

Neural Gas Sonification - Growing Adaptive Interfaces for Interacting with Data

Thomas Hermann and Helge Ritter
Neuroinformatics Group · Faculty of Technology · Bielefeld University
Bielefeld, Germany
{thermann|helge}@techfak.uni-bielefeld.de

Abstract

In this paper we present an approach using incrementally constructed neural gas networks to ‘grow’ an intuitive interface for interactive exploratory sonification of high-dimensional data. The sonifications portray information about the intrinsic data dimensionality and its variation within the data space. The interface follows the paradigm of model-based sonification and consists of a graph of nodes that can be acoustically ‘excited’ with simple mouse actions. The sound generation process is defined in terms of the node parameters and the graph topology, following a physically motivated model of energy flow through the graph structure. The resulting sonification model is tied to the given data set by constructing both graph topology and node parameters by an adaptive, fully data-driven learning process, using a growing neural gas network. We report several examples of applying this method to static data sets and point out a generalization to the task of process analysis

1. Introduction

Exploratory Data Analysis is the important and necessary first step to investigate datasets in search of structures, patterns and regularities that may be well hidden in the data. Particularly the investigation of high-dimensional data is aggravated by the missing intuitivity of high-dimensional spaces, and the suboptimal interface between the data and the human brain, provided by the sensory system on the side of the human, and ‘displays’ on the side of the computer. How can we facilitate the construction of exploratory data interfaces that can provide efficient bridges between these two sides?

Within the field of Sonification, the framework of *Model-based Sonification* (MBS) has been a major step towards a principled and task-oriented mediator between the ‘data-world’ and the perceptual space [3]. It combines an intuitive and physically motivated method for tuning a sonifica-

tion to a certain analysis task with a high degree of flexibility to remain generic in the data for which it is applicable. As a result, it offers a very good basis to make use of human learning skills.

This paper addresses the question, how the process of hand-crafting model-based sonifications can be replaced by a more automatic procedure, capable of creating (a range of) good MBS interfaces in a *purely data-driven way*.

We present an approach that employs graph structures of acoustically excitable elements as sonification models, created from a given data set by employing a growing neural gas network (GNG, [2]) in order to tie the model structure to the given data. The structure of the GNG is a network of neurons and connections. During the GNG adaptation process, the network graph is modified in different ways: by adapting neuron weight vectors, by inserting new neurons, and by updating edges between neurons according to a data-driven Hebbian criterion. The GNG increasingly adapts to the distribution until it finally shows over-fitting and adapts to the noise.

Regarding neurons as masses and connections as energetic couplings (e.g. springs), the comparison with acoustic systems is rather close-by. The GNG sonification model takes this analogy and creates a data-driven network of coupled acoustic systems that can be excited in the same way as a drum is excited, by tapping at a certain location.

The human’s auditory system is well trained to analyze sounds which emerge from the dynamical behavior of such connected acoustic units for its relevant source-related properties. Physics has actually given rise to evolutionary optimizations, engraved in our neural architecture to support such analysis [6]. Using an analogous mechanism for interactive exploration of datasets thus addresses a highly evolved set of information processing skills that we are rarely aware of in everyday life.

The paper is structured as follows: in Section 2 we briefly sketch the GNG algorithm. In Section 3 we first give a condensed introduction to the framework of Model-based Sonification and then introduce the GNG sonification model in detail. This is followed by a set of exploration examples that

illustrate the use on synthetically rendered data and also provides examples for real-world data sets. The extension of the model towards a sonification of the growth process is presented and exemplified in Section 4. The paper closes with a discussion of the particular benefits in this sonification type.

2. Growing Neural Gas Networks

In unsupervised learning all available variables in a dataset are regarded as input variables, assuming no information about a desired output. In these situations one possible objective for learning is the reduction of data dimensionality. A standard technique for this is to project the data on a few principal directions which span a linear subspace so that a good approximation is obtained. In the case that the data are being distributed on a low-dimensional curved (non-linear) manifold in data space, however, linear approximations require a higher dimensionality than the data intrinsically possesses. In such situations, non-linear projection methods allow reduced approximation errors with manifolds of lower dimensionality. Principal Curves (resp. Surfaces) are nonlinear manifolds which are able to reduce the dimensionality in this sense [7]. The Self-Organizing Map (SOM) can be regarded as a rough approximation of the nonlinear mapping, using reference points (neurons) whose topological neighborhood is defined in advance [12, 1]. But SOMs and principal surfaces deliver only poor approximations if the intrinsic data dimensionality deviates from the dimension of the surface or the neuron grid.

