Advanced visualization of Self-Organizing Maps with Vector Fields

Georg Pölzlbauer [a] Michael Dittenbach [b] Andreas Rauber [a,b]

[a]*Department of Software Technology, Vienna University of Technology, Favoritenstrasse 9–11, Vienna, Austria*

[b]*eCommerce Competence Center – ec3, Donau-City-Str. 1, Vienna, Austria*

Abstract

Self-Organizing Maps have been applied in various industrial applications and have proven to be a valuable data mining tool. In order to fully benefit from their potential, advanced visualization techniques assist the user in analyzing and interpreting the maps. We propose two new methods for depicting the SOM based on vector fields, namely the Gradient Field and Borderline visualization techniques, to show the clustering structure at various levels of detail. We explain how this method can be used on aggregated parts of the SOM that show which factors contribute to the clustering structure, and show how to use it for finding correlations and dependencies in the underlying data. We provide examples on several artificial and real-world data sets to point out the strengths of our technique, specifically as a means to combine different types of visualizations offering effective multidimensional information visualization of SOMs.

Key words: Self-Organizing Maps, Visualization, Gradient Field, Borderline

---

1 Introduction

The Self-Organizing Map (SOM; Kohonen (2001)) is an unsupervised artificial neural network especially valuable in exploratory data analysis. It is used in a variety of scientific and industrial applications (Kaski et al., 1998; Oja et al., 2001). In this paper, we use the SOM primarily as a visualization tool that performs a non-linear projection from a high-dimensional feature space onto a two-dimensional map lattice. We present our work on several flavors of a visualization method that shows the clustering structure of the map as a gradient field and show various extensions and possible application scenarios. The gradient field is displayed as arrows that point to the most likely cluster center for each map unit. One property of this technique is that it can be adjusted by interactively setting a kernel smoothing parameter to show the granularity of the clustering, similar to the choice of the number of clusters in clustering algorithms, revealing the structure at various levels of detail.

The rest of the paper is organized as follows: In Section 2 we describe related SOM visualization techniques. In Section 3 we discuss basic properties of the SOM and neighborhood kernel functions. In Section 4 we introduce the Gradient Field method and several extensions to it, namely the Borderlines representation as well as grouped gradient fields. In Section 5 we provide experiments on a benchmark data mining data set, a real-world industrial data set, and some experiments on artificial data. Section 6 summarizes this paper and gives directions for future work.

2 Related work

In this section, we provide a brief survey on SOM-related concepts and visualization techniques. The two primary qualities of the SOM are vector quantization and non-linear vector projection. The former aims at finding a reduced representation of the data samples by significant prototypes, while the latter applies a reduction of dimensionality and presentation in human-perceivable form (Grinstein et al., 2001). The most important vector quantization method is k-means (Hartigan and Wong, 1979) which is very similar to the SOM. Prominent non-linear vector projection methods include pair-wise distance preserving techniques like multi-dimensional scaling (Torgerson, 1952) and Sammon's Mapping (Sammon, 1969), which aim at preserving pair-wise distances in the resulting projections. Other techniques measure distance based on distribution and density of the data set to be projected, like Isomap and Curvilinear Distance Analysis (Lee et al., 2004). Isomap (Tenenbaum et al., 2000) relies on nearest neighbor graphs of the input data and finding the shortest path to construct the latent space. ViSOM (Yin, 2002) is an extension of the SOM that aims at positioning the prototype vectors in a roughly equidistant way in order to avoid post-processing visualization methods. With this model the training process itself is adapted to capture data space characteristics. A lot of effort has been directed to extend the topology of the SOM to more flexible topologies to reveal the cluster structure as part of the resulting architecture (Martinetz et al., 1993; Fritzke, 1994; Dittenbach et al., 2002).

SOM visualization usually uses the map lattice as a visualization platform (Vesanto, 1999), where quantitative information is most commonly depicted as color values or as markers of different sizes. More advanced approaches

exploit the analogy to geography (Skupin, 2004). SOM visualization methods mainly fall in two categories: techniques that rely solely on the codebook vectors, and ones that take the distribution of the data samples into account. We begin with the former category. Component planes are projections of one single dimension of the codebook. By plotting the component planes for all dimensions, all information on the prototype vectors is revealed. However, as with other methods in statistics, it becomes increasingly difficult to perceive important information such as clustering structure and underlying dependencies. We show how to aggregate component planes into groups and how to examine their differences in Section 4.3.

The unified distance matrix (U-Matrix; Ultsch and Siemon (1990)) is a visualization technique that shows the local cluster boundaries by depicting pair-wise distances of neighboring prototype vectors. It is the most common method associated with SOMs and has been extended in numerous ways. The Gradient Field that we describe in this paper has some similarities with the U-Matrix, but applies smoothing over a broader neighborhood and a different style of representation. One drawback of the U-Matrix is that it is less suited for large sparse SOMs (Emergent SOMs; Ultsch (1999)).

Another form of visualizing the map is clustering the prototype vectors (Vesanto and Alhoniemi, 2000), disregarding the SOM topology. This provides a vector quantization point of view and can be used to identify homogeneous regions. Another type of visualization is concerned with gradients (Kaski et al., 2000) for local models of the SOM, which is also closely related to our method.

The second category of visualization techniques takes the distribution of the data into account. The most simple ones are hit histograms, that show how

many data samples are mapped to a map unit, and labeling techniques, that plot the names and categories, provided they are available, of data samples onto the map lattice. More sophisticated methods include smoothed data histograms (Pampalk et al., 2002), which show the clustering structure by mapping each data sample to a number of map units, and graph-based methods (Pölzlbauer et al., 2005c), that show connections for map nodes that are close in feature space. The P-Matrix (Ultsch, 2003a) is another density-based approach that depicts the number of samples that are within a sphere with a certain radius around the prototype vectors. The radius is a quantile of the pair wise distances of the data vectors. The U*-Matrix (Ultsch, 2003b) is a combination of P-Matrix and U-Matrix, and is applied to large SOMs. The U-Matrix value, defined as the sum of distances of each node to its direct neighbors, is multiplied by a scaling factor induced by the local density of the data points around the corresponding prototype vector. This results in high values for areas where there the distance to neighboring areas is high and the data density is low, and low values otherwise.

The Generative Topographic Mapping (GTM; Bishop et al. (1998)) has been introduced as an alternative to the SOM with a continuous output space that models the probability distribution in feature space. The magnification factors visualization (Bishop et al., 1997) depicts local stretching of the mapping as ellipsoids in a discrete number of latent space centers. This method is related to our technique as it explains directional changes. Magnification factors can also be computed for the SOM, where a continuous interpolation is applied to the discrete SOM lattice in order to perform differential analysis. Apart from that, our method differs mainly in the way that smoothing is applied: While magnification factors for the SOM show similar results as the U-Matrix, we

apply a smoothing according to an adjustable parameter that defines the width of the area over which the differences are aggregated and investigated. Further extensions of magnification factors for GTM investigate their curvature (Tino et al., 2001).

3 Self-Organizing Maps and neighborhood kernels

A Self-Organizing Map is a mapping from an N-dimensional feature space to a low-dimensional output space. The output space is called the "map lattice" and consists of M neurons (map units). In this work, we assume the output space to be two-dimensional and the feature space to be a vector space over the real numbers ($\mathbb{R}^N$). We introduce a strict formal notation to differentiate between the various scenarios where vectors, coordinates or positions are used. We explicitly distinguish between a map node ξ_i on the two-dimensional map lattice and its corresponding prototype vector (or model vector) m_i in feature space. The index $1 \leq i \leq M$ connects a map node to its prototype vector. The set of all prototype vectors is called the codebook $\mathfrak{M}$. We denote the horizontal and vertical coordinates of ξ_i as ξ_i^{u} and ξ_i^{v}, respectively. The prototype vector m_i lives in the same space as the data samples. Another use of the word "vector" in this paper is in the context of the resulting visualization, that consists of a field of arrows pointing to cluster centers. We will refer to it as the Gradient Field method. There are two distance metrics involved in the SOM algorithm, one in feature space, the other one between units on the map lattice. The distance between prototype vectors m_i and m_j in feature space is denoted as

6

$$d_F(m_i, m_j) = \|m_i - m_j\|_F \tag{1}$$

where $\|\cdot\|_F$ is the distance measure of feature space also applied during SOM training, which can be basically any metric, such as the cosine distance, or any Minkowsky metric, while the most common distance usually being Euclidean (Aggarwal, 2003). d_F is also used to compute the distance between data samples and prototype vectors.

