being clustered together. However, at the boundary layer differences in terrain and land use may have significant localized impact and so we see variation in the clusters at low altitude. Also, at higher altitudes, particularly crossing the Tropopause, will produce noticeable change in the data. This can be seen in Figure 4.9b however the 3D nature of the plot makes interpretation a little difficult. Figure 4.9c shows the North-South plane, i.e. looking East to West, and allows for a clearer description. The main higher altitude Troposphere is shown in orange, however there are clearly new clusters formed in two distinct high altitude regions,at both the North and South poles. Examinations of the flight altitude show that it is likely to have been crossing the tropopause at these locations as, typically, the height of the Tropopause is lower at the poles.

It is the 4th image, Figure 4.9d that most clearly demonstrates the power of fully autonomous clustering. There is a green cluster formed in the South-East, around the major cities of Sydney and Melbourne, which is to be expected. There is also an unexpected cluster formed in the middle of Australia, in the deep outback. This turns out to be in the location of Mount Isa, a small town with a population of around 22,000 residents. However, it is also home to one of the worlds largest mining sites, mining for copper, zinc, lead and silver. It has its own smelting operations and a 302MW gas powered electricity generation plant on site as well. So although only O_3 and H_2O are used in this test it is possible to autonomously discover sources of heavy pollution,



(a) K-Means

(b) Mean Shift

(c) Hierarchical

(d) Hierarchical Without Noise

Fig. 4.10 Plots of various distance based clustering techniques applied to concentric circles of data. Because all 3 clusters have a common centroid they are unable to separate the natural cluster correctly. Hierarchical clustering can separate the concentric circles with no noise present, Figure 4.10d, as the linkages to the closest data samples moves around the circle and not across the noise in the gaps.

evidence of the value of DDCAR mining data and discovering information in large datasets entirely autonomously. The changes in O_3 in particular could be attributed to nitrogen oxides and volatile organic compounds creating Ozone, or to Ozone titration by nitric oxide. Having discovered the pollution source autonomously, the information can be made available for the relevant experts in their fields to discover the causes, and solutions.

4.4 Data Density Based Clustering for Arbitrary Shapes

Clustering techniques have moved on from the simplistic, distance based measures for cluster assignment. While there are many techniques that lay claim to 'making no assumptions' about the natural cluster shape assigning data to a cluster centre by a

distance metric alone cannot avoid producing clusters of hyper-elliptical, or in really simplistic cases, hyper-spherical clusters. Simply creating a large enough ellipse to contain a concave natural cluster and claiming that the technique can function is easily dismissed. A simple and basic test is concentric, circular natural clusters such as shown in Figure 4.10. Any technique that works on a purely distance based measure will fail by either dividing the natural clusters or by merging portions of each.

Hierarchical clustering, however, connects the data differently, by merging the nearest data samples. With noise present the linkages may cross the gaps on the low density data between the circles resulting in Figure 4.10c. Without the presence of noise the merging continues around the circles as shown in Figure 4.10d. It is this type of connectivity that forms the basis of DBScan and its variants and the DDCAS technique presented here. By ignoring connections to data samples with low local density 'crossing the gap' by the noise is avoided. Both techniques employ methods of determining that the linkage should ignore the noise data due to their low local density. The need for arbitrarily shaped clusters has been well documented and many examples of such are easily found, especially dynamical systems, rotating systems, electrical systems, hysteresis loops etc.

The main technique for discovering arbitrarily shaped clusters is DBScan [53], however it is relatively slow. DBScan functions by using two user defined values ε and D, the minimum local density. Each data sample is visited and, if the number of data samples within the radius, defined by ε, is above the threshold, D, then the data sample is considered dense. All data samples within that radius are then visited and also checked. In this way a list of connected dense samples are built up and these constitute a cluster. The technique requires visiting every data sample in a dataset and this is the key reason for the slow operation. Reliance on a single minimum density value also creates difficulties dealing with clusters of varying density.

One of the key speed advantages of DDC is that it does not require visiting every data sample to perform its calculations. As described in chapter 4.2 using a smaller initial radii r_0 to initialise DDC may result in divided natural clusters. These smaller clusters can be likened to micro-clusters, mC, and can be joined to form macro-clusters, MC. With this in mind an improvement to DDC is described here which retains much of the advantages of DDC, but uses smaller r_0 to generate a larger number of small mC which can then be merged. The technique was named Data Density based Clustering for Arbitrary Shapes, DDCAS.

4.4.1 Principles and operation of DDCAS algorithm

The full mathematical steps for the algorithm are given in Appendix C and a descriptive overview is provided here.

Algorithm 3: DDCAS Algorithm

Input: {Data}, r_0, T_{min}
while $\{Data\} \neq \emptyset$ **do**
  Find global densest sample from data set and assign as temporary *mC* centre
  Assign data to temporary *mC* centre
  Remove outliers
  Set the *mC* radius to the mean distance to the assigned data
  Find local densest sample and assign as *mC* centre
  Assign data to *mC* centre
  Remove outliers $||x_i - mC_j|| > 3\sigma$
  Remove assigned data from {Data}
end
Label all *mC* with less than T_{min} samples as outliers
Merge clusters whose centre lies within another cluster

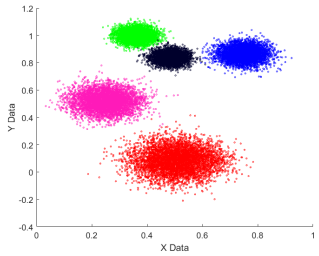
The primary functionality of DDCAS remains the same as DDC, i.e. we define a candidate *mC* centre as the densest remaining data sample among the un-clustered data. In DDCAS we have no use for the actual densities and do not calculate them, but rather acknowledge that the data sample with highest density is that closest to the mean. This data sample is assigned as the temporary *mC* centre and all the data samples within the distance r_0 are temporarily assigned to the *mC*. Data samples outside of $||x_i - mC_j|| > 3\sigma$ of the distances from the mean are discarded and the *mC* radius set to the mean distance to the clustered data. The local mean of the assigned data is then calculated and used as the final *mC* centre. Data samples within r_0 of this centre are assigned and outliers discarded.

To check whether the *mC* is part of a natural cluster, or of dense noise, the minimum support threshold value, T_{min}, is used. Any *mC* with fewer data samples than T_{min} is considered to be an outlier *mC* and will not be considered for merging into a macro-cluster *MC*. This process is repeated until all data are assigned to a *mC*. This assigns all the data to a *mC* and also retains the information as to those outliers which are similar.

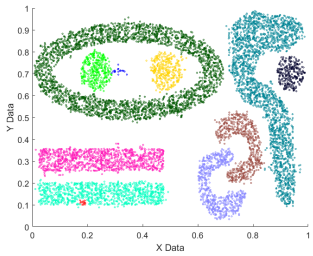
The final stage is to merge *mC* that overlap, i.e. whose centres are closer than the sum of their radii. Rather than combining all the *mC* data somehow, each *mC* is assigned a *MC* membership number.

4.4.2 Clustering of Synthetic Data Sets Using DDCAS

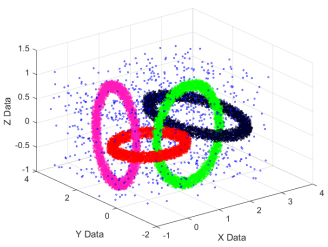
To examine the capabilities of DDCAS we compare it to the most widely known alternative technique for clustering arbitrary shapes, DBScan. We run each technique across 3 different synthetic datasets, shown in Figure 4.11a to 4.11c. The results are visualized



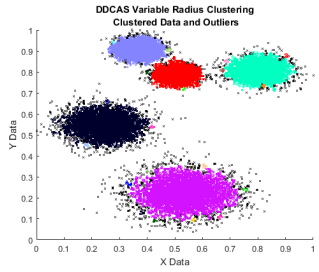
(a) Gaussian Data



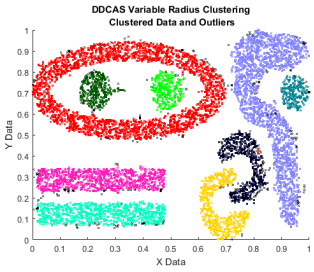
(b) DS2 Data



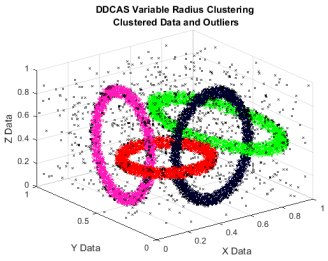
(c) Chain link data



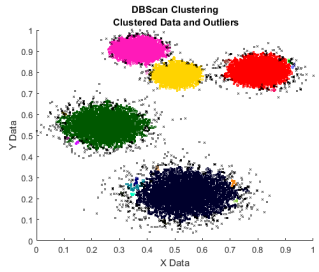
(d) DDCAS Gaussian Clusters



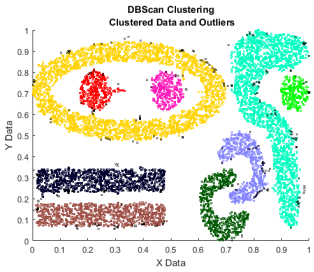
(e) DDCAS DS2 Clusters



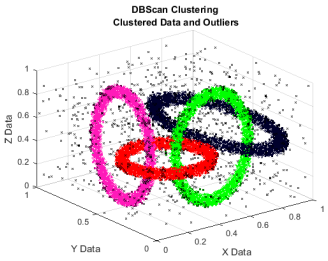
(f) DDCAS Chain Link Clusters



(g) DBScan Gaussian Clusters



(h) DBSCan DS2 Clusters

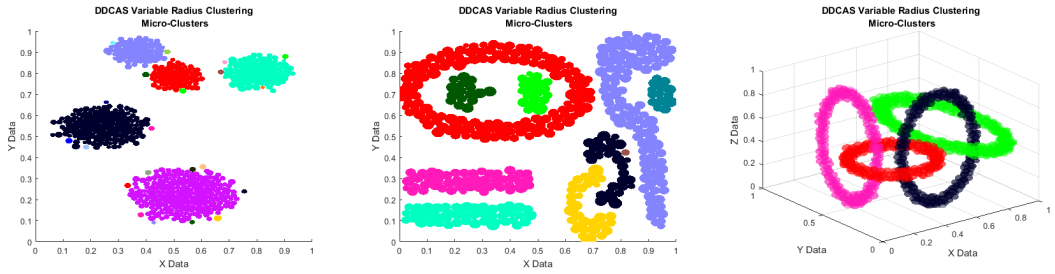


(i) DBScan Chain Link Clusters

Fig. 4.11 Image for the DDCAS comparisons with DBScan. Figures 4.11a to 4.11c shows the raw data sets coloured by class. Figures 4.11f to 4.11f show the clustering results for DDCAS while Figures 4.11g to 4.11i show the results for DBScan. For a detailed analysis see Table 4.6.

Table 4.6 Purity, Speed and Accuracy comparisons between DDCAS and DBScan.

Data Set	Technique	Purity (%)			Accuracy	Time / sample
		Min	Max	Mean	(%)	(ms)
Gaussian	DDCAS	91.67	100	99.97	99.98	0.04
	DBScan	99.96	100	100	99.99	0.52
DS2	DDCAS	94.88	100	99.35	99.66	0.05
	DBScan	95.22	100	99.3	99.63	0.27
Chain Link	DDCAS	100	100	100	100	0.10
	DBScan	100	100	100	100	0.53



(a) Gaussian *mC* (b) DS2 *mC* (c) Chain link *mC*

Fig. 4.12 Image for the DDCAS *mC* plots. The Figures show plots of the *mC* only which clearly summarises the data locations with less information.

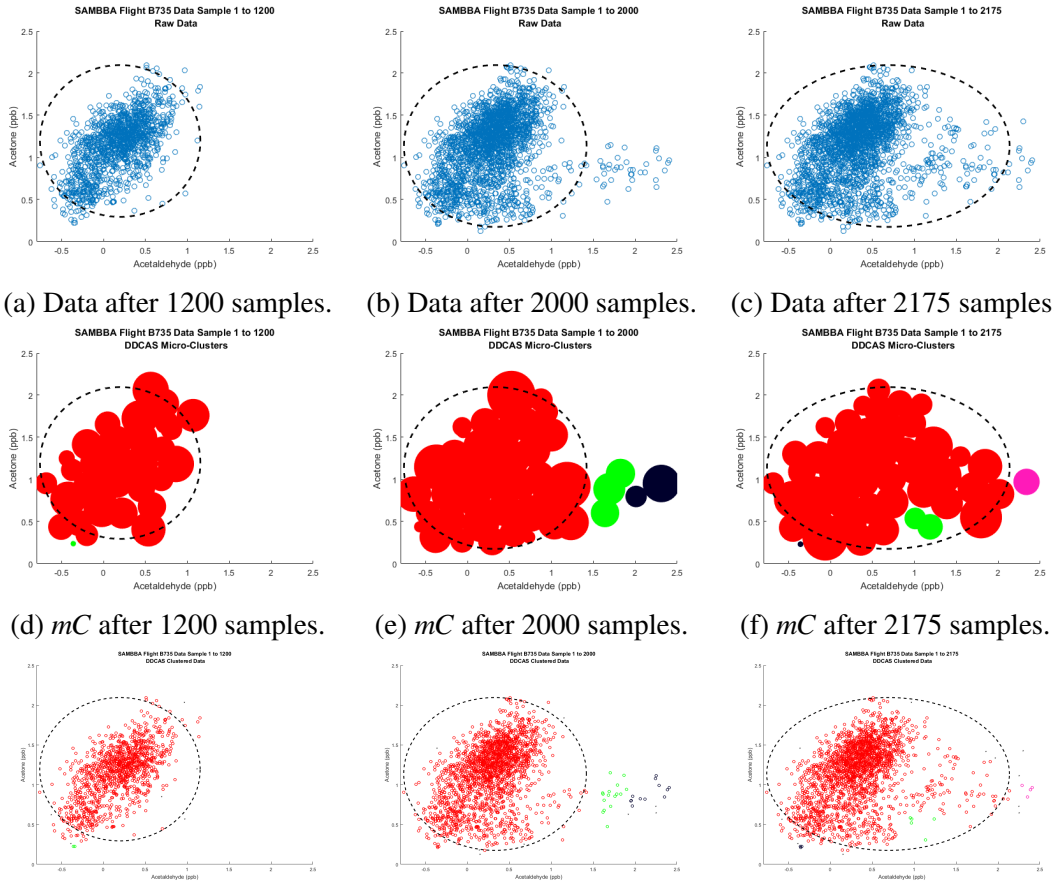
in Figure 4.11d to 4.11f for DDCAS and Figures 4.11g to 4.11i for DBScan. Detailed comparison of the results are given in Table 4.6.

The quality of the clustering results is similar across all 3 datasets for both techniques, with both techniques being marginally better in some cases. It is the speed where the greatest difference appears with DDCAS being considerably faster in all cases. It is also of significant interest that the parameters were much easier and more intuitive to adjust for DDCAS than DBScan. With DDCAS, if the *Initial Radius* is too large, or too small, then a plot of the micro-clusters indicates the source of the errors, whereas DBScan provides no indication of why the results are not as expected.

The display of DDCAS *mC* have additional uses. We show plots of the *mC* resulting from each of the previous tests in Figure 4.12. Here we can see the regions in data space that contain the data. With each *mC* potentially representing a large number of data samples these plots can provide rapid visualization of data compared to plotting every data sample.

4.4.3 Identification of Anomalies in Atmospheric Data Using DDCAS

To test the usefulness of DDCAS in a real data environment it was tested on data known to have anomalies of particular interest. The data comes from flight B735 of the South American Biomass Burning Analysis (SAMBBA) [110] data gathering campaign. The flight showed some unusual behaviour in O_3 and here we investigate other chemical species around the same time as these anomalies. The two chemicals of particular interest here are Acetaldehyde and Acetone. The nature of the data at three different times during the flights is shown in Figures 4.13a, 4.13b and 4.13c. It can be seen that the data is typically contained within a clear data region for the majority of the early flight. After approximately 1,400 data samples however, higher readings of Acetaldehyde start to occur without a matching change in Acetone. Dashed black ellipses superimposed on the



(a) Data after 1200 samples. (b) Data after 2000 samples. (c) Data after 2175 samples.

(d) *mC* after 1200 samples. (e) *mC* after 2000 samples. (f) *mC* after 2175 samples.

(g) Clustered data after 1200 samples. (h) Clustered data after 2000 samples. (i) Clustered data after 2175 samples.

Fig. 4.13 Results of DDCAS clustering at 3 different times during the SAMBBA B735 flight. The dashed black ellipse indicates the bounding region of the red cluster were it grouped by an elliptical distance based technique. By the third time period in Figures 4.13a, 4.13e and 4.13i the green anomalous data would not have been visible were it not for the arbitrarily shaped cluster definitions of DDCAS.

images indicate how a hyper-elliptical, distance based clustering technique may adapt to include these data and, indeed, leave out the high values as separated anomalies initially. Figures 4.13f and 4.13i indicate the values of arbitrarily shaped clusters produced by DDCAS. The green cluster of data that is slightly anomalous again from the main cluster would not show in a hyper-elliptical technique, but is shown by DDCAS. It is also seen that the anomalous data that initially appeared to be 'growing' across from the lower right of the original cluster now appears to be extending downwards from the top right of the initial cluster. This demonstrates how arbitrarily shaped clustering techniques can provide additional insight into data that may not otherwise be available.

It is not the purpose of this thesis to attach meaning and atmospheric chemistry analysis to these results. These data do allow a demonstration of the value of arbitrarily shaped clusters in revealing information that may not be available from other techniques and also demonstrates that DDCAS is capable of functioning on atmospheric science data.

4.5 Summary of Proposed Offline Clustering Techniques

This chapter has described the development of a number of offline clustering techniques. This section will summarise the techniques' capabilities and limitations and describe how the techniques fit in to the overall picture of solving the atmospheric science challenges formulated in chapter 2.4 and the solutions proposed in chapter 3, Section 3.2 and summarised in Table 4.7. Not all of the techniques developed are part of the proposed solutions, but they were part of the path to developing those that are.

To clarify the goals of the research into offline clustering techniques Table 4.7 shows how the offline clustering algorithm DDCAS will be used, and how, in the RASCAL software proposed in chapter 6. Absent from the table are DDC and DDCAR; DDC is the first step towards DDCAS and DDCAR was a branch from the main thrust of the research to investigate the potential for parameter-free clustering. It is seen that DDCAS provides a solution for many of the offline challenges, the exception being challenge 4, where temporal separation of the data is required. This could be achieved by windowing the data but a better solution is preferred for real time, discrete temporal differentiation.

DDC provides a fast, density based technique for finding a suitable number of clusters in a dataset. It requires no prior knowledge of the number of clusters, or of the relative densities of the natural clusters or the data between them. It is robust to a wide variation in initial cluster radii due to the adaptive nature of the radii during the clustering process. However, being a distance based clustering technique (density is distance based) the

Table 4.7 Summary of offline clustering techniques used to meet the defined atmospheric science challenges.

Challenge	Online	Offline	On \Offline	Arbitrary Shapes	Technique
1	Y	Y		Y	
2	Y		Y	Y	DDCAS
3	Y		Y	Y	
4	Y	Y		Y	
5		Y		Y	DDCAS
6	Y		Y	Y	
7		Y		Y	DDCAS
8		Y	Y	Y	DDCAS

cluster shapes are limited to hyper-elliptical clusters, relative to the distance measure used. Within this limitation it is comparable to any other techniques in terms of speed and accuracy. The desire for arbitrary shaped clusters means that DDC itself is unsuitable for meeting the atmospheric science challenges, however, it is an important first step in the development of DDCAS.

DDCAR uses a similar density measure to DDC to provide an initial estimation of the cluster radii. Although not fully explored the work demonstrated here indicates a robust and fully autonomous clustering process. Although still subject to the same limitations as DDC in the form presented here the ability to cluster data with no prior knowledge, or user input, is a marked step forward in machine learning. DDCAR was a side-track from the main algorithm development and so plays no part in meeting the atmospheric science challenges. The possibility of developing a fully autonomous clustering technique meant that this was an important piece of work for general application.

DDCAS uses a similar technique to DDC to discover clusters. Utilising a small initial radius divides natural clusters into many small micro-clusters. Joining these micro-cluster results in macro-cluster of arbitrary shape. The limitations of the technique are:

1. the initial radius should be no larger than the minimum gap between macro-clusters. If the radius is too large, then a micro-cluster may span the gap, erroneously joining macro-clusters. In many case this is intuitive, a user will have an idea of how separate data should be to be considered part of a separate cluster.

2. There must be no 'dense paths of noise' between macro-clusters, i.e. if there is even a small chain of noise above the support threshold joining two macro-cluster then they will be merged. This is actually a feature of the technique, and all other 'density-linked' types of clustering, but it does suggest that there will be

circumstances under which it is not suitable for use, e.g. where background noise is of a similar local density to the natural clusters.

DDCAS has an application to the atmospheric science solutions. Later, in chapter 6.7 results of DDCAS will be compared with those of the online techniques, CODAS, chapter 5.2, and CEDAS, chapter 5.3. The technique is able to re-produce the online clustering results of CODAS and CEDAS in a fast, offline technique that allows for post-flight or post-campaign analysis with similar results.

A secondary aim of DDCAS is to allow for fast offline clustering of historical data which allows the results to be used by an online technique. This will be discussed in further detail later when the online algorithms CODAS, chapter 5.2, and CEDAS, chapter 5.3, are presented.

Chapter 5

Development and Application of Online Clustering Techniques

5.1 Overview of Clustering Requirements

This chapter presents new online clustering techniques developed to solve the atmospheric science data challenges outlined in chapter 2, summarised in table 3.1. The combination of offline (see chapter 4) and these online techniques aims to provide a suite of compatible clustering algorithms which meet these challenges and overcome the difficulties associated with previous techniques described in chapter 3.

In this chapter the techniques called Clustering of online Data into Arbitrary Shapes (CODAS) and Clustering of Evolving Data-streams into Arbitrary Shapes (CEDAS) are developed. It is important to understand the differences between what is described here as an 'online data stream' and an 'evolving data stream' and the techniques described as 'Dynamic' and 'Evolving' clustering. The terminology and reasoning behind the different terms is explored in chapter 3.3.3 and, briefly, consider:

1. **Online Data Streams** are data that fall into a set of natural clusters. They are online, because the data arrive sequentially and may arrive at any time, in any order. Thus, the information that is used to summarise the clusters, e.g. the cluster centre and radius of influence, may adjust over over time, but the clusters will remain. Clustering of this type of data stream uses **Dynamic Clustering**. The clustering is referred to as 'dynamic' because a cluster may move or adjust its size and shape.

2. **Evolving Data Streams** are data that belong to natural clusters that evolve. These are termed evolving because not only do the data arrive sequentially, as per online streams, but older data becomes no longer relevant so at different times the clusters may be in a different location, have a different centre and / or different radii

of influence or may even no longer exist or new cluster come into existence. Clustering of this type of data requires the use of **Evolving Clustering**. The clustering techniques are referred to as evolving because the clusters may appear, merge, divide, and disappear as well as change size and shape.

This chapter describes the online techniques only, the reasons behind their development and a summary of the benefits of each technique and how they fit in to the overall picture.

5.1.1 Implementation and Testing of the Developed Algorithms

The system used to develop and test the online clustering algorithms is identical to that used for the offline system. Details of the system can be found in Section 4.1.1.

5.2 Development of Clustering for Online Non-Evolving Data Stream (CODAS)

This section is based on the paper "A new online clustering approach for data in arbitrary shaped clusters" presented at the 2015 IEEE 2nd International Conference on Cybernetics. [81]. The work is primarily that of the author with the aid of comments from the co-authors.

5.2.1 Reasons for Developing CODAS

In modern times there has been an ever increasing number of situations providing streams of data. Data streams may be defined as "a stream of data samples arriving in a time dependent manner in an unpredictable order". The need to make sense of the data in real time and in an adaptable real-time environment requires different techniques in data analysis from offline data. Not only are offline methods unsuitable for data streams, storage of the large volumes of data created by these streams is impractical. For the purposes of this thesis the definition of 'online clustering' shall be "clustering of data from data streams such that the cluster information is continually updated as new data arrives and such that past data is not required and can be discarded or archived". In Clustering of Online Data-streams into Arbitrary Shapes (CODAS) these concerns are addressed by dynamically adjusting the micro-clusters as new data are presented and by removing the need to store the data.

Alternative online data stream clustering techniques such as ELM [47] and DEC [15] provide real time dynamic clustering of data streams. Both of these techniques

operate on data streams in real time but are limited to hyper-ellipsoidal cluster shapes. The basis for ELM is to store the local mean as a cluster centre and to adjust the cluster centre and radii as more data arrives. DEC maintains a list of core and non-core clusters defined by the weight of the cluster. The weight decays over time or is increased as new data samples join the cluster. In this way core clusters may decay to non-core, non-core clusters may disappear or increase their weight to become core clusters or new, non-core, clusters may be created. In both techniques the clusters created are hyper-ellipsoidal. In the case of concave cluster shapes DEC may create many smaller hyper-ellipsoidal clusters or one large cluster encapsulating all the data.

SPARCL [30], Chameleon [89] and DBScan [26] are all techniques for clustering arbitrary shapes offline. Sparcl utilises a two layer approach whereby k-means [105] clustering is used to create a large number of micro-cluster centres. These micro-cluster centres are then further clustered using a hierarchical approach to join these micro-clusters. Chameleon and DBScan are techniques that successfully cluster arbitrary shapes however both work offline and so require the full data set. An incremental version of DBScan [26] was proposed which allows for incremental modification of the dataset. However after each increment the micro-cluster connections are made or broken according to the changes and so the whole dataset is required to be available for each increment.

A method known as DenStream was proposed in [26], based on a previous CluStream algorithm [3]. A set of core- and potential-micro clusters are maintained on-line, however the second stage creation of macro-cluster is offline. Each micro-cluster is created from a stored set of data with a decaying weight. By decaying the data samples those with a weight below a threshold are discarded and the memory requirement is limited somewhat, however the need to keep some sub-section of the data in memory blurs the definition of online clustering. While the original CluStream used k-means clustering for the second stage and so produced hyper-elliptical clusters, DenStream utilizes DBScan so producing arbitrarily shaped clusters. However, as seen in chapter 4 DBScan is significantly slower than K-Means. As a result, while the micro-clusters are maintained in an on-line fashion the process of combining the micro-clusters into final clusters is an off-line approach carried out on demand.

5.2.2 Principles of the CODAS Algorithm

Many clustering techniques for arbitrary shapes designate data samples as 'core' or 'non-core'. However, this requires storage of the data samples and ever increasing storage capacity which is to be avoided in on-line clustering. CODAS stores only the information

related to the micro-clusters and each micro-cluster has a 'core' and 'non-core' region, although to distinguish them the terms 'kernel' and 'shell' regions are used here.

In general, CODAS is a data driven approach to divide the data space into kernel and shell regions. Each micro-cluster consists of a shell region of radius r_0 and a kernel region being $0.5r_0$. This defines the shells as being the edge region of a cluster whereas the kernels form the main body of a cluster. Any micro-cluster above a given density threshold is considered for macro-cluster membership while those below the threshold are considered outliers. Micro-clusters with no intersections also form macro-clusters. Micro-clusters with kernel regions that intersect another micro-cluster shell region form a single, larger macro-cluster. Shell regions are considered to be edges of macro-clusters.

New data from the data stream will fall into one of 3 regions:

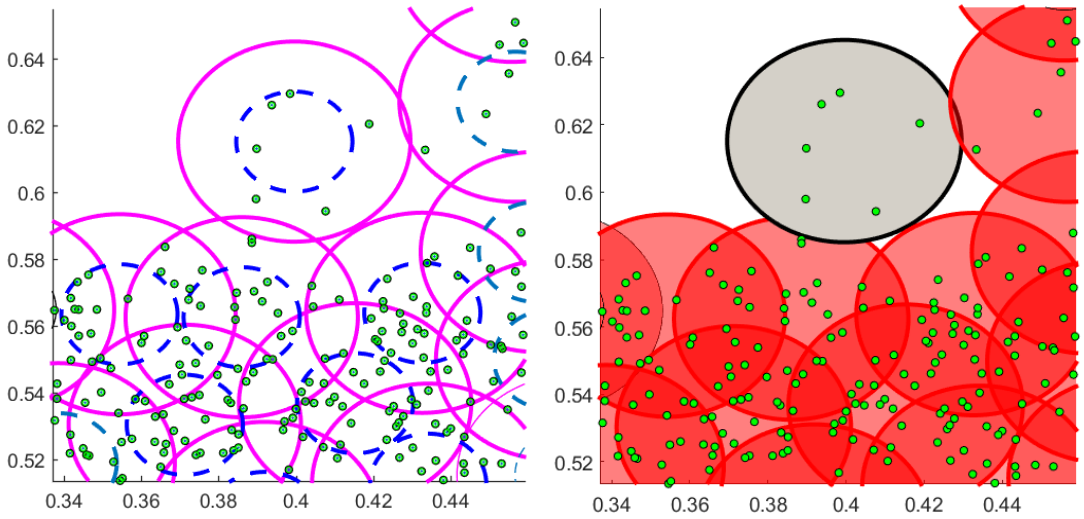
1. empty space where it will form a new, non-core-micro-cluster, i.e. not populous enough to be considered for merging.

2. micro-cluster shell region where it will be assigned to the cluster, the cluster count updated and the micro-cluster centre recursively updated to the mean of its samples. By only moving the micro-cluster centre if the data is in the outer shell this prevents a single micro-cluster following drifting data indefinitely, stopping the movement when the data fills the micro-cluster. (Restricting the centre update to when data samples arrive on the kernel and not the shell has the same affect.)

3. micro-cluster kernel region where it will be assigned to the micro-cluster and the cluster count updated

The micro-cluster that has been modified, or created, by this process is then checked to see if the local density is above the threshold. If it is, then it is checked for new intersections with other micro-clusters. If new intersections have been made then all the linked micro-clusters are assigned to the same macro-cluster. If a micro-cluster centre has been adjusted to the extent that an intersection with another micro-cluster is lost then the separated micro-clusters are also re-checked for macro-cluster membership. This maintains arbitrarily shaped data space regions of macro-clusters online. With this approach at any given time a data sample can be checked for its macro-cluster membership, any new sample is immediately clustered and outliers are identified as members of outlier-micro-clusters.