A more adaptive approach to solve this problem consists of an algorithm that also learns the required topological network structure. A very suitable approach is based on the idea of “competitive Hebbian learning” (CHL), combined with the neural gas algorithm [11]. CHL aims at finding a subgraph of the Delaunay triangulation, limited to those areas of input space which are covered by the distribution. Such a graph is called the “induced Delaunay triangulation” and is shown to optimally preserve topology [9].

It is constructed for a given set of reference vectors or centers by connecting the closest two centers to each input \mathbf{x} by an edge. Therefore a prerequisite for CHL is a suitable set of reference points which can for instance be obtained by vector quantization. As a special case, the neural-gas (NG) [10] can be taken, where basically for each input the coordinates of the K nearest centers are adapted with decreasing K during learning. Both the adaptation range K and the adaptation strength underly a decay schedule for which the number of adaptation steps has to be defined in advance. An alternative to the successive application of the NG and the CHL algorithms for topology learning is the Growing Neural Gas (GNG) proposed by Fritzke [2]. It has

the advantage that reference vector positioning and connection growth occur in an integrated fashion, within a single learning process. This seemed slightly more convenient for the present application, and therefore, we adopted that choice. The next section gives a sketch of the method to make the paper self-contained.

2.1. The GNG algorithm

The used GNG algorithm is taken from Fritzke [2]. A brief summary shall make the reader familiar with the basic concept and used nomenclature.

The GNG is given by a set of neurons or reference vectors \mathbf{w}_i which are regarded as positions in input space and a list of undirected edges e_j among neuron pairs. In addition, an error accumulator R_i is maintained for each neuron and a counter A_j for every edge to store its ‘age’. The algorithm starts with two units $\mathbf{w}_1, \mathbf{w}_2$ at random position in input space.

1. Draw a data point \mathbf{x} from the underlying distribution.
2. Find the nearest and second nearest neurons i_1, i_2 .
3. Increment the age A_j of all edges j emanating from neuron i_1 .
4. Update $R_{i_1} \leftarrow R_{i_1} + \|\mathbf{w}_{i_1} - \mathbf{x}\|$
5. Update neuron positions for neuron i_1 and its topological neighbors n by

$$\begin{aligned}\Delta \mathbf{w}_{i_1} &= \epsilon_b (\mathbf{x} - \mathbf{w}_{i_1}) \\ \Delta \mathbf{w}_n &= \epsilon_n (\mathbf{x} - \mathbf{w}_n)\end{aligned}$$

6. Create an edge j between neuron i_1 and i_2 , if it does not already exist. Set its age $A_j = 0$.
7. Remove those edges with $A_i > a_{max}$. Remove neurons without edges.
8. Every λ steps:
 - Insert a new neuron q half-way between the neuron $q_1 = \arg \max_i R_i$ and its topological neighbor neuron q_2 with the largest error. The edge between q_1 and q_2 is removed and edges from both neurons to the inserted neurons are added.
 - Update $R_{q_1} \leftarrow \alpha R_{q_1}$, $R_{q_2} \leftarrow \alpha R_{q_2}$ and set $R_q = R_{q_1}$.
9. Multiply all errors R_j with a constant $\rho < 1$.
10. Proceed with step 1 until a stopping criterion is fulfilled.

The constants $\epsilon_b, \epsilon_n, a_{max}, \lambda, \alpha, \rho$ determine the operation of the GNG algorithm. Their role within the algorithm (discussed in detail in [2]) is intuitively clear. For most datasets the following fixed values work fine: $\epsilon_b = 0.2$, $\epsilon_n = 0.006$, $\alpha = 0.5$, $a_{max} = 50$ and $\rho = 0.995$. λ was set to a constant fraction of the dataset size. As a stopping criterion, a final net size was used as indicated below. Some

modifications were introduced: (a) smoothing is obtained by adding noise to the inputs, (b) an optimization mode is introduced where simply neuron insertion is skipped [3].

