CorDex: a Knowledge Discovery Tool

Robert Leivian, William Peterson, and Mike Gardner

Motorola, Phoenix Corporate Research Laboratories
2100 E. Elliot Road, MD EL508, Tempe, Arizona 85284, USA
Bob_Leivian-RVNW60@email.mot.com fax:(602)413-7281

Abstract

CorDex is a program that organizes/categorizes/sorts items with a large set of attributes using a SOM (self-organizing map). This map allows the user to visually scan for similar items and the clustering of items based on their attributes. The program also allows users to visually search 'attribute planes' of this SOM's attributes to find: correlations among attributes, clusters which share similar attribute values but are very different in other respects, and attributes which are statistically unrelated. CorDex can process tens of thousands of items and find 'information' or 'rules' from 'raw data'. It has also been used to cluster items into distinct groups to allow traditional Rule Induction to find rules in large data sets that were not attainable without the CorDex pre-clustering. CorDex has been used to analyze semiconductor manufacturing yield problems and find root causes of problems that have eluded even the best process engineers for years.

1 Introduction

CorDex is a program that reads information about a set of items (rows) with attributes (columns) from any standard speadsheet or database 'view' and uses a SOM (self-organizing map) to 'sort' the data into a 2D grid. A CorDex map has several advantages over many traditional statistical data visualization tools because of the SOMs ability to handle much higher dimensional data; we have used data sets with over 200 attributes successfully. The properties of a SOM ensure that 'like' items are placed near each other on the map and dissimilar items are placed further apart. Once the map has been organized, a display mode can draw the name or icon of the closest item for each grid point in the map. In another display mode the user can choose any one attribute of each item and use it to display a color (or text code) based on its value that can be used to draw a colored rectangular 'pixel' or text string at the grid point in the map that corresponds to that item. This is referred to as a 'CorDex attribute plane'. The pattern these planes form can quickly reveal the details of the nature of that attribute in the set of items.

By visually examining several attribute planes, a user familiar with the items can derive 'rules' or 'knowledge' about the items. It is important to note that a human must be in the loop, CorDex is a tool much like a microscope in that the images it produces are not easy to 'computer analyze'. However, like the microscope it allows one to see things that they might not otherwise find. And since the grids can be made arbitrarily large and each item needs only one pixel, relationships among even tens of thousands of items can be scanned in seconds. An un-correlated attribute will look like 'snow' on a TV screen. A very strong first or second principle component will form a left to right or top to bottom 'rainbow'. A highly correlated but more complex principle component will look like a large splotch of a single hue. A significant but somewhat secondary attribute will form 'leopard spots'. An invariant or minor component will form a featureless monotone sheet. By looking for large splotches and leopard spots and ignoring the others you can quickly find the 'interesting' attributes of a given set of data items. Then by comparing the location of the splotches from different attributes you can quickly find correlations and anti-correlations. This visual method has proved very useful in a wide variety of real world data analysis applications, and unlike conventional statistics, appears to do better as more attributes are used.

1.1 The SOM Process

The CorDex program uses a fairly conventional SOM neighborhood-based learning algorithm to learn the mapping from N-space to 2-space[1]. Each item has N attributes and the scaled and encoded value of each

attribute forms a dimension in the input vector; thus each item represents a point in hyperspace. A few minor enhancements and optimizations to the standard SOM have been made by our group. If an attribute for a given item is not known, the mean value of the other known item values is used, this way incomplete or missing data can be accommodated. We use a 'stovepipe hat' neighborhood definition and Manhattan distance rather than euclidean distance for faster, simpler training. We also normalize all values to one byte to allow integer math, with faster execution and one-eighth the memory requirements of standard floating-point systems. The integer version of the math allows similar, if not better, convergence of the map and the granularity of one byte values seems to help avoid local minimums somewhat better. We have also looked at other methods for organizing the map, such as GTM[3], and they show promise for small to medium maps, but these batch methods are limited to smaller item sets because of the memory and time requirements. Standard Kohonen-type SOMs appear to scale much better.

1.2 The Item Location Map

The CorDex item location map display is based on our earlier work in the eighties of using SOMs to learn cursive handwriting and is similar to the poverty map in Kohonen[3]. This display is used for finding large, overall, patterns in the data and is the basis for determining how well the map organized. We have also added an option to display the borders between map cells in a shades of gray based on the hyperspace distance of adjoining 2D cells in the map. We call this display 'distance based borders' and it allows the user to better see folds or fractures in the map by drawing dark boarders along fractures and light borders between groups of similar cells, and retains the interior color of the cell for other uses. This border display mode is an improvement on Kaski's[4] display because each neighbor relationship is displayed not an average of all neighbor cells. There is a mechanism for zooming in or out to see more or less detail. These features have proved very useful in quickly determining visually how well the high dimensional data space mapped down to two dimensions.

