Using Competitive Learning to Handle Missing Values in Astrophysical Datasets

R.A. Browse, D.B. Skillicorn, S.M. McConnell
Technical Report 2002-458
Department of Computing and Information Science
Queen's University, Kingston
{browse, skill, mcconnell}@cs.queensu.ca

Abstract

We address the problem of classification of galaxies, using unsupervised clustering. This problem is made difficult by the large fraction of attribute values for galaxies that are missing from repositories. We enhance a Kohonen neural network so that it can learn from data with missing values and show that the resulting clusterings are subjectively better, have smaller dispersion, and can be learned more quickly than Bayesian approaches based on mixture models such as AutoClass. We also show that the enhanced Kohonen neural network is a better solution than replacing missing values by mean substitution.

Keywords: Kohonen competitive learning, neural networks, missing values, EM, AutoClass, galaxy classification, Hubble sequence.

1 Introduction

The application of data mining techniques to astrophysical data sets is still in its early stages. While most approaches have shown promising results, the range of techniques used to date is limited. Data mining applications in astrophysics must almost always use *unsupervised* techniques for there is no source for the 'right' answers other than the data itself. The exception is where predictors can be learned on high-quality data (for example, from close objects), or from data classified by humans. The latter approach is severely limited by the cost of using skilled human classifiers and experience has shown that human classifiers are prone to disagree.

One successful approach is the use of a supervised neural network for the separation of stars, galaxies and background noise in data obtained from the Minnesota Plate Scanner [Ode95]. Supervised neural networks have also been extensively used in astronomical applications such as forecasting [Wu97], morphological classification of galaxies [LNSS96] and spectral classifications of stars [SGG98].

In this paper, we address the unsupervised clustering of galaxies, concentrating on the problem of missing values, where some, perhaps most of the attributes of the galaxies being clustered are not available. The techniques described in this paper also apply to other domains. Astrophysics is unique in that data collected over millenia, in many different formats, are in use (for example to study long-period comets).

We introduce a variant of an unsupervised neural network which is able to build effective models even when attribute values are missing. The quality of the resultant clustering degrades slowly as the number of missing values increases. We compare the performance of our enhanced neural network with other approaches to handling missing values: the expectation maximization (EM) used by AutoClass, and Mean Substitution. Our technique is faster, and produces clusters of better quality. The ability to build models in the presence of missing values makes it possible to use available astrophysical datasets much more effectively.

In the following section we introduce some of the problems of working with astrophysical data, concentrating on the problem of missing values. In Section 3, we consider three kinds of solutions to missing value problems, and report comparative results for clustering using standard techniques and the enhanced neural network technique. In Section 4, we show that the neural network technique is stable with respect to the pattern of missing values. Section 5 contains some discussion of the properties of this approach.

2 Problems in Astrophysical Datasets

Determining the value of even a single attribute of a large number of celestial objects can require weeks of work, even though the data required is available in online repositories. There are three issues that make information gathering so difficult:

1. *Issues arising from distributed collection and ownership of data.* Data is spread out over physically and logically distributed datasets, because surveys of different subsets of objects were performed for different reasons, using different instruments, and over a large number of years. Many sites have to be queried to collect all available data for an object of interest. Existing tools such as VizieR, for example, aim at easing this task by allowing simultaneous queries of a number of catalogs. Additionally, larger datasets such as the NASA/IPAC extragalactic dataset (NED), are being compiled from smaller catalogs, but no single interface will give all available data for an arbitrary object.

2. *Issues arising from the measurements themselves.* Different naming conventions are used in different datasets, possibly resulting in duplicate entries for the same object. Duplicate entries were found for example in [HML94]. Large error bounds can be associated with the data and it is not easy to take these into account; even bounding them can be difficult because they arise from uncontrollable factors, such as the instrument used for the observation, or weather conditions.

3. *Missing values.* Attributes can be missing because they were not captured due to circumstances on the ground, unrelated to any characteristic of the object. Attributes can also be missing because they are hard to capture for some reason. The number of known objects is large (NED has data for over 4.3 million extragalactic objects) and so is the number of possible measurements for each object (the Lyon-Meuden extragalactic database, LEDA, returns up to 86 fields for queried objects).

The problem of missing values is not necessarily the hardest problem associated with astrophysical datasets, but until it is solved, it is difficult to get enough consistent data to address the more difficult problems.

Various mechanisms that produce missing values can be distinguished. Depending on the underlying pattern, a range of techniques can be utilized to deal with incomplete data samples. The following mechanisms for producing missing data were defined in [LR87]:

- **MCAR (Missing Completely At Random):** In this case, the probabilities that the values of the inputs are missing attributes are independent of the values of any of the variables.