3. GNG Sonification Model

A model-based approach to auditory display has been developed in [5, 3]. Different to *parameter mapping*, where a mapping of data features on sound features is created, here the data are used to specify the structure of a dynamical process (called sonification model), whose temporal evolution is determined by the equations of motion and external excitation. Thus, the data set can be queried by interacting with the data-driven sonification model. Well designed models connect certain structural properties of interest to structures in the sound in a canonic way, such that a model can be used with all sorts of datasets of arbitrary dimensionality.

The GNG sonification model is defined for datasets given by a data matrix $\mathbf{X} \in M(N \times d, \mathbb{R})$ whose row vectors $(\mathbf{x}_i^T, \mathbf{y}_i^T)$ are the given records with $i = 1, \dots, N$.

3.1. Setup

The virtual acoustic object defined in the GNG sonification model lives in a \mathbb{R}^d model space. The GNG graph is given by the neuron weight vectors \mathbf{w}_i , interpreted as coordinates of the neuron objects which are connected via the edges, which are shown in the plots as straight mesh lines. For the GNG sonification model, the edges are used as ‘conductors’ which transport energy between neurons. A typical GNG graph is illustrated in Fig. 3. It shows the projection of the setup in model space onto two selected axes.

For the GNG probing interaction which will be discussed first, the GNG configuration remains unchanged during exploration. Instead, each neuron i is given an energy variable E_i which is initially set to 0.

3.2. Dynamics

The dynamics describes the energy exchange between neurons that are connected by an edge. The energy vector \mathbf{E} is updated by numerical integration of the energy flow equation

$$\frac{dE_i}{dt} = -gE_i(t) - \sum_{j \in I_N(i)} q(E_i(t) - E_j(t)) \quad (1)$$

where $I_N(i)$ is the set of indices of all neurons that are connected with neuron i . The first term of eq. (1) causes an exponential energy decay. The dissipative energy loss may be imagined as radiated sound field energy. The second term causes energy diffusion through the network graph. With

$g = 0$, the equilibrium state would be a uniform distribution of the total energy $E_{tot} = \sum_i E_i$ on all neurons connected to the neuron that was excited. The speed of energy decay and energy transport can be controlled independently by adjusting g and q . Technically, the dynamics is implemented as a series of update steps iterating through all edges, where a fraction q_{step} of the energy difference between connected neurons is transported to the neuron with smaller energy.

3.3. Initial State and Excitation

The energy vector \mathbf{E} is initialized to 0. The system thus is in a state of equilibrium and silent. Excitation is done by probing the virtual acoustic object defined by the sonification model: selecting a neuron in the GNG visual display, by clicking the computer mouse, increases the neuron energy by a value 1. This interaction is analogue to striking (e.g. on a metal bar), where the kinetic energy of a system is punctually changed.

3.4. Model-Sound Linking and Listener

The sonification is the superposition of all neuron sounds. The neuron sound level resp. the sound amplitude is directly related to its energy E_i . Various choices may be suitable for defining the elementary neuron object acoustic behavior. This is the ultimate point where task-dependent information requirements may be integrated to the GNG sonification model. However, to keep the discussion simple we here use a damped linear oscillator with energy, mass and stiffness as the only determining parameters. For simplicity, in addition most of them are set to constant values. It should be emphasized, however, that more complex systems are expected to augment the auditory display beyond its elementary utility exemplified here.

The sonification thus is the superimposed sound signal obtained from all existing neuron objects. Although the network is defined in a spatial setup, the listener shall not be located anywhere in relation to the model space. The sonification is thus so far a mono sound vector.

3.5. Sound Synthesis

Additive Synthesis is used for sound computation. The sound computation is directly implemented in C, by interpolating amplitude and frequency values between equally spaced control points in time.

3.6. Data-Model Assignment

The GNG is characterized by the neuron positions w_i , the list of edges, the edge age vector and the error variables R_i . These variables are possible candidates to determine acoustic properties. However, as the aim was to learn something about the dimensionality or topology of the network, a derived quantity is used: the number of edges $n_e(i)$ emanating from neuron i . This variable is used since it correlates with the intrinsic dimensionality of the data. Fig. 1 shows the average number of edges per neuron for GNG networks with 100 neurons adapted to a uniform q -dimensional distribution in $[0, 1]^q$. The average number of edges grows

