RC22517 (W0207-065) July 15, 2002
Computer Science

IBM Research Report

KMAP - A Visualizer for Kohonen Self-organizing Map Clustering Results

George S. Almasi, Richard D. Lawrence
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

KMAP - A Visualizer for Kohonen Self-organizing Map Clustering Results

George S. Almasi and Richard D. Lawrence
IBM T.J. Watson Research Center
P.O. Box 218
Yorktown Heights, New York 10598
almasi@us.ibm.com, ricklawr@us.ibm.com

August 17, 2002

Abstract

We describe a Java-based visualizer that preserves the topological information inherent in the results of cluster data mining that used the Kohonen self-organizing map approach. The visualizer provides a tool with a number of views and drilldowns that an analyst can use to circumvent the data interpretation bottleneck and quickly evaluate the results of the map produced by the datamining run. Being written in Java, it is portable across a large number of computing platforms.

Keywords: visualizer, data mining, Kohonen, self-organizing map, clustering, Java.

1 Introduction

The Kmap visualizer described in this paper was designed to allow rapid evaluation of the results of data mining performed using the Kohonen self-organizing map [8] clustering approach. A key feature of this approach is that, in addition to grouping similar data records into clusters, it places the clusters on a two-dimensional map in such a way that similar clusters are near each other. The nature of a cluster in a particular neighborhood of the map says something about that neighborhood. In other words, the contents of individual clusters and the topography of the cluster map are both of interest.

The current authors wrote a highly scalable parallel version of the self-organizing map algorithm [9] which is used by IBM's Intelligent Miner for Data [7] . This program was recently put to the test in a demonstration of mining very large databases [4], and the parallel speedup that was obtained was so good (over 14X for a 16-processor SP2) that the normal ratio of time needed to perform the datamining vs. interpreting its results became inverted – data analysis became the new bottleneck. The reason was that the existing visualizer was good at conveying the nature of individual clusters, but the map had to be constructed basically by hand. Kmap was designed to remedy this problem.

This paper is organized as follows: we first describe the design of the Kmap visualizer, its modes of operation and its drilldown capabilities, and its implementation in Java. This is followed by four case studies that show examples of Kmap's features. We then present illustrative examples of Kmap's features in the form of four case studies using stock market, retail supermarket, credit bureau, and census data.

2 Implementation

The main Java classes in Kmap reflect the program's functionality:

- **Kmap** collects the clustering results from the input file (generic neural weights file or Intelligent Miner results file) and creates and manipulates the rectangular grid map of clusters.
- **Kdetail** handles the drilldowns into an individual cluster
- **Kplot** creates the contents of the rectangles for Kmap and Kdetail, using the JClass chart and table java components provided by KL Group Inc. [5].
- **Kmenubar** handles user interactions via the menu bar that is used for both Kmap and Kdetail.
- **NNClustRes** provides an API to Intelligent Miner results files in the form of a dozen "get" methods for obtaining
    1. the number of clusters (neural nodes) and data attributes (fields),
    2. the neural weights (mean clustering attributes for each node), and
    3. several sets of attribute statistics, used for drilldowns and described further below.

There are three different "modes" for viewing clustering results in Kmap and its drilldowns:

- "weights" mode, in which a plot of all the neural weights (i.e., attribute mean valuse) is shown for each cluster. This mode is useful when the attributes consist of a logical sequence like month-by-month prices or spending. Figure 1 shows such an example.
- "topten" mode, in which the names of the attributes with the top 3 or top 10 weights are shown for each cluster. This mode can be very useful when the number of attributes is large. Figure 3 shows a comparison of results viewed in weights and in topten modes, and Figure 4 gives a detailed example of topten mode in action.
- "sizes" mode, which simply shows the size of each cluster as a filled circle. Figure 3 contains an example of a "sizes" view.

The mode can be selected from the “View” menu, or by specifying w, t, or s after the filename when Kmap is called from the command line.

The cluster size is indicated by the background color in weights mode, by the grayness of the frame in topten mode, and by the size of the circle in sizes mode.

The available supplemental attributes, such as Total Spending or Age, can also be used to color the inner rectangle in topten mode or the background in weights or sizes mode. This is done via the “Color By” pulldown on the menu bar. The functions provided by the other menu bar pulldowns are as follows:

- “File” handles the opening of files and the printing of the displays.
- “View” provides a large number of methods for controlling the appearance of the visualization.
- “Sort By” allows sorting the attributes in topten mode by weights or by six other quantities.

The attributes used in neural clustering are often normalized first, and so the neural weights (the means of the normalized attributes) for each cluster can be different from the means of the raw values. The un-normalized values of the attributes can be included as supplemental fields, and Intelligent Miner will compute their statistics along with those of the active fields. Kmap has provision to read these supplemental field statistics and make them available for alternative ways to sort the attributes in topten mode, using the ”Sort By” menu. These quantities available for each cluster are:

1. “mean”, the mean value of the un-normalized attribute
2. “partcp”, the fraction of records with non-zero values of this attribute
3. “meanNZ”, the mean non-zero value of the un-normalized attribute, computed from the ratio of the two quantities above
4. “Rmean”, the mean value of the un-normalized attribute for this cluster divided by the mean value of this attribute for the entire data set.
5. “Rpartcp”, the quantity “partcp” normalized to the entire data set as above
6. “RmeanNZ”, the quantity “meanNZ” normalized to the entire data set as above

There are also popup menus and other panels available for drilldown and other purposes; these are shown in the examples that follow.

3 Case Studies

This section describes the use of Kmap to visualize several different kinds of datamining results.

- Standard and Poor 500 monthly closings (weights, drilldowns)
- Safeway (topten mode, more drilldowns)
- Credit Bureau(bankruptcy colorby)
- Census (categorical variables)

3.1 Stock Prices

For our first example, Figure 1 shows Kmap in "weights" mode, displaying the results of a 64-node clustering performed on the monthly closing prices of the Standard & Poor's stocks for the period September 1995 - August 1996. Each stock price is normalized by its maximum value during this period. The action of the Kohonen algorithm in placing similar clusters near each other on the map shows clearly in the gradual transition from gainers on the left to decliners on the right.

