Towards Information-Theoretic Visualization Evaluation Measure: A Practical example for Bertin's Matrices

Innar Liiv
Department of Informatics
Tallinn University of Technology
15 Raja Street, Tallinn 12618, Estonia
innar.liiv@ttu.ee

ABSTRACT

This paper presents a discussion about matrix-based representation evaluation measures, including a review of related evaluation measures from different scientific disciplines and a proposal for promising approaches. The paper advocates linking or replacing a large portion of indefinable aesthetics with a mathematical framework and theory backed up by an incomputable function – Kolmogorov complexity. A suitable information-theoretic evaluation measure is proposed together with a practical approximating implementation example for Bertin's Matrices.

Categories and Subject Descriptors

F.1.3 [Theory of Computation] *Complexity Measures and Classes*
H.5.2 [User Interfaces]: *Evaluation/methodology*

General Terms

Measurement, Experimentation, Human Factors

Keywords

Visualization, Evaluation metrics, Kolmogorov Complexity

1. INTRODUCTION

There has been some interesting work in information visualization community comparing, analyzing and discussing the differences between node-link diagrams and matrix-based representation [1,2]. MatrixExplorer is a (social) network visualization system using exactly those two representations [3]. These inquiries go back to 1940s, when a dialogue took place between Forsyth, Katz and Moreno [4,5,6].

However, there has been relatively little discussion from the visualization perspective regarding different strategies and theories to (re)order the data in matrices, as there are *row*!**column*! permutations to encode and visualize the same data without losing data consistency or transforming data. Henry and Fekete have addressed this issue through a user study to understand "how the layout of table data affects the user understanding and his exploration process" [7].

Bertin demonstrated the power of data rearrangement in matrices, stressing the importance of simultaneous availability of three information levels in every effective visual display of data, e.g. a classic example of townships in Fig. 1 [8,9]. One should be able to find an immediate visual reply to: 1) questions asked about the details of data presented in rows and columns, (e.g. Does township '08' have a railway station? Which townships have police stations); 2) local patterns found in the data (e.g. Where there is no water supply, there are no high schools); and 3) global patterns and trends found in the data (e.g. We are able to identify the transformation of rural areas to urban and what changes take place in the characteristics supporting such a transition).

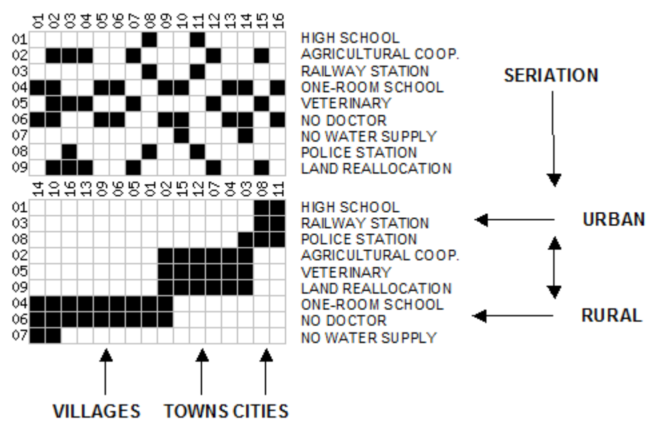


Figure 1. Bertin's [8,9] example of matrix reordering

Setting aside Bertin's elegant and intuitive example, two interconnected important question remain:

- What exactly is the goal of such generic reordering procedure (how to formalize it)?
- How to reach that goal!?

Bertin ([8], p.6) proposed the goal, "simplifying without destroying". He was convinced ([8], p.7) that simplification was "no more than regrouping similar things." Bertin stated that, with assistants and mechanical devices, "it only takes three days to construct a matrix and three weeks to process and interpret it more deeply", which he expected to improve over time. At the same time, several algorithms for automatic matrix reordering (seriation) already existed, but a quick propagation of such developments and results was restrained and muted by the barriers of different scientific traditions and disciplines.

