Visualizing Concept Associations Using Concept Density Maps

Nees Jan van Eck, Flavius Frasincar, and Jan van den Berg
Faculty of Economics, Erasmus University Rotterdam
P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
{*nvaneck,frasincar,jvandenberg*}*@few.eur.nl*

Abstract

The concept mapping algorithm proposed in an earlier paper is one of the dimensionality reduction techniques that can be used for knowledge domain visualization. Using this algorithm to visualize large knowledge domains may not always provide a good overview of the domain due to visual cluttering of concepts. In this paper, we propose to apply kernel density estimation to the visualization of concept maps in order to be able to better explore large knowledge domains. Kernel density estimation proves to be useful for the identification of concept clusters at different levels of detail. In addition to the visual exploration of large knowledge domains, we are also able to visually verify the hypothesis that the concept mapping algorithm places related concepts close to each other. The flexibility and effectiveness of our approach is validated by applying the proposed technique to different visualization scenarios for the field of computational intelligence.

1 Introduction

Knowledge domain visualization (KDViz) (e.g., [2]) is concerned with the creation of maps that help to present, analyze, and discover important aspects of the information specific to a certain scientific field. Following [2], we divide the process of KDViz into the following six steps: (1) collection of raw data, (2) selection of the type of item to analyze, (3) extraction of relevant information from the raw data, (4) calculation of similarities between items based on the extracted information, (5) positioning of items in a low-dimensional space based on the similarities, and (6) visualization of the low-dimensional space. The first step of the KDViz process is the collection of appropriate data. Since domain maps are typically constructed on the basis of a corpus of scientific texts, one has to collect these texts first. The second step of the KDViz process is the selection of the type of item to analyze. The type of item to analyze depends on the question one wants to answer. The most common types of items are journals, articles, authors, and descriptive words or terms. Each type of item can be used to visualize a different aspect of a scientific field. The third step of the KDViz process is the extraction of relevant information from the raw data collected in the first step. In many cases, the relevant information consists of co-occurrence frequencies of items. The fourth step of the KDViz process is the calculation of similarities between items based on the information extracted in the third step. A possible approach that can be taken to calculate similarities between items based on co-occurrence frequencies is to normalize the co-occurrence frequencies. The fifth step of the KDViz process is the positioning of items in a low-dimensional space based on the similarities calculated in the fourth step. This step is usually performed using dimensionality reduction techniques. The sixth step is the visualization of the low-dimensional space that results from the fifth step. The low-dimensional space has to be visualized in such a way that it can be effectively and accurately explored by human users.

In a previous paper [6], we have focused on the fifth step of the KDViz process, i.e., the positioning of items in a low-dimensional space. In that paper, we have presented an algorithm for constructing concept maps. In the remainder of this paper, we refer to this algorithm as the concept mapping algorithm. A concept map is a domain map that visualizes the associations between concepts in a scientific field. In a concept map, concepts are located in such a way that the distance between two concepts reflects the strength of their association. The stronger the association between two concepts, the smaller the distance between them. A concept map can be used to obtain an overview of a scientific field and, more specifically, of a field's important concepts and their mutual associations.

A deficiency of a concept map is that concept labels have the tendency to overlap when more concepts are displayed. This results in a decrease in insight into the structure of the concept map. Consequently, it may be difficult to get a quick overview of a scientific field.

In this paper, we want to focus more on the visualiza-

tion of the low-dimensional space, i.e., the sixth step of the KDViz process. Our goal is to visualize a concept map that is generated by the concept mapping algorithm in such a way that its structure is clear at a first glance. This is accomplished by visualizing the density of the concepts rather than all individual concepts. We refer to maps that visualize the density of concepts as concept density maps. To calculate the density of concepts, we use kernel density estimation (KDE) (e.g., [8]). In a case study that we describe in this paper, KDE in combination with the concept mapping algorithm is used to visualize the associations between concepts in the field of computational intelligence. It turns out that the resulting concept density maps give, at different levels of detail, a quick overview of this field.

The rest of the paper is structured as follows. Section 2 discusses related work. Section 3 presents the methods that are used in this paper for the positioning and visualization steps of the KDViz process. The application of these methods to the computational intelligence field is described in Section 4. Finally, Section 5 concludes the paper and proposes future research directions.

2 Related Work

In this section, we discuss some of the existing methods that can be used for the positioning of items in a low-dimensional space and for the visualization of the low-dimensional space.