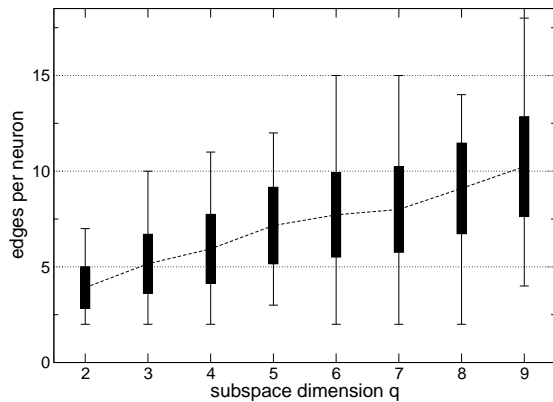


Figure 1. Number of edges per neuron for q -dimensional uniform distributed data (1000 records) in $[0, 1]^q$ using 100 neurons. The box limits indicate the standard deviation.

monotonously with subspace dimension.

Think about edges to fixate the neuron objects to their position – the more edges are connected, the higher the frequency with which the neuron will oscillate. So $n_e(i)$ determines the frequency of the time-variant oscillator that is used to generate the neuron sound, whereas the amplitude is determined by the energy as pointed out above. In GNG probing, $n_e(i)$ remains constant and thus each neuron contributes a tone of fixed frequency. The frequency is mapped from n_e by $f(n_e) = f_0 2^{ln_e/12}$ which causes a shift of l musical half-tones for each neighbor. Typical choices are $f_0 = 150$ Hz, $l = 3 \dots 5$, but this is subject to interactive adjustment by the user or model designer in order to make reasonable use of the audible frequency spectrum.

3.7. Implementation

Fig. 2 shows the visual display for interacting with the sonification model. A number of parameters can be changed

here like the sampling rate, the total duration, level, number of dynamic steps, the energy gain g and the flow speed q_{step} . Computation of the sonification is triggered by selecting a neuron in the scatter plot.

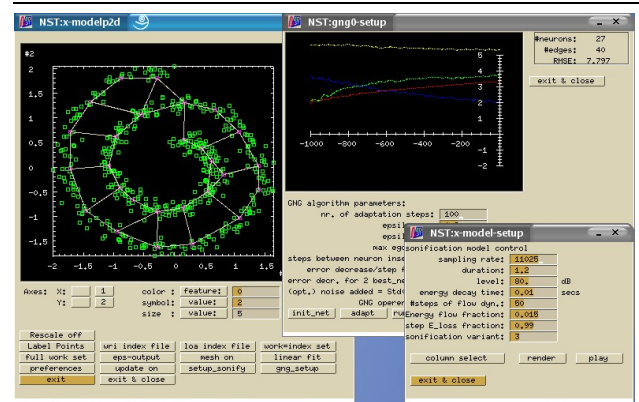


Figure 2. Screenshot of the User Interface for controlling the GNG sonification model. The plot shows a 2D noisy spiral dataset, and the GNG graph.

4. Exploration Examples

The particular advantage of sonification models is that they can be designed to work with arbitrary datasets, independent upon size, dimensionality, etc. This is a very important difference from other sonification paradigms like that of parameter mapping where every sonification is a 'unique creation' which needs its detailed mapping in order to be of use for the listener. In MBS, however, the dynamics is unchanged, and thus the listener can (*i*) adapt to the sounds of a model (*ii*) profitably make use of any previously perceived data sonifications, (*iii*) infer from sound the underlying meaning with respect to the data in all contexts of model use. Particularly (*ii*) gives good arguments for designing a set of benchmark structures in synthetically rendered data. The sonifications provide in a way templates for orienting in sound. A real world example would be to listen first to typical sounds of different types of vehicles (lorry, car, motorbike), which later help to make sense of sounds from unknown vehicles or even groups of vehicles (resp. mixtures of distributions that constitute a data structure).

As a simple starting point we exemplify GNG sonification for Gaussian distributions of different intrinsic dimensionality, followed by some topologically simple setups like a nonlinear one-dimensional manifold (e.g. the spiral) or

a toroidal structure. This is followed with real-world data sonifications - which here however serve only as examples.