Mueller et al [10] have recently extended the work of Ling [11] on visualizing similarity matrices to large-scale graphs and evaluate the interpretability of results from different one-mode vertex ordering algorithms, including sensitivity to the initial order of rows and columns [12]. For recent reviews of matrix reordering (seriation) methods, see refs [13,12].

This paper will provide a very cursory and generalized review of main evaluation strategies applied in different scientific disciplines (in section 2), present an information-theoretic visualization evaluation measure (in section 3) with a practical implementation example (in section 4), concluding with a discussion and directions for future research (in section 5).

2. RELATED EVALUATION MEASURES

Let us take another look at Bertin's example (Figure 1) and think again about the (definable) goal of such a procedure and result.

Should it make a big difference if the order of rows and columns were inverted? One could argue that the interpretation would not be much different. Unfortunately, there exist several domain-specific matrix visualization evaluation measures, where inversion of row order can change dramatically the result (e.g. grouping *efficiency* [14] and grouping *efficacy* [15] in cellular manufacturing). Researchers applying matrix reordering methods in archaeology for relative dating and sequential arrangement of events, reached an interesting conclusion: even if we find a visual order that suggests a true chronological ordering, external information is needed to give hints about the directions of time ([16], p.60).

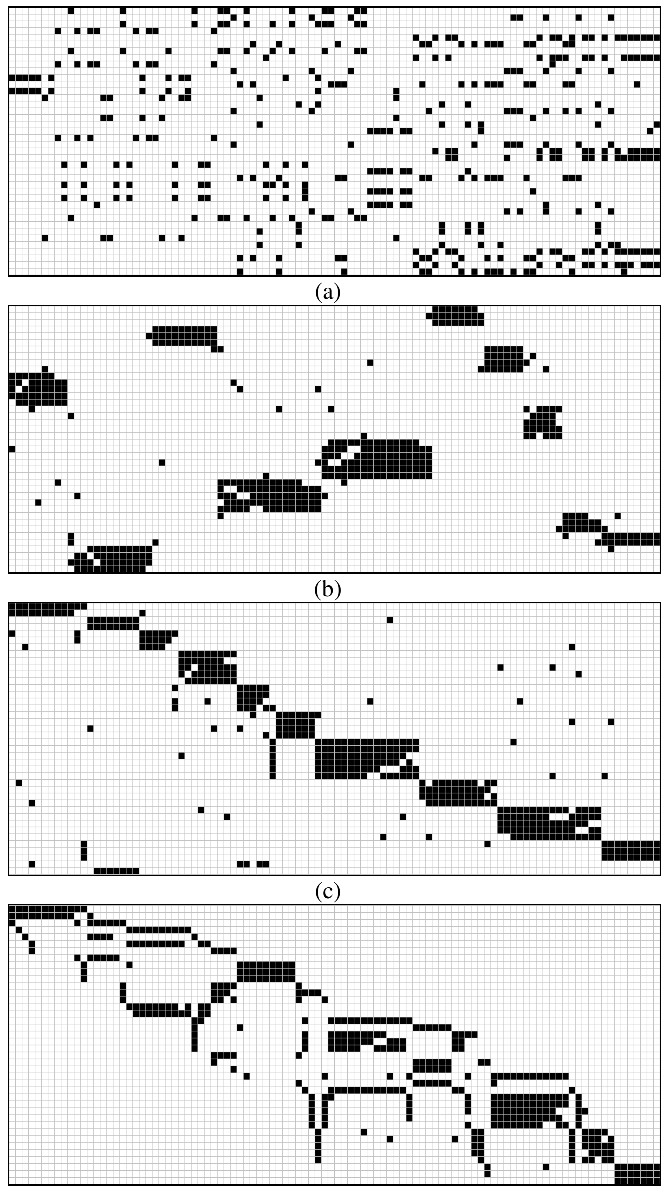
If we identify clusters from the results (e.g. "Villages" from Figure 1), should the inter-cluster order of rows (and columns) make a difference? (e.g. Would it make a difference to change the rows 4 and 7 in the result?) For matrix reordering methods using hierarchical clustering methods and a dendrogram to reorder the rows and columns accordingly to produce a result, it is difficult to identify the inter-cluster behavior. There are "2^{n-1} linear orderings consistent with the structure of the tree" [17] generated by hierarchical clustering. An arbitrary selection from all possible orderings works as a strong heuristic, but most probably will not result in satisfactory results if the similarity between neighbouring elements and high overall regularity within the data matrix is important. To remedy this situation, several authors have proposed additional procedures to perform optimal leaf ordering of the dendrogram [17,18].