The second stage combines micro-clusters that overlap into macro-clusters. In this way, arbitrary shaped macro-clusters can be produced. To simplify the calculations required for joining the micro-clusters they are limited to hyper-spheres. Thus, they overlap if the sum of the radii is greater than the distance between the centres. In fact, not all overlapping clusters are combined, but only those for which the micro-cluster



(a) Micro-cluster kernel and shell regions (b) Combined micro-clusters

Fig. 5.1 Illustration of kernel micro-cluster regions showing 5.1a micro-cluster radius in magenta and, micro-cluster kernel radius in blue 5.1b micro-clusters combined to the macro-clusters, the grey shaded micro-cluster kernel did not overlap another micro-cluster and so is not included in the macro-cluster.

kernel intersects another micro-cluster shell. This is similar in principle, and practice, to the merging for DDCAS. Section 4.4.1 provides a description and visualization of the reasons for this and so it is not repeated here. To summarise, consider a micro-cluster kernel region to be part of a macro-cluster and the outer shell region to be the edge of the macro-cluster of which it is a part. If it is found that another micro-cluster kernel is within the shell then that part of the shell is now no longer considered to be the edge. This is illustrated in Figure 5.1.

5.2.3 CODAS Algorithm Description

The mathematical description of the algorithm for CODAS is provided in Appendix D and this section provides a descriptive overview. The algorithm updates all the clustering results for a single data sample. For a data-stream this algorithm function is called on arrival of any new data sample. The algorithm is split into an initialization section and the 3 key functions which will be described separately:

1. Assigning new data to a micro-cluster
2. Updating the micro-cluster intersections
3. Updating the macro-cluster assignments

The following sub-sections describe each of these functions and use the following:

$\{C_\mu\}$ - the set of micro-cluster centres

$\{C_r\}$ - the set of micro-cluster radii

$\{C_n\}$ - the set of the number of micro-cluster members

$\{C_M\}$ -the set of the micro-cluster macro-cluster assignation

r_0 - micro-cluster radius

x - data sample

specific micro-cluster information is indicated by an index value, e.g. $\{C_\mu\}(i)$ refers to the centre of micro-cluster i.

Initialization

Algorithm 4: CODAS: Initialization

Input: $(x), r_0$
if *no micro-cluster exists* **then**
  $C_\mu(1) = x$
  $C_r(1) = r_0$
  $C_n(1) = 1$
  $C_M(1) = 1$
end

On the first function call the algorithm requires the values of r_0, the micro-cluster radius, and x, the new data sample. As there are no micro-clusters the first one is created and consists of the micro-cluster centre, C_μ, radius, C_r, number of micro-cluster members, C_n and the number of the macro-number cluster to which it belongs, C_M. The index (1) refers to the micro-cluster number.

Assign Data to Micro-Cluster

If there are micro-clusters already established then the main algorithm function is called and requires the data sample, x, micro-cluster radius, r_0 and the information for all current micro-clusters. The distance from the data sample to the nearest micro-cluster centre is calculated and, if the sample lies within that micro-cluster radius then the micro-cluster information is updated if not then a new micro-cluster is created. When updating the micro-cluster information the count of the number of data samples it contains is updated, but the cluster centre is only recursively updated to the mean if the data sample lies in the shell region. If the data lies in the core region it is considered to be well represented

Algorithm 5: CODAS: Assign Data to Micro-Custer

Input: $(x), r_o, C_\mu, C_r, C_n, C_M$
Find nearest micro-cluster centre to (x)
if *Data is within the micro-cluster* **then**
  Update micro-cluster
  if *Data is in shell region* **then**
    Recursively update micro-cluster centre
  end
else
  Create new micro-cluster
end

by the current centre and the micro-cluster centre is not changed. This has the effect of preventing the micro-cluster indefinitely following a drift in data.

Update Macro Clusters

This section of the algorithm is only required if a micro-cluster has been modified or created *and* is above the minimum threshold to be considered part of a cluster rather than noise or outliers.

Algorithm 6: CODAS: Update macro-clusters

if *a micro-cluster has changed and is above the minimum threshold* **then**
  Find all the previous intersections
  Find all current intersects
  if *Micro-cluster intersects have changed* **then**
    Find new intersections
    Update all intersection macro-cluster number
    else if *the changed micro-cluster has no intersections* **then**
      Assign new macro-cluster number
    end
  end
end
Find any orphaned micro-clusters
Assign new macro-cluster number

The algorithm compares the previous micro-cluster intersections with those of the updated micro-cluster, if there is no change then no further action is required. If there is a change then the newly intersecting micro-clusters are all assigned to the same macro-cluster. At the same time any micro-clusters that used to intersect, but no longer do, are assigned along with their intersects, to a different macro-cluster. If the change is

Table 5.1 CODAS Dimension Test Example

Sub Cluster	x	y	Dim3	Dim4	Dim5	Dim6	Dim7	Dim8
1	9.6447	-3.5968	0.6677	0.3340	0.3332	0.3332	0.3331	0.3333
1	9.6447	-3.5968	0.6674	0.3340	0.3332	0.3332	0.3331	0.3333
2	9.6447	-3.5968	0.3331	0.6673	0.3332	0.3332	0.3331	0.3333
2	9.6447	-3.5968	0.3330	0.6677	0.3332	0.3332	0.3331	0.3333

an orphaned, or newly formed micro-cluster, above the required threshold, but has no intersections, then this is assigned to its own macro-cluster.

5.2.4 CODAS Complexity and Data Dimensionality Penalty

The CODAS algorithm complexity is examined in relation to two parameters, the number of samples and number of dimensions of the data space. The number of micro-clusters is represented by N and the number of data space dimensions by D.

In the case of increasing number of samples the cluster results are available after each sample. The means a linear complexity of O(1).

For increasing dimensionality it is a key feature of CODAS that the calculations involved have low complexity. There are two key calculations. When checking the membership of the new sample to any current micro-cluster the Euclidean distance is calculated, with complexity $O(ND)$. After the sample is assigned to a micro-cluster the distances between the new or updated micro-cluster centre and all other micro-cluster centres is calculated with a complexity of $O(N)$. The resulting complexity is therefore O(2ND). This low complexity results in an algorithm that is not only fast, but has a low time penalty for increasing dimensionality.

The effect of dimensionality on CODAS is tested by taking the spiral dataset and dividing each natural cluster through a varying number of additional dimensions. To isolate the variance to that caused by dimensionality alone the same data samples are used, but are spatially separated by adjusting a single dimension to create new natural clusters across the data space.

An example is given in Table 5.1. The four data samples given were originally coincident in (x,y) and would be part of the same natural cluster. By adding 6 data dimensions of identical value this become 8 dimensional data. Adjusting the values in dimension 3 and 4, shown shaded, the data sample are now separated across these dimensions. Thus the number of micro-clusters, data samples and calculations remains

Table 5.2 Multi-Dimensional Speed Test Results

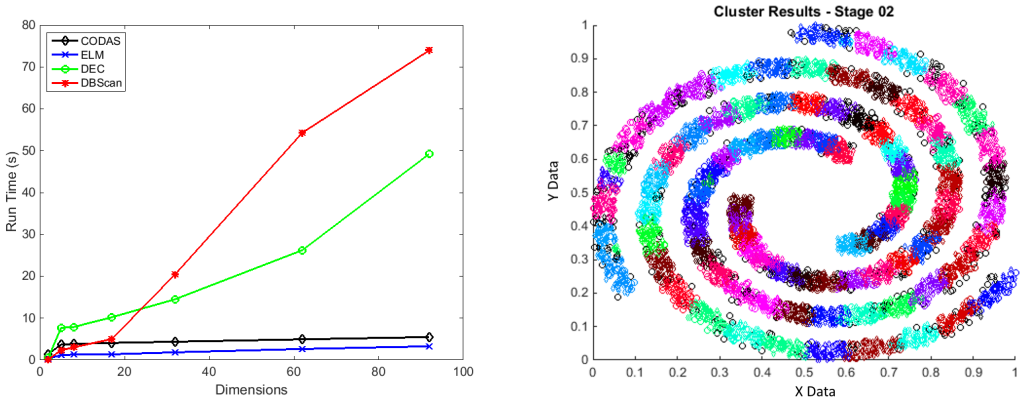
Number of Dimensions	Mean Run Time (s) CODAS	ELM	DEC	DBScan
2	1.3840	0.2558	0.4877	0.1400
5	3.6688	1.2200	7.6866	2.3300
8	3.8411	1.2800	7.8800	3.0500
17	4.0803	1.3533	10.2033	4.9966
32	4.3686	1.8366	14.5333	20.5133
62	4.9401	2.6300	26.2400	54.2966
92	5.4787	3.2733	49.2633	73.9933
Order of curve fit	$0.02x$	$0.024x$	$7e^{0.02x}$	x^4
R^2 value	0.9915	0.9941	0.998	0.9998
Projected time for 200 dim (s)	7.68	5.93	102×10^3	1.7×10^3

the same and the only change to have an affect on the algorithm operation time is the data dimensionality, and separation of the data and micro-clusters across those dimensions.

The results for these tests are given in Table 5.2. Although DBScan is not an online technique these results give an indication of the time penalty that would be expected for techniques such as DenStream that utilise DBScan as part of their algorithm. To estimate the time penalty for higher dimensionality data space a best fit line was generated, with R^2 values as shown in Figure 5.2a, and the estimated time for 200 dimensional data is given based on this projection. The time penalty for DEC and DBScan is seen to be considerable as dimensionality increases. ELM is slightly faster than CODAS, however it should be remembered that ELM produces hyper-elliptical clusters only.

5.2.5 Testing CODAS by Clustering of Synthetic Data

The CEDAS algorithm was first tested on the synthetic data sets used previously, Gaussian, Spiral, Chain and DS2. The Gaussian dataset is a relatively straightforward test, the only difficulty being the close proximity of some of the natural clusters. The Spiral and Chain datasets test the robustness to noise, with the chain dataset adding the difficulty of 3 dimensions and natural cluster that cannot be discovered by purely distance based techniques. The DS2 data set provides particularly difficult natural clusters with varying density and close proximity. The natural order of the provided data set is such



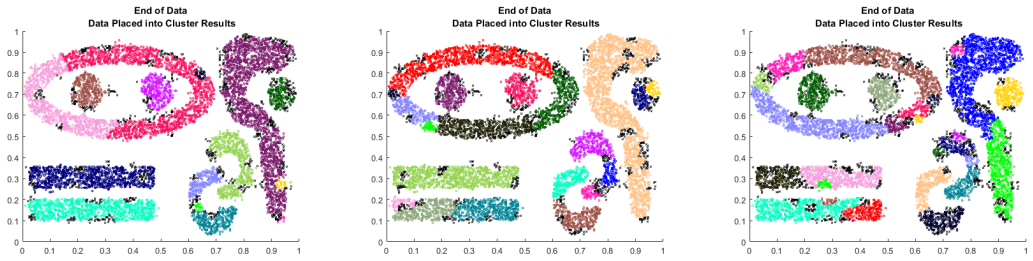
(a) Comparative times

(b) 92 Dimensional Data

Fig. 5.2 Figure 5.2a plots the run times for various techniques on higher dimensional data. Only ELM compares favourably, but it is limited to hyper-elliptical clusters only. Figure 5.2b shows the cluster results of CODAS projected back onto the x-y plane. Each coloured cluster is separated across 92 dimensions.

Table 5.3 CODAS synthetic data set test results.

	Data Set	Data Rate (samples / s)	Purity (%) Min	Max	Mean	Accuracy
CODAS	Gaussian	900	94.74	100	99.73	99.97
	Spiral	1,000	100	100	100	100
	Chain	700	100	100	100	100
	DS2	3,300	100	100	100	100
DBScan	Gaussian	n/a	99.95	100	99.99	99.98
	Spiral	n/a	87.1	90.8	88.9	89.47
	Chain	n/a	100	100	100	100
	DS2	n/a	95	100	99.3	98



(a) Randomised Data Order 1 (b) Randomised Data Order 2 (c) Randomised Data Order 3

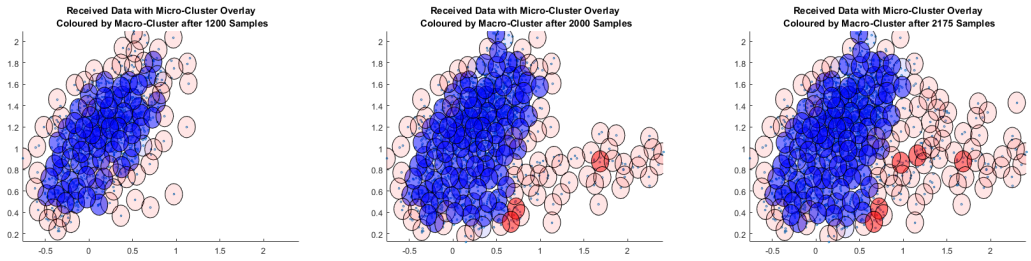
Fig. 5.3 These plots show the different clusters formed, after the same number of samples, for different, randomised, order of data. The cluster purity and accuracy are the same in all cases.

that the data appears predictably in order of each natural cluster, i.e. the data for each natural cluster appear sequentially. To fully test the algorithm and ensure it can deal with unpredictable data the data set order is randomised using Matlab's 'randperm' function. Examples of the results are presented in Table 5.3. The results show that CODAS can achieve a similar quality of results from a data stream as DBScan achieves in offline mode with the full datasets available. In the case of noisy data such as the Spiral data CODAS can outperform DBScan as it is more robust to dense noise.

CODAS is, as might be expected, order dependent. The micro-cluster creation and adjustment both depend on the data order and results may differ if the data order is randomised. Typical variations in the results after the same number of samples are shown in Figure 5.3 - the colour of the clusters is of no particular significance here. Although the clusters are forming in a different manner the purity and accuracy of the clusters formed remains constant. The order dependency has prevented some of the micro-clusters merging to form the full natural cluster, however, each macro-cluster formed is a subset of the natural cluster resulting in the high levels of purity and accuracy.

5.2.6 Visualization of Atmospheric Science Data Streams Using CODAS

To test CODAS on real data streams the algorithm is applied to the B735 flight data used for testing DDCAS in chapter 4.4.3. The data is presented to the CODAS algorithm sequentially in the order the data was captured during the data gathering mission. An initial radius of 0.3 was used and a minimum threshold for merging micro-cluster set at $T_{min} = 5$. A minimum threshold value is suggested by examining the initial data stream and estimating the rate of change of data values for typically 'normal' data. Using a value of 5 allows for data to the edge of a natural cluster, or rapidly changing data values to remain as outlier micro-clusters due to their lower local density.



(a) 1,200 Data Samples (b) 2,000 Data Samples (c) 2,175 Data Samples

Fig. 5.4 These plots show the different clusters formed, after the same number of samples, for different, randomised, order of data. The cluster purity and accuracy are the same in all cases.

The clustering has been paused after the same number of valid data samples as used for the DDCAS demonstration in chapter 4.4.3. For clarity the micro-clusters have been coloured such that the macro-cluster where typical data appears is blue and any outlier clusters that are not merged are coloured red. The transparency of each micro-cluster is inversely proportional to the number of data samples it contains. The results are shown in Figure 5.4. The blue micro-clusters have low transparency indicating many data samples in these regions. The red micro-clusters typically have higher transparency as they contain fewer data.

The plot shown in Figure 5.4a shows the results after 1,200 data samples. The main bulk of the data is in the blue macro-cluster with a few outliers on the fringes. By 2,000 data samples, Figure 5.4b, the anomalies spread out from the main data space region. By 2,175 data samples a secondary series of anomalous data 'fills in' the space to the top right of the main cluster. These results match with those obtained using DDCAS and provide a clear visual indication of the presence of anomalies as they occur.

The ability of CODAS to continually update the micro- and macro-clusters allows for anomalies to be identified immediately they occur. The delay expected from hybrid on/off-line techniques is not present. The clustering information returned by CODAS follows the same structure as DDCAS allowing the results to be interchangeable. This should lead on to the ability to cluster historical data using DDCAS followed by online clustering, as more data arrives, using CODAS. The key difference between the two is the cluster radius. DDCAS uses an adjustable radius to allow better, more detailed coverage of complex shapes whereas CODAS utilises a static radius to avoid inappropriate reduction of the micro-cluster radius, i.e. if CODAS utilised an adjustable radius there is a danger that micro-clusters will reduce in size to the extent that data originally assigned to a micro-cluster will fall outside of its influence over time.

5.2.7 Summary of the Benefits and Limitations of CODAS

CODAS is a fully online technique for clustering data streams into arbitrarily shaped clusters. By continually updating the macro-cluster assignations of the micro-clusters the clustering results are available online, an advantage over hybrid on/ offline techniques.

The natural clusters that data falls into must be somewhat fixed in nature as, although macro-clusters may adjust size, shape and position, macro-cluster changes are only achieved by the addition of micro-clusters. As there is no technique in the algorithm for reducing the relevance of older data the resulting macro-clusters have no way of following data drift or shift such as those identified and discussed in [102, 101]. As a result, once anomalous data has been identified, a repeat of similar anomalies at a later time will fall into micro-clusters that are already present and so will not be identified as anomalies. To overcome this issue a means of ageing the micro-clusters is required and this is addressed in the next Section, 5.3.

5.3 Development of Clustering for Online Evolving Data Streams (CEDAS)

This section is based on the paper "Fully online clustering of evolving data streams into arbitrarily shaped clusters" submitted to Information Sciences, July 2016. The work is primarily that of the author with the aid of comments and feedback from the co-authors.

As mentioned in the previous chapter, 5.2 there has been an increase in data availability in continuous data streams and clustering of this data has many advantages in data analysis. It is often the case that these data streams are not stationary, but evolve over time, and also that the clusters are not regular shapes but form arbitrary shapes in the data space. Previous techniques for clustering such data streams are either hybrid online / offline methods, windowed offline methods, or find only hyper-elliptical clusters.

This section presents a fully online technique for Clustering Evolving Data-streams into Arbitrary Shaped clusters (CEDAS). It is a two stage technique that is accurate, robust to noise, computationally and memory efficient, with a low time penalty as the number of data dimensions increases. The first stage of the technique produces micro-clusters and the second stage combines these micro-clusters into macro-clusters. Dimensional stability and high speed is achieved through keeping the calculations both simple and minimal using hyper-spherical micro-clusters. By maintaining a graph structure, where the micro-clusters are the nodes and the edges are its pairs with intersecting micro-clusters, the calculations required for macro-cluster maintenance are minimised. The micro-clusters themselves are described in such a way that there is no calculation required for the

kernel and shell regions and no separate definition of outlier (non-core) micro-clusters is necessary.

The abilities of the proposed technique to join and separate macro-clusters as they evolve in a fully online manner is demonstrated. There are no other fully online techniques that cluster data streams into arbitrarily shaped clusters that the author is aware of and so the technique is compared with popular online / offline hybrid alternatives for accuracy, purity and speed. The technique is then applied to real atmospheric science data streams and used to discover short term, long term and seasonal drift and the effects on anomaly detection.

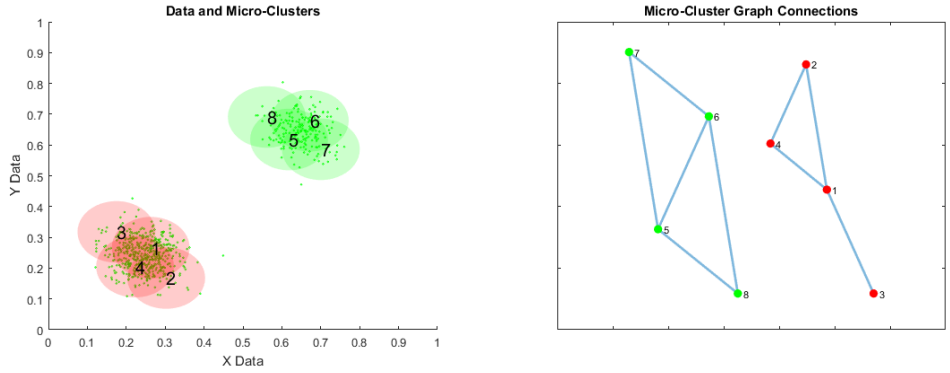
As well as having favourable computational characteristics, the technique can add analytic value over hyper-elliptical methods by characterising the cluster hyper-shape using Euclidean or fractal shape factors. Because the technique records macro-clusters as graphs, further analytic value accrues from the possibilities of characterising the order, degree, and completeness of the cluster-graphs as they evolve over time.

5.3.1 Reasons for Developing CEDAS

The reasons for developing the CEDAS algorithm are explained in Section 3.3.3 earlier. The evolution of natural cluster over time provide specific challenges that traditional online, or dynamic, clustering algorithms are unable to meet. In particular, the inability of many of these algorithms to generate arbitrarily shaped clusters, or to present these clusters in a fully online manner without windowing or a secondary offline stage, requires a new method of analysis. CEDAS is such an algorithm, operating on each data sample as it arrives and generating the arbitrarily shaped clusters immediately.

5.3.2 Principles of the CEDAS Algorithm

The CEDAS technique presented in this chapter has two distinct stages. The first adds data to current micro-clusters and adjusts their information, or creates new micro-clusters when data samples occur in un-clustered data space. The radius of the micro-clusters, r_0, is fixed and should be as small as is practical. In this newly proposed method a simple linear ageing process is used which reduces the 'energy' of a micro-cluster and allows unused micro-clusters to be removed completely. Alternative ageing processes could be used including those exponential types that may leave micro-clusters present, with insignificant energy, but allow them to be 're-born' and become relevant in the future with further data. The micro-cluster energy is fully renewed every time they receive new data but, again, other processes for the energy recovery of a micro-cluster could be used. When no data is received the micro-clusters lose some energy, gradually fading out. If



(a) Data and micro-clusters. (b) Graph structure.

Fig. 5.5 Example of the CEDAS algorithm micro-clusters and graph structure. The data together with two macro-clusters in red and green are shown in Figure 5.5a. Figure 5.5b shows the cluster graph structure with the nodes of the sub-graphs coloured according to the macro-clusters.

no data is received for a long time the micro-cluster energy will reach zero and they are are no longer recorded.

The second stage searches for overlapping micro-clusters. The micro-clusters are defined as having a kernel region $r \leq 0.5r_0$ and a shell region $r > 0.5r_0$. By only connecting those micro-clusters whose kernel regions overlap into another micro-cluster shell edge micro-clusters are automatically determined. Micro-clusters which do not have at least the user-specified local density, i.e. the minimum number of samples within the radius, remain as separate outlier micro-clusters. Each macro-cluster consists of the graph of intersecting micro-clusters; the adjacency relations for each micro-cluster are stored as a property of that micro-cluster. For convenience, the micro-clusters in adjacency relations (i.e. intersecting micro-clusters) are referred to as 'edges'. Those micro-clusters with no edges define graphs of order 1 without edges (i.e. without intersections) and constitute a macro-cluster graph by themselves. Using this graph structure reduces the calculations required to separate clusters if a micro-cluster dies and breaks a chain graph resulting in two groups of micro-clusters no longer being connected. Figure 5.5 shows a simple example. Two macro-clusters are shown with their respective micro-clusters numbered. Figure 5.5b shows the corresponding graph structure. The two sub-graphs have their nodes coloured according to the macro-cluster they represent. The edges between the nodes show which micro-clusters intersect to create the macro-cluster agglomeration.

5.3.3 CEDAS Algorithm Overview

Traditional offline clustering techniques for arbitrary shapes may categorize data samples as 'core' or 'non-core'. However, this requires storage of the data samples and ever-increasing storage capacity which is prohibitive for online clustering. CEDAS stores only the information related to the micro-clusters and a graph structure recording the micro-cluster connections.

The following terminology is defined for the CEDAS approach:

1. Cluster Graph: the structure that defines which micro-clusters join to form which macro-clusters. This is stored by recording the intersects of each micro-cluster in 'Edge', together with the appropriate macro-cluster assignation in 'Macro'.
2. Local density: the number of samples per micro-cluster
3. Macro-cluster: a cluster consisting of a number of intersecting micro-clusters.
4. Micro-cluster: a micro-cluster with a local density above the threshold.
5. Outlier-micro-cluster: a micro cluster with local density below the threshold.
6. Sample: any data point in 'D ' dimensions.
7. Threshold: the minimum number of samples within the micro-cluster radius of any sample to form a micro-cluster.

In general, CEDAS is a data-driven approach to divide the data space into kernel and shell regions based on a user defined radius, r_0. Each micro-cluster consists of a shell annulus region between radii $\frac{r_0}{2}$ and r_0 and a kernel region being $r \leq \frac{r_0}{2}$. Any micro-cluster above a given density threshold is considered for macro-cluster membership. Micro-clusters with kernel regions that intersect another micro-cluster shell region form macro-clusters. Micro-clusters above the threshold but with no intersections are also considered to be macro-clusters. Shell regions are considered to be edges of macro-clusters.

New data from the data stream will fall in to one of 3 regions:

1. empty space, where it will form a new, outlier-micro-cluster
2. a micro-cluster shell region, where it will be assigned to the cluster, the cluster count updated and the micro-cluster centre recursively updated to the mean of its samples.
3. a micro-cluster kernel region, where it will be assigned to the micro-cluster and the cluster count updated

The micro-cluster that has been modified, or created, by this process is then checked to see if the local density is above the threshold. If this is the case then this micro-cluster is checked for new intersections with other micro-clusters. If new intersections have been made then all the micro-clusters are linked and assigned to the same macro-cluster. This ensures that all linked micro-clusters have the same macro-cluster reference and maintains arbitrarily shaped data space regions of macro-clusters in a fully online manner.

With this approach at any given time a data sample can be checked for its macro-cluster membership, any new sample is immediately clustered and outliers are identified as members of outlier micro-clusters. It is the graph structure for storing the micro-cluster intersections that forms the basis of the advance from CODAS to CEDAS as this allows rapid merging and division of macro-clusters.

5.3.4 CEDAS Algorithm Description

There are 4 distinct steps for the full algorithm including initialisation and a $Step0$ is included where the user determines the parameters for the algorithm:

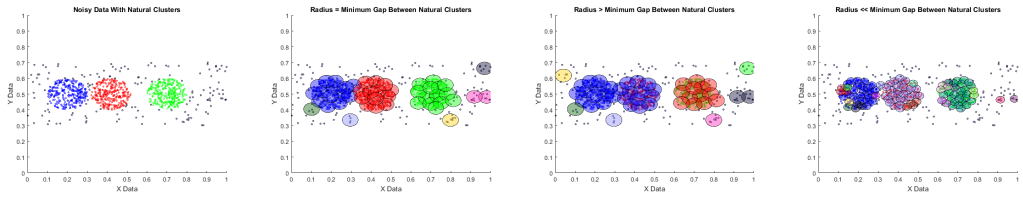
0. Parameter Selection
1. Initialization
2. Update Micro-Clusters
3. Kill Clusters
4. Update Cluster Graph

The full mathematical sequence for the algorithm is given in Appendix E and a detailed description of each of the key algorithm steps is provided here. The algorithm runs these sections sequentially for each data sample, if required. Not every section is required each time and the conditions for running each section are described.

Parameter Selection

CEDAS requires a number of parameters to function, *Decay*, *radius* and *Minimum Density Threshold*. The values for these parameters are application dependent and suitable values can be selected as follows:

1. *Decay*: the decay is directly related to the length of time over which the data is to be examined, e.g. at a sample rate of $1Hz$, to examine the data over a 28 day period the Decay would be 2,419,200, at $0.1Hz$ for 7 days, 60,480.



(a) Raw Data With Natural Clusters (b) Radius equal to the cluster gap (c) Radius Greater than the cluster gap (d) Radius Much less than the cluster gap

Fig. 5.6 Demonstration of varying CEDAS radius selection. Figure 5.6a shows raw data with noise and natural clusters. Figure 5.6b shows the cluster results with radius equal to the minimum gap between the clusters, Figure 5.6c shows the results of having a larger radius than the minimum gap and Figure 5.6d shows the effect of a much smaller radius. Thus radius is set by the user and should be less than the maximum dis-similarity data can have and still be considered a part of the same cluster.

2. *radius*: the radius of the micro-cluster is selected based on expert knowledge of the application. With any set of data there are distances between data samples. There is a maximum distance between data beyond which an expert will consider that the data belongs to a different cluster and this value is the maximum allowable radius, i.e. the radius should be set to the minimum allowable gap between macro-clusters. Using a radius below this value has little effect on the overall macro-cluster beyond compiling them from a greater number of micro-clusters and smoothing the edge of the macro-cluster. There is an effective lower limit to the radius below which it will not contain enough data samples for a micro-cluster to form. Visual examples are shown in Figure 5.6 where figure 5.6b shows successful clustering with the radius equal to the minimum gap between natural clusters, Figure 5.6c shows how increasing this value determines that two of the clusters are not different enough to be considered separate and become merged. Figure 5.6d shows how reducing the radius to half of the minimum gap has little effect on the macro-cluster results.

3. $MinimumDensityThreshold$ is required to differentiate clusters from background noise and / or outlier data. The value should be set based on expert knowledge as to the level of data required to be considered valid, natural clusters.

Initialization

This creates a structure to store the information related to each micro-cluster and takes place with the first data sample. The '$Centre$ ' defines the location of the micro-cluster in data space. '$Count$ ' stores the total number of data samples that have been allocated to the micro-cluster. The value of '$Count$ ' is recorded to allow recursive updates to the micro-cluster centre. '$Macro$ ' is a reference to the macro-cluster to which this micro-

Algorithm 7: CEDAS: Initialization

Input: x, r_0
Create micro-cluster structure containing:
$C_1(Centre) = x$
$C_1(Count) = 1$
$C_1(Macro) = 1$
$C_1(Energy) = 1$
$C_1(Edge) = 1$
Set number of micro-clusters to 1
Set modified micro-cluster number, for use updating the graph structure.

cluster belongs. The value of '*Macro* ' is the same for all micro-clusters in the '*Edge* ' list. '*Energy* ' is a value used to determine the length of time since a micro-cluster received new data. The decay algorithm reduces this value and is discussed later. '*Edge* ' is a list of intersecting micro-clusters, if a micro-cluster has no Edge list then it is a macro-cluster by itself. In graph theory terminology the micro-cluster number paired with each intersect constitutes an 'edge'of the form $\{mC_c, mC_i\}$, where the first term is the current micro-cluster and the second term is the intersecting micro-cluster.

Update Micro-Cluster

This part of the algorithm updates the micro-clusters when a new data sample arrives.

Algorithm 8: CEDAS: Update Micro-Cluster

Input: x, C, r_0
find distance to nearest micro-cluster centre, d_{min}
if $d_{min} < r_0$ **then**
  reset micro-cluster *Energy* to 1
  increment number of samples contained in micro-cluster
  if *data is within micro-cluster shell* **then**
    recursively update micro-cluster centre
  end
else
  Create new micro-cluster
end

The algorithm checks whether the new data sample belongs to any current micro-cluster. If it does not then a new micro-cluster is created. If the data sample is within a current micro-cluster then the *Energy* of that micro-cluster is re-set to 1 and the number of data samples it contains is incremented. A further check is made to find if the sample lies within the kernel or shell of the micro-cluster. If it is in the shell region then the centre of the micro-cluster is recursively updated to the mean of the data samples in the

shell. Only updating the centre if the data lies within the shell has the effect of preventing a single micro-cluster endlessly following drifting data by limiting its movement. (The same effect is also achieved by only updating the centre if the sample lies in the kernel.)