4.1. Mixture of Gaussians

For the first example, data were sampled from a mixture of Gaussians with density

$$p(\mathbf{x}) = \frac{1}{8} \sum_{i=1}^8 g_{0-i}(\mathbf{x}; i \cdot \hat{\mathbf{x}}_0, \sigma^2 \mathbf{I}_i) \quad (2)$$

and $\sigma^2 = 0.2$. g_{a-b} denotes here the density of a normal distribution in the variables $\{x_a, x_{a+1}, \dots, x_b\}$ which is taken to be concentrated on the null space of the remaining variables. The intrinsic dimension of the clusters ranges from 1 to 8. Fig. 3 shows a scatterplot of three of the clusters together with a GNG network with 50 neurons. The sound

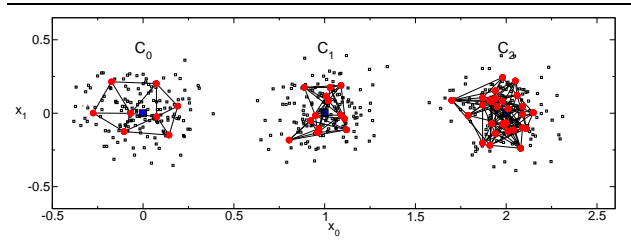
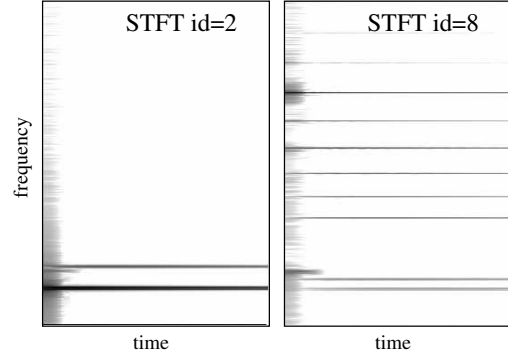


Figure 3. Clusters of different intrinsic dimensionality together with a GNG graph consisting of 50 neurons. From left to right, the IDs are 2, 4, and 8.

examples (see Table 4) are rendered by clicking on different neurons close to the different cluster centers. The brilliance of the sound increases with intrinsic data dimensionality. While the attack phase is determined by the selected neuron, at the end of each sonification the energy is distributed to all neurons in the same subgraph. Energy propagation to topological neighbors causes the spectrum to broaden since the connectivity n_e may be different.

4.2. Noisy Spiral Dataset

The following example demonstrates energy propagation for a curved one-dimensional data distribution. A 2D noisy spiral dataset is used, since it is suitable to demonstrate some dependencies of GNG sonification on network size while at the same time maintaining the ease of a simple visualization. Fig. 5 shows scatter plots of the dataset and typical GNG graphs for different network sizes. In the sound examples (see Table 1) for the GNG network with



sound examples for $d = 1 \dots 8$, for each dimension two sound examples (a),(b) are given, differing in the excitation location within the same cluster.

Figure 4. GNG sonification probing of Gaussian clusters of different intrinsic dimensionality in \mathbb{R}^{10} , some of them are shown in Fig. 3. The plots show the spectrogram of a sonification for the 2D and 8D case. Sound files are available at [4].

3 neurons (plot (a)), all neurons contribute the same pitch because all neurons are connected with two neighbors. Such a small network is incapable of properly reflecting data dimensionality in its graph. The situation changes with in-

Examples for Fig. 5

- (a) GNG with 3 neurons 1, 2
- (b) GNG with 20 neurons (a), (b), (c)
- (c) GNG with 45 neurons (a), (b), (b-c), (c)
- (d) GNG with 150 neurons (a), (b), (c)
- (e) GNG with 20 neurons (a), (b), (c)
- (f) GNG with 45 neurons (a), (b), (c)

Table 1. Sound examples for GNG sonifications for the noisy spiral dataset shown in Fig. 5. The sonification model is excited at the outer end (a), in the middle (b) and the inner end (c) of the spiral.

creasing network size. In (b), the GNG sonification changes to mixtures of 2 or 3 tones of higher pitch. The graph indicates that most neurons have about 4 topological neighbors. The initial pitch depends on the number of edges at the selected neuron. Since the energy is distributed equally among all neurons at the end of the sonification, the relative level at each pitch indicates how many neurons have the associate number of edges. Obviously, the GNG graph

