The Metro Visualisation of Component Planes for Self-Organising Maps

Robert Neumayer, *Student Member, IEEE*, Rudolf Mayer, *Student Member, IEEE*,
Georg Pölzlbauer, and Andreas Rauber, *Member, IEEE*

***Abstract*— The *Self-Organising Map* is a popular unsupervised neural network model which has successfully been used for clustering various kinds of data. To help in understanding the influence of single variables or components on clusterings, we introduce a novel method for the visualisation of *Component Planes* for SOMs. The approach presented is based on the discretisation of the components and makes use of the well-known metro map metaphor. It depicts consistent values and their ordering across the map for discretisations of various components and their correlations in terms of directions on the map. In our approach *Component Lines* are drawn for each component of the data, allowing the combination of numerous Component Planes into one plot. We also propose a method to further aggregate these Component Lines, by grouping highly correlated variables, i.e. similar lines on the map. To show the applicability of our approach we provide experimental results for two popular machine learning data sets.**

I. Introduction

The Self-Organising Map (SOM) is a prominent data mining method for clustering and data projection. Part of its popularity can be attributed to the various visualisation methods which summarise the characteristics of the underlying data set. One well-known method to get a better understanding of the characteristics of certain areas on the map and the rationale for mapping certain data points onto specific regions is the visualisation of *Component Planes*, i.e. the colour-coding of single components or variables. This visualisation thus partitions the SOM into projections of single variables, which, however, are hard to make sense of in case of high-dimensional data sets. Part of this complexity is usually overcome by clustering Component Planes to obtain groups of common characteristics. In this paper, we go one step further by proposing a novel method for the presentation of the mutual relationships of the various components. We propose an intuitive metaphor of maps of metro lines, which aims at showing a simplified representation of the components in a single illustration. Each variable is represented by differently coloured and connected line segments, called *Component Lines*, which are designed to connect the areas of the Component Planes from the lowest to the highest component value with several steps in between. Metro maps introduced the concept of skewed distances, as opposed to geographically correct distances, which is also used in our visualisation. This concept was originally developed for the maps of the London Underground transport network.

In a further aggregation step, we propose a technique that groups these Component Lines in case they are highly correlated, further simplifying the result and reducing the amount of redundant information displayed. Many of the steps involve a trade-off between the detail and amount of information and the clarity of the representation. We sometimes deliberately choose to sacrifice accuracy in order to communicate the data in an intuitive manner and to summarise only the most dominant characteristics of a data set. The resulting visualisation allows to intuitively communicate relationships between multiple variables and tendencies on a SOM in a single visualisation, capturing both positive and negative correlations. It can be overlaid onto any colour-coded visualisation of the SOM, abstracting from spurious details and focusing on the dominant attribute value distributions on the SOM.

The remainder of this paper is organised as follows: Section II describes related work, with a short introduction to the SOM algorithm, a survey of the most important and relevant SOM visualisations, and the origins of metro map based visualisations. Section III introduces our method for computing the metro visualisation. In Section III-A, we describe how the most basic line segments can be calculated from a SOM. Section III-B introduces an important measure of similarity, which is used in later aggregation steps to combine lines of correlating Component Planes. Section III-C describes how the final representation is calculated and several visual improvements that make the output more comprehensible. In Section IV, we apply our approach to the Iris and Boston Housing data sets. Section V concludes our findings and provides an outlook on future work.

II. Related Work

In this section, we introduce the Self-Organising Map and related concepts and visualisation techniques based thereon.

A. Self-Organising Map

The Self-Organising Map [4] is a well-known and widely used neural network model based on unsupervised learning. It provides a mapping from a high-dimensional input space to a lower-dimensional, often two-dimensional, output space. In the process of this mapping input patterns that are located close to each other in the input space will also be located closely in the output space, while dissimilar patterns will be mapped on opposite regions of the map. The SOM therefore provides

Robert Neumayer, Rudolf Mayer, Georg Pölzlbauer, and Andreas Rauber are with the Department of Software Technology and Interactive Systems, Vienna University of Technology, Favoritenstr. 9-11, Vienna, A-1040, Austria (phone: +43-1-58801-18876; fax: +43-1-58801-18876; email: {neumayer,mayer,poelzlbauer,rauber}@ifs.tuwien.ac.at).

