Large-Scale Data Exploration with the Hierarchically Growing Hyperbolic SOM

Jörg Ontrup, Helge Ritter

Bielefeld University, Faculty of Technology, Neuroinformatics Group
PO Box 100131, 33501 Bielefeld, Germany

Abstract

We introduce the Hierarchically Growing Hyperbolic Self-Organizing Map (H^2SOM) featuring two extensions of the HSOM (hyperbolic SOM): (i) a hierarchically growing variant that allows for incremental training with an automated adaptation of lattice size to achieve a prescribed quantization error and (ii) an approximate best match search that utilizes the special structure of the hyperbolic lattice to achieve a tremendous speed-up for large map sizes. Using the MNIST and the Reuters-21578 database as benchmark datasets, we show that the H^2SOM yields a highly efficient visualization algorithm that combines the virtues of the SOM with extremely rapid training and low quantization and classification errors.

Key words: Hyperbolic Self-organizing maps, Growing network, Hierarchical Clustering, Text Mining

1 Introduction

The rapid pace of technological advances has led to a continuously growing volume of large data sets. The Self-Organizing Maps as introduced by Kohonen (1982, 2001) have become a standard tool for the exploratory analysis of such data and have been extensively used for visualization purposes. As a result, there have been major efforts to overcome the problem of a strong rise in the required computational resources for training large maps that utilize a large number of nodes to offer a high resolution. Several approaches have been suggested to address this problem. Koikkalainen and Oja (1990) proposed the Tree-Structured Self-Organizing Map (TS-SOM), which consists of a fixed number of SOMs arranged in a pyramidal structure. The training of

Email address: jontrup@techfak.uni-bielefeld.de (Jörg Ontrup).

the pyramid is computed level-wise where the best match search is performed as a tree search reducing the complexity to $\mathcal{O}(log\ N)$. A Growing Hierarchical SOM (GHSOM) has been proposed by Rauber et al. (2002). Their approach combines individually growing SOMs with a hierarchical architecture and has successfully been applied to the organization of document collections and music repositories. Lately, Pakkanen et al. (2004) have described the Evolving Tree, which is constructed as a freely growing network utilizing the shortest path between two nodes in a tree as the neighborhood function for the self-organizing process. All of these approaches achieve a favorable computational complexity. However, the visualization of the learned hierarchies remains a demanding task. Either a map metaphor is not applicable anymore, or the transition between maps within or across the hierarchies introduces discontinuities making it hard to visualize and maintain the surrounding context. Thus, without guidance the user might be easily lost within the tree structure. Lamping and Rao (1994) discovered that hyperbolic space is ideally suited to embed large hierarchical structures. Their discovery motivated the introduction of the hyperbolic SOM (HSOM) (Ritter, 1999). By employing a lattice with a hyperbolic grid topology, it combines the virtues of the SOM and hyperbolic spaces for adaptive data-visualization. However, due to the exponential growth of its hyperbolic lattice, it also exacerbated the need for addressing the scaling problem of SOMs comprising very large numbers of nodes. In this contribution we show that a solution can be achieved by a very natural extension of the HSOM to a *Hierarchically Growing Hyperbolic SOM* (H^2SOM). It combines the virtues of hierarchical data organization, adaptive growing to a required granularity, good scaling behaviour and smooth, map-based browsing, thereby bringing together several strengths of separate, previous approaches within a single, uniform architecture.

2 Hyperbolic Geometry

Most of our spatiotemporal thinking is deeply rooted in the world of Euclidean geometry following Euclid's five axioms. However, hyperbolic space offers a completely consistent non-Euclidean geometry that is characterized by being negatively "curved". Standard textbooks on Riemannian geometry (Coxeter, 1957; Morgan, 1993) show that the relationships for the area A and circumference C for a circle of radius r are then given by $A(r) = 4\pi \sinh^2(r/2)$ and $C(r) = 2\pi \sinh(r)$, respectively. This bears two remarkable asymptotic properties: *(i)* for small radius r the space "looks flat" since $A(r) \approx \pi r^2$ and $C(r) \approx 2\pi r$. *(ii)* For larger r both A and C grow asymptotically *exponentially* with the radius.

Naturally, there exists no isometric embedding of $\mathbb{H}^2$ into $\mathbb{R}^2$, since a projection of the negatively curved space into flat space introduces distortions in

either length, area or angle. However, a locally isometric embedding into $\mathbb{R}^3$ is possible: we obtain a "wrinkled" structure, which resembles a saddle at every point of the surface. Sometimes, Nature approximated the growth behaviour of a hyperbolic surface, e.g. in some corals that need to maximize their contact area with the surrounding water that carries vital nutrients. In Fig. 1 it can be seen, that this is leading to structures resembling a 3-dimensional local embedding (of a patch) of the hyperbolic plane remarkably well. Note, that such a corrugated structure is also found in the human cerebral cortex which is comparatively thin (about 2-4 mm), but if laid out flat, covers about 2,500 cm^2.

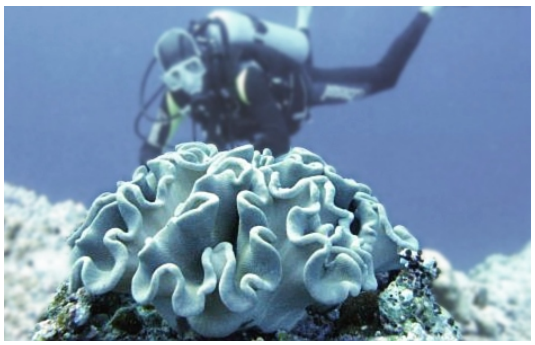


Fig. 1. A local embedding of $\mathbb{H}^2$ in $\mathbb{R}^3$ looks very similar to such a leather-coral for which nature found a solution to maximize its contact area in order to absorb vital nutrients from the surrounding water. (Photograph by courtesy of H. Toperczer.)

The geometric properties discussed above make the hyperbolic space an ideal candidate for embedding large hierarchical structures (Lamping and Rao, 1994; Munzner, 1998). For its display on a flat 2D screen one may choose the projection of $\mathbb{H}^2$ on the Poincaré Disk (Coxeter, 1957) that has a number of convenient beneficial properties for visualization: First, it is locally shape preserving, with a strong "fish-eye" effect: The origin of $\mathbb{H}^2$ - corresponding to the "fish-eye" fovea - is mapped almost faithfully, while distant regions become exponentially "squeezed". Second, the model allows to translate the original $\mathbb{H}^2$ in a very elegant way using so-called *Möbius transformations*. By describing the Poincaré Disk *PD* as the unit circle in the complex plane, the isometric Möbius transformation $T(z)$ for a point $z \in PD$ can be written as

$$T(c,\theta)(z) = e^{i\theta}\frac{z-c}{1-\bar{c}z}, \quad \|c\| < 1, \tag{1}$$

where the angle θ describes a pure rotation of the *PD* around the origin and c is a complex number specifying the mapping of the origin to $-c$ (with c becoming the new center of the *PD*). Consequently, the fovea can be moved to any other part of the infinite hyperbolic plane. This enables the user to selectively focus on interesting portions of a map painted on $\mathbb{H}^2$ while still keeping a coarser view of its surrounding context. For further details on the construction of the Poincaré Disk, see e.g. Ritter (1999).

