Mapping Nominal Values to Numbers for Effective Visualization*

Geraldine E. Rosario, Elke A. Rundensteiner, David C. Brown and Matthew O. Ward †
Computer Science Department, Worcester Polytechnic Institute

Abstract

Data sets with a large number of nominal variables, some with high cardinality, are becoming increasingly common and need to be explored. Unfortunately, most existing visual exploration displays are designed to handle numeric variables only. When importing data sets with nominal values into such visualization tools, most solutions to date are rather simplistic. Often, techniques that map nominal values to numbers do not assign order or spacing among the values in a manner that conveys semantic relationships. Moreover, displays designed for nominal variables usually cannot handle high cardinality variables well. This paper addresses the problem of how to display nominal variables in general-purpose visual exploration tools designed for numeric variables. Specifically, we investigate (1) how to assign order and spacing among the nominal values, and (2) how to reduce the number of distinct values to display. We propose that nominal variables be pre-processed using a *Distance-Quantification-Classing (DQC) approach* before being imported into a visual exploration tool. In the Distance Step, we identify a set of independent dimensions that can be used to calculate the distance between nominal values. In the Quantification Step, we use the independent dimensions and the distance information to assign order and spacing among the nominal values. In the Classing Step, we use results from the previous steps to determine which values within a variable are similar to each other and thus can be grouped together. Each step in the DQC approach can be accomplished by a variety of techniques. We extended the XmdvTool package to incorporate this approach. We evaluated our approach on several data sets using a variety of evaluation measures.

CR Categories: G.3 [Mathematics of Computing]: Probability and Statistics—Contingency Table Analysis; I.5.3 [Pattern Recognition]: Clustering—Similarity Measures D.2.12 [Software Engineering]: Interoperability—Data Mapping E.4 [Data]: Coding and Information Theory—Data Compaction and Compression

Keywords: nominal data, visualization, dimension reduction, correspondence analysis, quantification, clustering, classing.

1 Introduction

Nominal (or categorical) variables are variables whose values do not have a natural ordering or distance. High cardinality nominal variables (i.e., those with a large number of distinct values) are common in real-world data sets. Examples of high cardinality nominal variables include product codes and species names.

Visualization provides an efficient and interactive way of exploring high dimensional data [Ward 1994]. Unfortunately, nominal variables, especially high cardinality nominal variables, pose a serious challenge for data visualization tool developers. Difficulties arise due to several reasons.

First, visualization methods specifically designed for nominal data are not as commonly used as those designed for numeric data [Friendly 1999]. Possible reasons include: (1) They tend to be more special-purpose (e.g., Mosaic Displays [Friendly 1999] are designed for discovering associations whereas Parallel Coordinates [Inselberg and Dimsdale 1990], which are for numeric variables, can be used for exploring outliers, clusters, and associations). (2) Methods such as the Fourfold Display [Friendly 1999] cannot handle multiple nominal variables. (3) Methods such as the Mosaic Display cannot handle high cardinality variables well. (4) Most methods are not readily available in common visualization software [Friendly 1999].

Second, most visualization software packages only provide displays that are designed for numeric variables. Reasons for this include: (1) Data sets have traditionally contained only numeric data. (2) Numeric displays are more general-purpose. (3) The inherent order and spacing among numeric values makes it natural to convey notions such as magnitude and similarity.

One way to display nominal variables using numeric displays is to map the nominal values to numbers, i.e., assigning order and spacing to the nominal values. Display methods such as Parallel Coordinates (Figure 1) require both order and spacing among values. But care must be taken. Blindly casting nominal values into numeric displays may introduce artificial patterns and cause errors in the interpretation of the visualization. Existing nominal-to-numeric mapping techniques do not always assign both order and spacing to the values. For example, [Ma and Hellerstein 1999]'s technique only assigns order to the nominal values, but not spacing. As a motivating example of the need for order and spacing, refer to Figures 1 and 2 which both display the quality, color and size information of 6550 objects (from a synthetic data set). Figure 1 gives an example of a display where nominal values were assigned order and spacing using our DQC approach, whereas Figure 2 shows alphabetical ordering and uniform spacing of the nominal values. Figure 1 reveals that blue and purple objects have similar underlying distributions for quality and size. Such information is difficult to extract from Figure 2.

This paper addresses the problem of how to display data sets with a large number of nominal variables, some with high cardinality, in visual exploration tools designed for numeric variables. Specifically, we address two sub-problems:

- How do we map nominal values to numbers such that we effectively assign order and distance among the values? Order is used to position values along an axis, where the adjacency of values suggests similarity. Distance is used to space the values along that axis. The amount of spacing suggests the degree of similarity among values, making it easier to spot clusters as well as outliers.
- When a variable has many values, how do we group similar values together to reduce the number of distinct values to dis-

*This work is supported by NSF grant IIS-0119276
†e-mail: {ger,rundenst,dcb,matt}@cs.wpi.edu

IEEE Symposium on Information Visualization 2003,
October 19-21, 2003, Seattle, Washington, USA
0-7803-8154-8/03/$17.00 ©2003 IEEE

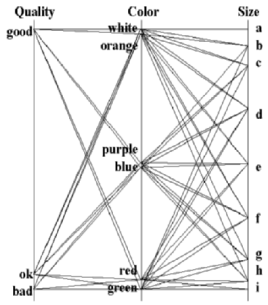

Figure 1: Parallel Coordinates with FCA Quantification.

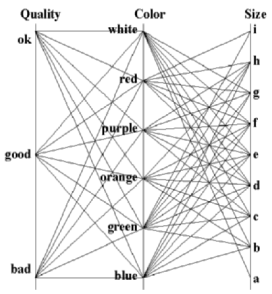

Figure 2: Parallel Coordinates with Arbitrary Quantification.

play? Reducing the cardinality is needed for displays such as Dimensional Stacking [LeBlanc et al. 1990] and Trellis Displays [Becker et al. 1997] which are limited by the number of values they can display.

