

# MVN-Reduce: Dimensionality Reduction for the Visual Analysis of Multivariate Networks

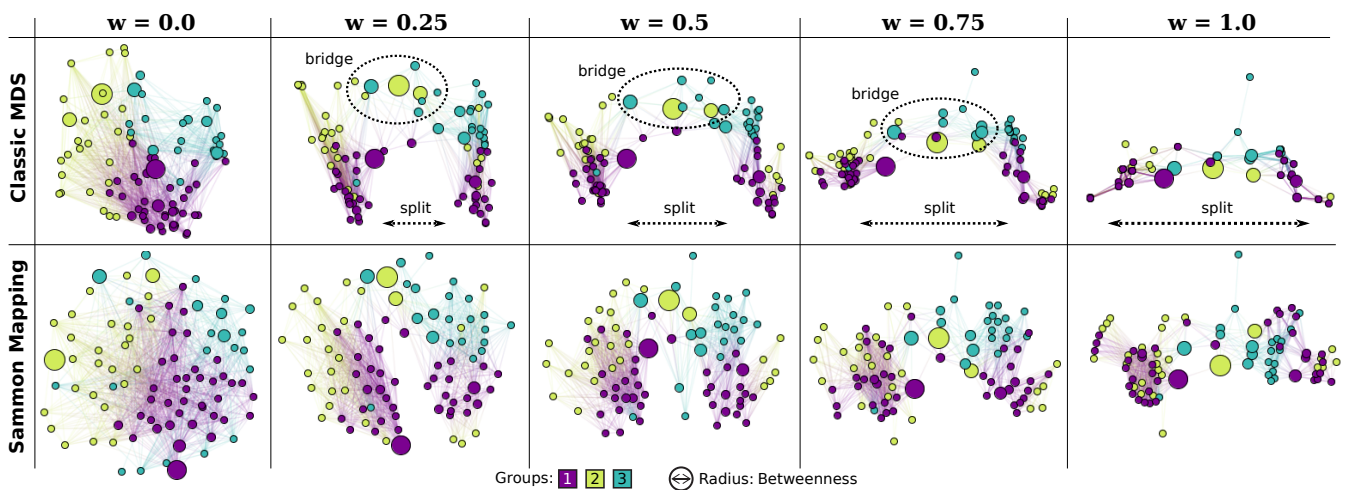
R. M. Martins<sup>1</sup>, J. F. Krueger<sup>2,3</sup>, R. Minghim<sup>4</sup>, A. C. Telea<sup>2</sup>, and A. Kerren<sup>1</sup>

<sup>1</sup>Linnaeus University, Växjö, Sweden

<sup>2</sup>University of Groningen, The Netherlands

<sup>3</sup>École Nationale de l'Aviation Civile, France

<sup>4</sup>University of São Paulo, Brazil



**Figure 1:** Co-authorship network (VisBrazil data set) visualized with MVN-Reduce, using Classic MDS, Sammon Mapping, and different values of the parameter  $w$  (see Eq. 2). Node colors encode attribute-based groups (from Bisecting K-Means) and node sizes encode betweenness. The visualization shows the split of the network into two main communities of papers, which are bridged by papers with common co-authors. At the same time, the nodes' content-based similarities and dissimilarities also influence the layout.

## Abstract

The analysis of Multivariate Networks (MVNs) can be approached from two different perspectives: a multidimensional one, consisting of the nodes and their multiple attributes, or a relational one, consisting of the network's topology of edges. In order to be comprehensive, a visual representation of an MVN must be able to accommodate both. In this paper, we propose a novel approach for the visualization of MVNs that works by combining these two perspectives into a single unified model, which is used as input to a dimensionality reduction method. The resulting 2D embedding takes into consideration both attribute- and edge-based similarities, with a user-controlled trade-off. We demonstrate our approach by exploring two real-world data sets: a co-authorship network and an open-source software development project. The results point out that our method is able to bring forward features of MVNs that could not be easily perceived from the investigation of the individual perspectives only.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Viewing Algorithms

## 1. Introduction

Relational data sets—also called networks or graphs—are present in many application areas, such as social network analysis, software comprehension, biology, and medicine. Such networks are

usually depicted by node-link metaphors, an approach that highlights relationships (edges) between actors (nodes) and groups of actors [BETT98]. A more general view on relational data sets leads to Multivariate Networks (MVNs)—graphs whose nodes and/or

edges have (multidimensional) attributes. There are many open challenges in the visualization of MVNs [KPW14]. One of these challenges is the effective, scalable, and interactive analysis and exploration of multiple relationships between nodes, i.e., edge-based and attribute-based similarities.

