

FACTOR ANALYSIS OF CARD SORT DATA: AN ALTERNATIVE TO HIERARCHICAL CLUSTER ANALYSIS

Miranda G. Capra

Center for Human-Computer Interaction and Grado Department of Industrial and Systems Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA

Software and product designers use card sorting to understand item groups and relationships. In the usability community, a common method of formal statistical analysis for open card sort data is hierarchical cluster analysis, which results in a tree of the items sorted into distinct, nested clusters. Hierarchical cluster analysis is appropriate for highly structured settings, like software menus. However, many situations call for softer clusters, such as designing websites where multiple pages link to the same target page. Factor analysis summarizes the categories created in card sorts and generates clusters that can overlap. This paper explains how to prepare card sort data for statistical analysis, describes the results of factor analysis and how to interpret them, and discusses when hierarchical cluster analysis and factor analysis are appropriate.

INTRODUCTION

Card sorting is a technique used in software and product design to understand relationships between items, to group items into categories, and to understand users' mental models of item organization. Many card-sorting programs that provide analysis functions use hierarchical cluster analysis, such as EZSort from IBM's Ease of Use group (Dong, Martin & Waldo, 2001; Martin & Kidwell, 2001). Hierarchical cluster analysis is appropriate for highly structured data because it results in a tree showing the relationships among items, with each item located in a single location in the tree. An example in software design is the creation of menu structures for applications where each function appears once in the menu hierarchy.

One feature of website design is that websites do not have to be rigidly structured. Many different web pages can link to the same target page, highlighting the connections between topics. If a particular page belongs in multiple locations, hierarchical cluster analysis will only show the single best match. Factor analysis may be a better choice for clustering items to include in a website because it generates clusters that can overlap. Factor analysis is also more appropriate when trying to understand general groups, rather than the specific placement of individual items. However, preparation of card sort data for formal statistical analysis is not a technique commonly discussed in usability literature, which may be a barrier to using factor analysis for analyzing card sort data.

The goals of this paper are to explain the techniques necessary to transform the results of a card sort into data to import into a statistical package, and to explain how to interpret the output of a factor analysis. When you analyze card sort results by hand or with a statistical package, rather than relying on the analysis functions built into card-sorting programs, you can overcome some of the common limitations of such software. For example, few card-sorting programs allow nested categories and duplicated items. In addition, statistical packages typically provide more parameters and descriptive statistics than the card sorting software, most of which only provide graphical output.

The technique described in this paper is more appropriate for open card sorts (which create categories) than for closed card sorts (which verify categories). See Maurer and Warfel (2004) and Robertson (2001) for introductions to card sorting and a discussion of the differences between open and closed card sorts. The specific details of conducting a hierarchical cluster analysis or factor analysis are outside the scope of this paper; readers unfamiliar with these techniques should consult a text on multivariate methods such as Johnson (1998).

ILLUSTRATIVE STUDY: HOW TO DESCRIBE A USABILITY PROBLEM?

A study where we used card sorting will illustrate the techniques described in this paper. The goal of the study was to create a set of guidelines for describing usability problems collected through formative usability evaluation. The study began with an open-ended questionnaire that resulted in 70 detailed items. This list was interesting, but too long for practical use or fast comprehension. We used an open card sort on these 70 items to identify a smaller set of general topics that usability practitioners need to address when describing usability problems. This study was part of the author's dissertation research; the complete results of the study will be reported in future publications. The next two sections provide a brief summary of this study.

Open-Ended Questionnaire

We used an exploratory questionnaire to gather ideas about what should be included in usability problem descriptions. Nineteen usability practitioners with 3 to 30 years of experience ($M=10.2$, $SD=4.7$) who had conducted at least 10 usability evaluations ($M=133.6$, $Median=50$, $SD=244.1$) completed this open-ended questionnaire. The questionnaire asked them to describe important qualities of a description.