Alternative solutions are present if the goal is defined as a combinatorial optimization problem for ordering objects to maximize overall adjacency similarity. With that approach (e.g.[3]) we could relate it to traveling salesman problem and solve it with available heuristics and computer programs. For example, Verin and Grishin [19] tried to find a permutation that would minimize the sum of Hamming distances in order to measure the "quality estimate of image smoothness" with matrices.

However, combinatorial optimization approaches bring us back to the classical problem – how to define similarity, results will be completely different. Instead of entity-to-entity similarity, we could also look at data cell neighborhood similarity, and even for neighborhoods, there are several different examples for evaluating matrix visualizations (McCormick *et al* [20] used von Neumann neighborhood and Niermann [21] used Moore's). All of those issues are far from being relevant only to Bertin's matrices, but emerge in all visualizations where similarity ordering is being introduced (e.g. [22,23,24]).

To summarize, main strategies and heuristics to reorder matrices have been: solutions using domain-specific heuristics (e.g. [14,15,16]); hierarchical clustering related solutions [25]; adjacency similarity maximization, both at entity-to-entity and neighborhood methods [3,19,20,21].

To further illustrate the problem, Figure 2 presents an initial matrix (a) and three possible (out of *rows!*columns!* combinations) "good" visualizations of the same data according to different algorithms and loss functions.



(a) (b) (c) (d)

Figure 2. Initial matrix and different visual permutations

Clearly all three are better than the initial, but which one is the best – would there be an unifying measure to compare them? The data compression approach presented in the next section is a good start to work towards unifying domain and task-specific objective functions.

3. INFORMATION-THEORETIC EVALUATION MEASURE

Kolmogorov complexity is the length of the shortest effective description of an object. We suggest looking at the matrix visualization evaluation, using data compression as a special case of Kolmogorov complexity of a string where it is allowed to "cheat" under specific restrictions to make regularity in the string more apparent and therefore more compressible. A pixel based representation of the matrix and string based (textual) representation of the matrix are considered to have a one-to-one mapping. The most efficient "cheating procedure" is then what we are looking for. The use of Kolmogorov complexity and the minimum description length principle [26,27] is gaining acceptance and popularity in the data mining and information visualization communities [28,29,30,31]. According to Wilkinson ([31], p.531) "a well-permuted image can be thought of as one requiring a minimum number of bits to encode it (compared to other permutations). In other words, a well-permuted image is more compressible." This statement fits perfectly with the approach of this paper. The proposed approach itself, however, will be different from Wilkinson's, as he presented two loss functions related to entropy, which were based on the neighborhood similarity of data cell. In this paper, instead of using entity-to-entity or cell neighborhood similarity measures, the whole dataset and its compressibility is considered at once. The main contribution of this approach is a possible future cross-fertilization of information visualization and data compression communities.

Definition. Matrix reordering can be defined as a combinatorial optimization problem for minimizing a loss function ***L*** on a matrix **A** using permutation matrices Π and Φ for reordering the rows and columns in a way that maximizes the local and global patterns:

$$\arg\min_{\Pi,\Phi} L(\Pi A \Phi) \qquad (3.1)$$

ΠAΦ denotes a matrix A multiplication with permutation matrix Π from the left (to reorder the rows of the original matrix) and with permutation matrix Φ on the right (to reorder the columns of the orgininal matrix). *L* denotes an arbitrary loss function, which describes a visual clutterness in the visualization. However, if we replace the arbitrary loss function with a specific mathematical function with an incomputable property, we are able to work towards practical approximating implementations.