We also want our solution to have the following features: *data-driven* (not relying on domain knowledge), *multivariate* (using the relationship of a nominal variable with *several* other variables to decide the ordering, spacing and classing of the values), *scalable* (can work with a large number of variables with high cardinality using limited memory), *distance-preserving* (the distance between two nominal values in nominal space is preserved in numeric space), *association-preserving* (nominal variables that are highly associated in nominal space are also highly correlated in numeric space), and *accessible* (readily available to data analysts). To our knowledge, no solution exists that has all these features (this is further discussed in Section 2).

To solve this problem, we propose that nominal variables be pre-processed using a *Distance-Quantification-Classing (DQC) approach* before being imported into visual exploration tools designed for numeric variables. In the *Distance Step*, we transform the data and search for a set of independent dimensions that can be used to calculate the distance between nominal values. This distance is based on each value's distribution across several other nominal variables. In the *Quantification Step*, we assign order and spacing among the nominal values based on the distance information. In the *Classing Step*, we determine which values within a variable are similar to each other and thus can be grouped together. Each of these three steps can be accomplished by more than one technique as we will show in Sections 4 to 6.

We implemented the DQC approach in XmdvTool, a public-domain visualization package developed at WPI [XmdvTool Home Page 2003]. For the Distance Step, we implemented and evaluated two alternatives – the well-established technique of Multiple Correspondence Analysis (MCA) [Greenacre 1993] from Statistics and our own Focused Correspondence Analysis (FCA) which we describe in this paper. FCA is our proposed alternative to MCA when memory is limited. For the Quantification Step, we used a modification of the Optimal Scaling technique [Greenacre 1993] to also make it work for data sets with perfectly associated variables. For the Classing Step, we used a Hierarchical Clustering algorithm [Johnson and Wichern 1988] so we can perform *multivariate classing* (using information from several variables to guide the classing).

To test our ideas, we pre-processed several data sets using the DQC approach and used numeric displays such as Parallel Coordinates to evaluate the usefulness of the quantified versions of the nominal variables. We compared MCA, FCA and arbitrary quantification using a wide range of evaluation measures such as time, memory, quality of quantification, quality of classing, and quality of visual display.

2 Related Work

Visualizing Nominal Variables: Several approaches to visualizing nominal variables exist. One can use displays that are specifically designed for nominal variables: sieve diagrams [Friendly 1999], mosaic displays [Friendly 1999], Correspondence Analysis maps [Greenacre 1993], fourfold displays [Friendly 1999], treemaps [Kolatch and Weinstein 2001], dimensional stacking [LeBlanc et al. 1990] and CatTrees [Kolatch and Weinstein 2001]. Unfortunately, these approaches are either special-purpose, not readily available in common data analysis software [Friendly 1999], or cannot handle high cardinality nominal variables well.

Others have mapped nominal values to numbers using some ordering technique and equal spacing between values, and then displayed them using numeric displays. Ordering techniques range from arbitrary ordering (e.g., alphabetical order), ordering based on the value of another variable [Ward 1994] (e.g., time), ordering based on domain expertise [Ma and Hellerstein 1999], to more intelligent ordering techniques (e.g., via natural clusters [Ma and Hellerstein 1999], using the spectral method [Beygelzimer et al. 2001]). Unfortunately, arbitrary ordering often creates artificial patterns which can lead to wrong conclusions. Furthermore, equal spacing does not convey the degree of similarity between nominal values.

Correspondence Analysis: Several research efforts on Correspondence Analysis (CA) have provided ideas for our research. [Friendly 1992] suggested using the coordinates from the first CA principal axis to order the values of nominal variables in mosaic displays to reveal the pattern of association. [Greenacre 1993] proposed using the coordinates from the first CA principal axis as input to create a classing tree. In this tree, the nominal values are grouped together using reduction in inertia to represent loss of information. [Greenacre 1993] also suggested the use of quantified versions of nominal variables as input to statistical techniques that require numeric variables such as regression. The SPSS Categories package uses CA to pre-process data for their Categorical Regression module and uses CA maps for visualizing nominal variables [Meulman and Heiser 2000]. These uses of the coordinates of the first CA principal axis seem to be due to the theory of Optimal Scaling, that states that these coordinates provide an optimal numeric representation of the nominal values [Greenacre 1993]. Unfortunately, when the nominal variable is perfectly associated with another nominal variable, such coordinates are not optimal, as we will show later.

[Milanese et al. 1996] used CA and clustering to group similar images and created a hierarchical tree for use in fast indexing into classes of images. This is similar to our approach in that we also use CA as a data reduction technique and use clustering to group similar nominal values together.

Classing: There are several approaches to grouping similar nominal values together. One could use expert knowledge but this can be tedious for high cardinality nominal variables. One could use information about the nominal variable itself (e.g., based on the frequency of occurrence of the values, the values can be grouped into popular, common or rare values). Or, one could use the relationship of the nominal variable with a target classification or regression variable [Micci-Barreca 2001] (e.g., group cities based on income level). But using only one specific variable to guide the classing (*bivariate classing*) may result in a classing that is believable only within the context of that specific variable (e.g., if we group cities based on income level alone, we may have to regroup cities if we want to visualize their relationship with land area). A better classing approach is to use several variables to guide the classing of a target variable (*multivariate classing*). One multivariate classing approach applies Clustering [Johnson and Wichern 1988] on a data

set where the records represent the nominal values and the variables contain summary information about each nominal value. We use this clustering approach for our Classing Step (Section 6).

