A SOM-view of oilfield data: A novel vector field visualization for Self-Organizing Maps and its applications in the petroleum industry

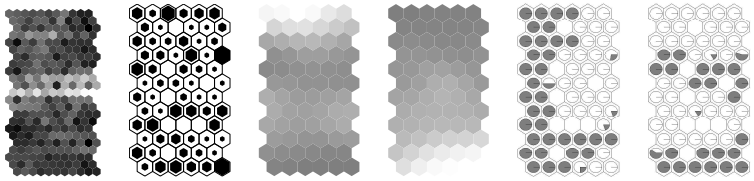
Georg Pölzlbauer, Andreas Rauber
(Department of Software Technology
Vienna University of Technology, Austria
{poelzlbauer, rauber}@ifs.tuwien.ac.at)
Michael Dittenbach
(eCommerce Competence Center – ec3
Donau-City-Str. 1, Vienna, Austria
michael.dittenbach@ec3.at)

Abstract: Self-Organizing Maps are a prominent tool for exploratory analysis and visualization of high-dimensional data. We propose a novel method for visualizing the cluster structure and coherent regions of the Self-Organizing Map that can be displayed as a vector field on top of the map lattice. Concepts of neighborhood and proximity on the map is exploited to obtain a representation where arrows point to the most similar region. The method is especially useful for large maps with a high number of map nodes. In our experiments, we visualize a data set that stems from applications in the petroleum industry, and show how to use our method to maximize the gas output.
Key Words: Self-Organizing Maps, Knowledge Visualization, Petroleum Industry

1 Introduction

Self-Organizing Maps (SOMs) [Kohonen 2001] are a valuable tool for exploratory data analysis and visualization, which map from a high-dimensional input space to a low-dimensional lattice, preserving the topology of the data set as faithfully as possible. It is a popular unsupervised neural network algorithm that has been used in a wide range of scientific and industrial applications. The projection can be visualized in numerous ways in order to reveal the characteristics of the input data or to analyze the quality of the obtained mapping. Our method is based on calculating distances from the SOM codebook with regard to the neighborhood kernel, introducing a concept of proximity on the map lattice. For each map unit, we compute a vector pointing in the direction of the most similar region in output space.

The remainder of this paper is organized as follows. In Section 2, we describe our data set, which comes from the domain of petroleum industry. Common SOM visualization techniques are discussed in Section 3. In Section 4, we outline the gradient field visualization. Section 5 provides experimental results. Finally, some conclusions are drawn in Section 6.



(a) (b) (c) (d) (e) (f)

Figure 1: 12×6 SOM: (a) U-Matrix, (b) Hit histogram, (c) Variable: "Produced Gas", (d) Variable "Stimulation Costs", (e) Pie Charts "Formation Type", (f) Pie Charts "Proppant Type"

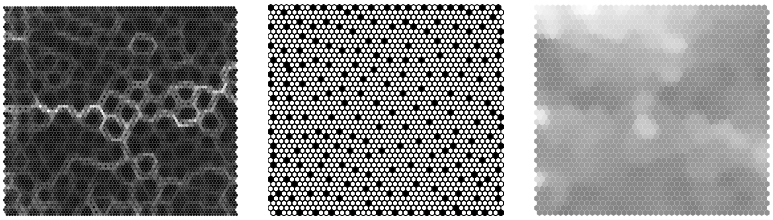
2 The Fracture Optimization Data Set

Data mining techniques have been applied in widely different domains. The petroleum industry with its mass of sensor data is one of the areas where SOMs can be used [Zangl et al 2003]. The aim is to investigate the characteristics of the data to find possibilities to optimize the gas production process.