Kmap offers two special drilldowns for relatively small datasets like this one. It can read a file listing the cluster for each record and display the membership of each cluster on a popup menu, as shown in Figure 1. In addition, the "View" menu has an entry that leads to a "Record Finder" panel that can find which cluster a particular record is in and highlight that cluster, as shown for the case of Western Atlas and cluster 25 in the figure. A hypothetical scenario for using these capabilities would be to check the membership of a gainer cluster like node 32, find that it contains two petroleum industry members (Mobil and Texaco), wonder where Western Atlas (also a petroleum industry member) had been placed, use the Record Finder, and discover it in cluster 25, a direct neighbor of cluster 32. There is a substantial number of such intuitively satisfying relationships in the map.

If the cluster membership file mentioned above also contains the attribute values for the individual records, Kmap can provide an additional drilldown that allows comparing the behavior of individual records in a cluster, as shown in Figure 2.

3.2 Supermarket Shoppers

The example in this section explains the motivation for "topten" mode and describes some of the features of that mode. The clustering in this case was performed on a month's spending data by some 36,000 supermarket customers in about 100 product categories (dairy, confectionery, baby products, etc.). As shown in Figure 3, the Kmap "weights" mode used in the previous example can be used to view these clustering results as well. Kmap provides point labels to help identify each point even in a crowded plot, and this feature can be used to identify the three attributes with the largest weights in cluster 3 as Sugar, Tea, and Desserts/Puddings, for example. But the large number of attributes makes this a slow process. Furthermore, unlike the months of the previous example, there is no sequential relationship among the attributes that makes a plot view particularly useful. A much faster grasp of the clusters' nature can be obtained in this case from an initial map that shows what the top 3 attributes are for each cluster, with a rich set of available drilldowns for further investigation. This is what "topten" mode does.

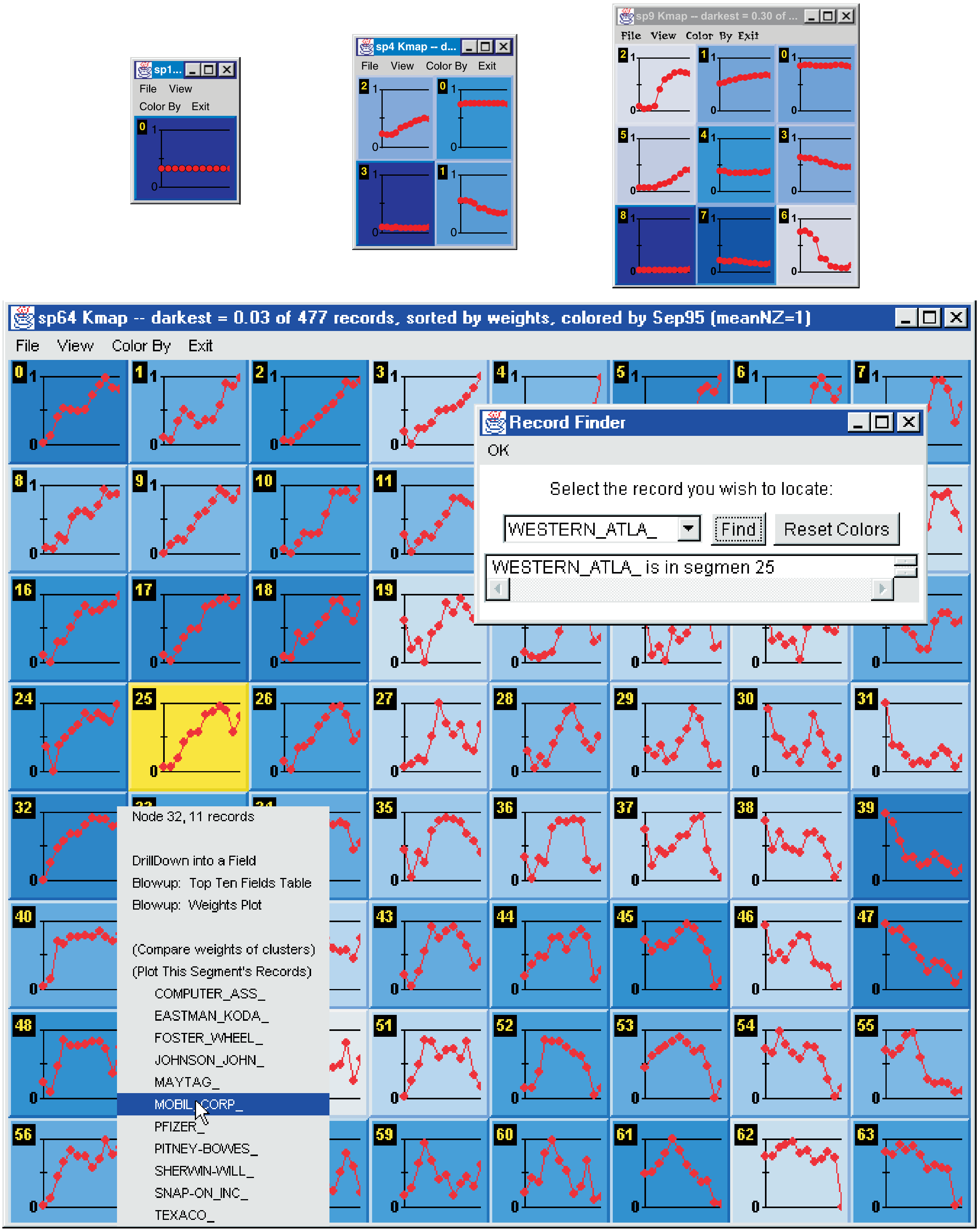


Figure 1: Monthly closing prices for the Standard & Poor's 500 stocks over a one-year period