- **MAR (Missing At Random):** This is a weaker version of MCAR, in which the cases with missing data differ from complete cases in the values of attributes that are present for both.
- **Non-Ignorable or NMAR (Not Missing At Random):** The missing data is caused by the values of the data itself.

The patterns of missing values in astrophysical datasets are complex and, even though newer datasets often specify why data are incomplete, it would be an impossible task to find out why a certain value is missing in older repositories. All three mechanisms are common in astrophysical datasets. Attributes are missing completely at random when they fail to be captured because of a mechanical failure of an instrument, or transient weather conditions. Attributes are missing at random when the objects surveyed are chosen for properties associated with their other attributes. An obvious examples is when objects in only one hemisphere are studied. Attributes are missing not at random when the attribute's values themselves are the explanation for their failure to be captured. An example is when the attribute value falls below the detection threshold of the observing device. Another example of a non-ignorable missing data mechanism is the *Malmquist Bias* – the underrepresentation of intrinsically faint objects at larger distances in flux-limited catalogs.

There are two main approaches to dealing with missing data: deletion of objects with missing attributes from the dataset altogether, or replacement of the missing attributes by values computed from the rest of the dataset, and possibly external knowledge. Deleting objects with missing attributes is simple, but it introduces bias unless the missing data is missing completely at random. It is also extremely limiting in practice, since many astrophysical datasets are sufficiently sparse that only a tiny remnant of the set of objects would remain. An example of casewise deletion for sample selection can be found in [GTI99], where, in a dataset containing 1279 white dwarfs, complete data for only 232 objects could be extracted.

We now introduce a dataset that we will use as a running example in the paper.

Example: Data for 4161 galaxies with radial velocities between 500 and 1500 km/s for which the Hubble Type is known were extracted from LEDA, the Lyon-Meuden extragalactic database. This roughly encompasses the galaxies contained within the Local Supercluster. The most frequent four attributes (the two colors U-B and B-V, the absolute B magnitude, and the 21 cm index) were then used to construct a complete subsample, containing 863 galaxies of the main Hubble types (Elliptical, Spiral and Irregular). Figure 1 shows the frequency of each attribute in the sample as a function of its value. The different distributions of these attributes with Hubble type suggest that they can all provide some information to discriminate among types.

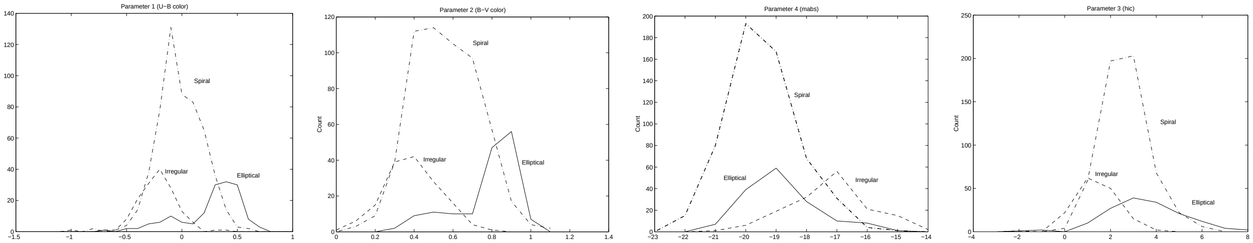

Figure 1: The frequencies of attributes (U-B color, B-V color, Absolute B magnitude, and 21 cm index respectively) as a function of their values, for each Hubble type

This dataset will act as a 'complete' baseline dataset that we will use to compare the performance of different clustering techniques. We will also use it to simulate datasets with varying numbers of missing values by uniformly deleting some of their elements. We note in passing that more than 75%

Cluster	0	1	2	3	4	5	6	7	8	9	10
Elliptical	9	10	5	11	4	47	56	1	7	0	2
Irregular	14	44	5	1	63	0	0	20	2	1	2
Spiral	150	77	97	98	33	57	10	9	14	14	0

Table 1: AutoClass clustering on complete data

of the originally available data had to be discarded even to produce a complete dataset with only four attributes.

Measuring the quality of a clustering is difficult. We will assess different clustering techniques first by comparing the clusters they produce with the Hubble types (elliptical, irregular, and spiral) and second by measuring the compactness of the clusters produced using dispersion,

$$\text{dispersion} = \left(\sum_{\text{clusters } i} \sum_{\text{objects } j} (x_{ij} - c_i)^2 \right)^{1/2}$$

where x_{ij} is the jth object in cluster i, and c_i is the center of cluster i. Dispersion depends on the number of clusters. When there are more cluster centers, it is more likely that an object will be close to one of them.

