SELF-ORGANIZING MAPS and GALAXY EVOLUTION

A Thesis Submitted to the Committee on Graduate Studies in Partial Fulfillment of the Requirements for the Degree of Master of Science in the

Faculty of Arts and Science

TRENT UNIVERSITY

Peterborough, Ontario, Canada

Applied Modelling and Quantitative Methods M.Sc. Graduate Program

January 2015

ABSTRACT

SELF-ORGANIZING MAPS and GALAXY EVOLUTION

Jacques Béland

Artificial Neural Networks (ANN) have been applied to many areas of research. These techniques use a series of object attributes and can be trained to recognize different classes of objects. The Self-Organizing Map (SOM) is an unsupervised machine learning technique which has been shown to be successful in the mapping of high-dimensional data into a 2D representation referred to as a map. These maps are easier to interpret and aid in the classification of data. In this work, the existing algorithms for the SOM have been extended to generate 3D maps. The higher dimensionality of the map provides for more information to be made available to the interpretation of classifications. The effectiveness of the implementation was verified using three separate standard datasets. Results from these investigations supported the expectation that a 3D SOM would result in a more effective classifier.

The 3D SOM algorithm was then applied to an analysis of galaxy morphology classifications. It is postulated that the morphology of a galaxy relates directly to how it will evolve over time. In this work, the Spectral Energy Distribution (SED) will be used as a source for galaxy attributes. The SED data was extracted from the NASA Extragalactic Database (NED). The data was grouped into sample sets of matching frequencies and the 3D SOM application was applied as a morphological classifier. It was shown that the SOMs created were effective as an unsupervised machine learning technique to classify galaxies based solely on their SED. Morphological predictions for a number of galaxies were shown to be in agreement with classifications obtained from new observations in NED.

Keywords: Self-Organizing Maps, Kohonen, SHARCNET, Parallel, Galaxy, Evolution, Multi-wavelength, Classification, Morphology

Acknowledgements

I would like to thank my thesis supervisor Dr. Sabine McConnell for her guidance and patience through the multiple versions of this research. Her suggestions on different approaches to certain problems and encouragement over obstacles was instrumental in the completion of this work.

I would also like to thank Dr. Judith Irwin for explaining and clarifying numerous astrophysical processes which affect the data I collected. Without this mentoring on the physics involved, the data being processed would have led to erroneous results. I am most appreciative of her sharing both her time and knowledge throughout the different stages of this research.

The work performed here would not have been possible without the additional support of numerous individuals and organizations. I would like to thank both John Lorenc and David Cullen for their support and understanding when academic needs required juggling of work timetables.

This research has made use of the NASA/IPAC Extragalactic Database (NED) which is operated by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. All of the data processed in this thesis were extracted from the NED database. I would like to thank Marion Schmitz from the NED team for assistance with my data extracts.

This work was made possible by the facilities of the Shared Hierarchical Academic Research Computing Network (SHARCNET:www.sharcnet.ca) and Compute/Calcul Canada. All of the results presented in this thesis were processed using resources on SHARCNET. From the SHARCNET team I would like to thank Dr. Alex Razoumov, Mark Hahn and Kaizaad Bilimorya. They were instrumental in getting my code to work on the SAW cluster as well as providing technical information on the computing environment. In all, just over 4.9 years of processing time

was used on the SAW cluster during the processing of the various datasets. This represents computing resources which would have made this project impossible using commodity based computers.

My final thanks go to my family. Carol and Eric have been both supportive and understanding of this seemingly never ending process.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The advent of relatively inexpensive computing technologies has brought about a vast increase in the amount of information created and shared amongst various parties. The Internet and new technologies such as camera phones and social media sites such as Facebook generate an enormous amount of information daily. Similarly, the medical, public safety, insurance and travel industries create their own volumes of data. Leveraging these new technologies has brought about the requirement for flexibility and quick access to individual data items.

The scientific community has also benefitted from advances in technology. The pace with which scientific data are now collected is impressive. Projects such as the Large Hadron Collider, the Hubble telescope and multiple automated sky surveys produce Terabytes of data daily when they are in operation.

In many cases, individuals are only interested in retrieving information in its original form. On Facebook, it is sufficient to have family and friends read a post or view a picture. There is, however, great potential in the ensemble of all of the data collected in any one specific field. This potential comes from what can be gleamed from the data itself, not just information about averages and other statistics but also patterns and inter-relationships within the data.

The volume of data present in any one of these databases makes the extraction of useful information impractical without automated approaches. *Data Mining* and *Knowledge Discovery* are names given to various techniques used to generate insights into the data. *Machine Learning* techniques automate the discovery process. One avenue of machine learning is simply the extraction of statistical information which can then lead to policy decisions such as insurance rates. The approach of interest, in this thesis, is the processing of data in the attempt to discover like data elements which are distinct enough to group into classes.

The processing of data to establish an automated technique of classifying objects is performed either through a supervised or unsupervised method. In the supervised case, a subject matter expert uses already established knowledge of object classes to guide the classifier into reproducing expected results. In the unsupervised case, the technique alone is responsible for finding commonalities within the data and from those, identifying objects which belong to the same class.

The research in this thesis will only focus on one specific approach to machine learning. The technique is called a Self-Organizing map. This is an unsupervised technique and will rely only on the attributes in the data to extract class information. There are numerous avenues for the application of machine learning algorithms. In this project, we will investigate the application of Self-Organizing Maps to the problem of galaxy classification. The objective is to create an automated systematic classifier which is capable of differentiating between different galaxy morphologies.

Historically, the classification of galaxies has been performed manually. Photographs in the visible range of the spectrum were used to group galaxies by properties [30, 68]. The existing classification schemes are therefore biased towards the visible part of the spectrum and subject to human interpretation and intuition. This has led to numerous studies which have shown that rarely do human experts agree on individual classifications [41].

Hubble had established his classification scheme before the advent of sensors capable of measuring signals in regions of the electromagnetic spectrum unavailable to the human eye. Sensor improvements and automated surveys have and will continue to provide a volume of information far exceeding classification capacities of human experts. The NASA-Extragalactic Database (NED) [47] is a data repository which incorporates the data from many separate studies into one single standard source of galaxy attributes. The galaxy properties used in this modelling effort will be restricted to the amount of energy the galaxies are emitting in various frequencies across the electromagnetic spectrum. Collectively, these measurements are known as the Spectral Energy Distribution (SED).

The research presented here had two aims. The first focus will be on the creation of an effective 3D version of a Self-Organizing Map classifier. This new implementation will be evaluated against known datasets to confirm its effectiveness as an unsupervised approach. It will be shown that the implementation of a 3D SOM was successful. The implementation was also used to investigate various approaches to improving both performance and the quality of the results obtained.

The second objective is to determine if, within the galaxy data collected, there exists combinations of frequencies which could lead to a valid classifier. There is no a priori knowledge of which frequencies within a SED are telltale signs of galaxy morphology. An investigation of the combinations of available frequencies will therefore be initiated. Though a significant amount of data was collected for analysis, the sparsity of the data had a direct impact on the results obtained. It will be shown that the 3D SOM was effective in resolving galaxies into major morphological classifications. The approach was also successful in predicting the class of a number of galaxies whose morphology was unknown at the onset of this research.

Chapter 2

Background

2.1 Data Mining

Data mining [62] is the process of extracting useful, sometimes non-intuitive information from a number of observed objects or data events. Each of these can be characterised by a list of properties which we will call *attributes*. These attributes describe all of the known facets of these objects that are deemed important in modelling their underlying rules and relationships. By leveraging various algorithms to compare the attributes of these objects or events, it is possible to group like items together.

The outcome of the modelling process can take the form of a set of rules describing the relationships between attributes in a dataset. This can be used, for example, to evaluate the probability of a specific outcome. In some applications, it is sufficient to have the data-mining process produce a set of results which can then be interpreted by subject matter experts. In others, what is desired, is to have the process recognize specific types of objects and associate them with unique predetermined classes. The associations are typically vetted by a human expert who is developing the process. The implementation of the algorithms in software which permits a computer to process process data and learn to classify events is called *machine learning*.

2.2 Machine Learning

The goal of teaching a machine to classify data typically derives from the fact that subject matter experts are limited in their ability to visualize and interpret datasets which possess a large number of dimensional attributes. Interpreting relationships with data possessing more than three dimensions can be difficult. Furthermore, the large number of events collected for analysis most often make the task of manual analysis impossible. Often there are relationships within the data that are unexpected and exist beyond the domain knowledge of such expert.

The construction of algorithms that analyse events in a systematic approach allows for the detection of these unexpected relationships.

The benefit of creating a process by which objects can be classified is not limited to the immediate dataset. The process can be used as a guide or model which will allow the classification of new events. This provides the capability of leveraging the model to detect new and unexpected events or just focus on specific known occurrences.

There are numerous different approaches to developing processes capable of classifying events. Some must be guided by human experts, while others only require post-processing interpretation. In this work, we shall focus only on a variation of *Artificial Neural Networks*, a technique called *Self-Organizing Maps*.

2.2.1 Artificial Neural Networks

Artificial Neural Networks, henceforth called ANN, are a machine learning technique which are used in an attempt to mimic the signaling processes involved in the human brain [39]. Though we will not examine all of the nuances of ANNs here, many of the concepts are important in regards to their similarities to the technique used in the data analysis portion of this thesis: the Self-Organizing Map.

In a model of the brain, a *neuron* can be treated as a simple processing unit. It receives a number of signals from various sources and processes them based on a learned response. This learned response evolves through repeated exposure to similar inputs and feedback from the resultant output. Though the chemical processes involved in the generation of the response are not known completely [39, 54], they can be mimicked through various means known as *activation functions*. In the software equivalent of neurons, the signal strength coming from the various input pathways are combined, through individual weighting, to produce an integrated response signal of the artificial neuron. Based on the strength of the normalized inputs, the neuron decides if it should fire an output signal or not. This output signal can be directed to follow-up stages consisting of a single neuron or to a set of multiple targets. A simple example of an artificial neuron configuration can be seen in Figure 2.1

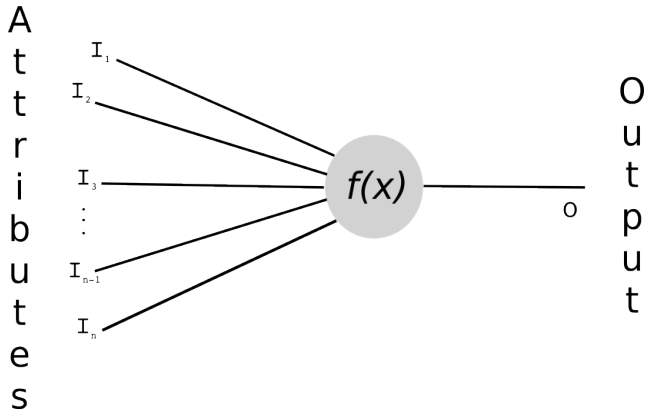

Figure 2.1: A simple model of an artificial neuron showing n inputs and a single output.

In the software implementation of a neural network the number of neurons is typically limited. The network should have a sufficiently large enough neuron population to accept all of the individual inputs and to represent all of the properties present in the input space. The neural network's output must also be configured to present information which can be interpreted to produce distinct classifications of the input.

The human body contains about a trillion 10^{12} of neurons [31, 54]. In a typical ANNs, however [31], 10 to 10,000 neurons are typically used. These are grouped into three main layers: *Input*, *Hidden* and *Output*.

The input layer: The input signals for this layer are derived directly from the attributes which describe the events we want to examine. If the objects or events we are classifying are characterised by n properties, the design of the neural network will typically have n neurons in the input layer.

The hidden layers: These layers receive signals from the input layer and combine them based on a set of input weights. If the signals satisfy the requirements of the activation function, a signal is sent to the next layer in the ANN. The next layer can either be another hidden layer or the output layer. Jadid [34] suggests that an upper bound to the number of neurons in the hidden layer is 10 or 20 percent of the ratio of the training set size to the total number of neurons present in both the input and output layers.

The output layer: This layer presents one or multiple output nodes. The signal provided by the output layer is used to interpret the classification of the event. This is either presented as a range of output values or by signals from specific output neurons.

An example of a simple three-layer neural network is shown in Figure 2.2

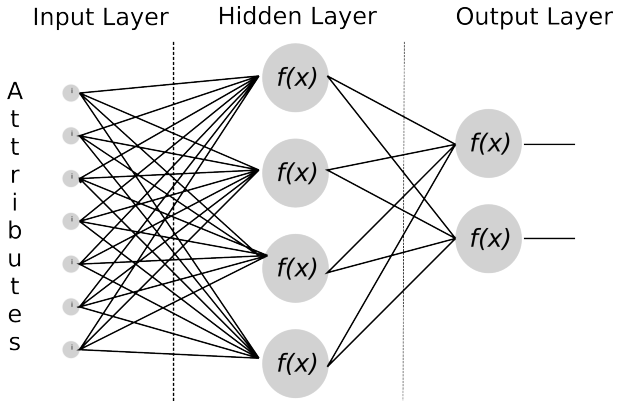

Figure 2.2: An example of a 7:4:2 ANN with seven inputs, 4 neurons in the hidden layer and two outputs.

Artificial Neural Networks can learn using a variety of supervised or unsupervised approaches. For a supervised ANN, the network must be trained using previously classified information. Example events of known classifications are presented to the network. Through various algorithms, the weights which affect the signal combinations are tuned to produce the desired output classifications. The tuning process is performed over numerous iterations through the training data. Each pass adjusts the weights between the various neurons in an attempt to provide the best classifications possible for the ensemble of the data. Attention must be taken to ensure that possible idiosyncracies in the data do not lead to over-tuning of the weights.

Unsupervised approaches do not rely on any a priori knowledge of the classes present in the input data. The data are presented to the network in multiple iterations. With each iteration, the network adjusts its responses based solely on the attributes of the input data. After several iterations, the weights associated with each neuron will stabilize and their variations will reach an agreed-upon stopping criteria.

The quality of the network can be evaluated in a number of ways. If a subject matter expert expects a specific distribution amongst the classes, an analysis of the output stages can reveal a qualitative measure of the goodness of the network. A more common approach is to use data of known classes, either from a subset of the data or a separate test dataset, and measure the effectiveness of the ANN in reproducing the results.

The machine-learning aspect of this approach is dependent on existing interpretations of the data and the training process is therefore subject to any bias in the training dataset. Once trained, an ANN can be used to quickly classify new data. For the purpose of this thesis, we will investigate the application of an unsupervised technique, the Self-Organizing Map.

2.2.2 Self-Organizing Maps

The Self-Organizing Map (SOM), is a unique category of neural networks. Their current form was introduced by Teuvo Kohonen [39] in 1990. Unlike previous neural networks which had neurons operating in sequenced layers, the neurons in Kohonen's maps compete as a whole for the input data. As such, Kohonen labelled his neural network a *competitive* learner [4, 39, 71].

The objective of the mapping process is to define a set of prototype vectors, represented by the neurons, to accurately represent the input data. The arrangement of these prototypes into a predetermined grid allows for the projection of a high-dimensional space represented by the number of input attributes, into an easier to interpret low-dimensional grid space. Kohonen's approach provides for a mechanism which can preserve the neighbourhood relationships from the input space to the SOM's geometry. The objective of the mapping process is that objects

that are similar in the input data will resolve to neighbouring positions within the final map. These groups of like neurons are called *clusters* and are the mechanism by which we can identify the classifications of the objects being mapped.

A real-world example of implementing the Kohonen algorithm could be a library classification system for books. As objects, books have a number of properties. A short list could include: subject matter, author, publisher, size, colour, keywords and binding method. If we consider all of these properties as independent from each other, they can define an attribute space for all books. The mapping process from the high-dimensional attribute space to the library shelves is what the SOM will attempt to perform.

Similar books would wind up with similar vectors. The measure of similarity, as we will discuss later in Section 2.3.3, will determine the proximity of the books within the library stacks. For this example, it will suffice to imagine that the similarity measure between "Mark Twain" and "H.G. Wells" is comparable to that of "Poetry" and "Chemical Engineering" as well as "Penguin Press" and "Random House". Other attributes which are numerical in nature, such as physical dimensions, weight and number of pages, have similarity measures which can be expressed more mathematically if desired.

We consider the stacks in a library as a collection of prototype vectors, each individual shelf having a potentially unique collection of properties from the input space. The SOM algorithm would then classify like books together, in doing so, creating sections in the library for Literature, History, Mathematics, Business and Sciences. Within this mapping, however, Biochemistry books would find themselves mapped in boundary regions between the two main classes: Biology and Chemistry. An example of a possible 3D SOM map of a library is shown in Figure 2.3.

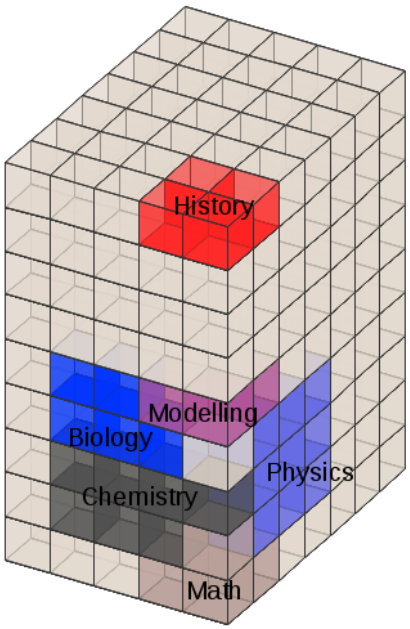

Figure 2.3: A simulated SOM of a science library classification system.

This example demonstrates the potential for SOMs. They can take a high-dimensional space and map it to a two-dimensional bookcase and shelf arrangement. Careful attention must be taken when choosing which attributes to include in the mapping process. In the above example attribute space, including the binding method may produce results wherein identical works (e.g. same author, same title), may not end up being co-located in the map if one were a paperback and the other hardbound. If the selection and weighting of the attributes is not initially carefully considered, the *binding* dimension could carry as much weight as author or language.

Once established, the SOM model can be used to quickly classify new arriving titles into the collection. It can also be used to identify books for which no existing mappings exist. Books that match existing prototypes will be easily classified. New arrivals with unobserved attribute values will be identified as outliers and could facilitate an expansion or a re-training of the existing map.

It is also important to note that adding dimension to the library, such as adding floor or building information, allows for greater flexibility in how books are attributed with locations within the library. Similarly, extending the standard 2D SOM to 3D allows for the possibility of higher granularity in the positioning of objects within a map. In this thesis, we will investigate the application of a 3D version of the Kohonen SOM to various datasets.

2.2.3 Summary

In this section we have introduced machine learning. The application of machine learning algorithms can be leveraged to facilitate the extraction of knowledge from a collection of data. The rules and patterns discovered in the data can then be used to determine the classification of new data events. We will now examine the algorithm used here in more detail.

2.3 Overview of the Kohonen Algorithm

The Kohonen algorithm provides a framework which can be implemented in a multitude of ways. Self-Organizing Maps based on Kohonen's approach are often called *Kohonen Maps*. The interpretation of each element of the algorithm can lead to the construction of vastly different maps. There are, however, a number of design guidelines which can be leveraged to produce more effective maps. The most important of these are: *Neuron Population Size*, *Distance and Similarity*, *Initialization of the map*, *Normalization of the Data*, *Missing Data*, *Adjustment of the prototype weights*, *Map Geometry*, *Quality Assurance Measures* and *Termination criteria*.

In the following sections, we will discuss these features and guidelines for their implementation will be suggested. Before doing so, a brief overview of the algorithm is necessary to provide context for these parameters.

2.3.1 Introduction to SOMs

Kohonen Maps provide a mechanism by which a set of data with a high-dimensional attribute space can be mapped to a lower-dimensional representation. The process is expected to enhance the appearance of interrelationships within the data, leading to a robust classification system.

The process begins with the selection of a target map geometry. The size and shape of the map must be chosen to facilitate the interpretation of the results while providing sufficient volume for the mapping process to work efficiently. The internal prototype vectors must be initialized to values which are representative of the input space. Techniques for initializing the map will be discussed in section 2.3.6.

The algorithm then involves an iterative process of choosing random data from the input dataset and finding where in the map that particular item should be placed. Data elements are presented to the map individually. The map is scanned for the prototype vector which most closely resembles the attributes of the candidate element. Once the prototype which is most similar to the datum is found, its attribute weights are adjusted to make it more representative of this particular data element. The process then extends the adjustment of the prototype attributes to neighbouring cells. This "tunes" this specific region of the map to attract data elements from the input dataset which are most similar to the current candidate. These prototype weight adjustments alter the map and can affect subsequent prototype-datum pairings.

For each iteration, all of the input data are presented to the map in a different random order. The mapping process continues until a predetermined condition is met. The termination criteria can be based on a map quality measure, when input data elements become bound to unique prototypes or simply a fixed number of iterations.

```
While (the termination criteria is not met)
{
 adjust the neighborhood radius
 adjust the learning rate
 randomize the presentation order of the data
 foreach data item
 {
  Find the closest matching prototype
  for all cells within the neighbourhood radius
  {
    adjust prototype weights of each cell
        based on distance and learning rate.
  }
 }
evaluate the termination criteria
}
```

Figure 2.4: A pseudocode version of the Kohonen Algorithm.

The completed Kohonen Map will, based on a sufficient quality measure, group like prototypes into areas which can identify clusters within the input data. The effectiveness of the algorithm and the quality of the results obtained are highly dependent on the characteristics of the target map. The following sections will address some of the more important facets involved in building an effective Kohonen Map.

2.3.2 Neuron Population Size

The choice of the number of neurons to use in a Self-Organizing Map is subjective and a balance must be struck between the number of neurons present and the computational cost they each represent. There should be a sufficient population in the output space to effectively represent the expected number of classes present in the input data. There should also be sufficient additional neurons in the map to help highlight demarcation boundaries between the various clusters of points.

Kohonen's paper [39] presents a taxonomy example map where an arrangement of 70 cells are configured in a 7x10 grid for this map. He uses this structure to map 32 data samples possessing five attributes each. This represents a greater than one-to-one relationship between the number of neurons and samples. For large datasets with multiple attributes, this would suggest very large maps are required. For reasons we shall explore shortly, this ratio is much too computationally expensive for sample sizes ranging into the tens of thousands.

For a sample size of N, an appropriate number of prototypes is on the order of $\sqrt{N}$ [61]. Vesanto [69] and Wendel [73] have adopted $5\sqrt{N}$ as the default number of

cells for the SOM for the SOMToolbox[1]. Wendel & Buttenfield [73] have extended on this by including the number of attributes into the calculation. They suggest that the SOM cell count should be more closely defined as in Equation 2.1, where a represents the number of attributes in the dataset.

$$N_{cells} = 5\sqrt{N \times a} \tag{2.1}$$

The term neuron is typically used to specify a unique element of the SOM grid. There is a one-to-one relationship between the neurons, their prototype vectors and the individual cells within the grid. We will therefore use the terms neuron, cell and prototype interchangeably.

2.3.3 Distance and Similarity

There are two very important concepts used in Kohonen maps. These will be called similarity and distance for the remainder of this thesis. Though synonyms in common usage, in SOMs these terms are used to describe two very distinct geometries present in the map. As such, we will adhere to the following definitions:

Distance: Distance will be understood to be a measure of proximity between two elements of the SOM grid structure. Typically it represents the Euclidean proximity of the two points and can be expressed in terms of grid units. For irregular geometries of the SOM, such as the surface of a toroid or a sphere, the evaluation of distance may not be as straight forward as calculating the Euclidean distance.

Similarity: Similarity will be used to measure proximity within the data's attribute space. It will be a measure that will evaluate how closely two entities are to being identical at the attribute level. There are numerous ways in which we can compute similarity. It is important to note, however, that many of these techniques are dependent on compatible ranges of values between the various attributes. If we are to calculate a Euclidean measure of similarity and one attribute is several orders of magnitude larger than the others, it will bias the similarity measure towards that single attribute. If a simple Euclidean similarity is used, the original data must be normalized to prevent this bias. Other techniques such as the *Pearson* similarity measure offers a uniform balance between the different dimensions in the dataset[7]. Another approach for evaluating similarity, the *Mahalanobis* measure, can account for correlations between the input dimensions[7].

[1]The SOM toolbox is an GNU GPL application made available by the Laboratory of Computer and Information Science. The authors are: Esa Alhoniemi, Johan Himberg, Juha Parhankangas and Juha Vesanto. See: http://www.cis.hut.fi/projects/somtoolbox/

In the context of this thesis, we will only deal with numerical data. The concept of similarity measures with categorical data, such as "Math", "Science" and "History" in the above library example, will not be addressed. Further reading on this topic can be found, for example, in Boriah's paper [6].

It is imperative to keep the concepts of similarity and distance distinct. Even though for our purposes we will use the Euclidean measure for both, one represents proximity in attribute space and the other by grid coordinates.

2.3.4 Normalization of the Data

Ensuring that the similarity measure provides unbiased values is paramount for the proper application of the SOM algorithm. It does, however, introduce the problem of how to place all of the data attributes on a common scale.

One technique that is often used [13, 31, 40, 69] is to preprocess the input data and, for each attribute, calculate a mean and a standard deviation. Attributes are then normalized by subtracting the mean from each value and by scaling the data's distribution variance to a range of plus or minus one. Note that since we are using Euclidean similarity values which required the sum of the squares of the differences, negative attribute values do not adversely affect the calculation.

A different technique involves finding the minimum and maximum value for each attribute and scaling all data to lie between zero and one. A drawback to this technique is that it does not leave any room for future data processed by the map if newly acquired data falls below the minimum or above the maximum. One could add extra padding to the scale factor but determining the proper amount would have to be estimated by a subject matter expert using knowledge not available in the input data. This could introduce bias into the map if all attributes are not normalized in the same fashion.

An alternate approach would see the global data range for all attributes employed as the scale factor. This would ensure that the scaling is linear across the range of values. If there exists subtle relationships between the attributes, the scaling may impact on our ability to detect it.

The selection of a normalizing methodology must minimize any bias imposed on the data. The application of any algorithm must ensure that the original data are normalized on a common footing. If some attributes are used under a logarithmic scale, such as pH or visible magnitudes, any scaling should be performed on the untransformed data. It must also ensure that the data are prepared in such a fashion to allow a well-balanced evaluation of the similarity measure between the input data and the prototype neurons of the SOM.

2.3.5 Missing Data

There are a number of different proposed techniques for addressing the problem of missing data [14, 37, 62]. The Kohonen algorithm is, by design, sensitive to the input data. Any bias in the input attributes will create a bias in the map. One can simply eliminate data elements with missing values from analysis. If the data requires processing items with missing values, attempts can be made to approximate their value [62] or their contribution can be ignored when evaluating similarity measures [37].

The introduction of a replacement value for a missing data attribute by its mean or interpolated value [62] may or may not be appropriate. In simple cases, interpreted values based on adjacent measurements may be acceptable. In situations where the behaviour of the property is less well known, assigning a replacement value may introduce bias. The choice of technique in itself can introduce a bias into the input data item as well. The choice would depend on the subject matter's expert opinion which in itself, is biased based on their experiences.

For the purpose of this thesis, we will reject any candidate items which do not have a complete set of attributes. Though this will significantly reduce the number of available items for the study, it will minimize the introduction of any additional bias into the map.

2.3.6 Initialization of the Map

In the Kohonen implementation, each neuron carries with it a measure of how strongly it expresses a value for each of the attributes. Each neuron, which exists in a specific grid location, is called a prototype vector. In the Kohonen algorithm the prototypes are tuned repeatedly in an attempt to allow the map to provide the best possible representation of the input space.

The objective of the initialization is to provide a set of candidate prototype neurons which span the expected input attribute space. Recalling the requirements of the normalization of the data, the initial values of the map should correspond to the same scale and range. Here are a few examples of different initialization techniques:

Random: This technique simply assigns a unique random value, between the observed minimum and maximum value of each attribute, to the corresponding attributes of each neuron in the map. Values can also be generated to cover $[-1, 1]$ or $[0, 1]$ for normalized attribute values.

Per-Attribute Random Assign a random value to each attribute of each neuron. The value assigned is based on the mean and standard deviation, or the range or values, for each specific attribute as determined by the normalization technique used.

SOM based Random Assigns random attribute values to each neuron based on the overall range of values observed during the normalization phase. The range is based on the whole SOM attribute space.

Random input data This technique samples a set number of random points from the input space to become a set of seed points. A similarity measure is evaluated between all of the chosen points. The seed data elements are introduced in the SOM and placed at separations from each other commensurate with their similarity in the input space. This placement attempts to preserve the data proximity attributes between both spaces.

Most dissimilar input data This technique parses all of the input data and selects a number of candidate data points which exhibit the most dissimilarity with all of the other elements of the input dataset. These most distant objects should represent the most separated classes of objects. It may also find the worst-case outliers in the population. It should isolate the most extreme prototypes for any of the classes of objects we are trying to find. All other objects in the dataset should reside within these boundaries.

Eigenvalue Using Principal Component Analysis [37, 40, 64] it is possible to extract the most significant eigenvectors and eigenvalues from a dataset. These can then be used in turn to seed the distribution of attribute values linearly across the maps to highlight the pre-existing structure within the data.

Each of these techniques imparts an order onto the original map. Though the map will evolve as data are processed by the algorithm, the initial conditions of the map impart a topological structure within the SOM. Completely random approaches could produce similar prototype neurons in disparate regions of the map. These regions may never coalesce, leaving quite similar prototypes in disjoint portions of the map. This will result in similar data items being mapped into disjointed cluster of nodes within the map. The similar but disjoint regions can have an influence on the convergence rate of the map as data items may be mapped to BMUs between regions at each iteration of the map. The evaluation of the number of actual clusters will be impacted as these separated regions will appear distinct in the grid space of the SOM.

As we shall soon see, the selection of the initialization technique has a direct bearing on the speed with which the Kohonen algorithm will converge on a solution. The first method, random, introduces no knowledge into the original map. Techniques involving seeding the map with some of the original data, however, initially favour data elements most similar to the seeds.

2.3.7 Adjustment of the Prototype Weights

The learning phase of the SOM involves two main steps. Every data item that we are trying to map will undergo these two actions. The first process is to find, within the SOM, which neuron is the most similar to the data elements under consideration. This step is called finding the *Best Matching Unit* or BMU. This involves evaluating the similarity measure between each neuron in the map and the data item under consideration. The node which is found to be the most similar is deemed the BMU.

Once the BMU has been found, the Kohonen algorithm then provides a method by which the similarity between the BMU and data element is reinforced. The attributes, or "weights" of the BMU are modified to more closely match those of the data item being mapped. To enhance the likelihood that neighbouring cells will attract similar data in future iterations, the prototype weights of adjacent nodes are adjusted to more closely resemble the datum being mapped. The amount of the adjustment is dependent on the grid distance between the BMU and the specific neighbour. This distance is called the *nearest neighbour distance*. Figure 2.5 shows the region surrounding a sample BMU located at (3,0,4). Notice that since this BMU is present on the edge of the SOM, it does not have a symmetrical distribution of nearest neighbours as some would reside outside of the map volume.

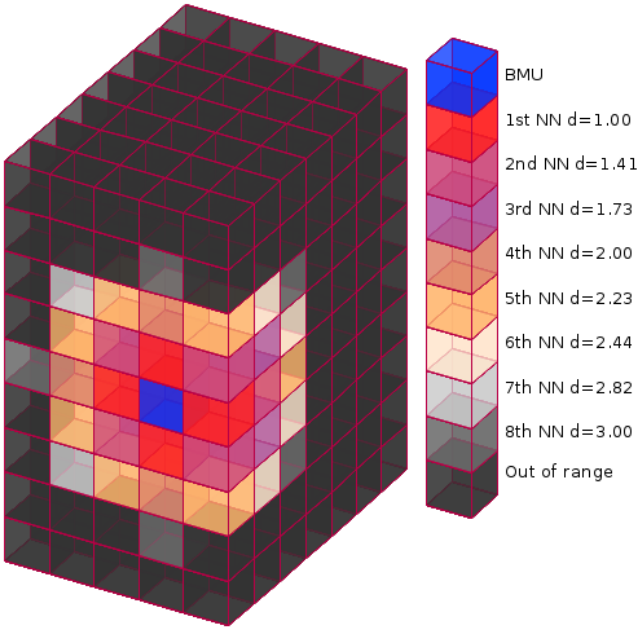

Figure 2.5: A sample BMU at coordinates (3,0,4). The BMU region's radius is set at 3 for this example. Adjacent cells, to a radius if 3, are shown with their Euclidean distances.

The extent to which the weights of individual grid neurons are adjusted will decrease over time. Most initialization schemes set the SOM node weights to values that are independent of the neighbour's values. When training takes place, the algorithm facilitates more and more similarity between adjacent nodes. At first, it is desirable to create large neighbourhoods of similarity to increase the chances of clustering like items from the dataset. As these neighbourhoods are strengthened, it is desirable to allow the algorithm to tighten the radius of similarity. By reducing the number of nearest-neighbour cells affected by the weight adjustments, over time the amount of similarity will become more concentrated around the BMU. This will lead to more compact clusters of nodes which represent the same type of data elements. It also has the benefit of allowing the nodes at the fringes of the clusters to become less tightly bound to the current BMU. This can provide for more clearly defined boundaries between clusters. It will also allow for a zone of nodes which more closely represent data which rightly belong to classifications shared amongst clusters.

2.3.8 Map Geometry

The neurons in a SOM are organized in a grid system. Each neuron resides in an individual grid cell. In some implementations, the grid takes the form of hexagonal tiles, giving each node six equidistant neighbours. In most other implementations, a regular Cartesian grid is chosen for simplicity. The quality of the resulting map is not affected by one's choice of tile shape [66]. The distance between two nodes is important when data are mapped into the SOM. A hexagonal grid system with six neighbours finds all immediate neighbours at a single grid unit distance. Cartesian grids, however, will produce, in two dimensions, four neighbours along the cell's edge at a distance of one unit and four more at the vertices with a distance of $\sqrt{2}$ units. The number of nearest neighbour cells in an SOM will directly affect the number of computations required as the map is built.

A simple Cartesian ordering of the cells of a SOM does bring about one major problem. In the process of building the map, weights of the BMU's neighbouring cells are adjusted. This promotes the creation of a gradient in the attribute weights within cell neighbourhoods. In the Cartesian model, however, a number of cells are found on the periphery of the map. They therefore do not enjoy the same number of adjacent cells as nodes within the bulk of the map. This creates a bias for these nodes as they are less likely to influence, or be influenced by, the same number of adjoining cells as their bulk counterparts. This can result in edge cells attracting a disproportionate number of the input data [29, 52, 74] and causing bias in the map.

A number of alternate solutions have been proposed to alleviate the Cartesian edge effect problem. One solution involves simply wrapping a two dimensional map, allowing cells on one edge to influence cells on the opposing edge. This essentially

turns a two-dimensional map into a surface. A number of different surfaces have been proposed for SOMs; toroidal and spherical are the most common [56, 74]. One issue with using such mappings is the positioning of the appropriate number of neurons across the volume's surface equidistant with each other. Another challenge is keeping track of which nodes neighbour every other node and at what distance. Placing ten, a hundred or even a thousand nodes at equidistant points on the surface of a sphere is not a trivial exercise [19, 74].

In the implementation of the SOM for this work, we have concentrated on the simple Cartesian grid.

2.3.9 Quality Assurance Measures

Self-Organizing Maps are generated in an unsupervised manner. The topology-preservation qualities of the mapping can be affected by local phenomenon within the SOM structure. Initialization techniques can create disparate regions within the map that contain similar prototype neurons. This can result in splitting a collection of the input space into multiple zones within the map when, in fact, the SOM regions represent the same class of data. Similar issues arise at the edges of the map when the attributes of the prototype neurons are not moderated by the same size neighbourhood as cells within the bulk of the map. It is therefore imperative to have a measure of how well the constructed map reflects the input data. Not only is it important to find clusters of nodes in the map, the topology of the input space must be preserved where like items in the input space, map to adjacent areas in the SOM.

The *Quantization Error* (QE) [4, 51, 66] is a measure of how well the cell's prototypes represent the input space. It is evaluated as the average difference between the data vectors and their associated cells prototype vector. The lower the value, the better the collection of data vectors are properly represented by their prototype. One difficulty with this measure is that it is an intra-cell measure of similarity and is therefore unable to gauge the overall effectiveness of the mapping.

The *Topographic Error* (TE) [4, 51, 66] attempts to address the local self-similar issue within the map. It is a measure of how similar adjacent prototype vectors are to the data points they represent. A data element's two BMUs are determined. If the map is well ordered, the two BMUs should be adjacent to each other. If they do not share a common edge or vertex, the algorithm asserts a penalty to the error measure. When all the data are evaluated, a score for the current map is obtained. The higher the score, the more distortion is present in the map. During the training phase of the map, the TE values should decrease in size as similar prototypes within the map migrate towards each other. This is a more sensitive measure to the overall topology preservation of the map as it places no restriction on the location of the data element's two BMUs.

The Topographic Product (TP) [4, 51] examines local distortions in the SOM. For each cell, a fixed number of nearest neighbours are used to generate a local

measure of similarity. This similarity is evaluated for both the SOM cells, and separately for the data points they represent. These two values are combined to give a measure of the effectiveness of the map at preserving the topology of the input space. The total map contribution of these values can be used to determine if more or fewer cells would provide a map with better topological preservation than the current SOM.

The *Trustworthiness Measure* [51, 71] also measures the effectiveness of neighbourhood preservation through the mapping process. It measures the number of dissimilarities between the SOM and input data memberships of a fixed number of nearest neighbours. The average is computed over the SOM, the Trustworthiness Measure is then evaluated as 1 minus the average. The closer the value is to 1, the less distortion is present in the mapping.

The *Neighbourhood Preservation* [51, 71] measure is related to Trustworthiness. The focus of this measure is on the input data and how well similar samples are mapped to the same neighbourhood in the SOM. Again in this case, the closer the measure is to a value of 1, the better the mapping preserves the topology of the input space.