3 Hierarchically Growing Hyperbolic Maps

3.1 Growing Network Topology

The core idea of the hierarchically growing Hyperbolic Self-Organizing Map is to employ the same sort of lattice structure already used for the plain HSOM and its applications (Ritter, 1999; Ontrup and Ritter, 2001; Walter et al., 2003).

1) Initialization: We start with the root node of the hierarchy placed at the origin of $\mathbb{H}^2$. Then the coordinates of the n_b nodes of the first sub hierarchy are placed at the vertices of n_b equilateral triangles as shown in Fig. 2(a). Since the sum of the angles in a hyperbolic triangle is always less than π, the angle α of an equilateral hyperbolic triangles has to obey $3\alpha < \pi$. Additionally the nodes of the first sub hierarchy must cover a full circle in $\mathbb{H}^2$ (c.f. Fig. 2(a)), hence $\alpha = 2\pi/n_b$ holds. When combining the two conditions we see that we need a branching factor of $n_b > 6$ for the tessellation scheme. Note, that there exists no upper bound for n_b and therefore the number of children a node can have. Since the side length l of the triangles in the Poincaré Disk is given by $l = (1 - 4\sin^2(\pi/n_b))^{1/2}$, the branching factor n_b also determines how "fast" the network is reaching out into hyperbolic space.

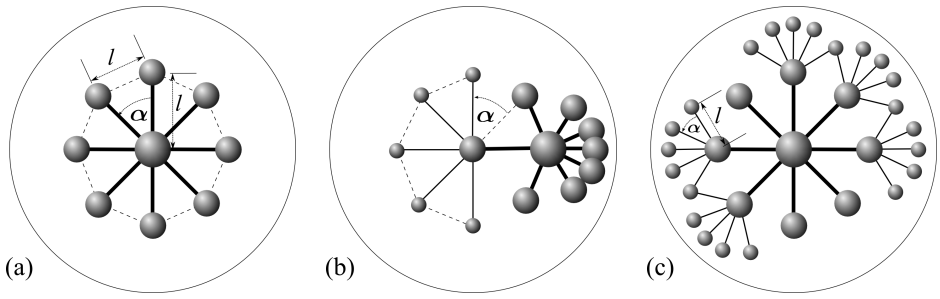


Fig. 2. Topology of the H^{2}SOM. (a) The nodes at the vertices of - in this case $n_b = 8$ - equilateral triangles form the first level in the hierarchy of the H^{2}SOM. (b) Each node can be translated into the center of the *PD* and expanded with $n_b - 3$ children nodes by applying a set of appropriate Möbius transformations. (c) By iteratively expanding the nodes, the networks grows towards the perimeter of the *PD*.

2) Growing Step: We can expand each node in the periphery of the existing network by surrounding it with $n_b - 3$ children nodes (there are already two sibling and one parent node present at this stage). Algorithmically this can be done by applying a Möbius transformation such that the to be expanded node now resides in the center of the *PD*. As an example, in Fig. 2(b) the leftmost node of Fig. 2(a) was translated to the center of the map (for illustration purposes the coordinates of the other nodes were translated accordingly). The coordinates of the children nodes are then obtained by iteratively applying

the Möbius transformation $T(z; c, \theta)$ with $c = 0$ and $\theta = \cos(\alpha) + i \sin(\alpha)$ to one of the sibling nodes as indicated in Fig. 2(b).

3.2 Learning and Growth Criterion

The training of the hierarchical network largely follows the traditional SOM approach. To each node a a reference vector $\mathbf{w}_a$ is attached, projecting into the input data space X. In addition, it will be convenient to attach to each node also its 2D position $\mathbf{z}_a \in \mathbb{C}$ in the complex Poincaré Disk $|\mathbf{z}| \leq 1$. The center node is initialized with the center of mass of the training data and does not take part in the training process. Its prototype vector stays fixed. The n_b nodes of the first sub hierarchy are initialized with small deviations from the center prototype and are trained in the usual way: After finding the best match neuron a^*, i.e. the node which has its prototype vector $\mathbf{w}_a$ closest to the given input $\mathbf{x}$, $a^* = \text{argmin}_a \|\mathbf{w}_a - \mathbf{x}\|$ all reference vectors are updated by the well known adaptation rule

$$\Delta \mathbf{w}_a = \epsilon(t)\ h(a, a^*)\ (\mathbf{x} - \mathbf{w}_a), \quad \text{with} \quad h(a, a^*) = \exp\left(-\frac{d^2_{a,a^*}}{\sigma(t)^2}\right) \tag{2}$$

Here $h(a, a^*)$ is a bell shaped Gaussian centered at the winner a^* and decaying with increasing distance d_{a,a^*} of the neurons. We can then compute the hyperbolic node distances d_{a,a^*} conveniently from their associated positions $\mathbf{z}_a$ in the Poincaré Disk (Coxeter, 1957):

$$d_{a,a^*} = 2\ \text{arctanh}\left(\frac{|\mathbf{z}_a - \mathbf{z}_{a^*}|}{|1 - \mathbf{z}_a \bar{\mathbf{z}}_{a^*}|}\right). \tag{3}$$

During the course of learning, the width $\sigma(t)$ of the neighborhood bell function and the learning step size $\epsilon(t)$ are continuously decreased in order to allow more and more specialization and fine tuning of the then increasingly weakly coupled neurons - just as in the standard SOM approach.

After fixed training intervals we repeatedly evaluate for each node an expansion criterion. In our experiments we have so far used the node's quantization error as the growth criterion. If a given threshold Θ_{QE} for a node is exceeded, that node is expanded as described in step 2 above and illustrated in Fig. 2(b). After the expansion step where all nodes meeting the growth criterion were expanded, all reference vectors from the previous hierarchies become fixed and adaptation "moves" to the nodes of the new structural level.

3.3 Fast Best Match Tree Search

The most time consuming step in a standard SOM is the global search for the best match unit. The peculiar, intrinsically "uniformly hierarchical" structure of the hyperbolic grid offers an intriguing possibility to significantly accelerate this most time-consuming step: we can approximate the global search for the winner unit a^* by *a fast tree search*, taking as the search root the initial center node of the growth process and following then the "natural" hierarchical structure in the hyperbolic grid: starting from this node, we recursively determine the k best-matching nodes among its n_b neighbors until we reach the periphery. For $k = 1$, this will generate a path with $\mathcal{O}(\log_{n_b} N)$ comparisons, instead of $\mathcal{O}(N)$ for a global search. For $1 < k \leq n_b$ we asymptotically must search $\mathcal{O}(N^p)$ nodes, with exponent $p = \log_{n_b} k \leq 1$ (restituting a full search with $p = 1$ for $k = n_b$). Fig. 3 shows, for $n_b = 10$, that the resulting scaling behaviour permits speed-ups of several orders of magnitude, as compared with a global (standard SOM) search.

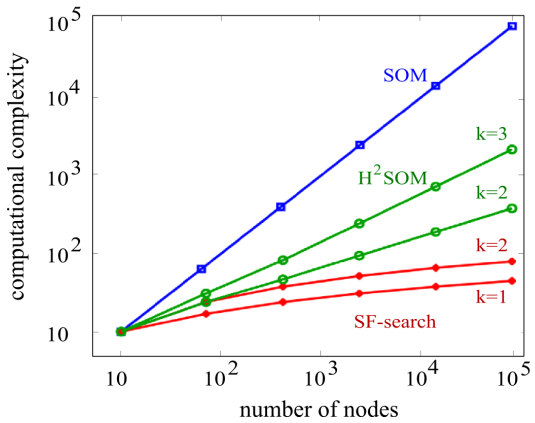


Fig. 3. Computational complexity for the best match search in SOM vs. H^{2}SOM (with a branching factor of $n_b = 10$).

