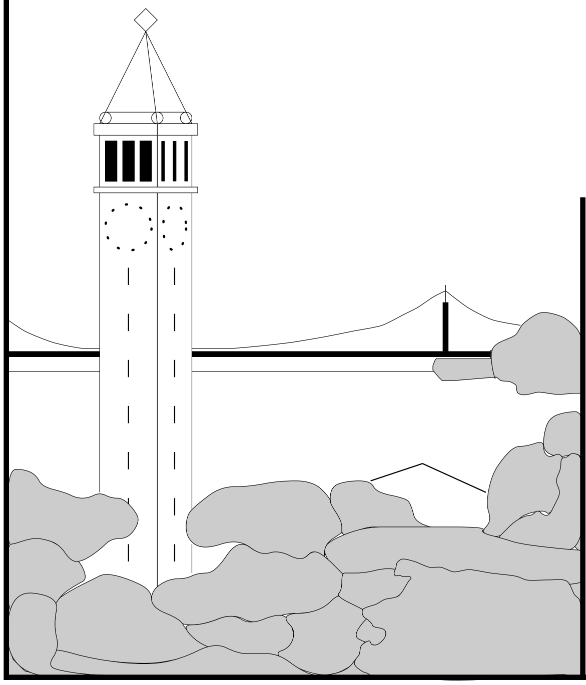
Visualizing Fuzzy Points in Parallel Coordinates

Michael R. Berthold and Lawrence O. Hall



Report No. UCB/CSD-99-1082

December 1999

Computer Science Division (EECS)
University of California
Berkeley, California 94720

1

Visualizing Fuzzy Points in Parallel Coordinates

Michael R. Berthold and Lawrence O. Hall*

Berkeley Initiative in Soft Computing (BISC)
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720
berthold@cs.berkeley.edu

*Dept. of Computer Science and Engineering, ENB 118
University of South Florida
Tampa, Fl. 33620
hall@csee.usf.edu

December 1999

Abstract

The ability to visualize data often leads to new insights. Data that is more than three dimensional must be visualized as a series of projections or transformed into some other representation which usually causes a loss of details. Parallel coordinates allows one to visualize data in two dimensions without a loss of information. In this paper, we discuss the use of parallel coordinates to visualize fuzzy data. Fuzzy data may consist of fuzzy rules, which can be viewed as *cutting a swath* through an n-dimensional space. Fuzzy clusters may also be considered fuzzy data in a similar way. Examples are given from three domains. The examples show that parallel coordinates can be used to find extraneous fuzzy rules, separate fuzzy clusters as well as validate previous findings about data sets.

1 Introduction

The automatic analysis of large datasets has gained increasing attention in the past couple of years. Scenarios where data analysis techniques can be used are many and stem from a wide range of diverse applications. Most techniques nowadays try to extract descriptors (usually in numerical form) or relationships (for example in the form of rules or mathematical functions). An introduction to the most commonly used techniques can be found in [1]. Due to the complexity of the resulting models (or their inaccuracy if the model's complexity is too low) increasingly methods have been proposed to visualize either the data itself or a model which aims to summarize all or a subset of the data. This way, the user can hopefully find structures of interest and, through a process commonly referred to as *exploratory data analysis*, fine tune the focus of analysis.

*BISC is supported by grant (ARO) DAAH 04-961-0341, (NASA) NAC2-1177, (ONR) N00014-96-1-0556, and (ONR) FDN0014991035. M. Berthold was supported by the Deutsche Forschungsgemeinschaft through grant DFG Be1740/7-1.

For data visualization many techniques exist, most prominently probably two-dimensional scatter plots along pairs of the features. Often this technique is extended to scatter plots of the most important principal components, hence providing projections of the data which have high variance. Of course histograms of each feature also belong in this category. However, such lower dimensional, linear projections always run into the risk of hiding interesting, higher dimensional structures. More sophisticated non-linear techniques are often based on multidimensional scaling [2] and try to find a lower dimensional representation that best matches the original data set.

Since data sets tend to be large and usually also redundant, processing and plotting all data points is often computationally expensive. Hence, attention has shifted to the visualization of models that summarize the data. Some techniques, such as Kohonen's self organizing feature map [3], try to find non-linear structures of fixed topology and size in the data which are then displayed in a lower dimensional (usually 2 or 3) space. Often hierarchical clustering techniques are used to display structure in the data, based on presumably more and less important prototypes. Data points themselves or representative cluster centers can also again be visualized through multi dimensional scaling techniques. These are often based on spring embedding [4] where the spring constants resemble relationships between parts of the model (often simply their distance in the original space).

All these techniques are based on some sort of compression, that is, high-dimensional structures are summarized and represented through elements in a lower dimensional visual representation. Hence complex, high-dimensional relationships are compressed into a much lower-dimensional relation. This loss in information can be harmful especially when the underlying dimensionality of the data (or representative model) is high[1]. A two dimensional feature map, for example, will fail to adequately model a higher dimensional structure. Therefore it is of great interest to find representations which enable a visualization of the entire structure of a dataset or the corresponding model without such a loss of information.

An interesting approach to visualize high dimensional data sets in the *same* dimension without any loss of information is *parallel coordinates* [5, 6] a technique which was recently proposed to find trends in data. Parallel coordinates allows one to visualize n dimensional data points in two dimensions. Essentially, parallel coordinates transforms multi-dimensional patterns into two-dimensional patterns without loss of information. Visualization is facilitated by viewing the two-dimensional representation of the n-dimensional data points as lines crossing n parallel axes, each of which represents one dimension of the original feature space. This approach scales well with increasing n and has been incorporated in a few data analysis tools such as Xmdv [7], XGobi [8], VisDB[9] and others [5].