3 Overview of Proposed Approach

Our proposed approach, the Distance-Quantification-Classing approach, consists of three steps (Figure 3). Each step can be accomplished by more than one technique. In this section, we describe the input, output and purpose of each step. In the succeeding sections, we discuss possible techniques for each step.

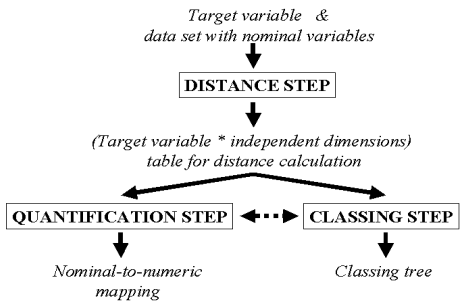

Figure 3: DQC Approach

Step 1: Distance Step – Given a data set with nominal variables, one of which is the nominal variable to be quantified and classed. The purpose of this step is to create a table where the rows represent the values of the nominal variable and the columns represent information about the other variables in the data set. For this table to be useful for the Quantification and Classing steps, we should be able to calculate the distance between two nominal values from this table.

To explain this better, consider a data set that contains quality, color and size information for 6550 objects. Quality has three possible values – good, ok, bad; color has six values – blue, green, orange, purple, red, white; and size has ten values – 'a' to 'j'. Suppose we want to analyze color (which we shall call our *target variable*) using quality and size (which we shall call our *analysis variables*). To analyze color, we look at the distribution of its values with respect to the analysis variables using a contingency or counts table (Figure 4). From the counts table, we can calculate row percentages (Figure 5) and get a glimpse of which colors are similar to each other based on row profiles; Figure 5 shows that blue and purple have similar row profiles. From the row percentage table, we may be tempted to calculate the distance between two rows using Euclidean Distance formula; however, there are two row percentage tables for color (color by quality and color by size). The technique to be used for this step must have a way to combine all the columns of all tables for color, extract new dimensions that are independent of each other, and transform the counts table into a table that uses the independent dimensions (Figure 6). These independent dimensions would then be the basis of distance calculations needed in the succeeding steps. Using independent dimensions ensures that the distance calculation is not biased by groups of highly associated columns. This argument is similar to performing Principal Component Analysis prior to Cluster Analysis to ensure that the dimensions are independent of each other as required by the Euclidean Distance calculations [Johnson and Wichern 1988]. Each row in the output table (Figure 6) can be thought of as a point in p-dimensional space defined by the p independent dimensions.

Often, the number of analysis variables is large although several may be highly associated with each other. This suggests that the number of independent dimensions to keep in the output table (Figure 6) can be reduced while still maintaining a high accuracy for the distance calculation. This Distance Step must also determine how many of the independent dimensions to keep.

This step is the most important step as it dictates the accuracy of the distance calculation needed in the Quantification and Classing Steps. It is also the most memory hungry and computationally intensive step as it involves transformations of the original (large) data sets and data reduction.

COLOR by QUALITY Counts

	Good	Ok	Bad	Total
Blue	187	727	546	1460
Green	267	538	356	1161
Orange	276	411	191	878
Purple	155	436	361	952
Red	283	307	357	947
White	459	366	327	1152
Total	1627	2785	2138	6550

Figure 4: Counts Table

Row Percentages

	Good	Ok	Bad	
Blue	13	50	37	100
Green	23	46	31	100
Orange	31	47	22	100
Purple	16	46	38	100
Red	30	32	38	100
White	40	32	28	100

Figure 5: Row Percentage Table Showing Row Profiles

Step 2: Quantification Step – Given a table with rows representing the values of the target variable and columns representing independent dimensions extracted from the analysis variables (Figure 6), this step uses the distance information to assign order and spacing to the values of the target variable. The output is a nominal-to-numeric mapping (Figure 7). The goal of this step is to create that mapping in a way that is distance-preserving and association-preserving.

Coordinates for Independent Dimensions

	Dim1	Dim2
Blue	- 0.02	- 0.28
Green	- 0.54	0.14
Orange	0.55	0.10
Purple	0	- 0.25
Red	- 0.50	0.20
White	0.57	0.19

Figure 6: Transformed Table with Independent Dimensions

Nominal	Numeric
Blue	-0.02
Green	-0.54
Orange	0.55
Purple	0
Red	-0.50
White	0.57

Figure 7: Nominal-to-Numeric Mapping

Step 3: Classing Step – This step uses the distance information derived in the Distance Step to determine which values of the target variable are similar to each other and thus can be grouped together with minimal loss of information. Ideally, the output is a hierarchical classing tree showing which values can be grouped together successively and the information lost with each grouping (Figure 8).

Note that the Quantification and Classing steps may or may not be dependent of each other, as suggested by the dashed line between them in Figure 3.

The DQC approach has several advantages. First, it is general-purpose. It provides a pre-processing approach that is useful not only for visualization purposes but also for other techniques that cannot handle high-cardinality nominal variables (e.g., clustering algorithms, association rules) or can only handle numeric variables. Second, it provides a hierarchical classing tree which gives users the flexibility to decide how many value-groups to use in visual displays, depending on their specific analysis goals. Third, it enables multivariate quantification and classing (i.e., determining the

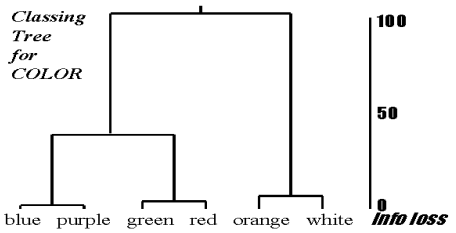

Figure 8: Classing Tree with Information Loss Measure

distance between the values based on their profiles across *several* other variables) which we believe provides more robust results.

