A Sparse Grid Based Generative Topographic Mapping for the Dimensionality Reduction of High-Dimensional Data

Michael Griebel and Alexander Hullmann

Abstract Most high-dimensional data exhibit some correlation such that data points are not distributed uniformly in the data space but lie approximately on a lower-dimensional manifold. A major problem in many data-mining applications is the detection of such a manifold from given data, if present at all. The generative topographic mapping (GTM) finds a lower-dimensional parameterization for the data and thus allows for nonlinear dimensionality reduction. We will show how a discretization based on sparse grids can be employed for the mapping between latent space and data space. This leads to efficient computations and avoids the 'curse of dimensionality' of the embedding dimension. We will use our modified, sparse grid based GTM for problems from dimensionality reduction and data classification.

1 Introduction

High-dimensional data often exhibit a correlation structure between the variables, which means that there are areas in the data space with little or no data points. A suitable low-dimensional projection of the data then allows a more compact description, a better visualization and a more efficient processing.

One approach to dimensionality reduction is to express the high-dimensional data in terms of latent variables. A well-known method is the Principal Component Analysis (PCA), which is based on the diagonalization of the data covariance matrix. However, the PCA is by construction a linear method. As such, it is not capable of modeling nonlinear lower-dimensional dependencies and sometimes may fail. A simple three-dimensional example, the so called 'Swiss roll', is given in Fig. 1. Here, the topological structure of the data is not preserved under the mapping into two dimensions and points originally far apart on the manifold are close-by in the two-dimensional projection.

This is why nonlinear methods are necessary. Some common approaches are multidimensional scaling (MDS), curvilinear component analysis (CCA), curvilin-

M. Griebel • A. Hullmann (✉)
Institute for Numerical Simulation, University of Bonn, Bonn, Germany
e-mail: griebel@ins.uni-bonn.de; hullmann@ins.uni-bonn.de

© Springer International Publishing Switzerland 2014

H.G. Bock et al. (eds.), *Modeling, Simulation and Optimization of Complex Processes - HPSC 2012*, DOI 10.1007/978-3-319-09063-4_5

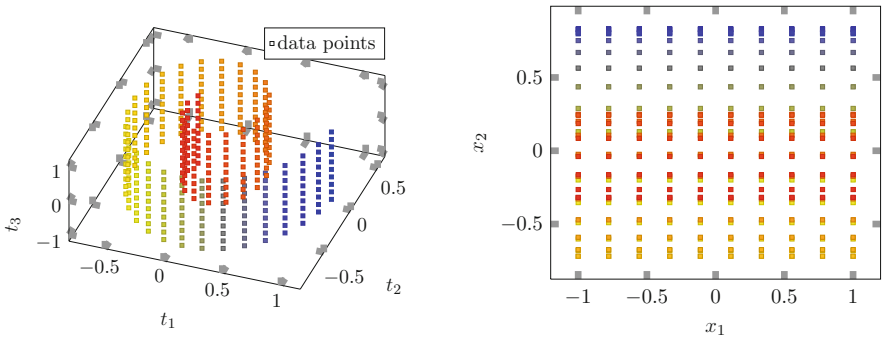

Fig. 1 The projection of the 'Swiss roll' data (*left*) onto the first two principal components results in a two-dimensional representation (*right*)

ear distance analysis (CDA), Laplacian eigenmaps (LE), locally linear embedding (LLE), Kohonen's self-organizing map (SOM), generative topographic mapping (GTM) and kernel PCA (KPCA), cf. [17]. Unfortunately, capturing nonlinearities comes at the price of a significant increase in computational complexity and with the problem of possibly finding only a locally optimal solution.

In this article we will focus on the generative topographic mapping (GTM) [4]. Usually, the latent space of the generative model is limited to two or three dimensions due to the 'curse of dimensionality'. It means that the cost complexity for the approximation to the solution of a problem grows exponentially with the dimension d, i.e. it is of the order $\mathcal{O}(h^{-d})$ with h being the one-dimensional mesh-width. Instead, we use sparse grids [6] for the discretization of the mapping between latent space and data space. Then, the number of degrees of freedom grows only by $\mathcal{O}(h^{-1}\,(\log h^{-1})^{d-1})$, which is a substantial improvement. This approach has also been followed for principal curves and manifolds in [10]. Of course, this saving in computational complexity comes at a cost, namely an additional logarithmic error term and a stronger smoothness assumption on the mapping. As a result, we get a sparse GTM (SGTM), which basically achieves the same level of accuracy with less degrees of freedom. In contrast to the conventional GTM, it can cope with higher-dimensional latent spaces.