Respondents described between 3 and 11 qualities ($M=5.2$, $SD=2.1$). These qualities varied greatly in level of granularity and specificity, with some respondents providing a brief description of each item and some providing a full

paragraph. Two researchers independently coded the responses, resulting in a combined list of 70 detailed guidelines. The next step was to identify high-level groups that summarized this detailed list.

Open Card Sort

The goal of the card sort was to create *general* guidelines for describing usability problems from the *detailed* list of 70 guidelines collected in the open-ended questionnaire. Eight usability practitioners with 3 to 15 years of experience (M=8.8, SD=4.6) who had conducted at least 10 usability evaluations (M=80, Median=75, SD=70) sorted the 70 items into categories. We asked them to give each category a name, allowed them to create one level of subcategories, and allowed them to put each item into multiple categories. Participants created a total of 77 categories and subcategories, with individual participants creating between 4 and 13 (M=9.6, SD=3.4). Figure 1 shows sample subset of sorted items submitted by one of our participants.

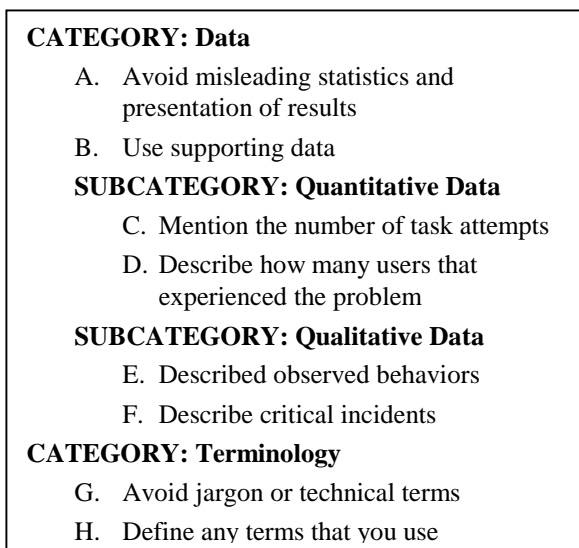


Figure 1. Eight Items, Sorted

When we reviewed our participants’ responses, it was clear that there were several overlapping concepts related to writing usability problem descriptions. For example, describing task success or failure is an aspect of describing the user’s actions, but it is also an aspect of backing your conclusions with quantitative data. Hierarchical cluster analysis was inappropriate for our data because it results in distinct, non-overlapping clusters. Factor analysis was a better fit because the concepts in our domain overlapped.

PREPARING THE DATA

Most clustering procedures require specially formatted data to analyze, such as binary variables, a similarity matrix, or a dissimilarity (distance) matrix. However, transforming card sort data into these formats is not an obvious process, and is not commonly discussed in usability literature about card sorts. This section describes the process of preparing the data for

import into a statistical package: transforming the category memberships into binary variables, measuring the similarity of each pair of items, and arranging the similarities into a similarity matrix. This section uses the sorted items shown in Figure 1 to illustrate the transformations and calculations.

Creating Binary Variables

The process of preparing the data for factor analysis begins by creating one binary variable for each category or subcategory created by each participant. Items in a subcategory belong to both the subcategory and the parent category. In our study, this resulted in 77 binary variables. Then assign each item a value of one (1) if it is a member of the category or subcategory, or a value zero (0) if it is not. Thus, the four categories and subcategories in Figure 1 transform into four binary variables, as shown in Table 1

Table 1. Example of Asymmetrical Binary Representation of Categorical Data

<i>Category</i>	A	B	C	D	E	F	G	H
Data	1	1	1	1	1	1	0	0
Quantitative Data	0	0	1	1	0	0	0	0
Qualitative Data	0	0	0	0	1	1	0	0
Terminology	0	0	0	0	0	0	1	1