Fuzzy models have not received a lot of attention with respect to visualization even though powerful algorithms to construct such models from data exist [10, 11, 12, 13, 14]. In this paper we therefore explore whether and how parallel coordinates can be used to visualize fuzzy points in n-dimensional space. In each dimension a fuzzy point has some fuzzy extent which in the narrowest case (that is, a singleton) would be a single value or point. Since fuzzy rules can also be considered as a fuzzy point in n dimensional space, this visualization is extensible to such rules as well. Another highly related example of fuzzy points is a fuzzy partition of a data set. The fuzzy partition can be represented by fuzzy clusters, which in turn can be viewed as centroids with a corresponding fuzzy neighborhood which again is representable in the framework of fuzzy points. Of course, fuzzy clusters have also been converted to fuzzy rules [15, 16, 17].

The rest of this paper proceeds as follows. In Section 2, we define parallel coordinates, Section 3 describes fuzzy rules and shows different approaches to plotting them in n dimensions. In Section 4, we show how parallel coordinates may be used to analyze data by using three different data sets. Lastly, in Section 5 a summary and discussion of the results is given.

[1] In contrast to a model in high dimensions which has a simple low-dimensional structure and can hence be transformed into a low dimensional representation without a substantial loss of information. Such models, unfortunately, are found rarely for real-world problems.

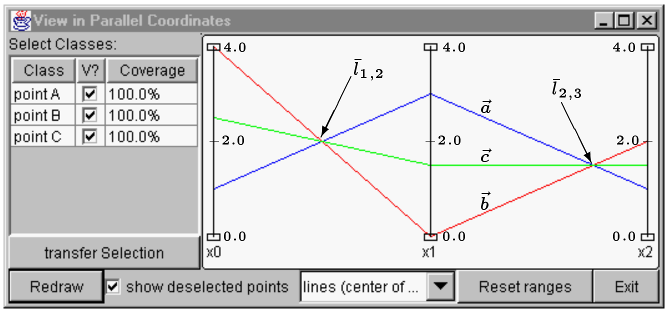


Figure 1: A parallel coordinate depiction of 3 points on a line with $\vec{a} = (1.0, 3.0, 1.0)$ (dark line), $\vec{b} = (4.0, 0.0, 2.0)$ (gray), $\vec{c} = (2.5, 1.5, 1.5)$ (light gray). The two intersection points at $\bar{l}_{1,2} = (0.5, 2)$ and $\bar{l}_{2,3} = (0.75, 1.5)$ uniquely describe the line going through all three points in the original Euclidean 3D.

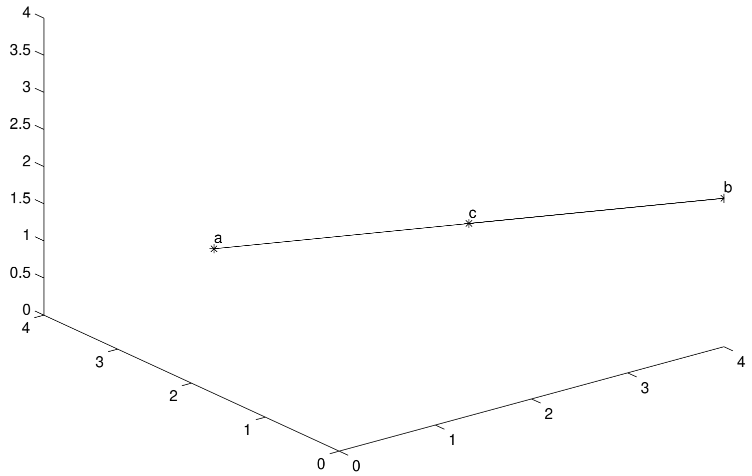


Figure 2: The point from Figure 1 in 3-D Cartesian coordinates

2 Parallel Coordinates

Parallel coordinates [18, 19] allows one to visualize n dimensional data in 2-D. Essentially, parallel coordinates transforms multi-dimensional problems into 2-D patterns without loss of information. Visualization is facilitated by viewing the 2-D representation of the n dimensional data.

If one takes each of n coordinate axes and lines them up in parallel, one has the basis for parallel coordinates. The distance between each adjacent axis is assumed to be equal to 1. A point in n dimensional space becomes a series of $n-1$ connected lines in parallel coordinates which intersect each axis at the appropriate value for that dimension. A parallel coordinates example of 3 points in 3-D, $\vec{a} = (1, 3, 1)$, $\vec{b} = (4, 0, 2)$, and $\vec{c} = (2.5, 1.5, 1.5)$, from a line is shown in Figure 1. The three points are also shown in a 3-D plot in Figure 2.

The dual of an n-dimensional line in Cartesian coordinates is a set of $n-1$ points in parallel coordinates [5, 20], for the example in Figure 1 these are indicated by $\bar{l}_{1,2} = (0.5, 2)$ and $\bar{l}_{2,3} = (0.75, 1.5)$ which uniquely describe a line in 3 dimensions.

The n-dimensional line in Cartesian coordinates can be represented by $(n-1)$ linearly independent equations each of which results from equating a different pair of the following fractions [19]:

$$\frac{x_1 - a_1}{u_1} = \frac{x_2 - a_2}{u_2} = \ldots = \frac{x_n - a_n}{u_n}. \tag{1}$$

Now it may be assumed that the $n-1$ linearly independent equations are obtained from pairing the $n-1$ adjacent fractions, with no loss in generality. This yields

$$x_{i+1} = m_i x_i + b_i,\ i = 1, 2, \ldots, n-1, \tag{2}$$

where $m_i = u_{i+1}/u_i$ represents the slope and $b_i = (a_{i+1} - m_i a_i)$ the intercept of the x_{i+1}-axis of the projected line on the $x_i x_{i+1}$-plane. The dual point of the n-dimensional line in parallel coordinates therefore corresponds the the set of n-1 indexed points

$$(\frac{i}{1-m_i)}, \frac{b_i}{1-m_i}),\ for\ i = 1, 2, \ldots, n-1. \tag{3}$$

There are other nice results about the parallel coordinate representation [21, 22, 20] that are not germane to this paper.