This paper is organized as follows: In Sect. 2, we describe our generative model, which is based on a mapping between the lower-dimensional latent space and the data space. In Sect. 3, we present a method to find the mapping by minimizing a certain target functional, i.e. the regularized cross-entropy between the model and the given data. Then, in Sect. 4, we show how we can obtain the original GTM as well as the sparse GTM by special discretization choices. In Sect. 5, we apply the sparse GTM to a benchmark dataset from literature and a real-world classification problem. Some final remarks conclude this paper.

2 The Generative Model

In the following, we will describe a generative model, which is based on a low-dimensional parameterization.

We want to represent a d-dimensional density $p(\mathbf{t}) \geq 0, \mathbf{t} \in \mathbb{R}^d$, by a density that is intrinsically low-dimensional. To this end, we introduce a mapping

$$\mathbf{y} : [0,1]^L \to \mathbb{R}^d$$

with $L \ll d$ that connects the L-dimensional latent space $[0,1]^L$ and the data space $\mathbb{R}^d$. The generated density is then

$$q_{\mathbf{y},\beta}(\mathbf{t}) = \left(\frac{\beta}{2\pi}\right)^{d/2} \int_{[0,1]^L} \exp\left(-\frac{\beta}{2}\|\mathbf{y}(\mathbf{x}) - \mathbf{t}\|^2\right) \mathrm{d}\mathbf{x} \,. \tag{1}$$

It can be interpreted as the image of an L-dimensional uniform distribution under the mapping $\mathbf{y}$ with additional Gaussian noise, which is controlled by the parameter β, see Fig. 2 for an illustration. It is easy to see that $\int_{\mathbb{R}^d} q_{\mathbf{y},\beta}(\mathbf{t})\mathrm{d}\mathbf{t} = 1$, i.e. $q_{\mathbf{y},\beta}$ is a density in the d-dimensional data space.

The aim is now to choose a mapping $\mathbf{y}$ and an inverse variance $\beta \in \mathbb{R}_+$, such that the dissimilarity between $q_{\mathbf{y},\beta}$ and p is minimized. To be precise, for a given regularization term $\lambda S(\mathbf{y})$ and density $p(\mathbf{t})$, we want to minimize the regularized cross-entropy [16]

$$\begin{aligned} \mathscr{G}(\mathbf{y},\beta) &:= H(p, q_{\mathbf{y},\beta}) + \lambda S(\mathbf{y}) \\ &= -\int_{\mathbb{R}^d} p(\mathbf{t}) \log \int_{[0,1]^L} \exp\left(-\frac{\beta}{2}\|\mathbf{y}(\mathbf{x}) - \mathbf{t}\|^2\right) \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{t} - \frac{d}{2}\log\frac{\beta}{2\pi} + \lambda S(\mathbf{y}) \end{aligned} \tag{2}$$

in $\mathbf{y}$ and β. For the remainder of this paper, we assume $S(\mathbf{y}) = \sum_{k=1}^{d} \|y_k\|_H^2$, where $\mathbf{y}(\mathbf{x}) = (y_1(\mathbf{x}), \ldots, y_d(\mathbf{x}))$ and $\|\cdot\|_H = (\cdot,\cdot)_H^{1/2}$ denotes a given norm or seminorm in a prescribed Hilbert space H. This naturally requires the components of the vector-valued function $\mathbf{y}$ to be an element of H. For an in-depth discussion of

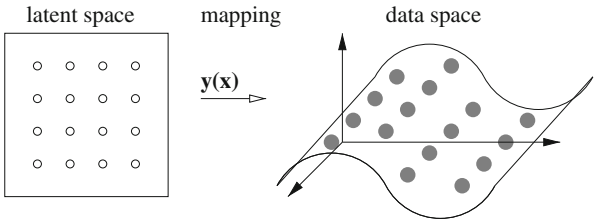

Fig. 2 The L-dimensional data space is mapped by $\mathbf{y}$ into the d-dimensional data space. There, the model assumes multivariate Gaussian noise with variance β^{-1}

the relation between regularization terms and associated function spaces, see [20]. A weak regularization with a too small λ or even $\lambda = 0$ leads to overfitting, i.e., the method models random noise instead of a meaningful underlying relationship between latent variables and the data set. A regularization that is too strong might prevent the method from discovering relevant features of the data. We recommend choosing the parameter λ for reconstruction and classification tasks by cross-validation techniques [7, 15].

