Neurocomputing 147 (2015) 60–70

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Self-organization and missing values in SOM and GTM

T. Vatanen [a,b,*], M. Osmala [a], T. Raiko [a], K. Lagus [a], M. Sysi-Aho [c], M. Orešič [d], T. Honkela [e], H. Lähdesmäki [a]

[a] Aalto University School of Science, Department of Information and Computer Science, P.O. Box 15400, FI-00076 Aalto, Espoo, Finland
[b] The Broad Institute of MIT and Harvard, 7 Cambridge Center, Cambridge, MA 02142, USA
[c] VTT Technical Research Centre of Finland, Espoo FI-02044, Finland
[d] Steno Diabetes Center, 2820 Gentofte, Denmark
[e] University of Helsinki, Department of modern languages, P.O. Box 24, FI-00014 Helsinki, Finland

ARTICLE INFO

Article history:
Received 12 April 2013
Received in revised form
18 January 2014
Accepted 19 February 2014
Available online 10 June 2014

Keywords:
Self-organizing map
Generative topographic mapping
Self-organization
Missing data
Data visualization

ABSTRACT

In this paper, we study fundamental properties of the Self-Organizing Map (SOM) and the Generative Topographic Mapping (GTM), ramifications of the initialization of the algorithms and properties of the algorithms in the presence of missing data. We show that the commonly used principal component analysis (PCA) initialization of the GTM does not guarantee good learning results with high-dimensional data. Initializing the GTM with the SOM is shown to yield improvements in self-organization with three high-dimensional data sets: commonly used MNIST and ISOLET data sets and epigenomic ENCODE data set. We also propose a revision of handling missing data to the batch SOM algorithm called the Imputation SOM and show that the new algorithm is more robust in the presence of missing data. We benchmark the performance of the topographic mappings in the missing value imputation task and conclude that there are better methods for this particular task. Finally, we announce a revised version of the SOM Toolbox for Matlab with added GTM functionality.

1. Introduction

Topographic mappings, such as the Self-Organizing Map (SOM) [1,2] and the Generative Topographic Mapping (GTM) [3], are useful tools in inspecting and visualizing high-dimensional data. The SOM was originally inspired by neuroscientific research on cortical organization, and the algorithm models the basic principles of the organization process at a general level. The SOM has been shown to serve its purpose well, especially when the faithfulness (precision) of the mapping from a high-dimensional space is considered [4]. In practice, the SOM has proved to be a robust approach tested in thousands of different applications [5–7]. The GTM was inspired by the SOM algorithm, while operating in the probabilistic framework which provides well-founded regularization and model comparison [3]. In this paper, we show that both methods have their own strengths over the other and the methods may even benefit each other. We investigate applicability of the methods in high-dimensional, real-life data sets and provide methodological improvements in the presence of missing data.

Visualization of biological and life science data is an important task in the rapidly evolving field of bioinformatics. New kinds of measurement techniques and visualization methods appear at a constant pace (see, e.g., www.vizbi.org), but many practitioners still turn to rudimentary methods, such as hierarchical clustering and heatmaps. Recently, [8,9] have used the SOM in order to cluster genome segmentation regions based on different assay signal characteristics gathered in the Encyclopedia of DNA Elements (ENCODE) project. The SOM is particularly well suited for many visualization tasks on biological data because of its computational simplicity and relatively loose prior assumptions on the data. As we will show, the Gaussian noise model assumed in the GTM is a critical constraint for many high dimensional data sets. Furthermore, a SOM-type mapping has also been adapted to arbitrary data for which the mutual pairwise distances are defined [10] allowing one to compute SOMs only based on pairwise distance matrices. A comprehensive review of visualization methods for large data sets can be found, e.g., in [11].