On the map lattice, the distance between units ξ_i and ξ_j is measured by the Euclidean Metric

$$d_O(\xi_i, \xi_j) = \|\xi_i - \xi_j\|_O = \sqrt{(\xi_i^{\mathrm{u}} - \xi_j^{\mathrm{u}})^2 + (\xi_i^{\mathrm{v}} - \xi_j^{\mathrm{v}})^2} \tag{2}$$

As our method is primarily concerned with post-processing of a trained SOM we do not introduce the training process in detail, but rather refer to (Kohonen, 2001) for an in-depth discussion. A crucial concept in SOM training is the neighborhood kernel h_σ, which is a monotonously decreasing function $h_\sigma : \mathbb{R} \to \mathbb{R}^+$. It converts the distance d_O into a measure of proximity. Far apart units ξ_i and ξ_j will have a low kernel value. Thus, the kernel acts as a weighting function for the influence of nodes ξ_i and ξ_j onto each other. Kernels are used in many fields of statistics such as probability density estimation; however, for use with SOMs, the kernel does not have to be a probability function with unit area.

There are numerous kernel functions, the most widely used one is the Gaussian kernel h_σ^G, which resembles a bell-shaped curve

$$h_\sigma^{\text{G}}(d_O) = \exp\left(-\frac{d_O^2}{2\sigma}\right) \tag{3}$$

The kernel value for distant nodes is exponentially decreasing and will be close to zero for $d_O > \sigma$. For computational reasons, the kernel can be cut off at this threshold ("cut-off Gaussian kernel"):

$$h_\sigma^{\text{G'}}(d_O) = \begin{cases} h_\sigma^{\text{G}}(d_O) \text{ if } d_O \leq \sigma \\ \\ 0 \qquad \text{otherwise} \end{cases} \tag{4}$$

A very simple kernel is the bubble kernel, which is a step function that is defined as

$$h_\sigma^{\text{b}}(d_O) = \begin{cases} 1 \text{ if } d_O \leq \sigma \\ \\ 0 \text{ otherwise} \end{cases} \tag{5}$$

It relies solely on the concept of cutting off outside the radius σ, weighting all distances up to this point equally. This function is not continuous in a mathematical sense.

Another choice is the inverse proportional kernel:

$$h_\sigma^{\text{ip}}(d_O) = \begin{cases} 1 - \frac{d_O^2}{\sigma^2} \text{ if } d_O \leq \sigma \\ \\ 0 \qquad \text{otherwise} \end{cases} \tag{6}$$

It is similar to the Gaussian Kernel but decreases faster to zero.

All of the above kernels are non-linear functions. The linear kernel is an ex-

ception:

$$h_\sigma^1(d_O) = \begin{cases} 1 - \frac{d_O}{\sigma} & \text{if } d_O \leq \sigma \\ \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

It decreases linearly from one to zero. The parameter σ deserves special attention. It determines the breadth of the neighborhood function, such that very high values correspond to high influence of far-away and close units alike, and very low values emphasize only the direct neighbors of the map unit. It is used in the proposed visualization techniques to control the granularity of the structures detected, serving as a smoothing parameter.

4 SOM visualization with vector fields

In this section, we introduce two methods for visualizing the clustering structure based on vector fields (Pölzlbauer et al., 2005b). The gradient is plotted on top of the SOM lattice with one arrow per map unit. The length and direction of each arrow indicate where the cluster centers are located. The entirety of the arrows forms a smooth vector field especially intended for use by professionals with engineering backgrounds, exploiting their familiarity with gradient and flow visualizations. The Borderline visualization is derived from the Gradient Field and provides an alternative view that emphasizes the cluster boundaries. In Subsection 4.3, we further extend the Gradient Fields to contrast contributing factors of the clustering structure.

4.1   Gradient Field Visualization

In this subsection we describe the algorithm for obtaining a vector field that shows homogeneous areas. Each of its arrows a_i is computed based on the prototype vectors, the map topology, and the choice of the neighborhood kernel. The a_i will be plotted on top of their corresponding map units ξ_i. The vectors a_i have a u and v component, denoted as a_i^{u} and a_i^{v}, corresponding to the horizontal and vertical coordinates, respectively. The algorithm outlined in the next paragraphs consists of two main steps which are repeated for each map unit: In the first one, weighted distances to all other prototype vectors are computed and separated along both the u and v axes in positive and negative directions. In the second step, these contributions are aggregated for each coordinate and normalized in order to avoid border effects.

From this point, we will outline the algorithm to compute the coordinates of a_i, for map unit ξ_i and its associated prototype vector m_i. Some of the formulas involve pair wise comparisons to other units and models vectors, which will be denoted with index j. These computations have to be performed for all $1 \leq j \leq M$ with $j \neq i$. First we need to obtain the vector connecting the positions of ξ_i and ξ_j in output space, which is defined as

$$\chi_{ij} = \vec{\xi_i \xi_j} = \xi_j - \xi_i \tag{8}$$

The angle α of this vector χ_{ij} is simply

$$\alpha_{ij} = \arctan\left(\frac{\chi_{ij}^{\mathrm{v}}}{\chi_{ij}^{\mathrm{u}}}\right) \tag{9}$$
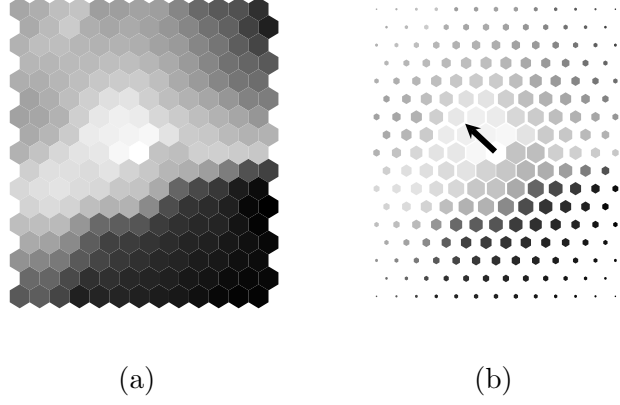
10

<center>(a)   (b)</center>

Fig. 1. (a) distances to prototype of map unit in the center; dark values denote high distances, (b) map unit size scaled according to Gaussian neighborhood kernel with $\sigma = 4$

Using the above notation, it is now possible to apply the neighborhood kernel to the distance between units ξ_i and ξ_j, which is the length of χ_{ij}, and to split up this weight into u and v directions:

$$\omega_{ij}^{\mathrm{u}} = \cos\alpha_{ij} \cdot h_\sigma(||\chi_{ij}||), \qquad \omega_{ij}^{\mathrm{v}} = \sin\alpha_{ij} \cdot h_\sigma(||\chi_{ij}||) \qquad (10)$$

The value of ω_{ij}^{u} will be close to zero when either the distance between ξ_i and ξ_j is high, resulting in a very low kernel value, or in case $\xi_i^{\mathrm{u}} = \xi_j^{\mathrm{u}}$, i.e. ξ_i is directly above or below ξ_j with no horizontal offset. The value of σ also influences the kernel function and thus the value of ω since high σ tend to weight far-away map units higher than with low σ values. Note that ω_{ij}^{u} will be negative in case ξ_j is to the left of ξ_i.

In the following, we will only explicitly provide formulas for the u coordinate, as v follows analogously. In the next step, the distances between the associated prototype vectors m_i and m_j are taken into account, weighting these distances by ω, and assigning them to either the positive or negative side of u:

<center>11</center>

$$
\delta_{ij}^{\mathrm{u}+} = \begin{cases} d_I(m_i, m_j) \cdot \omega_{ij}^{\mathrm{u}} & \text{if } \omega_{ij}^{\mathrm{u}} > 0 \\ \\ 0 & \text{otherwise} \end{cases} \tag{11}
$$

$$
\delta_{ij}^{\mathrm{u}-} = \begin{cases} d_I(m_i, m_j) \cdot (-\omega_{ij}^{\mathrm{u}}) & \text{if } \omega_{ij}^{\mathrm{u}} < 0 \\ \\ 0 & \text{otherwise} \end{cases} \tag{12}
$$

Dividing the weighted distance contribution of the prototype vectors along the positive and negative directions will provide a means to find the direction where the dissimilarity of the current vector m_i is relatively low, and where the arrow a_i will ultimately point to. If, for example, ξ_j is next to ξ_i on the right side, and the distances between the prototype vectors are high, then $\delta_{ij}^{\mathrm{u}+}$ will be high, which will contribute significantly to a_i pointing to the left, away from ξ_j. Figure 1 gives a schematic overview of the weighting process to the unit in the center of the map ξ_i: First, the pair wise feature space distances $d_I(m_i, m_j)$ are shown in Figure 1(a), where the ith node is the one in the middle of the map; then, the sizes of the hexagons are scaled according to the kernel value of their output space distance to the center node $h_\sigma(||\chi_{ij}||)$ in Figure 1(b). The arrow in this figure points in the direction where the least weighted distances are. Repeating this calculation for all the ξ_j, we can finally aggregate the $\delta_{ij}^{\mathrm{u}+}$ and $\delta_{ij}^{\mathrm{u}-}$ values:

$$
\rho_i^{\mathrm{u}+} = \sum_{j=1\ldots M, j\neq i} \delta_{ij}^{\mathrm{u}+}, \qquad \rho_i^{\mathrm{u}-} = \sum_{j=1\ldots M, j\neq i} \delta_{ij}^{\mathrm{u}-} \tag{13}
$$