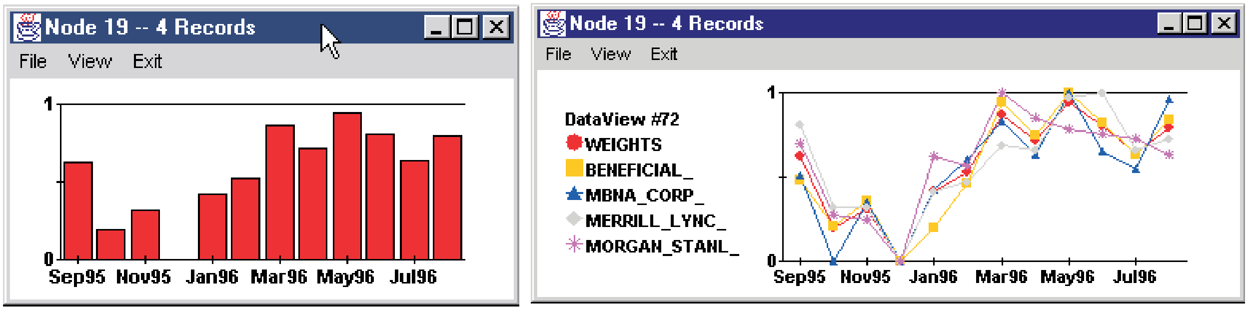


Figure 2: Drilldowns available in weights mode (for example in Figure 1)

In topten mode, each rectangle in the map conveys the following information about its corresponding cluster: the outer frame's degree of grayness is proportional to its population. the inner rectangle's color indicates the value of the supplemental attribute chosen from the "Color By" menu. In Figure 3, for example, blue indicates above-average customer Age, while pink denotes below-average attribute values. The numerical values of the population and Age are also shown for each cluster.

By default, the top 3 attributes are sorted by their neural weights, but this can be changed (for the main map as well as the drilldowns) via the "Sort By" menu, which allows re-sorting by the attribute mean value or the five other statistics mentioned earlier. The color of the bullets in this example correspond to a rainbow spectrum matched to the range of category numbers, so attributes with similar colors have similar category numbers. The bullet shapes carry no information in this example. However, for the wine recommender described in [3], we used a different scheme in which the bullet color indicated the wine color and the bullet shape indicated price (squares cost more than circles, etc.).

Clicking on a cluster's rectangle pops up a menu that can be used for further drilldown, as shown in Figure 3. The first item on the menu allows getting more information on a particular attribute (field), such as the distribution of its revenue among the clusters. For example, the 11% of the total customer population represented by cluster 3 accounted for 24% of the spending on Tea, for a "wallet share enrichment" of 2.18 – i.e., chances of finding a tea buyer in this cluster are more than twice that for the overall customer population.

The second item on the drilldown menu allows getting more detail on a particular cluster. Clicking it brings up a scrollable table showing the neural weights and other field statistics for all the attributes. For convenience, the distribution map for a given attribute can be brought up by clicking the attribute in this table.

The third item on the drilldown menu brings up an enlarged plot of the neural weights for that cluster, as shown in the figure.

The third Kmap view, "sizes", is shown at the top left of Figure 3 for reference. In this mode, the cluster population is conveyed by the size of the corresponding circle (the populations are very close in this case), and the color convention is the same as for the "topten" view.

The drilldowns shown are available in all 3 modes, and each drilldown figure can be printed individually.