Missing data are a common problem in many data-dependent fields ranging from social sciences to economics and from political research to entertainment industry. In fields where conducting surveys or polls is commonplace, missing data occurs, for instance,

* Corresponding author at: Aalto University School of Science, Department of Information and Computer Science, P.O. Box 15400, FI-00076 Aalto, Espoo, Finland.
E-mail addresses: tommi.vatanen@aalto.fi (T. Vatanen), maria.osmala@aalto.fi (M. Osmala), tapani.raiko@aalto.fi (T. Raiko), krista.lagus@aalto.fi (K. Lagus), marko.sysi-aho@vtt.fi (M. Sysi-Aho), matej.oresic@gmail.com (M. Orešič), timo.honkela@aalto.fi (T. Honkela), harri.lahdesmaki@aalto.fi (H. Lähdesmäki).

http://dx.doi.org/10.1016/j.neucom.2014.02.061
0925-2312/

when people refuse to answer to specific questions or some people cannot be contacted. In the movie business, predicting customer preferences is literally a million dollar quest. The Netflix Prize (see, e.g., [12]) was an open competition to devise the best recommendation system to predict user ratings for films based on previous ratings. In the second part of this paper, we present a revision to the batch SOM algorithm, called the Imputation SOM, which is shown to improve the behavior of the SOM algorithm in the presence of missing data.

This paper is organized as follows. Sections 2 and 3 introduce the SOM and the GTM models, respectively. In Section 4, the properties of the models are compared in terms of self-organization and convergence. We show that using the SOM for initializing the GTM may improve the learning results in some cases. Section 5 explains the treatment of missing values in the GTM and adapts the same principled way into the SOM. Performance of the algorithms is compared in a missing value imputation task. Finally, the results and possible future work are discussed in Section 6.

In all the experiments, the SOM Toolbox [13] and Netlab [14] software packages are used. The GTM scripts in Netlab are revised to handle data with missing values and a sequential training algorithm is contributed. Also, an issue of small probabilities being rounded to zero due to insufficient floating point precision was solved. Finally, we announce a revised version of the SOM Toolbox which incorporates GTM functionality. An up-to-date version of the SOM Toolbox is available at

http://research.ics.aalto.fi/software/somtoolbox

2. Self-organizing map

The self-organizing map (SOM) [2] discovers some underlying structure in data using K map units, prototypes or reference vectors $\{\boldsymbol{m}_i\}$. For the prototypes, explicit neighborhood relations have been defined. The classical sequential SOM algorithm proceeds by processing one data point $\boldsymbol{x}(t)$ at a time. Euclidean, or any other suitable distance measure is used to find the best-matching unit given by $\boldsymbol{m}_{c(\boldsymbol{x}(t))} = \arg\min_i \|\boldsymbol{x}(t) - \boldsymbol{m}_i\|$. The reference vectors are then updated using the update rule $\boldsymbol{m}_i(t+1) = \boldsymbol{m}_i(t) + h_{ci}(t)(\boldsymbol{x}(t) - \boldsymbol{m}_i(t))$, where an explicit neighborhood function $h_{ci} = \alpha(t) \cdot \exp\{-\|r_c - r_i\|^2 / 2\sigma^2(t)\}$ is used in order to obtain topological mapping. In the neighborhood function, $\|r_c - r_i\|$ is the distance between the best-matching unit r_c and unit i in the array, $0 < \alpha(t) < 1$ is scalar-valued learning-rate factor and $\sigma(t)$ is the width of the neighborhood kernel.

2.1. Batch SOM

In the Batch SOM, the reference vectors are updated using all data (or a mini-batch, a part of the data) at once and weighted accordingly. The batch update rule is

$$\boldsymbol{m}_i = \frac{\sum_n h_{ni} \boldsymbol{x}_n}{\sum_j h_{ni}}, \tag{1}$$

where the index n runs over the data vectors whose best-matching units satisfy $h_{ni} > 0$, that is, all data points up to the range of the neighborhood function are taken into account.