The data set that we use in our experiments stems from an industrial project and is taken from measurements collected in 199 wells for producing gas. It consists of 10 variable dimensions and contains metric as well as categorical values. Two of the variables are target variables, "produced gas" and "stimulation costs", which are unknown prior to actually starting the costly production process. Although the SOM is an unsupervised algorithm, it can be used by treating the values to predict as missing values. Regression and machine learning techniques perform better in forecasting target values than the SOM, but the aim is to learn about the correlation of the individual variables, similar to canonical correlation analysis, and the cluster structure of the data set rather than predicting the expected costs and amount of gas. The relationship of the variables in the data set and its visualization is the primary goal of the method introduced in this paper. The wells are described by features including geologic data as well as parameters that determine the technical process of producing gas. The measurements are "proppant mass", "average rate", "average pressure", "pad fluid volume", "total fluid volume", "netpay", "formation type", "proppant type", "produced gas", and "stimulation costs". From an optimization perspective, one is most interested in obtaining high "produced gas" and low "stimulation costs".

3 SOM Visualization Techniques

In this section, we briefly describe visualization concepts for SOMs [Vesanto 2002] related to the gradient field method. The first map we use for our experiments



(a) (b) (c)

Figure 2: 44×44 SOM: (a) U-Matrix, (b) Hit histogram, (c) "Produced Gas"

consists of 12×6 map units, the second one of 44×44 units, where the 199 data points are outnumbered by the units, both trained on the data set described in the previous section. In Figure 1, some of the most common visualization techniques are demonstrated with the 12×6 SOM. The U-Matrix [Ultsch 1999] is shown in Figure 1(a), where the distances between prototype vectors from neighboring map units are calculated, revealing the local cluster structure of the map. It can be seen that the upper and lower parts of the SOM are clearly separated by the horizontal border indicated by bright colors (high U-Matrix values). Another method for displaying SOMs are component planes, each depicting a single variable of the prototype vectors. Figures 1(c)–(d) show the two most important component planes "Produced Gas" and "Stimulation Costs". The lower right corner can be identified as having highest costs, and the upper part as yielding the highest gas output; thus, the upper part of the map contains the most desirable wells. It has to be noted that these variables can not be known in advance, and it is therefor important to learn which factors lead to these beneficial outputs. Figure 1(b) shows the hit histogram, which reveals the distribution of the data samples on the map, where the size of the marker indicates the frequency of how many times the unit was selected best-matching unit by data samples. A variant of this especially usefull for depicting categorical variables is displaying hits as pie charts, counting how often each map unit is selected by samples of each category level. This is shown in Figures 1(e)–(f) for the categorical variables "Formation Type" and "Proppant Type". The map is clearly divided by both attributes, with "Formation Type" dividing vertically, and "Proppant Type" separating the two clusters identified by the U-Matrix horizontally. The visualizations indicate that Proppant Type shown as bright in the pie charts gives more desirable results.

In Figure 2, different visualizations for the 44×44 SOM are shown. When comparing the U-Matrix in Figure 2(a) and the hit histogram in Figure 2(b), it is obvious that the U-Matrix draws small borders around each unit that is occupied by a data sample. The method that we present in the next section

is similar to the U-Matrix in the way that local distances are calculated, but can be parameterized to negate the effect of sparse data population to identify more global cluster borders. Another approach to deal with this problem of local cluster borders overshadowing global ones is the U*-Matrix [Ultsch 2003]. It is computed by weighting the U-Matrix values with the data density in input space. For maps of this size, which may also be sparsely populated like so-called Emergent SOMs [Ultsch 1999], methods like the pie charts can not be applied. One of the aims of our work is to provide further methods of visualization for large SOMs. Component planes for the two most important components "produced gas" and "stimulation costs" are shown in Figures 2(c) and 3(d). It can be seen that due to the increased space on the map lattice, slightly different structures emerge when compared to the smaller SOM.

4 A Vector Field Visualization

In this section, we describe the gradient field visualization. It is displayed as a vector field on top of the map lattice, and aims at making the SOM readable for persons with engineering background who have experience with flow and gradient visualizations. The information communicated through our visualization is similar to the U-Matrix, identifying clusters and coherent areas on the map, but allowing for extending the neighborhood width, and thus showing global distances. Another goal is to make explicit the direction of the most similar cluster center, represented by arrows pointing to this center. The method turns out to be most useful for SOMs with high numbers of map units. One of the concepts that is required is the neighborhood kernel that is used by the SOM training algorithm. The commonly used Gaussian kernel has the well-known form of the Gaussian Bell-Shaped Curve, formally

$$h_\sigma(dist) = \exp\left(-\frac{dist^2}{2\sigma}\right) \tag{1}$$

where $dist$ is the distance and $sigma$ a width parameter.

