A machine learning investigation of asteroid classification in the optical regime

by

Tricia Sullivan

A thesis submitted in partial fulfillment of the requirements of
Liverpool John Moores University
for the degree of
Doctor of Philosophy

September 2023

Declaration

Tricia Sullivan
Astrophysics Research Institute
Liverpool John Moores University
146 Brownlow Hill
Liverpool
L3 5RF
UK

September 2023

Abstract

Historically, asteroids have been characterised by features of their reflectance spectra in the optical and near-infrared (NIR) as well as their albedos. The Bus-DeMeo asteroid taxonomy and its closely-related predecessor (Bus-Binzel) both have their roots in spectrophotometric work from the 1970s based on principal component analysis (PCA) and clustering of small (few hundred) asteroid samples. The known asteroid population has grown exponentially since those times, and the taxonomies have been applied to as many as one million asteroids in the Sloan Digital Sky Survey (SDSS). Whether the classes they describe remain valid in a contemporary context is the topic of this thesis.

I used a range of machine learning techniques to investigate the robustness of asteroid classes in approximately two thousand reflectance spectra as well as in the photometric colour data available in three published catalogs extracted from the SDSS Moving Object Catalog. Beginning with spectra in the wavelength range $0.5 - 0.9$ μm I found that a support vector machine (SVM) classifier can identify 85% of asteroids correctly when considering the major classes (A, B, C, D, K, L, Q, S, V, X), but subclasses are not distinguishable from one another and/or their parent classes. Furthermore, the 15% of wrongly-classified objects include a high proportion of classes B, K, and Q being assigned to neighbouring classes in feature-space. The SVM performs better on the full 161 spectral wavelength points than on data that have been transformed with PCA, indicating that there is no need to reduce the dimensions before training.

In order to test an SVM on the large SDSS asteroid datasets, I generated a 'pseudo-broadband' training set by taking the average reflectance values of the spectra over a range corresponding to SDSS r, i, and z-bands. After training the SVM on this low-resolution data, classification accuracy was diminished by only 5%, but every A, B, K, and Q object was assigned to a neighbouring class. In the transition to broadband photometry these classes ceased to exist as far as the SVM was concerned.

For each dataset used, catalog classes had already been assigned according to either the Bus-Binzel (for spectra) or Carvano or DeMeo (for SDSS, respectively) systems. Objects having data in both SDSS and spectrum catalogs had been assigned to different classes in $\sim 30\%$ of cases, creating a challenging situation for the SVM since it cannot match to a conflicting label. In examining the SDSS datasets (one classified by the Bus-Binzel system and the other two by the Bus-DeMeo system) I discovered that the mean SDSS broadband values in each of bands r, i, z were brighter than pseudo-broadband values for the same object. After correcting for the difference in each band, the SVM was able to predict with 85% accuracy on the Carvano dataset and 74% on the DeMeo data. However, classes A, B, K, L, and Q were not verifiable by the SVM.

Since the above classes have small populations relative to the S-class, I set about increasing the sample size of the smaller classes. I did this using two augmentation methods: Synthetic Minority Oversampling Technique with Edited Nearest Neighbours (SMOTE-ENN) and a variational autoencoder (VAE). SMOTE-ENN interpolates between existing datapoints when resampling, whereas the VAE learns from existing samples and produces new spectra by sampling from a Gaussian distribution around each dimension of the data. After the SVM was trained on SMOTE-ENN, it was able to classify to 91% accuracy on the spectrum test set and 88% on pseudo-broadband. VAE achieved 92% accuracy at spectrum level and 90% at pseudo-broadband. Both augmentation methods recovered all of the 'missing' classes, but to poor accuracy.

I found that performance of augmented models fell by $30 - 40\%$ when testing on SDSS data. SMOTE-ENN's method of interpolation altered the distribution of data within class boundaries in a non-random way, effectively introducing artificial substructure in feature-space. The result was highly overfitted class boundaries that became obvious when examining plots of SMOTE-ENN's predicted classes on SDSS data. The VAE model was able to correct for some class imbalance, especially in the C-complex, but its accuracy results ran to 58% at best. The reason for much lower accuracy scores comes from the increased population of classes that are poorly-defined in the first place. Rather than improving the classifier, augmentation by the VAE exposes the weakness in claiming that classes A, B, K, L and Q exist at all in the SDSS.

Finally, I used unsupervised learning to search for evidence of asteroid classes in unlabelled data using K-Means, Gaussian mixture model (GMM) and Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN). After training and testing on spectra, PCA of spectra, and spectra augmented with both SMOTE-ENN and VAE, I found no evidence of 'real' classes apart from the C-complex and S-complex cores, which have been known since the 1970s. HDBSCAN characterises a large fraction of the data as noise.

I conclude that the variational autoencoder is a viable method to correct bias in the classification model without introducing new biases, and that SMOTE-ENN produces data prone to overfitting. With the VAE model in the $0.5 - 0.9$ μm regime, classes A, B, K, L, and Q are ambiguous at pseudo-broadband resolution and therefore their application to SDSS data is unreliable.

Tricia Sullivan September 2023

Contents

List of Figures

List of Tables

Acronyms

AGB	Asymptotic Giant Branch. 22
AGN	active galactic nuclei. 1
AIC	Akaike information criterion. 62
ALMA	Atacama Large Millimeter/submillimeter Array. 19
AMC	Asteroid Mining Corporation. 1
AU	Astronomical Unit. 15
BIC	Bayesian information criterion. 62
CAI	Calcium-rich inclusions. 23
CC	Carbonaceous chondrite. 21
EC	Enstatic chondrite. 22
EM	Expectation-Maximization. 48
GMM	Gaussian mixture model. 51
HDBSCAN	Hierarchical Density-Based Spatial Clustering Algorithms with Noise. 54
IR	Infrared. 4
IRAS	Infrared Astronomical Satellite. 4
IRTF	Infrared Telescope Facility. 10
KL	Kullbeck-Leibler. 43

MBA Main-Belt Asteroid. 3

ML Machine learning. 26

MLP Multi-layer perceptron. 29

MOC Moving Object Catalogue. 10

MSE Mean squared error. 29

NEA Near-Earth Asteroid. 3

NEOWISE Near-Earth Object Wide-field Infrared Survey Explorer. 4

NIR Near-infrared. 5

OC Ordinary chondrite. 21

PCA Principal component analysis. 7

PDF Probability distribution function. 54

RELAB Reflectance Experiment Laboratory. 24

ROC-AUC Receiver Operating Characteristic - Area Under the Curve. 29

S3OS2 Small Solar System Objects Spectroscopic Survey. 12

SMOTE-ENN Synthetic minority oversampling technique with edited nearest neighbors. 40

SSE Sum of squares error. 57

SVM Support vector machine. 29

TPM Thermophysical model. 3

UV Ultraviolet. 25

VAE Variational autoencoder. 29

VLBA Very Long Baseline Array. 4

WISE Wide-field Infrared Survey Explorer. 4

XGB Extreme gradient boost. 29

YORP Yarkovsky–O'Keefe–Radzievskii–Paddack. 4

Acknowledgements

The first person I am going to acknowledge is myself, because quite honestly nobody but me can know what I have been up against in the elevenish years it has taken to get from learning the definition of cosine to this moment.

Secondly I offer profound thanks to my primary supervisor, Professor Iain Steele. I was assigned to Iain at the beginning of the LIV.DAT CDT programme to work on something like 'The LSST Data Deluge,' but I didn't know how to code, much less handle a 'data deluge'. Iain set me up with a starter project on Boyajian's Star, helped me construct a plan for a PhD on AGN variability, got me a work placement with the Asteroid Mining Company, and then helped me change my project (twice) when plans weren't working. Everything I've attempted, Iain has supported and guided in a way that made me feel he had faith in me even when I didn't. I have been incredibly lucky to have him as my supervisor.

I'm also grateful to my co-supervisor Dr Paul Fergus, who has been wonderful in making sure I did not go off the rails with the machine-learning side of things. In the beginning I was extremely intimidated, but Paul has done everything to put me at my ease and has made valuable suggestions. I'd also like to thank co-supervisor Dr Chris Copperwheat for his encouragement and patience. The instructors of the data science courses that I attended in my first year were instrumental in my survival: Dr Daniel Perley, Dr Ivo Siekmann, and Dr Danushka Bollegala. Thank you all.

I do not recommend anyone doing an astrophysics PhD during perimenopause in a pandemic with multiple teenagers, but these things happen. I would like to thank the team at Pontesbury Medical Practice and Dr Pav Dhesi for working with me to get the right HRT, which has made an enormous difference to my ability to function especially in the last six months. The same team of people, together with Dr Sushma Rao, Dr

Rudolph Venter, Susan Darker-Smith, and staff at The Autism Centre and Sea Change Psychotherapy, have been brilliant with providing care for family members during an extended period of crisis during this PhD. Without their involvement I could not have gone on.

I am grateful to the STFC for pandemic-related financial help and to the ARI for providing the additional funds that allowed me to complete. Neither of my parents had the opportunity to go to college, but they went above and beyond for me, so I also thank my mother, Marion Sullivan. She sent me the money for a used Aygo that took me between Shropshire and Liverpool, and throughout my entire life she and my late father, Denis, were unflagging in their love and acceptance.

I avoided physics in high school and I got a D in Algebra 2 because instead of paying attention in class I was writing stories about exploding toilets with my friend Kim. That genre never took off for either of us, so I thank the Open University for making it possible for me to start learning maths and physics affordably at a time in my life when in-person learning would have been completely impossible. I especially thank Dr Cath Wilkins for her brilliant applied mathematics tutorials and the late Dr Iain Chapman for helping me with thermodynamics when I was pole-axed following the death of my father. I'm grateful to Professor Daniella Bortoletto and the other organisers of the 2014 Conference for Undergraduate Women in Physics, because of which I started to believe I might be capable of doing a PhD in this field. I thank Professor Phil James and my MSc supervisor Professor Ivan Baldry together with everyone involved in the ARI's MSc by distance learning, which enabled me to continue studying part-time after I ran out of road with the Open University. These unconventional access points to STEM changed my life.

I am a little bit of a lurker by nature, but being a part of the ARI community has meant more to me than I can say. I thank Danielle Coogan, Anna Hodgkinson, Maureen Pattullo, and Caroline Ramsay for their kindness and efficiency in helping us students navigate the university system. I thank my fellow ARI PhD students, who are all awesome and brilliant and kind. It has been a privilege. I'm grateful to Dr Helen Jermak for taking me under her wing especially at the difficult beginning, to Dr Marie Martig for the tissues and good advice, to Dr Fiona Murphy-Glaysher for her ongoing support, and to the LIV.DAT crew Alex, Joaquin, Alberto, Rob and Brandon for being awesome and making me laugh. Thanks to Dr Ross McWhirter for introducing me to autoencoders and encouraging me in general, and to Dr Marco Lam, Dr Egidijus Kukstas, Dr Doug Arnold, Dr Dan Harman, Dr ‘Zej Piascik and Dr Simon Pfeiffer for technical help of various kinds. I thank Dr Hannah Dalgleish, Dr Rhana Nicholson, and Dr Kate Furnell for lifting me up in my first year and Dr Manisha Shrestha for her personal kindness and generosity. I thank Dr Victoria Sheppard, Dr Jo McKeon, and the staff at Gladstone’s Library for the unforgettable Thesis Bootcamp. To anybody that I should have thanked and forgot to: please forgive the omission. I’m running on my last 2.3 brain cells.

I thank everyone at Stack Overflow, *sine qua non*.

Massive thanks to Stephanie Burgis, Cath Nockolds, and Karen Mahoney for putting up with me. I thank my soul-sister Caroline Holley for the weekly Zoom calls, ridiculous e-mails, recycled wool afghans and fingerless gloves that have seen me through many a chilly home-office work sesh, and for telling me the story behind our mantra *D is for Diploma*. I thank my sister Kathy for talking me through a crisis of confidence by telling me, ‘You have no idea how many stupid people have doctorates.’ It was exactly the right thing to say.

I thank my partner Steve for teaching me to fight back no matter what, and for having my back. Because of him I know that I can kick a door off its hinges. Everything else is gravy.

Finally, to Tyrone, Rhiannon, and Sean Morris: thank you for walking this road with me. I am hugely grateful to be a part of your lives. Here’s to the future.

The author of this thesis is supported by an STFC doctoral studentship within the Liverpool Big Data Science Centre for Doctoral Training, hosted by Liverpool John Moores University and the University of Liverpool [ST/P006752/1].

All analyses and plots presented in this thesis were produced using `Python`, most notably its packages `numpy, pandas, matplotlib, scikitlearn, keras, tensorflow, seaborn, astropy, scipy, imblearn`.

Tricia Sullivan September 2023

'...but the will must be stronger than the skill.'

Muhammad Ali

'Keep it up and go on'

Rex Orange County

This work is dedicated to my mother, Marion Sullivan, who has always been fascinated by science but spent her life caring for others. Thank you, Mom.

Chapter 1

The historical context of asteroid classification

1.1 Reflection on the project

Asteroids are a sensational topic. When they are in the news, it's usually because an asteroid big enough to cause impact damage has made a close pass to Earth. At the other extreme, they are seen as sources of great wealth for extractive hypercapitalist industry – after all, a few may contain rare metals in mind-boggling concentrations. People even view asteroids as humanity's bridge to space colonisation (Elvis, 2012). With all of this going on it is easy to forget that asteroids are also repositories of information about the history of the Solar System, from primordial times to the more recent delivery of water and organic molecules to Earth. In this chapter I will briefly outline the history of asteroid classification as well as what we know about the role of asteroids in the developing Solar System. But first, a few words about the project itself.

This work came about as a result of an industrial placement with the Asteroid Mining Corporation (AMC) that I undertook in late 2019 as part of the LIV.DAT doctoral training programme in data-intensive science. As a newcomer to data science, I was meant to acquire hands-on experience with real-world machine learning before returning to study optical variability of active galactic nuclei (AGN) light curves. The AMC is a start-up dedicated to mining asteroids for betterment of life on Earth; it has engineering and business experts on staff, but its contact with astronomers is through a relationship with the Liverpool Telescope via my primary supervisor, Professor Iain A. Steele. Because of this, my work at AMC was largely self-directed. AMC were planning to launch a CubeSat (miniature satellite) to record asteroid data from low Earth orbit; one question that had arisen was whether to mount a spectrograph or a photometric filter wheel on the cubesat. In order to estimate how effectively two different proposed instruments could pick out the commercially valuable (metallic) X-types from the asteroid population in general, I compared results of machine learning classifiers on each type of data. The results showed that when training algorithms to identify X-type asteroids, low-resolution multi-filter photometry performed as accurately as detailed reflectance spectra. This was rather surprising and made me wonder what would happen if the method were extended to other classes—and furthermore, which of the many asteroid types are 'real' from the point of view of an algorithm. Around the same time that I was thinking about this, COVID-19 began to cause difficulties in my personal life. Faced with an indefinite period of disruption, I took a parsimonious approach to both time and energy and decided to continue my work with asteroids instead of returning to AGN.

The questions I address are: When we look at optical asteroid reflectance spectra through the lens of statistics, are the asteroid classes 'real'? Are they real when we reduce resolution to a few bands? What about when we introduce large numbers of new objects without spectroscopy — do the classes still mean anything? As I shall show later in this chapter, it turns out that the asteroid science community has already accepted that asteroid classes do not equal mineralogical types and that they do not live in discrete clumps in colour-space, so why am I tilting at windmills? It is true that I do not undertake this work because there is a burning curiosity in the scientific community to reify asteroid classes, but rather because I am interested in what happens when the rubber of machine learning hits the road of nature. As a beginner in both machine learning and astronomy, I want to find out how these sophisticated statistical tools can be used to understand asteroid classification—and how they can't be used.

This thesis documents my learning process. It sits on the boundary between astrophysics and machine learning, and it probably embodies the bumpiness of the transition between the older analytical/small data approach to astronomy and the rising wave of today's big data approach in which so-called artificial intelligence is central. I have tried to present enough astrophysical context to explain the science, together with enough statistical/computational context to explain the machine learning. Of course, breadth comes at the expense of depth. Additionally, I must point out that my machine learning is not baked from scratch; I use tools that are available in Python packages without the need to write one's own implementations. These tools have enabled me to punch above my weight as a programmer, but they also occasionally permitted me to do stupid things without knowing it until I found myself deep in some rabbit-hole of wrong conclusions. Yet it is because I am limited to user-friendly tools that this work has relatively broad implications for other astronomers exploring this liminal space. I hope that my findings may offer some insight to both the challenges and the promise of applying garden-variety machine learning techniques to astrophysical questions.

The rest of this chapter is an introduction to asteroid science. Section 1.2 covers the fundamentals of asteroid measurements. In section 1.3 I outline the history of asteroid classification systems and briefly introduce those used in this project. Section 1.4 presents a very simplified set of possible Solar System histories that could result in the asteroid belt seen today. In 1.5 I touch on the relationship between meteorite classifications and the evolution of asteroids. Section 1.6 describes the processses that are known to alter asteroid surfaces and, in turn, their spectra. Section 1.7 summarises the above, while in Section 1.8 I outline the structure of the remainder of this thesis.

1.2 Fundamentals of observational physics

Asteroid knowledge comes from a wide range of observational techniques and their associated models. These include optical photometric lightcurves, thermal lightcurves, radar interferometry, reflectance spectra, multi-band photometry, and polarimetry. Physical characteristics can be derived from observations using thermophysical models (TPMs). In this section I outline how asteroids are measured.

1.2.1 Photometric light curves

Photometric lightcurves provide the angular positions that are used to derive the six orbital parameters necessary to solve the equations of motion for a small body orbiting the Sun (Collins, 2004).

Asteroid orbits are implicitly linked to their origins. Most asteroids live in the Main Belt, a region between Mars and Jupiter that represents the residue of the disk that formed the Solar System. The Main Belt is only slightly elliptical; however, gravitational effects, collisions, and the Yarkovsky effect can all alter the orbits of Main-Belt Asteroids (MBAs). Jupiter's strong gravitation has captured two groups of asteroids called the Trojans, which orbit the Sun at Jupiter's stables Lagrange points L4 and L5. Jupiter is also implicated in collisions that resulted in detectable asteroid dynamical families. For some asteroids, perturbations and collisions have changed orbital inclination above and below the plane of the Solar system. Increases in eccentricity resulted in asteroids crossing the orbits of Mars and Earth, leading to the populations of Mars-crossers and Near-Earth asteroids (NEAs) (Nesvorný, 2018). Classification systems recognise V-types, which are orbitally associated with the asteroid Vesta, and T-types (connected to Jupiter Trojans); however, these families have distinct spectral characteristics that permit this distinction. Otherwise, orbital families are not included in current classification systems.

Photometric light curves are valuable beyond orbital calculations. Good light curves also enable asteroid shape, rotation, and pole direction to be modelled through the method of light curve inversion Kaasalainen & Torppa (2001); Kaasalainen et al. (2001).

1.2.2 Asteroid size measurement

An asteroid's size is key to determining both its albedo and its thermal properties. In some cases, size can be calculated using stellar occultation, in which the asteroid is recorded transiting a star along the line of sight of the telescope (Ferreira, J. F. et al., 2022). The most accurate measurements of asteroid size and shape come from high-resolution delay-Doppler radar, which can produce detailed imaging of an asteroid surface, provided that the object is close enough to return a strong signal. Spin vectors can be estimated from the radar speckle technique, in which a monochromatic coherent radar signal is aimed at the asteroid by a transmitter such as Arecibo and received by an array system such as the Very Long Baseline Array (VLBA). The returning signal consists of high-and-low-intensity interference patterns called 'speckles' whose length scale depends on the distance to the object and its diameter; because speckles move with the surface, they also provide pole rotation sense/rate and pole direction (Benner, 2015). Due to its very high resolution, radar provides excellent astrometry that has been used to measure the Yarkovsky and Yarkovsky–O'Keefe–Radzievskii–Paddack (YORP) effects, which in turn have the potential to inform on the thermal inertia of the object (D. Vokrouhlicky, 2015), a proxy for its surface composition.

1.2.3 Thermophysical models

TPMs relate emitted energy (absorbed incoming Solar radiation plus scattered light from the asteroid's surface plus re-absorbed thermal self-emissions) to the thermal energy radiated into space (including heat conducted from deeper layers). The models rely on thermal infrared (IR) measurements taken from space with thermal noise controlled by cryogenic cooling. The Infrared Astronomical Satellite (IRAS), the Spitzer Space Telescope, AKARI, the Wide-field Infrared Survey Explorer (WISE) and its successor the Near-Earth Object Wide-field Infrared Survey Explorer (NEOWISE), and the Herschel Space Observatory have produced among them a large pool of asteroid data in thermal wavebands ($\lambda > 4\mu$m).

As described in Delbo' et al. (2015), TPMs break up the surface of the asteroid into triangular facets rotating about a spin vector to derive a temperature map across the surface of an asteroid. When the facets are summed, the spectral energy distribution at a given wavelength can be found from the Planck function, with the central wavelength being a function of distance to the Sun and the object's thermal properties (Mainzer et al., 2015). With phase angle and a thermal IR lightcurve, TPMs can be used to derive a size, shape, and albedo as well as rotation information. Rotation involves day/night temperature variation at the asteroid surface, the physics of which can result in the Yarkovsky and YORP effects. In these related phenomena, thermal emission from the cool side acts as a small thrust that can disturb the orbit and spin of a small asteroid (D. Vokrouhlicky, 2015). The Yarkovsky effect is thought to be an important mechanism in the migration of the NEA population from the Main Belt.

1.2.4 Albedo

Geometric albedo is defined as the radiation of an object relative to a Lambertian source (i.e., with uniform emission across a surface). Albedo measurements involve comparing the physical size of the object to its reflective brightness. Radiometry is a widely-used method for determining albedo, not least because it is suitable for large suites of objects. In broad terms, radiometry uses simultaneous visible and infrared measurements and performs a calibration to determine the relationship between size and brightness. Figure 1.1 illustrates how this works: dim asteroids radiate relatively more in the infrared than in the optical, so if an object is bright in both wavebands then it is large with a low albedo; but if it is bright in the optical and dim in the infrared, then the object is small with a high albedo.

Albedo was used in the Tholen classification system in the 1970s but in the twenty years between the introduction of the SMASS system Bus & Binzel (2002b) and the recent publication of Mahlke, M. et al. (2022) it was sidelined as the field focused on reconciling optical and Near-infrared NIR spectra with large broadband surveys such as SDSS.

If the object is bright enough, albedo can also be obtained from the degree of linear polarisation of light reflected from the asteroid surface. This is not a method suited to large surveys.

1.2.5 Reflectance spectra

When broadband light strikes a material, its constituent wavelengths are reflected or absorbed in a spectrum that is characteristic of the material's chemical makeup. Asteroid reflectance spectra are obtained from a telescope equipped with a spectrograph, an instrument that splits reflected sunlight into narrow wavelength ranges and measures the intensity at each wavelength. After removing the Solar contribution to each wavelength,

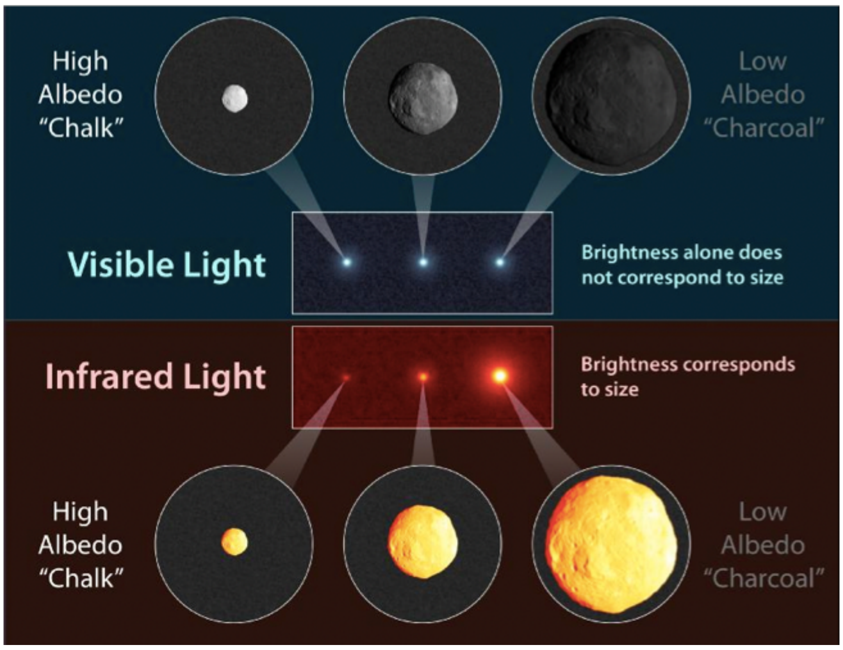


FIGURE 1.1: Illustration of how infrared emissions can tell us the albedo of an asteroid. From NASA/JPL-Caltech `https://www.nasa.gov/mission_pages/WISE/multimedia/gallery/neowise/pia14732.html`

intensity is plotted from short to long wavelength, and the features that remain offer insight as to the materials on the asteroid surface.

Much of we know about asteroid composition is based on comparing reflectance spectra to laboratory mineral spectra, especially from meteorites; this process has been shown to work well so far, at least in terms of surface properties. For example, reflectance spectra did a good job of predicting the surface composition of S-type asteroid Itokawa when compared to regolith samples collected by the Hayabusa mission and analyzed in the lab (Nakamura et al., 2011).

Most asteroid spectra have a positive slope[1]. The most common near-Earth asteroids, the SMASS S-group (which includes K, L, Q and R-types in addition to S-types) (Bus & Binzel, 2002a), have a strongly positive slope with an absorption feature around 1 μm associated with the presence of silicates (stony material). This means that their optical peak occurs at around 0.70 μm. In contrast, C-types (carbonaceous chondrites) and X-types (metallic) have a relatively flat, unpeaked slope in the optical, and are more difficult to separate from one another. Optical slope together with gross spectral features (e.g., bumps/absorption regions) have historically been treated as parameters for asteroid classification. See Figure 1.9 for an illustration of the variation of asteroid spectral types according to the major taxonomies.

[1]Sometimes called 'reddening' because the surface reflects more strongly at longer 'redder' wavelengths

The use of broadband colour photometry is an low-resolution extension of reflectance spectra that allows some measurement of asteroid colours, leading to statistical studies of asteroid populations. In both detailed spectra and broadband colour photometry, the flux measurement does not inform on the reflectivity of the asteroid; i.e., its albedo, which must be obtained independently.

1.3 The development of asteroid classification

The early history of asteroid classification is described engagingly in the PhD thesis of Tholen (1984), from which I have constructed this summary. A graphic summarising asteroid classes, their spectra, and their mineral associations is also given in Figure 1.8 for context.

While asteroids have been observed as far back as Herschel's 1801 observation of Ceres, asteroid classification as we now know it first began to take shape in the 1970s. First Chapman et al. (1971) found that asteroids measured in UVB filters had colour indices that clustered naturally into two groups. Zellner et al. (1974) later associated the groups with meteorite types. The first had absorption features in the NIR consistent with the silicates olivine and pyroxine and a red slope typical of iron, so they were associated with stony-iron meteorites and later labelled S-types. The second group were low-albedo objects with flat slopes, presumed carbonaceous (because of the abundance of carbon in the Solar System) and accordingly labelled C-types.

Over the next several years different research groups added more classes using a variety of measurements (multi-wavelength photometry, polarimetric light curves, radiometric albedo). The high albedo E-type ('enstatite') and mid-albedo M-type ('metallic') were found in 1976, followed by O ('ordinary chondrite'), T ('Trojan'), and R ('reddest') in the next few years. In the early 1980s D ('reddish dark' shortened to D for 'dark'), F ('flat'), 'A '(a separation of two R-type objects from the rest based on distinct albedo vs. infrared JHK relations (Veeder et al., 1983)), and P ('pseudo-M') were found. At this point the lack of consistency in the parameters used to determine classes was becoming a problem (Tholen, 1984).

A more consistent and mathematically rigorous classification system emerged from Tholen and co-authors in 1984 (Zellner et al., 1985). They used eight-colour photometric observations of 589 asteroids between roughly 0.34 - 1.0 μ m and performed principal component analysis (PCA) to reduce dimensions from eight to two (see 2.3 for a full description of this method), then clustered the data using a hierarchical (minimum tree) method to derive spectral classes. Finally, albedo was used to remove degeneracies;

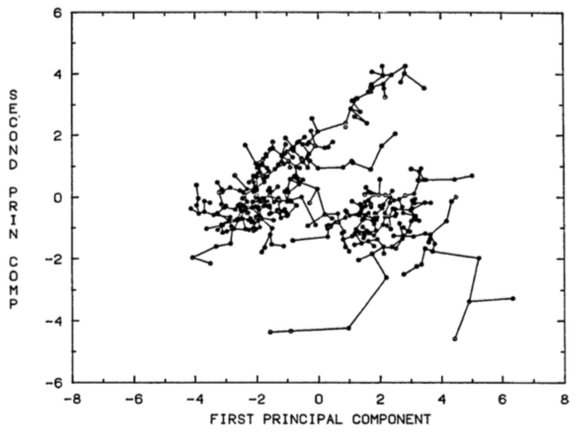


FIGURE 1.2: Connected graph of 405 asteroids from (Tholen, 1984) in principal component space. The graph is 'cut' at appropriate points to establish taxonomic types.

for example, E, M, and P (collectively the X type) are separated only by albedo. This resulted in classes A, B, C, D, E, F, G, M, P, Q, R, S, T, X [2]. Continuing in the vein of small populations, three of these classes were first created to account for individual asteroids: Q (Apollo), V (Vesta), and R (Dembrowska).

Central in Tholen's system is the ability of principal component analysis to capture 95% of the variance of the data within two principal components. This meant that he could plot the data in two dimensions and then construct a graph of relationships between points in order to separate clusters of similar objects (see Figure 1.2. Tholen's classes were eventually used as a basis for the Bus & Binzel (2002b) system based on SMASS optical spectra.

The Bus-Binzel taxonomic system forms the basis for the Bus-DeMeo taxonomy as well as the means by which Carvano et al. (2010) constructed a colour-based classification of asteroids in the Sloan Digital Sky Survey (sdss), so it underpins most of my work here. Example spectra for each taxonomic class are shown in Figure 1.3. The taxonomy is based on a sample around twice the size of Tholen's with wavelength range 0.435 - 0.925 μm, but it does not explicitly include albedo. The data are normalized at 0.55 μm and then fitted to a 'spline', which is a piecewise function that can fit different orders of polynomial to different regions of a curve (Subbotin, 2022), to produce datapoints at intervals of 0.05 μ. Slope is removed between 0.4 - 0.7 μm and PCA performed on the residuals, then boundaries are established on plots between slope and the first principal component.

Most classes seen before are here: A, B, C, D, K, O, Q, R, S, T, V, X. Classes F and G are not found because they depend on albedo, and the L class is introduced

[2]Here, X is used for E/M/P in the absence of albedo

as an intermediate between S and K. Additionally, a large number of subclasses are added, especially in the S-complex, for a total of 26. The handling of X-class is worth noting. In the absence of albedo, not only are E/M/P folded together as X, but also there is difficulty in differentiating X and C. The Bus system handles this by identifying spectral features in existing X-types (where albedo is known) and incorporating those features in identifying new X-types and at the same time articulating sub-classes of X based on detailed features only visible with the advent of CCDs. However, to achieve separation of X-types, PCA had to be performed a second time on X-types alone and in the shortest wavelength regime. This implies a fine distinction between these sub-classes; it is tempting to argue that some force-fitting has gone on.

Figure 1.4 illustrates the clumping of asteroids into two camps in PC space, denoted by ovals for C- and S-complexes, with a line drawn between C and X. Several capital letter sub-classes live within S-complex, and those outside it have the appearance of outliers rather than groups. Notably, V-class (associated with Vesta) has a very large scatter. It is already difficult to answer the question ‘what is a class?’ when asteroids believed to have the same parent body do not plot in a clump when compared to the rest of the population. We now see the other side of the problem faced by Tholen: he pointed out that naming classes based on different types of measurements led to potential inconsistencies. But when we restrict parameters to optical reflectance, dividing the asteroids into meaningful groups becomes much harder. The vagueness of many classes implicit in figure 1.4 is an issue that will become a theme as this thesis develops.

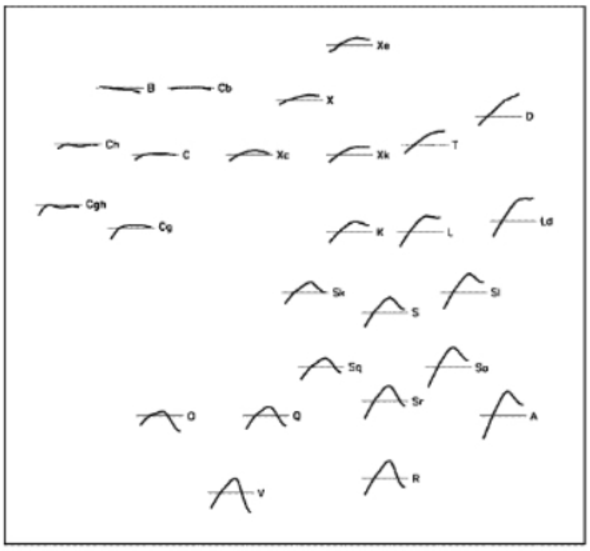


FIGURE 1.3: From (Bus & Binzel, 2002b), the taxonomy resulting from the SMASS survey showing an average spectrum from each Bus-Binzel class on a common scaling. The wavelength range is 0.435 - 0.925 μm and the horizontal line indicates normalised reflectance of 1.0.

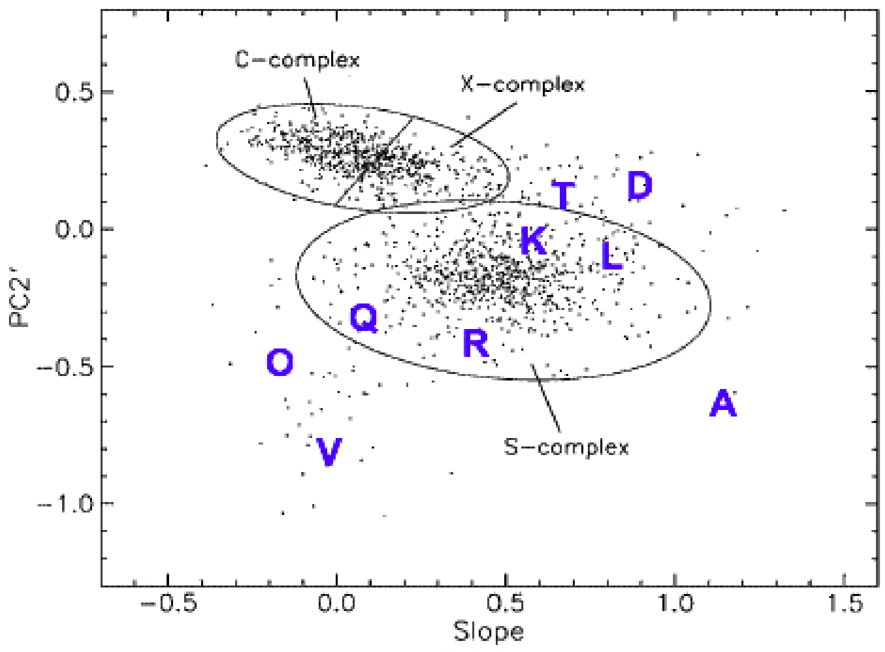


FIGURE 1.4: From Bus's paper Bus & Binzel (2002b), the main divisions of the Bus-Binzel classification are shown. The second principal component is plotted against slope (which is a proxy for first principal component). The rather arbitrary boundary imposed between C and X is based on features of known X-types (from albedo). Several other classes are determined from this plot (my notations in purple), whilst subclass B (in C-complex) is determined by the second principal component. The V- and A-type objects can be seen to have a large scatter. Viewed in terms of their numerical variance, V-types appear to have less in common with one another than, for example, any given X with any given K. For that matter, most 'capital letter' classes are subclasses of S-complex because they live inside its boundaries.

Successor to Bus-Binzel is the Bus-DeMeo (DeMeo et al., 2009) system, which extends the taxonomy into the NIR. The system is built around 321 objects with visible and NIR spectra that were already classified by Bus-Binzel, with most of the visible spectra coming from SMASS and the NIR coming from the SpeX medium-resolution 0.8–5.5 μm spectrograph instrument on the 3.2 m Nasa Infrared Telescope Facility (IRTF) on Mauna Kea. This taxonomy is constructed in the same manner as Bus-Binzel, with normalisation at 0.55 μm, splining at intervals of 0.05 μm, and removal of optical regime slope prior to PCA. In these data, a 'Grand Divide' between spectra is found when plotting PC2 against PC1. The Grand Divide corresponds to whether or not spectra have an absorption feature at 2 μm (see figure 1.5). (Because of this NIR feature, the Bus-DeMeo system has been adapted for NIR-only classification.) The rest of the divisions are artificial and comply with the Bus-Binzel system in that the authors add only 50 new spectra to the ones whose classes they already knew. They remove Sk, Sl, and Ld but add Sv.

The availability of a large asteroid dataset from the SDSS Moving Object Catalogue (MOC) led people to find ways to apply spectral classes to broadband photometric data. The SDSS data comprise five photometric bands (u, g, r, i, z) around effective wavelengths λ_{eff} as shown in Figure 3.6. In this thesis I work with data published by Carvano et al. (2010), DeMeo & Carry (2013), and Sergeyev, Alexey V. & Carry, Benoit (2021).

Carvano et al. (2010) use both the Tholen system and the Bus-Binzel system to arrive at nine photometric classes $(A_p, C_p, D_p, L_p, O_p, Q_p, S_p, V_p, X_p)$[3] They omit classes which their data is unsuitable to discern and merge L, Ld, and K into a single L_p class.

The Carvano system is probabilistic and accounts for overlaps in waveband ranges for each class. It avoids various problems associated with choice of normalisation wavelength for reflectance R by working in terms of the log reflectance colour defined as:

$$C_{\lambda_j} = -2.5(\log_{10} R_{\lambda_j} - \log_{10} R_{\lambda_{ref}})$$

A colour gradient is the important parameter here. Where C_j is the colour difference in band j relative to the g reference band and λ is the band centre, the colour gradient in band j is defined as:

$$\gamma_j = -0.4\frac{C_{\lambda_{j+1}} - C_{\lambda_j}}{\lambda_{j+1} - \lambda_j}$$

First the authors parametrized each observation in each filter by its colour gradient. Then each spectral class was assigned limits for the four colour gradients in the following

[3]The subscript indicates a photometric (as opposed to spectroscopic) derivation for the class, but I generally omit the distinction in this thesis as it should be clear from context when I am talking about SDSS classes.

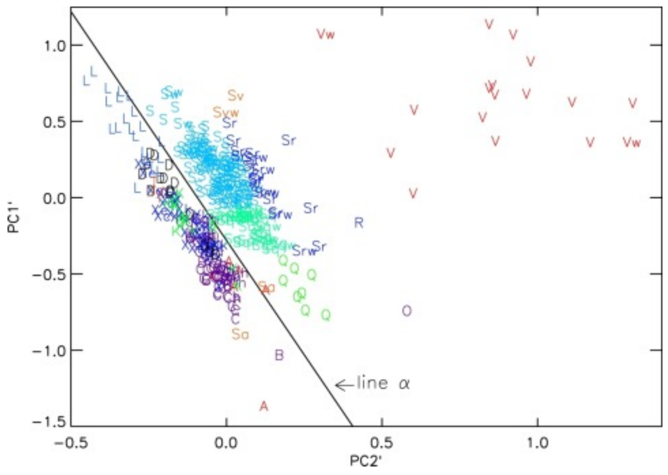


Figure 1.5: From (DeMeo et al., 2009), the 'Grand Divide' in asteroids plotted in principal component space. The split in data can be traced back to the 2 μm absorption band. Asteroids to the left of the divide do not have this feature; to the right, they do.

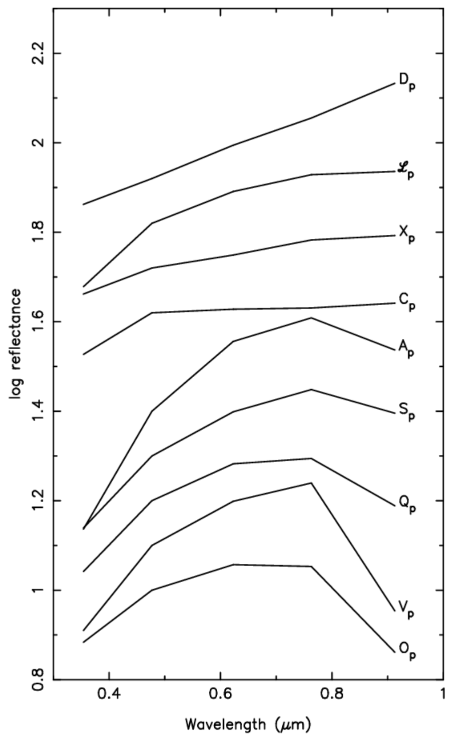


FIGURE 1.6: Median reflectance spectra for Carvano SDSS-MOC classification system. The subscript 'p' refers to photometric class to distinguish it from previous (spectral) taxonomies. L_p represents a fusion of L, Ld, and K classes unique to this system. Figure from Carvano et al. (2010).

manner. Previously-classified objects from SMASS and Small Solar System Objects Spectroscopic Survey (S3OS2) (Bus & Binzel, 2002b; Lazzaro et al., 2004) were used for most classes. For O and Q the authors used meteorite spectra of ordinary chondrites that matched class templates from Bus; for A they generated synthetic spectra based using the few known observations available. Spectra were convolved with SDSS filters and for each waveband they set a normal distribution around a mean γ with standard deviation defined by the measurement uncertainty. For any given observation, they measured its probability density according to the class distribution, one band at a time. The probabilities of all four bands were multiplied to give an overall result for each class. If a given class had a result greater than 60% of the sum of results for all classes, then that class was assigned to the object together with its final probability. If no class made the cut, then all classes having more than 30% of the sum were assigned (in no particular order) to indicate multiple classes.

This method of articulating class probability is advantageous for data analysis because subsets of the Carvano data can be used according to how strongly the classifier connects the object with the assigned class. The median reflectances for each Carvano class are shown in Figure 1.6.

DeMeo & Carry (2013) adapt the Bus-DeMeo system for SDSS and use the classifier to produce on a larger SDSS asteroid catalogue than Carvano (see 3.4.2 for details). The authors avoid using the noisy u band. They use two parameters: slope over $g-r-i$ bands and the $z-i$ colour (a proxy for band depth at 1 μ m). They define their class boundaries using class centres and standard deviations from the Bus DeMeo taxonomy for spectra, which runs into the NIR and allows them to confidently use z-band SDSS. They convolve a set of reference spectra with SDSS filters and convert to SDSS colours. No information is offered as to how they convolve with SDSS g-band spectra that almost invariably run no shorter than 0.435 μm in wavelength. Then they calculate the distance between each observation and each class centre, and fix class boundaries accordingly. They do some manual 'tweaking' based on their knowledge of individual objects. Example spectra for each class are shown in Figure 1.7, and the boundaries for classes in parameter space are shown in Figure 1.8.

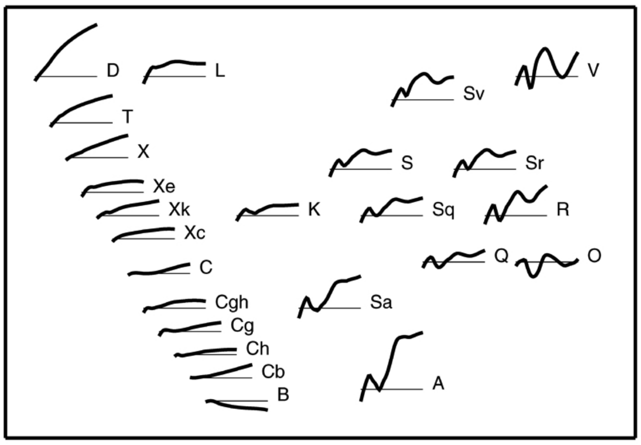


FIGURE 1.7: Representation of Bus-DeMeo classification system depicting average spectra for each class in the range 0.45 - 2.45 μm, with normalised reflectance of 1.0 indicated by a horizontal line for each spectrum. Classes are arranged roughly according to position in principal-component space (DeMeo et al., 2009). When comparing to the analogous Figure 1.3 from the Bus-Binzel system, classes A, S, V, and Q become easier to distinguish from one another given their different features to the right of the spectra.

The above describes the situation when I began working on asteroids. Given that quite a few classes have been assigned based on the unusual nature of a single object, one might wonder whether this is reasonable or whether a more statistical approach should be taken. Indeed, earlier this year a clustering study of mixed visible and NIR spectra together with albedo was published by Mahlke, M. et al. (2022) in which asteroids were clustered and reclassified according to a modified taxonomy, so clearly the field is moving in this direction. In my own work, I do not intend to obviate the need for detailed domain knowledge that underpins any classification scheme. Whatever the unanswered questions about the history of the asteroids (and there are many, as will be explained in the rest of this chapter), we know that many of them must have been subject to

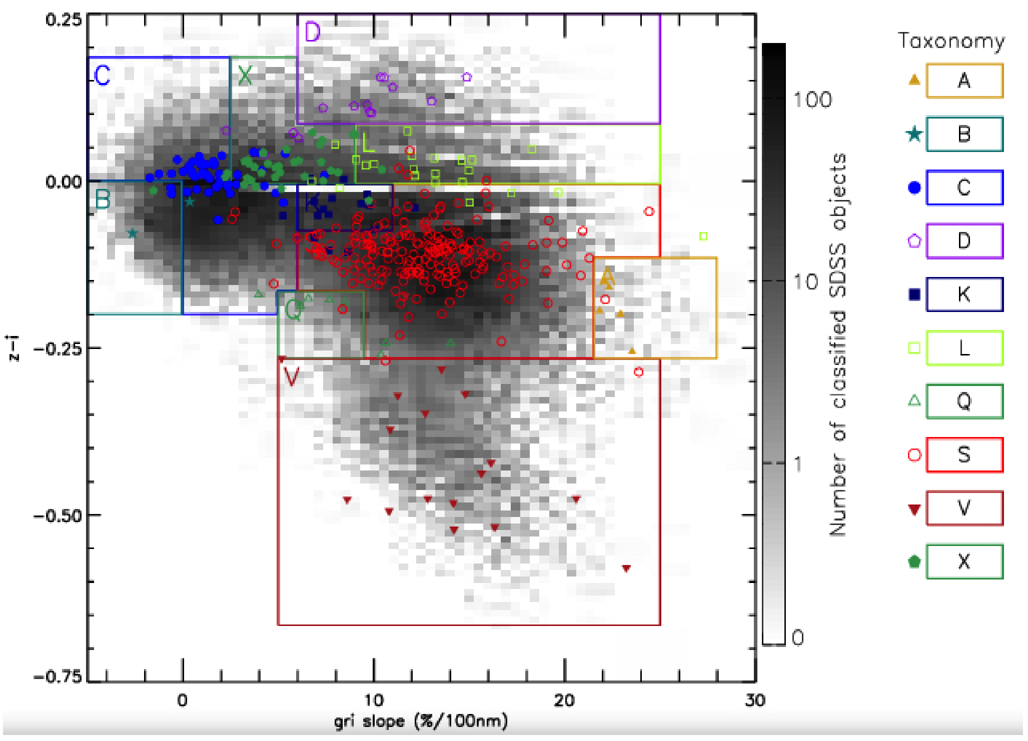


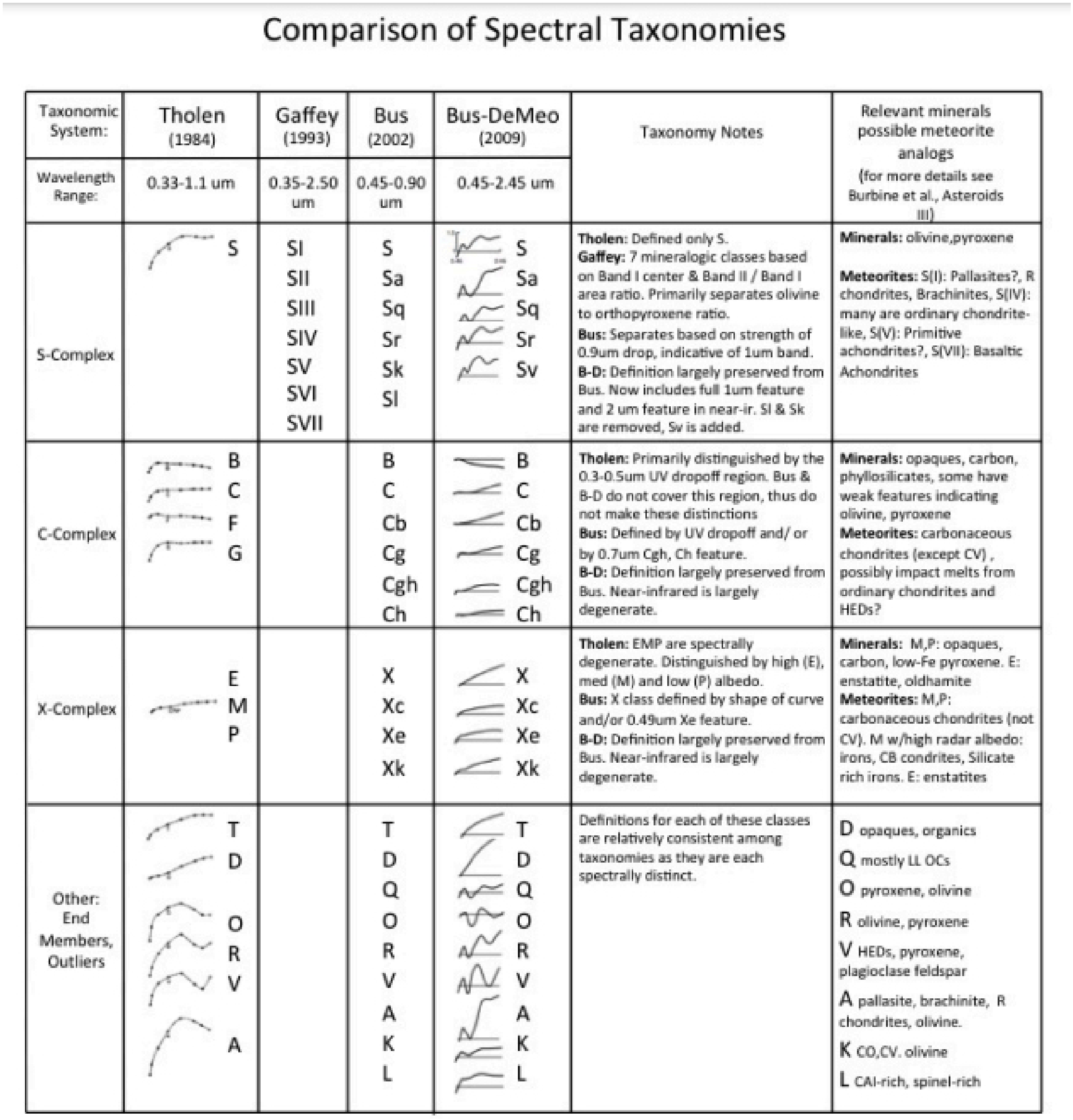
FIGURE 1.8: Boundaries for DeMeo & Carry (2013) SDSS classification system in their parameter-space (z-i colour vs. gri slope). Coloured points come from data having spectra and classified by DeMeo et al. (2009). Here they retain B and K classes in addition to the classes found by Carvano et al. (2010). Once again, the scattered and outlying position of V-types is notable.

.

collision, impact-melting, space-weathering, and differentiation, so it is sensible to think in terms of parent-and-child objects. For example, although I have remarked that there is a large scatter in the V-type classification in Bus-Binzel, the objects used to create the class are orbitally related to Vesta. I am not going to address orbital behaviour — the existence of asteroid families — but presumably it meaningfully relates to events that formed the asteroid belt. The larger context of Solar System formation is the topic of the next section.

1.4 Asteroids and Solar System formation

The Solar System's planets formed from a disk of gas and dust in which the dust initially coagulated to become mm- to cm-sized grains. To explain the evolution of grains to pebbles of around cm to meter radius, the streaming instability is invoked; here, simulations show that differential rates of movement between gas and grains lead to rapid growth of dense filaments in the disk (Youdin & Goodman, 2005) that eventually collapse to form planetismals. The terrestrial planets could have started with a 'pebble pile-up' in



Comparison of Spectral Taxonomies

Taxonomic System:	Tholen (1984)	Gaffey (1993)	Bus (2002)	Bus-DeMeo (2009)	Taxonomy Notes	Relevant minerals possible meteorite analogs (for more details see Burbine et al., Asteroids III)
Wavelength Range:	0.33-1.1 um	0.35-2.50 um	0.45-0.90 um	0.45-2.45 um		
S-Complex	S	SI SII SIII SIV SV SVI SVII	S Sa Sq Sr Sk Sl	S Sa Sq Sr Sv	**Tholen:** Defined only S. **Gaffey:** 7 mineralogic classes based on Band I center & Band II / Band I area ratio. Primarily separates olivine to orthopyroxene ratio. **Bus:** Separates based on strength of 0.9um drop, indicative of 1um band. **B-D:** Definition largely preserved from Bus. Now includes full 1um feature and 2 um feature in near-ir. Sl & Sk are removed, Sv is added.	**Minerals:** olivine,pyroxene **Meteorites:** S(I): Pallasites?, R chondrites, Brachinites, S(IV): many are ordinary chondrite-like, S(V): Primitive achondrites?, S(VII): Basaltic Achondrites
C-Complex	B C F G		B C Cb Cg Cgh Ch	B C Cb Cg Cgh Ch	**Tholen:** Primarily distinguished by the 0.3-0.5um UV dropoff region. Bus & B-D do not cover this region, thus do not make these distinctions **Bus:** Defined by UV dropoff and/ or by 0.7um Cgh, Ch feature. **B-D:** Definition largely preserved from Bus. Near-infrared is largely degenerate.	**Minerals:** opaques, carbon, phyllosilicates, some have weak features indicating olivine, pyroxene **Meteorites:** carbonaceous chondrites (except CV) , possibly impact melts from ordinary chondrites and HEDs?
X-Complex	E M P		X Xc Xe Xk	X Xc Xe Xk	**Tholen:** EMP are spectrally degenerate. Distinguished by high (E), med (M) and low (P) albedo. **Bus:** X class defined by shape of curve and/or 0.49um Xe feature. **B-D:** Definition largely preserved from Bus. Near-infrared is largely degenerate.	**Minerals:** M,P: opaques, carbon, low-Fe pyroxene. E: enstatite, oldhamite **Meteorites:** M,P: carbonaceous chondrites (not CV). M w/high radar albedo: irons, CB condrites, Silicate rich irons. E: enstatites
Other: End Members, Outliers	T D O R V A		T D Q O R V A K L	T D Q O R V A K L	Definitions for each of these classes are relatively consistent among taxonomies as they are each spectrally distinct.	D opaques, organics Q mostly LL OCs O pyroxene, olivine R olivine, pyroxene V HEDs, pyroxene, plagioclase feldspar A pallasite, brachinite, R chondrites, olivine. K CO,CV. olivine L CAI-rich, spinel-rich

FIGURE 1.9: Summary table of asteroids by type in different classification systems with possible meteorite links included. Figure by DeMeo et al. (2015).

an annulus starting at 0.3 – 1 AU and ending anywhere between 0.6 – 3 AU (Drakowska et al., 2016). When planetismals reach the size of around 100 kilometers, they continue to grow by colliding with one another (as in the inner disk) and/or by accreting from a flow of available 'pebbles' of centimetre to meter size drifting inward, slowed by the drag force of disk gas. (Johansen & Lambrechts, 2017).