2.2. Quality and size of the SOM

Selecting the size of the array of map units in the SOM is a subtle task. Previously many solutions, such as hierarchical [15] and growing maps [16,17], have been proposed to tackle this issue. The question of the size can be approached from the point of view of different quality measures. Two most commonly used error measures are the *quantization error* and the *topological error* [2]. The former measures the mean of the reconstruction errors $\|\boldsymbol{x} - \boldsymbol{m}_c\|$ when each data point used in learning is replaced by its best-matching unit. The latter measures the proportion of data points for which the two nearest map units are not neighbors in the array topology. As the number of map units increases, quantization error decreases and topological error tends to increase. Hence, there is no straightforward way of choosing the number of map units based on the measures above. Topographic preservation has been studied in detail, e.g., in [18,4,19]. In this work, we use an error measure proposed in [20]. This *combined error* is a sum of the quantization error and the distance from the best-matching unit to the second-best-matching unit of each data vector along the shortest path following the neighborhood relations. More formally, using the notation in [20], the distance metric used is given by

$$d(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{m}_c\| + \sum_{k=0}^{K_{c',i}-1} \|\boldsymbol{m}_{I_i(k)} - \boldsymbol{m}_{I_i(k+1)}\|, \tag{2}$$

where the first term is the quantization error and the second term computes the distance between the BMU and the second-best-matching unit along the map grid. Given a training data $\{\boldsymbol{x}_n\}_{n=1}^N$, combined error is given by

$$E_C = \sum_{n=1}^{N} d(\boldsymbol{x}_n), \tag{3}$$

where n runs over all the data vectors. We have added this feature in the SOM Toolbox `som_quality` function and demonstrate its use in the experiments.

3. Generative topographic mapping

The Generative Topographic Mapping (GTM) [3,21] is a nonlinear latent variable model which was proposed as a probabilistic alternative to the SOM. Loosely speaking, it extends the SOM in a similar manner as Gaussian mixture model extends k-means clustering. This is achieved by working in a probabilistic framework where data vectors have posterior probabilities given a map unit. Hence, instead of possessing only one best-matching unit, each data vector contributes to many reference vectors directly.

The GTM can be seen consisting of three parts: (1) discrete set of points in usually one or two-dimensional latent space, (2) nonlinear mapping, usually radial basis function (RBF) network, between the latent space and the data space, and (3) a Gaussian noise model in the data space such that the resulting model is a constrained mixture of Gaussians. In this paper, latent points $\{\boldsymbol{u}_i\}$, which are arranged in a regular grid, are mapped to the data space using M fixed radial basis functions $\boldsymbol{\phi}(\boldsymbol{u}_i) = \{\phi_j(\boldsymbol{u}_i)\}$, where $\phi_j(\boldsymbol{u}_i) = \exp\{-\|\boldsymbol{c}_j - \boldsymbol{u}_i\|/\sigma^2\}$, σ is the width parameter of the RBFs, $\{\boldsymbol{c}_j\}$ are the RBF centers and $j = 1,\ldots,M$. The number of RBFs, M, is a free parameter which has to be chosen by the experimenter. The radius of the RBFs is chosen according to $\sigma = d_{\max}/\sqrt{M}$, where $d_{\max}$ is the maximum distance between two RBF centers (this is a textbook choice for RBF networks; see, e.g. [22]). The node locations in latent space, $\boldsymbol{u}_i$, define a corresponding set of reference vectors $\boldsymbol{m}_i = \boldsymbol{W}\boldsymbol{\phi}(\boldsymbol{u}_i)$ in the data space, where $\boldsymbol{W}$ is a weight matrix defining the mapping from the latent space to the data space. In this work, each reference vector $\boldsymbol{m}_i$ serves as a center of an isotropic Gaussian distribution in the data space

$$p(\boldsymbol{x}|\boldsymbol{m}_i) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\frac{\beta}{2}\|\boldsymbol{m}_i - \boldsymbol{x}\|^2\right\}, \tag{4}$$

where β is the precision or inverse variance. The Gaussian distribution above also represents a noise model accounting for

the fact that the data will not be confined precisely to the lower-dimensional manifold in the data space. More general noise models have been proposed [21].