The positioning step of the KDViz process is generally performed using dimensionality reduction techniques. These techniques are able to represent multivariate data in a small number of dimensions. In the case of KDViz, this means that high dimensional item similarities are represented in a two- or three-dimensional space that can be visually interpreted by humans. Some of the dimensionality reduction techniques that can be used for KDViz are multidimensional scaling, principle component analysis, factor analysis, pathfinder network scaling, and self-organizing maps [2]. Due to space limitations, we will only discuss multidimensional scaling in more detail.

Multidimensional scaling (MDS) (e.g., [1]) is the most commonly used positioning method in the literature on KDViz. Given a set of items and the dissimilarities between these items, MDS positions the items in a low dimensional space in such a way that the distances between the items correspond as close as possible to the dissimilarities. The degree of correspondence is measured by a so-called stress function that penalizes the overall disparity between distances and dissimilarities. The optimal positioning is obtained by minimizing the stress function.

The concept mapping algorithm proposed in [6] can be seen as an alternative to MDS. In [6], we made a comparison between this algorithm and MDS by using both methods for constructing a concept map of the computational intelligence field. We compared the positioning generated by the concept mapping algorithm with the positioning generated by MDS. It turned out that the concept mapping algorithm generated a more satisfactory concept map than MDS.

We now consider the visualization step of the KDViz process. The simplest method to visualize the result of the positioning step is the so-called scatter visualization. In the scatter visualization, the spatial positions of items are visualized using points. When a lot of items have to be visualized, the scatter visualization tends to suffer from cluttering. The landscape visualization aims to improve on this. In the landscape visualization, a smooth terrain-like surface is constructed in such a way that the height of the surface indicates the concentration of items in an area. The concentration of items in an area can be calculated using density estimation methods. Most density estimation methods are based on a nearest-neighbor model, a histogram model, a kernel-based model, or a (Gaussian) mixture model [5]. KDViz tools that offer a landscape visualization are, e.g., VxInsight [3] and IN-SPIRE ThemeView (formerly known as ThemeScape [9]).

Methods adopted from the field of graph visualization are sometimes also used for the positioning and visualization steps of the KDViz process. In the field of graph visualization, spring embedding methods (also known as force directed methods) are typically used to position the nodes of a graph into a layout that satisfies similarity requirements as well as presentation requirements (e.g., as few as possible crossing edges). In spring embedding methods, nodes are seen as physical bodies that cause repelling forces on one another and edges between nodes are seen as springs that cause attraction forces between nodes. The final layout of the graph is a solution in which the forces on each node in the graph are in equilibrium.

GraphSplatting [7] is a method for visualizing large graphs as two-dimensional continuous fields. From the layout of the graph, a continuous field is obtained by placing two-dimensional Gaussian shaped basis functions on each node and then summing all basis functions. Although the authors of [7] do not mention it, this is (almost) the same as two-dimensional KDE with Gaussian kernel functions. In [4], GraphSplatting has been successfully applied to the visualization of domain model representations using resource description framework graphs.

3 Methods

In this section, we present the methods that we use for the positioning and visualization steps of the KDViz process. The methods are described in Subsection 3.1 and 3.2, respectively.

3.1 Concept Mapping Algorithm

In this section we briefly describe the concept mapping algorithm that is proposed in [6]. To position concepts at appropriate locations in a concept map, the algorithm needs a concept association matrix as input. Let $c_1, \ldots, c_n$ denote the concepts of interest, where n indicates the number of concepts. The concept association matrix $\mathbf{A}$ is an $n \times n$ matrix that contains for each combination of two concepts the strength of their association. Element a_{ij} of $\mathbf{A}$ is referred to as the association strength between concepts c_i and c_j. In the case study that is described in Section 4, the association strength between two concepts is calculated as the number of texts in which the concepts co-occur.

The underlying idea of the concept mapping algorithm is that each concept should be positioned as close as possible to its ideal location. For a two dimensional concept map, the location of concept c_i is denoted by the vector $\mathbf{x}_i = (x_{i1}, x_{i2})^T$ and the ideal location $\mathbf{x}_i^*$ of concept c_i is defined as

$$\mathbf{x}_i^* = \frac{\sum_{j=1}^{n} a_{ij}\mathbf{x}_j}{\sum_{j=1}^{n} a_{ij}}. \quad (1)$$