3 Functional Minimization

Let us now show how the GTM functional $\mathscr{G}$ can be efficiently minimized even though it is nonlinear and nonconvex in $\mathbf{y}$ and β. It is important to note that the functional equals the logarithm of the partition function, and we can rearrange it to its free energy form for easier numerical treatment. First we define the posterior probabilities $R_{\mathbf{y},\beta} : \mathbb{R}^d \times [0,1]^L \to \mathbb{R}$ by

$$R_{\mathbf{y},\beta}(\mathbf{t},\mathbf{x}) := \frac{e^{-\frac{\beta}{2}\|\mathbf{y}(\mathbf{x})-\mathbf{t}\|^2}}{\int_{[0,1]^L} e^{-\frac{\beta}{2}\|\mathbf{y}(\mathbf{x}')-\mathbf{t}\|^2} \mathrm{d}\mathbf{x}'} .$$

Next, we introduce the functional

$$\begin{aligned} \mathscr{K}(\psi,\mathbf{y},\beta) := &\int_{\mathbb{R}^d} p(\mathbf{t}) \int_{[0,1]^L} \psi(\mathbf{t},\mathbf{x}) \log \psi(\mathbf{t},\mathbf{x}) \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{t} \\ &+ \frac{\beta}{2} \int_{\mathbb{R}^d} p(\mathbf{t}) \int_{[0,1]^L} \psi(\mathbf{t},\mathbf{x}) \|\mathbf{y}(\mathbf{x}) - \mathbf{t}\|^2 \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{t} - \frac{d}{2} \log \frac{\beta}{2\pi} + \lambda S(\mathbf{y}) . \end{aligned} \tag{3}$$

Here, for all $\mathbf{t} \in \mathbb{R}^d$ it must hold that $\psi(\mathbf{x},\mathbf{t})$ is a density in $\mathbf{x}$. Then, a lengthy, but otherwise simple calculation reveals that

$$\mathscr{K}(R_{\mathbf{y},\beta},\mathbf{y},\beta) = \mathscr{G}(\mathbf{y},\beta) \quad \text{for all} \quad \mathbf{y},\beta. \tag{4}$$

We now minimize $\mathscr{K}$ by successively minimizing with respect to its single parameters ψ, $\mathbf{y}$ and β. This is advantageous, because these subproblems are convex even though $\mathscr{G}$ is not.

The following three minimization steps have to be carried out in an outer iteration until we converge to a local minimum. Minimizing with respect to β yields

$$\arg\min_{\beta} \mathscr{K}(\psi,\mathbf{y},\beta) = \left(\frac{1}{d} \int_{\mathbb{R}^d} p(\mathbf{t}) \int_{[0,1]^L} \psi(\mathbf{t},\mathbf{x}) \|\mathbf{y}(\mathbf{x}) - \mathbf{t}\|^2 \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{t} \right)^{-1} . \tag{5}$$

The posterior probabilities $R_{\mathbf{y},\beta}$ minimize $\mathscr{K}$ w.r.t. ψ, i.e.

$$\arg\min_{\psi} \mathscr{K}(\psi, \mathbf{y}, \beta) = R_{\mathbf{y},\beta} \ , \tag{6}$$

which is analogous to statistical physics, where the Boltzmann-distribution minimizes the free energy [18]. In combination with (4), this step can be understood as a projection back into the permissible search space since

$$\mathscr{K}(\arg\min_{\psi} \mathscr{K}(\psi, \mathbf{y}, \beta), \mathbf{y}, \beta) = \mathscr{K}(R_{\mathbf{y},\beta}, \mathbf{y}, \beta) = \mathscr{G}(\mathbf{y}, \beta) \ .$$

To minimize $\mathscr{K}$ in $\mathbf{y}$-direction, we need to solve the quadratic regression type problem

$$\arg\min_{\mathbf{y}} \int_{\mathbb{R}^d} p(\mathbf{t}) \int_{[0,1]^L} \psi(\mathbf{t}, \mathbf{x}) \|\mathbf{y}(\mathbf{x}) - \mathbf{t}\|^2 \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{t} + \frac{2\lambda}{\beta} S(\mathbf{y}) \ . \tag{7}$$

4 Discretization of the Model

We now discretize the mapping $\mathbf{y}$ by M basis functions $\phi_j : [0,1]^L \to \mathbb{R}, j = 1, \ldots, M$, and obtain $\mathbf{y}_M(\mathbf{x}) = \mathbf{W}\boldsymbol{\phi}(\mathbf{x})$ with the coefficient matrix $\mathbf{W} \in \mathbb{R}^{d \times M}$ and $\boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_M(\mathbf{x}))^T$. The minimization of the $\mathscr{K}$-functional in $\mathbf{y}$-direction (7) then amounts to solving d decoupled systems of linear equations for $r = 1, \ldots, d$

$$\mathbf{A}\mathbf{w}_r = \mathbf{z}_r \tag{8}$$

with $\mathbf{w}_r = ((\mathbf{W})_{r1}, \ldots, (\mathbf{W})_{rM})^T$, $\mathbf{A} \in \mathbb{R}^{M \times M}$ and $\mathbf{z}_r \in \mathbb{R}^M$. The entries of the matrix $\mathbf{A}$ and the vectors $\mathbf{z}_r$ can be computed for $i, j = 1, \ldots, M$ by