These binary variables are *asymmetrical* because lack of membership in a category is not the opposite of membership in a category. To illustrate, consider the categories in Figure 2. *Female* and *Male* transform into a single symmetrical binary variable, *IsFemale*. This variable is *symmetrical* because female implies not male, and not female implies male. *Animal*, *Vegetable* and *Mineral* transform into two asymmetrical binary variables, *IsAnimal* and *IsVegetable*; being neither animal nor vegetable implies being mineral. Unlike female and not female, which are opposites, animal and not animal are not opposites; not animal could be either vegetable or mineral. The distinction between symmetrical and asymmetrical binary variables is important because asymmetrical variables require noninvariant measures of similarity (Kaufman & Rousseeuw, 1990). In our study, we used Jaccard scoring to measure the similarity of our asymmetrical binary variables.

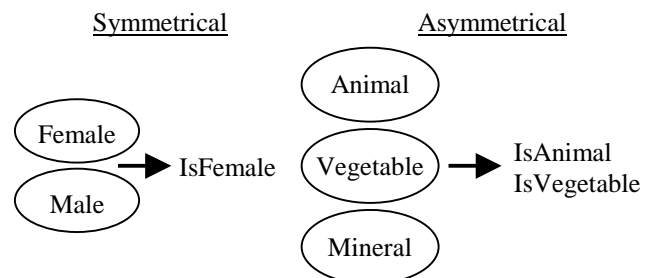


Figure 2. Symmetrical and Asymmetrical Binary Variables

Measuring Similarity

The next step in preparing data for factor analysis is to calculate the similarity of each pair of items sorted. We calculated the similarity using Jaccard’s Coefficient of Community (1912), also known as the *S-coefficient* (Kaufman & Rousseeuw, 1990) and the Jaccard score. Equation 1 shows the formula for calculating this score, and Table 2 illustrates the values of *a*, *b*, *c*, and *d*. The Jaccard score is the ratio of the commonality between two items to the spread of the two items, or the number of categories they have in common (intersection) to the total number of categories containing the items (union). When calculating the Jaccard score for items in nested categories, count items in a subcategory as belonging to both the subcategory and the parent category.

$$J = \frac{\text{intersection}}{\text{union}} = \frac{a}{a + b + c} \tag{1}$$

Consider the sorted items in Figure 1. To calculate the similarity between E and G, we count all of the categories that contain both items (*a*), only one of the two items (*b*, *c*), and neither item (*d*), as shown in Table 2. Using Equation 1, the Jaccard score for E and G is the ratio of the number categories containing both items (*a*) to the number of categories that have at least one of the items (*a + b + c*), or 0/3 (zero).

Table 2. Category Membership Counts for Items E, G

		Categories with E	Categories without E
Categories with G		a = 0	b = 1
Categories without G		c = 2	d = 1

Table 3 presents a sample of Jaccard score calculations for the items in Figure 1. Items A, C and E all have one category in common, *Data*. The similarity of A and C is 1/2 because they are spread across two (sub)categories: *Data* and *Quantitative Data*. In contrast, the similarity of C and E is smaller, 1/3, because they are spread across three (sub)categories: *Data*, *Quantitative Data*, and *Qualitative Data*.

Table 3. Sample Jaccard Score Calculations for Figure 1

Pair	Calculation	J
J (A, B)	1 / 1	1
J (A, C)	1 / 2	0.5
J (A, G)	0 / 2	0
J (C, D)	2 / 2	1
J (C, E)	1 / 3	0.33
J (E, G)	0 / 3	0

Creating a Similarity Matrix

The final step in preparing the data from the card sort for factor analysis is to create a similarity matrix, an *n x n* matrix

(where *n* is the number of cards sorted) measuring how similar each item is to every other item. The Jaccard score for each pair of items is a measure of similarity, so assemble the Jaccard scores into a symmetrical matrix with one row and column for each item. In our study, the result of this step was a 70x70 similarity matrix. Once you have imported this into a statistical program, you are ready for factor analysis.