3 Fuzzy data in parallel coordinates

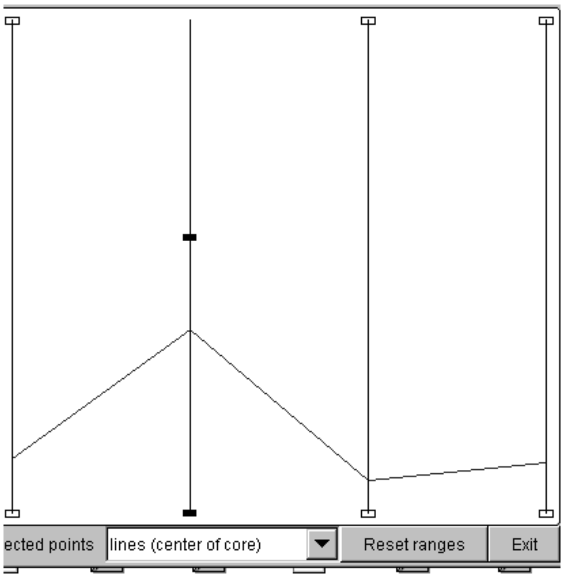
Our initial example showed several non-fuzzy points on a line. There are several ways that a fuzzy point might be represented. In Figure 3a we show that a fuzzy point can be represented as a single point which would be the core or centroid of the fuzzy rule, cluster, or point. It can also be represented by the number of training examples it covers (thicker lines for more examples), Figure 3b. A more intuitive, but perhaps more difficult to decipher, approach is to model the fuzzy point as a region in each dimension Figure 3c. When there are many fuzzy points each color or grey-level (as we use here) is overlaid. This can make it difficult to discern patterns in the data which is why the core point representation is offered. In Figure 3d, we show two overlapping fuzzy points/rules in different gray levels.

After the fuzzy data is displayed, it is necessary to view it in different ways in order to extract the information contained. For example, it may be desirable to view fuzzy points from one or two classes or regions only. One may wish to restrict the ranges of fuzzy points on one or more axes, in order to search for separation or trends.

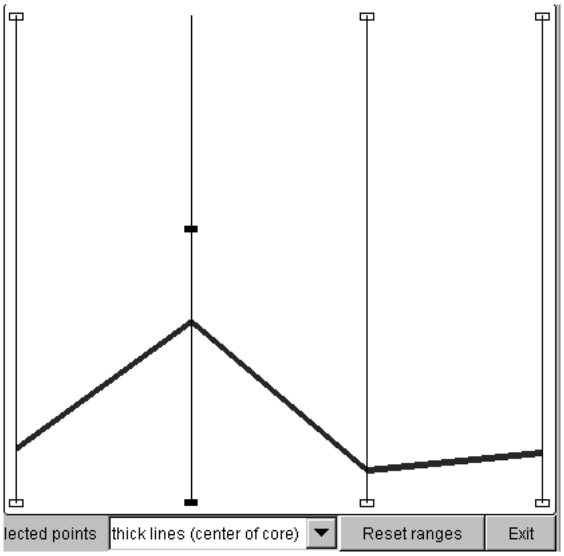
The tool used to display the fuzzy points in Figure 3 and in the following figures is part of the fuzzy data mining tool which contains a learning algorithm (and other features) described in [10, 23] and which was developed at UC Berkeley's BISC group (Berkeley Initiative in Soft Computing). The display feature allows for several different types of fuzzy point display of which parallel coordinates is one. Within parallel coordinates there are several options for the display of fuzzy points. The core point of the fuzzy point is a vector with each element the center of the core region for a fuzzy rule in the appropriate dimension. An example from a single fuzzy rule with 3 fuzzy attributes height, length and width is shown in Figure 4.

It is possible to display only the region of the fuzzy set where the membership is one (core) or the full range covered by a fuzzy set. Since, the learning tool used here provides a very large region of partial coverage (i.e. long arms of the trapezoidal fuzzy sets shown in Figure 4), only the core support of the fuzzy sets are displayed in each dimension here.

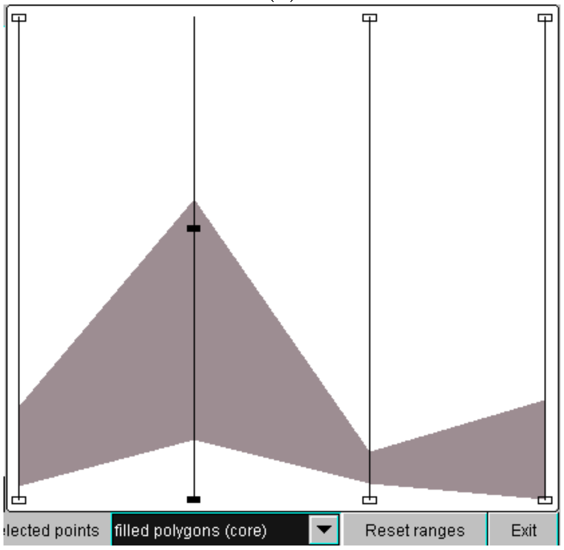
To display regions of less than full membership in a fuzzy set, it will be necessary to lighten the color or gray-level to indicate lessening of memberships in the set. An example is shown for the fuzzy rule from Figure 4 in Figure 5. Recognizing overlapping lighter colors will be challenging. It is possible to plot



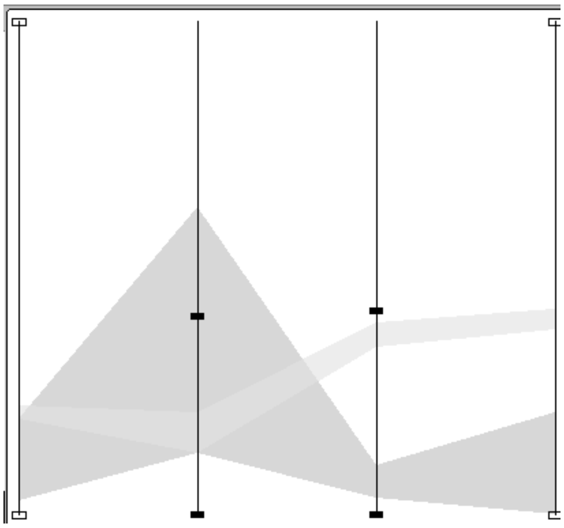
(a)



(b)



(c)



(d)

Figure 3: Displayed in parallel coordinates a) Fuzzy point as its core point, b) fuzzy point as its core point with thickness indicating support, c) fuzzy point as a region, d) overlapping fuzzy points or rules.