The SOM is typically a 1D, 2D or 3D arrangement of prototype vectors. The vectors have dimensionality equivalent to the number of attributes in the input data. The mapping process projects these high-dimensional vectors into the lower, typically 2D, SOM. Topology preservation is an essential requirement for self-organized maps. The process should ensure that similar objects in the high-dimensional input space are mapped close to each other in the SOM. Likewise, dissimilar objects should find themselves mapped further apart. If the map is not well ordered, then there exist within the map disconnected regions of similarity. These twists or knots in the map indicate that the map was malformed [26, 38, 58]. The map will have to undergo additional iterations, or a different initialization technique should be used and the map recreated. The above measures can be used to suggest a course of action.

2.3.10 Termination Criteria

The Kohonen algorithm is an iterative process. Every data point in the input space is mapped into the SOM. This process molds the SOM to represent the input space. Kohonen has suggested that this training phase of the SOM may take 100,000 iterations [39]. If the input dataset itself contains tens of thousands of data points, this could amount to a significant amount of resources and time. A balance between resources and the quality of the map must then be entertained.

One measure of the map's convergence is to monitor the allocation of the BMU. The selection of a BMU is based on how well it represents an individual datum. If, during the training phase of the map, we find that every data element is being consistently mapped to the same specific BMU, then we know that the SOM has converged on a possible solution. Further processing may not be required.

Every iteration, however, modifies not only the BMU's attributes but also those within its neighbourhood. These changes might lead to an eventual migration of the input space to new BMUs. As the algorithm progresses, the neighbourhood region affected by BMU selection becomes smaller and smaller. At the stage where the neighbourhood function only targets the BMU itself, additional iterations are of less benefit to the assignment of BMUs. Additional processing will tune the prototype vectors to more closely resemble their final set of data elements from the input space.

Each iteration trains the BMU and its neighbourhood to best represent its associated data from the input space. Since the attributes of the BMU are moderated by the attributes of the input data, the process may reach a state where the BMU is hovering around an average of the data and no longer converging. A simple example would be for a prototype that represents two one-dimensional data elements with values of 2 and 6. If the map converges and these two data elements are always presented to the same BMU, then every time the data element 2 is presented to the map, the prototype will become more similar to 2. When 6 is presented to the BMU, the prototype will be adjusted to be more similar to 6. Both data elements belong to the same BMU, however, since they do not have an identical value, the prototype will oscillate between two values ad-infinitum. Extending this to higher dimensions compounds the problem. Unless all of the data points associated with the same BMU have identical attributes, the prototype attributes will not converge to fixed values. The implementation of the SOM must then specify at what point further iterations of the algorithm are beneficial. If none of the data from the input space change their BMU for a fixed number of iterations, the map may be considered as complete. It may be sufficient to stop at this point.

Any algorithm used to terminate the mapping process must take into account the Quality Assurance techniques described in the previous section. The Topographic Error and the Topographic product give a measure of confidence in the map's ability to preserve the topology of the input space. This will allow for a simpler interpretation of the map and the clusters it represents.

2.3.11 Summary

The major concepts which affect the Kohonen Map have been reviewed in this section. Individually, any of these parameters can have a profound impact on both the performance and quality of the final SOM. In turn, this affects the quality of the resultant classifier.

The implementation decisions for any of these map parameters will vary from one subject matter to another. Determining the size of the map will depend on the number of attributes. Normalization will depend on the types of attributes. The similarity measure may need to include some special functions to accommodate categorical attributes such as author or keywords. Additionally, a number of these parameters may have to be determined through an iterative process of trial and error until a satisfactory mapping is achieved.

The configuration of the properties of the SOM will influence the the quality of the final map. Testing several different configurations will help produce a more robust and useful classifier.

2.4 Mathematical Description of the Kohonen Algorithm

The previous section examined the major concepts involved with the Kohonen Self-Organizing Map. The most important of these concepts are the measure of similarity, finding the BMU within the SOM and adjusting the weights once the BMU is found. This section will leverage these concepts and present in mathematical terms the approach used in this work to implement the algorithm.

2.4.1 Adjusting Attribute Weights

Kohonen suggests [39] that in a competitive learning environment, the neuron which most closely matches the input data should be rewarded by tuning its attribute values to more closely match those of the input. In this process, the input characteristics are being imprinted on the map. The extent of the tuning is proportional to the actual similarity between the input data D and the BMU prototype P. We can write the per-attribute weight W adjustment for attribute a as:

$$W_{new} = W_{old} + (\Gamma \times (D_a - P_a)) \tag{2.2}$$

In this equation, Γ is a weighting function which influences the intensity of the training adjustment. We will see in the next few sections that it is moderated by the rate at which the SOM learns as well as a cell's distance to the BMU.

2.4.2 The BMU and the Measure of Similarity

Each iteration of the algorithm requires that for each input data element, the best matching unit be determined. It is found by calculating a measure of similarity between the attributes of the data item and each and every prototype within the SOM. The prototype which expresses the smallest difference is elected as the BMU for that particular datum.

There are numerous measures of similarity available [6, 15, 40], the implementation used in this work is the Euclidean measure. If we represent the input data as d and the prototype vectors as p, their i^{th} attributes will be labelled as d_i and p_i respectively. The similarity measure S can therefore be evaluated as a Euclidean measure over n attributes. This equates to the square root of the sum of square differences between the n attributes:

$$S = \sqrt{\sum_{j=1}^{n} (d_j - p_j)^2} \tag{2.3}$$

In the implementation of the algorithm, the computationally expensive square root function has been omitted, for our purposes, the result of comparing the squares of numbers instead of comparing the numbers themselves is sufficient. The ordering, for all positive numbers, is preserved though not necessarily the relative magnitude. Since we are solely interested in which is numerically larger, omitting the square root will not alter the determination of the BMU.

2.4.3 The BMU Neighbourhood

Once the BMU has been found for each element of the input dataset, we have to determine the region of the map that should be adjusted to better resemble the candidate datum.

The BMU neighbourhood is used to tune the region surrounding the BMU to the properties of the input space, which trains the SOM to have a region sensitised to similar inputs. Initially, to help order the complete SOM, it is desirable to have the tuning process, for each input vector, affect a significant portion of the map. For this to happen, the initial BMU neighbourhood size should be roughly equivalent to the complete map. After a sufficient number of iterations (Kohonen suggests 1000 or so [39]), it is beneficial to reduce the range of influence of the input space. This allows the map to slowly develop regions that are more finely tuned to specific classes of input attributes. At the same time, data elements can migrate to more representative BMU regions if their current associations are less and less representative. Over numerous iterations of the algorithm, the neighbourhood size is slowly decreased until it reaches the size of a single SOM cell. This promotes the algorithm to establish, at the node level, a region of prototype vectors which best represents a distinct class of objects.

Starting from an initial radius of σ_0 equal to the majority of the map, the radius is decreased based on the number of remaining iterations [40]. Hence for the i^{th} iteration of a total of λ, we can calculate the BMU neighbourhood radius as:

$$\sigma_i = \sigma_0 e^{-i/\lambda} \tag{2.4}$$

For each iteration, this equation shrinks the range of influence of each object of the input space as it is re-introduced into the SOM.

2.4.4 The Learning Rate

When an input datum is associated with a specific BMU, the weights of BMU's neighbourhood prototypes must be adjusted to the new input. The extent to which we allow the adjustment to happen affects how quickly the map can approach an optimum solution.

In the early stages of building the SOM, large adjustments to the prototype weights allow the BMUs to migrate towards the attributes of the input data more quickly. These coarse adjustments permit a more rapid adjustment of the map's attributes. This, in turn, allows the BMUs to better represent the input data.

As the algorithm progresses, reducing the intensity of the learning or the coarseness of the adjustment will make the map more sensitive to small fluctuations in properties. These smaller variations in properties might help identify separate classes. We therefore allow the learning rate to vary with an exponential decay function similar to that of the neighbourhood size.

$$R_i = R_0 e^{-i/\lambda} \tag{2.5}$$

This equation [40] gives us a measure of the intensity of the learning component of the algorithm R_i, relative to the initial learning rate R_0, for the i^{th} iteration of λ.

2.4.5 Distance Effects

When weights are adjusted for the BMU, the intent is to make that specific neuron more attuned to the unique input it represents. It is also desirable to enhance the immediate region surrounding the BMU to like elements of the input space, which enforces the topology preserving properties of the SOM. If the strength of the adjustment is made uniformly over the BMU neighbourhood, the tuning effect is diluted over the region. We therefore introduce a distance-dependent factor to the weight adjustment scheme. As with the learning rate, this weighting factor must also decrease in strength as the algorithm progresses through each iteration.

If we define the Euclidean grid distance between the cell being adjusted and the BMU as g, we can write the distance modifier as [40]:

$$\Theta_i = e^{-g^2/2\sigma_i^2} \tag{2.6}$$

We can see that in this equation the strength of the adjustment is proportional to the ratio of the distance from the BMU and the maximum radius of the BMU region as determined by Equation 2.4.

2.4.6 The Mapping Equation

The above sections have introduced a number of factors which combine to influence the amount that a prototype's attribute weights are adjusted. We can now combine these findings into an final expression that can be used to describe the adjustments:

$$W_a' = W_a + \Theta_i R_i((D_a - W_a)) \tag{2.7}$$

In this equation [40], W_a' represents the new weight to be assigned to the prototype vector's a attribute. It is based on the attribute's old value W_a and the attribute value of the data item D_a. The extent of the adjustment is moderated by both the distance from the BMU using both Θ_i and the learning rate R_i. We can then express Equation 2.7 as a general expression for the attributes, but in terms of iterations:

$$W_{i+1} = W_i + R_0 e^{-i/\lambda} e^{-g^2/2\sigma_i^2}((D_i - W_i)) \tag{2.8}$$

In the Kohonen algorithm [40], only cells within the BMU region are affected by these weight adjustments.

2.4.7 Summary

In this section we have examined the moderators used in adjusting SOM cell weights based on the input data. These influences combine to give us a framework which we can use to implement the Kohonen algorithm and create Self-Organizing Maps.

The SOM algorithm maps a high-dimensional space into a lower dimensional space. This process is undertaken to facilitate the extraction of useful relationships within the data that are invisible in the original dataset. A proper topology preserving mapping will group like objects in the original space to closely neighbouring regions in the map space. The next section will cover techniques which can be used to discover patterns and relationships within the map. These patterns will help identify regions of the map that represent clusters of like objects which are of interest and potentially new unknown relationships.

2.5 Interpretation of the SOM

The objective of machine learning is to discover patterns and relationships within a dataset. These features can be leveraged to group like objects into distinct clusters. Further analysis can then focus on the characteristics of the cluster and not peculiarities of each individual object. For very large datasets, the computation cost of standard clustering techniques becomes significant [60, 69]. In such cases it is possible to use a SOM to not only reduce the dimensionality of the dataset but to also reduce the number of objects which require clustering. This allows a large volume of data, the input space, to be analysed through the investigation of the properties and relationships that exist between the prototype neurons in the SOM grid.

In the analysis of a SOM, groups of similar prototypes, representing multiple similar objects from the original dataset, will belong to the same cluster. Conversely, prototypes that are dissimilar should exist in their own separate clusters. Clustering techniques therefore need only investigate as many prototypes as make up the SOM and not the dimensionality of the original dataset.

There are many techniques that have been developed to help discover existing clusters in a dataset. It is sometimes possible to identify clusters of nodes through the visual inspection of the map. If the cluster boundaries are not very well defined, determination of cluster membership can become a subjective exercise. The methods described in the next few sections have evolved to help alleviate most of the subjectiveness surrounding the determination of cluster membership.

2.5.1 K-means

The k-means Algorithm [62] is based on the creation of a collection of k prototype cluster centres. The cluster centres are then compared to the SOM's prototype neurons and prototypes are assigned to cluster centres based on a similarity measure. Once the map prototypes are all allocated, they are used to generate new cluster centroid values and the group averages of the properties are then assigned to their respective cluster centroids. A new iteration is performed comparing the map neurons to each centroid. If required, cluster memberships are updated to better reflect similarities. The process is repeated and memberships adjusted until no further changes occur. When the clustering reaches a steady-state, the cluster centroids represent a set of prototypes which best describe the SOM.

This algorithm requires prior knowledge of the number of expected clusters within the map and fixing the initial number of clusters may introduce some bias into the process. If there is a known number of clusters, this algorithm may be appropriate. However if the actual number of clusters is unknown, fixing their number may results in some more subtle aspects of the map may be missed. In such cases, the initial number of assumed clusters can be varied and the algorithm processed a number of times. A measure would have to be developed to quantify a quality of clustering before a winning configuration can be determined. The measure would have to include some evaluation of the homogeneity of the clusters and would necessitate significant domain knowledge of the underlying dataset. One such measure is to calculate the *sum of the squared error* (SSE). This is calculated as the sum of the square of the differences between a cluster centroid and all of its members. The smaller the SSE, the closer the k-means centroids represent their data elements. If we compare multiple k-means calculations with differing number of centroids, the one which produces the lowest SSE yields the best representation of the data. The number of initial centroids must, however, be reasonable for the dataset being examined. Choosing an artificially high number of centroids will produce an artificially low SSE. The choice of the number of centroids should rely heavily on domain knowledge.

2.5.2 Single Linkage

The Single Linkage [62, 69] approach to clustering is an agglomerative hierarchical clustering technique. One benefit of this approach is that no assumptions are made for the number of clusters present in the data. Every prototype neuron in the SOM is initially considered an individual cluster, clusters are then joined together based on their similarity.

The first iteration will see the two neurons in the SOM which are most similar, joined together to create a new cluster of two points. The algorithm then proceeds to select the next two most-similar prototypes and joins them together. This can

result in either creating a new cluster or merging prototypes into an existing multi-prototype cluster. One advantage of this technique is that it is not influenced by the grid coordinates of the prototypes being clustered. This has the effect that distortions in the topology of the map are ignored and like-prototypes in disjointed portions of the map are clustered as if they were adjacent. Each iteration results in the merging of two clusters until only one cluster remains.

The benefit of this technique is that it does not require a priori knowledge of the number of clusters. The drawback though is that, when the algorithm terminates, we are left with a single cluster. It is then necessary to review the mergers and decide which mergers are appropriate and which occurred between two clusters which should have remained distinct. At a certain point, the similarity measure becomes too large and the representative clusters are too dissimilar to merge. This cut-off point can be evaluated using the same SSE technique as describe above.

2.5.3 Complete Linkage

The Complete Linkage algorithm [62, 69] is applied in much the same way as Single Linkage. The difference in the two techniques is the selection process for merging clusters.

In Complete Linkage, clusters are ranked based on their maximum distance from all other clusters. With this technique, for every pair of clusters, the similarity between their memberships is compared. Their most distant members are used to set the cluster-cluster distance. A tally is kept of the most dissimilar clusters. When the evaluation is complete, the clusters which are the least dissimilar, with the shortest cluster-cluster measure and hence most similar, are merged.

By selecting the least dissimilar clusters, the algorithm is less sensitive to noise and outliers [62]. Similar to Single Linkage, this algorithm is not bound by a pre-determined fixed set number of clusters of topology preservation. The necessary re-evaluation of the inter-cluster distances and determining the smallest magnitude makes this technique much more computationally expensive than Single Linkage.

2.5.4 Visualizing and Interpreting Clusters

The processing of the SOM and applying clustering algorithms to the resultant map all occur without direct intervention. There are no mechanisms, in the environment used in this work, to view the process as it moves forward. Providing such an environment would have a significant impact on the progress and efficiency of the SOM implementation. This is left for future work outside of the scope of this thesis.

The techniques discussed in the previous sections only provide a mechanism by which prototypes can be grouped or clustered based on their overall similarity. They do not provide a list of labels for each cluster, such as “Sciences” or “History”. These will have to be determined after the process is complete and known data are

presented to the map for classification. The number of clusters discovered within the SOM is also undetermined unless one uses a technique which requires this to be fixed beforehand.

Identification of the clusters will depend on the subject matter expert. A priori knowledge of the data will guide the decisions on how to separate the clusters into meaningful classifications. It is possible to plot the resultant SOM on a grid system using unique colours for each cluster discovered. For algorithms such a Single or Complete Linkage, this would results in a map containing only a single cluster. For such techniques, additional steps are required to separate the single cluster into the proper number of candidates.

A technique for evaluating the clustering process is to generate a *dendrogram*. A dendrogram is a tree-like representation each step of the clustering process. Each prototype vector is represented by its own unique cluster leaf-node. As prototypes are clustered together, branches are used to illustrate the merge. The height of the point where the branches are joined represents the similarity between the two clusters. Eventually, at the end of the process we are left with a single branch which is considered the trunk of the tree.

It is then up to the person analyzing the data to decide which merges should be pruned off into distinct clusters. Merges that are judged to represent clusters which are too dissimilar are used as pruning points. All branches belonging to the pruned region will be given the same cluster label. Once this step has been performed, the SOM can be re-plotted showing the unique clusters found within the data.

Techniques such as the Kohonen SOM do not require pre-training with data of known classification. Once trained, however, having data which represents known classes can facilitate labelling the identified clusters of the map. Leveraging such data will help associate known clusters with known classifications. At the end of the process, clusters which have not been tied to existing classification may indicate new and previously unknown classes or sub-classes. This could then lead to a review of the pruning process or investigation into the newly discovered classification.

2.5.5 Summary

We have examined three different techniques for determining the cluster relationships between the prototype neurons in the SOM. K-means as well as single and complete linkage allow to the grouping of like prototypes into clusters. Dendrograms and other techniques can be applied to the final cluster and map to identify distinct regions of the map which represent the desired classifications. In most cases, knowledge of the data and rough idea of the expected number of classifications are required to fully interpret the results of the SOM process.

2.6 Implementing the SOM

The description of the Kohonen algorithm given in previous sections is fairly straight forward. The concepts of reducing the neighbourhood size and the reduction in how aggressively the algorithm learns, once explained, make some intuitive sense. What is less clear, however, is the conversion of the algorithm into an efficient computer application. In this section we will introduce some of the challenges faced when implementing the algorithm.

2.6.1 Memory Consumption

Current consumer grade computers are significantly more powerful than those available only a decade ago. The amount of physical memory available today outstrips by a few orders of magnitude the amount of disk storage available when Kohonen first introduced the SOM. Expectations are that this trend will continue for the foreseeable future.

Though complex in its structure, a Self-Organizing map does not in itself occupy a significant amount of memory. To efficiently process a dataset through a SOM, the application will have to maintain a copy of the map in memory. For the best performance, it is also desirable to maintain a complete copy of the dataset in memory as well. The SOM algorithm repeatedly adjusts the map by exposing it to the original data, reading the data from disk for every iteration would incur a significant time cost.

Each data element that we want to examine brings with it a set number of attributes A. Each one of these properties will be allocated a unique region of memory. The size S, in bytes of this portion of memory will be dependent on the data type being stored. For each data element, we will require a minimum of $A \times S$ bytes of memory for integer attributes (4 bytes each for a single precision 32bit floating point number). Each prototype neuron in the SOM will also require $A \times S$ bytes of memory.

Table 2.1 illustrates memory requirements for a variety of SOM geometries. Most current computers possess several Gigabytes of main memory. All of the examples shown in the Table 2.1 represent but a fraction of the memory available. What is not immediately obvious though is that for every computation involving each of the data elements, the data must pass through the machine's cache. Even the most cache-rich server class machines rarely exceed 16Mb of cache memory. Hence, though the data may easily fit into the memory of a single machine, there is a performance degradation issue if we can not fit both the data and the whole map into the available cache.

				Map memory (kb)			Data memory (kb)		
				Number of Attributes			Number of Attributes		
SOM Size	**# cells**	**Dataset size**	**Bytes per attribute**	4	8	32	4	8	32
	64	2,000	4	1.00	1.02	8.00	31.25	62.50	250.00
	64	10,000	4	1.00	1.02	8.00	156.25	312.50	1,250.00
8x8	64	100,000	8	2.00	2.03	16.00	3,125.00	6,250.00	25,000.00
	100	2,000	4	1.56	1.59	12.50	31.25	62.50	250.00
	100	10,000	4	1.56	1.59	12.50	156.25	312.50	1,250.00
10x10	100	100,000	8	3.13	3.18	25.00	3,125.00	6,250.00	25,000.00
	256	2,000	4	4.00	4.07	32.00	31.25	62.50	250.00
	256	10,000	4	4.00	4.07	32.00	156.25	312.50	1,250.00
16x16	256	100,000	8	8.00	8.14	64.00	3,125.00	6,250.00	25,000.00
	400	2,000	4	6.25	6.36	50.00	31.25	62.50	250.00
	400	10,000	4	6.25	6.36	50.00	156.25	312.50	1,250.00
20x20	400	100,000	8	12.50	12.71	100.00	3,125.00	6,250.00	25,000.00

Table 2.1: Memory requirements (in kilobytes) for various sample SOM geometries

2.6.2 The Computational Load

The SOM algorithm is performed over numerous iterations until a stopping criterion is reached. For each iteration every data element is presented to the SOM in order to determine its BMU. This means that a similarity measure is calculated for each comparison. The similarity measure involves calculating the Euclidean distance (Equation 2.3) which is made up of a subtraction, a multiplication and an addition followed by a square root. If, as we have previously mentioned we omit the expensive square root calculation, we are left with three floating point operations per attribute.

These operations must be performed for each SOM cell and for each element of the input dataset. Once a BMU is found, its prototype weights as well as those in the BMU neighbourhood, must be adjusted. This involves evaluating Equation 2.4 for each attribute: two exponential functions, a subtraction, and an addition. Table 2.2 summarizes the computational load of implementing the SOM algorithm for a single iteration through the data.

For simplicity, the weight adjustment calculations are taken for the first iteration where we adjust all of the prototypes in the map, which is a slight overstatement of the load. The BMU region is typically circular and maps are usually square or rectangular hence a few prototypes in the corners of the map will be excluded. Our calculation will provide a load estimate for the worst-case scenario, as the algorithm progresses and the BMU region becomes smaller, the contribution to the total number of calculations from weight adjustments becomes smaller as well. In the final iterations, the neighbourhood radius will be one, and only the nearest neighbours of the BMU will be adjusted. At this point in time the computational load of the weight adjustments will be minimal.

If we simply look at the number of CPU *operations* required to perform the calculations required, we can get an estimate for the CPU load for various SOM scenarios. In this example we will use the term operations to represent a "time cost" for the processing of an operation such as addition or multiplication. Since the actual real-time cost of these operations is architecture specific, based on numerous performance enhancement tricks such as instruction pre-fetching etc., we will simply use discrete artificial units of time for our evaluations. In this simple model, we will assign a time cost of 1 to additions and subtractions, 2 to multiplication and 3 to division. It is important to note that this is an oversimplification of the whole process and does not provide an accurate measurement of the time it will take to produce the target SOM. This model ignores memory and cache access times, instruction look ahead and a number of other performance enhancements that are possible. This exercise does, however, allow us to gauge the computational requirements of the two main processes required for map building, namely the identification of the BMU and adjusting the weights in the BMU region. Table 2.2 provides an evaluation of the number of computations required in the first pass through the SOM algorithm for a dataset with 5 attributes.

			Number of operations		
SOM Size (nxn)	**# cells**	**Dataset size**	**Find the BMU**	**Adjust Weights (r=BMU region)**	**Last Iteration (r=1)**
	64	2,000	2,560,064	800	16
	64	10,000	12,800,064	800	16
8x8	64	100,000	128,000,064	800	16
	100	2,000	4,000,100	1,248	16
	100	10,000	20,000,100	1,248	16
10x10	100	100,000	200,000,100	1,248	16
	256	2,000	10,240,256	3,216	16
	256	10,000	51,200,256	3,216	16
16x16	256	100,000	512,000,256	3,216	16
	400	2,000	16,000,400	5,024	16
	400	10,000	80,000,400	5,024	16
20x20	400	100,000	800,000,400	5,024	16

Table 2.2: 5 attribute dataset CPU processing demands for various 2D SOM geometries.

It is important to stress that the numbers presented here are for a single iteration through a 2D SOM. The radius used in all calculations for the initial BMU region is equal to half the length of the side of the SOM. Extending these results to 3D increases these results by a factor of $\frac{4}{3}r$ which is the ratio of the the volume of a sphere of radius r to the surface area of a circle of the same radius.

If, as Kohonen proposes, the training of the SOM might entail 100,000 passes through the original dataset, the computational requirements can be significant.

2.6.3 Parallelization Opportunities

The significant number of computations required to satisfy the Kohonen algorithm present a challenge for implementation. The sequential execution of all operations can require several hours, if not days, to complete on a standard consumer-grade computer. A batch variation of the Kohonen algorithm has been developed to reduce the number of computations required while building the map [40, 42]. It achieves an improvement in speed by not updating the prototype weights until after all BMUs have been identified for the input data elements. Though this improves the speed of each iteration, it does so at the expense of allowing the map to adapt in a more gradual manner to the inputs.

Other approaches to improving the performance of the SOM is to exploit opportunities for parallelization within the algorithm. Lawrence [42] identifies two methods for implementing neural networks. The first, called *network partitioning*,

divides the processing workload across multiple processors. The other approach, *data partitioning*, presents distinct subsets of the input data to separate maps implemented across multiple processors.

The Batch SOM exploits a parallel approach to identify BMUs since the SOM is static until all BMUs are identified. With this approach, both partitioning schemes are possible though Lawrence argues that the data partitioning offers much greater opportunities for scalability. In this case, the exchange of information relating to the SOM updates can wait until the complete input dataset has been processed.

The original Kohonen algorithm, however, sees the SOM adapting individually to each input. This restricts our ability to parallelize the selection of BMUs using the data partitioning scheme as each iteration adjusts the prototype weights and therefore creates a new map space for subsequent data elements.

The algorithm does, however, allow for a certain amount of parallel processing. Unlike the batch SOM approach, we can leverage network partitioning. With this approach, there are two main opportunities. The first is in the selection of an individual input element's BMU. The search space for the BMU covers the whole map and it is possible to subdivide the map into equal portions and present each as an independent search space. The results from all of the searches can then be compared to find the overall BMU. For a simple 2D Cartesian map, this can be accomplished by subdividing the map by rows or columns. When the map is evenly distributed into regions of identical size, the processing time for the search should be identical for all sub-maps.

The second opportunity for a parallel approach is the adjustment of the weights after a BMU is found. Much like the search, it can be broken down into a per-column or per-row tasks and each assigned to a different parallel thread. Unlike the previous opportunity, the process of adjusting the weights of the map uses different resources based on the portion of the map being examined. The BMU region that is adjusted varies in size as the algorithm progresses and rarely covers the complete map. This means that some portions of the maps will not undergo any prototype modification while other regions will see all of their prototypes changed. This leads to an imbalance in the computational resources required by each parallel thread. In the worst case scenario, a single parallel stream will perform all of the calculations. Fortunately, this only happens when the radius is small which means the BMU region only contains a small number of nodes. The general case, however, will see multiple parallel threads sharing the workload which will improve the overall performance of the mapping process.

This 2D model can be extended to our 3D SOMs. The volume of the SOM can be subdivided into separate planes. Threads can then be created to process both the search and weight adjustments as required.

2.6.4 MPI versus OpenMP

There are two main type of parallel multi-processors systems: shared memory or distributed memory. The choice of which to use is driven by the algorithm being implemented. In the case of the Kohonen algorithm, the SOM itself is treated as a monolithic object. All aspects of the mapping process rely heavily on the current state of the prototype vectors and each iteration through the data has the potential of affecting the whole map. As such, any implementation of the algorithm will require all code running in parallel to have simultaneous access to a current image of the map.

In a simple distributed-memory approach to parallelization, each computer has a single processor and its own unique memory address space. An implementation of the algorithm on such an architecture would require a local copy of a portion of, or the complete SOM on each machine. It would also require that each machine possess its own copy of the input dataset or a portion thereof. Each processor would therefore be responsible for a specific region of the SOM or of the input data. Any changes to the local portion of the map would necessitate notifying all of the other processors as their local maps may be affected as well. The processor-to-processor update notifications and their data are implemented using a standard library, for example one known as MPI [42, 49].

In the non-batch version of the SOM, the benefit of sharing the computational load amongst separate memory spaces is lost when the algorithm reaches the stage of updating the SOM prototype weights. As an example, since the determination of the BMUs is based on prototype weights, all machines involved must therefore run in lock-step. Once each processor has determined its BMU, its results must be compared to that of all of the others. This requires each processor to communicate with every other processor. These inter-process data exchanges occur at a rate significantly slower than memory access times. The algorithm therefore is likely to spend a significant portion of its time communicating and less time computing. Implementing a non-batch version of the Kohonen algorithm using a distributed memory architecture is therefore not expected to be very efficient.

The second parallel approach is using a multi-processor system that uses shared memory. Such systems are programmed through the use of libraries which manage the overhead of parallelization. One such library is OpenMP [48]. The benefits of such a system are based on the fact that no inter-processor communications are required while accessing the SOM. All processors involved in performing the mapping process have complete access to a single shared copy of the map. Care must be taken in the implementation of the algorithm to ensure that no two processors are performing updates on a single prototype at the same time. In the algorithm, SOM updates are only performed when weights are adjusted. If, as we mentioned previously, updates are performed on a per-row, per-column or per-plane basis, no contention for prototypes will be experienced. The process of parallelizing the code must therefore properly segment the SOM to prevent overlaps in the updates. Implementations of the Kohonen algorithm using OpenMP are therefore well suited for shared-memory computers.

2.6.5 Performance Issues with OpenMP

Shared-memory computers are convenient in that they remove the need for explicit message passing and program coordination issues encountered on other architectures. Properly allocating unique portions of the SOM to distinct processor removes the possibility of multiple access to the same SOM elements by two processors. There are, however, a few aspects of such environment which must be taken into account when implementing an application.

Though the memory space of shared memory computers is shared, CPU access to specific elements stored in memory is greatly enhanced if the data required is stored in the machine's cache. If a requested data item is not in cache, retrieving it from main memory will incur a significant time cost. Members of the SHARCNET technical staff performed evaluations of the SAW cluster using the *lmbench*[2] tools. They found that access L1 cache items times were on the order of 1.5070ns, L2 cache was 8.7410ns and main memory on the order of 105.1ns [44]. From this we can conclude that if an item is not in at least the L2 cache, accessing that data would take over twelve times longer to access. This illustrates that as much data as possible should be in kept in cache. This is especially true for the SOM itself. If the SOM is not cache resident, then finding BMUs and adjusting weights will suffer a significant performance hit.

The dataset is typically much larger than the SOM. If the implementation of the code forces a large portion of the data to be stored in cache, it may cause portions of the SOM out of cache. This will have a significant negative impact on performance. The parallel portions of the algorithm should then be designed to operate on a per-iteration per-data element approach. This would see a minimum of the input data being pre-fetched in cache while having the SOM access more frequently, improving its chance of staying cache resident.

Another significant performance impact can come about from overhead in the OpenMP library. When a portion of code is parallelized, overhead occurs as the system creates threads to handle the requested task. The more often threads are created, the more overhead is incurred by the application and the more time is spent on thread creation and tear-down. To best manage these performance costs, the portions of code that are to be parallelized must attempt balance the thread creation rates and the gains achieved from multiple threads.

As an example, we can examine a simple 3D SOM consisting of a set number of j planes. Each plane in the SOM consists of n rows and m columns. Let us say that $j < n < m$. If the parallelization were implemented to maximize parallelism for the m items to be evaluated, the routine would create $j \times n \times m$ processes. If, however, the parallelization created a thread to process each plane, only j processes would be created. This would save the equivalent of $n \times m$ time the creation/teardown costs. Even for a small $5 \times 7 \times 9$ SOM, this represents a decrease from 315 threads

[2]The LMbench - Tools for Performance Analysis are available at http://lmbench.sourceforge.net/

to only 5, a factor of 63 reduction in overhead. Depending on the thread capacity of the system used for the computations, creating $j \times$n threads may be an equally effective variation with only a factor of n increase in overhead. This overhead may be less than the efficiency gained through additional threads resulting in a more efficient implementation.

2.6.6 Summary

This section discussed a number of important aspects which could affect the performance of the Kohonen algorithm's implementation. The shared and distributed memory approaches were compared. OpenMP was found to be more applicable to the efficient performance of this implementation of Kohonen Maps. Other aspects of the application design were also examined in terms of their possible impact on overall performance.

2.7 Selection a Subject Matter to Model

Data mining has always been an important part of astronomical research [2, 8]. Examples of this include astronomers examining photographic plates for objects of interest as well as researching ancient manuscripts for celestial events. As the volume of data increased and in order to mine existing data, automation of data collection started. As an example, the Automated Plate Scanner Catalog started a project to digitize the National Geographic-Palomar Observatory Sky Survey photographic plates for both the blue and red portions of the spectrum [59]. As early as 1992, neural networks were implemented for the automated analysis of the images to separate out stars from other objects.

Machine learning algorithms have long been applied to astronomical research [9, 23, 68]. Techniques such as ANN and SOM offer possible avenues in automating the classification of objects. Though a significant portion of the application of machine learning focuses on examining data through data mining efforts, some neural networks are applied directly to the raw data as they are acquired. It is known that some celestial events, such as gamma ray bursts, are precursors to other events of interest. Machine learning algorithms have been adapted to monitor streams of raw data from some instruments and generate alerts for events of interest [7, 9]. When such alerts are raised, additional resources can be brought to task for a more thorough examination of the target.

In most cases, the time-sensitive nature of the machine learning techniques is not as critical. As sky surveys become more expansive and farther reaching, they witness more and more objects. It is of great interest to apply automated processes to these objects as their rate of discovery far outstrips manual classification capacities [53].

One area of astronomy which relies on accurate morphological classification of galaxies is the creation of evolutionary models. It is believed that as galaxies age, they transition from one observable morphology to another [13, 68]. The study of the mechanisms involved and their effect on these transitions, or evolution, is an active area of research. To generate an understanding of the various stages of evolution, it is important to study galaxies of various morphologies. As the rate of discovery of new galaxies is increasing all of the time [2, 41], it is important to find an accurate automated process which could help assign morphologies to each.

The focus of the research in this thesis is to apply the technique of Self-Organizing Maps to the automated classification of galaxies. The following sections will describe the reasoning behind this choice as well as the properties of galaxies which are deemed important in identifying morphologies.

2.7.1 Galaxy Evolution

Observations of galaxies have revealed that they come in various shapes and manifest distinct attributes. In an attempt to create some order out of the observations, Edwin Hubble proposed a galaxy classification scheme, commonly referred to as the Hubble Tuning Fork, in 1926 [30]. The classification scheme not only proposes a way to separate out galaxies but it also hints at the possibility that galaxies evolve from one type to another. An example of the original hubble tuning fork diagram is shown in Figure 2.6 [68]. There have been numerous adaptations and proposed changes to the approach since then [13, 68]. These schemes have been proposed to adapt Hubble's original work to account for newly observed galaxy morphologies.

The original Hubble classes were derived from observations performed in the visible range of the light spectrum. The process was laborious and the results were often based on subjective evaluations of what could be derived from the photographic plates. Results often depended on the interpreter and could vary significantly from one specialist to another [25]. The objective of all of the classification schemes is to group galaxies into families. These classes and sub-classes are proposed as a model which can be used to study the evolutionary life cycle of galaxies. Astronomers could then use these classes to map out the different stages of galactic evolution similar to what they have done for stars in the H-R diagram [10, 41]. In this chapter we will examine different properties that can be used to describe a galaxy. From this, we will propose an approach which could be effective in mapping the different stages of a galaxy's life cycle.

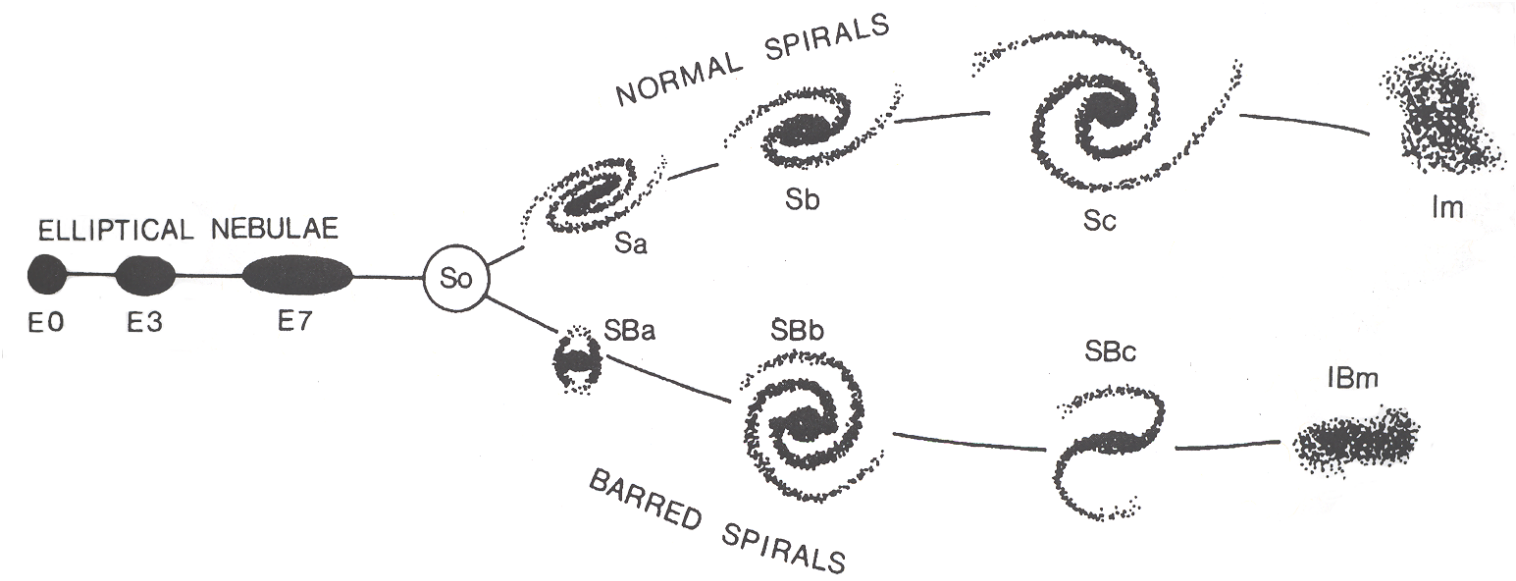

Figure 2.6: The Hubble Tuning Fork Diagram.