This paper’s main research question is: can we create meaningful visualizations of MVNs by reducing their high-dimensional variable space? We address this question by incorporating node attributes in the computation of graph layouts, so that the resulting drawing emphasizes both node connections and attribute-based node similarities. Layouts are generated using Dimensionality Reduction (DR) methods, which are commonly employed to project multidimensional data to lower dimensional spaces [vdMPvdH09] (hence the name MVN-Reduce). The connectivity- and attribute-based similarities of an MVN’s nodes are combined in a single distance matrix, based on a user-defined trade-off between the two types of similarities. The resulting distance matrix is then used as input to any suitable DR technique, yielding the low-dimensional (2D) node positions. The final results are depicted using a mix of traditional scatterplot and node-link metaphors. In summary, our contributions are:

- a new way to visualize MVNs using DR methods which is scalable in both the number of nodes and attributes, is simple to implement, and is nearly fully automatic;
- a user control for the trade-off between relational and multivariate data aspects in the final layout; and
- a demonstration of the benefits of MVN-Reduce for the exploration of two MVNs from software engineering and co-authorship analysis.

## 2. Background and Related Work

Let  $G = (V, E)$  be a graph with nodes  $V = \{\mathbf{x}_i\}_{1 \leq i \leq |V|}$  and edges  $E = \{(\mathbf{x}_i \in V, \mathbf{x}_j \in V)\}$ . A multivariate network (MVN) extends  $G$  by adding  $n$  attributes (or dimensions) to nodes and/or edges; the MVN is thus both a relational and a multivariate data set [KPW14]. In this work, we consider only node attributes of quantitative (continuous) type, modelled as a matrix  $A^V = (a_{ij}^V) \in \mathbb{R}^{|V| \times n}$ . An MVN is then denoted as  $G_A = (V, E, A^V)$ , with  $a_i = \{a_{ij}^V\}_{1 \leq j \leq n}$  being the  $n$ -dimensional vector of attribute values of node  $\mathbf{x}_i$ .

**Graph Drawing.** The Graph Drawing (GD) field studies the visualization of the *relational structure* of graphs, i.e., how to *embed* a graph  $G$  in 2D or 3D according to  $E$  [vLKS\*11]. GD techniques can handle large graphs of millions of nodes [GKN04], can reduce edge-crossing clutter to show the graph core structure [vdZCT16], or can use drawing styles or conventions to highlight specific graph substructures [EGK\*02]. However, for showing node and/or edge attributes, GD techniques are usually limited to classical encodings of a few attributes into shape, color, size, textures, or labels [Aub04, HB05]. In extreme cases, nodes themselves can become full-fledged visualizations of multidimensional data [BT09, JDK10], with a trade-off between the number of nodes and the number of attributes per node that can be shown.

**Dimensionality Reduction.** Multidimensional data sets can be explored by mapping the high-dimensional data to lower-dimensional, interactive, and easy-to-use visualizations by DR methods, so that aspects of the *high-dimensional structure* of the data are kept in the final layout. DR methods are visually scalable and computationally efficient in both the number of observations

and dimensions, work automatically, and are usually depicted by scatterplots [vdMPvdH09, SVP14, SZS\*17].