Mercedes-Benz S500	Mercedes-Benz S600	BMW 540i	BMW 740i	Cadillac "De Ville Concours (35,000	Cadillac De Ville Concours (new)	Buick Regal Gran Sport	Ford Crown Victoria LX	Ford Windstar LX	Ford Explorer XLT	Range Rover 4.0 SE
Jaguar XJ12	Lexus LS400	Mercedes-Benz E420	Cadillac SLS	Lincoln Continental		Oldsmobile Eighty Eight LSS	Lincoln Town Car Signature Series	Ford Ranger Splash	Jeep Grand Cherokee Laredo	Chevrolet Blazer LT
Jaguar XJR	Infiniti Q45	Mercedes Benz C36	BMW 750 iL		Audi A6 Quattro		Dodge Dakota Sport	Chevrolet S10 ZR2	Chevrolet Tahoe LT	Mitsubishi Montero SR
Jaguar XJ6	Volvo 850 T-5R			Volvo 960		Mazda 626 ES	Nissan Truck SE-V6	Toyota Tacoma SR5 V6	AM General Hummer	Chrysler Town & Country
Chevrolet '61 Impala SS 409		Mazda Millenia S (new)				Honda Accord LX		Chrysler Town & Country LXi		Volkswagon EuroVan Camper
Callaway Camaro C8	Pontiac Grand Prix 300GPX		Mazda Millenia S (35000 miles)	Nissan Maxima SE	Toyota Avalon XL	Volkswagen Passat GLX	Acura 2.5TL		Chrysler Electric Van	Subaru Impreza L
Chevrolet Corvette	Ford Mustang GT SVO GT-40	Ford Mustang Cobra R		Infiniti I30t		Saab 9000CDE		Honda Odyssey EX		Subaru Legacy Outback Wagon
Pontiac Firebird Formula Ram Air W	Lingenfelter Firebird 383	Chevrolet Camaro Z28	BMW M3 (35000 miles)	Lincoln Continental	Toyota Camry SE		Honda Accord EX (35000 miles)	Honda Accord EX (new)	Chrysler Sebring LX	
Dodge Viper RT/10		Lotus Espirit S4S	Ferrari 456 GT	BMW M3 Lightweight	BMW M3 (new)	Mitsubishi Galant GS (35000 miles)	Mitsubishi Galant GS (new)	Dodge Stratus ES	Acura Integra LS	Pontiac Grand Am GT Coupe
Guldstrand Corvette GS90	Porsche 911 Turbo			Toyota Supra (35000 miles)	Saab 900 SE Turbo	VW GTI	Saturn SL2	Toyota Tercel	Dodge Neon Sport Coupe	Nissan 200SX SE-R
Ferrari F355	Ferrari F355	Oldsmobile Cutlass Supreme IMSA GTS	Acura NSX-T	PFS Miata SC	Toyota Supra (new)	Saturn SC2	Nissan 200SX SE-R	Chevrolet Cavalier	Nissan Sentra GXE	BMW 318ti

Cars:Make Model Sort Engine Drive Passengers Doors Body Price_As_Tested Base_Price Fuel Type No_Cyl Bore_mm Stroke_mm Disp_cc Comp_ratio Fuel_sys ... Friday February 21, 1997 10:27:17 AM

Figure 1. The Item Location Map for the vehicles example

1.3 The Attribute Plane Map

The attribute pane map has proved most useful in finding more subtle and useful information about the data. In this display mode the same spacial arrangement of the item location map is maintained, but instead of putting the name of the item in text at the cell location, a colored pixel based on the value of a single selected attribute of each item is drawn. If there is room, a percentage above or below the mean for that attribute is also drawn (see Figure 2). A similar type display with blank and white hexagonal grid has been shown by Samuel Kaski in a paper on poverty attributes[4]. In this example there are only 121 cells but in

the other cases, such as in our semiconductor yield analysis there can be tens of thousands of cells to represent several months wafer production. The item location map will only show wafer lot serial numbers in text to small too read, however the attribute plane can show pictures of a single attribute, such as yield, and how it was distributed among the wafer lots. By comparing the patterns in other attribute planes for similar patterns, the user can find correlations in a manner similar to what early astronomers used to find planets and comets in pictures of thousands of stars with blink comparators.

