Socio-cultural Cognitive Mapping

Geoffrey P. Morgan[1], Joel Levine[2], Kathleen M. Carley[1]

[1]Carnegie Mellon University
Pittsburgh, PA

[2]Dartmouth College
Hanover, New Hampshire

Abstract. We introduce Socio-cultural Cognitive Mapping (SCM), a method for characterizing populations based on shared attributes, placing these actors on a spatial representation. We introduce the technique, taking the reader step-by-step through the algorithm. We conclude with two example uses of the tool: attribute data about the Hatfields and McCoys, and Sampson Network Data. In the Hatfield-McCoy case, the SCM process clearly delineates members of the opposing clans as well as gender. It also places maximal distance between the groups in both clans that were primarily responsible for the feud. The Sampson Network Data example takes multiple network inputs and uses them to inform a placement. Retrospective identifications of groups within the Sampson data are reasonably well-clustered, but could benefit from additional feature creation.

1 Introduction

Frequently in social science research, we have multiple attributes of a sample of a population of interest, but we want to understand the implicit communities of the sample. Are there multiple group of actors or is the sample relatively homogenous? What attributes show strong delineation between communities? Answers to these questions may be fruitful in understanding the different reactions of the community to change, as well as to develop a more nuanced understanding of a group of interest, in that there may be multiple important communities within that group-label. To explore these questions, we introduce Socio-cultural Cognitive Mapping (SCM).

In the SCM process, the user identifies information of interest, reified either as attribute data on a node-set or network data. This information is then used to inform a set of constraints between nodes – these constraints identify optimal distances between nodes. We then place nodes at random in a 1D, 2D, or 3D space with various geometries. Nodes move in a greedy but non-local fashion to the best available position based on their constraints. After all nodes have moved to best available positions without improvement in overall fit, a Chi-Square score is calculated and reported. We do multiple iterations of each geometry of interest. The best fitting space is then returned to the user for visualization and analysis.

Socio-cultural Cognitive Modeling, p. 1, 2017.

This work is supported by the Office of Naval Research, Grant #: N000141512797

SCM bears features in common with several other multi-dimensional scaling techniques that embed nodes in a space, thereby visualizing clustering, separation and dimensions of differentiation. SCM, like these other techniques, is a dimension reduction procedure. At this level, SCM is part of a class of tools that define a clustering among nodes based on their similarity in some other space – e.g., similarity in attribute or their connections to another set of nodes.

The canonical example is MDS (Multi-Dimensional Scaling). Unlike classical MDS (Torgerson, 1958), SCM does not rely on eigenvector decomposition to reduce the dimensions. In that sense, SCM is closest to general MDS (Borg & Groenen, 2005). A key difference between SCM and MDS is that MDS takes the attributes as given; whereas, in SCM these attributes are first converted to a set of binary attributes thus giving equal weight to each "category" of information. Like MDS, SCM can identify a set of dimensions that best characterize discriminate these clusters. Another key difference is that even in general MDS the user must specify the distance metric (e.g., Euclidean or Manhattan) and the number of dimensions (e.g. 2 or 3). In SCM, the distance metrics as well as the number of dimensions are part of the optimization. A third difference is that in SCM the nodes can vary in how much "constraint" they have on the position of the other nodes, whereas in MDS procedures all nodes contribute equally. Further that contribution is also optimized over. Finally, in SCM similarity and dissimilarity can be simultaneously taken into account; whereas, in MDS only dissimilarity is considered.

A second example is principal components analysis (Jolliffe, 2002). Principle components analysis presents variables as linear combinations of all other variables, the dimensional reduction rotates the space to visualize a small number of dimensions in which distance and variation approximates that of the original space. As ordinary two-variable regression reduces a two dimensional space to a one dimensional space on which variance approximates the whole, ordinary reduction techniques reduce a high dimensional space to a low dimensional space in which variance still approximates the whole.

By contrast, SCM reduction is built from a different base: closer to the data and shedding assumptions. Typically, a "variable" is reconceived as a collection of attributes and their joint distribution is attended to directly, rather than relying on a single number proxy for the whole distribution. Conceiving "input" as a collection of attributes, SCM is not restricted to number-valued variables.