Some DR methods directly use the attribute values  $a_i$  as input, such as the well-known Principal Component Analysis (PCA) [Jol02] or, more recently, LAMP [JPC\*11]. Other methods, known globally as Multidimensional Scaling (MDS) [CC01], take as input a real-valued distance matrix  $D = (d^n(\mathbf{x}_i, \mathbf{x}_j)_{ij})$ , where  $d^n$  is a distance metric over  $\mathbb{R}^n$ . Such methods can be seen as a mapping

$$P(D) : \mathbb{R}^{|V| \times |V|} \rightarrow \mathbb{R}^{|V| \times 2} \quad (1)$$

whose goal is to preserve the original similarities between the observations  $a_i$  in the final 2D layout. Several methods exist in this class, as follows. Sammon Mapping [Sam69] uses optimization to minimize an error (or stress) function between the original and the final distances between every observation pair. ISOMAP [TdSL00] uses nearest-neighbor distances to estimate the underlying geometry of the data’s high-dimensional manifold and uses classic MDS to depict the results. Landmarks MDS [ST04] and Pivot MDS [BP07] achieve speed-ups by using classical MDS on a subset of representative observations and fit remaining ones by local interpolation. Similarly, LSP [PNML08] positions representatives by a force-based scheme and fits the remaining observations by Laplacian smoothing. Distance-based DR methods have also been used to generate graph layouts, usually by applying  $P(D)$  directly to the distance matrix  $D$  obtained from the graph-theoretic (shortest path) distances between all nodes [KS80, GKN04].

**Combined Layouts.** Techniques for visualizing MVNs commonly position nodes based on either attributes or relations, offering different visual metaphors for filtering and browsing the two perspectives in single [Wat06, PW06] or multiple views [AS07, BCD\*10]. While these approaches take advantage of the unique characteristics of the two semantically-different perspectives of an MVN, they lack the potential advantages of combined 2D embeddings, such as the easy comparison of nodes and scalability on the numbers of nodes and attributes. In some cases the edges of the network also carry information related to node attributes. Edges may be created by applying thresholds to attribute similarities, such as code clones [Han13, VT14], or be enhanced with weights based on attributes of the nodes, such as categorical information [SGCP09, MAH\*12] or common topics in communications between actors [VNCD08, GKN\*09]. These are domain-specific techniques that may not be suitable for general MVNs with any number of continuous (quantitative) dimensions. In GrouseFlocks [AMA08], a GD method is modified to use attribute-based hierarchical clusters automatically extracted from the data. However, as with previous techniques, it does not directly treat similarities between nodes that are not already linked by an edge.

Our proposal (MVN-Reduce) aims to improve on previous work described in this section by (i) considering the two distinct perspectives of an MVN in a joint fashion when embedding the MVN’s nodes in 2D, (ii) allowing the analyst to explore if and how these two different perspectives are correlated, and (iii) taking advantage of the advances in DR research for the visualization of MVNs.

## 3. MVN-Reduce

MVN-Reduce aims to generate DR layouts of MVNs that are influenced by both the relational and the attribute-based similarities of nodes. The details of how to achieve this are described next.

**Step 1:** Turn relational and attribute-based similarities into comparable models. The relational data ( $G$ ) of an MVN is modeled as a distance matrix  $D_E = (d_{ij}^E)$ , where  $d_{ij}^E$  represents the connection strength between two nodes  $\mathbf{x}_i$  and  $\mathbf{x}_j$ , derived from the weighted shortest path distance in  $G$  between  $\mathbf{x}_i$  and  $\mathbf{x}_j$ , or any other structural distance metric defined over a graph  $G$ . Separately, the attribute-based distances between each pair of nodes ( $\mathbf{x}_i, \mathbf{x}_j$ ) of an MVN are compiled into a distance matrix  $D_A = (d_{ij}^A)$ . The value  $d_{ij}^A$  reflects the similarity of attributes  $a_i$  and  $a_j$ , and is computed using  $n$ -dimensional distance metrics such as Euclidean, Manhattan, or cosine. These transformations are done in similar ways in previous work [MAH\*12].

**Step 2:** Combine the similarities into a single model. We combine the relational and attribute-based similarities  $D_E$  and  $D_A$  into a single distance matrix  $D$  by linear interpolation:

$$D = I_w(D_E, D_A) = \frac{w}{\|D_E\|_F} D_E + \frac{1-w}{\|D_A\|_F} D_A. \quad (2)$$

The parameter  $w \in [0, 1]$  is a weight that models the trade-off between connectivity and attributes, and  $\|\cdot\|_F$  is the Frobenius norm of a matrix. The Frobenius norm division ensures that the interpolated values are of comparable scale, without changing the relative pairwise distances between nodes.