Because disk temperature decreases with radius, the concept of a 'snow line' is key to separating the terrestrial planets from the gas and ice giants. Above the snow line, the presence of ice in pebbles allows for much 'stickier' accretion than that possible for the interior silicate materials (which tend to bounce on collision), leading to the collapse of larger planetisimals (Johansen & Lambrechts, 2017) and hence earlier commencement of pebble accretion. The gas giant Jupiter is thought to have formed more or less on the snow line (Morbidelli, 2015) and at a radius where simulations show that large initial planetisimals can form, enabling highly efficient pebble accretion (Coleman, 2021); some re-condensation of water vapour could also boost the pebble population at the snow line (Liu et al., 2019). In any case, the growth of Jupiter would have depleted the gas disk locally and implanted (carbonaceous chondrite) planetismals from its neighbourhood inward, towards the asteroid belt (Raymond & Izidoro, 2017).

Once a planetisimal has reached about ten Earth masses, pebble accretion is exhausted, but large planetismals can then attract a gaseous envelope. The timings and efficiencies of these processes lead to a plausible formation pathway for the giant planets within the timescale of the gas disk (Johansen & Lambrechts, 2017). As the gaseous component of the disk dissipated from the Sun outward, the dynamics of the orbiting bodies – especially the largest, Jupiter and Saturn – would have evolved in relation to one another and to the changing fraction of gas remaining. At some point, the reached a state that includes several inter-related structures apart from planets: the icy Kuiper Belt (beginning around 30 AU at Neptune's orbit); the scattered disk of highly inclined objects just outside the main Kuiper Belt; and the Oort Cloud, which begins at a few thousand AU and may stretch up to 100,000 AU from the Sun (Carroll & Ostlie, 2014). Of particular importance for this thesis is the region between the orbits of Mars and Jupiter: the asteroid Main Belt.

The Main Belt acts as a record of the thermal, dynamical, and collisional events that led to the configuration of bodies observed today. Its dynamical and mineralogical characteristics (the latter inferred from analysis of meteorite samples) have been used as constraints in modelling formation. No single model yet has adequately explained all the unusual qualities of the asteroid belt, but several have successfully addressed parts of the problem. These include the Wetherill model, the Grand Tack model, the Nice Model (all as described in Morbidelli 2015), the Jumping Jupiter addition to Nice (Nesvorný, 2018), and the newer 'partitioning' model of (Brasser & Mojzsis, 2020), which proposes ring formation in the Solar System.

So, what exactly is it about the Main Belt that needs explaining? According to Morbidelli (2015), there are multiple lines of evidence that it has been dynamically disturbed after its initial formation:

1. Its current mass is 4.5×10^{-4} Earth masses, but interpolation of the mass densities needed to form the terrestrial planets and Jupiter (Weidenschilling, 1977) predicts

one Earth mass for the asteroid belt (given a uniform disk). Something must have happened to 99% of the asteroid belt.

2. MBAs have orbital inclinations of $0 - 30$ deg (average 11 deg) and eccentricities of $0 - 30$ (average 0.145) in contrast to the average planetary inclination of 1.85 and eccentricity of 0.060, implying disruption to asteroid orbits.

3. Since temperature varied radially in the gas/dust disk, similar minerals are expected to form at similar radial distances from the Sun. Yet there is considerable mixing of mineralogical types (ordinary vs. carbonaceous chondrites (see 1.5)) at the same radial distance from the Sun despite similar accretion ages (see (Villeneuve et al., 2009) and (Morbidelli, 2015)).

All these factors act as conditions that must be met by models. Additionally, the fact that nearby planet Mars is lighter than its expected mass has multiple explanations that lack consensus. A small Mars may be the result of a radially less-efficient streaming instability in a planetisimal-forming annulus in the inner Solar System, as predicted by modelling of dust accretion (Drazkowska et al., 2016). Or, since the asteroid belt lies outward to Mars, it may be that Mars was also affected by whatever event(s) depleted the asteroid belt (Walsh et al., 2011; Brasser & Mojzsis, 2020).

It appears generally accepted that any depletion event must have happened early in the history of the Solar System. By simulating the behaviour of the current Main Belt as well as a version with 1000 times as much mass (as predicted for a primordial asteroid belt based on disk constraints), Bottke et al. (2012) found that the belt must have been disturbed early in its history to get where it is today, suggesting that most asteroids' orbits were changed so much that they were kicked out of the system. This would result in a residue of objects that act as tracers of the extrema of possible stable orbits in the Main Belt (Morbidelli, 2015).

The 1992 Wetherill model (Wetherill, 1992) says that at the time of gas dissipation there was a mix of planetisimals and planetary embryos inside the orbit of Jupiter. If their growth was 'oligarchic' (i.e., the big get bigger fastest (Mandell, 2014)), then once the first planetismals became large enough to overcome the damping of smaller bodies on their eccentrities, those inside 2 AU would collide and form planets. However, in the Main Belt the effects of Jupiter and other planetary embryos would cause the ejection of most planetisimals and explain the belt's mass depletion. Simulations by O'Brien et al. (2006) cited in Morbidelli (2015) indicate that Main Belt orbits could be achieved by these disruptions (which could also reproduce the mixing of chondrite types after their respective origins at 2 AU and 3 AU). However, this would only work if the Nice model (see below) is added on as well.

A more recent alternative to Wetherill is the Grand Tack model (Walsh et al., 2011). The Grand Tack addresses both the size of Mars and the question of why Jupiter orbits so far from the Sun when planets tend to migrate inward across the disk; but it has important implications for the asteroid belt, too. In hydrodynamical simulations, Jupiter is shown to migrate towards the Sun if alone in the disk, and away from the Sun if paired with Saturn. The model finds that Jupiter formed first around 3.5 AU before migrating inward to 1.5 AU, at which point changes in torque due to the influence of a migrating Saturn caused it to change 'tack' and reverse direction until the gas disk was fully dissipated. During this process the asteroid belt would have been disturbed twice. As Jupiter drifted in, it pushed dry/pre-Terrestrial/ordinary chondrite (see section 1.5) material inward, adding to the mass inside 1 AU and kicking about 10% of the belt out of the system at the same time–this would reduce the material available to form Mars. As Jupiter and Saturn drifted out, they would again cross the snow line, hitting the ordinary chondrites first, followed by the carbonaceous chondrites. Their passage would have excited asteroid orbits with respect to eccentricities and inclinations such that most of the Main Belt would be kicked out altogether. The remainder would have re-formed with the inner belt dominated by inner-Solar-System material and the outer belt dominated by material from beyond the snow line (consistent with the prevalence of S-type and C-type asteroids in the inner and outer Main Belt, respectively–see Figure 1.10). One reason for the attraction to this model seems to be the need to reconcile the Solar System's distribution of planets with the then-recent discoveries of hot Jupiters in other systems. The model relies on complicated changes in torque involving the relationship between the disk and Saturn and Jupiter as they migrate either separately or together. The Grand Tack model is a poor predictor for the formation of the other planets, but it does deplete the Main Belt rapidly, as required.

The 2005 Nice model can be treated as an addition to either of the previous two. It consists of three interlinked papers (Gomes et al., 2005; Tsiganis et al., 2005; Morbidelli et al., 2005) that propose a disruption around 400 million years after gas dissipation. In simulations of the migration of Jupiter and Saturn, the authors show that an exchange of angular momentum with Main Belt bodies was involved in the two giant planets passing in and out of mean orbital resonance with one another (Fernández & Ip, 1984) — an event that led to instability in the whole system. The Nice model accounts for the Late Heavy Bombardment, the capture of the Trojan asteroids by Jupiter, and a final orbital-excitation/mass-depletion of the asteroid belt (in which Earth's water may have been delivered from carbonaceous chondrites from the outer Solar System (Trigo-Rodriguez et al., 2019)).

A variation to the Nice model is the 'Jumping Jupiter' model as described by Nesvorný (2018). Here, a primordial close encounter between Jupiter and an ice giant is posited, in which the ice giant was ejected while causing Jupiter to 'jump' inward by some fraction of an AU. Simulations by (Fernández & Ip, 1984) appear to support the inward motion of Jupiter due to interactions with planetisimals between Jupiter and Saturn. The change in orbital resonances would have been enough to excite both the inclination of Jupiter (which does not match the other planets well) and the asteroid belt (Raymond & Nesvorný, 2022).

However, there is a relatively simple version of events in which it is not necessary to deplete the asteroid belt because none of the Main Belt mass is missing in the first place. The invocation of annuli in the protoplanetary disk (as in the pebble pile-up simulations by Drazkowska et al. (2016) means that the mass projections for the Main Belt region of a uniform disk no longer apply. A 2020 model by Brasser & Mojzsis (2020) applies insights gained from observations by the Atacama Large Millimeter/Submillimeter Array (ALMA) in which planets are seen forming in annuli of high density in the gas/dusk disk. Addressing the need for ordinary and carbonaceous chondrites to form from separate reservoirs, their simulations show that Jupiter's formation by accretion would have been too slow to achieve this in a uniform disk. They propose instead a 'partitioning' of the disk in which a pressure maximum at ~ 5 AU led to pebble pile-up in a dense ringed

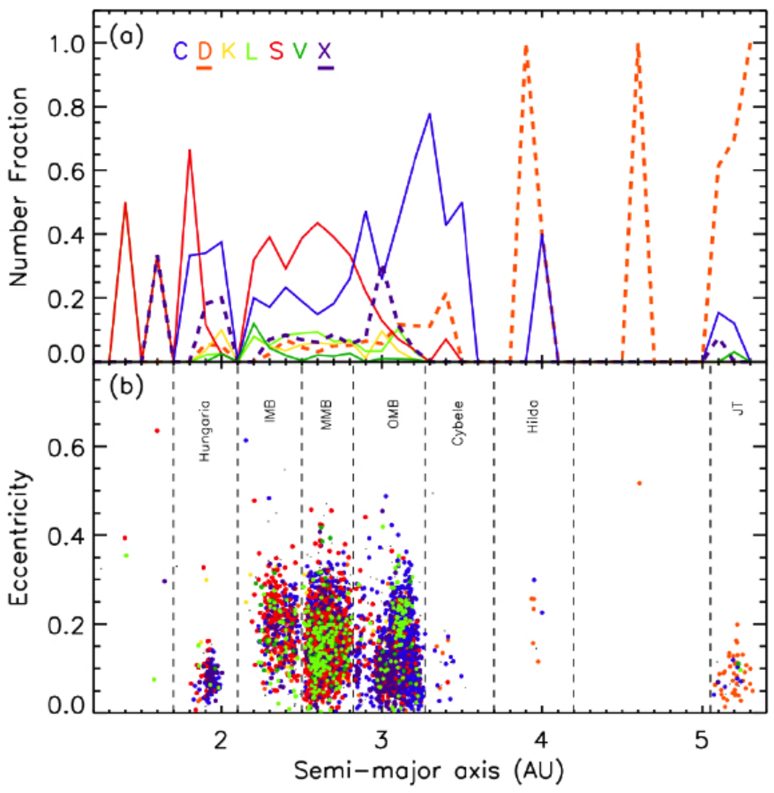


FIGURE 1.10: Distribution of asteroid types by orbital distance in AU according to types C, D, K, L, S, V, X by (Roh et al., 2022) for 4213 objects from the SDSS. The outer main belt is dominated by C-types, whilst S-complex make up a large fraction of the inner and middle main belt.

structure that both enabled the rapid formation of Jupiter and Saturn and stopped the infall of material towards the Sun (robbing Mars of mass). This assertion is supported with evidence from isotopes in materials from the inner and outer Solar System that shows a clearly bimodal distribution with carbonaceous chondrites (associated with giant planets) in one camp and the minerals of terrestrial planets and ordinary chondrites in the other (see Figure 1.11). The authors also find that a disk wind could explain the orbital excitation of the asteroid belt and that turbulent diffusion could have carried carbonaceous material inward without resorting to giant planet migration.

While there is some overlap in the different asteroid formation scenarios proposed so far, it is apparent that a single likely pathway to the existing orbital and chemical configurations has not yet been agreed. Moreover, the Yarkovsky effect is an important and ongoing mechanism in delivering asteroids from the Main Belt to the inner Solar System independently of formation model. Equally, the role of isotope information in the Brasser & Mojzsis (2020) new theory of formation must remind us of the importance of aspects of composition not found in reflectance spectra. The information about isotopes comes from meteorite science; in the next section I will sketch the basics of how meteorite analysis applies to history and asteroid type.

1.5 From meteorites to asteroids

Meteorite classification has its own nomenclature and conventions, central to which are the features of the specimen when viewed under an electron microscope—quite the opposite situation to observations of sunlight reflected hundreds of millions of kilometers away. When considering meteorites, it is important to realise that samples are biased in favour of NEAs, which are necessarily dynamically disparate to MBAs and may not be representative of the bulk of the asteroid population, especially when considering the role of past impacts in changing both orbital dynamics and physical characteristics in a single stroke. Then there are the (rare) cases of iron meteorites, where an unknown fraction of the meteor has burnt off in the atmosphere, meaning that any outer shell is lost. Yet drawing connections between meteorites and their Main Belt counterparts is the only available option, given that only a small amount of asteroid data can be collected in situ. Encouragingly, samples from Itokawa, Bennu and Ryugu offered surface grains associated with their types predicted from spectra (S, B, and C respectively), with Ryugu classification adjusted from C to subtype Cb after mineral samples were obtained (Nakamura et al., 2011; Lauretta et al., 2019; Vilas, 2008).

Asteroid taxonomy does not claim that membership in a class implies that a body has the same mineralogical composition as other members (DeMeo et al., 2015). Naturally

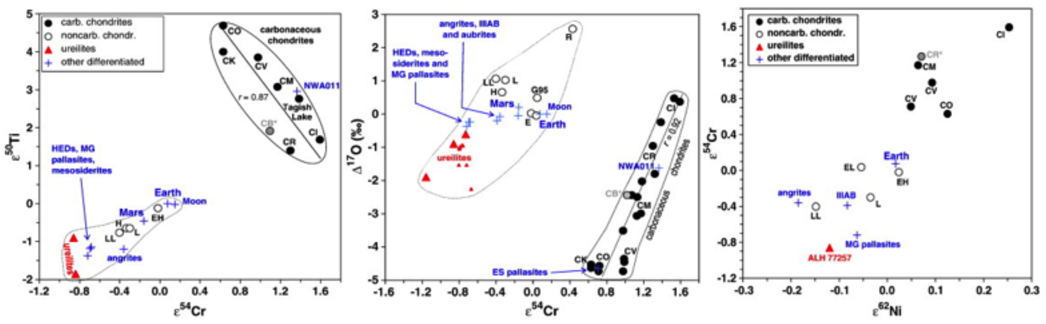


FIGURE 1.11: Evidence of bimodality in meteorite composition based on isotope ratios from Warren (2011), who argues that the primary taxonomic division lies between carbonaceous chondrites and all others.

it is tempting to map the one to the other, but we must be very careful about this. One reason for caution is that the picture within meteorite science itself is rather complex. On the one hand, samples have petrological (formation history) features that have been assembled into a taxonomy (see figure 1.12). Many efforts have been made to interpret this information in the context of Solar System formation physics. On the other hand, the isotope background of meteorites suggests a different taxonomy with different priorities and hierarchies (Warren, 2011) as shown in figure 1.13. A second reason for caution arises from specific cases where asteroid type and presumed meteorite type are obviously inconsistent. It is beyond the scope of this work to untangle the details required to state coherently exactly where the field stands today, but I will lay out the situation in broad strokes to give the reader a snapshot of the main issues that may be relevant to asteroid classification.

The most abundant meteorites are chondrites, including ordinary, carbonaceous, and enstatic chondrites. In general, ordinary chondrites (OCs) are silicates associated with S-type asteroids. Carbonaceous chondrites (CCs) are associated with both C-class (low

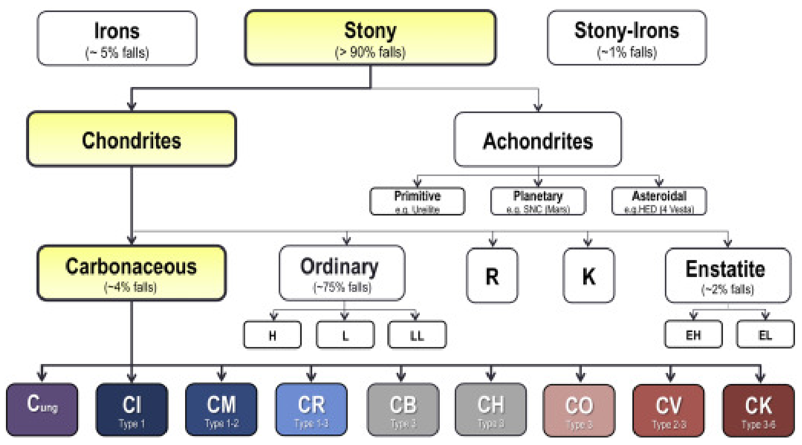


FIGURE 1.12: Schematic of classical meteorite classification system without consideration to isotope ratios, from (Glavin et al., 2018). Carbonaceous chondrites are highlighted.

albedo) and X-class (metallic) asteroids (DeMeo et al., 2015). Enstatic chondrites (ECs) are rare and have evolved more than OCs or CCs (Ridpath, 2016).

The oldest components of chondrites are tiny amounts (up to hundreds of parts per million) of presolar grains such as diamond, graphite, and silicon carbide, which became trapped in a fine silicate-rich dust matrix as it condensed and accreted in the Solar nebula. Matrix acts as a cement for chrondrules and other mineral inclusions (Zolensky et al., 2018). Isotopes in some presolar grains date them to as far back as 1 Gyr before the birth of the Sun. The apparent origins of presolar grains in asymptotic giant branch AGB stars, Type II supernovae shock ejecta, and winds from Wolf-Rayet stars offer information about the nature of the molecular clouds that collapsed to form the Sun (Jones, 2003).

The basic building blocks of chondrites are chondrules: small (usually ~ 1 mm) globules of melted dust grains, the first of which formed in the Solar nebula's first few million years before condensing. Chondrules' composition is dominated by silicates: olivine (a mixed crystal of Mg_2SiO_4 and Fe_2SiO_2 in varying proportions) and pyroxine (a crystalline structure found in basalt and containing oxygen in combination with various other metals and small ions) (Jones, 2003). Chondrules can be primordial or the complicated result of multiple generations with successive meltings. OC meteorites are always undifferentiated; i.e., there has been no melting in which heavier elements sink to the core of the parent body, meaning that they must have formed discretely in the early Solar System. The matrix of OCs is subject to some aqueous effects but otherwise in many cases is virtually unchanged since formation (Zolensky et al., 2018). Due to the absence/sparsity of organic molecules and water in the OCs' matrix, they are thought to have originated in the inner belt, where volatiles would have burnt off (and where most S-types still reside) (Morbidelli, 2015). Their chondrules have an oxygen isotope ratio that is distinct from ECs and CCs, pointing to a different origin to these other types (Jones, 2003) and leading to a proposed new taxonomy (Warren, 2011).

Carbonaceous chondrites are misleadingly named; they are not necessarily particularly high in carbon (Warren, 2011). CCs represent a small fraction of meteorites but have disproportionate importance in Solar System formation history due to their partial penetration of the asteroid belt from an origin above the snow line. They are also believed to have delivered organics and water to the Earth. CCs are easily distinguished from OCs by isotope ratios (see Figure 1.11), and under the microscope they show evidence of evolution after chondrule accretion (Zolensky et al., 2018). This diverse group of objects are sub-divided into eight petrological categories according to the degree of thermal and aqueous effects evident in their composition. Besides olivine, other silicates, and sulfides, the matrix of CCs contains up to 5% carbon in the form of graphite, carbonates, and organic compounds including amino acids, as well as water and evidence of its effects (Jones, 2003). The presence of volatile organic compounds implies formation above the snow line, although in cases where the object has been transformed by shock-heating of the parent body no volatiles are present. Some CCs contain presolar materials and calcium-rich inclusions (CAIs) that were early condensates of the cooling Solar nebula even prior to the first chondrule formation (Jones, 2003), all found in quantities well above the small traces that sometimes appear in ordinary chondrites (Zolensky et al., 2018).

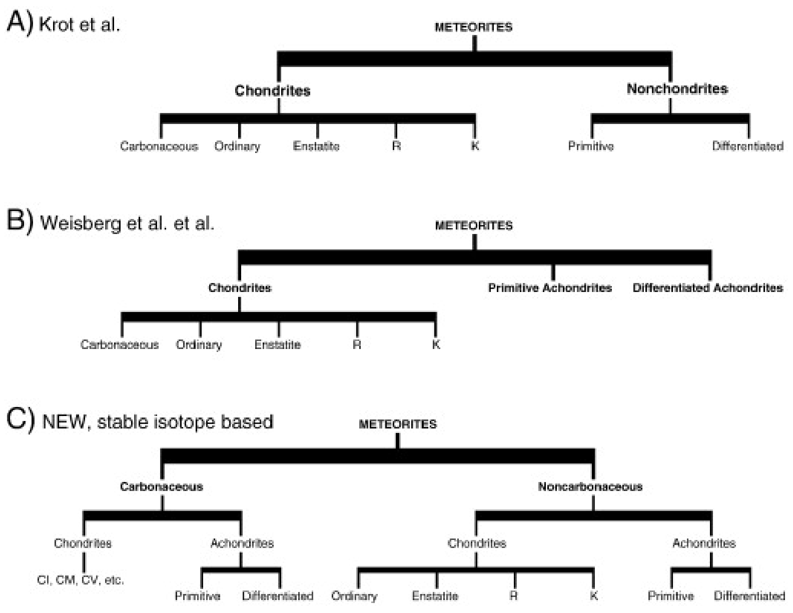


Figure 1.13: Proposed change in meteorite taxonomy from previous versions to one derived from isotope ratios (Warren, 2011). In the new version, differentiation (or not) and other aspects of petrologic type are subordinate to the sample's origin in the primordial disk. The 'carbonaceous vs. others' principle has been accepted but does not seem to be reflected in the nomenclature of the taxonomy itself (Krot et al., 2014).

Enstatite has chemical formula $Mg_2Si_2O_6$, but enstatite chondrite meteorites can include up to 35% iron-nickel content and their densities and grain sizes vary widely (Zolensky et al., 2018), with relatively little matrix. These rare objects are among those associated with X-type asteroids and may contain valuable Cr, Mn, and Ti (Hutson & Lewis, 1991). They are thought to have formed inside 1.4 AU (Keil, 2010) and their isotope profile is similar to Earth's, leading to the idea that enstatic material was involved in Earth's formation (e.g. Javoy et al. (2010); Piani et al. (2020); Lin (2022)).

Non-chondrites (or achondrites) are igneous and sometimes metamorphic meteorites characterised by formation in a reducing (oxygen-poor) environment and in which a degree of melting (likely due to radioactive decay) destroyed the chondrules. For example, enstatic achrondites are likely the result of melting in enstatic chondrites (DeMeo et al., 2015). Achondrites can be primitive or differentiated. Differentiation occurs when the meteorite parent body experiences enough heat and/or self-gravity to allow dense materials to sink to the core while silicates float to the surface. One group of basaltic achondrites are the howardites, eucrites, and diagenites, collectively called HED achondrites, likely originating in the crust of Vesta (making them V-types) (Jones, 2003).[4] A-type asteroids are believed to be examples of the mantles of differentiated achondrites for which the iron-nickel meteorites are supposed to be the cores (Reddy et al., 2005).

The relationship between meteorite and taxonomic types is neither simple nor clear-cut. It is acknowledged in asteroid science that types C, X and S each include a range of compositions (DeMeo et al., 2015); C and X spectra may be indicative of CCs, iron meteorites, enstatite, and some achondrites (Burbine, 2000). Among X-types, high-albedo examples (formerly type M) are thought to have iron-nickel composition; yet some examples have 3 μm hydration bands that would not be possible at the high temperatures needed to form such a differentiated body (Rivkin et al., 2000) cited in (Carvano et al., 2010). Differentiation has been posited as a contributing factor to the wide range of mineralogical features associated with S-types, notably the work of Gaffey et al. (1993) on partial differentiation within planetesimals. The authors grouped a sample of S-types into four subtypes according to mineralogical features (visible slope and relative band depths and ratios at 1 and 2 μm). Partial differentiation means that the asteroid may have fully melted but not fully crystallized into core/mantle/crust layers; or, given different melting points of iron, silicates, and other components, it may have only partially melted. Size of the body affects its differentiation through gravity, implying that different-sized segments of the same parent body could have different spectra even if they were severed before or during differentiation.

There is certainly evidence of mixing of minerals in meteorites. A simple example is pallasite, in which an iron matrix contains and connects a silicate crystal in a web-like structure (see figure 1.14). Despite the strikingly visible presence of iron (Buseck, 2022), pallasite is associated with S-types due to its silicate reflectance spectrum (DeMeo et al., 2015). When it comes to the iron meteorites, they are sub-divided into no less than 14 groups, of which one type is magmatic and associated with the cores of differentiated objects, whilst another is nonmagmatic, contains silicates, and is associated loosely with impact melts. More generally, when Carvano et al. (2010) applied their classification system to the RELAB spectroscopic database (Pieters & Hiroi, 2004) they found a wide distribution of meteorite classes for each asteroid class. The extremely low resolution of their reflectance data may well be to blame; taken in context of this brief review of the

[4]Spectrally, V-types are close to S-types and bear no resemblance to C-types; nevertheless, some C-type asteroids are thought to result from shock-melting during impacts between chondrites and HED achondrites (DeMeo et al., 2015)

literature, their findings suggest that it would be better to take meteorite/asteroid class correspondences with a grain of salt.

1.6 The changing appearance of asteroid spectra

Asteroid spectral slope and features, especially in the NIR, can change due to environmental conditions as well as observational phase angle. The latter is the angle between the observer, the Sun, and the moving asteroid. For MBAs, phase angle is typically no greater than 30–40 degrees, but NEA phase angle can change by as much as 100 degrees. A high phase angle can produce 'phase reddening'; i.e., spectral slope in the visible-NIR that increases with phase angle, attributed to surface roughness and scattering processes. Different surfaces are associated with different reddening behaviours, but these trends do not necessarily correspond to asteroid spectral class (Popescu M., 2019).

Environmental processes that alter surfaces include space weathering, collisional effects, and resurfacing. Modelling the effects of known processes and studying the effects of laser irradiation and ablation on meteorites in the laboratory has offered insight into competing processes that can both alter an asteroid's surface and refresh it. Space weathering can refer to micrometeorite bombardment and UV, X- and cosmic-ray irradiation, but the dominant process is the solar wind, which can increase spectral slope and sometimes darken the spectrum in the visible-NIR. However, asteroids can also be resurfaced in various ways: by gravitational encounters or YORP-effect spin changes that stir up dust, by collisions that expose fresh surfaces, by the influx of interplanetary medium particles, and by thermal changes at close perihelion, all of which tend to reduce spectral slope in opposition to space weathering (Graves et al., 2019; Brunetto et al.,

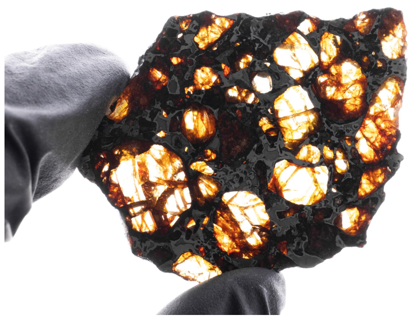


FIGURE 1.14: Example of stony-iron pallasite meteorite with an iron matrix surrounding silicate material. Pallasite is associated with S-types despite its iron content (Aerolite, 2022).

2015). The end result is an asteroid population whose spectral class may result as much from a range of events in its history as from some set of intrinsic physical characteristics.

1.7 The starting point of this work

In this chapter I have sketched the history of asteroid classification. I also explained the important role that asteroid types play in unravelling the history of the Solar System, including how difficult it can be to match reflectance-spectrum type to mineralogical type despite the rich and complex knowledge base surrounding the latter. I have introduced some models that govern how asteroid sizes (and albedo) are estimated and discussed some ways in which the physical properties of asteroid surfaces can change. These complexities indicate that asteroid classification is not as absolute as it might appear on first glance.

The contemporary Bus-DeMeo system builds directly on Tholen's work in the early 1970s, which used a combination of optical reflectance and albedo to cluster a few hundred asteroids into classes. Notably, virtually every time new data are collected there has been an increase in the number of subclasses identified, begging the question: are the original class labels still relevant? Similarly, the SDSS taxonomies not only have the same small-sample foundation but also carry the additional burden of low resolution, making class separation much more challenging. In the following chapters, I shall explore the robustness of the asteroid classes with machine-learning techniques.

1.8 How this thesis is structured

In this work I evaluate asteroid classification with a minimum of assumptions, using machine learning (ML) as a check on the internal consistency of human-made asteroid classes. In Chapter 2 I introduce the concepts behind ML in general and the methods I use. Chapter 3 introduces the various datasets and their preparation. I then proceed in Chapter 4 to establish a baseline for how well an ML algorithm can learn asteroid classes from spectra, with and without PCA. In Chapter 5 I compare this result to the result of classifying low-resolution broadband data, while in Chapter 6 I compare various methods of increasing the sample size to balance the numbers of objects in each class. Once classes are balanced I apply unsupervised learning, which is a way to group asteroid data without teaching the algorithm anything about the classes, in Chapter 7. I summarise my conclusions and briefly remark on possible future work in Chapter 8.

Chapter 2

Machine Learning Methods

2.1 Introduction

In this chapter I discuss the machine learning methods used in this thesis. Section 2.2 introduces some of the fundamental concepts underpinning machine learning. I list the metrics and visualisations used to assess the success of any given algorithm as well as the methods for classification and clustering that I have selected. Subsequent sections address each of the methods used in the thesis, beginning with dimensionality reduction using principal component analysis in Section 2.3 and the support vector machine classifier in Section 2.4, both used in Chapters 4 and 5 In Section 2.6 I then describe the augmentation algorithms from Chapter 6, including a breakdown of how the variational autoencoder works. Finally, in Section 2.7 I detail the methods used in Chapter 7 to determine the optimal number of clusters as well as algorithm stability, before describing each of the clustering algorithms in turn.

2.2 Fundamental concepts in machine learning

All machine learning involves training and validating a mathematical model using data. Models vary widely in scope, construction, and purpose, but common to the process is the division of data into a training set (to train the algorithm), a validation set (to tune the hyperparameters), and a test set (to evaluate the performance of the model on unseen data). A typical division of the data would be 70-20-10. Within the training set there may be nested a validation set, which works to set the model's hyperparameters. The training set alone determines the configuration of the final model, and it is always kept strictly separate from the test set or the entire process is invalid. Once a model is built, its trained weights can be saved for re-use with new data.

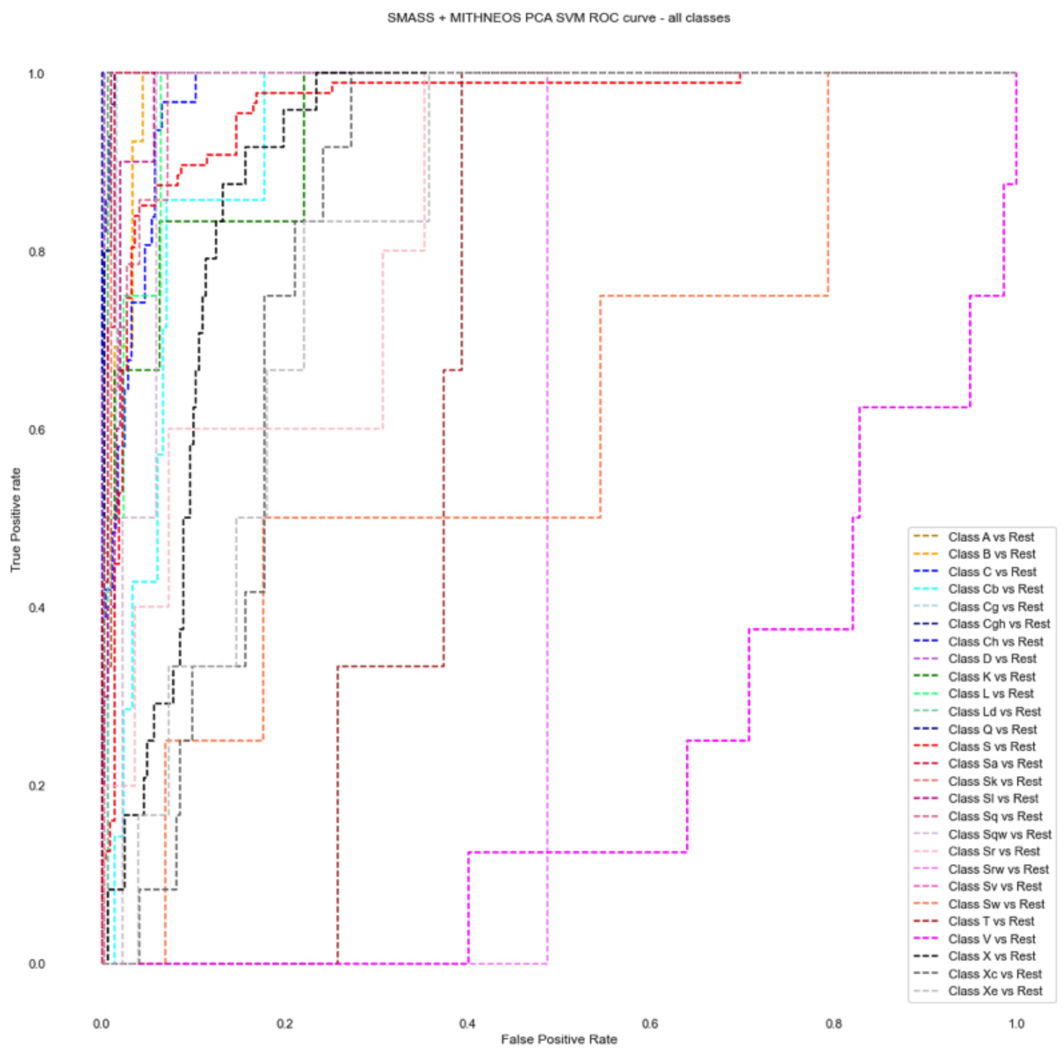


Figure 2.1: Example ROC-AUC plot for a classifier trained on all classes of spectra in this project, representing the true positive rate vs. false positive rate. The closer the lines are to the upper left corner, the better the model. This type of plot works well for binary classifiers but is very hard to read as multiple classes are introduced, so I do not include these plots.

I use three basic types of machine learning: supervised, self-supervised, and unsupervised. (Other types such as semi-supervised and reinforcement learning exist, but they are not relevant to the context here.) Supervised learning tries to match its outcomes to a target whereas unsupervised learning is a process of assigning labels from characteristics inherent in the data. For an asteroid classifier, the features are floating point decimals representing normalised reflectance flux values either by ascending wavelength or by broadband mean flux value, and the labels are asteroid classes. The model ‘learns’ to predict the best label according to features it has ‘seen’ during training.

After testing, a set of performance metrics is produced, which represent the performance of the model. Because an accuracy score (simply the fraction of correct classifications in the dataset) alone can mislead by obscuring important details, it is good practice to use additional metrix such as Sensitivity, Recall, Specificity, F1-score/mean-squared-error (MSE), and a precision-recall curve (or Receiver Operating Characteristic – Area Under the Curve (ROC-AUC) for binary classification problems (see figure 2.1)). I have used confusion matrices instead of ROC-AUC plots in this thesis because when there are multiple possible classes, the confusion matrix allows us to see not only correct and incorrect classifications, but the class where the model 'thinks' a given object belongs. The horizontal axis of the confusion matrix represents the predicted class, while the vertical axis represents the 'true' or 'actual' (i.e., previously recorded class); correct classifications then sit on the diagonal (see Figure 2.2).

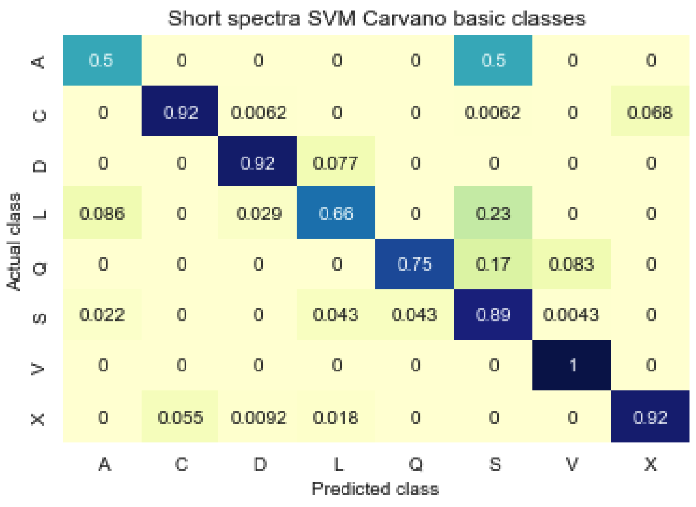


FIGURE 2.2: Example confusion matrix showing predicted classes on the horizontal axis and recorded classes on the vertical; anything on the diagonal is correctly classified. Instead of displaying actual numbers of objects in cells, this example shows fraction of objects by cell. The darker the cell, the higher the fraction.

Classification is an example of supervised learning. There are many models to choose from. I initially classified the asteroid datasets using a suite of four classifiers (random forest, support vector machine (SVM), multi-layer perceptron (MLP), and extreme gradient boost (XGB)). All the models performed well, but for simplicity I report only on the SVM because its results were the most consistently strong. The other methods are briefly outlined in Section 2.5.

Self-supervised learning means that the model is trying to reach a target, but because the data are unlabelled, the data themselves are also the target. In this work I use a variational autoencoder (VAE) to generate new samples of classes of reflectance spectra where there is a lack of data. A VAE is a deep learning method where a neural network learns a compressed representation of the data by encoding it in a 'latent space' and then resampling a new representation from the latent space to generate a similar, but non-identical, new sample.

Unsupervised learning is characterised by an absence of labels. Typically, it is used for grouping data into so-called 'clusters' of objects that are near to one another in some desired feature space according to a chosen metric. The method is exploratory; as part of problem-solving, often clustering is done first in order to loosely understand how many groups are in the data and how they may relate to one another. However, in this work I use unsupervised learning to explore the robustness of human-assigned classes.

Whether the model is supervised or not, for best performance some degree of hyperparameter tuning needs to be performed. A hyperparameter is a parameter that is controlled by the user and fixed before training begins; it affects how the model learns from the data. I have chosen to tune hyperparameters using a combination of trial-and-error and cross-validation.

In cross-validation, each time the model is run a different portion or 'fold' (usually around 20% the training set) is set aside as a validation set, and results are compared across the different combinations to obtain a more robust result than would be possible with a single validation set. Using a pipeline construction to permit scaling, the model is run iteratively on the training set, executing one stage of the pipeline at a time and tracking the score. In this case, the support vector machine is run and scored multiple times using the validation set as a mock test set. In cases where this process would be too computationally expensive, it is also possible to perform a randomised search in which combinations of parameters and folds are run at random. Here, models were tested using an exhaustive search for a range of parameter values that have been narrowed down manually to begin with and then refined through cross-validation. I note that folds were 'stratified'; that is, selected to reflect the class balance of the original data.

When it comes to unsupervised learning the most important hyperparameter is usually the number of clusters. There are a few techniques to help make a best guess as to how many clusters are inherent in a dataset; unfortunately, they are not always in agreement with one another! The methods used here are the elbow (with distortion and Calinski-Harabsz Index), the Davies-Bouldin index, and the silhouette method. Each of these will be described later in this section.

To ensure reproducibility, classification models are equipped with a 'random state' hyperparameter that allows one to initialize the algorithm using the same values with every run. Random state is also important in the division of training and test sets. In this work I have ensured that wherever appropriate, the random state has been fixed across all iterations and permutations. In some cases, I have repeated entire processes using multiple random states to ensure that results are generally sound.

Finally, a word about features and feature-space. The data are said to have 'features' or 'dimensions', which are simply columns of a matrix in which each row is a data sample. Features can be categorical variables having discrete labels (e.g., animal species) or continuous numerical variables that represent measurements of some quantity (e.g., mass). All features in this project are continuous variables expressed as floating-point decimals that measure sunlight reflected from the surface of asteroids. It is conventional to refer to features as 'dimensions'; appropriately, they are often used as plot axes when analysing data. But machine learning is typically required to work with far more dimensions than one can visualise, and often the user doesn't know which dimensions are the most important for a given task. Also, as the number of dimensions D increases, the volume of feature-space increases to the power of D. Unless the number of samples is astronomically large, the data will be sparsely distributed across this very large feature-space. Outside of deep-learning models, classifiers cannot do very much with a feature-space that is mostly empty. Another complication is the fact that many classifiers (including the support vector machine used here) measure similarity using a distance metric. The higher the dimension, the greater the pairwise distance between any two points, so that the difference between similar and dissimilar pairs is no longer significant. These problems, together with other statistical effects as described in Altman & Krzywinski (2018), are collectively called the 'curse of dimensionality' and have given rise to the development of dimensionality-reduction techniques. The next section describes the most prominent of these: principal component analysis.

2.3 Principal component analysis

Reflectance spectra consist of many dimensions (in this case, wavelengths) that tend to be correlated. Some of the information fed to a classifier is redundant, making for inefficient learning, whereas the large number of dimensions invokes the curse of dimensionality. Principal component analysis (PCA) is a linear transformation of the data into a feature space with a small number of new axes in which the covariance between features has been removed while the variance between datapoints has been maximized. Figure 2.4 shows a simple plot of features from the Iris dataset before and after PCA, illustrating the ability of PCA to tease apart different classes of data in which classes overlap in feature-space. This tool is available is available in Pedregosa et al. (2011b). [1]

The following breakdown is based on a description from the Open University text *Mathematical Methods and Models* (Open University, 2008) and lecture notes by Siekmann

[1]https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

(2021). If we treat each datapoint as a vector $\mathbf{X}$ of dimensions $X_1, X_2, ...X_p$, then the data have a mean $\mathbb{E}[\mathbf{X}]$ and a variance $Var(\mathbf{X}) = \mathbb{E}[(\mathbf{X} - \mathbb{E}[\mathbf{X}])^2]$. The covariance expresses how much each of the independent variables in $\mathbf{X}$ varies with respect to each of the others and takes the form of a symmetric, positive, semi-definite matrix: $\mathbf{\Sigma} = \mathbb{E}[(\mathbf{X} - E[(\mathbf{X}])(\mathbf{X} - E[(\mathbf{X}])^T]$

PCA is performed as follows:

1. Obtain the covariance matrix for the data.
2. Calculate the eigenvalues of the covariance matrix via solving the equation $\det(\mathbf{\Sigma} - \lambda\mathbf{I}) = 0$ where $\mathbf{I}$ is the identity matrix and the eigenvalues are $\lambda_1, ...\lambda_P$.
3. Place the eigenvalues in descending order by size.
4. Solve the set of eigenvector equations $(\mathbf{\Sigma} - \lambda_i)\mathbf{I})\mathbf{a_i} = \mathbf{0}$ where $\mathbf{a}$ are the p eigenvectors $a_1...a_p$
5. Subtract the mean vectors from the data to normalise around zero
6. Take the dot product of the normalised data with the eigenvectors to obtain the principal components:

$$\mathbf{U} = \begin{bmatrix} a_1 \\ \vdots \\ a_P \end{bmatrix} \cdot (\mathbf{X} - \mu)$$

The matrix $\mathbf{U}$ is ordered from largest to smallest eigenvalue and therefore from largest to smallest variance along each axis in principal component space. It is common for only two or three PCs to be used. (In this work I use three.)

Because PCA is a linear transformation, the eigenvectors act as weights applied to each dimension of the data individually in order to transform the axes. These weights, or 'loadings' can be examined to find out which dimensions of the data are contributing most to the variance. An example of eigenvectors from the SMASS2 survey is shown in the paper by Bus & Binzel (2002b) to elucidate which spectral features are associated with the greatest variance of their data. Their first principal component is roughly equivalent to slope, whilst the second shows up in an absorption band around 1 μm and the third PC impacts both in short wavelengths and as a feature around 0.7 μm. It is upon these variations in behaviour that the Bus-Binzel classification system is based.

In this work, PCA was used to check differences between datasets with respect to their variances, and as a pre-processing step before classification. Prior to performing PCA I scaled the data as described in the next subsection.

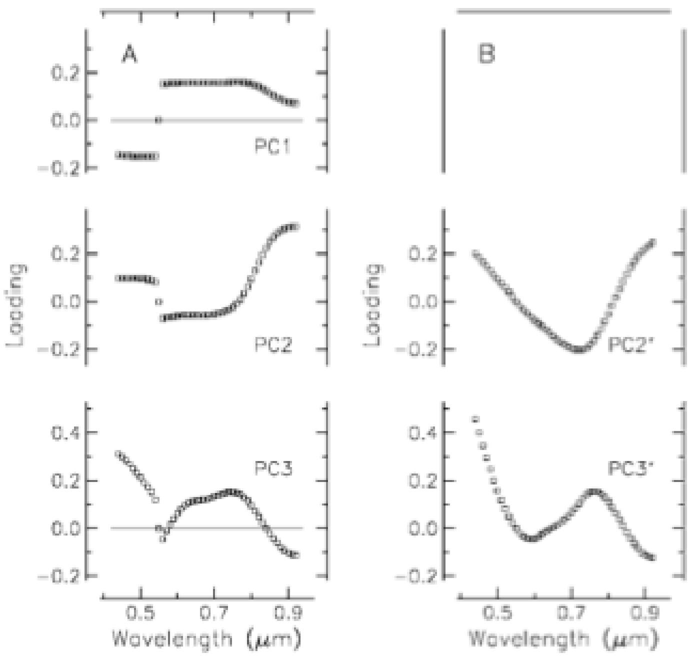


FIGURE 2.3: Eigenvalues by wavelength for SMASS classification system from Bus & Binzel (2002a).

.

2.3.1 Scaling for PCA

Usually in ML the reason for scaling is to ensure that each feature of the data is considered on equal footing with the others; for example, if one feature runs on a scale of 0–1000 and another runs from 0.001–0.010 then the variance in the second feature will be washed out by the variance in the first feature. In PCA it is therefore conventional to scale data before performing the analysis. For the spectra in this project it turns out that the original features are already on a scale that is close enough to one another, and scaling has a negligible effect on classifier accuracy. Nevertheless, at points in my workflow specified in the relevant text I preprocess data with RobustScaler, which removes the median and scales the data according to the range between the 1st quartile and 3rd quartile so as to reduce the impact of outliers (Pedregosa et al., 2011b). When running a support vector machine (see section 2.4) with cross-validation to tune hyperparameters, I scale in a pipeline that ensures the independent scaling of training and test sets in each fold of the cross-validation; otherwise, the classifier could 'see' the scaling from the entire training set during cross-validation, which would bias the result.

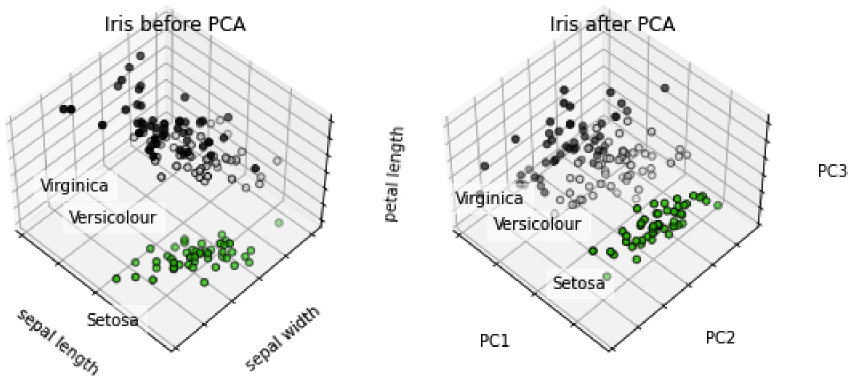


FIGURE 2.4: Example of the four-dimensional Iris dataset showing the first three dimensions on the left, and all the data after being recast into principal component space on the right. The PCA axis shift allows the data to be better separated, and reducing the dimensions from four to three facilitates visualisation. Example from scikitlearn documentation (Varoquaux, 2022).

2.4 Support Vector Machine Classifier

The support vector machine model is available in Pedregosa et al. (2011a) [2]. A detailed description of the mathematical and computational details of the support vector machine (SVM) can be found in Smola, Alex J. & Schölkopf, Bernhard (2004). In the simplest sense, the SVM looks for the optimal boundary between classes in feature-space. The easiest example of a boundary is a straight line in a plane that divides two classes of data. In figure 2.5 there are many different possible lines that separate the data perfectly. To find the optimal boundary, the algorithm selects the pair of datapoints that are closest to one another but opposite in class—these are the 'support vectors'—and selects the boundary that maximises the distance between them, called the margin. This principle extends to as many dimensions and classes as required, with boundaries consisting of $n-1$ dimensional hyperplanes embedded in n-dimensional space.

In cases where the data are not perfectly separable, a 'soft' margin may be used. This is a region either side of the decision boundary into which the model admits members of the opposite class, making it possible to build a model even though some of the data overlap.

Despite this concession, many datasets do not lend themselves to being separated linearly. In these cases, data may be transformed by a function called a kernel that maps it to $n+1$ dimensional space such that classes become linearly separable.

The calculations involved in transforming data in this manner rapidly become computationally expensive as dimensions are added. In two dimensions, for variables x1, x2 to transform by squaring the data requires computing the terms $x1, x2, x1^2, x2^2, x1x2$.

[2]https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

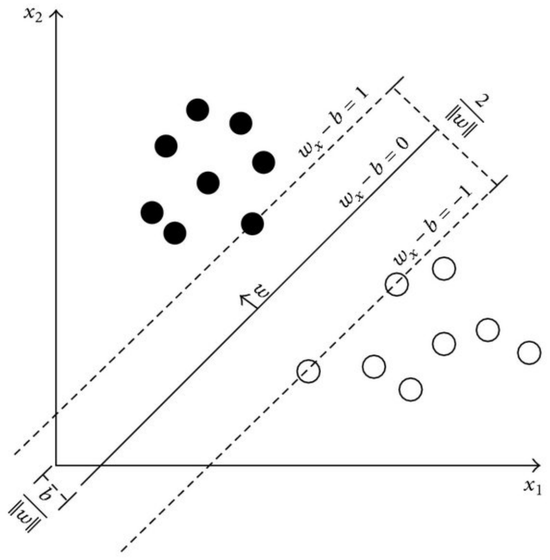


Figure 2.5: Most basic example of a linear decision boundary between two classes. Figure from Koo & Liew (2013)

Generalising, for the number of features k and degree of polynomial d, the number of computations can be described as $\frac{(k+d+1)!}{k!}$ which is exponential in degree d. However, the only information that is really needed is the spatial relationship between the vectors, and their distance from one another is the same irrespective of the number of dimensions in the hyperspace. This distance is expressed by the dot product of the vectors, which is invariant under translation. The following example from Wilimitis (2018) demonstrates how this works for two vectors **a** and **b**, but it applies to any number of dimensions. The function ϕ refers to the kernel, in this case a second-degree polynomial for simplicity:

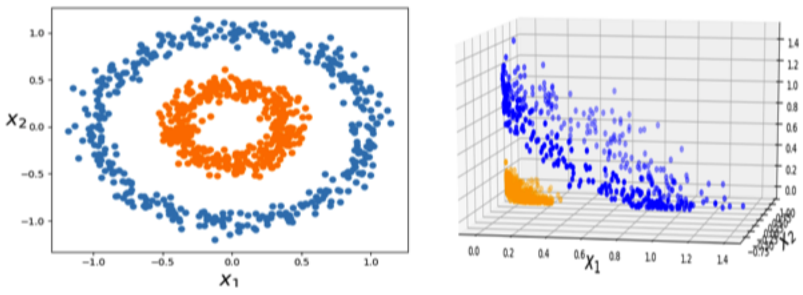


Figure 2.6: Illustration of the kernel trick for the polynomial kernel. Left: data in two dimensions, linearly inseparable. Right: After transformation into three dimensions via a quadratic kernel, data can be separated with a hyperplane. Example from Wilimitis (2018).

$$\begin{aligned} \phi(\mathbf{a}^T) \cdot \phi(\mathbf{b}) = \begin{pmatrix} a_1^2 \\ \sqrt{2}a_1a_2 \\ a_1^2 \end{pmatrix} \cdot \begin{pmatrix} b_1^2 \\ \sqrt{2}b_1b_2 \\ b_1^2 \end{pmatrix} \quad &= a_1^2b_1^2 + 2a_1b_1a_2b_2 + a_2^2b_2^2 \\ &= (\mathrm{a}_1b_1 + a_2b_2)^2 = \left(\begin{pmatrix} a_1 \\ a_2 \end{pmatrix}^T \cdot \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \right)^2 \\ &= (\mathbf{a}^T \cdot \mathbf{b})^2 \end{aligned} \tag{2.1}$$

This simplification is called the kernel trick, and it means that instead of working out all the dot products in higher-dimensional space, one can merely take the dot product of the vectors of interest *in the original space* and apply the kernel to the result. Instead of scaling exponentially, the process goes like $k + \log(d)$. The quadratic kernel is illustrated in Figure2.6.

The kernel most widely used in a support vector machine is the radial basis function (RBF), which is a Gaussian kernel defined as:

$$K(x, l) = \frac{e^{(-||x^2 - l^2||)}}{2\gamma^2} \tag{2.2}$$

where l is a support vector and x is the vector being measured. Here the γ hyperparameter determines the sharpness of the peak of the distribution, with a small value indicating a narrow region, steeply peaking, and a large value the opposite. The SVM decision boundary depends on the cost, a hyperparameter called C, which acts as a penalty applied to wrong classifications. Essentially C is what determines the softness of the margin (its values usually range from $10^{-4} - 10$ depending on context). The same cost is applied across all data points unless otherwise specified. In cases where data is imbalanced, the cost function can be multiplied by a weight for each class, so that there is a higher cost per data point for minority classes. Weighting the cost attempts to equalise the impacts of the classes on the shape of the decision boundary without resorting to oversampling. The downside to a weighted approach is the risk of overfitting when basing a decision boundary on a small number of points in a minority class.

In broad terms, this phenomenon is known as the 'bias-variance trade-off', where bias in this context refers to the crudeness of the boundary; i.e., the boundary is biased against the correct classification of some objects. Variance in this context refers to the detail of the boundary. In the bias-variance trade-off, one can either have a simple (but not very good) boundary, or a perfect (but overfitted) one. Even with the benefit of cross validation to select hyperparameters, there are judgement calls to be made in light of the fact that new data are almost certainly going to make a mess of an overfitted model.

The other important hyperparameter is γ, which determines the sharpness of the Gaussian peak in the radial basis function kernel that determines the region of influence governed by a support vector. It usually ranges between $0.1 - 100$ and like C can be optimised with cross-validation in a pipeline over a range of values.

2.5 Alternative classifiers

The other classifiers explored during the course of this project are described below. I have selected them because they each use different methods to arrive at their results (only random forest and XGB are related) and because they are easy to implement from Python libraries.