2.7.2 Galaxy Attributes

In the simplest of models, a galaxy is a collection of stars and interstellar material. The constituent stars are expected to be similar to those observed in our own galaxy. Stellar evolution has been extensively studied and modelled, resulting in the creation of the H-R diagram [13, 72]. This model has proven to be quite successful in describing the life cycle of stars. Younger galaxies would be expected to have younger, bluer stars. Conversely, older galaxies would present a larger fraction of older, redder stars. It can be argued that, in a isolated and relatively static galaxy, the populations of young and old stars could be used as a direct measure of its age. If according to our models such as the H-R diagram, a star requires eight billion years to reach a certain stage in its evolution, then observing such a star population in a distant galaxy would provide for a lower limit to the galaxy's age.

The interstellar material present in a galaxy is made up of planets, dust and gas. These are not, in themselves, capable of creating a significant, detectable amount of radiation from internal processes. These objects can, however, affect our detection of the radiation signals generated by the local stellar population [33]. The interstellar material can reflect or absorb portions the signal. If absorbed, this radiation can also serve to excite this interstellar material. It would then follow that once the material is sufficiently heated, it would emit radiation of its own. This radiation will be characteristic of the substances itself. These re-emissions are responsible for signals in the infrared portion of the spectrum.

Our ability to observe galaxies is dependent on our ability to measure their properties. Using instruments specifically crafted for this purpose, we can measure the amount of energy being received from each galaxy. In contrast to early investigations, modern instruments allow us to study galaxies in wavelengths which span the electromagnetic spectrum. The extended range of frequencies available through modern instruments allows for a more comprehensive representation of the physical processes at work within each galaxy candidate. Other tools allow us to measure the size and distance of each galaxy. Leveraging these various techniques allows for the direct comparisons of attributes between individual galaxies. It also permits the grouping of galaxies into families which possess similar properties.

2.7.3 Measuring Radiation

The original photographic plates used in astronomy provided a method of capturing the light emitted from galaxies. Depending on the chemicals applied to the plate and wavelength filters applied to the optics, it was possible to affect the plate's sensitivity to specific wavelengths. The darker images on the plate indicated that more chemical reactions took place implying that more radiation was received. From this, astronomers are able to get a sense of the brightness of a galaxy for the specific optical region under study. The brightness can then be compared to known

standards and a magnitude measurement determined for the galaxy. The observed galaxy would exhibit different magnitude values depending on the frequency of the light observed.

To extend the range of frequencies that can be studied, detectors capable of measuring radiation intensities outside of the visible spectrum have been developed. Also, a number of observatories have been launched into space to overcome the limitation to ground-based observations caused by the atmosphere. Currently, it is possible to measure the radiation received from a galaxy from the radio region into the γ-ray range. Measurements taken with these instruments record the amount of radiation per unit of surface over the detector. This quantity of energy per unit surface area is know as the *flux*. To facilitate the comparisons between the amount of signal received at different wavelengths, astrophysicists use a unit of measure to represent the flux density of a signal. This is a measure of the flux per frequency interval of the radiation being observed, essentially a measure of the radiation received per oscillation of the electromagnetic wave. The unit of measure used to describe flux density is the Jansky and is equivalent to 10^{-23} erg $s^{-1}cm^{-2}Hz^{-1}$.

2.7.4 The Observed Spectrum

Most objects either emit, reflect, transmit or absorb the flow of a detectable quantity of radiation. The amount of interaction between the radiation and the object is dependent on the composition of the object as well as the frequency of the radiation. Further, radiation might also be scattered as it flows through objects. Our ability to witness the presence of an object is dependent on our ability to detect this radiation. Extracting the qualities of the detected signal allows the determination of the physical processes occurring at or surrounding the source.

The stellar populations within galaxies generate a combination of measurable forms of radiation and the various stages in star development provide identifiable signatures that can be detected, measured and classified. Within the host galaxy, the radiation generated by stars is also subject to absorbtion and scattering. The effects of these processes on the incident radiation are dependent on the nature of the incident object; they can result in the generation of secondary emissions of radiation. The properties of the induced emissions will be indicative of the makeup of the material affected.

The resultant radiation emitted by a galaxy will therefore be a combination of the spectra from the various star populations, emissions from interstellar material as well as contributions from other more exotic processes. Components of the resulting spectrum can be used to identify specific physical processes that are taking place within each individual galaxy. As such, analysis of the emitted radiation and its intensity at various frequencies can be used to identify specific characteristics of the target galaxy.

2.7.5 The Spectral Energy Distribution

The collection of frequency specific measurements for a galaxy can reveal a great deal about the galaxy itself. The intensity of the individual signals allows us to calculate the amount of energy being emitted by the galaxy at each specific wavelength. A graph of the amount of energy being emitted by a galaxy versus the wavelength is known as a *Spectral Energy Distribution* or SED.

The SED of a galaxy is, as we have mentioned previously, depends on the characteristic processes occurring within the galaxy. As galaxies age it would be expected that the types of processes that can take place would change over time. Certainly, based on the H-R diagram, one would not expect to see indications of the late stages of a star population's life cycle to occur in a target galaxy unless the galaxy itself was at least as old as required by the stars. This would lead to the notion that most galaxies would exhibit different SEDs, not only from each other but one should also expect the SED of a galaxy itself to change over a long period of time. Like galaxies of like age should present similar SEDs if their internal constituent star populations are compatible. Conversely, galaxies of the same age but of very different morphology may provide substantially different SEDs. An opportunity exists where the SED of a number of galaxies could be compared and this might lead to some insight into their evolution.

Different models are being developed to reproduce SEDs based on known stellar processes [20, 72]. These models not only take into account stellar evolution, they also include effects on the SED caused by interstellar gas and dust as well as photometric redshifts.

Figures 2.7, 2.8, and 2.9 are examples of SEDs obtained from the *NASA Extragalactic Database* (NED) [47]. Examination of these figures shows distinct differences between the SED. The differences are not limited to the intensity of the measurements but to the shape of the overall curve as well, which is an indication that the SED may have the potential to lead to a classification system for galaxy morphologies. It is also important to note that the scatter in the data plotted at any one frequency is probably a better measure of the error present in the data than is given by the error bars present for any individual measurement.

2.7.6 Distance and Velocity

Hubble's other major contribution to astronomy was the observational verification of a relationship between the distance of a galaxy and its radial velocity [43]. Measurements of the distance to several galaxies when plotted against their velocities showed that with the exclusions of a few members of the local group of galaxies, the further a galaxy was, the faster it was moving away from our own galaxy.

This distance-velocity relationship has a direct impact on the study of galaxies. Large, automated galactic surveys do not allow for the detection and measurement

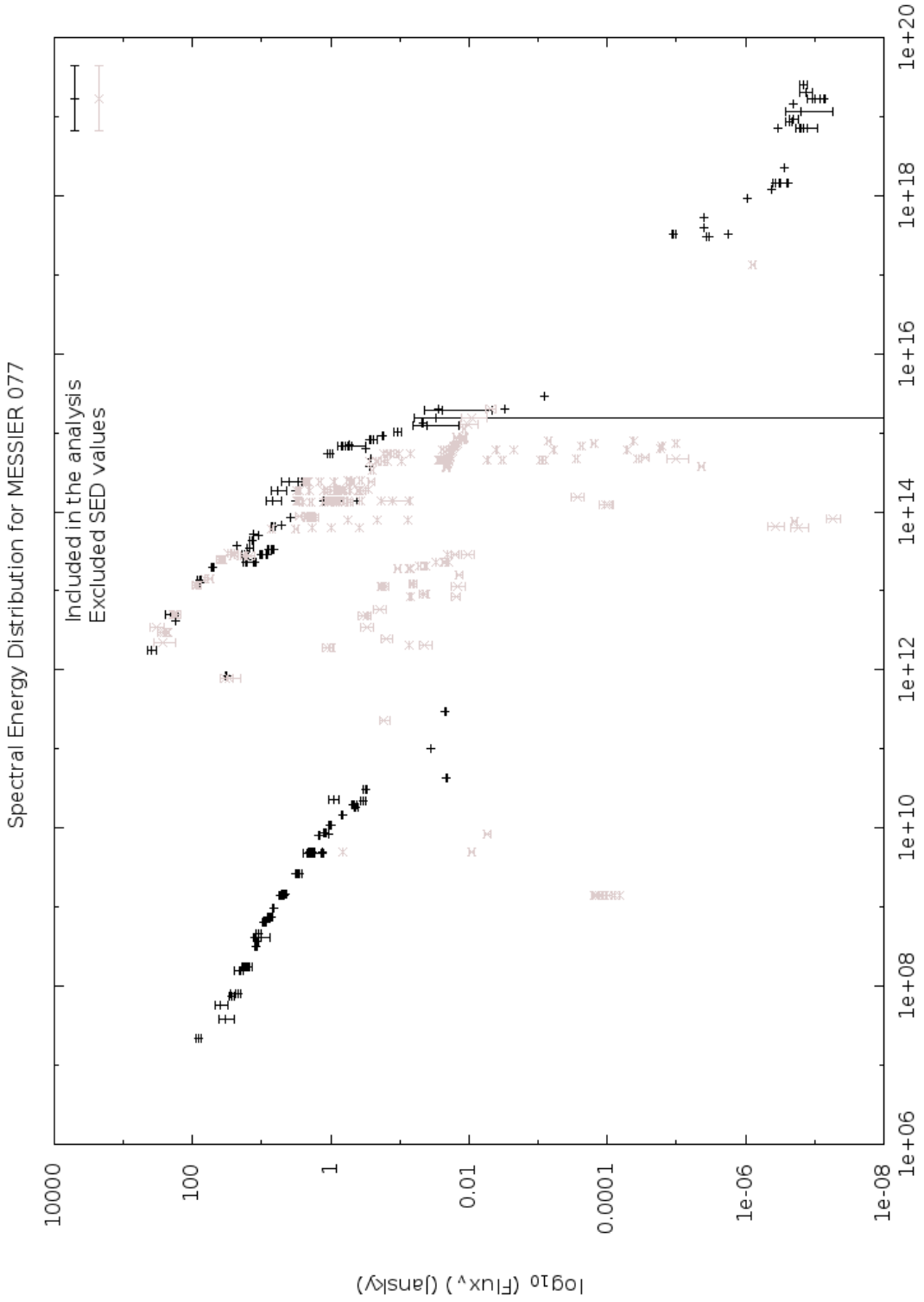

Figure 2.7: The SED for Messier 077, a spiral galaxy. (Exclusions are identified in section 3.3.4)

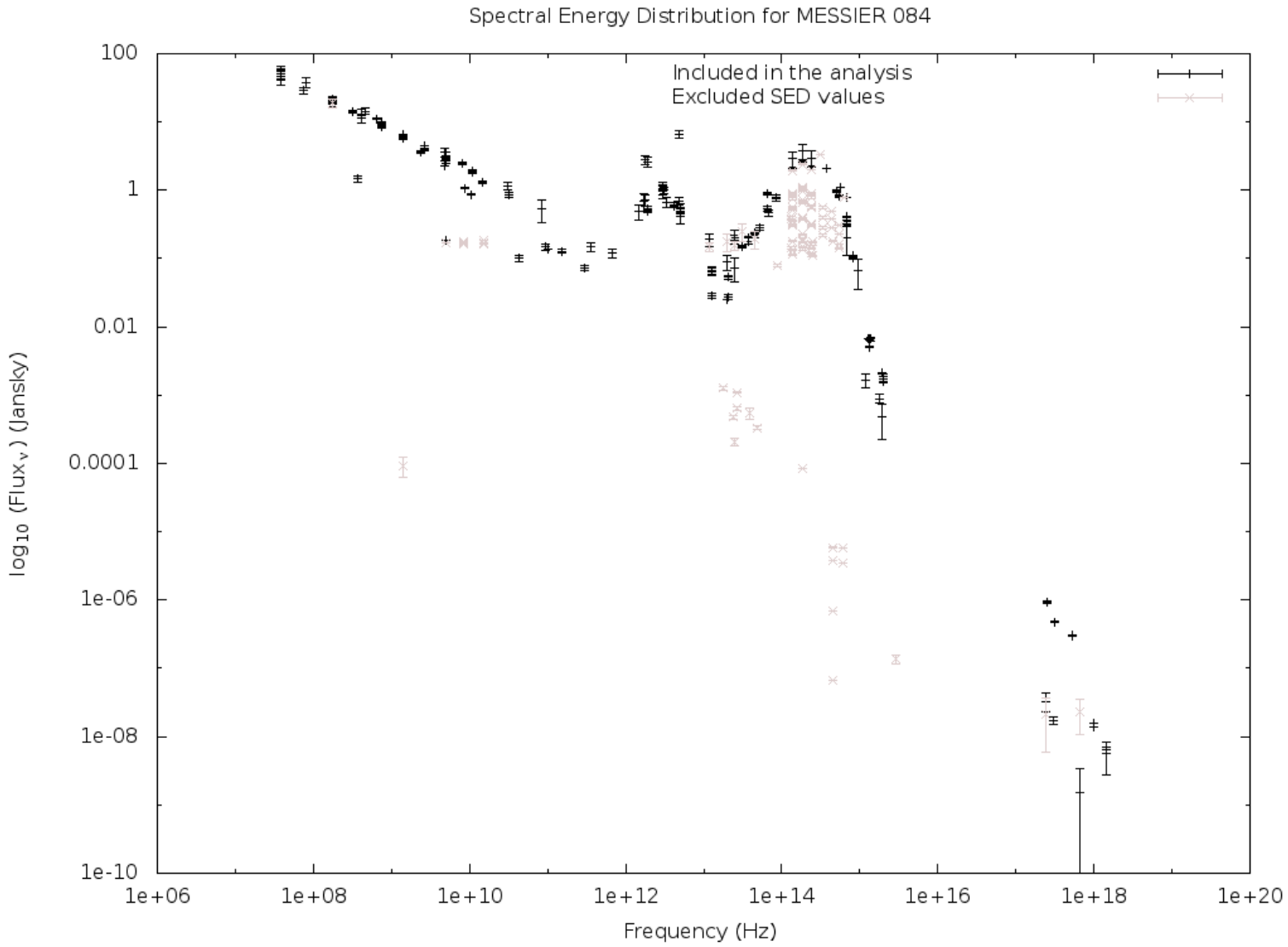

Figure 2.8: The SED for Messier 084, an elliptical galaxy. (Exclusions are identified in section 3.3.4)

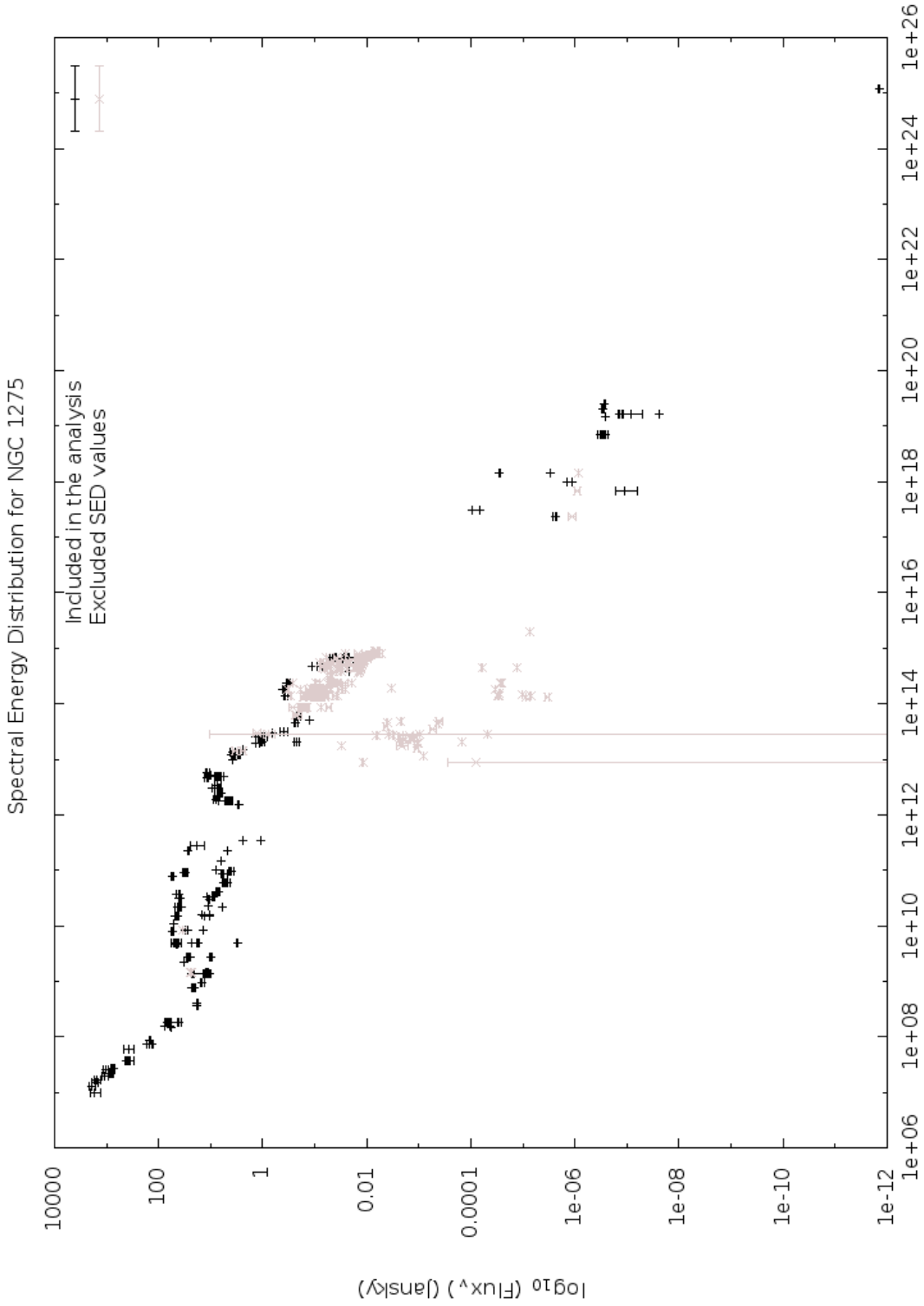

Figure 2.9: The SED for NGC 1275, a peculiar galaxy. (Exclusions are identified in section 3.3.4)

of standard distance markers to estimate the separation between each and the earth. Using the Hubble relationship, such distances can be determined.

Similar to the Doppler shift for sound, the velocity at which an object is moving in relationship to the earth affects the wavelength of the electromagnetic wave measured. For objects moving towards the Earth, the wavelength appears to be shortened. Similarly, for objects like galaxies which are moving away from us, the wavelength appears elongated by an amount proportional to its velocity. For low redshift objects, the effect is below a detectable limit. For larger velocities, the effect can become significant. Velocities of distant galaxies and hence their distance, can be determined using this Doppler shift and the *Hubble Relation* in Equation 2.9 [33].

$$cz = H_0 d \text{ km s}^{-1} \tag{2.9}$$

where d is the distance, H_0 is called the Hubble Constant with a value of $71 \text{ km s}^{-1} \text{ Mpc}^{-1}$ and z is obtained from:

$$z = \frac{\Delta\lambda}{\lambda_0} \tag{2.10}$$

where $\Delta\lambda$ is the shift in wavelength and λ_0 is the source wavelength. If spectral lines of known origin can be found in the spectrum of a remote galaxy, calculations can be performed on the amount of redshift observed to obtain a radial velocity.

2.7.7 Corrections

The radiation signal emitted by the target galaxies must travel from the source object to the detector and the distances over which the signal must travel will impact our ability to measure its intensity. Also, the pathway the signal follows is not devoid of sources of interference. These two factors must be accounted for before we can attempt to compare galaxies to one another.

The flux density of a detected electromagnetic signal is inversely proportional to the square of the distance between the source and the detector. To enable direct comparisons between measurements made of galaxies known to be at different distances, we must adjust our data. For the purpose of this work, all measurements were adjusted to a standard common distance of 10MPc (see Section 4.3, Equation 4.3).

There is an inherent lower limit to a device's ability to detect a signal. Much as the human eye is unable to see many faint astronomical objects, modern detectors also have their limitations. Due to this sensitivity limit, faint objects may be undetectable. This bias is known as the Malmquist bias [63], and means that at greater distances only brighter sources are seen, not fainter ones. Also, different studies use different types of detectors, which can lead to bright objects in one frequency band being invisible to detectors in other bands.

We know from studies within our own galaxy that the signal received from objects is affected by the interstellar medium present. This effect is known as *extinction*. Though we have no techniques to evaluate the level of extinction occurring within distant galaxies, we can compensate for signal degradation caused by dust within our own galaxy which has been reasonably well mapped. The internal dust distribution in the Milky Way is not uniform, however, studies [28] have measured the amount of signal extinction in various portions of the sky. We can use these known quantities to further normalize the signal strength from each individual galaxy. Extinction affects mostly the frequencies in the visible, U.V. portions of the spectrum [28, 32]. We must therefore compensate by different amounts for the various frequencies observed.

The last correction to the data that will be investigated, known as the *K correction*, is one which attempts to compensate for the effects of the expansion of the universe. The Hubble relation related a measured redshift to a distance measure. Redshift affects the SED that is measured from each galaxy. The SED is a measurement of the amount of energy received from an object plotted against the frequency of the observation and the frequency of a measurement is inversely proportional to its wavelength. If a redshift is measured and found to affect the wavelength by say five percent, the impact on the frequency will be different in different regions of the spectrum. For a very long wavelength the effect is minimal on the frequency. For wavelengths several orders of magnitude shorter, which are common in SEDs, the shift in frequency will be significantly different. The implication of this effect is that observations made at specific frequencies may in fact be measuring different regions of the SED depending on the redshift of the target galaxy. This implies that if we were to observe the same galaxy's SED at two significantly different redshifts, the SEDs would appear quite different. The K correction attempts to compensate for these redshift-induced distortions [18].

Different approaches have been taken to evaluate the K correction. Pence [50] presented his results based on observational data. Blanton and Roweis [5] present an approach based on templates derived from stellar populations. Chilingarian [11] created a series of polynomials which allowed for the approximation of the K correction based solely on a redshift value and observations at a single frequency. It is this paper which we will use as a reference for correcting out SED data.

2.7.8 Summary

The concept of galaxy classifications and evolution was refined by individuals such as Edwin Hubble [30] and Sidney van den Bergh [68]. A proper understanding of the differences between galaxies is important in creating a model of their evolution. Galaxies are classified based on their detectable characteristics.

In this chapter we have discussed the methods and challenges encountered when collecting qualitative data from galaxies. Measurements are affected by the limitations of the detectors, extinction, distance and the opacity of out atmosphere are certain wavelengths. Careful corrections must be made the the observations to ensure that like attributes can be compared on an equal basis.

2.8 Conclusion

In this section we have introduced the concept of machine learning. It is expected that the implementation of such techniques will help provide a consistent interpretation of the data being modelled. Once properly trained, the SOM should provide for rapid and efficient classification of new data.

Some of the major ideas governing the creation of Self-Organizing Maps were presented and the mathematical and computational aspects of the algorithm were investigated in order to identify optimization opportunities though parallelization efforts.

This chapter also focused on an area of study upon which we will leverage our interpretation of the Kohonen map: a model for the evolution of galaxies. The characteristics of galaxies, which we shall use as attributes were identified. We discussed the spectral energy distribution of galaxies which will be obtained and corrected for distance and extinction. These will form the data upon which we will create our classification model.

In the following chapters, we will investigate the acquisition of our data and our selection of valid candidates for the study. We will also undertake an investigation of the appropriateness of various SOM configurations.

Chapter 3

Model and Data

3.1 Introduction

The previous chapter provided an introduction to the concepts of machine learning and Self-Organizing Maps. Details were given on how, in mathematical terms, the SOM is tuned to best reflect the input data's attributes. The following sections will concentrate on how the algorithm was implemented for use within this thesis. Emphasis will be placed on various techniques used to make the application more efficient and reduce the overall run time involved in building individual maps.

This chapter will also describe the dataset that was acquired for study. The study of galaxy evolution will be conducted through an analysis of galaxy SEDs. To perform this study, a significant amount of information will have to be collected to cover the known galaxy types as well as much of the electromagnetic spectrum for which data is available. In the second section of this chapter we will investigate different sources of data for our study as well as the processes used to acquire the data.

3.2 Implementing the Self-Organizing Map

The SOM algorithm was implemented using a custom application written in the C language. There exist a number of utilities such as R and MatLAB which provide SOM packages. The decision to write a distinct version of the SOM was based on the desire to gain detailed knowledge of the details of implementation. It also allowed the opportunity to investigate variations to the standard SOM geometry. Finally, the code allowed for a study of different map initialization techniques not found in off-the-shelf packages.

3.2.1 Geometry

The SOM algorithm allows for the mapping of a high-dimensional space representing the data's attribute space, to a typically two dimensional map. In the mapping process it is possible that some of the information contained in the data is obscured by other artifacts in the map. The majority of the datasets used in this work were comprised of multiple attributes. When these datasets were examined using PCA, more than three eigenvectors were identified. For this reason, it was decided to extend the normal Kohonen algorithm to a three-dimensional map to allow the expression of more eigenvectors in the resultant map. The SOM algorithm will still be mapping a high dimensional space to a lower one. By expanding the map space to three dimensions it is expected that we can enhance the SOM's ability to resolve object classes.

The 3D interpretation of the SOM was restricted to investigating maps on a regular Cartesian grid. Other geometries such as toroidal or spherical were not included due to their inherent difficulties in positioning nearest neighbours at uniform distances from each other [29, 56, 74]. Similarly, extending the hexagonal grid commonly used in 2D maps was not possible as the shape does not translate to 3D without introducing gaps in the map coverage.

Extending the map to a third dimension has a direct impact on the number of nearest neighbours. In a 2D Cartesian grid arrangement, each prototype can be represented by a single grid square. A prototype would have four neighbours adjacent to the four faces of the square. These would be neighbours at a distance of one grid unit. The prototype would also have four neighbours which touch it at the corners. These would be at a distance of $\sqrt{2}$ from the prototype. In 3D, the number of nearest neighbours soars to a total of 26. This is made up of six neighbours along the faces of the now cubic prototype, twelve neighbours that are co-planar along the edges of the cube and another eight adjacent to the vertices of the cube. The prototype would then have six neighbours at a distance of 1, twelve at a distance of $\sqrt{2}$ and eight more at a distance of $\sqrt{3}$. An illustrative drawing is shown in Figure 3.1.

Extending the map to a third dimension changes the map from a surface topology to that of a volume. The direct impact is more than just increasing the number of computations by the size of the new dimension. Interactions between the "layers", which can be represented by the standard 2D arrangements, also come into effect. The increased number of nearest neighbours does not directly affect the determination of the BMU. This calculation is directly proportional to the size of the third dimension, however, the stage of adjusting the weights changes drastically. This is where BMU regions in the bulk of the volume affect prototypes in adjoining layers. Instead of updating the prototypes which fall within a region of the map defined as $4\pi r^2$ in the case of 2D maps, we now must consider a membership which spans the volume of $\frac{4}{3}\pi r^3$. Except for just updating the nearest neighbours, this always incurs significantly more computations. The additional computation costs can be extrapolated from the increase in the number of target prototypes summarised in Table 3.1. In this summary we can see that using a nearest neighbour radius as small as 3 increases the workload for weight adjustments from 45 updates to 251. This represents a factor of 557% in workload.

	Conventional 2D SOM		3D SOM		
BMU region radius	**Number of prototypes**	**SOM cell count**	**Cube volume**	**Spherical BMU cell count**	**Percentage increase in workload**
1	9	9	27	27	300.00%
3	49	45	343	251	557.78%
5	121	109	1331	895	821.10%
7	225	193	3375	2103	1089.64%
9	361	305	6859	4139	1357.05%

Table 3.1: Workloads for weight adjustment within the BMU region based on radius.

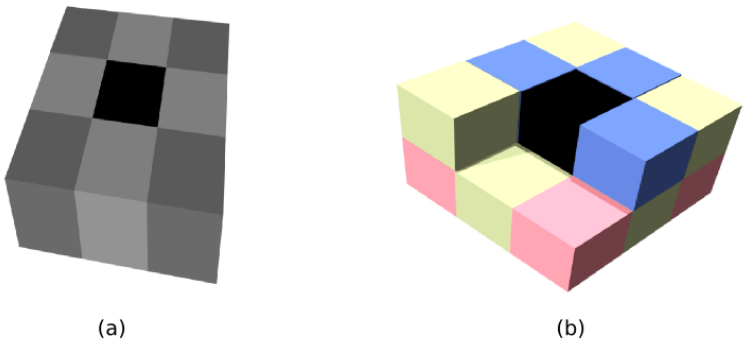

Figure 3.1: Nearest Neighbours in a Cartesian grid. (a) 2D SOM. (b) 3D SOM

3.2.2 Randomizing the Input Data

When data are presented to the SOM each element has an immediate impact on the structure of the map. If the data elements are always presented to the map in the same order it might induce into the map bias for the first few data elements over the others.

As suggested by Kohonen [39], all of the data were presented to the map in a different random order for each iteration. An array of indices was created and sorted into a random order, that array was then processed sequentially to extract the next random data element to present to the map.

3.2.3 Optimizations

There are a number of optimizations that were brought into the code to improve performance. The first was a decision to not compute any square roots. Equation 2.3 describes the calculation for the similarity between a datum and a prototype vector, this equation is used solely for finding the BMU. Determining the BMU is based on finding the most similar measure which equates to smallest similarity measure. In this case, it is not the magnitude of the number that is important, only the ordering. Since comparing numbers or their squares does not change the ordering, for our purposes the calculation of the square root is superfluous.

Square roots can also be eliminated from the process of determining if a prototype falls within a certain grid distance from the BMU. As the algorithm progresses, the BMU region changes in size. Once a BMU has been chosen for a specific datum, all of the nodes within the BMU region have their weights tuned to the new input. For this, we need to scan a volume of the SOM for all prototypes within a BMU radius in all directions. In the Cartesian grid of the SOM, this means that we need to scan in all three dimensions for all nodes within the target zone. Geometrically this equates to scanning a cube of prototypes for those which might fall within the enclosed sphere. In code, this equates to scanning in the x, y and z directions

and evaluating the distance to the BMU for each prototype's coordinates. As in the case of the similarity, comparing the square of the BMU region's radius to the sum of the squares of the prototype's coordinates is sufficient to determine if it lies within the BMU regions or not.

Recalling Table 2.2, we can get an estimate for the overall savings in time by not computing square roots for the Euclidean measure. For each datum presented to the SOM at each iteration, we save a square root for every prototype present in the SOM. In addition, based on the size of the BMU regions for that iteration, we save additional square roots every time we determine if a prototype falls within the region or not.

In Equation 2.7, Θ_i and R_i represents the weighting factor that need to be applied to tune the prototype to the newly presented datum. Θ_i describes the distance sensitive component of the adjustment while R_i describes the learning rate factor. For any one iteration through the input dataset, the learning factor remains a constant. Similarly, the distance-based values for the influence of Θ are constant for the iteration and are only moderated by the distance to the BMU. This allows us to further reduce the number of calculations by evaluating these values once per iteration and not every time a distance evaluation is required.

In a 3-dimensional volume, using discrete prototype coordinates such as in matrix, only discrete distance values are possible. If we evaluate the distances from the origin to the points at coordinates (3,4,0), (0,3,4), (3,0,4) (0,0,5), we find that they are all equidistant. The sum of the squares of the differences in the coordinate pairs with those of the origin all equate to a value of 25. Within the grid structure of the SOM, each one of these points will mirror points relative to the BMU. As an example, if we consider the BMU at the origin, then the point (3,4,1) has distance clones at (-3,4,1), (-3,-4,1), (3,-4,1) and four others changing the z coordinate from 1 to -1. In fact, any point in the first quadrant should have at least one equivalent point within the other eight quadrants around the origin. The discrete coordinate system within the SOM ensures that multiple prototype coordinates have identical distance measures from the BMU. Also, because we are summing the squares of integers, a number of distances will never be encountered. As an example, there are no three integers which once squared, will result in a distance (without the square root) of 7. We can leverage this to reduce the number of overall calculations in the processing of the weight adjustments.

In the implemented code, arrays were created to track which combinations of coordinates yielded valid distances as well as what the weight adjustment factor would be for those distances. The first array, updated at the beginning of the routine, calculated valid distances possible within the constraints of the dimensions of the SOM. For each iteration through the input dataset, a one-time pass was used to calculate the weight factors Θ for the current learning rate and all valid distances. This alleviated the need to calculate the weight factor for each prototype within the BMU region of each datum, saving two exponentiations and a couple of divisions for each weight adjustment.

3.2.4 Leveraging parallel processing

In the previous chapter we presented an overview of both the MPI and OpenMP approaches to parallelization. The MPI based approach was eliminated from consideration because of the complexities of managing and coordinating updates to the SOM for the original Kohonen algorithm used in this research. The frequent changes to the SOM would require the rebroadcast and synchronization of the complete SOM at every iteration and for each datum processed. For these reasons, only the OpenMP approach was implemented.

The objective of parallelization is to break down a large computation task into multiple sub-tasks. These stand-alone tasks could then be processes independently on multiple processors to achieve an overall improvement in performance of the code. The better the code performs, the more quickly results can be obtained. It also allows for the opportunity to explore larger and more complex problems which might have been impractical to run before.

The SOM algorithm on first glance seems to lend itself quite well to the idea of parallelizing its execution. The iterative process of presenting each datum to the SOM, the selection of the BMU and the adjustment of the BMU region weights all appear to be valid candidates. The one overriding constraints on our ability to parallelize the solution is the SOM itself. The objective of the algorithm is that each time a datum is presented to the SOM, the later is enhanced to better reflect the input space. Subsequent updates to the map will therefore be dependent on the order that the updates are performed in. If parallelization was implemented to present BMU region updates to the map simultaneously from unique data elements in each thread, the sequence of the updates would be disrupted. Since the order of the data being presented to the SOM is by design random, this reordering is not problematic. The drawback to this approach is that it will create memory access contention when updating prototypes of the map as the BMU region weights are adjusted. The contention will be greatest at the early stages of creating the map when the BMU regions essentially cover the entire volume.

The second opportunity for OpenMP would be in the selection of the BMU. This step in the algorithm does not modify the map and is therefore a easy candidate to select for parallelization. Without parallelization, the BMU is found by iterating through the SOM through each of the three dimensions. When implementing a parallelized version, care must be taken to balance the number of prototypes being investigated in parallel with the time costs of creating additional thread to perform the task. Recalling from Section 2.6.5 that the outer loop solution would only see 5 threads created for a $5 \times 7 \times 9$. A factor of 63 reduction in the number of threads created. Performance testing revealed a significant impact with the inner-loop solution. In this implementation of the code, the SOM was sliced into 2D layers as shown in Figure 3.2. Each layer was presented to a separate thread. Each thread searched by row and column through its assigned plane. Once all of the threads

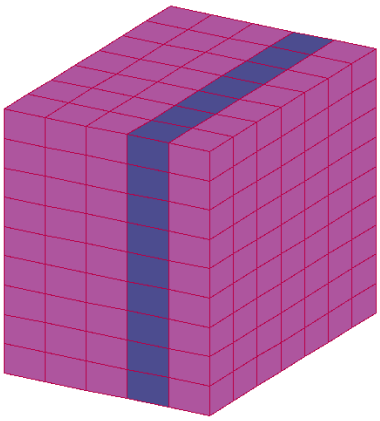

Figure 3.2: SOM processing by assigning the $x = 3$ plane to a unique thread.

had completed, their layer-level BMUs were compared and the winning candidate was elected as the BMU for the candidate datum.

The other major opportunity for a performance enhancement through OpenMP was with the adjustment of weights within the SOM once a BMU was elected. The adjustment of weight for any unique prototype is only dependent on its relationship with the BMU in terms of a distance measure, the weight adjustment is not a function of the weights or attributes of any of the surrounding prototypes. The independence of a prototype from its neighbours allows us to adjust any prototype's attribute weights without having to be concerned about what order the adjustments are being made or adjacent neighbours. The BMU region was processed in much the same way as the BMU was determined. 2D layered slices of the SOM were presented separate threads. Each layer was then scanned for prototypes which might fall within the BMU weight adjustment radius. When a valid target was identified, the weight was adjusted.

3.2.5 The Control File

The processing of a dataset into a SOM requires that attributes be identified for both the map and the data. In the case of the data, it is sufficient to identify the filename where the data resides as well as the number of attributes it contains.

For the SOM algorithm, we need to specify the dimensions of the map, the number of iterations through the data, the initial BMU radius, the learning rate as well as which technique we want to use to initialize the SOM.

An additional parameter used to moderate the selection of the BMU was added to the control file. This parameter, when selected, moderated the process of selecting a BMU. The implementation of this parameter will be discussed in AppendixA.

All of these parameters were included in a control file for the SOM application. This permitted the quick processing of the same data file and SOM geometry for multiple initialization techniques by simply changing one line in the control file.

Similarly, changing the geometry of the SOM for multiple runs of the same data file was just as easy. This facilitated the bulk processing of multiple SOM runs to explore how changes in the initialization or geometry of the map affected the quality of the final results.