For example, "height-weight" data are sometimes used to demonstrate standard technique. For these data, standard procedure would have us improve prediction by adding variables. SCM procedure looks at the detail. It automatically notes that weight for a given height is not normally distributed but more like a Laplacian . And, with proper modeling SCM makes the joint frequencies of height and weight are more predictable — without the complexity of higher dimensions.

With high-variable data, assumptions about the space itself are shed. For example, in some cases substitution of a "Manhattan metric" for a Euclidean metric will enhance the prediction of joint frequencies — without additional parameters and without higher dimensions.

We continue this paper by describing the algorithm in detail, and then following that explanation with a case-study example of the Hatfields and McCoys followed by the Sampson Monastery data. We conclude with a summary of key points from the paper.

2 Algorithm

The SCM process has multiple steps. For additional technical details on the steps in the SCM, see Levine and Carley (2016). The user selects data to be used to inform similarity, identifies how similarity should be assessed, select a set of geometries for the nodes to be placed on, and then allow the process to proceed, with the tool returning the best fitting positions across all geometries as coordinates. Finally, once all nodes have been unable to find a better position, the layout is evaluated and a score reported. We go into more detail on each of these steps in the following sub-sections.

2.1 Selecting Data

The SCM supports multiple types of data, and one of the goals of the SCM process is to make it easier to consider node attributes and network matrices as more interchangeable. The three types of input (attributes, binary network data, weighted network data) it evaluates are each processed differently.

Attribute data is first pre-processed to identify the type of data the attribute embodies. While ORA (Carley, 2014) has multiple data types included in its attribute schema (e.g., number, number category, text, text category), these data types are only rarely enforced. Instead, we process the values present for that attribute to determine how it shall be treated in the SCM process.

- Attributes with exactly two distinct values (including a blank value) are treated as binary attributes.
- Attributes which are not numbers are treated as categorical attributes, these are then turned into a set of binary variables.
- Attributes which are numbers but there are seven or less distinct values are also treated as categorical attributes, and then turned into a set of binary variables.
- Attributes which are numbers and have more than 7 distinct values are treated as quantitative attributes, five quantitative bins are identified (which are intended to present an equal distribution of values but when many numbers are at the minimum or maximum value may not be perfect) and binary variables are created for each bin.

Binary network data is treated as a set of binary attributes. For the SCM process to recognize that a network is binary, the checkbox “binary values” must have been selected on the network tab in ORA. Otherwise, the network data is treated as weighted network data.

Weighted network data is treated explicitly as a set of constraints for the SCM process, while the previous two inputs are used to generate a similarity matrix (more

details will be given on how the similarity matrix is calculated shortly). Link weights are assumed to be event rather than distance counts, and so high counts indicate closer distance constraints for the SCM process. If your values instead a distance metric (e.g., number of miles to a given city), the values will need to be inverted before the network can be usefully used in the SCM process.

2.2 Generating the Similarity and Constraint Matrices

A constraint matrix is used to inform the ideal position of points in the various evaluated geometries. The constraint matrix is calculated as a transformation of a similarity matrix or a weighted network if that option is selected. The similarity matrix is generated based on the binarized SCM attributes rather than the network or node-attributes from which they spring. Multiple control variables can inform the generation of the similarity matrix calculation, including:

- Remove Redundant Attributes: This setting examines whether two SCM attributes are redundant, and if so, removes one of the two attributes. Two binary attributes (X, Y) are redundant if every node that has attribute X has attribute Y, and every node that does not have attribute X does not have attribute Y.
- Remove Mutually Exclusive Attributes: This setting examines whether two SCM attributes are mutually exclusive, and if so, removes one of the two attributes. Two binary attributes (X, Y) are mutually exclusive if every node that has attribute X does not have attribute Y, and every node that does not have attribute X does have attribute Y. As an example, in our Hatfield-McCoy case-study, every person that is a Hatfield is not a McCoy, and every McCoy is not a Hatfield.
- Use Negative Similarity: By default, the SCM uses only positive similarity to inform the similarity matrix. For every attribute two nodes share, their similarity increases. With this option, the SCM uses both positive and negative similarity, where negative similarity is defined as two nodes gain in similarity the more attributes they both do not have. Negative similarity is most appropriate when "not being part of something" is an important element of the social context in which the actors are operating. In future iterations of the tool, the user will be able to control which attributes inform both positive and negative similarity.
- Treat Zero-Similarity as Non-Information: Two nodes may have nothing in common, and have a similarity score of 0. If this option is selected, then this element of the constraint matrix will be ignored and these two nodes will be able to "slide" past each other on the geometry. If it is not, then a maximal distance value is placed in the cell. By default, this option is selected.

Once the similarity matrix has been generated, we convert the similarity matrix into the constraint matrix. There are multiple ways of generating a constraint matrix, but for the examples in this work, we use a simple inverted similarity. Future iterations will include other transformations.

2.3 Moving nodes to satisfice constraints

After calculating the constraint matrix, we run a number of iterations across each geometry and attenuation setting, note that the SCM supports 1D, 2D, and 3D node placements. Typical geometry and attenuation settings are 0.7, 1.0, 2.0, and 3.0. We confine ourselves to 2D Euclidean spaces for this paper. See Figure 1 for example geometries.

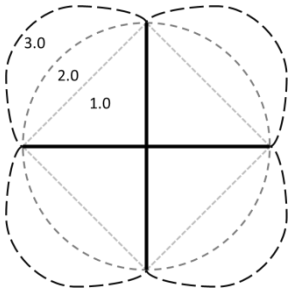

Fig. 1. The SCM process supports multiple geometries, including Manhattan, Euclidean, and more, geometries presented on a 2D plane for convenience.

For each run, we place the nodes at random. We then move the nodes to best available positions via local and jump movements. We use jump movements to avoid local optima in a neighborhood.

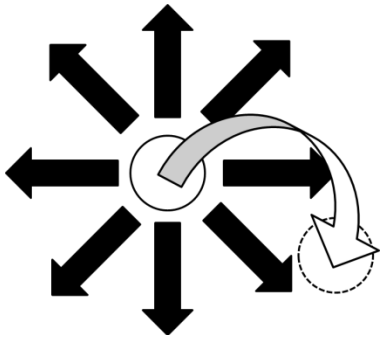

Fig. 2. Node movements include both local (black arrows) and jump (white arrow) movements. A 2D plane is used strictly for illustrative purposes.

Currently, the tool uses a greedy algorithm to determine the best movement for the node at each point. The random jump is the best random position found after 100 attempts. While greedy algorithms tend to fall into local optima, we believe this addition of random jumps greatly softens the negative impact. We have seen that the standard deviations of the ChiSquare fits have been significantly reduced in every geometry for tested data-sets since we added the non-local jump to the process. Jumps are most common early on as the initial random configuration is mostly discarded. Eventually, most movements are local, with some jumps still occurring on rare occasions.