2.5.1 Random Forest

The random forest classification method is based on the fundamental unit of a classification and regression tree () more commonly known as a decision tree (Breiman & Stone, 1984). When classifying data, a decision tree will split the data according to a hierarchical series of features (e.g., color, size). Each feature implies a question that the input data must answer, travelling down branches and sub-branches until finally the leaves on the tree form the output classes. The tree is grown (and pruned) using relatively simple machine learning optimization methods.

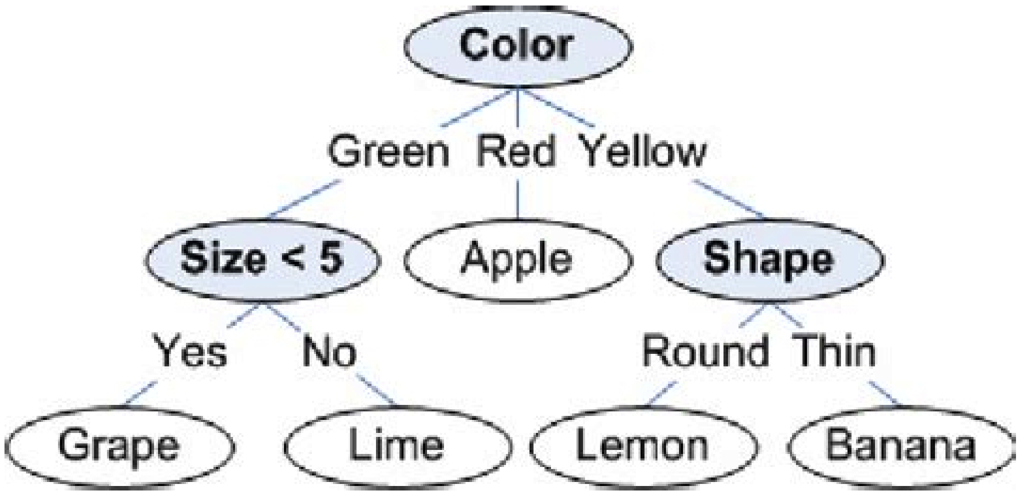


FIGURE 2.7: Simple decision tree. Image from Mikulski (2006)

The problem with decision trees is that they depend sensitively on the data used to construct them. There is a tendency for the individual tree to become too specialized, and as in the case of high variance for the SVM, specialization leads to overfitting. A process called 'regularization' (which effectively prunes branches relating to features of less importance) can address overfitting to some extent, but at the price of less accuracy.

Random forests are a very successful answer to overfitting that can greatly improve flexibility and performance of decision trees. A random forest is a group of decision trees constructed in such a manner that the trees are relatively uncorrelated to one another. The trees are grown from data that has been bootstrapped and bagged. Bootstrapping means that many copies of the original data have been made in which the data is randomly 'messed up' a little, so that each decision tree will be slightly different. Once the bootstrapped decision trees have been grown, they are grouped together, or 'bagged,' and the majority output class is chosen as 'correct'. In the example in Figure 2.7, if a bag of five trees produced 'lime', 'lemon', 'lemon', 'lemon', 'lime', the class 'lemon' would be chosen.

Bagging is an example of ensemble learning, and it increases the robustness of predictions. Crucially, when growing the individual trees, the random forest method doesn't allow the machine to 'see' all of the features (or levels) as it is growing; instead, the tree is made of a random sampling of decision branches. Because of this randomness, the correlation between individual trees is reduced; i.e., they are less likely to behave in similar ways, and less likely to overfit (Brownlee, 2019).

2.5.2 Extreme Gradient Boost (XGB)

Extreme Gradient Boosting (XGB) is a random forest algorithm that employs a technique known as 'gradient boosting' introduced by Friedman (2001). The gradient here refers to the cost (or loss) function that characterises the difference between the true data and the model's approximation of it. We are always trying to find the lowest error by getting to the bottom of the loss function, which we do by moving 'down' its gradient (see stochastic gradient descent in Figure 2.8).

Just as random forest uses ensemble learning by taking the majority 'vote' from multiple decorrelated models (trees), gradient boosting also uses ensemble learning, but in a sequential way. The key to this method lies in using individual models with high bias (these are called 'weak learners') to train one another in sequence. Each new model uses gradient descent to add a function of the residuals from the first model such that variance is reduced. The new model remains simple with low bias, but it because its variance is reduced, the method avoids the bias-variance trade-off. Effectively we end up with the best of both worlds. XGB in particular uses parallel processing at the node level to speed up learning, and flexible regularization to reduce overfitting (XGBoostDevelopers,

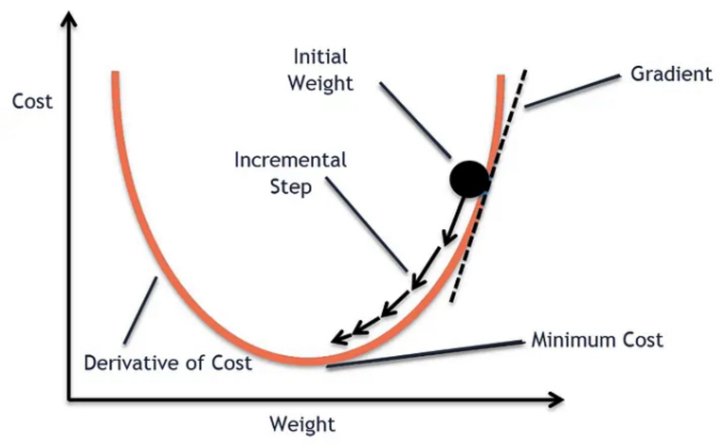


FIGURE 2.8: Intuition for stochastic gradient descent. The optimiser updates each weight to travel 'down' the gradient to get closer to the (local) minimum using a stochastic process to set the step size toward each new weight. Adam ('adaptive moment estimation') is a type of stochastic gradient descent. Figure from Kapil (2019)

2022). A large number of hyperparameters need to be tuned using cross-validation as described for SVMs.

2.5.3 Multi-layer perceptron (MLP)

In classification tests I also used the multi-layer perceptron (MLP) algorithm in Scikit-learn (Pedregosa et al., 2007). MLP is an implementation of an artificial neural network (), described in Section 2.6.2.1.

2.6 Interpolation vs. self-supervised learning for data augmentation

One of the characteristics of asteroid data is the predominance of S-types and, to a lesser extent, C-types in the data. This phenomenon is known as data imbalance, and it is problematic for machine learning. When a person is assigning a linear boundary 'by eye' it may not matter if different groups of data in feature-space have different sizes, but it can be a severe problem for machine learning classification. One reason for this is that many models learn by accumulating examples of characteristics, and the characteristics of a minority class example are easily swamped by the majority class. Even in PCA we can envisage how overall data variance can be more heavily affected by the features that characterise many samples at the expense of features present in only a few samples.

I have stated that the use of cost weighting with the SVM as described in the previous section runs a risk of overfitting to a small number of minority-class datapoints. On the other hand, the generation of synthetic data is always speculative. If the synthetic data are too similar to the source data, overfitting is ensured; but if they are too different, there is no guarantee that the class of the new sample would be the same as its parent object. Additionally, since the true underlying distribution of data is unknown, we cannot know which points in our sample are outliers and which are central to the distribution; we risk oversampling points that are actually noise.

To deal with this problem, we could cut the sample size of the larger classes down, but the disadvantages of this are obvious. We want to keep the data that we have. We could repeatedly draw more samples from the minority class population, replicating the minority data that we already have; but this risks the classifier overfitting to the specific features. So instead, the approach is to augment the data by generating some form of synthetic samples that are similar but not identical to what we already have.

I have sought to balance these concerns by choosing one augmentation method that is conservative in that it uses datapoints within the spatial confines of existing classes, and a second method that is speculative because it introduces noise to existing classes. The methods are, respectively, synthetic minority oversampling with edited nearest neighbours, and the variational autoencoder.

2.6.1 Synthetic minority oversampling with edited nearest neighbours (SMOTE)

In synthetic minority oversampling (SMOTE-ENN), the algorithm leaves the majority class alone. For each minority class, it generates new samples in the neighbourhood of existing datapoints. This is done feature by feature. For each point x in the sample, SMOTE finds a set of nearest neighbours (usually five) by measuring the Euclidean distance between x and every other datapoint. It randomly selects one neighbour. For every feature, it then takes the difference between the datapoint and the neighbour and multiplies that value by a random number between 0 and 1. The new value is now a feature in a new vector. The process is repeated for all features until a synthetic sample has been generated Chawla et al. (2002). The number of nearest neighbours ($k_{neighbours}$) is the only real tunable hyperparameter here.

In order to reduce the risk of introducing new samples that fall in a different class to the parent object, the Edited Nearest Neighbours (ENN) algorithm is added to the mix. ENN only allows SMOTE to resample on data that has the same class as at least two of its three nearest neighbours (Wilson, 1972). I have used SMOTE-ENN from scikit-learn's imblearn package (Lemaître et al., 2017), which is fast and easy. An example of the result on spectra can be seen in Figure 6.2. The new data are confined to the original class boundaries, but the linear nature of the interpolation is noticeable, especially in the sparser classes like A and V. The repercussions of this effect are a tendency to leave an artificial spiderweb-style substructure in the n-dimensional space of the data. Neural networks, described in the next section, are primed to detect substructure of this kind in the data, and there is a risk of overfitting to the substructure.

2.6.2 Variational autoencoder (VAE)

The variational autoencoder VAE is a type of ANN designed specifically to generate new samples. I will first address the basic features of an ANN before explaining what a variational autoencoder specifically does.

2.6.2.1 Artificial neural network (ANN)

The simplest unit of the network, the artificial neuron, is a mathematical simplification of a biological neuron. As shown in Figure 2.9, in the brain a neuron receives multiple neurotransmitter inputs from neighbouring neurons via thousands of dendrites. When the strength of a chemical signal exceeds a fixed threshold ('action potential'), the neuron fires and delivers neurotransmitter to the next neuron in the network via its axon. If the action potential threshold isn't reached, then the neuron does not fire (Newman & Han, 2017).

The first artificial neuron was called the perceptron, developed by Rosenblatt (1958) and subsequently Minsky & Papert (1969). The perceptron remains the fundamental unit of current ANNs. In a network, the neuron takes as input the output of all neurons connected to it on the input side. Each neuron has a set of associated weights–which refer to how much importance it attaches to the input it receives from every other neuron on the input side—as well as a bias that is specific to the neuron itself. The neuron

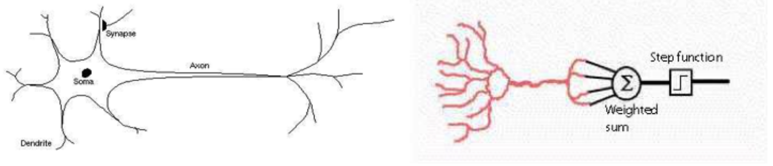


FIGURE 2.9: Cartoon of a biological neuron (left) and the perceptron it inspired (right). Figure from Roberts (2022)

.

calculates a weighted sum of inputs with bias and applies an activation function, so named because the outcome determines whether the artificial neuron is ‘activated’ and subsequently fires.

$$output = a(\sum_{i=0}^{i} x_i w_i + b)$$

The output of the activation function is a number between zero and one. This value becomes the input to the next neuron in the array.

Figure 2.10 shows the difference between four different activation functions, each of which will ‘fire’ when the output exceeds zero. The simplest activation function is the step function, which feeds forward nothing unless the output is greater than zero, in which case it always delivers one. The sigmoid, tanh, and rectified linear unit () functions are more nuanced in their output. In this work, I used a grid search to choose the ReLU activation function, defined as $a(x) = max(0, x)$. In a deep network, when the maximum that can be returned by each neuron is unity, then after many multiplications the signal becomes smaller and smaller until finally the network cannot learn. This is called the ‘vanishing gradient problem’ where the gradient refers to the gradient of the loss function (explained below).

ANNs are constructed in layers. Each layer represents a parallel set of neurons (sometimes called ’filters’) that receives input from the preceding layer and delivers output to the subsequent layer but does not interact with its own layer (see Figure 2.11. The input layer has as many neurons as there are dimensions in the data, whereas the size of the output layer varies according to the number of classes described in the problem space. For a binary classifier, there are only two outputs. Only the input and output

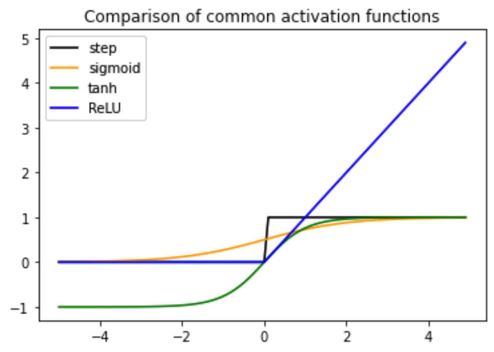


FIGURE 2.10: Illustration of four different activation functions: step, sigmoid, tanh, and relu. Relative strength of input signal is shown on the x-axis and of output on the y-axis. The neuron will always feed forward when the output of the activation function exceeds zero, but the weight to be fed forward depends on the nature of the activation function itself.

layers are seen by the user. All others are 'hidden layers' and in this case the process is described as 'deep learning'. In a deep network, 'forward propagation' refers to the flow of information from the input side through the hidden layers of the network to the output side.

An ANN is a sophisticated trial and error machine. It starts out knowing nothing: all weights and biases are assigned randomly to values between zero and one at the start. The network adjusts these parameters individually during training as it 'learns' until it reaches the optimum weights and biases required to produce the target output. To do this, an algorithm must compare the outcome of each iteration of forward propagation to a desired target and calculate the error (or cost). The error used in the variational autoencoder is the Kullback-Leibler (KL) divergence (see 2.7.4.5 for details), which is a method to quantify the difference between two probability distributions,the known data P and the hidden distribution Q in a probability space χ:

$$D_{KL}(P||Q) = -\sum_{x \epsilon \chi} P(x) \log \left(\frac{Q(x)}{P(x)} \right)$$

The output of this function is both positive and differentiable, which is important considering the role that the cost function plays in training the network. Training involves a process of repeatedly running the data through the network, updating weights and biases each time, for a set number of repetitions called 'epochs'. The idea is to converge on the best parameters possible for the data.

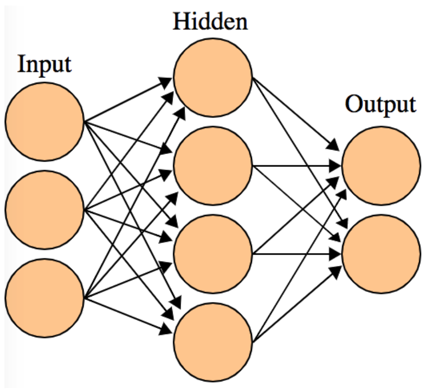


FIGURE 2.11: Cartoon of an ANN with three inputs and two outputs. There can be any number of filters in the hidden layers of the network, and any number of layers. Figure from Cburnett (2022).

2.6.2.2 Backpropagation

Optimization of the ANN's weights and biases occurs through a process called gradient descent with backpropagation. Gradient descent is the optimization itself, which is performed by an algorithm that seeks to reach the global minimum of the KL loss function as quickly as possible. To do this, weights are updated to move the output in the opposite direction to the gradient of the loss function at that point; i.e., to move 'downhill' towards the smallest error. (For an intuitive view of gradient descent, see Figure 2.8.) The method that facilitates this 'learning' is backpropagation.

Backpropagation is an algorithm that takes the partial derivatives of the KL divergence for successive hidden layers using the chain rule. It is applied backward from the output side of the network to the input side. In this way, every weight and bias is updated in such a fashion that KL divergence becomes smaller in the subsequent epochs. There are various ways to customize how far and how fast a parameter moves down the gradient to enable the optimisation to happen more efficiently as well as to avoid getting 'stuck' in a local minimum of the error function. For the variational autoencoder in this work, I use the Adaptive Moment (Adam) optimizer (Kingma & Ba, 2015), described below based on material from (Jiang, 2020) and (Brownlee, 2021)

This walk-through of Adam considers only a single parameter (e.g., a single weight) denoted x; in reality, the computations would be done on a vector. As its name suggests, Adam uses the moments of the KL divergence work with momentum, allowing it to converge more quickly on a global (rather than local) minimum. To do this, Adam computes the exponential moving average of the first and second moments of the loss, and it also contains decay terms that are analogous to friction, helping to prevent oscillation around a minimum.

The first moment (derivative) of x is denoted m and the second moment (derivative of the square, or 'uncentered variance') is denoted ν. The variance here refers to how much oscillation there is as we descend to the lowest value of the KL divergence. There are two hyperparameters $\beta_1(t)$ and $\beta_2(t)$ that work as decay terms for the first and second moments, set at 0.9 and 0.99 respectively, as well as ϵ, set to a small number such as 10^{-8} to prevent a division by zero error (if a number gets too small, the compiler will round it to zero otherwise). The step size α is set here at 0.001. all these are typical/default values. Since we are moving through the network in epochs, we start the time at $t = 1$ and increment it by one for each training epoch. We then compute the gradient and first and second moments.

$$\nabla(t) = f'(x(t-1))$$

$$m(t) = \beta_1 * m(t-1) + (1-\beta_1) * \nabla(t)$$

$$\nu(t) = \beta_2 * \nu(t-1) + (1-\beta_2) * \nabla(t)^2$$

The use of the decay parameters here means that both the gradient of x and the rate of change of x are adaptive, so that the descent is as fast as possible but also as smooth as possible. We do not know the initial m and ν, but they are initialized to zero. Because this choice creates a bias in the initial moments, a correction is introduced:

$$\hat{m}(t) = \frac{m(t)}{1-\beta_1(t)}$$

and

$$\hat{\nu}(t) = \frac{\nu(t)}{1-\beta_2(t)}$$

In order to nudge the weight in the opposite direction to the gradient of the loss function, the updating of a single weight then looks like this:

$$x(t) = x(t-1) - \frac{\alpha * \hat{m}(t)}{(\sqrt{\hat{\nu}(t)} + \epsilon)}$$

One of the reasons why Adam is so widely used is because it is very effective at optimising 'mini-batches' of data, which is necessary when dealing with big data, especially image processing. In hindsight, for a relatively small and shallow neural network like the one I use here, a simpler stochastic gradient descent probably would have worked fine.

The ANN, then, consists of an input layer, some hidden layers of neurons that apply weights and biases to the data and pass it on, and an output layer. A loss function is used to compare the output to some target value (maybe a class, if the network is a classifier) and optimise the way that the weights are adjusted. Adjustment of weights proceeds backward through the system, and the data are run through the network again. This process repeats for a set number of epochs. Once the network has been trained, new data can be run through it with the expectation that the network will (for example) classify it accurately. However, in this project I use ANNs not for classification, but for generating new spectra of various classes to augment the data. Specifically, I use a pair of one-dimensional convolutional networks arranged as a variational autoencoder.

2.6.2.3 Autoencoders

Before describing the variational autoencoder I will introduce the architecture of an autoencoder (AE). I rely heavily on blog articles by Rocca (2019) and Chollet (2016) for the explanation of both autoencoders and of variational autoencoders (in the next subsection). For the work in this thesis I use the Chollet et al. (2015) package and I base my code on the example by Chaudhary (2020).

An autoencoder is a self-supervised method comprised of two convolutional neural networks that mirror one another's architecture. The first network is an encoder that compresses large numbers of input features into a small set of encoded features called the latent state. The second is a decoder that reverses this process, taking the latent state variables and deconvolving them through the network to generate a full-dimensional output that acts as an approximation of the input. Figure 2.12 illustrates the generic architecture for an autoencoder. Just as an ANN uses gradient descent and backpropagation to set weights to differentiate between target classes (for example), the autoencoder backpropagates errors through both encoder and decoder to compress and decompress a sample as accurately as possible. This process is determined entirely by the data.

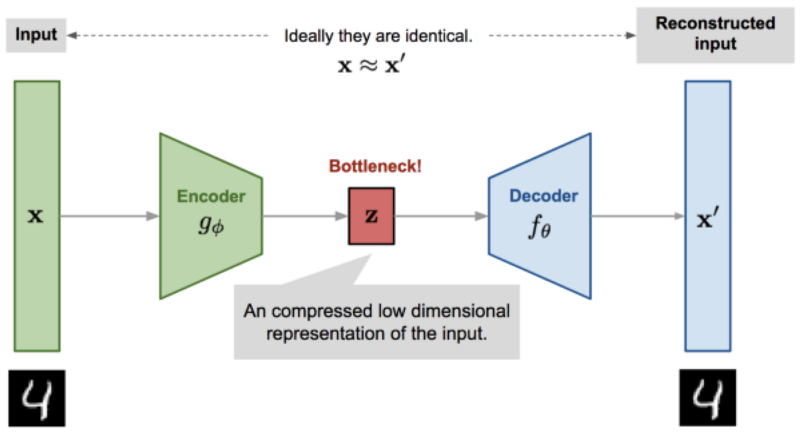

Input
Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Reconstructed input
$\mathbf{x}$
Encoder
g_ϕ
Bottleneck!
$\mathbf{z}$
Decoder
f_θ
$\mathbf{x}'$
An compressed low dimensional representation of the input.

FIGURE 2.12: Illustration of a vanilla autoencoder structure. This example shows a reconstruction of the handwritten number 4 from the MNIST dataset using a trained autoencoder. Image from Saikia (2019).

A typical application is the denoising autoencoder, where noisy input is presented together with a clean target. The model would learn how to extract the characteristic noise of that dataset, so that subsequent noisy samples could be cleaned automatically. Another use is dimensionality reduction. Like any neural network, a trained autoencoder has a set of weights for both encoder and decoder, and it has a 'bottleneck', or the latent state, where all features are encoded in a compressed manner. The process is similar to principal component analysis, and the latent state would be an analogous representation of the data if there were only one layer in the model and no convolutions were used. Like PCA, a trained autoencoder allows one to extract the latent state and plot it in two or three dimensions for the purposes of classification. However, unlike PCA, the latent space of a deep convolutional model does not give access to eigenvalues that are in turn traceable to the original features. In fact, if there are no constraints on what the latent space is, the model can create extremely complex encodings and decodings through the layers even if the latent space is one-dimensional. I saw this when attempting to use an autoencoder for dimensionality-reduction as an alternative to PCA. The autoencoder produced a good reproduction of the spectra, but with a bothersome effect: due to the stochastic nature of the model, on different runs identical data sometimes occupied two dimensions of latent space, sometimes three, and sometimes only one dimension. When all spectra are sitting on the same point, obviously it is not possible to use dimensionality reduction to assist in classification.

This outcome points to a larger issue for autoencoders: lack of regularisation of the latent space. As far as the autoencoder knows, the latent space has no existence independent of the data on which it has been trained. This is a problem if one wishes to use the autoencoder as a generative model. The use of the reconstruction loss (i.e., the loss function on the reconstruction of the data vs. the original data) shapes the latent space such that similar points are clustered near to one another but are separated from dissimilar points with empty space that essentially has no meaning. People got the idea of feeding in coordinates to the latent space that are close to the coordinates that the model has learned, in hope of getting an output similar but non-identical to the training example. However, selecting a latent space point that the decoder doesn't 'know' would generate a meaningless output because all weights and biases in the model are specific to the training data; the network is grossly overfitted. For the same reason, the latent space is not mathematically continuous. In a continuous space, points that are near to one another are expected to represent similar objects and vice versa. This is not the case for a vanilla autoencoder.

2.6.2.4 Variational autoencoders

The variational autoencoder was first proposed by Kingma & Welling (2014). It is designed to answer the problem of lack of regularization of the latent state; i.e., lack of completeness and lack of continuity. In doing so, it becomes capable of generating outputs that are similar, but not identical, to the inputs that trained it. The VAE is then a generative model with different capabilities to a classical autoencoder. While a vanilla autoencoder encodes data as points in the latent space, the VAE encodes data as samples from a distribution where the mean and variance are determined by the network during training.

When the network is trained, not only is the reconstruction loss computed, but it is added to the KL divergence between the Gaussian around the point in question and the Gaussian defined by the latent space from which it was drawn. In practice, this means

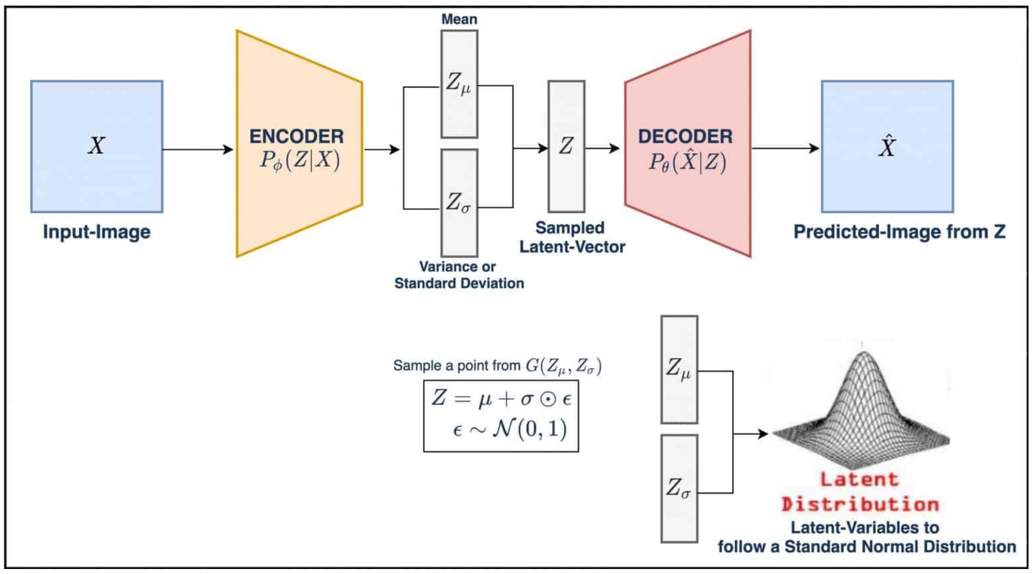

Mean
Z_μ
X
ENCODER
$P_\phi(Z|X)$
Z
DECODER
$P_\theta(\hat{X}|Z)$
$\hat{X}$
Z_σ
Input-Image
Sampled Latent-Vector
Predicted-Image from Z
Variance or Standard Deviation
Sample a point from $G(Z_\mu, Z_\sigma)$
$Z = \mu + \sigma \odot \epsilon$
$\epsilon \sim \mathcal{N}(0,1)$
Z_μ
Z_σ
Latent Distribution
Latent-Variables to follow a Standard Normal Distribution

FIGURE 2.13: Illustration of a variational autoencoder structure. The reconstructed sample is drawn from a latent space that has been forced into the shape of a Gaussian distribution to ensure continuity and completeness. The encoder is described in the illustration as an inference model because it is inferring the Gaussian underlying the data on which it is trained. Image from Sharma (2022).

that the latent space is defined continuously and that all points in the latent space have some meaning in terms of a Gaussian distribution. It now becomes possible to select a new point in the space that is midway between two existing points, and the output will be a blend of the characteristics of both.

With autoencoders the tuning of hyperparameters is not an issue in the same way as for other models discussed. Rather, the details of the network such as number of layers and dimensions of layers, need to be determined in a bespoke way by the user.

2.7 Unsupervised learning

In this section I will outline the workings of the three clustering algorithms used in this work (K-means, Gaussian mixture model, and HDBSCAN). I then describe each of the methods used to determine number of clusters and/or evaluate stability of the clustering.

2.7.1 K-means

This description of K-means is based on pp 424-425 in Bishop (2006) and the scikitlearn sci-kit learn developers (2022) documentation. The algorithm can be treated as an example of an Expectation Maximization (EM) problem, so named because it has an expectation step and a maximization step that follow one another iteratively. The

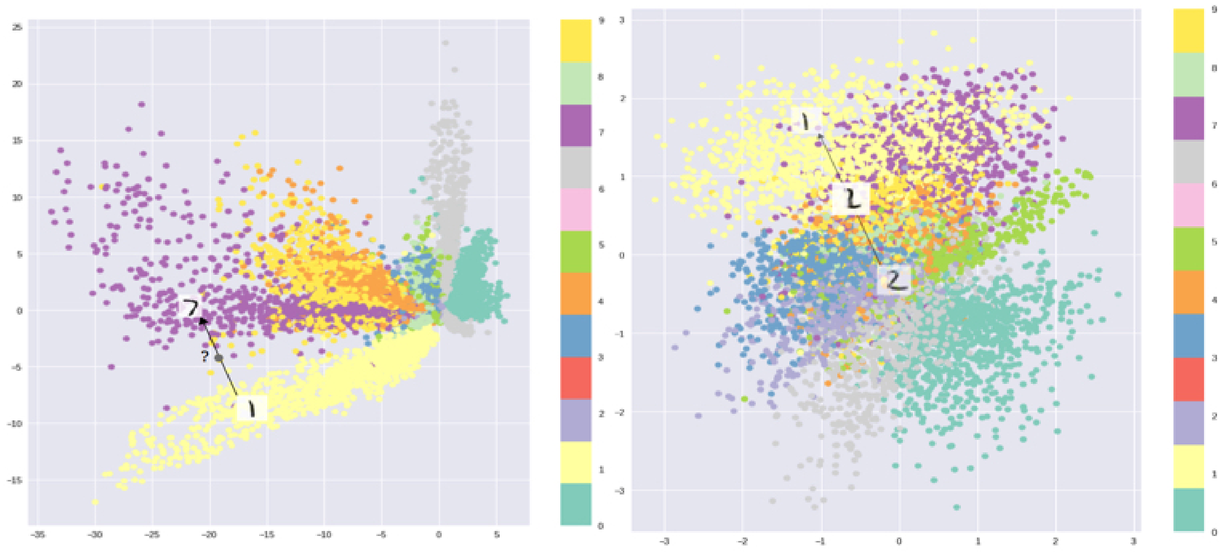


FIGURE 2.14: Plot of MNIST dataset of handwritten digits when transformed into two-dimensional latent space at the bottleneck of an autoencoder. Some variation of this plot is widely used to illustrate the difference between autoencoders and variational autoencoders. On the left, the model is trained by a vanilla autoencoder (i.e., on the reconstruction loss only) to reproduce the input samples exactly. On the right, a variational autoencoder is trained on the reconstruction loss plus KL divergence to generate a new sample midway between 1 and 2. By setting up a latent distribution and computing both the difference between it and the output in addition to the difference between input and output vectors, the algorithm maps the data as an overlapping set of distributions rather than a set of discrete clusters, while still learning the features of the input data. It is now possible to draw a sample from anywhere in the space and get a result that resembles its neighbours even though the network has never 'seen' that datapoint. Image from Shafkat (2018)).

expectation step makes an estimation for the expected values that we seek (in this case, the best available cluster centroid for each datapoint). The maximization step takes that result and alters it to maximize the output; here, this means minimizing the error within each centroid.

Assuming a dataset of N observations in D dimensions $\{\mathbf{x}_1, .., \mathbf{x}_N\}$ we want to arrange the data in K clusters such that the datapoints in each cluster are closer to one another than to the data outside that cluster. Given a number of clusters K, the algorithm finds a set of cluster means $\boldsymbol{\mu}_k$ such that the sum of squared Euclidean distances between the datapoints in the cluster and its mean (equivalent to sum of squared error, SSE, also called 'inertia') is as small as possible. Cluster membership is set by an indicator $r_{nk} \in \{0, 1\}$ where k represents a cluster number from 1 to K such that $r_{nk} = 1$ if x_n belongs in cluster k, otherwise $r_{nk} = 0$. The inertia between each point and its assigned μ_k is then expressed by the quadratic function

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \parallel \mathbf{x}_n - \boldsymbol{\mu}_k \parallel^2$$

To begin with, a random vector $\boldsymbol{\mu}_k$ is selected from among the datapoints. These are the initial expectation values, or cluster centroids. Expectation and maximization proceed as follows:

1. Expectation: For each $\mathbf{x}_n$, the algorithm measures the squared distance to each of the μ_k, then sets $r_{nk} = 1$ for the smallest value given the set of $\boldsymbol{\mu}_k$ it started with. Each point now has its best available expectation value.

2. Maximization: the algorithm finds the 'best' μ_k of the clusters that were assigned in the prior step. For each point, the r_{nk} chosen in the first step is held constant and the algorithm measures inertia to find the smallest J. The minimum of a quadratic function can be found by setting its derivative to zero, so

$$2\sum_{n=1}^{N} r_{nk}(\mathbf{x}_n - \mu_k) = 0$$

which rearranges to:

$$\mu_k = \sum_n \frac{r_{nk}\mathbf{x}_n}{\sum_n r_{nk}}$$

that represents the mean of datapoints assigned to a given cluster.

This process is repeated, with some points being re-assigned after the new means have been computed, until convergence. The process is sensitive to the initial $\boldsymbol{\mu}_k$ chosen and may converge on a local minimum. To answer this, the scikitlearn implementation will run the entire procedure ten times by default and choose the best $\boldsymbol{\mu}_k$ at the end.

K-means has several disadvantages. Because it measures distances between points, in high-dimensional spaces the data become sparse (the curse of dimensionality). The use of inertia also favours clusters whose members are isotropically distributed, which is not realistic for most data (sci-kit learn developers, 2022).

The most important hyperparameter for K-means is the number of clusters, which can either be known in advance or found automatically using methods such as elbow, AIC, and silhouette method discussed in this chapter. If the number of clusters is known, one can initialise with cluster centroids. Inertia can also be fine-tuned if desired.

2.7.2 Gaussian mixture model

The equations and rationale in this sub-section have been assembled using pages 14-15, 78-94, 110-113, and 430-439 of Bishop (2006) to the best of my understanding.

Unlike K-means, which assigns each point definitively to a single 'hard' cluster, a Gaussian mixture model (GMM) is a probabilistic model that uses 'soft' clustering. The model assumes that the data are drawn from a set of K Gaussian distributions whose means and variances are unknown. Like K-means, GMM initializes a set of means $\boldsymbol{\mu}_k$ before running the EM algorithm to find the the most likely component (cluster) for each point and the best mean for each component. However, GMM also involves calculating covariance matrices $\boldsymbol{\Sigma}_k$ in order to define the latent Gaussians. Another difference is that the K-means binary variable r_{nk} (which ensures that each datapoint is assigned to only one cluster) will be replaced by a vector of mixing coefficients $\boldsymbol{\pi}_k$ that represents the probabilities of the point being assigned to each Gaussian component.

I will explain the basis of the GMM model before laying out the optimisation procedure for EM. Let's first write down the univariate Gaussian.

$$\mathcal{N}(\mathbf{x}|\mu, \sigma^{\mathbf{2}}) \equiv \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{1}{2\sigma^2}(x-\mu)^2)$$

For a D-dimensional vector $\mathbf{x}$ the multivariate Gaussian can be written as:

$$\mathcal{N}(\mathbf{x}|\mu, \boldsymbol{\Sigma}) = \frac{1}{2\pi^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp(-1/2(\mathbf{x}-\boldsymbol{\mu})^T(\mathbf{x}-\boldsymbol{\mu}))$$

Note that in both above equations π takes its usual meaning, but it is conventional to repurpose the symbol as the mixing coefficient. To simplify notation, we will write the probability distribution for the mixture as

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_{\mathbf{k}}, \boldsymbol{\Sigma}_{\mathbf{k}}))$$

in which there are K components and the mixing coefficients $\boldsymbol{\pi}$ represent percentage contributions; i.e., each element lies in the interval (0,1) and their values sum to 1.

We can think of $\mathbf{X}$ as the observed data sampled from an unknown 'true' latent mixture of Gaussians, and we will be evaluating the probabilities that each observed datapoint belongs to each component k. We need to work out not 'which component does x_n belong to' but instead 'to what degree does it belong to each of these components?' In order to deal with the latter question, we introduce a latent binary variable vector $\mathbf{z}$, for which z_k can either be 0 or 1, meaning that there are K possible permutations of $\mathbf{z}$. By introducing $\mathbf{z}$ we invoke a joint probability distribution.

At this point we say, for a given component (that is, for the k in which $z_k = 1$), the probability of a datapoint being sampled from it is expressed by the mixture coefficient π_k

$$\pi_k = p(z_k = 1)$$

Eventually, π_k is going to allow us to get a log-likelihood in terms of $\mathbf{X}$ only. To start with, we are going to use $\mathbf{z}$ to find out how 'responsible' each Gaussian component is for a given datapoint. Responsibility is defined in terms of $\mathbf{z}$ conditional on $\mathbf{x}$. For this we need Bayes' Theorem

$$P(Y|X) = \frac{p(X|Y)p(Y)}{P(X)}$$

where $P(X)$ will be the sum of all possible components from the counter $j = 1$ to K:

$$\gamma(z_k) = p(z_k = 1|\mathbf{x}) = \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)}$$

We can now replace these probabilities with π_k and the normal distribution of $\mathbf{x}$:

$$\gamma(z_k) = \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \mathbf{\Sigma_k})}{\sum_l \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_\mathbf{l}, \mathbf{\Sigma_l})}$$

Now that we have responsibility, we can use it to work out the average means, covariances, and mixing coefficients. Once we have those, we need to get rid of $\mathbf{z}$, because ultimately we need our probabilities in terms of x_n only. For this a technique called marginalisation will be used. To marginalize (in this case) means to sum over all possible values of $\mathbf{z}$ and so obtain a probability distribution for $\mathbf{X}$ alone.

We treat every observed (feature) vector $\mathbf{x}$ as normally distributed in each component z_k:

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{\Sigma})$$

By the product rule of probabilities, the joint distribution is then:

$$p(\mathbf{z}, \mathbf{x}) = p(\mathbf{z}|\mathbf{x})p(\mathbf{x})$$

By summing over all possible states of $\mathbf{z}$, we can marginalize out the latent variable and get back the distribution in $\mathbf{x}$ that we started with, noting that for every x_n

there is a corresponding latent vector $\mathbf{z}_n$ as required. Marginalization looks like this:

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_{\mathbf{k}}, \mathbf{\Sigma_k}))$$

Finally, we need to define the log-likelihood of our probability distribution. There are several reasons for choosing the natural log of the distribution, the most obvious of which is that products of logs can be transformed into sums. This is desirable because when dealing with probabilities, products of small numbers can be lost when they fall below the computational limits of floating-point decimals. The log-likelihood over the entire dataset of length N is defined as:

$$\ln(\mathbf{X}|\pi, \boldsymbol{\mu}, \mathbf{\Sigma}) = \sum_{n=1}^{N} \ln \left[\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \mathbf{\Sigma}_k) \right]$$

To find the maximum, the derivative is set to zero with respect to the mean and covariance in turn. In the case of the mean this is:

$$0 = \sum_{n=1}^{N} \left[\frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \mathbf{\Sigma}_k)}{\mathbf{\Sigma}_j \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \mathbf{\Sigma}_j)} \right] \Sigma^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k)$$

The term in brackets is the responsibility of z_k, and with some manipulation can be written as $\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})\mathbf{x}_n$ where $N_k \equiv \sum_{n=1}^{N} \gamma(z_{nk})$ to indicate the number of points assigned to component k. We are obtaining the average responsibility for component k over all N datapoints.

All the components needed for EM are now in place. The algorithm flows as described in Bishop (2006) p. 438-9:

- Initialize $\boldsymbol{\mu}_k$ (cluster mean) and π_k (cluster identity). Calculate the covariance matrix $\mathbf{\Sigma}_k$ and get the log-likelihood for these values.
- Expectation step: Calculate every $\gamma(z_{nk})$ from current parameter values.
- Maximization step: Maximize the log-likelihood with respect to each variable:
  - $\boldsymbol{\mu}_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})\mathbf{x}_n$
  - $\mathbf{\Sigma}_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k^{new})(\mathbf{x}_n - \boldsymbol{\mu}_k^{new})^T$
  - $\pi_k^{new} = \frac{N_k}{N}$ where $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$
- Evaluate the log-likelihood and repeat the Expectation and Maximization steps until convergence.

The main hyperparameters for GMM are $n_{components}$, the number of Gaussians in the mixture, and covariance type, which describes the form of covariance matrix used. In scikit-learn developers (2022) there are a few ways to initialise the centroids to fit the Gaussians. They are based on K-means and their main differences are processing speed. Since GMM uses pairwise Euclidean distances anyway, the curse of dimensionality is the same as for K-means, but the problem of cluster shape that occurs with K-means does not happen with GMM as long as the spherical version of the covariance is not chosen. I have set the model for full covariance rather than the spherical, tied, or diagonal alternatives offered because full covariance performed best in Bayesian Information Criterion (BIC) tests on the asteroid data (method described in scikit-learn developers (2022)).

2.7.3 HDBSCAN

For most clustering methods it is necessary to input the number of clusters desired before running the model. The notable exception is Hierarchical Density-Based Spatial Clustering of Algorithms with Noise (HDBSCAN) (Campello et al., 2013).

HDBSCAN treats the data as a probability distribution but makes no assumptions about cluster shapes, sizes, or even densities, and unlike other methods discussed it makes allowance for both noise and outliers. It builds a hierarchical tree of clusters under the basic principle that a cluster is a group of points that are close together compared with the relative sparsity around them, like 'islands in the sea' (McInnes et al., 2016). Accordingly, the level of the 'sea' determines what gets labelled as a cluster.

In order to recast the data as a probability distribution function (PDF), the algorithm sets many sea levels for the data to find the optimal number of islands (clusters). One way to do this is to pick a number K that represents how many nearest neighbours to consider. A large K implies low sea level and vice versa. As described in McInnes et al. (2016), for a given K, HDBSCAN performs the following routine on every point:

- Find the smallest radial distance from the point that encloses K points. This is called the 'core distance' and is inversely proportional to the probability density function for that point.
- Compute the 'mutual reachability distance' to every pair of points (a, b), defined as: $d_{mreach-k}(a, b) = \max\{core_k(a), core_k(b), d(a, b)\}$ where $d(a, b)$ is the metric distance between a, b and the core distances refer to a particular value of K. If either of the points lives in a sparse region (i.e., the sphere drawn for that K is larger than the distance between (a, b)), their mutual

reachability distance will be *larger* than their original distance. This means that regions of low density become spread out even more under the metric $d(a, b)$.

- Now that data have been transformed into a minimum-reachability space, the idea is to identify the islands. HDBSCAN finds the 'minimum spanning tree' using Prim's algorithm, a fast and efficient way to connect all vertices of a graph. Prim's algorithm begins with an arbitrary datapoint (vertex) and picks the smallest possible weight (minimum reachability distance) on the edge to a new vertex. The process repeats until the entire dataset is connected (Prim, 1957). Figure 2.15 illustrates the outcome for some sample data with $K = 5$, where colour-coding emphasizes the importance of mutual reachability distance. Each 'island' has its own internal characteristics, and they are joined by long 'bridges.'

Once the above process is complete, HDBSCAN creates a hierarchy in the following steps:

- Edges are sorted by ascending distance
- Treat each potential cluster as a set. Starting with the two smallest mutual reachability distances, find the roots of the trees containing them, and if the roots are different, subsume one of the sets into the other by the union-find algorithm (see Galler & Fisher (1964) for details). The outcome can be visualised as a dendrogram that looks like the left panel of Figure 2.16
- HDBSCAN employs a user-determined parameter for minimum cluster size. If a split in the dendrogram leads to a cluster that is smaller than the minimum size, these datapoints are treated as noise/outliers and removed from

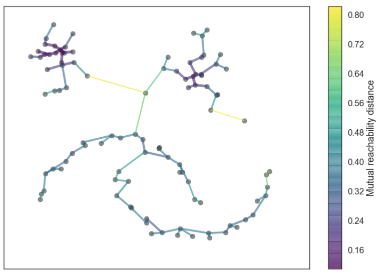


FIGURE 2.15: Illustration from McInnes et al. (2016) of the connected graph of some data plotted by mutual reachability distance: $d_{mreach-k}(a, b) = \max\{core_k(a), core_k(b), d(a, b)\}$ where $d(a, b)$ is the metric distance between a, b and the core distances refer to a particular value of K. The graph starts at any point (vertex) and agglomerates vertices where edges are weighted by mutual reachability distances, starting with the closest pairs of points and adding according to Prim's algorithm as described in text. In this case $K = 5$ and mutual reachability distance is encoded in the colour of the edges.

consideration. The dendrogram is now condensed and simplified as per the right panel of 2.16.

- In hierarchical clustering methods it is conventional to assign clusters by cutting across the lines in the dendrogram (equivalent to setting a global sea level). HDBSCAN departs from its predecessors by introducing the concept of 'persistence,' which is found through determining 'excess of mass' in a portion of a distribution, with area under a distribution analogous to mass in a probability density scenario. Figure 2.17 is included to provide intuition. Two distributions are shown, each with three peaks but of different shapes. In one case, the sum of the areas under the (separate) blue and yellow shaded regions is greater than the area under the shared green region, so blue and

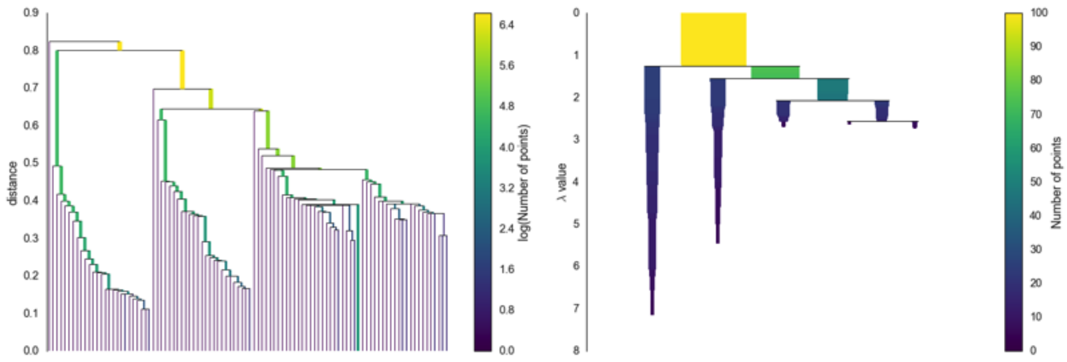


FIGURE 2.16: Dendrogram of sample datapoints by mutual reachability distance (left) and condensed version (right) from McInnes et al. (2016). To sever clusters from their parent distributions, one would cut through the vertical lines at a suitable point. HDBSCAN allows for a bespoke cluster selection process that relies on 'persistence'. Persistence boils down to measuring the area under each hierarchical coloured region in the right panel. If the child areas sum to more than their parent areas, the clusters are retained. Otherwise, the parent is the final cluster on that branch.

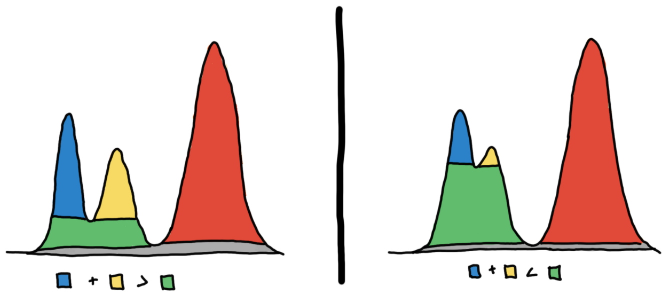


FIGURE 2.17: Illustration of the intuition behind 'persistence' in HDBSCAN. Viewing the data as a probability distribution, then one can compare the total area under each of the blue and yellow clusters to the shared area (green). If the blue and yellow areas are greater, these two clusters are said to have an 'excess of mass' and to be 'persistent' enough to constitute individual clusters (left); otherwise, they are both included in the green cluster (right). Implicit in this approach is the appropriate initial setting of the 'sea level' (grey). Figure from Berba (2020)

yellow are genuine clusters. In the other case, the opposite is true, so the entire region is treated as a single green cluster. Through this method, clusters that persist through having an excess of mass can have a variety of sizes, shapes, and densities relative to the whole distribution.

Although the parameter-setting process in HDBSCAN is relatively simple, it is very sensitive to minimum cluster size as an input parameter. When performing HDBSCAN I have initialized from a range of minimum cluster sizes to minimize the number of points assigned as noise. This process was manual and iterative. No matter how sophisticated the algorithms used here, there is no getting around manual fine-tuning.

2.7.4 Finding the number of clusters

2.7.4.1 Elbow plot

The elbow method (Yuan, 2019) is simple and can sometimes be judged 'by eye'. It runs K-means for a range of different cluster numbers (k) and computes the mean sum-of-squares error SSE between data points and their assigned cluster centres each time. A plot of the SSE (called 'distortion') against k will show the error dropping as cluster number increases. The 'elbow' is judged as the point of maximum curvature; that is, there are diminishing returns for increasing the number of clusters beyond this. On a relatively smooth curve, there isn't necessarily an obvious elbow. In the left side of figure 7.1 the python library Yellowbrick (scikit-yb developers, 2019) shows the distortion on K-means for SDSS

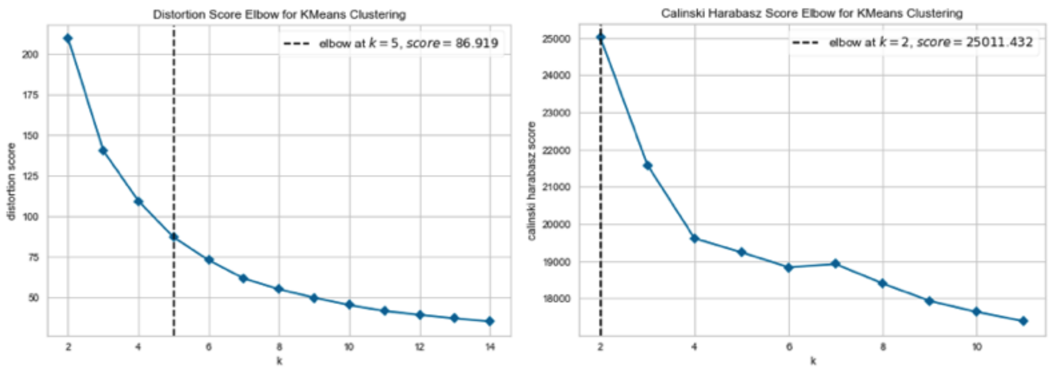


FIGURE 2.18: Example plots of SDSS asteroid data clustered with K-means. Left: elbow with distortion. In Yellowbrick (scikit-yb developers, 2019), the Kneedle algorithm (Satopaa et al., 2011) determines the point of maximum curvature. Right: Calinski-Harabsz index of the same data. In this case the two methods do not agree.

.

Carvano data. In this case the elbow, or point of maximum curvature, is found automatically at k=5 using the Kneedle algorithm (Satopaa et al., 2011).

2.7.4.2 Silhouette method

The silhouette is also used to determine the optimal number of K-means or Gaussian components in a dataset. It represents a measure of how tightly bound each point is to its assigned cluster relative to the nearest neighbouring cluster (Yuan, 2019). Silhouette is defined as:

$$s(i) = \frac{b(i) - a(i)}{max[a(i), b(i)]}$$

where $a(i)$ is the average distance between each point i and all the other points in its assigned cluster and $b(i)$ is the average distance between i and all the points in the nearest neighbouring cluster. The value of the silhouette ranges from -1 to 1, with 1 representing an ideal member of a cluster and -1 representing a point that fits perfectly to a different cluster.

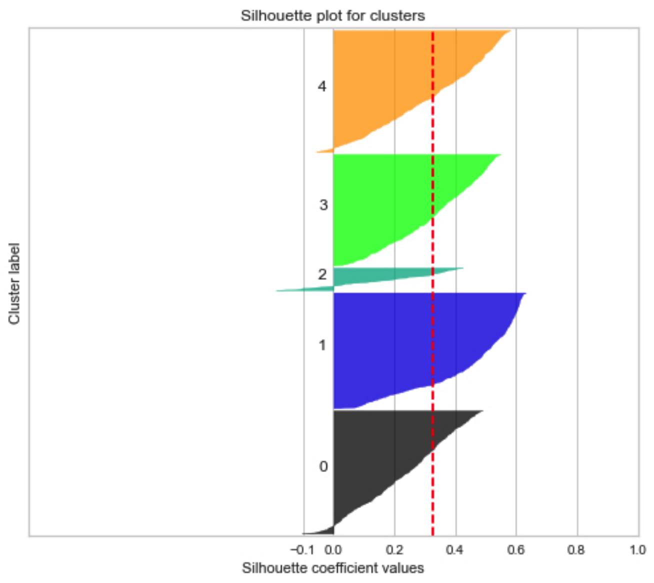


Figure 2.19: Illustration of a silhouette plot for five k-means clusters showing the relative sizes of clusters. The more distinct the cluster is from others, the higher its co-efficient on the x-axis. A negative coefficient means that the object is closer to other clusters than to its assigned cluster. The red line indicates the average silhouette score for all data; if a cluster fails to achieve this or barely achieves it, the number of clusters is not a good fit for the data. Here, Cluster 2 is not only much smaller than the others, but it also has a number of points that are closer to neighbouring clusters, and it barely reaches the red line. These data do not fit well with five k-means clusters

.

The visual representation of the silhouette is its most useful aspect (code used to generate silhouette plots for this project can be found at Drakos (2020)). Given a clustering algorithm (here, K-means and GMM) one runs the model for a range of cluster values, calculating the silhouettes each time. The silhouette score can be plotted against the number of clusters and the maximum value chosen. Much more information can be found by plotting a given set of clusters as an area. As in figure 2.19 a plot of each cluster's silhouette shows not only its height (whether it is above or below zero, and by how much) but also its width; i.e., how many points are contained in that cluster relative to the others. When used to find the number of clusters in K-means, the intention is for all the silhouettes to have similar widths, since K-means is optimised to find clusters of similar sizes. If some of the clusters have very slender silhouettes relative to the others it is an indication that K-means is not working well for the data. The silhouette also gives some feeling for the distribution of data at a glance. The example in figure 2.19 depicts K-means run on nine clusters in the SDSS. Each cluster from $0-8$ is shown in a different colour, increasing from 0 to 1 on the x-axis. The red dotted line is the average value of the silhouette. Clusters 2, 3, 4, 5 and 8 have some members that are nearer (on average) to the neighbouring cluster than to their assigned cluster, shown by their negative values of s. It is also evident that the cluster sizes are asymmetric, with 2 and 5 being particularly low in population.

2.7.4.3 Calinski-Harabsz score

The Calinski-Harabasz score is a ratio of the dispersion within a cluster to the average dispersion between clusters, so that a high score indicates good clustering (at least for algorithms like K-means in which proximity is more important than density or some other intra-cluster relationship). As described in Pyshark (2022), the score sums the SSE of points from their respective cluster centres to calculate a within-group sum-of-squares error for each cluster k over the number of points in that cluster n_k:

$$WGSS_k = \sum_{i=1}^{n_k} ||X_{ik} - C_k||^2$$

where C_k is the centroid of cluster k of K and X_{ik} is a point in that cluster.

The sum of squares of each point from the centre of the dataset C is summed separately:

$$BGSS_k = \sum_{k=1}^{K} n_k \times ||C_k - C||^2$$

The above two quantities are treated as a ratio to obtain the Calinski-Harabsz (CH) Index. If CH is high, the clustering is good.

$$CH = \frac{\frac{BGSS}{K-1}}{\frac{WGSS}{N-K}} = \frac{BGSS}{WGSS} \times \frac{N-K}{K-1}$$

An example is shown in the right-hand frame of Figure 2.18. Note that despite Yellowbrick's automatic (and misleading) title, the Calinski-Harabasz index does not look like an elbow except by coincidence in this case; the best number of clusters occurs at the point of highest value.