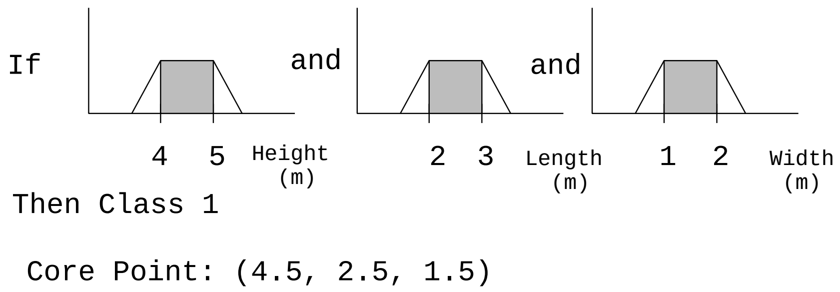


Figure 4: An example of a fuzzy rule represented as its core point. The core region of each set is shown by the shaded area.

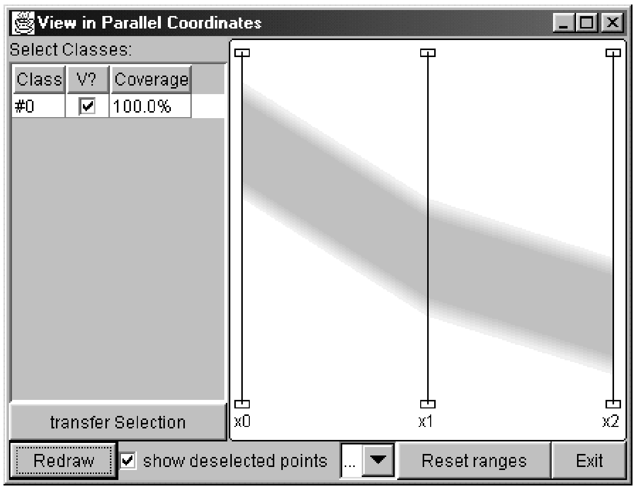


Figure 5: An example of a fuzzy rule represented as its core support region and in lighter gray its partial support region.

alpha-cuts of each fuzzy set so that only the region whose points have a membership equal to or above α are shown. Under an alpha-cut paradigm the remaining figures show plots with $\alpha = 1$.

A fuzzy point can also be represented by its coverage in each dimension as shown in Figure 3c. Overlapping fuzzy points tend to obscure the overlap unless colors are used and allowed to intermix. There is a mode for overlapping fuzzy points in which the colors for each point appear one vertical pixel slice at a time, which we call interlaced or mixed lines. This helps visualize the overlap of two or more fuzzy points by seeing different colors.

It is also possible to remove labeled classes from a plot. This is a useful feature when one wants to concentrate on the fuzzy points from a subset of classes. There are sliders on each axis which allow fuzzy points to be masked out. Any fuzzy point in a dimension which is not within the range allowed by the sliders is masked out and shown in a distinct color. These points can also (optionally) be removed from the parallel coordinate display.

The described functionality is illustrated by a set of examples in the next section. This set of examples shows the value of each way of displaying fuzzy points in parallel coordinates.

4 Fuzzy Data analysis

The operations applied to fuzzy data sets are best illustrated by examples. Three data sets are used to show the utility of parallel coordinates in analyzing fuzzy data. They are the well-studied Iris plant database and the Statlog shuttle database available from the UC Irvine [24] collection of data. Also, a set of cluster centers obtained from color satellite images off the West Florida shelf are utilized [25, 26].

4.1 Iris experiments

The Iris plant database consists of 150 examples each of 4 features which describe 3 types or classes of Iris. In our experiment, fuzzy points, fuzzy rules in this instance, were created from a training set of 75 examples. In Figure 6 we show the 11 fuzzy rules that were learned for three classes. The rules for a given

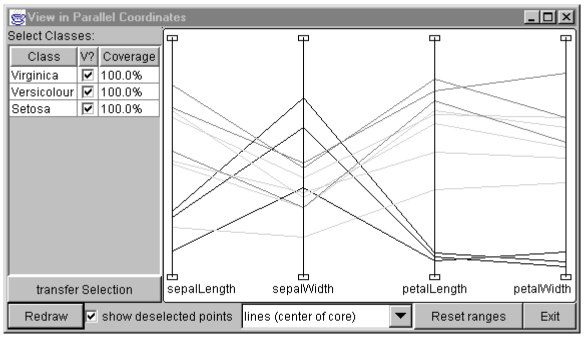
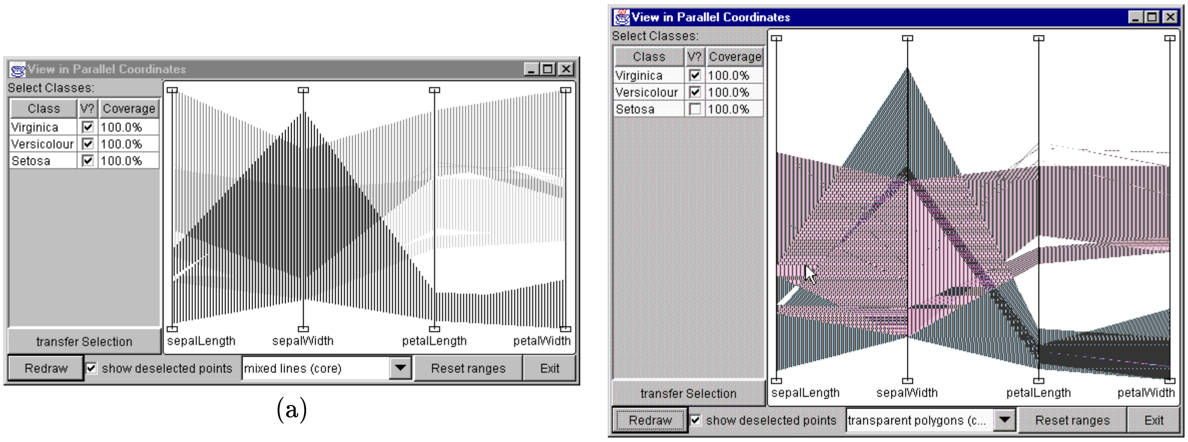


Figure 6: The parallel coordinate view of the Iris data set rule cores. Rules for each class are in the same gray level.