Kill Micro-Cluster

This part of the algorithm reduces the energy of the micro-clusters and removes them if the energy has fallen below zero.

Algorithm 9: CEDAS: Kill Micro-Cluster

```
Input: C, Decay
Reduce all C(Energy) by Decay
if Any C(Energy) < 0 then
    Remove micro-cluster
    Remove all edges containing the micro-cluster
    Decrement the number of micro-clusters
end
```

First all the micro-cluster energies are reduced by the decay amount. Then, if any micro-cluster energies are below zero they are removed and all edges that refer to this micro-cluster are removed and the total number of micro-clusters is reduced.

Update Micro-Cluster Graph

Algorithm 10: CEDAS: Update Graph

```
if A micro-cluster has been modified then
    if the micro-cluster edge list has changed then
        Set a new macro-cluster number throughout the graph
    end
end
if Any micro-clusters have died then
    Set new macro-number for the graphs of its edges
end
```

This section only makes any changes if either:

1. a new cluster has been created and reached the minimum density threshold
2. a cluster centre location has been modified
3. a cluster has died and been removed.

First the changes are made to any micro-cluster that has been modified by either having its centre location moved or by virtue of being a micro-cluster that has newly reached the threshold. In either case the graph edges for that micro-cluster may have changed. If the edge list has changed then the new graph has its macro-cluster number set to a new value.

The changes made by any micro-cluster that have died out are then addressed. Any micro-clusters that the dead micro-cluster used to have an edge with have their graphs updated with a new macro-cluster number.

5.3.5 Testing CEDAS by Clustering of Synthetic Data Streams

The following sections analyse the performance of the CEDAS algorithm and presents the results and discussion across a range of experiments. In Subsection 5.3.6 the ability of CEDAS to accurately deal with data drift, cluster separation, cluster merging and noise over time is validated. The speed and accuracy is then compared with alternative techniques CluStream, DenStream and MR-Stream across high dimensionality data in Subsection 5.3.7. In Subsection 5.3.8 CEDAS is compared again to these techniques with regard to complexity, processing speed, cluster quality and memory efficiency. Finally in Subsection 5.3.9 the CEDAS algorithm is applied to a real data stream from the London Air Quality monitoring system to demonstrate how evolving clustering can aid data mining of data streams containing short term drift, long term drift and short and long term anomalies.

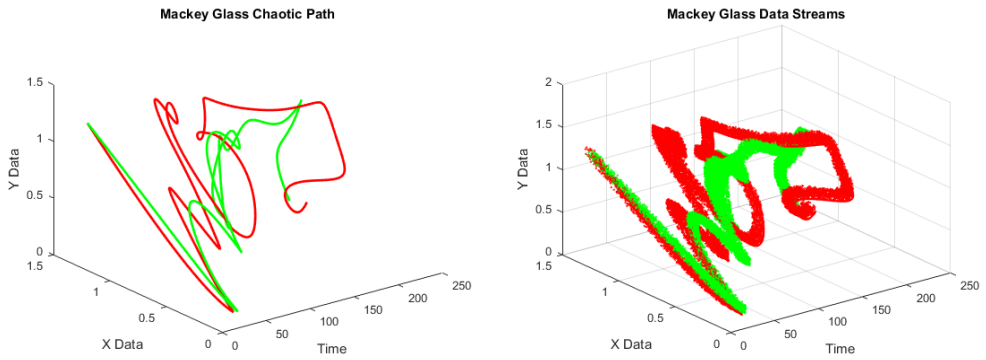
5.3.6 CEDAS Functionality with Cluster Separation, Cluster Merging, Drift and Noise

A 3D data stream consisting of 2 Mackey-Glass time series is presented as a data stream. The data stream is a pair of solutions of the Mackey-Glass non-linear time delay differential equation [104, 67]. shown in equation 5.1.

$$\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1+x(t-\tau)^{10}-bx(t)} \tag{5.1}$$

For each data stream the equation is solved twice, once for the x values and again for the y values using the parameters for a and b given in Table 5.4.

The equation is solved numerically at discrete time steps using the 4th order Runge-Kutta method using different values for a and b to create x and y values as shown in Figure 5.7a. For each time step 10 random data samples were created around the core value to provide a data stream of 40,020 samples illustrated in Figure 5.7b. Early in the data the values of both data streams are coincident. They later separate and come



(a) Chaotic Mackey Glass data path (b) Random data around the chaotic path

Fig. 5.7 Illustration of the Mackey-Glass data sets, a) the chaotic path b) the data stream created around that path. The two Mackey-Glass streams are shown in red and green. When considering the data over the previous 'N ' samples the data may form separate streams, two clusters, or streams that are joined at some point, a single cluster.

together at various times. It would be expected that 'recent' data will produce a changing number of macro-clusters, as the data streams separate and rejoin, and that an online, evolving clustering technique will detect these changes as they occur. A further data set was created by the addition of 10% random noise samples creating a dataset of 44,022 samples. These data are used to test the robustness of CEDAS to detecting the clusters in a noisy environment. By presenting the data sequentially a continually evolving data stream is generated, rather than a data stream of similar values with sporadic variation, such as the KDDCup data set below. This tests the ability of the algorithm to add, merge and separate macro-clusters in a continuously evolving environment.

To validate the correct functionality of CEDAS the algorithm was applied to the Mackey-Glass data streams using $Decay = 1,000$ samples, $Radius = 0.05$ and $MinimumThreshold = 15$. The decay is a suitable time period for investigation such that the macro-clusters will be large enough to visualise, and the two Mackey-Glass data

Table 5.4 Values for a and b used to solve the Mackey-Glass equations for the test data streams.

Stream	X-Data		Y-Data	
	a	b	a	b
1 (Red)	0.2	0.1	0.25	0.11
2 (Green)	0.18	0.12	0.22	0.10

streams will be merged and separated for sufficient time to indicate the correct operation of CEDAS. The radius is selected such that the 'width' of the data streams is encapsulated minimising the plotting time for multiple micro-cluster spheres. The minimum cluster size is such that it is larger than the expected density of the noisy data ensuring that the noise remains as outliers.

The data is presented to the CEDAS algorithm one sample at a time to imitate an online data stream and the results plotted at each time step to create a video of the results. The CEDAS algorithm was used to detect and report in the plot title the following information:

1. Definite Clusters: these were defined as clusters containing > 15 data samples and > 1 micro-cluster. These are settings specific to the investigation to interpret the algorithm results and are not algorithm parameters.

2. Outlier Clusters: these were defined as containing > 15 data samples all contained in 1 micro-cluster. These are also specific to interpretation of the results and not algorithm parameters.

3. Last Change: the time period at which the last change in the number of Definite Clusters occurred. This information was recorded to allow the state at that time to be reproduced.

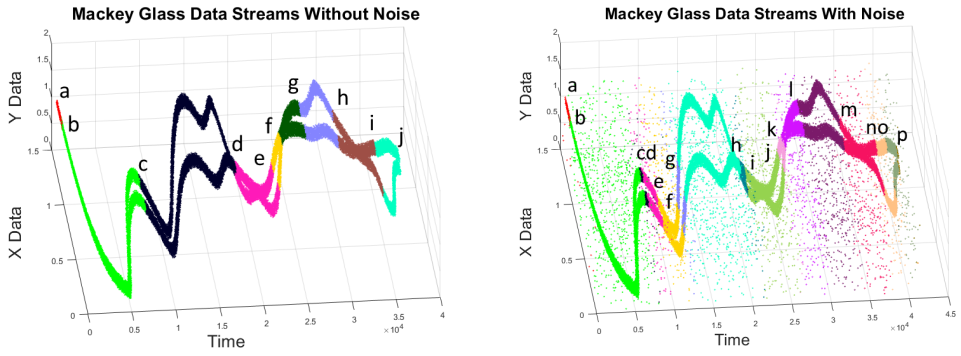
Cluster Separation and Merging

Using the clean Mackey-Glass data stream the sample number at which a change in the number of macro-clusters was detected was stored. After the analysis data is plotted and coloured differently each time the number of macro clusters changed. This is shown in Figure 5.8a.

After the initial settling period (red), it is seen that at each colour change the number of clusters in the data contained in the preceding decay period has changed. For example, in the green period the data was contained in a single cluster. At the time the colour changes to black, the data in the previous 1,000 samples had just separated to 2 separate macro-clusters. When the colour changes to magenta, now the previous 1,000 samples create 1 macro-cluster. The colours of the data do not represent the clusters themselves, but represent the period preceding the time at which the number of macro-clusters changes.

The Effects of Noise

To test the effects of noise on CEDAS the Mackey-Glass dataset is used with a random noise sample added every 5 data samples as described above. The random nature of the



(a) Without Noise (b) With Noise

Fig. 5.8 CEDAS Auto Change Detection, changes in colour represent changes in the number of clusters. Thus in figure 5.8a while the data is coloured green previous 'N ' samples form a single cluster, joined at the beginning. At the point the data colour changes to black, the data in the previous 'N ' samples has separated into two clusters. It should be noted that the colours of the data are not the clusters themselves, but represent the time periods during which the data forms different numbers of clusters. The changes detected without noise are also detected with noise with the additional changes caused by temporary separate micro-clusters before they rejoin the main clusters.

noise will have some effect on the initial positions of micro-clusters if the noise falls within them. This increases the likelihood of an initial micro-cluster separating from the main macro-cluster group. If this occurs then the number of macro-clusters may change briefly. This would give the appearance of false positives when compared with the results from dataset without noise. These additional clusters are, in fact, present at that time and it is accepted that the noise has changed the clustering.

The results are shown in Figures 5.8a and 5.8b. Figure 5.8b illustrates that trigger points c, e, f, g, h, o have been created by the noise and could easily be discounted based on the number of samples, if required. The trigger points without noise can be matched to those with noise as shown in table 5.5. These are discussed in the following Subsections.

False Positives

With any evolving technique, apparent changes at some point in time may turn out to be irrelevant at a later time. An example of such soon-to-be-irrelevant data anomalies are those that result from the added noise. Rather than calling these 'false positives', they could be considered as 'temporary or short-term true positives'. In the event these are caused by temporary misplacement of micro-clusters caused by noise, which are rapidly re-absorbed into the macro-cluster, then these addition clusters will have an unusually short lifespan, i.e. considerably shorter than the set decay period. In this way any triggers that are within a user-defined short time span from a previous trigger could be discounted

Table 5.5 Comparison of trigger points with and without noise. The trigger points in brackets with noise are short term and caused by the effect of noise in moving the micro-cluster positions briefly.

Trigger Points		
Group	Without Noise	With Noise
1	a	a
2	b	b (c)
3	c	d (e, f, g, h)
4	d	i
5	e	j
6	f	k
7	g	l
8	h	m
9	i	n
10	j	(o) p

if required. However, this is not always desirable, as even short term anomalies may be of interest. They may, for example, indicate the start of a general drift or shift in the data.

False Negatives

With appropriate settings for decay time and micro-cluster radius, i.e. ones that match the users definition of what constitutes a cluster, false negatives do not occur. At any time that the number of data samples within the given radius are present, then a micro-cluster above the minimum threshold occurs, and therefore a macro-cluster will be present. It follows then that the algorithm cannot deny the presence of a cluster where one exists. It must be remembered that a different decay time will create different times for cluster separation. This is not indicative of false negatives, but rather a deliberate function of the technique to consider clusters based on data within a defined time frame.

True Positives

As demonstrated, all changes to clusters are correctly detected. With the noisy dataset some temporary true positives may occur, as discussed, but CEDAS has successfully detected the same true positives as with the clean dataset as shown in table 5.5.

True Negatives

Taking the definition of a ' true negative' to be that ' no changes in macro-clusters are detected when there are none' then this occurs with every sample that does not create new clusters.

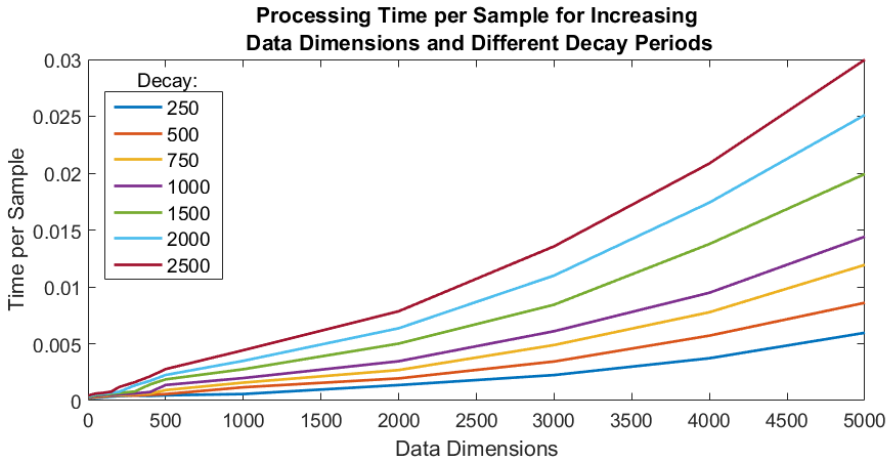


Fig. 5.9 Plot of mean processing time per sample in seconds for varying data dimensionality. Each line represents the processing time for different decay periods which create a proportional increase in micro-clusters, e.g. the top, red line represents the processing time per sample for a decay period of 2,500 samples for data with dimensions from 1 to 5,000.

5.3.7 High Dimensional Data Test

The dataset used in this section comprises three helical data streams, two of which join mid-way through the test while the other stays separate. These data streams are moved through a range of multiple dimensions to examine the time variance of the analysis with higher dimensional data. The data was analysed using CEDAS with a range of values for *Decay* and settings of $InitialRadius = 0.05$ and $MinimumThreshold = 4$. As the aim of this experiment is to test the speed penalty across high dimensional data and not to test the efficacy, a-priori knowledge of the data streams was used to ensure valid clustering occurs. *Decay* was set at a reasonable number of samples to ensure macro-clusters of a suitable size to demonstrate the effectiveness of the technique. The radius is set smaller than the width of the helices to ensure multiple micro-clusters at all times, and below the minimum expected gap between natural clusters. The minimum threshold was set to 4 to restrict micro-clusters from forming on the very edge of the natural clusters. The data set is then moved into higher dimensional data space by adding additional dimensional data coordinates. By projecting the data back into 3 dimensions the clustered data can be plotted and the results of cluster membership checked while increasing the complexity of the clustering calculations.

Speed and Dimensionality Comparison

By utilising hyper-spheres for micro-clusters the cluster joining technique checking for micro-cluster overlap is much simpler than, e.g. hyper-ellipsoidal micro-clusters. Micro-

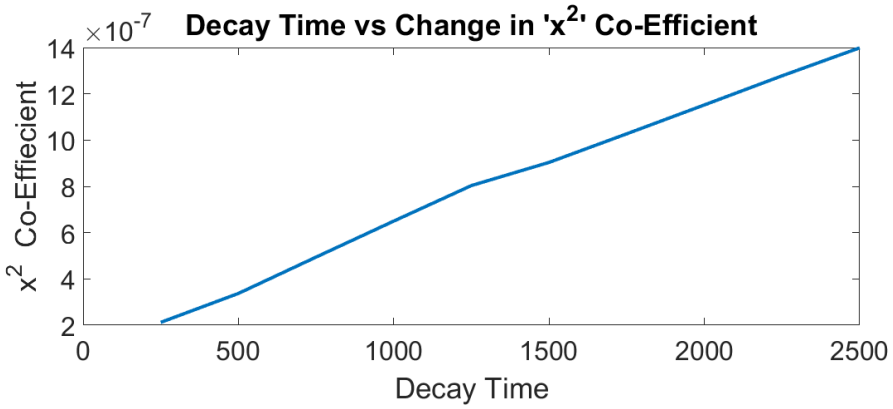


Fig. 5.10 Comparison of the processing time per sample with the decay time showing a linear relationship between processing speed and decay time. For the data used in this example the decay time is directly proportional to the number of micro-clusters. Where a longer decay time does not result in additional micro-clusters, then the time per sample remains constant. In practice the processing time will lie somewhere between the two.

clusters are joined if the edge of the core hypersphere intersects another hyper-sphere shell. This requires only a comparison between the Euclidean distance between cluster centres and the sum of the micro-cluster radii. Therefore, the only calculation that is dimensionally dependent is the Euclidean distance with complexity $O(D)$ where 'D ' is the number of dimensions. The relationship between the number of data dimensions and processing time per sample is linear.

With each new data sample being assigned to a single micro-cluster it is only necessary to check the intersections for that micro-cluster and then only if the micro-cluster centre has been modified, a new micro-cluster has been created, or a micro-cluster has been removed. This further reduces the required number of calculations. The radii of the micro-clusters is constant and so the only calculation is to compare the Euclidean distance between the changed micro-cluster and all others with $1.5r_0$.

The relationship between the number of data dimensions, decay period and calculation time is plotted in Figure 5.9. In the case of an evolving data stream with continuous drift the decay time is also proportional to the number of micro-clusters. To investigate the relationship between decay time, and so the number of micro-clusters, and run time per sample CEDAS is tested for varying numbers of data dimensions from 3 to 2,500 for a range of decay periods. The processing time per sample is plotted in each case and the best fit polynomial line of degree 2 is found. A linear relationship is found between the decay period and the number of data dimensions as shown in Figure 5.10. These results concur with the predicted linear time penalty for both the number of dimensions and the number of micro-clusters.

By comparison, Figure 5.11 shows the relationship between processing time and dimensionality with the same data set for comparison with both DenStream [26] and CluStream [3]. The Massive Online Analysis [20] implementation running on R3.2.2 in RStudio 0.98.1102 was used, analysing the same helical high dimensional dataset as for CEDAS. CluStream was also limited to a maximum of 100 micro-clusters. For both of these techniques, two tests were run using a decay time of 1,000 samples:

1. Both DenStream and CluStream without carrying out the 2nd stage re-clustering until the end of the data stream.

2. An approximation of a fully online technique by carrying out the 2nd stage clustering technique at frequent intervals - every 100 samples for DenStream and every 10 samples for CluStream.

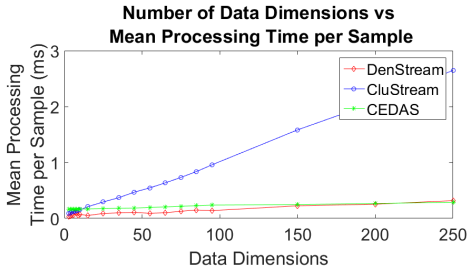
For the DenStream 2nd stage re-clustering DBScan [53] was used as implemented in the 'R' package by Hahsler [74] to allow for arbitrary shaped macro-clusters to form in a similar manner to CEDAS. The results shown in Figures 5.11a and 5.11b are for test 1 and the results shown in Figure 5.11c and 5.11d are for test 2. Without 2nd stage re-clustering both DenStream and CluStream are faster than CEDAS for low dimensionality data. The break-even point is approximately $12D$ for CluStream and $220D$ for DenStream. When the second stage re-clustering of the micro-clusters is done frequently enough to approximate fully online analysis there is significant time penalty for both DenStream and CluStream. In both cases CEDAS is noticeably faster than both DenStream and CluStream and suffers significantly less time penalty for increasing data dimensionality.

5.3.8 Using CEDAS to Identify Computer Network Intrusion Attacks

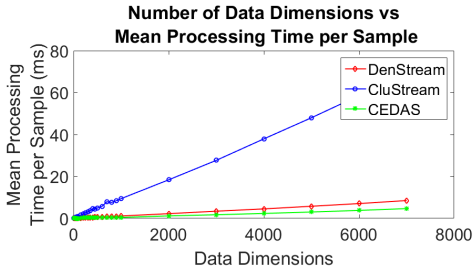
To further test the CEDAS algorithm in a different environment the KKDCup99 [77] dataset was used as a data stream by presenting the data to the algorithm sequentially. The data set consists of approximately 5 million samples in the full data set, 500,000 samples in the 10% reduced set, simulating network intrusion attacks on a military installation. The dataset has 42 features and information to classify the data into 22 attack types in addition to the normal network traffic. This data is used to determine the cluster purity and memory use for comparison with alternative techniques and also to validate the clustering results in relation to the number of attack types which occur in a time period.

Speed and Cluster Quality

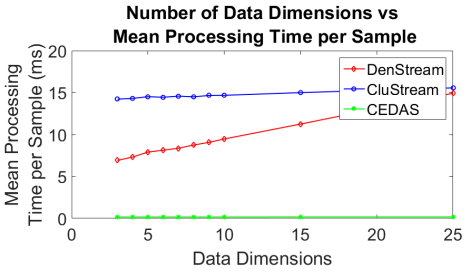
The KDDCup99 data stream is a popular dataset for testing evolving clustering algorithms such as eClass [10] and it is used here to allow direct comparisons with D-Stream and



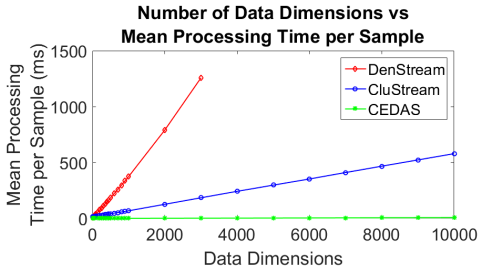
(a) Without Re-Clustering Low Dimensionality



(b) Without Reclustering High Dimensionality



(c) With Re-Clustering, Low Dimensionality



(d) With Re-Clustering High Dimensionality

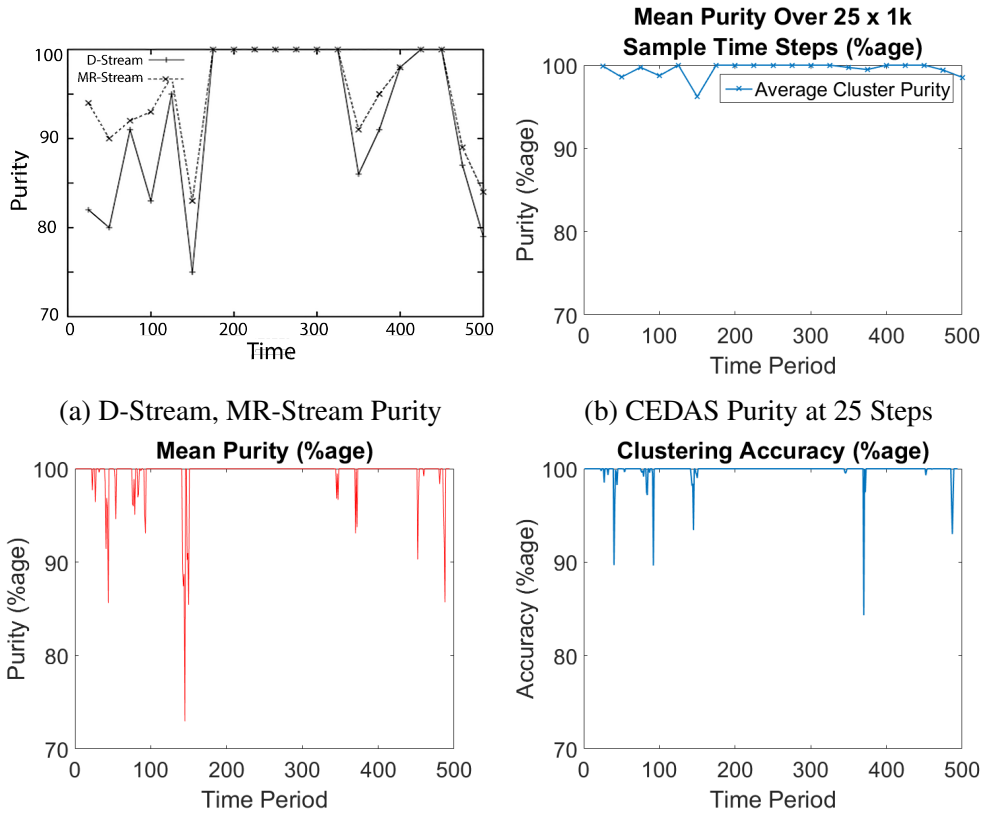
Fig. 5.11 Typical analysis time per sample for DenStream, CluStream and CEDAS across various dimensional data. a) and b) show CluStream and DenStream without 2nd stage re-clustering until the end of the data stream. c) and d) show DenStream and CluStream with frequent 2nd stage re-clustering. In all plots CEDAS is shown in green. DenStream and CluStream have a faster 1st stage clustering, but for fully online clustering CEDAS is shown to be faster.

MR-Stream purity results provided by Wan et al [155]. Two sets of results are presented. The first is the same analysis used by Wan et al. of creating 500 time intervals spaced at 1K samples and placing these into groups of 25 and taking the mean cluster purity over these groups of 25. Taking the mean of a set of results can disguise individual poor results and so the cluster purity for CEDAS at each of the 500 time intervals is also provided. These results are shown in Figure 5.12.

It should be noted that the mean cluster purity alone, as defined by equation 5.2, may be a poor measure by itself.

$$mean\,purity = \frac{\sum_{i=1}^{N} \frac{|C_i^d|}{|C_i|}}{N} \times 100\% \tag{5.2}$$

$$accuracy = \frac{\sum_{i=1}^{N} |C_i^d|}{\sum_{i=1}^{N} |C_i|} \times 100\% \tag{5.3}$$



(a) D-Stream, MR-Stream Purity

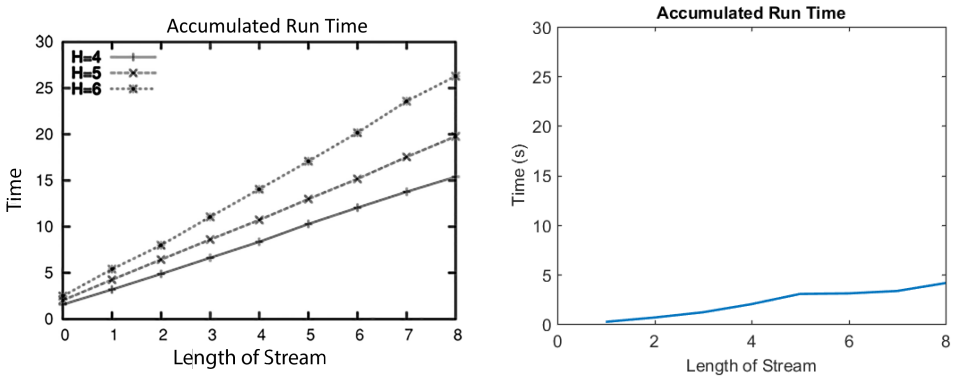
(b) CEDAS Purity at 25 Steps

(c) CEDAS Purity at All Steps

(d) CEDAS Accuracy

Fig. 5.12 (a) Plot of mean cluster purity (taken from [155]), (b) Mean cluster purity for CEDAS by the same measure as Wan et al. [155]. (c) Cluster purity at each time step showing instances of reduced mean purity. (d) CEDAS accuracy measure.

Here C_i is the number of samples in a cluster, C_i^D is the number of these samples assigned to the dominant class and N is the number of clusters. In cases where a high number of samples are contained in one cluster with low purity, yet few samples are contained in a high number of clusters with high purity the result is a high mean purity even though most samples are incorrectly assigned. Equally, the reverse is true when few clusters are present, if 99% of the data is correctly assigned in one cluster and two sample are contained in a second, one of which is mis-assigned the mean purity looks poor. In Wan et al. the relevance of this measure is further reduced by taking the mean of these means and so the purity measure is included here for comparison to Wan et al. only and not to attach any particular significance to the result. The cluster accuracy measure as defined in equation 5.3 is presented in Figure 5.12d which is a measure of the number of clustered samples that have been correctly assigned to the dominant class. By using both the purity and accuracy measures the quality of the clustering can be stated with greater confidence.



(a) Processing Speed for MR-Stream (b) Processing Speed for CEDAS

Fig. 5.13 Figure 5.13a shows plots of the processing for MR-Stream for various grid depths (from [155]). Figure 5.13b show the processing time for CEDAS to the same scale.

The results of the quality analyses are shown in 5.12. Figure 5.12a shows the results provide by Wan et al. for the mean purity over 25 time steps for both D-Stream and MR-Stream. Figure 5.12b shows the same analysis for CEDAS. Also shown are the mean purity at all steps, Figure 5.12c, and the cluster accuracy for CEDAS, Figure 5.12d.

Although the purity at time period 145 is 73%, the mean over the 25 time periods this is 96%. Using the two time periods selected by Wan et al, 27 and 52, the CEDAS purity was 96% and 99.85% compared with MR-Stream at 97.5% and 92% respectively. It is interesting to note that at time periods 26 and 28 CEDAS purity is 100% suggesting that CEDAS adapts quickly to this variation. Using the 25 time periods measure favoured by Wan et al. the CEDAS mean purity exceeds that of MR-Stream. When considering the accuracy measure at the time periods 27 and 52 the accuracy measurements are 98.5% and 99.98% respectively. This indicates that nearly all the samples are correctly assigned to the dominant clusters, but the purity is reduced due to few incorrectly assigned samples in clusters with few members. The accuracy of CEDAS remains close to 100% at all times except for 3 single occasions where it drops to around 90% and 2 at around 95%.

The processing times for the MR-Stream is provided in Figure 5.13a and the equivalent measure for CEDAS is shown in Figure 5.13b. These graphs show the accumulated processing time at stages throughout the data stream. CEDAS is noticeably faster as well as producing clusters of a higher purity and greater accuracy.