3 Approaches to Handling Missing Values

We consider three kinds of solutions to the problem of missing values: discarding objects for which the attribute measurements are incomplete, replacing missing attributes by simple functions of those attribute values for other objects, and building models that treat missing values in more sophisticated ways.

3.1 Approach 1: Use only complete records.

Building models using only those objects for which a complete set of attribute values is known dramatically limits the number of objects that can be used. If the attribute values are not missing completely at random, using only complete records inevitably introduces bias into the resulting models. This approach, therefore, cannot be completely satisfactory. We will demonstrate the results of several techniques on our example dataset as a prelude to understanding the results of techniques that handle missing values.

There are two broad approaches to unsupervised clustering: parametric and non-parametric. The most common parametric technique is to assume that the underlying distribution of data is a mixture of simpler distributions, perhaps Gaussians. The EM algorithm is an effective (maximal likelihood) way to compute both the parameters of these simpler distributions and an allocation of objects to clusters. Note that EM naturally handles missing values – the cluster labels can be regarded as missing attributes of the objects. AutoClass [CKS$^+$88] is a widely-available implementation of the EM algorithm.

Example: Table 1 shows the clustering produced when AutoClass is run on the example dataset. AutoClass was told to cluster the data 50 times and then selects the clustering that best fits the data. Each clustering may contain a different number of clusters.

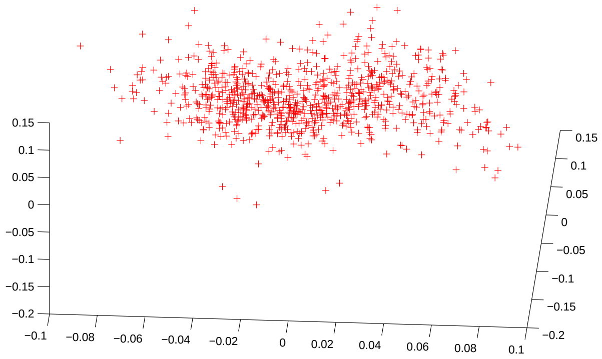

Figure 2: SVD plot of example dataset

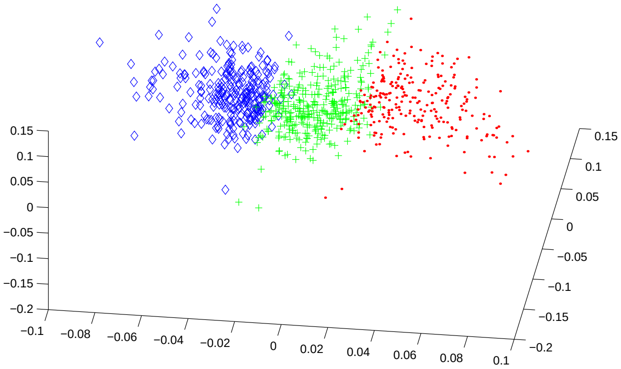

Figure 3: SDD plot of example dataset

Several clusters in Table 1 are relatively homogeneous, but there are many that contain large numbers of galaxies of different Hubble types. The dispersion for this clustering is 957.

We now consider some non-parametric clustering techniques. *Singular value decomposition* (SVD) [BDO95] regards each object as a point in a space indexed by the attributes. If there are n objects and m attributes, then each object is a point in m-dimensional space. SVD transforms the original space into a new m-dimensional space in which the variation along the first axis is maximized, then the variation along the second axis, and so on. Each object is a point in this new space; however, ignoring many of the later dimensions gives a representation of the data in a low-dimensional space that is as faithful as possible to original data. High-dimensional data can be displayed in 2 or 3 dimensions where visual inspection can often reveal its properties.

Example: Figure 2 shows a plot of the example dataset in 3 dimensions. Information from the singular values indicates that almost all of the variation is present along the horizontal axis. This axis corresponds roughly to the progression from elliptical to irregular galaxies, with the spiral galaxies in the center. No clear clusters are visible in this plot.

Semi-discrete decomposition (SDD) [MS01] is an unsupervised bump hunting technique that is related to SVD. It constructs a ternary decision tree for objects, placing each region of the dataset that stands out from the rest into a separate branch. Those branches closest to the root are the most distinct, so SDD is especially good at outlier detection.

Example: Figure 3 shows the top level decision tree classification of the example dataset, superimposed on the positions derived from the SVD above. The SDD has partitioned what SVD considers to be one big cluster into three subclusters. Table 2 shows the allocation of Hubble types to these clusters. Cluster 0 contains almost all of the Irregulars; cluster 2 contains almost all of the Ellipticals; while the spiral galaxies are spread across all three classes, with a concentration in the middle. There is substantial agreement between this clustering and the Hubble sequence. The dispersion of this clustering is 1233.