(a) (b)

Figure 7: (a) The parallel coordinate view using interlaced lines of the Iris data set rules and (b) the parallel coordinate view using transparent colors of the Iris data set rules for just the Virginica and Versicolour classes.

class are all the same gray level. This means the three rules whose cores are at the bottom of the petal width feature are from the same class. It can also be seen that the other two classes have 3 and 5 fuzzy rules describing them. Also shown in the upper left hand corner of the plot is the percentage of training examples correctly covered by the sum total of the fuzzy rules for each class. The check boxes allow the user to choose not to display one or more classes by turning the check off. Here, the union of the rules for each class correctly cover all examples in the training set for the class in this experiment.

One thing that is clear from viewing in parallel coordinates is that features 3 and 4 (petal length and width) appear to cleanly separate one class. In fact, for this data set it appears that all classes can be separated using only feature 3. However, the coverage of the fuzzy points is not given by the cores, so in Figure 7a we can see the interlaced line view of the overlaps among the rules. For a different point of view, we show overlaid or mixed colors (we call this transparent colors) in Figure 7b with just classes Virginica and Versicolour displayed.

Now, we investigate isolating classes by using the sliding boxes on each axis (sliders). Only the fuzzy points whose cores lie within the region between the two sliders on each axis are displayed and only their coverage (on the train set) is reflected in the upper left hand coverage area. In Figure 8 the class of Virginica has been isolated from the other two classes by just using petal length. This can be seen because the only fuzzy points between the sliders are of one gray level. Also, the display in the upper left shows that 100%

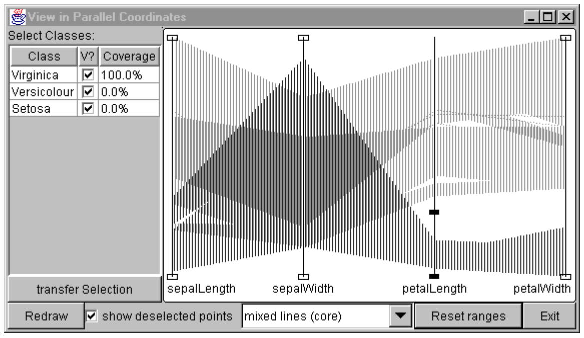


Figure 8: The parallel coordinate view after using the slider on petal-length to isolate only the fuzzy points describing the Virginica class.

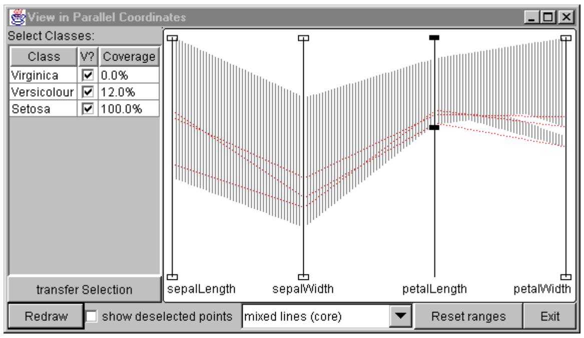


Figure 9: The parallel coordinate view after using the slider on petal-length to attempt to isolate only the fuzzy points describing the Setosa class.

of the training examples for Virginica are covered and no other (0% for the other two classes) examples are covered.

Given known results about the Iris data set [24], the fact that a single dimension can separate this class is simply confirmed in the parallel coordinate view. In Figure 9, we attempt to isolate the fuzzy points for the Setosa class using just petal-length. We can almost do it, but are left with 12% of the examples from the Versicolour class. This is because we cannot remove the last few fuzzy points for Versicolour without also removing some points/rules for Setosa. This fact is consistent with what is known about the Iris data set, namely that these two classes overlap.

4.2 Satellite data

The satellite data used here comes from a satellite used primarily to examine the ocean. The images are from the Coastal Zone Color Scanner (CZCS) and are of the West Florida shelf [25, 26]. The CZCS was a scanning radiometer aboard the Nimbus-7 satellite which viewed the ocean in six co-registered spectral bands 443, 520, 550, 670, 750 nm, and a thermal IR band. It operated from 1979-1986.

The features used were the 443, 520, 550, 670 nm bands and the pigment concentration value was derived from the lowest 3 bands. Atmospheric correction was applied to each image [27] before the features were extracted. A fast fuzzy clustering algorithm, mrFCM [28], was applied to obtain 12 clusters per image. There were five regions of interest in each image. These consist of red tide, green river, other phytoplankton blooms, case I (deep) water and case II (shallow) water. Twenty-five images were ground-

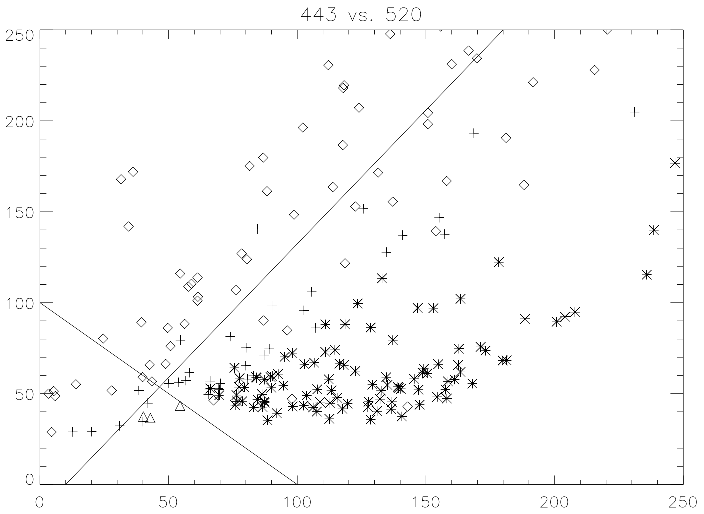


Figure 10: The 2-D projection with non-axis parallel lines which split off most of the red tide clusters. The red tide clusters are the triangular shaped objects on the graph, the +'s are other phytoplankton blooms (including green river), the stars are case I water and the diamond like objects are case II water.

truthed by oceanographers [29] and eighteen of these were used for training.