(a) Component Plane (b) Connected region centres

Fig. 1. Component Plane and its discretisation (map size is 12 × 18 units, discretisation is done for three regions)

a sort of clustering of the data, however, without explicitly assigning data items to clusters.

Basically, a Self-Organising Map is a low-dimensional lattice, in this work assumed to be two-dimensional, consisting of M neurons or units. The map lattice can have different topologies, in this paper we use rectangular maps. We further assume the feature or input space to be the vector space over the real numbers ($\mathbb{R}^N$). For each unit in the output space, a codebook vector $\mathbf{m}_i$ of the dimensionality of the input space is linked to a position on the two-dimensional map lattice, denoted as $\xi_i = (\xi_i^x, \xi_i^y)$. The codebook $\mathfrak{M}$ is the set of all codebook vectors. In the training phase, the best matching codebook vector is identified for all input vectors by using a distance function, the Euclidean distance in our case. Once the best matching unit is identified, its codebook vector and the codebook vectors of neighbouring units are shifted towards the input vector. This results in a topology-preserving mapping of input vectors onto units of the map. Self-Organising Maps have been applied to a wide range of tasks, ranging from control interfaces for industrial processing plants and other engineering problems [5] to document organisation in digital libraries [9].

B. Self-Organising Map Visualisations

SOM visualisations can utilise the map lattice as a visualisation platform [13], where quantitative information is most commonly depicted as colour values or as markers of different sizes. More advanced approaches exploit e.g. the analogy to geography [10].

Component Planes are projections of single dimensions of the codebook. By plotting the Component Planes for all dimensions, all information about the codebook vectors is revealed. However, as with other methods in statistics, with increasing dimensionality it becomes more difficult to perceive important information such as clustering structure and underlying dependencies.

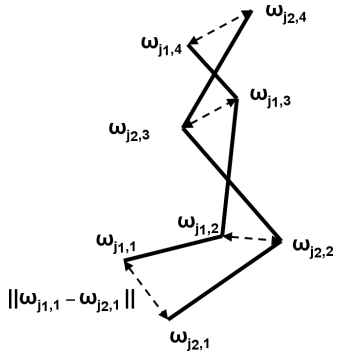
The unified distance matrix (U-Matrix [12]) is a visualisation technique that shows the local cluster boundaries by depicting pair-wise distances of neighbouring prototype vectors. It is the most common method associated with Self-Organising Maps and has been extended in numerous ways. The Gradient Field [7] has some similarities with the U-Matrix, but applies smoothing over a broader neighbourhood and uses a different style of representation. It plots a vector field on top of the lattice where each arrow points to its closest cluster centre. This is used to contrast different groups of Component Planes [8] with a similar goal as our method introduced in this paper. Similarly, [14] applies clustering of and projection techniques on the Component Planes with the aim of visually ordering them such that similar ones are grouped together making them more easily identifiable by users.

The second category of visualisation techniques take into account the distribution of the data. The most simple ones are hit histograms, which show how many data samples are mapped to a map unit, and labelling techniques, which plot the names and categories, provided they are available, of data samples onto the map lattice. More sophisticated methods include smoothed data histograms [6], which show the clustering structure by mapping each data sample to a number of map units, or graph-based methods, showing connections for units that are close to each other in the feature space. The P-Matrix [11] is another density-based approach that depicts the number of samples that lie within a sphere of a certain radius around the codebook vectors. The radius is a quartile of the pair-wise distances of the data vectors.

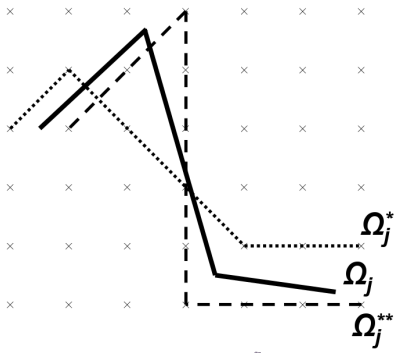
C. The London Underground Map

The principle of metro map visualisations as used in this approach was first introduced and designed by Harry Beck in 1931 [2], and is with only slight modifications still used for today's London metro maps. At the time of its introduction, the concept of the map was revolutionary. Contrary to previous metro maps, the one designed by Beck disregarded geographic aspects. The geometric representation of the river Thames is the only link between the map and the actual landform of the area it represented. Furthermore, distances in the map did not correlate to geographic distances anymore.