4 Distance Step

A well-known family of techniques from Statistics suitable for the Distance Step is the Correspondence Analysis (CA) family [Greenacre 1993; SAS Institute Inc 2000; StatSoft Inc 2002]. CA has been reinvented under different names such as Dual Scaling, Optimal Scaling and Reciprocal Averaging. Its simplest version, called *Simple Correspondence Analysis* (SCA), is designed to analyze the relationship of two nominal variables. SCA takes as input a 2-way counts table (Figure 4). The rows of the counts table can be thought of as data points in a p-dimensional coordinate space defined by the p columns. As such, there is a distance between two data points. CA eliminates the dependencies among the columns by extracting a reduced set of new columns that are independent of each other, while still preserving all or most of the information about the differences between the rows. Figure 6 shows an example output from CA. CA is similar to Principal Component Analysis (PCA) except that CA is for nominal variables while PCA is for numeric variables. Just like PCA, each successive independent dimension (called a principal axis) explains less and less of the overall information.

In its general form, CA can analyze n-way tables that contain some measure of correspondence between the rows and columns (not just counts). In this Distance Step, one can use any version of Correspondence Analysis, as long as it can analyze the relationship of more than two variables and it can provide as output the coordinates of the top independent dimensions for each value of the target nominal variable (as in Figure 6). In the following subsections, we describe two versions of CA suitable for the Distance Step.

4.1 Multiple Correspondence Analysis

Multiple Correspondence Analysis (MCA) extends SCA to analyze more than two nominal variables [Greenacre 1993; SAS Institute Inc 2000; StatSoft Inc 2002]. To perform MCA, simply create a Burt Table (Figure 9) and use that as input to SCA. If a counts table is a cross between two nominal variables, a Burt Table is a cross of all variables by all variables. If V is the total number of unique values across all variables, then the size of the Burt Table is V*V.

The Burt Table structure allows MCA to simultaneously analyze all variables. That is, for every target variable, it can build row profiles using information from *all* other variables. This simultaneous analysis is efficient in terms of processing time because certain calculations can be reused, though wasteful in memory. When the number of nominal variables to analyze is large and some have high cardinality, MCA could run out of memory, depending on how it is implemented.

The coordinates of the first principal axis from MCA follow an optimal scaling property [Greenacre 1993]. This means that such coordinates represent a quantification of all nominal values in all variables. Note, however, that this quantification is sub-optimal when the target variable has a perfect 1-to-many or many-to-many association with another variable, as we show in Section 7.

	QUALITY	COLOR	SIZE
QUALITY	Quality by Quality Counts Table	Quality by Color Counts Table	Quality by Size Counts Table
COLOR	Color by Quality Counts Table	Color by Color Counts Table	Color by Size Counts Table
SIZE	Size by Quality Counts Table	Size by Color Counts Table	Size by Size Counts Table

Figure 9: Example MCA Input Table (Burt Table)

	QUALITY	SIZE
COLOR	Color by Quality Counts Table	Color by Size Counts Table

Figure 10: Example FCA Input Table (Compressed Burt Table)

4.2 Focused Correspondence Analysis

Due to the memory-intensive nature of MCA, we have designed an alternative solution, which we call Focused Correspondence Analysis (FCA), aimed at processing a large number of nominal variables, some possibly having high cardinality.

Unlike MCA which analyzes all variables simultaneously, FCA analyzes one variable at a time, making FCA less computationally efficient than MCA. The memory savings in FCA come from this key idea: instead of comparing value profiles across all other nominal variables, just compare value profiles across the set of nominal variables most associated (i.e., correlated) with the target variable. For example, to analyze one nominal variable color against its most associated variables, say quality and size, we use a compressed Burt table such as Figure 10 as input to SCA. This table is a concatenation of counts tables of color*quality and color*size.

We now discuss why such a table would be a valid input for SCA. In Section 4, we mentioned that the basic version of SCA uses a counts table as input. In Section 4.1, we indicated that we can perform MCA by using a Burt Table as input to SCA. In general, SCA can use as input any table that has the following properties [Greenacre 1984]: (1) the table must use the same physical units or measurements, and (2) the values in the table must be non-negative. If the input table does not meet these assumptions, the table must be transformed before performing SCA. The table in Figure 10 follows these properties.

Two pre-processing steps are needed for FCA: (1) Measure the pairwise association between nominal variables, and (2) Determine the top k associated variables for each nominal variable.

4.2.1 Measure the pairwise association between nominal variables

Given the counts table of two nominal variables, we can state how closely related the variables are with each other using *measures of nominal association* [Agresti 1990]. These measures are analogous to measures of correlation between numeric variables. Several measures of nominal association exist. The choice depends on factors such as the size and shape of the counts table and the presence of low counts [Agresti 1990]. For our purpose, we want a measure of association that is valid for counts tables that may be large, non-square and may contain low cell counts – all properties of counts tables from high cardinality variables. We also want a measure of association that has a bounded range of values, so it is easy to compare two values. One such measure is the *Uncertainty*

Coefficient Asymmetric measure U(R|C) [SAS Institute Inc 2000]. U(R|C) gives the proportion of uncertainty in the row variable R that can be explained by the column variable C. If U(R|C) = 1, the value of the row variable can be known precisely given the value of the column variable.

4.2.2 Determine top k associated variables for each nominal variable

For now, we select some k greater than 2, depending on the memory space available. Since there may be variables that are only weakly associated with other variables, we cannot use a threshold on the measure of association chosen in Section 4.2.1. By selecting k to be greater than 2, we ensure that we use at least one analysis variable for each target variable.