The eighteen training images were clustered into 12 classes. Each class or cluster was labeled from the ground truth image as its majority class. In [25, 29] red tide, triangle shaped objects, was separated from most other fuzzy cluster centroids by using non-axis parallel cuts in a two dimensional projection on the training images as shown in Figure 10. The lines can be used to build a rule to separate most of the red tide from the rest of the clusters.

The same training data set was plotted in parallel coordinates in Figure 11. Of interest is whether a rule can be obtained with similar accuracy from visualizing this training set with just one view, thereby eliminating the time consuming job of searching the projections. One item of note is that our current tool only allows for axis parallel partitions to be made of the fuzzy point space. The sliders on the axes only form a single axis parallel closed interval in each dimension. This made the task (artificially) somewhat more challenging.

In Figure 12a the 443nm feature is used to separate red tide from most of the other classes. In Figure 12b the 520nm feature is also used (the black sliders show the range) to isolate red tide from all but a few bloom and green river clusters. Finally in Figure 13, the 670nm band is also restricted to leave only some fuzzy points representing blooms which are very similar to red tide and accomplishing an equivalent rule to that from [29] without the necessity of creating and viewing many different projections. The cost of using axis parallel cuts is that one more feature had to be used to isolate the red tide clusters.

4.3 Shuttle data set

The Shuttle data set is from the Statlog project and consists of a training set of 43,500 examples and a test set of 14,500 examples. It is available as part of the UC Irvine database [24]. Each example consists of 9 continuous attributes and each example belongs to one of 7 classes. Eighty percent of the examples belong to Class one.

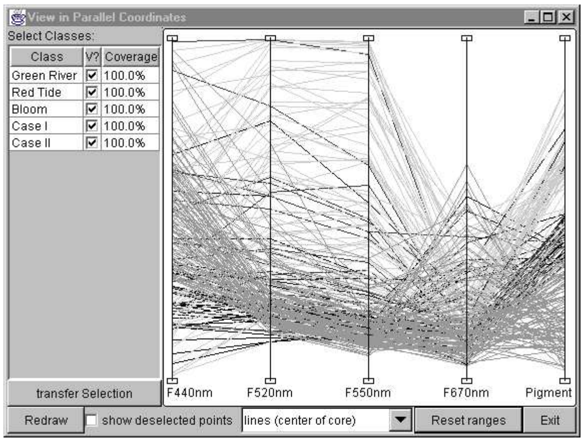
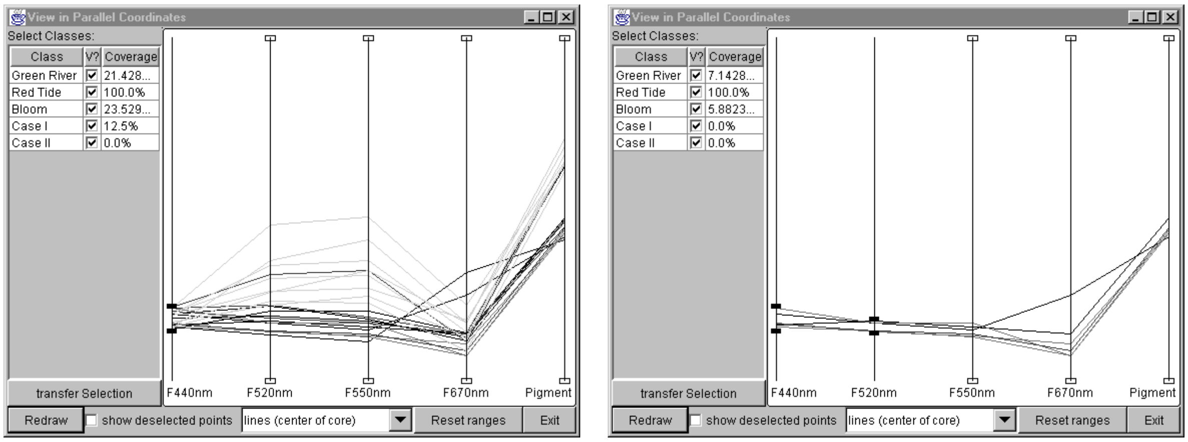


Figure 11: The parallel coordinate view of the 216 fuzzy cluster centers from the 18 ground truthed images.



(a) (b)

Figure 12: (a) Restricting the 443nm range of the training data set isolates some of the red tide clusters and (b) additionally restricting the 520nm band helps isolate most of the red tide clusters.

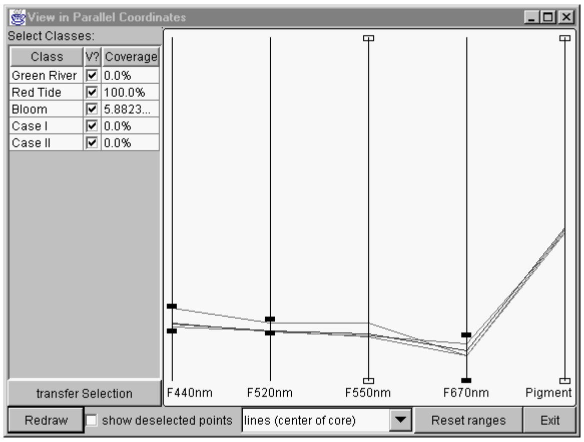


Figure 13: Finally restricting the 670nm band will allow the extraction of a rule which is quite good at identifying red tide clusters with few false positives (exactly 2 as before).