The SOMs discussed in this paper have a two-dimensional output space (map lattice) with map units p_i, where index i is between 1 and M. The map units are attached to a model vector m_i (also codebook or prototype vector) which lives in the same space as the input samples, and has dimension N. The SOM algorithm creates an ordering of the model vectors, such that close map units correspond to similar prototype vectors. We explicitly denote p_i as the position vector in u, v-coordinates of the map unit that represents codebook vector m_i, with i connecting the input and output space representation. The u, v-coordinates of the map units are denoted as p_i^u and p_i^v, respectively. Distance can thus be measured both in input and output space, either between m_i and m_j or p_i and p_j, where $d_{input}(m_i, m_j)$ denotes the Euclidean distance between m_i and m_j in input

space, and $d_{output}(p_i, p_j)$ the distance on the map lattice, i.e. approximately the number of map nodes between p_i and p_j.

For each unit p_i we compute a two-dimensional vector a_i. We distinguish between u and v coordinates, denoted as a_i^u and a_i^v, along the horizontal and vertical axes of the map lattice. The gradient field method determines these vectors in two-steps: First, the computations for each map unit are performed separately for the positive and negative directions of axes u and v, and finally, these components are aggregated by obtaining the ratio between the positive and negative directions, and the lengths of the a_i are normalized, which can then be visualized.

First, we define $\alpha(p_i, p_j)$ as the angle between the direction of p_j seen from p_i on the map lattice and the u-axis. We can thus decompose the contribution of the neighborhood kernel into u and v coordinates given positions p_i, p_j:

$$w^u(p_i, p_j) = \cos(\alpha(p_i, p_j)) \cdot h_\sigma(d_{output}(p_i, p_j)) \tag{2}$$

$$w^v(p_i, p_j) = \sin(\alpha(p_i, p_j)) \cdot h_\sigma(d_{output}(p_i, p_j)) \tag{3}$$

w^u and w^v serve as a weighting factor for the rest of the computations. The neighborhood kernel relies on width parameter σ, which determines how far-away map units are weighted.

Next we split the dissimilarity that can be measured in input space into positive and negative direction for both axes. This is performed for each pair of map units p_i, p_j, formally

$$con_+^u(p_i, p_j) = \begin{cases} d_{input}(m_i, m_j) \cdot w^u(p_i, p_j) \text{ if } w^u(p_i, p_j) > 0 \\ 0 \qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \tag{4}$$

$$con_-^u(p_i, p_j) = \begin{cases} -d_{input}(m_i, m_j) \cdot w^u(p_i, p_j) \text{ if } w^u(p_i, p_j) < 0 \\ 0 \qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \tag{5}$$

Here, con_+^u denotes the contribution of map unit p_j's distance from p_i in positive direction along u, and con_-^u in negative direction. Note that similar calculations are performed for the U-Matrix, when considering only adjacent map units, and without the use of the neighborhood kernel. con_+^v and con_-^v are defined analogously. For example, a map unit p_j that lies to the lower right of p_i results in $con_-^u(p_i, p_j) = con_+^v(p_i, p_j) = 0$, and some positive values for $con_+^u(p_i, p_j)$ and $con_-^v(p_i, p_j)$ according how far these units lie apart on the SOM lattice and weighted by the neighborhood kernel, and also its distance in input space, which is directly measured by factor d_{input}.

Then we aggregate the positive and negative contributions for p_i

$$diss_u^+(p_i) = \sum_{j=1\ldots M, j\neq i} con_+^u(p_i, p_j) \tag{6}$$

$$diss_u^-(p_i) = \sum_{j=1\ldots M, j\neq i} con_-^u(p_i, p_j) \tag{7}$$

$diss_+^v$ and $diss_-^v$ follow analogously. $diss_+^u(p_i)$ indicates how much m_i is different from the region that lies to the right of it. In a gradient field analogy, this value shows how much it is repelled from this direction.