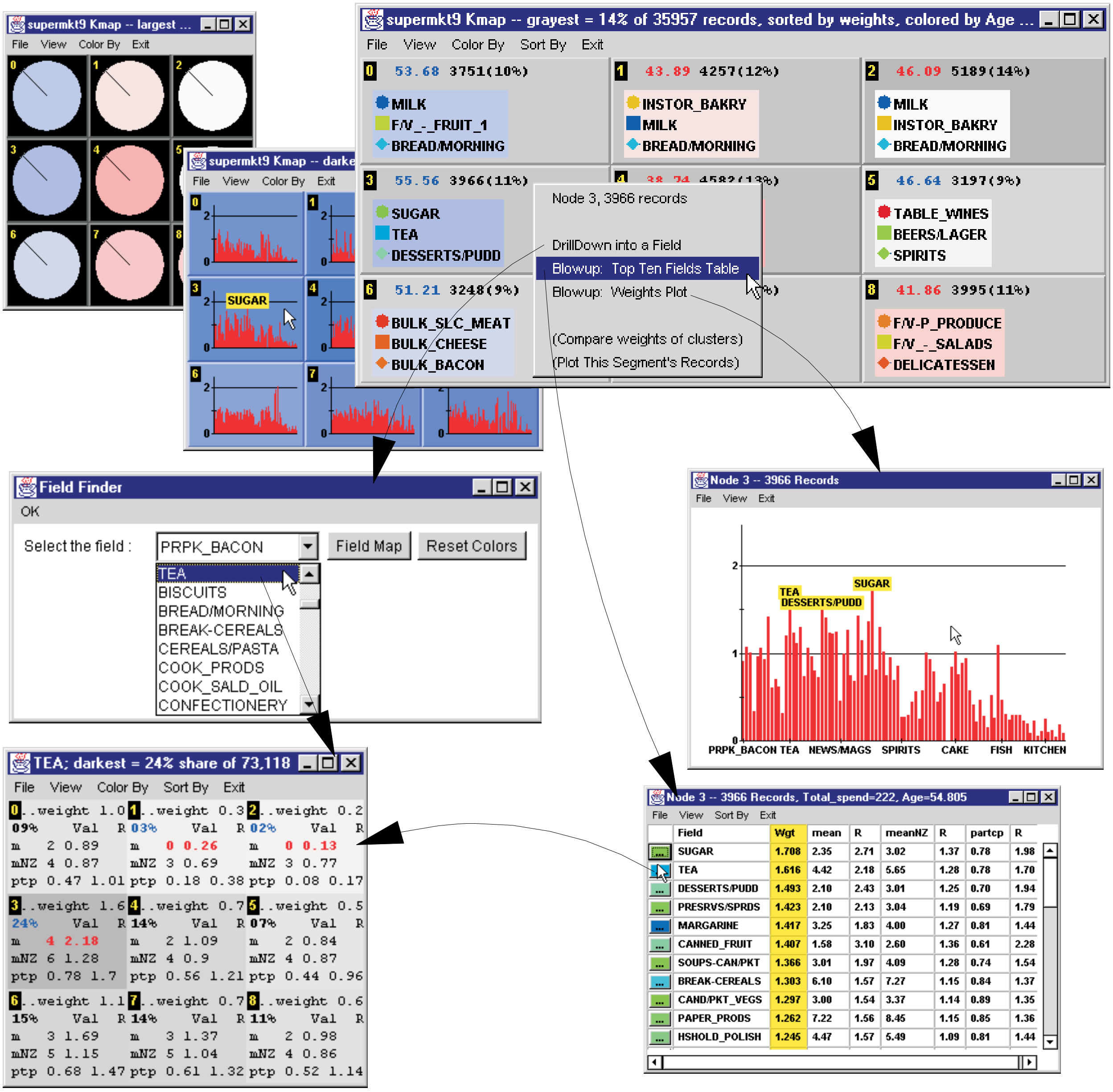


Figure 3: Comparison of views and drilldowns. Starting at top left, the “sizes”, “weights”, and “topten” views of the supermarket shopper clustering results for 9 clusters. The top right view also shows popup menu leading to several drilldowns: into a specific attribute (left center and right), into the weights view of a cluster (right center), and into the topten view of a cluster (bottom right). Clicking an attribute’s button in the view at bottom right is a second way to obtain the attribute map at bottom left.