We now turn to the unsupervised neural network technique that is the focus of this paper. We use an unsupervised neural network with Kohonen competitive learning. The architecture of such a network is shown in Figure 4. The set of attributes for each object is fed to the inputs of the network. The layers of the network are fully forward connected. Each of the outputs of the network signals the classification of an object into the cluster to which it corresponds. If a partitioning of objects into two clusters is desired, then a network with two outputs is used; if three clusters are

Cluster	0	1	2
Ellipticals	20	26	106
Irregulars	136	16	0
Spirals	106	305	148

Table 2: SDD clustering on the complete dataset

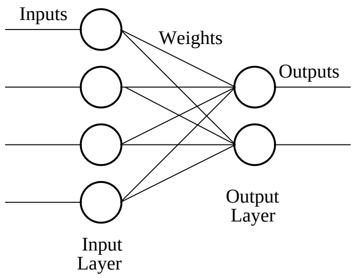

Figure 4: Kohonen network

desired, three outputs are used, and so on. The network clusters objects such that the similarity between objects allocated to the same cluster is maximized.

The objects are fed to the network one at a time, one attribute per input. For each output node, the Euclidean distances of the weights and the input attributes are summed over all input nodes to produce a final value in each of the output nodes. The output node containing the smallest value is selected as the *Winner*. Its weights are situated closest in m-space to the input pattern. In a Kohonen network, the output value in the winning node is set to 1 and values in all other output nodes are forced to 0 (winner take all), resulting in an allocation of the given object to the cluster represented by the winning node only. (Alternatively, the values in the output nodes could be left unmodified and interpreted, if normalized, as the probability that the object should be allocated to each of the clusters.)

In a second step for each object, the weights leading into the winning node are adjusted to bring them even closer to the input pattern. Objects are fed to the network during multiple passes through the dataset until cluster membership remains stable. (Alternatively, the computation could be terminated when the dispersion falls below a predetermined threshold.)

Example: The clustering produced for the example dataset using the Kohonen neural network and three clusters is shown in Table 3. This clustering is very similar to that produced by SDD, but each cluster is slightly more homogeneous. The average dispersion of this clustering is 1603. (Results are similar for two and four clusters.)

All of the techniques discussed so far produce global models of the dataset. It is also possible to use rule-based techniques, which produce local models of regions of the dataset. For example, a

Cluster	0	1	2
Ellipticals	19	19	114
Irregulars	134	16	2
Spirals	88	327	144

Table 3: Kohonen neural network clustering for the complete dataset

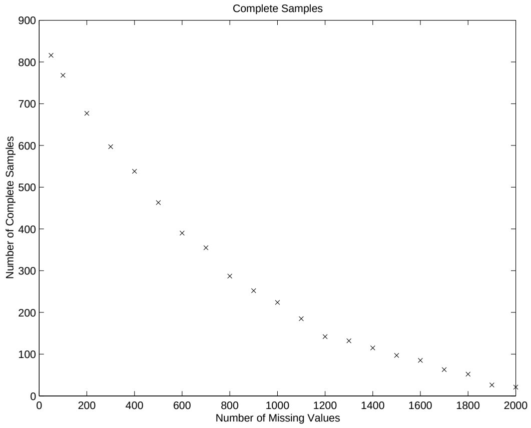

Figure 5: Number of complete objects as a function of incidence of missing values

rule is typically of the form

$$\text{conjunction of inequalities on the attributes} \longrightarrow \text{Hubble type}$$

with some associated probability. In some settings, such rules can be useful. However, such probabilities are large when there are regions in the dataset with unusual density – they are useful in pointing out localized anomalies in data. The SVD plot suggests that such regions do not exist in this dataset.

In summary, AutoClass, semidiscrete decomposition, and the Kohonen neural network technique all produce reasonable clusterings, although AutoClass prefers a large number of clusters. AutoClass has complexity $\mathcal{O}(nmk)$ where k is the number of clusters; SVD has complexity $\mathcal{O}(nm^2)$, and the Kohonen network has complexity $\mathcal{O}(nmk)$. However, the constant factor for AutoClass is much larger than for the Kohonen network – AutoClass takes minutes when the Kohonen network takes seconds. Although SVD has been successfully used for clustering in other contexts, it is not able to find any clusters on this data.

The main limitation of using only objects with complete sets of attributes is that it imposes a limit on the number of objects that can be considered. Figure 5 shows how, in our example dataset, the number of objects with a complete set of attributes in this dataset decreases as the incidence of missing values increases. Only about 20 complete sets of attributes remain when there are 2000 missing values. We now turn our attention to techniques that are able to handle missing attribute values.