2.7.4.4 Davies-Bouldin Index

The Davies-Bouldin index is constructed from a cluster separation measure $R_{ij} \equiv \frac{S_i+S_j}{M_{ij}}$ where S_i, S_j are the respective dispersions of clusters i and j and M_{ij} is the distance between vectors that characterise clusters i and j (e.g., their centroid vectors). For each cluster i one calculates a similarity measure R_{ij} by iterating over the other clusters (as indexed by j) and retaining the maximum value R_i as a measure of the cluster's similarity with its nearest neighbour. The Davies-Bouldin index $\overline{\mathrm{R}}$ consists of a system-wide average of maximum similarity: $\overline{\mathrm{R}} \equiv \frac{1}{N}\sum_{i=1}^{N} R_i$. By minimising the index, one ensures that clusters are generally as different from one another as possible (Davies & Bouldin, 1979).

Figure 2.20 shows the Davies-Bouldin index for SDSS Carvano data after applying K-means. There is clearly a preference for two clusters; however, it also looks like four would be better than three, at least by this metric.

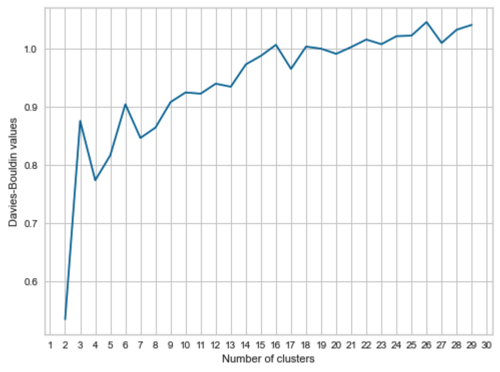


FIGURE 2.20: Example plot of SDSS asteroid data clustered with K-means over a range of clusters after evaluation of the Davies-Bouldin index. In this case the lowest value indicates the number of clusters where each cluster is most distinct from the others; i.e., two in this case.

2.7.4.5 Kullbeck-Leibler divergence & the Jensen-Shannon distance

The Kullbeck-Leibler divergence between two distributions originates in information theory and Shannon's concept of entropy in which information is conceptualised in terms of bits (Shannon, 1948). With respect to a randomly distributed discrete variable X, the quantity called information conveys how 'surprised' we are by a sample in terms of its probability distribution $p(x)$. In turn, entropy is the average information over the set X of all instances of X where every element lies in [0,1]. For example, in a uniform distribution we are always going to be surprised so entropy is high, but in a normal distribution there is less entropy because we already expect more samples near the mean than at the tails (Carter, 2011).

Entropy for a probability distribution $p(x)$ can be written as the expectation of the negative log of the probability, i.e.,

$$\mathbf{H}(p) = \mathbb{E}(-\log(p))$$

which is written for a discrete set of data as

$$\mathbf{H}[X] = -\sum_{x\in\mathcal{X}} p(x)\log p(x)$$

A related expression is the cross entropy, typically used to evaluate the information loss (or error, or cost) between a datapoint and the distribution predicted by a model. It refers to Shannon's idea of the expected number of bits needed to encode data drawn from distribution $p(x)$ so as to decode it based on distribution $q(x)$ (Murphy, 2022). This invokes a joint probability.

$$H(p,q) = -\sum_{x\in\mathcal{X}} p(x)\log(q(x))$$

The KL divergence also has to do with the change in entropy between one distribution and another. In information theory it is the difference between the cross-entropy and the entropy of $p(x)$, or:

$$D_{KL}(p\|q) = \mathbf{H}(p,q) - \mathbf{H}(p)$$

For a discrete dataset the KL divergence is written as:

$$D_{KL}(p\|q) = -\sum_{x\in\mathcal{X}} p(x)\log\left(\frac{q(x)}{p(x)}\right)$$

The KL divergence is also called the relative entropy between the true probability distribution $p(x)$ and some model distribution $q(x)$. It is used in variational autoencoders to train the encoder and decoder based on Gaussian distributions for the training data. KL divergence is not a true metric; for one thing, it is not symmetric with respect to p and q. However, using the KL divergence together with m, the pointwise mean between distributions p and q, the Jensen-Shannon distance is defined as

$$JS = \sqrt{\frac{KL(p||m) + KL(q||m)}{2}}$$

Thanks to the square root, the Jensen-Shannon distance works as a true metric that can quantify the difference between data drawn from two distributions. Here I have used it to assess the stability of the Gaussian mixture model algorithm with the method and code from Lavorini (2018). The basic idea is to divide the data in half randomly, fit to a GMM on each half independently, and check the similarity between the probability distributions of each half of the data. This is done multiple times with different 50/50 splits each time, and for a range of cluster numbers, always seeking a) the smallest Jensen-Shannon distance, and b) the smallest error on the Jensen-Shannon distance between one run and the next. If the errors on the Jensen-Shannon distance are high, then the GMM is unstable and the data cannot be well-fitted by the model.

2.7.4.6 Akaike & Bayesian Information Criteria

In his paper Akaike (1974) named his measurement 'AIC' with the intention that it would be the first of many versions in an alphabetical series ('BIC', 'CIC', etc.), but the 'A' is now used to denote Akaike Information Criterion whereas BIC stands for the related Bayesian Information criterion. Both are methods of model selection rooted in Shannon's theory of entropy in that they rely on the maximum likelihood estimate, also called the log-likelihood, which is an expression referring to optimization of the parameters θ_{MLE} of a model to maximize the probability of generating the data given all possible parameters $\boldsymbol{\theta}$. From section 4.1 of Murphy (2022), this optimisation boils down to minimizing the negative log likelihood for a set of independently adjusted parameters $\boldsymbol{\theta}$, expressed over a dataset of N points:

$$\boldsymbol{\theta_{MLE}} = -\sum_{n=1}^{N} \log(p(\mathbf{y_n}|\mathbf{x_n}, \boldsymbol{\theta}))$$

In AIC, the maximum likelihood is used to compute goodness of fit for a model:

$$AIC(\boldsymbol{\theta}) = (-2)\log(\boldsymbol{\theta_{MLE}}) + 2k$$

where k is the number of parameters $\boldsymbol{\theta}$.

In the context of this work (finding the optimal number of clusters) the parameters are the components in a Gaussian mixture model with full covariance. The so-called MAICE is a combination of AIC and 'minimum estimate', since the best value for AIC is found for the parameters at which it is minimised.

The BIC (Schwarz, 1978) represents a variation to AIC that penalises models having many parameters (to reduce overfitting) by altering the second term:

$$BIC(\theta) = (-2)\log(\boldsymbol{\theta_{MLE}}) + k * \log(N)$$

As before, for a good model fit we want a small value for the BIC, which means that a smaller number of parameters k is favoured, leading to simpler models. When selecting the number of clusters we are also looking for the change in AIC or BIC between one value of k and another, similar to the idea behind an elbow plot.

2.8 Summary of machine learning methods

In this chapter I have introduced the fundamental concepts in machine learning that underpin the methods used in my work. These include the train/test split, the meaning of an accuracy score, the ROC-AUC plot and confusion matrix plot, and the difference between supervised, self-supervised, and unsupervised learning as well the curse of dimensionality. I explained how principal component analysis works to reduce dimensions and introduced the support vector machine classifier. I outlined two different methods to augment the number of samples: SMOTE-ENN and the variational autoencoder. I then moved on to unsupervised learning, describing the K-means, Gaussian Mixture Model, and HDBSCAN algorithms. Finally, I explained the workings of several methods used in unsupervised learning to determine the number of clusters as well as the stability of the models. The techniques described are by no means an exhaustive list of tools appropriate to the tasks I undertake, but they represent the best methods of those that I was able to test.

Chapter 3

Data

All data in this project come from public catalogs. Every object has been assigned a classification according to either Bus-Binzel or Bus-DeMeo. Because there is an abundance of sparsely populated subclasses in the spectral reflectance data, I have had to make decisions about what to call a 'basic' class and what to call a subclass. I take the naïve approach and treat everything with a single capital letter as a full class and everything with a secondary letter as a subclass. When training an algorithm on a basic class I fold all subclasses into that class.

The issue of imbalanced classes becomes evident as soon as we look at the distribution of classes in each dataset. As discussed in Chapter 2, the reliability of some ML algorithms can be compromised by imbalanced classes and has a bearing on clustering algorithms; hence the need for Chapter 6. But the reason for the imbalance also needs to be considered. In the context of the Main Belt, the nearer objects tend to be S-types, while the dimmer C-types tend to dominate the outer Main Belt. With magnitude limits typically around 19, this leads to a selection bias in favour of S-types. A similar bias will apply to the size of objects that can be viewed; whether size and class could be correlated is an open question. Higher-albedo objects are more likely to be observed for the same reason (although albedo is not handled explicitly here, it plays a part in some subclasses as well as the B/C separation). Additionally, some surveys deliberately set out to observe Trojans (T), Hungarias, and other families of asteroids preferentially. The MIT-Hawaii Near-Earth Object Spectroscopic Survey (mithneos) and Isaac Newton Telescope (int) studies target NEAs, which have a different class balance to MBAs. NEA spectra are therefore likely to have different class distributions to photometric surveys.

3.1 Reflectance spectra

3.1.1 SMASS

The Small Main Belt Asteroid Spectroscopic Survey II consists of 1447 asteroids, including the objects that determine the Bus-Binzel taxonomy, observed by the MDM Observatory at Kitt Peak. These data come from a public repository [1]. They form the bulk of the training sample of spectra used in this project and I refer to them frequently, either as 'SMASS2' or simply 'SMASS.' The data span the wavelength range 0.435-0.925 μm, and 1333 of the spectra were of sufficient range and quality for inclusion here.

The distribution of classes and subclasses is shown in figure 3.1. S-types are seen to greatly outnumber all other classes; even C-types, in which most of the mass of the Main Belt is included, are underrepresented here compared to S.

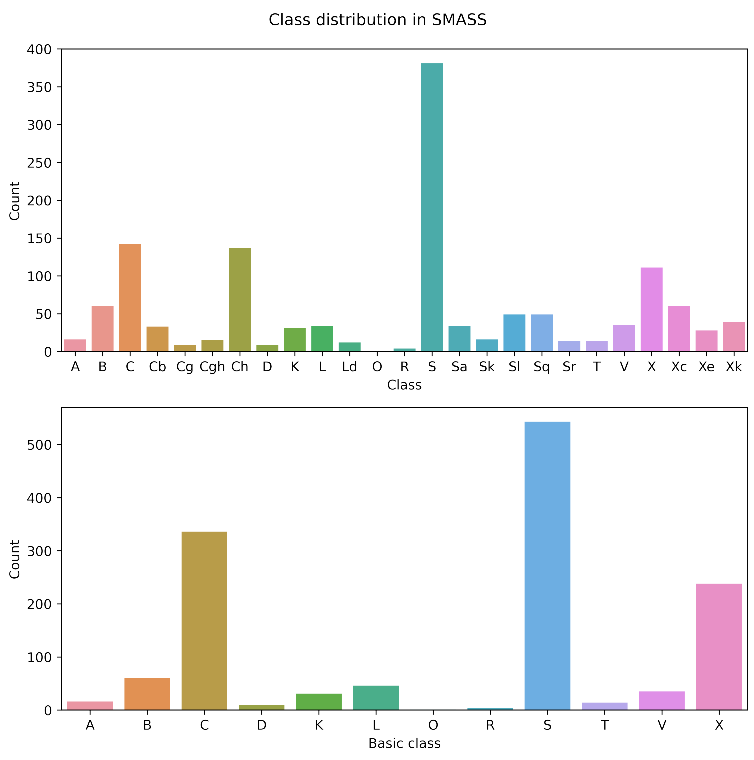


FIGURE 3.1: Distribution of full classes and 'basic' classes in SMASS2 survey. S-class dominates this sample even though C-type objects make up the greater mass of the Main Belt. Some classes are too sparsely populated to use in training an algorithm.

[1]`http://smass.mit.edu/smass.html`

3.1.2 MITHNEOS

The MIT-Hawaii Near-Earth Object Spectroscopic Survey (MITHNEOS) (Binzel et al., 2019) has published a dataset[2] in which observations come from the 0.8 – 2.5μm wavelength range from the SpeX instrument on the 3-meter NASA Infrared Telescope Facility (IRTF) on Mauna Kea, Hawaii as well as the FIRE (Folded-port InfraRed Echellette) instrument on MIT's 6.5m Baade Telescope at the Magellan Observatory in Las Campanas, Chile. Included are optical spectra compiled from work by Lazzarin et al. (2005) and others in the community (not specified by the authors). I have extracted 317 objects having visible-wavelength range spectra for use here. Of these, the Lazzarin et al. objects were observed by the Telescopio Nazionale Galileo (TNG) at La Palma with the Low-Resolution Spectrograph (LRS) in the 0.5–0.95 μm range (Lazzarin et al., 2005). There are also spectra in the range 0.435–0.925μm of unknown provenance, so I have divided the MITHNEOS data into two datasets: the 'long' one (188 spectra) and the 'short' one (317 spectra). All are classified according to the Bus-Binzel taxonomy. As these objects are Near Earth Asteroids (NEAs) the datasets include Q class, which is not observed in the Main Belt. The distribution of classes and subclasses is shown in Figure 3.2.

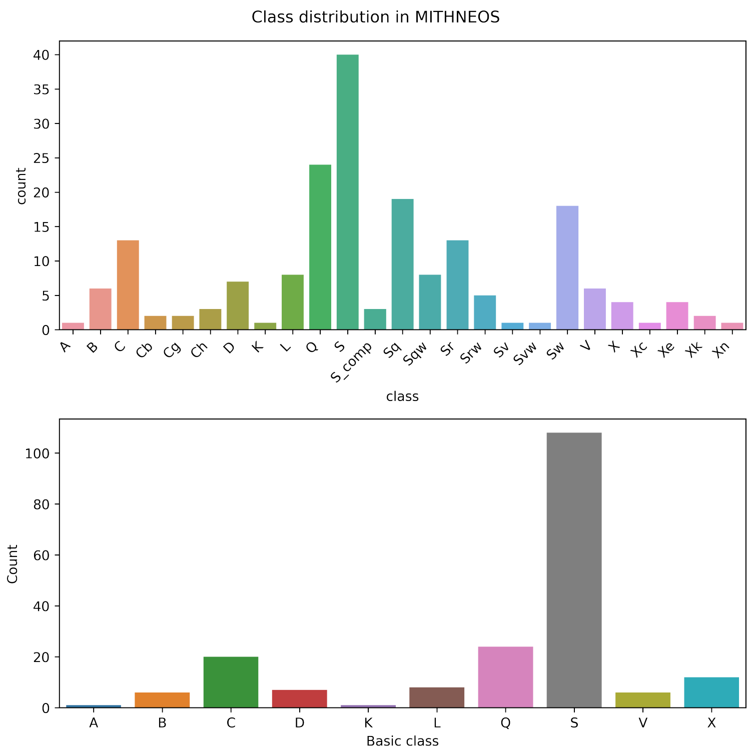


FIGURE 3.2: Distribution of full classes and 'basic' classes in MITHNEOS survey.

[2]available at `http://smass.mit.edu/minuspubs.html`

3.1.3 S3OS2

The Small Solar System Object Spectroscopic Survey (S3OS2) (Lazzaro et al., 2004) is comprised of 820 asteroids observed by the European Space Observatory's 1.52 m telescope at La Silla, Chile. Spectra are meant to cover the range $0.49 - 0.92\mu m$ but I was only able to get 719 objects with wavelength coverage from 0.5 - 0.9 μm for use here, of which 645 are unique to the dataset (others overlap with either SMASS or MITHNEOS). These are MBAs, including some intentional coverage of asteroid families, and all have been classified according to Bus-Binzel. Data are available at (Lazzaro & Florczak, 2006). The distribution of classes and subclasses is shown in figure 3.3.

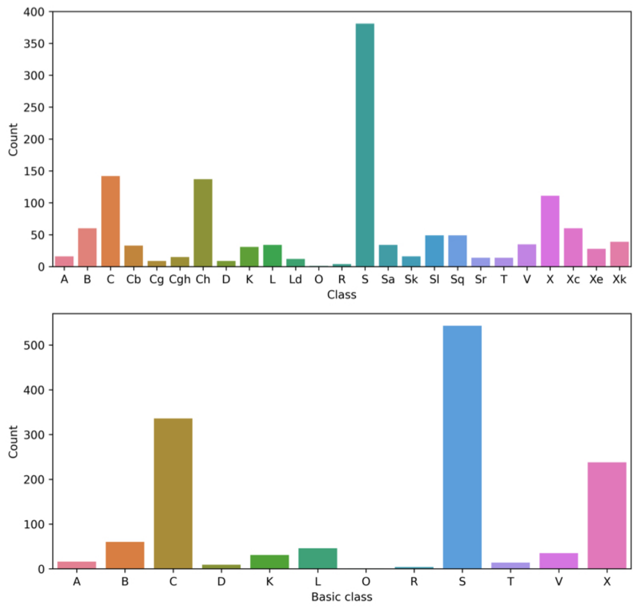


FIGURE 3.3: Distribution of full classes and 'basic' classes in S3OS2 survey.

3.1.4 INT

The dataset denoted INT refers to the published data of Popescu et al. (2019)[3] that represent NEA spectra taken from the Isaac Newton Telescope (INT) with the IDS spectrograph and then combined with NIR data from an (unpublished) SMASS NIR database. This is a small dataset in which only 35 objects were suitable for use in the 0.5 - 0.9 μm wavelength range. They have been classified according to the Bus-DeMeo taxonomy, which is equipped to handle features that appear only in the infrared (such as the 2 μm absorption band) even though the important new observations reported here are only in the optical. I therefore suggest that the

[3]available from http://cdsarc.u-strasbg.fr/viz-bin/qcat?J/A+A/627/A124

recorded class is not guaranteed to match the classification that has been assigned by the optical-only Bus-Binzel system. The distribution of classes and subclasses is shown in figure 3.4. The fraction of Q-types is notably large here.

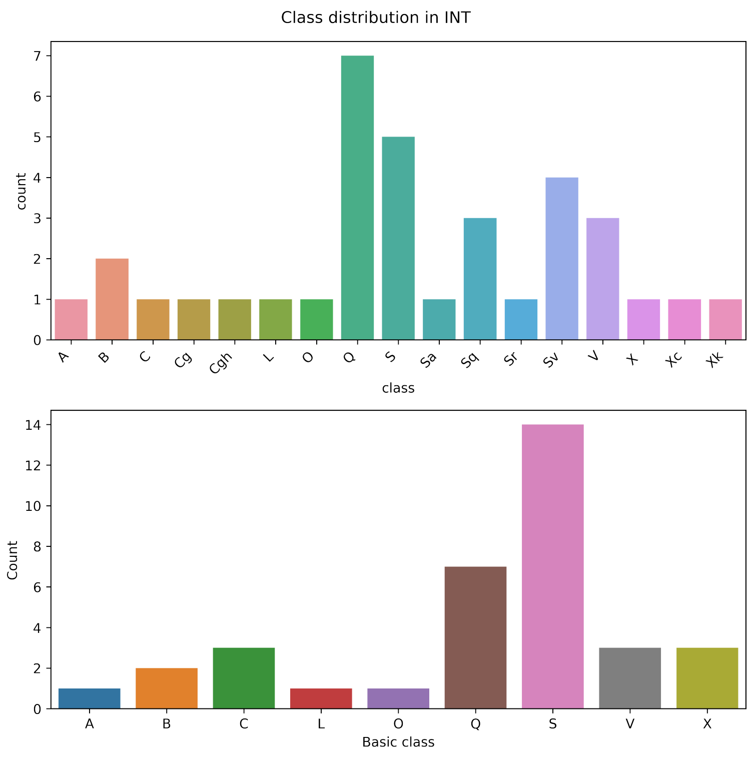


FIGURE 3.4: Distribution of full classes and 'basic' classes in INT survey.

3.2 Calibration of spectra at source

In this section I describe the approaches to calibration of spectra for each catalogue used in this work. As described by the authors associated with each dataset, the main potential sources of uncertainty for reflectance spectra are:

(a) Dispersion of some wavelengths outside the slit due to atmospheric refraction

(b) Loss of flux due to asteroid moving outside the slit

(c) Poor calibration of instrument

(d) Random colour changes, possibly due to cloud-cover-induced chromaticity

(e) Colour differences between standard stars used as solar analogues

(f) Lack of multiple observations for averaging

(g) Use of a mean extinction model (in SMASS2)

(h) High phase angle leading to phase reddening (if uncorrected)

(i) Different faces of asteroid showing different composition when rotating

The partial loss of flux at long or short wavelengths due to the first effect is probably the most problematic effect because it results in a change in slope from one end of a spectrum to another. All authors took steps to avoid this pitfall, but they took different steps.

The majority of spectra in this work come from SMASS2. The authors used the Mark III spectrograph with a slit 5 arcmins long and 4.5 arcsec wide, oriented North-South to ensure that any dispersion would fall within the slit and observing close to zenith to minimise atmospheric diffraction. They took Hg, Ar and Xe spectral line images with the same instrument regularly for wavelength calibration. Most spectra were averaged across multiple observations. Occasional random variations in slope up to ± 0.15 occurred, usually when the object was observed through cloud, suggesting that flux at the long or short end may have been lost due to diffraction outside the slit. In cases where slope variations exceeded 0.03 (in dimensionless units of relative reflectance) the spectrum was discarded and re-observed on a better night. The authors discovered that their use of a mean extinction model was a major source of error, but applying mainly to the near-UV range (used only in the early stages of this work) and limited to a maximum slope variation of ± 0.02 over the range 0.44 – 0.92 μm seen at the very highest airmasses (Bus & Binzel, 2002a).

Other authors took different approaches. Lazzaro et al. (2004) oriented a 5 arcsec slit East-West to handle asteroid motion because the ESO telescope did not permit automatic tracking. They trusted the wide slit to capture full dispersion because they observed close to zenith at low airmasses. They also recorded phase angles (median 13.5 degrees) in case of phase reddening. They did not average spectra taken on different nights, but selected the observation with lowest noise for each object.

In contrast, INT observations were performed with differential tracking plus manual adjustments, with a narrower 1.5 arcsec-wide slit oriented along the parallactic angle taking 3–5 observations per object, always close to zenith. At least one solar analogue close to the object was taken, with preference to the same solar analogues as SMASS2 when possible (Popescu, M. et al., 2019).

The MITHNEOS objects come from a variety of sources, many unpublished; some are associated with the SMASS2 survey, but information on observational conditions and data reduction is unavailable for the others, except for 24 objects where a 5 arcsec slit oriented on the parallactic was used (Lazzarin et al., 2005).

The heterogeneity of these data inevitably leads to fuzzy results. This is not necessarily a bad thing. Training and testing on this diverse set of spectra helps to characterise the robustness of the classification system when confronted with the variations that are expected in natural systems. Differences in surveys and techniques act as a proxy for the random errors that are otherwise problematic to incorporate within machine-learning models.

3.3 Preparation of spectra

The resolution at which reflectances were recorded varies from dataset to dataset and object to object. Some spectra also have gaps of varying sizes and spectral locations, necessitating some pre-processing to establish a consistent set of wavelengths at which to record reflectance. The Bus-Binzel taxonomy relies on data splined at 0.01 μm. I used a similar method but as the data I use has more detail available, I splined the spectra at intervals of 0.0025 μm using a the Scipy's (Virtanen et al., 2020) cubic spline interpolation function as described in (Qingkai Kong, 2020). I then divided all spectra by their value at 0.55 μm in accordance with the Bus-Binzel and Bus-DeMeo taxonomic convention. When gaps between datapoints are larger than a few hundredths of a micron, the splining process can make wrong interpolations. Spectra were checked by eye and interpolated values were adjusted manually as necessary.

When converting the spectra to their SDSS equivalents, I had to choose how to approximate what each SDSS filter would pick up (see figures 3.6 and 3.5 from 3.4). To obtain correspondences between SDSS and optical/NIR spectra, the DeMeo classification system (DeMeo & Carry, 2013) integrated spectra over the SDSS filter curves. They do not report how they handled end values where short-wavelength data were unavailable. My data are unsuitable for convolution with SDSS filters due to lack of values at both the short at long end of the spectra. I tried to approximate the flux expected from a filter by making pseudo-broadband in two different ways:

(a) Take the full-width-half-maximum value for the SDSS filter and subtract half of its value from the effective wavelength (band centre) to get the minimum for the band, add half of its value to the effective wavelength to get the maximum for the band, and then find the mean of the data between those two values.

(b) Take the exact value at (or near) the effective wavelength.

I tested classifiers using both methods; the results were virtually indistinguishable. In the thesis, I use the mean value over the filter. I note that SDSS z-band is inaccurate, especially the short spectrum dataset; SDSS z peaks at 0.9132 μm whilst my data stop at 0.9 μm.

Once the pseudo-broadband values were obtained, I divided the results by the spectrum value at (or near) the SDSS g-band band centre of 0.4774 μm so that the pseudo-broadband data are all relative to the SDSS g-band as in the Carvano et al. (2010) catalog.

3.4 SDSS photometry

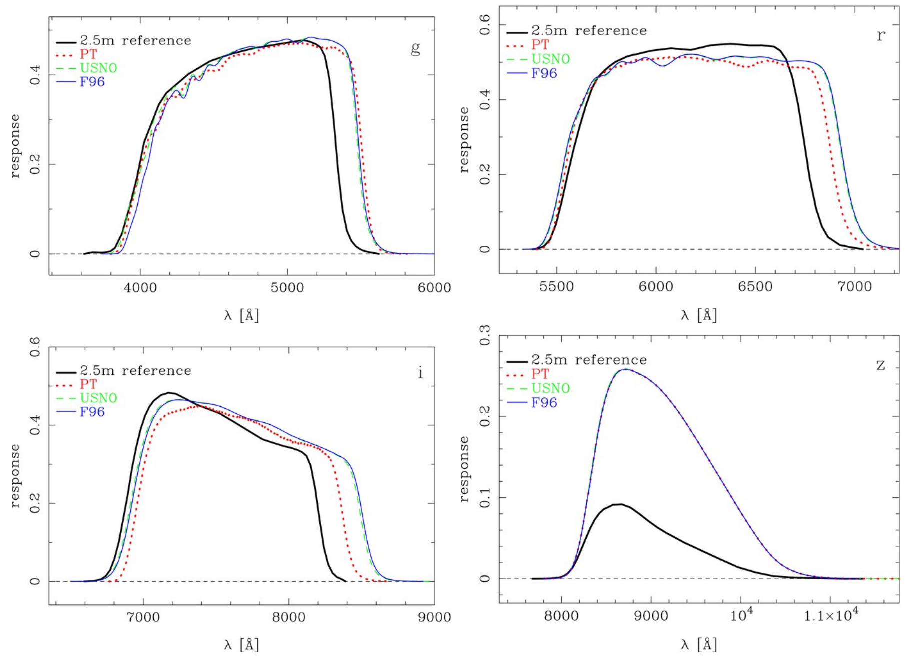


FIGURE 3.5: Filter response curves for the SDSS bands g, r, i, and z (Mamoru, 2010) used in this work. When converting spectra to pseudo-broadband, part of the SDSS g-band is lost (below 4350 angstroms for long spectra and below 5000 for short). The z-band is lost above 9250 angstroms in long spectra and above 9000 angstroms in short spectra.

The SDSS Survey recorded CCD photometry using a 2.5 m wide-angle optical telescope between 1998-2009. Data are recorded five filters u-g-r-i-z as shown in figure 3.6.

3.4.1 Carvano catalog

The classification system by Carvano et al. (2010) has been applied to the fourth release of the SDSS-MOC and is publicly available at Hasselmann et al. (2011). Of the 107,360 asteroids available I have selected for use in this work 20,389 with no flags for bad photometry. A large fraction of these objects were assigned multiple classes, so I also use a subset of the data consisting of 10487 objects of unambiguous class where classes are defined in SDSS r-i-z space according to the examples in figure 3.7. I refer to this subset as the Carvano 'pure' data.

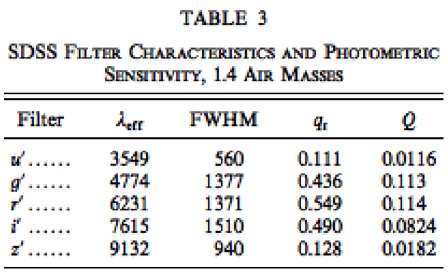


TABLE 3

SDSS Filter Characteristics and Photometric Sensitivity, 1.4 Air Masses

Filter	$\lambda_{\rm eff}$	FWHM	q_t	Q
u'......	3549	560	0.111	0.0116
g'......	4774	1377	0.436	0.113
r'......	6231	1371	0.549	0.114
i'......	7615	1510	0.490	0.0824
z'......	9132	940	0.128	0.0182

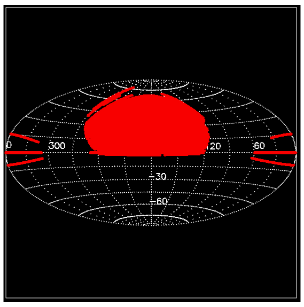


Figure 3.6: Left:SDSS waveband centres and full width half maximum values used for all SDSS data in this project. Waveband centres are used to calculate colour gradients in Carvano et al. (2010). To generate a pseudo-broadband value from a spectrum, I take the mean value across the waveband from minimum to maximum. Figure from (Carvano et al., 2010)

. Right: Sky coverage map showing a large fraction of the Northern hemisphere sky covered by the SDSS. Image from `https://classic.sdss.org/dr7/coverage/`

The data are expressed as solar corrected 'log reflectances' as described in section 1.3, which I have converted into reflectances by taking the antilog. The data are already normalized to unity in g band.

3.4.2 DeMeo et al.

The dataset from DeMeo & Carry (2013) (available from Ivezic (2008)) has been used to create and apply the DeMeo SDSS classification scheme. These data originate in a later data release than the Carvano et al. (2010) and contain more objects. There are 34148 asteroids, all classified according to the DeMeo system. The mean error on r-band dimensionless relative flux is 0.0378. Figure 3.9 shows the data in r-i-z space, but with K and L classes merged to match the SDSS L_p class.

3.4.3 Sergeyev data

In addition to the previous catalog asteroids from DeMeo & Carry (2013), data from Sergeyev, Alexey V. & Carry, Benoit (2021) comprise 1,040,690 SDSS objects that were collected from the 'trash' of the SDSS after filtering out stars, galaxies, and objects with spurious photometry. The authors then classified according to the DeMeo SDSS taxonomy. By their nature the data are messy; I have trimmed the according to magnitude error <= 0.05, of which 218515 are numbered and the rest are unidentified. The 'reflectance' error on these objects is on average 0.0195 in dimensionless units of relative reflectance. They also spill out over a wider wavelength range than the other two SDSS datasets.

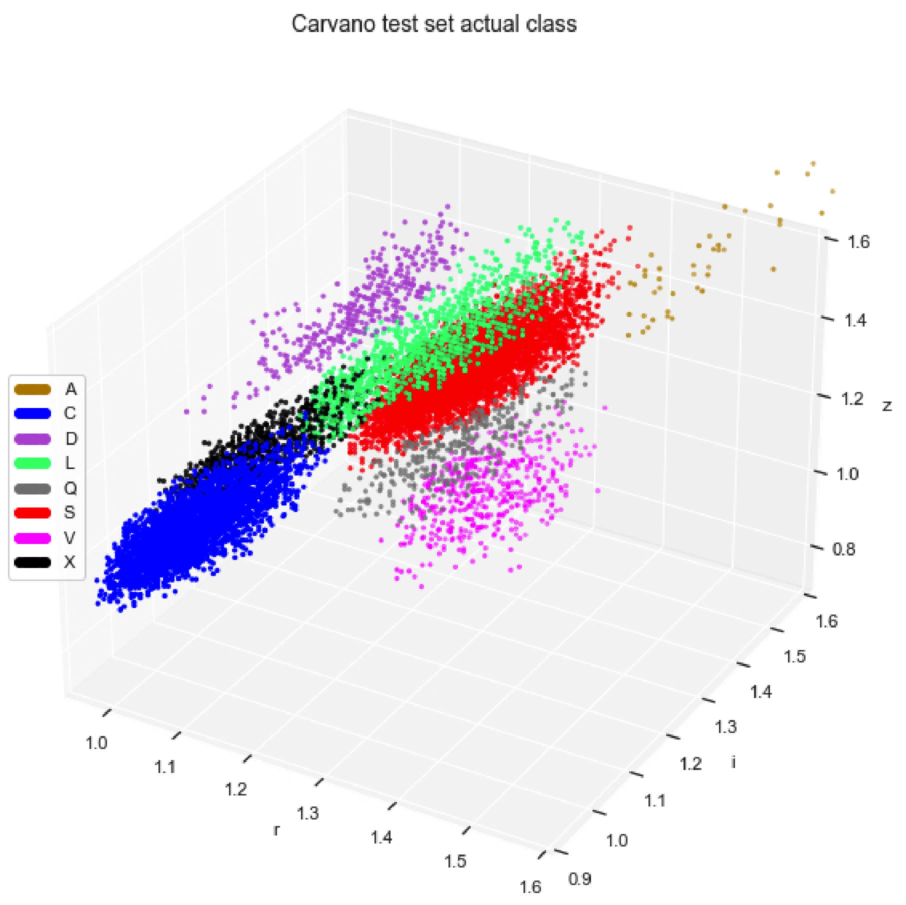


FIGURE 3.7: Recorded classes of Carvano et al. (2010) data in SDSS r-i-z space on which spectrum classifiers will be tested. Mixed classifications have been removed.

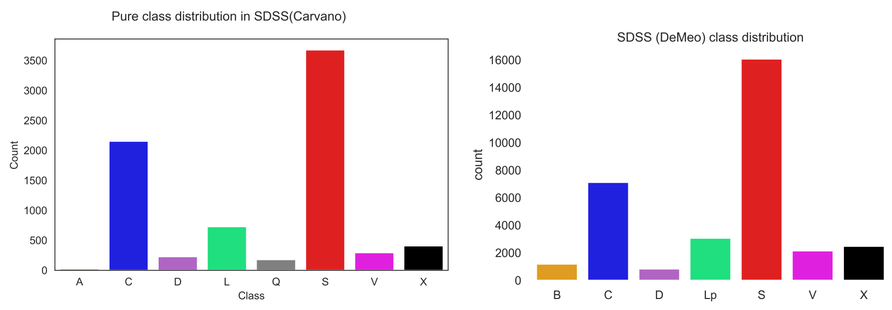


FIGURE 3.8: Class distribution for the two main SDSS datasets in the project as determined by their respective classification systems. Left: Carvano (pure), which consists only of unambiguous classes (i.e. $\geq 60\%$ probability of being in that class), and right: DeMeo.

.

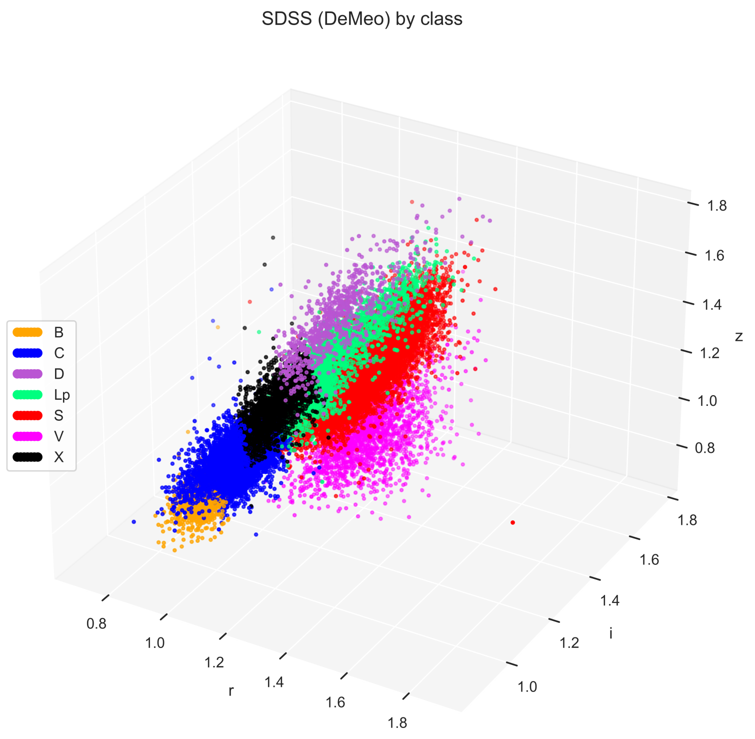
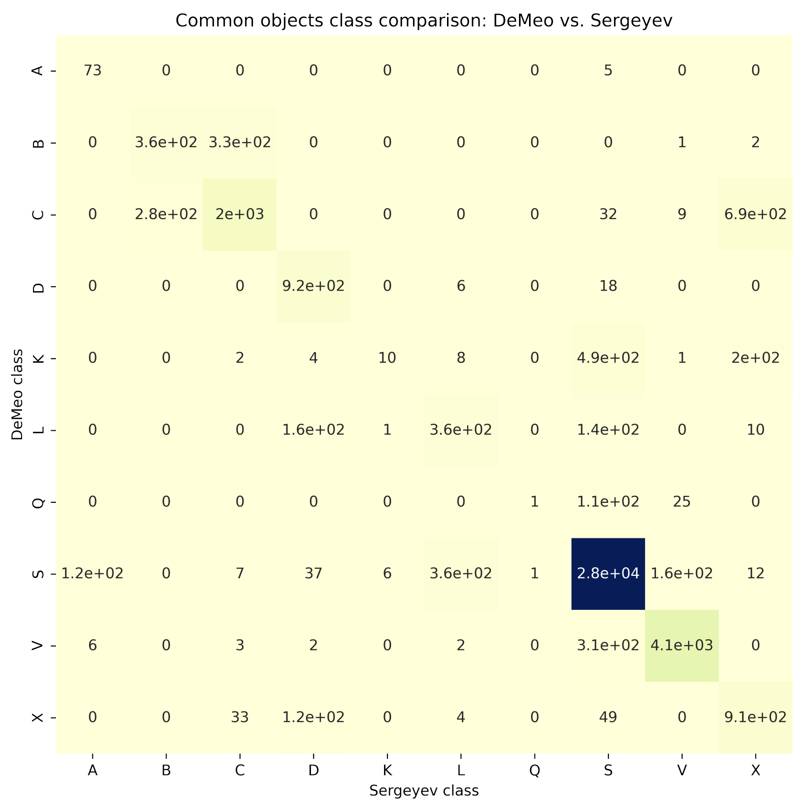
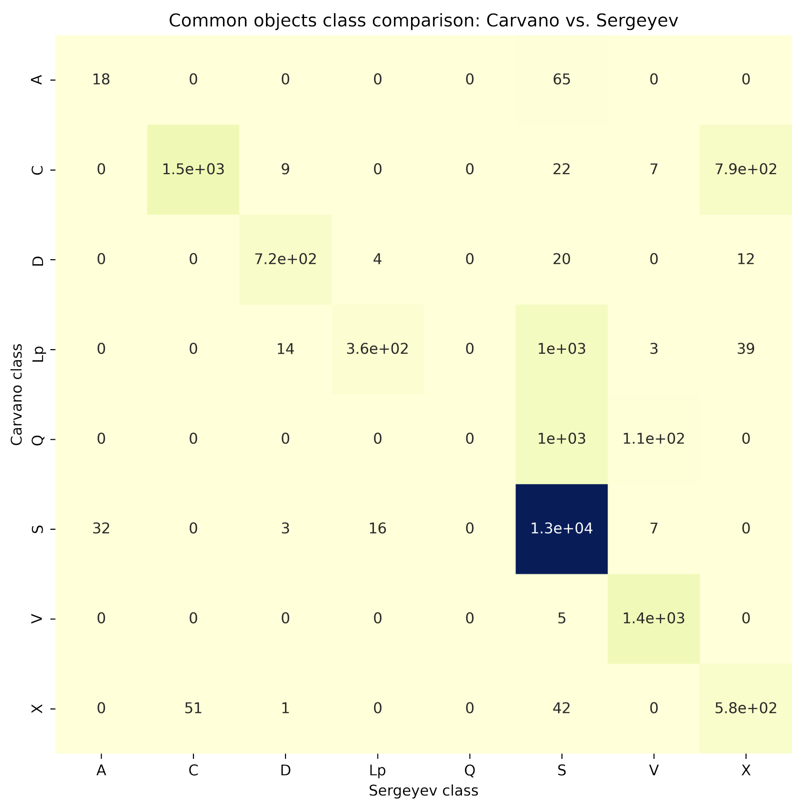


FIGURE 3.9: Data from (DeMeo & Carry, 2013) in r-i-z space. I have grouped K and L objects under the label Lp to match the Carvano et al. definition of L. There are no O or Q objects in these data.



DeMeo class \ Sergeyev class	A	B	C	D	K	L	Q	S	V	X
A	73	0	0	0	0	0	0	5	0	0
B	0	3.6e+02	3.3e+02	0	0	0	0	0	1	2
C	0	2.8e+02	2e+03	0	0	0	0	32	9	6.9e+02
D	0	0	0	9.2e+02	0	6	0	18	0	0
K	0	0	2	4	10	8	0	4.9e+02	1	2e+02
L	0	0	0	1.6e+02	1	3.6e+02	0	1.4e+02	0	10
Q	0	0	0	0	0	0	1	1.1e+02	25	0
S	1.2e+02	0	7	37	6	3.6e+02	1	2.8e+04	1.6e+02	12
V	6	0	3	2	0	2	0	3.1e+02	4.1e+03	0
X	0	0	33	1.2e+02	0	4	0	49	0	9.1e+02



Carvano class \ Sergeyev class	A	C	D	Lp	Q	S	V	X
A	18	0	0	0	0	65	0	0
C	0	1.5e+03	9	0	0	22	7	7.9e+02
D	0	0	7.2e+02	4	0	20	0	12
Lp	0	0	14	3.6e+02	0	1e+03	3	39
Q	0	0	0	0	0	1e+03	1.1e+02	0
S	32	0	3	16	0	1.3e+04	7	0
V	0	0	0	0	0	5	1.4e+03	0
X	0	51	1	0	0	42	0	5.8e+02

FIGURE 3.10: The DeMeo & Carry (2013) SDSS data have been classified according to the same system as the Sergeyev, Alexey V. & Carry, Benoit (2021) data. The confusion matrix on the left shows classifications for objects present in both catalogs, with 91% matching but an obvious blurring between B/C/X and between S and K/L/Q/V. On the right the Carvano classifications are compared with Sergeyev's.

Chapter 4

Supervised Learning with Optical Spectra

4.1 Introduction

As described in Chapter 1.3 the Bus-Binzel classification system relies on PCA to extract meaningful features from high-resolution spectra. In this chapter I will use PCA to explore variability in reflectance spectra across multiple small datasets (about 300 - 1300 objects each). The main objective of this chapter is to determine whether PCA is still the most effective basis for classification given the ability of machine learning to easily handle high-dimensional data. Along the way I will also explore the implications of fitting PCA on one dataset and applying it to another. By the end of this chapter, a baseline will have been established for the ability of the spectra to be classified with machine learning.

4.2 About the data

The following datasets as described in 3 are used in this section:

(a) SMASS2 0.435 – 0.925 μm (1333 objects)

(b) SMASS2 + MITHNEOS 0.435 – 0.925 μm (1521 objects)

(c) SMASS2 + MITHNEOS 0.50 – 0.90 μm (1650 objects)

(d) SMASS2 + MITHNEOS + S3OS2 - 0.50 – 0.90 μm (2369 objects)

(e) INT 0.5 - 0.9 μm (35 objects)

Tensions between datasets affect this work in various ways. To begin with, around 27% of asteroids common to both S3OS2 and SMASS have been assigned different

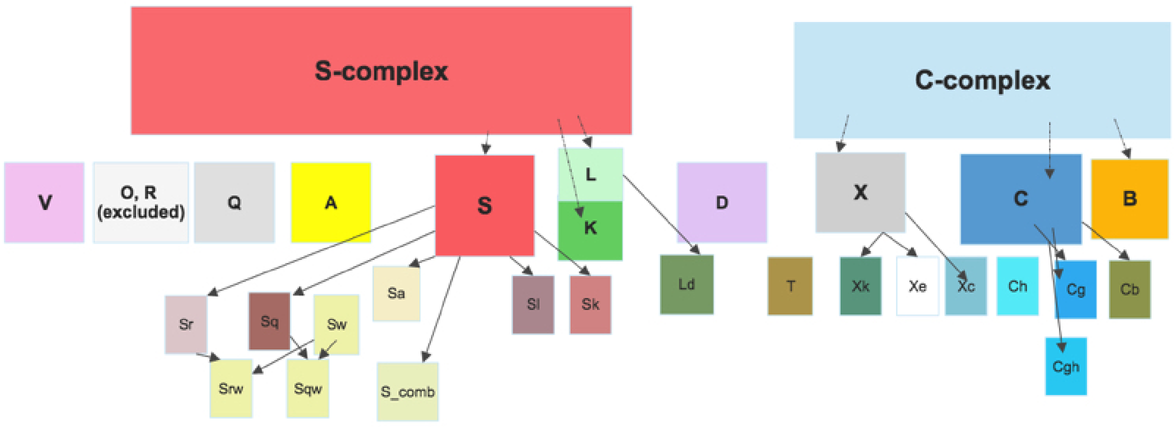


FIGURE 4.1: This chart roughly shows the hierarchy among spectrum classes. When combining subclasses to establish basic classes, objects below the second row are absorbed into their parent classes above. The exceptions are O, R, and T (excluded).

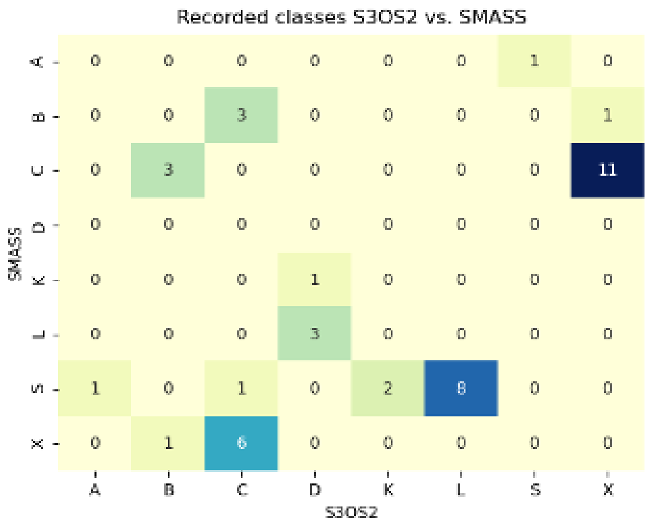


FIGURE 4.2: Of the objects present in both SMASS and S3OS2, 27% of them have different recorded classes. Here the classes are compared. B/C/X tend to be classified as one another, as do K/L/S. We are seeing that although subtle features are identifiable, there is some degree of reversion to the two fundamental complexes identified more than 50 years ago.

classes in the Bus-Binzel system (see 4.2). Generally, these disagreements lie within (not between) the two foundational groups, S-complex and C-complex. The S3OS2 authors ascribe this to the loss of the shorter wavelengths in S3SO2 as well as differences in user judgement in the SMASS classifier itself, but they also note that some spectra show up differently across surveys.

Examining these spectra reveals the latter to be true, e.g., 729 and 1139 (Figure 4.3). As will be seen throughout this work, the data do not necessarily behave as expected. In many of the other cases, however, there is little visible difference between the spectra, usually in the C/X complex (figure 4.4). Finally, figure 4.5

suggests that noise in S3OS2 leads to a different classification than in cleaner SMASS data taken of the same object.

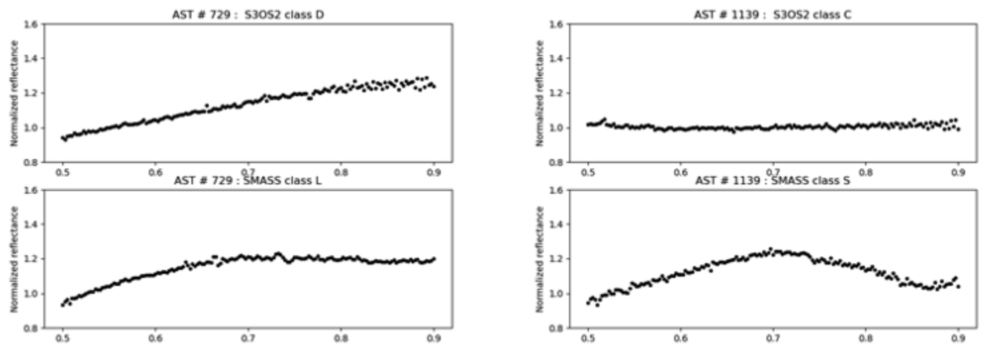


FIGURE 4.3: Entirely believable change in class between S3OS2 and SMASS for asteroids 729 and 1139.

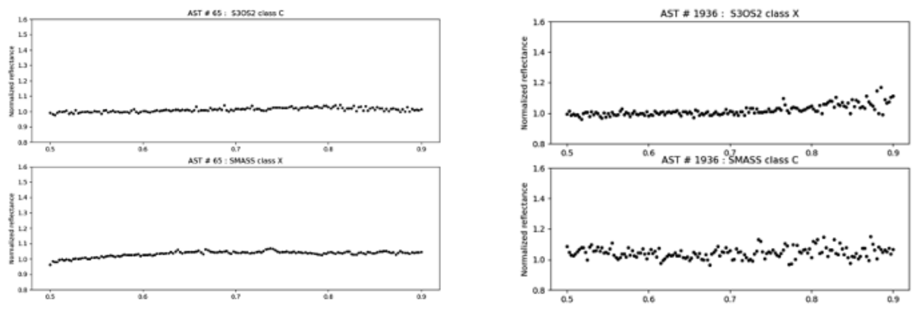


FIGURE 4.4: Specctra of asteroids 45 and 1936. Without albedo for an object there can be very little to separate C from X-types.

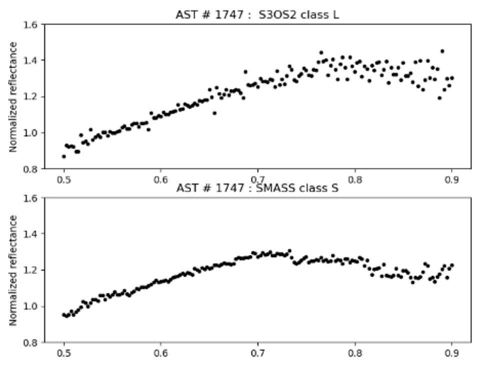


FIGURE 4.5: Object 1747 is class L in S3OS2 and S in SMASS2. Apparently random errors in S3SO2 scatter the spectrum. By tracing lower range of S3OS2 points, something very like the SMASS2 version would be achieved.

Differing distributions of classes within surveys presents some difficulties. SMASS contains no B or Q. S3OS2 contains no Q. INT contains no D or K. MITHNEOS contains no A or K. Because of this, classifiers trained on any individual dataset will automatically fail when presented with unfamiliar classes.

The range in dataset sizes complicates the training process. A classifier trained on the SMASS with 1333 objects has an advantage over one trained on the smaller S3OS2 or MITHNEOS data; as for INT, the data are too small to split into training and test sets, with several classes containing only one specimen. For these reasons I have not attempted to disentangle problems within a dataset from problems across datasets. With only 688 objects, the maximum success that S3OS2 can achieve on its own data is around 80%. Does this mean that the data are self-contradictory? Maybe, but in the absence of a full analysis of individual objects I will also suppose that the size of the data plays a part in this result.

Class imbalance is always a problem for machine learning, no matter the algorithm, because the most populous class will be most heavily weighted in determining decision boundaries and conversely the smaller classes are at risk of erasure. In order for any classifier (human or otherwise) to discriminate between very small, subtly different sub-classes, a degree of overfitting is unavoidable. Here, aside from the obvious dominance of S across the board, there are a profusion of sub-classes of S, each having few samples (see figures 3.1, 3.2,3.3, 3.4). I will show that in general, these sub-classes do not hold up well in a machine-learning context.

A final caveat concerns the classes Q and A. The INT data are heavily populated by Q, a class that is absent entirely from SMASS. Examination of plots in figure 4.6 suggests that Q is not well-defined, especially in S3OS2. The smallest class, A, has only 24 objects in the combined dataset $0.5 - 0.9$. Looking at the spectra in figure 4.7, it is difficult to conclude that they are all members of the same group. It is becoming obvious that even a 'basic' classification can be challenging in this wavelength regime when moving between surveys.

4.3 Method

To standardize the spectrum binning across surveys, all spectra were splined at 0.0025 μm intervals. In some cases, the survey's reported wavelength range is longer than what actually exists in individual files. Splining performed very poorly when extrapolating beyond the reported data. Therefore, any spectra that are shorter than the needed wavelength range by $> 0.01\mu m$ in either direction were discarded. After eyeball checks, any remaining spectra whose splines appeared to grossly misrepresent the data were corrected by hand. There are a total of 28 classes spanning the full breadth of data; however, some subclasses are only present in the combined set. Classes O and R were removed because they are too sparse

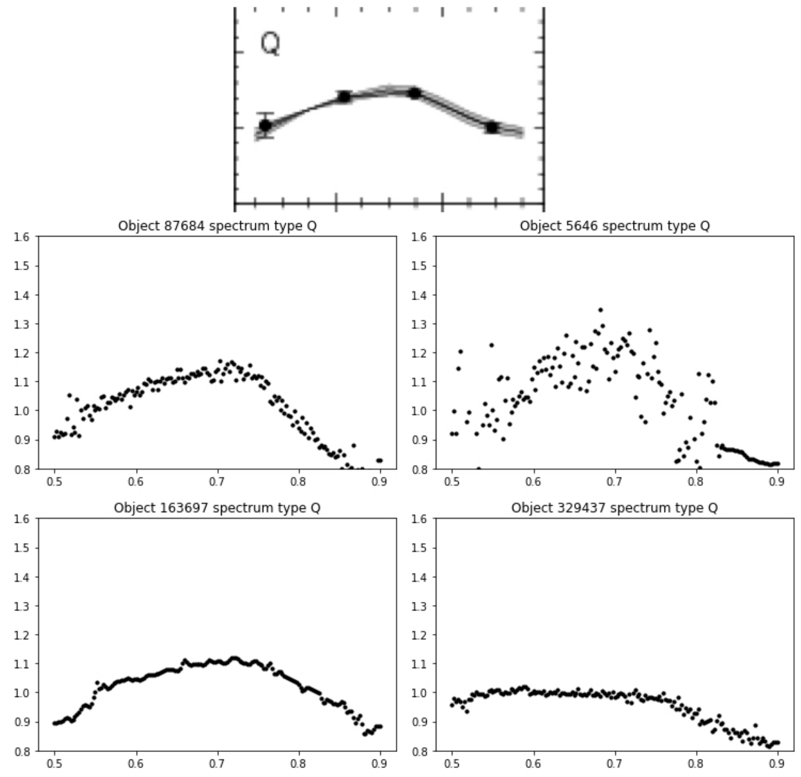


FIGURE 4.6: Top: Template for Q-type for SDSS from DeMeo & Carry (2013)) Bottom: Examples of Q-type objects that are not particularly consistent with one another.

to split for training and testing. A few subclasses (S comp, Svw, Xn) have been relabelled according to their basic classes for the same reason. Here, a basic class is designated by a single letter and includes the spectra originally labelled as such as well as data from any subclass under it; for example, 'Sl' would be designated 'S' as its basic class. The designation of a class as 'basic' is somewhat arbitrary at this point; for example, the classes K and L are understood to be part of the S complex but are not treated as subclasses in the SDSS classifiers. Conversely, SMASS class T is a small class not seen outside that survey, so here it is treated as a basic class in the early stages but discarded when the work moves on to consider the SDSS.