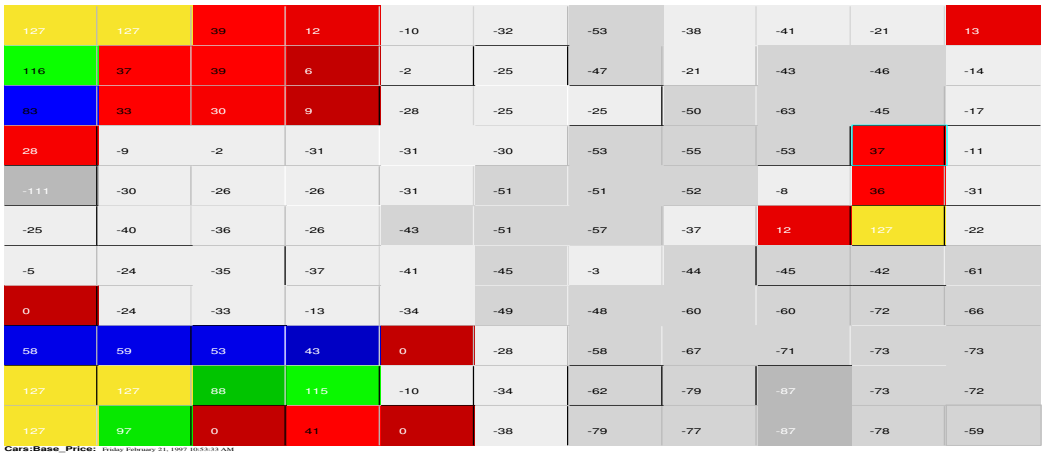

Figure 2. The Attribute Plane display for the vehicles attribute ‘base selling price’

2 Some Example Database Experiences

We have been working with CorDex for a few years internally at Motorola. We have developed experience with some toy projects, some interesting but non-practical, and several real world, very hard problems. Our progress has been excellent and matches/exceeds human application-expert level analysis in many cases.

This paper will discuss three of the problems that CorDex has addressed in our research.We hope that these examples will give the reader a good idea of the generality and usefulness of the program. They represent some discovered information in these data sets that range from ‘obvious’ to ‘not found by a year of expert analysis’.

2.1 Vehicles

Our first example is a very good for understanding how CorDex works, and is the first example new users use to train themselves. The data is already familiar to most people and the conclusions and patterns found in the data are obvious but illustrate the connection between attribute plane shapes and known relationships. The data set consists of 99 cars and trucks reviewed in 1996 consumer magazines and a few older ‘outliers’ of more unusual vehicles to make it interesting. There are up to 73 attributes recorded for each car (mileage, passenger seating, engine size, repair ratings, price, features, etc.). The SOM quickly learns and builds the map in a few seconds on a Power Macintosh. The item display map is reasonable and very intuitive: Mercedes and BMWs end up in one area of the map, Corvettes and Porsches in another, Ford Broncos and Land Rovers in another. The outliers (e.g. a Chrysler electric van, and a '61 Impala SS409) are separated from the others.

It is interesting, however, to look at an attribute plane such as ‘base price’ (see Figure 2). The picture has three large widely separated dark spots and a single large light spot and a background of white. The dark spots represent high priced vehicles and the light spot inexpensive vehicles. CorDex is telling you some discovered information (that you already knew, but it makes a nice confirmation of the system) about this set of vehicles: there are three very different kinds of expensive cars but most inexpensive cars are very similar. With a mouse click CorDex can show you the item names in each cell of the expensive cluster and it shows BMW, Lexis, Jaguar etc. in one cluster (luxury cars), Porsches, Ferrari etc. in another (sports/race

cars) and Land Rovers, HumVees and Broncos etc. (sport/utility) in the third. However, all the Tercels, Sentras and Cavaliers were in the same cluster of inexpensive vehicles. It is also possible to look at the 'fuel economy' plane and find anti-correlations with 'quarter mile race time', and find two kinds of high horsepower vehicles (sport/utility and race/luxury).

2.2 Movies

We have also have a data set of over 5,000 movies on video tape with 10 attributes (year made, academy awards won, how long it runs, rating etc.) that has some very real patterns that are not so obvious since the size and variety of items is much larger. This is usually the second example most users explore. When you look at the various attribute planes in this example you see more leopard spots and fewer rainbows, signifying a more complex set of relationships. However, it is easy to see some interesting relationships, for example, a large percentage of the films that have won academy awards over the years have been longer than average films. This is easy to see once you know what to look for; the picture produced by both the length and awards attribute planes produce leopard spots. CorDex has a 'region' tool (more on this later in the paper) that can draw borders around the spots in one plane and preserve this outline when you switch to other planes, you can easily see where the spots line up when there is a correlation, and the spots are in different places when unrelated.