**Step 3:** Create a low-dimensional embedding from the resulting combined model. The resulting distance matrix  $D$  (Eq. 2) is used as input into any DR method that accepts a distance matrix as input (thus, of MDS type), which generates 2D node positions. Finally, the MVN is drawn using classical node-link techniques, with optional attribute encoding into node size, shape, and color.

Some aspects of Eq. 2 are important to highlight. First, the above-mentioned normalization removes any specific requirements on the types of distance metrics used to compute  $D_E$  and  $D_A$ . Depending on the MVN, the task, and the application, different distance metrics for both the graph’s structure and the high-dimensional attributes can be used, such as domain-specific similarities [Han13, VT14]. This will not sacrifice the generality of the technique. Second, finding an optimal weight  $w$  is largely context-dependent, involving factors such as the specific task being performed, whether the user is more interested in the graph’s structure or its attributes, or the specific distribution of distance values in  $D_E$  and  $D_A$  for a given MVN. Hence, instead of defining a fixed  $w$ , we provide interactive means for the user to browse the space of possible combinations and decide, on a case-by-case basis, which trade-off  $w$  is best for each situation.

#### 4. Applications

In this section we present the use of MVN-Reduce for exploring co-authorship networks and multivariate software networks, and show which new insights MVN-Reduce helped to obtain atop what visual exploration using classical GD and DR layouts provide.

**Co-authorship Networks.** The *VisBrazil* MVN consists of papers (nodes) published by Brazilian visualization researchers from 2003 to 2010 [MAH\*12]. The edges represent paper co-authorship and are weighted by the number of common co-authors. This is complemented with attributes extracted from the papers’ abstracts and represented with the classical Vector Space Model (VSM) [Sal86].

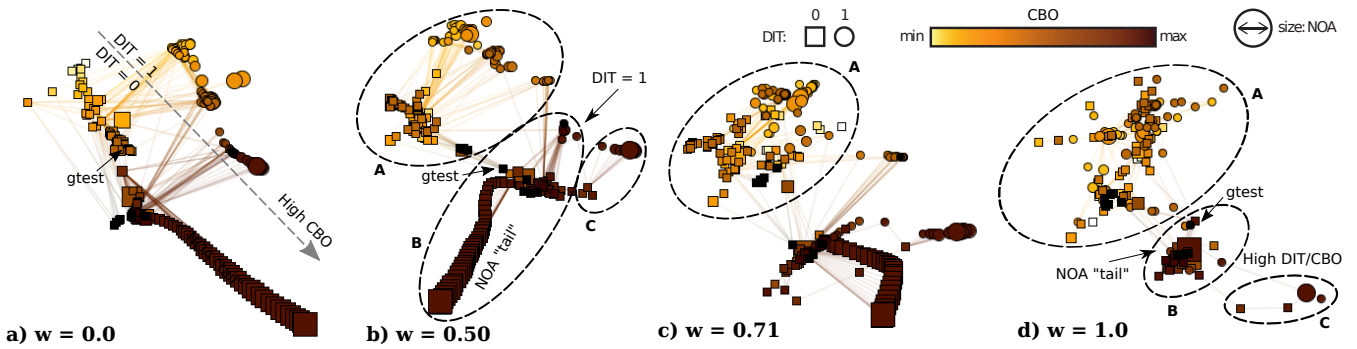
Fig. 1 shows two sets of layouts for *VisBrazil* created with MVN-Reduce, using two DR techniques to project  $D$ : Classic