The probability density function of the GTM is obtained by summing over the Gaussian components yielding

$$p(\boldsymbol{x}|\boldsymbol{W},\beta) = \sum_{i=1}^{K} P(\boldsymbol{m}_i)p(\boldsymbol{x}|\boldsymbol{m}_i) = \sum_{i=1}^{K} \frac{1}{K}\left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\frac{\beta}{2}\|\boldsymbol{m}_i - \boldsymbol{x}\|^2\right\}, \tag{5}$$

where K is the total number grid points in the latent space, or map units in the SOM terminology, and the prior probabilities $P(\boldsymbol{m}_i)$ are given equal probabilities $1/K$.

The GTM represents a parametric probability density model, with parameters $\boldsymbol{W}$ and β, and it can be fitted to a data set $\{\boldsymbol{x}_n\}$ by maximum likelihood. The log-likelihood function of the GTM is given by

$$\log(\mathcal{L}(\boldsymbol{W},\beta)) = \sum_{n=1}^{N} \log\ p(\boldsymbol{x}_n|\boldsymbol{W},\beta), \tag{6}$$

where $p(\boldsymbol{x}_n|\boldsymbol{W},\beta)$ is given by (5) and independent, identically distributed (iid) data is assumed. We solved numerical issues in Netlab GTM implementation [14] by computing log-likelihood as follows:

$$\log(\mathcal{L}(\boldsymbol{W},\beta)) = \log p(\boldsymbol{x})_{\max} + \log\left(\sum_{n=1}^{N} \exp(\log p(\boldsymbol{x}_n) - \log p(\boldsymbol{x})_{\max})\right), \tag{7}$$

where $p(\boldsymbol{x})_{\max} = \max_n p(\boldsymbol{x}_n|\boldsymbol{W},\beta)$. In the experiments, negative log-likelihood-per-sample, given by

$$E_{\mathrm{GTM}} = -\log(\mathcal{L}(\boldsymbol{W},\beta))/N, \tag{8}$$

is used as a training error. The error can be minimized using the EM algorithm or alternatively any standard non-linear optimization technique.

4. Self-organization and convergence

Both the GTM and the batch SOM require careful initialization in order to self-organize [23,24]. For both algorithms, the common choice is to initialize according to the plane spanned by the two main principal components of the data. In the batch SOM, the neighborhood is annealed during the learning which decreases the rigidness of the map. The most important advantages of the batch SOM when compared to the classical sequential SOM are quick convergence and computational simplicity [24].

As we will show, initializing the GTM using PCA does not always lead to appropriate results. Instead, we propose using the batch SOM for initializing the GTM. In the SOM initialization, using few epochs of 'rough training' with wide neighborhood will suffice. Next, $\boldsymbol{W}$ can be determined by minimizing the error function:

$$E_{\mathrm{init}} = \frac{1}{2}\sum_i \|\boldsymbol{W}\boldsymbol{\phi}(\boldsymbol{u}_i) - \boldsymbol{m}_i^{\mathrm{SOM}}\|, \tag{9}$$

where $\boldsymbol{m}_i^{\mathrm{SOM}}$ are the reference vectors of the initializing SOM. The initializing SOM can, in turn, be initialized using PCA, which makes the whole process deterministic.

Differences between the SOM and the GTM, and efficacy of the SOM initialization are demonstrated using several high-dimensional data sets. These data sets were chosen to demonstrate cases where the different initialization methods make a difference to the resulting mapping.

In the first example, we use the ISOLET data set from the UCI machine learning repository [25]. The data contains 7797 spoken samples of the letters of the alphabet. The 617 features are described in [26] and include, e.g., spectral coefficients, contour features and sonorant features. The data was normalized to have zero mean and unit variance. The class labels, i.e., the letter identifiers, were not used in training of the maps.