Definition. Given all N!•M! permutations of A, we want to select permutation matrices Π and Φ such that the length of the shortest encoding (according to minimum description length principle [26,27] of matrix multiplication ΠAΦ is minimal.

We can look at this definition also as a combinatorial optimization problem of finding permutation matrices Π and Φ to minimize the following:

$$\arg\min_{\Pi,\Phi} K(\Pi A \Phi) \qquad (3.2)$$

However, as Kolmogorov complexity is incomputable, we will make an approximation using the length *l* of the result of an arbitrary compression algorithm (e.g. `gzip`):

$$\arg\min_{\Pi,\Phi} l(compress(\Pi A \Phi)). \qquad (3.3)$$

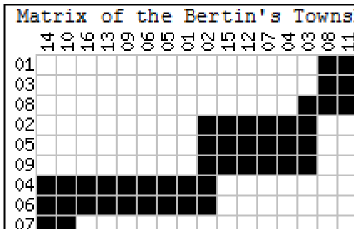
Besides the formal notation, we will make a practical implementation example of the proposed measurement, using shell scripting and Unix piping to secure rapid repeatability and scrutiny for everyone with the access to some Unix-based operating system.

4. A PRACTICAL EXAMPLE

We will provide a practical example of implementing the proposed measure with a standard compression tool `gzip`, which is included by default under most and available for all Unix-based operating systems (Mac OS X, Linux, FreeBSD, Solaris etc.). `gzip` is also freely available for Microsoft Windows platforms and the following examples are easily repeatable there as well with possible minor modifications.

First, we have to encode our example datasets into binary strings and textfiles as shown in Fig. 3. We will be measuring matrix visualization quality with the classical *Townships* example with the solution by Jacques Bertin ([8], p.33), which forms almost a reverse block diagonal form, but similarly to real-world datasets, does not provide a completely pure decomposition to partitions.

```
Matrix of the Bertin's Townships example:
```




```
Corresponding content of the file bertin_solution:
0000000000000011
0000000000000011
0000000000000111
0000000011111100
0000000011111100
0000000011111100
1111111110000000
1111111110000000
1100000000000000
```

Figure 3. Guideline for encoding the Bertin dataset

It is easy to print out the initial matrix and evaluate the correctness of matrix transposition directly in a shell environment of the operating system using a trivial matrix transposition script and standard `md5` *hashing tool:*

```
> cat bertin
0000000100100000
0111001000010010
0000000100100000
1100110011001101
0111001000010010
1100110011001101
0000000001000100
0010000100100000
0111001000010010
```

```
> cat bertin | md5
b650d20c6c224076c8b6baf69c61fcfc
> cat bertin | ./TRANSPOSE_MATRIX |
./TRANSPOSE_MATRIX | md5
b650d20c6c224076c8b6baf69c61fcfc
```

Measuring the "goodness" of Bertin's initial and unordered dataset can be done by printing the content of the file bertin to gzip input using piping and measuring the bytes using `wc -c`:

```
> cat bertin | gzip --best | wc -c
      57
> cat bertin | ./TRANSPOSE_MATRIX | gzip --
best | wc -c
      62
```

Such output should be interpreted as follows: Matrix stored in *bertin* can be compressed to **57** bytes using gzip and the transposed matrix to **62** bytes, respectively. The lowest from those two is **57**, making it the goodness measure (equation 3.3) for that permutation of the matrix, which is the initial and unordered version of the matrix in the specific case.