For each dataset, the following routine is executed:

(a) Spline, clean, scale with RobustScaler and split the data as described in 3.3

(b) Perform PCA

(c) Examine the fraction of variability captured by each principal component

(d) Visualise the classes according to first three principal components

(e) Visualise the PCA loadings

(f) Cross-validate and train an SVM to classify the validation set

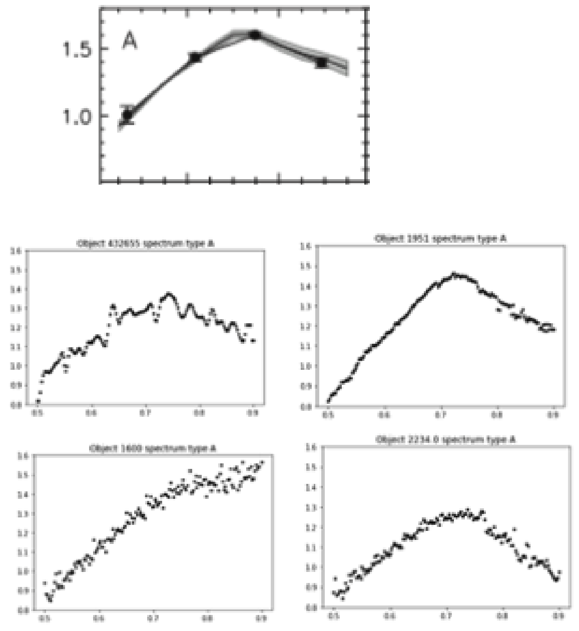


FIGURE 4.7: Template for A-type (DeMeo & Carry, 2013) Below: Examples of A-type objects that seem to have little in common with one another.

(g) Test on test set from the same data

(h) Test on test set from other datasets, if available in that wavelength range

(i) Produce confusion matrices and/or ROC-AUC plots

The same datasets are then re-processed as above, but without using PCA.

The SVM has been chosen because after trialling three other classifiers (random forest, MLP, and XGB) models, all methods performed within a few percentage points of one another in accuracy. Overall, optimally consistent results came from the support vector machine.

4.3.1 PCA preliminary findings

As described in 1.3, the SMASS2 classifier was designed based on splined SMASS data using the slope of the entire spectrum as a proxy for PC1 and taking PCA of residuals (Bus & Binzel, 2002b). I decided to consider what happens when new PCA is performed on different combinations of data and without explicitly calculating slope. I produced a series of heatmaps to visualise the eigenvalues, some of which are shown in figure 4.10.

In reading these plots it is important to recall that each principal component is calculated and plotted on the same scale. The first PC contains the majority of the dataset's variance, with each successive component representing a smaller share. Dramatic-looking features in a heatmap have impact if they occur in the first PCs but mean very little for higher-numbered PCs because the amount of variance that they represent is tiny; see figure 4.8 to understand how quickly the variance diminishes with additional PCs.

Heatmaps of loadings are very useful in flagging the presence of extreme outliers because anomalous values become very obvious. When combining datasets of different provenance, there is a concern that the principal components will change meaningfully. This is because each dataset will have a unique covariance matrix, leading to unique eigenvalues and unique PCs. Figure 4.10 offers some reassurance that the covariance matrices for SMASS, MITHNEOS, and S3OS2 are all reasonably similar. Each of these heatmaps has the same pattern of variance. For PC1 the correlation becomes negative shortward of 0.55 μm and positive longward, but is otherwise almost uniform. For PC2 we are seeing variance occur where some spectra fall off at higher (and lower) wavelengths, but others do not. These changes present as a smooth transition. The middle of the plot is relatively uniform because in this wavelength range there is little difference between the spectra on the PC2 axis. For PC3 some spectra are very strongly shaped just short (and just long) of the normalisation, and in addition some are very different above 0.875 μm. These broad characteristics are the same across all the datasets to be tested.

PCA also assists with visualisation of class relationships. Figure 4.9 shows each dataset in three-dimensional PC-space, labelled by class. Again, it is important to remember that the PC1 axis encodes most of the variance when viewing these plots.

The results of PCA screeplots (figure 4.8) were used in setting the number of principal components to be tested in cross-validation of classification models. These plots show the percentage of variance in the data captured by each of the first 25 PCs. The main difference in the screeplots lies in the greater number of PCs needed to express the variance in the full-class datasets, except for the combined SMASS/MITHNEOS/S3OS2 dataset, where the distribution of PCs is very similar for all classes vs. basic classes.

Because of the long tails of some of these distributions, tests were done on PCA up to 85 classes and 100% of the variance. Even so, improvements in classification were marginal. For this reason, the number of PCs was limited to the range 3-7 to be tuned to each individual dataset using cross-validation.

4.4 Results

4.4.1 Classifying with PCA and SVM

Tables 4.1 and 4.2 record scores for SVM models on all-class and basic-class models, respectively. The reason for progressing by dataset across the columns of each table is to take account of changes in class population one the one hand, and changes in wavelength range in the other. By testing SMASS + MITHNEOS at the longest-possible wavelength range and then at the shorter range, a baseline is established against which to measure the shorter (but more numerous) spectra from 0.5 – 0.9 μm. A small decrement in accuracy can be seen when moving to the latter model, which is attributable to wavelength limits rather than changes in class population. Conversely, when adding S3OS2 data to the combined model, changes in behaviour can be pinned to the new survey constraints (including class diversity) rather than a change in wavelength range.

Some feeling for the performance of the basic-class models vs. the all-class models can be gained from Figure 4.12, which shows the ROC-AUC results for each. The basic class model performs well, with all classes performing close to the top left

Test set	SMASS long	SMASS + MITH long	SMASS + MITH short	SMASS + MITH S3OS2
Self	0.69	0.672	0.636	0.621
SMASS	-	-	0.722	0.796
MITHNEOS	0.30	-	0.50	0.359
S3OS2	-	-	0.382	0.412
INT	-	-	0.540	0.628

Table 4.1: SVM models trained and tested on PCs of all classes. Models struggle to do well even when tested on their own data; the reasons that SMASS is the most successful can be attributed to its large sample size relative to the others.

Test set	SMASS long	SMASS + MITH long	SMASS + MITH short	SMASS + MITH S3OS2
Self	0.876	0.855	0.812	0.822
SMASS	-	-	0.890	0.889
MITHNEOS	0.689	-	0.800	0.736
S3OS2	-	-	0.730	0.794
INT	-	-	0.762	0.78

Table 4.2: SVM models trained and tested on PCs of basic classes. The outcome here is much better than for full classes. The final column represents the best results achievable when training on principal components.

of the plot; the full class model shows a much lower rate of true positives to false positives as evidenced by lines close to (and below) the diagonal midline.

In Table 4.1, at first glance some results look worse than they are. For example, testing the SMASS model on MITHNEOS data yields only 30% accuracy for full classes; however, in examining the confusion matrix (Figure 4.13) all of the Q-type objects are mis-classified for the simple reason that the SMASS model has not been trained on Q-types. For eleven Q-types the SMASS model predicts 'Sq' (reasonably) and for ten 'V', with two predictions of 'Sr' and a single prediction of 'A'. This 'misclassification' has some value; it helps to illuminate the relationship between Q and its neighbouring classes in PC-space. Similarly, there are three sub-types of 'S' in MITHNEOS that do not exist for the SMASS model; these are also automatically misclassified. In general, the profusion of subclasses of 'S' seem to be testing poorly, and about half of the misclassifications disappear in the basic class version due to all S-types being treated as a single group.

The problem of missing subclasses can be handled by combining datasets to allow the model to train on a wider variety of subclasses. For shorter wavelength-range SMASS + MITHNEOS 0.5 – 0.9 μm model, the full-class model predicts six of eight Q-types correctly in the MITHNEOS test set and four of seven correctly in INT.

There is some blurring between B, C and X visible in the basic classes that is not corrected with the addition of S3OS2 data (Figure 4.11). Overall, though, the fully combined short dataset tests well on each of its components and on INT at the basic class level, averaging around 80% correct across 11 classes. The final column of 4.2 then becomes the best-case scenario for classification with available spectra using PCA.

Finally, dimensionality reduction techniques such as PCA invariably involve information loss. It could be argued that, if some small fraction of variance is contained in the higher PCs (Figure 4.8), then including all variance in the training set would permit (theoretically) perfect classification. To test this, the combined SMASS2 + MITHNEOS 0.435 – 0.925 dataset is used. Classification improves only marginally when up to 85 principal components are fed into the SVM. More PCs are not the answer.

4.4.2 Classifying with SVM on full dimensions

Full results of all datasets with an SVM trained on all dimensions are shown in Tables 4.3 and 4.4. Training without PCA produces slightly better results in most combinations of training and test sets, and in the combined dataset it performs better across the board as summarised in Table 4.5.

Test set	SMASS long	SMASS + MITH long	SMASS + MITH short	SMASS + MITH S3OS2
Self	0.774	0.708	0.666	0.653
SMASS	-	-	0.838	0.872
MITHNEOS	0.295	-	0.609	0.500
S3OS2	-	-	0.599	0.441
INT	-	-	0.47§	0.780

Table 4.3: SVM models trained and tested on all dimensions of all classes. Poor results here are due in part to surveys not containing a complete set of classes in common.

Test set	SMASS long	SMASS + MITH long	SMASS + MITH short	SMASS + MITH S3OS2
Self	0.936	0.898	0.836	0.850
SMASS	-	-	0.940	0.938
MITHNEOS	0.622	-	0.875	0.785
S3OS2	-	-	0.736	0.837
INT	-	-	0.853	0.824

Table 4.4: SVM models trained and tested on all dimensions of basic classes. Results are an improvement on the all-class version because subclasses have been lumped together, making the classifier's job much easier.

Test set	PCA all	All	PCA basic	Basic
Self	0.621	0.653	0.822	0.850
SMASS	0.796	0.872	0.889	0.938
MITHNEOS	0.359	0.5	0.738	0.785
S3OS2	0.412	0.441	0.794	0.837
INT	0.628	0.780	0.798	0.824

Table 4.5: Comparison of ccuracy results for SVM models trained on SMASS + MITHNEOS + S3OS2 with and without PCA. The model trained on full spectra improves on the PCA model by $3-5\%$ depending on the test set.

4.5 Discussion

4.5.1 The value of PCA

The Bus-Binzel classification system is built on the use of PCA, predicated by the same method in the Tholen (1984) system. PCA is extremely helpful in breaking down high-dimensional data to a low-dimensional, visual representation, but some weaknesses also exist. If a new spectrum is observed with features that vary dramatically from the dataset on which PCA was performed, when it is transformed into PC-space the unique variance of its features is at risk of being washed out rather than enhanced. The effect has not been observed here, but it remains a risk for new data. Additionally, the use of PCA should be reconsidered in machine-learning classification, because results here indicate that it slightly disadvantages the classifier. Although I have only reported results for SVM, this point holds for random forest and multilayer perceptron models as well. Skipping PCA in modelling also offers simplicity. New data can be immediately classified using a non-PCA model without first transforming to the PC space of the model. The model itself can be updated by retraining on new data without need to change the PCA.

4.5.2 Role of uncertainties in spectra

I have not been able to find a straightforward way to incorporate errors on flux into the training and testing of models. In the absence of errors, I have sought to use diversity of sources and cross-validation in hyperparameter training to mitigate overfitting due to . In Section 3.2 I claimed that using spectra from multiple sources is beneficial because it implicitly introduces uncertainties, and the fact that prediction accuracy in this chapter generally falls when new data are added seems to support this assertion. Also, the cross-validation performed in setting hyperparameters for SVM has created some buffer in the models against random errors in new data.

As explained in Section 3.2, the literature suggests that variations in spectral slope are likely to dominate uncertainties associated with these data. The physical causes of slope change (e.g., phase reddening, rotation) are presumed random. Therefore I would expect the SVM models to behave with patterns of prediction similar to those seen in the confusion matrices in this chapter; that is, they would fail or succeed in the same ways, albeit with some overall deterioration of accuracy to be expected. However, errors associated with spectral calibration carry risk of systematic bias in the slope of all spectra. For this there can be no cure, because the SMASS classification system and its successors (including the most recent by Mahlke, M. et al. (2022)) all rely on the spectra and, especially, their slope (as a proxy for the first principal component). If the taxonomy itself were founded on biased data, then the meaning of all results would be greatly diminished.

4.5.3 Classes and subclasses according to machine learning

Detailed spectra may offer insight into asteroid mineralogy, and even in cases where mineralogy is ambiguous, high-quality data are always preferable to simple colour information. Any analysis of large-scale colour surveys implicitly depends on solid classifications at spectrum level for the robustness of its own taxonomic assignments. The problem is that with the sample of spectroscopically observed objects still relatively small (few thousands), each new survey yields new subclasses and injects new variability into the population. Testing classes with an SVM shows that many subclasses are too small and too similar to be picked apart by machine learning. For the rest, most are well-defined, at least at the basic type level, with the exception of B, K, and Q-types. B lives at one end of the C class in PC space and tends to be lumped with C; this is not necessarily the fault of missing albedo, because B-types generally have a negative slope. It is possible that more samples of B would resolve this problem. I have already stated that Q-types do not appear consistent on inspection, so it is not surprising that the SVM struggles to classify them; as for K, it is not present in the largest survey (SMASS) and is most often mistaken for the large S-class despite also being close to L in PC space. As a class, K is barely distinguishable from L at spectrum level, yet its presence as a major class in the DeMeo system means that SDSS broadband data can be readily designated as K. We will find out later if this designation is tenable.

For asteroids, features that distinguish certain classes (such as the 2 μm absorption band) may appear only in the infrared; this is the whole point of the Bus-DeMeo taxonomy. Within the visible regime, though, results here indicate that small changes in wavelength range of surveys are not too problematic in matching classes, and the range 0.5 - 0.9 μm offers the best variety and quantity of spectra with minimal loss of information.

4.6 Conclusion

In this chapter I have shown that a support vector machine can predict basic classes within each of the SMASS, MITHNEOS, and S3OS2 datasets to around 90% accuracy, but when models are transferred from one dataset to another performance can deteriorate dramatically. In order to move forward, I have combined all the available data into a single model from 0.5 – 0.9 μm, which achieves 85% accuracy in classification on a test set without resorting to PCA. Recalling that there could be around a 27% difference in classifying the same objects in SMASS as in S3OS2 (see 4.2) this can be treated as a good performance, and the SVM has no trouble with high dimensionality. I have trained a second model on data from SMASS and MITHNEOS in order to increase the wavelength range from 0.435 – 0.925 μm whilst retaining all the important classes. The long model achieves

90% classification accuracy, but because it is based on a smaller, less diverse population, performance will almost certainly degrade more on new data than in the short model.

Despite these encouraging numbers, I note that some of the major types are difficult to distinguish from neighbouring classes despite the high level of detail available in their spectra: namely, B, K, L, and Q. Generally, these classes can be differentiated by models tested *within* their original datasets, but not by models trained on all available data. While there is some reason to believe that the problem is exacerbated by the absence of B, K, and Q classes in the large SMASS dataset that makes up much of the data used for training, the fact that L-types are also compromised indicates that some more fundamental problem exists in the taxonomy. The implication is that models need to overfit in order to find these classes. This is important because it shows that some classes do not have a strong identity even given the best possible resolution.

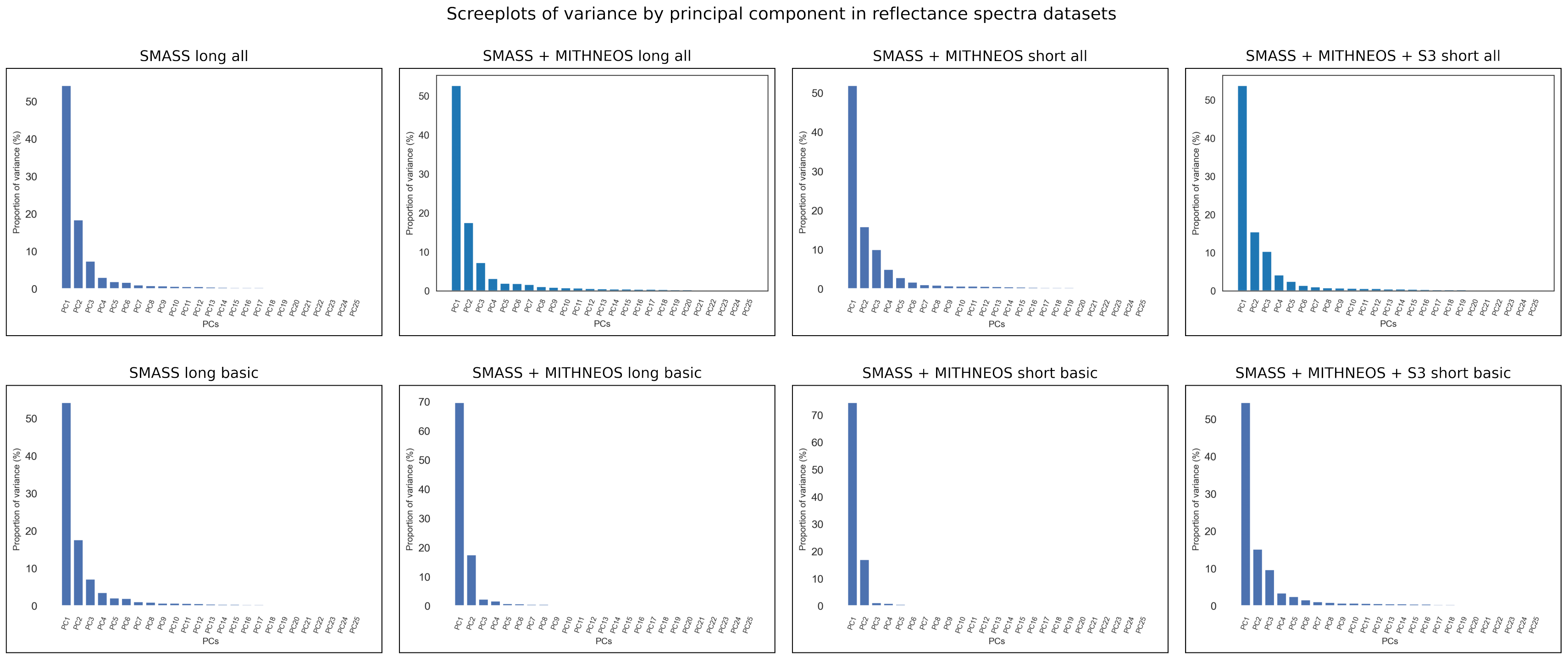


FIGURE 4.8: Screeplots showing percentage of variance in principal components of classes for (left to right): SMASS $0.435 - 0.925\mu m$, SMASS + MITHNEOS $0.435 - 0.925\mu m$, SMASS + MITHNEOS $0.5 - 0.9\mu m$, SMASS + MITHNEOS + S3 $0.5 - 0.9\mu m$. Top row shows all classes, bottom row shows respective basic classes. Code from `https://www.reneshbedre.com/blog/principal-component-analysis.html`

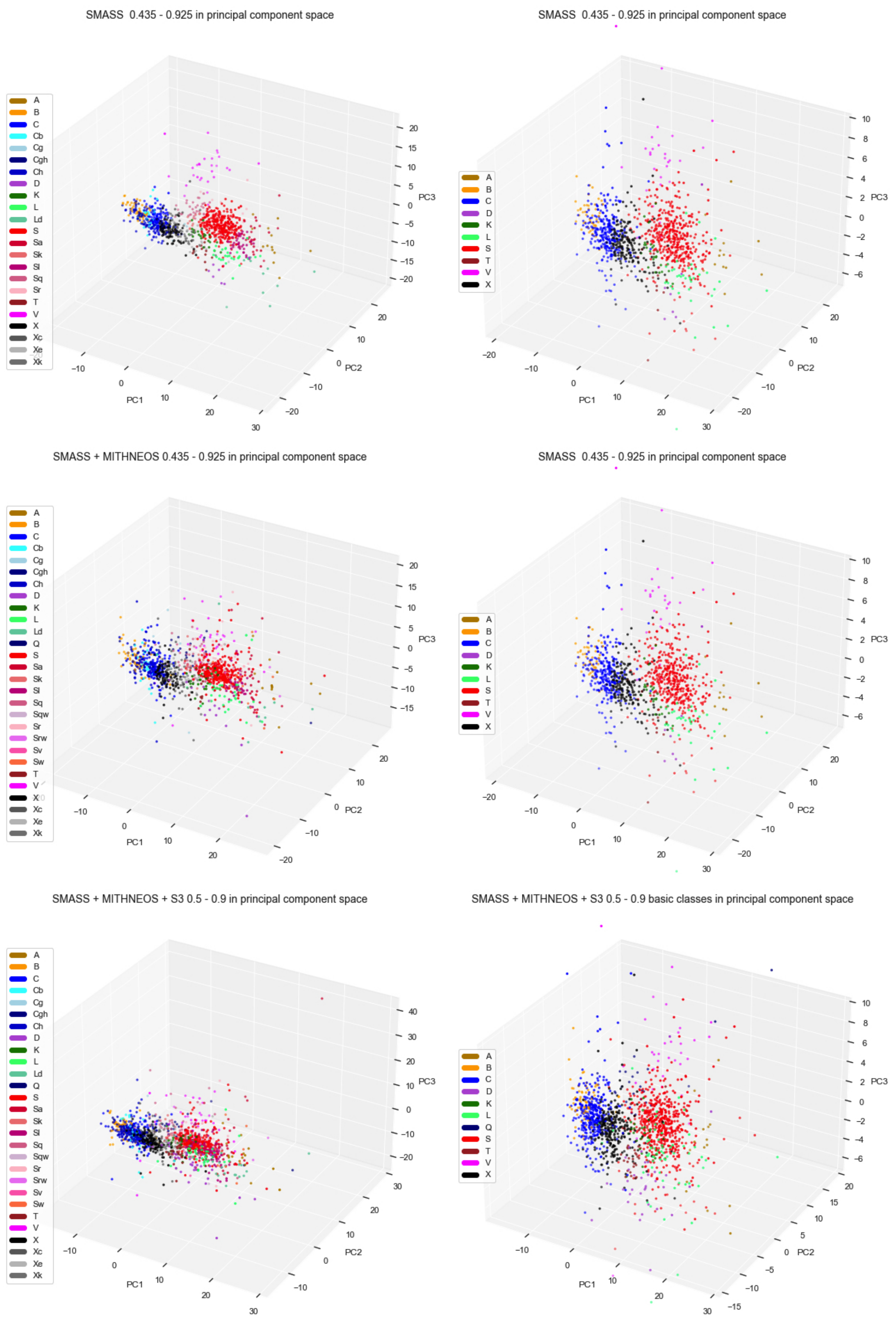


Figure 4.9: First three PCs for SMASS, SMASS + MITHNEOS long, SMASS + MITHNEOS + S3. Left: all classes, right: basic classes. For all its uses, PCA necessarily shifts its axes as the data are changed, which we can see in the different distributions of the data as more objects are added successively to the plot.

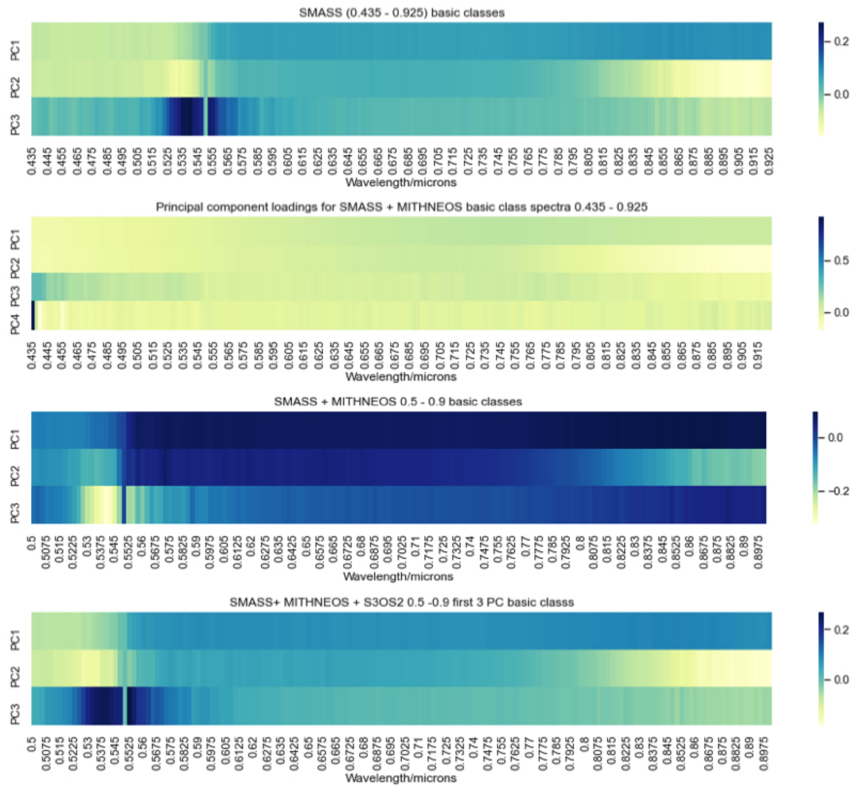
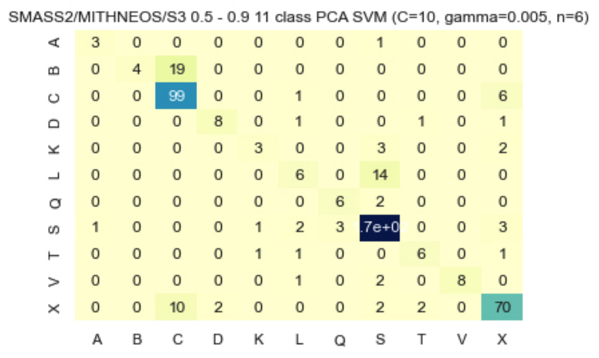


FIGURE 4.10: Eigenvalues for PCA expressed as a heatmap of 'loadings' (weights); the values of each unit on the x-axis represent the importance of each wavelength in setting the axis that maximises variance. Top to bottom: SMASS 0.435 – 0.925 μ m, SMASS + MITHNEOS 0.435 – 0.925 μm, SMASS + MITHNEOS 0.5 – 0.9 μm, SMASS + MITHNEOS + S3OS2 0.5 – 0.9 μm All are basic classes. Note that colour schemes are independent for each principal component as well as each subplot. The greatest variance is always indicated by the greatest deviation from zero. Qualitatively the features expressed in the eigenvalues are consistent throughout.



	A	B	C	D	K	L	Q	S	T	V	X
A	3	0	0	0	0	0	0	1	0	0	0
B	0	4	19	0	0	0	0	0	0	0	0
C	0	0	99	0	0	1	0	0	0	0	6
D	0	0	0	8	0	1	0	0	1	0	1
K	0	0	0	0	3	0	0	3	0	0	2
L	0	0	0	0	0	6	0	14	0	0	0
Q	0	0	0	0	0	0	6	2	0	0	0
S	1	0	0	0	1	2	3	.7e+0	0	0	3
T	0	0	0	0	1	1	0	0	6	0	1
V	0	0	0	0	0	1	0	2	0	8	0
X	0	0	10	2	0	0	0	2	2	0	70

FIGURE 4.11: Confusion matrix of all three datasets combined in the range 0.5 - 0.9 μm and tested on the combined test set. A problem remains in defining B, K, and L-types. The two most populous classes, S and C, tend to 'pull in' members of classes that are similar to them.

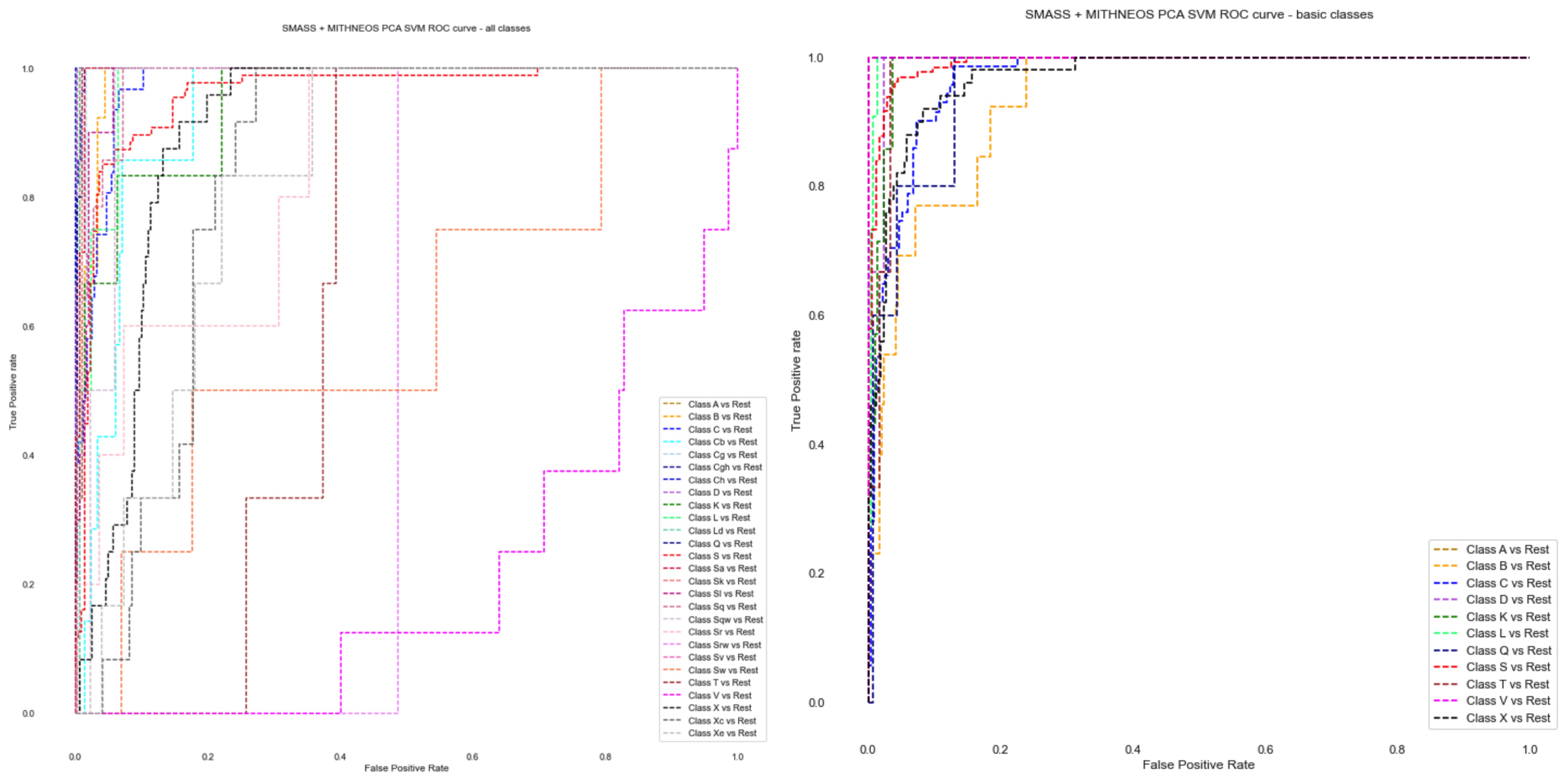


FIGURE 4.12: Example ROC curves for SMASS2 + MITHNEOS models. Here, the basic class model is clearly the more effective because lines for most classes stay close to the top left of the box.

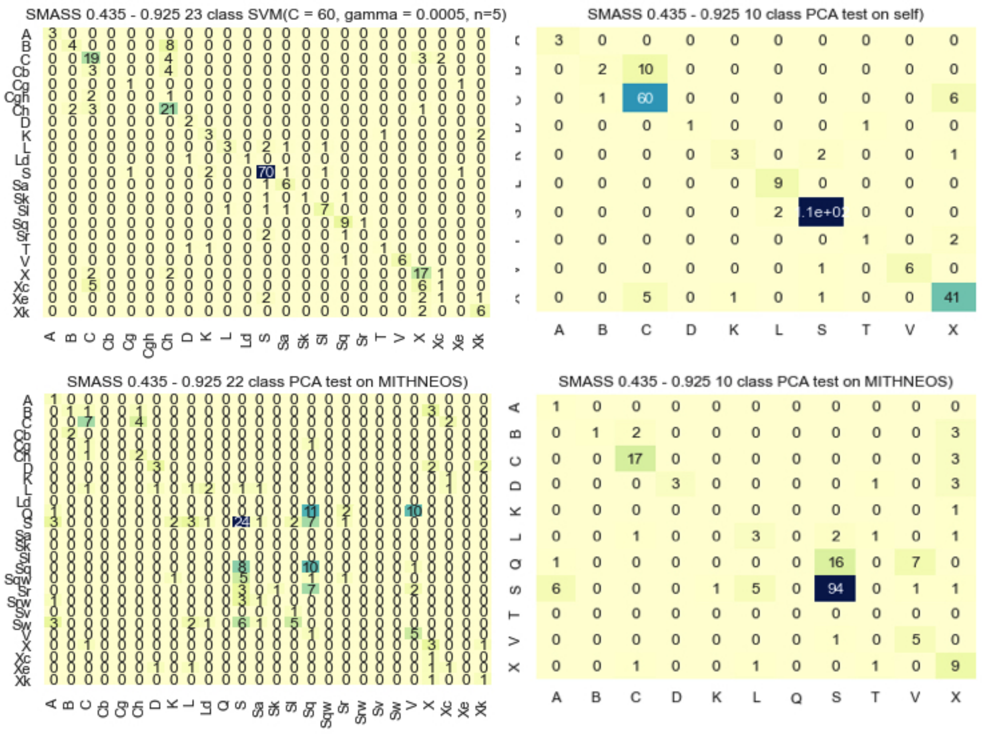


FIGURE 4.13: Confusion matrices for PCA/SVM model trained on SMASS2 0.435 - 0.925 μm and tested on (top): self all classes, self basic classes; (bottom): MITHNEOS all classes, MITHNEOS basic classes.

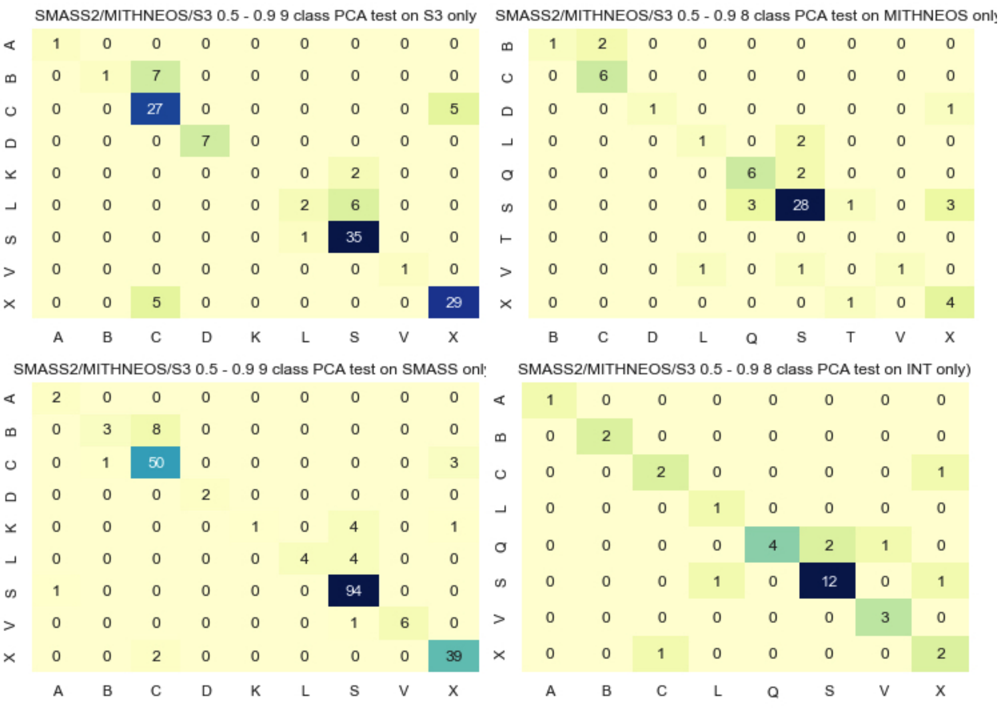


FIGURE 4.14: Confusion matrices for a model trained on PCA of combined SMASS2 + MITHNEOS + S3 0.5 - 0.9 and tested on each individual training set (top: S3 left, MITHNEOS right; bottom: SMASS left, INT right).

Chapter 5

Pseudo-broadband classification tests on SDSS

5.1 Introduction

This chapter addresses the question: what happens to the integrity of each asteroid type when moving from spectra to SDSS broadband photometry? There are two impacts of this shift, one being felt at the level of spectra converted to pseudo-broadband and a second when comparing the result to SDSS recorded classes, for which there are three (partially overlapping) datasets and two separate (but related) classification systems. To understand the shift better I work with a subset of the data having both spectra and broadband values, hereafter known as the common objects.

Both the Carvano et al. (2010) and DeMeo & Carry (2013) SDSS classification systems were designed with the benefit of expertise in interpreting asteroid spectra, a process that (for better or worse) carries biases. When I began this work, I hoped to triangulate on these systems by letting the spectra speak for themselves, so that for SDSS it would be possible to train on pseudo-broadband from spectra and predict on broadband to achieve a more objective understanding of class boundaries. Over time it became clear that the process is less than straightforward, largely due to apparent differences in scaling between SDSS and spectra. This work has ended up being more about what cannot be done than what can be done.

In this chapter I attempt to separate the problems with low-resolution data in general from the problems with matching SDSS broadband to spectrum pseudo-broadband. The short (SMASS + MITHNEOS + S3 + INT 0.5 - 0.9 μm) and long (SMASS + MITHNEOS 0.435 - 0.925 μm) datasets described in 4 are converted to pseudo-broadband and a classifier is trained on each of them. I compare the outcome to the results obtained with full spectra to show which classes are the most difficult to separate at this much lower resolution. Then the model is tested on three different SDSS datasets. The difficulties that arise are addressed using the data themselves in two ways: 1) calculation of conversion factors between pseudo-broadband and each of the SDSS datasets, and 2) training a model to 'recognise' the correct spectrum class from SDSS broadband values based on objects having both spectra and SDSS data. The improved results are then examined to deconstruct failures.

5.1.1 Method

The short and long combined datasets are used to create pseudo-broadband filter data for comparison with SDSS-MOC objects.

Spectra were prepared as pseudo-broadband to match SDSS broadband reflectance data from the literature. I did some prior work that approximated u band by using the value of the spectra at 0.435 μm, but the extra dimension did not offer a meaningful improvement for classification even within spectrum datasets, so here I use only g, r, i, z (where g is simply unity in all cases). This enables me to plot in three dimensions over r-i-z.

Short spectra from 0.5 – 0.9 μm were normalized to unity at 0.5 μm and long spectra from 0.435-0.925 μm were normalized to unity at 0.4775 μm, as close as possible to SDSS g-band centre of 0.4774 μm. For each dataset, r and i values were obtained by taking the mean wavelength over the full-width half-maximum of the SDSS bandpass. The z values were obtained by taking the value at 0.9 μm for short data and the mean value immediately around the central wavelength for z band (0.9132 μm) for long data. A bias is therefore introduced to the short data because the g-band is shifted to higher wavelength and z-band is set to slightly lower wavelength than the SDSS z band centre. For positively-sloped spectra (which the overwhelming fraction of the sample are) these wavelength limitations will compress the range of fluxes at both ends, with a greater effect on the strongly-sloped classes (e.g., D and the S-complex) than on the flatter classes (B, C, X).

For each dataset, an SVM was trained and tested on the spectrum pseudo-broadband data and the model saved. Each model was tested on each of three SDSS datasets: Carvano, DeMeo, and Sergeyev, each prepared as described in 3.4. Because the Carvano system has no B and folds K into L, separate models are trained for the DeMeo and Carvano data.

Proceeding on the assumption that pseudo-broadband is more precise than SDSS, I attempt to bring the SDSS data in line with the spectrum data to improve classification accuracy.

The first method (bridge model) takes a machine-learning approach. The bridge model involves retraining the SVM to 'recognise' SDSS data associated with a given class spectrum class in the hope that it can become a 'bridge' between spectra and SDSS. A major assumption is that the difference in SDSS data vs spectrum data for the same objects follows a consistent pattern that can be learned. The problem is challenging for machine learning due to the small number of common observations and the imbalance in class numbers.

(a) The bridge model was trained on common objects in three permutations:
   i. train on spectrum pseudo-broadband and test on the same
   ii. train on SDSS broadband and test on the same
   iii. train on SDSS broadband but test on spectrum classes

(b) Results were compared to the original fraction of mismatched classes between spectra and SDSS

The second method is to simply correct the SDSS data. Histograms of common objects by band show bimodality in both the spectrum and SDSS data (likely attributable to the two fundamental groupings, C-types and S-types). Objects with matching classes have similar distributions to objects with unmatching classes, also irrespective of class. However, the SDSS is systematically brighter in all bands; that can be corrected.

(a) Corrections were assigned based on the median value for SDSS divided by the median value for spectra.

(b) Each set of SDSS data was divided by the appropriate correction factor in each band.

(c) An SVM was trained on all spectra and tested.

(d) Classes are predicted on each SDSS dataset with and without correction.

Predictions on SDSS are compared to recorded classes and some discrepancies are investigated.

Band	DeMeo short	Carvano short	Sergeyev short	DeMeo long	Carvano long	Sergeyev long
r	1.08	1.09	1.13	1.09	1.08	1.10
i	1.09	1.1	1.13	1.13	1.12	1.13
z	1.05	1.07	1.1	1.15	1.18	1.15

TABLE 5.1: SDSS data are divided by the values in this table to bring them in line with pseudo-broadband. The corrections are derived by dividing the median SDSS value in each band by its pseudo-broadband counterpart.

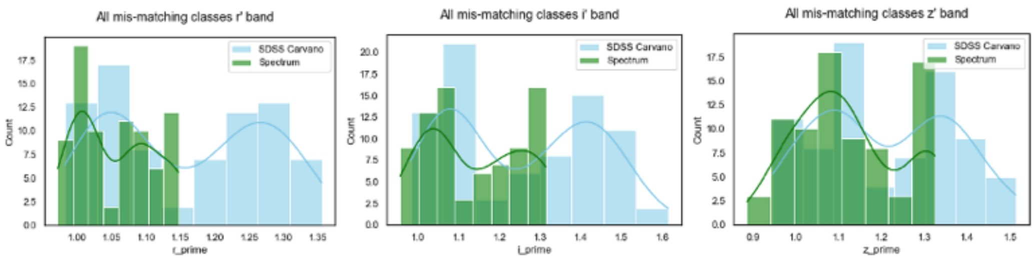


FIGURE 5.1: Offsets in band values between common spectra and SDSS data used to calculate conversion factors.The difference in the distributions between datasets is the most dramatic at r' band; for z' band the distribution peaks are quite close to one another.

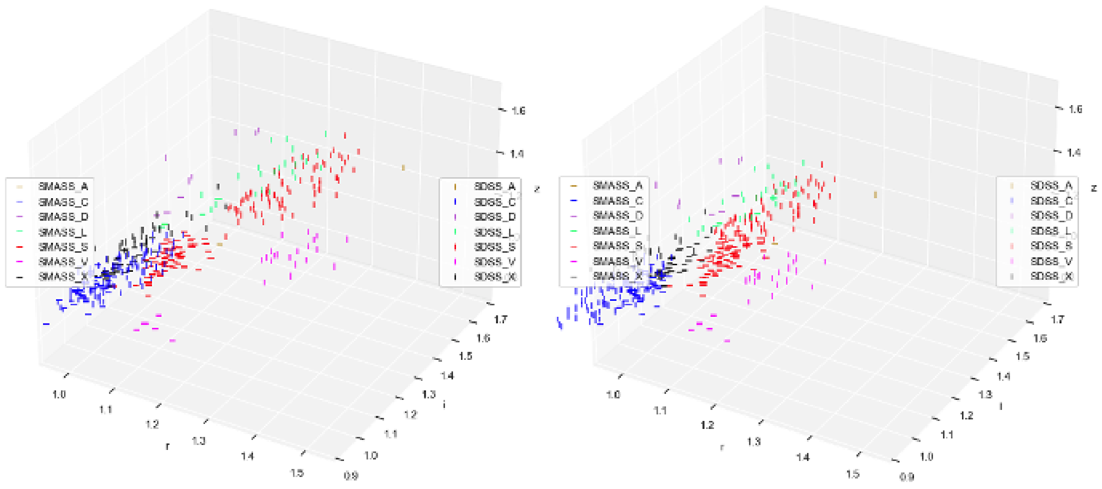


FIGURE 5.2: Common objects before (left) and after converting Carvano SDSS values (right). Horizontal and vertical bars are used to differentiate datasets visually (they do not denote error). After the correction factor is applied to SDSS the S and L-types move closer to their pseudo-broadband equivalents, but there is still a significant visible difference between the data for V-types.

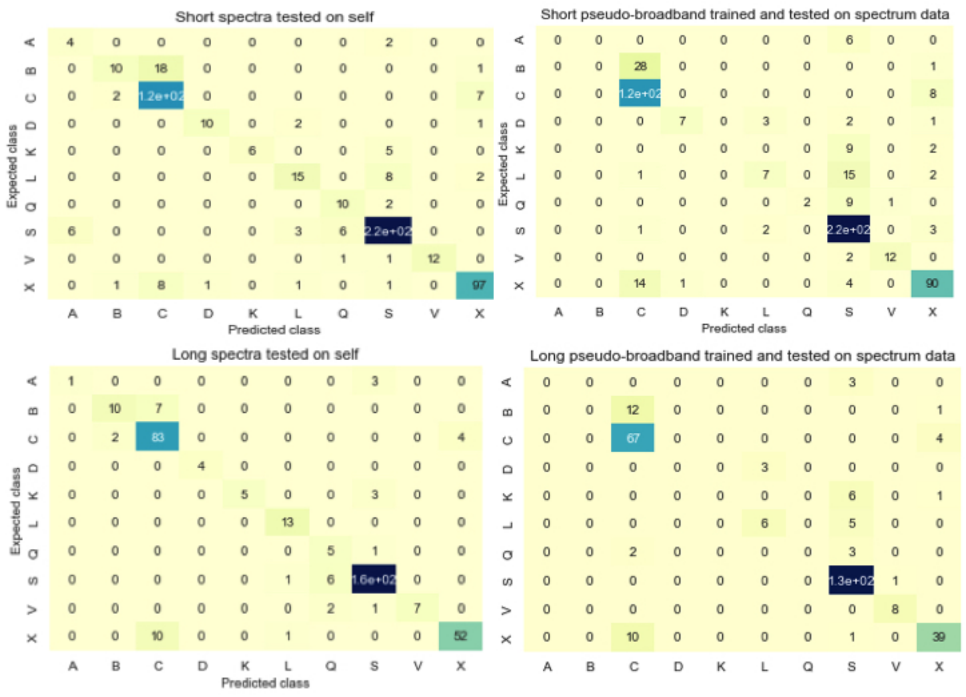


FIGURE 5.3: Confusion matrices showing predictions from broadband model vs. spectrum model on short spectra test set for Carvano classes (left) and DeMeo classes (right). Top row are short spectra, bottom row are long. The absence of A, B, and K from the broadband predictions suggest that the data resolution is too low for the classifier to pick out these types.

5.2 Results

5.2.1 Spectrum model predictions on broadband data

The simplest measure of changes of class boundaries after reducing to pseudo-broadband can be obtained by running the broadband-trained model on the original test set and comparing its predictions to the predictions of the spectrum-trained model. If the broadband version could perfectly represent the spectrum data, the results would be the same; where there is disagreement, there must be deterioration in the quality of the data because this is the only thing that has changed.

There are a few insights to be gleaned from Figure 5.3. The first is that the short data are more robust when switching to pseudo-broadband, especially for the DeMeo data. There is very poor matching between spectrum predictions and broadband predictions for the long data. The short data do experience problems with some classes. Of Carvano classes, the broadband model does not find type A at all, whilst of DeMeo classes it loses A, B, and K. Without doing any comparisons to recorded classes, it's already possible to say that the use of classes A, B, and K is not meaningful for broadband data, at least not at these population levels. This is not the fault of the method, but of the data resolution.

Classes	Short spectra	Short griz	Difference	Long spectra	Long griz	Difference
Carvano	N/A	83.7	N/A	N/A	86.4	N/A
DeMeo	85.0	79.6	-5.4	89.8	82.8	-7.0

TABLE 5.2: Results of griz models from short and long spectra as percentages. We are interested in the change in accuracy for DeMeo classes between spectra and griz; this is $5 - 7\%$, and it means that the best we can hope for from a pseudo-broadband classifier is around 85% accuracy. The result on Carvano classes is included to enable later comparison to tests on SDSS Carvano data.

5.2.2 Pseudo-broadband model on pseudo-broadband data

Now the results of the broadband models are compared to actual recorded classes for these spectra. As seen in Table 5.2, after reducing to broadband the classifier score drops by 5.4% for short spectra and 7% for long. Carvano classes have been introduced to facilitate comparison with SDSS; they score higher because B-types have been folded into C and K-types into L at the training stage, making less work for the classifier

A benchmark is now established for the pseudo-broadband classifier that reflects the expected upper limit of its performance. Figure 5.3 illustrates how well each class is predicted using DeMeo classes; deterioration of the result is actually worse than the score alone would indicate because the A, B and K classes disappear completely from predictions; all A are predicted as S, all B as C, and most K as L (otherwise X). In the long model predictions, the same thing happens; but D and Q also disappear. Type D are predicted as L, and Q are mostly classed as S (otherwise B). We cannot know if these three classes are truly ‘lost’ at this stage; but the classifier so far does not recognise any examples from the test set.

Given the predominance of S-types in the sample it is hard to know to what extent class imbalance is involved in misclassifications. The fact that a few Q and D objects, which disappear in the long model, are successfully identified in the short model (which contains more data) suggests that the role of class imbalance is more pronounced when 1) there are very few samples of the smallest classes and 2) the smallest classes are immediately adjacent to the majority class.

5.2.3 Comparing pseudo-broadband to SDSS

Before attempting to predict on SDSS using spectrum pseudo-broadband it is important to find out how well the SDSS and spectrum classes match up. Table 8 indicates that for common objects recorded classes align between spectrum and SDSS classes to about 70.7% (see Figure 5.6 for breakdown by class). It cannot be

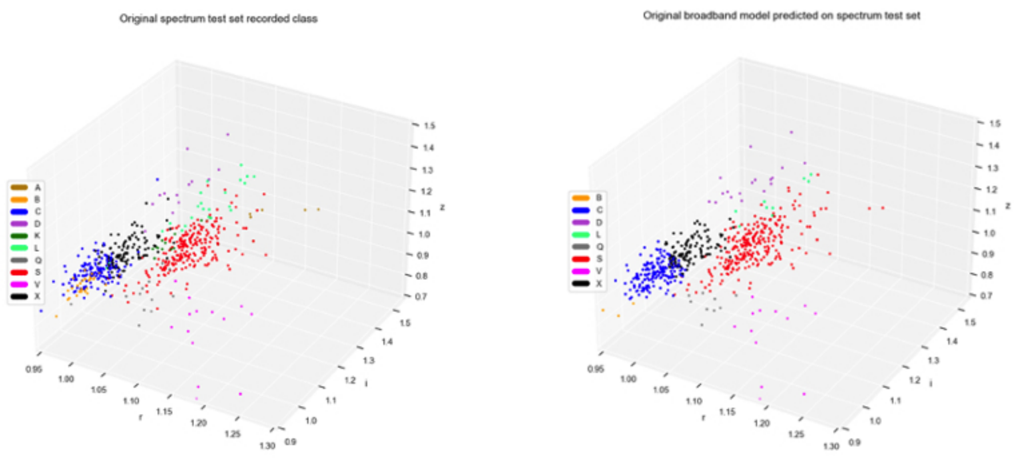


FIGURE 5.4: Recorded class for original spectra in pseudo-broadband r-i-z space (left) vs class predicted by model from spectra in the same space (right). The absorption of B (gold) into C (blue) is evident in the right panel.

known whether this fraction generalises to the entire dataset, but the implication that SDSS labels are only about 70% trustworthy makes it effectively impossible to establish a ground truth for models to learn. It is particularly difficult to say anything about A- and Q-types, which are barely represented in the common objects; these classes are also vulnerable to being overlooked by the models trained so far.

Data	Matching long (%)	Matching short (%)
Carvano	76	82
DeMeo	70	70
Sergeyev	0.71	0.78

TABLE 5.3: Class matching between spectra and SDSS for common objects

Before jumping to conclusions about classification systems, it's important to look at the data in r–i-z space. Common objects exhibit a larger scatter in the values for each SDSS bandpass than the values in spectrum pseudo-broadband. The effect is shown in for Carvano data in Figure 5.5. If this increased variability in SDSS were related to loss of flux at the high and low ends of the spectra when converting to pseudo-broadband, then a comparison of the data in i– vs. r–band should reveal a similar relationshp for both spectra and SDSS. However, the bottom subplots in Figure 5.5 shows that this is not the case.

The fact that spectra can redden due to high phase as well as variations in the surface presented due to rotation are both possible reasons for differing values at the high end of photometric/pseudo-broadband wavelength range. However, these differences should be random. Additionally, pseudo-broadband and SDSS catalogues rely on averaged values. There is no reason to expect SDSS to have systematically higher values (for example) in z-band due to random physical changes.

I note that I have obtained Carvano reflectances by converting from log reflectances, whereas DeMeo and Sergeyev data come from linear reflectances calculated by the authors; in both cases the divergence from pseudo-broadband is similar in nature. It seems more likely that the problem comes from differences in spectrophotometric calibration (covered in Section 3.2) between the SDSS system and the various reflectance spectrum datasets.

Finally, the fact that the data are yet more widely spread in the DeMeo and especially Sergeyev datasets relative to Carvano SDSS can be observed by comparing the distribution of the data in Figures 5.9, 5.10, and 5.11, which are all plotted on the same axes. I suspect that the increased uncertainty associated with the larger populations included in the DeMeo and especially Sergeyev catalogues contributes to the observed scatter.

5.2.4 Results of model predictions on SDSS datasets

Common objects test set	Score (%)
Pseudo-broadband train/test	78
SDSS train/test	81
Pseudo-broadband vs. SDSS recorded class	71
Bridge model	72

Table 5.4: Results of training and testing on the common objects using pseudo-broadband data vs. SDSS data on the same objects. For the bridge model, the SVM was trained on SDSS photometry with pseudo-broadband classes to try to teach the model to recognise the classes derived from spectra. The result is very slightly better than recorded class matching between SDSS and pseudo-broadband classes, but not enough to solve the problem.

Table 5.4 shows results of training and testing on common objects for the bridge model. Training pseudo-broadband and testing on pseudo-broadband is slightly less successful than training on all SDSS and testing on SDSS. Small sample size is likely a problem here, because the SDSS as a whole can train and test on its own classes at $> 90\%$. For the bridge model in which the classifier tried to 'learn' the spectrum classes from SDSS data, only 72% were correctly classified. This is higher than the 70.7% matching in recorded classes, but at low significance (78%) at this sample size. The bridge model is not used further in this work.