In summary, FCA has its own strengths and weaknesses. With FCA, memory usage is reduced and, in fact, controllable. Also, we empirically show in Section 7 that FCA provides better classing trees compared to MCA for some data sets. FCA however needs a longer run time compared to MCA. This is due to the one-at-a-time analysis as well as the need for pre-processing. In the context of visualization tools, intelligently mapping nominal values to numbers is a pre-processing step that can be run in batch mode. Hence, the run time may not be as important compared to memory space in some situations.

4.3 Reduce Number of Dimensions to Keep

The CA family of techniques uses forms of decomposition (e.g., Singular Value Decomposition, Eigenvalue Analysis) to extract the set of independent dimensions. By default, all forms of CA will keep all independent dimensions calculated [Greenacre 1993] which, for high dimensional high cardinality data sets, require a lot of space. These independent dimensions are ordered by diminishing importance. Part of the CA output is the set of eigenvalues (principal inertia) that indicate the importance of each independent dimension. The first dimension, which is the most important dimension, will have the highest eigenvalue. We plot the eigenvalue by dimension number (called a Scree Plot) and find the 'elbow', the point at which the change in consecutive eigenvalues is small. We keep only the dimensions up to the 'elbow'. This is a common technique used in Factor Analysis [SAS Institute Inc 2000]. This technique is independent of the particular version of CA we use for the Distance Step.

In summary, the MCA-based Distance Step algorithm is as follows:

```
1. BurtTable(rawdataMatrix) -> burtMatrix
2. SCA(burtMatrix) -> coordMatrix, evaluesVector
3. ReduceNumberDim(coordMatrix, evaluesVector) -> coordMatrixSubset
```

while the FCA-based Distance Step algorithm is as follows:

```
1. PairwiseAssociation(rawdataMatrix) -> assocMatrix
2. Set k
3. FCATable(rawdataMatrix, k, assocMatrix) -> fcaInputMatrix
4. SCA(fcaInputMatrix) -> coordMatrix, evaluesVector
5. ReduceNumberDim(coordMatrix, evaluesVector) -> coordMatrixSubset
```

5 Quantification Step

Quantification is the process of assigning order and spacing to the nominal values. For this step, we want a technique that can take as input the independent dimensions from the Distance Step and produce a nominal-to-numeric mapping for each nominal variable.

As mentioned in Section 2, a popular technique used for quantification is based on the theory of Optimal Scaling [Greenacre 1993]. Based on Optimal Scaling, we can use the coordinates from the first CA independent dimension as the quantified version of the nominal values. Unfortunately, when a nominal variable is perfectly associated with another variable (e.g., one-to-many association: one state has many zip codes, or many-to-many association: specific products are only sold in specific regions), we have found in our experiments that this technique fails (see Section 7).

Since we want our technique to work without the need for domain knowledge, we want it to automatically handle cases of perfect associations. Hence, we propose an adjustment to the Optimal Scaling approach: If the first n CA eigenvalues are 1.0, let $scale_i = \sum_{j=1}^{n} coordinate_{i,j}$ where $coordinate_{i,j}$ is the coordinate of the jth independent dimension for row i. Else set $scale_i = coordinate_{i,1}$ (coordinate of the first independent dimension). *Scale* is the term used in Optimal Scaling for the quantified version of a nominal variable. In Section 7, we show that this proposed adjustment gives more effective results for cases with perfect association.

By using independent dimensions extracted via CA to create the quantified versions of nominal values, we have essentially defined the order and spacing of two nominal values to be a function of the chi-squared distance between them. Chi-squared distance is the distance function used in CA [Greenacre 1993]. Chi-squared distance is the weighted Euclidean Distance between a row profile and the average (or expected) row profile. Put differently, the quantified version of a nominal value depends on how different its profile is from the average profile. This implies that even if the nominal variable has an underlying order (i.e., even if it is actually a discretized numeric variable), that order is not likely to be recreated in the quantified version.

An alternative to our modified optimal scaling is to use an algorithm similar to [Ankerst et al. 1998]'s algorithm for rearranging dimensions for a visualization. We search for an ordering of the rows of Figure 6 that minimizes the sum of the distances between all pairs of adjacent rows. This defines the order of the nominal values. The spacing between values can be defined using the distance between the row values. Our Optimal Scaling quantification is faster than this algorithm because Optimal Scaling directly uses output from CA at no extra cost.

6 Classing Step

Classing (or intra-dimension clustering) is the process of finding which values within a nominal variable are similar to each other and thus can be grouped together. For this step, we want a technique that can take as input a table with rows representing the values of the target variable and columns representing independent dimensions extracted from the analysis variables, and produce a hierarchical classing tree showing value groupings and the amount of information lost with each grouping (shown in Figure 8). One method for solving this is to apply a hierarchical clustering algorithm on the CA output table (Figure 6), where each value (row point) is weighted by its counts.

Classing is a data reduction technique, thus it results in loss of information. In this step, we also want to show the amount of information lost whenever two values are grouped together, and display this alongside the classing tree. To approximate the loss of information incurred in classing the nominal variable X, we follow four steps (inspired by [Greenacre 1993]): (1) Determine the variable V with the highest association with X. (2) Create a contingency table between variables X and V. (3) Calculate the total table measure of association (e.g., Uncertainty Coefficient). (4) Starting from the bottom of the classing tree and going all the way to the top, for every pair of nodes merged together, calculate the loss of information incurred, defined by the cumulative percentage loss of information $InfoLoss = 100 * (A(fullTable) -$

Data Characteristics	NOTPERF Synthetic	PERF Synthetic	Mushroom	Auto-mobile	Census
# nominal variables	3	3	15	10	9
# records	6550	3643	8124	205	3969
Max # values/var	10	15	12	22	41
Total # values	19	35	102	61	101
Min strength of association (U)	0.003	0.6	0.02	0.01	0.01
Max strength of association (U)	0.13	1.0	0.53	1.0	0.57

Figure 11: Evaluation Data Sets

$A(afterMerging))/A(fullTable)$, where A(t) is the association measure for table t. An alternative measure of information loss is the R-squared measure that can be calculated with Cluster Analysis [SAS Institute Inc 2000].