Next, we will measure the compressibility of the matrix permutation solution provided by Bertin ([8], p.33):

```
> cat bertin_solution
0000000000000011
0000000000000011
0000000000000111
0000000011111100
0000000011111100
0000000011111100
1111111110000000
1111111110000000
1100000000000000
```

```
> cat bertin_solution | gzip --best | wc -c
      49
> cat bertin_solution | ./TRANSPOSE_MATRIX |
gzip --best | wc -c
      47
```

The evaluation of compressibility of that permutation gives us **49** bytes for the proposed solution matrix and **47** bytes for the matrix' transposition, resulting evaluation measure of **47** according to equation (3.3).

If we compare this solution with an alternative permutation *bertin.min* using the same initial dataset, we get the following result:

```
> cat bertin.min
0000000011111100
0000000011111100
0000000011111100
0000000000000111
0000000000000011
0000000000000011
0000001100000000
1111111110000000
1111111110000000
```

```
> cat bertin.min | gzip --best | wc -c
      48
> cat bertin.min | ./TRANSPOSE_MATRIX | gzip
--best | wc -c
      48
```

The evaluation measure for this permutation is **48** for the matrix and its transposition. By visual inspection, we can see that the alternative solution is also able to decompose the system into groups, however, not providing as seamless transformation as Bertin's manual solution, which also concords with the slightly better result attained with data compression.

5. CONCLUSIONS AND FUTURE WORK

The paper advocates that it is more than reasonable to link indefinable aesthetics with a mathematical framework and theory backed up by an incomputable function – a perfect match! Using the Kolmogorov complexity in an information visualization metric would enable us to replace a number of aesthetics issues (yet undefined) with the minimum description length principle and Occam's razor. Additionally, a myriad of methods and techniques built on Kolmogorov complexity and the minimum description length principle become available for information visualization community.

This kind of practical approach of Kolmogorov complexity should be further researched toward pixel-based [22,23] and other visualization techniques, where there is one to one and reversible mapping between an original data point and visualization without a transformation or aggregation process.

6. ACKNOWLEDGEMENTS

The author wishes to thank Peter Kinnaird for helpful discussions and comments.

7. REFERENCES

[1] Ghoniem, M., Fekete, J., and Castagliola, P. 2005. On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. *Information Visualization* 4, 2, 114-135.

[2] Henry, N. and Fekete, J., & McGuffin, M. J. 2007. NodeTrix: a Hybrid Visualization of Social Networks. *IEEE Transactions on Visualization and Computer Graphics* 13, 6, 1302-1309.

[3] Henry, N. and Fekete, J. 2006. MatrixExplorer: a Dual-Representation System to Explore Social Networks. *IEEE Transactions on Visualization and Computer Graphics* 12, 5, 677-684.

[4] Katz, L. 1947. On the Matric Analysis of Sociometric Data. *Sociometry*, 10(3), 233-241.

[5] Forsyth, E. , & Katz, L. 1946. A Matrix Approach to the Analysis of Sociometric Data: Preliminary Report. *Sociometry*, 9(4), 340-347.

[6] Moreno, J. L. 1946. Sociogram and Sociomatrix: A Note to the Paper by Forsyth and Katz. *Sociometry*, 9(4), 348-349.

[7] Henry, N. and Fekete, J. 2006. Evaluating visual table data understanding. In *Proceedings of the 2006 AVI Workshop on Beyond Time and Errors: Novel Evaluation Methods For information Visualization* (Venice, Italy, May 23 - 23, 2006). BELIV '06. ACM, New York, NY, 1-5.

[8] Bertin, J. 1981. *Graphics and Graphic Information Processing*. Berlin: Walter de Gruyter (Translated by W. J. Berg and P. Scott).

[9] Bertin, J. 1967. *Sémiologie graphique: les diagrammes, les réseaux, les cartes*. Paris: Mouton.

[10] Mueller, C., Martin, B., and Lumsdaine,A. A comparisonof vertex ordering algorithms for large graph visualization, In *Proceedings of Asia-Pacific Symposium on Visualization* 2007, 141–148.