3.2.6 Clustering the SOM Prototypes

The final step in processing the data is to provide a way of interpreting the results present in the completed SOM. To this end, a separate application was written to apply the Single Linkage algorithm to the map.

In Single Linkage, the most similar prototypes are clustered together. The process continues until all of the clusters have been merged into one. To facilitate extracting the actual data clusters from the output, a separate routine was written to help generate dendrograms. A sample dendrogram is presented in Figure 4.3. These diagrams help visualize the various levels of similarity that were used to join the clusters together. They can also help identify the levels at which the branches represent distinct classes of objects.

3.2.7 The Runtime Environment

All of the SOM processing for this thesis was performed on resources made available through the *Shared Hierarchical Academic Research Computing Network*(SHARCNET). More specifically, all of the data was processed on the *SAW* cluster.

SAW was chosen as its resources were sufficient to run all of the SOM geometries used in the analysis phase of this thesis. The geometry chosen for this work was such that at most the implementation would use fewer than eight threads. It would therefore be inefficient to run the code on clusters where each node provided more than eight. The additional capacities on these computers would be wasted. The choice of forcing all processing within the same cluster was made to ensure that performance comparisons could be made from model to model. Though the mapping exercises were run on a cluster, no job required more than one node to run on as this would have necessitated using MPI. The convenience of running on a cluster, however, permitted for several families of jobs to be run concurrently on separate nodes.

The cluster of computers is comprised of 336 processing nodes. Each node in the cluster has the following characteristics: 8 cores (2 sockets x 4 cores per socket), Intel Xeon @ 2.83 GHz E5440 processors [35] as well as 16.0 GB of memory. Each processor has 12Mb of L2 cache with an access time measured by *lmbench* of 8.74ns [44].

As an extreme example of memory requirements, processing a SOM which had dimensions of $30 \times 30 \times 30$ would represent $27,000$ prototypes. If we add to this say 5,000 data elements, we would have a grand total of $32,000$ attribute vectors which are the bulk of the memory requirements for the SOM. If each vector is made up of 20 attributes of 16 bytes each, the SOM and data would require about 10Mb of memory. For the Xeon CPUs available on SAW, this fits quite nicely in L2 cache as we will see in Chapter 4

3.2.8 Summary

This section has provided an overview of the process used to implement a version of the Kohonen SOM. The implementation leverages a number of techniques which help reduce the overall number of calculations required. To further reduce the run time of the SOM creation process, the code utilized the OpenMP libraries to help parallelise the code to run in a shared memory environment.

3.3 Data: Galaxy Attributes

Galaxy classifications have historically been driven by observations in the visible range of the spectrum. Hubble's Tuning Fork diagram, as an example, is a classification system based on the observational data available at the time. Since galaxies contain billions of stars, it would follow that many of the observational properties of a galaxy could be deduced from observations of stars. It should also follow that if we know how stars change over time, we could use this knowledge to develop a model for how galaxies might be expected to change over time as well. There are a number of research efforts which attempt to reproduce the observable attributes of galaxies through the simulation of large populations of stars [20, 72]. The success of such efforts indicate that it should be possible to obtain a model for the evolution of galaxies by observing the properties of the stars they contain and not simply their appearance in the visual part of the spectrum. Attributes which have been used as modifiers of the Hubble Tuning Fork, such as bars or no-bars, Spiral or elliptical will not be used in the analysis phase. In this research, we will only use the SED signatures collected from galaxies as a measure of their attributes.

Studies of galaxies in the past were performed on a per-observatory and per-project basis. Each observatory research group would collect a series of image plates and they would be physically stored in a library. Study of the plates required physical access. Today, most of the large-scale studies make their data available to other researchers in electronic format.[1, 2, 3]. Great care is taken at all sites to generate the most accurate measurements possible. The advent of large-scale automated surveys has brought about a deluge of such data. Each of these surveys typically only observe a small portion of the electromagnetic spectrum. Some study the visible, some observe the radio or x-ray region. Due to the vast number of objects included in most large scale surveys, the resulting datasets share one common similarity: the lack of a common naming convention. Most surveys catalogue their results based on the equatorial coordinates of the detected objects. The lack of an authoritative central object catalogue, where each object has a unique agreed upon designation, makes cross-study investigations more difficult. Compounding

[1]European Southern Observatory (ESO): http://www.eso.org/
[2]The Sloan Digital Sky Survey (SDSS): http://www.sdss.org/
[3]The Arecibo Legacy Fast ALFA Survey (ALFALFA): http://egg.astro.cornell.edu/index.php/

this problem are the facts that not all objects are visible to all surveys and not all surveys can study the sky with the same angular resolution.

Large groups within the astronomical community have created common databases which congregate their research team's data. Examples of these *Virtual Observatories* include the Canadian[4], US[5] and International web sites[6].

3.3.1 NASA Extragalactic Database (NED)

One such effort in creating a central repository of multiple independent studies is the NASA Extragalactic Database (NED)[7]. The NED database has aggregated information from about 90 thousand journal articles [47] and catalogs[8]. Objects have been uniquely identified across studies and NED has created a standardized cross-reference system to link the various galaxy names. The NED preferred names allows the extraction of spectral data from numerous studies to be obtained through a single query. Other attributes that are available include:

Equatorial Coordinates: This defines the Right Ascension (RA) and Declination (DEC) of the galaxy. It allows for the object to be located in the sky.

Redshift: The redshift of a galaxy is a measure of the magnitude by which a galaxy's velocity affects its spectrum. The redshift of a galaxy is also used as a measure of distance. Negative values represent galaxies moving towards our own, positive values are for galaxies that are moving away. The more rapidly it is moving, the larger the magnitude of the redshift. At high redshift, the effects of velocity on the measured SED become significant [32]. For the purpose of this study, our samples were restricted to galaxies having a positive redshift value which was less than 0.1. This eliminated the requirement of having to compensate for high redshift using the K correction.

Angular Size: The apparent size of the galaxy in degrees. Each wavelength used in a study will measure a different angular size for the galaxy. Different regions of a galaxy generate different amounts of radiation at various wavelengths. The measured dimensions will therefore vary based on the frequency under study. Both the major and minor axis values are given, they will be used later to remove unwanted measurements from the study. In this analysis, we are only interested in whole-galaxy measurements. Only galaxies which contained valid values for the major axis were included. As well, SED measurements that only represented a sub-region of the target were excluded.

[4]http://www3.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/cvo/

[5]http://www.us-vo.org/

[6]http://www.ivoa.net/

[7]The NASA/IPAC Extragalactic Database (NED) is operated by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration

[8]http://ned.ipac.caltech.edu/samples/NEDmdb.html

Distance: This represents the physical distance to a galaxy. The distance is critical in adjusting the measured flux to a common measure for all galaxies. NED contains both redshift-derived distances and in some cases, distances obtained through direct measurement of stellar properties within the host galaxy. Each galaxy in our sample is required to have a distance measurement. This value is required to ensure that we are able to translate all SED measurements to a standard common distance. The distance measure that is used is one that is determined from the Hubble relation and also compensates for the gravitational effects of the Virgo cluster. NED establishes a distance value called the Hubble Flow Distance for most galaxies, only galaxies with such a value have been included.

Morphology Classification: Individual studies attempt to assign a Hubble class to each object; when possible, these classifications have been extracted for comparison with the results of the SOM. Galaxies of unknown morphology were kept for possible morphology prediction from our SOM classification model.

Extinction: Extinction is the reduction in the measured signal due to interstellar material between the source and the sensor. These values were retrieved on a per-galaxy basis. The extinction values provided represent the magnitude of the extinction within our own galaxy. There are currently no methods available to measure internal extinction in distant galaxies. The NED database provides extinction values for the optical portion of the spectrum; though there are methods for correcting for extinction in other frequency bands the corrections are not as significant as for those in the optical regime and are beyond the scope of this thesis.

Synonyms: Each astronomical study assigns unique names or labels to each object. Authors referencing these studies also include these specific names in their work. Unfortunately, from study to study, these names are not consistent. One of the many contributions of NED is to maintain a cross-reference of all of these names and the objects they refer to. This allows the correlation of data from multiple studies for the same object. A version of the NED cross-reference table was created locally for all objects, linking the NED preferred name to other identifier names used in contributing research. This data was used to prevent duplicating data downloads and allowing the same object from entering our dataset multiple times under different names. This would have biased the analysis.

The NED information store was the sole source of data for this thesis. All data items within NED have been compiled from disparate sources and have undergone a consistent set of transformation to preset all data in a consistent manner and a standard set of units.

3.3.2 Galaxies

Each of the surveys contributing to NED has specific goals in mind. The NED database therefore contains objects of a number of different types such as stars, super novae etc. In this work, we are only interested in objects which are galaxies. Based on the preferred name in NED, an object inherits that specific study's classification. If the object is found in multiple studies, it may exist in NED's cross-reference data with multiple classifications. When extracting objects from NED, objects were included as a galaxy if any of their cross-reference classifications matched the galaxy type. This permitted the inclusion of radio sources or infrared objects into the study as long as another study in a different frequency region declared the object as a galaxy.

3.3.3 The Spectral Energy Distribution

NED provides individual measurements of the radiation output of a galaxy at specific wavelengths. Individually, these measurements can be used to identify and compare specific processes occurring with the host galaxy [13]. As a collection of a series of measurements spanning the whole spectrum, these measurements represent a picture of the state of the galaxy.

Studies have been conducted in an attempt to reproduce the spectral energy distribution of galaxies [20, 72]. These studies are based on a number of assumptions about the population sizes of different types of stars. It is believed [72] that variations in population densities amongst the types of stars is a direct measure of the evolutionary stage of the galaxy.

In a static galaxy, where start formation has ceased, stars go through their life cycle and rarely ever affect their surroundings. If a galaxy is static, then once all of the gas clouds that can collapse into stars have done so, what is left is a fixed collection of stars. No new stars should be born after this point. Individual stars would age according to their position on the well-understood HR diagram. In such an environment, a measurement of the age of the oldest stars would provide an age for the galaxy. Measurements of the number of stars at specific positions within the HR diagram would then give us a snapshot of the galaxy and allow us to predict what it will look like in the future. A measurement of the complete spectrum would indicate where the galaxy resides within its life-cycle.

However, observable galaxies are not static and motions within the galaxy ensure that new opportunities for star formation occur over time. As stars age and die, their impact on their surrounding presents opportunities for the creation of new stars. Measurements of the spectral energy distribution of a target galaxy can therefore tell us more than just an inventory count of star types and their numbers, it can also tells us about the dynamics in play within the galaxy. The combination of the information contained in the SED should allow us to get a measure of

the forces driving change within the galaxy and hence, from that, determine its evolution.

The objective of this thesis is to apply the SOM algorithm in an attempt to extract a classification scheme for galaxy evolution. It is expected that the combinations of stellar types contained within each galaxy can lead to a determination of its current position on the evolutionary pathway. The spectral energy distribution from each galaxy should give us the information required for the mapping process. For the purpose of this analysis, we will not introduce any information contained in atomic or molecular line spectra.

3.3.4 The Data

The first data to be downloaded from NED consisted of extracts from some of the major surveys. Targeted were some of the more common-named surveys such a objects with a designation of NGC, UGC and ESO. The next phase tried to ensure that we would have data in a number of different spectral regions. Targeted were ALFALFA in the Radio, IRAS[9] in the Infra-red, 2MASS[10] in the IR and visible as well as GALEX[11] in the ultraviolet.

Additional galaxies were extracted from NED for the *Sloan Digital Sky Survey* (SDSS). Due to the large number of objects in this study, it was not feasible to acquire a complete list of the galaxies it contains, there is a hard limit to the number of objects NED can return from any one query. Follow-up queries then consisted of subdividing the sky into patches of fixed angular size in both right ascension and declination. Each patch then returned as many galaxy type objects as possible. Of note, however, is that most of these surveys register any object in their field of view. For each extract from NED, only candidates identified as galaxies were retained.

For all of the galaxy name acquisition phases, generating duplicate data was an issue. Galaxies often have multiple different names, one for each study. For the galaxies whose SED are shown in the previous chapter, *NGC 1275* is known in NED under 99 different names, *Messier 084* has 66 while *Messier 077* has 80. For each galaxy, a cross-id table was downloaded with all NED synonyms. At each data import, either by the name of the study such as SDSS or by region of the sky, each name was checked against the known synonyms for data already imported.

Values for distance, Redshift, morphology and distance were downloaded for each galaxy. In addition, galactic extinction values for the visible wavebands were also acquired for later correction to the luminosity in each band. In all, there were 680,162 combined galaxies in the bulk downloads. Of these, more than 60% were lost due to insufficient data, see Table 3.2. These galaxies are represented by 2,519,121 unique synonyms in the various studies used to collect their SED data.

[9]NASA/IPAC Infrared Science Archive: http://irsa.ipac.caltech.edu/data/ISSA/

[10]The 2MASS Redshift Survey: https://www.cfa.harvard.edu/ dfabricant/huchra/2mass/

[11]The Galaxy Evolution Explorer: http://www.galex.caltech.edu/index.html

	Number of SED records
Complete dataset:	11,092,432
LINE data:	103,004
SDSS uncertainty:	1,600,150
Angular FOV:	759,350
Quality Issues:	6,551,996
Included in the analysis:	2,077,932
Unique galaxy-freq. pairs:	1,456,948

Table 3.3: SED records excluded from the analysis.

	Number of Galaxies
Complete dataset:	680,162
Redshift out of range:	417,445
Invalid object type:	1,572
Available to study:	261,145

Table 3.2: Contributions to the number of galaxies lost from the analysis.

The next phase of acquiring data consisted of downloading SED data for each candidate galaxy. This was performed one-by-one for all of the unique galaxy names collected in the first phase. In all, 11,092,432 individual SED data points were retrieved from NED. The observational data extracted represents SED measurements in 1,137 different frequencies.

Further processing of the SED data was required. As stated previously, SED data whose angular field of view (FOV) was constrained to just a portion of the complete galaxy, were removed from consideration. Additionally, many SED measurements belonged to atomic and molecular emission spectra. Data from these narrow spectral "lines" are not considered part of the same SED processes under study in this work. Similarly, SED entries associated with comments containing keywords such as: "poor quality", "multiple objects", "nucleus only" or "confusion" were omitted. The final reduction was the weighted averaging of the SED measurements when multiple measurements were available for the same galaxy-frequency pairings. This further reduces the number of available SED measurements for analysis down to 1,456,948. A summary of the records removed from consideration is shown in Table 3.3.

3.3.5 Summary

The NASA Extragalactic Database was used as a single source for all of the data used in this work. The database combines information from thousands of independent research projects into one self-consistent source. NED also provided unit conversions for the SED values, from all of the studies into a standardised set of units. This prevented the introduction of any systematic errors into this work due to an error in unit conversion. The NED cross-IDs were also leveraged to avoid introducing duplicate copies of the same galaxies into the dataset. This would have introduced bias into the dataset as some galaxies would be represented more than once.

3.4 Conclusion

In this chapter we have discussed the implementation of the SOM algorithm. We have also introduced some techniques to improve the performance of the code. This includes the use of the OpenMP libraries to create a parallelized version of the code to run in a shared-memory environment.

The objective of the implemented code was to develop a tool that could be used to study the evolution of galaxies. Through an analysis of common elements of the SED, it is expected that the SOM will be able to help classify galaxies into their appropriate morphological classes. Data was exported from NED and prepared for analysis. In the next chapter, we will review the results of the analysis.

Chapter 4

Results

4.1 Introduction

In the previous chapters, the Kohonen SOM was introduced. The algorithm was examined in depth and approaches which would allow performance improvements were examined. This lead to the implementation of 3D SOM which leverage OpenMP to further enhance processing through the use of parallelization. A candidate dataset, representing the spectral energy distribution of numerous galaxies, was selected for processing. It is the aim of this work to analyze these data using SOMs and produce an effective galaxy morphology classifier.

In the following sections, the implemented version of the 3D SOM will be evaluated using well studied datasets to ensure that it is effective in classifying these standards. Attention will then be placed on the data collected, the processing required to correct for astronomical effects and to bring it into a self-consistent state. From the acquired NED data, candidate datasets with unique attributes will be created and processed through various SOM configurations. The results of these investigations and their effectiveness as classifiers will be compared.

4.2 SOM Code Verification

The SOM code implementation was tested using two distinct datasets before it was used to analyse the galaxy data. The first test data presented to the code was a large set of data points where each datum was made up of three attributes. Each attribute representing an intensity value for the red, green or blue colour channels (RGB). When combined, these three attributes are used to produce display colours for most graphics software. The data was presented to the SOM for classification. The three attributes present in the data matched the three dimensions of the SOM. The resultant map showed that the colour data elements were mapped separating out the data into red, blue and green regions with the overlap giving rise to the other intermediary colours. One of the test mappings is shown in Figure 4.1.

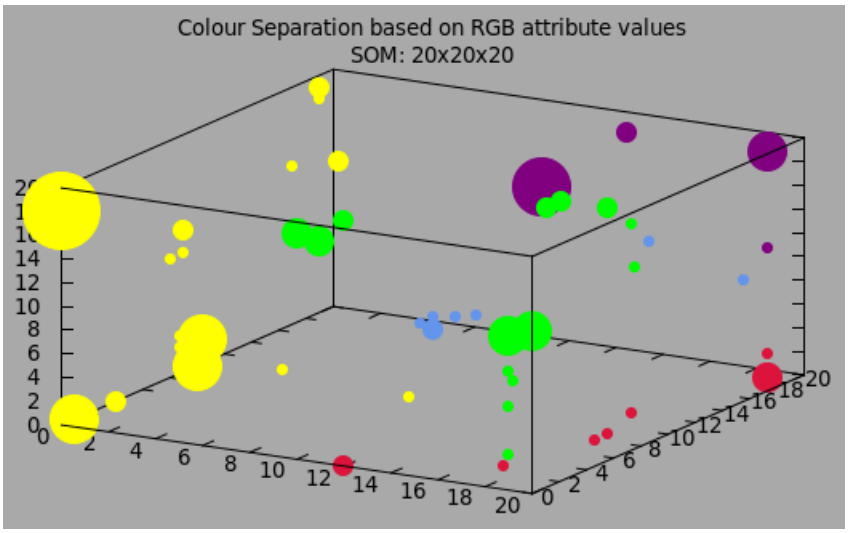

Figure 4.1: SOM separation of the RGB colour dataset.

The second dataset used to analyzed with the SOM is the zoo datasets made available through the Machine Learning Repository (MRL)[1]. The dataset is comprised of 101 records each with 18 attributes. 17 of the attributes are simple boolean values, two are numeric. The data represent a series of records describing the characteristics of a number of living creatures. These attributes include: *do they have hair?, teeth? or a backbone? Are they aquatic? or airborne?, predators? or venomous?*

A manual separation of the dataset was performed by sorting the attributes. Visual examination of the list and breaking it down when attributes differed, allowed for the grouping like creatures together. The raw data was then presented to the SOM code and the resultant map was compared to the manual classifications. The results were identical.

The results of the two test datasets indicated that the algorithm was implemented correctly and was producing consistent results.

A more thorough testing of the application and its features was performed on the Iris dataset from the MRL. The effects of the SOM geometry, number of prototypes as well as normalization techniques were explored in depth. Details of this additional testing and results, see Appendix A, further support the correctness of the implementation of the Kohonen algorithm in this application.

4.3 Data Reduction

A significant amount of data were collected from NED for this research. Constraints were placed on the data to ensure that the analysis was not compromised by known artifacts in the bulk data. The following is a list of factors which were used to preprocess data to ensure that they were appropriate for analysis:

[1]Machine Learning Repository: http://archive.ics.uci.edu/ml/datasets.html

- Field of view
- Redshift
- Extinction
- Distance normalization

Each of these factors was addressed as follows:

A large portion of the SED data resident within NED is comprised of measurements which, by design, only include specific regions of the target. Certain studies only focus on the central bulge of a galaxy or portions of its disk. For this analysis all of the SED entries in the dataset which had an angular field of view which was less than that of the complete galaxy were excluded.

Similar constraints were placed on redshift. In Chapter 2 the K correction was introduced to correct for distortions of the SED caused by redshift. As the redshift becomes larger, relativistic effects become significant and distortions are introduced into the SED. To reduce the requirement for significant K corrections, the candidate galaxies were restricted to those with a redshift less than or equal to 0.1 [32]. An analysis was performed on the data collected from NED. It was found that for small redshifts, the K correction was smaller than the existing uncertainty in the data. For this analysis, the SED data collected was not K corrected.

Before flux values can be adjusted for distance, they must be corrected for galactic extinction. Previously we have stated that the extinction is a measure of the reduction in intensity of a signal due to dust and gas between the source and the observer. Within NED, extinction values are available for specific bands within the visible portion of the spectrum. Care was taken to reduce the extent of the impact of extinction by choosing galaxies which did not lie directly in the galactic plane. A plot of the galaxies present in the database is shown in Figure 4.2.

The NED database contains extinction values for the U, B, V, R, I, J, H and K filters. These values were used to correct the respective SED measurements when applicable. Equation 4.1 relates an extinction value for the V filter A_V in terms of the flux density measured f_V and the flux density one would have measured without extinction: f_{V_0}.

$$A_V = -2.5 \log \left(\frac{f_V}{f_{V_0}} \right) \tag{4.1}$$

Which, solving for f_{V_0}, simplifies to:

$$f_{V_0} = \frac{f_V}{10^{(A_V/-2.5)}} \tag{4.2}$$

Each SED measurement in the visible portion of the spectrum, for which we had a corresponding extinction value, was corrected before further processing.

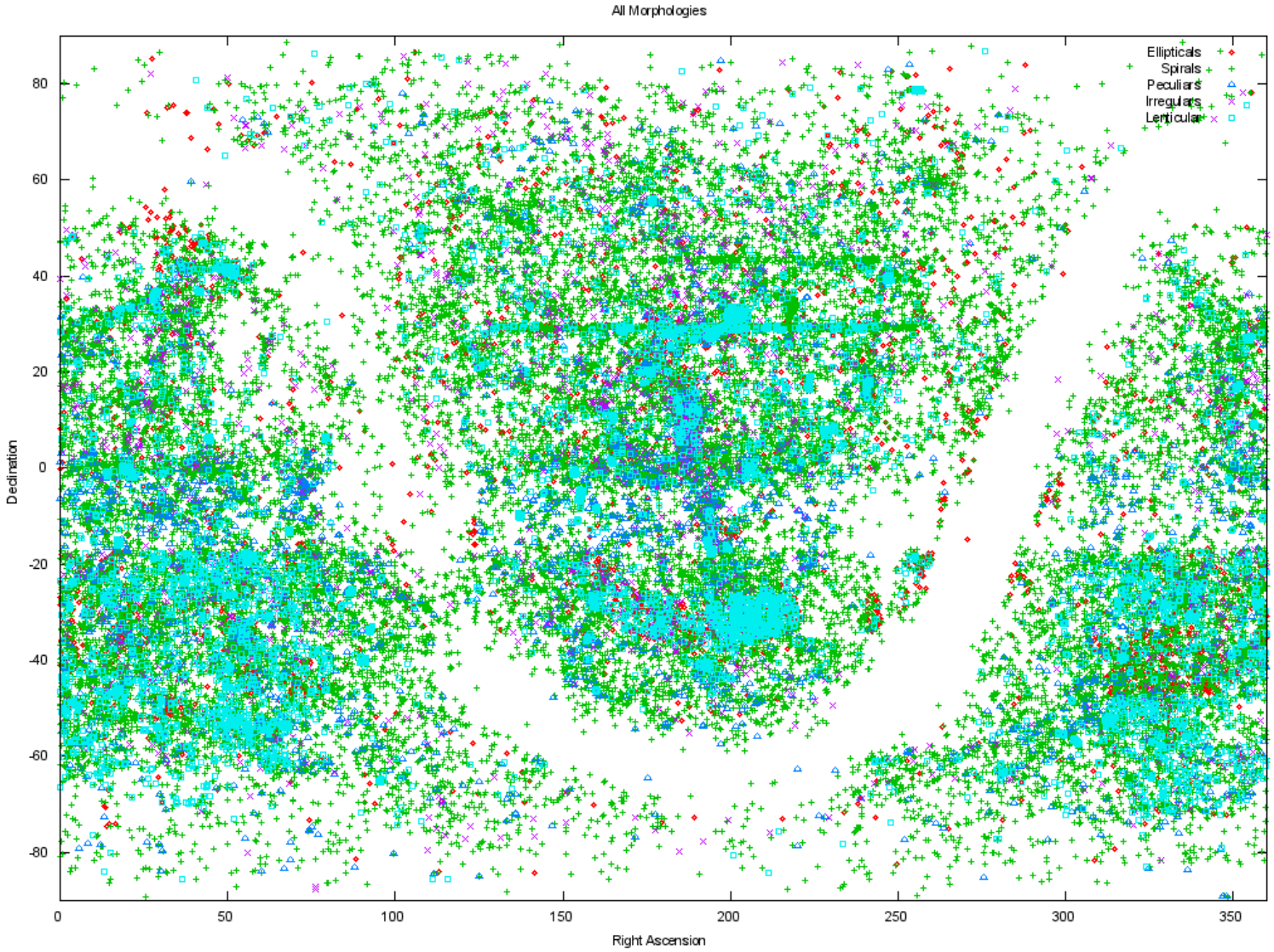

Figure 4.2: Galaxies present in the dataset plotted by right ascension versus declination. The plot shows the "Zone of avoidance" for measurements made through the plane of our own galaxy.

A proper comparison of the luminosity of galaxies requires that we account for their relative distances. Without a proper distance correction, it is impossible to distinguish between a bright galaxy that is far away and a faint galaxy that is nearby if their detected flux are identical. The data collected by NED is not by default corrected for distance. To properly correct the data before they are included in the mapping process, all SED data must be corrected for distance. We must therefore only include galaxies and SED data for objects with known distance measures. Equation 4.3, was used to convert the fluxes present in NED to the apparent luminosity. The conversion is dependent on the galaxy's distance D_g, the galaxy's measured luminosity from NED L_0 and on setting standard distance for comparisons. In this analysis we used a standard distance of 10Mpc. This conversion permitted the direct comparison of luminosity values across the SED.

$$L_{std} = L_0 \times \frac{4\pi D_g^2}{4\pi(10Mpc)^2} = L_0 \times \left(\frac{D_g}{10Mpc}\right)^2 \tag{4.3}$$

The final processing of the SED data involved summarizing the existing data to provide a single SED value for every galaxy-frequency pair in our dataset. The SOM algorithm requires that each attribute of each object be unique. For measurements from multiple studies which provided NED with their results, a measurement error based weighted average was used to generate a single value.

The process of filtering out unwanted data significantly reduced both the number of galaxies and the number of SED data points available for study. Reductions due to redshift and distance considerations reduced the total number of galaxies from 680,162 to 261,145. Applying our constraints and uniqueness requirements to the SED data reduced our spectral data from 11,092,432 to 1,456,948 records.

4.3.1 Additional Quick-Validation Data

The only attributes used by the SOM to classify galaxies are the SED measurements at different frequencies. The SOM technique does not rely on any external sources to moderate the creation of the map or its final clustering. To facilitate a quick visualization of the effectiveness of the created SOM, an additional galaxy property was maintained in the dataset. This property was the NED classification of the morphology of the object.

Extensions to the Hubble diagram provide a large number of morphologies and sub-morphologies for galaxy classification. For the purpose of an initial view of the data such granularity was not required. Because this property of the galaxy was not going to be involved in the mapping process itself, the sub-morphologies were re-mapped to the broadest of classes: Spiral, Peculiar, Irregular, Elliptical and Lenticular.

The objective of carrying this extra piece of information was to make interpreting the maps as simple as possible. With the broad morphologies in place, it was a quick exercise to witness the locations of the various galaxies within the map. This view would reveal the effectiveness of the SOM process at extracting sufficient information from the SED data to map like morphologies close together.

4.4 Data Pre-processing

The objective of the SOM is to group objects together based on their similarities across a number of attributes. However, the relationship between the data was one-to-many. For any particular galaxy, there exists a series of frequencies for which we have measurements. What is missing from this dataset is a method of generating a list of galaxies which all shared a set list of frequency measurements. To this aim, an additional step was included in the data pre-processing.

An analysis of the SED data was performed and a list of unique frequencies was extracted. This list contained 1137 different frequency values which spanned all of the observations. If the list of attributes for each prototype in the SOM were to be made up of 1137 values, it would cause two issues. The first issue would be the fact that for most galaxies, most of their attributes would consist of missing data. Though there exist techniques to deal with missing values, most are implemented to substitute surrogate values for only a small subset of the input data and typically only for any one attribute per datum. The replacement value is based on some expert based best-estimate or on some average based on similar data items. The use of all 1137 frequencies would present such a sparse dataset that values for missing data would be more numerous that the actual number of measurements. In the data collected from NED, the galaxy with the largest number of known attribute values contained 195 unique frequency measurements.

The second difficulty with using all of the frequencies is the amount of memory required to store the information. Through using all 1137 attributes, we require that every prototype and every data element presented to the SOM consumes almost six times more memory than required by the single object which exhibits the maximum number of observations. Also, including all of the frequencies would imply that all of the galaxies in the dataset would be eligible for inclusion in the analysis. Combined, the impact on memory consumption would far outstrip the available cache memory on even the largest processors. This would lead to a significant degradation in the performance of the SOM routine.

To solve the above issue, an analysis of the most common frequency pairings was undertaken. Each galaxy in the dataset was taken and a list of the frequencies for which we have observations was made. To make comparisons easier, a boolean vector was created for each galaxy. The vector contained a dimension for each of the 1137 frequencies. If a galaxy was observed to have data for a specific frequency, the bit value at that position was marked with a '1'. Otherwise all bits were set to '0'. The result of the exercise was a set of vectors indicating where observations were available.

Each galaxy's frequency vector was then compared to all of the others maintaining a count which indicated how many galaxies were represented by the exact same bit vector. The resultant group of bit vectors represented a 'family' of galaxies with an identical list of observed frequencies.

Bandpass	Frequency ν (Hz)	Number of families
Radio	$3.0e^{10} > \nu$	2,477
Millimeter	$3.0e^{11} > \nu > 3.0e^{10}$	250
Sub-Millimeter	$1.0e^{12} > \nu > 3.0e^{11}$	11
Far Infrared	$7.5e^{12} > \nu > 1.0e^{12}$	223
Mid-Infrared	$6.0e^{13} > \nu > 7.5e^{12}$	404
Near Infrared	$3.0e^{14} > \nu > 6.0e^{13}$	378
Visual	$9.0e^{14} > \nu > 3.0e^{14}$	22,307
Ultraviolet	$2.0e^{16} > \nu > 9.0e^{14}$	194
X-rays	$2.0e^{19} > \nu > 2.0e^{16}$	561
Gamma Rays	$\nu > 2.0e^{19}$	9
Multi-Bandpass	*all*	17,227

Table 4.1: Number of bit vector families per spectral region

Once a popularity list was created for the bit vectors, they were then compared amongst themselves. For each of the observed vectors a comparison was made of the frequencies present with those of all of the other vectors. This was done to generate frequency bit-vectors present within the data which were not affiliated to any specific galaxy. This permitted the creation of galaxy families which, though they represented fewer frequencies, contained a larger number of candidate galaxies.

The intent of creating the above datasets was to explore the possibility that a SOM would be capable of classifying galaxies in such a way that we could interpret the results in terms of morphologies. The source studies which created the data in NED typically only examine a specific portion of the spectrum. An additional avenue of investigation would be to examine if any one region of the spectrum would contain sufficient information to lead to SOM morphological classifications on their own. To explore this, the above frequency bit vectors were reprocessed to create new patterns. These new patterns would only have '1' for the region of the spectrum they represent and '0' everywhere else. This then created additional bit vector families for Radio, Millimeter (mm), sub-millimeter (sub-mm), Far Infrared (FIR), Mid-Infrared (MIR), Near infrared (NIR), Visual, Ultraviolet (UV), X-ray and Gamma-Ray regions. [33, 47]

The result of this process was a list of families of frequencies present in the data. This permitted the creation of numerous dataset for SOM processing. By using the bit vector approach, the datasets were guaranteed to not contain any missing data.

Table 4.1 shows the number of bit vector families that were found in each of the spectral regions. It is interesting to point out that the visual range generated a larger number of bit vectors than the overall multi-region approach. This artefact is caused by the sub-vector matching. By masking out the other regions, more variations of bit patterns were generated in the visual region.

Bandpass	Frequency ν (Hz)	Number of candidate datasets
Radio	$3.0e^{10} > \nu$	—
Millimeter	$3.0e^{11} > \nu > 3.0e^{10}$	—
Sub-Millimeter	$1.0e^{12} > \nu > 3.0e^{11}$	—
Far Infrared	$7.5e^{12} > \nu > 1.0e^{12}$	6
Mid-Infrared	$6.0e^{13} > \nu > 7.5e^{12}$	11
Near Infrared	$3.0e^{14} > \nu > 6.0e^{13}$	22
Visual	$9.0e^{14} > \nu > 3.0e^{14}$	26
Ultraviolet	$2.0e^{16} > \nu > 9.0e^{14}$	—
X-rays	$2.0e^{19} > \nu > 2.0e^{16}$	3
Gamma Rays	$\nu > 2.0e^{19}$	—
Multi-Bandpass	*all*	428

Table 4.2: Number of candidate datasets per spectral region

4.5 Generating the Dataset

Once all of the bit vector families had been identified, the process of creating the input files for SOM processing could begin. The large volume of families made processing each impractical. A scoring algorithm was developed to determine the most suitable candidates. The objective of the scoring was to provide the largest number of frequencies and the largest number of galaxies to the SOM. It was found that these two metrics were often diametrical opposites. Families with very few frequencies tended to have a large number of galaxies. Conversely, families with many frequencies tended to have a minimal contingents of galaxies. Some base requirements helped in the selection process. Families represented by fewer frequencies, hence attributes, than the dimensionality of the SOM were excluded. Additionally, families which contained fewer than several hundred galaxies were also omitted.

Table 4.2 shows the results of the scoring process. The final tally of acceptable datasets was 496. Unfortunately, a number of frequency regions failed to produce sufficient candidates and were therefore dropped from further processing. Note also that, though the visual region generated a large volume of candidate families in Table 4.1, only 26 of these produced candidates acceptable for processing.

With the number of families determined, each was converted into an input dataset for the SOM.

The SOM process is sensitive to the data presented to it. As we have stated previously, even the order with which the data are presented to the SOM can affect the outcome. If the procedure used to present the data inadvertently favours some data over the rest, bias can be imprinted on the map. It is therefore important to

Morphology	Number of Galaxies
Spiral	38,706
Peculiar	2,533
Irregular	1,755
Elliptical	7,793
Lenticular	7,985

Table 4.3: Galaxy sample size by morphology.

ensure that the data we use to build the maps reflect as best we can, a uniform distribution of the various known morphologies.

As shown in Table 4.3, the complete set of galaxies present in this study is heavily weighted towards the spiral morphology. In many of the candidate datasets, this ratio is even more severe. To provide for a more even representation, the populations of the different morphologies were balanced in the final datasets. Each dataset was inventoried for the different populations it contained. To arrive at a sample size for each morphology the high and low population counts were excluded and an average of the remaining three class populations was calculated. This average was then used as the required representation from each class. To build the dataset used for study, some morphologies were under-sampled, selecting at random candidates from the full population. Other less abundant classes were over-sampled by randomly selecting candidates (with replacement).

The objective of this investigation is to evaluate the effectiveness of the SOM algorithm at classifying galaxies by morphology. Though forcing equal morphological representation in the dataset would seem in contradiction with the objective of this research, it is in fact imperative for the production of effective mappings. The morphologies themselves are not used to create the map. All the normalization process provides is a mechanism to ensure one population does not monopolize the complete map, overshadowing all others.

4.5.1 Exploring the Mapping Process

In Chapter 2 Wendel & Buttenfield [73] suggested that a an appropriate number of neurons for a SOM can be expressed as shown in Equation 2.1. In this equation N represents the number of data elements while a is the number of attributes.

In our datasets we have an average of 1100 galaxies. The number of attributes ranges from four to twelve. For our purposes this yields a range from 66 to 114 cells. This figure is, however, based on an equation originally derived for a two dimensional SOM.

A number of tests were performed where both the dimensions of the SOM were varied as well as the method used to initialize the map. SOMs were created in the following geometries: $30 \times 30 \times 30$, $3 \times 5 \times 7$, $3 \times 9 \times 27$, $9 \times 9 \times 9$ and $5 \times 7 \times 9$. Initialization techniques for the SOM included:

NONE: The raw, unscaled data is used for the map.

PNORM: Per-attribute scaled data. Prototypes in the map are initialized by generating per-attribute random values. These values are scaled to be within the range of each attribute as seen in the training dataset.

SNORM: SOM based normalization. Prototype attribute values are generated as per the PNORM technique. These values are then scaled based on the overall range of all attributes present in the training dataset.

GNORM: Similar to the PNORM technique. In an attempt to reduce edge effects, a buffer area consisting of ten percent of the prototypes on each face of the 3D SOM are forced to have zero for all attributes.

ADATA: The training dataset was examined. Similarity measures were evaluated for all of the possible pairings within the data. The eight most dissimilar data points were selected. These data items were then placed at the vertices of the 3D SOM grid, keeping the ten percent buffer from the edges as in the GNORM case. As the data points were placed within the grid, the surrounding prototypes had their weights adjusted using the same rules as those for a typical BMU region.

As a result of testing various geometries, a size of $5 \times 7 \times 9$ was chosen for all future maps. Also, our selection of normalization techniques was reduced to: ADATA, NONE and PNORM.

4.6 Results

Of all of the frequency families created, 496 were chosen for preliminary analysis. The selection was based on the number of galaxies present as well as trying to maximize the number of spectral regions represented. From this first round of investigation, the list was reduced to twelve candidate datasets.