The only way to position each concept at its ideal location is to assign all concepts to the same location. This, of course, does not result in a useful concept map. The algorithm therefore attempts not only to position concepts as close as possible to their ideal location but also to prevent concepts from being located too close to each other. To achieve this, the algorithm minimizes the following objective function

$$E = \sum_{i=1}^{n} \left(\bar{w}_i \|\mathbf{x}_i - \mathbf{x}_i^*\|^2 + \beta \sum_{\substack{j=1 \\ j \neq i}}^{n} e^{-\|\mathbf{x}_i - \mathbf{x}_j\|} \right), \quad (2)$$

where $\bar{w}_i$ is a weight indicating the importance of concept c_i, β is a parameter, and $\|\cdot\|$ denotes the Euclidean norm. For the exact calculation of $\bar{w}_i$, we refer to [6]. In (2), the first term within the parentheses is responsible for positioning concepts as close as possible to their ideal location. This term pays more attention to concepts with higher weights. The second term within the parentheses is responsible for preventing concepts from being located too close to each other. In the case study in this paper, a gradient descent algorithm is used to find a (local) minimum of the objective function.

3.2 Kernel Density Estimation

Kernel density estimation (KDE) (e.g., [8]) is a statistical method for constructing a smooth estimate of a probability density function from observed data points. In this paper, KDE is used as a method for displaying the density of concepts in a concept map. The general idea of this approach is that an estimate of the density of concepts is obtained by first placing a symmetric probability density function, called a kernel function, at each concept location and then taking the average of the kernel functions. To speed up calculations, the density of concepts in a concept map is only estimated for a finite grid of points that is specified by the user. The kernel density estimate of a grid point at location $\mathbf{x} = (x_1, x_2)$ is given by

$$\hat{D}(x_1, x_2) = \frac{1}{nh_1h_2} \sum_{i=1}^{n} K\left(\frac{x_1 - x_{i1}}{h_1}, \frac{x_2 - x_{i2}}{h_2} \right), \quad (3)$$

where $K(\cdot)$ is a two-dimensional kernel function centered at each concept location $\mathbf{x}_i = (x_{i1}, x_{i2})$, and $\mathbf{h} = (h_1, h_2)$ is a bandwidth parameter that controls the degree of smoothness.

In the case study that is described in Section 4, the two-dimensional kernel function is taken to be the product of two Laplace density functions, leading to

$$K(t_1, t_2) = \frac{1}{4} e^{-(|t_1| + |t_2|)}. \quad (4)$$

In our experience, the use of other density functions, such as the Gaussian, the Epanechnikov, or the triangular one, leads to worse results with respect to the identification of concept clusters.

Choosing good bandwidths is important. Too small bandwidths do not remove insignificant bumps and result in too rough density estimates, while too large bandwidths smear out real peaks and result in too smooth density estimates. In the case study in this paper, we use the so-called normal scale bandwidth selector [8] to produce estimates of the bandwidths. In the case of the Laplace product kernel, the normal scale bandwidth selector is equal to

$$\hat{h}_j = \left(\frac{\sqrt{\pi}}{6n} \right)^{1/5} \hat{\sigma}_j \qquad j = 1, 2, \quad (5)$$

where $\hat{\sigma}_j$ is the standard deviation of the concept locations in the jth dimension.

4 Case study: Visualizing Concept Associations

To illustrate the usefulness of KDE in combination with the concept mapping algorithm, we used both methods to construct visualizations of the associations between concepts in the field of computational intelligence (CI). The CI field, which can be seen as a part of the larger artificial intelligence field, deals with topics like neural networks, fuzzy systems, and evolutionary computation. Table 1 summarizes the approach taken in our research.

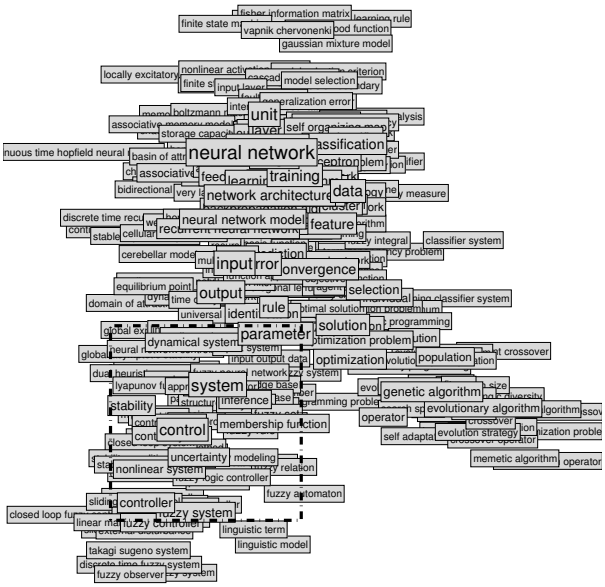

Figure 1. Concept map of the CI field constructed using the concept mapping algorithm.