[11] Ling, R. L. 1973. A computer generated aid for cluster analysis. *Commun. ACM* 16, 6 (Jun. 1973), 355-361.

[12] Liiv, I. "Seriation and Matrix Reordering Methods: An Historical Overview", Statistical Analysis and Data Mining, 2010, in press.

[13] Wilkinson,L. and Friendly, M. 2009. The History of the Cluster Heat Map. *The American Statistician,* 63(2),179-184.

[14] Chandrasekharan, M. P. , & Rajagopalan, R. 1989. Groupability: an analysis of the properties of binary data for group technology. *International Journal of Production Research*, *27*, 1035-1052.

[15] Kumar, K. R. , & Chandrasekharan, M. P. 1990. Group efficacy: a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology. *International Journal of Production Research*, *28*(2), 233-243.

[16] O'Brien, M. J., & Lyman, R. L. 1999. *Seriation, Stratigraphy, and Index Fossils: The Backbone of Archaeological Dating*. New York: Kluwer Academic/Plenum Publishers.

[17] Bar-Joseph, Z. , Gifford, D. K. , & Jaakkola, T. S. 2001. Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics*, *17*(S1), S22-S29.

[18] Bar-Joseph, Z. , Demaine, E. D. , Gifford, D. K. , Srebro, N., Hamel, A. M. , & Jaakkola, T. S. 2003. K-ary Clustering with Optimal Leaf Ordering for Gene Expression Data. *Bioinformatics*, *19*(9), 1070-1078.

[19] Verin, L. L. , & Grishin, V. G. 1986. Algorithm for interactive forming matrix data representation and estimation of its efficiency. *Pattern Recognition Letters*, 4, 193-200.

[20] McCormick, W. T. , Schweitzer, P. J. , & White, T. W. 1972. Problem Decomposition and Data Reorganization by a Clustering Technique. *Operations Research*, *20*(5), 993-1009.

[21] Niermann, S. 2005. Optimizing the Ordering of Tables With Evolutionary Computation. *The American Statistician*, *59*(1), 41-46.

[22] Keim, D. A. 2000. Designing Pixel-Oriented Visualization Techniques: Theory and Applications. *IEEE Transactions on Visualization and Computer Graphics* 6, 1, 59-78.

[23] Keim, D.A, Schneidewind, J., & Sips, M. 2007. Scalable Pixel-based Visual Data Exploration in "Pixelization Paradigm", *Lecture Notes in Computer Science,* Vol. 4370, Springer.

[24] Kosara, R, Bendix, F., & Hauser, H. 2006. Parallel Sets: Interactive Exploration and Visual Analysis of Categorical Data. *IEEE Transactions on Visualization and Computer Graphics* 12, 4, 558-568.

[25] Seo, J., Shneiderman, B. 2006. Knowledge Discovery in High-Dimensional Data: Case Studies and a User Survey for the Rank-by-Feature Framework. *IEEE Transactions on Visualization and Computer Graphics* 12, 3, 311-322.

[26] Rissanen, J. 1978. Modelling by shortest data description. *Automatica*, *14*, 465-471.

[27] Grünwald, P. D. 2007. *The Minimum Description Length Principle*. Cambridge, MA: MIT Press.

[28] Keogh, E. , Lonardi, S. , & Ratanamahatana, C. A. 2004. Towards Parameter-Free Data Mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, Seattle, WA, USA, August 22-25, 2004* , 206-215.

[29] Dhillon, I. S. , Mallela, S. , & Modha, D. S. 2003. Information-Theoretic Co-Clustering. In *Proceedings of The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD-2003), Washington, DC, USA, August 24 - 27, 2003,* 89-98.

[30] Faloutsos, C. , & Megalooikonomou, V. 2007. On data mining, compression, and Kolmogorov. *Data Mining and Knowledge Discovery*, *15*(1), 3-20.

[31] Wilkinson, L. 2005. The Grammar of Graphics (2nd Edition). Springer: New York.