2.3 Semiconductor Yield

The previous examples are used mostly for tutorials, however we have used CorDex to analyze very tough real world problems and it has solved some world class problems. We have used to it find production yield problems in an existing semiconductor wafer manufacturing line. The line is mature and optimized as well as current process engineering allows, at a yield rate above the industry average. However, it experiences an occasional small yield dip that was not explained by any existing analysis. This problem was the hardest one we could find, there was no 'low hanging fruit' involved here so we thought it would make a good test case. CorDex combined with rule induction[5] was able to identify the cause by sorting two months production, one from a 'good' month and one from a 'bad' month, a set of over 17,000 items.

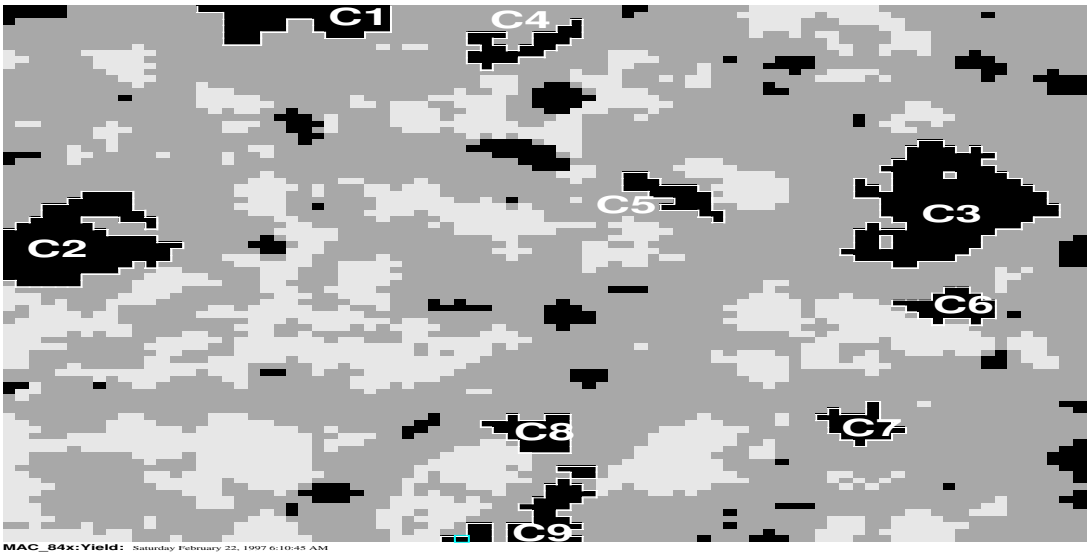

Figure 3. Semiconductor Low Yield Clusters with Rule Induction border overlays

For this problem we gathered 132 attributes for each wafer produced, including such varied information as which production machines were used, material vendor for each lot, the individual step's operator's initials, the time of day each production step was performed, results of process control sites on each wafer, in-process and post-process probe tests, the final test analog performance parameters. This data comprised over 14 megabytes of information. Although we consider this a 'medium sized' real world problem, many

of the other data analysis tools consider this 'huge' and are unable to even load a data set this big. We ran this problems on a standard, 32-megabyte RAM, Macintosh PowerPC in 72 hours of processing time.

The program showed the yield attribute plane to be leopard spots with two large spots and several medium sized spots (see Figure 3) so we knew the data was correlated and we hoped we could find the reasons for the yield dip. It was not process limited (i.e. random miscellaneous failures with no pattern). This quickly told us that there were at least several separate and distinct failure modes: two major and several minor. However there was no single strong correlation among the groups. Also, we found at least 30 of the 132 attributes at least somewhat involved visually. Although we had narrowed it down quite a lot, we could not manually find the cause without further help.The group had previously tried Rule Induction on this data set to no avail. However, when we tried it again with only the CorDex selected main failure modes as classifications it was able to find several significant rules involving a few material lots and a certain EPI machine. When we presented this data to the process engineer, he looked at the data and several other attributes and quickly hypothesized that the rules were suggesting the failure was a 'stacking fault' (the details of epitaxy, stacking faults, etc. are beyond the scope of this paper, but is addressed in standard semiconductor processing texts). The engineer then looked at the two major leopard spots and quickly found that the main difference between them was the final acceptance tests they were failing and that they were actually the same failure mode; if they occurred in an input transistor one set of tests would fail and if occurring in an output transistor would fail a different set of tests. These two spots were, in fact, of the same basic type and accounted for most of the yield dip of the wafers. CorDex had found the problem: stacking faults probably caused by poor material combined with a marginal EPI machine.