Having established via the purity and accuracy measures that the clusters are meaningful it is useful to see if they demonstrate any results of interest. To do this the number of clusters in a time period are compared with the number of classes given in the data. The plot of these is given in Figure 5.14 where it can be seen that each time there is a rise in the number of classes, i.e. attacks, the number of clusters also rises. Given that these

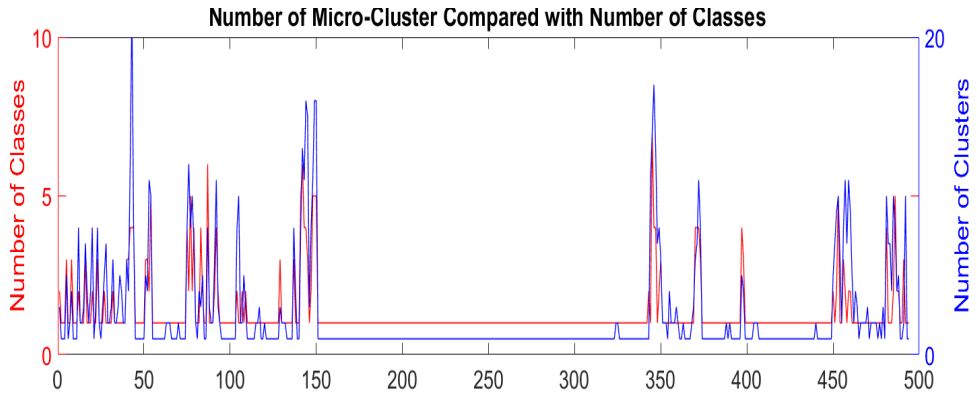
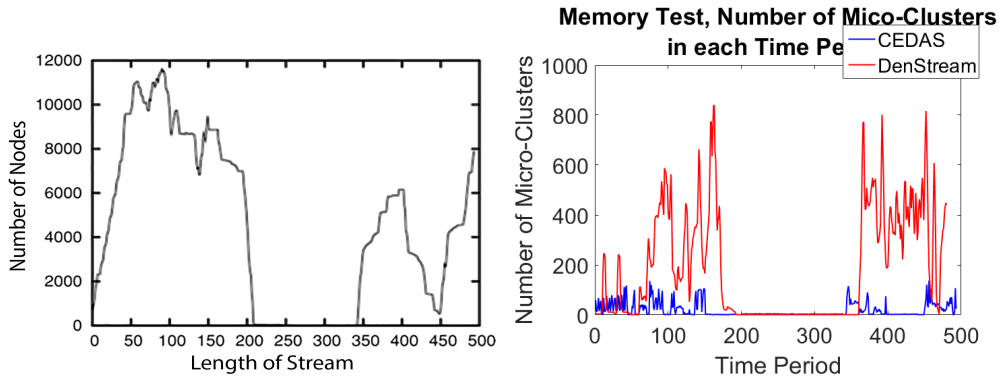


Fig. 5.14 Plot of the number of classes of attack and the number of clusters found by CEDAS in each time period. The number of clusters is proportional to the number of classes throughout.

clusters have high purity, and the accuracy of clustering is also high, these additional clusters must contain attack vectors unique to each type of attack. There are 50 time periods with attack vectors present and these are detected 100%. As discussed above, occasional separated micro-clusters are a feature of evolving techniques and providing they are short-lived and re-absorbed into the main clusters they can be ignored with reasonable confidence. When the number grows beyond 1 sample per cluster, however, they may be indicative of possible attacks. Thus with a threshold of 1 then 20 false positives occur. However increasing the threshold to 2 to allow for occasional separated micro-clusters reduces this Figure to 4, and a threshold of 3 reduces this to a single instance. This compares favourably with a mean number of clusters per attack of 8.2.

Memory Efficiency

To demonstrate the efficient memory use of CEDAS, the storage required by MR-Stream and DenStream with that required by CEDAS is compared when clustering the KDDCup99 datastream. The results presented by Wan et al. for MR-Stream are shown in Figure 5.15a and, when the data stream is evolving and has variety, MR-Stream reaches figures in the thousands of nodes with a peak approaching 12,000. By contrast, the number of micro-clusters required by DenStream and CEDAS for the same data stream are shown in Figure 5.15b. DenStream has a mean value of 181 and maximum of 839 whereas CEDAS has a mean of 20 and peaks at 137. This demonstrates the significant memory saving of micro-clusters over grid based techniques. Even allowing for the CEDAS cluster description consisting of 5 values there is significant saving over MR-Stream.



(a) Memory Usage for MR-Stream (b) Memory Usage for CEDAS, DenStream

Fig. 5.15 Plot of the number of nodes or micro-clusters, which equates to memory use, for MR-Stream (from [155]), DenStream and CEDAS. CluStream is not shown as it uses a maximum number of micro-clusters set by the user. CEDAS shows the lowest memory use.

5.3.9 Data Mining of Atmospheric Data Streams Using CEDAS

In this section CEDAS is applied to data from the Kings College London Air Quality Website [50]. The data is from one monitoring site, Westminster Marylebone, and 2 dimensions are used, labelled NO_x, PM_{10}. Here and throughout, NO_x is defined as the reactive oxides of Nitrogen, primarily NO and NO_2, and PM_{10} is defined as the mass concentration of microscopic airborne particles with aerodynamic diameter of $10\mu m$ or above. The data, which is recorded operationally to monitor breaches of air pollution legislation [130] and to inform the public of adverse air pollution conditions, is captured at 15 minute intervals and ranges from 1^{st} January 2010 to 30^{th} December 2014 for a total of 87,600 samples. This data is used to test CEDAS ability to differentiate short and long term anomalies and follow the temporal drift of real data.

To allow for clustering to take place the data is normalised to a suitable range relative to the micro-cluster radius, $r0$. Here the range was based on the data available in the dataset and scaled to $0-1$. The data had an actual range from $min = 7.200$ to $max = 1,447, ppbv$ (parts-per-billion by volume) for NO_x and $min = -0.9$, $max = 422.8$ (μgm^{-3}) for PM_{10} and so predicted ranges of 0 to 1500 and 0 to 200 respectively were used. The scaling introduced by this normalisation has an effect on the local density, joining and separation of micro-clusters and so expert knowledge is required to find suitable values for scientific research involving the cluster results.

Anomaly detection differs between long-, medium and short-term analysis and how CEDAS copes with such variation is shown. To demonstrate this '*Short Term*' is defined as 7 days and '*Medium Term*' as 28 days and '*Long Term*' as being one year. The *decay* values used correspond to the number of data samples collected in the respective time

period. For the *radius* a value of 0.05 is used. This value is arrived at by looking at historical data and estimating the distance from the main natural cluster to data that would be considered an outlier. The definition of what is considered different enough to be an outlier is at the discretion of the expert user. The analysis carried out here is robust to a range of *radius* values with little change in the macro-clusters or their visual appearance. All data space regions containing data are of interest, including single outliers and so the minimum threshold is set to 1.

The data used was collected between 2010 and 2014 inclusive. The data was presented to the CEDAS algorithm sequentially, in NO_x, PM_{10} pairs, to mimic an online data stream. The micro-clusters were plotted and the transparency of the micro-clusters set according to the value of the Energy in each. In this way if anomalous data appears for a short period of time the cluster adjusts, but it fades over the subsequent time period providing an online visualisation of the Energy of the micro-clusters. This provides a clear visual indication of CEDAS adapting to the changes in the data stream and following long term and short term drift. By using different decay times different clusters are created and this is shown to be useful to investigate different time periods for drift, shift and anomalies.

The Subsection 'Short Term Drift and Anomalies' the use of a short decay period is demonstrated to reveal short term data drift that would be disguised in medium term decay periods. Subsection 'Medium Term Drift and Anomalies' describes the use of medium term decay periods to investigate possible seasonal variations. Finally, in Subsection 'Long Term Drift and Anomalies' demonstrates how medium term decay periods can be used to investigate long term variations. Visual indications of how CEDAS reacts to the evolving data stream are used to demonstrate how the technique can add value for the expert user. Potential numerical analysis of the clustering results is discussed in Section 5.4 Conclusions.

Short Term Drift and Anomalies

Using a decay period, as a number of samples, equivalent to 7 days of data changes in NO_x and PM_{10} over time can be detected. Sample plots are shown in Figure 5.16 (a)-(c) showing the cluster analysis at 3 different dates for the preceding 7 days. The data for the preceding 28 day period, for the same dates, is shown in Figures 5.16 (d)-(f).

The 7 day period preceding 24/03/11 is markedly different from the 7 day period preceding 06/02/11. Despite these differences in the 7 day data, by comparing the plots (d)-(f) it can be seen that, overall, for the preceding 28 day periods the spread of data values has been more consistent. The data shown in the black and green clusters of the 7 day analysis in 5.16 (b) may be considered anomalous for that week, but in Figure 5.16 (e) it is shown not to be unusual over the preceding 28 day period. However, data such as

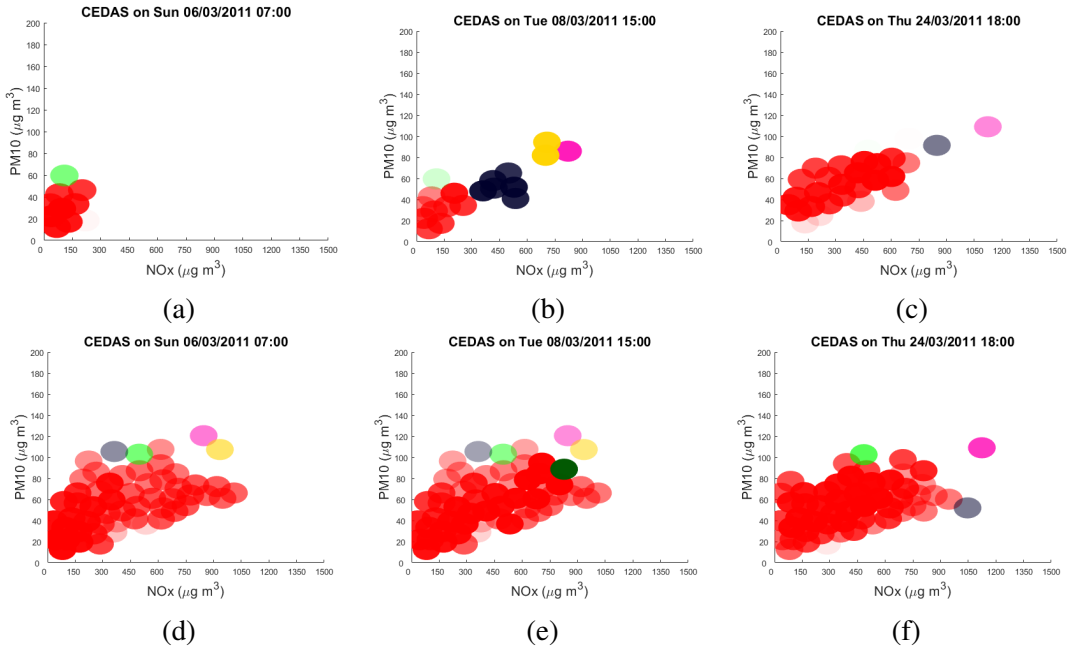


Fig. 5.16 Sample plots of short term decay periods (a)-(c) and medium term decay periods (d)-(f). The short term variations indicated in (a)-(c) show the data varies over different 7 day periods. The medium term variations in (d)-(f) show that the data over the 28 day periods is more consistent and disguises the 7-day variation.

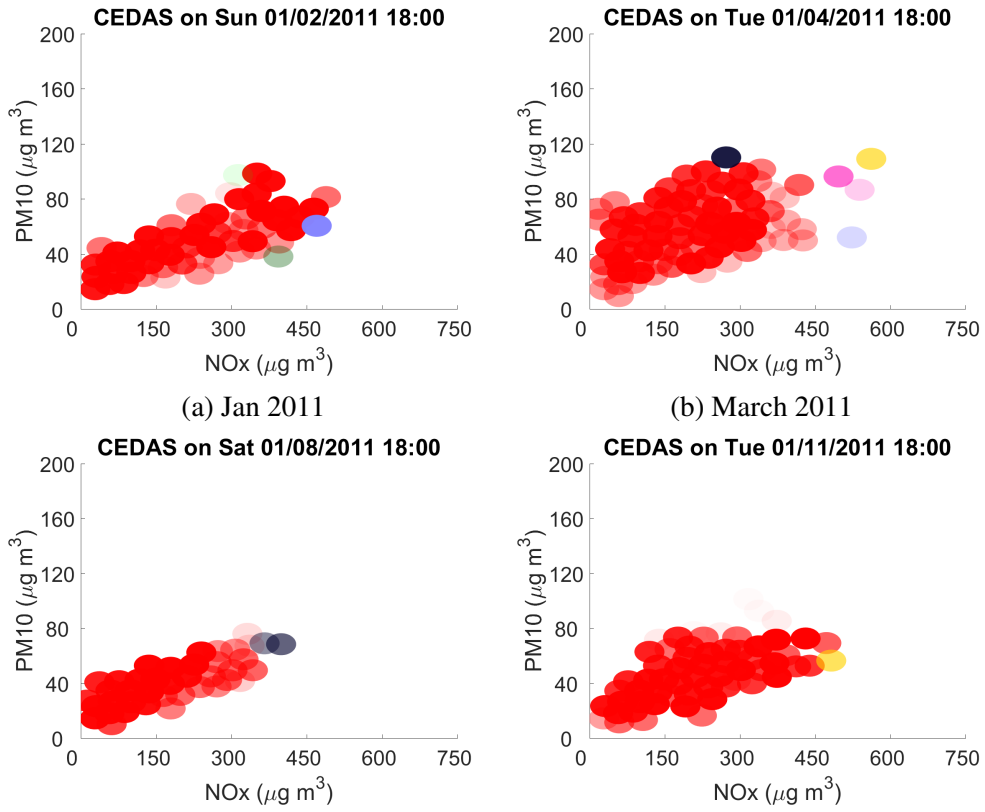
that in the yellow and magenta cluster of 5.16 (b) is seen to still be anomalous over the 28 day period, Figure 5.16 (e), where the clusters are now coloured khaki and blue.

This demonstrates that, by selecting suitable decay periods, the clustering results from the proposed algorithm provides relevant analysis of how data behaves over different time periods and how CEDAS can follow these changes in a fully online manner.

Medium Term Drift and Anomalies

The plots in Figure 5.17 are the cluster results for a 28 day decay period taken at different dates throughout the year. Over the 5 year period of the data streams this approximate pattern is repeated each year. The primary variation is not in the maximum, minimum or range of either NO_x or PM_{10} but rather in the range of the $PM_{10} : NO_x$ ratio. This is particularly noticeable when comparing, e.g. March and July where at any given value of NO_x the range of PM_{10} values is greater in March. Anomalous data can also be seen in March indicating that some unusual events are present.

This demonstrates the ability of CEDAS to follow such seasonal drifts, if they exist, and find data that is anomalous within that local time frame.



(a) Jan 2011 (b) March 2011

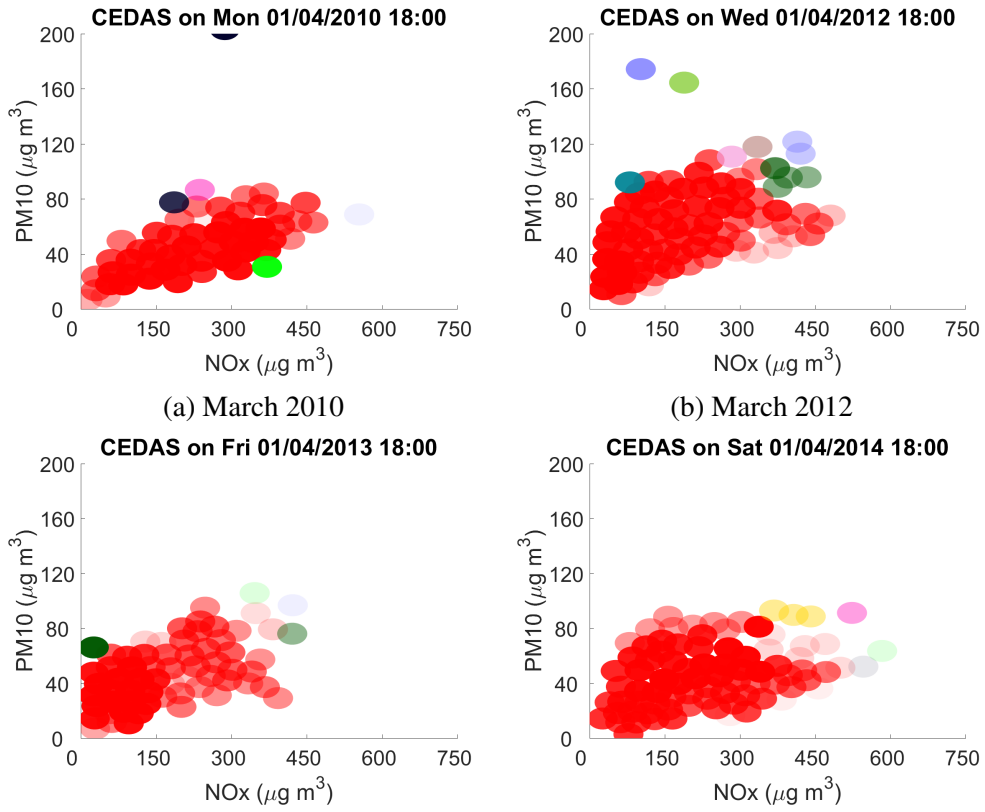
(c) July 2011 (d) Oct 2011

Fig. 5.17 Plots of CEDAS clustering for a 28 day decay period showing a variation of the data spread at different dates during a single year.

Long Term Drift and Anomalies

For long term changes, i.e. changes across years, the data could be analysed in multiple ways. For example, the data could be clustered on the full 365 day decay period. However, as has already been indicated in the Subsection 'Medium Term Drift and Anomalies' there are variations within that year which may be hidden in the way described in Subsection 'Short Term Drift and Anomalies'. With this information it is reasonable to consider an analysis of 28 day decay periods, at the same date, for subsequent years. Examples of these cluster results are provided in Figure 5.18 and shows the results for data of the 28 days preceding 01/04 for the years 2010-2015.

The shape of the main cluster can be seen to vary between years indicating the changes in data values. Anomalies are indicated and are for the particular month and year under consideration. In all cases some relatively minor anomalies with values that are slightly different from the main cluster can be seen. These could be symptomatic of the data undergoing normal drift and changes. March 2012, however, shows some more extreme anomalies, shown in blue and green, with particularly high PM_{10} values.



(a) March 2010

(b) March 2012

(c) March 2013

(d) March 2014

Fig. 5.18 Plots of CEDAS clustering with a 28 day decay period showing variation of the data for March over a 5 year period.

These anomalies detected in March 2012 were not measured in any other year. This demonstrates how CEDAS may be used to analyse yearly changes, i.e. long term shifts in data and find anomalies independent of drift.

5.4 CEDAS Summary and Conclusions

A new, fully online clustering technique for clustering data into arbitrarily shaped clusters is proposed. In Section 5.3.3 the algorithm has been described. The technique has been applied to the various data sets described in Section 5.3.5 and the results presented and discussed. In this section the results are summarised together with appropriate conclusions.

5.4.1 Technique Validity

Section 5.3.6 demonstrates the ability of CEDAS to accurately divide and merge evolving data streams where appropriate demonstrating the validity of the technique. The proposed algorithm is also shown to be robust to noise.

5.4.2 Cluster Quality

In Section 5.3.8 the proposed algorithm is compared to CluStream, DenStream and MR-Stream and demonstrated that in the tested scenarios CEDAS performed as well as, or better, than all three alternatives. Including the additional accuracy measure provides evidence that the mean cluster purity measure is, in the case of CEDAS, a fair measure of the cluster quality.

5.4.3 Computational Efficiency

When working with stable data-streams with few micro-clusters, DenStream and CluStream approach the speed of the newly proposed technique. However, when the data stream evolves more rapidly, or there are a higher number of micro-clusters, the offline portion of combined online/ offline techniques becomes a limiting factor and CEDAS becomes significantly faster. In the case of low dimensionality and where the second stage, offline, technique is not required often then DenStream and CluStream may also be faster. However, this precludes these techniques from being considered as fully online. If excessive periods of time are allowed between second stage clustering important clusters and their information may go unnoticed. By being fully online the proposed technique will not suffer from this limitation. It should also be noted that CluStream finds only hyper-elliptical and not arbitrarily shaped clusters.

5.4.4 Memory Efficiency

In general, the similarities in the micro-cluster stage means there is a similarity between memory use for CEDAS, DenStream and CluStream. For micro-clusters of a similar size the number will be similar for each technique. MR-Stream is highly memory intensive, not only does it store data for all the cluster nodes, but also for those nodes on the higher plane. MR-Stream claims to use this information to reduce the calculations required for the second stage clustering. However, in the case of a highly populated data space this will result in an increase in memory storage and calculations as a high proportion of the nodes and their parents need to be stored and visited during the second stage clustering. In an effort to reduce the memory requirements MR-Stream prunes nodes with a low

density, however this implies a possible loss of data without regard to its relevance to the current state.

5.4.5 Dimensionality

The proposed algorithm has a linear complexity and time penalty relative to the number of data dimensions. DenStream and CluStream have a similar linear complexity and time penalty, however, it is shown in Section 5.3.8 that the penalty is lower for CEDAS. MR-Stream has penalty of n^{DH} for dense data space rendering it more suitable to low dimensional sparse data, particularly when considered with the memory requirements.

5.4.6 Decay Time and the Number of Micro-Clusters

The proposed algorithm has a linear time penalty related to the number of micro-clusters. This is common to all two stage clustering techniques, including those alternatives discussed previously. In cases where the data is fairly static in the data space this has little relevance, however, if the data samples are continuously drifting through the data space there is a relationship between the speed of drift and the number of micro-clusters exists. The maximum number of possible micro-clusters are present when each micro-cluster contains T_{min} data samples and the number of micro-clusters is given by equation 5.4.

$$number\,of\,micro-clusters = \frac{Decay}{T_{min}} \tag{5.4}$$

Thus the worst case is a linear relationship between the decay time and the number of micro-clusters and so it follows a linear relationship between the processing time and the decay period. In practice data streams that drift at such a high rate are likely to be rare and may require a different type of analysis in any case. In the opposite extreme of fairly static data the number of micro-cluster will vary little and no time penalty results from an increase in decay time.

5.4.7 Anomalies, Drift and Time

Sections 5.3.8 and 5.3.9 discuss the ability of the proposed algorithm to cope with drift and anomaly detection in real data streams. In both these sections CEDAS proved capable of accurately detecting anomalies within the defined time periods demonstrating possible applications in network security and atmospheric science research. The results in Section 5.3.8 demonstrate how CEDAS could be used to automate detection across multiple dimensions that cannot be easily visualised, whereas Section 5.3.9 presents a visualization for primary interpretation by the user.

Table 5.6 Summary of clustering techniques required to meet the defined atmospheric science challenges. [1] CODAS will perform spatial separation only.

Challenge	Online	Offline	On \Offline	Arbitrary Shapes	Technique
1	Y	Y		Y	CEDAS, DDCAS
2	Y		Y	Y	DDCAS, CODAS, CEDAS
3	Y		Y	Y	CEDAS
4	Y	Y		Y	CODAS[1], CEDAS
5		Y		Y	DDCAS
6	Y		Y	Y	CODAS, CEDAS
7		Y		Y	CEDAS, CODAS, DDCAS
8		Y	Y	Y	DDCAS

5.4.8 Summary of the Benefits of CEDAS

Clustering of Evolving Data Streams into Arbitrary Shapes has been demonstrated to be a robust and accurate technique with linear complexity across both data stream size and data stream dimensionality. It is a fully online technique providing constant and immediate access to the clustering results as they change with each data sample. This technique has been applied to real life datasets and shown to produce useful insights into evolving data streams. It is intended to be used with a decay period to reduce the importance of older data, however, if the decay period is set to zero then the results become the same as CODAS.

The ability to perform fully online clustering is an important step towards solving the Atmospheric Science data challenges set in chapter 2 by addressing the clustering issues discussed in chapter 3. In chapter 6 CODAS and DDCAS are brought together in some demonstration software illustrating possible uses for the novel data mining abilities they provide.

5.4.9 Overall Summary Of Online Clustering Techniques

To review the goals of the online clustering algorithm research, Table 5.6 lists the clustering algorithms that will be used, and how, in the RASCAL software proposed in chapter 6. This table also include the the offline algorithms from Chapter 4 and the online algorithms developed in this chapter, to show that solutions to all the challenges introduced have been achieved.

Chapter 6

Online Real Time Atmospheric Science Cluster Analysis with Offline Compatibility

6.1 Background of RASCAL

The spatially inhomogeneous and temporally intermittent fluid mixing of air, combined with localised sources and sinks of sensible heat, latent heat and trace gases, typically results in variability within regions, atmospheric 'compartments' (the boundary layer, the free troposphere, maritime air, etc.), together with more-or-less sudden changes between these atmospheric regions compartments. The edges of what could can be considered homogeneous air parcels are rarely clearly defined, and this is true whether we think of edges in physical (3D) space or in hyper-dimensional data space. Therefore, the result is a cloud of 'similar' measurements that seek to describe these air parcels, which are similar but not identical, and this 'fuzzy' similarity is ideal for cluster analysis.

Although it is not novel to apply clustering algorithms to atmospheric data of one kind or another [33, 28, 127, 41], the assumptions underlying such clustering have not always been made explicit, and methods for on-line clustering have not been available. On-line, or real-time, methods are necessary for time-critical applications such as natural hazard monitoring [66, 142] and pollution threshold exceedances prompting public health alerts [111, 72, 17]. There has been less consideration in the atmospheric science community about real-time adaptation of measurement or monitoring schedules, from stationary or mobile measurement platforms, but the advent of large robotic aircraft capable of returning data to ground stations at rates approaching megabits per second (up to 3Mbs during 3 coordinated flights during the CAST campaign [75]) prompts considerations of how to help the Mission Scientist when confronted by such a data torrent.

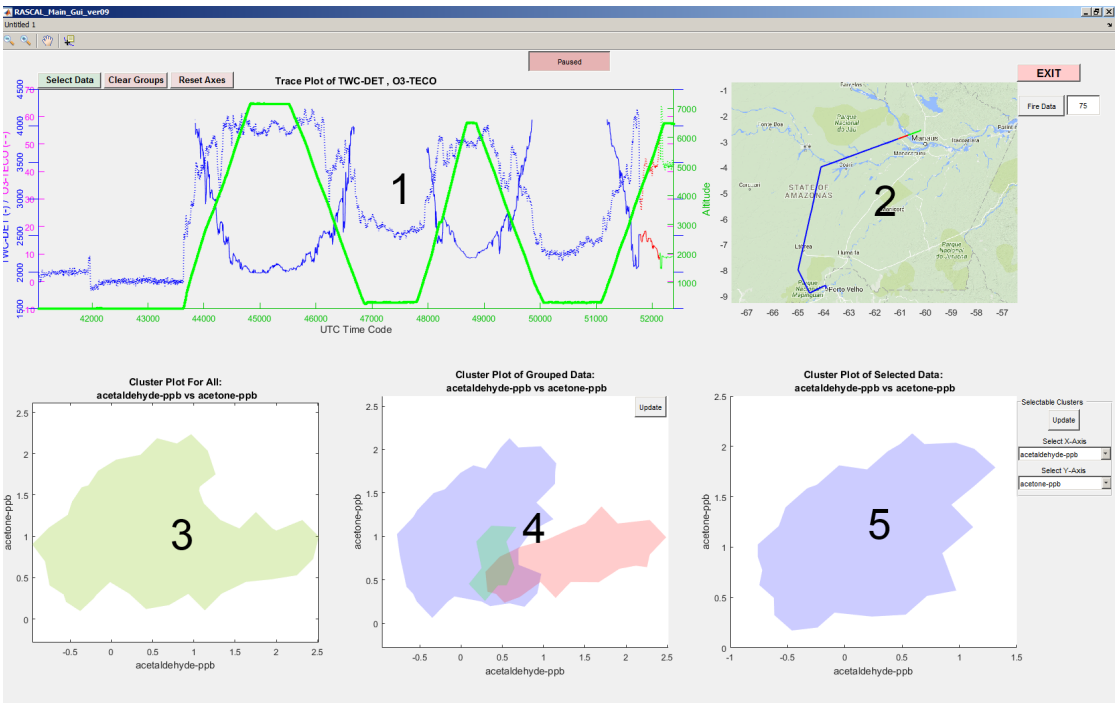


Fig. 6.1 RASCAL operating screen showing (1) trace plot, (2) map plot, (3) online clustering and (4),(5) offline clustering.

Real-time Atmospheric Science Cluster AnaLysis (RASCAL) is a demonstration of just such an aid. There are two versions, utilising the different, compatible online and offline clustering techniques developed in the earlier chapters. 'RASCAL' is an online version to demonstrate the functionality as appropriate for use in-flight, in real time, and there is an accompanying offline version 'RASCAL Offline' which allows for reproducing the same analysis offline, post-flight. The main text of this chapter will deal with the on-line version, followed by a separate Subsection on the offline version. The offline version was developed after feedback from atmospheric scientists on the online version and incorporates many upgrades to its user interface, although the underlying functionality of the clustering is unchanged except for, where appropriate, the use of offline clustering techniques.

This chapter introduces a software framework to enable Mission Scientists to explore data streams in real time during sorties. This framework assists in-flight identification of data of interest and immediate feedback for adapting the current flight path or for future flight planning. The basic analysis provided at this stage runs in real time (>1Hz sample rate) without parallelization for multiple processor cores and the computational and memory efficiency ensures that there is potential for future inclusion of basic on-line chemical modelling and/ or data analysis to provide further contextual information. The techniques demonstrated can function with any available chemistry or flight data,

although some may require basic pre-processing such as normalization, from a pre-determined range, to scale different measurement scales appropriately. These data streams are usually selected pre-flight but a combination of the techniques in this thesis to allow full in-flight flexibility will be discussed.

Figure 6.1 show the main RASCAL data visualizations and these are:

1. Trace Plots: real-time trace plots of a number of data streams, typically these would include flight altitude, ozone, water vapour but any available data stream can be included. Groups of data can be selected by the user in this plot.

2. Map Plot: a map window provides a basic map, linked from Google Earth, that can be panned and zoomed and can be overlaid with additional data. The demonstration includes:

   (a) Flight Path: a real-time plot of the flight path, coloured according to data group (see 1)

   (b) Fire Data: This can be either near real time data or non-live archive data as used here downloaded from the EOSDIS FIRMS website, [117]. The data is used in the GIS Shape file format, [52], and is imported at run time. Visibility of this data is selectable and can be filtered by the confidence level contained within the shape file as to whether it is a genuine fire as opposed to another type of heat source. The MODIS FIRMS approach is described in [65, 64]

3. On-Line Cluster Plot: On-line clustering of pre-selected data streams is performed using the CODAS clustering algorithm, [81], and the results plotted.

4. Off-line Clustering of the pre-selected data streams is performed using a modified DDC clustering algorithm, [80]. The results are plotted and, if data groups are selected (see 1), then clusters are coloured to match the selected groups.

5. Off-Line clustering of selectable data streams. Off-line clustering is performed on data streams selected from drop down menus and the results displayed. Any available data stream can be selected. If data groups are selected (see 1) then clusters are coloured to match the selected groups.

The rationale for the outputs described above is discussed through the rest of this chapter, which is structured as follows. Section 6.1.1 introduces the key terms used throughout this chapter. Section 6.2 provides an overview of clustering techniques and Alpha Hulls, while Section 6.2.1 describes the specific techniques used in RASCAL. An overview of the use of RASCAL is given in Appendix F, and the methodology used to demonstrate the software is given in Section 6.4. Examples of typical in-flight

information that could be gleaned from RASCAL are given in Section 6.5. The discussion in Section 6.6, considers the benefits of the current RASCAL implementations while the Future Work in Section 6.8 considers some of the possibilities for the future of this type of software aid. Finally we summarise the work to date, its limitations and future possibilities in Section 6.9, Conclusions.