These days, this kind of schematic representation of transportation networks has become familiar to a great number of people. Therefore, this representation becomes a highly attractive metaphor for map visualisations. Furthermore, it is common knowledge that the distances on metro maps are skewed and do not conform with real-world distances, which is also true for the metro map visualisations of Component Lines



(a) Measure of distances between lines



(b) Snapping of region centres to units of the SOM

Fig. 2. Computation of distances between metro lines and snapping of region centres

we describe in this paper. Its prime attraction, however, lies in its simplicity, abstracting from spurious details and resulting in a more abstract representation that is more easily memorised and compared across different variations.

III. Component Line Visualisation

A. From Component Planes to the Metro Visualisation

Our method starts with the vector representation of the variables of the codebook. Its $j-th$ component is denoted as $\mathbf{c_j} \in R^M$.

The classic Component Plane visualisation is shown in Figure 1(a). In order to achieve discretisation of the Component Planes, each component is split into a number n of disjoint ranges. In our experiments, this division is performed by calculating the threshold values as equidistant points between the lowest and highest values in the particular Component Planes. This results in n partitions of the SOM. The upper limit l for region k can be computed for every component $\mathbf{c_j}$ as follows:

$$l_k(\mathbf{c_j}) = \frac{k \cdot (\max \mathbf{c_j} - \min \mathbf{c_j})}{n} + \min \mathbf{c_j} \tag{1}$$

where $\max \mathbf{c_j}$ and $\min \mathbf{c_j}$ denote the maximum and minimum values for a particular component, respectively. Alternatively, this division into regions could also be done in other ways such as by using percentiles as delimiters.

The resulting limits $[l_{k-1}, l_k]$ for all n possible values of k are therefore set so that the components are represented as the number of instances in each range. The set of units that fall within these intervals are denoted as:

$$\Theta_{j,k} = \{\xi_i \mid m_{i,j} \in [l_{k-1}, l_k]\} \tag{2}$$

where j is an index over the dimensions/components, k an index over the number of regions, i over the number of codebook vectors. Therefore, $m_{i,j}$ denotes the $j-th$ component of codebook vector $\mathbf{m_i}$.

Region centres $\omega_{j,k}$ for component j and region k are computed as the centres of gravity for individual groups of components as follows:

$$\omega_{j,k} = \frac{1}{|\Theta_{j,k}|} \sum_{\xi_i \in \Theta_{j,k}} \xi_i \tag{3}$$

Further, Ω_j denotes the entire tuple of centres $\{\omega_{j,k} \mid 1 \leq k \leq n\}$ and implicitly represents the $n-1$ lines, which are obtained by linking all centres of regions of a specific component ordered by their value, henceforth referred to as Component Lines.

B. Measuring Distance between Component Lines

Figure 1(b) depicts the centres marked by black dots. Adjacent centres are linked by lines. Note that the coordinates obtained for ω do not necessarily coincide with the integer unit coordinates, as they can take continuous values. In order to perform the subsequent computational steps, we need to introduce a metric that measures the distance between two component lines Ω_{j_1} and Ω_{j_2}. This function d introduces a concept of dissimilarity, such that pairs of lines that are mutually more similar than others can be identified. We define this measure as

$$d(\Omega_{j_1}, \Omega_{j_2}) = \min\left(\sum_{k=1}^{n} \|\omega_{j_1,k} - \omega_{j_2,k}\|, \sum_{k=1}^{n} \|\omega_{j_1,k} - \omega_{j_2,(n+1-k)}\|\right) \tag{4}$$

where $\|\cdot\|$ denotes the Euclidean norm. The idea behind this is that the lines are a simplified representation of the gradient of a single variable, which should be visually similar in case the variables are correlated, be it negatively or positively, as denoted by the two parts of Equation 4. Thus, Component Lines which share approximately the same path are assigned a low distance. Figure 2(a) illustrates the computation of distances between Component Lines as the sum of the distances between the pairs of centre points of the same indices. Inverting the indices of Ω_{j2} as in the second argument of *min* in Equation 4 stems from the fact that Component Planes can be negatively correlated, i.e. the lines point in opposite directions. For similarity, however, only the absolute value of the correlation is of interest.