3.2 Approach 2: Replace missing values by substitutes

A standard technique for handling missing values is to replace them with values computed in some other way, usually by considering all of the values for that attribute present in the dataset. Here we illustrate what happens when each missing attribute value is replaced by the mean value of that attribute over all objects for which it is present, and then the Kohonen neural network is used on this 'complete' data to create a clustering. The dispersion resulting from the clustering was calculated based on the original data values.

Example: Attribute values were randomly removed uniformly over both objects and attributes. Datasets with between 0 and 2000 (57.9%) missing attribute values were generated. These missing values were replaced by the means of the appropriate attributes, and the objects were clustered into two, three and

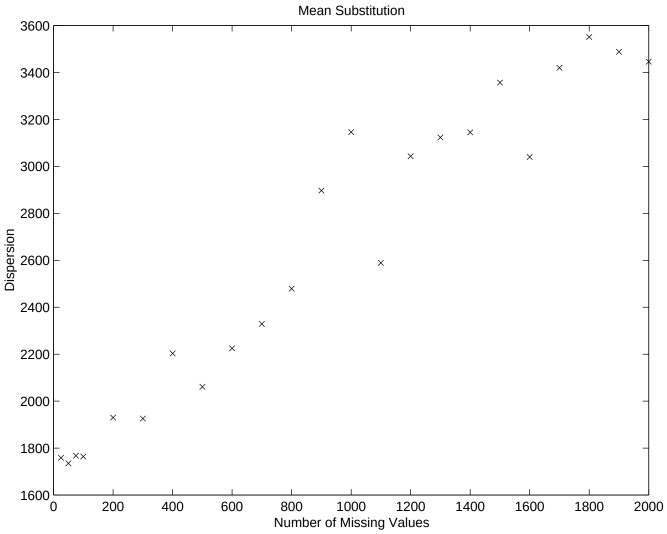

Figure 6: Dispersion for mean substitution as a function of the number of missing values

Cluster	0	1	2
Ellipticals	6	54	92
Irregulars	15	65	72
Spirals	93	230	236

Table 4: Mean substitution then Kohonen neural network clustering for a dataset with 2000 missing values

four groups, using the Kohonen network. A representative result is shown in Figure 6 for the case of three clusters. The results are similar for two and four output nodes. The number of missing attribute values is shown on the horizontal axis, and the average dispersion over 50 executions on the vertical axis. It is clear that the dispersion increases linearly with the number of missing attribute values.

The relationship of the clustering to the Hubble types is shown in Table 4. This appears to be quite a poor clustering, with each cluster containing proportional numbers of Ellipticals, Irregulars, and Spirals. The running time is only affected by the preprocessing step which requires two passes through the dataset, one to compute the mean, and one to replace the missing attributes. (Note that for an out of core dataset, this step, though conceptually simple, could be extremely expensive.)

3.3 Approach 3: Work around missing values.

Since AutoClass (and EM in general) can handle missing values, it can be used directly on datasets where some attribute values are not present.

Example: The clustering produced from the dataset with 2000 missing values is shown in Table 5. AutoClass has selected 6 clusters (it preferred 11 clusters on the complete dataset).

This clustering appears quite poor compared to earlier clusterings because there is no cluster containing predominantly irregular galaxies, and no combination of clusters that would contain predominantly elliptical galaxies. Furthermore, the clusters containing a predominance of spiral galaxies do not represent any meaningful subclass of the spirals, for example the clusters do not contain spirals of particular subtypes.

Cluster	0	1	2	3	4	5
Elliptical	38	94	14	4	2	0
Irregular	199	6	6	15	5	1
Spiral	183	96	151	125	2	2

Table 5: AutoClass clustering on a dataset with 2000 missing values

Missing values	Dispersion	Number of clusters
0	957	11
500	1307	8
1000	1526	9
1500	1820	6
2000	2171	6

Table 6: AutoClass dispersions and number of clusters as a function of the number of missing values

Table 6 shows the dispersion and number of clusters AutoClass produces for increasing numbers of missing values.

We now modify the neural network technique to handle missing values as shown in Figure 7 (with the changes from the standard Kohonen algorithm shown in italics). The algorithm determines cluster allocations based on known input attributes, disregarding the possible values of missing attributes and, at the same time, disregarding the fact that they are missing (this latter property is discussed further in Section 4). The architecture of the network changes dynamically, depending on the available input attributes for any given object. This is illustrated in Figure 8, which shows the structure of the network from Figure 4 for the case of an input pattern missing the second attribute.

Example: Figure 9 shows the dispersion of the clusterings achieved by the Kohonen network as the number of missing values increases. For ease of comparison with AutoClass, both the average dispersion over 50 runs and the best (smallest) dispersion are shown. Since AutoClass selects the best clustering it finds, it is best compared with the lower set of results.