Once we know $\rho_i^{\mathrm{u}+}$ and $\rho_i^{\mathrm{u}-}$, we can tell whether and to which extent one of the directions outweighs the other. For $\rho_i^{\mathrm{u}+} > \rho_i^{\mathrm{u}-}$, the accumulated distances

from the right side are bigger than the ones on the left, so the arrow will point to the left. Close to the map borders, the distance contributions in the direction of the border will always be lower than in the direction inside of the map since there are simply no prototype vectors for which a distance could be computed. The resulting arrows are thus biased to always point outside of the map. To avoid this, a normalization has to be performed that sums the ω_{ij}^{u} values in positive and negative directions

$$
\omega_i^{\mathrm{u}+} = \sum_{j=1...M, j \neq i} \begin{cases} \omega_{ij}^{\mathrm{u}} & \text{if } \omega_{ij}^{\mathrm{u}} > 0 \\ \\ 0 & \text{otherwise} \end{cases} \tag{14}
$$

$$
\omega_i^{\mathrm{u}-} = \sum_{j=1...M, j \neq i} \begin{cases} -\omega_{ij}^{\mathrm{u}} & \text{if } \omega_{ij}^{\mathrm{u}} < 0 \\ \\ 0 & \text{otherwise} \end{cases} \tag{15}
$$

Finally, we can determine the components of a_i as

$$
a_i^{\mathrm{u}} = \frac{\rho_i^{\mathrm{u}-} \cdot \omega_i^{\mathrm{u}+} - \rho_i^{\mathrm{u}+} \cdot \omega_i^{\mathrm{u}-}}{\rho_i^{\mathrm{u}+} + \rho_i^{\mathrm{u}-}} \tag{16}
$$

where the accumulated input space differences $\rho_i^{\mathrm{u}+}$ and $\rho_i^{\mathrm{u}-}$ are weighted by the opposing accumulated $\omega_i^{\mathrm{u}-}$ and $\omega_i^{\mathrm{u}+}$, respectively. In case the node ξ_i does not lie close to the edge of the u axis, $\omega_i^{\mathrm{u}+}$ and $\omega_i^{\mathrm{u}-}$ will be equal, and the effect of this normalization will vanish. The most important of the computational steps are performed in (13) and (16). The roles of ρ_i^+ and ρ_i^- of either u or v coordinates deserve special attention and will be briefly discussed:

- $\rho_i^+ \approx \rho_i^-$: In this case, distances are balanced and the component of a will be small. If the values of ρ_i^+ and ρ_i^- are small, the prototype vectors m_j of the surrounding map units of ξ_i are very similar to m_i and likely to be in

the center of a cluster. In case both ρ_i^+ and ρ_i^- are large, ξ_i is likely to be right in between two clusters to neither of which it belongs. Such units are called interpolating units and are easily recognized as arrows of neighboring units are pointing away from them.

- $\rho_i^+ > \rho_i^-$: The distances in positive direction outweigh distances in negative direction. m_i is more similar to its neighbors in negative direction and a_i will reflect this by pointing there.

- The length of a_i is determined by the difference between ρ_i^+ and ρ_i^-. Large differences result in long arrows.

Our method differs from the U-Matrix in the way that it can be represented as a field of arrows, and by the smoothing that is performed by the kernel to override small cluster boundaries that may be artifacts that come from the choice of the map size. The choice of the kernel width σ plays an important role in what the visualization actually shows, since a small value of σ weights the direct neighbors of the map unit ξ_i much stronger than the other units, while a large value takes the surrounding units into account, weighting them nearly equally, and thus smoothing over wide areas. The effect of choosing σ lies in whether the visualization is a fine-grained vector field, where the neighboring arrows vary sharply, or a coarse representation that show the global clustering structure. One of the main advantages of our method is that it allows interactively setting and tuning σ to different levels, so different kinds of clustering structures can be observe. The value of σ has to be seen in relation to the size of the map. For usual settings, we recommend setting σ to around one sixth of the number of units along the shorter side of the map, which usually results in a balanced visualization that is neither coarse nor granular.We provide examples and further discussion of σ in Subsection 5.1.

14

The computational complexity of calculating the vector field is $O(M^2)$ since it requires the pair wise distances between the map nodes and prototype vectors. In case the kernel is cut off after a certain threshold, like $h^{\mathrm{G'}}$, h^{b}, h^{ip}, or h^{l}, which is plausible since distant map nodes influence each other by negligible amounts, the complexity reduces to $O(M)$. The computational cost of the U-Matrix also scales linearly with the number of map units, thus computation of the vector field is not more expensive in terms of complexity than the U-Matrix. The experiments in Section 5, which include SOMs with up to 2000 nodes, have been computed in several seconds even for the $O(M^2)$ case, running under an implementation in Matlab code, thus offering itself for interactive visualization with different sets of σ.

4.2 Borderlines representation

Another representation that emphasizes cluster boundaries over vector fields pointing towards cluster centers can be easily derived. By depicting the orthogonal of each a_i as a line from both sides of the center instead of an arrow, the resulting visualization shows likely cluster boundaries. The length of the arrows is preserved such that long lines hint at a strong separation between clusters. We call this representation the Borderline visualization. An example is shown in the next section.

4.3 Extension to groups of component planes

In this subsection, we describe an extension of the Gradient Field technique. As data mining methods usually assume that the observed data samples follow

an unknown statistical distribution, local correlations and dependencies between variables are often an interesting property to assess. We assume that the clustering structure is induced by certain groups of variables. The variables of these distinct groups are either correlated to a certain degree or statistically dependent in a non-linear way. Our assumption implies that the clustering structure can be decomposed into these groups of variables. The basic idea is to plot two or more groups simultaneously with the Gradient Field method. This combined visualization shows the contributing factors of the clustering and gives an additional feel about the underlying data.

The reliance on variables and component planes rather than codebook vectors and their distances requires us to introduce some additional formal definitions. A projection of the the codebook's i^{th} variable is called the ith component plane C^i, which lies in a one-dimensional subspace of the feature space. Component planes can be conveniently visualized on the map lattice. We define groups of component planes as the combination of several variables, and denote $\mathfrak{S} = \{1, \ldots, m\}$ as the set that consists of all the indices of these variables. We are further interested in comparing g subsets $\mathfrak{S}_{1,\ldots,g} \subseteq \mathfrak{S}$. The sets must be disjoint, i.e. $\mathfrak{S}_i \cap \mathfrak{S}_j = \emptyset, \quad i, j \leq g, \forall i \neq j$ must hold. The union $\bigcup_{i=1}^{g} \mathfrak{S}_i$ does not necessarily have to contain all the variable indices if only a subset of the variables is of interest. We denote by $\mathfrak{M}^{(\mathfrak{S}_i)} = \bigtimes_{k \in \mathfrak{S}_i} C^k$, where $\times$ refers to the Cartesian Product, the reduced codebook which consists of the component planes $\mathfrak{S}_i$. $\mathfrak{M}^{(\mathfrak{S}_i)}$ is a "sub-SOM" with the same number of map units, for which e.g. the U-Matrix and Gradient Field visualizations can be computed. Interesting subsets $\mathfrak{S}_{1,\ldots,g}$ can be chosen either based on correlations between component planes (Vesanto and Ahola, 1999; Himberg, 1998) or by grouping variables that are known to be semantically similar, for example variables

16

that have a common source. The former choice can be performed by either investigating the correlation matrix or by visual inspection of the component planes, combining the ones that are similar. Determining the correlation of component planes can be performed by

$$d_{comp}(i,j) = \text{abs}(\gamma(C^i, C^j)) \tag{17}$$

where $\gamma(C^i, C^j)$ is a suitable measure of correlation such as the Pearson correlation coefficient on the variables C^i and C^j. For highly similar component planes, the absolute value is close to 1. If there is no linear correlation, its value is close to zero. Grouping components together is then performed by partitioning the set of component planes. The more interesting choice is grouping variables that are semantically similar, since this exploits a-priori knowledge about the data. We then investigate whether the groups provide contrasting clustering structures. For a thorough discussion of this approach, see (Pölzlbauer et al., 2005a).