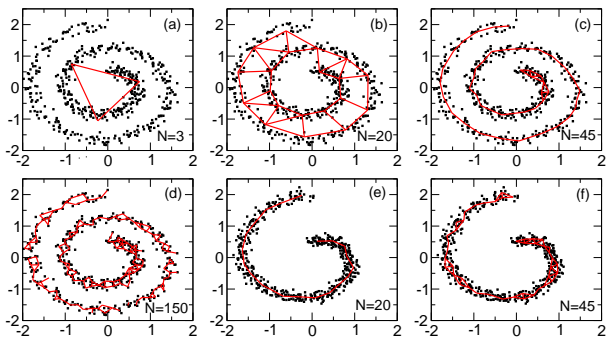


Figure 5. Noisy Spiral dataset and GNG graphs: (a) - (d) show the noisy spiral with 2 rotations. Obviously, with increasing GNG size, the GNG adapts to a suitable fit until the GNG is over-fitting the data. (e) and (f) show a noisy spiral with only 1 rotation. Here the GNG begins to overfit with 20 neurons.

is a two-dimensional ring. The limited network size prevents the GNG from discovering the actual spiral structure. This is achieved at a network size of about 45 neurons, shown in (c). The GNG sonification sound examples now again show a simpler sound consisting mainly of one pitch, since the occurrence of 2-edged neurons dominates. However, at the inner end of the spiral, the GNG starts to adapt to the noise which becomes audible as a more complex sound when the GNG is probed there. If the adaptation is continued, the neurons converge more and more to the samples and the GNG graph can collapse into disconnected subgraphs. Some GNG sonification examples are given for the GNG graph shown in plot (d).

The required GNG network size may depend on the given data set to appropriately describe its structure. Plot (e) and (f) in Fig. 5 show a noisy spiral with one rotation. Here, 20 neurons suffice to get a suitable graph and over-fitting begins with 45 neurons. It is up to the researcher to evaluate the quality of a fit and exploratory tools can aid this process.

5. GNG Growth Process Monitoring

From the previous discussion it became apparent that the growth process itself reflects valuable information about the structure in data spaces. This provided the inspiration for an extension of the GNG probing sonification towards a process monitoring sonification. The continuous changes of audible structure during GNG adaptation matches our high sensitivity in discerning time-evolving auditory structures – in fact, our ear is better in comparing sound structures than in drawing conclusions from a single sonification alone. For

instance, it is easier to say that the intrinsic dimension of one cluster is higher than that of another, but difficult to assess what exactly it is.

Since sound complexity corresponds to network graph complexity, the process monitoring sonification may even be useful to evaluate when the GNG starts to converge or overfit the data. The latter feature is particularly of interest since no canonical criterion exists when to stop the GNG growth. While in the toy examples shown so far over-fitting is easily detected from the graphs, this is much more difficult in high-dimensional settings and GNG process sonifications may provide here valuable assistance. Besides the sonification rendering introduced so far, there are numerous events occurring during GNG adaptation, like neuron insertion, neuron deletion, edge creations, etc., which are usually not attended while using GNG models. Such events can easily be integrated into the sonification, either by creating additional auditory streams, or by defining excitatory effects on their local vicinity. As a result, this can keep the user better in touch with the progress of the adaptation.

In contrast to the GNG sonification model presented above, here the acoustic energy is now distributed uniformly among all neurons for the whole sonification time so that no energy flow takes place and the sonification provides a summary over the whole network at any instance. The GNG process sonification is rendered by performing a sequence of adaptation/sonification cycles. N_a adaptation steps are performed in a single adaptation cycle. In the following examples, $N_a = 500$ is used and neurons are inserted every 300 adaptation steps. In a sonification cycle, the sound is computed by superimposing an auditory grain for each neuron. The auditory grain frequency is associated to the number of edges per neuron n_e exactly as in the previous approach. The auditory grains are further characterized by their amplitude envelope which is here chosen as a triangular window function. The density of the auditory grains in time is determined by the number of cycles and the total duration of the sonification.

Sonifications are presented for the four datasets shown in Fig. 6. The sonifications are rendered with 80 cycles in 5 s sonification time. A sampling rate of 44100 Hz is used and the auditory grains last 200 ms.