MDS [CC01] (top row) and Sammon Mapping [Sam69] (bottom row). In both cases,  $D_E$  captures the length of the shortest path between two nodes, i.e., the graph-theoretic or geodesic distance [BETT98], and  $D_A$  contains the cosine-based distances between the VSM vectors of each node, a common metric to compare text documents [FS07]. The columns in Fig. 1 correspond to different values of  $w$  ranging from  $w = 0$  (show attributes only) to  $w = 1$  (show connectivity only). By varying  $w$  between 0 and 1, we smoothly change the view between attribute-only and structure-only, which further helps tracing how nodes in the two views correspond to each other. View interpolation is well known in information visualization (see, e.g., [HTCT14]). However, an important difference exists: we do not interpolate the 2D view  $P(D)$ , but the high-dimensional input data  $D$ . As such, all intermediate views obtained when varying  $w$  from 0 to 1 correspond to valid projections. To allow the user to easily compare how the different layouts map content-based similarity, the set of nodes was partitioned into three color-coded groups based on their attribute values, using the Bisecting K-Means algorithm [SKK00]. Additionally, to allow the comparison of how the layouts encode the MVN’s topology, the nodes’ sizes reflect their betweenness centrality [Bra01]. Edges are drawn with colors that linearly interpolate those of their endpoint nodes.

The leftmost layouts ( $w = 0$ ) show well-defined and separated content-based groups, but are highly cluttered regarding edge-crossings. As  $w$  increases, we see a “split” in the network, as some nodes move to form tightly-connected communities in the left and right regions of the layout. At the same time, a subset of nodes forms a “bridge” between the communities, including most of the nodes with the highest betweenness values in the MVN. This matches the expected behavior of nodes with high betweenness—to connect many other nodes through shortest paths. The original attribute-based groups are still visible up to  $w = 0.75$  even after the split: Group 1 (purple) is divided into two, occupying the lower parts of the two communities; Group 2 (green) stays mostly on the left (apart from two high-betweenness nodes); and Group 3 (cyan) occupies the upper part of the layout, mostly on the right side. The split gets clearer as we approach the rightmost layout ( $w = 1$ ). This is not, however, a good *overall* view of the MVN, as nodes from the three (initially visible) content-based groups have collapsed into each other in the layout’s cluttered view.

The second row of Fig. 1 presents the results of using the nonlinear DR method Sammon Mapping. Nonlinear DR methods are known to generate results that are better adapted to each data set’s own high-dimensional nonlinear manifold structures [TdSL00, RS00, vdMPvdH09]. As such, the results of MVN-Reduce applied to *VisBrazil* are improved, as the resulting layout and node groups are less cluttered. The flexibility of MVN-Reduce in allowing the use of any distance-based DR method makes it possible for the analyst to explore the available techniques and take advantage of such improved results.

Summarizing, the main contributions of MVN-Reduce for the exploration of the *VisBrazil* data set are: (i) the combined views given by  $0.25 \leq w \leq 0.75$  make it possible to identify, at different levels of detail (given by  $w$ ), the main characteristics of both attribute- and edge-based similarities between nodes of the MVN; and (ii) the nodes’ positions respect, at the same time and with varying trade-offs, the arrangement of the graph as two separate (but bridged) strongly-connected co-authorship communities (which is a feature of the network topology [MAH\*12]) and the content-



**Figure 2:** Layouts of the *caffe* data set created with the LSP projection technique and Euclidean distances, with  $0.0 \leq w \leq 1.0$ . The three visually-coded metrics are: Coupling between objects (CBO), Depth of Inheritance Tree (DIT), and Number of Attributes (NOA).

based similarities that make it possible to distinguish the color-coded groups, which is a feature of the nodes' attributes.

**Multivariate Software Networks.** In this section we use MVN-Reduce to explore the active open-source C++ project *caffe*, a deep learning framework [JSD\*14]. The nodes are classes and their attributes are 37 code metrics frequently used in software maintenance, including object-oriented measures (e.g., structural complexity and coupling) and descriptive measures (e.g., total number of classes, methods, and attributes) [LM06]. The edges are derived from method calls and represent the strength of the two-way dependencies between classes: the weight of an edge  $(x_i, x_j)$  is the sum of the number of method calls from  $x_i$  to  $x_j$  and from  $x_j$  to  $x_i$ .

One common way to visually explore source code is to use a GD method to create a 2D layout of classes according to their dependencies, which helps software analysts to find groups of tightly-connected nodes that form the core components of a project. However, such a view shows only system *structure* but offers no insight on how classes are related in terms of similar metric values. With this problem in mind, we show how MVN-Reduce can help us answer the following question: “How can a software analyst find dependency groups in source code and further investigate how they behave internally regarding quantitative metrics?”