Both, the geometry of the hyperbolic lattice, together with the hierarchical growing scheme, tend to organize the prototype vectors $\mathbf{w}_a$ in such a manner that the above search scheme provides a very good approximation to global search. In fact, our experiments indicate that we may even truncate the tree branching factor to $k = 1$ for all search steps beyond the innermost ring, leading to a "super-fast" search scheme ("SF-search") scaling as $\mathcal{O}(k \cdot \log_{n_b} N)$ (lower curves in Fig. 3). For instance, in the test problem reported in Table 1 we found that for $k = 2$ ($k = 1$) SF-search led to the correct best match unit or the very vicinity of it in 92% (65%) of all cases, leading to maps that were on par with or outperformed Euclidean SOMs constructed with global search.

The distinctive difference of the H^{2}SOM to other hierarchical SOM variants such as the Tree-Structured SOM (TS-SOM) (Koikkalainen and Oja, 1990), the Hierarchical SOM (Rauber et al., 2002), the Self-Organizing Tree Algorithm (SOTA) (Herrero et al., 2001), the Adaptive Topological Tree Structure (ATTS) (Freeman and Yin, 2004) or the Evolving Tree by Pakkanen et al. (2004) is that the complete hierarchy is embedded within a continuous, browsable space. When selecting a deeper level within the hierarchy the user does not need to carry out a discrete "jump", where the surrounding context might be lost, but instead can traverse the complete hierarchy in a smooth way. We believe that this is a very important property for visualization and have developed a framework using the open source visualization library VTK [1] to display a 3D scene where the user can interact with the Poincaré Disk in two ways: *(i)* The disk can be "grabbed" with the mouse and freely moved in 3D space, such that a suitable viewpoint might be chosen. *(ii)* The user can click on any arbitrary point z_0 on the Poincaré Disk and drag it to a new position z_1. The corresponding Möbius transformation for this mapping is given by $T(c, \theta)(z_0) = z_1$. With Eq. (1) and a rotation angle of $\theta = 0$, this results in c given by

$$c = \frac{z_0(\|z_1\|^2 - 1) - z_1(\|z_0\|^2 - 1)}{\|z_0\|^2\|z_1\|^2 - 1} \tag{4}$$

By evaluating mouse events during a drag operation, we continuously solve Eq. (4) and apply the corresponding Möbius transformation to all visible nodes on the *PD*. Consequently, the focus on the map can be moved in a continuous way, providing a means to smoothly navigate through the hierarchical space spawned by the H^{2}SOM nodes.

Additionally, for each node the visualization framework allows the display of different graphical attributes such as 3D glyph type, color, size, texture, or text labels which are dynamically adjusted in size with respect to their distance to the origin. A GUI allows the user to select features such as the number of data items mapped to a node, assigned class labels, average distance of prototype vectors to those of neighboring nodes or the variance of data items in the node's Voronoi cells to these graphical attributes. The overall architecture is based on a client-server approach with all data items stored in a SQL database. This allows for an elegant "drill down" where a mouse click on a H^{2}SOM node selects all corresponding data items in the database which then provides views on this data. After discussing numerical benchmarks in the next section, we will give some examples of this visualization approach.

[1] `http://public.kitware.com/VTK/`

4 Benchmarking the H^{2}SOM

We have chosen two datasets to benchmark and compare the H^{2}SOM to the standard SOM approach: the MNIST database of handwritten digits and the Reuters-21578 corpus of newswire articles. Both datasets feature a large collection of high-dimensional patterns and carry additional labels which makes them good candidates for benchmarks in a classification scenario.

4.1 The MNIST database

The MNIST database [2] consists of 60.000 training samples from approximately 250 writers and 10.000 test samples from a disjoint set of 250 other writers. We used the original 784-dimensional dataset which resembles 28x28 pixel grey level images of the handwritten digits. Since we used the scalar product as our data metric, all samples were normalized to unit length.
We have trained four standard SOMs of the sizes 7x7, 13x13, 25x25 and 48x48 with 49, 169, 625 and 2304 nodes, respectively. In comparison we have trained five H^{2}SOMs with a branching factor of $n_b = 8$. As a termination criterion we used a combination of maximal depth and quantization error: The growth process was stopped when either a predetermined hierarchy level was reached (in our case 2, 3, 4, 5 or 6 rings with maximal 41, 161, 609, 2281 or 8521 nodes, respectively), or a node's quantization error was less than a third of its parent's quantization error. In all cases 600.000 training steps were performed and the given results were averaged over 10 runs (except for the large SOM which was just trained twice due to the long computing times).

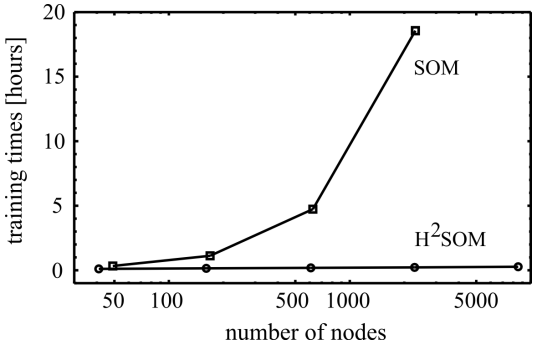


Fig. 4. Training times for different sized SOMs and H^{2}SOMs for the MNIST database. Note, that the abscissa is drawn with a logarithmic scale.

Fig. 4 shows the training times for computing the maps. From the graph the favourable scaling behaviour of the fast best match search in the H^{2}SOM becomes evident: even very large maps are trained within a few minutes, while standard SOMs quickly take several hours to complete.

[2] `http://yann.lecun.com/exdb/mnist/`

Table 1
Comparison of the H^{2}SOM to similar sized standard SOMs. The table shows the training times in hours and minutes for the map formation of the 60000 training samples and the seconds for the best match lookups for the 10000 test samples of the MNIST database. For the H^{2}SOM the test runs were performed with (a) the rapid SF-search with $k = 2$ and (b) a slower global search. (All results were obtained on a standard laptop with 1.5 GHz Pentium-M processor).