Once the relevant variable groups have been selected, the Gradient Field method can be applied to both sets and visualized simultaneously. In order to adjust the length of the arrows to negate the effect of different numbers of variables between the sets, we compute

$$\hat{a}_i^{(\mathfrak{S}_1)} = a_i^{(\mathfrak{S}_1)} \frac{|\mathfrak{S}_1|}{|\mathfrak{S}_1| + |\mathfrak{S}_2|} \tag{18}$$

where $|\cdot|$ denotes the cardinality of a set. In the next section, we provide examples that explain the relation to variable dependence and applications on a real-world data set.

17

5 Experiments

In this section, we will investigate the usefulness of the previously described techniques with two real-world data sets and several artificial ones. The first one is the "Phonetic" benchmark data[1], which describes 20 distinct phonemes from continuous Finish speech, which are measured in 20 variables. It consists of 1962 data samples. The SOM trained on this data set consist of 30×40 map units. The other real-world data set comes from the domain of petroleum engineering (Zangl and Hannerer, 2003). The "Fracture Optimization" data set has been collected from 199 gas wells in 10 dimensions. The variables can be grouped according to their source and use, which we will discuss in more detail below. We have trained two SOMs on this data set, one with 7×10 map units, and one with 44×44 to show the characteristics of our method for largely differing SOM sizes of both conventional as wel as emergent SOM types. We show the general properties of our method with the Phonetic data set, and the dual visualization and correlation discussion with the Fracture Optimization data. The artificial data sets thereafter are used to stress our method's capabilities of visualizing non-linear correlations between groups of variables.

5.1 Effects of the neighborhood radius

At first, we want to show the effects of the kernel width σ with the SOM trained on the Phonetic data set with a Gaussian kernel h^{G}. In this dataset, many of the phoneme classes coincide with the homogeneous areas on the

[1] Included in LVQ-PAK, available at http://www.cis.hut.fi/research/software

18

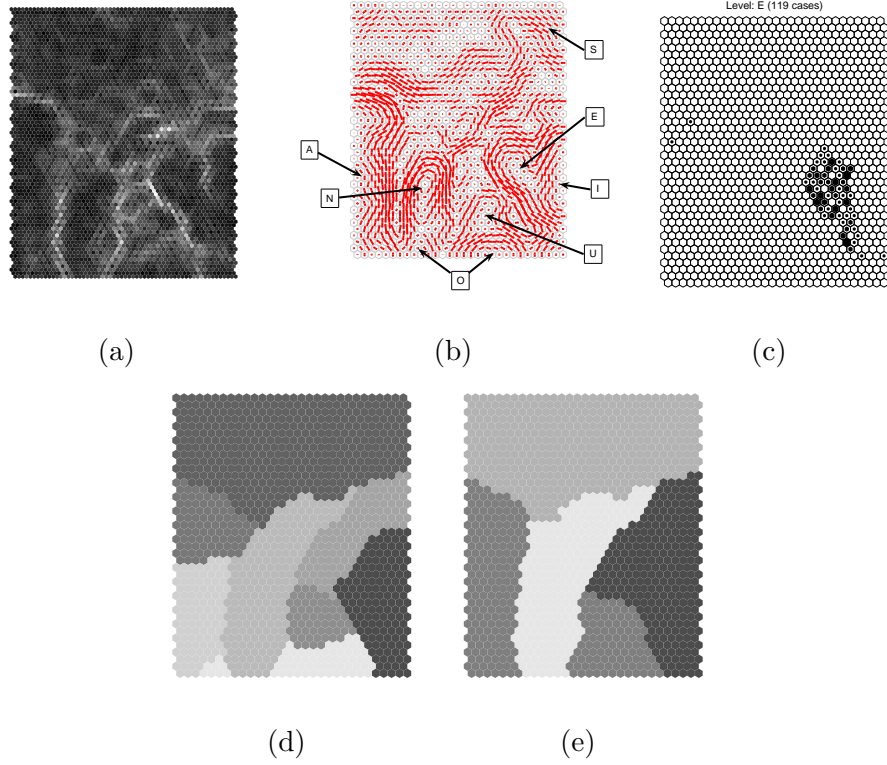|		
(a)	(b)	(c)

(d)	(e)

Fig. 2. Phonetic data mapped to the SOM: (a) U-Matrix, (b) Clusters and corresponding labels, (c) hit histogram of data with label "E", (d) k-means clustering with 8 clusters, (e) k-means clustering with 4 clusters

map. The U-Matrix of this 30×40 SOM is shown in Figure 2(a). Figure 2(b) highlights some regions that are primarily occupied by one vowel or consonant. As an example for a phoneme that is highly clustered, Figure 2(c) shows the hit histogram for data samples of phoneme "E". In Figure 3, the results for both Gradient Field and Borderline methods for $\sigma = 1, 5, 15$ are shown. The σ values "1", "5", and "15" have been chosen to represent low, medium, and high radii, respectively, in relation to the map size of 30×40. Low values of this parameter lead to very granular visualizations, where only direct neighbors are taken into account for the computation of each arrow and thus only local gradients can be observed, as visualized in Figures 3(a),(d). By increasing σ, the clustering structure revealed shifts gradually from local towards global. Figures 3(b),(e) provide a far better overview on the clustering structure, and
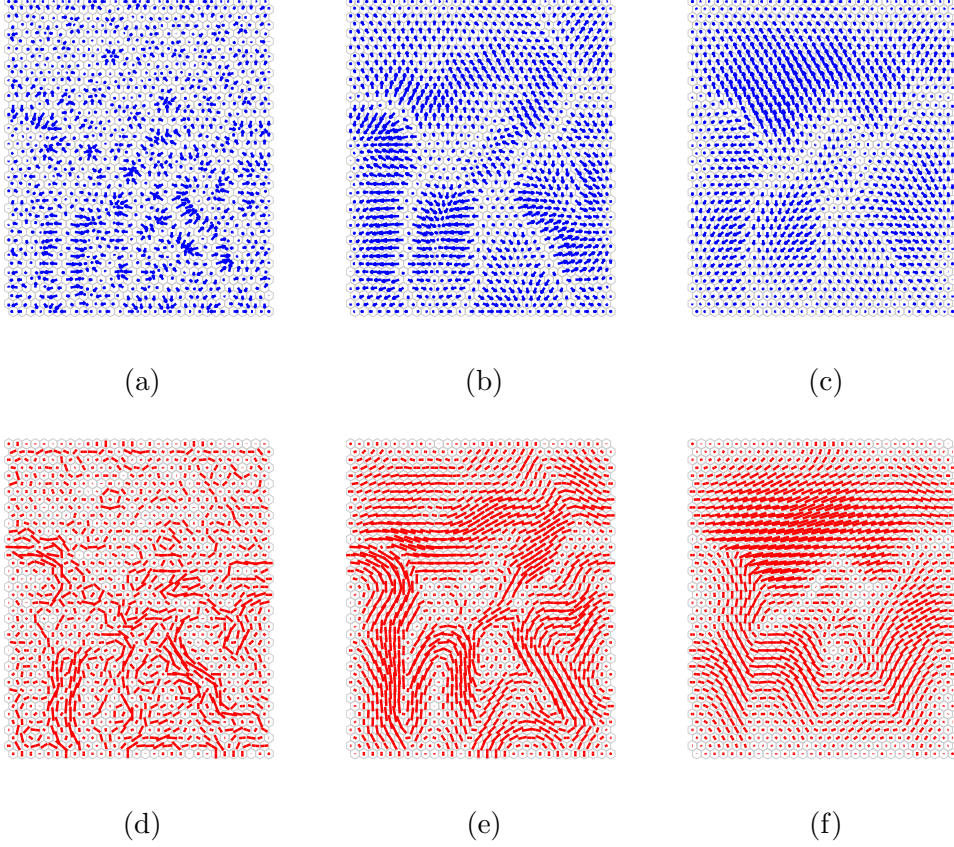
19