Fig. 2 shows layouts for the *caffe* data set created with MVN-Reduce using Euclidean distances ( $D_A$ ), shortest-path distances ( $D_E$ ), and the nonlinear DR method Least Square Projection (LSP) [PNML08]. The values of three selected metrics are visually encoded: Coupling Between Objects (CBO, encoded in color), Depth of Inheritance Tree (DIT, encoded in node shape), and Number Of Attributes (NOA, encoded in node size). In Fig. 2(a), nodes are spread according to their CBO values from top-left to bottom-right, as shown by the color gradient along this diagonal dashed line. A gap clearly separates two bands of nodes located on the two sides of the diagonal, corresponding to the two different DIT values in the data set:  $DIT = 0$  below the diagonal and  $DIT = 1$  above it. Additionally, the nodes in the tail-like structure in the bottom of the layout have high CBO and low DIT values, but vary significantly with respect to their NOA values. On the other hand, Fig. 2(d), with  $w = 1$ , shows characteristics of the network topology. Three tightly-connected node groups—labeled A, B and C—are visible, with Group A containing mainly low-CBO nodes and Group B the nodes in the NOA “tail” in Fig. 2(a).

Looking now at the intermediate views, we see that Fig. 2(b) shows the same three connectivity-based groups found in Fig. 2(d).

This shows the ability of MVN-Reduce in answering the research question: most of the complexity of Group B is lost in Fig. 2(d), since its nodes are laid out in a very small area. The layout in Fig. 2(b), however, was able to unfold this group into a more meaningful presentation regarding its attributes, while still keeping it separate from the other two structural groups (A, C). For instance, the NOA “tail” completely disappeared in Fig. 2(d); an analyst using this classical structure-only view would not have discovered such a unique attribute-based distribution of nodes. The tail, however, is visible in Fig. 2(b). Similarly, Fig. 2(b) allows us to identify a small node subgroup in B with  $DIT = 1$ , which is indistinguishable in the structure-only view (Fig. 2(d)). Regarding Group A, Fig. 2(b) shows that it is split quite evenly into nodes with  $DIT = 0$  and  $DIT = 1$  and that most edges are between nodes with different DIT values, something we also see in the attribute-only view on the left. When  $w \geq 0.71$ , this characteristic of the MVN is no longer clear (see Figs. 2(c-d)). We conclude that the layout in Fig. 2(b), with  $w = 0.5$ , is the best view to answer the research question among the presented ones, as it visually separates tightly-connected groups and also shows attribute-value distributions.

## 5. Conclusion

We have presented MVN-Reduce, a method that generates node-link views of Multivariate Networks (MVNs). The core of MVN-Reduce is the unification of the concepts of connectivity-based and attribute-based similarity of nodes of an MVN in a single real-valued distance matrix. This matrix is used with existing DR methods to create 2D layouts that reflect a continuum of views between an attribute-only and a structure-only one. Users can interactively change the mix of information reflected in the view, thereby allowing both control of what the view shows and a smooth transition between different views for visual linking purposes. MVN-Reduce is simple to implement, generic, and easy to use. Future work will consider (i) using more than two perspectives (structure and attributes), leading to the exploration of an MVN along additional dimensions; (ii) exploring the automatic suggestion of DR methods and parameter settings based on the input data set; and (iii) evaluating the technique more extensively with user studies, comparative analyses and applications with larger and more complex data sets.

## Acknowledgements

This work was partly supported by CNPq, FAPESP, project CAPES-NUFFIC 028/11, and project MOTO (H2020-SESAR-2015-1), grant 699379, offered by the European Commission.