6.1.1 Terminology

1. Age: the age of a cluster is the time (i.e. number of data ingestions) since the cluster was last updated. Clusters which have not been updated for some period, determined by parameters in the ageing algorithm, can be winnowed out of the cluster list. Ageing is, therefore, a useful way of preventing the number of cluster growing indefinitely over time as well as providing a diagnostic of change in cluster patterning over time.

2. Clusters: Groups of data created by their similarity either directly to each other, or by being linked through a chain of data with local similarity. The techniques discussed here are CODAS, CEDAS and DDCAS as described in the earlier Chapters 5.2, 5.3 and 4.4 respectively. Although this has been defined earlier it is repeated here to clarify the difference from data groups.

3. Data Groups: Groups of data defined by the user, typically through selection of data in any of the visualization windows. Data Groups are therefore distinct from clusters by virtue of being user-defined rather than algorithm-defined. When examining data that is similar in nature, and therefore forms a cluster, it may be useful to separate these data into groups by reason of spatial, temporal, or other separation known to, or observed by, the user.

4. Alpha Hull: first defined by [48], alpha hulls are a technique for creating an area enclosing a set of points whereby the minimum distance between any two points is a function of α. The α parameter allows for concave shaped regions by providing a lower limit on the internal chord length between samples.

6.2 Clustering and Visualisation Techniques Used in RASCAL

Although an overview of clustering has been given previously this section provides both a re-cap and more detailed analysis of clustering for use in the RASCAL software. The

analysis of the requirements here is much more focussed on Atmospheric Science and how the techniques can be applied together with the benefits they can bring.

Cluster analysis is the grouping of similar data based on the data alone with no user defined limits to each cluster dimension such as we find with classification techniques. There are a number of different underlying methods typically based on data-space distance measures [105, 163, 108], density [80, 53, 37], distribution [57] etc. and many further developments of these exist. Some of these techniques discover the clusters, some force the data into a set number of clusters. Some are based on assumptions of cluster shape, hyper-ellipses being the most common, while others will discover arbitrarily-shaped groups. In most cases the cluster technique will group the data samples provided and provide these data groups as the results. However, with discrete sampling of a continuous system, such as that found in atmospheric science, it could be advantageous to provide the results as 'cluster regions' describing where the data between the samples may also lie in the data space, i.e. a single data sample is taken to represent the data in a local region, however the data in that region is unlikely to be identical and so a region of data space is used to allow for such local variations as may occur. For techniques that provide groups of data as results this would require an additional layer of calculations to provide the regions surrounding the data.

Clustering has been used as a data mining tool in atmospheric science in areas such as clouds [70], ozone measurements [103], classification of climate regions [168] and investigations of climate station siting [139] all of which used hyper-ellipsoidal off-line clustering of data.

6.2.1 RASCAL Clustering Requirements

This subsection considers the key features required of the clustering techniques to be used in the RASCAL software and describes the reasoning behind their choice.

Arbitrarily Shaped Clusters

The nature of the atmospheric data suggests that it is likely to form arbitrarily shaped clusters rather than hyper-ellipsoids, particularly when flight patterns move through different air parcel compositions, or if we include spatial co-ordinate data. Consider such atmospheric composition changes as may result from a pollution source or local variation in land use type. Initially we have a hyper-elliptical pocket (or air parcel) of distinctly unusual chemistry surrounded by 'standard' chemistry for the region. Atmospheric stirring and mixing will spread the pollution unevenly in physical space due to local air currents giving an outer region of normal air, and inner region of polluted air and a meandering border region chemistry between the two. Over time these variations

in atmospheric chemistry will spatially drift in physical space forming 3-dimensional shapes (e.g. [36], Section 2.6). Furthermore, during these movements in physical space the polluted air will come into contact with surrounding air of different composition from the initial surrounding air resulting in new types of chemistry at the interface (e.g. [148]). One can make a similar argument for the deformation of clusters in chemical-composition-space, as a result of the non-linear chemistry occurring within the 'pocket'. Even when the chemistry alone, without spatial data, forms hyper-elliptical clusters under normal conditions the evolution of one or more chemical species may alter the readings to produce distinctly non-hyper-elliptical cluster shapes. Figure 3.1 shows illustrative examples of the differences between clusters results from common types of clustering methods. In some cases it would be possible to improve the results somewhat by careful, application specific, tuning of the ellipse radii however, this requires a priori knowledge of the expected clustering. The difficulties associated with using distance based measures, and therefore hyper-ellipses, to represent non-hyper-elliptical shapes will remain.

Common distance based clustering techniques such as hierarchical [109] or subtractive are not only unsuitable due to the nature of the clusters, but also due to the off-line method of calculation. Clustering methods that provide for arbitrarily shaped clusters fall into two broad categories; techniques such as Chameleon, [89], and DBScan, [53], are limited to offline use and have limitations when used with high dimensional data, [6]; online techniques for arbitrarily shaped clusters such as DenStream [26], CluStream [3] or MR-Stream, [155], are actually hybrid online/ offline techniques where 1st stage micro-clustering is carried out online, but final clustering is offline and on-demand, effectively restricting their use for constant monitoring of data streams.

Pre-Determined Numbers of Clusters

Setting aside the issue of cluster shape, unless the chemical sources are known and the chemistry well understood it is impossible to predict the number of clusters that might exist at any point in time or in any locality. This negates the use of clustering techniques which require a pre-defined number of clusters. They can either force data that should be in separate clusters to be assigned to a cluster to which the data do not naturally belong to or it can force clusters to divide unnecessarily, rendering the results unreliable. Therefore any clustering technique that require a pre-determined number of clusters such as k-means and its variants, [105] can be discounted.

On-Line Versus Off-Line Clustering

To deliver useful information to the Mission Scientist in real time the clustering technique is required to act on data as it is sampled so we require an on-line technique. Few fully

online techniques exist and, of those that do, most deliver hyper-elliptical cluster shapes, [3, 15, 26, 47]. On-line techniques for arbitrary shaped clusters are limited to incremental versions of DBScan or grid based methods such as MR-Stream, [155], or other multi-stage techniques [26]. Two-, or multi-, stage techniques typically maintain the 1st stage micro-clusters on-line, but the final stage clusters are produced off-line and on demand. The final off-line stage is too computationally expensive to occur continually and needs to be carried out at regular, pre-determined intervals or under a prompt from the Mission Scientist.

To complement the on-line clustering being displayed the Mission Scientists will be able to carry out off-line clustering of alternative sets of data. To provide similarity between the results there is a need to ensure compatibility between the on-line and off-line clustering techniques, i.e. they should be based on the same principles and produce results in the same format.

To achieve all these requirements the new techniques presented in this thesis were developed to achieve the following aims:

1. Low dimensional complexity
2. Low memory requirements
3. Low calculation complexity
4. Online and Offline capability
5. Similarity of clusters between on-line and off-line techniques
6. Compatibility between on-line and off-line cluster descriptions

6.2.2 Advantages Gained by Clustering

Clustering has a number of advantages for online data analysis of Atmospheric Science data streams; (i) it provides summary information of the data; (ii) allows easy visualization; (iii) aids the simplification of subsequent calculations; and (iv) identifies outliers and anomalies.

Summary Information

In any set of measurements of a dynamical, non-stationary system the measurements will be at discrete intervals. As such, they are point representations of a continuous value. Had the measurements been taken at a slightly different time the results would have varied slightly. It is not unreasonable to assume that measurements within a certain range could all have been possible. Using clustering we can summarise a region of data

space within which any of the possible measurements would lie. Such data-space regions need not be data-space-filling: clustering using arbitrary cluster shapes can generate fractal-like clusters, [82]. Outliers are automatically displayed as separate clusters, or may be removed.

Visualization

Having clustered the data into a region within which subsequent data samples are expected to be found it is a simple exercise to visualize this region. We can create a coloured, transparent 'patch' for each cluster providing easier to interpret visualisations. This is best used in 2D data space although it has some advantages in 3D as well. The two dimensions can be individual data dimensions or composites, similar to those in principal component analysis. Beyond two or three dimensions alternative visualization techniques are required or the use of automated cluster interpretation techniques may be necessary.

Calculation Simplicity

By clustering data samples into separate groups the data can be represented by a reduced number of points, or even a single point to represent a much greater number. This has clear implications for any future additional chemistry analysis that may be implemented, see Section 6.8.10. Reducing the number of calculations can simplify analysis to a level achievable by computer, or even expert user, in real time.

Identification of Outliers and Anomalies

Clustering of data into groups of similar data also has the inverse effect of identifying data that are either in small outlier groups or are single points. Outlier identification is useful in two key areas: identification of invalid data due to instrument failure or readings during calibration; and, importantly, identification of data that does not fit the typical profile of the rest of the data. The use of arbitrarily shaped clustering also allows the identification of unusual drift and shift in data as the cluster shape alters. Using summary information to speed up data analysis allows for faster exploratory analysis, rather than relying exclusively on posterior hypothesis-driven analysis [42].

6.3 Clustering Techniques Used in RASCAL

6.3.1 Data Density based Clustering (DDC)

Data Density based Clustering is described in detail in Chapter 4.2 and is a technique designed for computational and memory efficiency. Two implementations have been developed, Data Density based Clustering (DDC), which is utilised here, and Data Density based Clustering for Arbitrary Shapes (DDCAS), which has been developed for future inclusion. In the DDC implementation here the clusters are hyper-circular, the cluster information consisting of one cluster centre and a radius. All the data in a group are considered to define a macro-cluster allowing separate of groups of data to be displayed with clarity. The technique is off-line and requires an initialisation parameter, known as the initial radius r_0, and defined in the initialisation screen. The technique is robust to variations in r_0 due to the elimination of outliers and the in-process adaptation of the radius to the actual data spread.

DDCAS, Chapter 4.4, is a two stage technique built on a simplified DDC in which this simplified DDC creates hyper-spherical micro-clusters which are then joined in a secondary process to form the macro-clusters of arbitrary shape. DDCAS macro-clusters consist of a list of connected micro-clusters each of which is defined by a centre and a common radius. This technique is also off-line and uses the same initial radius, r_0, parameter. DDCAS is also robust to variations in r_0 due to the elimination of outliers and the in-process adaptation of the radius to the actual data spread. DDCAS has been developed to allow clustering of historical data to be followed by on-going on-line clustering using CODAS or CEDAS, however, this has not been implemented at this stage.

6.3.2 Clustering of Online Data-streams in Arbitrary Shapes (CODAS)

Clustering of Online Data streams in Arbitrary Shapes (CODAS), detailed in Section 5.2, was developed to address the typically exponential complexity of on-line clustering algorithms to multi-dimensional data. CODAS is a two stage algorithm, the first creating hyper-spherical micro-clusters, the second joining these micro-clusters to form macro-clusters. In its full form the algorithm requires two data based user parameters, an initial radius r_0 and a minimum number of data samples T_{min} that must lie within that radius for a micro-cluster to form. In practice in RASCAL we typically use a value of 1 for T_{min} to include all possible micro-clusters.

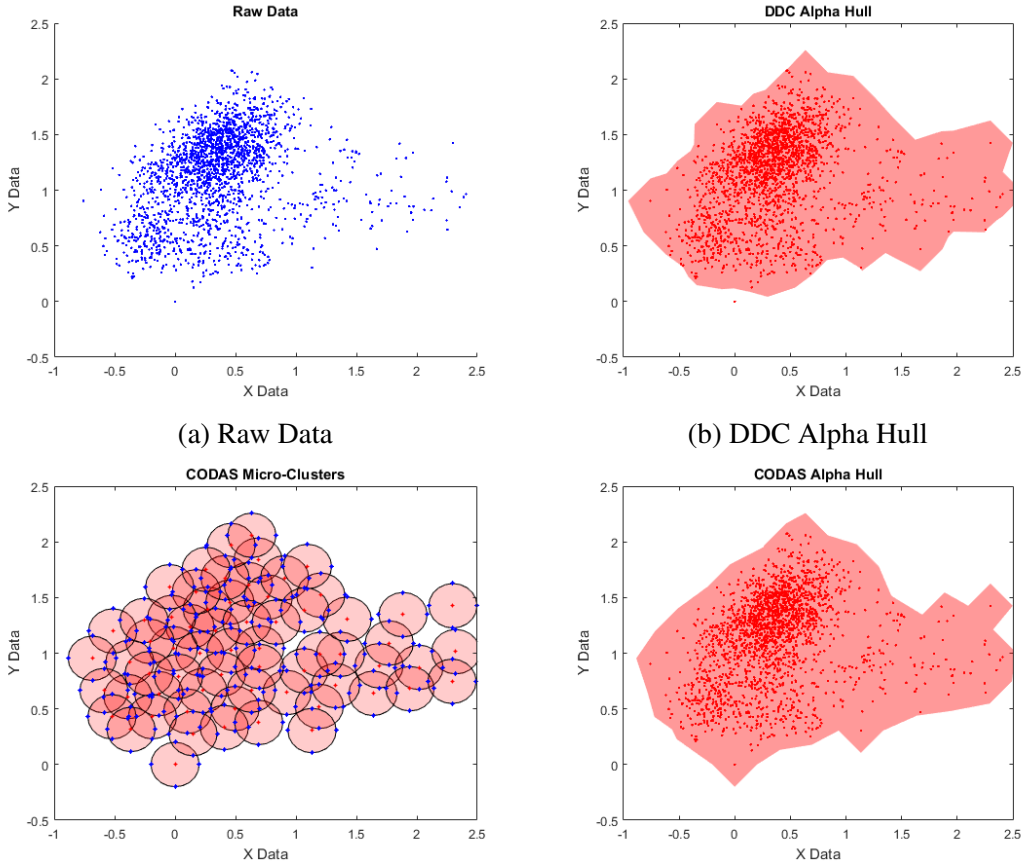
As further data arrives in the data stream the samples are either added to existing micro-clusters, or are available to form new micro-clusters. If the data is added to a current micro-cluster the micro-cluster centre may adjust to better represent the data contained within it. In CODAS the micro-clusters do not age and remain throughout the analysis providing a historical record of data from the initial stages of the flight.

6.3.3 Clustering of Evolving Data-streams in Arbitrary Shapes (CEDAS)

Clustering of Evolving Data streams into Arbitrary Shapes (CEDAS), detailed in Section 5.3, was developed for use over extended time periods. In its full form the algorithm requires two data based user parameters, an initial radius r_0 and a minimum number of data samples m_0 that must lie within that radius for a micro-cluster to be considered for merging into a macro-cluster. With extended flight times potentially extending over multiple days, details such as repetitive temporal changes in data may not be apparent if the data falls in the same data space as previous time periods. The ability to 'decay' the importance of older data therefore becomes significant to allow visualization of recent data. The CEDAS algorithm is interchangeable with CODAS receiving data and returning results in an identical format. For the demonstration in this Chapter CODAS is used due to the short term nature of the flight data used.

6.3.4 Alpha Hulls

The cluster techniques used in RASCAL provide regions within which the data have been observed. In the case of two stage clustering such as CODAS we have a number of micro-cluster centres and radii. When plotting these for visualisation the micro-clusters are coloured according to the macro-cluster they form. Transparency is used to aid visualisation where clusters from different data groups overlap some of the same data space. Plotting these micro-clusters achieves a display similar to that shown in Figure 6.2c. This can be confusing to the eye, especially when multiple clusters are shown in different colours and shades. To overcome this a technique known as Alpha Hulls [48] is used to encompass the micro-clusters. Alpha hulls are distinguished from convex hulls [71] by an additional 'alpha' value which relates to the minimum internal radius of the hull edge.



(a) Raw Data

(b) DDC Alpha Hull

(c) CODAS Micro-Clusters

(d) CODAS Alpha Hull

Fig. 6.2 Plots of the visualization of cluster stages for DDC and CODAS. In plots b and d the data samples have been included to illustrate the alpha hull fit, these are not normally displayed. In on-line mode this data is no longer available.

6.4 Methodology

To test the use of the RASCAL software it is applied to a dataset from the South AMerican Biomass Burning Analysis (SAMBBA) campaign, [35, 110]. The SAMBBA data sets includes readings from a number of core and non-core instruments and numerous aircraft flight parameters, [23]. For the purposes of demonstrating the RASCAL software the investigations will be limited to flight B735 as the O_3 readings show some unusual spikes suitable for online investigation.

Also merged in the data stream are model data for CO, $Isoprene$, NO, NO_2, and O_3 from CiTTyCat, [132], initialised from the UKCA global chemistry-climate model into the data set. The model output was at 1 minute intervals at $3.75^o x 2.5^o$ resolution and we used linear interpolation between values to provide data of the same time resolution as the data stream.

These data allow us to demonstrate cases of:

1. Identification of data of interest versus data not of interest.
2. Use of clustering results to identify anomalies.
3. Use of model data output to identify data of interest.

and these are detailed in the following Chapter 6.5.

6.5 Using RASCAL to Investigate Atmospheric Science Data In Flight

The screen captures used to illustrate the following discussion have been created during a normal run of the software and can be easily reproduced, no special prior knowledge is required.

6.5.1 Identification of Data of Interest

There are two aspects to identifying data that may be of interest for adapting a flight path, or for identifying data for analysis after the campaign. Firstly separating data that is truly anomalous from that which appears anomalous but is not and, secondly, identifying multiple regions of data that have similar anomalous characteristics. These are illustrated in figure 6.3.

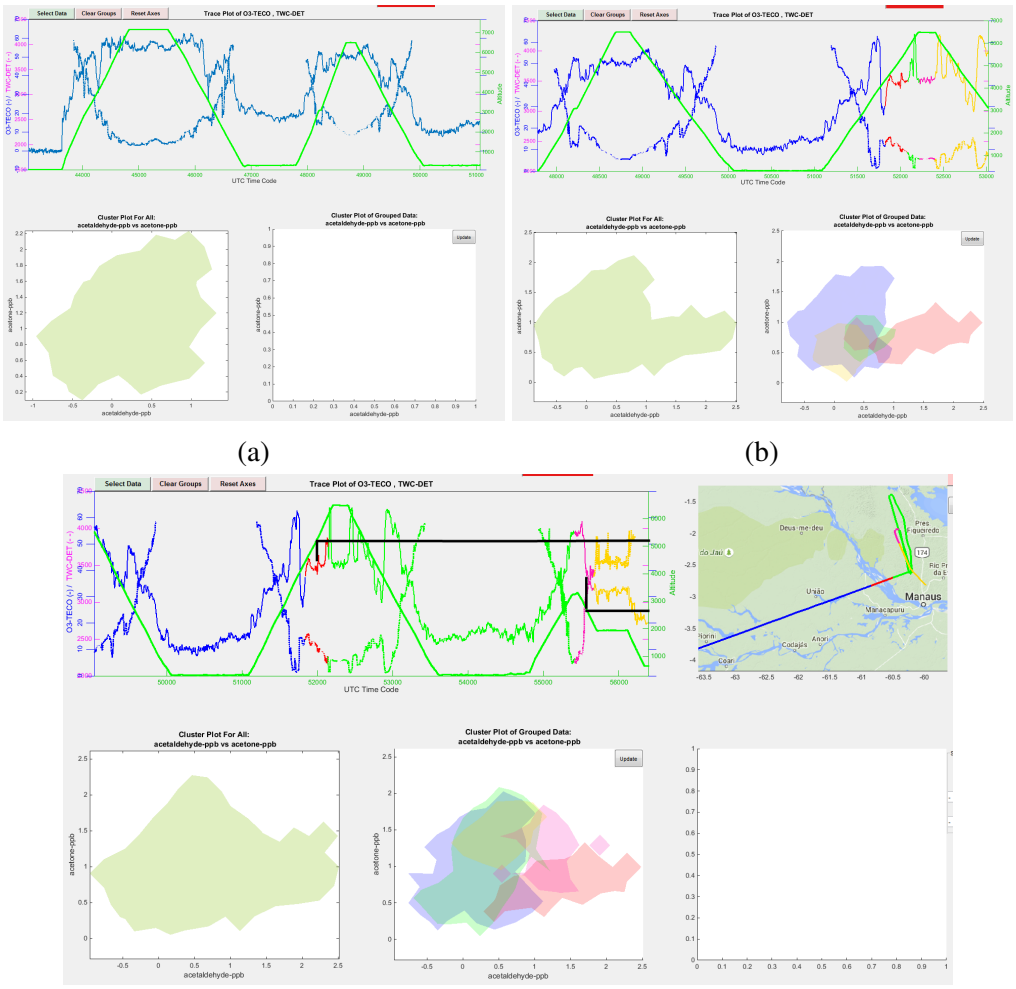


Fig. 6.3 RASCAL screen views showing (6.3a) The early part of the flight with no significant anomalies and measurements falling with a range of standard values. (6.3b) Two data regions in similar locations relative to O_3 spikes. The red data shows abnormal levels of acetaldehyde in the cluster plot. (6.3c) Two data regions with abnormal acetaldehyde levels in the cluster plot. We also see their location on the flight path and the altitudes marked in black on the trace plot.

Using RASCAL Clustering for Differentiating True Anomalies

Figure 6.3a shows the data with no significant anomalies. The cluster plot (lower left) shows a consistent data region within which all the samples so far have been measured.

Later in the flight this cluster plot starts to show an unusual bulge as seen in figure 6.3b. The cluster plot is automatically re-scaled to include all the data in the plot and a significant 'tail' of data can be seen bulging out to the right. By selecting data in the trace screen the data shown in red can be identified and shown to be consecutive. Looking at the trace plot (top) this data can be seen after a significant spike in O_3. Immediately after the next spike and before the following one is a similar region coloured magenta. However, when the group cluster plot (lower right) is examined it can be seen that whereas the red region constitutes the bulge, the magenta region has measurements in the normal region.

Using RASCAL Clustering to Identify Multiple Anomalies of Interest

Still later in the flight it is noticed that a second bulge begins to appear, also related to an O_3 spike as shown in figure 6.3c. This time the acetone values are higher than the previous bulge, however, they remain consistent with normal values and it is the acetaldehyde measurements that increase. In this image the two regions are identified as red and magenta. These data values are shown in the lower-centre cluster plot. This time the map view of the flight path is also shown and the two data regions can be seen separated in time, however, due to the flight path they are somewhat physically closer than might be expected. The trace plot, where the altitudes are indicated by the black lines, shows that the magenta region is at a lower altitude. This is consistent with a plume of air, rising in height and drifting geographically with a slow loss in, or mixing out of, acetone as it does so.

Using Selectable Data Clustering for Additional Information

Having identified multiple regions of anomalous data further exploration of these regions can be carried out using DDC clustering on alternative selected data streams. Using the drop down menus to the right side of the lower-right plot, see figure 6.4, DDC is performed on any pair of data sets. Figure 6.4 shows the resulting plot from selecting Acetaldehyde and MVK-MACR (i.e., the signal derived from proton-reaction mass spectrometry corresponding to methyl vinyl ketone and methacrolein, first-generation reaction products of the biogenic hydrocarbon, isoprene). In the cluster plot on the lower right it is seen that both the regions have raised levels of both MVK-MACR and

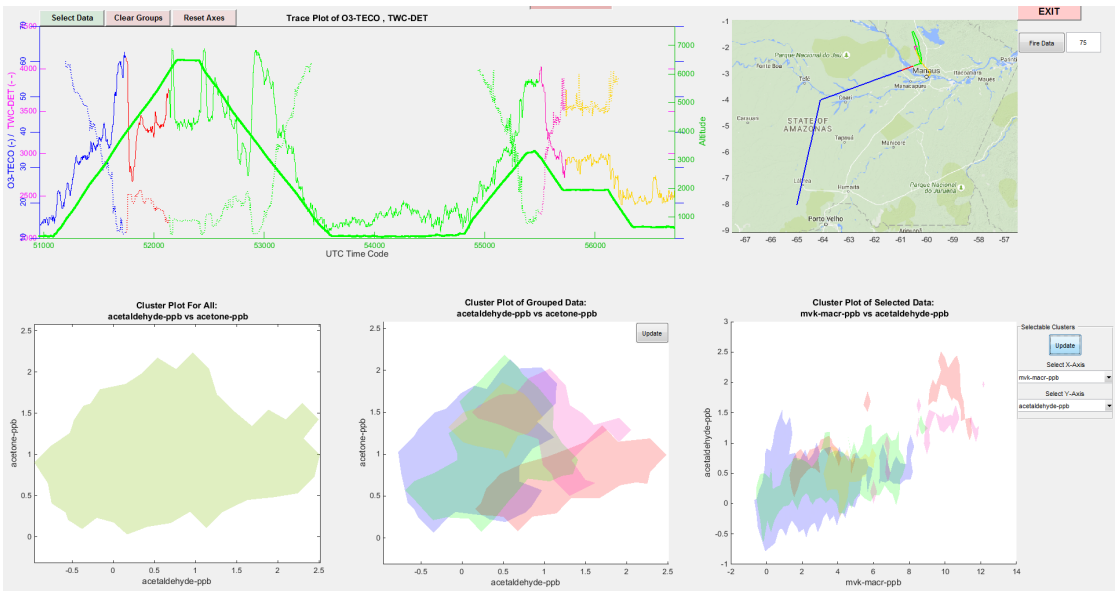


Fig. 6.4 Using selectable data-stream clustering to explore data streams. By selecting suitable data streams from the drop down menus we can apply clustering to MVK-MACR and Acetaldehyde. The display shows that the two selected data regions, red and magenta, both have raised levels of MVK-MACR and Acetaldehyde.

acetaldehyde. These parameters are known to have a correlation with biomass burning, [166, 78, 39].

It can also be seen from the flight path that the magenta region is at lower altitude and appears to be geographically narrower than the higher altitude red region. These data appear consistent, therefore, with the spread of pollutants as a biomass burning plume rises and drifts.

6.5.2 Using Model Outputs to Identify Data of Interest

In figure 6.5 the output from the CiTTyCAT Lagrangian model ([96, 132]), has been used to demonstrate the use of RASCAL with model data. The trace plots are used to compare the CiTTyCat model output, dashed line, with the actual instrument reading, solid line, for O_3 and indicate three sets of data for further investigation. The solid green line is the flight altitude.

Red Data Region

Where the data is highlighted in red, the model outputs bears little relation to the actual readings. Whereas, generally, throughout the flight the model values rise and fall in approximate synchronization with the readings, in the red zone the model predicts a large drop in O_3 despite the rising altitude. Overall, throughout the flight there is a general

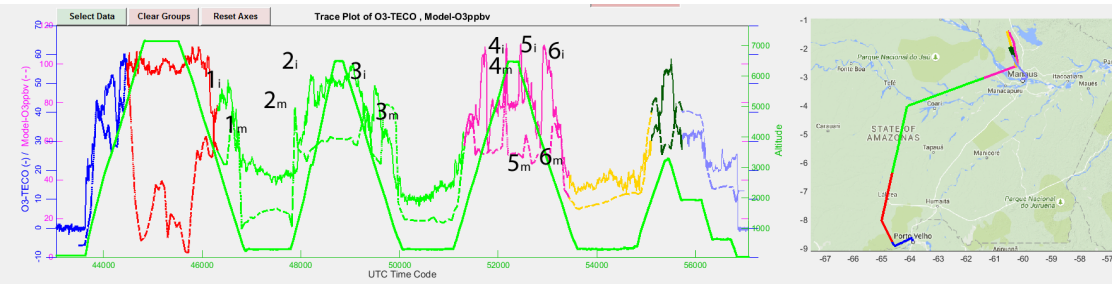


Fig. 6.5 Identification of Further Regions of Interest: RASCAL screen view showing model data comparison where the regions in red and black indicate where the model prediction is incorrectly predicting dips in O_3 and magenta where the model accurately predicted narrow spikes in O_3

tendency for the model output to become closer to reality suggesting the red anomaly may be caused by the initial model parameters. CiTTyCAT is a research modelling tool; for operational mode it would be a straightforward extension of the software to provide on-the-fly skill metrics.

Numbered Spikes

Where the data is indicated with numerical values the model (subscript 'm ') has predicted spikes in O_3 approximately the matching instrument readings (subscript 'i '). The region highlighted in magenta (labels 4-6) shows spikes with a good temporal match to the readings and, in one case a good match in magnitude.

Dark Green Data Region

In the dark green data region, from time code $\approx 55,000$, the model is predicting a drop in O_3 where the actual readings show a spike. Other than this, the model is still providing a reasonable match to the overall profile of the instruments. Although there is some temporal shift for most of the spikes, and the scaling and the amplitude of the peaks may be different, the general shape of the profile is similar from the red region up until this point. This indicates some data worthy of investigation as it may be, for example, an unknown, unexpected cause for the actual chemistry change, or some accumulated error in the model output.

6.6 Discussion

RASCAL is a core basis on which to build future on-line analysis tools. This Chapter has demonstrated how basic clustering can help identify data of specific interest for further

analysis. This can be carried out in-flight or post flight. Clustering groups of similar data together can reduce the amount of data required for analysis by representing the data in a more concise manner, simplifying and speeding up the analysis.

Sections 6.5.1 demonstrated how RASCAL could aid atmospheric scientists in identifying anomalies of interest and also in finding multiple groups of data with similar anomalies. Identifying this data in real time has a number of uses including possible flight re-routing and for indicating data to analyse prior to planning the next flight.

It can be argued that the data of most interest are those which are the result of unexpected anomalies. Data which behaves as expected may reinforce current thinking and analysis but is unlikely to add new insight. Data which is unexpected however may provide information key to new understanding or to improve current knowledge. With current post campaign analysis the available data relating to these anomalies may be limited if they are not identified and explored in-flight. With on-line techniques indicating when abnormal data is encountered, re-routing, or appropriate planning for the next flight could enable additional data to be gathered for detailed analysis later.

Section 6.5 also demonstrates how RASCAL can go beyond finding anomalous data and also provide deeper insight into the possible causes of these anomalies by clustering data from other chemical species.

It has been demonstrated how RASCAL can help to identify further instances of similar anomalies. Identification of these anomalies has demonstrated how the path of possible plumes of pollution from biomass burning can be traced. Having traced the path of the plume at two altitudes it is possible to narrow down locations for the plume at higher altitudes thus aiding future flight planning. It may even be possible, using arbitrarily shaped clustering, to make first estimates of atmospheric dispersion characteristics based on the macroscopic shape and/ or fractality of the cluster in physical space.

Section 6.5.2 illustrates how RASCAL can be used in conjunction with model data to compare the model outputs with actual measurements. How well the model predicts the real data can be seen in real time providing rapid feedback to modellers and strengthening the rationale for their participation in field intensives. Again, this can be used to guide current and future flight plans to gather more data in regions that are of most interest, whether this is where the model is good or poor, thus providing more useful data for post flight analysis.