The appropriate model complexity for the GTM, i.e, the number of RBFs and latent points, can be chosen, e.g., by cross-validating the negative log-likelihood. Using cross-validation for the ISOLET data, a suitable number of RBFs was found to be 400 (20×20) and a suitable number of map units 4004 (77×52). We used the same data to demonstrate effects of the SOM initialization already in [27]. However, after improving the Netlab GTM functions to tackle numerical issues (see Eq. (7)), the results obtained are significantly different.

Fig. 1 shows two GTM visualization of the ISOLET data. In Fig. 1 (a), PCA initialization was used, whereas in Fig. 1(b) the GTM was initialized using the SOM. The map initialized using the SOM has better cluster structure where most of the letters form distinct clusters. Furthermore, similar sounding letters are mapped close to each other. On the left side of the map, the data is more ambiguous and different letters, such as B, D, E, P, and V, are mixed together.

The GTM in Fig. 1(a), initialized with PCA, also has some cluster structure, but most of the letters are spread over wider area compared to the Fig. 1(b). However, when comparing to the results in [27], where the GTM with PCA initialization was not able to learn any interesting structure in the ISOLET data, there is a significant improvement caused by our implementations of

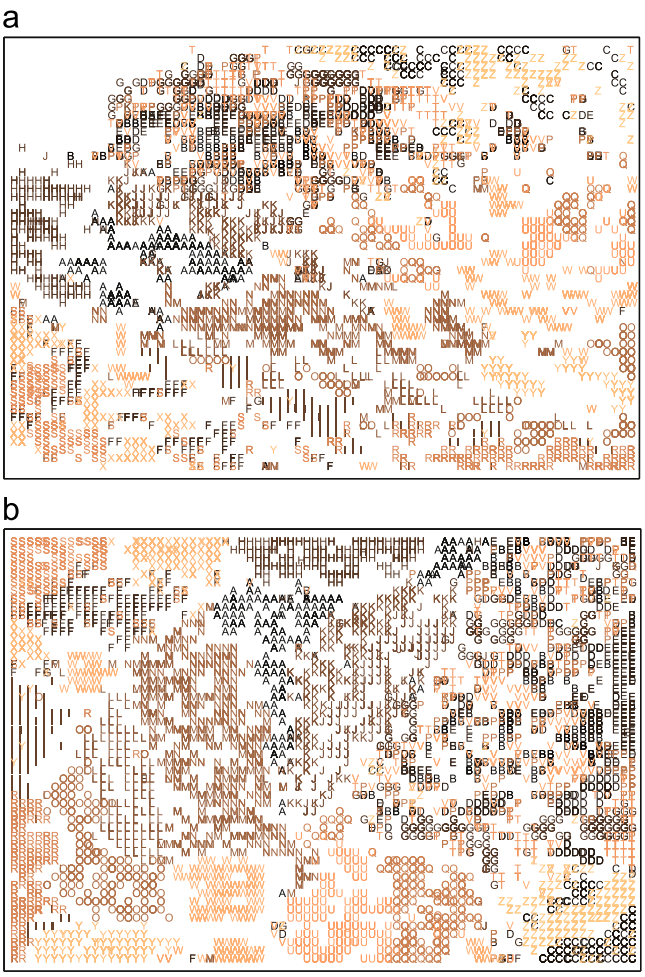


Fig. 1. A GTM of the ISOLET data with 4004 (77×52) map units and 400 (20×20) RBFs initialized using (a) PCA and (b) the SOM. Bootstrapped mean training error, E_{GTM} (8), is (a) 563.1 and (b) 545.8. (a) PCA initialization. (b) SOM initialization.