	SOM		H^{2}SOM, $n_b = 8$				
	13x13	48x48	3 rings		5 rings		6 rings
nodes	**169**	**2304**	**161**		**2281**		**8521**
QE	0.2094	0.1510	0.1993		0.1441		0.1175
t_{train}	1:07h	18:34h	0:09h		0:13h		0:16h
t_{test}	7.8s	181s	(a) 1.8s	(b) 8.4s	(a) 3.0s	(b) 101s	(b) 514s
Class	classification performance [%]						
0	93.9	98.3	96.0	**98.1**	**98.3**	**99.2**	99.5
1	**98.3**	**98.6**	**98.3**	98.1	98.5	98.5	99.1
2	86.6	**94.6**	**89.1**	**92.4**	92.4	93.1	94.7
3	**80.2**	91.3	76.1	79.5	90.0	**92.7**	94.6
4	69.0	88.3	**73.2**	**76.3**	**90.4**	**93.6**	94.4
5	66.9	90.0	**83.5**	**89.1**	87.4	**92.5**	93.1
6	**93.9**	**97.1**	89.7	92.7	96.0	96.3	97.7
7	81.2	91.0	**81.7**	**85.9**	**91.4**	**92.8**	93.9
8	**76.4**	88.8	59.1	67.6	88.1	**90.9**	91.8
9	**59.8**	**88.0**	55.9	57.8	**88.3**	87.7	90.7
total	81.0	92.7	80.5	**85.3**	92.2	**94.4**	95.8

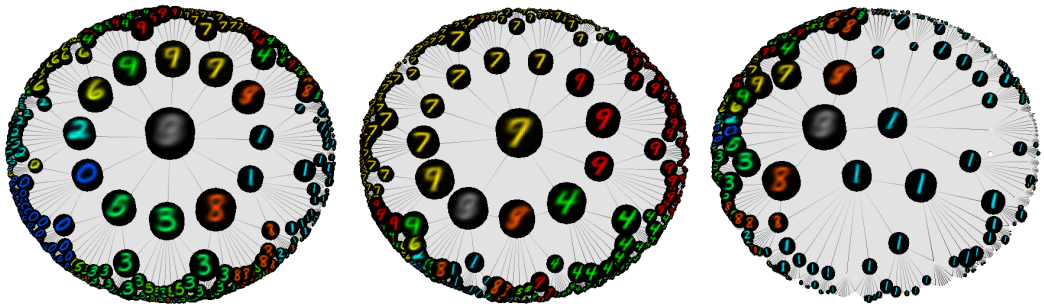
We additionally applied the SOMs as a classification tool for classifying the handwritten digits of the MNIST test dataset. To this end, the labeled training data is mapped to the SOM and all nodes are labeled with the most frequent label of the training items mapped to it. If there is no training item mapped to a node, i.e. the node is a interpolating node, it is labeled according the majority of votes from the neighborhood on the lattice grid. To each test item then the class label of its corresponding best match node is assigned.

In Table 1 the classification accuracies for different SOMs are given. Again, the most prominent difference is the time needed for the training of the networks. Due to the high data dimensionality ($d = 784$) the large SOM took more than 18 hours to compute, while the large H^{2}SOM using the "super-fast" SF-search was finalized in only 16 minutes. Despite using a full search for the SOM during training, the H^{2}SOM achieves a better mean quantization error. When using the SOMs as a classification tool, we used (a) the SF-search with $k = 2$, and (b) a slower global search to find the best match nodes for the 10.000 test samples. In the first case, the overall performance of the SOM is 0.5% better, though for half of the classes the H^{2}SOM achieves the same or better results. When using the slower global search only for retrieval *after*

the fast training of the H^{2}SOMs, the classification performances for the latter become considerably better and now clearly outperform the SOM. The last column shows the results for a large H^{2}SOM with 8521 nodes (it does not have a SOM counterpart, since it would have taken too long to compute). In terms of quantization error and classification accuracy, the results for this very large H^{2}SOM are superior without investing significantly more time in training the network.

Visualizing the MNIST database

Turning to the visualization capabilities of the H^{2}SOM we show in Fig. 5 a H^{2}SOM with a branching factor of $n_b = 12$. In (a) the Poincaré Disk is shown in a centered view, such that the top-level structure of the dataset is visible as the innermost ring of nodes. The prototype vectors are overlayed as textures on the node's glyphs. The colors are just a visual hint to indicate the class to which the majority of training samples belong to in the corresponding region of the map.



(a) (b) (c)

Fig. 5. Screenshots from different focus positions in the MNIST database. (a) shows the overall coarse structure of the dataset is shown, in (b) the user moved the focus to the "7" node from the 1 o'clock position in (a). In (c) the focus of attention was moved to the area covering the "1". Here several nodes were not expanded, because the low variation of the data resulted in low quantization errors of the nodes.

The H^{2}SOM can be seen to have learned the following top level structure from the data: The upper three nodes resemble mixtures between "4"s, "9"s and "7"s. Clockwise follows a node with a prototype looking like a blurred slanted "9", then two different orientated "1"s follow. At the bottom, three prototypes similar to an "8", "3" and "5" are shown, and then an articulated "0", "2" and "6" appear. In Fig. 5(b) the user has moved the focus towards the one o'clock node which is now centered. Here it can be seen, that at this next structural level the data splits up into equally slanted "7"s at the top, "9"s to the right and "4"s at the bottom right of the map.

5 Text Mining with the H^{2}SOM

Building on ideas how to use SOMs to semantically organize textual data (Ritter and Kohonen, 1989) the pioneering work on the WebSOM project (Lagus et al., 1996; Kaski et al., 1998; Kohonen et al., 2000) has amply demonstrated the strengths of the self-organizing map principle as a valuable interactive exploration tool to analyze large amounts of unstructured text corpora. In earlier work we have reported results obtained with the hyperbolic SOM (Ontrup and Ritter, 2001; Walter, 2003). Skupin (2002) has produced aesthetically very pleasing maps motivated by geographic metaphors. There has also been work on hierarchical variants of SOM (Merkl, 1997; Rauber et al., 2002; Freeman and Yin, 2004) which addressed the issue of computational complexity and advanced user interfaces. However, to our knowledge there has been so far no approach achieving a hierarchical self-organization in combination with smooth map-like browsing.

5.1 The Reuters-21578 Corpus

We here mainly report results on the Reuters-21578 [3] corpus of newswire articles from 1987 which has become a standard benchmark in text mining applications (Joachims, 1998; Yang, 1999; Sebastiani et al., 2000; Hotho et al., 2003).
There has been extensive work on different document representations, feature selection or term weighting approaches. For simplicity we here follow the widely used vector-space-model in Information Retrieval – commonly referred to as the *bag of words* model and first build a set of distinct terms $\{t_i\}$ for the text corpus. After word stemming and stop word removal we arrive at a vocabulary of unique word stems $\{w_i\}$. For each document d we then construct a feature vector $\vec{f_d}$, where the components w_i are determined by the frequency of which word stem w_i occurs in that document. Following standard practice (Salton and Buckley, 1988) we choose a *term frequency* $\times$ *inverse document frequency* weighting scheme. Distances and therewith dissimilarities of two documents are computed with the cosine metric

$$d(i,j) = 1 - \cos(\vec{f}_{d_i}, \vec{f}_{d_j}) = 1 - \vec{f}'_{d_i}\vec{f}'_{d_j}, \text{ with } \vec{f}' = \frac{\vec{f}}{||\vec{f}||} \tag{5}$$

and efficiently implemented by storing the normalized document feature vectors $\vec{f}'$.
In case of the Reuters-21578 collection our training set (obtained from the

[3] `http://www.daviddlewis.com/resources/testcollections/reuters21578/`

'ModApte" split into 9603 training and 3299 test documents) contained 5093 distinct terms after preprocessing, word stemming and stop word removal.