The first example is a process sonification for a GNG growth in a dataset given by the noisy spiral shown in Fig. 6, (a). The sonification starts with a low pitched spectrum indicating the presence of only a few edges per neuron. With increasing network size a stable pitch structure becomes audible for some time, consisting of a low pitched tone. During this time the GNG structure is a one-dimensional chain with two edges per neuron. With increasing network size the GNG more and more adapts to the noise. As a result, the number of edges increases and the sound brilliance rises. In the end the level in the lowest tone increases again. This is

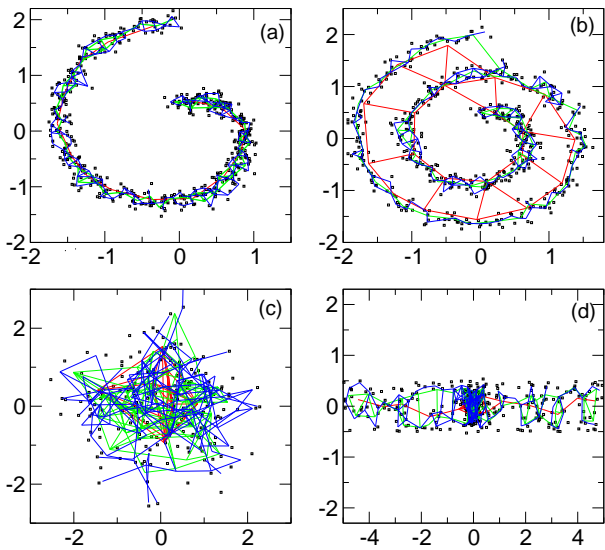


Figure 6. Scatter plots of example datasets and GNG networks at 3 phases of the growth process. The GNG graphs correspond to network size 20 (red), 50 (green) and 135 neurons (blue). (a) noisy spiral with one rotation, (b) noisy spiral with two rotations, (c) 5D Gaussian (d) mixture of a 2D rectangular uniform distribution and a 5D Gaussian. Sound examples are available at [4].

caused by deletion of edges due to over-fitting. Some neurons are then only activated by single data points. Such an increase of loudness at the lowest tone generally indicates over-fitting and may be used as an auditory feature to value model complexity.

The next example is a sonification for the noisy spiral dataset shown in Fig. 6, (b). A remarkable difference can be perceived during the first half second of the sound: the spectral mean grows rapidly, indicating that at the beginning, the net is two-dimensional to approximate the data. With increasing network size, the GNG adapts to the finer one-dimensional curve and edges between adjacent braces of the spiral are deleted. This causes a shift of the spectral centroid towards lower frequencies. Finally, as in the last example, over-fitting begins. The dynamical evolution of spectrum is typical for situations where a higher-dimensional data distribution is built up from lower dimensional pieces.

Very different from the latter sound is the sonification of GNG growth for a 5D Gaussian, shown in Fig. 6, (c). The number of edges per neuron now grows much faster than in the previous two-dimensional examples. However, it can be heard, that directly after the beginning of the sonification, neurons with few edges are very rare. This is typical for data that is distributed rather homogeneously in a high-

dimensional space. As with the other examples, edges are getting deleted due to over-fitting which leads to a growth of acoustic energy in the low-frequency part at the end.

The final sound example for the dataset in Fig. 6 (d) sounds similar to the previous one. In contrast to that, however, the sound level at the lower modes does not disappear in the beginning. Carefully comparing both examples, it is possible to follow the lower pitched plateau. This part of the spectrum is caused by the neurons in the 2D distribution.

The following examples involves a 64-dimensional real-world dataset, containing pixel intensities of 8×8 bitmaps for handwritten digits. The data were obtained by subsampling the 24×24 pixel data from the MNIST database [8]. For the characters '1' and '2', more than 1000 records are used. A PCA plot together with the GNG are shown in Fig. 7. The PCA eigenvalues shown below suggest that class '2' shows a higher complexity, corresponding to more internal degrees of freedom in writing a '2' digit. This difference is becoming audible in the sonifications, see [4] for sound files and further discussion.

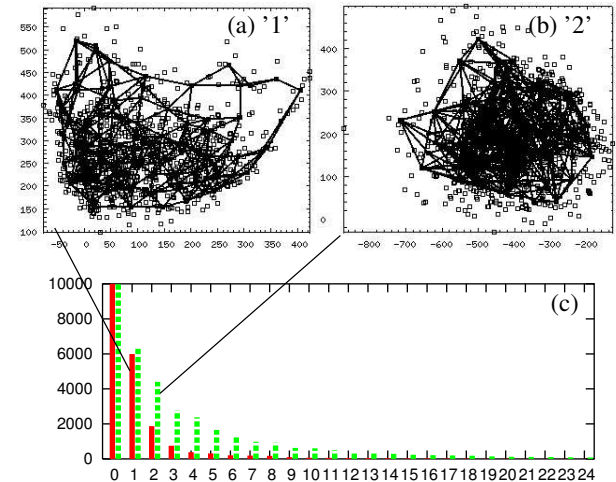


Figure 7. GNG for the MNIST handwritten digits datasets for classes '1' and '2'. The plot shows the variances along the principal components.