Next, another aggregation is performed for the total possible contribution in positive and negative directions according to the neighborhood functions

$$w_+^u(p_i) = \sum_{j=1\ldots M, j\neq i} \begin{cases} w^u(p_i, p_j) \text{ if } w^u(p_i, p_j) > 0 \\ 0 \qquad\qquad \text{otherwise} \end{cases} \tag{8}$$

$$w_-^u(p_i) = \sum_{j=1\ldots M, j\neq i} \begin{cases} -w^u(p_i, p_j) \text{ if } w^u(p_i, p_j) < 0 \\ 0 \qquad\qquad \text{otherwise} \end{cases} \tag{9}$$

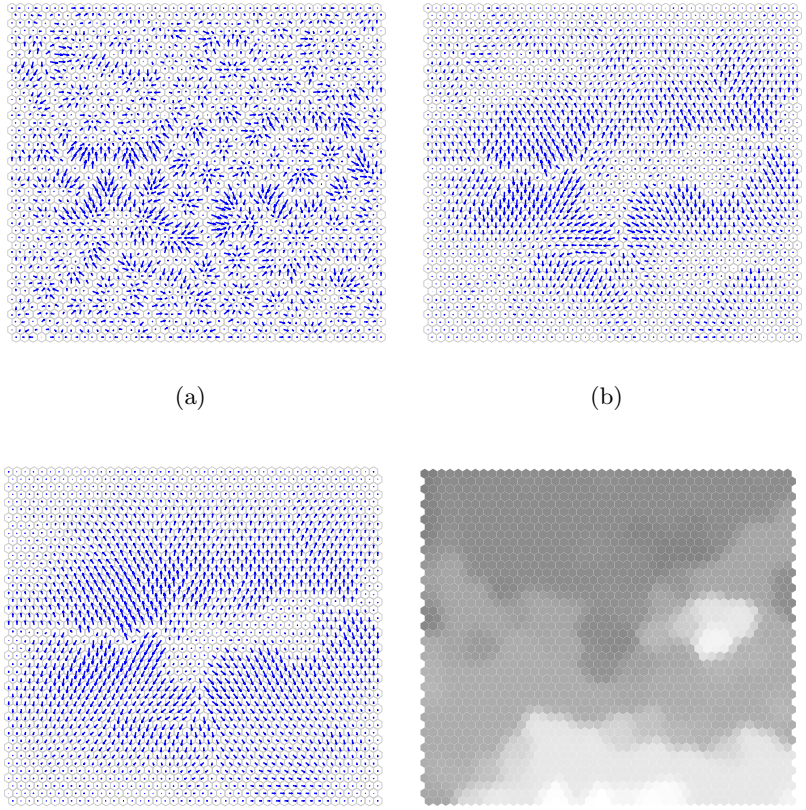
Finally, the coordinates of a_i can be computed as the ratio of positive and negative dissimilarities. The normalization by w_+^u counters the effect that units close to the borders of the SOM would have dissimilarities of zero in directions that actually point outside of the map, resulting in a strong bias and long arrows, which is not desired. The u component of the gradient vector a is computed as

$$a_i^u = \frac{diss_-^u(p_i) \cdot w_+^u(p_i) - diss_+^u(p_i) \cdot w_-^u(p_i)}{diss_+^u(p_i) + diss_-^u(p_i)} \tag{10}$$

and likewise for the v direction.