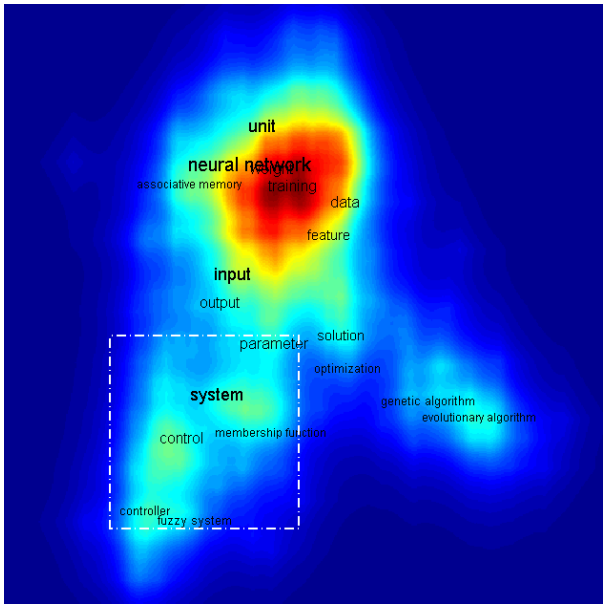

Figure 2. Colored concept density map of the CI field constructed using KDE.

4.1 Data

The visualizations were constructed on the basis of a corpus of scientific texts. The corpus that we used was taken from our previous research [6]. This corpus consists of about 3,800 English-written abstracts that were taken from five leading scientific journals in the CI field using the Science Citation Index Expanded (SCIE). Using a thesaurus of the CI field, 294 different concepts were identified in the corpus of abstracts. The association strengths of the identified concepts were calculated and stored in a concept association matrix. The association strength of two concepts was calculated as the co-occurrence frequency of the concepts in the corpus of abstracts, i.e., the number of abstracts in the corpus in which the concepts co-occur.

Table 1. Summary of the way in which the KDViz process is implemented in this paper.

Step of the KDViz Process	Implementation
(1) Collection of data	Corpus of CI abstracts
(2) Selection of type of item	Concepts
(3) Extraction of information	Co-occurrences frequencies
(4) Calculation of similarities	Association strengths
(5) Positioning of items	Concept mapping algorithm (Subsection 3.1)
(6) Visualization	Kernel density estimation (Subsection 3.2)

4.2 Map Overview

The concept mapping algorithm was used to map the concept association matrix to a two-dimensional concept map. A more detailed description of this mapping, together with parameter settings, can be found in [6]. Figure 1 shows the resulting concept map of the CI field. The goal of the concept map is to obtain an overview of the CI field, but due to the overlap of concept labels it is hard to get this overview at a first glance.

We applied KDE to the concept map in Figure 1 to gain more insight into the structure of the CI field. We used the Laplace kernel and the normal scale bandwidth selector. Figure 2 shows the resulting colored concept density map constructed using a grid size of 500×500. The used color scheme ranges from the colors blue to red, and passes through the colors green, yellow, and orange. Blue denotes low densities, while red denotes high densities.

By looking at the colored concept density map, we can easily detect some clusters (i.e., areas of high densities), which indicate the presence of a large number of highly associated concepts. A large and very dense cluster of concepts can be identified in the top center of the map (colored by red). In addition, three smaller and less dense clusters can be found in the bottom left and the bottom right of the map (colored by green). The concept labels that are shown on top of the map should give an indication of the topic of

each cluster, and consequently may point to the main research topics in the CI field. The concepts in the top center cluster of the map (indicated by the labels *neural network*, *unit*, *weight*, and *training*) are related to the topic of neural networks. The concepts in the two bottom left clusters (indicated by the labels *system*, *control*, *controller*, *fuzzy system*, and *membership function*) are related to the topic of fuzzy systems. The concepts in the bottom right cluster (indicated by the labels *evolutionary algorithm*, and *genetic algorithm*) are related to the topic of evolutionary computation.