$$(\mathbf{A})_{ij} = \int_{\mathbb{R}^d} p(\mathbf{t}) \int_{[0,1]^L} \psi(\mathbf{t}, \mathbf{x}) \phi_j(\mathbf{x}) \phi_i(\mathbf{x}) \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{t} + \frac{2\lambda}{\beta} (\phi_i, \phi_j)_H \quad \text{and} \tag{9}$$

$$(\mathbf{z}_r)_i = \int_{\mathbb{R}^d} p(\mathbf{t}) \int_{[0,1]^L} \psi(\mathbf{t}, \mathbf{x}) (\mathbf{t})_r \phi_i(\mathbf{x}) \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{t} \ , \qquad r = 1, \ldots, d. \tag{10}$$

Note here that the derivation of our model in Sect. 2 started with the explicit knowledge of the continuous density $p(\mathbf{t})$. This is however in general not the case in most practical settings. There, rather an empirical density $p_N^{\text{emp}}(\mathbf{t})$ based on N data points $(\mathbf{t}_n)_{n=1}^N$ is given instead. Therefore, for the remainder of this paper, we furthermore replace the continuous density $p(\mathbf{t})$ by a sum of Dirac-delta-functions $p_N^{\text{emp}}(\mathbf{t}) = \frac{1}{N} \sum_{n=1}^N \delta_{\mathbf{t}_n}(\mathbf{t})$. Then, the $\mathrm{d}\mathbf{t}$-integrals in (9) and (10) get replaced by sums, which corresponds to discretization by sampling.

4.1 Original GTM

Now, two further discretization steps can be taken. First, we choose L-variate Gaussians as the specific functions in the basis function vector $\phi : [0,1]^L \to \mathbb{R}^M$. Their centers lie on a uniform mesh in the L-dimensional latent space with mesh width h_1. Then $M = \mathcal{O}(h_1^{-L})$ and $h_1 = \mathcal{O}(M^{-\frac{1}{L}})$, respectively. Secondly, we choose a tensorized rectangle rule on a uniform mesh with width h_2 for the numerical quadrature of the d$\mathbf{x}$-integrals in (9) and (10), which results in $K = \mathcal{O}(h_2^{-L})$ quadrature points $(\mathbf{x}_i)_{m=1}^K$. This is equivalent to assuming a grid-based latent space distribution, as it is done in [4].

We obtain the resulting systems of linear equations (8), where now

$$(\mathbf{A})_{ij} = \frac{1}{NK} \sum_{n=1}^{N} \sum_{m=1}^{K} \psi(\mathbf{t}_n, \mathbf{x}_m) \phi_j(\mathbf{x}_m) \phi_i(\mathbf{x}_m) + \frac{2\lambda}{\beta} (\phi_i, \phi_j)_H \quad \text{and} \tag{11}$$

$$(\mathbf{z}_r)_i = \frac{1}{NK} \sum_{n=1}^{N} \sum_{m=1}^{K} \psi(\mathbf{t}_n, \mathbf{x}_m) (\mathbf{t}_n)_r \phi_i(\mathbf{x}_m) \,, \qquad r = 1, \ldots, d, \tag{12}$$

for $i, j = 1, \ldots, M$.

Note that in our successive minimization of $\mathcal{K}$, see Sect. 3, the minimization (6) with respect to ψ equals the E-Step and the minimization steps (5) and (7) with respect to β and $\mathbf{y}$ equal the M-Step of the well-known Expectation Maximization-algorithm [8]. In all steps, the discretized versions of $\mathbf{y}$ and the d$\mathbf{x}$-integrals now need to be employed. Altogether, we finally obtain the GTM [4], or, the other way around, we see that the original GTM is a special discretization of our generative model (1).

Note furthermore that the M degrees of freedom in the discretization and the K function evaluations for numerical quadrature have both an exponential dependence on the embedding dimension L. This severely limits the GTM to the cases $L \leq 3$. To overcome this issue, we will choose some other type of discretization of our generative model in the following.

4.2 Sparse GTM

We now suggest to use a sparse grid discretization [6] for the components of the mapping $\mathbf{y}$ instead of a uniform, full mesh. We denote the resulting numerical method as the sparse GTM. To explain our new approach in detail, let us first consider a one-dimensional level-wise sequence of conventional sets of piecewise linear basis functions on the interval $[0,1]$. There, the space V_l on level $l \geq 0$ contains $n_l = 2^l + 1$ hat functions $\phi_{l,i} : [0,1] \to \mathbb{R}$