7 Experimental Evaluation

In this section, we compare the MCA-based implementation, FCA-based implementation and the common approach of arbitrary quantification (arbitrary ordering and uniform spacing) using a wide range of evaluation measures. We focus our evaluations on the Distance Step (MCA vs. FCA) because it is the most important step in the DQC approach. All implementations and evaluations were done within XmdvTool [XmdvTool Home Page 2003].

7.1 Setup

We used real as well as synthetic data sets, as listed in Figure 11. The real data sets used are popular benchmark data sets taken from [Blake and Merz 1998]. We have used only the nominal variables for most of these data sets. The NOTPERF synthetic data set has three variables (quality, color, size) and is intended to simulate varying degrees of association. This is the data set used in all examples given in earlier sections. The PERF synthetic data set has three variables (region, country and product code) and is intended to simulate perfect associations (1-to-many: region-country, many-to-many: specific set of products are only sold in specific countries).

7.2 Quality of Visual Display

Intuitively, quantification A is better than quantification B if the visual display resulting from A allows the data analyst to confirm or discover (true) patterns in the data that are otherwise harder or impossible to learn using B. The quality of a visual display is more difficult to measure and quantify. One alternative is to conduct user studies and have subjects answer questions using data sets for which they have some domain knowledge. Example questions include: Based on your domain knowledge, are the values that are positioned close together for the most part similar to each other? Are the values that are positioned far from the rest of the other values for the most part that different? Are there fewer line crossings because of the ordering and spacing? Did you discover any new patterns (e.g., outliers, clusters, strength of association between two nominal variables)? In general, which quantification do you feel is better (easier to understand, more believable ordering and spacing)?

7.2.1 Automobile Data Set Case Study

We chose the Automobile Data Set because it is easy to interpret. Figures 12, 13 and 14 display the quantified versions of selected variables in a Parallel Coordinates display. In Parallel Coordinates, each vertical line represents one variable, and each polyline cutting across the vertical axes represents one instance in the data set. Parallel Coordinates is one type of display that requires ordering and spacing of values and it can display several variables compactly. In these figures, we have ordered the variables such that the vertical axes of highly associated variables are adjacent to each other for easier interpretation.

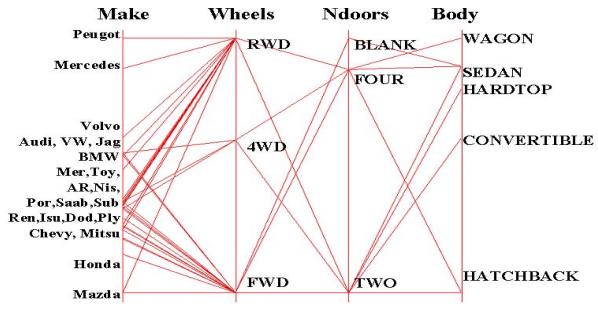

Figure 12: Automobile Data, MCA-Based Quantification

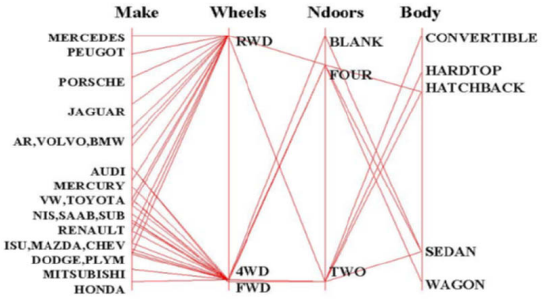

Figure 13: Automobile Data, FCA-Based Quantification

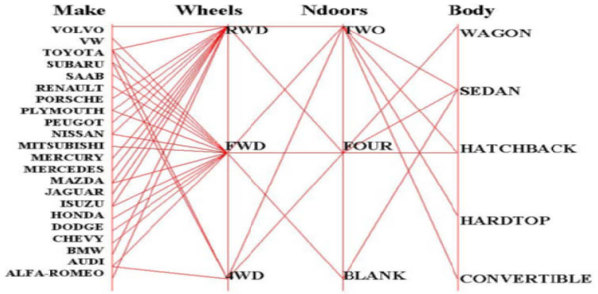

Figure 14: Automobile Data, Arbitrary Quantification

The MCA-based display (Figure 12) and the FCA-based display (Figure 13) present alternative notions of similarity among the values. Some results are similar (Peugot/Mercedes are positioned away from Honda/Mazda), some are different (the spacing between Convertible/Hardtop/Hatchback and Sedan/Wagon). But both MCA and FCA displays confirm our domain knowledge. Which is better depends on the user's preference. Also, both MCA and FCA-based displays have fewer line crossings than the Arbitrary Quantification display (Figure 14).

7.2.2 PERF Data Set Case Study

Figures 15 and 16 display the quantified versions of the variables in the PERF Data Set. Recall that the region-country pair has a 1-to-many association while the country-product code pair has a many-to-many association. These perfect associations are revealed in all CA-based quantifications but are hidden in the arbitrary quantification.