To validate the hypothesis that related concepts are placed in the same cluster, we constructed a contoured concept density map in which the locations of the concepts are also displayed. This map is shown in Figure 3. For each of the concepts, we manually determined whether the concept is relevant to the topic of neural networks, to the topic of fuzzy systems, to the topic of evolutionary computation, or to more than one of these topics. A green dot (●) refers to a neural network concept, a red cross (×) refers to a fuzzy systems concept, a blue plus sign (+) refers to an evolutionary computation concept, and a grey star (✳) refers to a general concept. The colored contour lines indicate points that have the same density. Since the contour lines roughly indicate the boundaries of the clusters, we can see that the top center cluster contains mainly neural networks concepts, the bottom left clusters contain mainly fuzzy systems concepts, and the bottom right cluster contains mainly evolutionary computation concepts. From this observation we can conclude that the hypothesis that related concepts are placed in the same cluster has been proven valid.

4.3 Zooming into the Map

Up to now, we have only looked at a concept density map (Figure 2) that is constructed on the basis of the complete concept map of the CI field (Figure 1). As we have seen, this concept density map gives a quick overview of the global structure of the CI field. To gain more insight into local details of the CI field, we zoomed into a region that seems interesting. The region of interest is indicated by a dashed bounding box in Figure 1 and 2 and corresponds to the area in which most of the concepts related to fuzzy systems are located. Figure 4 shows the concept map of this region of interest. Subsequently, we applied KDE to this concept map. Again, we used the Laplace kernel and the normal scale bandwidth selector. Figure 5 shows the resulting colored concept density map constructed using a grid size of 500×500. It should be clear from Figure 5 that we take a look at a more detailed level and that we again obtain a quick overview of the knowledge structure, this time that of the selected region. The most important concepts in the region are now visible while most of them are not visible in Figure 2. In addition, the concept density map shows many more details like the distinct, red colored clustering in three regions, which is not that well visible in Figure 2.

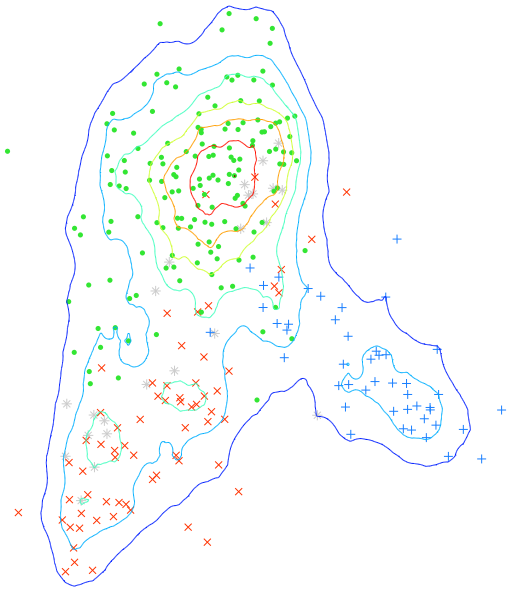

Figure 3. Contoured concept density map of the CI field including the concept locations.

The concept density map in Figure 5 was inspected more carefully, both for validation purposes and for knowledge discovery. We showed the map to two experts in the field of fuzzy logic and fuzzy systems. They both agreed that (fuzzy) *inference* and *rule base* are indeed semantically close concepts, relatively different from concepts like *control system* and *controller*. They also observed that the concepts that lie just between the three clusters reveal the most interesting information. The fact that the concept *parameter* is situated in the upper right corner could easily be explained since this concept is also of importance for neural networks (see also Figure 2). The fact that the concept *Lyapunov function* is situated in the upper left came for both experts as a surprise. One of the experts knew that this concept is used in the field of *neural networks*, especially related to the concept *associative memory* (see Figure 2). Since the concept *Lyapunov function* lies between *associative memory* and (fuzzy) *control system*, he started to think that this concept is also of importance within fuzzy control, which is indeed the case. For the other expert, who is familiar with fuzzy control, it was a new fact that *Lyapunov functions* are used in the field of neural networks. So, careful inspection of the concept density maps by two domain experts revealed that these maps enable a process of knowledge discovery.