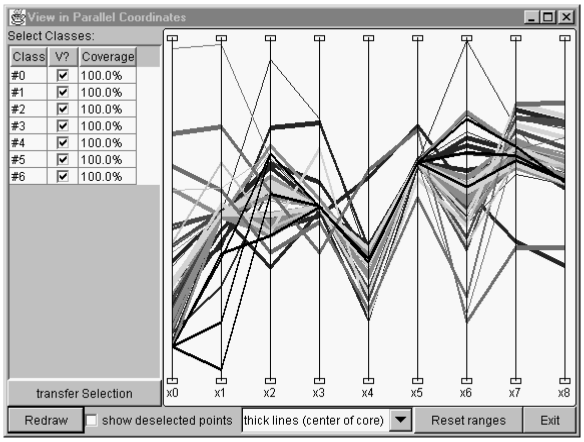


Figure 14: The 62 fuzzy rules learned on the Shuttle data. The view is with thick core lines, where line thickness indicates the number of examples a rule covers.

A set of 62 fuzzy rules were generated from the training data via the fuzzy learning algorithm that is integrated with the display tool. The rules or fuzzy points are shown in Figure 14 and they make only 9 errors on the test set (0.062% error). Of interest is whether it is possible to reduce the number of rules, visually, without seriously impacting performance.

Figure 15a shows the removal of a single rule that appears to be an outlier. This removal causes no increase in the error rate on the test set. The removal of a second rule, Figure 15b causes one more error on the test set, a very small increase.

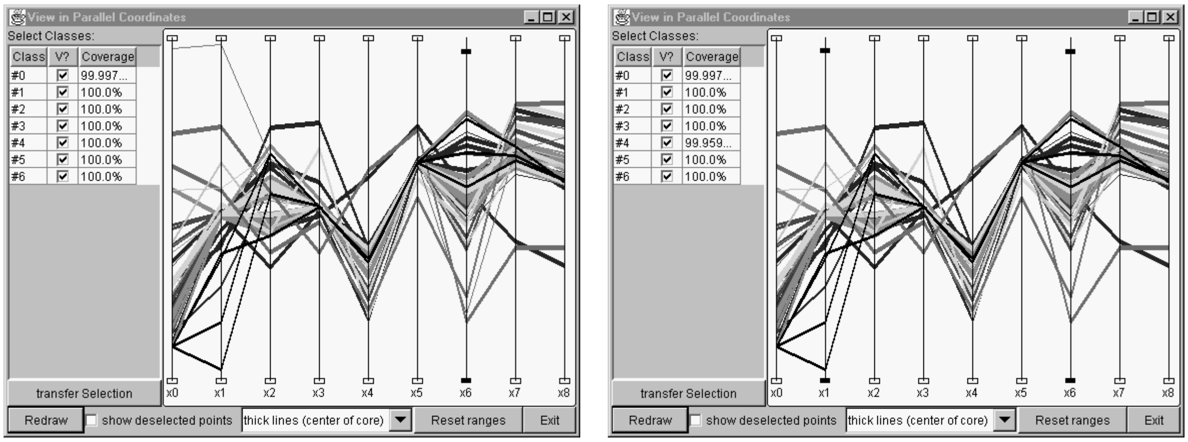
5 Summary and Future Work

Parallel coordinates allow n-dimensional fuzzy points to be viewed in 2 dimensions without loss of information. This paper shows how parallel coordinates can be used to both view fuzzy points in several different ways and understand them. Fuzzy points can be viewed by their cores or centroids, by thickline cores which indicate how many examples each point covers, by interlaced lines with each point assigned a color or gray-level and by transparent colors or gray levels. The last representation is difficult to see with gray-levels.

A java-based tool has been developed to display fuzzy points in each of the above described ways in n-dimensional parallel coordinates. The tool also allows for restrictive ranges to be applied to each dimension. For fuzzy points which correspond to a particular fuzzy or non-fuzzy class, each time restrictions are applied to an axis, the percentage of examples removed for each class is shown.

The visualization can be used to remove fuzzy rules that have no impact as shown in the Shuttle data set experiments. It can be used to remove features that are not helpful in separating classes as shown with experiments on the Iris data set in which the overlap between the fuzzy points in the two sepal features clearly precludes their use as class separators.

Parallel coordinates was also effective in visualizing a set of fuzzy clusters (by centroids). The visualization was used to separate red tide phytoplankton blooms from other types of blooms and water. We were able to recreate the results of a painstaking search through 2-D projections that produced a similar rule for isolating red tide. The re-creation required less than 5 minutes.



(a) (b)

Figure 15: (a) The removal of a single rule does not affect the test performance (we have removed some over fitting) and (b) the removal of a second apparent outlier rule causes a single error increase on the test data set. The view is with thick core lines, where line thickness indicates the number of examples a rule covers.

5.1 Future work

We have applied the concept of viewing fuzzy points in parallel coordinates only to fuzzy points which are rules or centroids of potential rules that will be used to place an object in a class. The technique may also be useful for fuzzy control rules in which the consequent of a rule is a fuzzy set or fuzzy function. The visualization process may indicate where coverage is not fine enough (light overlap) and appears likely to make it easy to view uncovered regions.

To apply the idea to fuzzy control rules, an indication of *goodness* of control would need to be displayed after every change to the fuzzy points/rules.

One intriguing possibility with parallel coordinates is to add features that are, for example, projections onto non-axis parallel lines or more complicated features to allow for the data to be separated. The extra features can still easily be visualized, may separate the data much better even though they may be redundant.

The tool we have built could be further extended to allow fuzzy sets to be defined on each axis. Then a fuzzy rule/point could be created by linking a choice of 1 or more fuzzy sets per axis. Not all axes need to be included in every rule. Hence, it seems a natural approach to the visual development of fuzzy sets. In a way analogous to our current accuracy percentage for the training set, after a rule is *visually* created an accuracy measure can be displayed for it when it is applied to classification data. Then the fuzzy sets may be modified to achieve the desired accuracy on the training or validation set. This approach seems to hold promise for developing fuzzy classification rules and interactive data exploration.