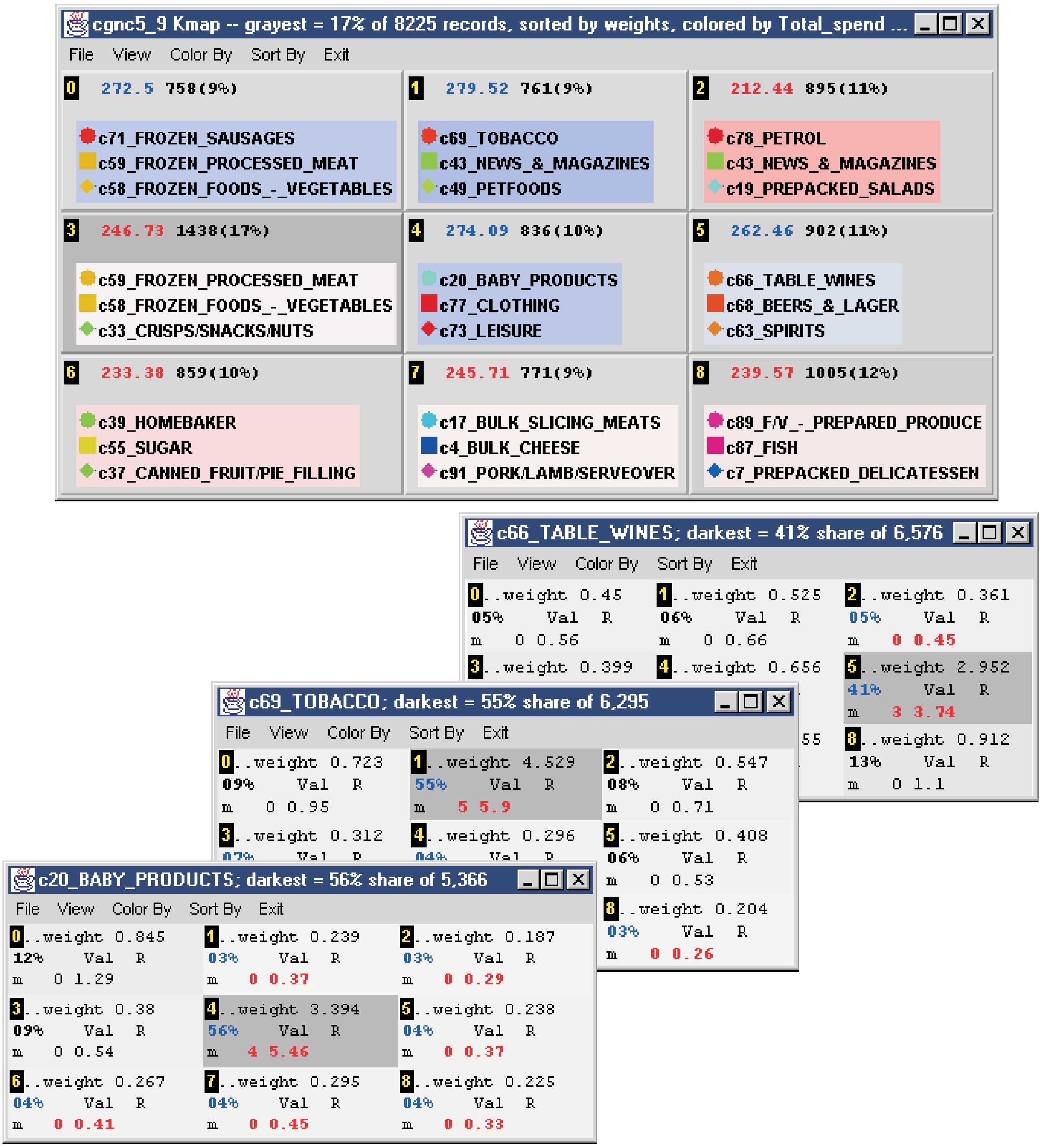


Figure 4: Supermarket retail data clustering results

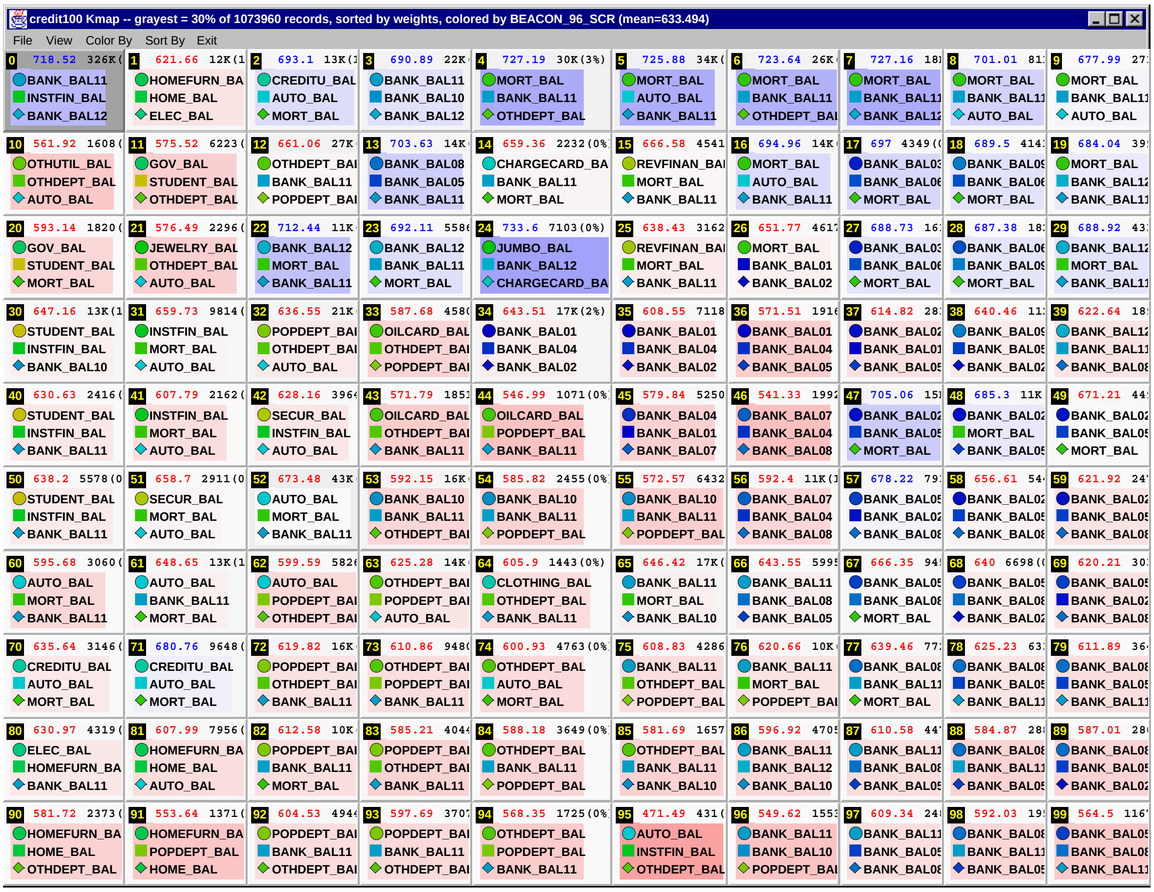
One application of this clustering capability is in the generation of personalized recommendations, as we describe in [2]. These recommendations were being generated for participants in a trial program using PDAs(personal digital assistants) for supermarket shopping. Since all participants had above-average spending [1], we clustered only the above-average spenders and obtained the results shown in Figure 4. Note the emergence of a cluster (4) with strong spending on baby products. As the three drilldown maps of attribute share show, these clusters are sharper than the previous ones in that the "wallet share enrichments" are larger. For example, cluster 4 has 10% of the total population but 56% of the total spending on baby products, for a an enrichment of 5.6. This sort of enrichment can be quite valuable for customer relationship and marketing purposes. It also impacted the generation of recommendations. For example, the most popular chocolates in cluster 4 turn out to be quite different from those most frequently preferred by the population as a whole. [2].

3.3 Credit Data

The example in this section shows the value of adroitness in the selection of the attributes used for clustering. Figure 5 shows clustering results from a study on mining a large database from a credit bureau. Twelve of the 44 attributes that were used were created by categorizing each outstanding credit card balance into a small matrix of several ranges of balance and several ranges of the age of the account. This helped the domain expert to find about a dozen different segments on the map. For example, the top attributes for clusters 18 and 28 are "BANK_BAL03" and "BANK_BAL06" , which both correspond to large balances in accounts less than 6 months old, hence the analyst's label "BIG CHURN" for people who continually accept bank offers for new credit cards with low initial interest rates, and then transfer to another card at the end of the low interest rate period. This ability to discern various segments of credit usage can be very helpful to a bank's efforts to market its various products.

The second example in this section shows the clustering results on a subset of the population whose credit rating is within a "gray area" that normally makes it difficult to obtain credit. One of the supplemental attributes (not used in the clustering, of course) was whether an individual in this subset actually went bankrupt within the next six months. When this attribute is used to color the cluster map (Figure 6) , an interesting pattern appears: far from being randomly distributed, the actual bankruptcies are concentrated on the right edge of the map. The bankruptcy rate for the population as a whole is about 0.5%, and most of the clusters have a rate below this value, while one of the clusters at the right of the map has an 8% bankruptcy rate, which is 16 times higher than the average value. The dominant attributes of this cluster are all old accounts with very large balances, which tend to be prominent in the other clusters with above-average bankruptcies as well. The suggestion is that a number of people are being denied credit without good reason, and that a more refined rating system could free up credit to a group that could probably really use it. Customers and the bank would both benefit.