5.2 Performance Measures - Map Quality

For exploratory data analysis tasks where the self-organizing map is used as a tool to display similarity relationships from a high-dimensional input space on a low dimensional mapping space, the quality of this mapping is essential. There have been several proposals for such a quality measure, see e.g. Goodhill and Sejnowski (1997) for an extensive overview. We here report results on the approach of Bezdek and Pal (1995) based on Spearman's rank correlation coefficient to measure the degree of topology preservation. It is based on the preservation of the rank order of all pairwise distances,

$$\rho = \frac{\sum_i (R_i - \overline{R})(S_i - \overline{S})}{\sqrt{\sum_i (R_i - \overline{R})^2} \sqrt{\sum_i (S_i - \overline{S})^2}} \tag{6}$$

where the vectors R and S denote the ranks in the high-dimensional input space and in the low-dimensional map space, respectively. The value of ρ is limited to the range $[-1, 1]$, where a value of $\rho = 1$ corresponds to a "metric topology preserving" transformation (Bezdek and Pal, 1995), which describes a perfect mapping. As Table 2 indicates, the H^{2}SOM achieves a lower quantization error and a better global rank correlation than a SOM of comparable size (in the experiment, the threshold Θ_{QE} for the node expansion was set to zero, but growing was limited to a depth of 5 rings). Note, that the training times differ by a factor of ≈ 60, i.e. several minutes vs. several hours.

Table 2
Comparison of SOM and H^{2}SOM for the single performance measures training time, quantization error and Spearman's rho.

	t_{train}	QE	ρ
SOM (48x48; 2304 nodes)	13h 25min	0.3586	0.1829
H^{2}SOM (8,5; 2281 nodes)	**13min 49s**	**0.3484**	**0.2688**

Spearman's rho measures the overall global mapping quality of the SOM. However, for an interactive visualization framework where the user explores the data on a map, a local measure quantifying the goodness of a local patch on the map might be more meaningful. Venna and Kaski (2001) point out that any multi-dimensional scaling method introduces two kinds of errors when considering local neighborhoods in the input or map space: *(i)* Data items within an ϵ-neighborhood in the map space might actually come from distant regions in the input space; and *(ii)* data items within an ϵ-neighborhood in the input space might be mapped to distant regions in the map space. The first type of error might mislead a user to accept similarities in patterns which

in fact are not present in the data, while the second type introduces discontinuities resulting in the loss of original data relationships within the mapping. Venna and Kaski (2001, 2005) propose the two measures of *trustworthiness* and *continuity* to quantify the two errors described above. They are defined as

$$T(k) = 1 - S \sum_{i=1}^{N} \sum_{j \in \tilde{X}_k(i)} (r_{\mathcal{X}}(i,j) - k) \tag{7}$$

and

$$C(k) = 1 - S \sum_{i=1}^{N} \sum_{j \in \tilde{M}_k(i)} (r_{\mathcal{M}}(i,j) - k) \tag{8}$$

where N is the number of data items, $\tilde{X}_k(i)$ is the set of items within a neighborhood of k samples around data item i in the map space $\mathcal{M}$, but *not* in the input space $\mathcal{X}$; and $r_{\mathcal{X}}(i,j)$ is the rank of item j in the ordered list of distances to item i given by their distance in the input space. $\tilde{M}_k(i)$ and $r_{\mathcal{M}}(i,j)$ are defined accordingly with the role of $\mathcal{X}$ and $\mathcal{M}$ reversed. S is a normalization factor scaling the results of $T(k)$ and $C(k)$ between zero and one.

Fig. 6 shows the trustworthiness and continuity for both map types. For very small neighborhoods the standard SOM achieves a higher trustworthiness, for larger neighborhoods however, the H^{2}SOM performs persistently better.

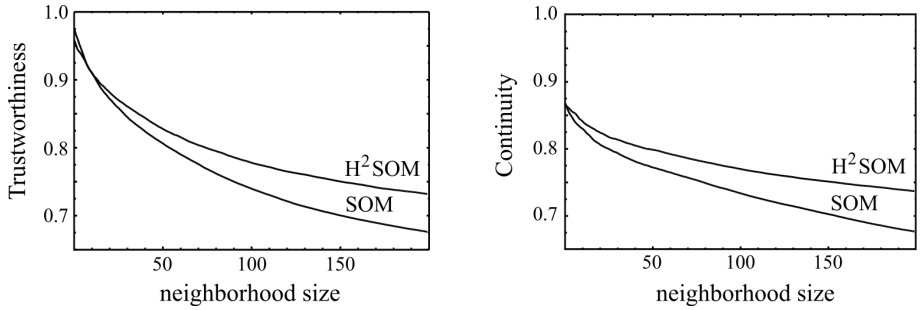


Fig. 6. Trustworthiness and continuity of SOM vs. H^{2}SOM for the Reuters-21578 text corpus as functions of the neighborhood size.

5.3 Performance Measures - Precision/Recall

In the context of document clustering the ability of an algorithm to classify a document into one or several categories is of high interest to the user. In classical information retrieval this ability is usually measured in terms of *precision* and *recall*, defined as the fraction of correctly classified documents, and the fraction of relevant documents from a retrieval set, respectively (Baeza-Yates and Ribeiro-Neto, 1999). Ideally, a system should achieve a high precision at high recall levels, but naturally there exists a trade-off between both: as the recall rises, precision tends to get lower. In order to compute precision-recall curves for the SOMs, we use the following rank function which sorts all

documents in a retrieval set:

$$r(C, d_i) = (\delta_{C,C^*} + 0.1) \cdot N_C^* \cdot d(w^*, d_i), \tag{9}$$

where C is the to be retrieved class, d_i the document, δ_{C,C^*} the Kronecker delta, C^* the class label assigned to the best match node of d_i, N_C^* the number of training documents with label C mapped to the best match node of d_i and $d(w^*, d_i)$ the distance between best match prototype and the feature vector of d_i.

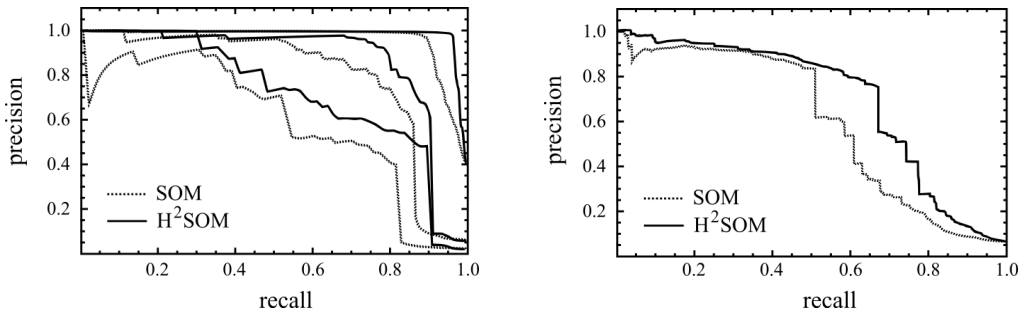


Fig. 7. Precision over recall for the Reuters-21578 data. In (a) the curves for the categories "earn", "grain" and "wheat" are shown (from top-right to bottom-left), in (b) the micro-averaged results over all categories.

The three exemplary precision-recall curves for the Reuters categories "earn", "grain" and "wheat" in Fig. 7(a) show that the H^{2}SOM consistently achieves higher precision levels than the standard SOM. This is also reflected by the micro-averaged results over all categories in Fig. 7(b), showing that the H^{2}SOM is able to "keep up" a higher precision for longer time.
Table 3 shows the maximal achievable F1-measures for all categories (micro- and macro-averaged), as well as for the two most and three least frequent categories from the top 20 topics in the Reuters corpus. It is defined as the harmonic mean of precision and recall (Baeza-Yates and Ribeiro-Neto, 1999) and yields values in the interval $[0, 1]$, with $F1 = 0$ when no relevant documents are found, and $F1 = 1$ when all documents from a given class are retrieved with no errors.