$$\phi_{l,i}(x) = \max(1 - 2^l \, |x - x_{l,i}| \, , 0) \,,$$

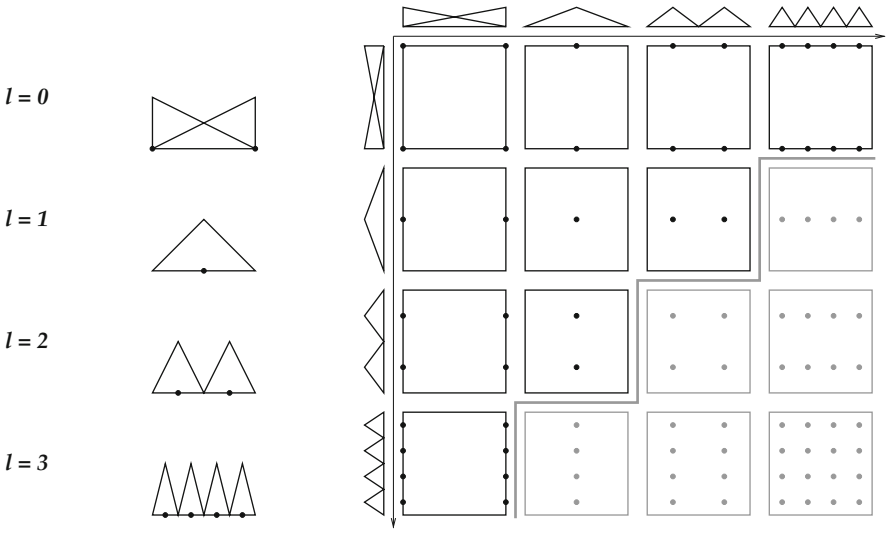

Fig. 3 The first four hierarchical surplus spaces of the one-dimensional hierarchical basis (*left*). Two-dimensional tensorization and the sparse subspace (*right*)

which are centered at the points of an equidistant mesh $x_{l,i} = 2^{-l}i, i = 0, \ldots, n_l - 1$. Next, we consider the hierarchical surplus spaces W_l, where $V_{l+1} = V_l \oplus W_{l+1}$, see also the left-hand side of Fig. 3. They can be easily constructed by

$$W_l = \text{span}\{\phi_{l,i} : i \in \xi_l\} \quad \text{with} \quad \xi_l := \begin{cases} \{0, 1\} & \text{for } l = 0 \\ \{i \text{ odd}, 1 \leq i \leq 2^l - 1\} & \text{else.} \end{cases}$$

With the multi-indices $\mathbf{l} = (l_1, \ldots, l_L) \in \mathbb{N}^L$, $\mathbf{i} = (i_1, \ldots, i_L) \in \mathbb{N}^L$, the d-variate functions $\phi_{\mathbf{l},\mathbf{i}}(\mathbf{x}) = \phi_{l_1,i_1}(x_1) \cdots \phi_{l_L,i_L}(x_L)$ and the Cartesian products $\xi_{\mathbf{l}} := \times_{s=1}^{L} \xi_{l_s}$, we obtain L-dimensional spaces $W_{\mathbf{l}} = \text{span}\{\phi_{\mathbf{l},\mathbf{i}} : \mathbf{i} \in \xi_{\mathbf{l}}\}$. Then,

$$V_J^{(\infty)} := \bigoplus_{|\mathbf{l}|_\infty \leq J} W_{\mathbf{l}} = \bigotimes_{s=1}^{L} \bigoplus_{l_s=0}^{J} W_{l_i} = \bigotimes_{s=1}^{L} V_J$$

resembles just a normal isotropic full grid (FG) space up to level J, while

$$V_J^{(1)} := \bigoplus_{|\mathbf{l}|_1 \leq J+d-1} W_{\mathbf{l}} \tag{13}$$

denotes the sparse grid (SG) space[1] on level J. The former has $M^{\mathrm{FG}} = \mathcal{O}(2^{LJ})$ degrees of freedom, while the latter has only $M^{\mathrm{SG}} = \mathcal{O}(J^{L-1}2^J)$ degrees of freedom. However, under the assumption of bounded mixed derivatives, both discretizations have essentially the same L_2-error convergence rate, see [6, 9] for further analysis and implementational issues. The use of this kind of discretization for every component of the vector-valued mapping $\mathbf{y}$, i.e. $\mathbf{y}^{\mathrm{SG}} = (y_1^{\mathrm{SG}}, \ldots, y_d^{\mathrm{SG}})$ with $y_r^{SG} \in V_J^{(1)}, r = 1, \ldots, d$, then yields a sparse GTM.

The corresponding L-dimensional integration problems (9) and (10) for setting up the associated systems of linear equations (8) are approximated by evaluation points $\mathbf{x}_m$ with fixed weights γ_m for $m = 1, \ldots, K$. Here, methods like Quasi Monte-Carlo or sparse grid quadrature [11] can be used. Then, K does not exhibit the 'curse of dimensionality' with respect to L.