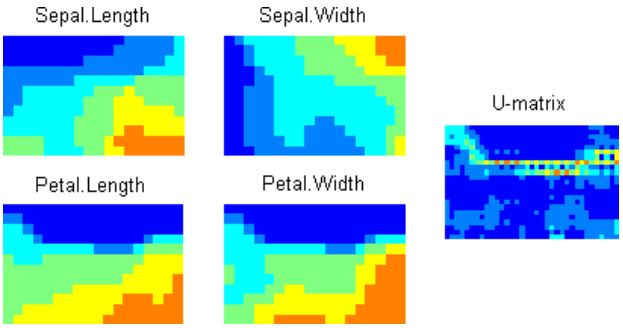


Fig. 3. Component Planes (six regions) and U-Matrix in the Iris data set

C. Visual Enhancements

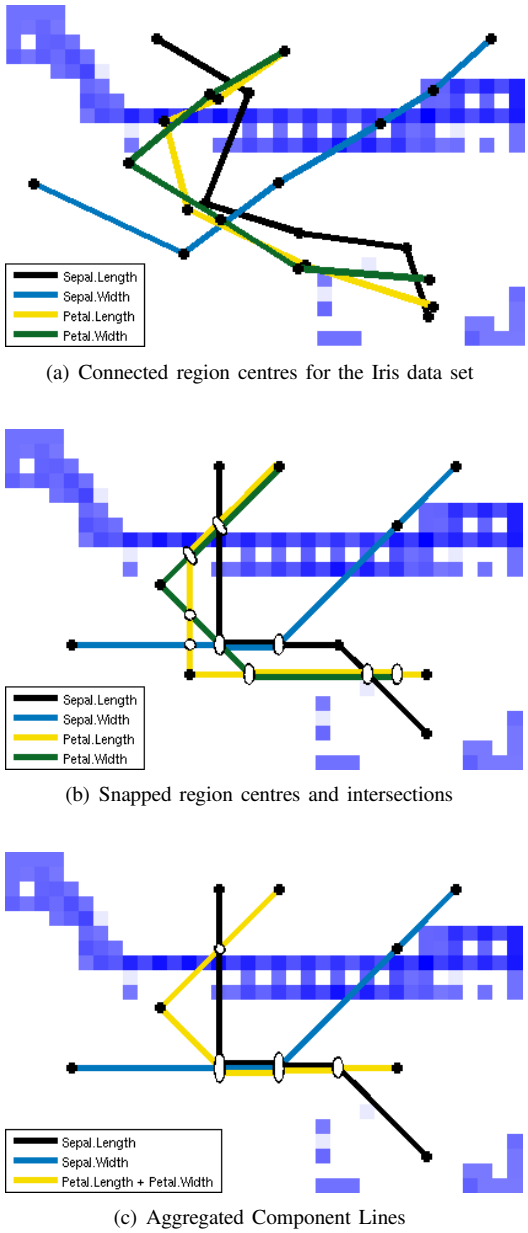
1) Snapping: For a more intuitive and smooth representation, and to more closely resemble the metaphor of an actual metro map, the locations of the region centres are adjusted in a way that the lines Ω_j are drawn only horizontally, vertically, or diagonally. In order to achieve this kind of representation, we compute new Component Lines Ω_j^* where the centres $\omega_{j,k}$ are restricted to the discrete unit positions on the map. Formally, this is performed by minimising the energy function

$$\min_{\Omega_j^*} d(\Omega_j, \Omega_j^*) \mid \Omega_j^* \in \mathfrak{S} \tag{5}$$

where $\mathfrak{S}$ is the set of valid candidate Component Lines as defined above, i.e. with integer positions for the map coordinates and only adjacent lines with multiples of 45 degrees. We solved this by a heuristic optimisation algorithm that compares several candidates. First, a number of possible 'snapping points', i.e. the coordinates of the closest unit on the map, is computed for each region centre. Then all remaining centres are aligned to unit coordinates too, with the constraint of only considering lines in an angle of multiples of 45 degrees to each other. The the best solution in terms of an overall lowest distance to the actual region centres out of the resulting set of candidate alignments is chosen. Figure 2(b) illustrates the process of aligning the original Ω_j to candidate lines Ω_j^* and Ω_j^{**}. The small crosses represent the discrete positions of the map units. In this example, the candidate with the smallest distance to the initial Component Line would be Ω_j^{**}.