The selected datasets were studied using maps which were $5 \times 7 \times 9$ in size. The maps were initialized using the ADATA, NONE and PNORM techniques. In addition, the BMU selection was moderated using the CUBE, CUTOFF or FULL approachA. All maps were constructed over a maximum of 500,000 iterations.

The results of all analyses are presented in Appendix C. Each set of results begins with a description of the data. This includes, broken out by morphology, the training set size. Also shown are the number of unique galaxies present in the training data as well as the full dataset. In the case of each dataset extracted from NED, there existed galaxies which matched the frequency profile but did not have a NED defined morphology. A count of these galaxies is also presented. These *unknown* galaxies will be discussed in the next section.

The second portion of the dataset description is a list of the frequencies present in the dataset. These are presented along with the frequency band they belong to.

The final data shown is a breakdown of the results of processing the data through the various SOM configurations. For each experiment, a list of percentages is shown which represents how well the SOM performed at classifying the input data. Since all of the training data were of known morphology, prototypes within the SOM were allocated to morphologies based on the predominant class present within the SOM Cell.

The data presented is broken down into 3 different lines. The first line labeled *Training* shows how well the training data were classified when it was presented back to the SOM after the completion of the map build.

The second line contains information on the mapping accuracy of the SOM for data that was not present in the training set. This line only contains statistics for new data which mapped to prototypes in the SOM which were previously classified by the training data. This second line has been labeled as *New Data*.

The final line presents, for each dataset, contains classification results for data in the complete dataset which mapped to cells not classified by the training data. Of the 315 cells in the SOM, not all were the recipients of mapped data during the training phase. These prototypes were, however, subject to the training process. When new data were presented to the SOM, often it was mapped to unclassified prototypes. Similar to the classification during training, these new prototypes were associated with a classification based on the majority class of the data mapped. The third line, labeled *No Map*, shows how well the SOM performed in reproducing the morphology of the new data. Next to the percentage, the number of data elements is shown in brackets.

Each dataset in the appendix also contains additional information on each SOM. These items include measures of the quality of the map such as the quantization error and the termination error. Other data includes information on the build process. These include the CPU time used to build the map, the number of iterations required as well as the number of BMU changes still taking place had the map not converged by the $500,000^{th}$ iteration. Finally, an examination of the training set's map utilization is presented. This includes how many prototypes represented a single morphology, how many were associated with mixed-morphologies and a measure of the percentage of the map used to represent the training data.

Interpretation of the results obtained leads to a number of interesting observations of the data. The following sections will discuss some of the major factors which influenced the outcome of the mapping process.

4.6.1 Class Imbalance

The number of candidate galaxies available within each dataset is constrained by the number of observed frequencies. The larger the number of frequencies, the lower the availability galaxies. Similarly fewer frequencies lead to a larger population to study. One factor remains fairly consistent across the sample set, however, the count imbalance in the morphologies.

Further complications arise when inspecting the classes present in the original data. In most cases, spiral galaxies account for over 80% of the data. The other two predominant morphologies as elliptical and lenticular types. Efforts were made to ensure that an appropriate representation was provided to the SOM during training. an examination of the various datasets will show that in some cases, very few candidate galaxies existed for certain morphologies. To provide sufficient data for analysis, these under-represented classes might have their galaxies included an order of magnitude more often than any galaxy of a different class. As an example, Pat_952_1386_5 only contains 11 distinct peculiar galaxies and 2 irregulars. To create the training dataset, these were duplicated numerous times each to provide for the 151 samples required. Similarly Pat_1090_1145_7 required multiple duplicates of its irregular and elliptical candidates. The result of this over sampling of a few unique galaxies may have biased the SOM training towards these few examples. The resultant map would therefore be biased to just a few representatives for specific classes. Larger datasets with a more even representation did not encounter the same difficulties.

4.6.2 Map Coverage

Three other measures recorded for each analysis also display a relationship with the dataset size. These are the number of prototypes which represent unique morphologies, those that represent multi-morphology cells and the percentage of the SOM's volume mapped by the training data.

For very small sample sizes, the SOM provides ample space for associating input data to prototypes. In such cases, competition for prototypes is low. This leads to both an artificially high success rate in mapping morphologies as well as a low demand for prototypes. The low requirement for prototypes is reflected in a low map utilization of the training data. The side effect of the reduced competition for prototypes is that there are very few cells, if any which contain a mix of morphologies.

In the results, as the number of candidate galaxies increases, so does the number of cell prototypes mapped by the training data. The more competition for the prototypes, the larger the number of mixed-morphology cells. This leads directly to a better utilization of the SOM map space. In the perfect case, the objective would be to have only single-morphology prototypes in the SOM. In a real case, it should not be unexpected that candidates which lie between valid morphologies might

congregate in '*border*' cells between regions of unique morphologies. The difficulty with such border regions is that it is difficult to assign a specific morphology to the data elements they represent. Typically a majority rule approach is used. When measuring the effectiveness of the SOM, the greater the number of border cells of mixed-morphology, the lower the expected morphology prediction accuracy.

Examination of the results does show that for the simple, small training sets, the reproducibility of the SOM mapping is much higher than for large datasets. As the mixed morphology prototypes become more numerous, the prediction accuracy diminishes.

4.6.3 Error Measures

As a verification step, both the quantization and topographic errors were determined for each iteration of the maps produced. Overall, their appearance is similar to those obtained while investigating the Iris datasetA.

Reviewing the values (Appendix C) for QE, which represents how well the prototypes represent their associated data points, did not yield any consistent preference for any specific analysis technique. Similarly, TE values did not reveal one technique to be more effective than any other.

The TE (dist) variation of the topographic error did provide the opportunity, in a couple cases, to differentiation map quality. Results for Pat_1052 and Pat_1030 show instances where identical QE values generated TE(dist) values that were significantly different. The default QE measure only provides for a 1 or 0 score if a data item's BMU are not adjacent. By measuring the map grid distance between the first and second BMU, a more sensitive measure of the map's ordering can be made. Smaller distances would indicate maps which, for the same datasets, are better aligned to the data.

It was found that, for maps with a larger percentage of coverage, values for both QE and TE produced less variance from technique to technique. These maps were also found to be those belonging to datasets with the larger number of candidate galaxies.

4.6.4 Related Datasets

Of the twelve datasets examined, three were found to be subsets of the dataset Pat_1052_1126_12. These four datasets have been placed at the end of Appendix C. As a quick summary Table 4.4 shows the frequency overlaps in the different datasets.

The extent to which they cover the SOM as well as the mix of single and multi-morphology cells mirror what is seen in the other datasets. Based on their number of attributes and dataset set sizes, their individual effectiveness at classifying the training data is similar as well. A dendrogram of the Pat_1052 dataset, processed

Band	Frequency	Pat_1052	Pat_139	Pat_192	Pat_Visual_49
Radio	1400000000	✓	✓	✓	
FIR	3000000000000	✓	✓		
	5000000000000	✓	✓		
NIR	138000000000000	✓	✓		
	182000000000000	✓	✓		
	240000000000000	✓	✓		
Visual	325000000000000	✓		✓	✓
	389000000000000	✓			✓
	477000000000000	✓		✓	✓
	617000000000000	✓		✓	
	681000000000000	✓			✓
	836000000000000	✓		✓	✓
Known morphology		934	6,190	1,937	5,089
Unknown Morphology		290	2,512	1,944	3,593
Total Number of Galaxies		1,224	8,702	3,881	8,682

Table 4.4: Frequency membership of the Pat_1052 family

without adjusting the initialization of the SOM (NONE) and no restrictions to the BMU movement (FULL) is shown in Figure 4.3.

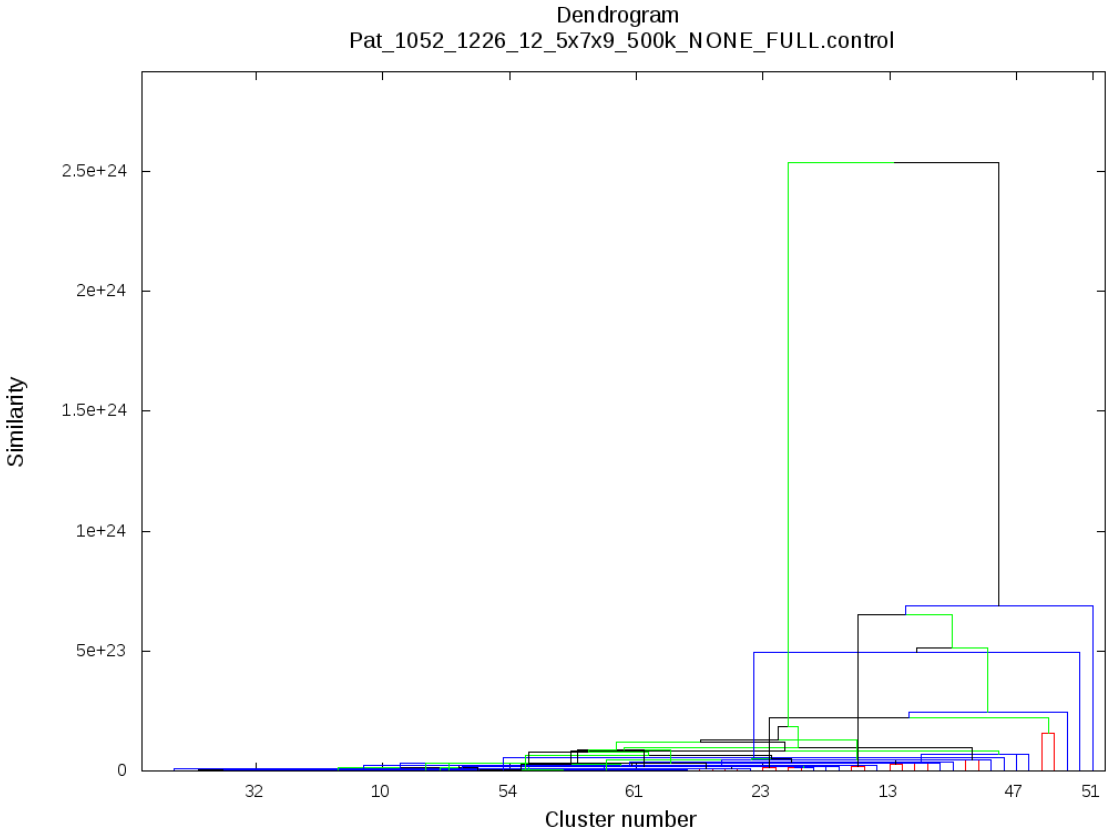

Figure 4.3: A dendrogram showing the clustering of the Pat_1052 dataset.

The similarities are such that it is not clear if the improvement in the classification seen in Pat_1052_1226_12 is due to the resultant sample size or the increase in information contained in the additional attributes.

4.6.5 Predicting Morphology

Each of the datasets contained a number of galaxies for which NED had not determined a morphological classification. These galaxies were included in this study as they contained the appropriate frequencies to be included in each analysis. This provided for a larger sample size without impacting on the SOM build itself. These 'unknowns' were not part of the training data and could therefore be used to measure the effectiveness of the SOMs at predicting a galaxy's classification.

The list of unknown galaxies were evaluated by their respective SOMs and given a classification. Each of these classifications was taken as a vote. The list of all predicted morphologies for all of the unknown galaxies was compiled. The SOMs were then used as an ensemble method [62] to predict an overall morphology for

each object. Table 4.5 presents a summary of the morphology populations present in the datasets analysed. It is important to note that many galaxies are counted numerous times to make up these counts. As an example, recalling the Pat_1052 datasets, since Pat_1052 is a subset of Pat_139, at_192 and Pat_Visual_49, all of its galaxies are accounted for at least 4 times in the Table 4.5. A count of the unique galaxies included in the datasets is also presented.

The galaxy dataset was collected at the beginning of this research project. The NED database is constantly being updated with new classifications. The list of unknown galaxies was presented back to NED to see if any morphologies had been updated since the original data snapshot. It was found that a large number of the previously unknown galaxies had now been given official NED morphologies. We were then able to compare our predicted classes with those present in NED.

A summary of the predicted classes can be found in Appendix D where results are given for galaxies with the largest number of votes per morphology.

Ensemble results for the peculiar, irregular and lenticular galaxies did not provide any consistency across the multiple SOMs. This resulted in a very poor performance in predicting galaxy morphology. The spiral and elliptical predictions provided results which were much more encouraging. With only a few exceptions, both spiral and elliptical galaxies were predictable when the vote count stayed above the 55% level.

4.7 Discussion

A number of difficulties were encountered within this project. The most interesting question arising from the various tests was trying to determine which factors influenced the convergence of the SOM the most. In the cases of the RGB or Iris data, it was well established that the attributes in the data lead directly to a valid classification scheme. The colour example was rather simplistic but did show that the implementation was able to discover the ordering present in such cases. In the case of the Iris data, historically, physical dimensions have been used to classify species. Those data attributes were known to be the drivers of the classification. It would therefore be expected that the SOM algorithm would produce a map which could effectively classify the data. In the galaxy data, however, there were no guarantees that any of the datasets contained sufficient embedded information to lead to a valid classification scheme.

The main question which arose was whether the quality of the map was impacted by the lack of information contained in the data or if it was an artefact of the parameters and constraints placed on the map itself. If the dataset has no intrinsic morphological knowledge embedded in it, an effective reliable map cannot be produced. If, however, the selection of the map size is incorrect or if the topological order is not maintained during the building of the map, the process will not lead to an effective classifier. Similarly, if the normalization of the data

Dataset Name	S	P	I	E	L	NED Unknowns	NED Knowns
Pat_130_10042_5	6,999	630	134	244	427	1,575	8,434
Pat_952_1386_5	202	11	2	240	485	318	940
Pat_1030_1261_7	1,063	136	11	6	21	15	1,237
Pat_1046_1235_7	1,017	162	17	9	13	13	1,218
Pat_1090_1145_7	989	113	7	4	20	8	1,133
Pat_1100_1121_6	846	110	22	59	74	10	1,111
Pat_1126_1069_10	1,028	7	9	1	6	10	1,051
Pat_Visual_42_11641	1,267	112	67	319	287	6,580	2,052
Pat_1052_1226_12	790	62	16	17	49	290	934
Pat_139_8938_6	5,076	530	123	183	278	2,512	6,190
Pat_192_6934_5	1,489	122	35	150	141	1,944	1,937
Pat_Visual_49_8831	3,741	223	116	487	522	3,593	5,089
Totals:	24,507	2,218	559	1,719	2,323	16,868	31,326
Unique:	13,343	1,045	347	1,272	1,621	13,621	17,628

Table 4.5: Morphology representation in each dataset

before it is introduced to the map harms the embedded knowledge, the map will exhibit the effects of these changes. The process of creating an effective SOM then becomes an iterative search for the proper geometry and initialization techniques. Since each dataset contains unique data, each dataset in turn would have to be explored individually.

In the data extracted from NED, 1137 different frequencies were identified. Data was extracted for 680,162 galaxies and used to produce all of the sample dataset for the SOM discovery process. Though the data volume seems very large, the sparsity of the dataset significantly reduces the number of candidate datasets. Finding datasets which contained multiple frequencies as well as more than a few hundred galaxies in each of the morphologies proved quite difficult. An examination of the datasets in Appendix C shows that this was simply impossible in most cases. Large datasets with more than 5 frequencies simply did not exist.

The results obtained from the different analyses do suggest that using SOMs to classify galaxies by morphology is possible. The individual maps did not provide as clear a distinction between morphologies as expected. The quality of the results maybe a product of the geometry of the SOM selected, the lack of available data or a combination of both. Collectively though, the maps produced did exhibit some consistency in classification. For both the Spiral and Elliptical classes, it was possible to correctly predict the morphology of a number of galaxies.

Chapter 5

Conclusion and Future Work

5.1 Conclusions

We have shown that extending Self-Organizing maps to 3D is an effective technique in discovering patterns and order within a set of high dimensionality data. The RGB colour test data as well as the MRL's Zoo dataset were successfully explored using the SOM code implemented for this research.

Further analysis of the code implementation was performed on the Iris dataset. Results from these tests showed the effectiveness level of various map dimensions and normalization techniques. This information was then used to guide our decisions regarding the use of SOMs for analysing SED data for galaxies in an attempt to create an automated morphology classifier.

Of the 496 viable datasets generated from the NED data, only a dozen were chosen for further processing. This reduction in the number of candidates can be attributed to poor SOM performance in the preliminary round of analysis. The performance of the mapping process is impacted by the information contained in the attributes of the data as well as our choice of the geometry of the SOM. The exhaustive search for the perfect map geometry for each individual candidate dataset was beyond the processing capacity of the current thesis. A standard configuration was chosen to help provide a basis for comparison between datasets. Though the optimal solution for each dataset may be different due the number of attributes and data elements present, the SOMs produced did show promise in separating morphologies. For the maps that didn't produce any distinct separation between classes, it may be that the frequencies chosen for those datasets do not provide sufficient information to distinguish Spirals from Ellipticals or any other morphology.

Of the dozen datasets chosen, results from individual SOM showed promise in identifying morphologies. By adopting an ensemble method approach it was possible to provide morphology predictions for a number of galaxies. The more effective predictions were found to be for spiral and elliptical classes.

5.1.1 Future work

The objective of creating an automated classifier for galaxy morphology has been shown to be possible. To produce more efficient SOMs will require additional investigations.

The primary constraint in this work has been the limited size of the datasets. Though the results for the smaller sets yield impressive reproducibility, it is not evident that this is not just due to the small sample size yielding high percentages. Without larger samples it is difficult to not think of this as a statistical effect of a limited number of data points. The implication of the small sample sizes for certain datasets may also imply that the frequencies used in the analysis are not those that are of general interest to a large fraction of the studies producing SED data. If this is the case, then classifiers based on these frequencies may be quite good at determining galaxy morphology but they would be ineffective for the vast majority of galaxies present in NED. Conversely, if the SOM proved very efficient, it may guide future SED measurements for existing galaxies to include these specific frequencies.

Another approach to the frequency issue may be to generate a reduced number of data points per frequency band. Though this would reduce the overall number of frequencies present, it has the potential of increasing the number of candidate galaxies which share this new common attribute. This approach would require developing a technique to merge multiple values into a single representative value. This averaging, or bining of the data would have to take into account the number of data points, if an average was better than the integral over the region etc. This process is beyond the scope of the present research.

The two other factors which were found to be the most significant were the map size as well as the initialization method. Improvements in the initialization which would be of interest for future study would include using PCA to guide prototype weighting [37, 64]. This would involve finding the three most significant components embedded within the data and initializing the map accordingly. The most significant component along the longest axis, the second along the mid-length axis and the least significant of the three, along the shorter dimension of the SOM. This may lead to a better ordering of the mapped data. It should be noted though that it is possible that some of the attributes are not linear in nature and techniques such as normal PCA may not be appropriate [24, 37].

The size of the map is difficult to determine other than trial and error. Due to the edge effects, the smallest dimension should be kept above 3 units and from our testing at or above 5. Kohonen suggests asymmetrical maps and one avenue would be to scale the dimensions based on the impact of the three most significant components determined by PCA. Even with these additional guidelines, it is expected that an iterative process will be required to find the optimal map size for each individual dataset. From the current work, it is apparent that a one-size fits all approach may not be optimal.

For morphological identification, mixed results were obtained. The SOM algorithm is an unsupervised learner, it is possible that the inclusion of Peculiar and Irregular galaxies may have hindered the processing efficiency. In this work, attempts were made to consider these morphologies as significant as those present on the original Tuning Fork Figure 2.6. As such, the selection of datasets to process as well as attributes focused on equal representations of the 5 classes: S, P, I, E and L. It is possible that further work would benefit from focusing on the three major classes and allow the SOM to allocate prototypes for the P and I classes on its own. This would benefit the analysis by allowing for larger datasets as well as reducing the requirement for aggressive over and under-sampling of the data.

In closing, it is expected that further work will provide direction on the appropriate frequencies which could be used in an automated process to help assign galaxy morphologies to unknown targets. The results here do indicate that identifying Spirals and Ellipticals is possible. Future work may allow the tuning of the SOM to help resolve sub-morphologies such as E0 from E3 and Sb spirals from SBc Spirals.

Bibliography

[1] Jorge Sánchez Almeida, J. Alfonso L. Aguerri, Casiana Muñoz-Tuñón, and Angel De Vicente. Automatic unsupervised classification of all sloan digital sky survey data release 7 galaxy spectra. *The Astrophysical Journal*, 714(1):487, April 2010.

[2] Nicholas M. Ball and Robert J. Brunner. Data mining and machine learning in astronomy. *International Journal of Modern Physics D*, 19:1049–1106, 2010.

[3] Hans-Ulrich Bauer and Klaus R. Pawelzik. Quantifying the neighborhood preservation of self-organizing feature maps. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 3(4):570–579, 1992.

[4] Derek Beaton, Iren Valova, and Daniel MacLean. Cqoco: A measure for comparative quality of coverage and organization for self-organizing maps. *Neurocomputing*, 73:2147–2159, June 2010.

[5] Michael R. Blanton and Sam Roweis. K-corrections and filter transformations in the ultraviolet, optical, and near-infrared. *The Astronomical Journal*, 133:734–754, February 2007.

[6] Shyam Boriah, Varun Chandola, and Vipin Kumar. Similarity measures for categorical data: A comparative evaluation. In *In Proceedings of the eighth SIAM International Conference on Data Mining*, pages 243–254, 2008.

[7] Kirk D. Borne. A machine learning classification broker for the lsst transient database. *Astronomische Nachrichten*, 329(3):255–258, 2008.

[8] Kirk D. Borne. Scientific data mining in astronomy. *Next Generation of Data Mining*, pages 91–114, November 2009.

[9] Mathieu Bringer and Michel Boër. An automatic astronomical classifier based on topological neural networks. In *Astronomical Data Analysis Software and Systems IX, ASP Conference Proceedings*, volume 216, page 640. Crabtree. Astronomical Society of the Pacific, 2000.

[10] Ronald J. Buta. Galaxy morphology. In Terry D. Oswalt and William C. Keel, editors, *Planets, Stars and Stellar Systems*, pages 1–89. Springer Netherlands, 2013.

[11] Igor V. Chilingarian, Anne-Laure Melchior, and Ivan Zolotukhin. Analytical approximations of k-corrections in optical and near-infrared bands. *Monthly Notices of the Royal Astronomical Society*, 405(3):1409–1420, 2010.

[12] Igor V. Chilingarian and Ivan Zolotukhin. A universal ultraviolet-optical colour-colour-magnitude relation of galaxies. *Monthly Notices of the Royal Astronomical Society*, 419:1727–1739, 2011.

[13] Christopher J. Conselice. The fundamental properties of galaxies and a new galaxy classification system. *Monthly Notices of the Royal Astronomical Society*, 373:1389–1408, 2006.

[14] Marie Cottrell and Patrick Letréemy. Missing values: processing with the kohonen algorithm. In *Proceedings of Applied Stochastic Models and Data Analysis, Brest, France*, May 2005.

[15] Pierre Demartines and François Blayo. Kohonen self-organizing maps: Is the normalization necessary? *Complex Systems*, 6(2):105–123, 1992.

[16] Ching-Wa Yip et al. Distributions of galaxy spectral types in the sloan digital sky survey. *The Astronomical Journal*, 128(2):585–609, 2004.

[17] Chris Lintott et al. Galaxy zoo 1: data release of morphological classifications for nearly 900 000 galaxies. *Monthly Notices of the Royal Astronomical Society*, 410:166–178, January 2011.

[18] David W. Hogg et al. Ned: Level 5: The k correction. `http://ned.ipac.caltech.edu/level5/Sept02/Hogg/frames.html`.

[19] Evguenii A Rakhmanov et al. Minimal discrete energy on the sphere. *Mathematical Research Letters*, 1(6):647662, 1994.

[20] Julien Devriendt et al. Galaxy modelling. i. spectral energy distributions from far-uv to sub-mm wavelengths. *Astronomy and Astrophysics*, 350:381–398, October 1999.

[21] Jürgen Bernard et al. Multiscale visual quality assessment for cluster analysis with self-organizing maps. In *Proceedings of the SPIE Visualization and Data Analysis 2011, 78680N*, volume 7868, January 2011.

[22] Manda Banerji et al. Galaxy zoo: reproducing galaxy morphologies via machine learning. *Monthly Notices of the Royal Astronomical Society*, 406(1):342–353, 2010.

[23] Michael Storrie-Lombardi et al. Morphological classification of galaxies by artificial neural networks. *Monthly Notices of the Royal Astronomical Society*, 259 NO. 1:8–12, November 1992.

[24] Nicholas M. Ball et al. Galaxy types in the sloan digital sky survey using supervised artificial neural networks. *Monthly Notices of the Royal Astronomical Society*, 348:1038–1046, March 2004.

[25] Ofer Lahav et al. Galaxies, human eyes and artificial neural networks. *Science*, 267:859–862, Feb 1995.

[26] Thomas Villmann et al. Topology preservation in self-organizing feature maps: exact definition and measurement. *Neural Networks, IEEE Transactions on*, 8, March 1997.

[27] Sir Ronald Aylmer Fisher. Machine learning repository - iris data set. `http://archive.ics.uci.edu/ml/datasets/Iris`, 1936.

[28] Edward L. Fitzpatrick. Correcting for the effects of interstellar extinction. *Publications of the Astronomical Society of the Pacific*, 111(755):63–75, January 1999.

[29] Shafaatunnur Hasan and Siti Mariyam Shamsuddin. Multistrategy self-organizing map learning for classification problems. *Computational Intelligence and Neuroscience*, 2011:1:1–1:11, January 2011.

[30] Edwin P. Hubble. Extragalactic nebulae. *Astrophysical Journal*, 64:321–369, December 1926.

[31] Basheer I.A. and Hajmeer M. Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods*, 43:3–31, December 2000.

[32] Dr. Judith Irwin. personal communication.

[33] Judith A. Irwin. *Astrophysics, Decoding the cosmos*. John Wiley & Sons Ltd., 2008.

[34] Mansour Nasser Jadid and Daniel R. Fairbairn. Neural-network applications in predicting moment-curvature parameters from experimental data original. *Engineering Applications of Artificial Intelligence*, 9:309–319, June 1996.

[35] SHARCNET Technical Support Kaizaad Bilimorya. personal communication.

[36] Jari Kangas, Teuvo Kohonen, and Jorma Laaksonen. Variants of self-organizing maps. *IEEE Transactions on Neural Networks*, 1:93–99, March 1990.

[37] Samuel Kaski. Data exploration using self-organizing maps, 1997.

[38] Kimmo Kiviluoto. Topology preservation in self-organizing maps. In *Proceedings of International Conference on Neural Networks (ICNN)*, volume 1, pages 294–299, June 1996.

[39] Teuvo Kohonen. Self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, September 1990.

[40] Teuvo Kohonen. *Seld-Organizong Maps*, volume 30 of *Springer Series in Information Sciences*. Springer-Verlag, 3 edition, 2001.

[41] Ofer Lahav. Galaxy classification by human eyes and by artificial neural networks. *Astrophysical Letters and Communications*, 31:73, 1995.

[42] Richard Lawrence, George Almasi, and Holly Rushmeier. A scalable parallel algorithm for self-organizing maps with applications to sparse data mining problems. *Data Mining and Knowledge Discovery*, 3(2):171–195, 1999.

[43] Abbé Georges Lemaître. Un univers homogène de masse constante et de rayon croissant, rendant compte de la vitesse radialle deb nébuleuse extragalactiques. *Annales de la société scientifique de Bruxelles*, 64, Avril 1927.

[44] SHARCNET Technical Support Mark Han. personal communication.

[45] MLR. UC irvine machine learning repository. `http://archive.ics.uci.edu/ml/index.html`.

[46] Avi Naim, Kavan U. Ratnatunga, and Richard E. Griffiths. Galaxy morphology without classification: Self-organizing maps. *The Astrophysical Journal Supplement Series*, 111:357367, August 1997.

[47] NED. The nasa/ipac extragalactic database. `http://ned.ipac.caltech.edu/`.

[48] OpenMP. The OpenMP api specification for parallel programming. `http://openmp.org/wp/`.

[49] OpenMPI. OpenMPI: Open source high performance computing. `http://www.open-mpi.org/`.

[50] W. Pence. K-corrections for galaxies of different morphological types. *The Astrophysical Journal*, 203:39–51, January 1976.

[51] Georg Pölzlbauer. Survey and comparison of quality measures for self-organizing maps. In *In Fifth Workshop on Data Analysis (WDA'04)*, pages 67–82, 2004.

[52] Georg Pölzlbauer, Michael Dittenbach, and Andreas Rauber. Advanced visualization of self-organizing maps with vector fields. *Neural Networks*, 19:911922, 2006.

[53] Andrej Prša and Edward Guinan et al. Fully Automated Approaches to Analyze Large-Scale Astronomy Survey Data. In *astro2010: The Astronomy and Astrophysics Decadal Survey*, volume 2010 of *Astronomy*, page 25, 2009.

[54] Kenneth S. Saladin. *Anatomy & Physiology - The Unity of Form and Function*. McGraw-Hill, New York, 6th edition, 2012.

[55] John W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18:401–409, May 1969.

[56] Charles R. Schmidt, Sergio J. Rey, , and André Skupin. Effects of irregular topology in spherical self-organizing maps. *International Regional Science Review*, 34(2):215–229, December 2010.

[57] R. Sibson. Slink: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16:30–34, January 1973.

[58] Mu-Chun Su, Chien-Hsing Chou, and Hsiao-Te Chang. Adding a healing mechanism in the self-organizing feature map algorithm. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, volume 6, pages 171–176, 2000.

[59] National Geographic-Palomar Observatory Sky Survey. Minnesota automated plate scanner (MAPS) catalog. `http://aps.umn.edu/`.

[60] Kadim Taşdemir and Erzsébet Merényi. A new cluster validity index for prototype based clustering algorithms based on inter- and intra-cluster density. In *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*, pages 2205 – 2211. IEEE, Neural Networks, August 2007.

[61] Kadim Taşdemir, Pavel Milenov, and Brooke Tapsall. Topology-based hierarchical clustering of self-organizing maps. *IEEE Transactions on Neural Networks archive*, 22:474–485, March 2011.

[62] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Pearson, Addison Wesley, 2006.

[63] Pekka Teerikorpi. Observational selection bias affecting the determination of the extragalactic distance scale. *Annual Review of Astronomy and Astrophysics*, 35:101–136, September 1997.

[64] Carole Thiebaut, Michel Böer, and Sylvie Roques. Steps toward the development of an automatic classifier for astronomical sources. In *Proc. SPIE 4847, Astronomical Data Analysis II, (19 December 2002)*, volume 4848, December 2002.

[65] Alfred Ultsch and H. Peter Siemon. Kohonen's self organizing feature maps for exploratory data analysis. In *In Proceedings of International Neural Networks Conference (INNC)*, pages 305–308, 1990.

[66] E. Arsuaga Uriarte and F. Díaz Martín. Topology preservation in som. In *PROCEEDINGS OF WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY*, volume 15, pages 187–190, 2006.

[67] Iren Valova, Derek Beaton, Alexandre Buerand, and Daniel MacLean. Fractal initialization for high-quality mapping with self-organizing maps. *Neural Computing and Applications*, 19:953–966, October 2010.

[68] Sidney van den Bergh. *Galaxy Morphology and Classification*. Cambridge University Press, 1998.

[69] Juha Vesanto and Esa Alhoniemi. Clustering of the self-organizing map. *IEEE Transactions on Neural networks*, 11 No. 3, May 2000.

[70] Juha Vesanto, Johan Himberg, Markus Siponen, and Olli Simula. Enhancing som based data visualization. In *Proceedings of the 5th International Conference on Soft Computing and Information/Intelligent Systems*, pages 64–67. World Scientific, 1998.

[71] Juha Vesanto, Mika Sulkava, and Jaakko Hollmen. On the decomposition of the self-organizing map distortion measure. In *Proceedings of the Workshop on Self-Organizing Maps (WSOM'03*, pages 11–16, 2003.

[72] Carl Jakob Walcher, Brent Groves, and Budavári Tamás. Fitting the integrated spectral energy distributions of galaxies. *Astrophysics and Space Science*, 331(1):1–51, 2011.

[73] Jochen Wendel and Barbara. P. Buttenfield. Formalizing guidelines for building meaningful self-organizing maps. In *GIScience 2010,Sixth international conference on Geographic Information Science,Zurich*, September 2010.

[74] Yingxin Wu and Masahiro Takatsuka. The geodesic self-organizing map and its error analysis. In *Proceeding ACSC '05 Proceedings of the Twenty-eighth Australasian conference on Computer Science*, volume 38, pages 343–351, 2005.

Appendix A

SOMs and The Iris Data Set

The purpose of this appendix is to provide an overview of the process used in modelling a set of data using self-organizing maps. The investigations performed were also used to verify the proper functioning of the three dimensional implementation of the non-batch Kohonen SOM algorithm. The content of this section will include the selection of a standard modelling dataset of known classifications. This will be used to measure the effectiveness of the code implementation at resolving the different classes present. Various SOM configuration parameters will be explored to investigate how they affect the quality of the produced map.

A.1 The dataset

The data used in this exercise is the Iris dataset from the Machine Learning Repository at U.C. Irvine[1]. The Iris dataset is comprised of fifty data points for each of for three types of irises: Iris Setosa, Iris Versicolour and Iris Virginica. Each data point has four attributes: sepal length, sepal width, petal length and petal width.

The dataset has been studied by many groups [45, 62] and is known to have one family of irises, Setosa, which is easily separable from the others. The other two species are more closely related and are a bit more difficult to resolve.

A.2 Sizing the SOM

Choosing an appropriate size for a SOM has a direct effect on the quality of the maps produced [39]. If the dimensions of a map are too small, important small-scale variations in the data may be masked. Having a map that is too large has the opposite effect where details are smeared out over a large region and become less prominent.

Two other factors which must be considered are the quality of the map and the amount of computational resources we are willing to use to obtain the final result.

[1]Machine Learning Repository: http://archive.ics.uci.edu/ml/datasets.html

The larger the map, the more prototype vectors come into play in determining the BMU. Doubling the map size in any two dimensions quadruples the number of BMUs. Doubling also affects the effective radius of the BMU region which is tuned to the new input data at each iteration of the map. For a three-dimensional SOM, the number of cells affected by the weight smoothing over the BMU region is defined by the volume of a sphere of a set radius. If we double the radius, the volume of the sphere, hence the number of affected prototypes, increases by a factor of eight. This represents a significant increase in the number of computations required.

In terms of the quality of the map, we are concerned with the possible existence of warping or twisting in the attribute space spanned by the map. In an ideal SOM, all prototypes which are similar will occupy adjacent regions in the map. As the map is tuned for the input data, it is possible to introduce variations in the prototype weights which generate like prototypes in disjoined portions of the map. For smaller maps, these separated regions cannot be too distant from each other. The smoothing of the BMU regions will tend to blend in the differences and reintegrate these prototypes into a common area. For very large maps though, it is possible that the BMU region radius is not sufficient to bridge between these distinct areas. In such cases, the regions remain distinct and the topology of the input attribute space is not preserved in the mapping process.

To evaluate the effect of different size maps, four different size maps will be investigated: $3 \times 5 \times 7$, $5 \times 5 \times 5$, $5 \times 7 \times 9$ and $10 \times 10 \times 10$.

A.3 Initializing the data

Three different methods of initializing the data and the SOM were used in this exercise. These are a direct result of the map initialization techniques discussed in section 4.5.1.

The first was to provide the data to the SOM in its raw form. The normalization label for such data will be NONE.

The second approach was to also use the raw data but to use the eight most dissimilar points in the dataset to seed the weights of original map corners, keeping the most dissimilar points the furthest apart in the SOM's grid. This approach will be labeled ADATA.

The final technique used was to perform a normalization of the input data's attribute space. This was done to ensure that all of the attributes present were scaled to a common range. This reduced the possibility of one attribute taking precedence in all of the similarity measure calculations. The per-attribute normalization data will be labeled PNORM.

A.4 The stopping criteria

A rule of thumb for the number of iterations has been given as 500 times the number of prototype vectors [29, 39]. This number was found to be insufficient for the 3D SOMs used in this study. In this exercise, all of the datasets were processed through the SOM for $500,000$ iterations or until there were no changes to the BMUs[2]. Kohonen [39] and others [29] have suggested that other measures, such as Quantization Error (QE) and Topographic Error (TE), can also be used to evaluate the convergence of the map. Though not used as termination criteria, these two measures will be investigated.

A.5 Modifications to the BMU selection

In much of the preliminary work for this thesis, it was found that upon completion of the maximum number of iterations, many data elements had not converged to a single specific BMU. Further research into the problem, by mapping the movements of the data, revealed that many of these unsettled data points were oscillating between individual sets of two or more BMUs.

In an attempt to reduce or eliminate these oscillations, two dampening techniques were investigated. The CUBE method attempts to restrict movement between adjacent BMUs in the map's coordinate system. The CUTOFF approach restricts BMU movements based on a selected minimum separation in the attribute space. These will be described in the following sections.

A.5.1 CUBE - Moderating a BMU's Nearest Neighbours

Tracking the locations of the oscillating BMUs revealed that many were adjacent to each other. As the data elements were allocated to one BMU, the neighbourhood adjustment function was enough to alter adjoining prototypes to make them more similar to the data being presented. In the next iteration, the item would jump back to the original BMU.

To reduce the oscillations, the BMU selection process was given an extra option. When selected, this option would prevent a data element from jumping to a new BMU if that new target prototype was adjacent within a unit cube surrounding the original within the SOM grid structure. This technique, CUBE, was compared to the original to see if it would help the SOM converge more quickly.

[2]Changes to the BMUs had to remain at zero for 50 iterations before the map was considered complete

A.5.2 CUTOFF - Similarity Restrictions

The second technique that was developed to moderate the selection of the BMU was based on the existing similarities within the data. If a set of data contains elements which can be classified as similar, then, there must exist within the attribute space a certain heightened similarity between those items. If an evaluation is made of the similarities between all of the data elements in a dataset, some will be found to be more similar and some more dissimilar. This is the basis upon which the SOM is built.