Turning to the corrections derived from common objects, for short spectra, pseudo-broadband can be predicted at 80.1% accuracy (with or without correction) on their own test set. Results on SDSS score between 62.3 – 75.3%. The latter

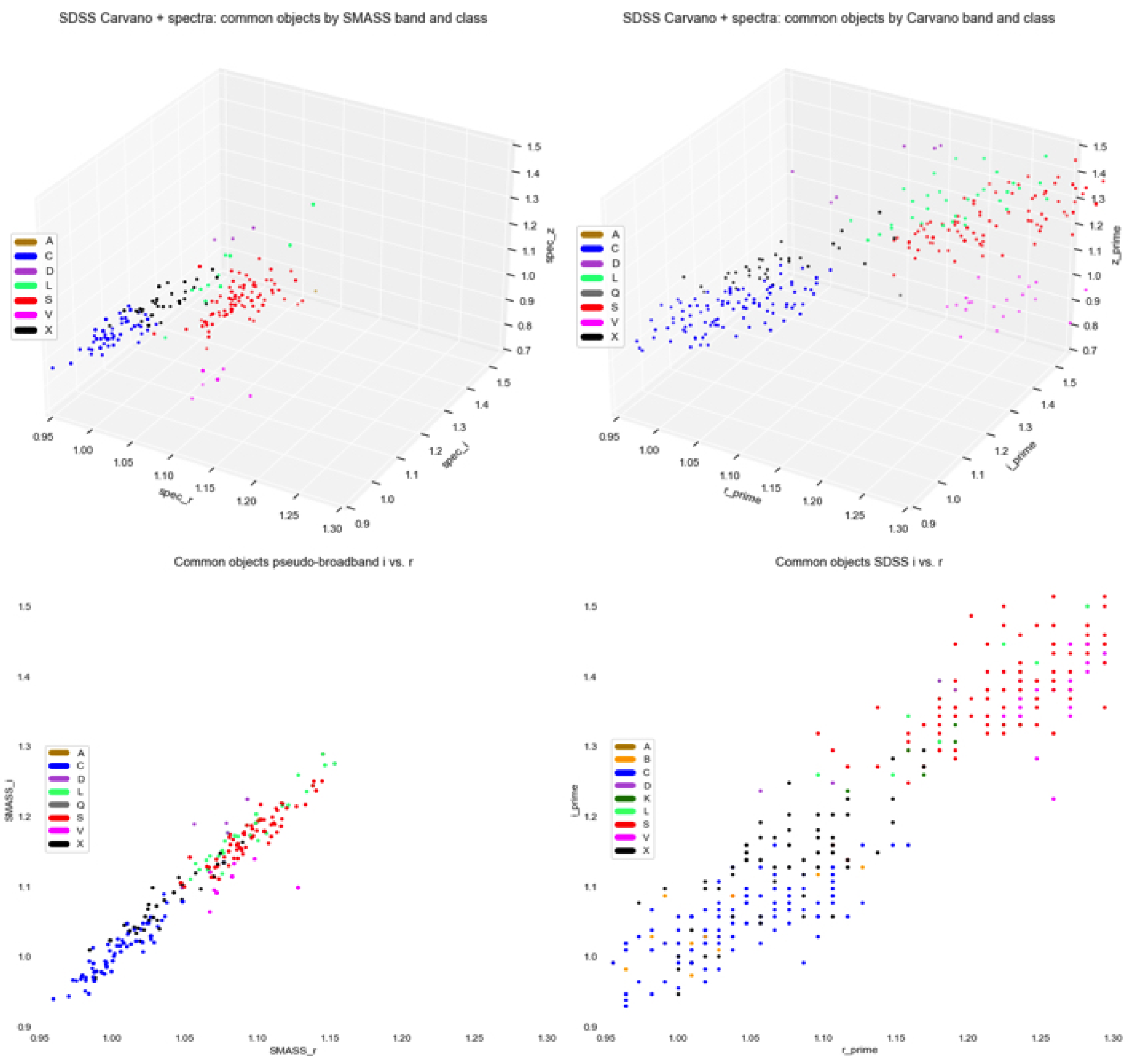


FIGURE 5.5: Comparison of reflectance values in common objects as seen from spectra converted to pseudo-broadband (left) and Carvano SDSS (right). The top row shows values in r, i, z relative to g-band, while the bottom row shows r − i only, where loss of flux in g and z should not matter. A wider spread in SDSS values as well as an apparent offset between the two datasets is evident.

scores improve after correction for DeMeo (+ 11.6%) and Carvano (+21%), but for Sergeyev data they degrade (-3.8%). (see Table 5.5). In a contest between the bridge model and the correction factor, the correction factor wins; however, figure 5.2 makes it clear that improvements with this method are limited. The classes are not fully superposed even after correction.

Short	Before (%)	After (%)	Difference (%)
Carvano	67.4	88.5	+21.1
DeMeo	62.3	73.9	+11.6
Sergeyev	75.3	71.5	-3.8

TABLE 5.5: Correction factor: before and after

In studying the plots of recorded classes in Figures 5.10, 5.9, for DeMeo and Sergeyev the boundaries of recorded classes seem arbitrary because the data show up as a cloud; however, it is reassuring that the classes mostly appear consistent between the three SDSS datasets. On the other hand, when machine-learning predictions are made, the S-type becomes subdivided and the plots take on more of a patchwork appearance, especially in Sergeyev where the recorded classes show severe dominance of S-types. In principle, this effect would indicate that the complexities of spectrum classes, once flattened into pseudo-broadband, project into $r-i-z$ space in complicated ways. That is, degeneracies in class are implied at broadband level, which is expected at such low resolution. But due to the poor correspondence between pseudo-broadband and real SDSS values, it is not possible to be more specific about how those degeneracies work. In fact, it is more likely that the fitted model is attempting to label some distortion of what the 'true' data would be if spectra for these objects were available, and the nature of the distortion is not-well characterised. This is the heart of the problem.

For Carvano, the appearance of discrete clumps occurs in part because I have removed objects with mixed or uncertain classes as well as photometric flags. However, the mixed Carvano classes are a minority of the dataset, and by looking at figure 5.11 it's obvious that the mixed classes don't make much difference. In fact the data have been more carefully curated in the first place, especially compared to the Sergeyev dataset, where the authors explicitly set out to find as many asteroids as they possibly could.

When predicting on DeMeo, the SVM boundary in C/X runs nearly parallel to the i-axis, similar to the recorded class C/X boundary in the Carvano classifier. However, the recorded DeMeo C/X boundary runs almost parallel to the r-axis, resulting in a different interpretation of the differences between the two classes. I am going to assert that the SVM classifier trained on spectra is 'less wrong' than

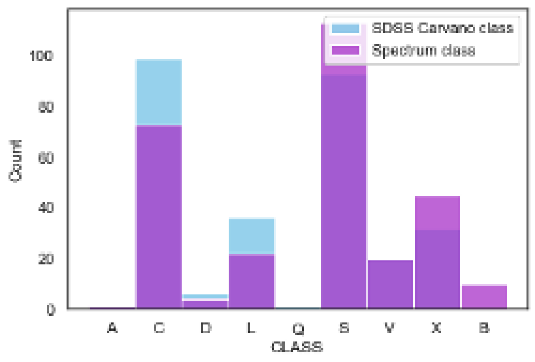


FIGURE 5.6: Comparison of class distributions between spectra and SDSS from common objects. For about 70% of objects the classes agree. The dearth of common objects from A and Q is a problem

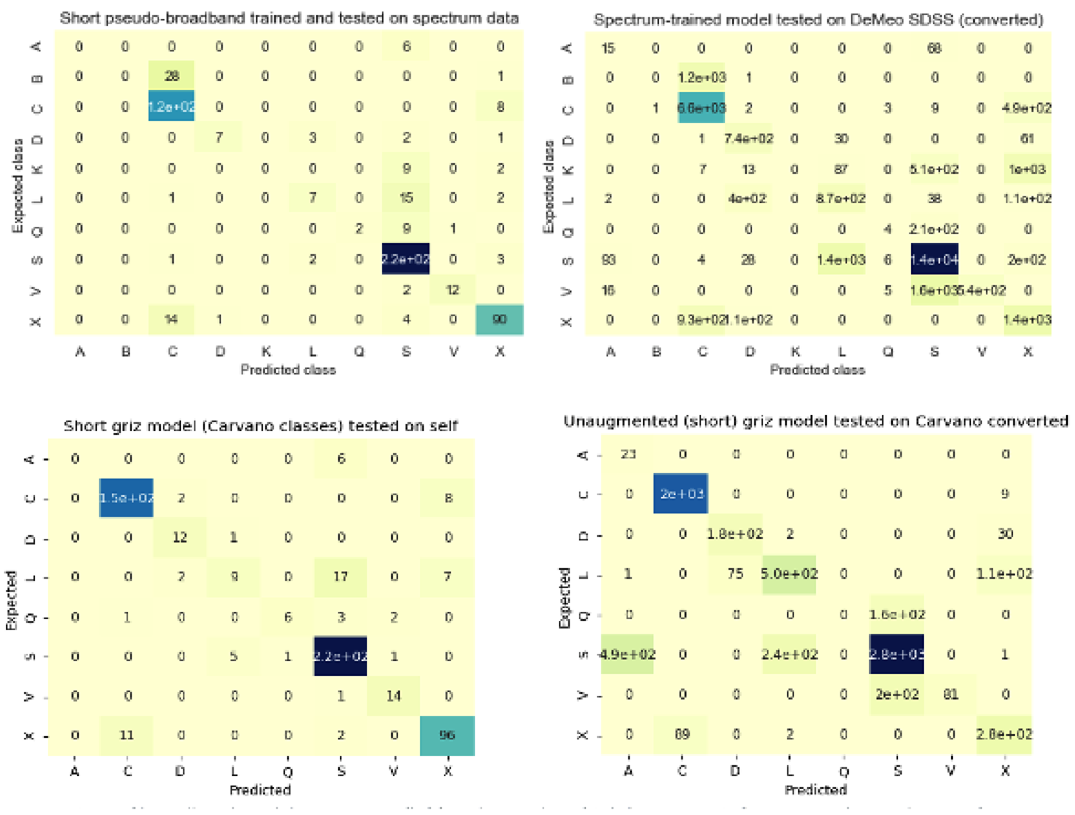


FIGURE 5.7: Comparison of broadband model on test sets (left) and SDSS data (right). Top row is for DeMeo classes, bottom for Carvano classes. There are few D-types for training, but the model agrees with SDSS classification well. Note that Type A 'reappears' in SDSS after being misclassified in the model's own test set. However Q 'disappears' when tested on SDSS data for Carvano. Classes B and K are 'lost' in the DeMeo test set as well as SDSS predictions, suggesting that the Carvano approach may be more realistic here.

the DeMeo SDSS classifier, especially because some of the objects used to create the Bus-Binzel classes will have had albedos that make separation of C and X much easier. There is a clear loss of B-types in the predictions, which again seem to be operating on the i-axis whilst the DeMeo classifier uses the r-axis. The SVM model is preferable to recorded classes in this specific point as well.

On the C/X boundary, Carvano predictions line up better to recorded classes than DeMeo predictions. Type A manages to reappear, even though the pseudo-broadband model failed to predict A-types on its own test set. At the same time, Q disappears, despite the presence of a distinct region associated with Q in the 3D plot of the test set. From the plots of recorded vs. predicted class for Carvano, class boundaries of the spectrum classifier are not a good match when it comes to Q and D. The Q (grey) region in Carvano SDSS appears distinct from the S-complex (red) but borders on V (magenta)—yet the spectrum classifier groups it with S. A smattering of D-types are predicted as X even though they are relatively

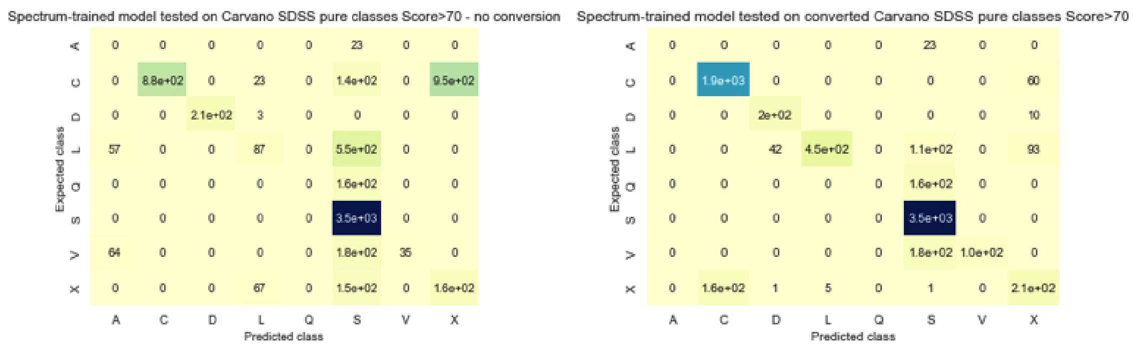


FIGURE 5.8: Results of spectrum-trained model on Carvano (left) before and (right) after correction factor applied. The corrected model still cannot find Q or A-types, but C and L recognition is greatly improved.

distant from the main X cluster in the Carvano plot; at the same time, a patch of Carvano L-types are assigned to D when, to the eye, they do not belong there.

The Carvano A-class consists of a small cluster of outliers in r-i-z space, but the spectrum classifier also predicts a substantial chunk of Carvano S-types as A. Given the expanded domain of values in SDSS, this is a slightly suspicious outcome. Looking at the full distribution of Carvano classes vs the model predictions, it is clear that the model captures the expected A-types plus some others that are recorded as S and L, but the intrusion of A into S/L is not too egregious (see Figure 5.11)

Overall, the recorded Carvano classes appear more convincing than the predicted ones and there does not seem to be a problem with imbalance in the sense that both A and D-types end up over-represented by the model (possibly because the overrepresentation involves moving 'inward' in an r-i-z space where the SDSS data occupies a larger domain). On the other hand, the two largest classes, S and C, appear to have 'gained territory', which could be the result of imbalance and warrants further investigation.

5.2.5 Long spectrum version

Results for the long spectra are presented next. In the interest of space, plots are not included here as they are similar to the short spectrum model results.

Table 5.6 shows the improvement in recorded class matching for common objects after correcting SDSS data. The long model improves less from correction than the short model when tested on Carvano data. In terms of scoring, the short model works better on Carvano data, but the long model works better on DeMeo and Sergeyev data. Here once again entire classes are lost, but this effect is worse than in the short model. A few things to note:

- Most objects are predicted to be C, L, S, or X
- A, B, K disappear completely; A becomes S, B becomes C, and K is spread between C, S, X.
- All but a few Q are lost to S. About half of L is predicted as S and a quarter as X. Most of D is lost to L.

At this point it seems that small sample size is beginning to bite when it comes to the overfitting of models, and the 'cleanliness' of the long data are not an advantage. In the contest between quality and quantity of data, when it comes to models trained on spectra, quantity wins.

Short	Before (%)	After (%)	Difference (%)
Carvano	68.8	80.7	+ 11.9
DeMeo	64.8	76.2	+ 11.4
Sergeyev	75.7	78.0	+ 2.3

Table 5.6: Results of classification with the long pseudobroadband model for each SDSS dataset, before and after the correction factor is applied. There are distinct benefits for the alignment of Carvano and DeMeo data to pseudobroadband values.

5.3 Discussion

Remembering that the common objects can be matched at no more than 80% agreement between spectra and SDSS, the short spectrum model achieves a good score on SDSS after corrections are made. In fact, given the 20% mismatch in recorded classes it is hard to see how it could possibly do better. Yet the accuracy score alone masks an important problem of smaller classes being misrepresented or not represented at all. The classes A, B, K, and Q are no longer able to be verified by the short spectrum test set.

There are two challenges to overcome here: firstly, poor calibration between spectrum pseudo-broadband and SDSS broadband, and secondly, loss of information in moving from spectra to broadband. Calibration is challenging for several reasons:

Neither spectral dataset is ideal for comparison with SDSS data. The short dataset is sourced from four surveys whose class assignments are not in complete agreement, and their wavelength limits force broadband normalization at 0.5 μm when it should be at 0.4775 μm to match SDSS. The long 0.435 – 0.925 μm data extend to better coverage of the SDSS g and z bands and are correctly normalised, but they come from only two surveys and still carry biases (described in Section 5.1.1

SDSS is normalised at the centre of g-band (0.4776 μm) and extends into the NIR beyond 0.9 μm. Carvano et al. (2010) converted magnitudes to log reflectances, which I have re-converted to linear values relative to g in order to render them equivalent to the reflectances used so far. DeMeo & Carry (2013) used a different approach, referencing 371 optical/infrared spectra which they were able to convolve with SDSS filter profiles for the sake of comparison with SDSS broadband. They do not say how they handled the short end of the SDSS g-band filter, which extends below the limit of 0.435 μm for any publicly-available spectra that I have been able to find.

Because I was unwilling to extrapolate into the end regions of the spectra, and because most S3OS2 and INT data are available only between 0.5 - 0.9 μm, both the z- and g-bands are incomplete in my work. I also note that the Bus-DeMeo system (DeMeo et al., 2009) relies on albedo, especially to separate B from C —also unavailable here.

The real problem in training a classifier on spectra and testing on SDSS lies in the scaling of the data. I have discussed the possible reasons for this in section 5.2.3. While I have ruled out some possible explanations, I am unable to pinpoint the problem. Since the same object is often brighter in r, i, and z in SDSS than in spectra, some problem with calibration is assured. But there is evidence that inclusion of a larger population of asteroids also results in greater scatter of the data, which I attribute to two intertwined causes. The first is error. Spectrum data have a selection bias in favour of brighter objects, which in turn implicitly reduces the error on reflectance for any given wavelength. Spectra were carefully curated and processed with eyeball checks; SDSS data are more widely inclusive and processed automatically. The size of SDSS data increases from Carvano to DeMeo to Sergeyev, and so does the size of the data domain. I attribute the second possible cause of the scatter to inclusion of a genuinely more diverse sampling of the population, especially in the dimmer parts of the Main Belt that are not observed by small spectroscopic surveys.

An apparent offset in the values of the three bands also exists between SDSS and pseudo-broadband. The offset has been improved as much as possible with the introduction of a correction factor. This adjustment is crude and certainly insufficient; it is clear from comparing 3D plots of pseudo-broadband with SDSS broadband that the former has values concentrated in a narrower region of each of the three bands. The most obvious reason for the result is the increased error range together with the sheer size of the SDSS data. In the absence of a clearly defined cause of the discrepancy, ideally a systemic correction would come from the data themselves by training a machine learning model to find the relationship. The so-called 'bridge model' that I have attempted here is not effective enough for good classification and cannot even be shown to enable a statistically significant improvement on the models trained on unaltered data. Small sample size of objects-in-common may be to blame for the failure, but there is not enough evidence to know one way or another.

In terms of information loss in the conversion to pseudo-broadband, all the worst-affected classes have small populations. To lose A is hardly unexpected because it is a very small class; but size is not the only factor because type D tests well even though it is also small; this could be because D-types are more tightly clustered than A. It is a little surprising that B is lost because it appears to have a well-defined position in r-i-z at the extreme end of the C-complex, but there are few samples for the classifier to train on. On the other hand, K is difficult to distinguish from L even at spectrum level, and Q has already been shown to be a class whose spectra are rather variable in shape. It is understandable that the model struggles on these classes.

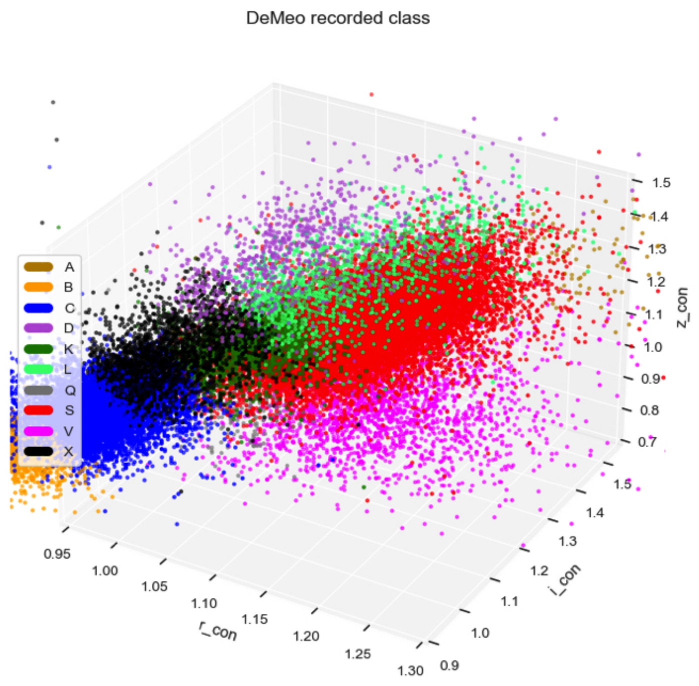
Overall, the increased scatter and brightness offset of the SDSS vs. spectra make the use of pseudo-broadband problematic without even thinking about known errors on magnitude/reflectance. It is to be expected that the inclusion of known errors would wash out most of the value of models trained on pseudo-broadband. The results of training on data that have been deliberately 'blurred' with a variational autoencoder in Chapter 6 will offer some feeling for the impact of random errors on the broadband classification models.

5.4 Conclusion

The transition to pseudo-broadband reduces model score by only 5% compared to full spectra, but this is enough to wash out the identifying characteristics of classes A, B, and K in the test set. A problem like this might possibly be correctable with careful augmentation of minority classes, but it turns out to be secondary to the issue of domain size in r-i-z space for SDSS data. The SVM can achieve a score comparable to the percentage of recorded classes that match between spectra and SDSS, but examination of 3D plots and confusion matrices makes it clear that the recorded classes have a more believable place in r-i-z space than the predictions of the SVM do. Therefore, the SVM fails to help us better understand the data.

I have shown that it is possible to improve matching between spectrum pseudo-broadband and SDSS with a correction factor derived from the mean differences in reflectance of common objects, but there are not enough data to teach a model to do a better job of translation. I am reluctant to recommend this correction factor without a deeper understanding of how SDSS magnitudes translate to flux, and anyway I consider the approach itself to run contrary to the machine-learning focus of this work.

The boundaries of classes are destined to overlap when broadband photometry is involved. However, we need to know what kind of groupings are latent in the data, irrespective of mineralogical knowledge or any other preconceptions. Clustering has the potential to offer unexpected information in this respect. However, in order to compare human classifier results to clustering results, it is important to understand the role of class imbalance on the data. It is also necessary to correct the sparsity that leads to some classes being washed out of the model, for which augmentation, the subject of Chapter 6 will be performed. Since the performance of the short data model is shown to be slightly more robust than the long version, I will carry on with the short dataset only.



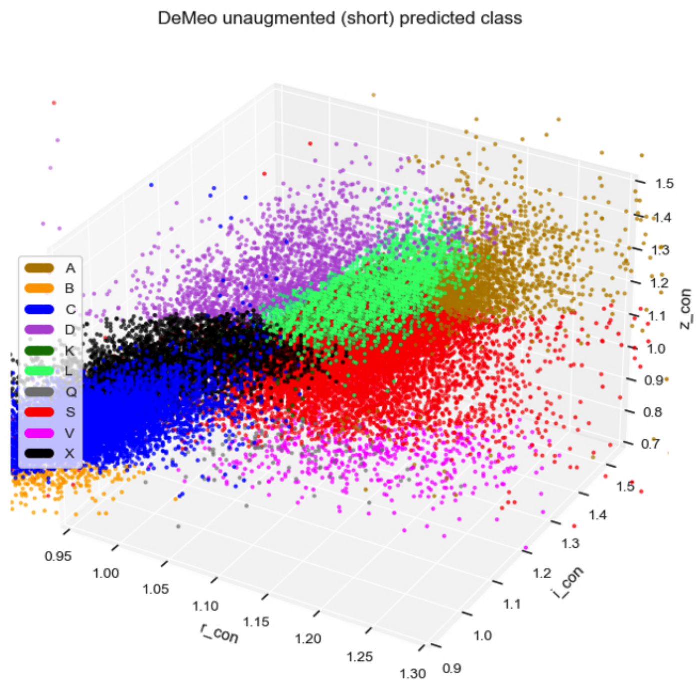


FIGURE 5.9: Recorded (left) vs. corrected pseudo-broadband model predicted classes for SDSS DeMeo. A large number of (gold) A-types are predicted because the main body of data now intrudes into a region of $r - i - z$ space that contains very few spectra; orange B-types are now found mostly out-of-bounds of the space defined in terms of spectra

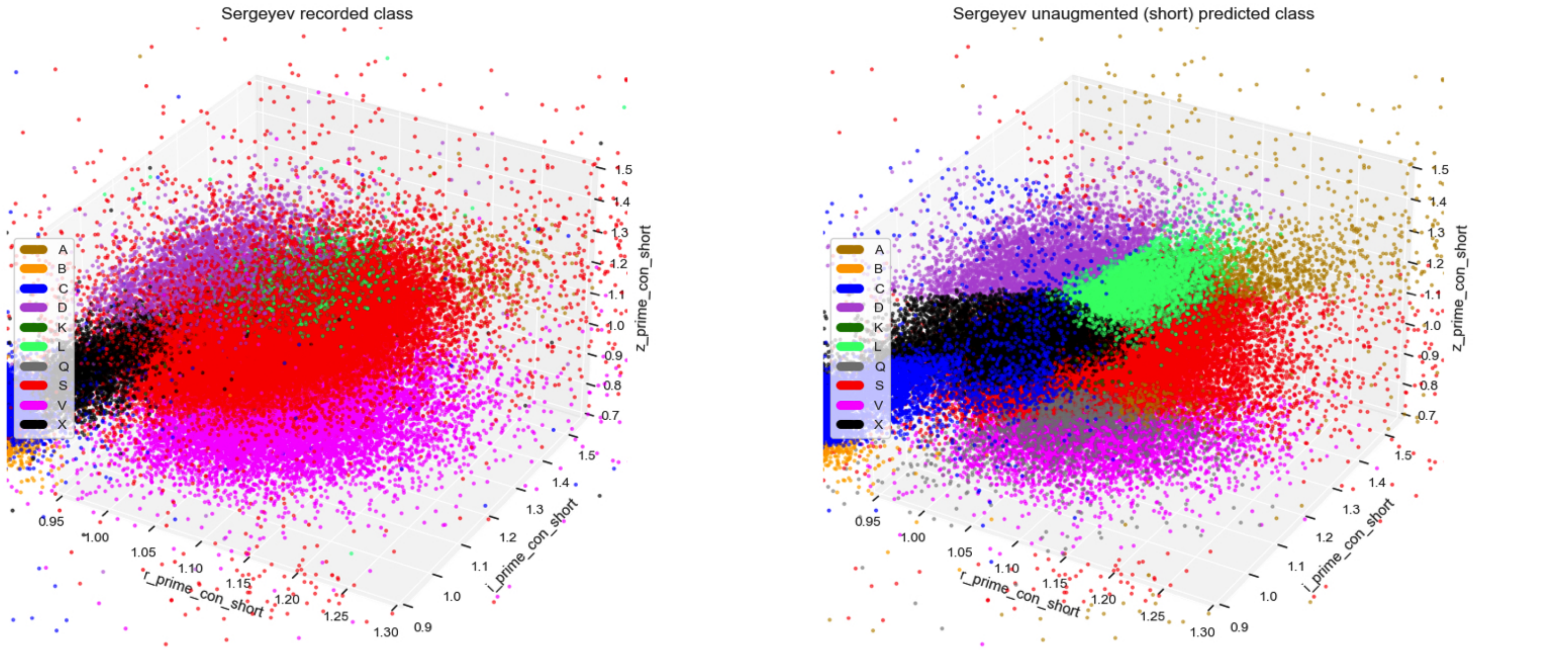


FIGURE 5.10: Recorded (left) vs. corrected pseudo-broadband model predicted classes for SDSS Sergeyev. The classifier turns a large red cloud into a patchwork of different classes, while C-complex as a whole is well out of bounds of the spectrum C-complex region.

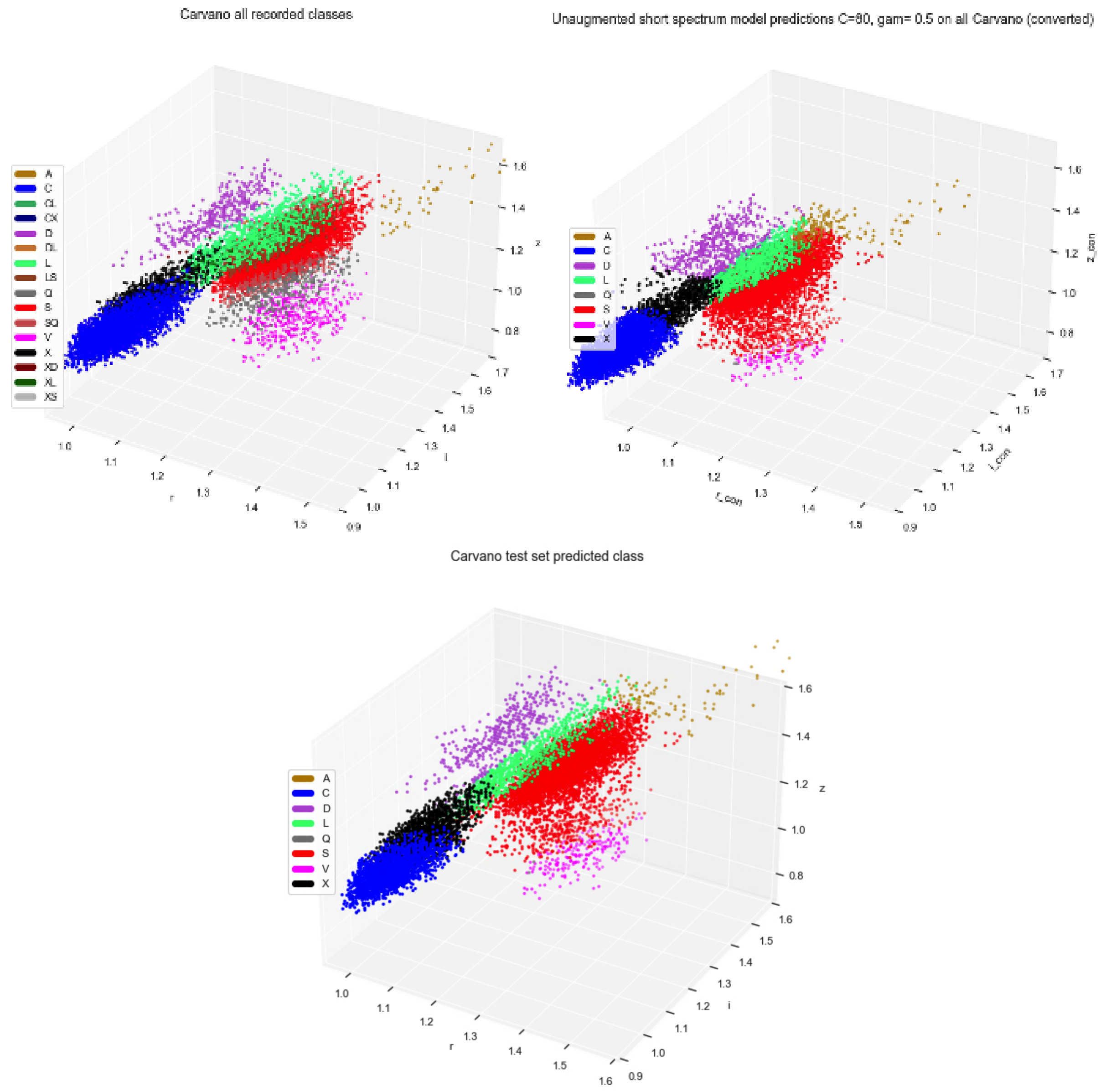


FIGURE 5.11: Top row: comparison of Carvano recorded classes (including subclasses) with pseudo-broadband predictions on basic classes. Bottom: predictions of 'pure' class model containing fewer borderline objects. Even when mixed classes are included, D and A are visibly separable, and a case could be made for Q-class having its own region. Even with multi-class objects, it is easier to 'see' structure in the cleaner Carvano data.

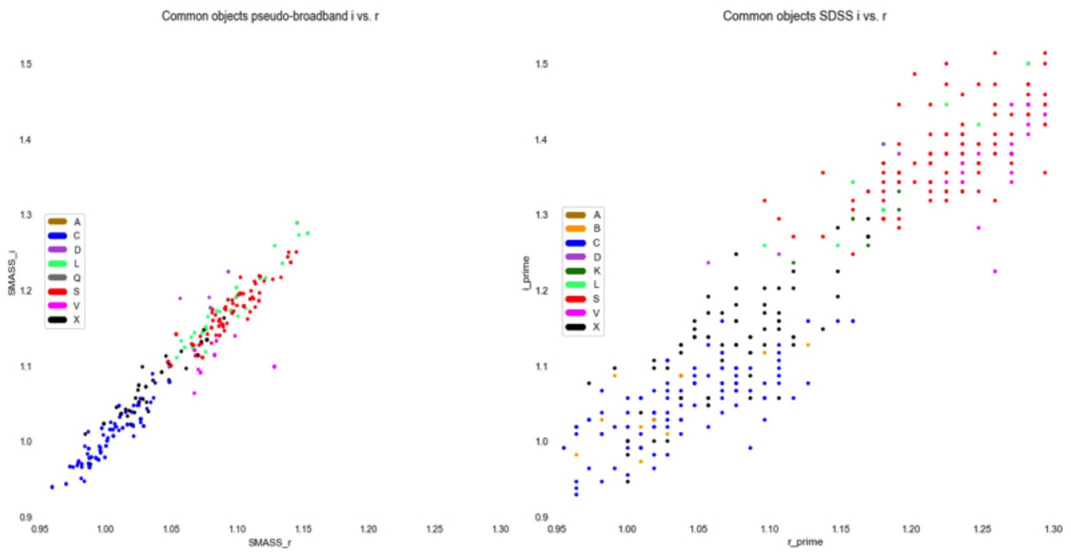


FIGURE 5.12: Comparison of common objects i vs. r in pseudobroadband (left) and SDSS (right). The SDSS objects are scaled over a wider range of values even in the bands where pseudo-broadband contains the full SDSS range of wavelengths.

Chapter 6

Addressing class imbalance with augmentation

6.1 Introduction

Up until this point some smaller classes (A, B, K) have been elided by models, while others such as D and V have generally survived intact. It has been unclear to what degree the population dominance of S and C classes over the others has factored into this result. Also, it has been impracticable to incorporate the effects of measurement error on models, nor to make allowances for the movement of class boundaries in the event of presumed new data becoming available in the border regions between classes. In this chapter I will attempt to address these limitations as forensically as possible.

I will compare results of an SVM trained on original data to models trained in ways designed to overcome imbalance. The first and simplest balancing method is to weight the classes, which does not require adding new data. The second and third involve augmentation using, respectively, SMOTE-ENN and VAE oversampling. Each of these methods carries its own characteristic advantages and risks, as discussed in 2.6.

6.2 Method

For each of the balancing methods, data from short spectra used in the previous chapters were divided into training, validation, and test sets by setting aside 20% of the data as a test set and then splitting the remaining 80% in a 4:1 ratio of train:validation. The training spectra were used for augmentation and model training, after which each model was tested on its own validation set as well as on the original (unaugmented) test set. Each model was tested on each of the three SDSS datasets using SDSS data with and without correction as described in Chapter 5.

6.2.1 Class weighting

The class weighting approach applies to the existing data without augmentation by giving equal emphasis to all classes, irrespective of their population size. Basic classes were used to train an SVM as in the previous section, except for the addition of a class weights parameter in the call to scikit-learn's support vector machine. By setting class weights to 'balanced', the algorithm automatically calculates an appropriate weighting for each class and multiplies the cost function by this weighting (Pedregosa et al. (2011a)). This is the most conservative approach to the problem, but with small sample size overall it is still likely to produce a weaker model, nor can it address the effects of errors or the anticipated scatter in new samples.

6.2.2 SMOTE-ENN

SMOTE-ENN was described in detail in Chapter 2.6.1. Here it is used to raise the numbers of each minority class to match the majority class, S. For the short spectra, this means 689 spectra of each class. The SVM model was trained on original spectra and then tested on SMOTE-ENN augmented spectra and vice versa. The data were then reduced to pseudo-broadband values and a new model trained and tested on SMOTE-ENN before being tested on the three SDSS datasets. As before, a separate model was trained for the Carvano system.

Unlike in the weighted version, SMOTE-ENN doesn't rely on a small number of points to determine a given class. Examples of SMOTE-ENN synthetic spectra for each class are shown in Figure 6.1 to illustrate that they are plausible representations of real spectra of their respective classes. Because SMOTE-ENN works within existing class boundaries, it seems like a safe method of augmentation, but there are unwanted consequences of interpolating between existing samples to generate new ones. Although the SMOTE-ENN dataset was created in the full 161 dimensions of the original spectra, when reducing to r-i-z space a substructure is introduced that results in complicated class boundaries and subsequent overfitting (see Figure 6.2).

6.2.3 Variational Autoencoder

As described in Chapter 2 the variational autoencoder resamples individual spectra according to a trained model that includes variance. For this work I used the Keras library's 1DConvnet to build a variational autoencoder for the spectra. There was a good deal of trial and error in this process. In order to get good approximations of the original classes it was necessary to train the VAE for each class individually and then combine all the samples into one dataset, working with a laptop and using Google Colab to do the heavy lifting. As with SMOTE-ENN, the number of samples was set by the number of S-types. Unlike SMOTE-ENN, it's possible to compare the variational spectrum to its progenitor directly. Most VAE spectra are very close to the originals in shape while usually being smoother, as shown in Figure 6.3. On the other hand, figure 6.4 shows that in a minority of samples, the approximation strays from the original. This is the reason for using a VAE: it is desirable to simulate some error and some scope for the variation that is expected from new observations. It is worth noting that the way the VAE represents the latent state of a class seems to depend on how similar the training examples are to one another. When looking at many synthetic spectra it's apparent that the tendency of the VAE to produce something different to the parent spectrum is greater if the set of parent spectra for that class has a large scatter (and vice versa). It is easier to find A-types that vary noticeably from their parents than it is to find C-types that vary, for example. I did not attempt to engineer the autoencoders to address this, but in future work it could be attempted.

Finally, it can't be stated whether a Gaussian distribution around an accurately encoded state is the best representation of variability in unknown asteroids, but at least there is scope for the new samples to move outside the original class boundaries in a statistically believable way.

6.3 Predicting on spectra with balanced data

Results of SVM classification on these new datasets are presented in Table 6.1. The weighted model does not improve on the imbalanced model. The SMOTE-ENN model produces a 91% accuracy result on the test set of unaugmented spectra, and 88% after conversion to pseudo-broadband; however, in the pseudo-broadband 3D plots (see Figure 6.2) the augmentation produces undesirable substructure in the upsampled data, creating a risk of overfitting.

At spectrum level the VAE gets a similar result of 96% and 92%, respectively, for augmented and original test spectra and 90% in pseudo-broadband.

Data 0.5 - 0.9 μm	Imbalanced	Weighted	SMOTE-ENN	VAE
Spectra (self)	0.851	0.829	0.969	0.957
Spectra (other)	0.839 (SMT)	N/A	0.912 (orig)	0.918 (orig)
	0.797 (VAE)	N/A		
griz spec (self)	0.825	0.692	0.875 (self)	0.898 (self)
griz spec DM	0.739*	0.299*	0.351 conv	0.306*
			0.386 unconv	
griz spec S	0.715*	0.298*	0.434 conv	0.374*
			0.415 unconv	
griz spec C**	0.885*	0.441*	0.517 conv	0.585*
			0.436 unconv	
* with conversion				
** fold B into C, K into L				

TABLE 6.1: Augmentation results summary showing the training/test data in the left column and model results for different class-balancing methods on the right. The VAE-trained model performs very nicely on test spectra, but when tested on the SDSS it is 30-40 percentage points worse than models trained on imbalanced data. It is likely that Carvano performs slightly better than the others because there are fewer classes to differentiate.

But when it comes to testing on the SDSS, the VAE and SMOTE-ENN models do not improve on the imbalanced model when classifying the original data, whereas I will show in Section 6.4 that the matching to SDSS classes is nothing short of disastrous. To get to the bottom of what is happening, some 3D plots of the spectra will be helpful.

Figures 6.5 and 6.6 show that weighting the classes solves the disappearing of A, B, and K, but it also over-assigns samples to these classes, usually by removing them from S. This model scores at 0.692, markedly worse than the original imbalanced model. Figure 6.7 and indicates that the results for SMOTE-ENN (score:0.75) and VAE (score: 0.745) are very similar in their pattern of misclassification. The 'missing' classes have been recovered, but there is still a tendency to confuse them with their main-complex neighbours; i.e., C and S.

In examining the misclassifications of this model, in many cases there is an argument to be made in favour of the VAE classifier's decisions. For example, the problematic Q-type has seven misclassifications by the VAE, but examination of spectra shows that five of these are type Sq and could be considered borderline objects. Similarly, half of wrong A-predictions are labelled Sa. Of the five spectra wrongly predicted as D-type, three are Ld. Only one S is misclassified (as L) and there are no mistakes for V-type. K fares worst, but even in this case, of the eleven wrong K-types, three are Xk.

When it comes to B/C, these classes are tricky to distinguish even with full spectra. Looking at plots, of those wrongly predicted as B-type, most have a slight downward slope longwards of 0.8 μm and one is subtype Cb. Conversely, there are several B-types predicted as C by the model; visually, in these cases there is very little to distinguish these classes (see Figure 6.9). However, DeMeo et al. (2009) did use albedo to determine definition of B-types in that taxonomy, which could be a factor in their choice of labels for these asteroids.

The VAE model is sometimes wrong, but not greatly wrong, and it clearly helps to correct training set class imbalance. Although performance by SMOTE-ENN is similar, I will show in the next section that overfitting causes concern with SMOTE-ENN.

6.4 Predicting on the SDSS with balanced data

The results of applying the balanced models to the SDSS are described in this section. They are all worse than the unbalanced models. The weighted model (6.10 second row) greatly reduces the population of S and, to a lesser extent, X. It predicts many more A, B, and L-types than expected, across the board. The effect on DeMeo and Sergeyev data is to render them almost unrecognisable. For Carvano, matters are a little better. However, D is divided into two populations, A spreads into S, L, and Q, and X has been nearly overrun by C even though C (as a large class) should have been de-emphasized by this model.

For the VAE model, figure 6.11 shows a similar proportion of class redistribution, but the boundaries look very different. Each class shows up as an oblate cloud with simple but soft boundaries rather than the severe and rather arbitrary lines as of the weighting method. Across all three datasets the nature of the boundary between C/X now resembles the original Carvano boundary. However, in the Carvano data there is still a tendency for D to appear in S and S in Q, and fewer X-types are predicted here. Because the Carvano data have clearly-defined boundaries, what emerges is an imprint of the disagreements between spectra and SDSS class boundaries, because here the classifications are visually inconsistent with the shape of the Carvano data. There are a few examples of X and L-types popping up in unexpected places, indicating that the VAE has altered some spectrum parameters beyond the bounds of their original classes.

Turning to SMOTE-ENN, with an accuracy score of 88% on its own test set it appear to outperform the imbalanced classifier from the previous chapter. This is misleading because the ‘spiderweb’ structure that was seen in the spectrum augmentation produces undesirable effects on class boundaries in the SDSS. In the bias-variance tradeoff discussed in Chapter 2, this model is an example of emphasis on variance run amok. A slightly subtler point can be made from Table 6.1 regarding the risks of relying on a conversion factor to translate pseudo-broadband to SDSS broadband. The gap between scores on converted and unconverted data greatly narrows for SMOTE-ENN results. In other words, even the small amount of variability introduced by augmentation invalidates the correction because the model overfits to SMOTE-ENN class boundaries.

6.5 Discussion

There is good news and bad news to be found in these results. The good news is that both SMOTE-ENN and VAE can produce believable synthetic examples of the spectrum classes, with individual SMOTE-ENN samples appearing a little more realistic than the rather smooth VAE objects. They also appear to perform well at pseudo-broadband level. The bad news is that when the augmented data are examined in r-i-z space SMOTE-ENN has introduced unwanted substructure through its interpolation between existing datapoints. There does not seem to be any harm in this when working with the small spectrum test set, but when a SMOTE-ENN model is applied to the large cloud of SDSS data the class boundaries become segmented and excessively complicated: bad news.

Once the data are augmented with VAE, the SVM is quite effective at identifying very similar classes B, C, and X in pseudo-broadband, especially considering that no albedo has been provided. For spectra, I would recommend the VAE model as an improvement on the unaugmented model developed in earlier chapters, and I would go so far as to say that VAE augmentation is a credible way to balance classes with asteroid spectra if one is happy with a little dirt in the outcome. It seems to offer a little insurance against overfitting that SMOTE-ENN and weighting do not.

However, just as seen in chapter 5, the problem of matching to SDSS has not gone away, and sadly is only exacerbated by augmentation. For a VAE model tested on corrected SDSS data, accuracy scores drop by around 30% to between 31% and 59% depending on the dataset. This occurs despite the encoder's tendency to produce synthetic objects that appear very similar to their progenitors. It is difficult to square this result with the fact that a) unbalanced pseudo-broadband models worked better than this, and b) augmentation of the spectra was very conservative in its changes. But the reason for the worse result probably has to do with the nature of the accuracy score itself. In unbalanced versions of the models, there were only a handful of K, Q, etc. for the classifier to get wrong, so the scores were artificially inflated. Now are looking at a model that continues to fail to separate certain classes (K/L/Q/S and also C/X/B) but now there are *even more* of the minority classes included in the sample, so we are seeing the true nature of the problem.

The Carvano data give the clearest picture of what the distribution of asteroids in r-i-z space looks like, because there are visible clusters of objects. It may be that the larger DeMeo & Carry (2013) dataset depends too heavily on NIR features to be useful in this context, but it seems more likely that this blurriness is the inevitable result of their inclusion of objects with larger errors on magnitudes. Certainly, the errors on the very large dataset reported by Sergeyev, Alexey V. & Carry, Benoit (2021) can account for the difficulty of classifying those data and I would hesitate to blame any classifier for failing.

6.6 Where does this leave us?

With augmentation I set out to balance the classes and in turn, the classifiers — but the results are anything but balanced. This should not have surprised me, but it did. What these results are saying is that assumptions about the nature of a class relate not only to the relationships of vectors in feature-space, but to the *quantities* of vectors in regions of feature-space; i.e., the density of the data. When densities are changed to equalise class numbers, the boundaries of classes naturally shift, and they do not necessarily shift the way our classification system says they should. This entire process is about the nature of biases, and it is quite difficult to know what to conclude. Weighting gives a different result to linear interpolation, which gives a different result to Gaussian resampling, and no set of results comes close to replicating the classification systems that have been inherited and progressively updated from the 1970s.

Well, besides messing up the classifiers there was a reason for doing the augmentation in the first place, and that was to enable the data to be clustered — which cannot be done effectively if it is unbalanced. Chapter 7 takes the augmented data and clusters them to find out what groups they fall into naturally.

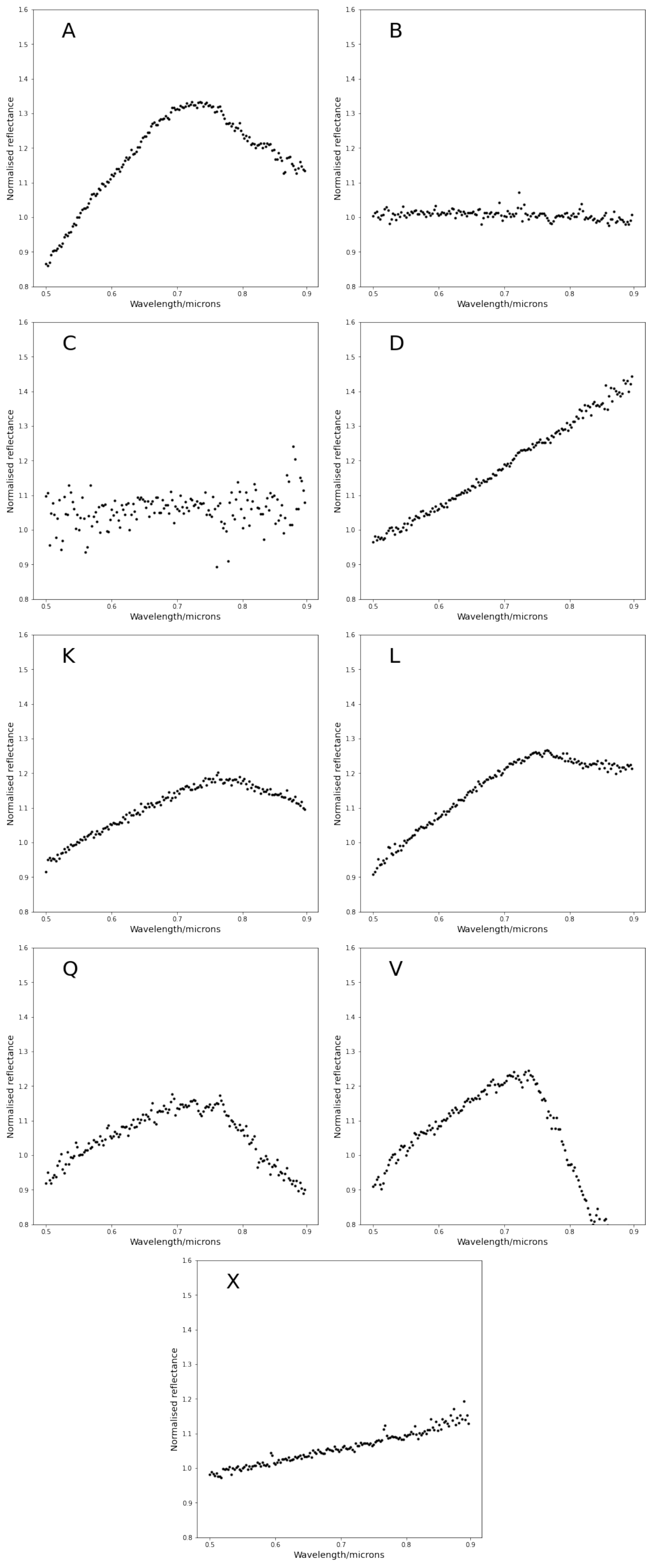


FIGURE 6.1: Randomly-selected SMOTE-ENN spectra for each augmented class.

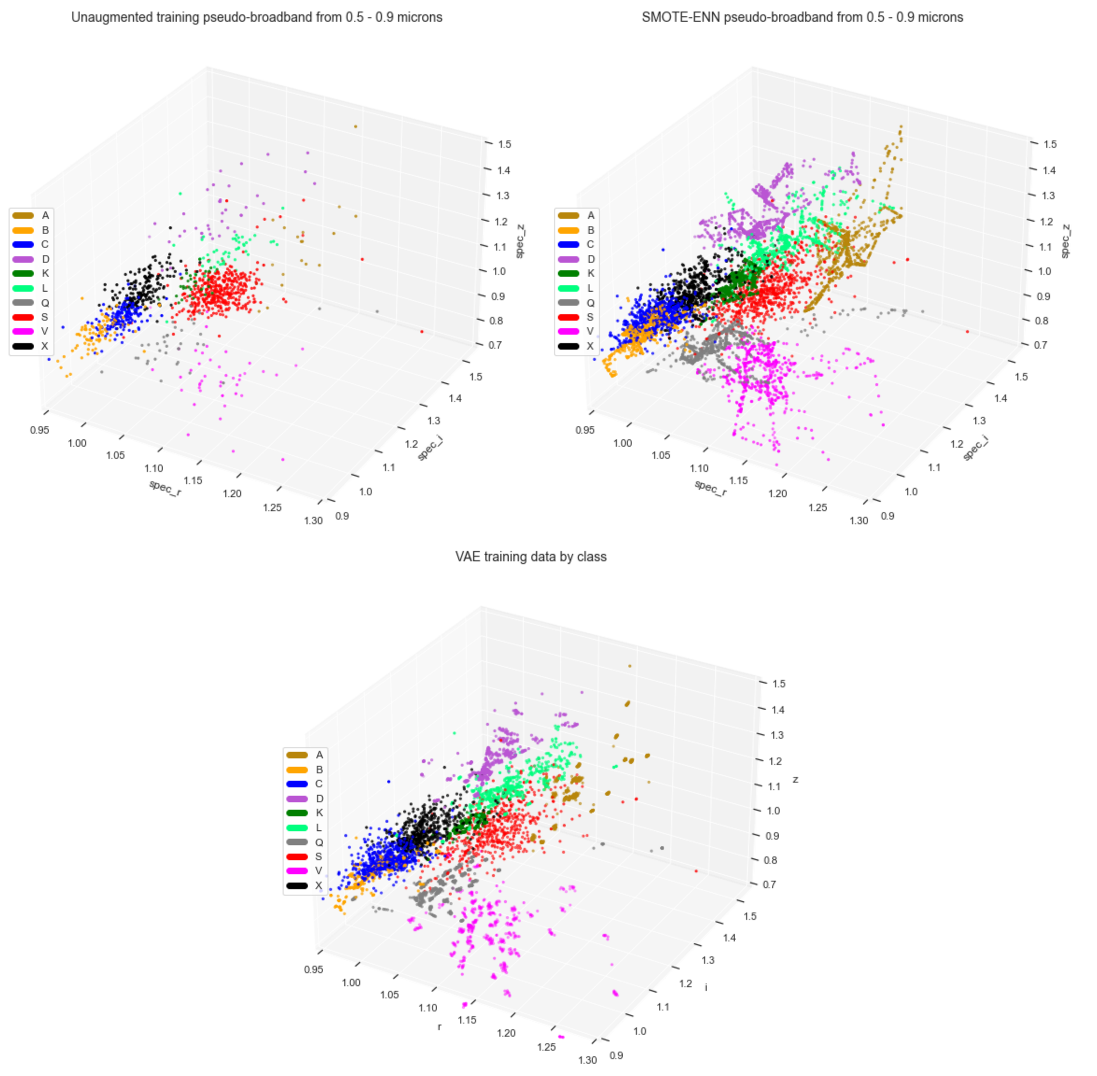


Figure 6.2: Comparison of original training set (top left) with SMOTE-ENN (top right) and VAE (bottom) training sets in pseudo-broadband. Despite SMOTE-ENN being performed on 161 dimensions, in r-i-z space we see that linear features are introduced. VAE augmentation produces a 'blurring' of datapoints by resampling near to the original datapoints.