6.7 Development of RASCAL for Offline Cluster Analysis

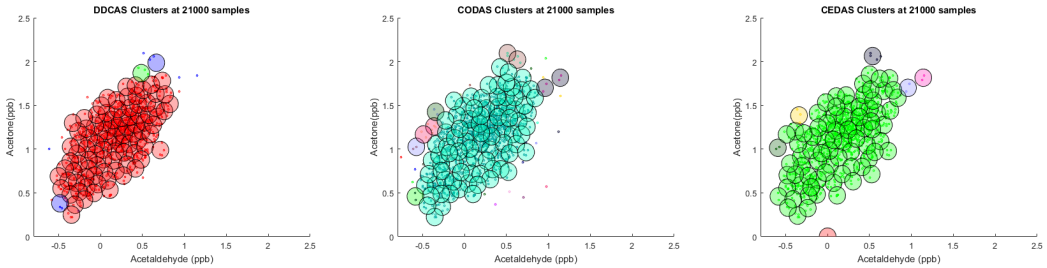
The version of RASCAL described in the preceding sections is designed for online use during data gathering missions. In practice RASCAL exceeds the required data sampling rate however it processes the data sequentially on arrival. When working with large datasets such sequential techniques become prohibitive for use in an offline environment. In consideration of the need to reproduce the same clustering analysis results offline and alternative software has been developed, known as RASCAL Offline. In this version, rather than processing the data sequentially, the whole data set is loaded and processed at once.

While much of the processing such as the trace plots, or mapping and flight path information display remains the same, the clustering algorithms are required to function offline, on the complete dataset. The proposed offline version of RASCAL therefore requires that the offline clustering algorithms produce similar results to the online algorithms they replace. DDCAS (Chapter 4.4), when used with a fixed radius, should produce similar results to both CODAS and CEDAS and we investigate that in this section.

To demonstrate the similarities the SAMBBA B735 data is used to produce plots of the clustering results for each technique at the key anomaly identification stages detailed earlier. Stage 1 approximates typical data for the flight, Stage 2 after identification of the first anomaly and Stage 3 at the end of the flight, including the second anomaly. Sample plots for each technique are shown in Figure 6.6 for visual comparison of the output. Figure 6.7 shows alpha hulls covering the data assigned to the largest cluster. Visually they are similar at each stage and to measure the similarity of the results they are analysed using various standard properties and shape factors of the enclosed data space. The properties and shape factors are given in Table 6.1. The similarity of the centres of the hulls indicate a similar location in the data space while the other properties indicate the shape similarities.

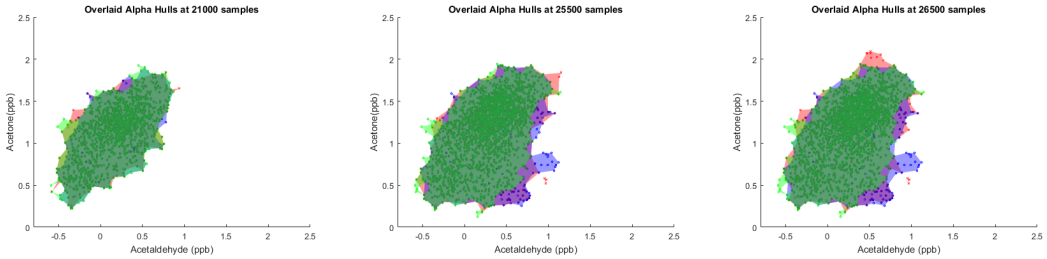
With such similarities of results DDCAS is used to replace CODAS and CEDAS in the RASCAL Offline software to allow rapid reproduction of the analysis offline. This allows the mission scientists to produce additional analyses of data of interest that may not have occurred in flight and also to reproduce plots and visualization for presentation and discussion.

The similarity of the both the results and the data structure of the techniques is also an indication that DDCAS is compatible as a first pass technique to cluster historical data. These results can then be acted on directly by CODAS or CEDAS for continued online analysis.



(a) DDCAS Stage 1 (b) CODAS Stage 1 (c) CEDAS Stage 1

Fig. 6.6 Clustering technique outputs of the B735 flight showing clustering of typical data.



(a) Stage 1 (b) Stage 2 (c) Stage 3

Fig. 6.7 Alpha hulls of the data assigned to the main cluster at the 3 stages of B735 analysis. Stage 1 shows typical data, stage 2 is after identification of the first anomaly and stage 3 at the end of the flight after discovery of the second anomaly. The three techniques are overlaid in each plot to indicate the similarity.

Table 6.1 Shape factor information to compare DDCAS, CODAS, CEDAS

Stage	Technique	Centre		Perimeter	Area	2nd moments I_2		Elongation	Compactness
1	DDCAS	0.172	1.142	5.842	1.264	0.089	0.097	0.913	1.937
	CODAS	0.173	1.136	5.512	1.249	0.090	0.104	0.865	1.806
	CEDAS	0.169	1.137	6.011	1.300	0.093	0.104	0.895	1.925
2	DDCAS	0.283	1.156	6.441	1.913	0.096	0.141	0.684	3.410
	CODAS	0.287	1.145	6.604	1.900	0.100	0.146	0.679	3.250
	CEDAS	0.268	1.158	6.906	1.721	0.098	0.141	0.693	2.642
3	DDCAS	0.286	1.161	6.716	1.914	0.096	0.142	0.674	3.398
	CODAS	0.288	1.146	6.604	1.900	0.100	0.145	0.692	3.262
	CEDAS	0.269	1.159	6.906	1.721	0.098	0.141	0.694	2.651

6.8 Future Developments

6.8.1 Off-Line RASCAL Analysis

Section 6.7 covers the basics for the offline version of RASCAL. The speed of analysis of offline data sets is not only useful for repeating previous in-flight analysis, but also for further data exploration. The offline version also provides a base for software development. Suggestions can be easily incorporated and the effects of additional software controls, analysis techniques or visualization methods .

6.8.2 Archiving Analysis Results

After carrying out analyses there is no ability to save the details such as the start and end time codes, chemistry used, flight location etc. Although screen captures, or pen and paper could be used to record this a single button should be available to save the current state to file. It could then be retrieved so post-flight analysis can more easily reproduce the same results. Feedback from the Atmospheric Science community on what and how best to record such data would be valuable.

6.8.3 Multi-Dimensional Cluster Display

Demonstrating the potential of RASCAL required a clear, easy to understand display of the clustering results. To achieve this 2 dimensional clusters are used. The clustering techniques used here are largely dimensionally independent, or exhibit linear complexity, [81, 82], meaning many more than 2 instrument data streams could be used for clustering. Displaying these clusters in an easy to understand format will require different visualization techniques [29, 99] or by the extraction of information for visualization [90].

6.8.4 Clustering of Evolving Data streams in Arbitrary Shapes

Implementing Clustering of Evolving Data streams in Arbitrary Shapes (CEDAS) [82] into RASCAL will provide additional options for online clustering. Although the anomalies illustrated above can be found with CODAS due to their different locations in data space CEDAS would also be able to find and visualise them due to their temporal difference. Thus CEDAS would add the ability to distinguish anomalies with the same data values, at different times during the flight.

6.8.5 Shading of Micro-Clusters by Data Density Instead of Alpha Hulls

Current visualisation of the data clusters is focused on the location of the data in data space only. Thus clustering and alpha hulls to 'draw round' the data are used. This hides any information relating to the amount of data within the alpha hull. Future work should investigate shading the alpha hull in relation to the amount of underlying data, similar to the diagram shown in Figure 6.2c.

6.8.6 Data Group Selection Improvements

Data selection is only available on the trace plot screen. This can make it difficult to find the relevant data, especially at a later date. Future versions should allow selection of the data in any of the plot windows. (This has been implemented in the offline version of RASCAL). For example, for the bulge in the cluster plot discussed in Section 6.5.1 it would be useful to select the bulge and see where the data lies on the trace plots and flight path.

6.8.7 On-Line / Off-Line Clustering Combination

CODAS provides on-line clustering of the data stream, DDC provides off-line clustering of all the data received so far. As it stands it is not possible to change the on-line clustering to an alternative set of data streams. If this were done the clustering would start at the time of change and would not include any earlier data so losing information for a period of time. Were DDCAS to be implemented here it would provide results in a format compatible with CODAS and CEDAS. A hybrid algorithm could be developed such that when the data streams are changed DDCAS is invoked to provide clustering of the relevant archived data followed by CODAS or CEDAS to provide on-line updating of these clusters.

6.8.8 Additional Map Data Overlay

Additional map overlays can be developed to add data from any applicable source that would be of interest. This could include such things as population areas, known pollution sources or different types of vegetation etc.

6.8.9 Live Data Feed and Integration

RASCAL currently loads a data file containing the instrument measurements. This data is fed to the RASCAL analysis routines one sample at a a time to simulate a live data feed. To work in the field a method of interfacing with the on board instruments is required. Liaison with the climate teams responsible for the instruments is necessary to develop a preferred technique and common interface between all the instruments and RASCAL.

6.8.10 On-Line Chemistry Analysis

The algorithms in RASCAL have been designed to minimise processing time in a typical PC and have not been optimised, or parallelized, for speed yet utilise around 20% of the total processing time available to a corei7 processor. Therefore there is spare computing capability which could be used for more detailed chemistry analysis. Development of a common interface between plug-in type sub-functions would allow independent development of such analyses.

6.9 Conclusions

This Chapter has demonstrated RASCAL software with on-line and off-line cluster analysis of data streams. It has illustrated some of the benefits of the techniques, demonstrating how they can improve data collection by using the RASCAL results to alter flight paths or to feed in to future flight plans. The historical nature of the plots throughout the flight and the repeatable cluster analysis, supports multiple Mission Scientists which could be important on extended sorties such as those of the Global Hawk (e.g. [85]). Handover is enabled by allowing a new shift to re-analyse previous data and also to display the geographical source of the data and to compare this with map overlays showing additional information from alternative databases, e.g. data from the FIRMS ([51, 117]) database can be used to indicate possible biomass burning sources. The techniques involved allow on-line analysis to inform Mission Scientists in real time of significant anomalies identifying data of interest for future detailed investigation (and false alarms that can be safely ignored).

The information from RASCAL can feed in to flight planning improving the nature and quality of data gathered during campaigns. By improving the data collection to maximise the data most relevant to the campaign target investigation, post-campaign analysis will be enhanced by the availability of a greater amount of more relevant data maximising the potential return on investment for campaigns.

The memory and computational efficiency of RASCAL indicate its suitability as a basic framework upon which future on-line atmospheric science analysis can be built. It is intended that future collaborations between computer scientists and atmospheric scientists will develop a suite of on-line techniques that can further enhance and improve data analysis in the 'Big Data' era for atmospheric science.

Chapter 7

Summary, Conclusions and Future Work

This chapter is divided into three sections. First, in 7.1, Research Summary, the research is summarised considering the initial research proposal, Advanced Analysis and Visualization Techniques for Atmospheric Science. Following this, the 'Conclusions ' Section 7.2 considers each clustering algorithm, and the RASCAL demonstration software indicating the progress and contributions made by each. Finally, Section 7.3, 'Future Work ', suggests possible directions for future research arising from the work in this thesis. This includes work outside of the area of initial research proposal.

7.1 Research Summary

This thesis has introduced a suite of novel clustering algorithms specifically targeted at aiding mission scientists on data gathering campaigns. The proposed algorithms address different aspects of the challenges faced and together provide the solutions demonstrated in the RASCAL software. Not all of the developed algorithms are directly applicable to the final solution, but were part of the research path and have specific uses of their own. The benefits of each can be summarised as follows:

- DDC is an offline, recursive density estimation based technique for discovering hyper-elliptical natural clusters. If the natural clusters are not hyper-elliptical, then they will be divided into hyper-elliptical parts. It requires a single parameter to operate, an initial estimation of the cluster axes dimensions. The dimensions of the final clusters are adjusted to best suit the data and it provides high purity and high accuracy results. It is capable of separating outliers into separate small clusters to aid identification of different types of anomaly.

- DDCAR is also an offline technique and uses a data density based technique for estimating the initial radius required for DDC. This removes any user interaction and results in a fully autonomous technique. The work described here is limited to hyper-elliptical clusters only.
- DDCAS is the third offline technique and extends the DDC algorithm to a two stage process that first discovers hyper-elliptical micro-clusters and then merges these to form macro-clusters. By adapting the micro-cluster radii the technique builds macro-clusters consisting of a chain of variously sized micro-clusters allowing arbitrarily shaped natural clusters to be accurately represented.
- CODAS is an online clustering algorithm that places data from data-streams into arbitrarily shaped clusters. This technique is a dynamic technique in that the clusters can move and adjust their size, however they do not fully evolve.
- CEDAS is a fully evolving, online clustering technique for clustering data into arbitrarily shaped clusters. CEDAS clusters are updated with every sample, rather than periodically as with hybrid online/ offline techniques. By using a decay period over which the data remains relevant the clusters are fully evolving, changing size, shape and position and being removed or created if required.

The algorithms have been tested on a range of applications and shown to be effective, particularly in respect to their intended role in the analysis of atmospheric science data streams.

The DDC, DDCAS, CODAS and CEDAS clustering algorithms have been implemented into the RASCAL software to demonstrate how they can aid missions scientists in discovering anomalies, specific targeted pollution and other variations in atmospheric chemistry. Applications such as this can contribute to improvement in flight planning, data collection, tracking of pollution plumes and the evaluation of climate models. Outside of atmospheric science, the new clustering algorithms present a novel contribution to the science of machine learning.

7.2 Conclusions

During the initial research the requirements for utilising clustering for online analysis of atmospheric data streams were proposed and these are detailed in chapter 3.2. As these developed it became apparent that no current single, or multiple clustering algorithms could achieve all the goals. The research then focussed on proposing novel algorithms which were designed, from the outset, to work together in a cohesive manner to create

a suite of clustering algorithms functioning in a similar manner, or based on similar underlying principles.

The previous section outlines the algorithms themselves, whereas this section highlights their novelty and contribution to the field of unsupervised learning and their application to different fields of research, and atmospheric science in particular.

7.2.1 Novel Offline Clustering Solutions

Data Density Based Clustering (DDC)

Data Density based Clustering (DDC) provides a high speed, density based clustering algorithm, requiring no a-priori knowledge of the number of natural clusters, and intuitive selection of its parameters given expert knowledge of the field of application. Such expert knowledge consist of simply deciding on the minimum data density that would be considered part of a cluster and not general background noise or outliers. DDC can be set to cluster all data, leaving outliers in 'small' cluster, or to leave all 'small clusters' as a set of outliers. This ease of use makes DDC a useful addition to the field of offline data clustering.

As evidenced by its ability to continue to generate clusters after the data from cluster one has been removed, DDC can work effectively on sub-sets of the full data set. Thus clustering could be effectively parallelized to work with big data by dividing the data space. This is easily demonstrated, however, in cases where the sub-divisions split a natural cluster work should be carried out on ways to effectively merge these. An easily parallelized, high speed, data driven clustering algorithm is also a novel addition to the clustering field, however fully implementing such an approach was not carried out and so this will appear in the 'future work' section in more detail.

Data Density Based Clustering for Arbitrary Shapes (DDCAS)

Data Density Based Clustering for Arbitrary Shapes (DDCAS) extends DDC to work with natural clusters of any shape. It consistently outperforms other, popular algorithms for arbitrarily shaped groupings, either by speed, accuracy, memory use or a combination of the three. User input is minimal, requiring only knowledge of the maximum data density of background noise, or outliers such that an expert would not consider that data part of a natural cluster or the minimum gap required between two clusters for them to be considered separate. This is fairly intuitive, with knowledge of the data source. DDCAS can also be used to cluster outliers either as a single group of all outliers, or as 'small clusters' of outliers. This generates additional information about the outlier data many other clustering algorithms do not find. The combination of intuitive use, speed,

accuracy, memory use and additional information available regarding outliers are a novel addition to the field of clustering of arbitrarily shaped clusters.

A further benefit of the DDCAS algorithm is that it produces micro-clusters and clusters in the same manner as CODAS and CEDAS, see later. CODAS and CEDAS are online algorithms and as such may be sensitive to the data order. In particular, online algorithms have no knowledge of historical data from before the algorithm is initiated. DDCAS can be used to prime CODAS and CEDAS using recent, historical data. This priming is much faster than, e.g. running the online algorithm over the historical data. Once primed, the online algorithms produce accurate cluster results immediately they are initiated. With large, or temporally extensive data sets this can reduce the time to results by orders of magnitude. This is an important, novel contribution allowing clustering to be achieved in data streams in a way not previously achievable.

Autonomous Clustering (DDCAR)

Data Density Based Clustering with Automated Radii (DDCAR) is a step to a fully autonomous, data driven clustering algorithm requiring no user input. This is the first algorithm to demonstrate full autonomy, albeit on hyper-elliptical clusters only at this time. Full automation of unsupervised learning is a major contribution to machine learning and this initial success should be developed further.

7.2.2 Novel Online Clustering Solutions

Dynamic Clustering (CODAS)

Clustering of Online Data-streams into Arbitrary Clusters (CODAS) is a fully online, data density based clustering algorithm. While many online algorithms for arbitrarily shaped clusters are, in fact, two stage algorithms consisting of online micro-clustering combined with offline macro-clustering, CODAS is fully online providing up to date clusters with each data sample from a dynamic data stream. It is this fully online state that creates the contribution to the field of online clustering.

Evolving Clustering (CEDAS)

Previous online methods utilize an online micro-structure update, but use offline macro-structure generation. Thus the results only reflect the true cluster state at the instant they are generated. Increasing the rate of macro-cluster generation in these algorithms slows them to a level that is impractical for many uses. Clustering of Evolving Data-streams into Arbitrary Clusters (CEDAS) provides the first known algorithm to provide full online clustering for arbitrarily shaped clusters.

7.2.3 Applications of Novel Clustering Algorithms

The primary focus of this thesis was the development of an online system for use in-flight during atmospheric science data gathering missions. This is demonstrated and discussed in earlier chapters. Although it remains to be tested in the field, the RASCAL software demo allows for in-flight detection on anomalous data to aid decision making, flight planning and to target specific data of interest. This could make a significant improvement to the quality, relevance and detail of data gathered on a per-flight and per-campaign level.

Consideration should also be given to possibilities of abuse of such a system whereby targeting specific data could be used to artificially skew results. As such research into the practical and ethical use of such a system should take place.

7.3 Future Work

The potential future work can be divided into three distinct sections, that of algorithm development, Section 7.3.1, that of atmospheric science, or other software applications directly related to the research proposal, Section 7.3.2 and other applications. Such alternative applications for future work include Autonomous Detection, Section 7.3.3, climate model comparison and 7.3.4 and complex system analysis, 7.3.5.

7.3.1 Algorithm Development

Autonomous Clustering

The DDCAR algorithm, in particular, has scope for future development. The research goals did not call for full autonomy and so the work did not move beyond the feasibility stage. Further work into better radius estimation techniques could yield improved results.

Online and Offline Compatibility

The implementations of DDCAS, CODAS and CEDAS are all designed to produce cluster results in the same format. If DDCAS is used with a fixed micro-cluster radius, such as that used for CODAS and CEDAS, then the resulting algorithm could be considered to be their offline equivalent. The techniques employed for micro-cluster definitions and the agglomeration of these into macro-clusters indicates that the cluster results are directly compatible. Future work could extend the work presented here to investigate the validity of the cross-over of these technique and how they offer new opportunities in mining data streams. In particular DDCAS could be used to quickly cluster historical data such that

online techniques can take over and continue updating the cluster results. This provide opportunities for rapidly moving between data streams.

Algorithm Parallel Processing

The offline algorithms work in such a way that suggest they are suitable for use in a highly parallel processing environment. Either division of the data followed by merging of the clusters, or possibly data space sub-division may be suitable. Similarly, the online algorithms may also function in a parallel processing environment with each data sample being processed by a different node into the micro-clusters and the macro-cluster structures also being processed in parallel. Parallel processing would provide a significant increase in processing speed, but presents its own challenges for future research.

Autonomous Decision Making

The applications considered here add value to the data streams to enhance the expert users decision making. These decisions are based on visual cues provided by visualization of the clusters. The information contained within these clustered could be considered for the suitability of autonomous decision making.

7.3.2 Software Application Development

RASCAL Software

Future work on the RASCAL software holds a number of possibilities. Improvements to the software itself with regard to feedback from mission scientists and other users. Implementation of the offline to online crossover just mentioned would provide considerable flexibility. Perhaps the most important development for future work is further testing across historical data sets together with testing of the software by mission scientists on data gathering missions, data streams from monitoring sites or other applications.

The techniques have been applied to a variety of atmospheric science ‘big data’, uncovering hard-to-find anomalies and subtle changes in atmospheric composition. Moving beyond these proofs-of-concept, there is the opportunity to automate such detections, allowing more complex analysis of very high-dimension, hard to visualize, relationships (i.e., 10s to 100s of chemical dimensions) in atmospheric chemistry.

The clustering algorithms, and the RASCAL software, have been designed for applications in atmospheric science in the context of this thesis. Future work should consider that the algorithms work on data streams and these could originate from many different sources. Applications of the work to other disciplines should be considered,

for example machine condition monitoring, object tracking or social and behavioural sciences. In addition there are applications in complex systems, discussed later.

7.3.3 Autonomous Risk Assessment

Investigation of the use of the cluster results for autonomous detection of changes in atmospheric composition, and of variance from predictive models for the assessment of environmental risk should be investigated. In the context of this thesis, human visible variations in cluster results are frequently seen indicating anomalies and significant or unexpected changes in the environment. Macro-cluster analysis in terms of standard shape factors are able to detect these changes within the limited scope tested here. Further research could result in fully autonomous detection of atmospheric changes and environmental risks.

Where the algorithms have been demonstrated in alternative scenarios, e.g. computer network intrusion detection, they may also provide for autonomous detection and early warning systems.

7.3.4 Alternative Application - Climate Model Comparison

During the course of researching this thesis some initial tests were done utilizing the various clustering algorithms to analyse the variations between climate models. A quick demonstration showed that clustering could be used to compare the models in a spatio-temporal manner not feasible before. This analysis could form the basis for improved model predictions by best model selection, or as an ensemble technique for improving results. This work has formed the basis for a pilot study under the Research on Changes of Variability and Environmental Risk (ReCoVER) project for which funding has been awarded.

7.3.5 Alternative Application - Complex System Analysis

The two stage techniques, DDCAS, CODAS and CEDAS provide additional information on the data that has yet to be explored. In particular the graph theory utilised in CEDAS provides insights into the underlying complex dynamical system. This thesis only begins to scratch the surface of the potential in the techniques. By storing cluster information in the form of graphs of micro-clusters, the rich literatures in graph theory and fractal geometry can be applied. The edge information, the number of samples within the micro-clusters, and the hyper-dimensional space-filling behaviour of the clusters may all be used to uncover hidden properties of the data streams. These investigations make possible comparisons across different models as well as comparing them with measurement data.

This same, climate data based argument can also be applied to streams of data from any complex system and may have a much wider range of applications suitable for investigation.

Papers Published and Submitted

This lists the academic papers published, submitted and arising, but not yet written based on the work in this thesis. An estimation of the contribution of each author is provided.

Published Papers

Journal Papers

R. Hyde, P. Angelov, and A. R. MacKenzie, "Fully online clustering of evolving data streams into arbitrarily shaped clusters," Inf. Sci. (Ny)., vol. 382–383, pp. 96–114, 2017.

Initial concept, design of experimental work, algorithm and test software, experimental work carried out by Richard Hyde. Checking of experimental results was the work of all three authors. The first draft paper was authored by Richard Hyde, with contributions to the atmospheric science by Rob MacKenzie. The final paper was the work of all three authors.

N. R. P. Harris, L. J. Carpenter, J. D. Lee, G. Vaughan, M. T. Filus, R. L. Jones, B. OuYang, J. A. Pyle, A. D. Robinson, S. J. Andrews, A. C. Lewis, J. Minaeian, A. Vaughan, J. R. Dorsey, M. W. Gallagher, M. Le Breton, R. Newton, C. J. Percival, H. M. A. Ricketts, S. J.-B. Baugitte, G. J. Nott, A. Wellpott, M. J. Ashfold, J. Flemming, R. Butler, P. I. Palmer, P. H. Kaye, C. Stopford, C. Chemel, H. Boesch, N. Humpage, A. Vick, A. R. MacKenzie, R. Hyde, P. Angelov, E. Meneguz, and A. J. Manning, "Co-ordinated Airborne Studies in the Tropics (CAST)," Bull. Am. Meteorol. Soc., p. 160229122059003, Feb. 2016.

This work is a summary paper for the Coordinated Airborne Studies in the Tropic (CAST) project and is primarily the work of Neil Harris, the principal investigator. Contributions by Richard Hyde, Plamen Angelov and Rob MacKenzie are based on the work and papers reported in this thesis.

Conference Papers

The following papers were submitted to peer reviewed conferences and published in the conference proceedings.

R. Hyde and P. Angelov, "Data density based clustering," in 2014 14th UK Workshop on Computational Intelligence (UKCI), 2014, vol. UKCI 2014, no. Ddc, pp. 1–7.

The initial concept was provided by Plamen Angelov. Design of experimental work was by Plamen Angelov and Richard Hyde. The algorithm and test software, carrying out of experimental work and results were the work of Richard Hyde. The first draft of the paper was authored by Richard Hyde and the final paper was the work of Richard Hyde and Plamen Angelov.

R. Hyde and P. Angelov, "A Fully Autonomous Data Density Based Clustering Technique," in 2014 IEEE Symposium on Evolving and Autonomous Learning Systems (EALS), 2014, pp. 116–123.

The initial concept, design of experimental work, algorithm and test software, experimental work and results were the work of Richard Hyde. The first draft of the paper was authored by Richard Hyde and the final paper was the work of Richard Hyde and Plamen Angelov.

R. Hyde and P. Angelov, "A new online clustering approach for data in arbitrary shaped clusters," in 2015 IEEE 2nd International Conference on Cybernetics (CYBCONF), 2015, pp. 228–233.

The initial concept, design of experimental work, algorithm and test software, experimental work and results were the work of Richard Hyde. The first draft of the paper was authored by Richard Hyde and the final paper was the work of Richard Hyde and Plamen Angelov.

Submitted Papers

Journal Papers

R. Hyde, P. Angelov, A. R. MacKenzie, and N. R. P. Harris, "Supporting Mission Scientists in the Face of Atmospheric Data Torrents," Bull. Am. Meteorol. Soc., p. Submitted June 2016.

The initial concept, design of experimental work, algorithm, and demonstration software, experimental work and results were the work of Richard Hyde. The first draft of the paper was the work of Richard Hyde and Rob MacKenzie. The submitted paper was the work of all the authors. (After completion of this thesis, this paper was rejected

as not consistent with the remit of BAMS and is currently being re-written for submission to an alternative publication.)

References

[1] Achachi, A., Benatia, D., Vi, C., and Vi, W. G. (2015). New Model of a Solar Wind Airplane for Geomatic Operations. *Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL(1W4):137–142.

[2] Aggarwal, C. C. and Reddy, C. K., editors (2014). *DATA Clustering Algorithms and Applications*. CRC Press, Boca Raton.

[3] Aggarwal, C. C., Watson, T. J., Ctr, R., Han, J., Wang, J., Yu, P. S., Watson, T. J., Ctr, R., Han, J., Wang, J., and Yu, P. S. (2003). A framework for clustering evolving data streams. *Proceedings of the 29th international conference on Very large data bases*, pages 81–92.

[4] Aggarwal, C. C., Wolf, J. L., Yu, P. S., Procopiuc, C., and Park, J. S. (1999). Fast algorithms for projected clustering. *ACM SIGMOD Record*, 28(2):61–72.

[5] Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. (2005). Automatic Subspace Clustering of High Dimensional Data. *Data Mining and Knowledge Discovery*, 11(1):5–33.

[6] Ali, T., Asghar, S., and Sajid, N. A. (2010). Critical analysis of DBSCAN variations. *2010 International Conference on Information and Emerging Technologies, ICIET 2010*.

[7] Allan, J. D., Morgan, W. T., Darbyshire, E., Flynn, M. J., Williams, P. I., Oram, D. E., Artaxo, P., Brito, J., Lee, J. D., and Coe, H. (2014). Airborne observations of IEPOX-derived isoprene SOA in the Amazon during SAMBBA. *Atmospheric Chemistry and Physics Discussions*, 14(9):12635–12671.

[8] Amini, A., Wah, T. Y., and Saboohi, H. (2014). On Density-Based Data Streams Clustering Algorithms: A Survey. *Journal of Computer Science and Technology*, 29(1):116–141.

[9] Angelov, P. (2012). *Autonomous Learning Systems*. John Wiley & Sons, Ltd, Chichester, UK.

[10] Angelov, P. and Zhou, X. (2008). Evolving Fuzzy-Rule Based Classifiers From Data Streams. *IEEE Transactions on Fuzzy Systems*, 16(6):1462–1474.

[11] Annan, J. D. and Hargreaves, J. C. (2011). Understanding the CMIP3 multimodel ensemble. *Journal of Climate*, 24(16):4529–4538.

[12] Appel, K. W., Gilliland, A. B., Sarwar, G., Gilliam, R. C., Appel, K. W., Gilliland, A. B., Sarwar, G., Gilliam, R. C., Appel, K.W., Gilliland, A. B., Sarwar, G., and Gilliam, R. C. (2007). Evaluation of the Community Multiscale Air Quality (CMAQ) model version 4 . 5 : Sensitivities impacting model performance Part I — Ozone. *Atmospheric Environment*, 41(40):9603–9615.

[13] Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., and Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1):243–256.

[14] Babcock, B., Babu, S., Datar, M., Motwani, R., and Widom, J. (2002). Models and issues in data stream systems. *Proceedings of the twentyfirst ACM SIGMODSIGACT-SIGART symposium on Principles of database systems PODS 02*, pages(2002-19):1.

[15] Baruah, R. D. and Angelov, P. (2013). DEC: Dynamically evolving clustering and its application to structure identification of evolving fuzzy model. *Transaction on Cybernetics*, 44(9):1–16.

[16] Batagelj, V. and Bren, M. (1995). Comparing resemblance measures. *Journal of Classification*, 12(1):73–90.

[17] Belojevic, G. (2013). Biopreparedness and Public Health. In *Biopreparedness and Public Health*, pages 187–195. Springer Netherlands.

[18] Bengtsson, T. and Cavanaugh, J. E. (2006). An improved Akaike information criterion for state-space model selection. *Computational Statistics and Data Analysis*, 50(10):2635–2654.

[19] Bezdek, J. C., Ehrlich, R., and Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2-3):191–203.

[20] Bifet, A., Holmes, G., Pfahringer, B., Kranen, P., Kremer, H., Jansen, T., and Seidl, T. (2010). MOA: Massive online analysis, a framework for stream classification and clustering. *HaCDAIS 2010*, 11:3.