Acknowledgements: This research was partially done at BISC while L. Hall was on sabbatical. Thanks to UC Berkeley's Div. of CS and Prof. Zadeh for the use of their facilities.

References

[1] M. Berthold and D. H. (eds.), *Intelligent Data Analysis, An Introduction.* Springer-Verlag, 1999.

[2] J. Meulman, *A distance approach to nonlinear multivariate analysis.* DSWO Press, Leiden, The Netherlands, 1986.

[3] T. Kohonen, *Self-Organizing Maps.* Springer-Verlag, Berlin, Heidelberg, 1995.

[4] N. Q. Jr. and M. Breuer., "A force directed component placement procedure for printed circuit boards.," *IEEE Transactions on Circuits and Systems*, vol. 26, no. 6, pp. 377–388, 1979.

[5] A. Inselberg, "Multidimensional detective," in *IEEE Conference on Visualization?*, 1997.

[6] A. Inselberg and T. Avidan, "The automated multidimensional detective," in *IEEE Visualization Conference*, 1999.

[7] Y.-H. Fua, M. Ward, and E. A. Rundensteiner, "Hierarchical parallel coordinates for exploration of large datasets," in *IEEE Conference on Visualization*, 1999.

[8] D. F. Swayne, D. Cook, and A. Buja, "Xgobi: Interactive dynamic data visualization in the x window system," *Journal of Computational and Graphical Statistics*, vol. 7, no. 1, 1998.

[9] D. Keim and H. Kriegel, "Visualization techniques for mining large databases," *IEEE Trans on Knowledge and Data Engineering*, vol. 8, no. 6, pp. 923–938, 1996.

[10] M. R. Berthold and K.-P. Huber, "Constructing fuzzy graphs from examples," *Intelligent Data Analysis*, vol. 3, no. 1, pp. 37–54, 1999. (http://www.elsevier.nl/locate/ida).

[11] J.-S. R. Jang and C.-T. Sun, "Neuro-fuzzy modeling and control," *Proceedings of the IEEE*, 1995.

[12] Z. Chi and H. Yan, "ID3-derived fuzzy rules and optimized defuzzification for handwritten numeral recognition," *IEEE Transactions on Fuzzy Sets*, vol. 4, no. 1, pp. 24–31, 1996.

[13] O. Cordon, M. D. Jesus, F. Herrera, and M. Lozano, "MOGUL: a methodology to obtain genetic fuzzy rule-based systems under the iterative rule learning approach," *International Journal of Intelligent Systems*, vol. 14, pp. 1123–1153, Nov 1999.

[14] M. Figueiredo and F. Gomide, "Design of fuzzy systems using neurofuzzy networks," *IEEE Transactions on Neural Networks*, vol. 10, pp. 815–827, July 1999.

[15] S. Chiu, "Fuzzy model identification based on cluster estimation," *Journal of Intelligent & Fuzzy Systems*, vol. 2, no. 3, 1994.

[16] M. Delgado, G.-S. A.F., and M. F, "A fuzzy clustering-based rapid prototyping for fuzzy rule-based modeling," *IEEE Transactions on Fuzzy Systems*, vol. 5, pp. 223–233, May 1997.

[17] M. Friedman and A. Kandel, *Introduction to Pattern Recognition : Statistical, Structural, Neural and Fuzzy Logic Approaches.* World Scientific, 1998.

[18] A. Inselberg, "The plane with parallel coordinates," *Visual Computer*, vol. 1, pp. 69–91, 1985.

[19] A. Inselberg and B. Dimsdale, "Multidimensional lines i: representation," *SIAM J. Applied Math*, vol. 54, no. 2, pp. 559–577, 1994.

[20] S.-Y. Chou, S.-W. Lin, and C.-S. Yeh, "Cluster identification with parallel coordinates," *Pattern Recognition Letters*, vol. 20, pp. 565–572, 1999.

[21] A. Inselberg and B. Dimsdale, "Multidimensional lines ii: proximity and applications," *SIAM J. Applied Math*, vol. 54, no. 2, pp. 578–596, 1994.

[22] C. Gennings, K. Dawson, W. Carter, and R. Myers, "Interpreting plots of a multidimensional dose-response surface in a parallel coordinate system," *Biometrics*, vol. 46, pp. 719–735, 1990.

[23] K.-P. Huber and M. R. Berthold, "Building precise classifiers with automatic rule extraction," in *IEEE International Conference on Neural Networks*, 3, pp. 1263–1268, 1995.

[24] C. Blake and C. Merz, "UCI repository of machine learning databases," 1998.

[25] M. Zhang, L. Hall, and D. Goldgof, "Knowledge-based classification of czcs images and monitoring of red tides off the west florida shelf," in *The* 13^{th} *International Conference on Pattern Recognition*, vol. B, pp. 452–456, 1996.

[26] M. Zhang, L. O. Hall, D. B. Goldgof, and F. E. Muller-Karger, "Fuzzy analysis of satellite images to find phytoplankton blooms," in *IEEE International Conference on Systems Man and Cybernetics*, 1997.

[27] G. H.R., D. K. Clark, J. L. Mueller, and W. A. Hovis, "Phytoplankton pigments derived from the nimbus-7 czcs: comparisons with surface measurements," *Science*, vol. 210, pp. 63–66, 1980.

[28] T. W. Cheng, D. B. Goldgof, and L. Hall, "Fast fuzzy clustering," *Fuzzy Sets and Systems*, vol. 93, pp. 49–56, 1998.

[29] M. Zhang, L. Hall, and D. Goldgof, "Knowledge guided classification of coastal zone color images off the west florida shelf," Tech. Rep. ISL-99-11, University of South Florida, Dept. of CSE, USF, Tampa, FL., 1999. Under review in International Journal of Pattern Recognition and AI.