3 CorDex Visual Search Features

CorDex has added several tools to make pattern finding easier. We have added conventional cluster analysis to group items and compare its grouping to SOM grouping. We have added user defined 'Regions' using conventional boolean operations on the attributes (e.g. show only the cells that have attribute: price < $20,000 and attribute: mileage > 20 m.p.g.). There is also a method for masking the display to show only cells that are selected by any of the above groupings, or showing them but drawing a bright border around only the selected cells. All these tools allow you to find patterns much easier.

3.1 Clusters

We have added and investigated several well known clustering methods such as K-means and nearest neighbor for grouping cells and looking for patterns. In general these have not proved to be very useful as they suffer the well known 'curse of dimensionality', when used on very large number of attributes. SOM's in general appear to avoid at least some of this curse.

3.2 Regions

CorDex supports a general concept that we call 'Regions' that allow the user to group items by the region of the 2D map they occupy rather than clusters they form in the original data N-space. This appears to be a key to finding the subtle but useful patterns that some traditional methods can't. We believe that this is because of the non-linear nature of SOMs to simply ignore conflicting or irrelevant data, and let the neighborhood competition find relevant attributes when forming the maps. There are two main methods of defining regions or cells. You can use the mouse to drag selection rectangles for broad regions of the map, then shift-click to refine that selection. You can also use boolean operations to select cells with a union or intersect (AND/OR) based on the map cell attribute values being: greater than, less than, equal to, different, or within X units of: a selected reference item's specified attribute, specific value, or a given percentage of the mean of that attribute in the current distribution. These regions can be given user names such as 'economy cars' or 'good yield wafers' and the region of the map that they occupy highlighted or masked when comparing attributes looking for commonality.

3.3 Masking and/or Outlining Region Displays

The Regions of the map, once defined, can be masked, and/or outlined to aid inspection. Some users like to see only a selected regions or two and mask others, (i.e. draw those cells in the background color). Other users prefer to draw bright outlines around selected regions to aid in the visual search. CorDex supports both methods and both have proved to be very useful for finding things.

4 Improved Rule Induction with CorDex

Another very useful discovery of CorDex is that once a user has grouped the items in to regions based on the map and visual searching for attribute commonality, the items in each region can define classes of items and given to rule induction systems such as C4.5[6]. With this more effective classification of the important attributes and elimination of irrelevant items, C4.5 can have much more success in discovering the rules that define the class. In our wafer yield experiment we found that once CorDex found the major clusters of low yield, we ran C4.5 and it found much better rules that did eventually explain the basis for the low yield problem. Runs of C4.5 without CorDex pre-classification could not find any significant rules because of all the conflicting and coincidental attributes buried in the entire data set.

This ability to use CorDex as a preprocessor for other well-known tools such as rule induction is a very important feature. It allows CorDex to be used in addition to rather than instead of existing tools. This fact should help in the acceptance of CorDex into areas without having replace tried and true techniques. Our experience also shows that interpretation of 2D SOMs and their attribute planes is very intuitive and easy to learn by a wide range of people, since visual pattern recognition is very natural.

5 Conclusions

CorDex has combined many graphic displays and conventional tools into a significant knowledge discovery tool. The attribute plane, regions and rule induction can be used to quickly scan thousands of items and relationships and narrow down to only the 'interesting' attributes. From that point a user can find simple relationships immediately or use rule induction to find more complex or hidden patterns. The Self-Organizing Feature Map is the key to sorting data items into data structure very suitable for visual searching of attributes. This has proved to be effective in several problems here at Motorola.

6 Acknowledgments

This work is part of ongoing research at Motorola and has received much help and encouragement from the members of the Phoenix Corporate Research Lab including Kari Torkkola, John Summers and others.

References

[1] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, 1995.

[2] C. Bishop, M. Svensen, and C. Williams, *GTM: A Principled Alternative to the Self-Organizing Map*. Neural Computer Research Group, Aston University, Birmingham B4 7ET, UK, April 4, 1996.

[3] T. Kohonen. *Self-Organizing Maps*, page 119. Springer-Verlag, 1995.

[4] S. Kaski, and T. Kohonen. Exploratory data analysis by the self-organizing map: Structures of welfare and poverty in the world. In *Apostolos: Neural Networks in Financial Engineering. Proceedings of the Third International Conference on Neural Networks in the Capital Markets*, London, England,. pages 498-507, World Scientific, Singapore, 1996.

[5] J. R. Quinlan. *C4.5 Programing f or machine Learnin*g. Morgan Kaufmann, 1993.