Fig. 3. Phonetic data with 30×40 SOM, Gradient Field visualization: (a) $\sigma = 1$, (b) $\sigma = 5$,(c) $\sigma = 15$; Borderline visualization: (d) $\sigma = 1$, (e) $\sigma = 5$,(f) $\sigma = 15$

individual regions can be distinguished. In Figures 3(c),(f), the global structure is shown for $\sigma = 15$. It reveals that the upper third is most strongly separated from the rest of the map, which has not been indicated by any of the former, more local representations. The relation to a clustering algorithm is shown in Figures 2(d),(e), which show the results of k-means of the codebook vectors with 8 and 4 clusters. The map topology is omitted for the clustering process, so clusters can consist of non-adjacent nodes. σ is loosely related to the number of clusters, as high values of σ show few boundaries and are comparable to clustering with few cluster centers, while a low σ results in many local cluster boundaries and is thus comparable to clustering with many cluster centers. When compared to Figure 3, it can be seen that the $k = 4$ clustering reveals
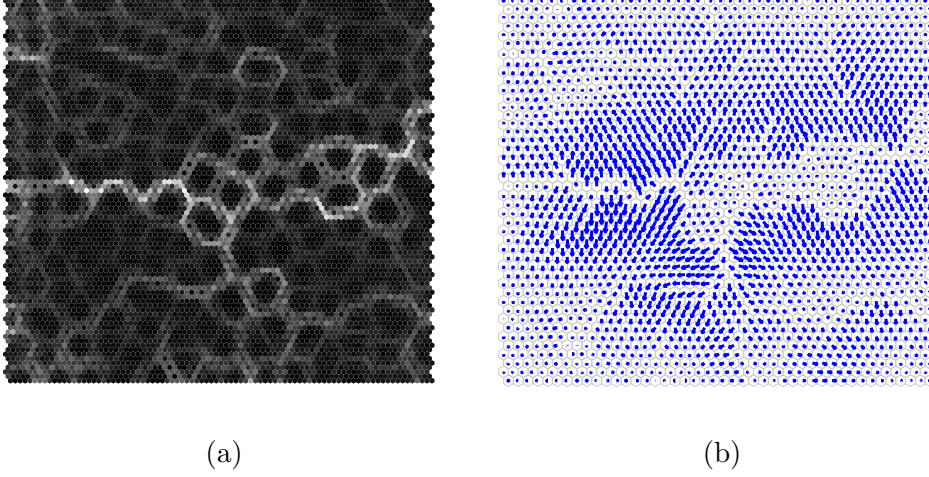
(a)   (b)

Fig. 4. Fracture Optimization data with 44×44 SOM: (a) U-Matrix, (b) Gradient Field with $\sigma = 8$

similar information as the Gradient Field method with a high σ, while $k = 8$ is comparable to the ones with a lower smoothing parameter. The choice of σ has to be performed interactively and varies for the type of kernel used, but a good starting point for the Gaussian kernel h^{G} is $1/6^{\mathrm{th}}$ of the shorter side of the map. σ always has to be seen in relation to the size of the map since it is defined in absolute terms over the number of units over which the smoothing is performed. While the choice of σ strongly influences the type of information on the structure revealed, the results are insensitive to the choice of the neighborhood kernel function. We have not noticed significant differences between different neighborhood kernels. The only exception is the Bubble kernel h^{b}, which is is not continuous in a mathematical sense and does not allow a smooth convergence across the neighborhood range, and is also hardly used for SOM training. Thus, a cut-off kernel variant may be employed, resulting in linear complexity for calculating the vector field.

21

5.2 Smoothing sparse maps

The next example shows the smoothing effect on sparse maps which are sometimes preferred if the SOM is used for visualization only and vector quantization is not of interest. We use a 44×44 as an example of an oversized SOM trained on the Fracture Optimization data set with an inverse proportional kernel h^{ip}. The Fracture Optimization data set that we use for this experiment consists of 199 samples in 10 variable dimensions, where each sample corresponds to a well for gas drilling. Since the number of the map nodes (1936) is about ten times higher than the number of data samples (199), the U-Matrix visualization shown in Figure 4(a) shows artifacts that come from the fact that most samples are mapped to nodes that they occupy solely. The remaining units are merely interpolating units. The U-Matrix implies that these transitions are actually cluster boundaries, while the significant boundary between the upper and lower parts are overshadowed. In Figure 4(b), a Gradient Field with $\sigma = 8$ is visualized, the radius being roughly one sixth of the map axes. The smoothing effect overrides the insignificant cluster boundaries and focuses on a more global clustering structure, which consists of two big clusters on the upper and lower part of the map and a transition region slightly right of the center. It is thus possible to adapt the visualization accordingly for sparse SOMs.

5.3 Dual Gradient Fields on petroleum engineering data

Next, we discuss the combination of Gradient Fields to show how groups of component planes influence the overall clustering structure with a 7×10 SOM
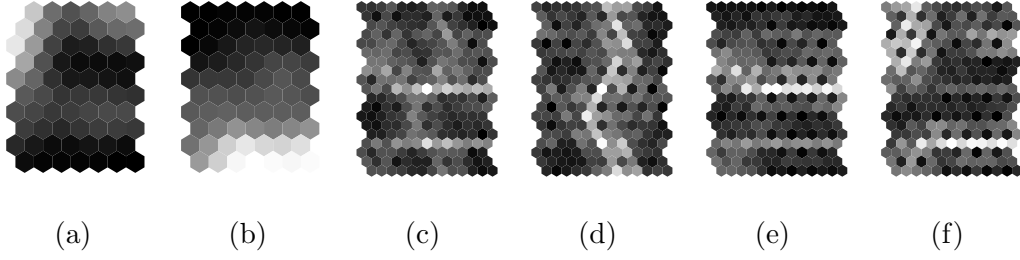
22

(a)　　　(b)　　　(c)　　　(d)　　　(e)　　　(f)

Fig. 5. Fracture Optimization, 7×10 SOM: (a) Component plane "produced gas", (b) component plane "stimulation costs"; (c) U-Matrix, (d) U-Matrix of $\mathfrak{M}^{(\mathfrak{S}_{geo})}$, (e) $\mathfrak{M}^{(\mathfrak{S}_{param})}$, (f) $\mathfrak{M}^{(\mathfrak{S}_{output})}$
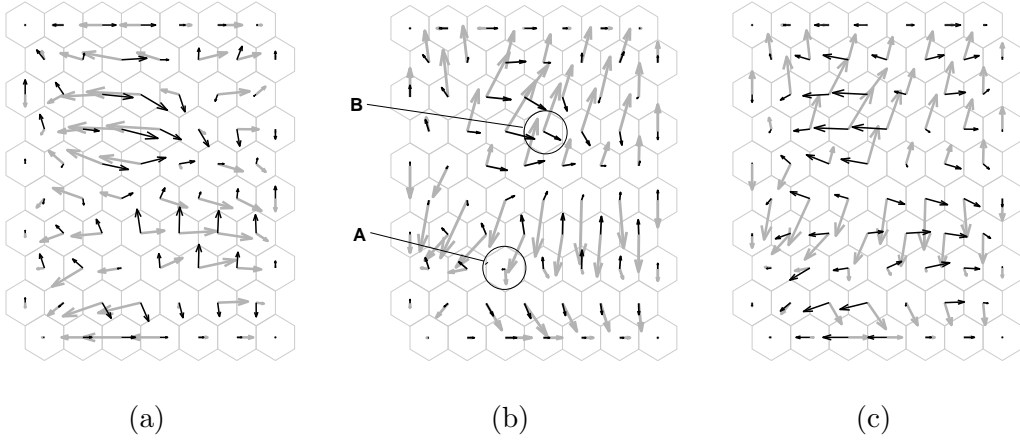

(a)　　　　　　　(b)　　　　　　　(c)

Fig. 6. Fracture Optimization SOM Dual Gradient Fields (the first group is indicated by black, and the latter by grey vectors): (a) $\mathfrak{M}^{(\mathfrak{S}_{out})}$ vs $\mathfrak{M}^{(\mathfrak{S}_{geo})}$, (b) $\mathfrak{M}^{(\mathfrak{S}_{out})}$ vs $\mathfrak{M}^{(\mathfrak{S}_{param})}$, (c) $\mathfrak{M}^{(\mathfrak{S}_{geo})}$ vs $\mathfrak{M}^{(\mathfrak{S}_{param})}$

trained on the Fracture Optimization data. The map size is determined by the number of training data samples and the ratio of the axes is calculated from the ratio of the first to the second largest eigenvalue from principal component analysis, as performed by default in the Matlab SOM Toolbox (Vesanto et al., 1999). The variable dimensions come from 3 types of sources: (1) Geological factors that describe properties mainly determined by the choice of the geographic position; (2) Engineering parameters set during the gas pumping process; and (3) Output parameters that assess the success of the operation: "Stimulation Costs" and "Produced Gas". The data is gathered during three steps, each corresponding to one of these groups. First, the position where to

23

build the well after checking the geological data is selected; then, the engineering parameters are determined and the pumping proppant and fluids is started; and finally, after the gas has been obtained, the output variables can be assessed. The index sets are denoted as $\mathfrak{S}_{\mathrm{geo}}$ (3 dimensions), $\mathfrak{S}_{\mathrm{param}}$ (5 dimensions), and $\mathfrak{S}_{\mathrm{out}}$ (2 dimensions), respectively. We have trained a 7×10 SOM on this data set and want to find out how these groups of variables depend on each other and how they decompose the clustering structure. Note that we use the output variables in the same way as the other ones for training since the SOM is an unsupervised learning method. What we intend to do is thus related to canonical correlation analysis rather than to regression. Data analysts are concerned with measuring the impact of the choice of the well's position or the fine-tuning of the engineering parameters on the output.