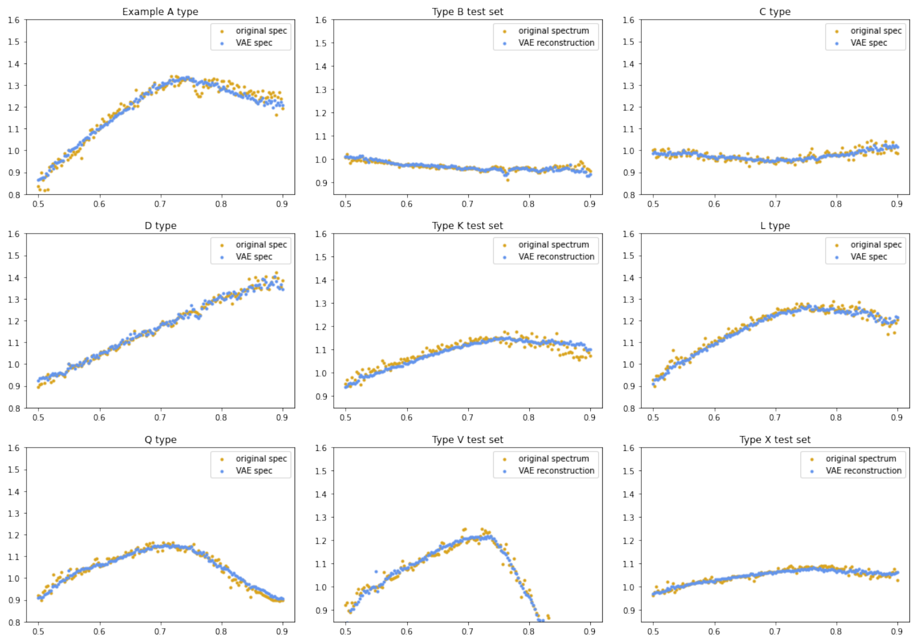


FIGURE 6.3: Examples of original and augmented plots of reflectance vs. wavelength (μm) for each class show that the VAE can generate good representative samples of each type. Simulated spectra tend to be less noisy than originals, but most noise will wash out in conversion to pseudo-broadband anyway.

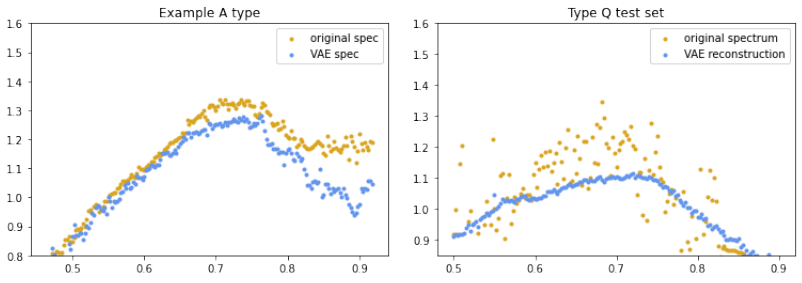


FIGURE 6.4: Examples of VAE spectra reflectance vs. wavelength (μm) that show some structural variation from their progenitor spectra. These are more likely to be found in classes that are poorly-defined, and can be treated as representations of what might happen in the presence of errors, for which the models do not otherwise take account. This is some insurance against overfitting.

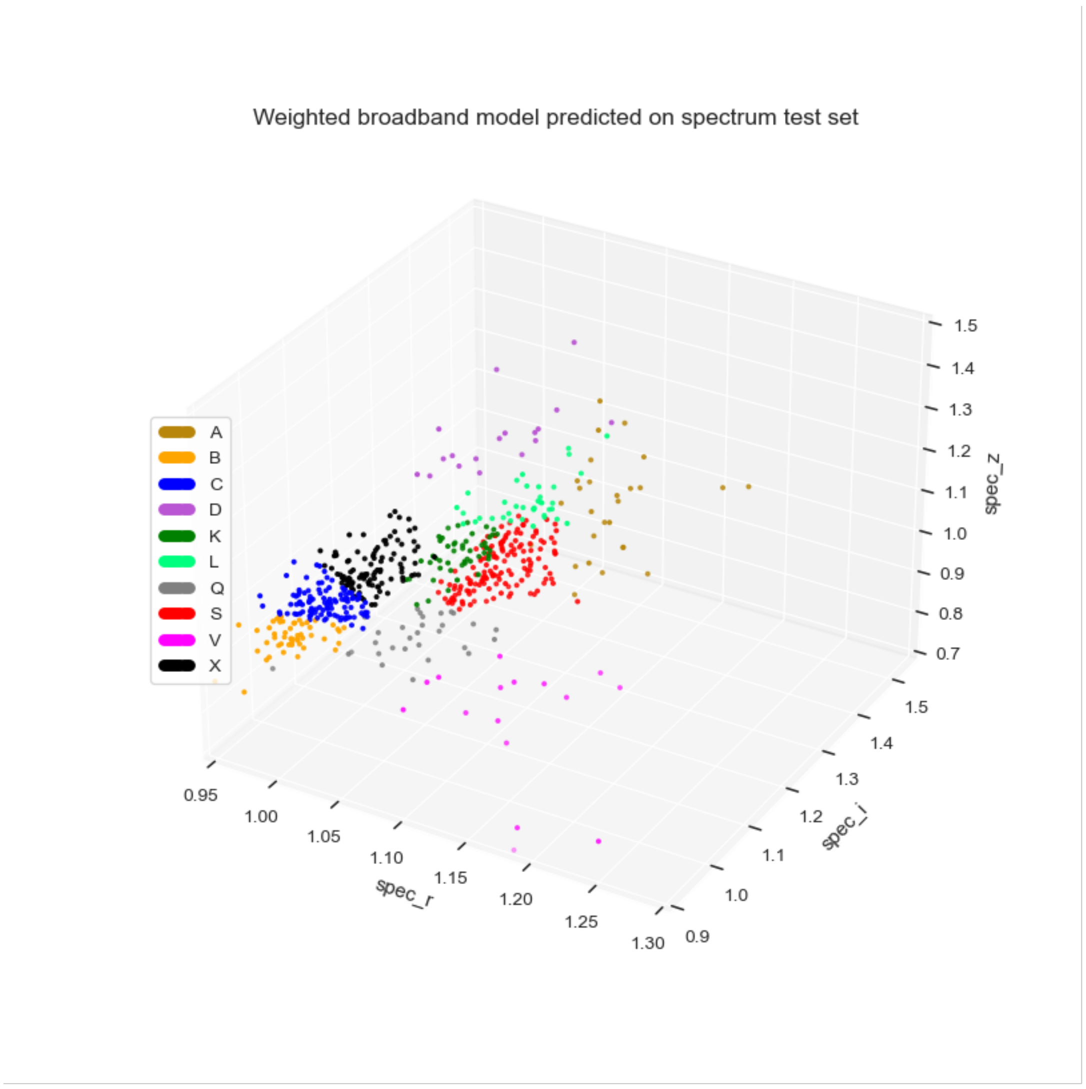
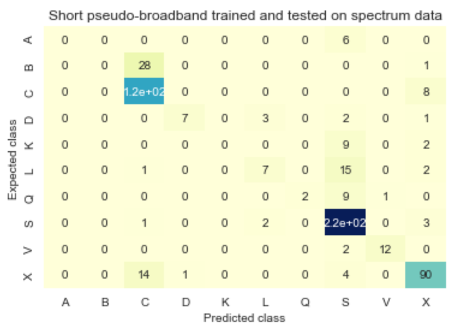
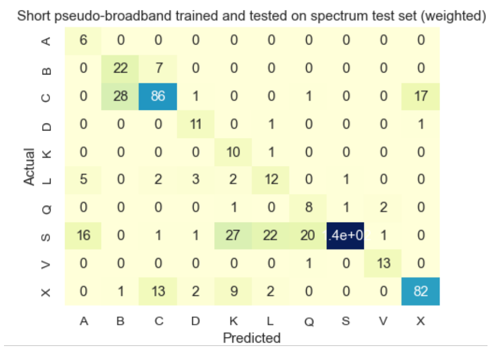


Figure 6.5: Test set classes as predicted by weighted model on pseudo-broadband. Compared to Figure 5.4 classes A, B, Q, L, and K all gain members, mostly at the expense of S. These designations do appear somewhat arbitrary, however.


Short pseudo-broadband trained and tested on spectrum data

Expected class \ Predicted class	A	B	C	D	K	L	Q	S	V	X
A	0	0	0	0	0	0	0	6	0	0
B	0	0	28	0	0	0	0	0	0	1
C	0	0	1.2e+02	0	0	0	0	0	0	8
D	0	0	0	7	0	3	0	2	0	1
K	0	0	0	0	0	0	0	9	0	2
L	0	0	1	0	0	7	0	15	0	2
Q	0	0	0	0	0	0	2	9	1	0
S	0	0	1	0	0	2	0	2.2e+02	0	3
V	0	0	0	0	0	0	0	2	12	0
X	0	0	14	1	0	0	0	4	0	90


Short pseudo-broadband trained and tested on spectrum test set (weighted)

Actual \ Predicted	A	B	C	D	K	L	Q	S	V	X
A	6	0	0	0	0	0	0	0	0	0
B	0	22	7	0	0	0	0	0	0	0
C	0	28	86	1	0	0	1	0	0	17
D	0	0	0	11	0	1	0	0	0	1
K	0	0	0	0	10	1	0	0	0	0
L	5	0	2	3	2	12	0	1	0	0
Q	0	0	0	0	1	0	8	1	2	0
S	16	0	1	1	27	22	20	4e+0	1	0
V	0	0	0	0	0	0	1	0	13	0
X	0	1	13	2	9	2	0	0	0	82

Figure 6.6: Confusion matrices of test set classes as predicted by imbalanced pseudo-broadband model with score 0.825 (left) and weighted model with score 0.692 (right). To compensate for the effects of the majority classes, the weighted model reassigns some S-types to classes A, K, L, and Q and some C-types to class B.

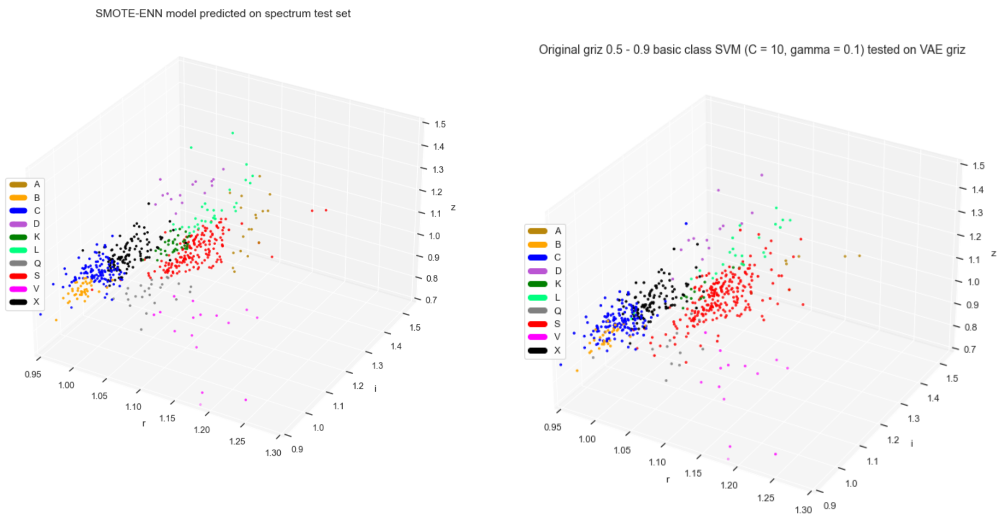
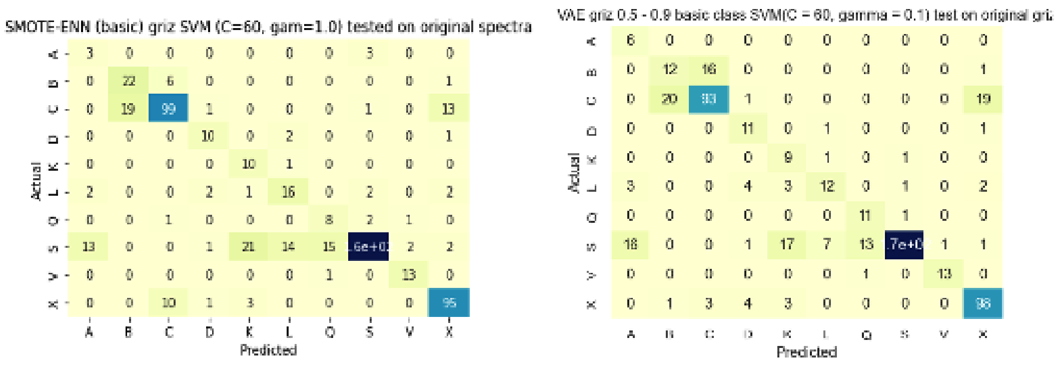


FIGURE 6.7: Comparison of pseudo-broadband test set classes as predicted by SMOTE-ENN model (left), and VAE model (right). Similar to the weighted model in 6.5, in the SMOTE-ENN model, B objects gain territory within C-complex and more Q-types appear. The VAE model is fairly close to the predicted classes from Figure 5.4; it is able to predict the smaller classes without over-predicting them.



Actual \ Predicted	A	B	C	D	K	L	Q	S	V	X
A	3	0	0	0	0	0	0	3	0	0
B	0	22	6	0	0	0	0	0	0	1
C	0	19	99	1	0	0	0	1	0	13
D	0	0	0	10	0	2	0	0	0	1
K	0	0	0	0	10	1	0	0	0	0
L	2	0	0	2	1	16	0	2	0	2
Q	0	0	1	0	0	0	8	2	1	0
S	13	0	0	1	21	14	15	1.6e+02	2	2
V	0	0	0	0	0	0	1	0	13	0
X	0	0	10	1	3	0	0	0	0	95

Actual \ Predicted	A	B	C	D	K	L	Q	S	V	X
A	6	0	0	0	0	0	0	0	0	0
B	0	12	16	0	0	0	0	0	0	1
C	0	20	83	1	0	0	0	0	0	19
D	0	0	0	11	0	1	0	0	0	1
K	0	0	0	0	9	1	0	1	0	0
L	3	0	0	4	3	12	0	1	0	2
Q	0	0	0	0	0	0	11	1	0	0
S	18	0	0	1	17	7	13	1.7e+02	1	1
V	0	0	0	0	0	0	1	0	13	0
X	0	1	3	4	3	0	0	0	0	98

FIGURE 6.8: Comparison of spectrum test set classes predicted by SMOTE-ENN model with score 0.750 (left), and VAE model with score 0.745 (right). The pattern is virtually identical. Classes B, K and Q are now detectable, but at low accuracy, and L continues to be absorbed by S despite balancing the data. The VAE has an edge for A, K, L, and Q, but the tendency of S to be overpredicted persists.

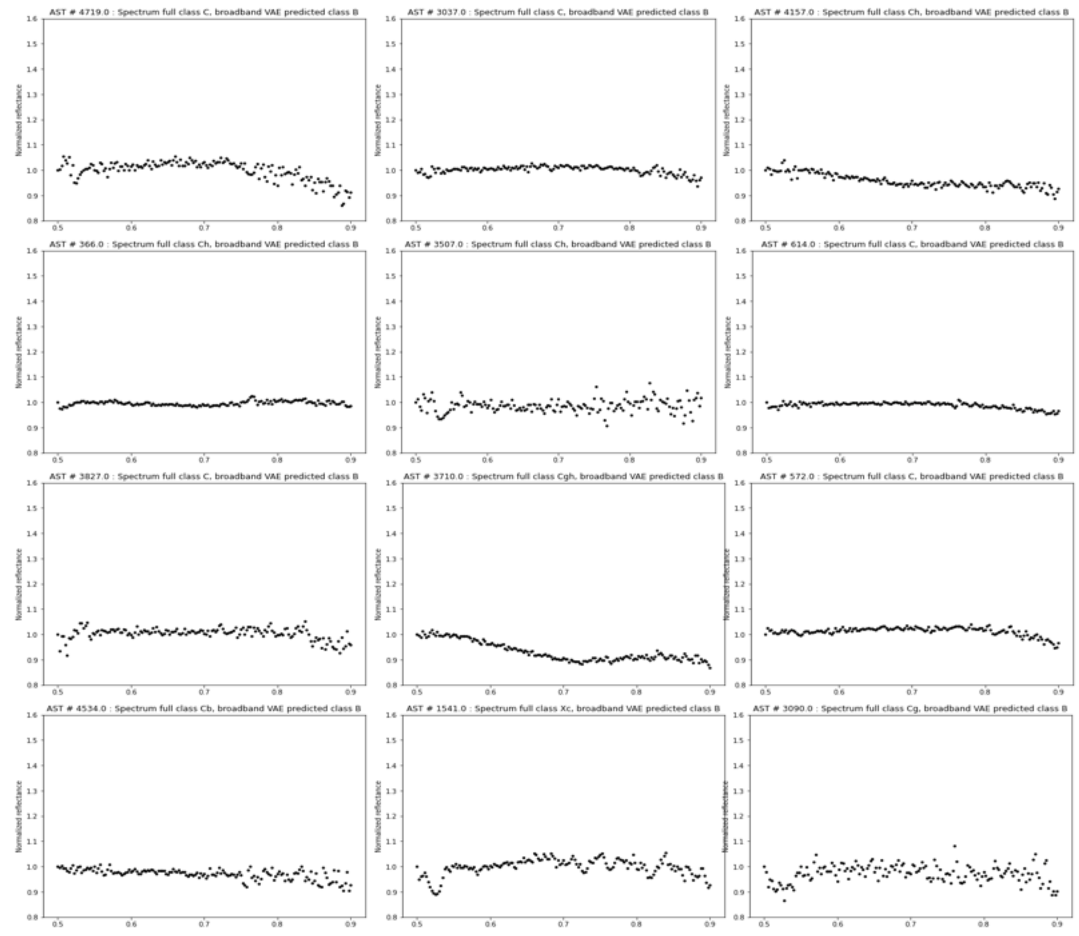


FIGURE 6.9: Examples of C and X-type asteroids predicted as B-type by the VAE SVM. Most do have a slight negative slope characteristic of B-types. The assignment of these objects to C-type may be a consequence of albedo being factored into the Bus-DeMeo taxonomy—which the SVM cannot see in the spectra alone.

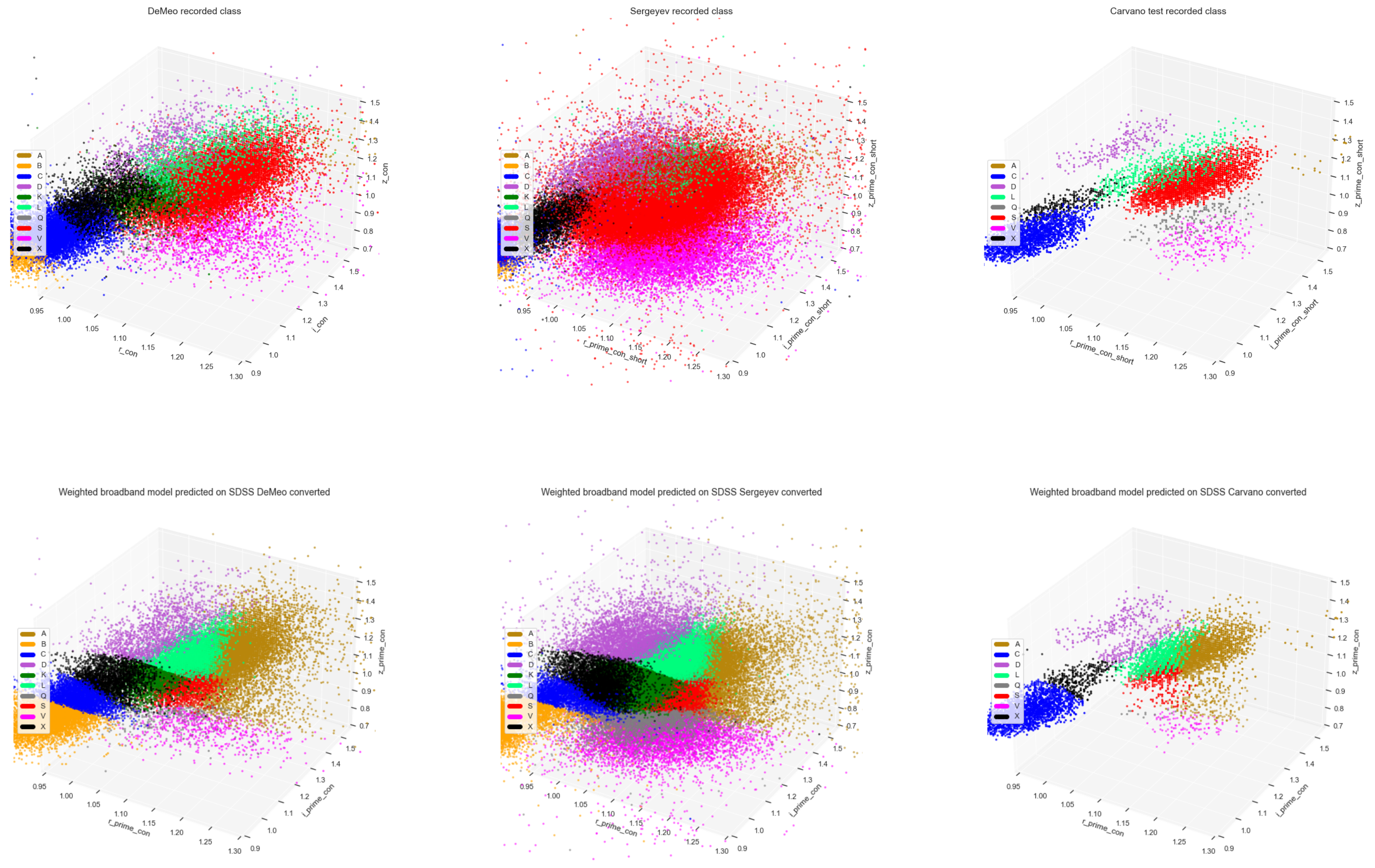


FIGURE 6.10: True (top) and weighted model predictions (bottom)(l to r): DeMeo, Sergeyev, Carvano SDSS.

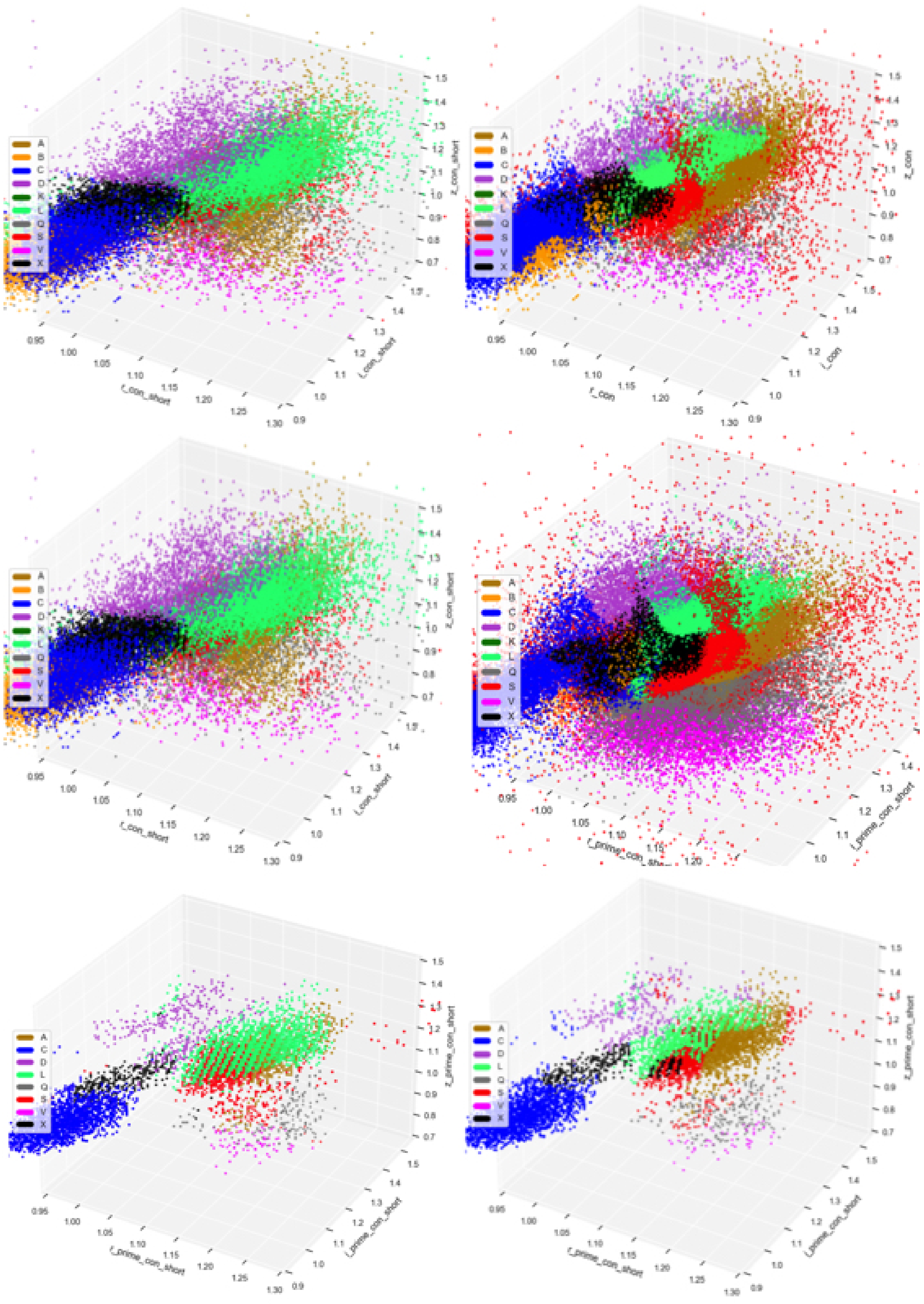


FIGURE 6.11: VAE (left) and SMOTE-ENN model predictions (right)(top to bottom): DeMeo, Sergeyev, Carvano SDSS.

Chapter 7

Implications of unsupervised learning for asteroid classes

7.1 Clustering to validate classes (or not)

In previous chapters I have shown that trained supervised learning models are reasonably successful at finding classes both at full (161) dimensionality and when reduced to pseudo-broadband. There is a meaningful decrement when moving from one dataset to another, however; for example, SMASS spectra can be classified to within 93% accuracy on their own test set, but only 62% when tested on MITHNEOS. Part of the problem comes from the tendency of S-class to over-attract objects that belong in A,K, L, and Q. When I changed SVM weighting and augmented to correct for this, in both cases the result was an over-correction; i.e., minority classes excessively gained datapoints. At this stage, it is difficult to avoid the suspicion that some of these classes are not well-defined to begin with.

So, what is a class? If a class label is intended to serve as a pointer to some unifying set of qualities possessed by a subset of the data, the label needs to be detachable from the data that gave rise to it. For asteroids, this unifying set of qualities refers to knowledge of how mineralogical features translate to reflectance spectra in the laboratory by means of absorption, emission, and scattering, a summation of which has been encoded in the label[1]. In that sense the label originates in a knowledge base extrinsic to the data at hand. Given the power implicit in this knowledge, why would we want to throw the classes out the window and entrust ourselves completely to statistics? The main motivation is to uncover latent structure in the data that may not meet with preconceptions. A secondary reason is to find

[1]I am going to pretend there are no conflicts or complexities between asteroid classes and mineralogical interpretations, or between historical assumptions and isotope distributions in the literature, and take the existing taxonomy as gospel.

out whether the proposed classes are reinforced by any latent statistical structure; to the extent that they are, the classes can be said to have a meaning intrinsic to the data. If that happens, the class is granted some integrity independent of preconceptions from the external knowledge base.

To think for a moment about what preconceptions could be involved here we need to remember that each successive classification system has been designed to build on the foundations of its predecessor, all the way back to the Tholen system created from 589 asteroids in eight bands. In 1984, this represented around 50% of the recorded population, with a declared bias in favour of objects belonging to families of interest (Zellner et al., 1985). The most recent public report on NASA's website claims 1,113,527 known asteroids as of July 2019. In 2020 alone nearly 3000 new near-Earth asteroids were recorded (Witze, 2021), pointing to the other obvious problem: sample size. For most of the million-plus known asteroids, high-quality spectra are not yet available. For the known data, classes described by Tholen have formed the basis for articulation of new classes as well as subclasses, but if Tholen's system had not existed and a million (or even, as here, a few thousand) asteroids had been observed first time round, would the categories have been the same? Where, for example, would the boundaries for Q-type be if Tholen had observed Q-types? We have seen how the Bus-Binzel system added classes and subclasses to handle new data and greater detail. The subsequent Bus-DeMeo system improved the classification by extending into the infrared, where the 2 μm feature enables better distinctions between types. Yet at every stage comes the potential for bias. This is not a criticism of the decision to privilege consistency, which is perfectly practical, but a caution that every such decision has consequences. To a newcomer it seems that there is a lot of adding of classes going on, but apparently very little taking away.

Here, I am simply using the available optical spectra to see what the statistics say about them via unsupervised learning. By plotting the data in either $r - i - z$ or PCA space, one can already see that the points do not sit in separate regions. Instead, there are unevenly distributed clouds of points that need to be marshalled into groups somehow. How many groups? That is also to be determined.

In this chapter I use a battery of statistical methods to try to find out how many groups of objects are detectable in the full spectra, the augmented spectra, the reduced-dimension spectra using PCA and pseudo-broadband, and finally the SDSS. For the latter I focus on the high-quality Carvano dataset in which some clustering is visible. Armed with this information, I then attempt clustering using a suite of complementary methods.

7.2 Method

The familiar datasets from previous chapters are explored here:

- Unaugmented full spectra 0.5 - 0.9 μm
- Augmented full spectra (VAE)
- First three PCs of unaugmented spectra
- Unaugmented pseudo-broadband g-r-i-z (denoted 'griz' hereafter)
- SDSS griz (Carvano, DeMeo, and Sergeyev datasets)

As described in Chapter 2 there are several clustering methods available, each having a different definition of a cluster. The choice of method depends on the characteristics of the data and one's knowledge of the relationship between its dimensions. In this case, the data are not in clear groups, nor are the classes particularly spherical, so k-means is not an ideal choice. However, its status as one of the most widely used clustering algorithms warrants its inclusion. I chose the GMM because it does not rely on clusters of uniform size or shape. HDBSCAN is selected because its documentation claims that for data where clusters are in unusual shapes/configurations it performs better than other algorithms (McInnes et al., 2017) and because, uniquely, HDBSCAN can refuse to cluster items that it judges to be noise. In addition to these methods, I did perform tests with spectral clustering (Ng et al., 2001), which is a method based on a similarity matrix between datapoints that allows for non-spherical and/or nested cluster shapes. I found it concerning that spectral clustering could not form a closed graph with the data, and as its results added no new information, I have excluded it from the reported results.

I have run a set of tests to assess the number of k-clusters and the number of Gaussian components in the data. I use multiple methods in order to find out whether the number of clusters remains consistent across different metrics. Informally, an identifiable optimum number of clusters is a necessary but not sufficient criterion for a robust clustering result. For k-means, I have used the elbow method with distortion, the Calinski-Harabasz Index, the silhouette score, and the Davies-Bouldin Index. For GMM I calculated the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) and the Jensen-Shannon distance as well as the silhouette method. I ran HDBSCAN iteratively to determine the optimal minimum number of points per cluster. I then visualised the clusters and performed some simple comparisons between clusters and classes.

7.3 Results

7.3.1 Full spectra clustering results

Method	Number of clusters
K elbow (distortion)	5
K Calinski-Harabsz	2
K Davies-Bouldin	2
K silhouette	2
GMM silhouette	2
GMM AIC/BIC	2
GMM Jensen-Shannon distance	2
HDBSCAN	2

Table 7.1: Number of clusters - full spectra

In Table 7.1 there is strong consensus for only two clusters in the data here. The elbow plot calculates five clusters, but a glance at figure 7.1 shows that curvature between k=4 and k=6 is slight, making the case for five clusters quite weak. However, it's not safe to conclude that there really are two clusters in these data, because it turns out that the GMM clusters show signs of instability.

In figure 7.2 for the silhouette on GMM, we can see that even for two clusters the average silhouette values sit at around 0.1, and above four clusters the silhouette score is negative, indicating that the objects are closer to the neighbouring cluster than to their assigned cluster. This is a bad sign. After clustering with GMM, the Jensen-Shannon distance between the each half of the dataset is clearly minimised at two clusters, but there is a large error on this value indicating instability. Figure 7.3 illustrates what is going on, with Gaussians plotted for two, three, and four clusters (left to right). Visualised in three dimensions, the Gaussians are fully superposed. This is, of course, inconsistent with even the primary division between S-complex and C-complex that we would expect to see. If stability were observed in the Jensen-Shannon distance between two versions of these data, it might be possible to conclude that some hidden structure exists in the high-dimensional data that is invisible in r-i-z. But the evidence of instability is there, so GMM can be said to fail.

The only algorithm that seems to work at all with these data is HDBSCAN, which offers an option for classifying points as noise. As seen in Figure 7.4, this model can only find the cores of C-complex and S-complex, with everything else considered noise. It is clear, then, that although classifiers could train successfully on labelled data at the full-spectrum level, unsupervised learning can only 'find' the two highest-density regions in the data, and only with one algorithm.

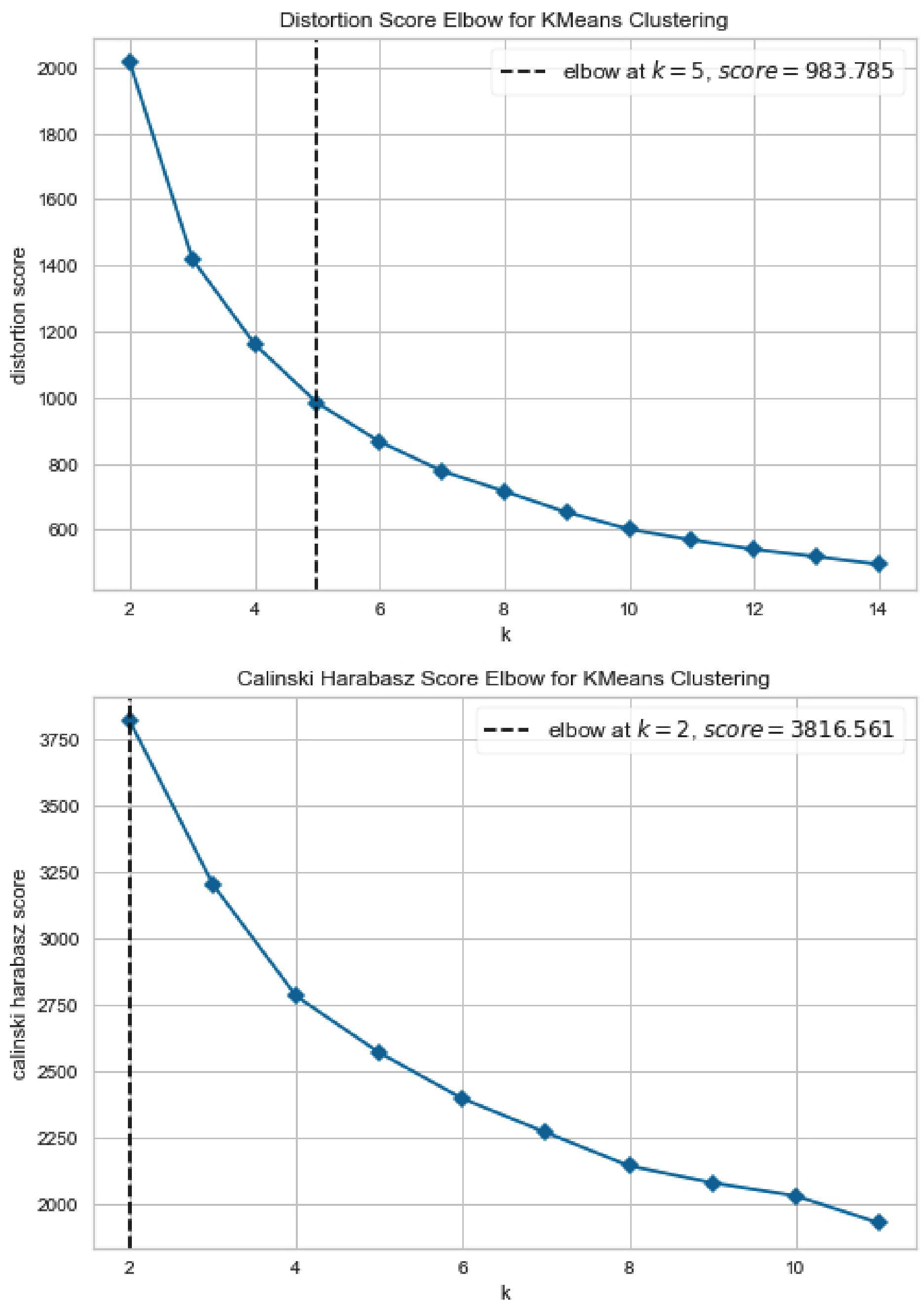


FIGURE 7.1: Determining number of k-means clusters for full spectra. Top: Elbow plot of the average SSE for each set of clusters showing an almost imperceptible preference for five clusters. Bottom: Calinski-Harabsz index results indicating preference for two clusters.

7.3.2 Clustering on spectra augmented with VAE

In Chapter 6 the VAE data turned out to be well-suited to classification with an SVM and offered the advantage of separating B, C, and X classes better than unaugmented data. When viewed in r-i-z space, there was no sub-structure introduced, unlike linear sub-structures that were seen when spectra were augmented

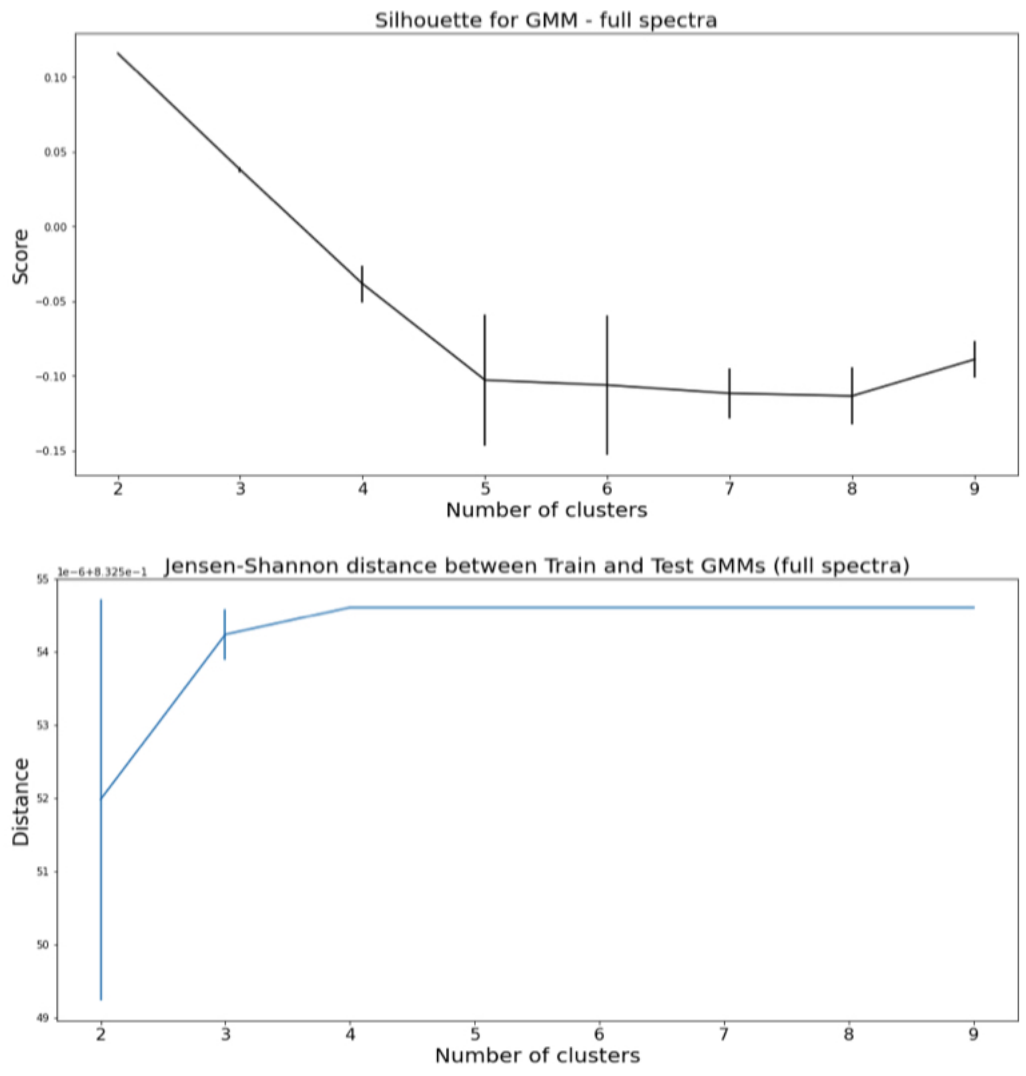


FIGURE 7.2: Determining number of Gaussian components for full spectra. Top: Silhouette of the full-spectrum GMM showing that the best result for two clusters is still poor at 0.1. Bottom: After splitting data 50/50, the Jensen-Shannon distance between each half is lowest for two clusters, but the large error bar shows that the clusters are unstable.

with SMOTE-ENN. However, the VAE data do appear in small clumps because each synthetic spectrum is drawn from a Gaussian distribution around its parent.

Table 7.2 indicates a mixed result for optimal number of VAE clusters. For k-means the measures of cluster integrity are split between two and six.

Six k-means clusters are shown in the left of figure 7.5 with the ten classes shown to the right. The only improvement on unaugmented K-means clustering is that the V-class objects (in orange on the left plot) have been assigned their own group, and some A-types have also been picked out. However, a lot of X-types have been grouped with S despite having different shapes in r-i-z, and there is no indication

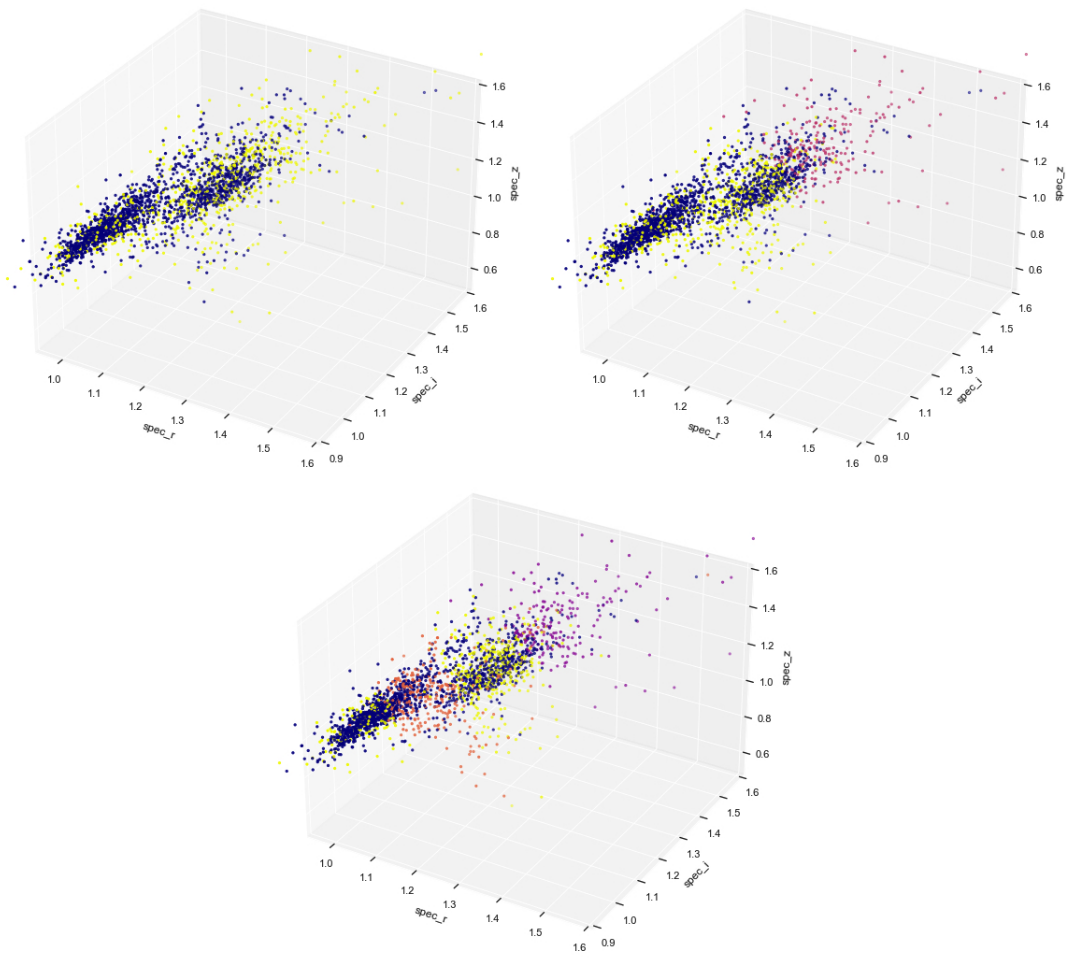


FIGURE 7.3: GMM fitted to two, three, and four clusters (clockwise from top left). The GMM is not able to cluster believably on these data.

that D will be seen as its own group. In fact, S-types have been split along a different axis by the clustering algorithm than by scientists who built the classification system.

What about GMM? The silhouette, AIC, and BIC all predict three clusters, but the Jensen-Shannon distance indicates that the most stable clustering happens at seven and eight clusters. Figure 7.6 shows this result. Note that the silhouette peaks at around 0.36 of a possible 1.0 at three clusters, which is not a large fraction and so does not inspire confidence.

In Figure 7.7 the three Gaussian components in the left plot bear no resemblance to visible structure in r-i-z space, and in the right plot the seven components are heavily intermixed, especially in the region between C-complex and S-complex: the part of the plot that would be expected to be well-differentiated. So this is also a failure.

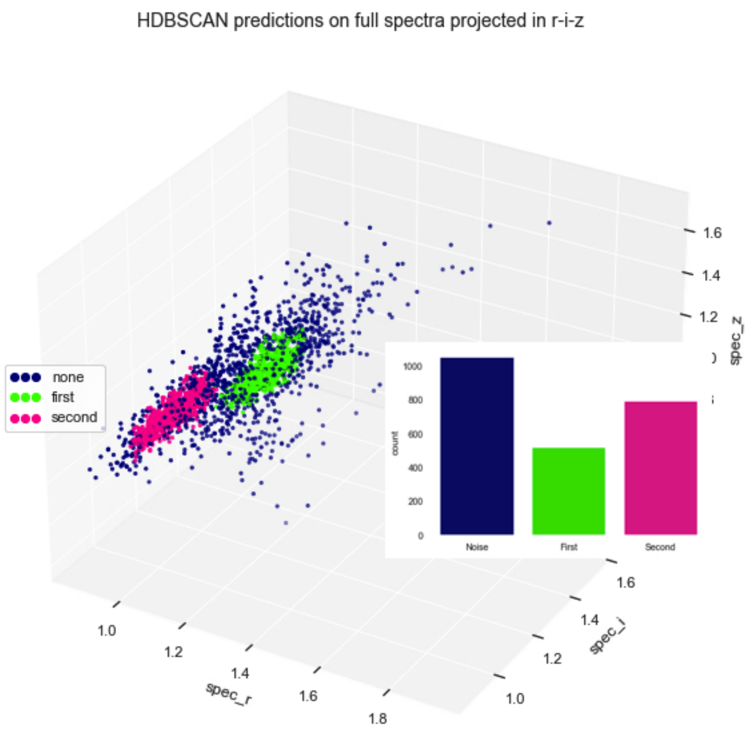


FIGURE 7.4: HDBSCAN plot of unaugmented spectra. Noise is the largest component, followed by C-complex, with the core of S-complex identified as the second cluster.

Method	Number of clusters
K elbow (distortion)	6
K Calinski-Harabsz	2, 6
K Davies-Bouldin	6
K silhouette	2
GMM silhouette	3
GMM AIC/BIC	3
GMM Jensen-Shannon distance	7,8
HDBSCAN	2

TABLE 7.2: Number of clusters - VAE augmented full spectra. There is a lack of consensus both within and between the different methods.

HDBSCAN again finds (roughly the same) two clusters, although in Figure 7.8 this time the noise level dominates, and the smaller cluster (S-complex) appears to be split in two even though it has been labelled as a single cluster.

The introduction of augmented spectra has only succeeded in increasing the noise. Considering that the Gaussian distribution around observed spectra is meant to act as a proxy for new data, a picture begins to emerge of a relatively smooth continuum in the r-i-z space. Of course, this is what is seen in the most inclusive SDSS data reported by Sergeyev, Alexey V. & Carry, Benoit (2021) in which classification results were so disappointing.

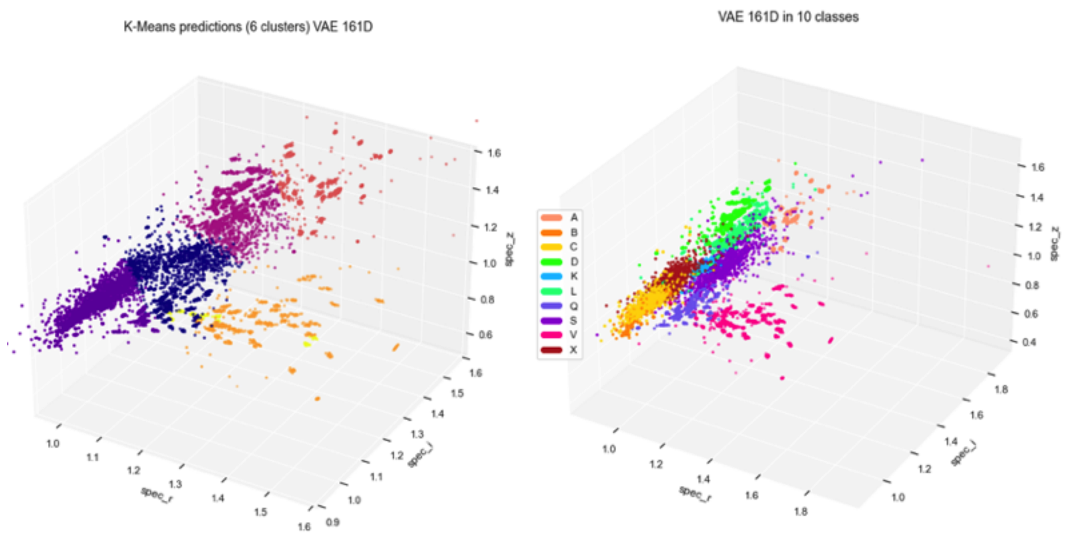


Figure 7.5: VAE spectra comparison of six k-means clusters (left) with ten classes (right) projected in r-i-z space. There is some overlap, but k-means is still making cuts across the i-axis as a major way of dividing the data.

7.3.3 Comparison of clustering on reduced dimensions: griz vs PCA

Method	PCA clusters	griz clusters
K elbow (distortion)	5	6
K Calinski-Harabsz	2	2
K Davies-Bouldin	2	2
K silhouette	2	2
GMM silhouette	2	2
GMM AIC/BIC	5	4
GMM Jensen-Shannon distance	2	2
HDBSCAN	2	2

Table 7.3: Number of clusters - griz vs PCA on full spectra. Reduced dimensions improve consensus on two clusters.

Because the VAE clustering resulted in increased noise, in this section the focus returns to unaugmented spectra. In Table 7.3 there is once again a mix of results for numbers of k-means clusters. PCA is predicted to have two, four, and five with an emphasis on two, and griz is predicted to have two, four, and six, also with an emphasis on two. HDBSCAN predicts two as usual. Once again there is a lack of consensus amongst the models.

For both PCA and griz, HDBSCAN clusters are consistent with what was found for spectra. The improvement comes in a smaller amount of noise, especially for PCA (see Figure 7.10). However, only two clusters are found by this method, consistent with many of the results in Table 7.3.

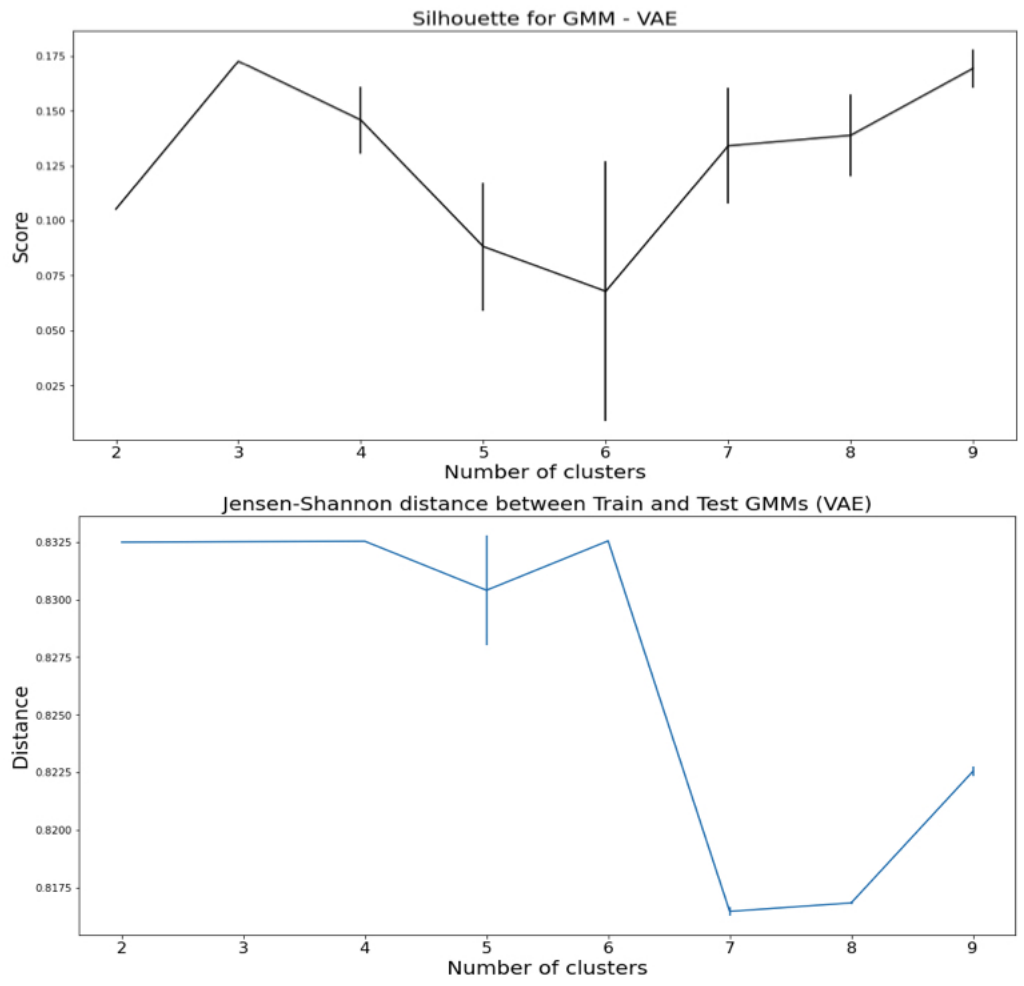


FIGURE 7.6: For GMM, the VAE silhouette plot (top) predicts three clusters, but the most stable clustering is found at either seven or eight clusters, where the Jensen-Shannon distance between two halves of the data is the smallest (bottom). It now remains to determine by eye which number of clusters is most realistic for these data.

Figure 7.9 shows that there is no argument for using k-means here. Because most of the variability is captured in the first principal component, k-means will 'stack' the classes along the axis of PC1. When this effect is projected into r-i-z space the effect is a series of diagonal stripes across the data that correspond poorly to classes.