Figure 10 compares the dispersion of the neural network clustering and the clustering produced using mean substitution. The deviations from a smooth upward line for both the average Kohonen network dispersion and mean substitution dispersion replicate across repeated experiments and need to be investigated further.

The dispersion achieved on this dataset (with 2000 missing values) and a target of 6 clusters is 2693 on average, and 1476 for the best solution. With a target of 11 clusters, the dispersion is 2681 on average and 1195 for the best solution.

Cluster	0	1	2
Ellipticals	29	119	4
Irregulars	134	1	17
Spirals	114	196	249

Table 7: Enhanced Kohonen neural network clustering with 2000 missing values

```
Randomly initialize the weights $w_{ij}$
repeat
  for Each of the k input pattern represented by the input vector $\vec{X}_k$
    for Each of the n output nodes
      Calculate the distances $D_1..D_n$ between weight vector and input vector
          *taking into account the known input values only*
      $D_i = \| \vec{W}_i - \vec{X}_k \|^2$
    endfor
    Calculate the winner by choosing the minimum $D_i$ for all i = 1..n
    Let $y_w$ denote the winning output node.
    for The Winning Output Node $y_w$
      for All n Weights $w_{iw}$ (where i = 1..n) leading into $y_w$
          *where the corresponding input value is known*
          Update weight $w_{iw}$ based on the difference between input value $x_i$ and weight $w_{iw}$
      end for
    end for
  end for
until cluster membership does not change anymore or the weight changes fall
      below a predetermined threshold
```

Figure 7: Enhanced Kohonen neural network learning algorithm

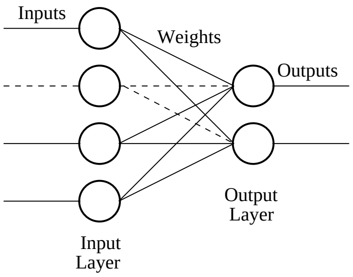

Figure 8: Kohonen network for input pattern missing the second attribute

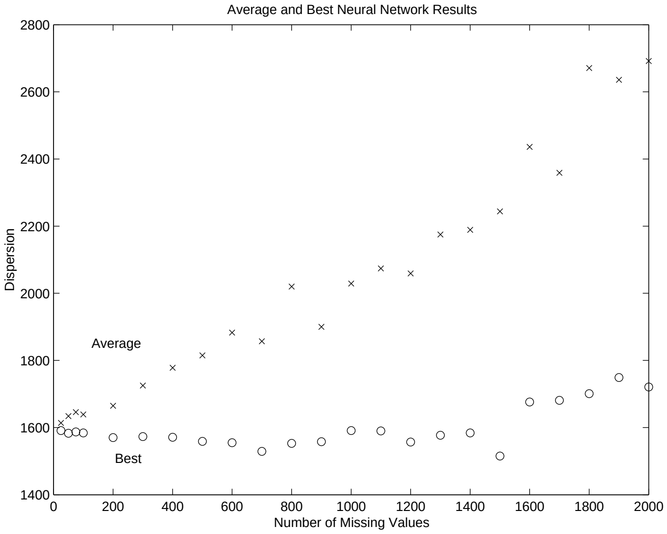

Figure 9: Dispersion for the enhanced Kohonen neural network clustering as a function of the number of missing values

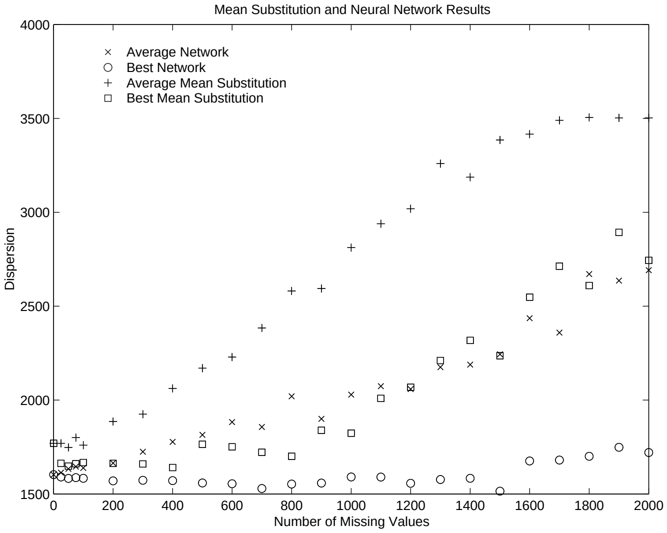

Figure 10: Comparison of dispersion for the enhanced Kohonen neural network clustering and mean substitution as a function of the number of missing values