If we plot the similarities within the raw data, we will be able to examine the structure of the graph and find regions which share the same level of similarity. For this purpose, it is not necessary to know which items are more similar to others, just what the overall maximum similarity might be. If we know that a significant portion of items within the data share a high degree of similarity, then any similarity measure encountered during the BMU selection process which exceeds that maximum value would indicate that the prototype and data item share a similarity beyond what exists in the original dataset. The new BMU would therefore be the optimum choice for that data element.

In the implementation of the Kohonen algorithm, the BMU is chosen based on minimizing the Euclidean distance in the attribute space. A high degree of similarity is synonymous with a small Euclidean distance in the attribute space. To select an appropriate limit for restricting the BMU's movement, an analysis of the inter-data distances can be performed. Figure A.1 is a plot of the attribute space distance profile which exists within the Iris data. An examination of the figure reveals that a significant portion of inter-data distances have a value of less than 5. For the Iris data, a Euclidean distance of 5 accounts for just over 48% of all of the inter-data distances present in the original dataset.

In the analysis that follows, a number of distance restrictions will be chosen in order to evaluate their impact on the quality of the SOM. Cutoff points were chosen to restrict BMU changes at the $5\%, 10\%, 15\%, 20\%, 30\%, 40\%$ and 50% of all possible distances. These are indicated in the distance chart in Figure A.2. This new chart is just a sub-chart of the data shown in Figure A.1. Datasets analysed using this technique will be identified as CUTOFF.

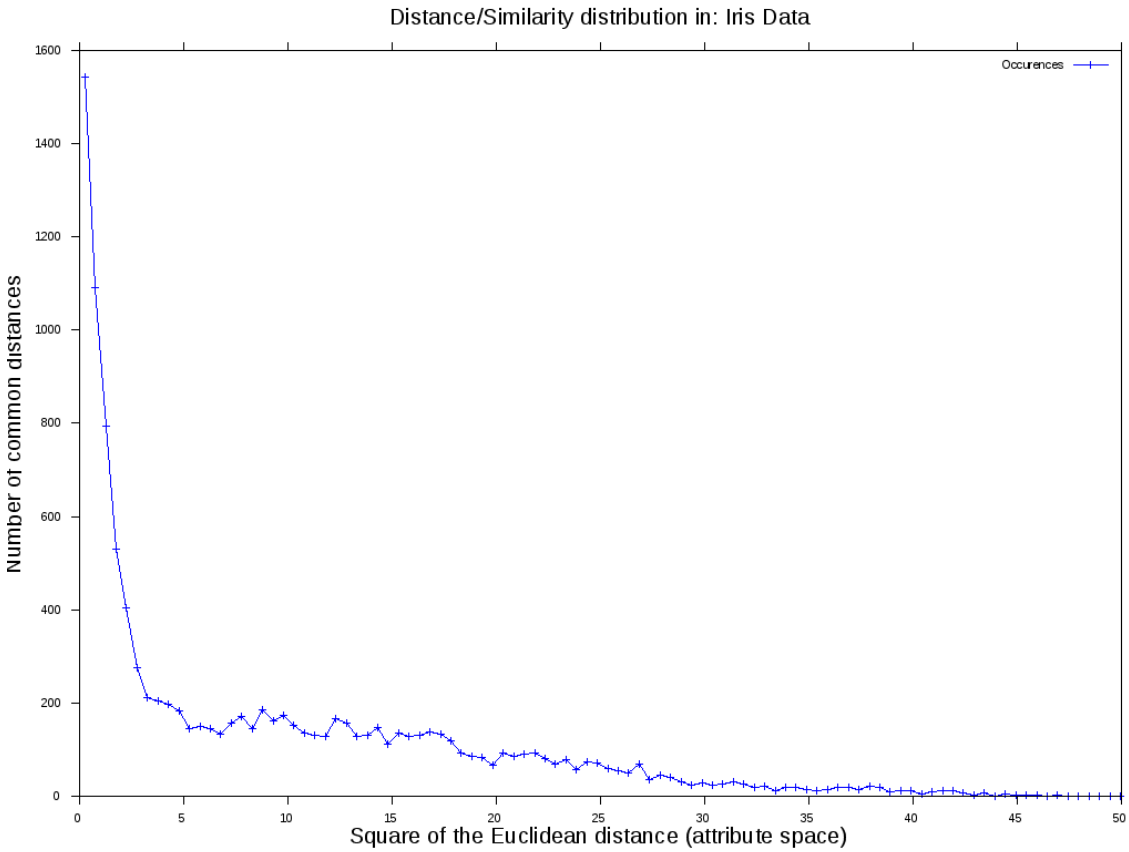

Figure A.1: The Euclidean distance profile of the Iris data.

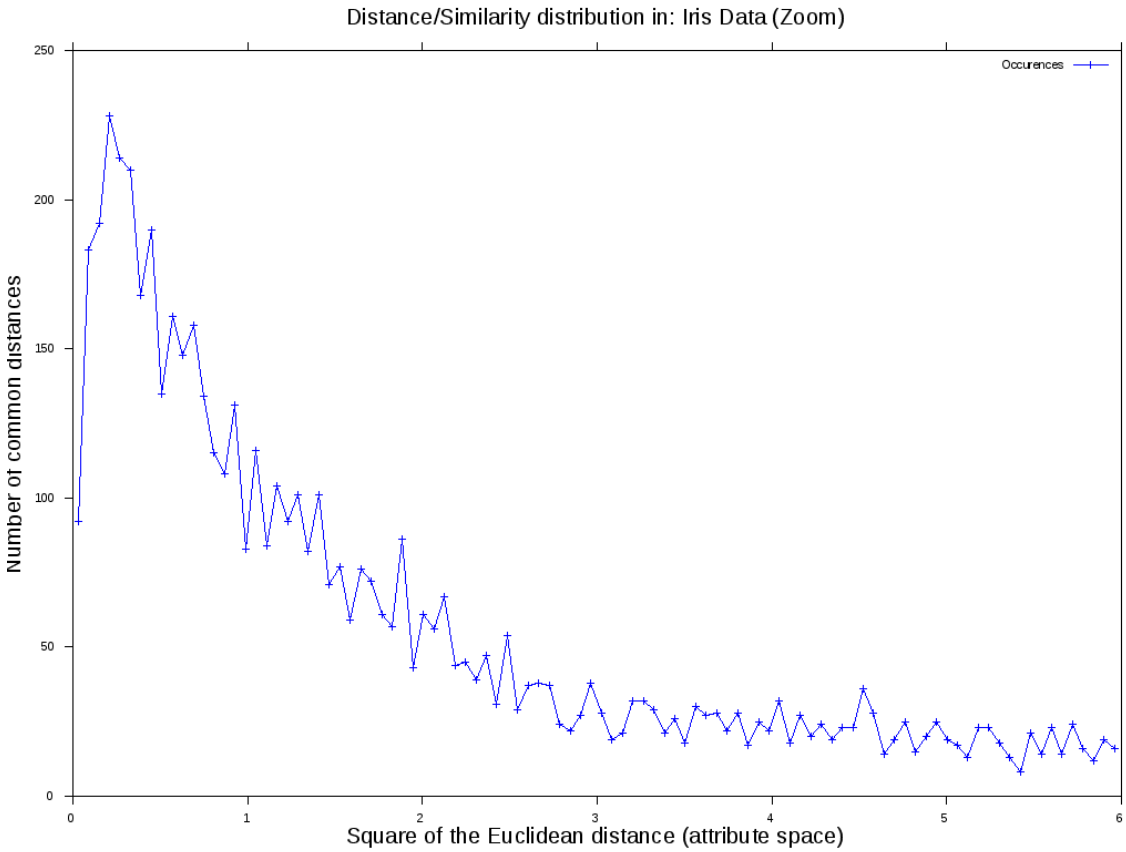

Figure A.2: The Euclidean distance profile of the Iris data (details).

A.6 The Analysis of the Iris Data

The Iris data were investigated through implementing a number of different SOM configurations. The data were preprocessed using either the ADATA, NONE or PNORM normalization schemes. The impact of the map size was investigated using maps of dimensions: $3 \times 5 \times 7$, $5 \times 5 \times 5$, $5 \times 7 \times 9$ and $10 \times 10 \times 10$. The final modifier of the mapping process was applying restrictions to the selection of the BMU to help control the oscillation problem.

The results of the investigation can be found in Tables A.1, A.2, A.3 and A.4. The main limit placed on all of the analyses performed was to set a maximum number of iterations for each dataset. This limit was set to $500,000$ iterations independent on the size of the SOM. This was done to limit the amount of resources consumed for this investigation and to provide for a common measure for cross-comparison. The size of the SOM has a direct bearing on the number of calculations required per iteration. To provide a more consistent measure for comparison, all of the data provided will be plotted against elapsed time instead of by iteration.

The effectiveness of the SOM produced can be evaluated using a number of parameters. The most important is how effective the map is at classifying the

Name	Number of Iterations	Run Time	QE	TE (Boolean)	TE (Distance)	BMU Changes	Populated Cell Count	Single Class Cells	Multi-Class Cells	# pure Setosa Cells	# pure Versicolor Cells	# pure Virginica Cells	Species Confusion
ADATA (5%)	499999	632	1.36E-001	1.13E-001	3.32E-001	42	50	47	3	38	6	3	Virginica-Versicolor=3
ADATA (10%)	499999	484	6.65E-002	2.40E-001	6.03E-001		58	53	5	10	26	17	Virginica-Versicolor=5
ADATA (15%)	499999	477	1.45E-001	8.00E-002	2.19E-001	14	57	53	4	1	23	29	Virginica-Versicolor=4
ADATA (20%)	353129	369	1.68E-001	1.27E-001	3.42E-001		57	51	6	28	10	13	Virginica-Versicolor=6
ADATA (30%)	298299	313	2.12E-001	1.47E-001	3.38E-001		66	62	4	17	19	26	Virginica-Versicolor=4
ADATA (40%)	239857	259	3.54E-001	1.53E-001	3.85E-001		65	62	3	12	24	26	Virginica-Versicolor=3
ADATA (50%)	176951	253	9.03E-001	3.93E-001	1.04E+000		57	49	8	11	21	17	Virginica-Versicolor=5 Versicolor-Setosa=3
ADATA (CUBE)	499999	571	7.12E-002	1.67E-001	4.93E-001	4	62	60	2	32	12	16	Virginica-Versicolor=2
ADATA (FULL)	499999	487	9.80E-002	2.33E-001	6.17E-001	32	57	53	4	33	9	11	Virginica-Versicolor=4
NONE (5%)	499999	530	7.35E-002	1.93E-001	5.38E-001	11	65	62	3	31	15	16	Virginica-Versicolor=3
NONE (10%)	499999	517	7.23E-002	2.47E-001	7.14E-001	1	67	65	2	27	21	17	Virginica-Versicolor=2
NONE (15%)	499999	477	8.96E-002	2.33E-001	7.67E-001	2	63	60	3	39	13	8	Virginica-Versicolor=3
NONE (20%)	371507	395	1.30E-001	1.20E-001	2.69E-001		58	56	2	35	13	8	Virginica-Versicolor=2
NONE (30%)	298274	312	2.22E-001	1.73E-001	4.11E-001		62	59	3	16	24	19	Virginica-Versicolor=3
NONE (40%)	229707	247	3.74E-001	2.13E-001	5.09E-001		71	68	3	14	25	29	Virginica-Versicolor=3
NONE (50%)	174234	200	1.09E+000	2.47E-001	7.03E-001		59	49	10	11	22	16	Virginica-Versicolor=8 Versicolor-Setosa=2
NONE (CUBE)	499999	488	6.78E-002	3.07E-001	1.16E+000	3	74	71	3	33	20	18	Virginica-Versicolor=3
NONE (FULL)	499999	514	8.86E-002	2.47E-001	7.54E-001	24	60	58	2	37	10	11	Virginica-Versicolor=2
PNORM (CUBE)	499999	525	7.69E-003	1.60E-001	4.27E-001	4	62	58	4	31	15	12	Virginica-Versicolor=4
PNORM (FULL)	497626	1785	4.60E-003	2.00E-001	6.60E-001		58	56	2	10	21	25	Virginica-Versicolor=2
PNORM (cutoff)	499999	543	7.10E-003	2.07E-001	5.49E-001	14	62	61	1	9	23	29	Virginica-Versicolor=1

Table A.1: Job Performance statistics: Iris SOM 3x5x7.

Name	Number of Iterations	Run Time	QE	TE (Boolean)	TE (Distance)	BMU Changes	Populated Cell Count	Single Class Cells	Multi-Class Cells	# pure Setosa Cells	# pure Versicolor Cells	# pure Virginica Cells	Species Confusion
ADATA (5%)	499999	570	6.819236e-02	1.933333e-01	5.896039e-01	8	66	65	1	8	29	28	Virginica-Versicolor=1
ADATA (10%)	499999	578	1.444179e-01	1.933333e-01	6.024480e-01	5	72	71	1	5	36	30	Virginica-Versicolor=1
ADATA (15%)	435211	515	1.986325e-01	1.666667e-01	4.689788e-01		65	63	2	9	30	24	Virginica-Versicolor=2
ADATA (20%)	399820	494	1.899061e-01	2.400000e-01	7.391343e-01		69	69		8	28	33	
ADATA (30%)	290768	376	6.459961e-01	2.533333e-01	8.309786e-01		62	60	2	13	25	22	Virginica-Versicolor=2
ADATA (40%)	186158	253	8.890148e-01	4.133333e-01	1.215271e+00		81	74	7	20	26	28	Virginica-Versicolor=5 Versicolor-Setosa=2
ADATA (50%)	149165	208	1.809067e+00	4.266667e-01	1.396771e+00		67	56	11	13	25	18	Virginica-Versicolor=10 Versicolor-Setosa=1
ADATA (CUBE)	418380	643	1.285176e-01	1.333333e-01	3.760777e-01		62	61	1	1	24	36	Virginica-Versicolor=1
ADATA (FULL)	499999	592	7.046591e-02	2.866667e-01	8.677894e-01	30	64	62	2	9	23	30	Virginica-Versicolor=2
NONE (5%)	499999	607	7.279311e-02	2.333333e-01	6.536873e-01	10	68	66	2	7	33	26	Virginica-Versicolor=2
NONE (10%)	499999	781	6.464116e-02	2.266667e-01	6.196376e-01	3	59	58	1	7	25	26	Virginica-Versicolor=1
NONE (15%)	430795	516	9.763395e-02	2.400000e-01	8.052421e-01		69	66	3	10	33	23	Virginica-Versicolor=3
NONE (20%)	499999	569	6.654377e-02	2.133333e-01	6.516890e-01	1	66	62	4	6	30	26	Virginica-Versicolor=4
NONE (30%)	290447	760	2.067503e-01	2.400000e-01	7.483821e-01		68	63	5	14	27	22	Virginica-Versicolor=5
NONE (40%)	174665	783	9.738952e-01	3.866667e-01	1.203951e+00		77	69	8	16	24	29	Virginica-Versicolor=5 Versicolor-Setosa=3
NONE (50%)	143043	559	1.785980e+00	4.400000e-01	1.498465e+00		72	57	15	10	29	18	Virginica-Versicolor=8 Virginica-Versicolor-Setosa=1 Versicolor-Setosa=6
NONE (CUBE)	499999	567	6.467389e-02	1.733333e-01	4.782373e-01	3	72	72		12	33	27	
NONE (FULL)	499999	572	6.906731e-02	2.533333e-01	6.595172e-01	19	67	65	2	10	35	20	Virginica-Versicolor=2
PNORM (CUBE)	499999	579	1.115112e-02	3.466667e-01	1.133831e+00	43	51	47	4	30	7	10	Virginica-Versicolor=4
PNORM (FULL)	499999	576	5.215203e-03	2.400000e-01	6.584921e-01	12	75	72	3	24	19	29	Virginica-Versicolor=3
PNORM (cutoff)	499999	877	4.218992e-03	1.733333e-01	4.862848e-01	18	72	70	2	24	21	25	Virginica-Versicolor=2

Table A.2: Job Performance statistics: Iris SOM 5x5x5.

Name	Number of Iterations	Run Time	QE	TE (Boolean)	TE (Distance)	BMU Changes	Populated Cell Count	Single Class Cells	Multi-Class Cells	# pure Setosa Cells	# pure Versicolor Cells	# pure Virginica Cells	Species Confusion
ADATA (5%)	499999	953	4.833422e-02	2.800000e-01	8.421910e-01	9	84	82	2	42	21	19	Virginica-Versicolor=2
ADATA (10%)	499999	1022	3.561658e-02	1.733333e-01	5.239058e-01	1	88	88		14	40	34	
ADATA (15%)	499999	983	5.728496e-02	1.933333e-01	5.632779e-01	1	82	81	1	44	24	13	Virginica-Versicolor=1
ADATA (20%)	328879	719	1.052941e-01	1.733333e-01	4.653005e-01		112	112		26	43	43	
ADATA (30%)	257196	597	2.413838e-01	2.666667e-01	7.664471e-01		100	95	5	40	29	26	Virginica-Versicolor=5
ADATA (40%)	221916	537	3.381061e-01	3.733333e-01	1.096343e+00		101	98	3	34	36	28	Virginica-Versicolor=3
ADATA (50%)	133685	343	2.487384e+00	6.466667e-01	2.324395e+00		89	84	5	17	40	27	Virginica-Versicolor=4 Versicolor-Setosa=1
ADATA (CUBE)	467359	933	2.094064e-02	2.466667e-01	8.939514e-01		98	96	2	39	35	22	Virginica-Versicolor=2
ADATA (FULL)	499999	959	4.149481e-02	2.200000e-01	5.982525e-01	1	94	93	1	39	36	18	Virginica-Versicolor=1
NONE (5%)	428250	862	2.105294e-02	2.266667e-01	7.389346e-01		99	98	1	31	35	32	Virginica-Versicolor=1
NONE (10%)	499999	981	5.679647e-02	1.733333e-01	4.758344e-01		74	70	4	34	21	15	Virginica-Versicolor=4
NONE (15%)	426510	851	3.711802e-02	2.733333e-01	8.862215e-01		106	106		37	38	31	
NONE (20%)	350312	764	1.105303e-01	2.000000e-01	7.153354e-01		71	68	3	17	33	18	Virginica-Versicolor=3
NONE (30%)	323702	695	1.133377e-01	1.066667e-01	2.673473e-01		109	108	1	39	40	29	Virginica-Versicolor=1
NONE (40%)	201734	493	9.888106e-01	3.600000e-01	9.206449e-01		91	89	2	16	40	33	Virginica-Versicolor=2
NONE (50%)	133288	344	2.367762e+00	4.933333e-01	1.939465e+00		84	76	8	18	33	25	Virginica-Versicolor=6 Versicolor-Setosa=2
NONE (CUBE)	499999	1058	2.765561e-02	2.266667e-01	7.385278e-01		103	101	2	42	39	20	Virginica-Versicolor=2
NONE (FULL)	499999	983	1.642954e-02	3.266667e-01	9.969395e-01		103	102	1	28	40	34	Virginica-Versicolor=1
PNORM (CUBE)	488213	971	1.794212e-03	2.533333e-01	8.364826e-01		100	98	2	36	35	27	Virginica-Versicolor=2
PNORM (FULL)	499999	980	7.831071e-03	1.466667e-01	3.912768e-01	13	82	82		5	32	45	
PNORM (cutoff)	499999	1033	1.945248e-03	1.866667e-01	4.849081e-01	6	100	100		13	41	46	

Table A.3: Job Performance statistics: Iris SOM 5x7x9.

Name	Number of Iterations	Run Time	QE	TE (Boolean)	TE (Distance)	BMU Changes	Populated Cell Count	Single Class Cells	Multi-Class Cells	# pure Setosa Cells	# pure Versicolor Cells	# pure Virginica Cells	Species Confusion
ADATA (5%)	499999	1942	1.098038e-01	8.666667e-02	3.834543e-01	21	94	94		1	47	46	
ADATA (10%)	354604	1567	5.356607e-03	1.333333e-02	3.123705e-02		138	138		40	49	49	
ADATA (15%)	330560	1432	1.934673e-02	2.600000e-01	9.435321e-01		138	137	1	43	47	47	Virginical-Versicolor=1
ADATA (20%)	255183	1263	8.126591e-02	4.200000e-01	1.238538e+00		130	130		37	50	43	
ADATA (30%)	180616	945	2.716049e-01	5.133333e-01	1.874511e+00		124	123	1	37	44	42	Virginical-Versicolor=1
ADATA (40%)	154170	826	4.203493e-01	5.600000e-01	2.104259e+00		125	123	2	41	46	36	Virginical-Versicolor=2
ADATA (50%)	80143	474	1.518571e+00	7.066667e-01	4.691634e+00		108	107	1	31	45	31	Virginical-Versicolor=1
ADATA (CUBE)	354653	1514	5.126083e-03	2.000000e-02	1.227419e-01		139	139		44	49	46	
ADATA (FULL)	499999	1910	1.028355e-02	1.533333e-01	5.312344e-01		118	118		24	46	48	
NONE (5%)	435269	1727	9.595036e-03	6.000000e-02	1.754261e-01		123	123		29	48	46	
NONE (10%)	340287	1463	1.598064e-02	2.000000e-01	8.518810e-01		136	136		42	47	47	
NONE (15%)	330361	1439	2.450332e-02	2.466667e-01	1.036988e+00		136	136		43	46	47	
NONE (20%)	244943	1173	1.430467e-01	4.133333e-01	1.263208e+00		126	126		37	45	44	
NONE (30%)	245834	1171	1.462891e-01	4.333333e-01	1.299084e+00		122	121	1	33	46	42	Virginical-Versicolor=1
NONE (40%)	155898	838	5.285568e-01	5.866667e-01	2.143314e+00		123	122	1	37	47	38	Virginical-Versicolor=1
NONE (50%)	83503	496	2.092594e+00	6.866667e-01	4.439119e+00		116	112	4	30	43	39	Virginical-Versicolor=3 Versicolor-Setosa=1
NONE (CUBE)	436268	1975	1.635742e-02	1.066667e-01	5.643787e-01		112	112		22	48	42	
NONE (FULL)	499999	1923	2.881843e-02	5.333333e-02	1.852544e-01	12	102	102		8	46	48	
PNORM (CUBE)	431537	1697	8.259257e-04	1.266667e-01	5.119261e-01		122	122		30	47	45	
PNORM (FULL)	499999	1962	1.156172e-03	1.333333e-01	3.662994e-01	4	118	118		24	50	44	
PNORM (cutoff)	387622	1437	2.989526e-04	3.333333e-02	9.511646e-02		140	140		47	49	44	

Table A.4: Job Performance statistics: Iris SOM 10x10x10.

input data. Other factors such as resource consumption costs in creating the map must also factor in the evaluation. The following sections will investigate each in turn.

A.7 Tracking BMU changes

The selection of a BMU and the subsequent tuning of the prototype values allows for the establishment of an effective SOM. As the map converges to a final representation of the input data, the number of BMU changes should decrease and eventually reach zero.

In some cases, however, distortion in the map or data with no discernable structure may delay the convergence of the map. Figures A.3 and A.4 are typical graphs showing the number of BMU changes for each of the 9 BMU selection modifiers as time series. Note that in these charts, only every 1000th point is plotted to reduce the overwhelming density of points.

Examination of these figures reveals that there is a significant impact on the convergence speed of the SOM as we increase the BMU similarity distance modifier. Variations of the CUTOFF technique provide for a dramatic decrease in the number of BMU changes. As we will see in the following sections, this come at a cost in the quality of the resultant map. The CUBE technique and the 5% CUTOFF more closely follow the unmodified approach while providing at times, a measurable decrease in the number of BMU changes. Once can also notice that the CUBE technique seems to extend the runtime of the algorithm by an appreciable amount.

All of the plots of BMU changes over time exhibit some interesting characteristics. In the two examples provided, it is possible to see a number of occasions where there are drastic increases in the number of BMU changes. In Figure A.3, the 10% data shows such a change at the 350 second time interval. the 5% data shows the same behaviour between 500 and 600 second. In Figure A.4 the same two datasets show similar patterns between the 1000 and 1300 second marks.

The runtime of all of the datasets is directly impacted by the size of the maps. This can be seen both graphically in these charts as well as in the results raw data in tables A.1 through A.4.

A.8 Quantization Error

The quantization error if often stated as a mechanism for measuring the quality of the mapping process. The QE value represents a measure of the attribute space distance between the prototypes and all of the data elements associated with it. The QE is averaged over the number of data elements present. Lower values represent higher similarity between the data and the SOM. As the map converges it is expected that these values will decrease since the SOM will be trained to represent the data more and more reliably.

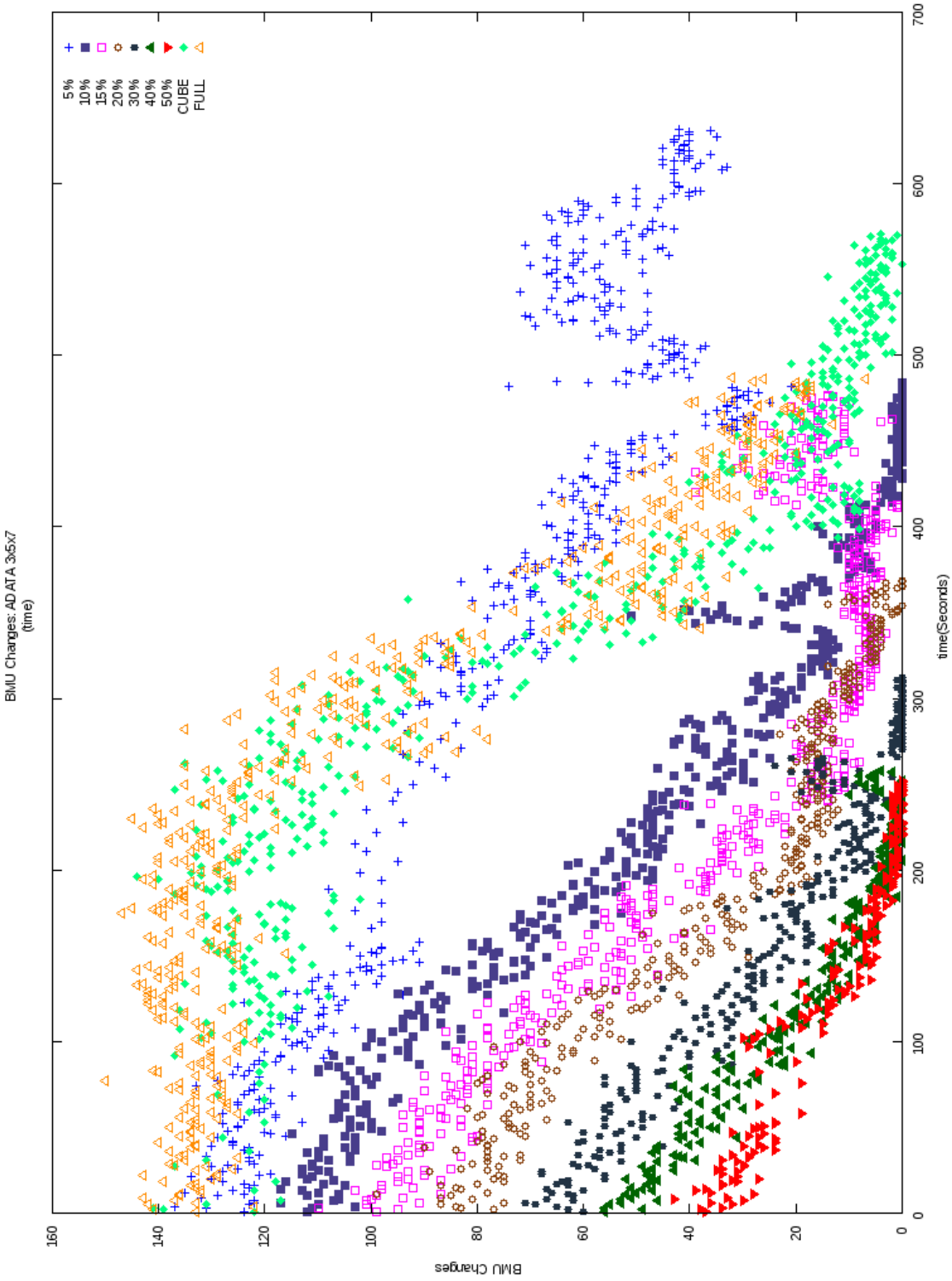

Figure A.3: BMU changes vs. time: 3x5x7 SOM using ADATA.

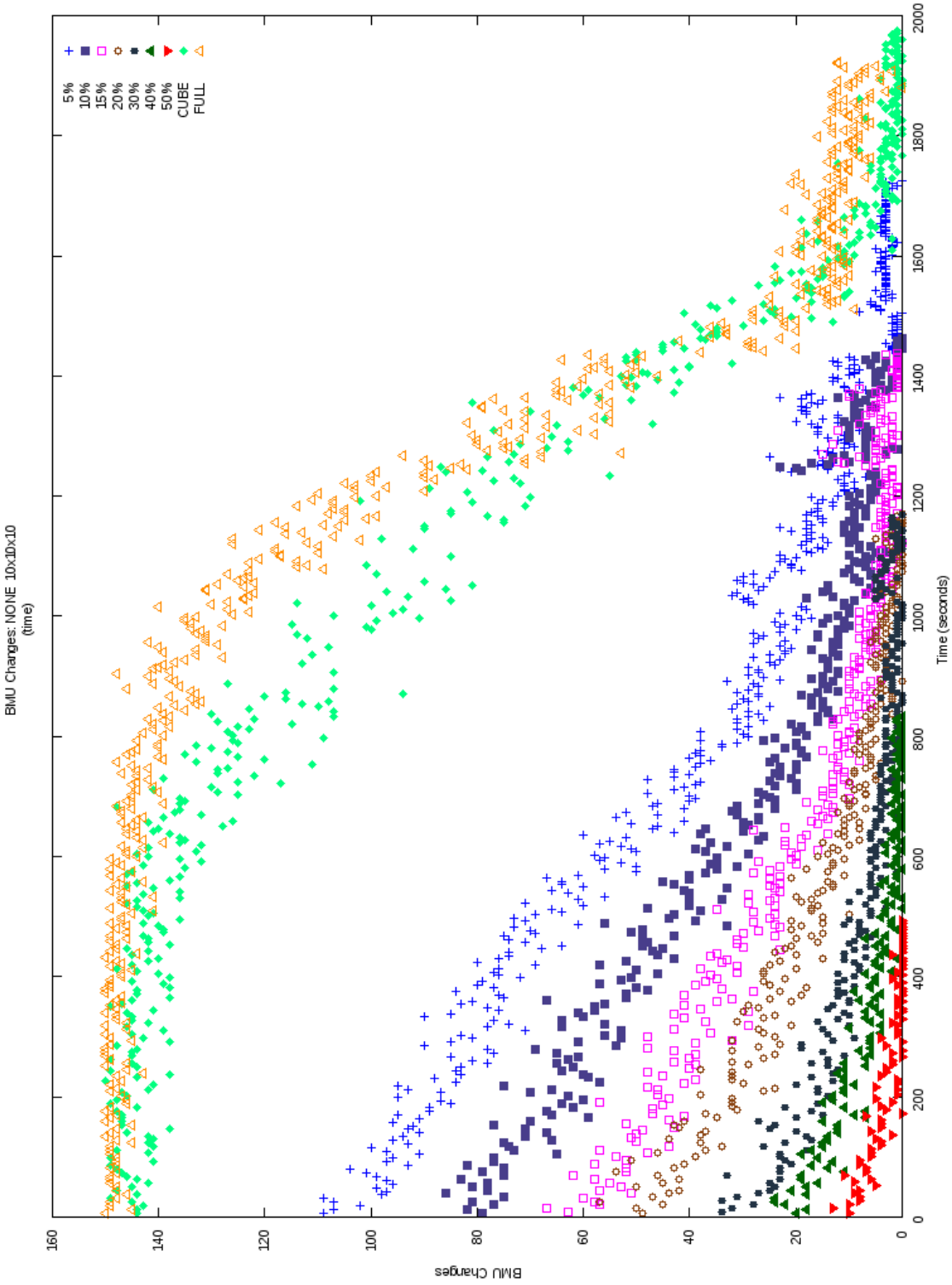

Figure A.4: BMU changes vs. time: 10x10x10 SOM with NONE.

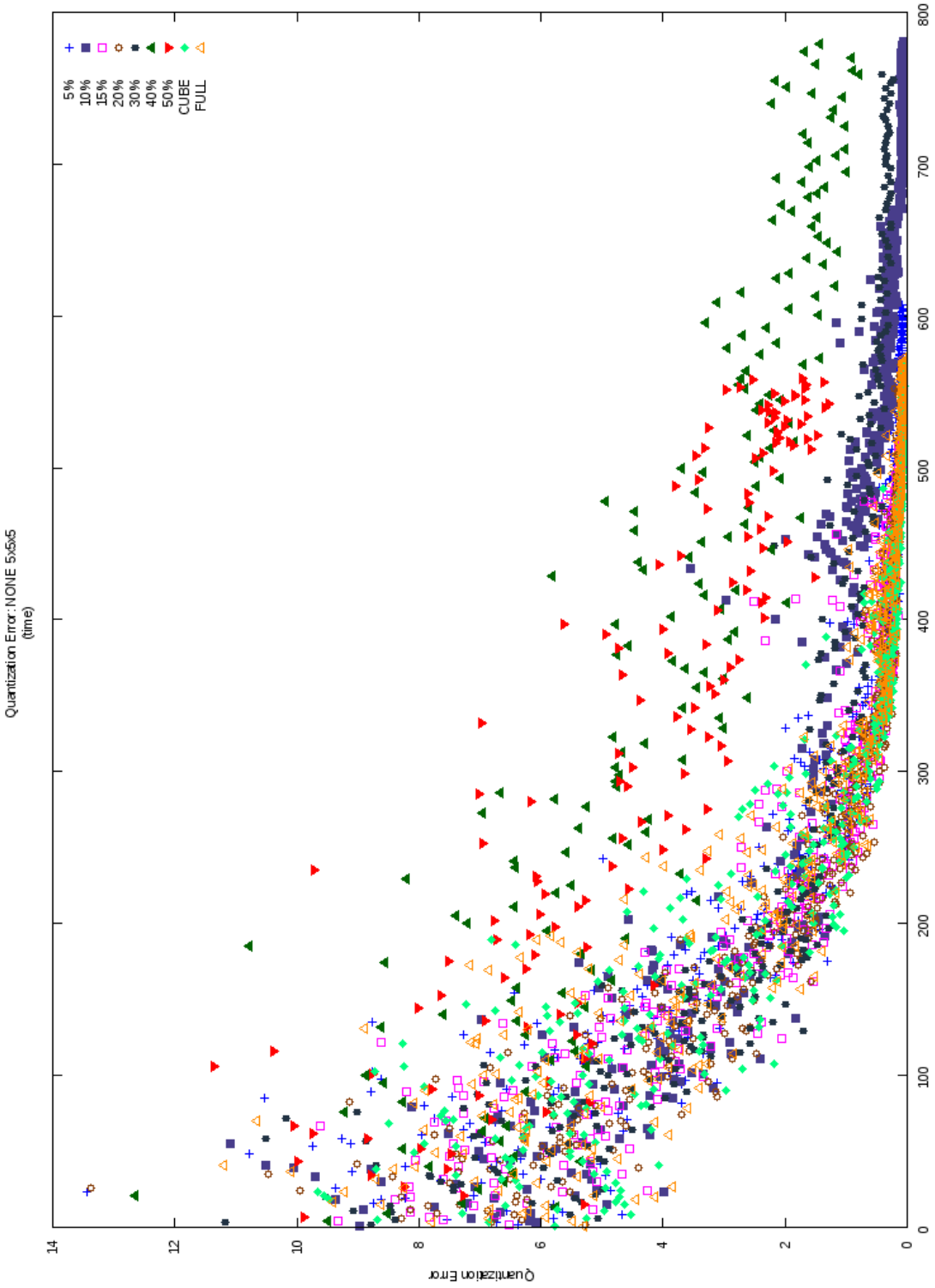

Figure A.5: Quantization Error vs. time: 5x5x5 SOM using NONE.