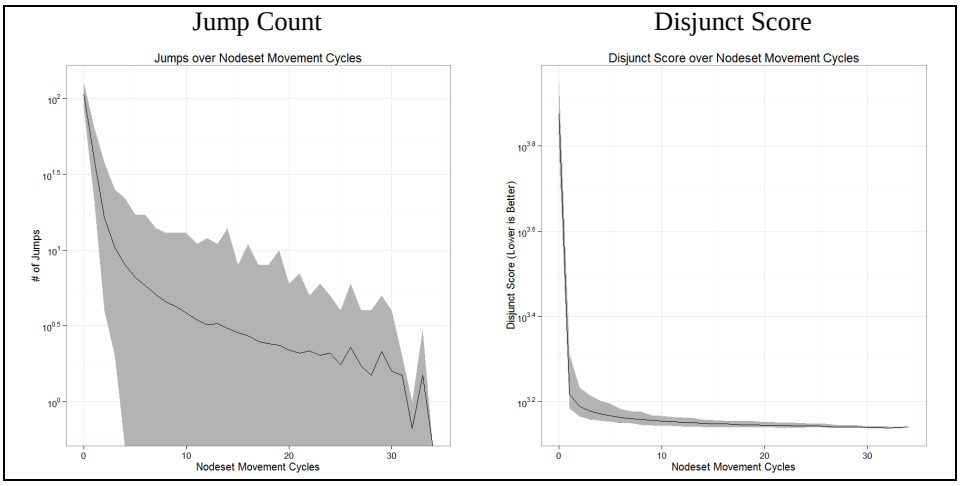

Fig. 3. Random jumps decrease greatly over time and greatly improve the overall fit of the models. Weighted Averages, Max and Min values drawn from iterations of the Hatfield-McCoy data. Disjunct score is converted to Chi-Square before being presented to the user. Nodes may jump multiple times in a cycle.

Future iterations will be outfitted with a simulated-annealer optimization process.

2.4 Evaluating fit and returning results

Once all nodes have been unable to find a better position, the SCM evaluates the overall fit based on the constraint set and calculates a Chi-Square. We are using the Chi-Square in this fashion as a goodness of fit, and not as a statistical evaluation. We calculate the Chi-Square, and report the best fit per geometry and attenuation setting, the standard deviation of Chi-Square per geometry and attenuation setting, and the best Chi-Square over all.

For this best fitting Chi-Square, we return (by default) the coordinate positions for all nodes, but we can also report back the Similarity Matrix, the Constraint Matrix, and the Chi-Square Error Cell Matrix. Because we return the Error Matrix, we can visualize the level of satisficing and conflict in the node's current positions, providing more confidence in the ultimate groupings. We can also return the transformed SCM binary attributes to each node. Once we have node coordinate positions, we can use visualizations to examine the resulting groupings.

3 The Hatfield-McCoy Case Study

In the Hatfield-McCoy Case study, we wanted to demonstrate the utility of the technique on a relatively well-known historical scenario. The Hatfields and McCoys were two rural families living across the Big Sandy River from each on the West Virginia and Kentucky sides respectively. The two families were in a bitter feud from

1863 to 1891. "Devil Anse" Hatfield led the Hatfields of West Virginia while Randolph McCoy led the McCoys of Kentucky. The Hatfields were wealthier, owning a timbering operation and serving in local government, while the McCoys were farmers. Both families made and sold moonshine. Intermarriage between the families happened before and even during the feud, which was first kicked off by the death of Asa McCoy, who was the only member of either family who sided with the Union during the Civil War. The McCoys did not forget, and eventually killed a mutual relative of both Hatfields and McCoys, Bill Staton, who sided with the Hatfields on the ownership of a hog. The feud escalated over time until eventually most of the McCoys moved to Pikesville (20 or 30 miles farther from the border) to escape the violence and Devil Anse was arrested after an armed shootout.

We used multiple sources to generate this dataset, since the exact composition of the clans is difficult to determine, and several sources contradict. This dataset is primarily intended to for illustrative purposes. We identified 66 members of the two families, and for each individual, we identified these binary attributes: Man, Woman, Hatfield, McCoy, Harmed Hatfield, Harmed McCoy, Killed in Feud, Intermarried, Devil Anse Family, and Randolph Family.

Table 1. Binary Attributes of the Hatfield-McCoy Case Study

Attribute	*Percent with Attribute*
Man	75.8%
Woman	24.2%
Hatfield	45.5%
McCoy	54.5%
Devil Anse (Hatfield) Family	18.2%
Randolph (McCoy) Family	24.2%
Intermarried	10.6%
Harmed Hatfield	7.6%
Harmed McCoy	6.1%
Killed in Feud	16.7%

Given the nature of the feud, it seems clear that Hatfield and McCoy would clearly differentiate the nodes. We would also expect that gender, given the era, would be clearly differentiated. Those that are intermarried are probably not clearly differentiated. We would hope that Devil Anse and Randolph family members are widely separated.