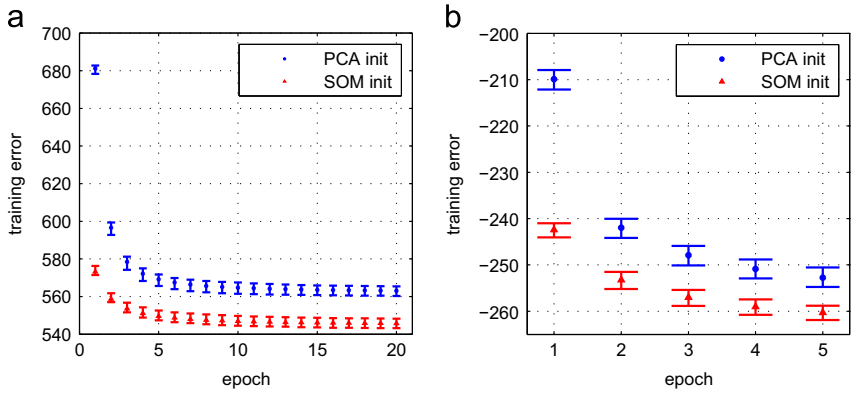


Fig. 2. Bootstrapped means and ranges of training error (8) evolution of GTMs trained using (a) the ISOLET data and (b) the MNIST data. For the MNIST data, sequential training with mini-batch size of 10 000 is used and only five training epochs is needed. In both cases, the GTMs with the SOM initialization converge to smaller training error. (a) ISOLET data. (b) MNIST data.

standard numerical precision tricks in the GTM algorithms. Evolution of bootstrapped training errors for the both GTMs is shown in Fig. 2(a). The GTM with the SOM initialization converges to lower training error, E_{GTM} (8). Mean bootstrap estimates (standard deviations in parentheses) for final training errors are 563.1 (1.4) for PCA initialization and 545.8 (1.3) for SOM initialization.

The MNIST data set (see http://yann.lecun.com/exdb/mnist/) contains 60 000 training samples and 10 000 test instances of handwritten digits. We use this separation in order to assess generalization of the GTM. Each feature in the data set is a gray scale value of a pixel, between zero and one. Fig. 3 shows the MNIST test instances mapped on GTMs trained using the MNIST training data with (a) PCA and (b) the SOM initialization. The maps were trained using the sequential training algorithm, which speeds up the convergence [21]. When mini-batch size of 10 000 was used, only five epochs of training was needed for sufficient convergence. The evolution of bootstrapped training errors is shown in Fig. 2(b). When comparing the resulting GTMs, the same observations can be made as for the ISOLET data. Both GTMs show some cluster structure, but this structure is clearer in the GTM initialized using the SOM. Bootstrapped mean estimates and standard deviations in parenthesis for test error (8) evaluated using test data are -255.0 (1.4) and -249.7 (1.3) for SOM and PCA initialization, respectively (standard deviations in parentheses).

For third high-dimensional data demonstration we use epigenomic measurements obtained by the ENCODE consortium [8]. The ENCODE data was recently used in [9] in order to train a large SOM to identify complex relationships in epigenomic measurements and other genomic data. The authors computed the RPKM (Reads Per Kilobase per Million reads) values from the signal of 72 measurements over six cell lines on 1.5 million genome segments. The resulting data matrix was used to train a SOM of size 30 times 45 units. The clusters, revealed by SOM, are associated with general and cell type-specific gene activity, and regulatory regions such as promoters and enhancers. The authors show that the distinct combination of epigenetic signals associated to a particular cluster can be used to find new genomic loci of that particular type. Here we use ChIP-seq read density profiles of only five different histone modifications (H3K4me1, H3K4me3, H3K79me2 and H3K9ac and H3K27me3) in human chronic myelogenous leukemia (K562) cell line. The goal is to investigate whether enhancers, promoters and random background regions form distinct clusters in GTM. The 1000 most significant p300 binding sites distal to any transcription start site (TSS), and the 1000 most significant DNase I hypersensitivity sites (DHSs) overlapping any TSS were used as a set of true enhancers and active promoters, respectively. In addition, histone modification signals at 1000