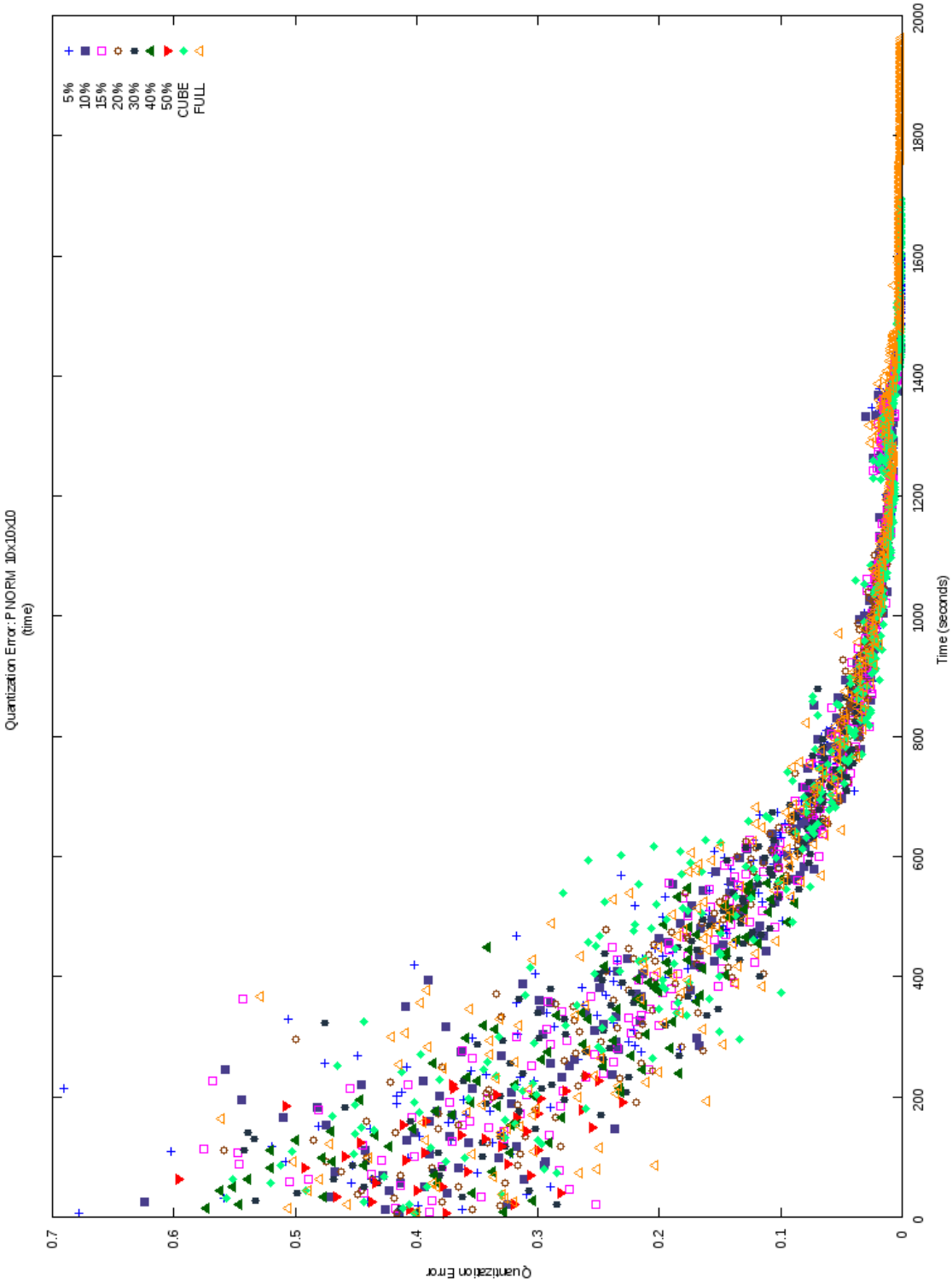

Figure A.6: Quantization Error vs. time: 10x10x10 SOM using PNORM.

QE can also be used as a termination criteria for the convergence of the SOM. Results from the analysis show that the QE behaviour is consistent over the different normalization approaches and map sizes. Two examples of the QE versus time plots are shown in Figures A.5 and A.6.

We can see that for the $5 \times 5 \times 5$ chart that for most of the life cycle of the SOM creation, the QE for all but the CUBE approach is significantly higher than that for the unmodified approach (FULL).

It is also interesting to note that when the SOM is expanded to the $10 \times 10 \times 10$ size, the range of QE values changes drastically. In Figure A.5 we find the y-axis rage from 0 to 14 whereas in Figure A.6 this drops to 0 to 0.7. This can be explained by looking at the number of SOM cells the data must map to. In the first example, there are only 125 cells in the SOM to map the 150 Iris data points. In the larger map, there are 1000 cells. There is almost an order of magnitude more room in the second map than in the first. Since there is less contention for the prototypes, each datum can eventually be assigned to a unique SOM cell. In this case, it is not evident that the lower QE values represent a better SOM and not just an attribute space with very little variance from point to point.

A.9 Termination Error

The termination error is a direct measure of the adjacency of the BMU and the second BMU selected for each data item. It is therefore, a measure of the ordering of the map. The value for TE is only increased if the first and second BMU are not adjacent within the SOM. The normal measure for TE is to simply score 0 is the BMUs are adjacent and an error of +1 if they are apart. In this thesis, this has been extended to a TE(Distance) measure which accumulates the distances in the offending BMUs. Both of these measure can be seen in the results tables.

Figures A.7 and A.8 are plots of TE for two sample runs of the Iris data. We can see in the first figure that the early convergence in the 50% and 40% cases bring about large TE. In the CUBE and 5% cases, however, we see that we get TE values that are lower but at a small runtime cost.

Figure A.8 reflects how map size can have an effect on the building of the SOM. In this example, the 50% and 40% cases have higher TE values while not significantly decreasing the run time. In the 20%, 30% and 40% samples There is little improvement in the TE values but the runtime is drawn out by almost 30%.

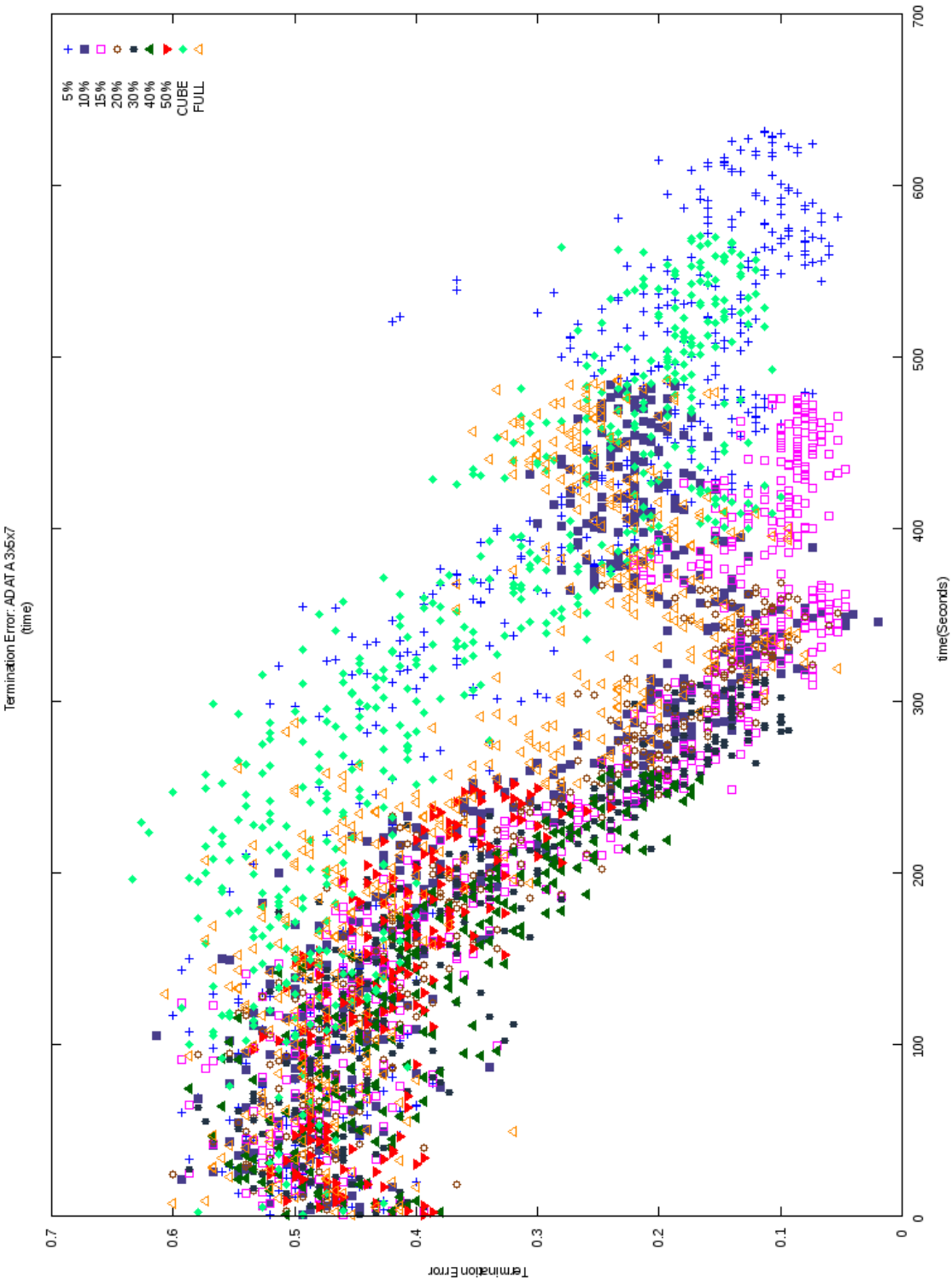

Figure A.7: Termination Error vs. time: 3x5x7 SOM using ADATA.

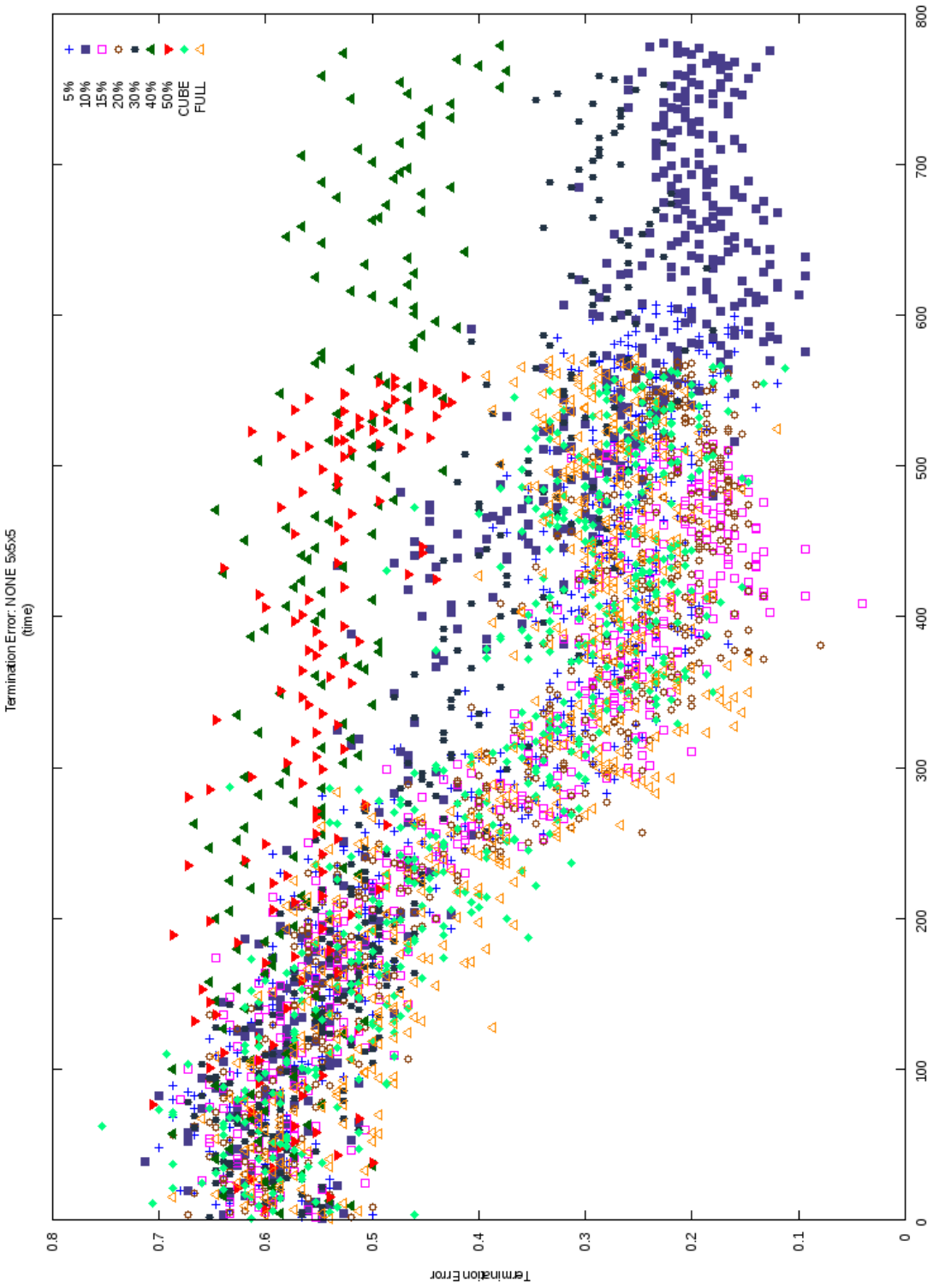

Figure A.8: Termination Error vs. time: 5x5x5 SOM using NONE.

A.10 Discussion

In this appendix we have explored the effectiveness of the implemented code in analyzing the Iris dataset. Multiple SOM sizes and data normalization approaches were tested.

From this analysis, a number of conclusions can be made. First is the importance of selecting an appropriate map size. Maps that are too large allow the data to spread out and weight interactions between cells is minimized. Such maps can be identified through the large number of populated cells and a small quantization error. An example of such a map can be seen in Figure A.9. Maps which are tool small cannot adequately represent the attribute space of the dataset. These maps will therefore have higher attribute gradients and will tend to have cells which contain multiple classes. An examination of the results tables show that for the smaller SOM such as $3 \times 5 \times 7$ the number of multi-morphology cells is larger than those for the $5 \times 7 \times 9$ and $10 \times 10 \times 10$ maps.

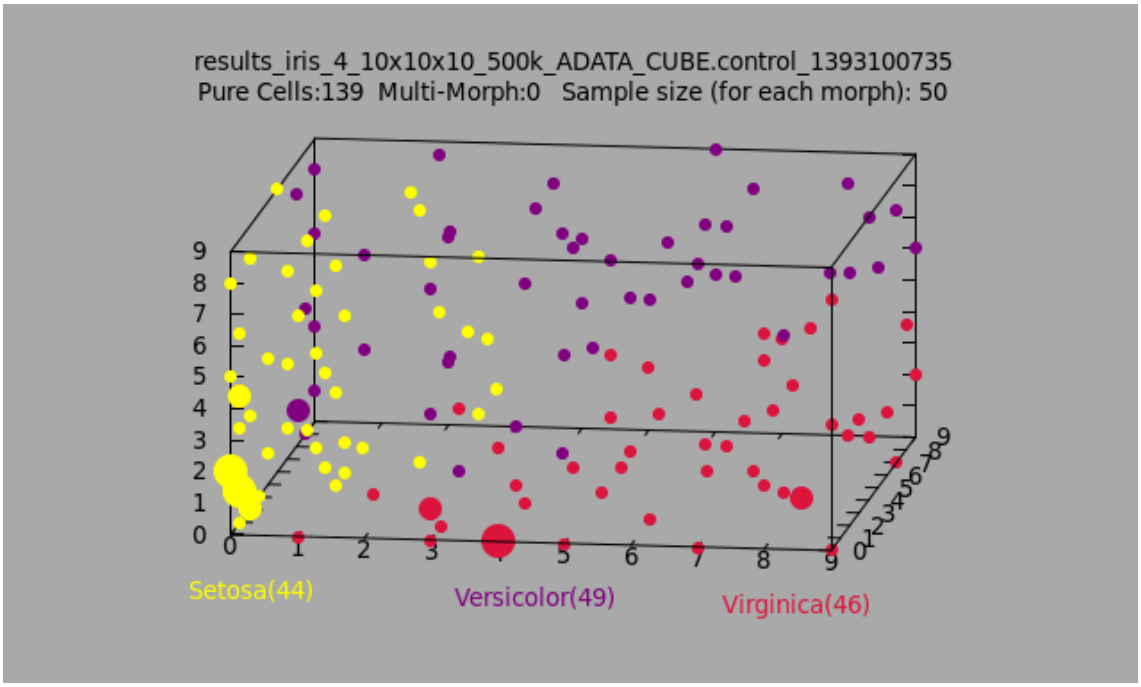

Figure A.9: SOM - 10x10x10 Iris data: ADATA and CUBE.

Mapping the number of BMU changes, TE and QE have shown that being too aggressive in moderating the BMU changes yields maps which terminate prematurely. These SOMs are also associated with larger QE and TE measures which implies, poorer maps.

Figure A.10 and A.11 are maps which provide a balance of these different approaches. Both provide for a good separation between the classes while consuming half of the resources required for larger maps.

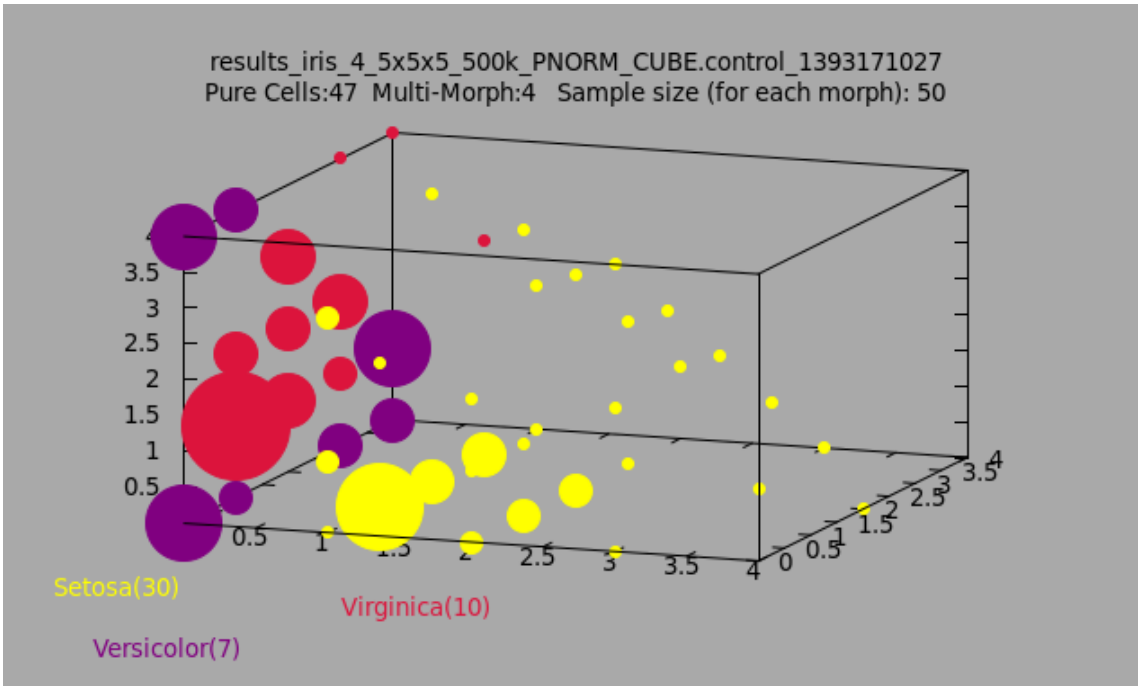

Figure A.10: SOM - 5x5x5 Iris data: PNORM and CUBE.

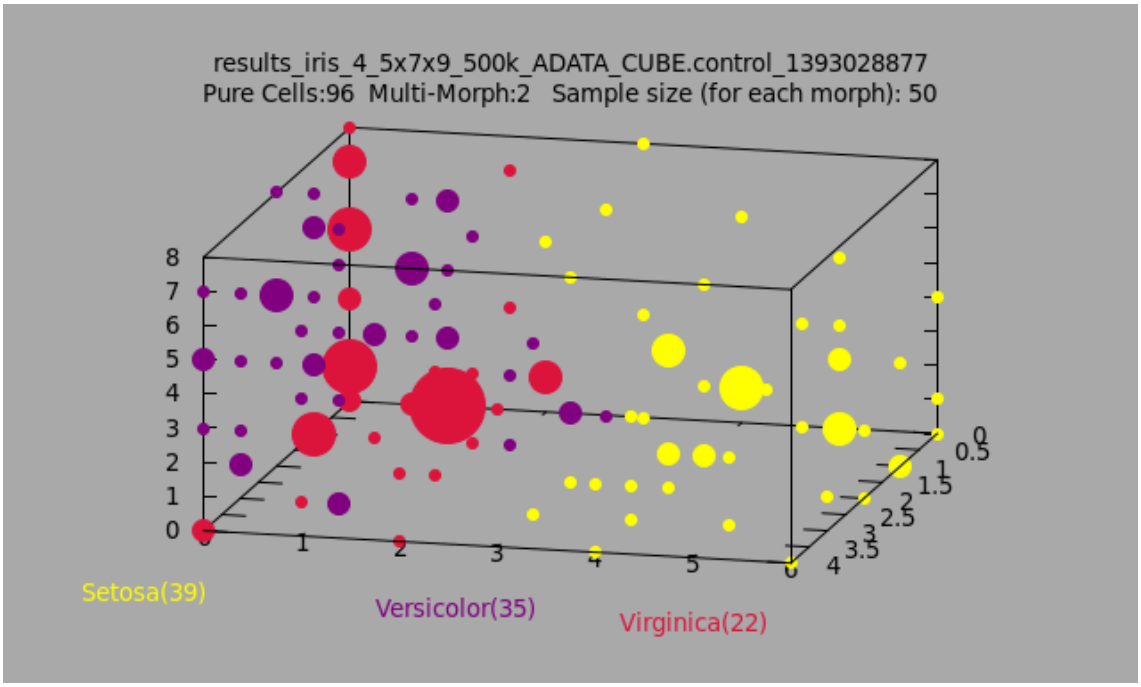

Figure A.11: SOM - 5x7x9 Iris data: ADATA and CUBE.

A.11 Conclusion

We have shown that the implementation of the Kohonen algorithm is effective in producing viable maps for classification purposes. Investigation of the BMU oscillations within the map indicate that they may be an artefact of the attribute gradient within the map being too "flat", resulting in adjacent cells being essentially identical. The examination of the various BMU selection modifiers has shown that the CUBE approach is effective in minimizing the QE and TE values though at the expense of sometimes increasing the run time of the overall algorithm.

Experimentation in managing the BMU oscillations through restricting attribute space distances showed promise. Most of these approaches caused significant changes in quality of the maps produced. Only the 5% moderator showed promise in improving both QE and TE values as compared to those of the raw data. As with the CUBE case, this often lead to slightly longer run times. Restricting the CUTOFF fraction to the 5% level will therefore be used for the analyses in this thesis.

Further work will have to be undertaken to investigate some of the artefact present in the BMU, QE and TE charts. Though it is thought that these are caused by a reordering within the map, it is also plausible that some of these irregularities may be caused by over-fitting the data. Such investigations, however, are beyond the scope of this thesis.

Appendix B

NED Data Extracts

This appendix examines the data retrieval steps used to extract data from NED. This site was used as the exclusive source for the galaxy and SED data used in this research. It provides for a consistent dataset which incorporates results from multiple independent research teams. All of the data exported from NED were kept in a MySQL database for local processing.

The NED site allows for data to be exported two different ways. The first technique involves submitting batch requests. The batch interface involves submitting email requests to nedbatchipac.caltech.edu. The content of the body of the email contains the parameters for the request.

For the purpose of this research, the NED batch requests were used to collect as many galaxy names as possible which conformed to our $Z < 0.1$ requirement. Other parameters which helped refine the searches include: Right ascension, declination and object type. In all cases, batch searches of NED required that all objects returned be of type: Galaxy. Preliminary requests were based on general catalogue names such as NGC, UGC, SDSS. Because the NED batch requests were limited in the number of objects they could return, this approach was not as effective as expected. Follow up requests to find more galaxies subdivided the night sky into quadrants. Requests which returned an incomplete set of responses were subdivided into smaller quadrants and resubmitted.

Batch requests submitted to NED returned the NED preferred object name, the object type, RA and DEC as well as the number of photometric measurements present in NED. A sample header of a batch submission is shown below. These lists were then used to download detailed information using the second NED interface.

```
File Name: BELAND_d_, Processing Time: Wed Mar 13 15:39:03
Output option: standard
************************** SEARCH REQUESTS ******************************
IN_CSYS : equatorial   ,  IN_EQUINOX : J2000.0
OUT_CSYS: equatorial   ,  OUT_EQUINOX: J2000.0
EXTENDED_NAME_SEARCH: Yes
SORTED_BY_FIELD     : RA_or_Longitude
REDSHIFT_VELOCITY   : 10000.0
DEFAULT_RADIUS      : 5.000000
DEFAULT_BEGIN_YEAR  : 1700
DEFAULT_END_YEAR    : 2013
IAU NAME STYLE      : S

PARAMETERS:
   RA:  Between 14h00m00.00000s and 15h00m00.00000s
```

```
   DEC:  Between 10.0000000000 and +20d00m00.0000s
   Name prefix (ALL of the following):
   INCLUDE ANY:
          Galaxies
   EXCLUDE ANY:

   REDSHIFT:
           Less Than   30000.000000

************************* SEARCH RESULTS *****************************

PARAMETERS
2946 object(s) found.
#--Object Name------------Type-------------Position---------Dist.-Ref-Note-Phot
1   2MASX J14000229+13470 G      14h00m02.26s  +13d47m10.0s   0.0   0   0   24
2   KUG 1357+161          G      14h00m02.59s  +15d55m14.0s   0.0   6   0   14
3   2MASX J14000420+10515 G      14h00m04.24s  +10d51m58.8s   0.0   0   0   24
```

Sample NED Batch results file.[1]

The second approach used to extract data from NED involved an iteration of multiple steps.

Check duplicates: Every list of galaxy names provided to this routine checked the local database to see if the galaxy name was already present. It also verified to ensure the galaxy wasn't present under another name or synonym.

Bulk Data: Download some basic information on each galaxy: RA, DEC, number of photometric measurements, major and minor axes dimensions, Galactic extinction for the visible wavelengths, D (Virgo+GA) and z.

Get SED data: For each galaxy download the current spectral energy distribution data.

Get cross-references: Each galaxy description in NED can be made up of contributions from multiple studies. the cross-reference table lists all of the synonyms under which a galaxy is known. This information is used to prevent having duplicate entries in the database.

The iterative portion of the data extracts involved adding new galaxies to the database. Every time a new set of galaxy names was generated, all of the above steps were performed. Additionally over the course of this research, many updates were performed by the NED team. These updates included new or updated data for the required fields as well as many new SED measurements. Periodical checks were performed to see if the number of photometric measurements changed for any particular galaxy. If so, updates were performed.

All of the downloads through the web interface to NED returned bar-separated data. These files were subsequently parsed and the appropriate database tables updated. Though mostly automated, these steps still required significant manual intervention. The process of obtaining data from NED, however, was quite efficient once the processes involved were grasped.

[1] note sure if I need this? I am not showing the format of any other downloads for comparison.

Appendix C

Results

C.1 Processing Results

The purpose of this appendix is to provide an overview of the datasets and how they were analyzed through the course of this thesis.

Results are presented using three tables for each dataset. The first table describes the number of galaxies of each morphology used in the training set for building the SOM, the number of unique galaxies present in that training set, the actual number of galaxies present in the complete dataset as well as the percentage representation of each morphology.

The second table shows the attributes present in the data. For each dataset a list of the individual frequencies and the frequency band they represent is given.

The last table gives SOM performance measures for each technique used in the analysis. Also present are runtime characteristics for each job so that a comparison can be drawn between each approach.

Pat_130_10042_5	S	P	I	E	L	NED Unknowns	NED Knowns
Training sample size:	434	434	434	434	434		
Training (unique galaxies):	424	311	134	244	427		
Full dataset:	6999	630	134	244	427	1575	8434
% Morph.:	82.99	7.47	1.59	2.89	5.06		

Observed Frequencies: 5	
Frequency Band	Observed Frequencies
Radio	
Millimeter	
SubMillimeter	
FIR	3000000000000
MIR	
NIR	138000000000000, 182000000000000, 240000000000000
Visual	681000000000000
Ultraviolet	
X-ray	
Gamma-ray	

Analysis Technique	S	P	I	E	L	Overall %	Δ BMU	CPU Seconds	QE	TE	TE (dist)	Single Morph.	Multi Morph.	% Map Coverage
ADATA_CUBE:							398	119528	5.59E+34	3.82E-01	1.37E+00	63	211	86.98
	43	51	79	67	51	54								
	29	26	0	0	0	29								
	100 (119)	0 (14)	0 (0)	0 (0)	0 (0)	89								
ADATA_CUTOFF:							443	122834	4.94E+34	2.81E-01	1.09E+00	83	191	86.98
	43	50	77	58	51	52								
	33	30	0	0	0	33								
	100 (164)	25 (8)	0 (0)	0 (0)	0 (0)	96								
ADATA_FULL:							1141	123538	8.26E+34	3.13E-01	1.12E+00	90	141	73.33
	48	35	82	51	47	49								
	34	17	0	0	0	33								
	100 (189)	13 (15)	0 (0)	0 (0)	0 (0)	93								
NONE_CUBE:							412	121802	6.84E+34	4.53E-01	1.69E+00	63	197	82.54
	56	51	78	63	46	55								
	39	30	0	0	0	38								
	100 (120)	11 (19)	0 (0)	0 (0)	0 (0)	87								
NONE_CUTOFF:							612	121503	6.92E+34	2.58E-01	8.10E-01	106	123	72.70
	52	37	75	51	48	50								
	40	17	0	0	0	39								
	100 (190)	17 (18)	0 (0)	0 (0)	0 (0)	92								
NONE_FULL:							1130	120420	8.73E+34	2.60E-01	8.76E-01	118	95	67.62
	36	39	71	44	55	46								
	24	24	0	0	0	24								
	100 (205)	50 (12)	0 (0)	0 (0)	0 (0)	97								
PNORM_CUBE:							1081	55548	1.79E-03	4.73E-01	1.83E+00	49	216	84.13
	58	43	81	55	43	52								
	50	24	0	0	0	49								
	100 (190)	8 (13)	0 (0)	0 (0)	0 (0)	94								
PNORM_CUTOFF:							1164	124920	1.52E-003	4.42E-001	1.73E+000	64	202	84.44
	50	50	79	58	43	52								
	0	0	0	0	0	0								
	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0								
PNORM_FULL:							1071	120210	1.68E-03	4.48E-01	1.59E+00	56	212	85.08
	54	44	79	62	52	55								
	40	30	0	0	0	39								
	100 (18)	50 (4)	0 (0)	0 (0)	0 (0)	90								

Analysis results for Pat_130_10042_5.

Pat_952_1386_5	S	P	I	E	L	NED Unknowns	NED Knowns
Training sample size:	151	151	151	151	151		
Training (unique galaxies):	107	11	2	109	122		
Full dataset:	202	11	2	240	485	318	940
% Morph.:	21.49	1.17	0.21	25.53	51.60		

Observed Frequencies: 5	
Frequency Band	Observed Frequencies
Radio	
Millimeter	
SubMillimeter	
FIR	
MIR	
NIR	138000000000000, 182000000000000, 240000000000000
Visual	465000000000000, 641000000000000
Ultraviolet	
X-ray	
Gamma-ray	

Analysis Technique	S	P	I	E	L	Overall %	Δ BMU	CPU Seconds	QE	TE	TE (dist)	Single Morph.	Multi Morph.	% Map Coverage
ADATA_CUBE:							35	37875	2.34E+34	2.54E-01	1.20E+00	73	41	36.19
	56	100	100	71	70	67								
	44	0	0	38	53	48								
	87 (15)	0 (0)	0 (0)	76 (25)	93 (40)	86								
ADATA_CUTOFF:							131	39315	9.35E+34	2.17E-01	7.03E-01	83	65	46.98
	75	100	100	72	69	73								
	41	0	0	46	52	49								
	92 (25)	0 (0)	0 (0)	62 (8)	94 (32)	89								
ADATA_FULL:							114	36738	1.29E+34	1.13E-01	3.14E-01	101	43	45.71
	66	100	100	65	70	68								
	44	0	0	31	54	47								
	83 (18)	0 (0)	0 (0)	83 (24)	96 (56)	90								
NONE_CUBE:							25	38696	1.28E+34	1.91E-01	5.49E-01	82	58	44.44
	79	100	100	68	68	73								
	52	0	0	26	51	46								
	88 (16)	0 (0)	0 (0)	96 (25)	94 (33)	93								
NONE_CUTOFF:							97	37456	4.70E+34	3.02E-01	1.01E+00	116	73	60.00
	78	100	100	79	76	78								
	41	0	0	41	38	39								
	75 (20)	0 (0)	0 (0)	71 (24)	91 (56)	83								
NONE_FULL:							90	37864	2.34E+34	8.48E-02	2.26E-01	70	38	34.29
	63	100	100	72	69	69								
	46	0	0	50	62	57								
	100 (14)	0 (0)	0 (0)	95 (19)	95 (22)	96								
PNORM_CUBE:							55	37117	2.87E-04	1.70E-01	4.98E-01	108	42	47.62
	67	100	100	79	64	71								
	0	0	0	0	100	62								
	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0								
PNORM_CUTOFF:							104	38229	3.82E-04	1.70E-01	5.08E-01	78	39	37.14
	65	100	100	66	65	67								
	0	0	0	0	100	62								
	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0								
PNORM_FULL:							93	38330	2.10E-04	2.34E-01	8.18E-01	116	60	55.87
	82	100	100	80	73	79								
	0	0	0	0	100	62								
	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0								

Analysis results for Pat_952_1386_5.

Pat_1030_1261_7	S	P	I	E	L	NED Unknowns	NED Knowns
Training sample size:	56	56	56	56	56		
Training (unique galaxies):	54	44	11	6	21		
Full dataset:	1063	136	11	6	21	15	1237
% Morph.:	85.93	10.99	0.89	0.49	1.70		

Observed Frequencies: 7	
Frequency Band	Observed Frequencies
Radio	
Millimeter	
SubMillimeter	
FIR	5000000000000
MIR	12000000000000
NIR	138000000000000, 182000000000000, 240000000000000
Visual	379000000000000, 681000000000000
Ultraviolet	
X-ray	
Gamma-ray	

Analysis Technique	S	P	I	E	L	Overall %	Δ BMU	CPU Seconds	QE	TE	TE (dist)	Single Morph.	Multi Morph.	% Map Coverage
ADATA_CUBE:							2	13810	1.35E+34	2.57E-01	7.80E-01	62	21	26.35
Training:	78	75	82	100	81	79								
New data:	51	22	0	0	0	48								
No Map:	100 (308)	4 (23)	0 (0)	0 (0)	0 (0)	93								
ADATA_CUTOFF:							2	13770	8.65E+33	1.68E-01	5.33E-01	87	16	32.70
Training:	89	82	91	100	100	89								
New data:	65	20	0	0	0	61								
No Map:	100 (433)	33 (36)	0 (0)	0 (0)	0 (0)	94								
ADATA_FULL:							27	13942	2.73E+34	1.25E-01	3.78E-01	40	16	17.78
Training:	70	52	82	100	67	66								
New data:	48	12	0	0	0	45								
No Map:	100 (140)	50 (14)	0 (0)	0 (0)	0 (0)	95								
NONE_CUBE:							2	12900	1.91E+34	8.21E-02	2.49E-01	53	15	21.59
Training:	74	57	82	100	62	68								
New data:	46	13	0	0	0	43								
No Map:	100 (206)	33 (15)	0 (0)	0 (0)	0 (0)	95								
NONE_CUTOFF:							1	14660	8.24E+33	1.04E-01	2.59E-01	65	17	26.03
Training:	76	66	91	100	81	76								
New data:	48	9	0	0	0	44								
No Map:	100 (354)	19 (26)	0 (0)	0 (0)	0 (0)	94								
NONE_FULL:							26	13723	9.02E+33	1.46E-01	4.53E-01	59	18	24.44
Training:	85	64	91	100	67	76								
New data:	47	29	0	0	0	45								
No Map:	100 (237)	45 (20)	0 (0)	0 (0)	0 (0)	95								
PNORM_CUBE:							43	13537	9.04E-04	1.04E-01	3.16E-01	49	14	20.00
Training:	69	59	100	100	62	68								
New data:	21	9	0	0	0	20								
No Map:	100 (80)	43 (7)	0 (0)	0 (0)	0 (0)	95								
PNORM_CUTOFF:							26	13454	2.62E-03	1.75E-01	4.78E-01	40	14	17.14
Training:	72	77	91	100	76	77								
New data:	0	0	0	0	0	0								
No Map:	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0								
PNORM_FULL:							13	14192	1.01E-03	2.54E-01	7.27E-01	52	16	21.59
Training:	67	70	82	100	71	71								
New data:	44	11	0	0	0	42								
No Map:	100 (39)	67 (3)	0 (0)	0 (0)	0 (0)	97								

Analysis results for Pat_1030_1261_7.

Pat_1046_1235_7	S	P	I	E	L	NED Unknowns	NED Knowns
Training sample size:	64	64	64	64	64		
Training (unique galaxies):	60	50	17	9	13		
Full dataset:	1017	162	17	9	13	13	1218
% Morph.:	83.50	13.30	1.40	0.74	1.07		

Observed Frequencies: 7	
Frequency Band	Observed Frequencies
Radio	
Millimeter	
SubMillimeter	
FIR	3000000000000, 5000000000000
MIR	
NIR	138000000000000
Visual	379000000000000,468000000000000,641000000000000, 681000000000000
Ultraviolet	
X-ray	
Gamma-ray	

Analysis Technique	S	P	I	E	L	Overall %	Δ BMU	CPU Seconds	QE	TE	TE (dist)	Single Morph.	Multi Morph.	% Map Coverage
ADATA_CUBE:							0	15431	2.85E+33	1.41E-01	5.69E-01	84	16	31.75
	87	80	94	100	85	86								
	45	34	0	0	0	44								
	100 (403)	29 (51)	0 (0)	0 (0)	0 (0)	91								
ADATA_CUTOFF:							87	16941	1.01E+34	1.53E-01	4.91E-01	72	20	29.21
	85	86	88	100	69	85								
	59	37	0	0	0	57								
	98 (215)	43 (37)	0 (0)	0 (0)	0 (0)	89								
ADATA_FULL:							9	16037	4.49E+33	3.00E-01	9.06E-01	81	16	30.79
	80	78	100	100	77	83								
	39	43	0	0	0	39								
	99 (337)	38 (52)	0 (0)	0 (0)	0 (0)	91								
NONE_CUBE:							3	15212	2.64E+34	1.75E-01	6.57E-01	61	16	24.44
	68	68	71	100	69	70								
	37	37	0	0	0	37								
	100 (244)	39 (36)	0 (0)	0 (0)	0 (0)	92								
NONE_CUTOFF:							90	15921	1.09E+34	1.59E-01	5.09E-01	62	15	24.44
	68	68	76	100	85	72								
	31	30	0	0	0	30								
	100 (318)	29 (42)	0 (0)	0 (0)	0 (0)	91								
NONE_FULL:							3	16614	1.41E+34	1.28E-01	2.89E-01	41	14	17.46
	62	64	35	78	85	62								
	38	37	0	0	0	38								
	99 (195)	36 (33)	0 (0)	0 (0)	0 (0)	89								
PNORM_CUBE:							50	16134	5.59E-04	1.84E-01	5.51E-01	79	17	30.48
	85	82	76	100	100	85								
	1	98	0	0	0	11								
	100 (1)	0 (0)	0 (0)	0 (0)	0 (0)	100								
PNORM_CUTOFF:							27	16889	1.99E-03	1.31E-01	3.99E-01	54	19	23.17
	68	66	82	100	54	70								
	0	2	0	0	0	0								
	100 (4)	0 (0)	0 (0)	0 (0)	0 (0)	100								
PNORM_FULL:							18	15595	1.91E-03	2.59E-01	7.54E-01	67	18	26.98
	73	70	71	100	85	74								
	0	1	0	0	0	0								
	0 (0)	100 (1)	0 (0)	0 (0)	0 (0)	100								

Analysis results for Pat_1046_1235_7.