We obtain the resulting systems of linear equations (8), where now

$$(\mathbf{A})_{\mathbf{l},\mathbf{i},\mathbf{k},\mathbf{j}} = \frac{1}{N}\sum_{n=1}^{N}\sum_{m=1}^{K}\gamma_m\psi(\mathbf{t}_n,\mathbf{x}_m)\phi_{\mathbf{l},\mathbf{i}}(\mathbf{x}_m)\phi_{\mathbf{j},\mathbf{k}}(\mathbf{x}_m) + \frac{2\lambda}{\beta}(\phi_{\mathbf{l},\mathbf{i}},\phi_{\mathbf{k},\mathbf{j}})_H \quad \text{and} \tag{14}$$

$$(\mathbf{z}_r)_{\mathbf{l},\mathbf{i}} = \frac{1}{N}\sum_{n=1}^{N}\sum_{m=1}^{K}\gamma_m\psi(\mathbf{t}_n,\mathbf{x}_m)(\mathbf{t}_n)_r\phi_{\mathbf{l},\mathbf{i}}(\mathbf{x}_m)\,, \qquad r = 1,\ldots,d, \tag{15}$$

with $|\mathbf{l}|_1, |\mathbf{k}|_1 \le J + d - 1, \mathbf{i} \in \xi_{\mathbf{l}}, \mathbf{j} \in \xi_{\mathbf{k}}$.

When we minimize the functional $\mathscr{K}$ in $\mathbf{y}$-direction the systems (8) have to be solved. As recommended in [4], we use a direct method. An LU factorization of the matrix $\mathbf{A}$ costs $\mathcal{O}\big((M^{SG})^3\big)$. Then, the forward and backward substitution steps for d different right-hand sides of (8) cost $\mathcal{O}\big(d \cdot (M^{SG})^2\big)$. For high-dimensional data sets with $d > M^{SG}$, these steps can be more relevant cost-wise than the initial factorization of $\mathbf{A}$.

It is also possible to solve the system (8) for each right-hand side by an iterative method. Then the costs are $\mathcal{O}(d \cdot \#it \cdot X)$, where $\#it$ denotes the number of necessary iteration steps and X is the cost of one matrix-vector multiplication. Typically, the unidirectional principle [2, 5] is used for the fast multiplication with sparse grid operator matrices, but this algorithm is not applicable here since the function ψ in (14) does not allow a product representation in $\mathbf{x}$. However, in contrast to the Original GTM from Sect. 4.1, our sparse GTM results in a somewhat sparse matrix $\mathbf{A}$. This can be exploited in the matrix vector multiplication of $\mathbf{A}$. Note that the regularization term $\frac{2\lambda}{\beta}(\cdot,\cdot)_H$ prevents the matrix $\mathbf{A}$ from being severely ill-conditioned. Here, however, keeping $\#it$ low and bounded independently of the discretization level J is a matter of preconditioning the matrix (14), which is nontrivial and future work. Since we presently cannot guarantee that the costs

[1] We can replace $|\mathbf{l}|_1$ by $|\mathbf{l}|_1 + |\{s : l_s = 0\}|$ in (13), which leads to a slightly different treatment of boundary functions, but has otherwise the same asymptotic properties, see [9].

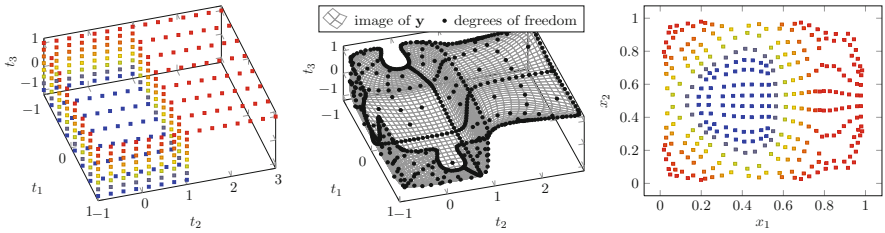

Fig. 4 The three-dimensional 'open box' (*left*), a sparse GTM fitted to this dataset (*middle*) and the two-dimensional projection of the data points (*right*)

$\mathcal{O}(d \cdot \#it \cdot X)$ are lower than $\mathcal{O}\left(d \cdot (M^{SG})^2\right)$ for the direct method in high-dimensions, we decided to stick with the LU factorization for now.

To demonstrate the nonlinear quality of the method, we apply the sparse GTM to the 'open box' benchmark dataset [17] in Fig. 4. We see a reasonable unfolding of the box in the two-dimensional embedding, which would not be possible with linear methods, like e.g. a conventional PCA.

5 Numerical Experiments

In this section, we will now present the results for the sparse GTM for some problems from dimensionality reduction and data classification.