FACTOR ANALYSIS

The next step is to perform a factor analysis. In your factor analysis, use the similarity matrix from the previous step as a covariance matrix. Readers that are unfamiliar with factor analysis should consult a text on multivariate methods such as Johnson (1998) for discussions of selecting the number of factors and choosing a factor rotation technique.

Output of a Factor Analysis

A factor analysis of our data resulted in 10 factors. For each item and each factor, a factor analysis generates a *loading*, which is a score of how strongly the item is associated with the factor. These loadings range from 0 to 100. We dropped items with loadings below 40, the default cutoff value in SAS. We had five items that did not load on any factors and eight items that loaded on two factors. For example, “Be pragmatic/practical; avoid theories/jargon that non-HCI people wouldn’t appreciate” loaded on both Factor 2, *Clarity/Jargon*, and Factor 9, *Politics/Diplomacy*. Table 4 shows the loadings for these two factors. Creating names for the factors, such as *Clarity/Jargon*, was part of the interpretation process. We relied on two sources of information for interpreting the factors: the factor loadings for the items and the names the participants gave their categories that best matched each factor.

Table 4. Factor Loadings for Two Factors

Factor 2, <i>Clarity/Jargon</i>	
Loading	Item Text
91	Be concise, avoid wordiness
88	Use precise terminology
85	Avoid jargon or technical terms
80	Define any terms that you use
78	Don’t use vague terms and descriptions; be concrete
73	Be clear and precise
61	Be pragmatic/practical; avoid theories/jargon that non-HCI people wouldn’t appreciate (also in Factor 9)
40	Avoid so much detail that no one will want to read to description (also in Factor 5)
Factor 9, <i>Politics/Diplomacy</i>	
Loading	Item Text
63	Avoid judging the system or decisions made by other team members
61	Avoid pointing fingers or assigning blame
54	Mention good design elements and successful user interactions
41	Be pragmatic/practical; avoid theories/jargon that non-HCI people wouldn’t appreciate (also in Factor 2)

Matching Category Names

The participants in our card sort provided a name for each category and subcategory that they created. To help us interpret our factors we wanted to identify the participant-supplied category names that best matched each factor. To do this we once again used Jaccard scoring to calculate similarities, but this time between categories instead of items. Jaccard scoring measures the similarity of two items by counting their overlapping categories, but it can also measure the similarity of two categories by counting their overlapping items (Jaccard, 1912). To find the names that matched a particular factor, we measured the similarity of that factor to each of the 77 categories and subcategories created by our participants.

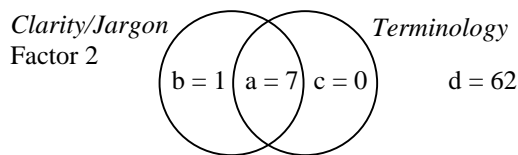


Figure 3. Overlap Between Factor 2, Clarity/Jargon, and Participant-Supplied Category Terminology

For example, one of our participants created a category, *Terminology*, which contained the first seven of the eight items in Factor 2, *Clarity/Jargon*. To calculate the similarity between *Clarity/Jargon* and *Terminology*, we counted all of the items in both categories (a), in only one category (b, c), and in neither category (d), as shown in Figure 3. Using Equation 1, the Jaccard score for these two categories is the ratio of the intersection of the categories (a) to the union of the categories (a + b + c), or 7/8 (0.87). Table 5 shows the names that matched *Clarity/Jargon* in our study (shown in Table 4). We multiplied the Jaccard scores by 100, and only used category names above a cutoff value of 30. We marked parent category names with brackets.

Table 5. Jaccard Scores For Categories Matching Factor 2, Clarity/Jargon

Score (J*100)	Participant-Supplied Category Name
87	Terminology
75	[General] Be Clear and Precise
66	Wording
58	Style
46	General
43	General Style
37	[General] Precision

Factor Interpretation

The final step in the analysis is to interpret each factor based on the loading of each item on each factor. Those items with the highest loadings on a factor are most strongly associated with the factor. In our study, we also used the Jaccard scores of the names for the categories that matched each factor. Those names with the highest Jaccard score for a factor are most strongly associated with that factor. We

interpreted our factors by creating a summary description of each factor. The author wrote the first sentence of each summary based on the most strongly associated items and names for the factor. The remainder of the summary was based on the remaining items that loaded on the factor. The summaries for Factors 2 and 9 are as follows.