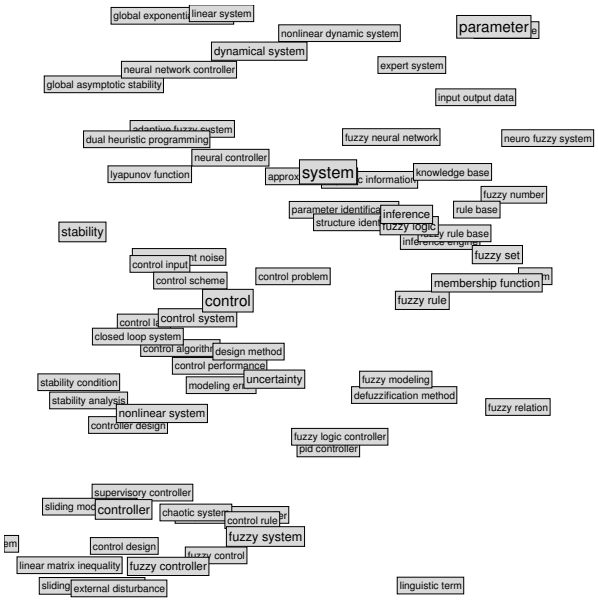

Figure 4. Concept map of a region of interest.

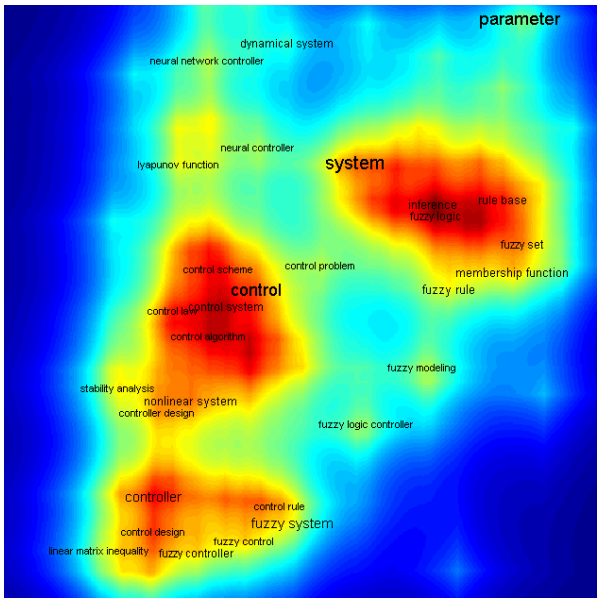

Figure 5. Colored concept density map of a region of interest.

5 Conclusions

To visualize knowledge domains consisting of a large number of concepts, one can use the concept mapping algorithm in combination with kernel density estimation. In this paper, we employed as similarity measure the number of concept co-occurrences in a collection of texts from the computational intelligence field. After experimenting with several kernels, we found that the Laplace kernel gives the best results with respect to the identification of concept clusters. Using our visualization approach, we were able to identify clusters at different levels of detail by zooming into regions of interest. We were also able to test our assumption that related concepts are placed in the same cluster.

At the moment, we can extract concept associations from a large set of scientific texts and visualize these associations as spatial relationships. In the future, we would like to extend our method by (1) extracting the semantic relationships between concepts in a knowledge domain and (2) providing a suitable visualization metaphor to graphically depict these semantic relationships. Also, we would like to complement the current visualization techniques with querying facilities that provide an enhanced exploration of knowledge domains. In our endeavor, we are encouraged by the recent developments for the Semantic Web that we plan to investigate for the discovery, representation, and visualization of semantic concept maps.

References

[1] I. Borg and P. J. Groenen. *Modern Multidimensional Scaling*. Springer, New York, 2nd edition, 2005.

[2] K. Börner, C. Chen, and K. W. Boyack. Visualizing knowledge domains. *Annual Review of Information Science and Technology*, 37:179–255, 2003.

[3] G. S. Davidson, B. Hendrickson, D. K. Johnson, C. E. Meyers, and B. N. Wylie. Knowledge mining with VxInsight: Discovery through interaction. *Journal of Intelligent Information Systems*, 11(3):259–285, 1998.

[4] F. Frasincar, A. Telea, and G. J. Houben. *Visualizing the Semantic Web*, chapter 9: Adapting Graph Visualization Techniques for the Visualization of RDF Data, pages 154–171. Springer, 2nd edition, 2006.

[5] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*, volume 26 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, London, 1986.

[6] N. J. van Eck, L. Waltman, and J. van den Berg. A novel algorithm for visualizing concept associations. In *Proceedings of the 16th International Workshop on Database and Expert Systems Applications*, pages 405–409, 2005.

[7] R. van Liere and W. de Leeuw. Graphsplatting: Visualizing graphs as continuous fields. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):206–212, 2003.

[8] M. P. Wand and M. C. Jones. *Kernel Smooting*, volume 60 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, London, 1995.

[9] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: Spatial analysis and interaction with information from text documents. In *Proceedings of the 1995 IEEE Symposium on Information Visualization*, pages 51–58, 1995.