6. Conclusion

We have presented a new approach to construct model-based sonification interfaces in a data-driven way, using an adaptive growth process inspired from neural gas models. The method “embodies” a given data set as a compact,

graphical sonification model that mediates between user actions and data space in a rather intuitive fashion by offering a graph of nodes that can be acoustically excited in a direct and physically motivated way, employing simple mouse actions. The neural network here serves two purposes: offering a segmentation of data that preserves the topology of the data, and by constituting a connection between the dataset and a model-based setup of acoustically active elements. This sonification approach is extremely flexible: the method is easily tuned to support other exploratory tasks, e.g. by binding the single neuron sound to other local properties of interest, like local entropy etc..

The probing sonifications start with information about very localized properties and end – due to the propagation of energy along edges – with a summary over a larger environment. This allows a comparison at different levels of resolution, and also assists the comparison of different probings. The phenomenon of timbre evolution is encountered in many acoustic systems, there mainly due to nonlinearities of a dynamic system. However, we expect that the parallels to sound structure in real-world sound systems supports the perception and analysis of structure in the model use.

The extension of GNG Process Sonification showed that adaptation processes can be sonified by using iterated adaptation and sonification cycles using a modified probing model. With the modified GNG sonification model the evolution of network connectivity becomes audible as temporal evolution of spectrum. This enables the listener to perceive dynamical patterns (e.g. timbre plateaus) which may help to understand or explain the data at hand.

In the GNG sonification model, the element of energy propagation along edges is an important new building block that may be used also in other sonifications as for instance for rendering sonifications for Self-Organizing Maps.

Finally, the Growing Neural Gas Sonification model offers a principled generic versatile, task-oriented approach how to bind auditory representations, visualization, and interaction together to a coherent exploratory system. We regard the move towards real-time sound synthesis with real-time visual feedback as the most important next step to increase the binding of the different modalities and focus on this in our ongoing research.

References

- [1] V. Cherkassky and F. Mulier. Self-organizing networks for nonparametric regression. In *From Statistics to Neural Networks*, pages 188–212, 1994.
- [2] B. Fritzke. A growing neural gas network learns topologies. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 625–632. The MIT Press, 1995.
- [3] T. Hermann. *Sonification for Exploratory Data Analysis*. PhD thesis, Bielefeld University, Bielefeld, 2 2002.
- [4] T. Hermann. Sonification for exploratory data analysis – demonstrations and sound examples. <http://www.techfak.uni-bielefeld.de/~thermann/projects/index.html>, 2002.
- [5] T. Hermann and H. Ritter. Listen to your data: Model-based sonification for data analysis. In G. E. Lasker, editor, *Advances in intelligent computing and multimedia systems, Baden-Baden, Germany*, pages 189–194. Int. Inst. for Advanced Studies in System research and cybernetics, 1999.
- [6] T. Hermann and H. Ritter. Sound and meaning in auditory data display. *Proceedings of the IEEE, Special Issue on Engineering and Music - Supervisory Control and Auditory Communication*, 92(4), April 2004.
- [7] M. LeBlanc and R. Tibshirani. Adaptive principal surfaces. *Journal of the American Statistical Association*, 89(425):53–65, 1994.
- [8] Y. LeCun. The MNIST Database of Handwritten Digits. <http://www.research.att.com/~yann/ocr/mnist/index.html>, 1998.
- [9] T. M. Martinetz. Competitive hebbian learning rule forms perfectly topology preserving maps. In *Proc. of the ICANN'93*, pages 427–434, Amsterdam, 1993. Springer.
- [10] T. M. Martinetz and K. J. Schulten. A "neural-gas" network learns topologies, volume 1, pages 397–402. North-Holland, Amsterdam, 1991.
- [11] T. M. Martinetz and K. J. Schulten. Topology representing networks. *Neural Networks*, 7(3):507–522, 1994.
- [12] H. Ritter, T. Martinetz, and K. Schulten. *Neural Computation and Self-Organizing Maps. An Introduction*. Addison-Wesley, Reading, MA, 1992.