## References

- [AMA08] ARCHAMBAULT D., MUNZNER T., AUBER D.: Grouse-Flocks: Steerable exploration of graph hierarchy space. *IEEE Transactions on Visualization and Computer Graphics* 14, 4 (2008), 900–913. 2
- [AS07] ARIS A., SHNEIDERMAN B.: Network visualization by semantic substrates. *Information Visualization* 12, 5 (2007), 733–740. 2
- [Aub04] AUBER D.: Tulip—A huge graph visualization framework. In *Graph Drawing Software*. Springer, 2004, pp. 105–126. 2
- [BCD\*10] BEZERIANOS A., CHEVALIER F., DRAGICEVIC P., ELMQVIST N., FEKETE J.-D.: GraphDice: A system for exploring multivariate social networks. *Computer Graphics Forum* 29, 3 (2010), 863–872. 2
- [BETT98] BATTISTA G. D., EADES P., TAMASSIA R., TOLLIS I. G.: *Graph Drawing: Algorithms for the Visualization of Graphs*, 1st ed. Prentice Hall PTR, 1998. 1, 3
- [BP07] BRANDES U., PICH C.: Eigensolver methods for progressive multidimensional scaling of large data. In *Graph Drawing. GD 2006. Lecture Notes in Computer Science*, vol 4372, Kaufmann M., Wagner D., (Eds.). Springer, Berlin, Heidelberg, 2007, pp. 42–53. 2
- [Bra01] BRANDES U.: A faster algorithm for betweenness centrality. *The Journal of Mathematical Sociology* 25, 2 (2001), 163–177. 3
- [BT09] BYELAS H., TELEA A.: Visualizing multivariate attributes on software diagrams. In *13th European Conference on Software Maintenance and Reengineering* (2009), pp. 335–338. 2
- [CC01] COX T. F., COX M. A. A.: *Multidimensional Scaling*. Chapman & Hall, 2001. 2, 3
- [EGK\*02] ELLSON J., GANSNER E., KOUTSOFIOS L., NORTH S. C., WOODHULL G.: Graphviz—open source graph drawing tools. In *Graph Drawing. GD 2001. Lecture Notes in Computer Science*, vol 2265, Mutzel P., Jünger M., Leipert S., (Eds.). Springer, Berlin, Heidelberg, 2002, pp. 483–484. 2
- [FS07] FELDMAN R., SANGER J.: *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge university press, 2007. 3
- [GKN04] GANSNER E., KOREN Y., NORTH S.: Graph drawing by stress majorization. In *Graph Drawing. GD 2004. Lecture Notes in Computer Science*, vol 3383., Pach J., (Ed.). Springer, Berlin, Heidelberg, 2004, pp. 239–250. 2
- [GKN\*09] GLOOR P. A., KRAUSS J., NANN S., FISCHBACH K., SCHODER D.: Web science 2.0: Identifying trends through semantic social network analysis. In *International Conference on Computational Science and Engineering* (2009), pp. 215–222. 2
- [Han13] HANJALIĆ A.: ClonEvol: Visualizing software evolution with code clones. In *First IEEE Working Conference on Software Visualization (VISOFT)* (2013), pp. 1–4. 2, 3
- [HB05] HEER J., BOYD D.: Vizster: visualizing online social networks. In *IEEE Symposium on Information Visualization (INFOVIS)* (2005), pp. 32–39. 2
- [HTCT14] HURTER C., TAYLOR R., CARPENDALE S., TELEA A.: Color tunneling: Interactive exploration and selection in volumetric datasets. In *IEEE Pacific Visualization Symposium* (2014), pp. 225–232. 3
- [JDK10] JUSUFI I., DINGJIE Y., KERREN A.: The Network Lens: Interactive exploration of multivariate networks using visual filtering. In *Information Visualisation (IV), 14th International Conference* (2010), pp. 35–42. 2
- [Jol02] JOLLIFFE I. T.: *Principal Component Analysis*, vol. XXIX. Springer, 2002, ch. Series in Statistics, p. 487. 2
- [JPC\*11] JOIA P., PAULOVICH F. V., COIMBRA D., CUMINATO J., NONATO L.: Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2563–2571. 2
- [JSD\*14] JIA Y., SHELHAMER E., DONAHUE J., KARAYEV S., LONG J., GIRSHICK R., GUADARRAMA S., DARRELL T.: Caffe: Convolutional architecture for fast feature embedding. In *22nd ACM International Conference on Multimedia* (2014), pp. 675–678. 4