2) Metro Stations and Intersections: The centres ω, which represent the centres of gravity of single components, are indicated by markers on the Component Lines, intuitively mimicking metro stops. To even more emphasise the metaphor of real-life metro maps, certain intersections of Component Lines are displayed as metro stations (white circles with black borders). These stations clearly point out the meaning of parallel lines, namely their homogeneity with respect to a certain local similarity. They might furthermore provide a useful reference point for comparing different SOMs trained on the same data.

3) U-Matrix: As described in Section II-B, the U-Matrix can be utilised to visualise cluster boundaries. We use this technique in our map to show distinct boundaries between



(a) Connected region centres for the Iris data set

(b) Snapped region centres and intersections

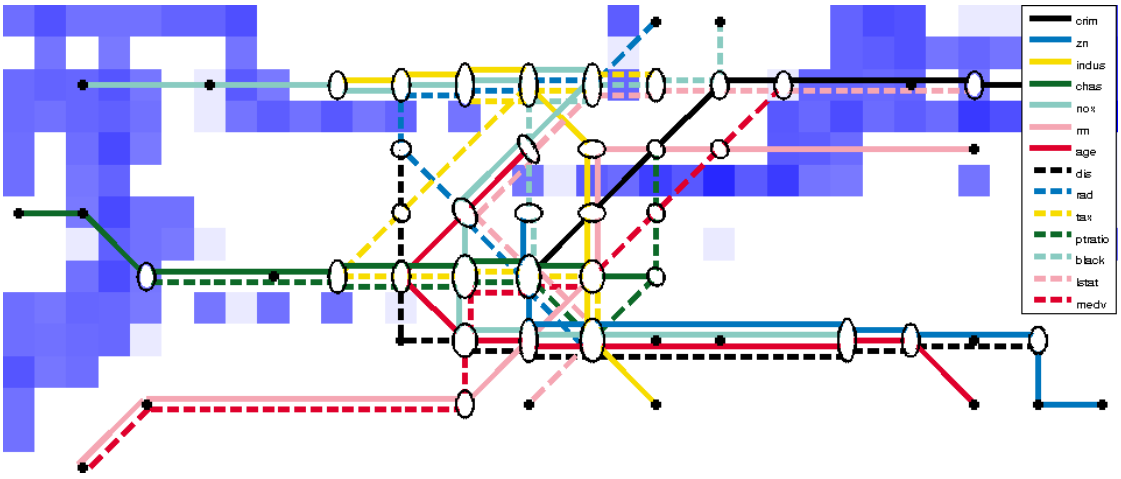
(c) Aggregated Component Lines

Fig. 4. Metro visualisation for the **Iris data set** (map size 12×18, four regions)

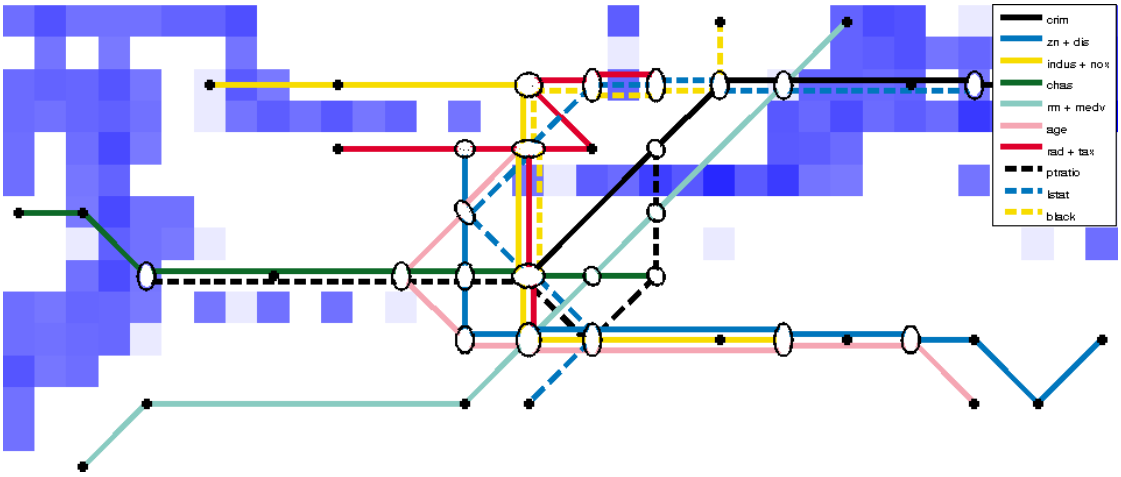
clusters as representational *rivers* or *lakes*. This again resembles the metaphor of rivers or lakes, which are often featured in the background of real-world metro maps. Analogously to a city being divided into different areas, our visualisation divides data into clusters. For this visualisation we only visualise very high values of the U-Matrix on a two-coloured palette.