We ran the SCM removing redundant but not mutually exclusive attributes, counted both positive and negative similarity, and used a Euclidean space. The Chi-Square was 907.2 out of 2145 degrees of freedom. The standard deviation of the Chi-Square in this geometry was 225.3. This model has removed a substantial amount of noise – the Wilson-Hilferty Z-Score approximation (Wilson & Hilferty, 1931) is -24.58.

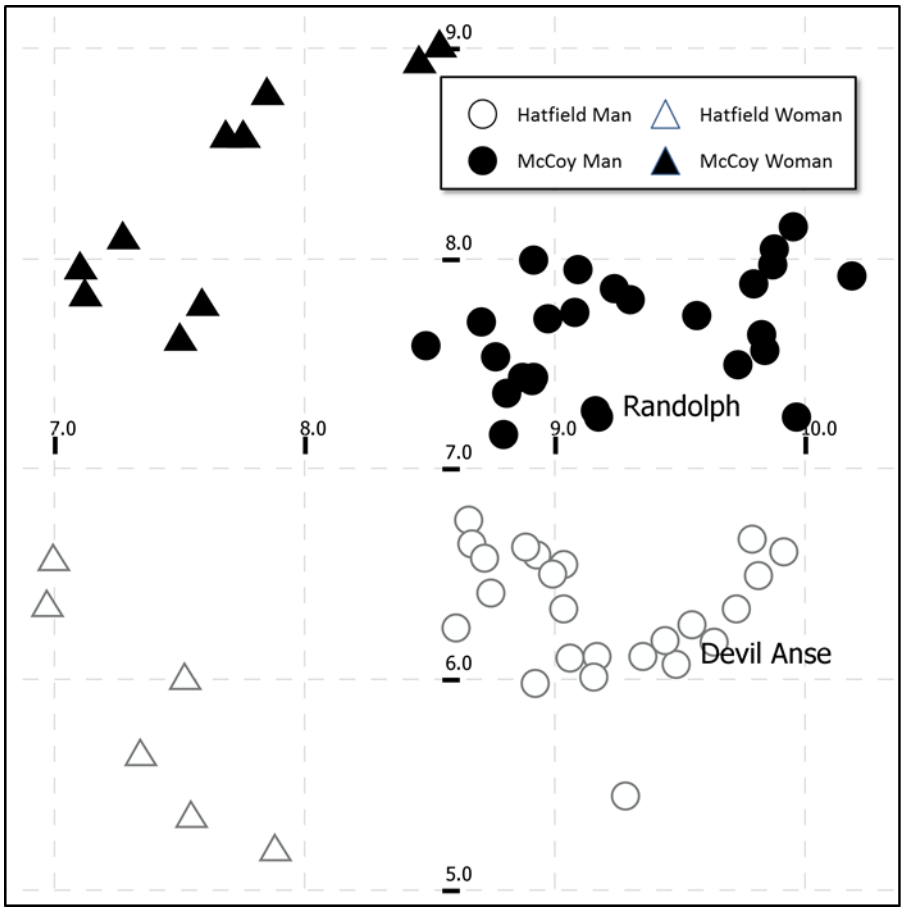

Fig. 4. Position of nodes reflects clean separation between clans (color) and gender (shape). The leaders of the two clans are labeled.

4 The Sampson Monastery Case Study

The Sampson Monastery data was first collected by Sampson when he explored the relationships of monks in a cloister. A previously stable collection of monks, later labeled retrospectively the "Loyal Opposition", were joined by a new cohort of monks, labeled the "Young Turks", who questioned policies of the group and attempted to enact change. A third and smaller group, "Outsiders", were not accepted by either of the two groups. Eventually, most of the monks left the monastery.

Sampson collected panel data over time about who liked and disliked others. Each monk could offer up to three names for both who they liked and who they disliked. There are various ways of representing this data, but the SCM process gave us an option otherwise not easily available: we want to provide the SCM process information on both who likes who and who dislikes who, but it's impossible to contain

both network semantics in a single binary matrix. However, converting each of the binary matrices to attributes allowed us to create a set of relationship attributes that accurately describe the state of affairs within the Monastery without leaving out information. We did create special "symmetric like" and "symmetric dislike" features. We withheld the retrospective group affiliations from the attribute data and only added them afterwards to color nodes.