- [KPW14] KERREN A., PURCHASE H., WARD M. O.: *Multivariate Network Visualization: Dagstuhl Seminar #13201, Dagstuhl Castle, Germany, May 12-17, 2013, Revised Discussions*, vol. 8380 of *Lecture Notes in Computer Science*. Springer, 2014. 2
- [KS80] KRUSKAL J. B., SEERY J. B.: Designing network diagrams. In *1st General Conference on Social Graphics* (1980), pp. 22–50. 2
- [LM06] LANZA M., MARINESCU R.: *Object-Oriented Metrics in Practice*. Springer, 2006. 4
- [MAH\*12] MARTINS R. M., ANDERY G. F., HEBERLE H., PAULOVICH F. V., DE ANDRADE LOPES A., PEDRINI H., MINGHIM R.: Multidimensional projections for visual analysis of social networks. *Journal of Computer Science and Technology* 27, 4 (2012), 791–810. 2, 3
- [PNML08] PAULOVICH F. V., NONATO L. G., MINGHIM R., LEVKOWITZ H.: Least square projection: a fast high-precision multidimensional projection technique. *IEEE Transactions on Visualization and Computer Graphics* 14, 3 (2008), 564–575. 2, 4
- [PW06] PRETORIUS A. J., WIJK J. J. V.: Visual Analysis of Multivariate State Transition Graphs. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006), 685–692. 2
- [RS00] ROWEIS S. T., SAUL L. K.: Nonlinear dimensionality reduction by locally linear embedding. *Science* 290, 5500 (2000), 2323–2326. 3
- [Sal86] SALTON G.: *Introduction to modern information retrieval*. McGraw, 1986. 3
- [Sam69] SAMMON J. W.: A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers C-18*, 5 (1969), 401–409. 2, 3
- [SGCP09] SMITH M., GIRAUD-CARRIER C., PURSER N.: Implicit Affinity Networks and Social Capital. *Information Technology and Management* 10, 2-3 (2009), 123–134. 2
- [SKK00] STEINBACH M., KARYPIS G., KUMAR V.: *A comparison of document clustering techniques*. Tech. Rep. 00-034, University of Minnesota, 2000. 3
- [ST04] SILVA V., TENENBAUM J.: *Sparse multidimensional scaling using landmark points*. Tech. rep., Stanford University, 2004. 2
- [SVP14] SORZANO C., VARGAS J., PASCUAL-MONTANO A.: A survey of dimensionality reduction techniques, 2014. arxiv:1403.2877. 2
- [SZS\*17] SACHA D., ZHANG L., SEDLMAIR M., LEE J. A., PELTONEN J., WEISKOPF D., NORTH S. C., KEIM D. A.: Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 241–250. 2
- [TdSL00] TENENBAUM J. B., DE SILVA V., LANGFORD J. C.: A global geometric framework for nonlinear dimensionality reduction. *Science* 290, 5500 (2000), 2319–2323. 2, 3
- [vdMPvdH09] VAN DER MAATEN L., POSTMA E., VAN DEN HERIK H.: Dimensionality reduction: A comparative review. *Journal of Machine Learning Research* 10, 1 (2009), 66–71. 2, 3
- [vdZCT16] VAN DER ZWAN M., CODREANU V., TELEA A.: CUBu: Universal real-time bundling for large graphs. *IEEE Transactions on Visualization and Computer Graphics* 22, 12 (2016). 2
- [vLKS\*11] VON LANDESBERGER T., KUIJPER A., SCHRECK T., KOHLHAMMER J., VAN WIJK J., FEKETE J.-D., FELLNER D.: Visual analysis of large graphs: State-of-the-art and future research challenges. *Computer Graphics Forum* 30, 6 (2011), 1719–1749. 2
- [VNCD08] VELARDI P., NAVIGLI R., CUCCHIARELLI A., D’ANTONIO F.: A New Content-Based Model for Social Network Analysis. In *IEEE International Conference on Semantic Computing* (2008), pp. 18–25. 2
- [VT14] VOINEA L., TELEA A. C.: Visual clone analysis with SolidSDD. In *Second IEEE Working Conference on Software Visualization (VISOFT)* (2014), pp. 79–82. 2, 3
- [Wat06] WATTENBERG M.: Visual exploration of multivariate graphs. In *SIGCHI Conference on Human Factors in Computing Systems* (2006), pp. 811–819. 2