Figures 5(c)–(f) show the U-Matrices for the SOM and the sub-SOMs $\mathfrak{M}^{(\mathfrak{S}_{\mathrm{geo}})}$, $\mathfrak{M}^{(\mathfrak{S}_{\mathrm{param}})}$, and $\mathfrak{M}^{(\mathfrak{S}_{\mathrm{out}})}$. From these, an impression of the the cluster boundaries can be gained. The engineering parameters seem to divide the map horizontally, while the geological factors are responsible mainly for a vertical boundary. The output parameters are the most interesting ones, since we aim to explain which regions of the map correspond to desirable outcomes, i.e. where wells with low costs and high produced gas are located. Figures 5(a),(b) show the component planes for these variables. It can be seen that the costs are high in the lower part of the map, and low in the upper regions. The gas production is high for wells that are mapped to the left border and slightly below the center of the map. Thus, the most desirable position for wells is the upper left corner with both low costs and high output. Figures 6(a)–(c) show the pair wise Dual Gradient Fields of the three groups. For the arrows pointing in different directions for most parts of the map, the underlying variable

24

groups are likely to be statistically independent and explain different parts of the overall clustering structure, which will be discussed in the next example. Figure 6(c) shows $\mathfrak{S}_{\text{geo}}$ and $\mathfrak{S}_{\text{param}}$, where the arrows are orthogonal in most cases. This information can be exploited in order to improve the fracture optimization process. The horizontal position of the sample projected onto the map is apparently determined by the geological factors since the black arrows are parallel to the horizontal axis. The vertical position corresponds to the engineering parameters. Thus, once the well is physically built, the geological factors cannot change anymore, and the horizontal position on the SOM describes a constraint for the effect of tuning the engineering parameters. It is desirable to shift a well towards the upper left corner to optimize output and costs. The lengths of the arrows correspond in how much a parameter has to be changed to achieve a change in the node that the sample is projected to. For example, Figure 6(b) shows $\mathfrak{S}_{\text{out}}$ (black) and $\mathfrak{S}_{\text{param}}$ (grey). Suppose a well is mapped to position "A", where the black arrow is short, while the grey arrow is long, thus moving one node up would require changing the engineering parameter by a large amount, while resulting only in small differences in output. For position "B", the arrows are approximately orthogonal, thus changing the parameters would only have marginal effects on the output since the gradients do not indicate that there is a change in output vertically.

5.4   *Statistical dependencies between groups of variables*

Next, we examine the effect of the Dual Gradient Field method on 4 artificial data sets where we want to find out whether one variable is statistically dependent on the other two. The data sets consist of $10,000$ samples. The
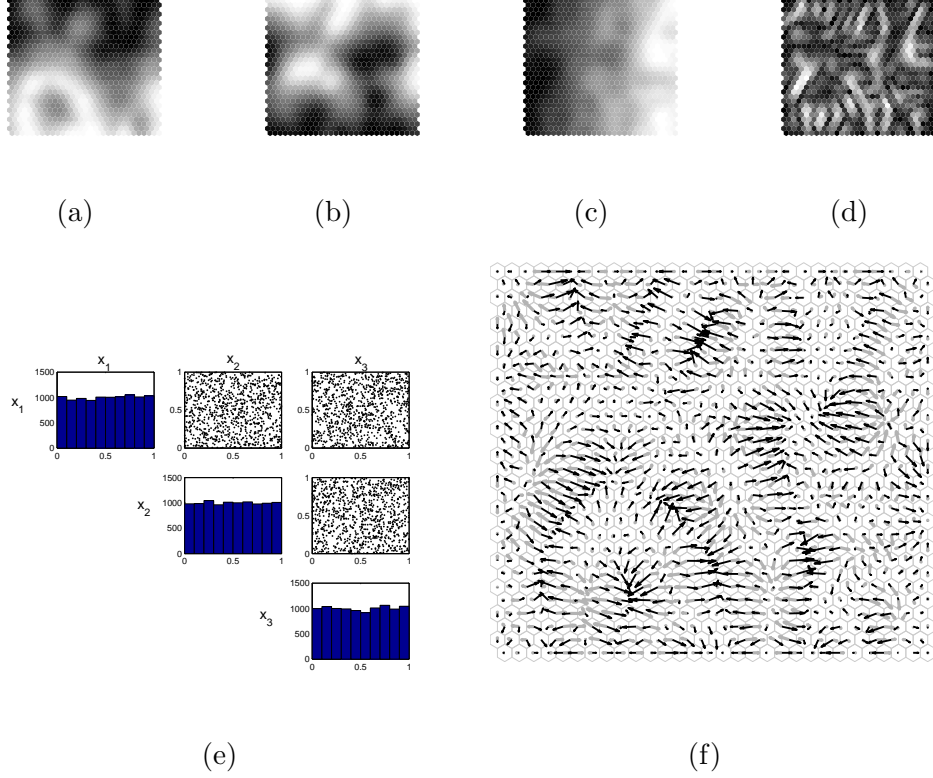
(a)   (b)   (c)   (d)

(e)   (f)

Fig. 7. Artificial Data SOM (no relationship): Component planes: (a) x_1, (b) x_2, (c) x_3; (d) length of difference vector, (e) scatterplots and distribution of x_1, x_2, x_3, (f) Dual Gradient Fields $\mathfrak{M}^{(\mathfrak{S}_{\text{uniform}})}$ vs $\mathfrak{M}^{(\mathfrak{S}_{\text{prob\_dep}})}$

first 3 examples are 3-dimensional, and the last one is 20-dimensional. In the 3-dimensional examples, the first two variables x_1 and x_2 are uniformly distributed between 0 and 1, and are statistically independent. The set of indices forming this group is denoted as $\mathfrak{S}_{\text{uniform}} = \{1, 2\}$. The second group consist of the third probably dependent variable $\mathfrak{S}_{\text{prob\_dep}} = \{3\}$ for which we want to know whether it can be explained by the former two variables.

In the first example, we consider a third variable x_3 that is independent of the former two and is also uniformly distributed. Scatterplots for this data set are provided in Figure 7(e), which show the variables x_1, x_2 and x_3 in rows and columns, and pair-wise scatterplots where they intersect; the bar charts show the distribution of each single variable. The scatterplots clearly show that
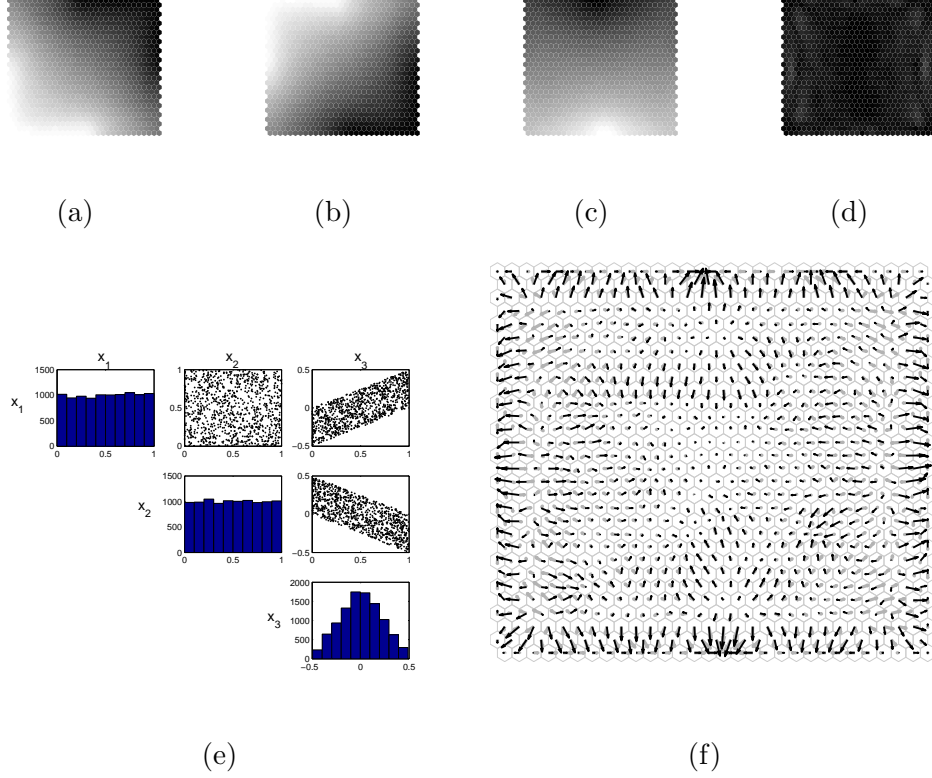
26

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

(e)   (f)

Fig. 8. Artificial Data SOM (linear relationship): Component planes: (a) x_1, (b) x_2, (c) x_3; (d) length of difference vector, (e) scatterplots and distribution of x_1, x_2, x_3, (f) Dual Gradient Fields $\mathfrak{M}^{(\mathfrak{S}_{\text{uniform}})}$ vs $\mathfrak{M}^{(\mathfrak{S}_{\text{prob-dep}})}$

there is no correlation between any of the variables. Figure 7(a)–(c) shows the component planes after training a 30×30 SOM with a cut-off Gaussian kernel $h^{\text{G'}}$ on this data. The Dual Gradient Fields in Figure 7(f) show that most arrows do not have common directions. To emphasize this, we have calculated the length of the vector connecting the two arrows $\|a_i^{\mathfrak{S}_{\text{uniform}}} - a_i^{\mathfrak{S}_{\text{prob-dep}}}\|_O$ in Figure 7(d). This value is high (light values) if the black and grey arrows point in different directions and low (dark values) if the black and grey arrows are similar. The figure has numerous light nodes, and thus indicates that there are no dependencies between the variable groups.