Technique	Missing	Dispersion		
		3 clusters	6 clusters	11 clusters
AutoClass	none	1901	1317	957
SDD	none	1233	–	–
Kohonen	none	1603	1055	749
AutoClass	2000	2445	2171	2047
Mean substitution	2000	2744	2776	2570
Kohonen	2000	1721	1476	1195

Table 8: Comparison of minimum dispersions

For further comparison with the enhanced Kohonen network and additionally to the number of clusters it selected originally, AutoClass was also utilized to group the original dataset into 3 and 6 clusters and the dataset containing 2000 missing values into 3 and 11 clusters. Each of those clusterings was achieved by selecting the best (as determined by AutoClass) of 50 runs.

We summarize the results of all of these techniques in Table 8. The enhanced Kohonen network outperforms AutoClass whenever their performance can be directly compared, and AutoClass's clustering appears qualitatively poorer. The Kohonen network also outperforms mean substitution by a large margin.

The running time for AutoClass increases rapidly as the number of missing values increases: it increases by a factor of 3.8 over the time required for the complete dataset by the time 2000 values are missing. The running time for the enhanced Kohonen network does not increase as the incidence of missing values increases.

Overall, the enhanced Kohonen network outperforms AutoClass in subjective quality of clustering, dispersion, execution time, and simplicity (since priors in some form must be provided to AutoClass).

4 Do missing values convey extra information?

One possible problem with the enhanced Kohonen neural network is that it might treat the fact that particular attribute values are missing as an extra attribute when determining a clustering. In other words, the "holes" in the dataset might convey extra information that the algorithm can use. This could be a good thing, if the missing values are not missing at random; but a bad thing if they are. Since it is hard to tell which is the case, this at least introduces some uncertainty. We now show that, in fact, the pattern of missing values is not used by the algorithm.

To investigate this, a second experiment was performed. For each of the three main Hubble Types, 50% of the values for one of the attributes were removed. The resulting dataset contained 432 missing values representing 12.5% of the input data. The galaxies in this modified dataset were then clustered into two, three and four groups. The results for the case of two output nodes are shown in Figure 11. The graphs for other numbers of clusters are similar. Allocations to both of the clusters for the three major Hubble Types, elliptical, spiral and irregular galaxies are shown. The allocations to the clusters are shown in Figure 11.

Each of the bar graphs show the total allocations to the two clusters when the dataset was complete (left) and when there were missing values (right). The relative heights of the two kinds of input data show that missing values do not alter the allocation of objects to clusters. For example, instead of dividing the irregular galaxies over both clusters based on the fact that in half

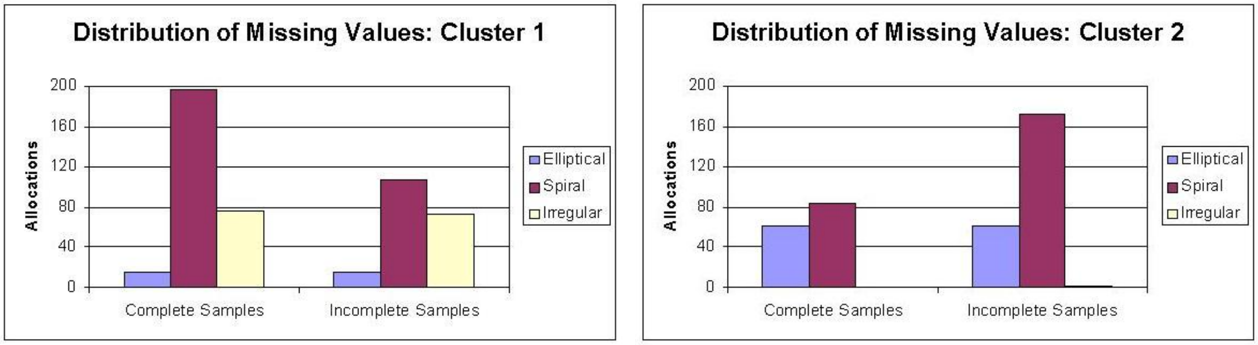

Figure 11: Allocation to clusters: complete versus incomplete attributes

the samples on attributes was missing, all Irregulars except for one were allocated to cluster 1. Also, the number of complete and incomplete samples for elliptical galaxies was split evenly over the clusters. In the case of the spiral galaxies, more incomplete samples were allocated to cluster 2 than to cluster 1, with a ratio of 1:2 for incomplete to complete samples. However, the same allocations were observed when clustering the complete data sample and are inherent to the actual data. We can therefore conclude that the algorithm did not differentiate across the clusters based on the fact that the input attributes were missing: the data was distributed across the clusters regardless of whether an attribute was missing or not.