2. **Be clear and precise while avoiding wordiness and jargon.** Define terms that you use. Be concrete, not vague. Be practical, not theoretical. Use descriptions that non-HCI people will appreciate. Avoid so much detail that no one will want to read the description.

9. **Consider politics and diplomacy when writing your description.** Avoid judging the system, criticizing decisions made by other team members, pointing fingers or assigning blame. Point out good design elements and successful user interactions. Be practical, avoiding theory and jargon.

DISCUSSION

Jaccard Scoring

Jaccard scoring can calculate similarities for card sorts with both duplicated items and nested categories. It can score both simple and complex nesting, including overlapping and duplicated categories. While most card sorting packages do not allow nested categories, it is common to encourage nesting in physical card sorts, and essential to use nesting with large sets of cards. You can calculate the Jaccard score directly from the raw category memberships, but the transformation into binary variables is simpler. You can then use a spreadsheet or statistical program to calculate the Jaccard scores.

Jaccard scoring is a common technique for calculating item similarities when clustering asymmetrical binary data (Kaufman & Rousseeuw, 1990). It is similar to techniques for calculating semantic similarities between words using word taxonomies. If you locate two words in a taxonomy, the semantic similarity is the ratio of the path intersection, or overlapping portion of the paths, to the path union, or total path counting the overlap only once (Resnick, 1999). There are many similar scoring techniques, also called *matching coefficients* (Lorr, 1983), for binary variables (see Everitt, 1974; Kaufman & Rousseeuw, 1990; Lorr, 1983;).

Hierarchical Cluster Analysis vs. Factor Analysis

Hierarchical cluster analysis is an *individual*-directed technique (Johnson, 1998). It focuses on the relationships between the individual items, and items can only appear in a single place in the hierarchy. It is best suited for data where a clear hierarchical organization exists (Everitt, 1974). For example, plants are naturally organized into species, then genera, orders, etc., so hierarchical cluster analysis is appropriate for analyzing plant samples. However, an analysis that forces each item to belong in a single location is not appropriate for every situation.

Factor analysis is a *variable*-directed technique (Johnson, 1998). It creates new variables (factors) that summarize the input variables (card sort categories). This worked well for our study because we were seeking a small number of factors that summarized the many card sort categories that our participants created. Factor analysis was also appropriate for our study because it allows for overlapping clusters; items can load on more than one factor. For example, one item loaded on both Factor 2 and Factor 9. When we did a hierarchical cluster analysis on the same data, one cluster was identical to Factor 2. Another cluster was close to Factor 9, but lacked this item. With hierarchical cluster analysis, we missed the association with Factor 9 because it places each item in the single best location. Had our items had been web pages we might have left out an important cross-link on our site.

Applications

In software design, hierarchical cluster analysis is appropriate for rigidly structured situations, such as creating menu structures for software applications where each function appears once in the menus. The technique described in this paper is suitable for more flexible situations, such as creating top-level categories for a website. Participants can perform an open card sort on the pages in the site, and factor analysis will identify the major categories for the site. The potential for overlapping categories is appropriate for web site design, since multiple pages on a website can link to the same target page. The designers could also use the factors identified through the factor analysis as categories for a closed card sort, both to precisely locate specific pages within the website and to verify the names chosen for the factors.