In the second setup, the third coordinate of each sample is defined as $x_3 = \frac{x_1 + x_2}{2}$. Scatterplots can be seen in Figure 8(e). Pearson's correlation coefficient
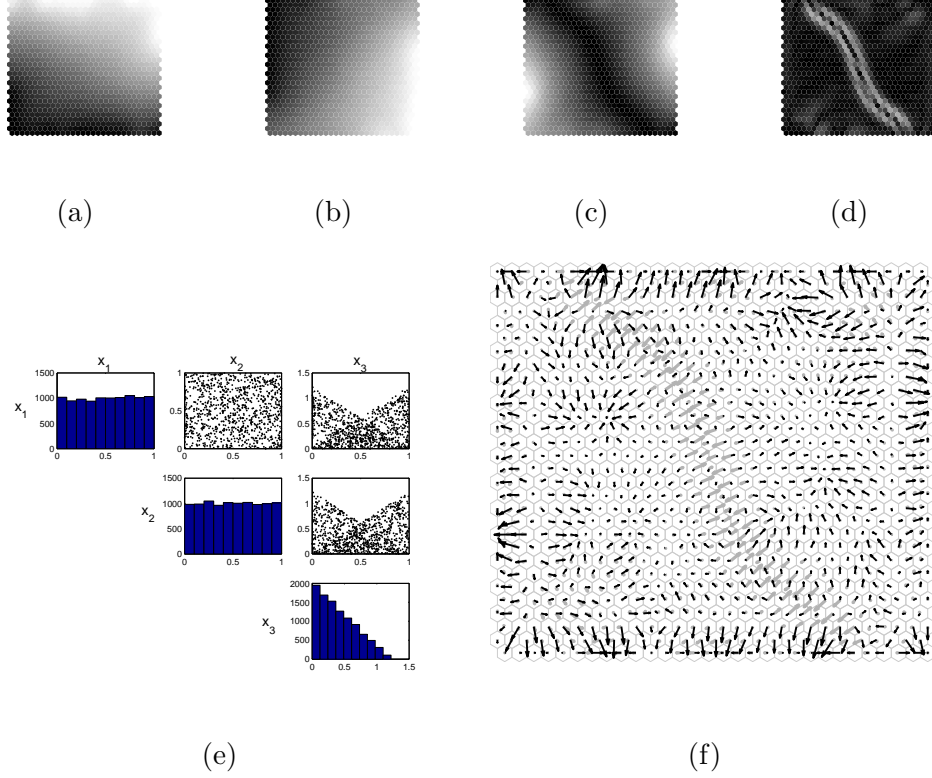
27

(a)　　　　　(b)　　　　　(c)　　　　　(d)

(e)　　　　　　　　　　　　(f)

Fig. 9. Artificial Data SOM (non-linear relationship): Component planes: (a) x_1, (b) x_2, (c) x_3; (d) length of difference vector, (e) scatterplots and distribution of x_1, x_2, x_3, (f) Dual Gradient Fields $\mathfrak{M}^{(\mathfrak{S}_{\mathrm{uniform}})}$ vs $\mathfrak{M}^{(\mathfrak{S}_{\mathrm{prob\_dep}})}$

between x_1 and x_3 is 0.7, and between x_2 and x_3 is -0.7, indicating strong linear dependence. The component planes plots for the 30×30 SOM on this data are shown in Figures 8(a)–(c). It can be seen that the projection results in linear ascent along diagonal lines, which are orthogonal for x_1 and x_2, stressing their independence. This has not happened in the previous example. Here, the data set is a 2-dimensional subspace embedded in the 3-dimensional feature space, and thus equal in dimension to the map lattice. The dependent component x_3 interferes with the other axes. When we apply the Dual Gradient Field method for groups $\mathfrak{S}_{\mathrm{uniform}}$ and $\mathfrak{S}_{\mathrm{prob\_dep}}$, the result can be seen in Figure 8(f). Aside from some deviations introduced by the SOM's border effect, it shows that the cluster structure of this map is caused by the same factors,

28

since the black and grey arrows are very similar both in angle and length. This is an expected result, since the third variable is predictable, and it will not introduce a different clustering structure than the one already present from the previous coordinates. Figure 8(d), where again the differences of the black and grey arrows are depicted, shows that the arrows are very similar for all parts of the map.

In the third case, the dependent variable is given as $x_3 = \text{abs}(x_1 + x_2 - 1)$ which is then multiplied by a factor to normalize its variance to the other variables. Although there exists a deterministic relationship between the variables, the correlation is zero for all pairs of variables, since there is no global linear relationship. x_3 is only piece-wise linearly dependent. Scatterplots are shown in Figure 9(e), which reveal that there is some sort of dependency between x_1 and x_3, and between x_2 and x_3. Component planes for the SOM trained on this data are visualized in Figures 9(a)–(c). The component plane for x_3 shows that the peak vales are on two edges of the map, the SOM thus has adjusted properly to this 2-dimensional manifold. By applying the Gradient Fields, the arrows in the regions with a linear relationship are almost identical. The smoothing performed to obtain the arrows only weights gradients within a certain radius, and the linear relationship can be observed within this radius. In the transition region, however, where x_3 approaches zero, no linear relationship is found, resulting in high differences between the arrows. Again, in Figure 9(d) the lengths of the difference vector between the arrow is depicted. This deviation shows where the linear relationship is not given, but recognizing that there is a linear relationship in most other areas of the map. When compared to Figures 7(d) and 8(d), where the same is performed for the independent and the linearly dependent case, we can now show in Figure 9(d)
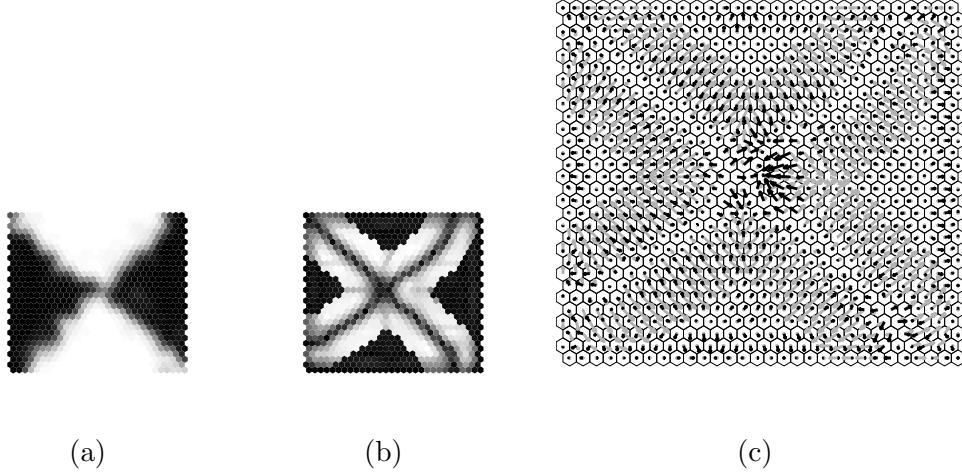
Fig. 10. Artificial Data SOM (XOR-like relationship): (a) Component plane x_{11}, (b) length of difference vector, (c) Dual Gradient Fields $\mathfrak{M}^{(\mathfrak{S}_{\text{uniform}})}$ vs $\mathfrak{M}^{(\mathfrak{S}_{\text{prob\_dep}})}$

where the visualization recognizes piece-wise linear relationships. While most statistical coefficients fail to quantify the deterministic dependence of x_3 in this case, our method can be used in order to identify piecewise linear portions of the data. Also, this implies that if cluster structures of variable groups overlap in certain regions of the map, we can learn about piece-wise dependencies between the variables.