Table 3
F1-measures for micro- and macro-averaging over all categories as well as for the two most frequent and three least frequent categories.

	$F1_{\text{micro/macro}}$	$F1_{\text{earn}}$	$F1_{\text{acq}}$	$F1_{\text{gold}}$	$F1_{\text{nat-gas}}$	$F1_{\text{soybean}}$
SOM	0.628/0.633	0.933	0.854	0.706	0.473	**0.426**
H^{2}SOM	**0.705/0.674**	**0.974**	**0.938**	**0.830**	**0.591**	0.382

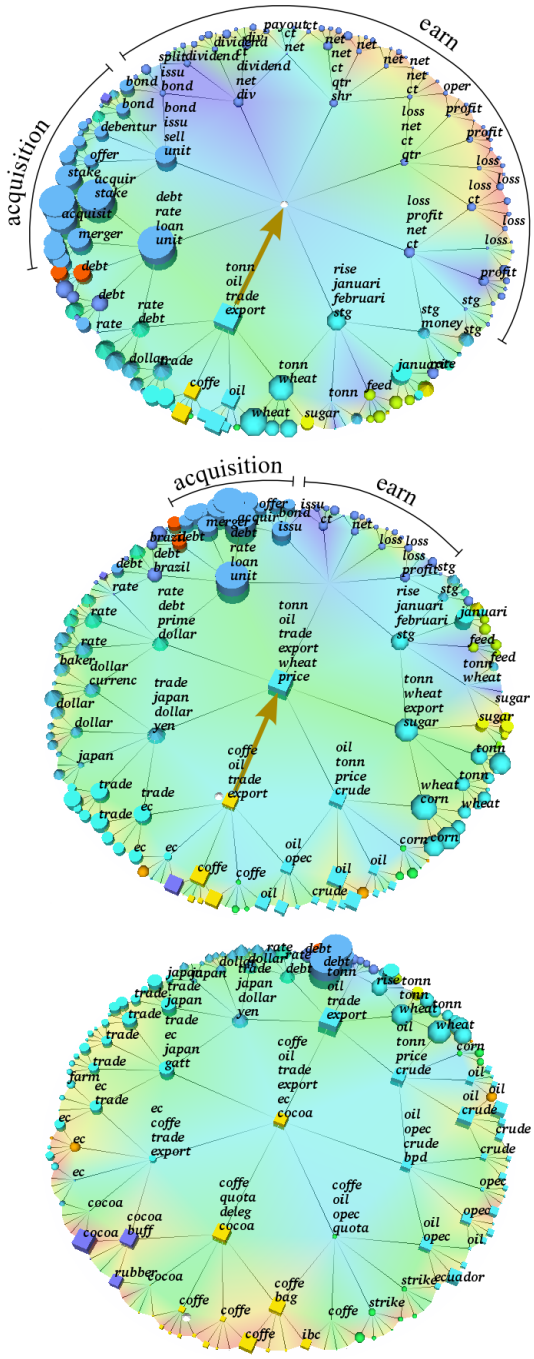


Fig. 8. The Reuters-21578 collection.

Fig. 8 shows the application of our visualization framework to the Reuters corpus. Category labels are reflected by different node symbols and nodes are annotated with key words coming from the bag of words feature space, i.e. nodes are labeled with words whose entries have high values in the node's prototypes. Consequently, each node carries prototypical descriptive words giving hints to the human about their semantic content. The top two categories, *earn* (dark blue spheres at the top right) and *acquisition* (dark blue cylinders at the left) account for approximately 50% of all documents from the training set. Consequently, the substantial part of the map shows these two categories. In order to draw the focus more into the direction of the other topics, the user can adjust the fovea as indicated by the arrow in Fig. 8(a). Here, the node of the first hierarchical level at the 7 o'clock position is dragged towards the center of the map. The resulting image is shown in Fig. 8(b). Now, the top two categories are less dominant, and more visual space is used for displaying the other messages from the corpus. This brief example points out how the visualization framework of the H^{2}SOM might be used to explore a large text archive: The distribution of node symbols and keywords in the upper level provides the user with an overview such that he or she can quickly grasp, "what is going on in the archive". Possibly interesting regions can be brought into focus by smoothly browsing the Poincaré Disk. For example, in (c) the user has moved

the focus towards the *coffee* cluster at the bottom of the map (which contains 86% of all training items labeled by Reuters with the "coffee" topic). Note, that the neighboring area on the left covers the semantically close *cocoa* topic.

5.5 Visualizing Time in Document Streams

Today, many text domains like e-mails, news feeds, chatroom messages, web forums or web logs contain temporal information. Havre et al. (2002) have proposed the *ThemeRiver* which "depicts thematic variations over time within a large collection of documents".
We here pursue a similar approach and utilize the time-stamp which is attached to each document to order the documents in time and then compute a sequential mapping to the H^{2}SOM. By attaching to each node of the H^{2}SOM a time dependent activation potential defined as

$$a_i(t) = \beta\, a_i(t-1) + \mathcal{S}_i(t) \quad \text{with} \quad \mathcal{S}_i(t) = \begin{cases} 1 & \text{if } i \text{ is best-match} \\ & \text{node at arrival time } t \\ 0 & \text{otherwise} \end{cases}$$

where β is a decay factor controlling the amount of leakage, each node of the H^{2}SOM acts like a *leaky integrator*. As news items "flow" in, the neuron activities of the corresponding best match nodes in the hierarchy increase. At times with no news coverage, node activations decrease again. By continuously mapping the incoming data stream to the H^{2}SOM a "movie" of news activities can be generated. Fig. 9 shows a sequence of still images of such an animation where the time dependent activation potential $a_i(t)$ is mapped to the *z-axis* perpendicular to the Poincaré Disk.

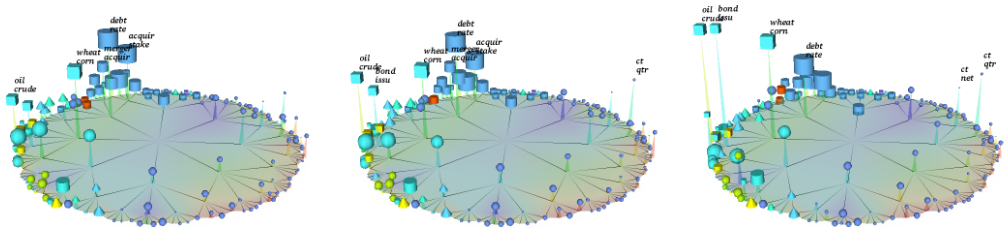
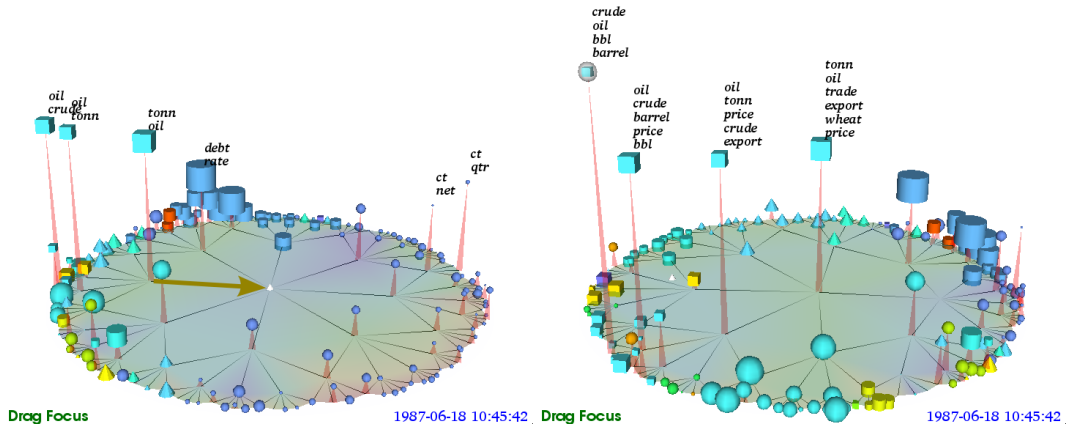


Fig. 9. Animation of news activities through time. The three still images grabbed from a movie stream show a developing news peak in the left part of the map.