[1] like the children in Lake Wobegon



LOW			Medium OLD	Stodgy Mortgage	Stodgy Mortgage	Stodgy Mortgage	Stodgy Mortgage	Stodgy Mortgage	Stodgy Mortgage
	STUDENT	Shoppers	Medium 1-2 year	Stodgy Mortgage	Stodgy Mortgage	AUTO	**BIG** CHURN	Stodgy Mortgage	Stodgy Mortgage
STUDENT				**JUMBO**	Stodgy Mortgage	Little Churn	**BIG** CHURN	Small Business	Small Business
GOVMT			OIL	Little Churn	Little Churn	Little Churn	MEDIUM CHURN		
GOVMT		Popular Dept	OIL	OIL	Little Churn		MEDIUM CHURN	MEDIUM CHURN	MEDIUM CHURN
GOVMT		AUTO					MEDIUM CHURN	MEDIUM CHURN	MEDIUM CHURN
AUTO	AUTO	AUTO	Serious Shoppers		Medium OLD	Medium OLD	Medium 6-12 months	Medium 6-12 months	Medium 6-12 months
AUTO	AUTO	Popular Dept	Serious Shoppers	Serious Shoppers	Ser Shop / Medium OLD	Medium OLD	Medium 1-2 year	Medium 1-2 year	Medium 1-2 year
		Popular Dept	Serious Shoppers	Serious Shoppers	Serious Shoppers	Medium OLD	Medium OLD	Medium 1-2 year	Medium 1-2 year
		Popular Dept		Serious Shoppers	AUTO	Medium OLD	Medium OLD	Medium 1-2 year	Medium 6-12 months

Figure 5: Credit usage customer segments

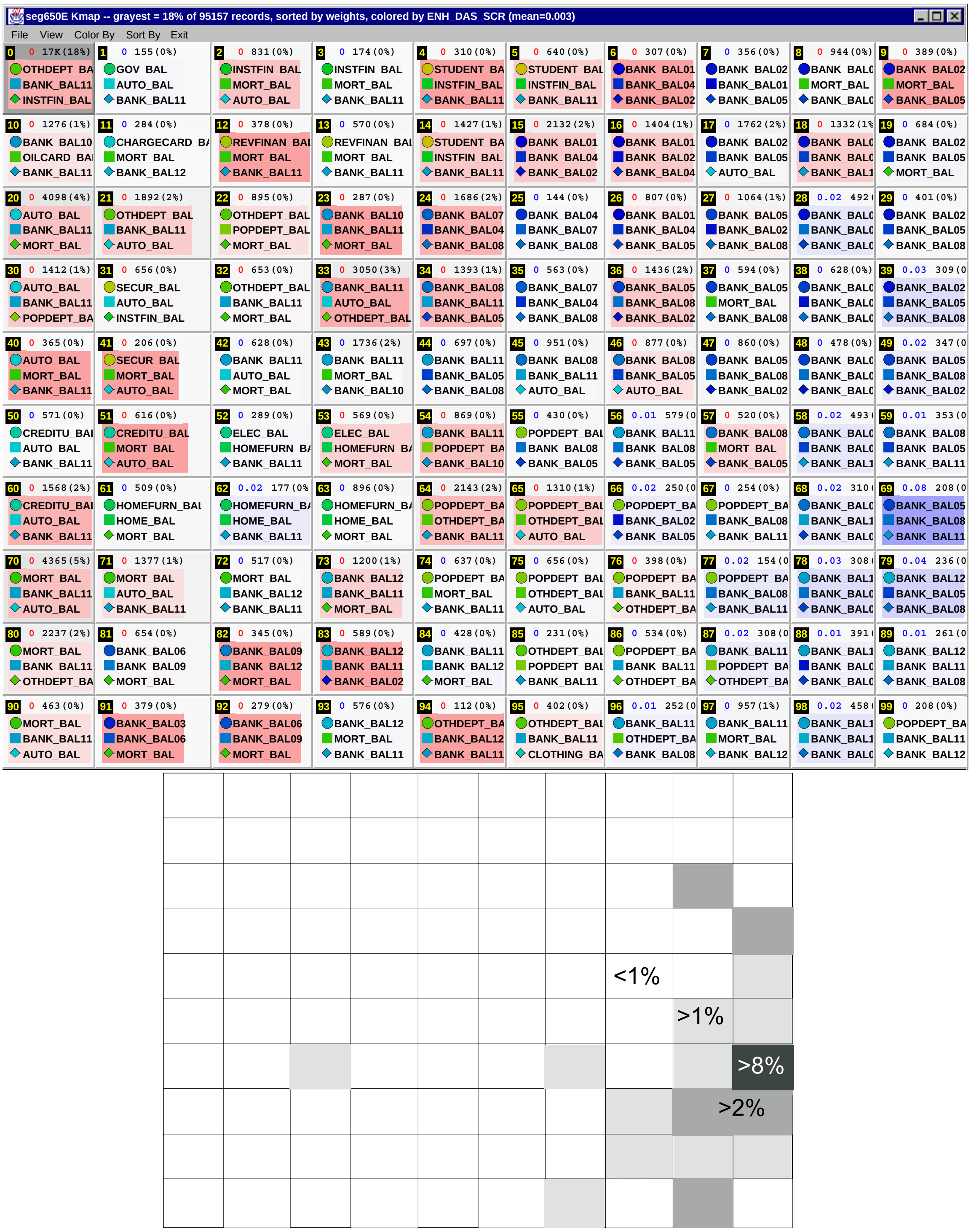


Figure 6: Actual bankruptcies among customers rated as marginal credit risks.

Figure 7: Actual bankruptcies among customers rated as marginal credit risks.

3.4 Census Data

The example in this section shows how Kmap can handle categorical values. The clustering was performed using publicly available census data with a mixture of 12 continuous and categorical attributes for each person. The data comes from a municipality located near a National Laboratory, which accounts for the unusually large number of advanced degrees and large diversity of backgrounds.

Categorical attributes pose an extra degree of complexity in neural clustering because they are expanded to a one-to-N mapping: for example, if the original attribute were "animal type" with values "cat, dog, cow", it would be replaced for clustering purposes by three attributes "is_a_cat", "is_a_dog", "is_a_cow" that can each assume values 1 or 0. So the total number of attributes actually used in the clustering is data-dependent.