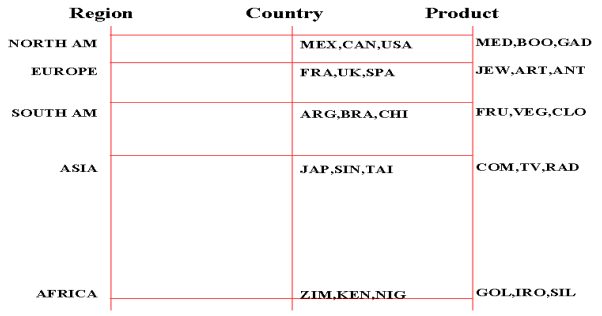

Figure 15: Perfect Association Data, FCA-Based Quantification

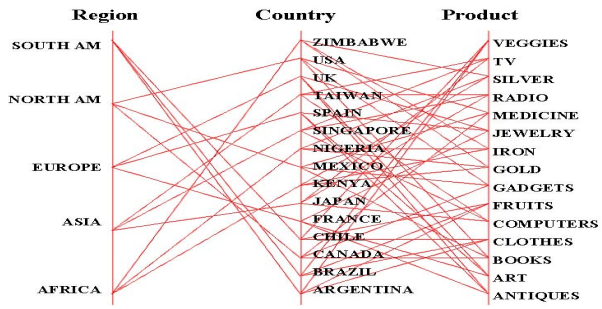

Figure 16: Perfect Association Data, Arbitrary Quantification

7.3 Memory Space and Processing Time

The most memory-intensive part of our implementation is the use of CA in the Distance Step, so we only focus on the memory needed there. Ignoring any specific memory optimization that may be employed by some CA implementations, in general, the MCA input table (Figure 9) requires $(sum\_of\_cardinality)^2$ while the FCA input table (Figure 10) requires at most $max\_cardinality * (sum\_of\_cardinality - max\_cardinality)$ for each nominal variable to be processed. These formulas and the example tables show that MCA uses more memory than FCA.

Figure 17 shows the percentage of time the FCA-based approach runs longer than MCA-based using the formula $100 * (total\_time - MCA\_total\_time)/(MCA\_total\_time)$. For each MCA bar, we show the actual number of seconds that the MCA-based approach ran. So although the gap between FCA and MCA run times seems large, the actual run time of the FCA-based approach is still fast.

7.4 Quality of Quantification

Intuitively, a given quantification is good if (a) instances that are close to each other in nominal space are also close together in quantified space, and (b) if two variables are highly associated with each other, we expect their quantified versions to also have high correlation measure.

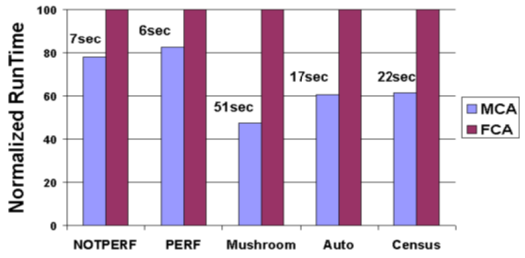

Figure 17: Total Run Time of Entire DQC Approach

[Greenacre 1993] suggests the use of Average Squared Correlation to measure the quality of a quantification. Given the original dataset, replace each nominal variable V_j with its quantified version Q_j (i.e. scale). For each instance i, calculate $score_i = average(Q_{ij})$ for all variables j. For each quantified variable Q_j, calculate the correlation of Q_j and *score* for the entire data set. Then calculate the $average\_squared\_correlation = average((correlation(Q_j, score))^2)$ across all Q_j. The higher the average squared correlation, the better the quantification. Intuitively, if two variables are highly associated with each other, we expect their quantified versions to also have a high correlation measure. If all nominal variables are highly associated with each other, then the score of each observation should be highly correlated with each individual quantified variable. This further implies that if two observations are close together in nominal space, then they would also be close together in quantified space; so the scores of these observations would be close to each other.

Figure 18 shows the Average Squared Correlation for MCA-based, FCA-based and arbitrary quantifications. It shows that both CA-based quantifications are better than arbitrary quantification. The figure also verifies the Optimal Scaling theory, namely, that the quantification based on the coordinates of the first MCA extracted dimension is optimal [Greenacre 1993]. Figure 19 shows how close the FCA scales are to the MCA scales. This figure uses boxplots to show, for the real data sets, the distribution of the correlation between MCA and FCA scales. These boxplots show the minimum and maximum values as well as the 25th, 50th and 75th percentile values of each set of correlation values. Correlation values close to 1.0 mean the FCA scales closely agree with the MCA scales.

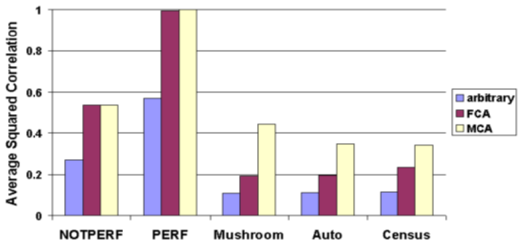

Figure 18: Average Squared Correlation

7.5 Quality of Classing

Intuitively, classing A is better than classing B if, given a classing tree, the rate of information loss with each merging is slower. One way of calculating information loss is given in Section 6.

Figure 20 compares the rate of information loss of MCA compared to FCA for one variable. Each line shows the cumulative

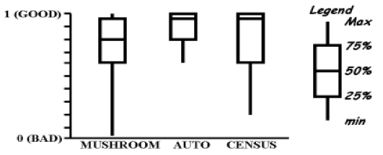

Figure 19: Correlation between MCA Scales and FCA Scales

information loss incurred at each merging of nodes. The lower the line, the slower is the information loss, the better the classing. The gap between the lines (*MCA_cumulative_loss* minus *FCA_cumulative_loss*) can be calculated for all variables. Its distribution has been summarized in Figure 21. This plot shows that the FCA-based classing is better than MCA-based for some data sets.