[21] Birch, C. E., Brooks, I. M., Tjernström, M., Shupe, M. D., Mauritsen, T., Sedlař, J., Lock, a. P., Earnshaw, P., Persson, P. O. G., Milton, S. F., and Leek, C. (2012). Modelling atmospheric structure, cloudand their response to CCN in the central Arctic: ASCOS case studies. *Atmospheric Chemistry and Physics*, 12(7):3419–3435.

[22] Brito, J., Rizzo, L. V., Morgan, W. T., Coe, H., Johnson, B., Haywood, J., Longo, K., Freitas, S., Andreae, M. O., and Artaxo, P. (2014). Ground based aerosol characterization during the South American Biomass Burning Analysis (SAMBBA) field experiment. *Atmospheric Chemistry and Physics Discussions*, 14(8):12279–12322.

[23] Buck, C. R. (2011). Instrument Configuration List. Technical report, FAAM.

[24] Cabrerizo, A., Larramendi, R., Albar, J.-P., and Dachs, J. (2017). Persistent organic pollutants in the atmosphere of the Antarctic Plateau. *Atmospheric Environment*, 149:104–108.

[25] Calinski, T. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods*, 3(1):1–27.

[26] Cao, F., Ester, M., Qian, W., and Zhou, A. (2006). Density-based clustering over an evolving data stream with noise. In ... *Conference on Data Mining*, pages 328–339.

[27] Carpenter, L. J., Jones, C. E., Dunk, R. M., Hornsby, K. E., and Woeltjen, J. (2009). Air-sea fluxes of biogenic bromine from the tropical and North Atlantic Ocean. *Atmospheric Chemistry and Physics*, 9(July 2006):1805–1816.

[28] Cervone, G., Franzese, P., Ezber, Y., and Boybeyi, Z. (2008). Risk assessment of atmospheric hazard releases using K-means clustering. *Proceedings - IEEE International Conference on Data Mining Workshops, ICDM Workshops 2008*, pages 342–348.

[29] Chan, W. W.-y. (2006). A Survey on Multivariate Data Visualization. *Science And Technology*, June(June):1–29.

[30] Chaoji, V. (2009). *Efficient Algorithms for Mining Arbitrary Shaped Clusters*. PhD thesis, Rensselaer Polytechnic Institute.

[31] Chen, L., Feng, Q., He, Q., Huang, Y., Zhang, Y., Jiang, G., Zhao, W., Gao, B., Lin, K., and Xu, Z. (2017). Sources, atmospheric transport and deposition mechanism of organochlorine pesticides in soils of the Tibetan Plateau. *Science of The Total Environment*, 577:405–412.

[32] Chen, Y. and Tu, L. (2007). Density-based clustering for real-time stream data. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining KDD 07*, volume d, page 133, New York, New York, USA. ACM Press.

[33] Christiansen, B. (2007). Atmospheric circulation regimes: Can cluster analysis provide the number? *Journal of Climate*, 20(Christiansen 2003):2229–2250.

[34] Ciscar, J.-c. (2009). Climate change impacts in Europe Final report of the PESETA research project. Technical Report October, European Commission Joint Research Centre.

[35] Coe, H. (2012). SAMBBA - The South American Biomass Burning Analysis (SAMBBA).

[36] Collier, C. G. (2013). Atmospheric Dynamics - by Mankin Mak. *Meteorological Applications*, 20(1):E1–E1.

[37] Comaniciu, D., Meer, P., and Member, S. (2002). Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619.

[38] Committee on Environment and Natural Resources (2001). Intercontinental Transport of Air Pollution: A Review of Federal Research and Future Needs. Technical report, Committee on Environment and Natural Resources.

[39] Corrigan, A. L., Russell, L. M., Takahama, S., Äijälä, M., Ehn, M., Junninen, H., Rinne, J., Petäjä, T., Kulmala, M., Vogel, A. L., Hoffmann, T., Ebben, C. J., Geiger, F. M., Chhabra, P., Seinfeld, J. H., Worsnop, D. R., Song, W., Auld, J., and Williams, J. (2013). Biogenic and biomass burning organic aerosol in a boreal forest at Hyytiälä, Finland, during HUMPPA-COPEC 2010. *Atmospheric Chemistry and Physics*, 13(24):12233–12256.

[40] Council of Economic Advisers, The Council of Economic Advisers, Council of Economic Advisers, The Council of Economic Advisers, Council of Economic Advisers, and The Council of Economic Advisers (2014). The cost of delaying action to stem climate change. Technical Report July, Council of Economic Advisors.

[41] Crawford, I., Robinson, N. H., Flynn, M. J., Foot, V. E., Gallagher, M. W., Huffman, J. A., Stanley, W. R., and Kaye, P. H. (2014). Characterisation of bioaerosol emissions from a Colorado pine forest: Results from the beachon-rombas experiment. *Atmospheric Chemistry and Physics*, 14(16):8559–8578.

[42] Dasgupta, A., Poco, J., Bertini, E., and Silva, C. T. (2016). Reducing the Analytical Bottleneck for Domain Scientists: Lessons from a Climate Data Visualization Case Study. *Computing in Science and Engineering*, 18(1):92–100.

[43] Davies, D. L. and Bouldin, D. W. (1979). A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227.

[44] Defays, D. (1977). An efficient algorithm for a complete link method. *The Computer Journal*, 20(4):364–366.

[45] Dempster, A., Laird, N., Rubin, D., and Dempster, A. P., Laird, N . M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

[46] Dunn, J. C. (1974). Cybernetics and Systems A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, 3(3):32–57.

[47] Dutta Baruah, R., Angelov, P., Baruah, R. D., Angelov, P., Dutta Baruah, R., and Angelov, P. (2012). Evolving local means method for clustering of streaming data. *IEEE International Conference on Fuzzy Systems*, pages 10–15.

[48] Edelsbrunner, H., Kirkpatrick, D., and Seidel, R. (1983). On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559.

[49] El-Sonbaty, Y., Ismail, M., and Farouk, M. (2004). An efficient density based clustering algorithm for large databases. In *16th IEEE International Conference on Tools with Artificial Intelligence*, pages 637–677. IEEE Comput. Soc.

[50] Environmental Research Group, K. C. L. (2015). London Air Quality Network :: Welcome to the London Air Quality Network » Data Downloads.

[51] EOSDIS (2015). Fire Information for Resource Management System.

[52] Esri, A. and Paper, W. (1998). ESRI Shapefile Technical Description. *Computational Statistics*, 16(July):370–371.

[53] Ester, M., Kriegel, H. P., Sander, J., Xu, X., and Martin Ester Jorg Sander, Xiaowei Xu, H.-P. K. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland, Oregan. AAAI.

[54] European Centre for Medium-Range Weather Forecasts (2017). Climate reanalysis | ECMWF.

[55] European Commission (2015). Supporting climate action through the EU budget - European Commission.

[56] Ferchichi, S. E., Zidi, S., Laabidi, K., Ksouri, M., Maouche, S., El Ferchichi, S., Zidi, S., Laabidi, K., Ksouri, M., and Maouche, S. (2011). Feature extraction for atmospheric pollution detection. In *Communications, Computing and Control Applications (CCCA), 2011 International Conference on*, pages 1–6.

[57] Figueiredo, M. A. T. and Jain, A. K. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396.

[58] Foster, A. and Ford, N. (2003). Serendipity and information seeking: an empirical study. *Journal of Documentation*, 59(3):321–340.

[59] Fowlkes, E. B. and Mallows, C. L. (1983). A Method for Comparing Two Hierarchical Clusterings. *Journal of the American Statistical Association*, 78(383):553.

[60] Franklin, J. (2006). Long-Range Transport of Chemicals in the Environment. Technical report, Monitoring & Environmental Chemistry Working group.

[61] Gama, J., Rodrigues, P. P., and Lopes, L. (2011). Clustering distributed sensor data streams using local processing and reduced communication. *Intelligent Data Analysis*, 15(1):3–28.

[62] Gentle, J. E., Kaufman, L., and Rousseuw, P. J. (1990). *Finding groups in data : an introduction to cluster analysis*. New York : Wiley.

[63] Georges Hebrail, A. B. (2012). Individual household electric power consumption Data Set .

[64] Giglio, L., Csiszar, I., and Justice, C. O. (2006). Global distribution and seasonality of active fires as observed with the Terra and Aqua Moderate Resolution Imaging Spectroradiometer (MODIS) sensors. *Journal of Geophysical Research*, 111(G2):G02016.

[65] Giglio, L., Descloitres, J., Justice, C. O., and Kaufman, Y. J. (2003). An Enhanced Contextual Fire Detection Algorithm for MODIS. *Remote Sensing of Environment*, 87(2-3):273–282.

[66] Gill, J. C. and Malamud, B. D. (2016). Hazard Interactions and Interaction Networks (Cascades) within Multi-Hazard Methodologies. *Earth System Dynamics Discussions*, 2016(January):1–35.

[67] Glass, L. and Mackey, M. (2010). Mackey-Glass equation. *Scholarpedia*, 5(3):6908.

[68] Goil, S., Nagesh, H., and Choudhary, A. (1999). MAFIA: Efficient and scalable subspace clustering for very large data sets. ... *Discovery and Data Mining*, 5:443–452.

[69] Golding, B., Clark, P., and May, B. (2005). The Boscastle flood: Meteorological analysis of the conditions leading to flooding on 16 August 2004. *Weather*, 60(8):230–235.

[70] Gordon, N. D. and Norris, J. R. (2010). Cluster analysis of midlatitude oceanic cloud regimes: Mean properties and temperature sensitivity. *Atmospheric Chemistry and Physics*, 10(13):6435–6459.

[71] Graham, R. L. (1972). An efficient algorith for determining the convex hull of a finite planar set.

[72] Gramsch, E., Cereceda-Balic, F., Oyola, P., and von Baer, D. (2006). Examination of pollution trends in Santiago de Chile with cluster analysis of PM10 and Ozone data. *Atmospheric Environment*, 40(28):5464–5475.

[73] Gulliver, J. S. (2012). *Transport and fate of chemicals in the environment : selected entries from the Encyclopedia of sustainability science and technology*. Springer.

[74] Hahsler, M., Arya, S., and Mount, D. (2015). Density based clustering of applications with noise (DBSCAN) and related algorithms.

[75] Harris, N. R. P., Carpenter, L. J., Lee, J. D., Vaughan, G., Filus, M. T., Jones, R. L., OuYang, B., Pyle, J. A., Robinson, A. D., Andrews, S. J., Lewis, A. C., Minaeian, J., Vaughan, A., Dorsey, J. R., Gallagher, M. W., Le Breton, M., Newton, R., Percival, C. J., Ricketts, H. M. A., Baugitte, S. J.-B., Nott, G. J., Wellpott, A., Ashfold, M. J., Flemming, J., Butler, R., Palmer, P. I., Kaye, P. H., Stopford, C., Chemel, C., Boesch, H., Humpage, N., Vick, A., MacKenzie, A. R., Hyde, R., Angelov, P., Meneguz, E., and Manning, A. J. (2016). Co-ordinated Airborne Studies in the Tropics (CAST). *Bulletin of the American Meteorological Society*, page 160229122059003.

[76] Hemond, H. F., Fechner, E., and Fechner-Levy, E. J. (2000). *Chemical fate and transport in the environment*. Academic press, San Diego, 2nd edition.

[77] Hettich, S. and Bay, S. D. (1999). The UCI KDD Archive. Technical report, University of California, Department of Information and Computer Science, Irvine.

[78] Hornbrook, R. S., Blake, D. R., Diskin, G. S., Fried, A., Fuelberg, H. E., Meinardi, S., Mikoviny, T., Richter, D., Sachse, G. W., Vay, S. A., Walega, J., Weibring, P., Weinheimer, A. J., Wiedinmyer, C., Wisthaler, A., Hills, A., Riemer, D. D., Apel, E. C., Hills, A.Riemer, D. D., and Apel, E. C. (2011). Observations of nonmethane organic compounds during ARCTAS-Part 1: Biomass burning emissions and plume enhancements. *Atmospheric Chemistry and Physics*, 11(21):11103–11130.

[79] Horseman, a. M., MacKenzie, a. R., and Chipperfield, M. P. (2009). Tracers and traceability- implementing the cirrus parameterisation from LACM in the TOMCAT SLIMCAT chemistry transport model. *Geoscientific Model Development Discussions*, 2(2005):1299–1333.

[80] Hyde, R. and Angelov, P. (2014). Data density based clustering. In *2014 14th UK Workshop on Computational Intelligence (UKCI)*, pages 1–7, Bradford. IEEE.

[81] Hyde, R. and Angelov, P. (2015). A new online clustering approach for data in arbitrary shaped clusters. In *2015 IEEE 2nd International Conference on Cybernetics (CYBCONF)*, pages 228–233, Gdynia. IEEE.

[82] Hyde, R., Angelov, P., and MacKenzie, A. R. (2017). Fully online clustering of evolving data streams into arbitrarily shaped clusters. *Information Sciences*, 382-383:96–114.

[83] Jaccard, P. (1912). The Distribution of the Flora in the Alpine Zone.1. *New Phytologist*, 11(2):37–50.

[84] Jennings, G. (2008). Global Hawk sets endurance record.

[85] Jensen, E. J., Pfister, L., Jordan, D. E., Bui, T. V., Ueyama, R., Singh, H. B., Thornberry, T., Rollins, A. W., Gao, R.-S., Fahey, D. W., Rosenlof, K. H., Elkins, J. W., Diskin, G. S., DiGangi, J. P., Lawson, R. P., Woods, S., Atlas, E. L., Navarro Rodriguez, M. A., Wofsy, S. C., Pittman, J., Bardeen, C. G., Toon, O. B., Kindel, B. C., Newman, P. A., McGill, M. J., Hlavka, D. L., Lait, L. R., Schoeberl, M. R., Bergman, J. W., Selkirk, H. B., Alexander, M. J., Kim, J.-E., Lim, B. H., Stutz, J., and Pfeilsticker, K. (2015). The NASA Airborne Tropical TRopopause EXperiment (ATTREX): High-Altitude Aircraft Measurements in the Tropical Western Pacific. *Bulletin of the American Meteorological Society*, page 151221155301005.

[86] K Bache, M. L. (2013). UCI Machine Learning Repository.

[87] Kageyama, Y., Sato, I., and Nishida, M. (2007). Automatic classification algorithm for NOAA- AVHRR data using mixels. *2007 IEEE International Geoscience and Remote Sensing Symposium*, pages 2040–2043.

[88] Kailing, K., Kriegel, H.-P., and Kröger, P. (2004). Density-Connected Subspace Clustering for High Dimensional Data. *4th SIAM Int. Conf. on Data Mining*, pages 246–257.

[89] Karypis, G., Han, E.-H., and Kumar, V. (1999). Chameleon: hierarchical clustering using dynamic modeling. *Computer*, 32(8):68–75.

[90] Kim, M. C., Zhu, Y., and Chen, C. (2016). How are they different? A quantitative domain comparison of information visualization and data visualization (2000–2014). *Scientometrics*, 107(1):123–165.

[91] Kohlhepp, R., Ruhnke, R., Chipperfield, M. P., De Mazière, M., Notholt, J., Barthlott, S., Batchelor, R. L., Blatherwick, R. D., Blumenstock, T., Coffey, M. T., Demoulin, P., Fast, H., Feng, W., Goldman, A., Griffith, D. W. T., Hamann, K., Hannigan, J. W., Hase, F., Jones, N. B., Kagawa, A., Kaiser, I., Kasai, Y., Kirner, O., Kouker, W., Lindenmaier, R., Mahieu, E., Mittermeier, R. L., Monge-Sanz, B., Morino, I., Murata, I., Nakajima, H., Palm, M., Paton-Walsh, C., Raffalski, U., Reddmann, T., Rettinger, M., Rinsland, C. P., Rozanov, E., Schneider, M., Senten, C., Servais, C., Sinnhuber, B. M., Smale, D., Strong, K., Sussmann, R., Taylor, J. R., Vanhaelewyn, G., Warneke, T., Whaley, C., Wiehle, M., and Wood, S. W. (2012). Observed and simulated time evolution of HCl, ClONO 2, and HF total column abundances. *Atmospheric Chemistry and Physics*, 12(7):3527–3556.

[92] Kolusu, S. R., Marsham, J. H., Mulcahy, J., Johnson, B., Dunning, C., Bush, M., and Spracklen, D. V. (2015). Impacts of Amazonia biomass burning aerosols assessed from short-range weather forecasts. *Atmospheric Chemistry and Physics Discussions*, 15(13):18883–18919.

[93] Kranen, P., Assent, I., Baldauf, C., and Seidl, T. (2011). The ClusTree: Indexing micro-clusters for anytime stream mining. *Knowledge and Information Systems*, 29(2):249–272.

[94] Lamarque, J. F., Shindell, D. T., Josse, B., Young, P. J., Cionni, I., Eyring, V., Bergmann, D., Cameron-Smith, P., Collins, W. J., Doherty, R., Dalsoren, S., Faluvegi, G., Folberth, G., Ghan, S. J., Horowitz, L. W., Lee, Y. H., MacKenzie, I. A., Nagashima, T., Naik, V., Plummer, D., Righi, M., Rumbold, S. T., Schulz, M., Skeie, R. B., Stevenson, D. S., Strode, S., Sudo, K., Szopa, S., Voulgarakis, A., and Zeng, G. (2013). The atmospheric chemistry and climate model intercomparison Project (ACCMIP): Overview and description of models, simulations and climate diagnostics. *Geoscientific Model Development*, 6(1):179–206.

[95] Leggett, J. A., Lattanzio, R. K., and Bruner, E. (2013). Federal Climate Change Funding from FY2008 to FY2014. Technical report, Congressional Research Service.

[96] Levine, J. G., MacKenzie, A. R., Squire, O. J., Archibald, A. T., Griffiths, P. T., Abraham, N. L., Pyle, J. A., Oram, D. E., Forster, G., Brito, J. F., Lee, J. D., Hopkins, J. R., Lewis, A. C., Bauguitte, S. J. B., Demarco, C. F., Artaxo, P., Messina, P., Lathière, J., Hauglustaine, D. A., House, E., Hewitt, C. N., and Nemitz, E. (2015). Isoprene chemistry in pristine and polluted Amazon environments: Eulerian and Lagrangian model frameworks and the strong bearing they have on our understanding of surface ozone and predictions of rainforest exposure to this priority pollutant. *Atmospheric Chemistry and Physics Discussions*, 15(17):24251–24310.

[97] Liu, B. (2006). A Fast Density-Based Clustering Algorithm for Large Databases. In *International Conference on Machine Learning and Cybernetics*, pages 996–1000. IEEE.

[98] Liu, L.-x., Guo, Y.-f., Kang, J., and Huang, H. (2009). A three-step clustering algorithm over an evolving data stream. *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, 1:160–164.

[99] Liu, S., Maljovec, D., Wang, B., Bremer, P., and Pascucci, V. (2015). Visualizing High-Dimensional Data : Advances in the Past Decade. *Eurographics Conference on Visualization (EuroVis)*.

[100] Liu, Y., Li, Z., Xiong, H., Gao, X., and Wu, J. (2010). Understanding of Internal Clustering Validation Measures. In *2010 IEEE International Conference on Data Mining Understanding*.

[101] Lughofer, E. (2011). Dynamic Evolving Cluster Models Using On-line Split-and-Merge Operations. In *2011 10th International Conference on Machine Learning and Applications and Workshops*, volume 2, pages 20–26. IEEE.

[102] Lughofer, E. and Angelov, P. (2011). Handling drifts and shifts in on-line data streams with evolving fuzzy systems. *Applied Soft Computing*, 11(2):2057–2068.

[103] Lyapina, O., Schultz, M. G., and Hense, A. (2016). Cluster analysis of European surface ozone observations for evaluation of MACC reanalysis data. *Atmospheric Chemistry and Physics*, 16(11):6863–6881.

[104] Mackey, M. and Glass, L. (1977). Oscillation and chaos in physiological control systems. *Science*, 197(4300):287–289.

[105] MacQueen, J. B. (1967). Some Methods for classification and analysis of multivariate observations. *Procedings of the Fifth Berkeley Symposium on Math, Statistics, and Probability*, 1(233):281–297.

[106] Marenco, F., Amiridis, V., Marinou, E., Tsekeri, A., and Pelon, J. (2014). Airborne verification of CALIPSO products over the Amazon: a case study of daytime observations in a complex atmospheric scene. *Atmospheric Chemistry and Physics Discussions*, 14(7):9203–9224.

[107] McLachlan, G. J. and Chang, S. U. (2004). Mixture modelling for cluster analysis. *Statistical methods in medical research*, 13(5):347–361.

[108] McQuitty, L. (1966). Similarity Analysis by Reciprocal Pairs for Discrete and Continuous Data Educational and Psychological Measurement. *Educational and Psychological Measurement*.

[109] McQuitty, L. L. (1967). Expansion of Similarity Analysis By Reciprocal Pairs for Discrete and Continuous Data. *Educational and Psychological Measurement*, 27(2):253–255.

[110] Morgan, W. T., Allan, J. D., Flynn, M., Darbyshire, E., Hodgson, a., Johnson, B. T., Haywood, J. M., Freitas, S., Longo, K., Artaxo, P., and Coe, H. (2013). Overview of the South American biomass burning analysis (SAMBBA) field experiment. *AIP Conference Proceedings*, 1527(September 2015):587–590.

[111] Mullins, J. and Bharadwaj, P. (2015). Effects of short-term measures to curb air pollution: Evidence From Santiago, Chile. *American Journal of Agricultural Economics*, 97(4):1107–1134.

[112] Munsell, E. B., Sippel, J. A., Braun, S. A., Weng, Y., and Zhang, F. (2015). Dynamics and Predictability of Hurricane Nadine (2012) Evaluated through Convection-Permitting Ensemble Analysis and Forecasts. *Monthly Weather Review*, 143(11):4514–4532.

[113] Muthers, S., Kuchar, A., Stenke, A., Schmitt, J., Anet, J. G., Raible, C. C., and Stocker, T. F. (2016). Stratospheric age of air variations between 1600 and 2100. *Geophysical Research Letters*, 43(10):5409–5418.

[114] Namadchian, A. and Esfandani, G. (2012). DSCLU: A New Data Stream Clustring Algorithm for Multi Density Environments. In *2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pages 83–88. IEEE.

[115] NASA (2013). ATTREX Platforms - Airborne.

[116] NASA (2015). ATTREX Missions. https://espo.nasa.gov/missions/attrex/.

[117] NASA EOSDIS Earthdata (2015). NASA EOSDIS Eartdata. https://earthdata.nasa.gov/active-fire-data#tab-content-6.

[118] NCAR (2014). CONTRAST Payload. https://www2.acom.ucar.edu/contrast/gv-payload.

[119] NCAR (2016). CONTRAST Project. https://www2.acom.ucar.edu/contrast.

[120] NERC (2014). CAST- Co-ordinated Airborne Studies in the Tropics.

[121] NERC (2015a). NERC Atmospheric Grants on the Web 150907. http://gotw.nerc.ac.uk/thematic.asp.

[122] NERC (2015b). NERC Grants on the Web 150907. http://gotw.nerc.ac.uk/thematic.asp.

[123] Ng, R. T. and Han, J. (2002). CLARANS: A method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering*, 14(5):1003–1016.

[124] Nisha and Kaur, P. J. (2015). A Survey of Clustering Techniques and Algorithms. *2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 304–307.

[125] Norby, R. J., De Kauwe, M. G., Domingues, T. F., Duursma, R. A., Ellsworth, D. S., Goll, D. S., Lapola, D. M., Luus, K. A., MacKenzie, A. R., Medlyn, B. E., Pavlick, R., Rammig, A., Smith, B., Thomas, R., Thonicke, K., Walker, A. P., Yang, X., and Zaehle, S. (2015). Model-data synthesis for the next generation of forest free-air CO2 enrichment (FACE) experiments. *The New phytologist*, 209(1):17–28.

[126] Pacifico, F., Folberth, G. A., Sitch, S., Haywood, J. M., Artaxo, P., and Rizzo, L. V. (2014). Biomass burning related ozone damage on vegetation over the Amazon forest. *Atmospheric Chemistry and Physics Discussions*, 14(14):19955–19983.

[127] Papenhausen, E., Wang, B., Ha, S., Zelenyuk, A., Imre, D., and Mueller, K. (2013). GPU-accelerated incremental correlation clustering of large data with visual feedback. *Proceedings - 2013 IEEE International Conference on Big Data, Big Data 2013*, pages 63–70.

[128] Parsons, L., Haque, E., Liu, H., Parsons, L., Haque, E., Haque, E., Liu, H., and Liu, H. (2004). Subspace Clustering for High Dimensional Data: A Review. *ACM SIGKDD Explorations Newsletter - Special issue on learning from imbalanced datasets*, 6(1):90–105.

[129] Partington, K. and Cardille, J. (2013). Uncovering Dominant Land-Cover Patterns of Quebec: Representative Landscapes, Spatial Clusters, and Fences. *Land*, 2(4):756–773.

[130] PCC (2012). 2012 Air quality updating and screening assessment. Technical Report April, Plymouth City Council, Plymouth.

[131] Pöelitz, C., Andrienko, G., and Andrienko, N. (2010). Finding arbitrary shaped clusters with related extents in space and time. *EuroVAST 2010: International Symposium on Visual Analytics Science and Technology*, pages 19–25.

[132] Pugh, T. A. M., Cain, M., Methven, J., Wild, O., Arnold, S. R., Real, E., Law, K. S., Emmerson, K. M., Owen, S. M., Pyle, J. A., Hewitt, C. N., and MacKenzie, A. R. (2012). A Lagrangian model of air-mass photochemistry and mixing using a trajectory ensemble: the Cambridge Tropospheric Trajectory model of Chemistry And Transport (CiTTyCAT) version 4.2. *Geoscientific Model Development*, 5(1):193–221.

[133] Rand, W. M. (1971). Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, 66(336):846.

[134] Reisen, F., Meyer, C. P. M., and Keywood, M. D. (2013). Impact of biomass burning sources on seasonal aerosol air quality. *Atmospheric Environment*, 67:437–447.

[135] Ren, J. and Ma, R. (2009). Density-based data streams clustering over sliding windows. In *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, volume 5, pages 248–252. IEEE.

[136] Rendón, E., Abundez, I., Arizmendi, A., and Quiroz, E. M. (2011). Internal versus External cluster validation indexes. *International Journal of Computers and Communications*, 5(1):27—-34.

[137] Roberts, R. (1989). Serendipity: Accidental Discoveries in Science. *Serendipity: Accidental Discoveries in Science*.

[138] Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

[139] Schkoda, R. F., Lund, R. B., and Wagner, J. R. (2014). Clustering of Cyclostationary Signals with Applications to Climate Station Sitings, Eliminations, and Merges. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 7(5):1754–1762.

[140] Sibson, R. (1973). SLINK: an optimally efficient algorithm for the single-link cluster method.

[141] Silva, J. A., Faria, E. R., Barros, R. C., Hruschka, E. R., de Carvalho, A. C. P. L. F., and Gama, J. (2013). Data stream clustering: A Survey. *ACM Computing Surveys*, 46(1):1–31.

[142] Simpson, R. W., Thatcher, W., and Savage, J. C. (2012). Using cluster analysis to organize and explore regional GPS velocities. *Geophysical Research Letters*, 39(18):n/a–n/a.

[143] Šmídl, V. and Hofman, R. (2013). Tracking of atmospheric release of pollution using unmanned aerial vehicles. *Atmospheric Environment*, 67:425–436.

[144] Solar Impulse SA (2016). Solar Impulse Clean Technologies to Fly Around the World.

[145] StatSoft (2016). Finding the Right Number of Clusters in k-Means and EM Clustering: v-Fold Cross-Validation.

[146] Stiller, G. P., Von Clarmann, T., Haenel, F., Funke, B., Glatthor, N., Grabowski, U., Kellmann, S., Kiefer, M., Linden, A., Lossow, S., and López-Puertas, M. (2012). Observed temporal evolution of global mean age of stratospheric air for the 2002 to 2010 period. *Atmos. Chem. Phys. Atmospheric Chemistry and Physics*, 12(7):3311–3331.

[147] Stocker, T. (2013). Climate change 2013 : the physical science basis : Working Group I contribution to the Fifth assessment report of the Intergovernmental Panel on Climate Change. Technical report, Intergovernmental Panel on Climate Change 2013.

[148] Tan, D. G. H., Haynes, P. H., MacKenzie, A. R., and Pyle, J. A. (1998). Effects of fluid-dynamical stirring and mixing on the deactivation of stratospheric chlorine. *Journal of Geophysical Research: Atmospheres*, 103(D1):1585–1605.

[149] The World Bank (2009). *World Development Report 2010*. The World Bank.

[150] Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423.

[151] Triantafyllou, E., Giamarelou, M., Bossioli, E., Zarmpas, P., Theodosi, C., Matsoukas, C., Tombrou, M., Mihalopoulos, N., and Biskos, G. (2016). Particulate pollution transport episodes from Eurasia to a remote region of northeast Mediterranean. *Atmospheric Environment*, 128:45–52.

[152] UCAR (2011). Earth's Atmosphere | UCAR Center for Science Education.

[153] Vinh, N. X., Epps, J., and Bailey, J. (2009). Information theoretic measures for clusterings comparison. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, pages 1–8, New York, New York, USA. ACM Press.

[154] Vriend, S. P., van Gaans, P. F. M., Middelburg, J., and de Nijs, A. (1988). The application of fuzzy c-means cluster analysis and non-linear mapping to geochemical datasets: Examples from Portugal. *Applied Geochemistry*, 3(2):213–224.

[155] Wan, L., Ng, W. K., Dang, X. H., Yu, P. S., and Zhang, K. (2009). Density-Based Clustering of Data Streams at Multiple Resolutions. *ACM Transactions on Knowledge Discovery from Data*, 3(3):1–28.

[156] Watkiss, P., Bosello, F., Buchner, B., Catenacci, M., Goria, A., Kuik, O., and Karakaya, E. (2007). Climate change: the cost of inaction and the cost of adaptation. Technical Report 13, Office for Official Publications of the European Communities.

[157] Wofsy, B. C., Daube, R., Jimenez, R., Kort, E., Pittman, J. V., Park, S., Commane, R., Xiang, B., Santoni, G., Jacob, D., Fisher, J., Pickett-Heaps, C., Wang, H., Wecht, K., Wang, Q.-Q., Stephens, B. B., Shertz, S., Watt, A., Romashkin, P., Campos, T., Haggerty, J., and Coop, W. A. (2011). CDIAC HIPPO Data and Documentation Download.

[158] Wofsy, B. C., Daube, R., Jimenez, R., Kort, E., Pittman, J. V., Park, S., Commane, R., Xiang, B., Santoni, G., Jacob, D., Fisher, J., Pickett-Heaps, C., Wang, H., Wecht, K., Wang, Q.-Q., Stephens, B. B., Shertz, S., Watt, A., Romashkin, P., Campos, T., Haggerty, J., and Coop, W. A. (2012). HIPPO Merged 10-second Meteorology, Atmospheric Chemistry, Aerosol Data (R20121129).