5.1 Dimensionality Reduction

On the left-hand side of Fig. 5, we present a toy example with data points stemming from a wave-shaped manifold. Since we here have a sufficiently large amount of data points, we need no regularization term. We measure the GTM functional value $\mathscr{G}(\mathbf{y}, \beta)$, see (2), after 5 minimization cycles for $\mathscr{K}$, see (3). On the right-hand side, we see that the sparse GTM achieves about the same reduction in the GTM functional value with substantially less degrees of freedom than the GTM based on a full grid.

Next, we consider a real-world problem. Figure 6 shows a three-dimensional projection of a 12-dimensional data set. It consists of 1,000 data points with diagnostic measurements of oil flows along a multi-phase pipeline. The three different class types in the plot represent stratified, annular and homogeneous multi-phase configurations, compare [3] for further details. In [4], it was shown how a two-dimensional embedding of the data with the GTM gives an improved separation of the clusters compared to the embedding with the PCA. We now run this experiment with a sparse GTM with $L = 2$ and $L = 3$, discretization level

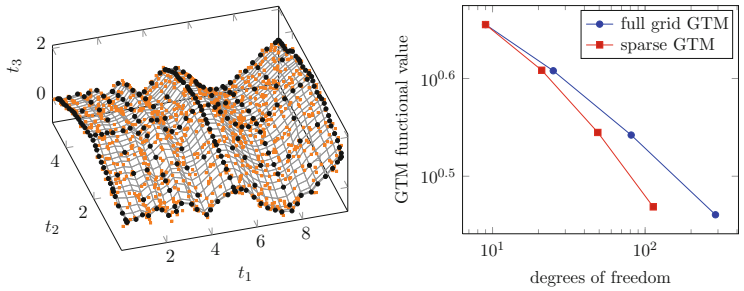

Fig. 5 Reduction in the GTM functional value with respect to the degrees of freedom per **y**-component for a GTM with a full grid discretization and the sparse GTM

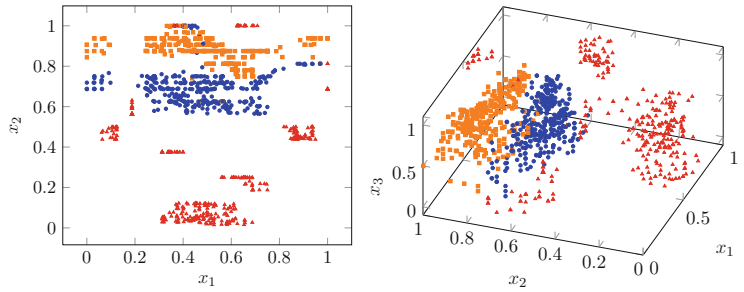

Fig. 6 Embedding of a 12-dimensional data set with three class labels by the sparse GTM in two dimensions (*left*) and three dimensions (*right*)

$J = 4$, H^1_{mix}-seminorm regularization and $\lambda = 4.0 \times 10^{-3}$. We see that the three-dimensional latent space offers an even more detailed picture of the data than the two-dimensional embedding and a slightly better separation of the different clusters.

5.2 Classification

We now use the sparse GTM for classification. To this end, we append a class variable $c_n \in \{-1, 1\}$ to the data points by

$$\mathbf{t}'_n := ((\mathbf{t}_n)_1, \ldots, (\mathbf{t}_n)_d, c_n)^T \quad \text{for} \quad n = 1, \ldots, N \ . \tag{16}$$

We first use the sparse GTM to fit the mapping $\mathbf{y}$ and the inverse variance β to these points. Then, we can classify new data points with help of the density $q_{\mathbf{y},\beta}$ of (1) by

$$c(\mathbf{t}) := \begin{cases} 1 & \text{if } q_{\mathbf{y},\beta}(t_1, \ldots, t_d, 1) \geq q_{\mathbf{y},\beta}(t_1, \ldots, t_d, -1) \\ -1 & \text{else.} \end{cases}$$

We apply this technique to 'Connectionist Bench (Sonar, Mines vs. Rocks)', a real-world data set from the UCI Machine Learning Repository [1]. It consists of approximately 200 measurements with 60 dimensions and two class labels.

In [12], this data was randomly split into two parts. One part was used to train various neuronal networks, the other one was used to measure the quality of the model. The best neuronal network achieved an average classification rate of 84.7 %.

We use our sparse GTM with latent space dimensions $L = 2$ and $L = 3$ and a regularization term based on the H^1_{mix}-seminorm. We achieve classification rates between 72.0 and 84.6 % already for $L = 2$, depending on the regularization parameter λ and the discretization level J. For $L = 3$, $J = 3$ and $\lambda = 1.0 \times 10^{-4}$, we even reach a classification rate of 85.6 %, which clearly shows the potential of our new approach.