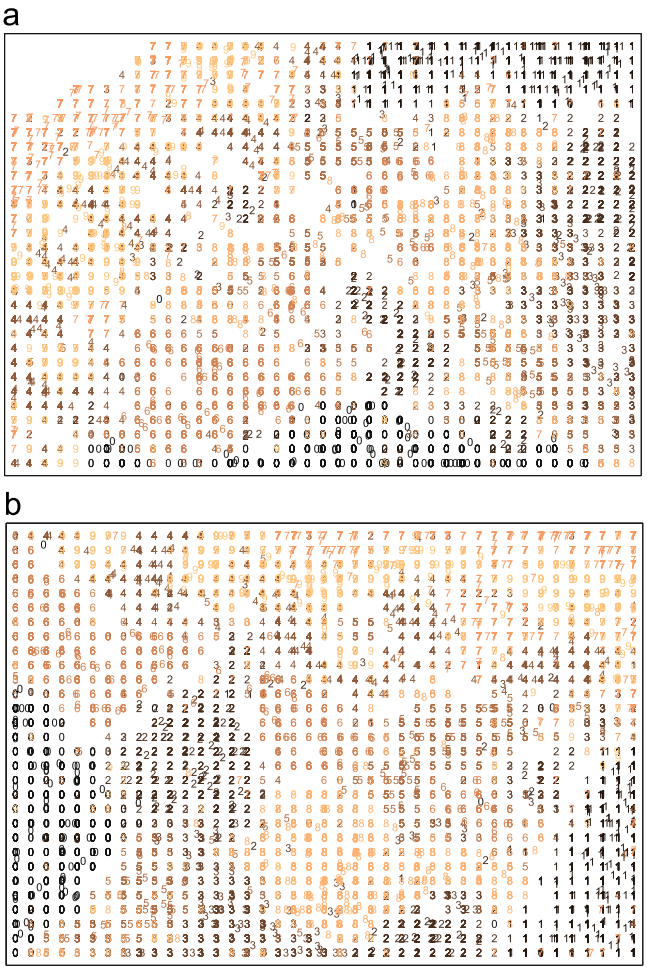


Fig. 3. The mapping of MNIST validation data on GTMs trained using MNIST training data and initialized using (a) PCA and (b) the SOM. Bootstrapped mean test error, E_{GTM} (8), is (a) -249.7 and (b) -255.0. (a) PCA initialization. (b) SOM initialization.

random locations were used as background samples. The random locations were sampled from genomic regions having the total read density signal greater or equal to 10, hence avoiding sampling regions with zero signal. The read density profiles at these sites were extracted using a 5000 base pair window centered on the region of interest, and further averaged over every fifth nucleotide. Finally the five histone modification signal profiles were