During the animation of news activities keywords are generated and displayed at those node positions exceeding a certain activation threshold. In case of the developing peak in Fig. 9, these are "tonn", "oil" and "crude" as shown in the larger image of Fig. 10(a). The user interface allows to halt the animation at any time and to use the focus and context navigation framework for inspecting a possibly interesting region more closely. For our example, this is shown in

Fig. 10(b), where the peak region was moved towards the center of the Poincaré Disk, revealing more details. In order to inspect the underlying data at a single message level, the user can select a node and display the set of messages for which this node is the best match unit. Due to the hierarchical organization and the exponential growing behavior of the H^{2}SOM, the number of data items drastically decreases for nodes deeper in the hierarchy. In Fig. 10(b) the highest peaked node has been picked which selects the set of messages shown in the user interface of Fig. 11. The selection consists of 66 from more than 12000 messages, i.e. corresponds to a significant drill-down to approximately 0.5% of the data.



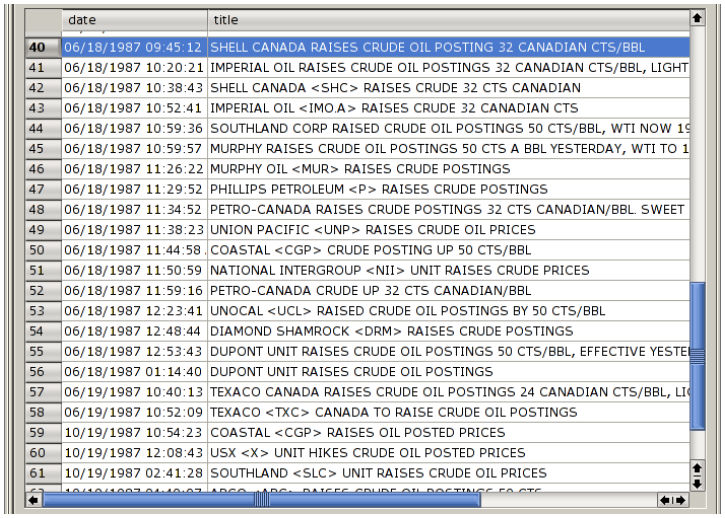
(a) (b)

Fig. 10. Two screenshots showing the news activity at a point in time where several oil companies raised their crude oil prices in parallel.

From the titles displayed in Fig. 11 it becomes evident, that the user has identified a point in time where many oil companies raised their crude oil prices almost in parallel, causing a burst of corresponding messages on the Reuters newswire. Interestingly, a few hours later, a second burst appears which covers messages about economic growth forecasts and different gross national products - indicating a close relation of the oil price to economic factors.

6 Conclusions

We have presented the Hierarchically Growing Hyperbolic SOM (H^{2}SOM), a new extension to the hyperbolic self-organizing map (HSOM). It enhances the HSOM along two important and mutually reinforcing directions: *(i)* the hyperbolic lattice structure is built incrementally by an adaptive growth process



	date	title
40	06/18/1987 09:45:12	SHELL CANADA RAISES CRUDE OIL POSTING 32 CANADIAN CTS/BBL
41	06/18/1987 10:20:21	IMPERIAL OIL RAISES CRUDE OIL POSTINGS 32 CANADIAN CTS/BBL, LIGHT
42	06/18/1987 10:38:43	SHELL CANADA <SHC> RAISES CRUDE 32 CTS CANADIAN
43	06/18/1987 10:52:41	IMPERIAL OIL <IMO.A> RAISES CRUDE 32 CANADIAN CTS
44	06/18/1987 10:59:36	SOUTHLAND CORP RAISED CRUDE OIL POSTINGS 50 CTS/BBL, WTI NOW 19
45	06/18/1987 10:59:57	MURPHY RAISES CRUDE OIL POSTINGS 50 CTS A BBL YESTERDAY, WTI TO 1
46	06/18/1987 11:26:22	MURPHY OIL <MUR> RAISES CRUDE POSTINGS
47	06/18/1987 11:29:52	PHILLIPS PETROLEUM <P> RAISES CRUDE POSTINGS
48	06/18/1987 11:34:52	PETRO-CANADA RAISES CRUDE POSTINGS 32 CTS CANADIAN/BBL SWEET
49	06/18/1987 11:38:23	UNION PACIFIC <UNP> RAISES CRUDE OIL PRICES
50	06/18/1987 11:44:58	COASTAL <CGP> CRUDE POSTING UP 50 CTS/BBL
51	06/18/1987 11:50:59	NATIONAL INTERGROUP <NII> UNIT RAISES CRUDE PRICES
52	06/18/1987 11:59:16	PETRO-CANADA CRUDE UP 32 CTS CANADIAN/BBL
53	06/18/1987 12:23:41	UNOCAL <UCL> RAISED CRUDE OIL POSTINGS BY 50 CTS/BBL
54	06/18/1987 12:48:44	DIAMOND SHAMROCK <DRM> RAISES CRUDE POSTINGS
55	06/18/1987 12:53:43	DUPONT UNIT RAISES CRUDE OIL POSTINGS 50 CTS/BBL, EFFECTIVE YESTE
56	06/18/1987 01:14:40	DUPONT UNIT RAISES CRUDE OIL POSTINGS
57	06/19/1987 10:40:13	TEXACO CANADA RAISES CRUDE OIL POSTINGS 24 CANADIAN CTS/BBL, LI
58	06/19/1987 10:52:09	TEXACO <TXC> CANADA TO RAISE CRUDE OIL POSTINGS
59	10/19/1987 10:54:23	COASTAL <CGP> RAISES OIL POSTED PRICES
60	10/19/1987 12:08:43	USX <X> UNIT HIKES CRUDE OIL POSTED PRICES
61	10/19/1987 02:41:28	SOUTHLAND <SLC> UNIT RAISES CRUDE OIL PRICES

Fig. 11. Messages from the drill down shown in Fig. 10(b).

which is guided in a top-down fashion, focusing computational resources initially on the extraction of the upper levels of a hierarchical structure, and then, guided by the formed "map nucleus", gradually spreading resources across the significant finer levels of the hierarchy. *(ii)* the entailing efficiency gain is further amplified by replacing the time-consuming SOM bestmatch search by an extremely fast approximation that exploits the intrinsically hierarchical structure of the hyperbolic lattice to search only an exponentially small fraction of all existing nodes for identifying a close-to-optimal match.