Pat_1090_1145_7	S	P	I	E	L	NED Unknowns	NED Knowns
Training sample size:	47	47	47	47	47		
Training (unique galaxies):	46	37	7	4	20		
Full dataset:	989	113	7	4	20	8	1133
% Morph.:	87.29	9.97	0.62	0.35	1.77		

Observed Frequencies: 7	
Frequency Band	Observed Frequencies
Radio	
Millimeter	
SubMillimeter	
FIR	5000000000000
MIR	25000000000000
NIR	138000000000000, 182000000000000, 240000000000000
Visual	379000000000000, 681000000000000
Ultraviolet	
X-ray	
Gamma-ray	

Analysis Technique	S	P	I	E	L	Overall %	Δ BMU	CPU Seconds	QE	TE	TE (dist)	Single Morph.	Multi Morph.	% Map Coverage
ADATA_CUBE:							1	12157	1.85E+34	1.79E-01	5.78E-01	64	17	25.71
	83	68	100	100	95	82								
	45	17	0	0	0	42								
	100 (325)	26 (23)	0 (0)	0 (0)	0 (0)	95								
ADATA_CUTOFF:							0	10295	1.88E+34	8.51E-02	2.57E-01	60	16	24.13
	85	70	100	100	85	82								
	43	16	0	0	0	41								
	100 (244)	33 (21)	0 (0)	0 (0)	0 (0)	94								
ADATA_FULL:							0	11632	2.94E+34	1.45E-01	4.58E-01	52	9	19.37
	72	68	100	100	75	74								
	36	15	0	0	0	34								
	100 (240)	48 (23)	0 (0)	0 (0)	0 (0)	95								
NONE_CUBE:							0	11379	1.78E+34	1.19E-01	5.10E-01	75	12	27.62
	80	78	86	100	95	83								
	45	17	0	0	0	43								
	100 (361)	18 (34)	0 (0)	0 (0)	0 (0)	92								
NONE_CUTOFF:							4	11206	2.00E+34	1.11E-01	3.48E-01	58	11	21.90
	83	68	86	100	85	79								
	46	11	0	0	0	43								
	100 (224)	33 (21)	0 (0)	0 (0)	0 (0)	94								
NONE_FULL:							0	10397	1.53E+34	1.28E-01	3.94E-01	87	9	30.48
	91	95	100	100	100	95								
	41	17	0	0	0	40								
	100 (504)	21 (47)	0 (0)	0 (0)	0 (0)	92								
PNORM_CUBE:							6	11584	1.35E-03	8.51E-02	2.46E-01	49	19	21.59
	76	76	86	100	85	79								
	34	16	0	0	0	33								
	100 (210)	42 (12)	0 (0)	0 (0)	0 (0)	96								
PNORM_CUTOFF:							5	11412	6.70E-004	1.79E-001	6.22E-001	64	16	25.40
	78	78	86	100	95	82								
	0	0	0	0	0	0								
	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0								
PNORM_FULL:							0	11074	1.44E-04	1.83E-01	5.39E-01	92	8	31.75
	93	92	100	100	100	95								
	12	14	0	0	0	12								
	100 (314)	33 (33)	0 (0)	0 (0)	0 (0)	93								

Analysis results for Pat_1090_1145_7.

Pat_1100_1121_6	S	P	I	E	L	NED Unknowns	NED Knowns
Training sample size:	81	81	81	81	81		
Training (unique galaxies):	77	59	22	59	74		
Full dataset:	846	110	22	59	74	10	1111
% Morph.:	76.15	9.90	1.98	5.31	6.66		

Observed Frequencies: 7	
Frequency Band	Observed Frequencies
Radio	1400000000, 2380000000
Millimeter	
SubMillimeter	
FIR	
MIR	
NIR	138000000000000, 182000000000000, 240000000000000
Visual	681000000000000
Ultraviolet	
X-ray	
Gamma-ray	

Analysis Technique	S	P	I	E	L	Overall %	Δ BMU	CPU Seconds	QE	TE	TE (dist)	Single Morph.	Multi Morph.	% Map Coverage
ADATA_CUBE:							31	18985	2.23E+34	3.75E-01	1.32E+00	112	49	51.11
	81	78	95	88	64	78								
	40	13	0	0	0	39								
	100 (250)	20 (20)	0 (0)	0 (0)	0 (0)	94								
ADATA_CUTOFF:							99	19723	4.20E+34	2.05E-01	6.63E-01	84	58	45.08
	81	80	100	92	66	80								
	43	32	0	0	0	42								
	100 (225)	6 (17)	0 (0)	0 (0)	0 (0)	93								
ADATA_FULL:							92	19966	1.23E+34	3.83E-01	1.22E+00	116	49	52.38
	73	78	95	88	70	78								
	45	20	0	0	0	43								
	100 (227)	38 (16)	0 (0)	0 (0)	0 (0)	95								
NONE_CUBE:							20	19735	1.27E+34	2.22E-01	6.25E-01	84	44	40.63
	58	64	91	80	64	68								
	39	29	0	0	0	39								
	100 (64)	67 (6)	0 (0)	0 (0)	0 (0)	97								
NONE_CUTOFF:							105	19521	2.55E+34	2.89E-01	9.75E-01	96	60	49.52
	77	83	100	90	70	81								
	45	18	0	0	0	43								
	100 (158)	17 (12)	0 (0)	0 (0)	0 (0)	94								
NONE_FULL:							118	18535	3.47E+34	1.60E-01	4.28E-01	61	37	31.11
	64	54	59	75	64	64								
	47	21	0	0	0	45								
	100 (41)	50 (4)	0 (0)	0 (0)	0 (0)	95								
PNORM_CUBE:							341	19735	7.59E-04	2.62E-01	7.51E-01	66	55	38.41
	58	53	95	83	74	69								
	38	21	0	0	0	36								
	100 (58)	33 (3)	0 (0)	0 (0)	0 (0)	96								
PNORM_CUTOFF:							340	20671	6.90E-004	1.23E-001	3.84E-001	86	35	38.41
	53	51	86	75	64	62								
	100	100	0	0	0	100								
	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0								
PNORM_FULL:							343	19706	2.89E-04	4.05E-01	1.29E+00	88	63	47.94
	65	63	95	85	72	73								
	43	43	0	0	0	43								
	100 (92)	50 (4)	0 (0)	0 (0)	0 (0)	97								

Analysis results for Pat_1100_1121_6.

Pat_1126_1069_10	S	P	I	E	L	NED Unknowns	NED Knowns
Training sample size:	8	8	8	8	8		
Training (unique galaxies):	8	7	5	1	6		
Full dataset:	1028	7	9	1	6	10	1051
% Morph.:	97.81	0.67	0.86	0.10	0.57		

Observed Frequencies:10	
Frequency Band	Observed Frequencies
Radio	
Millimeter	
SubMillimeter	
FIR	
MIR	
NIR	138000000000000, 182000000000000, 240000000000000
Visual	325000000000000,389000000000000,391000000000000, 477000000000000,479000000000000,617000000000000, 836000000000000
Ultraviolet	
X-ray	
Gamma-ray	

Analysis Technique	S	P	I	E	L	Overall %	Δ BMU	CPU Seconds	QE	TE	TE (dist)	Single Morph.	Multi Morph.	% Map Coverage
ADATA_CUBE:	100 0 100 (979)	100 0 0 (0)	100 0 50 (4)	100 0 0 (0)	100 0 0 (0)	100 0 99	0	1265	1.02E+31	1.00E-01	4.81E-01	29	0	9.21
ADATA_CUTOFF:	100 33 100 (977)	100 0 0 (0)	100 0 0 (4)	100 0 0 (0)	100 0 0 (0)	100 33 99	0	1346	2.32E+20	1.75E-01	4.10E-01	27	0	8.57
ADATA_FULL:	100 35 100 (906)	100 0 0 (0)	100 0 0 (4)	100 0 0 (0)	100 0 0 (0)	100 35 99	0	1265	3.00E+20	1.50E-01	5.37E-01	27	0	8.57
NONE_CUBE:	100 12 100 (932)	100 0 0 (0)	80 0 0 (3)	100 0 0 (0)	83 0 0 (0)	93 12 99	0	1263	8.01E+32	2.00E-01	4.70E-01	25	1	8.25
NONE_CUTOFF:	100 7 100 (963)	100 0 0 (0)	100 0 25 (4)	100 0 0 (0)	100 0 0 (0)	100 7 99	0	1356	2.53E+20	2.75E-01	7.05E-01	27	0	8.57
NONE_FULL:	100 49 100 (892)	100 0 0 (0)	100 0 25 (4)	100 0 0 (0)	100 0 0 (0)	100 49 99	0	1249	7.39E-004	2.75E-001	8.65E-001	25	1	8.25
PNORM_CUBE:	100 14 100 (851)	100 0 0 (0)	100 0 0 (4)	100 0 0 (0)	100 0 0 (0)	100 14 99	0	1296	6.46E-04	5.00E-02	1.39E-01	27	0	8.57
PNORM_CUTOFF:	100 0 0 (0)	100 0 0 (0)	100 0 0 (0)	100 0 0 (0)	100 0 0 (0)	100 0 0	40	1890	7.08E+04	2.50E-02	5.59E-02	27	0	8.57
PNORM_FULL:	100 35 100 (989)	100 0 0 (0)	100 0 0 (4)	100 0 0 (0)	100 0 0 (0)	100 35 99	0	1261	6.62E-04	4.50E-01	1.33E+00	27	0	8.577

Analysis results for Pat_1126_1069_10.

Pat_Visual_42_11641	S	P	I	E	L	NED Unknowns	NED Knowns
Training sample size:	240	240	240	240	240		
Training (unique galaxies):	212	112	67	165	142		
Full dataset:	1267	112	67	319	287	6580	2052
% Morph.:	61.74	5.46	3.27	15.55	13.99		

Observed Frequencies: 6	
Frequency Band	Observed Frequencies
Radio	
Millimeter	
SubMillimeter	
FIR	
MIR	
NIR	
Visual	325000000000000,389000000000000,477000000000000, 617000000000000,641000000000000,836000000000000
Ultraviolet	
X-ray	
Gamma-ray	

Analysis Technique	S	P	I	E	L	Overall %	Δ BMU	CPU Seconds	QE	TE	TE (dist)	Single Morph.	Multi Morph.	% Map Coverage
ADATA_CUBE:							96	62365	1.41E+34	2.34E-01	6.25E-01	96	80	55.87
	51	55	85	50	65	58								
	31	0	0	23	48	32								
	100 (105)	0 (0)	0 (0)	73 (11)	86 (7)	96								
ADATA_CUTOFF:							107	64619	1.03E+34	2.68E-01	8.44E-01	116	118	74.29
	56	68	97	45	80	64								
	23	0	0	24	50	26								
	100 (115)	0 (0)	0 (0)	55 (11)	57 (7)	93								
ADATA_FULL:							225	62819	2.58E+34	1.82E-01	6.74E-01	129	53	57.78
	40	38	93	46	70	52								
	18	0	0	20	40	20								
	98 (127)	0 (0)	0 (0)	95 (20)	92 (13)	97								
NONE_CUBE:							186	64483	1.63E+34	3.08E-01	9.74E-01	112	94	65.40
	54	54	91	43	72	59								
	26	0	0	17	45	27								
	99 (118)	0 (0)	0 (0)	50 (10)	70 (10)	93								
NONE_CUTOFF:							185	63398	3.07E+34	3.46E-01	1.41E+00	86	154	76.19
	59	62	91	48	80	64								
	33	0	0	18	57	34								
	100 (75)	0 (0)	0 (0)	40 (10)	71 (7)	91								
NONE_FULL:							375	63536	1.11E+34	3.29E-01	9.06E-01	114	74	59.68
	53	54	88	47	64	57								
	36	0	0	32	42	36								
	100 (119)	0 (0)	0 (0)	62 (21)	83 (12)	93								
PNORM_CUBE:							294	44030	5.93E-04	2.30E-01	7.08E-01	91	87	56.51
	43	56	94	39	70	55								
	15	0	0	14	37	17								
	99 (82)	0 (0)	0 (0)	40 (10)	83 (6)	91								
PNORM_CUTOFF:							456	62535	6.79E-004	2.83E-001	8.68E-001	89	107	62.22
	51	51	85	47	68	57								
	0	0	0	0	0	0								
	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0								
PNORM_FULL:							428	62562	1.03E-03	3.04E-01	1.35E+00	96	87	58.10
	45	54	96	44	64	55								
	25	0	0	23	34	26								
	99 (111)	0 (0)	0 (0)	50 (12)	33 (9)	90								

Analysis results for Pat_Visual_42_11641.

Pat_1052_1226_12	S	P	I	E	L	NED Unknowns	NED Knowns
Training sample size:	43	43	43	43	43		
Training (unique galaxies):	41	28	16	17	26		
Full dataset:	790	62	16	17	49	290	934
% Morph.:	84.58	6.64	1.71	1.82	5.25		

Observed Frequencies: 12	
Frequency Band	Observed Frequencies
Radio	1400000000
Millimeter	
SubMillimeter	
FIR	3000000000000, 5000000000000
MIR	
NIR	138000000000000, 182000000000000, 240000000000000
Visual	325000000000000,389000000000000,477000000000000, 617000000000000,681000000000000,836000000000000
Ultraviolet	
X-ray	
Gamma-ray	

Analysis Technique	S	P	I	E	L	Overall %	Δ BMU	CPU Seconds	QE	TE	TE (dist)	Single Morph.	Multi Morph.	% Map Coverage
ADATA_CUBE:							0	10031	2.77E+34	2.14E-01	7.32E-01	75	17	29.21
	73	86	94	88	92	84								
	38	25	0	0	25	37								
	100 (281)	10 (10)	0 (0)	0 (0)	27 (11)	94								
ADATA_CUTOFF:							13	10754	5.13E+34	1.49E-01	4.84E-01	56	20	24.13
	76	71	100	76	88	80								
	48	12	0	0	20	46								
	100 (233)	20 (10)	0 (0)	0 (0)	12 (8)	94								
ADATA_FULL:							0	10764	2.53E+34	2.14E-01	5.93E-01	71	13	26.67
	78	79	88	94	81	82								
	39	24	0	0	23	38								
	100 (293)	23 (13)	0 (0)	0 (0)	30 (10)	94								
NONE_CUBE:							1	10332	5.91E+34	2.14E-01	5.42E-01	53	12	20.63
	51	61	100	71	65	65								
	21	15	0	0	12	20								
	100 (184)	0 (8)	0 (0)	0 (0)	0 (7)	92								
NONE_CUTOFF:							14	11564	6.08E+34	2.05E-01	5.70E-01	54	17	22.54
	63	71	94	76	73	73								
	34	23	0	0	29	33								
	100 (244)	12 (8)	0 (0)	0 (0)	22 (9)	94								
NONE_FULL:							4	10944	2.32E+34	2.09E-01	6.46E-01	65	19	26.67
	73	75	88	88	92	81								
	31	25	0	0	29	31								
	100 (296)	10 (10)	0 (0)	0 (0)	0 (9)	94								
PNORM_CUBE:							9	10243	6.26E-03	2.33E-01	7.10E-01	72	12	26.67
	68	86	100	82	73	79								
	0	0	0	0	0	0								
	100 (747)	0 (34)	0 (0)	0 (0)	0 (23)	92								
PNORM_CUTOFF:							2	10638	9.46E-03	1.44E-01	4.53E-01	60	9	21.90
	59	75	88	76	65	70								
	0	0	0	0	0	0								
	100 (1)	0 (0)	0 (0)	0 (0)	0 (0)	100								
PNORM_FULL:							6	10557	1.88E-02	1.44E-01	3.96E-01	57	15	22.86
	68	64	88	82	88	76								
	0	0	0	0	0	0								
	100 (2)	0 (0)	0 (0)	0 (0)	0 (0)	100								

Analysis results for Pat_1052_1226_12.

Pat_139_8938_6	S	P	I	E	L	NED Unknowns	NED Knowns
Training sample size:	331	331	331	331	331		
Training (unique galaxies):	315	256	123	183	278		
Full dataset:	5076	530	123	183	278	2512	6190
% Morph.:	82.00	8.56	1.99	2.96	4.49		

Observed Frequencies: 6	
Frequency Band	Observed Frequencies
Radio	1400000000
Millimeter	
SubMillimeter	
FIR	3000000000000, 5000000000000
MIR	
NIR	138000000000000, 182000000000000, 240000000000000
Visual	
Ultraviolet	
X-ray	
Gamma-ray	

Analysis Technique	S	P	I	E	L	Overall %	Δ BMU	CPU Seconds	QE	TE	TE (dist)	Single Morph.	Multi Morph.	% Map Coverage
ADATA_CUBE:							187	92377	5.07E+34	2.69E-01	8.22E-01	70	154	71.11
	47	39	74	58	50	51								
	34	21	0	0	0	33								
	100 (132)	38 (16)	0 (0)	0 (0)	0 (0)	93								
ADATA_CUTOFF:							35	91933	1.06E+35	1.90E-01	5.54E-01	74	78	48.25
	45	34	72	52	35	44								
	30	16	0	0	0	29								
	100 (60)	50 (8)	0 (0)	0 (0)	0 (0)	94								
ADATA_FULL:							785	89887	1.43E+35	2.50E-01	7.31E-01	73	100	54.92
	46	39	72	54	39	47								
	32	21	0	0	0	31								
	100 (154)	41 (17)	0 (0)	0 (0)	0 (0)	94								
NONE_CUBE:							204	91402	7.14E+34	2.63E-01	9.20E-01	85	117	64.13
	53	32	76	57	37	48								
	30	15	0	0	0	29								
	99 (172)	58 (19)	0 (0)	0 (0)	0 (0)	95								
NONE_CUTOFF:							13	90685	6.83E+34	2.29E-01	7.87E-01	65	126	60.63
	51	39	79	59	33	48								
	40	18	0	0	0	39								
	99 (86)	45 (11)	0 (0)	0 (0)	0 (0)	92								
NONE_FULL:							774	92395	1.10E+35	2.94E-01	8.79E-01	66	104	53.97
	59	37	71	56	37	50								
	41	16	0	0	0	40								
	100 (107)	50 (12)	0 (0)	0 (0)	0 (0)	94								
PNORM_CUBE:							922	53585	1.76E-03	4.73E-01	1.74E+00	87	130	68.89
	43	39	71	49	46	47								
	22	8	0	0	0	22								
	100 (148)	11 (18)	0 (0)	0 (0)	0 (0)	90								
PNORM_CUTOFF:							893	89747	5.15E-04	5.27E-01	2.45E+00	67	162	72.70
	39	36	76	61	40	46								
	0	0	0	0	0	0								
	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0								
PNORM_FULL:							750	90076	6.45E-04	3.69E-01	1.39E+00	60	158	69.21
	53	39	80	54	42	50								
	30	11	0	0	0	29								
	100 (69)	20 (10)	0 (0)	0 (0)	0 (0)	89								

Analysis results for Pat_139_8938_6.

Pat_192_6934_5	S	P	I	E	L	NED Unknowns	NED Knowns
Training sample size:	138	138	138	138	138		
Training (unique galaxies):	131	122	35	90	93		
Full dataset:	1489	122	35	150	141	1944	1937
% Morph.:	76.87	6.30	1.81	7.74	7.28		

Observed Frequencies: 5	
Frequency Band	Observed Frequencies
Radio	1400000000
Millimeter	
SubMillimeter	
FIR	
MIR	
NIR	
Visual	325000000000000,477000000000000,617000000000000, 836000000000000
Ultraviolet	
X-ray	
Gamma-ray	

Analysis Technique	S	P	I	E	L	Overall %	Δ BMU	CPU Seconds	QE	TE	TE (dist)	Single Morph.	Multi Morph.	% Map Coverage
ADATA_CUBE:							41	34311	3.93E+34	3.22E-01	9.45E-01	102	72	55.24
	73	52	94	83	58	68								
	36	0	0	41	35	36								
	100 (352)	0 (0)	0 (0)	33 (9)	0 (2)	97								
ADATA_CUTOFF:							65	34064	1.57E+34	4.30E-01	1.77E+00	93	110	64.44
	69	59	100	83	74	72								
	34	0	0	49	44	35								
	100 (270)	0 (0)	0 (0)	20 (5)	60 (5)	97								
ADATA_FULL:							227	34224	1.66E+34	3.17E-01	1.08E+00	83	89	54.60
	61	47	97	80	63	64								
	33	0	0	52	26	33								
	100 (151)	0 (0)	0 (0)	50 (10)	33 (6)	94								
NONE_CUBE:							81	34612	1.91E+34	3.84E-01	1.31E+00	104	86	60.32
	78	61	91	74	58	70								
	37	0	0	33	26	37								
	100 (296)	0 (0)	0 (0)	17 (6)	67 (6)	97								
NONE_CUTOFF:							90	35007	3.22E+34	3.61E-01	1.48E+00	91	94	58.73
	77	54	91	82	59	70								
	38	0	0	37	34	38								
	100 (233)	0 (0)	0 (0)	33 (9)	0 (1)	97								
NONE_FULL:							211	33376	4.53E+33	4.35E-01	1.39E+00	125	70	61.90
	76	61	80	77	62	70								
	33	0	0	36	26	33								
	100 (173)	0 (0)	0 (0)	67 (18)	70 (10)	95								
PNORM_CUBE:							257	34768	1.18E-03	2.62E-01	8.98E-01	108	50	50.16
	67	52	80	76	60	64								
	0	0	0	0	100	3								
	100 (9)	0 (0)	0 (0)	0 (0)	0 (0)	100								
PNORM_CUTOFF:							219	34469	1.05E-03	3.26E-01	9.37E-01	83	72	49.21
	73	45	86	71	57	63								
	99	0	0	0	0	92								
	100 (13)	0 (0)	0 (0)	0 (1)	0 (0)	92								
PNORM_FULL:							192	34521	7.40E-04	3.93E-01	1.65E+00	100	79	56.83
	67	42	80	73	66	62								
	98	0	0	0	0	91								
	100 (5)	0 (0)	0 (0)	100 (1)	0 (0)	100								

Analysis results for Pat_192_6934_5.

Pat_Visual_49_8831	S	P	I	E	L	NED Unknowns	NED Knowns
Training sample size:	411	411	411	411	411		
Training (unique galaxies):	387	223	116	272	299		
Full dataset:	3741	223	116	487	522	3593	5089
% Morph.:	73.51	4.38	2.28	9.57	10.26		

Observed Frequencies: 5	
Frequency Band	Observed Frequencies
Radio	
Millimeter	
SubMillimeter	
FIR	
MIR	
NIR	
Visual	325000000000000,389000000000000,477000000000000, 681000000000000,836000000000000
Ultraviolet	
X-ray	
Gamma-ray	

Analysis Technique	S	P	I	E	L	Overall %	Δ BMU	CPU Seconds	QE	TE	TE (dist)	Single Morph.	Multi Morph.	% Map Coverage
ADATA_CUBE:							377	116060	2.69E+34	3.76E-01	1.31E+00	82	175	81.59
	51	53	85	43	62	55								
	35	0	0	22	39	34								
	100 (43)	0 (0)	0 (0)	80 (5)	62 (8)	92								
ADATA_CUTOFF:							521	116250	3.36E+34	2.73E-01	8.70E-01	110	129	75.87
	49	39	81	38	68	52								
	31	0	0	22	45	32								
	99 (108)	0 (0)	0 (0)	56 (16)	75 (12)	91								
ADATA_FULL:							1404	117414	5.68E+34	4.17E-01	1.52E+00	96	131	72.06
	51	35	80	48	60	52								
	35	0	0	23	34	34								
	100 (123)	0 (0)	0 (0)	62 (13)	42 (12)	91								
NONE_CUBE:							375	114869	3.41E+34	3.97E-01	1.38E+00	77	156	73.97
	51	49	84	43	58	53								
	38	0	0	29	42	38								
	94 (64)	0 (0)	0 (0)	79 (14)	56 (16)	85								
NONE_CUTOFF:							500	115925	4.85E+34	4.66E-01	1.69E+00	66	200	84.44
	53	57	71	53	66	58								
	37	0	0	42	49	38								
	100 (57)	0 (0)	0 (0)	0 (1)	50 (2)	96								
NONE_FULL:							680	114013	5.32E+34	2.71E-01	8.57E-01	116	140	81.27
	49	36	89	44	55	51								
	29	0	0	27	30	29								
	99 (121)	0 (0)	0 (0)	62 (13)	29 (17)	88								
PNORM_CUBE:							1030	42522	1.33E-03	4.51E-01	1.64E+00	60	214	86.98
	57	55	73	55	56	58								
	38	0	0	29	26	37								
	100 (281)	0 (0)	0 (0)	1 (76)	5 (21)	74								
PNORM_CUTOFF:							789	113863	1.42E-003	3.07E-001	1.20E+000	91	118	66.35
	48	30	84	38	51	47								
	0	0	0	0	0	0								
	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0								
PNORM_FULL:							844	119687	2.22E-03	2.65E-01	9.93E-01	59	88	46.67
	46	40	78	38	52	48								
	14	0	0	20	54	17								
	100 (187)	0 (0)	0 (0)	4 (77)	6 (16)	68								

Analysis results for Pat_Visual_49_8831.

Appendix D

Predictions

For the dataset processed in this thesis, multiple approaches were taken to create an effective morphological classifier for galaxies.

The present appendix combines the results obtained in Appendix C. The technique uses an *Ensemble* method to combine the morphology predictions (for each unknown galaxy) of each separate model. Galaxies that are present in multiple models receive a morphology *vote* for each predicted morphology. Using a majority vote process, the various models are combined to forecast a combined morphological prediction.

Since the original snapshot date of the database used in this work, NED has updated morphologies for a significant portion of the galaxies for which we did not know the morphology. Additional downloads were performed to update the database for these previously unclassified galaxies in order that we may compare our model's predictions.

The following tables show results for each of the major classes used in this study: S, P, I, E and L. Results for the spiral and elliptical galaxies are the most promising. This is not surprising since these two classes represent the bulk of the data present in the study. These two classes also make up the majority of the original Hubble diagram. Morphology predictions for the other threes classes were not as successful.

Galaxy Name	NED Morph.	# of Votes	S	P	I	E	L	Pred. Morph.
MCG +10-16-020	Sc	10	80.00	10.00	0.00	0.00	10.00	S
SBS 1511+515C	Sc	10	80.00	10.00	0.00	0.00	10.00	S
CGCG 265-026 NED01	Sbc	10	80.00	10.00	0.00	0.00	10.00	S
NGC 3349	Sb	10	80.00	10.00	0.00	0.00	10.00	S
UGC 05941 NED02	Sb	10	80.00	10.00	0.00	0.00	10.00	S
2MASX J08433610+3431131	Sa	10	80.00	10.00	0.00	0.00	10.00	S
UGC 09315 NOTES01	Sa	10	80.00	10.00	0.00	0.00	10.00	S
KUG 0855+462	Sb	11	73.00	18.00	0.00	0.00	9.00	S
SBS 1320+551	S0^	11	73.00	18.00	0.00	0.00	9.00	S
FBQS J2327-1023	Sb	10	70.00	10.00	0.00	0.00	20.00	S
IC 4611	Sa	10	70.00	10.00	0.00	0.00	20.00	S
CGCG 039-202 NED02	Sbc	10	70.00	10.00	10.00	0.00	10.00	S
LCSB S2204P	Sb	10	70.00	10.00	10.00	0.00	10.00	S
2MASX J15311118+4657560	Sab	10	70.00	10.00	10.00	0.00	10.00	S
MCG +08-24-021	Sc	10	70.00	20.00	0.00	0.00	10.00	S
UGC 00793 NED01	Sb	10	70.00	20.00	0.00	0.00	10.00	S
2MASX J12055599+4959561	S0/a	10	70.00	20.00	0.00	0.00	10.00	S
2MASX J20080914-5630214	Sb	9	67.00	0.00	0.00	0.00	33.00	S
2MASX J21483574-5732124	Sb	9	67.00	0.00	0.00	0.00	33.00	S
UGC 11673 NED02	Sc	12	67.00	17.00	0.00	0.00	17.00	S
MCG -02-58-023	Sbc	12	67.00	17.00	0.00	0.00	17.00	S
MCG -02-05-064	Sbc	20	65.00	15.00	5.00	5.00	10.00	S
2MASX J15474417+4124089	S0/a	11	64.00	9.00	0.00	9.00	18.00	S
CGCG 242-039	Sbc	19	63.00	26.00	5.00	0.00	5.00	S
CGCG 383-012	compact	61	61.00	8.00	8.00	10.00	13.00	S
2MASX J05331977-2830433	Sb	10	60.00	0.00	0.00	10.00	30.00	S
2MASX J05243408-3148577	S	10	60.00	0.00	0.00	10.00	30.00	S
CGCG 186-049	compact	20	60.00	10.00	5.00	10.00	15.00	S
2MASX J04462502-2020295	Sb	10	60.00	0.00	0.00	0.00	40.00	S
2MASX J20083036-5639163	Sa	10	60.00	0.00	0.00	0.00	40.00	S
2MASX J10081910+3729039	Sb	10	60.00	10.00	0.00	0.00	30.00	S
CGCG 036-026 NED01	Sc	10	60.00	20.00	0.00	0.00	20.00	S
KUG 0951+369	Sbc	20	60.00	20.00	5.00	0.00	15.00	S
UM 207	Sc	10	60.00	10.00	20.00	0.00	10.00	S

Table D.1: Ensemble method morphology predictions: Spirals.

Galaxy Name	**NED Morph.**	**# of Votes**	**S**	**P**	**I**	**E**	**L**	**Pred. Morph.**
CGCG 138-054	Sb	20	25.00	55.00	15.00	0.00	5.00	P
2MASX J15505152+4202325	S0/a	10	30.00	50.00	0.00	10.00	10.00	P
VIII Zw 100	compact	10	30.00	50.00	10.00	0.00	10.00	P
CGCG 266-044 NED02	Sc	12	25.00	50.00	17.00	0.00	8.00	P
CGCG 207-043	Sbc	25	16.00	48.00	4.00	8.00	24.00	P
CGCG 387-054	Sb	30	40.00	47.00	7.00	3.00	3.00	P
CGCG 039-113	S0 pec	24	17.00	46.00	4.00	4.00	29.00	P
UGCA 372	Wolf-Rayet Galaxy	13	8.00	46.00	8.00	31.00	8.00	P
2MASX J21305810-0705079	Sc	22	9.00	45.00	14.00	14.00	18.00	P
NGC 0617	Sb	33	33.00	45.00	6.00	3.00	12.00	P
MCG +08-23-080	Sc	11	27.00	45.00	9.00	9.00	9.00	P
CGCG 039-163	Sb	20	20.00	45.00	25.00	5.00	5.00	P
CGCG 067-009	Sc	21	19.00	43.00	10.00	24.00	5.00	P

Table D.2: Ensemble method morphology predictions: Peculiars.

Galaxy Name	**NED Morph.**	**# of Votes**	**S**	**P**	**I**	**E**	**L**	**Pred. Morph.**
ESO 233-IG 044	Pec	18	6.00	6.00	83.00	0.00	6.00	I
ESO 379-IG 035	pair	38	5.00	8.00	79.00	0.00	8.00	I
ESO 554-IG 027	B bar	38	3.00	18.00	71.00	0.00	8.00	I
UGCA 280	very com-pact	58	10.00	7.00	67.00	7.00	9.00	I
SBS 1401+490	Im	9	0.00	0.00	67.00	22.00	11.00	I
UGC 06074	S?	51	10.00	14.00	63.00	6.00	8.00	I
IC 1235	Sc	60	17.00	5.00	63.00	5.00	10.00	I
NGC 3353	Sb? pec	68	15.00	9.00	60.00	1.00	15.00	I
CGCG 275-003	Scd	49	14.00	10.00	59.00	2.00	14.00	I
CGCG 247-005	Sc	49	14.00	10.00	57.00	2.00	16.00	I
2MASX J17025023+3642246	Sc	21	19.00	10.00	57.00	10.00	5.00	I
NGC 6275	Sdm	62	18.00	8.00	56.00	11.00	6.00	I
ESO 079- G 005	SB(rs)d pec?	29	14.00	3.00	55.00	3.00	24.00	I

Table D.3: Ensemble method morphology predictions: Irregulars.

Galaxy Name	NED Morph	# of Votes	S	P	I	E	L	Pred. Morph.
2MASX J01312761+0109470	E	10	20.00	10.00	0.00	60.00	10.00	E
2MASX J07591742+4200308	E	10	20.00	10.00	0.00	60.00	10.00	E
2MASX J08004242+4001385	E	10	20.00	10.00	0.00	60.00	10.00	E
2MASX J08110429+3559085	E	10	20.00	10.00	0.00	60.00	10.00	E
2MASX J08243796+3715241	E	10	20.00	10.00	0.00	60.00	10.00	E
2MASX J08553525+4027535	E	10	20.00	10.00	0.00	60.00	10.00	E
2MASX J09031116+5403515	E	10	20.00	10.00	0.00	60.00	10.00	E
2MASX J09305872+0348281	E	10	20.00	10.00	0.00	60.00	10.00	E
2MASX J12530320+4500444	E	10	20.00	10.00	0.00	60.00	10.00	E
2MASX J13253392+1304122	E	10	20.00	10.00	0.00	60.00	10.00	E
2MASX J13490723+0504127	E	10	20.00	10.00	0.00	60.00	10.00	E
2MASX J13544229+0528557	E	10	20.00	10.00	0.00	60.00	10.00	E
2MASX J15520363+2751595	E	10	20.00	10.00	0.00	60.00	10.00	E
2MASX J17045631+2100219	E	10	20.00	10.00	0.00	60.00	10.00	E
B3 1026+386	E	10	20.00	10.00	0.00	60.00	10.00	E
B3 1052+456	E	10	20.00	10.00	0.00	60.00	10.00	E
MCG +06-35-001	E	10	20.00	10.00	0.00	60.00	10.00	E
MCG +09-19-173	E	10	20.00	10.00	0.00	60.00	10.00	E
MCG +09-20-073	E	10	20.00	10.00	0.00	60.00	10.00	E
NGC 5776	E	19	11.00	11.00	5.00	58.00	16.00	E
MCG -01-27-002	E	9	0.00	0.00	0.00	56.00	44.00	E
2MASX J01154523+0014445	S	9	0.00	11.00	0.00	56.00	33.00	E
2MASX J07572674+2328306	S0^	11	18.00	9.00	0.00	55.00	18.00	E
2MASX J07563043+4102099	E	11	18.00	9.00	0.00	55.00	18.00	E
2MASX J14243629+0244423	E	11	18.00	9.00	0.00	55.00	18.00	E
2MASX J14281992+4559098	E	11	18.00	9.00	0.00	55.00	18.00	E

Table D.4: Ensemble method morphology predictions: Ellipticals.

Galaxy Name	NED Morph	# of Votes	S	P	I	E	L	Pred. Morph.
2MASX J23181204-1246176	Sb	9	0.00	0.00	0.00	0.00	100.00	L
2MASX J03122491-2716295	Sab	9	0.00	0.00	0.00	0.00	100.00	L
2MASX J21480355-5720413	S	9	0.00	0.00	0.00	0.00	100.00	L
2MASX J21464982-5701361	Sc	10	10.00	0.00	0.00	0.00	90.00	L
2MASX J21394244-1412163	S	9	0.00	0.00	0.00	11.00	89.00	L
2MASX J23174294-1242341	Sb	9	11.00	0.00	0.00	0.00	89.00	L
2MASX J03104250-2641313	Sa	9	11.00	0.00	0.00	0.00	89.00	L
2MASX J21385510-1416411	S	9	11.00	0.00	0.00	0.00	89.00	L
2MASX J05261419-3153059	Sab	11	18.00	0.00	0.00	0.00	82.00	L
2MASX J05233086-3133372	S	11	18.00	0.00	0.00	0.00	82.00	L
2MASX J03103728-2655054	Sb	10	20.00	0.00	0.00	0.00	80.00	L
2MASX J21471111-5718401	Sa	10	20.00	0.00	0.00	0.00	80.00	L
2MASX J01150233+0015470	S0	18	0.00	0.00	6.00	17.00	78.00	L
2MASX J02063361-0323483	Sc	9	0.00	11.00	0.00	11.00	78.00	L
2MASX J04480634-2033514	Sb	9	22.00	0.00	0.00	0.00	78.00	L
2MASX J22205029-0305055	Sb	9	22.00	0.00	0.00	0.00	78.00	L

Table D.5: Ensemble method morphology predictions: Lenticulars.