D. Aggregation of Component Planes

With an increasing number of dimensions in the input space, and therefore an increasing number of Component Plane visualisations the perception of this visualisation becomes increasingly difficult. Component Lines can combine the Component Plane information in one plot, but even this only makes



(a) Snapped region centres forming the Component Lines



(b) Aggregated Component Lines

Fig. 5. Metro map visualisation for the **Boston Housing data set** (map size is 8×18, six regions)

sense up to a certain dimension. In order to further summarise the information communicated through our illustration, we propose an optional step of aggregating similar Component Lines into representative prototypes. This works by clustering the Component Lines. With the distance measure between two such lines defined in Equation 4, a matrix of pair-wise distances can be calculated. Subsequently, hierarchical clustering with any of the common linkage metrics can be performed. In our approach, we use Ward's clustering, and the (fewer) aggregated Component Lines are computed by averaging over the Component Line centres within each cluster. A threshold value for the Ward's clustering can be used to influence the level of aggregation and to suit user's subjective information needs or desired levels of aggregation. Examples are given in the next section.

IV. Experiments

In this section we apply the preceding methods to standard machine learning benchmark data sets to demonstrate the characteristics, benefits, and the applicability of the metro visualisation. We chose the Iris and Boston Housing data sets from the UCI machine learning repository as well-known examples.

A. *Iris Data Set*

The Iris data set [1] is a well-known standard reference data set, containing three classes of 50 instances each, where each class refers to a type of Iris plant. One class is linearly separable from the remaining two (see the upper region of the Component Planes visualisation in Figure 3), while the latter are not linearly separable from each other. The data consists of four features: sepal length, sepal width, petal length, and petal width.

For this experiment, we trained a map of size 12×18 units and discretisation into eight regions. Figure 3 depicts the U-Matrix of the map, which visualises the above mentioned linear separation of one of the classes from the other two. It

moreover shows visualisations of the four Component Planes of the data set, each grouped into four regions.

Figure 4(a) depicts the metro map after the first step of the division of Component Planes into regions and connecting the centres of the regions from the same Component Plane with lines. The black circles on the Component Lines denote the centres of the regions. The next step is shown in Figure 4(b), where the centres are snapped onto unit locations to simplify the visualisation and to more closely resemble a metro map. Besides that, the intersections between two or more lines are added.

Finally Figure 4(c) shows the aggregation of the four Component Lines grouped into three remaining ones. The components *petal length* and *petal width* have been aggregated to one Component Line, indicating that the components are highly correlated. Also, *sepal length* is somewhat correlated to the *petal length* and *petal width*, while *sepal width* has clearly no correlation to the other components. These correlations can be conveniently displayed and overlaid in a single visualisation on top of other colour code clusterings, such as e.g. the U-Matrix.

B. Boston Housing

To evaluate our method, and to demonstrate the effect of aggregation techniques, we also performed experiments on a data set having a higher input dimensionality. The Boston Housing data set [3] consists of 506 instances containing information collected by the U.S Census Service concerning housing in the area of Boston, Massachusetts. The data is described in thirteen continuous and one binary attributes.