To quantify the ability of the H^{2}SOM to combine visualization and classification of high-dimensional data sets, we have conducted benchmark studies with the MNIST database of handwritten digits and with the Reuters-21578 newswire articles dataset. With respect to similar sized SOMs we obtain comparable or superior classification results, but with speed-ups of two orders of magnitude and more for maps with several thousand nodes. Moreover, analyzing for the Reuters corpus map quality in terms of a rank-correlation measure for global topology preservation, we find that the H^{2}SOM achieves better topology preservation at the same time with a lower quantization error as compared to a similar-sized SOM. Also at the local level, using the trustworthiness and continuity measures of Venna and Kaski (2001), we find superior H^{2}SOM map quality in most cases - only in a very narrow band of small neighborhood ranges can the SOM achieve a slightly higher trustworthiness than the H^{2}SOM. In addition, an evaluation of precision-recall curves (and the related F1-measure) indicates that the H^{2}SOM achieves a significant improvement both within individual categories as well as after micro- or macro-averaging over categories.

When comparing the H^{2}SOM to other hierarchical self-organizing methods, we find that all implementations are able to achieve a computational complexity

of $\mathcal{O}(\log N)$. Consequently, all algorithms should require similar calculation times for large-scale data sets. A quantitative comparison with respect to quantization errors or classification performance is more difficult to obtain, since to our knowledge the available publications on the TS-SOM, the GH-SOM, and the Evolving Tree do not offer quantitative benchmarks on publicly available large-scale data sets. Both Koikkalainen (1994) and Pakkanen et al. (2004) mention that the capability of their algorithms to find the "true" best match unit is very similar to that of a much slower global search. This result is very much in line with our findings for the tree search within the H^{2}SOM (c.f. Sec. 3.3). Similar to the Evolving Tree, the H^{2}SOM does not form regular SOM layers as the TS-SOM, but allows for a more flexible growing of its nodes. We therefore expect the H^{2}SOM to perform similar to the Evolving Tree with respect to classification accuracy.

We conclude that the H^{2}SOM provides a computationally very efficient and with regard to map quality and classification performance highly competitive alternative to both the standard SOM and the HSOM, enabling hierarchical self-organization in combination with smooth, map-like browsing in a way that so far - to the best of our knowledge - is not offered by existing approaches.

Acknowledgement: The authors would like to acknowledge the support of the Parmenides Foundation.

References

Baeza-Yates, R. A. and Ribeiro-Neto, B. A. (1999). *Modern Information Retrieval*. ACM Press / Addison-Wesley.

Bezdek, J. and Pal, N. (1995). An index of topological preservation for feature extraction. In *Pattern Recognition*, 28(3):381–391.

Coxeter, H. S. M. (1957). *Non Euclidean Geometry*. Univ. of Toronto Press, Toronto.

Freeman, R. T. and Yin, H. (2004). Adaptive topological tree structure for document organisation and visualisation. In *Neural Networks*, pages 1255–1271.

Goodhill, G. J. and Sejnowski, T. (1997). A unifying objective function for topographic mappings. In *Neural Computation*, 9:1291–1303.

Havre, S., Hetzler, E., Whitney, P., and Nowell, L. (2002). Themeriver: Visualizing thematic changes in large document collections. In *IEEE Transactions on Visualization and Computer Graphics*, 8(1).

Herrero, J., Valencia, A., and Dopazo, J. (2001). A hierarchical unsupervised growing neural network for clustering gene expression patterns. In *Bioinformatics*, 17(2):126–136.

Hotho, A., Staab, S., and Stumme, G. (2003). Explaining text clustering results using semantic structures. In *Principles of Data Mining and Knowledge Discovery, PKDD*.

Joachims, T. (1998). Text Categorization with Support Vector Machines: Learn-

ing with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, 1398, pages 137–142. Chemnitz, DE.

Kaski, S., Lagus, K., Honkela, T., and Kohonen, T. (1998). Websom–self-organizing maps of document collections. In *Neurocomputing*, 21:101–117.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. In *Biological Cybernetics*, 43:59–69.

Kohonen, T. (2001). *Self-Organizing Maps*. Springer Series in Information Sciences. 3rd edition.

Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Paatero, V., and Saarela, A. (2000). Organization of a massive document collection. In *IEEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery*, 11(3):574–585.

Koikkalainen, P. (1994). Progress with the tree-structured self-organizing map. In *11th European Conference on Artificial Intelligence (ECAI 1994)*, pages 211–215.

Koikkalainen, P. and Oja, E. (1990). Self-organizing hierarchical feature maps. In *Proc. of the IJCNN 1990*, volume II, pages 279–285.

Lagus, K., Honkela, T., Kaski, S., and Kohonen, T. (1996). Self-organizing maps of document collections: A new approach to interactive exploration. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 238–243.

Lamping, J. and Rao, R. (1994). Laying out and visualizing large trees using a hyperbolic space. In *ACM Symposium on User Interface Software and Technology*, pages 13–14.

Merkl, D. (1997). Exploration of text collections with hierarchical feature maps. In *Proceedings of the Annual Int'l ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97)*. ACM Press, Philadelphia.

Morgan, F. (1993). *Riemannian Geometry: A Beginner's Guide*. Jones and Bartlett Publishers, Boston, London.

Munzner, T. (1998). Exploring large graphs in 3D hyperbolic space. In *IEEE Computer Graphics and Applications*, 18(4):18–23.

Ontrup, J. and Ritter, H. (2001). Text categorization and semantic browsing with self-organizing maps on non-euclidean spaces. In *Proceedings of the 5th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 338–349. Springer, LNAI 2168.

Pakkanen, J., Iivarinen, J., and Oja, E. (2004). The evolving tree – a novel self-organizing network for data analysis. In *Neural Processing Letters*, 20(3):199–211.

Rauber, A., Merkl, D., and Dittenbach, M. (2002). The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data. In *IEEE Transactions on Neural Networks*, 13(6):1331–1341.

Ritter, H. (1999). Self-organizing maps in non-euclidian spaces. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 97–110. Amer Elsevier.

Ritter, H. and Kohonen, T. (1989). Self-organizing semantic maps. In *Biological Cybernetics*, 61:241–254.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. In *Information Processing and Management*, 24(5):513–523.

Sebastiani, F., Sperduti, A., and Valdambrini, N. (2000). An improved boosting algorithm and its application to automated text categorization. In *Proceedings*

of CIKM-00, 9th ACM International Conference on Information and Knowledge Management, pages 78–85.

Skupin, A. (2002). A cartographic approach to visualizing conference abstracts. In *IEEE Computer Graphics and Applications*, 22(1):50–58.

Venna, J. and Kaski, S. (2001). Neighborhood preservation in nonlinear projection methods: An experimental study. In G. Dorffner, H. Bischof, and K. Hornik, editors, *Artificial Neural Networks—ICANN 2001*, pages 485–491. Springer, Berlin.

Venna, J. and Kaski, S. (2005). Local multidimensional scaling with controlled tradeoff between trustworthiness and continuity. In *Proceedings of WSOM'05, 5th Workshop On Self-Organizing Maps*, pages 695–702. Paris.

Walter, J. (2003). H-MDS: a new approach for interactive visualization with multidimensional scaling in the hyperbolic space. In *Information Systems, Elsevier*.

Walter, J., Ontrup, J., Wessling, D., and Ritter, H. (2003). Interactive visualization and navigation in large data collections using the hyperbolic space. In *Proceedings of the Third IEEE International Conference on Data Mining*. IEEE.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. In *Information Retrieval*, 1-2(1):69–90.