6 Conclusions

We presented a generative model that can be used for dimensionality reduction and classification of high-dimensional data. For a certain choice of discretization involving uniform grids, we obtained the original generative topographic mapping from [4]. Using a sparse grid discretization for the mapping, we obtained our new sparse GTM. It gives about the same quality with less degrees of freedom. Moreover, it has the perspective to overcome complexity issues of the grid-like structures, which limit the conventional GTM to a low number of latent space dimensions. For example, in dimension $L = 4$ and discretization level $J = 5$ the sparse grid approach with index set $\{\mathbf{l} : |\mathbf{l}|_1 + |\{s : l_s = 0\}| \leq J + d - 1\}$ has 7,681 degrees of freedom, which is still treatable using a direct solver, whereas the full grid already has 1,185,921 degrees of freedom. For dimensions like $L = 10$ the situation is as follows: Full grids with $L = 10$ and $J = 4$ have $2.0 \cdot 10^{12}$ degrees of freedom ($5.8 \cdot 10^{11}$ inner functions and $1.4 \cdot 10^{12}$ boundary functions), which is clearly beyond the capabilities of current computers. Sparse grids have $1.1 \cdot 10^7$ degrees of freedom, of which only 2,001 are inner functions and 10,817,088 are boundary functions. Of course, this is still too much for a direct solver, but now only the number of boundary functions poses a bottleneck. Modified boundary functions with improved properties can be found in [9, 19], so there is some hope to treat higher dimensional latent spaces. Furthermore, note that the runtime complexity depends only linearly on the data space dimension d and the number of data points N. This makes the sparse GTM a suitable tool for high-dimensional data sets. For further experiments and results, cf. [13, 14].

References

1. Bache, K., Lichman, M.: UCI Machine Learning Repository. http://archive.ics.uci.edu/ml (2012)
2. Balder, R., Zenger, C.: The solution of multidimensional real Helmholtz equations on sparse grids. SIAM J. Sci. Comput. **17**, 631–646 (1996)
3. Bishop, C., James, G.: Analysis of multiphase flows using dual-energy gamma densitometry and neural networks. Nucl. Instrum. Methods Phys. Res. Sect. A: Accel. Spectrom. Detect. Assoc. Equip. **327**(2–3), 580–593 (1993)
4. Bishop, C., Svensen, M., Williams, C.: GTM: the generative topographic mapping. Neural Comput. **10**(1), 215–234 (1998)
5. Bungartz, H.: Dünne Gitter und deren Anwendung bei der adaptiven Lösung der dreidimensionalen Poisson-Gleichung. Dissertation, Fakultät für Informatik, Technische Universität München (1992)
6. Bungartz, H., Griebel, M.: Sparse grids. Acta Numer. **13**, 1–123 (2004)
7. Craven, P., Wahba, G.: Smoothing noisy data with spline functions. Numer. Math. **31**(4), 377–403 (1978)
8. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. **39**, 1–38 (1977)
9. Feuersänger, C.: Sparse Grid Methods for Higher Dimensional Approximation. Südwestdeutscher Verlag für Hochschulschriften AG & Company KG, Saarbrücken (2010)
10. Feuersänger, C., Griebel, M.: Principal manifold learning by sparse grids. Computing **85**(4), 267–299 (2009)
11. Gerstner, T., Griebel, M.: Dimension–adaptive tensor–product quadrature. Computing, **71**(1), 65–87 (2003)
12. Gorman, R., Sejnowski, T.: Analysis of hidden units in a layered network trained to classify sonar targets. Neural Netw. **1**, 75 (1988)
13. Griebel, M., Hullmann, A.: Dimensionality reduction of high-dimensional data with a nonlinear principal component aligned generative topographic mapping. SIAM J. Sci. Comput. **36**(3), A1027–A1047 (2014)
14. Hullmann, A.: Schnelle varianten des generative topographic mapping. Diploma thesis, Institute for Numerical Simulation, University of Bonn (2009)
15. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence – Volume 2 (IJCAI'95), San Francisco, pp. 1137–1143. Morgan Kaufmann (1995)
16. Kullback, S.: Information Theory and Statistics. Wiley, New York (1959)
17. Lee, J., Verleysen, M.: Nonlinear Dimensionality Reduction. Springer, New York/London (2007)
18. Neal, R., Hinton, G.: A view of the EM algorithm that justifies incremental, sparse, and other variants. In: Learning in Graphical Models, pp. 355–368. Kluwer Academic, Dordrecht/Boston (1998)
19. Pflüger, D., Peherstorfer, B., Bungartz, H.: Spatially adaptive sparse grids for high-dimensional data-driven problems. J. Complex. **26**(5), 508–522 (2010)
20. Schölkopf, B., Smola, A.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT, Cambridge (2001)