The technique described in this paper would be particularly useful for large card sorts. We found that it was difficult to interpret the dendrogram (tree) created by a hierarchical cluster analysis of our card sort data, in part because the overlapping nature of our application domain resulted in weak hierarchical clustering, and in part because we had so many items. Some authors use visual inspection of card sort data (e.g. Nielsen & Sano, 1994; Lamantia, 2003), but this may be difficult with large numbers of items or participants. Factor analysis tends to result in easily interpretable factors (Johnson, 1998), and is well suited for very large data sets.

CONCLUSION

The paper describes a straightforward way to prepare card sort data for formal statistical analysis. Calculating the similarities and running the analysis with statistical software can be time-consuming, but can also be more powerful than visual inspection of the data or using the clustering algorithms that come packaged with many of the card-sorting programs. Statistical packages can perform tests to evaluate the shaping and distribution of your data, and allow you to adjust the parameters of the analysis and select a specific clustering algorithm. In addition, they provide descriptive statistics of results for making decisions such as how many clusters to use. You can select the analysis that is best suited for your specific

application, of which hierarchical cluster analysis and factor analysis are just two of many clustering and multivariate technique (see Everitt, 1974; Jardine & Sibson, 1971; Johnson, 1998; Kaufman & Rousseeuw, 1990; Lorr, 1983).

ACKNOWLEDGEMENTS

This research was conducted while the author was supported by the Alexander E. Walter Fellowship from the Grado Department of Industrial and Systems Engineering at Virginia Tech. The author is grateful for the advice of her advisor, Tonya Smith-Jackson. Susanne Aref of the Virginia Tech Statistical Consulting Center suggested using factor analysis to generate overlapping clusters for the illustrative study. Robert Capra assisted with data analysis.

REFERENCES

- Dong, J., Martin, S., & Waldo, P. (2001). A user input and analysis tool for information architecture. In *Extended Abstracts of the ACM Conference on Human Factors in Computing System – CHI 2001* (pp. 24-25). New York: ACM.
- Everitt, B. (1974). *Cluster Analysis*. London: Heinemann Educational Books.
- Jaccard, P. (1912). The distribution of flora in the alpine zone. *The New Phytologist*, 11(2), 37-50.
- Jardine, N. & Sibson, R. (1971). *Mathematical Taxonomy*. New York: John Wiley & Sons.
- Johnson, D. (1998). *Applied Multivariate Methods for Data Analysis*. Pacific Grove, CA: Duxbury.
- Kaufman, L., & Rousseeuw, P.J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: John Wiley & Sons.
- Lamantia, J. (2003). Analyzing Card Sort Results with a Spreadsheet Template. Retrieved on April 12, 2005 from the Boxes and Arrows web site: http://www.boxesandarrows.com/archives/analyzing_card_sort_results_with_a_spreadsheet_template.php
- Lorr, M. (1983). *Cluster Analysis for Social Scientists*. San Francisco: Jossey-Bass.
- Martin, S. & Kidwell, D. (2001). A case study in cluster analysis for intranet organization. In A. Ambler, K. Graham & P. Jenson (Eds.), *Proceedings of the 2nd International Workshop on Engineering Management for Applied Technology (EMAT 2001)* (pp. 57-64). Los Alamitos, CA: IEEE Computer Society.
- Maurer, D. & Warfel, T. (2004). Card sorting: a definitive guide. Retrieved on February 8, 2005 from the Boxes and Arrows web site: http://www.boxesandarrows.com/archives/card_sorting_a_definitive_guide.php
- Nielsen, J. & Sano, D. (1994). SunWeb: User Interface Design for Sun Microsystems's Internal Web. In *Electronic Proceedings of the Second World Wide Web Conference '94: Mosaic and the Web*. Retrieved February 8, 2005 from the NCSA web site: <http://archive.ncsa.uiuc.edu/SDG/IT94/Proceedings/HCI/nielsen/sunweb.html>
- Resnick, P. (1999). Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research*, 11: 95-130.
- Robertson, J. (2001). Information Design Using Card Sorting. Retrieved on April 12, 2005 from the Step Two Designs website, <http://www.steptwo.com.au/papers/cardsorting/>