The trained map consists of 8×18 units, discretisation is done for six regions. The Component Lines, already snapped onto units, are shown in Figure 5(a). With the higher number of dimensions in this data set, the visualisation becomes more crowded. However, it still gives some hints about correlated Component Planes. For example, the components *'Median value of owner-occupied homes'* ('*medv*' in the map legend) and *'average number of rooms per dwelling'* ('*rm*') both run from the lower left to the upper right corner of the map. Consequently, these two components are grouped together to one Component Line in Figure 5(b). As another example, the components *'proportion of residential land zoned for lots over 25,000 sq.ft.'* ('*zn*') and *'weighted distances to five Boston employment centres'* ('*dis*'), both running from the lower right corner to the centre of the map, are grouped into one Component Line.

V. Conclusions and Future Work

In this paper, we presented a novel and intuitive method for the visualisation of Component Planes and their correlations, by using Component Lines as a metaphor of metro maps. Driven by the problems arising when trying to plot all components, namely the resulting high number of plots, this is achieved by the discretisation of single components of the input data. This subsequently allows to highlight these components by a user-chosen level of aggregation. Parameter settings include the number of regions over a component's range. For higher-dimensional input vectors, grouping the Component Lines can reduce the complexity of the visualisation. The experiments presented show that our method is feasible for visualising both low and higher-dimensional feature sets. We showed that the proposed aggregation technique for components is capable of covering the variance within Component Lines.

As future work, we want to investigate using different distance functions between Component Lines, focusing on local distance functions and edit distances, which are used for snapping and aggregation. We therein plan to put the emphasis of future research on the conjunction with the aggregation. Further, we will work on improving the coherence of the visualisation (e.g. metro lines not being intertwined after intersections). We also want to investigate the robustness and usefulness of our method on very high-dimensional data sets, for example from the text mining or music information retrieval domains, particularly concentrating on aggregation issues.

References

[1] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. In *Annual Eugenics, 7, Part II*, pages 179–188, 1936.

[2] Janin Hadlaw. The london underground map: Imagining modern time and space. *Design Issues*, 19:25–36, Winter 2003.

[3] David Harrison Jr. and Daniel L. Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102, 1978.

[4] Teuvo Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, 1995.

[5] Teuvo Kohonen, Erkki Oja, Olli Simula, Ari Visa, and Jari Kangas. Engineering applications of the self-organizing map. *Proceedings of the IEEE*, 84(10):1358–1384, October 1996.

[6] Elias Pampalk, Andreas Rauber, and Dieter Merkl. Using Smoothed Data Histograms for Cluster Visualization in Self-Organizing Maps. In *Proceedings of the International Conference on Artifical Neural Networks (ICANN'02)*, pages 871–876, Madrid, Spain, August 27-30 2002. Springer.

[7] Georg Pölzlbauer, Michael Dittenbach, and Andreas Rauber. A visualization technique for self-organizing maps with vector fields to obtain the cluster structure at desired levels of detail. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'05)*, pages 1558–1563, Montreal, Canada, July 31 - August 5 2005. IEEE Computer Society.

[8] Georg Pölzlbauer, Michael Dittenbach, and Andreas Rauber. Advanced visualization of self-organizing maps with vector fields. *Neural Networks*, 19(6-7):911–922, July-August 2006.

[9] Andreas Rauber and Dieter Merkl. The SOMLib digital library system. In *Proceedings of the 3rd European Conference on Research and Advanced Technology for Digital Libraries (ECDL'99)*, Lecture Notes in Computer Science (LNCS 1696), pages 323–342, Paris, France, September 22-24 1999. Springer.

[10] André Skupin. A picture from a thousand words. *Computing in Science and Engineering*, 6(5):84–88, 2004.

[11] Alfred Ultsch. Maps for the visualization of high-dimensional data spaces. In *Proceedings of the Workshop on Self organizing Maps (WSOM'03)*, pages 225–230, Kyushu, Japan, 2003.

[12] Alfred Ultsch and Hans Peter Siemon. Kohonen's self-organizing feature maps for exploratory data analysis. In *Proceedings of the International Neural Network Conference (INNC'90)*, pages 305–308. Kluwer, 1990.

[13] Juha Vesanto. SOM-based data visualization methods. *Intelligent Data Analysis*, 3(2):111–126, 1999.

[14] Juha Vesanto and Jussi Ahola. Hunting for correlations in data using the self-organizing map. In *International ICSC Congress on Computational Intelligence Methods and Applications (CIMA'99)*, pages 279–285. ICSC Academic Press, 1999.