5 Discussion

The Kohonen competitive learning algorithm uses each data point independently to develop steps towards an unsupervised clustering solution. The influence of an individual data point depends on many factors: the current value of the cluster centers, the learning rate and the order of presentation. Each contribution is inherently inexact and only provides a small increment in the general direction of the final solution. Thus missing data at any particular step of the algorithm has little detrimental effect on the solution. Kohonen learning allows each point to contribute evenly to the solution, and yet, in the case of our modification, allows each data point to be considered only for the components of the data that are present. The result is a technique that is extremely simple and straightforward, but apparently rivals and even surpasses existing more complicated techniques.

The proposed modification allows us to keep larger numbers of samples in the data sets. For the sample data extracted from LEDA, this means that we can now include all galaxies in the dataset for which at least one input attribute is known instead of restricting the sample size to 863. This was also evident in another sample extracted for a related experiment: a selection of seven input attributes for spiral galaxies contained only 460 complete samples, while the enhanced algorithm allowed us to include 2680 spiral galaxies in the sample, resulting in an increase in objects that could be included of 582%. In the latter case, 21.77% of the values were missing, but the sample included all of the spiral galaxies originally extracted from LEDA except for five, for which none of the attributes was known.

In general, the more data that can be included in building the model, the better the model will be. This is especially true when we do not have a clear prior knowledge of the structure of the model. The algorithm not only allows us to utilize *more* samples by allowing incomplete data, but also enables us to use *different* data as input to the neural network. The approach allows us to include data that would otherwise not be able to contribute to constructing the model, which would be especially obvious in cases of censored data for example, where the reason that the attribute is missing is related to the value of that attribute itself. It will allow us to include attributes which are appropriate for a subset of the data only, such as for example including the line widths for spiral galaxies and flagging them as missing values for any other type of galaxy.

6 Conclusion

We have presented an enhanced Kohonen unsupervised neural network algorithm that is able to handle missing input values, and hence to build models of datasets in which the values of some attributes are not present. The new algorithm provides better clusters in the sense of smaller dispersion about cluster centers than mean substitution, while preserving the simplicity and performance of such straightforward approaches. The ability to handle missing attribute values effectively is useful for many datasets, but we have suggested that it is particularly useful for astrophysical datasets where missing value problems are commonplace.

We have also shown that the pattern of missing values is not used by the new algorithm as an extra 'virtual' attribute. This makes it possible to use it on datasets with variant sets of attributes, modelling objects that do not have certain attributes as if these attributes were missing.

Acknowledgement: We are grateful to Judith Irwin for assistance with selection of attributes and input about the appropriateness of the clustering results produced by our techniques.

References

[BDO95] M.W. Berry, S.T. Dumais, and G.W. O'Brien, *Using linear algebra for intelligent information retrieval*, SIAM Review **37** (1995), no. 4, 573–595.

[CKS$^+$88] P. Cheeseman, J. Kelly, M. Self, J. Stutz, W.Taylor, and D. Freeman, *Autoclass: A bayesian classification system*, Proceedings of the Fifth International Conference on Machine Learning (Ann Arbor, MI.), Morgan Kaufmann Publishers, 1988, pp. 54–64.

[GTI99] E. García-Berro, S. Torres, and J. Isern, *Neural network classification of white dwarf populations*, ASP Conf. Ser. 169: 11th European Workshop on White Dwarfs, 1999, pp. 30+.

[HML94] E. S. Howell, E. Merényi, and L. A. Lebofsky, *Classification of asteroid spectra using a neural network*, Journal of Geophyiscal Research **99** (1994), 10847+.

[LNSS96] O. Lahav, A. Naim, L. Sodré, and M.C. Storrie-Lombardi, *Neural computation as a tool for galaxy classification: methods and examples*, Monthly Notices of the RAS **283** (1996), 207+.

[LR87] R.J.A. Little and D.A. Rubin, *Statistical analysis with missing data*, New York:John Wiley and Sons, 1987.

[MS01] Sabine McConnell and D. B. Skillicorn, *Outlier detection using semidiscrete decomposition*, Tech. Report 2001-452, Queen's University, Department of Computing and Information Science, November 2001.

[Ode95] S. C. Odewahn, *Automated Classification of Astronomical Images*, Publications of the ASP **107** (1995), 770+.

[SGG98] H.P. Singh, R.K. Gulati, and R. Gupta, *Stellar spectral classification using principal component analysis and artificial neural networks*, Monthly Notices of the RAS **295** (1998), 312–318.

[Wu97] J. Wu, *Prediction of geomagnetic storms from solar wind data using Elman recurrent neural networks*, 1997.