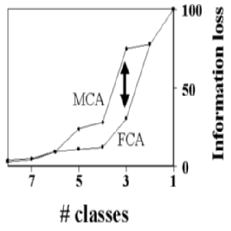

Figure 20: Information Loss Due To Classing For One Variable

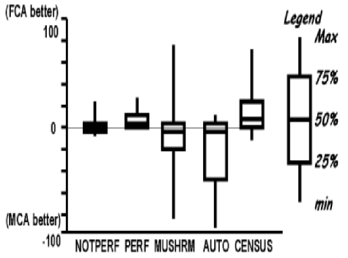

Figure 21: Distribution of the Difference in MCA and FCA Information Loss

8 Conclusions

In this paper, we proposed the Distance-Quantification-Classing (DQC) approach which enables the exploration of data sets containing nominal variables using visualization tools that have been designed exclusively for numeric variables. To make the approach accessible to data analysts, we implemented it in XmdvTool, a public-domain multivariate data visualization package. For our implementation, we used Multiple Correspondence Analysis (MCA) and our own Focused Correspondence Analysis (FCA) for the Distance Step, a modification of the Optimal Scaling formula for the Quantification Step, and Hierarchical Clustering for the Classing Step. We evaluated our approach in terms of memory space requirement, run time, quality of quantification, quality of classing, and quality of visual display. MCA-based and FCA-based quantifications are clearly better than the common practice of arbitrary quantification. In terms of the quality of classing and quantification, MCA seems to perform better than FCA but in terms of the quality of the visual displays, which one is better depends on the eye of the beholder. When memory space is limited, FCA provides a viable alternative to MCA for the Distance Step. The adjustment made to the quantification function to make it work for variables with perfect association improves upon the existing technique of taking only the coordinates of the top CA dimension. Producing classing trees further allows users to reduce the data for displays requiring low cardinality nominal variables.

The DQC approach is a general-purpose pre-processing step which can also be used for other techniques that require low cardinality nominal variables as input (e.g., such as clustering algorithms, association rules, neural networks), or require numeric variables as input (e.g., regression). Possible future work includes allowing the user to interactively modify the ordering, spacing and classing of the nominal values, conducting formal evaluations, and trying other alternatives for each step.

Acknowledgments: We gratefully acknowledge our colleagues in the XmdvTool group at WPI for their contributions to this research, as well as to NSF and NSA for the funding for XMDV research.

References

AGRESTI, A. 1990. *Categorical Data Analysis*. New York: John Wiley & Sons.

ANKERST, M., BERCHTOLD, S., AND KEIM, D. A. 1998. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. *Proc. of IEEE Symposium on Information Visualization, InfoVis'98, p. 52-60*.

BECKER, A., CLEVELAND, S., AND MARTIN, R. 1997. Trellis graphics displays: A multidimensional data visualization tool for data mining. *Knowledge Discovery and Data Mining '97*.

BEYGELZIMER, A., PERNG, C.-S., AND MA, S. 2001. Fast ordering of large categorical datasets for better visualization. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*, ACM, San Francisco, CA.

BLAKE, C., AND MERZ, C., 1998. UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

FRIENDLY, M. 1992. Mosaic displays for loglinear models. *Proceedings of the Statistical Graphics Section, ASA* (Aug), 61–68.

FRIENDLY, M. 1999. Visualizing categorical data. In *Cognition and Survey Research*, M. G. Sirken, D. J. Herrmann, S. Schechter, N. Schwarz, J. M. Tanur, and R. Tourangeau, Eds. John Wiley & Sons, Inc., New York, 319–348.

GREENACRE, M. J. 1984. *Theory and Applications of Correspondence Analysis*. Academic Press, London.

GREENACRE, M. J. 1993. *Correspondence Analysis in Practice*. Academic Press, London.

INSELBERG, A., AND DIMSDALE, B. 1990. Parallel coordinates: A tool for visualizing multidimensional geometry. *Proc. of Visualization '90, p. 361-78*.

JOHNSON, R. A., AND WICHERN, D. W. 1988. *Applied Multivariate Statistical Analysis*, second ed. Prentice Hall International, Inc.

KOLATCH, E., AND WEINSTEIN, B., 2001. Cattrees: Dynamic visualization of categorical data using treemaps. http://www.cs.umd.edu/class/spring2001/cmsc838b/Project/Kolatch_Weinstein.

LEBLANC, J., WARD, M., AND WITTELS, N. 1990. Exploring n-dimensional databases. *Proc. of Visualization '90, p. 230-7*.

MA, S., AND HELLERSTEIN, J. L. 1999. Ordering categorical data to improve visualization. In *IEEE Information Visualization Symposium Late Breaking Hot Topics*, 15–18.

MEULMAN, J., AND HEISER, W. J., Eds. 2000. *SPSS Categories 10.0*. SPSS Inc.

MICCI-BARRECA, D. 2001. A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *SIGKDD Explorations 3*, 1 (Jul), 27–32.

MILANESE, R., SQUIRE, D., AND PUN, T. 1996. Correspondence analysis and hierarchical indexing for content-based image retrieval. In *Proc. 3rd IEEE Int. Conf. on Image Processing, ICIP'96*, 859–862.

SAS INSTITUTE INC, 2000. SAS OnlineDoc Version 8 with PDF Files.

STATSOFT INC, 2002. Electronic statistics textbook: Correspondence analysis. http://www.statsoftinc.com/textbook/stcoran.html.

WARD, M. 1994. Xmdvtool: Integrating multiple methods for visualizing multivariate data. *Proc. of Visualization '94, p. 326-33*.

XMDVTOOL HOME PAGE, 2003. http://davis.wpi.edu/˜xmdv.