5 Examples

In this section, we give examples based on a the 44×44 SOM trained on the Fracture Optimization data. In another publication [Pölzlbauer et al 2005], we evaluate a standard machine learning data set which more densely populated. Figure 3(a) shows our visualization technique with a Gaussian kernel of $\sigma = 2$. If compared to the U-Matrix in Figure 2(a), it can be seen that the longest arrows are observed near the cluster borders, pointing to the interior of their cluster and away from these borders. It can be seen that adjacent units, for which the arrow points in different directions, are clearly along a cluster border. The length of the arrow indicate the sharpness of the border. Between these transitions, arrows sometimes have no distinguishable length. The corresponding prototype vectors are likely to be far away from either cluster, and are referred to as interpolating units, since they do not represent data vectors in a vector quantization sense, but are only a link connecting two distant data clouds. Cluster centers also have small dotlike arrows pointing in no distinguishable direction, but the difference is that the surrounding arrows are pointing in their direction, and not away from them as is the case with interpolating units. The effects of increasing σ can be



(a) (b)

(c) (d)

Figure 3: 44×44 SOM: Vector Field with (a) $sigma = 2$, (b) $sigma = 7$, (c) $sigma = 15$, (d) component plane "stimulation costs"

observed in Figures 3(b), (c) for values 7 and 15, respectively. Higher σ results in a smoothing effect, emphasizing the global structures over local ones, thus this parameter has to be selected depending on what the user is interested in. What can be learned from these plots is that the map is divided into two major regions on top and bottom, and there is a transition area on the center right, where very small arrows are located. The cluster centers are close to the top and bottom edges. Figure 3(d) shows the component plane for the "stimulation costs" dimension. The center right region corresponds to higher costs than in its neighboring areas, and when compared to the vector field visualization, this region is clearly identified as untypical, because it belongs to neither of the two big clusters on top and bottom of the map. While the "produced gas" component shows a more monotonic decrease on the map from top to bottom

in Figure 2(c), and the component plane for "stimulation costs" has a more irregular distribution, both can be aggregated into the vector field visualization that can be interpreted as having a coherent change of attributes from top to bottom, with an irregularity in the center right region. The vector field technique does not know which of the components are important or even beneficial, but wells located in either location of the map can be identified as belonging to a more or less typical region. In our case, the stimulation costs should be low and the produced gas high. Thus, wells on the upper edge of the map combine the best properties of low costs and high output, while the lower edge represents wells with high costs and mediocre output. There is a transition in between, and also a sharp border separating the two main regions, while the center right area holds undesirable wells with both high costs and very low output. The information about the coherency of the regions can not be obtained simply by looking on the two component planes presented here, but since the vector field method aggregates all the components, this visualization reveals the cluster structure, while not placing borders where transitions from high to lower values occur uniformly.

6 Conclusion

In this paper, we have introduced a novel method of displaying the cluster structure of Self-Organizing Maps. The gradient field method is distantly related to the U-Matrix. It is based on the neighborhood kernel function and on aggregation of distances in the proximity of each map node. It requires a parameter σ that determines the smoothness and the level of detail of the visualization. The direction of the most similar region is pointed to by an arrow. Our experiments have shown that this method is especially useful for maps with high numbers of units and is intended for use in engineering applications, because experts from these domains are generally accustomed to vector field plots. Usability studies of our technique are ongoing with experts from the petroleum industry.

References

[Kohonen 2001] Kohonen, T.: "Self-Organizing Maps"; 3rd edition, Springer, 2001

[Pölzlbauer et al 2005] Pölzlbauer, G., Dittenbach, M., Rauber, A.: "A visualization technique for Self-Organizing Maps with vector fields to obtain the cluster structure at desired levels of detail"; Intl. Joint Conf. on Neural Networks (IJCNN), 2005

[Ultsch 1999] Ultsch, A.: "Data Mining and Knowledge Discovery with Emergent Self-Organizing Feature Maps for Multivariate Time Series"; In "Kohonen Maps", Elsevier Science, 33–46, 1999

[Ultsch 2003] Ultsch, A.: "U*-Matrix: a Tool to visualize Clusters in high dimensional Data"; Technical Report, Philipps-University Marburg, 2003

[Vesanto 2002] Vesanto, J.: "Data Exploration Process Based on the Self-Organizing Map"; PhD Thesis, Helsinki University of Technology, 2002

[Zangl et al 2003] Zangl, G., Hannerer, J.: "Data Mining: Applications in the Petroleum Industry"; Round Oak Publishing, 2003