However, things are a little better for the GMM. Figure 7.11 compares the results of clustering on PCA versus clustering on griz versus recorded classes. The chosen numbers of components represents the optimal results indicated for that dataset (where we are looking at more than two clusters). PCA does a better job of separating C from X and generally best resembles the class distribution, which is to be expected given that classes were allocated based on principal components in the first place. There is no indication of a V-class or a D-class here even though both of these classes are easy for an SVM to find. This is as good as it gets

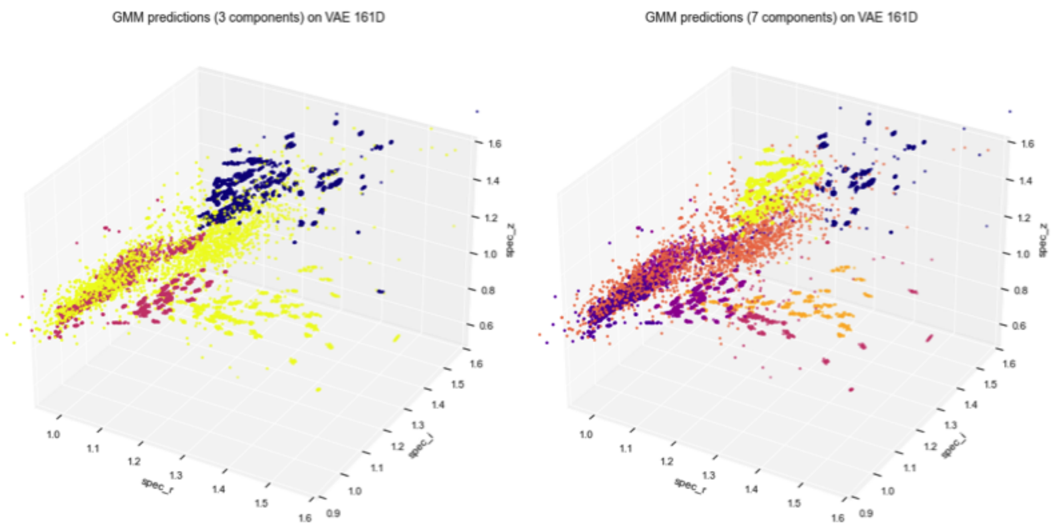


FIGURE 7.7: Comparison of Gaussian mixture model on VAE spectra for (left) three components and (right) seven components. Both plots show a failure to separate C- and S-complex, the most fundamental division in types

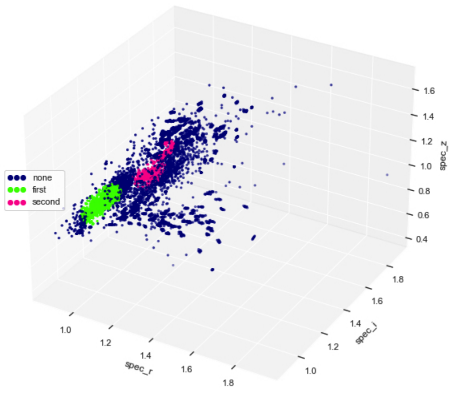


FIGURE 7.8: HDBSCAN on VAE spectra projected in r-i-z space. The perceived noise component now dominates the data, with the smaller (S-complex) cluster being divided.

for unsupervised learning in terms of ‘finding’ classes in the best data that are available.

7.3.4 Clustering the SDSS

Of the three SDSS datasets, the most amenable to clustering is the ‘pure’ Carvano dataset of 10487 objects. As roughly sketched in Figure 7.12 these data appear distributed in five clumps, or perhaps four if the sparsely populated A-class at the extreme end of all three bands is treated as noise. The question is, will machine learning see it the same?

It turns out that SDSS broadband data suffer even more than spectra, griz, or PCA data when it comes to finding the true number of clusters, which could be

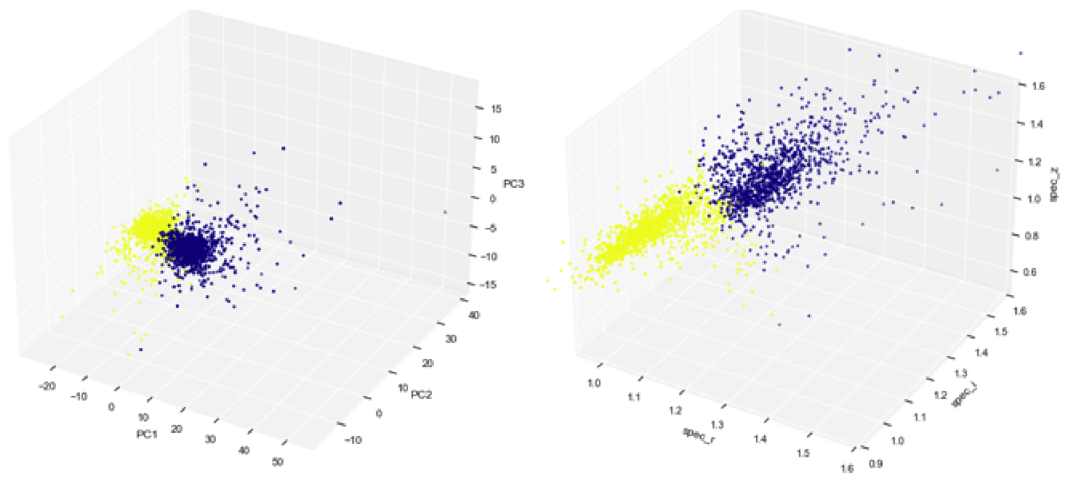


FIGURE 7.9: Two k-means clusters on PCA data in PC-space (left) and r-i-z space(right). Even in PC-space, there is very little structure to the data from which to make clusters. The boundary drawn by k-means is too simple to capture the differences in C- and S-complex that we observe in r-i-z space.

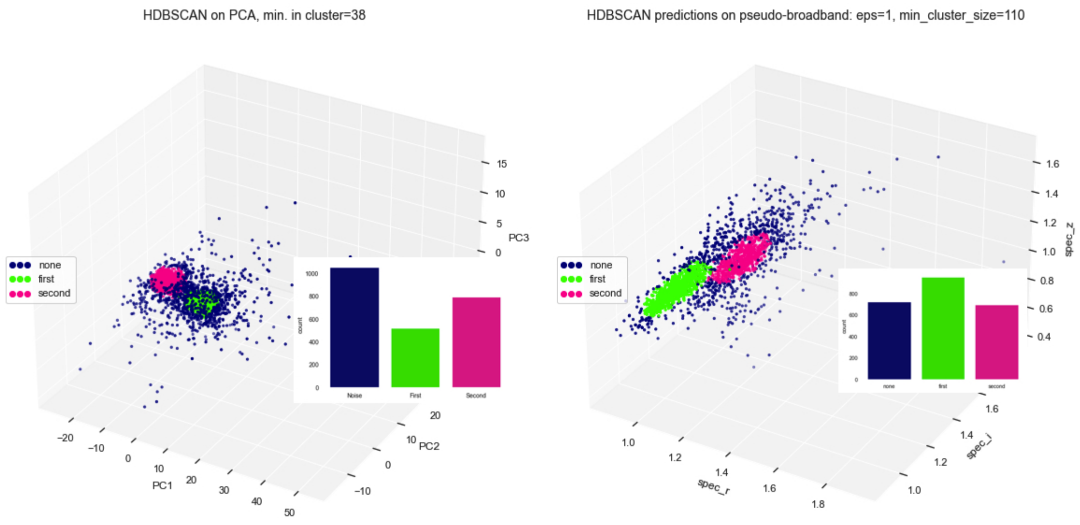


FIGURE 7.10: Comparison of clusters found with PCA (left) and griz (right) showing that there less noise in the HDBSCAN clusters when using griz data. The results of Chapter 4 found that griz was preferable to PCA for classification, and this also seems to be the case for clustering with HDBSCAN.

anything between two and seven depending on the method followed (see Table 7.4). By now it is no surprise that k-means has trouble with the data, but the fact that HDBSCAN still finds only two clusters with a lot of noise is worrisome, because two of the outlying groups shown here (D, V/Q) are relatively dense with gaps maintained between themselves and the main complexes. As for GMM, there are only two clusters found by the silhouette method, but BIC predicts eight. The latter is important because BIC penalises the addition of new classes.

Because the Carvano system accounts for mixed classes, we can examine the ‘pure’ classes only. There is a marked improvement of HDBSCAN’s ability to cluster on

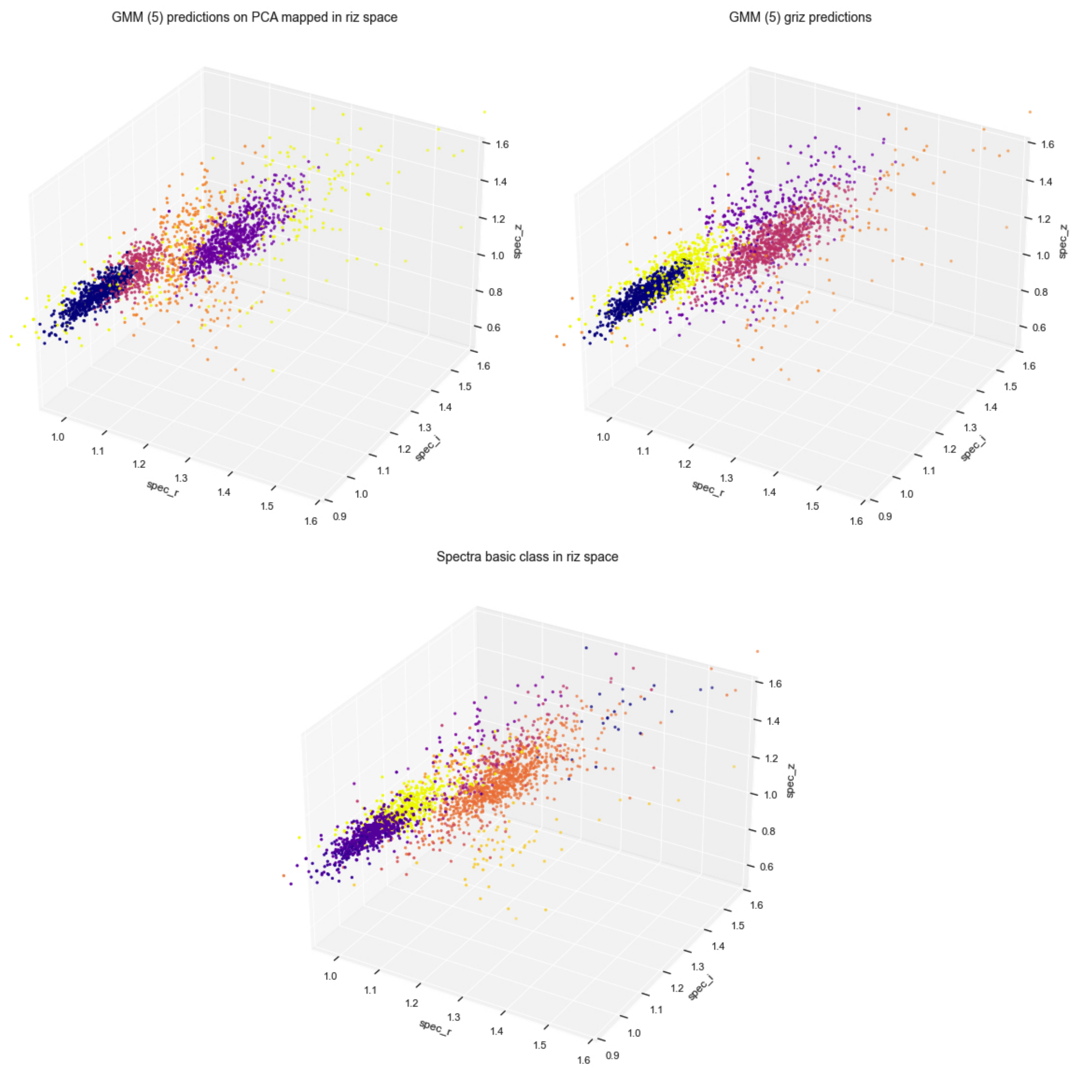


FIGURE 7.11: Visual check on PCA vs. griz clustering results. Top left: Five GMM components on PCA spectra projected in r-i-z space. Top right: Five GMM r-i-z clusters. Bottom: Ten classes in r-i-z. The GMM seems to 'see' the C/X distinction in PCA space, but there is no indication of a D or V cluster despite classifiers performing well on these classes.

the pure Carvano classes when compared to the larger DeMeo dataset with its inclusion of classes B and K, seen in Figure 7.14. The probabilistic nature of the Carvano classification system helps because it permits a flexible approach to the selection of groups of objects without resorting to adding new subtypes.

To get to the bottom of the discrepancy between BIC and other methods, the eight-cluster Gaussian mixture model is shown in the centre of Figure 7.15. This model produces clusters with some fundamental differences to classes. It breaks C- and S-complexes into five rather artificial-looking subsets, only one of which corresponds roughly to a class (B). The model also interprets D-class as part of a larger distribution around the central S-complex. The four-cluster Gaussian

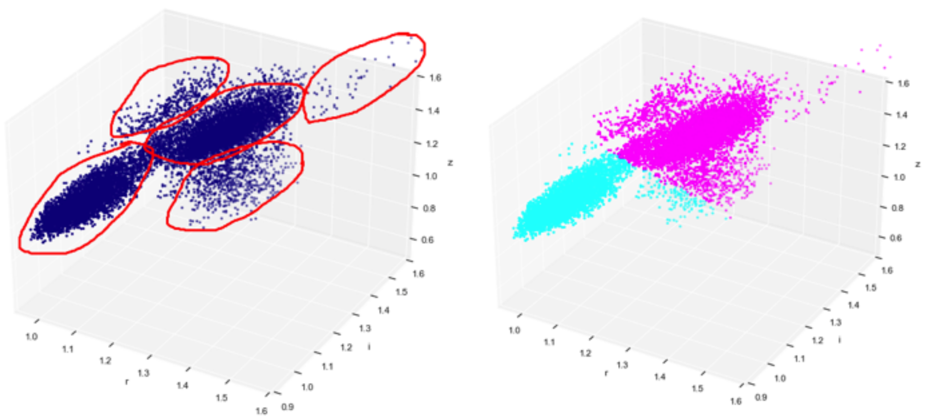


FIGURE 7.12: Left: rough sketch of the clusters one might expect from the Carvano SDSS data in r-i-z. Right: k-Means with two clusters is completely unsuitable for the data.

Method	Number of clusters
K elbow (distortion)	5
K Calinski-Harabsz	2
K Davies-Bouldin	3
K silhouette	2
GMM silhouette	2
GMM AIC/BIC	8
GMM Jensen-Shannon distance	2
HDBSCAN	2

TABLE 7.4: Number of clusters - SDSS Carvano

(left) consists of C-complex, S-complex, Q/V as a group, and once again the large, diffuse distribution encompassing D, L and some S- and Q-types. It seems that GMM and HDBSCAN may be finding the same thing after all, only the GMM model is not ‘allowed’ to declare a point as noise; instead, it creates a large, diffuse distribution encompassing what appear to be multiple groups.

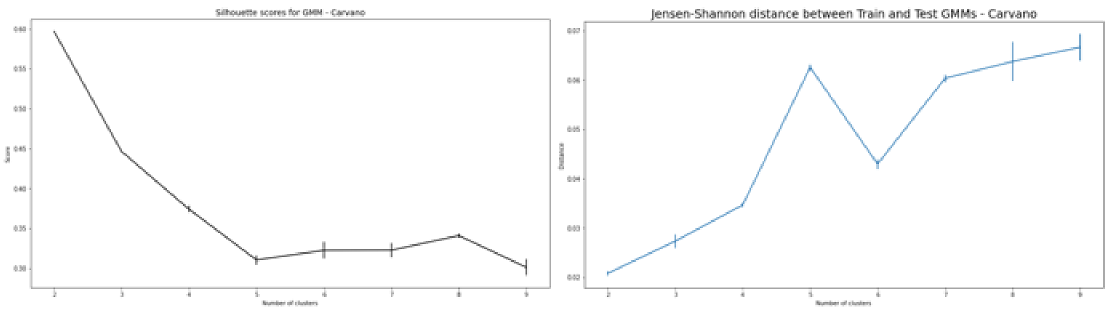


FIGURE 7.13: Number of Gaussians for SDSS Carvano data. There is no consensus.

7.4 Discussion

Usually, people perform unsupervised learning as part of a discovery process prior to doing other work. I have done everything backward, using unsupervised learning as an independent test of classification. I hoped this method would shed light, but it has produced mostly heat instead. In interpreting the results, I find that most of the classes fail to be recognised; but I also find that unsupervised learning fails to capture features that appear important.

Depending on the algorithm used and even on the method of determining number of clusters, the results of unsupervised learning can be utter nonsense. I have tried

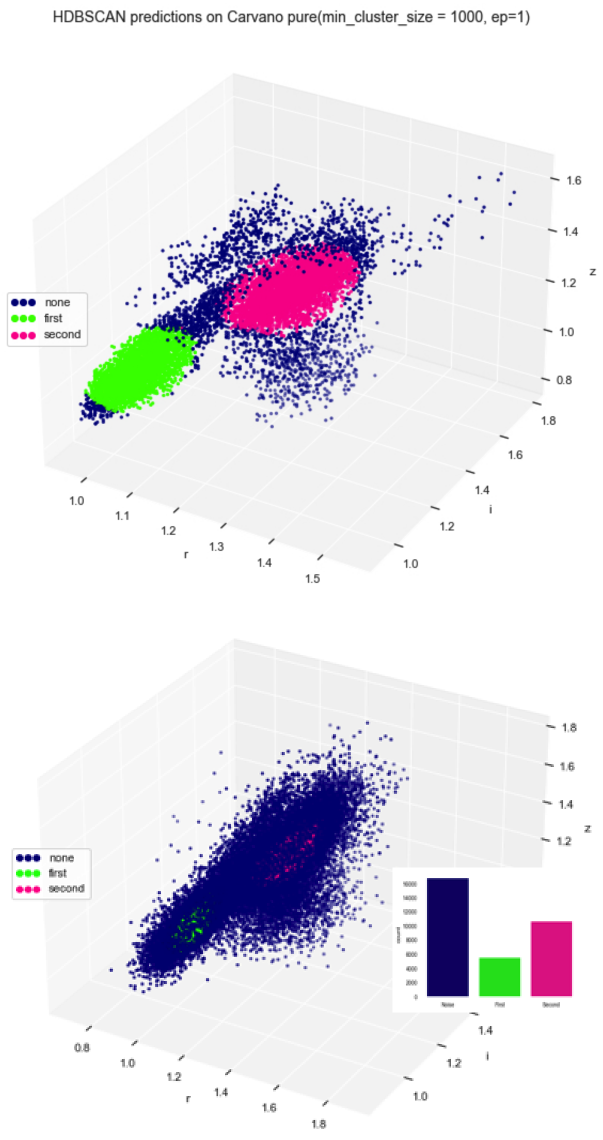


Figure 7.14: Comparison of HDBSCAN results for: (top) Carvano 'pure' class data, with regions belonging to D, V, Q, and A considered noise; (bottom) DeMeo data, where the two main complexes are there, but the noise is overpowering.

to select methods with as much care as possible and have presented only results that I consider robust. Statistically, apart from C-complex and S-complex, the classes do not seem to be 'real'. It is possible to squeeze out clusters roughly corresponding to C, S, Q/V, and X with PCA and a GMM, but not K, L, D, or A. In considering PCA results one must recall that classification systems are founded on PCA, making even this limited success less than independent (although PCA, of course, does refer to 'real' variability).

For unsupervised learning with these data, augmentation degrades the result rather than improving class imbalance. This is an ironic result, given that the primary reason I built the variational autoencoder was to balance the sizes of expected clusters and correct implicit algorithmic bias that favours large classes. Since the autoencoder produces credible spectra of each class, I attribute the failure to the

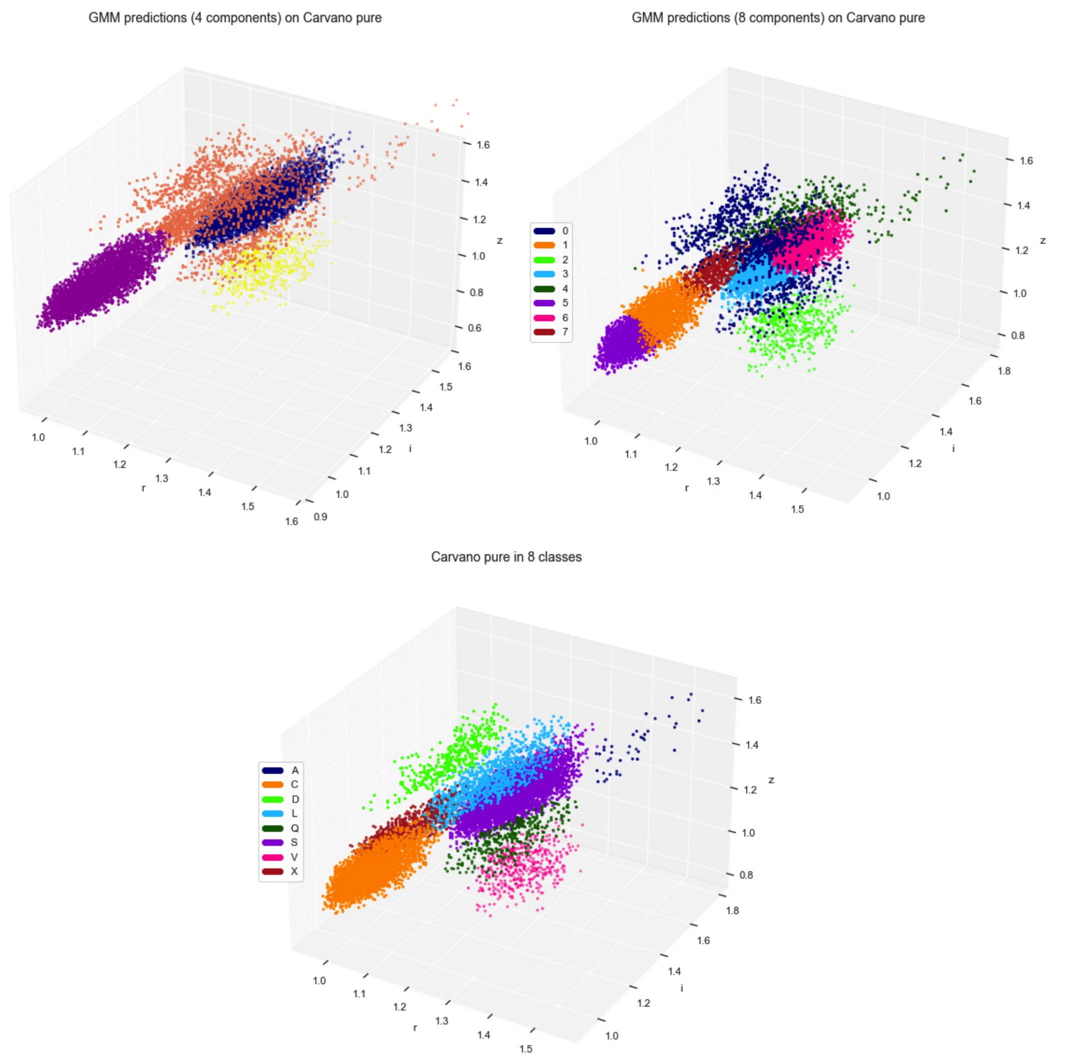


FIGURE 7.15: Carvano 'pure' class data according to (top left) four Gaussians, (top right) eight Gaussians, (bottom) recorded class. The Gaussian mixture interprets D-class as a member of a larger distribution around the S-complex.

introduction of more noise to an already-noisy dataset, which can be seen in the HDBSCAN results.

There is clearly a use for dimensionality-reduction despite the power of modern computing making light work of 161 dimensions for classification. With both PCA and griz simplification, noise is reduced, and visualisation becomes easy. PCA works well for this purpose, but as discussed in Chapter 4, griz is more visually intuitive and also allows easy comparison between datasets.

When it comes to the SDSS, though, even the cleanest dataset only shows C-complex and S-complex. Unsupervised learning does not 'find' the visually distinct D-types, and it groups V together with some of Q. Does this mean that unsupervised learning has failed, or does it mean that D, K, L, and some Q-types are part of a more general pattern of variation around the central S-complex? It is hard to say. With only around 2000 spectra at 161 dimensions the GMM understandably struggles to find discrete clusters. But even after PCA, the GMM still only finds a single large component surrounding the S-types in the place of four putative classes. The individual classes described by such a Gaussian do not appear to have a reality independent of the classification system.

When classifying asteroids purely by optical spectra/colour, their dominant distinguishing feature is slope. Section 1.6 discussed the physical reasons for degeneracy in asteroid spectral slope. Phase reddening occurs on short timescales related to orbit, and in some cases it can be corrected; but slope increase due to space weathering from the solar wind and cosmic ray bombardment, etc. works in opposition to the slope decreases associated with surface refreshment from collisions and gravitational interaction. Disentangling these different sets of causes and effects is not possible. Clustering in the optical is to some extent a futile effort due to data appearing as a continuum in feature-space. For that matter, classifying in the optical is equally limited, especially when optical slope can be physically unrelated to surface composition for some objects.

Is clustering of no value, then? In order to progress, it is necessary to break the degeneracy by including new physical parameters. The NIR has been incorporated into the Bus-DeMeo classification system and is included in the new Mahlke, M. et al. (2022) system, together with albedo – which has been absent from systems since Tholen (1984). Mahlke et al. have used clustering to good effect in their work, but they treat it as an early step in the process of defining a classifier much as Tholen did (rather than as a test of the classification system, as I do here). They have also taken a number of specific steps to process their data to avoid some of the pitfalls apparent in this thesis:

(a) They use NIR preferentially, with most of their data going up 0.8 μm and only a small fraction having data below 0.6 μm.

(b) Instead of normalising at 0.55 μm (which necessarily reduces variance in this region) they use a novel method to estimate normalisation constants.

(c) They use a probabilistic variant of PCA that handles missing data, meaning that they can have spectra of varying lengths and ranges and still work out the dominant variances.

(d) They specify the number of Gaussians required at 20 and then manually combine groups based on prior knowledge of types.

(e) They scale albedo carefully to ensure a good balance of spectra vs. albedo in setting classification boundaries.

The Mahlke system represents a significant evolution of prior classification systems, in particular because of the inclusion of albedo. But it is not an independent check on any system.

7.5 Conclusion

In this chapter I have shown that even for PCA the classification system appears to be reaching when it goes beyond C-complex and S-complex. It is true that a GMM can be trained on four components to produce a fair representation of visually recognisable clusters, but only the C-complex and S-complex clusters match the recorded classes (with X folding into C). PCA can produce a five-cluster model that reflects classes a little better, which is to be expected given that the classification system has been designed based on PCA. There is no case to be made for A, B, D, K, L, or Q as distinct groups based on these results.

When addressing the question of classification of asteroids it is probably most important to consider the purpose of the classification. If we are trying to understand solar system formation history then large numbers of objects are more important than the details of mineralogy; but if we are thinking about mining an asteroid for its resources, the details of a specific object supercede the statistics of types. It is already widely acknowledged in the field that asteroids are distributed in a continuum in colour space, so in that sense they are never going to be suited to clustering methods. Yet without any independent verification of classes, the taxonomic system will carry the biases of the domain knowledge that gave rise to it.

In working on clustering, I developed progressively greater faith in HDBSCAN as a reliable method, until I came to the Carvano data and found that even after hyper-parameter tuning it could not ‘see’ what seemed obvious to the eye. It may well be that a more skilled machine-learning practitioner could squeeze more value out of HDBSCAN for these data, but if that is the case then this is a method poorly suited for use by beginners (which at the time of this writing describes many astronomers).

Chapter 8

Summary and conclusion

The questions driving this research were:

(a) Can a machine-learning classifier detect the major asteroid classes in the Bus-Binzel and/or Bus-DeMeo taxonomies at spectrum level, at photometric colour level, or at both?

(b) Are these classes 'real' from the point of view of unsupervised learning algorithms?

8.1 Summary of main results

In Chapter 4 I examined the consistency of the Bus-Binzel asteroid classification system by using a support vector machine on detailed spectra as well as on their principal components. I trained and tested within and between four datasets (SMASS, MITHNEOS, S3OS2 and INT). The SVM trained with spectra performed around 3% better than the SVM trained with principal components. In a combined dataset, major classes could be classified to 85% accuracy in wavelength range 0.5 - 0.9 μm, and to 90% in range 0.435 - 0.925 μm. Failures showed a pattern: class K was difficult to separate from L, Q from S, and B from C. The SVM was able to pick out ambiguous objects better when applied to a single dataset; but I concluded that it must overfit to do so, or it would not fail on the combined data. More B-type samples might have solved the problem with that class, but an eyeball check of Q-types showed that Q is not well-defined to begin with. There is doubt that classes B, K, L, and Q are anything other than outlier members of the major complexes C and S. I found that none of the thirteen subclasses identified by the taxonomy's authors were convincingly distinguishable from their parent classes.

Having established the best-case scenario, in Chapter 5 I converted the data to a simplified 'pseudo-broadband' to make it compatible with SDSS photometry. As a result the SVM only lost 5% of its accuracy, but the aforementioned B and K completely ‘disappeared’, as did A, whilst Q is nearly always misclassified. Photometric classification therefore has a ceiling at about 80% and tells us that only classes C, S, L, S, V and X are ‘real’ (if imperfectly separable) at that resolution. Given that SDSS-based taxonomies use all these classes, the question became: what happens when the SVM meets SDSS data? An examination of objects-in-common indicated that only 71% have the same assigned class in photometric catalogs that they do in spectrum-based catalogs, with ambiguity concentrated amongst C/X, B/X and L/S particularly. This fuzziness in class targets placed limits on what could be achieved by the SVM and hinted at difficulties to come.

Indeed, when I trained on spectrum-based pseudo-broadband and tested on SDSS values in the Carvano et al. (2010) and DeMeo & Carry (2013) catalogs, accuracy deteriorated to 67% and 62% respectively. I proposed that the partial cause was an apparent offset in each of the $r - i - z$ bands’ brightness despite using the SDSS band centres and full-width-half-maxima to define pseudo-broadband values. I was able to recover some of these losses by using objects common to both spectrum and SDSS datasets to define a correction factor. After applying the correction, accuracy improved to 88.5% on Carvano data and 73.9% on DeMeo. The ‘disappeared’ class A gained a small number of objects but B and K had no correct predictions. Overall, the distribution of SDSS data in r-i-z space was found to be excessively large relative to the spectrum pseudo-broadband, which was not remediable. Ultimately, A, B, K, L, Q are not verifiable classes in the SDSS.

Because of the observed tendency of some minority classes to disappear from predictions, Chapter 6 related the quest for an unbiased classifier by balancing the numbers of classes to remove the disproportionate effects of S and C on training. I compared two methods: SMOTE-ENN and a VAE. SMOTE-ENN data can be classified to accuracy of 91% as spectra and 88% as pseudo-broadband. VAE can achieve 92% accuracy at spectrum level and 90% at pseudo-broadband. Both augmentation methods can recover all the ‘missing’ classes, but only to low levels of accuracy, and they continue to mistake L for S. However, I found that VAE resamples very closely to each of its parent datapoints, and SMOTE-ENN introduces unacceptable substructure that is visible in 3D plots of pseudobroadband.

The performance of the augmented models collapsed when testing on the SDSS. Accuracy of the augmented models is $30 - 40\%$ lower on the SDSS than on pseudo-broadband. The SMOTE-ENN model’s substructure created highly overfitted class boundaries that became obvious when examining plots of its predicted classes on SDSS data. The VAE was able to correct for some class imbalance, especially in the C-complex, but its accuracy results ran to 58% at best. The reason for much lower accuracy scores is the fact that the classes which I have already asserted are not 'real' (i.e., B, K, L, Q) make up a very substantial fraction of the sample after augmentation, and they remain at least as unreal as they were when the best data was used. These objects would naturally raise the failure rate of the classifier.

In Chapter 7 I went on to show that unsupervised learning cannot find any verifiable groups in spectrum data apart from the C-complex and S-complex, which were established over 50 years ago. K-means failed to produce any credible clusters. Gaussian mixture models showed indications of instability when subjected to statistical tests, and the best Gaussian mixture models produced clusters that correspond roughly to S, C, and X, with everything else being treated as part of some larger, diffuse distribution around the central classes. Trying to emphasise variance in the data by working with principal components did not meaningfully change this outcome. Similarly, HDBSCAN recognised only the cores of C and S and treated everything else as noise.

When it came to clustering on the SDSS, the GMM preferred to separate the data into only two clusters; however, these did not match the observed C-complex/S-complex groups! If forced to create four or five clusters, the GMM treated C-complex as a single group and broke S-complex into a core cluster plus a smaller cluster encompassing some Q, V, and a little bit of S, plus a larger cloud around S-complex that includes D, A, and some Q. It therefore does not find correspondences to classes apart from the two major complexes. Again, HDBSCAN found only the cores of S-complex and C-complex, with everything surrounding them relegated to noise. The visual appearance of clusters in r-i-z space–particularly D and V classes in Carvano SDSS–does not matter to the algorithms.

At some point, it must be decided what is useful about all these experiments. Based only on data spanning a 0.4 μm range in the optical, I am not going to be the person to claim that classes A, B, K, L, and Q are not real! But I will say that the taxonomy does not hold up in this wavelength regime for these classes. With the asteroids, we have a sample of spectra that we know are biased towards larger/closer (brighter) and more orbitally interesting objects (for example, the V-types and the Trojans), but the data quality are very good. We also have a large amount of not-so-good but unbiased data, exemplified by the Sergeyev dataset. It has long been an assumption of the big data approach that more data is the answer to all things, but if the data quality is low, we do not learn very much.

In practical terms, if one were using machine-learning to select follow-up targets there could be some benefit in tweaking the feature-space with augmentation to widen the net for rare objects. I have shown that simple weighting of the classifier to place more emphasis on the minority class can lead to over-selection for that class. SMOTE-ENN (for all its faults) also accomplished this enhancement of small classes. There are situations it would be better to select some unwanted objects and later discard them than to miss the rare object entirely — when searching for metallic asteroids, for example. But the real difficulty in that case lies in separating X from C, and then finding the metallic subtypes within X. The existence of X subtypes in modern taxonomies is misleading in that we do not actually have albedos for most of these objects; as described in 1.3 all metallic asteroids are supposed to live in the same visible feature-space as their high-albedo relatives, but this assignation is based on a small sampling. Luckily, Mahlke, M. et al. (2022) recently introduced a taxonomy that splits X-types according to albedo; their sample size is larger than Bus-DeMeo's by an order of magnitude, so their classifications should supersede or at least complement the Bus-DeMeo system. And, of course, there is the little problem of collecting albedos for objects that we wish to classify.

Machine learning can be good for showing up hidden relationships and acting as a check on our own implicit biases; but it is also perfectly capable of introducing biases of its own. Most machine learning methods discussed in this thesis are relatively easy to implement, and techniques like cross-validation in a pipeline enable some automation of parameter tuning. Still, like any learning process, machine learning involves trial and error — this is true on the part of the operator as well as the machine. In this work, given the abundance of statistical tools to handle asteroid classification and clustering and their power to fit to the data they are fed, the quality of the data (and its labels) is the limiting factor. The absence of NIR and albedo in this work has degraded the quality of data; this is the main issue that still hangs unresolved.

8.2 Future work

Overall, the population size of known asteroids has now reached the point where big data approaches are needed, but this project has probably come a little too soon as Gaia spectra of asteroids were unavailable. Some machine-learning methods that I attempted (such as the bridge model) failed due to lack of data. This is only a temporary problem.

The 'big data' that I worked with here was low-resolution SDSS photometry. My expectation is that new data from Gaia will supercede the SDSS, and an analysis of clusters versus classes in the Gaia data would be interesting to pursue to find out whether the classes (or clusters) have more integrity when data quality and quantity are both increased.

This work was performed entirely in the optical regime because the project originated with those data through my AMC placement. As the results of the work became clearer over time, I had occasion to regret restricting myself to the optical because of the 1 and 2 μm features that are incorporated into the Bus-DeMeo system and so implicitly affect optical classification using this system. I am therefore unable to say whether classification or clustering results would improve with inclusion of the 2 μm feature especially, although of course the SDSS photometry does not cover this regime anyway, nor does Gaia. Instead, a multi-wavelength approach is required.

One of the limitations of this project has been the paucity of high-quality spectra for training the models. For its ability to generate synthetic spectra, the variational autoencoder has been one fruitful outcome, and it could remain useful in the future because even if the size of the asteroid dataset increases exponentially, class imbalances are likely to persist. However, I have not yet explored the full capabilities of this technique. One niggling concern is the tendency of my VAE to produce overly smooth spectra. It would be worth the effort to explore the capability of the VAE to generate more variable random noise and thereby to simulate more realistic spectra.

Rather than attempting to refine the unsupervised learning methods used in this thesis, I would address the new Mahlke system instead. An improved version of K-Means as developed by Lisboa et al. (2013) facilitates a rigorous determination of appropriate cluster centres for the data. This algorithm, as well as HDBSCAN and GMM models, could provide some independent checks on the Mahlke boundaries that were assigned based on prior knowledge of classes. Given that the Mahlke training spectra have significant overlap with the training spectra used in this work, the results of such a study could also offer insight to the response of the classification system to the addition of albedo as a feature. Intuitively, one would expect albedo to hold its value more readily than spectral slope in the face of the environmental stressors discussed previously. If albedo could demonstrably break any of the degeneracies associated with spectral slope, the classification system would be greatly improved. Testing with unsupervised learning could tell us whether the new system is founded on statistical bedrock.

Bibliography

Aerolite 2022, NWA 10023 9.3g., https://aerolite.org/product/nwa-10023-9-3g/

Akaike H., 1974, IEEE Transactions on Automatic Control, 19, 716

Altman N., Krzywinski M., 2018, Nature Methods, 15, 399

Benner L. A. M. ; Busch M. W. . G. J. D. . T. P. A. s. b. o. . M. J. L., 2015, Asteroid Models from Multiple Data Sources. pp 166–8

Berba P., 2020, Understanding HDBSCAN and density-based clustering, https://towardsdatascience.com/understanding-hdbscan-and-density-based-clustering-121dbee1320e

Binzel R. P., et al., 2019, Icarus, 324, 41

Bishop C., 2006, Pattern Recognition and Machine Learning. https://www.microsoft.com/en-us/research/publication/pattern-recognition-machine-learning/

Bottke W., D. V., et al. M. D., 2012, Nature, 485, 78

Brasser R., Mojzsis S., 2020, Nature Astronomy, 4, 492–499

Breiman L. F. J. O. R., Stone C., 1984, Classification and regression trees. Wadsworth Books

Brownlee J., 2019, Bagging and Random Forest Ensemble Algorithms for Machine Learning, https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/

Brownlee J., 2021, Code Adam Optimization Algorithm From Scratch, https://machinelearningmastery.com/adam-optimization-from-scratch/

Brunetto R., Loeffler M. J., Nesvorný D., Sasaki S., Strazzulla G., 2015, Asteroids iv, pp 597–616

Burbine T. H., 2000, Forging asteroid-meteorite relationships through reflectance spectroscopy, `https://dspace.mit.edu/handle/1721.1/9170`

Bus S. J., Binzel R. P., 2002a, Icarus, 158, 106

Bus S. J., Binzel R. P., 2002b, Icarus, 158, 146

Buseck A., 2022, Pallasites, `https://meteorites.asu.edu/stony-iron-meteorites/pallasites`

Campello R., Moulavi D., Sander J., 2013, in Pei J., Tseng V. S., Cao L., Motoda H., Xu G., eds, Advances in Knowledge Discovery and Data Mining. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 160–172

Carroll B. W., Ostlie D. A., 2014, An Introduction to Modern Astrophysics, Second Edition. Pearson Education Limited

Carter T., 2011, An introduction to information theory and entropy, `http://astarte.csustan.edu/~tom/SFI-CSSS/info-theory/info-lec.pdf`

Carvano J. M., Hasselmann P. H., Lazzaro D., Mothé-Diniz T., 2010, Astronomy and Astrophysics, 510

Cburnett 2022, Neural networks: perceptrons, `https://en.wikipedia.org/wiki/Artificial_neural_network#/media/File:Artificial_neural_network.svg`

Chapman C. R., Johnson T. V., McCord T. B., 1971. p. 51, `https://adsabs.harvard.edu/full/1971NASSP.267...51C`

Chaudhary K., 2020, Variational Autoencoder, `https://github.com/kartikgill/Autoencoders/blob/main/Variational%20Autoencoder.ipynb`

Chawla N., Bowyer K., Hall L., Kegelmeyer W., 2002, J. Artif. Intell. Res. (JAIR), 16, 321

Chollet F., 2016, Building Autoencoders in Keras, `https://blog.keras.io/building-autoencoders-in-keras.html`

Chollet F., et al., 2015, Keras, `https://github.com/fchollet/keras`

Coleman G. A. L., 2021, Monthly Notices of the Royal Astronomical Society, 506, 3596

Collins G. W., 2004, The Foundations of Celestial Mechanics. Pachart Publishing House, pp 93–106

D. Vokrouhlicky W. F. Bottke S. R. C. D. J. S. T. S. S., 2015, The Yarkovsky and YORP effects. p. 506

Davies D. L., Bouldin D. W., 1979, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1, 224

DeMeo F. E., Carry B., 2013, Icarus, 226, 723

DeMeo F. E., Binzel R. P., Slivan S. M., Bus S. J., 2009, Icarus, 202, 160

DeMeo F., C.M. A., K.J. W., C.R. C., R.P B., 2015, The Compositional Structure of the Asteroid Belt. The University of Arizona Press, pp 13–31

Delbo' M., Mueller M., Emery J. P., Rozitis B., Capria M. T., 2015.

Drakos G., 2020, Davies-Bouldin Index vs Silhouette Analysis vs Elbow Method Selecting the optimal number of clusters for KMeans clustering, `https://tinyurl.com/yzucp6ks`

Drakowska J., Alibert Y., Moore B., 2016, A&A, 594, A105

Drazkowska J., Alibert Y., Moore B., 2016, Astronomy & Astrophysics, 594, A105

Elvis M., 2012, Nature, 485, 549

Fernández J., Ip W., 1984, Icarus, 58, 109

Ferreira, J. F. Tanga, P. Spoto, F. Machado, P. Herald, D. 2022, A&A, 658, A73

Friedman J. H., 2001, Annals of Statistics, 29, 1189

Gaffey M. J., Bell J. F., Brown R., Burbine T. H., Piatek J. L., Reed K. L., Chaky D. A., 1993, Icarus, 106, 573

Galler B. A., Fisher M. J., 1964, Commun. ACM, 7, 301–303

Glavin D. P., Alexander C. M., Aponte J. C., Dworkin J. P., Elsila J. E., Yabuta H., 2018, in Abreu N., ed., , Primitive Meteorites and Asteroids. Elsevier, pp 205–271, doi:https://doi.org/10.1016/B978-0-12-813325-5.00003-3, `https://www.sciencedirect.com/science/article/pii/B9780128133255000033`

Gomes R., Levison H., Tsiganis K. e. a., 2005, Nature, 435, 466–469

Graves K. J., Minton D. A., Molaro J. L., Hirabayashi M., 2019, Icarus, 322, 1

Hasselmann P. H., Carvano J. M., Lazzaro D., 2011, SDSS-based Asteroid Taxonomy V1.0. EAR-A-I0035-5-SDSSTAX-V1.0., `https://sbn.psi.edu/pds/resource/sdsstax.html`

Hutson M. L., Lewis J. S., 1991, Enstatite Chondrites And Achondrites As Asteroidal Resources, https://ntrs.nasa.gov/citations/19910016742

Ivezic Z., 2008, SDSS-MOC4, https://faculty.washington.edu/ivezic/sdssmoc/sdssmoc.html#catalogs

Javoy M., et al., 2010, Earth and Planetary Science Letters, 293, 259

Jiang L., 2020, A Visual Explanation of Gradient Descent Methods (Momentum, AdaGrad, RMSProp, Adam), https://tinyurl.com/yc8mny37

Johansen A., Lambrechts M., 2017, Annual Review of Earth and Planetary Sciences, 45, 359

Jones R. H., 2003, in Meyers R. A., ed., , Encyclopedia of Physical Science and Technology, third edition edn, Academic Press, New York, pp 559–574, https://www.sciencedirect.com/science/article/pii/B0122274105004348

Kaasalainen M., Torppa J., 2001, Icarus, 153, 24

Kaasalainen M., Torppa J., Muinonen K., 2001, Icarus, 153, 37

Kapil D., 2019, Stochastic vs. Batch Gradient Descent, https://medium.com/@divakar_239/stochastic-vs-batch-gradient-descent-8820568eada1

Keil K., 2010, Geochemistry, 70, 295

Kingma D. P., Ba J., 2015, CoRR, abs/1412.6980

Kingma D. P., Welling M., 2014, Auto-Encoding Variational Bayes, https://arxiv.org/pdf/1312.6114.pdf

Koo C. L., Liew M., 2013, Biomed Research International, 1

Krot A., Keil K., Scott E., Goodrich C., Weisberg M., 2014, in Holland H. D., Turekian K. K., eds, , Treatise on Geochemistry (Second Edition), second edition edn, Elsevier, Oxford, pp 1–63, https://www.sciencedirect.com/science/article/pii/B9780080959757001029

Lauretta D., et al., 2019, Nature, 568, 55

Lavorini V., 2018, Gaussian Mixture Model clustering: how to select the number of components (clusters), https://tinyurl.com/34wz8pn2

Lazzarin M., Marchi S., Magrin S., Licandro J., 2005, Monthly Notices of the Royal Astronomical Society, 359, 1575

Lazzaro D. A. C. C. J. M.-D. T. D. R., Florczak M., 2006, Small Solar System Objects Spectroscopic Survey, NASA Planetary Data System, `https://sbn.psi.edu/pds/resource/s3os2.html`

Lazzaro D., Angeli C., Carvano J., MothÃ©-Diniz T., Duffard R., Florczak M., 2004, Icarus, 172, 179

Lemaître G., Nogueira F., Aridas C. K., 2017, Journal of Machine Learning Research, 18, 1

Lin Y., 2022, Prog Earth Planet Sci, 9, 483

Lisboa P. J., Etchells T. A., Jarman I. H., Chambers S. J., 2013, BMC Bioinformatics, 14

Liu B., Ormel C., Johansen A., 2019, Astronomy and Astrophysics, 624

Mahlke, M. Carry, B. Mattei, P.-A. 2022, A&A, 665, A26

Mainzer A., Usui F., Trilling D., 2015, Space-based thermal infrared studies of asteroids. University of Arizona Press, pp 89–106

Mamoru D. e. a., 2010, Astrophysics Journal, 139, 1628

Mandell A. M., 2014, Oligarchic Growth. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 1–1, doi:10.1007/978-3-642-27833-4˙1102-4, `https://doi.org/10.1007/978-3-642-27833-4_1102-4`

McInnes L., Healy J., Astels S., 2016, HDBSCAN: How it works, `https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html`

McInnes L., Healy J., Astels S., 2017, The Journal of Open Source Software, 2, 205

Mikulski D., 2006, PhD thesis, U.S. Army Tank Automotive Research, Development and Engineering Center, `https://www.researchgate.net/publication/236671728_Rough_Set_Based_Splitting_Criterion_for_Binary_Decision_Tree_Classifiers`

Minsky M., Papert S., 1969, Perceptrons. MIT Press

Morbidelli A., 2015, The Dynamical Evolution of the Asteroid Belt. The University of Arizona Press, pp 493–507

Morbidelli A., Levison H., Tsiganis K. e. a., 2005, Nature, 435, 462–465

Murphy K. P., 2022, Probabilistic Machine Learning: An introduction. MIT Press, `https://probml.github.io/pml-book/book1.html`

Nakamura T., et al., 2011, Science, 333, 1113

Nesvorný D., 2018, Annual Review of Astronomy and Astrophysics

Newman T., Han S., 2017, All you need to know about neurons, `https://www.medicalnewstoday.com/articles/320289`

Ng A., Jordan M., Weiss Y., 2001, in Dietterich T., Becker S., Ghahramani Z., eds, Vol. 14, Advances in Neural Information Processing Systems. MIT Press, `https://proceedings.neurips.cc/paper/2001/file/801272ee79cfde7fa5960571fee36b9b-Paper.pdf`

O'Brien D. P., Morbidelli A., Bottke W. F., 2006, Icarus, 191, 434

Open University 2008, Mathematical Methods and Models: Block 3. The Open University, Milton Keynes

Pedregosa F., et al., 2007, sklearn.neural$_{n}etwork.MLPClassifier$,

Pedregosa F., et al., 2011a, 1.4.1.3 Unbalanced Problems, `https://scikit-learn.org/stable/modules/svm.html`

Pedregosa F., et al., 2011b, Journal of Machine Learning Research, 12, 2825

Piani L., Marrocchi Y., Rigaudier T., Vacher L. G., Thomassin D., Marty B., 2020, Science, 369, 1110

Pieters C., Hiroi T., 2004, 35

Popescu M. Vaduvescu O. d. L. J.-G. R. L. J. B. I. S. A. A. R. M. T. M. D. P. M. D. P. M. F. C. S. H. D. A. M. O.-E. I. L.-M. F. E. R., 2019, @doi [Astronomy & Astrophysics] 10.26093/CDS, 627

Popescu, M. et al., 2019, @doi [A&A] 10.1051/0004-6361/201935006, 627, A124

Popescu M., et al., 2019, @doi [Astronomy & Astrophysics] 10.1051/0004-6361/201935006, 627, A124

Prim R., 1957,] 10.1002/j.1538-7305.1957.tb01515.x., pp 1389–1401

Pyshark 2022, Calinski-Harabsz Index for K-Means Clustering Evaluation Using Python, `https://pyshark.com/calinski-harabasz-index-for-k-means-clustering-evaluation-using-python/`

Qingkai Kong Timmy Siauw A. B., 2020, Cubic Spline. Academic Press, https://pythonnumericalmethods.berkeley.edu/notebooks/chapter17.03-Cubic-Spline-Interpolation.html

Raymond S., Izidoro A., 2017, Icarus, 297, 134

Raymond S. N., Nesvorný D., 2022, Origin and Dynamical Evolution of the Asteroid Belt. Cambridge University Press, p. 227–249, @doi10.1017/9781108856324.019

Reddy V., Hardersen P., Gaffey M., Abell P., 2005

Ridpath I., 2016, A Dictionary of Astronomy (2 ed.), https://www.oxfordreference.com/view/10.1093/acref/9780199609055.001.0001/acref-9780199609055-e-1156

Rivkin A. S., Howell E. S., Lebofsky L. A., Clark B. E., Britt D. T., 2000, Icarus, 145, 351

Roberts E., 2022, Neural networks: perceptrons, https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Neuron/index.html

Rocca J.; Rocca B., 2019, Building Autoencoders in Keras, https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

Roh D.-G., Moon H.-K., Shin M.-S., DeMeo F. E., 2022, @doi [Astronomy & Astrophysics] 10.1051/0004-6361/202039551, 664, A51

Rosenblatt F., 1958, Psychol Rev., 65, 386

Saikia S., 2019, Building a Convolutional Autoencoder with Keras using Conv2DTranspose, https://tinyurl.com/4sxp3rk3

Satopaa V., Albrecht J., Irwin D., Raghavan B., 2011, in 2011 31st International Conference on Distributed Computing Systems Workshops. pp 166–171, @doi10.1109/ICDCSW.2011.20

Schwarz G., 1978, The Annals of Statistics, 6, 461

Sergeyev, Alexey V. Carry, Benoit 2021, @doi [A&A] 10.1051/0004-6361/202140430, 652, A59

Shafkat I., 2018, Intuitively understanding variational autoencoders, https://tinyurl.com/5dt24mc8

Shannon C. E., 1948, @doi [The Bell System Technical Journal] 10.1002/j.1538-7305.1948.tb01338.x, 27, 379

Sharma A., 2022, Variational Autoencoder in Tensorflow, `https://learnopencv.com/variational-autoencoder-in-tensorflow/`

Siekmann I., 2021, Lecture notes: 7022DATASCI Efficient algorithms for complex data sets

Smola, Alex J. Schölkopf, Bernhard 2004, @doi [Statistics and Computing] 10.1023/B:STCO.0000035301.49549.88, 14, 199

Subbotin Y., 2022, Encyclopedia of Mathematics, `http://encyclopediaofmath.org/index.php?title=Spline_interpolation&oldid=48784`

Tholen D. J., 1984, Asteroid Taxonomy from Cluster Analysis of Photometry., `http://hdl.handle.net/10150/187738`

Trigo-Rodriguez J., Rimola A., Tanbakouei S., Cabedo V., Lee M., 2019, @doi [Space Science Reviews] 10.1007/s11214-019-0583-0, 215

Tsiganis K., Gomes R., Morbidelli A. e. a., 2005, Nature, 435, 459–461

Varoquaux G., 2007—2022, PCA Example with Iris Data-set, `https://scikit-learn.org/stable/auto_examples/decomposition/plot_pca_iris.html#sphx-glr-auto-examples-decomposition-plot-pca-iris-py`

Veeder G., Matson D., Tedesco E., 1983, Icarus, 55, 177

Vilas F., 2008, @doi [The Astronomical Journal] 10.1088/0004-6256/135/4/1101, 135, 1101

Villeneuve J., Chaussidon M., Libourel G., 2009, Science, 325, 985

Virtanen P., et al., 2020, @doi [Nature Methods] 10.1038/s41592-019-0686-2, 17, 261

Walsh K., Morbidelli A., Raymond S. e. a., 2011, Nature, 475, 206

Warren P. H., 2011, @doi [Earth and Planetary Science Letters] https://doi.org/10.1016/j.epsl.2011.08.047, 311, 93

Weidenschilling S., 1977, Astrophys Space Sci, 51, 153–158

Wetherill G. W., 1992, Icarus, 100, 307

Wilimitis D., 2018, The Kernel Trick in Support Vector Classification, `https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f`

Wilson D. L., 1972, @doi [IEEE Transactions on Systems, Man, and Cybernetics] 10.1109/TSMC.1972.4309137, SMC-2, 408

Witze A., 2021,] https://doi.org/10.1038/d41586-021-00641-8, 591

XGBoostDevelopers 2022, Introduction to Boosted Trees, `https://xgboost.readthedocs.io/en/stable/tutorials/model.html`

Youdin A. N., Goodman J., 2005, @doi [The Astrophysical Journal] 10.1086/426895, 620, 459

Yuan C.and Yang H., 2019, @doi [J] 10.3390/j2020016, 2, 226

Zellner B., Gehrels T., Gradie J. C., 1974, The Astronomical Journal, 79, 1100

Zellner B., Tholen D., Tedesco E., 1985, Icarus, 61, 355

Zolensky M. E., Abreu N. M., Velbel M. A., Rubin A., Chaumard N., Noguchi T., Michikami T., 2018, in Abreu N., ed., , Primitive Meteorites and Asteroids. Elsevier, pp 59–204, @doihttps://doi.org/10.1016/B978-0-12-813325-5.00002-1, `https://www.sciencedirect.com/science/article/pii/B9780128133255000021`

sci-kit learn developers 2022, scikitlearn documentation 2.3.2 K-means, `https://scikit-learn.org/stable/modules/clustering.html#k-means`

scikit-learn developers 2007-2022, Guassian mixture models, `https://scikit-learn.org/stable/modules/mixture.html#gmm`

scikit-yb developers 2016-2019, Yellowbrick, `https://www.scikit-yb.org/en/latest/api/cluster/elbow.html`