We used a Euclidean space, assumed that shared absence of an attribute matters, and assumed zero similarity was important. These assumptions seemed appropriate given the context of the Sampson Monastery data.

When we run this model in a Euclidean space, we get a best Chi-Square of 48.48, with a standard deviation of the Chi-Square being 34.97. Given the 153 degrees of freedom, this is a Wilson-Hilferty Z-Score approximation of -8.13, and thus the model has removed substantial noise.

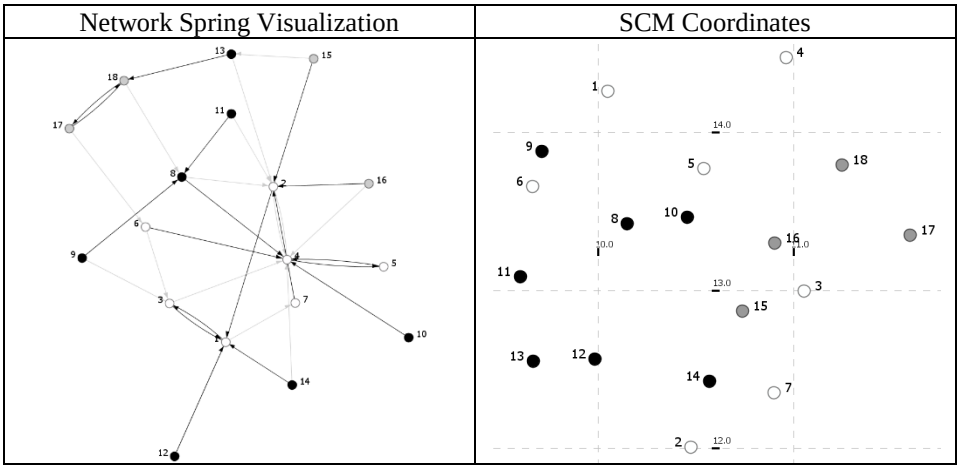

Fig. 5. A Network Visualization and SCM Coordinates are compared. In the network graphic, dark lines indicate like, while lighter lines indicate dislike. On the SCM visualization, The Young Turks (Black) dominate the west, with the Loyal Opposition (White) divided by the mass of Outsiders (Gray).

5 Discussion and Conclusion

In this work, we have introduced Socio-cultural Cognitive Modeling (SCM), a technique we developed to characterize populations and identify implicit groups. We have used the technique in two separate illustrative use-cases. In each case, it is clear that good features lead to better classification.

To support use, SCM is available in ORA (Carley, 2014). ORA is an analysis and visualization toolkit for high dimensional network and social network data. Network images shown here in (e.g., Figures 4 and 5) were done in ORA. Consequently, SCM is currently available for use by the community.

More work remains to be done. We plan to add a sophisticated optimization package, rather than relying on greedy stochasticity. This should support finding an SCM

configuration for complex data with an improved overall goodness of fit. We are also interested in evaluating our best fit positions in other ways than a Chi-Square. We also plan to add the ability to support counterfactual simulation with the tool.

Nonetheless, the process already provides a novel way to take advantage of information available either as attributes or network data to produce an estimate of each node's appropriate position in relation to each other. Implicit groups of significant import may be discovered.

References

Borg, I., & Groenen, P. J. (2005). *Modern multidimensional scaling: Theory and applications*. New York: Springer Science & Business Media.

Carley, K. M. (2014). ORA: a toolkit for dynamic network analysis and visualization *Encyclopedia of Social Network Analysis and Mining* (pp. 1219-1228): Springer.

Jolliffe, I. (2002). *Principal component analysis*: Wiley Online Library.

Levine, J. H., & Carley, K. M. (2016). *SCM System*. Retrieved from Pittsburgh, PA:

Torgerson, W. S. (1958). *Theory and methods of scaling*. New York: Wiley.

Wilson, E. B., & Hilferty, M. M. (1931). The distribution of chi-square. *Proceedings of the National Academy of Sciences, 17*(12), 684-688.