The last example examines a more complex high-dimensional data set in 20 variables. Again, the data are split into two groups, and the dependence of the second group $\mathfrak{S}_{\text{prob\_dep}} = \{11, \ldots, 20\}$ is investigated. The first group $\mathfrak{S}_{\text{uniform}} = \{1, \ldots, 10\}$ is further divided into two subgroups of 5 dimensions. The variables in these subgroups are equally distributed with zero mean, and are constructed in a way that they are highly correlated within each subgroup with correlation coefficient of 0.9. Pairs of variables from different subgroups are not correlated. The second group of 10 variables is dependent on the first one and is constructed by an XOR-like function

$$x_k = \text{sign}(x_i) \cdot \text{sign}(x_j) \tag{19}$$

with $11 \leq k \leq 20$ the variable from the second group to be computed, $1 \leq i \leq 5$ a variable of the first subgroup, and $6 \leq j \leq 10$ a variable from the second subgroup. The results for the Dual Gradient Flow can be seen in Figure 10. Figure 10(a) shows an example component plane of $\mathfrak{S}_{\text{prob\_dep}}$ computed by (19). In Figure 10(c), the results of the Dual Gradient Flow method are depicted. The arrows are highly divergent in the regions where the boundaries are: The grey arrows, denoting $\mathfrak{M}^{(\mathfrak{S}_{\text{prob\_dep}})}$, point away from these boundaries towards their 4 cluster centers, while the black arrows are almost uniform over the map, with a small disturbance in the middle of the map that was probably introduced during training. The difference is visualized in Figure 10(b), which shows the deviation of the arrows. In the dark areas, the statistical relationship between the two groups is evident, while the light areas correspond to transitions. This is another example of a non-linear dependency that cannot be captured by a linear correlation coefficient, which is zero for pairs of variables from $\mathfrak{S}_{\text{prob\_dep}}$ and $\mathfrak{S}_{\text{uniform}}$.

6 Conclusion and Future Work

In this paper, we have described a visualization technique for Self-Organizing Maps with gradient fields that are especially aimed at professionals with engineering backgrounds. The method can be displayed either as a flow diagram where arrows point in the direction of most likely cluster centers, or as an equivalent that emphasizes at showing cluster boundaries. It has a parameter that determines how much smoothing is applied to the resulting visualization.

We have also provided an extension to simultaneously plot multiple groups of variables to show the decomposition of the clustering structure in contributing factors. We have also shown that this method can be used to detect linearly and non-linearly dependencies between variables.

Future work will be directed towards efforts to cluster variables for the dual visualization, such that this task can be performed automatically without need for explicit user interaction. Another enhancement may be achieved by integrating data density information in input space, as the current model relies solely on the prototype vectors. A further possible extension is the analysis and visualization of the rate of change of the arrows, thus investigating the second order differentiation of the feature space distances. Also, the applicability of the vector field visualization to other related mapping models is currently under investigation.

References

Aggarwal, C. C., 2003. Towards systematic design of distance functions for data mining applications. In: ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'03). ACM Press, New York, NY, USA, pp. 9–18.

Bishop, C., Svensen, M., Williams, C., 1997. Magnification factors for the GTM algorithm. In: Workshop on Self-Organizing Maps (WSOM'97). pp. 333–338.

Bishop, C., Svensen, M., Williams, C., 1998. GTM: The generative topographic mapping. Neural Computation 10 (1), 215–234.

Dittenbach, M., Rauber, A., Merkl, D., 2002. Uncovering hierarchical structure

in data using the growing hierarchical self-organizing map. Neurocomputing 48 (1–4), 199–216.

Fritzke, B., November 1994. Growing cell structures – a self-organizing network for unsupervised and supervised learning. Neural Networks 7 (9), 1441–1460.

Grinstein, G., Trutschl, M., Cvek, U., 2001. High-dimensional visualizations. In: Data Mining Conference KDD Workshop 2001. ACM Press, New York, San Francisco, CA, USA, pp. 7–19.

Hartigan, J. A., Wong, M. A., 1979. A K-means clustering algorithm. Applied Statistics 28, 100–108.

Himberg, J., 1998. Enhancing SOM-based data visualization by linking different data projections. In: International Symposium on Intelligent Data Engineering and Learning (IDEAL'98).

Kaski, S., Kangas, J., Kohonen, T., 1998. Bibliography of self-organizing map (SOM) papers: 1981–1997. Neural Computing Surveys 1, 102–350.

Kaski, S., Nikkilä, J., Kohonen, T., 2000. Methods for exploratory cluster analysis. In: International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet. L'Aquila, Rome, Italy.

Kohonen, T., 2001. Self-Organizing Maps, 3rd edition. Springer, Berlin, Heidelberg.

Lee, J., Lendasse, A., Verleysen, M., 2004. Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis. Neurocomputing 57, 49–76.

Martinetz, T., Berkovich, S., Schulten, K., 1993. "Neural-gas" Network for Vector Quantization and its Application to Time-Series Prediction. IEEE Transactions on Neural Networks 4 (4), 558–569.

Oja, M., Kaski, S., Kohonen, T., 2001. Bibliography of self-organizing map

(SOM) papers: 1998-2001 addendum. Neural Computing Surveys 3, 1–156.

Pampalk, E., Rauber, A., Merkl, D., August 27–30 2002. Using smoothed data histograms for cluster visualization in self-organizing maps. In: Proceedings of the International Conference on Artifical Neural Networks (ICANN'02). Springer, Madrid, Spain, pp. 871–876.

Pölzlbauer, G., Dittenbach, M., Rauber, A., September 5–8 2005a. Gradient visualization of grouped component planes on the som lattice. In: Cottrell, M. (Ed.), Proceedings of the Fifth Workshop on Self-Organizing Maps (WSOM'05). Paris, France, pp. 331–338.

Pölzlbauer, G., Dittenbach, M., Rauber, A., July 31 – August 5 2005b. 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). IEEE Computer Society, Montreal, Canada, pp. 1558–1563.

Pölzlbauer, G., Rauber, A., Dittenbach, M., May 30 – June 1 2005c. Advanced visualization techniques for self-organizing maps with graph-based methods. In: Jun Wang, Xiaofeng Liao, Z. Y. (Ed.), Proceedings of the Second International Symposium on Neural Networks (ISNN'05). Springer-Verlag, Chongqing, China, pp. 75–80.

Sammon, J. W., May 1969. A nonlinear mapping for data structure analysis. IEEE Transactions on Computers C-18 (5), 401–409.

Skupin, A., 2004. A picture from a thousand words. Computing in Science and Engineering 6 (5), 84–88.

Tenenbaum, J. B., de Silva, V., Langford, J. C., 2000. A global geometric framework for nonlinear dimensionality reduction. Science 290 (5500), 2319–2323.

Tino, P., Nabney, I., Sun, Y., 2001. Using directional curvatures to visualize

folding patterns of the GTM projection manifolds. In: International Conference on Artificial Neural Networks (ICANN'01). Springer, pp. 421–428.

Torgerson, W., 1952. Multidimensional scaling: I. theory and method. Psychometrika 17, 401–419.

Ultsch, A., 1999. Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series. In: Oja, E., Kaski, S. (Eds.), Kohonen Maps. Elsevier Science, pp. 33–46.

Ultsch, A., 2003a. Maps for the visualization of high-dimensional data spaces. In: Proceedings of the Workshop on Self organizing Maps. Kyushu, Japan, pp. 225–230.

Ultsch, A., 2003b. U*-matrix: a tool to visualize clusters in high dimensional data. Tech. rep., Departement of Mathematics and Computer Science, Philipps-University Marburg.

Ultsch, A., Siemon, H. P., 1990. Kohonen's self-organizing feature maps for exploratory data analysis. In: Proceedings of the International Neural Network Conference (INNC'90). Kluwer, pp. 305–308.

Vesanto, J., 1999. SOM-based data visualization methods. Intelligent Data Analysis 3 (2), 111–126.

Vesanto, J., Ahola, J., 1999. Hunting for correlations in data using the self-organizing map. In: International ICSC Congress on Computational Intelligence Methods and Applications (CIMA'99). ICSC Academic Press, pp. 279–285.

Vesanto, J., Alhoniemi, E., 2000. Clustering of the self-organizing map. IEEE Transactions on Neural Networks 11 (3), 586–600.

Vesanto, J., Himberg, J., Alhoniemi, E., Parhankangas, J., 1999. Self-organizing map in matlab: the SOM toolbox. In: Proceedings of the Matlab DSP Conference 1999. pp. 35–40.

Yin, H., 2002. ViSOM - a novel method for multivariate data projection and structure visualisation. IEEE Trans. on Neural Networks 13 (1), 237–243.

Zangl, G., Hannerer, J., 2003. Data Mining: Applications in the Petroleum Industry. Round Oak Publishing.