[159] Woo, K.-G. G., Lee, J.-H. H., Kim, M.-H. H., and Lee, Y.-J. J. (2004). FINDIT: a fast and intelligent subspace clustering algorithm using dimension voting. *Information and Software Technology*, 46(4):255–271.

[160] World Bank (2010). The economics of adaptation to climate change: A Synthesis Report. Technical Report August, The World Bank Group, Washington DC.

[161] Wyche, K. P., Monks, P. S., Smallbone, K. L., Hamilton, J. F., Alfarra, M. R., Rickard, a. R., McFiggans, G. B., Jenkin, M. E., Bloss, W. J., Ryan, A. C., Hewitt, C. N., and MacKenzie, A. R. (2015). Mapping gas-phase organic reactivity and concomitant secondary organic aerosol formation: chemometric dimension reduction techniques for the deconvolution of complex atmospheric data sets. *Atmospheric Chemistry and Physics*, 15(14):8077–8100.

[162] Xu, D. and Tian, Y. (2015). A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, 2(2):165–193.

[163] Yager, R. and Filev, D. (1994a). Generation of Fuzzy Rules by Mountain Clustering. *Journal of Intelligent & Fuzzy Systems*, 2(3):209–219.

[164] Yager, R. R. and Filev, D. P. (1994b). Approximate clustering via the mountain method. *IEEE Transactions on Systems, Man and Cybernetics*, 24(8):1279–1284.

[165] Yip, K. Y., Ng, M., and Cheung, D. (2003). A Review on Projected Clustering Algorithms. *International Journal of Applied Mathematics*, 13(1):35–48.

[166] Yuan, B., Liu, Y., Shao, M., Lu, S., and Streets, D. G. (2010). Biomass burning contributions to ambient VOCs species at a receptor site in the Pearl River delta (PRD), China. *Environmental Science and Technology*, 44(12):4577–4582.

[167] Zhang, T., Ramakrishnan, R., and Livny, M. (1996). BIRCH: An Efficient Data Clustering Databases Method for Very Large. *ACM SIGMOD International Conference on Management of Data*, 1:103–114.

[168] Zscheischler, J., Mahecha, M. D., and Harmeling, S. (2012). Climate classifications: The value of unsupervised clustering. *Procedia Computer Science*, 9:897–906.

Appendix A

DDC Algorithm

This appendix provides the mathematical steps of the algorithm, as implemented in Matlab, for the experiments in this thesis. These steps are not intended to represent the most computationally efficient method of implementing the algorithm but to implement a clear technique with all the steps clearly separated.

α learning parameter

μ_0 Global Mean

μ_l Local Mean

μ_j the vector of the mean of cluster j

μ_{jd} dimension d of the mean vector of cluster j

$\sigma(\{C_j\})$ standard deviation of the set of clustered data $\{C_j\}$

$\{C_j\}$ the set of members of cluster j

$\{C_{jd}\}$ the set of dimension d of the set of members of cluster j

$\{Data\}$ the set of remaining un-clustered data, initially the full data set

D_i density of sample x_i

d number of data dimensions

i generic index

j index of next cluster to be created

k generic index

N the number of data samples

r_0 initial radius of the clusters, may be the same for each axis or a vector individual values for each axis

r_{jd} radius of cluster j in dimension d

x_i data sample i

x_{id} dimension d of data sample i

X_0 Global Scalar product

X_l Local Scalar product

Input: $\{Data\}$, r_0

Initialization:

$\mu_0 = \frac{1}{N} \sum_{i=1}^{N} (x_i \in \{Data\})$

$X_0 = \frac{1}{N} \sum_{i=1}^{N} (x_i \in \{Data\})^2$

$j = 1$

while $\{Data\} \neq \emptyset$ **do**

Find Global Densest Point, Assign Data to Cluster, Remove Outliers

for $\forall x_i \in \{Data\}$ **do**

$D_i = \frac{1}{1+\|x_i-\mu_0\|^2+X_0-\|x_0\|^2}$

end for

$\mu_j = x_{argmax_{i=1}^{N} D_i}$

$\{C_j\} \ni \sum_{i=1}^{N} \sum_{k=1}^{d} \frac{[(x_{ik} \in \{Data\} - \mu_{ik}]^2)}{r_0^2} \leq 1$

$\{C_j\} \ni \left\| (x_i \in \{C_j\}) - \mu_j \right\| < (3 \times \sigma(\{C_j\}))$

Find Local Densest Point, Assign Data to Cluster, Remove Outliers:

$\mu_l = \frac{1}{N} \sum_{i=1}^{N} (x_i \in \{C_j\})$

$X_l = \frac{1}{N} \sum_{i=1}^{N} (x_i \in C_j)^2$

for $\forall x_i \in \{C_j\}$ **do**

$D_i = \frac{1}{1+\left\| (x_i \in \{C_j\}) - \mu_0 \right\|^2 + X_0 - \left\| (x_i \in \{C_j\}) \right\|^2}$

end for

$\mu_j = x_{argmax_{i=1}^{N} D_i}$

$\{C_j\} \ni \sum_{i=1}^{N} \sum_{k=1}^{d} \frac{[(x_{ik} \in \{C_j\} - \mu_{ik}]^2)}{r_0^2} \leq 1$

18: $\{C_j\} \ni \left\|(x_i \in \{C_j\}) - \mu_j\right\| < (3 \times \sigma(\{C_j\}))$

Adjust Radii to Match Farthest Cluster Data in Each Axis, Assign Data and Remove Outliers

19: **for** $k = 1$ to d **do**

20: $r_{jk} = max(\|(x_{ik} \in \{C_{jk}\}) - \mu_{jk}\|)$

21: **end for**

22: $\{C_j\} \ni \sum_{i=1}^{N} \sum_{k=1}^{d} \frac{[(x_{ik} \in \{C_j\} - \mu_{ik}]^2)}{r_{jk}^2} \leq 1$

23: $\{C_j\} \ni \left\|(x_i \in \{C_j\}) - \mu_j\right\| < (3 \times \sigma(\{C_j\}))$

Adjust Radii to Match Final Data Assignment

24: **for** $k = 1$ to d **do**

25: $r_{jk} = max(\|(x_{ik} \in \{C_{jk}\}) - \mu_{jk}\|)$

26: **end for**

27: $\{Data\} = \{Data\} - \{C_j\}$

28: **end while**

End of Base Algorithm

Merging of Clusters if a cluster centre is within another cluster ellipse.

29: $Merged = 1$

30: **while** $Merged = 1$ **do**

31: $Merged = 0$

32: **for** $i = 1$ to j **do**

33: **for** $k = 1$ to j **do**

34: **if** $\sum_{k=1}^{d} \frac{[(\mu_{kd} - \mu_{id}]^2)}{r_{id}^2} \leq 1$ **then**

35: $\{C_i\} = \{C_i\} \cup \{C_k\}$

36: $\mu_i = \overline{x_i}$

37: $r_i = max(x_i - \mu_i)$

38: $Merged = 1$

39: **end if**

40: **end for**

41: **end for**

42: **end while**

Appendix B

DDCAR Algorithm

This section provides the algorithm for the radius estimation performed by DDCAR only. The radii resulting from this algorithm are used to feed into the DDC algorithm given in Appendix A so this will not be reproduced here. The mathematical steps of the algorithm, as implemented in Matlab, for the experiments in this thesis are given and are not intended to represent the most computationally efficient method of implementing the algorithm but for clarity.

d - data dimension index

$\{D_{id}\}$ – the density of data sample x_i in dimension d

$\{Data\}$ - the set of all data

$\overline{\delta}$ - mean density change

$\overline{\delta}_{in}$ – smoothed value of previous n density changes up to sample i

i - generic index

j - index of selected sample to use for radius estimation

μ_{0d} Global Mean in dimension d

μ_j – the mean of cluster j

N – the number of samples being considered.

n – the number of density changes to use for smoothing (see Section 4.3 for explanation)

r_d – the radius estimation in dimension d

x_{id} – value data sample i in dimension d

X_{0d} Scalar product in dimension d

Input: $\{Data\}$ $n \in \{n|10 < n < 75\}$

1: $\forall d$

2: $\mu_{0d} = \frac{1}{N} \sum_{i=1}^{N} (x_{id} \in \{Data\})$

3: $X_{0d} = \frac{1}{N} \sum_{i=1}^{N} (x_{id} \in \{Data\})^2$

4: $D_{id} = \frac{1}{1+\|x_{id}-\mu_0\|^2+X_0-\|x_0\|^2}$

5: $\{D\} = \{D_{id}|D_{(n+1)d} > D_{nd}\}$

6: $\overline{\delta} = \frac{1}{N} \sum_{n=2}^{N} D_n - D_{i-1}$

7: $\overline{\delta}_{in} = \frac{1}{n} \sum_{i-n}^{i} D_i - Di = 1$

8: $j = argmin_{i=1}^{N}(\overline{\delta}_{in} > \overline{\delta})$

9: $r_d = \mu_{0d} - x_{jd}$

Appendix C

DDCAS Algorithm

This section provides the algorithm for Data Density based Clustering into Arbitrary Shapes (DDCAS). The mathematical steps of the algorithm, as implemented in Matlab, for the experiments in this thesis are given. These steps are not intended to represent the most computationally efficient method of implementing the algorithm but to implement a clear technique with all the steps clearly separated.

$\{C_j\}$ - the set of data in cluster j

d - data dimension index

$\{Data\}$ - the set of all data

g - macro-cluster number

i - generic index

j - index of a micro-cluster

k - generic index

μC - micro-cluster

μ_{0d} - Global Mean in dimension d

μ_j – the mean of cluster j

N – the number od data samples in $\{Data\}$

$\{O_k\}$ - the set of data in outlier cluster k

r_0 - the vector of initial radii

r_j – the vector of radii of cluster j

T_{min} - minimum threshold below which a micro-cluster is considered to be noise or outliers

x_{id} – value data sample i in dimension d

Input: $\{Data\}$, r_0, T_{min}

Initialization:

$j = 0$ {micro-cluster index}
$k = 0$ {outier-cluster index} *Main Function*
while $\{Data\} \neq \emptyset$ **do**
Find Global Densest Point (closest to the Data mean), Assign Data to μC
$j = j + 1$ {Increment candidate μC number}
$\mu_0 = mean\{Data\}$
$i = argmin(\|x_i - \mu_0\|)$
$\mu_n = x_i$ {set candidate μC centre}
$r_j = r_0$ {set candidate μC radius}
$\{C_j\} \ni \|(x_i \in \{C_j\}) - \mu_j\| < r_j$ {Assign data}
$\{C_j\} \ni \left\|(x_i \in \{C_j\}) - \mu_j\right\| < (3 \times \sigma(\{C_j\}))$ {Remove Outliers}
$r_j = mean\{\|C_j - \mu_j\|\}$ {set radii to mean distance to members}
$\mu_j = mean\{C_j\}$ {move centre}
$\{C_j\} \ni \|(x_i \in \{C_j\}) - \mu_j\| < r_j$ {re-assign data to new centre and radii ellipse}
$\{Data\} = \{Data\} - \{C_j\}$ {Remove $\{C_j\}$ from Data set}
if $|\{C_j\}| < T_{min}$ **then**
$k = k + 1$
$\{O_k\} = \{C_j\}$ {assign $\{C_j\}$ to outlier cluster}
$j = j - 1$ {decrement μC number}
end if
end while
Assign M_c *numbers to* μC *by merging*
$g = 1$
for $j = 1$ to i **do**
for $k = i$ to i **do**
if $r_j + r_k < 2r_0$ **then**
if g_j **then**
$g_k = g_j$
else
$g_j = g$
$g_k = g$
$g = g + 1$

```
31:             end if
32:         end if
33:     end for
34: end for
```

Appendix D

CODAS Algorithm

This section provides the algorithm for Clustering of Online Data into Arbitrary Shapes (CODAS). The mathematical steps of the algorithm, as implemented in Matlab, for the experiments in this thesis are given. These steps are not intended to represent the most computationally efficient method of implementing the algorithm but to implement a clear technique with all the steps clearly separated.

$\{C\}$ - the set of all micro-clusters

C_{min} - the minimum membership for a micro-cluster to be considered for merging

$\{C_\mu\}$ - micro-cluster centre, $C_\mu(i)$ micro-cluster information of micro-cluster i

$\{C_r\}$ micro-cluster radius

$\{C_n\}$ - number of micro-cluster members

$\{C_M\}$ - micro-cluster macro-cluster assignation

D - distances from new sample to micro-cluster centres

Δ - flag for updating orphan macro-clusters

i - generic index

$\{I\}$ - set of intersecting micro-clusters with indices $\{I\}(i)$ where i is: 1 - previous intersections; 2 - current intersections; 3 - new intersections; 4 - orphans that no longer intersect the changed micro-cluster's macro-cluster

j - generic index

k - index of changed micro-cluster

μ_{0d} - Global Mean in dimension d

μ_j – the mean of cluster j

N – the number of macro-clusters

M - macro-cluster number

x - data sample

Input: r_0, C_{min}, x, C

Initialization:

$N = |\{C\}|$ {count number of micro-cluster}
if $N < 1$ **then**
  Set up first micro-cluster
  $C_\mu(1) = x$
  $C_r(1) = r_0$
  $C_n(1) = 1$
  $C_M(1) = 1$
else
  Main function
  Assign data sample to micro-cluster
  $D = \|x - \{C_\mu\}\|$ {Distance from sample to micro-cluster centres}
  $i = arg[min(d)]$ {Find index of closest micro-cluster}
  if $D_i < r_0$ **then**
    If sample is within a micro-cluster, then up date the micro-cluster
    $k = i$ {Record index of changed micro-cluster}
    $C_n(i) = C_n(i) + 1$
    if $D(i) > 0.5 \times r_0$ **then**
      If samples is in micro-cluster shell, update micro-cluster centre
      $C_\mu(i) = \overline{C_\mu(i)}$
    end if
  else
    {Create new micro-cluster}
    $N = N + 1$
    $C_\mu(N) = x$
    $C_r(N) = r_0$
    $C_n(N) = 1$
    $C_M(N) = N$
  end if

{Update new micro-cluster intersections}
if k and $C_N(k) > C_{min}$ **then**
{If a micro-cluster has changed and is above the minimum threshold}
$\{I_1\} = arg[C_M = C_M(k)]$ {Find all the previous intersections}
$\{I_2\} = arg[\|C_\mu(k) < 1.5 \times r_0]$ {Find all current intersects}
if $\{I_1\} \neq \{I_2\}$ **then**
{If micro-cluster intersects have changed}
$\{I_3\} = \{I_2\} - \{I_1\}$ {Find new intersections}
if $|\{I_2\} > 0$ **then**
$M = max\{C_M(CI_3)\}$ {Find max macro-cluster number of intersecting micro-clusters}
if $\{I_3\} = \emptyset$ **then**
{If the changed micro-cluster has no intersections}
$C_M(k) = max\{C_M\} + 1$ {Assign new macro-cluster number}
else
$\{C_M(\{I_2\})\} = max\{C_M(\{I_2\})$ {Set the macro-cluster number for all interscting micro-clusters}
end if
end if
end if
else
Return
end if
{Update any micro-cluster that have separated from a macro-cluster}
$\{I_4\} = \{I_1\} - [\{I_2\} + \{I_3\} + k]$ {Find any micro-clusters that were previously intersected, but are no longer intersected}
if $\{I_4\} \neq \emptyset$ **then**
{Update all to new macro-cluster number}
$\forall i \in \{I_4\}$
$M = max\{C_M\} + 1$
$\Delta = 1$
while $\Delta = 1$ **do**
$\Delta = 0$
$j = arg_i[\|\{I_i\} - \{I_1\}\| < 1.5 \times r_0]$
if $|j| > 0$ **then**
$\Delta = 1$
$C_M(j) = M$
end if

```
52:         end while
53:     end if
54: end if
```

Appendix E

CEDAS Algorithm

This appendix provides the detailed algorithm for CEDAS. The mathematical steps of the algorithm, as implemented in Matlab, for the experiments in this thesis are given. These steps are not intended to represent the most computationally efficient method of implementing the algorithm but to implement a clear technique with all the steps clearly separated.

C_i^{μ} - micro-cluster $'i'$ data structure containing:

$C_i^{\mu}(Centre)$ - $vector \in \mathbb{R}$ with $length = number\,of\,dimensions$, micro-cluster $'i'$ centre co-ordinates

$C_i^{\mu}(Count)$ - $integer$, number of data samples that have been assigned to micro-cluster $'i'$

$C_i^{\mu}(Macro)$ - $integer$, micro-cluster $'i'$ macro-cluster membership

$C_i^{\mu}(Energy)$ - $Energy \in \mathbb{R}$, current value of assigned to micro-cluster $'i'$

$C_i^{\mu}(Siblings)$ - $vector\,of\,integers$, list of $'Sibling'$ micro-clusters linked to micro-cluster $'i'$

C^{μ} for all the above, but without subscript refers to all micro-clusters

d_i - $\in \mathbb{R}$, distances from new data sample to the micro-cluster centre i

d_{min} - $\in \mathbb{R}$, distance to the nearest micro-cluster centre

$\{D\}$ - $vector\,of\,integers$, set of indices of dead micro-clusters. (For the decay process described here this is a vector of length 1).

$Decay$ - $\in \mathbb{R}$, rate at which $C_i^{\mu}(Energy)$ is reduced

G - temporary variable for re-assigning macro-cluster numbers

i - *integer*, index value

N_c - *integer*, number of micro-clusters

r_0 - $\in \mathbb{R}$, micro-cluster radius, user input

x - $vector \in \mathbb{R}$ with $length = number of dimensions$, current data sample

u - *integer*, index of updated or created micro-cluster

Input: x, R_0

Initialization:

while $x \neq \{\}$ **do**
  if $C^{\mu} = \emptyset$ **then**
    $C_1^{\mu}(Centre) = S$
    $C_1^{\mu}(Count) = 1$
    $C_1^{\mu}(Macro) = 1$
    $C_1^{\mu}(Energy) = 1$
    $C_1^{\mu}(Sibling) = 1$
    $N_c = N_c + 1$
    $u = N_c$
  end if
  Update Micro-Cluster:
  $u = 0$
  $d_{min} = \left|\left|x - C_i^{\mu}(Centre)\right|\right|_{min}$
  if $d_{min} < r_0$ **then**
    $i = argmin_{j=1}^{K}\{d_j\}$
    $C_i^{\mu}(Energy) = 1$
    $C_i^{\mu}(Count) = C_i^{\mu}(Count) + 1$
    if $d_{i(min)} > \frac{R_0}{2}$ **then**
      $u = i$
      $C_u^{\mu} = \frac{(C_u^{\mu}(Count)-1)\times C_u^{\mu} + S}{C_{u(Count)}^{mu}}$
    end if
  else
    $C_1^{\mu}(Centre) = x$
    $C_1^{\mu}(Count) = 1$
    $C_1^{\mu}(Macro) = 1$
    $C_1^{\mu}(Energy) = 1$

$C_1^{\mu}(Sibling) = 1$
$N_c = N_c + 1$
end if
Kill Clusters:
$C_i(Energy) = C_i(Energy) - Decay$
$\{D\} = find(C_i(Energy) < 0)$
if $\{D\} = \emptyset$ **then**
return
else
for all D_i:
delete $C_{D_i}^{\mu}$
delete $C(Sibling) = D_i$
$C_i^{\mu}(Sibling) > D_i = C_i(Sibling) - 1$
$N_c = N_c - 1$
end if
if $u \neq 0$ **then**
$d_{ui} = \left|\left|C_u^{\mu}(Centre) - C_i^{\mu}(Centre)\right|\right|$
$\{j\} = find(D_{uj} < (1.5 \times r_0))$
$C_u^{\mu}(Sibling) = C_u^{\mu}(Sibling) \cup \{j\}$
$G = min\{C_{C_u(Sibling)}(Macro)\}$
$C_u^{\mu}(Macro) = G$
$C_{C_u(Sibling)}^{\mu}(Macro) = G$
end if
Housekeeping: Reassign Macro Cluster Numbers
$C^{\mu}(Macro) = 0$
$G = 0$
for $i = 1 to N_c$ **do**
if $C_i(Macro) = 0$ **then**
$G = G + 1$
$C_i^{\mu}(Macro) = G$
$C_{C_i^{\mu}(Sibling)}^{\mu} = G$
else
$C_{C_i^{\mu}(Sibling)}^{\mu} = C_i^{\mu}(Macro)$
end if
end for
end while

Appendix F

RASCAL Software

This Appendix provides an overview of the operation of the Real-time Atmospheric Science Cluster AnaLysis (RASCAL) software. The software is intended to demonstrate the capabilities of the techniques employed, typical use-age and the type of output produced.

F.1 Overview

RASCAL in its current form runs in a simulated real-time environment using data stored in a simple comma-delimited (csv) format text file. The data is loaded and then presented to the RASCAL analysis routines 1 sample vector at a time thus simulating a live data feed. The data and analysis are presented in a number of graphical windows for interpretation by the expert user (Mission Scientist). The data to be traced and monitored are fixed at the start of the flight and cannot currently be changed in-flight. The selectable data clusters can be changed in flight and the data overlays on the flight map can be switched on and off or adjusted. When data groups are selected data is displayed in all windows in the colour corresponding to the data groups.

F.2 RASCAL Initialisation

The initialisation screen for RASCAL has 3 distinct sections as indicated in Figure F.1. Section A contains parameters relating to the initial state of the plots and the data to be monitored throughout the flight. Section B allows selection of the data streams to be used for the flight path and timing. Section C contains relevant parameters related to the clustering algorithms and alpha hull displays. The values in section C would not normally be adjusted by a lay-user and should not be modified without a firm understanding of

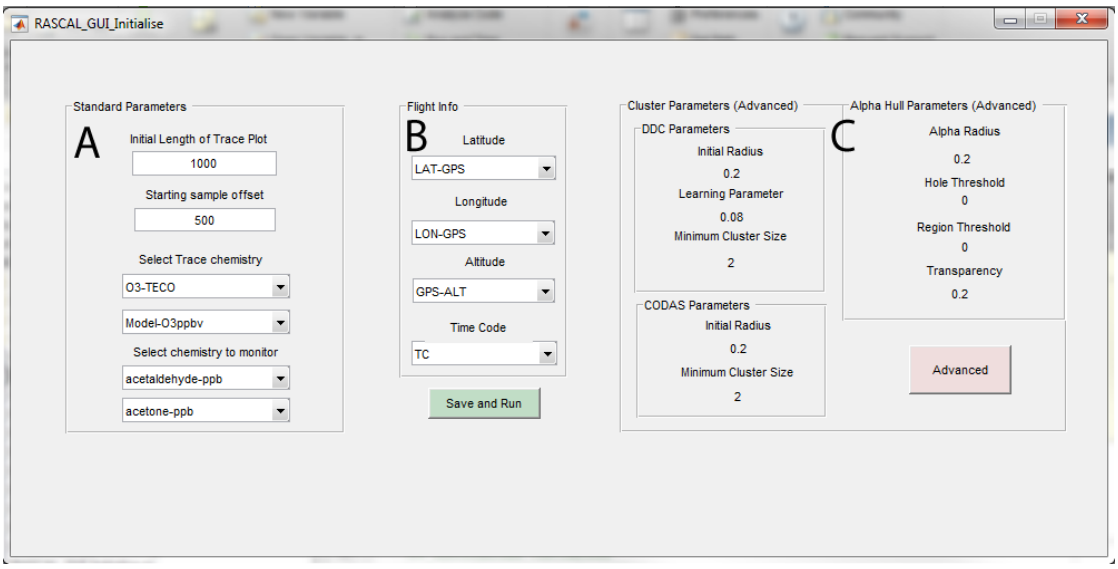


Fig. F.1 RASCAL initialization screen showing (A) trace plot information, (B) flight information and (C) clustering and visualization parameters.

the related techniques. For detailed information about these parameters, see the relevant Chapters on the respective clustering techniques.

F.3 Set Up and User Parameters

The parameters in Section (A) of the initialization screen are largely self-explanatory. The first first relates to the initial length of the plot tracing the chemicals selected below. The trace plot always tracks the aircraft altitude. The starting sample off-set value relates to testing the software on data from a stored file. In many cases the data collections starts long before take-off and so a large amount of irrelevant data may be included. This offset allows for by-passing that data for the sake of convenience.

The next two selections are the chemistry to trace during the flight. The chemistry that is selected for monitoring in the next two menus is that chemistry that will be continuously clustered, online, using CODAS (or CEDAS if implemented later). The chemistry that is monitored may be the same as that which is traced if required.

Section (B) allows for selection of the various aspects of the flight data as different data sets, or aircraft equipment, may label these variables differently.

Section (C) allows for varying the parameters of the cluster analysis and alpha hulls used for visualization. For a detailed explanation of these see the relevant Chapters pertaining to each technique.

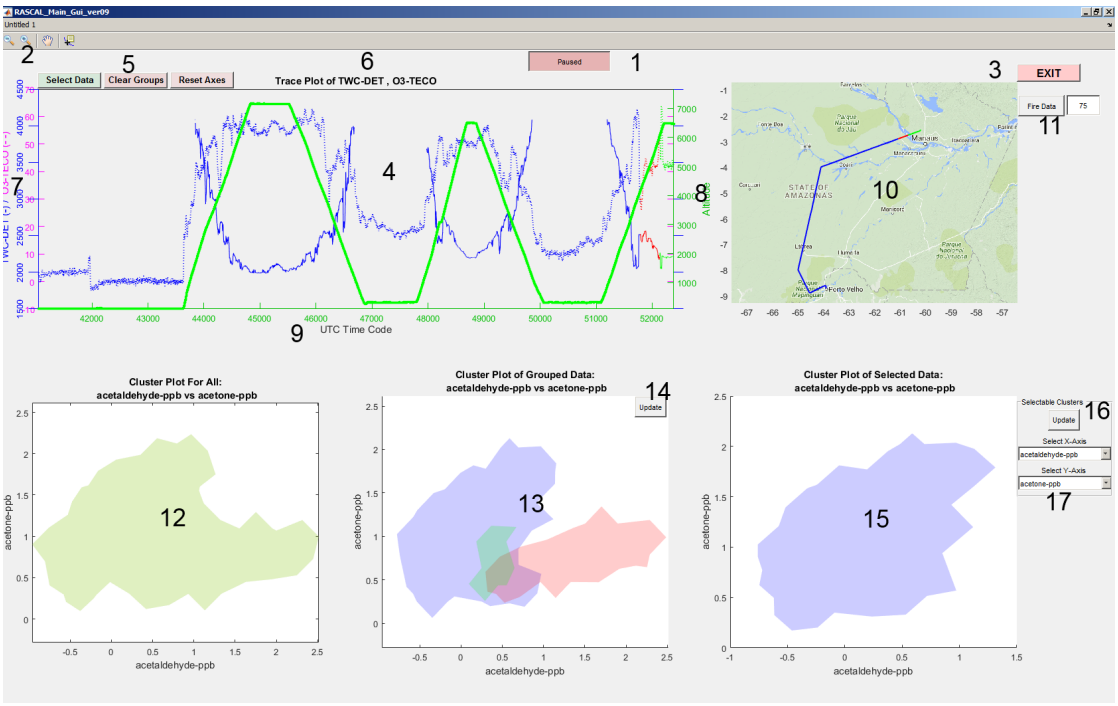


Fig. F.2 RASCAL operating screen showing the items discussed in section F.4

.

F.4 RASCAL Operating Screen

This section provides a detailed description of the various parts of the RASCAL main operation screen, i.e. the screen visible to the mission scientist throughout the flight.

1. Pause Button: Pauses the data collection and analysis. This is used for pausing the display for discussion and screen captures.

2. Zoom and Pan Buttons: Some windows can be zoomed with a mouse scroll wheel button. Using these buttons allows pan and zoom on any window.

3. Exit Button: Halts all analysis and exits immediately

4. Trace Plot Window: Shows the on-line trace data.

5. Data Selection Buttons: When data of interest appears these buttons can be used to select data in the trace plot window to form groups. Any analysis carried out on these groups is coloured to match and the flight path colour is also changed to illustrate the data source.

6. Trace Plot Title: Title of the trace plot showing the names of the tracers selected.

7. Trace Plot Scales: The names of the tracers and the scales are coloured independently for clarity. These colours do not match the trace lines as the trace lines are coloured by group (see 5). The scales automatically adjust to include the displayed data.

8. Altitude Scale: The scale for the flight altitude. This automatically adjusts to the displayed data.

9. Time Scale Axis: Shows the time scale as selected in the set up screen.

10. Map Window: A display of the map region. This is dependent on the intended flight area initially but can be panned and zoomed. Maps are downloaded from Google and require an internet connection. Off-line maps could be easily incorporated.

11. Map Plot Overlays: Overlays of alternative data sources can be incorporated. Here we have included fire data from the NASA FIRMS database, [117]. This is not live data but is downloaded prior to the flight.

12. CODAS Cluster Window: The results of CODAS clustering on the data received so far.

13. DDC Cluster Window: When data is selected in to groups DDC off-line clustering can be performed on these groups and is displayed here. The clusters for each data group are coloured to match the other displays.

14. DDC Update Button: To update the DDC plot this button is pressed. It currently operates as a toggle button and DDC runs continuously until pressed again. With large volumes of data this slows down RASCAL and it may drop below real-time so it is not recommended to run it continuously.

15. DDC on Selected Data Window: This window displays the results of DDC off-line clustering of the data groups on any data selected in 17.

16. Selected Data Update Button: This button force an update to the DDC cluster results on the selected data.

17. Data Selection Menus: Drop down menus allow selection of any of the available data streams.

F.5 RASCAL Offline

This section of he appendix describes some of the enhancements implemented in the offline version of RASCAL. The operating screen remains essentially the same. The

underlying changes to the clustering algorithms used are described in Chapter 6.7 and this section deals with the use of the software. A number of enhancements based on feedback on the usability have been implemented. In particular these include:

1. Trace Plot Selection: The data used for the trace plots can be selected form the drop down menus.

2. Trace Plot Colours: for clarity, when no data groups have been selected for analysis the trace plots are separate colours and match by colours of the drop down menus for selecting the trace data.

3. Trace Plot Data Selection: Selection of groups of data in the trace plots has been simplified; left click to add a group edge at the mouse cursor position; right click to delete the nearest edge to the mouse cursor.

4. Selecting Data for Analysis: data can be selected in any visualization window, not just on the trace plot by clicking the 'select data ' button and drawing an area in the windows. Any data sample within the drawn area are selected.

5. Pan and Zoom: all the visualization window can be panned and zoomed.

Further feedback on these changes has been positive and these, together further enhancements should be implemented in the online version as well in the future.