Kmap handles this by making two passes through the results file, obtaining the number of distinct values for each categorical attribute on the first pass and constructing the effective attributes coming from each original attribute on the second pass. Sample Kmap results are shown in Figure 9 and are compared to the default Intelligent Miner visualizer's view in Figure 8. The two views each present interesting information in different ways, but they are indeed consistent with each other. For example, one of the supplemental attributes (i.e., not used for clustering) is whether household income is above or below $50,000. Using this attribute to color the Kmap in Figure 9 points to clusters 18 and 24 as being highest in this attribute, with 70% of the households in cluster 24 having income above $50,000. This agrees with the corresponding pie chart in Figure 8.

One of the original categorical attributes was "education", and one of its values was "doctorate". Drilling down into the effective categorical attribute "Doctorate(education)" (or "has_a_PhD" in our earlier terminology) in Figure 9 (comfortingly, perhaps) an overlap between clusters with high PhD populations and clusters with high income. Drilling down into the "educ_num" attribute, a number proportional to the number of years of education, shows similar results. Both results are consistent with the drilldowns in Figure 8. Kmap does a good job showing the attribute distributions on the map. The default visualizer does a good job showing the categorical attributes in pie charts. Both can be useful.

3.5 Appendix: A Tale of Two Clusters

Figure 10 offers a more detailed look at the kind of information that Kmap can extract from the supermarket customer clustering results mentioned earlier.

4 Summary

Kmap has proved to be a versatile visualizer for Kohonen self-organizing map clustering results obtained from a variety of datamining engagements. Analysts value being able to see the map topography, as well as the drilldowns provided, many of which resulted from their suggestions. We provided examples of Kmap's use in four different areas of commerce. Currently we are applying Kmap to technical data generated by the simulators and hardware control systems of massively parallel supercomputers.

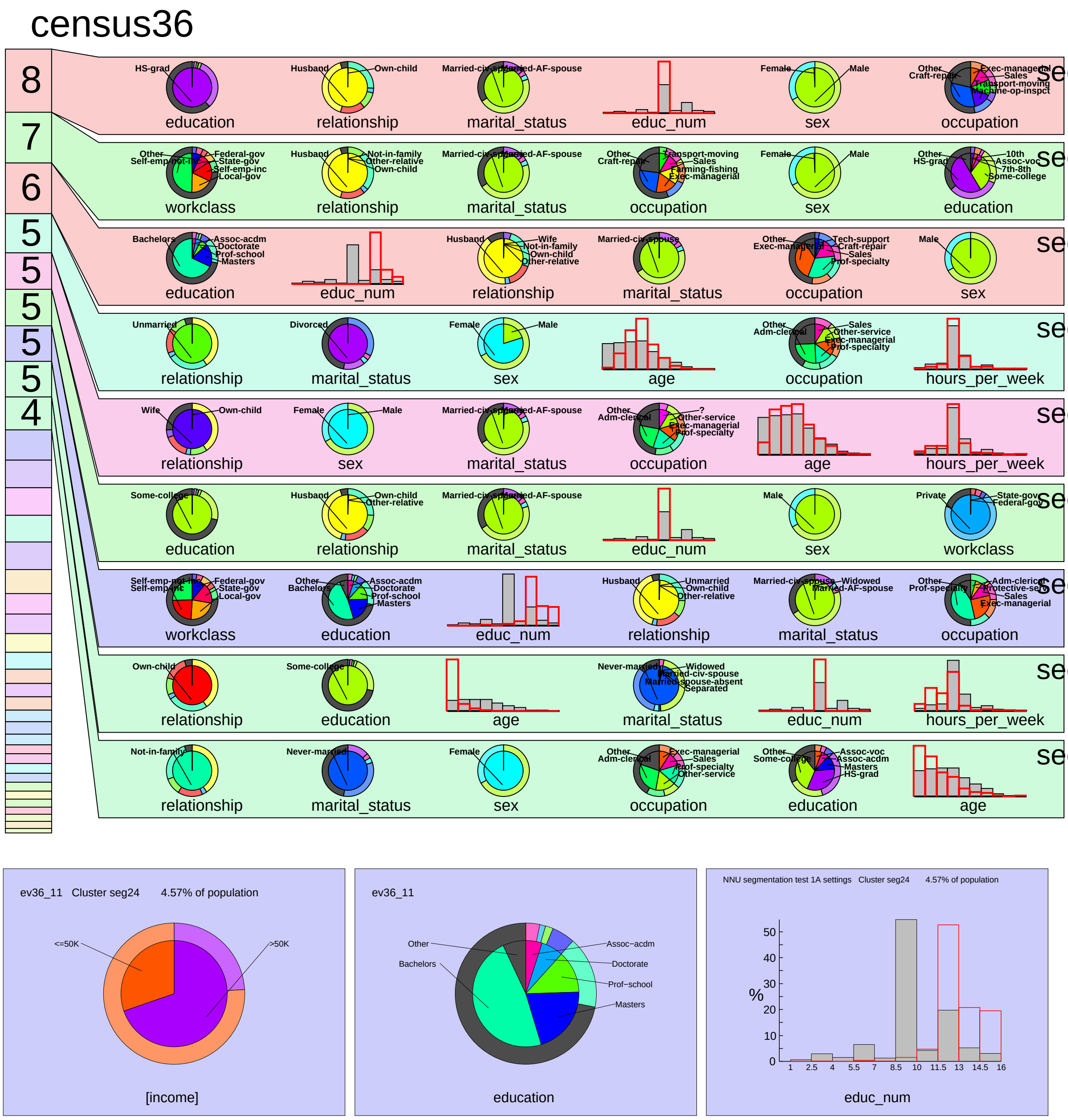


Figure 8: Census data clustering results seen with the Intelligent Miner default visualizer

Figure 9: Census data clustering results seen with Kmap

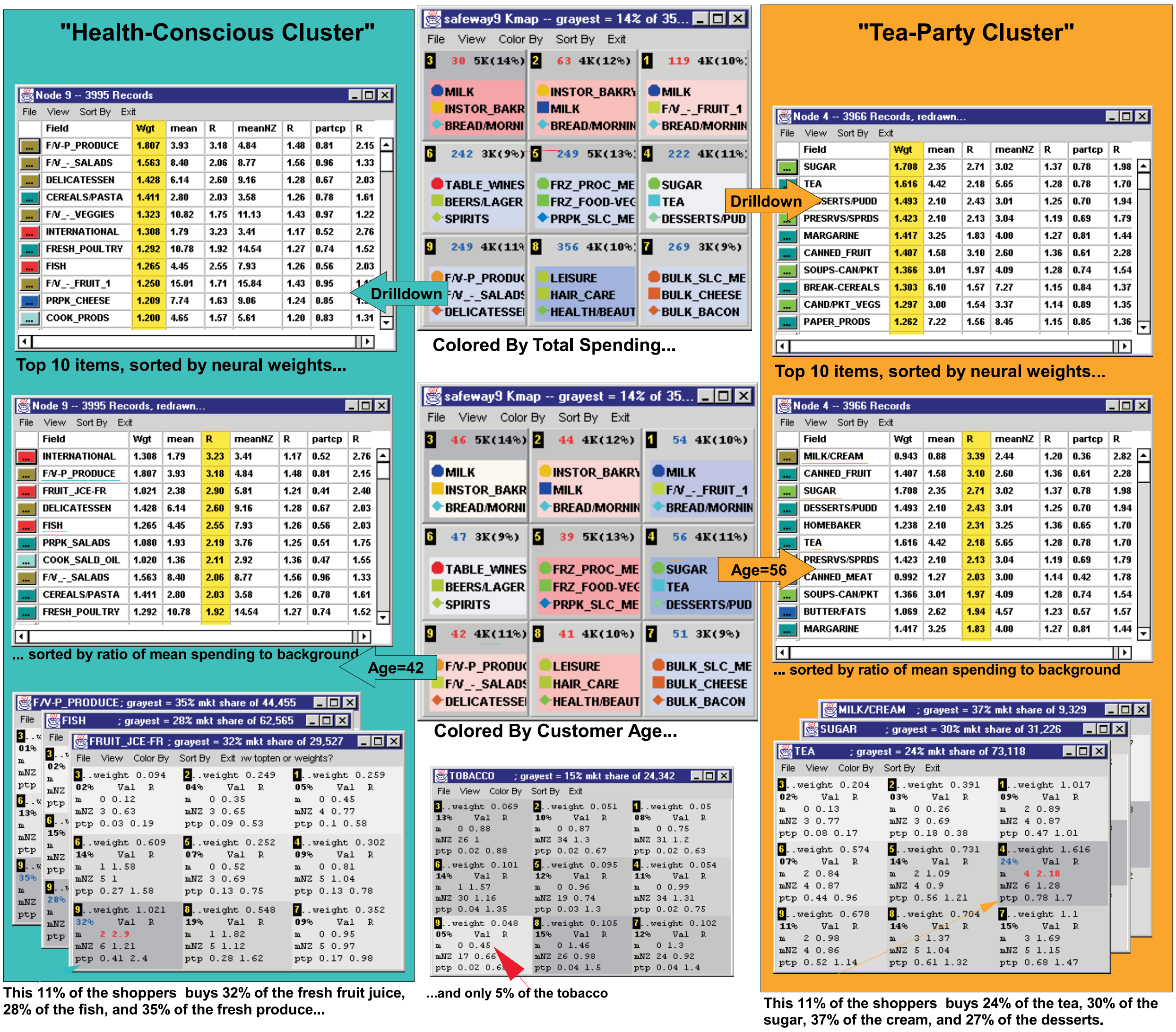


Figure 10: A Tale of Two Clusters: rather different lifestyles emerge during examination of the supermarket shopper clustering results. Cluster 4 seems to be an older group of serious homebakers likely to be found having a tea party, whereas cluster 9 seems to be a younger, health-conscious group more likely to be found jogging - and not smoking!

5 Acknowledgements

We are grateful for many helpful discussions with our colleagues Joseph Bigus and Michael Rothman.

References

[1] Kotlyar V., Viveros M.S., Duri S.S., Lawrence R.D., Almasi G.S. "A Case Study in Information Delivery to Mass Retail Markets" In the Proceedings of the 10th International Conference on Database and Expert Systems Applications (DEXA), Florence, Italy, August/September 1999, pp. 842-851. Published by Springer-Verlag as Lecture Notes in Computer Science, vol 1677

[2] R. D. Lawrence, G. S. Almasi, V. Kotlyar, M. S. Viveros, and S. S. Duri , "Personalization of Product Recomendations in Mass Retail Markets", *Data Mining and Discovery 3*(2): 171-195 (1999).

[3] G. S. Almasi and A. J. Lee, "PDA-based Personalized Recommender Agent", Proceedings of the 5th International Conference on the Practical Application of Intelligent Agent and Multi Agent Technology (PAAM 2000), Manchester, England, April 2000, pp. 299-309.

[4] H. Edelstein, "Mining Large Databases: A Case Study", *www.twocrows.com/largedb.pdf*

[5] "JClass Java Components", *www.klgroup.com*

[6] G. Almasi, "Kmap Demo", *www.gsalmasi.com*

[7] Intelligent Miner for Data, *www.ibm.com/software/data/iminer/fordata*

[8] T. Kohonen, "Self-Organizing Maps", Springer-Verlag, 1995.

[9] R. D. Lawrence, G. S. Almasi, and H. E. Rushmeier, "A Scalable Parallel Algorithm for Self-Organizing Maps with Applications to Sparse Data Mining Problems", *Data Mining and Knowledge Discovery*, Vol. 3, 1999, pp. 171-195. (U. S. Patent 6,260,036, July 10, 2001.)

[10] J. C. Bezdek and N. R. Pal, "Some New Indexes of Cluster Validity", *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics* Vol. 28, No. 3, June 1998, pp. 301 - 315.