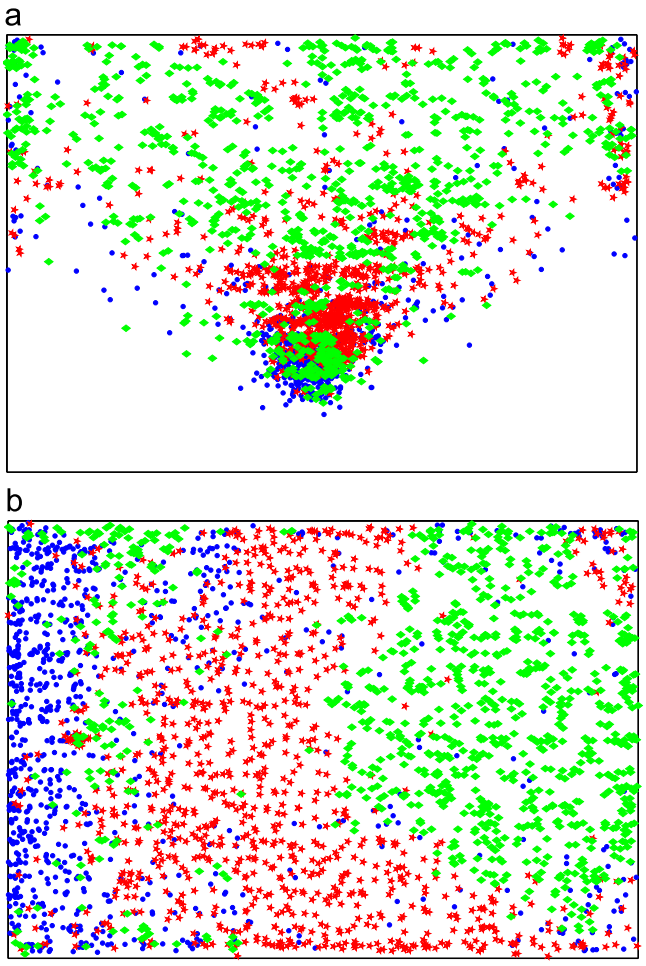


Fig. 4. GTMs trained using epigenomic data measured by the ENCODE consortium and initialized using (a) PCA and (b) the SOM. The promoter (green), enhancer (red) and background (blue) samples form semi-consistent clusters. Jitter is added to make data points distinguishable. (a) PCA initialization. (b) SOM initialization. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

concatenated to form 5000 dimensional vectors. The data was normalized by subtracting the mean and dividing by the standard deviation.

Fig. 4 shows GTMs trained using the ENCODE data. The difference between PCA and the SOM initializations is obvious, even though the difference of the bootstrapped final training errors, E_{GTM} (8) (standard deviation in parentheses), is only less than 2%, 4662.3 (53.0) vs. 4463.5 (51.1) for PCA and SOM initialization, respectively. In Fig. 4(b), different genomic regions, promoters (green), enhancers (red) and background sequences (blue) are clustered in distinct regions of the map. The mixing of enhancers, promoters and random background clusters may reflect the fact that not all TSS overlapping DHS or TSS-distal p300-binding sites necessarily are active promoters and enhancers, respectively, and hence do not show promoter or enhancer-specific epigenetic marks. The random background samples may also contain active promoters and enhancers.

5. Missing values

In this section, we discuss the behavior of topographic mappings in the presence of missing values. We start by showing how missing values are treated in the GTM and develop the same idea for the SOM. The section is concluded by an experimental study where even low-dimensional data sets reveal differences between the studied algorithms.

In all what follows, missing-at-random (MAR) data is assumed. This means that the probability of missingness is independent of missing values given the observed data. Even though this assumption can be questioned in many real-life scenarios, this is usually a reasonable assumption given that only a small proportion of the data is missing.

5.1. GTM and missing values

The GTM offers a robust framework for dealing with missing values, noted already in [3]. As with any method operating in the probabilistic framework, missing values can be handled by integrating them out. If the missing values are MAR, this does not introduce any bias. Hence, the maximum-likelihood estimation of the model parameters θ reduces to maximizing $\mathcal{L}(\theta|\boldsymbol{X}_{\mathrm{obs}}) = p(\boldsymbol{X}_{\mathrm{obs}}|\theta)$, where $\boldsymbol{X}_{\mathrm{obs}}$ denotes the observed data. For the GTM, the likelihood function is given by

$$\mathcal{L}(\boldsymbol{W},\beta|\boldsymbol{X}_{\mathrm{obs}}) = p(\boldsymbol{X}_{\mathrm{obs}}|\boldsymbol{W},\beta) = \int p(\boldsymbol{X}_{\mathrm{obs}}|\boldsymbol{X}_{\mathrm{mis}},\boldsymbol{W},\beta)\, d\boldsymbol{X}_{\mathrm{mis}}, \tag{10}$$

where $\boldsymbol{X}_{\mathrm{mis}}$ denotes the missing or unobserved data. This integration can be performed analytically for the standard GTM with an isotropic noise model.

The handling of missing data can be incorporated in the EM algorithm in a straightforward manner. In the E-step, where posterior probabilities of data vectors given the map units are calculated, missing values are simply omitted. That is, the distance between the map units and a data vector with missing value(s) is evaluated only in the dimensions observed for the corresponding data vector. In the M-step, the expected values of the missing data and other sufficient statistics are used. The details of learning the GTM with missing values using the EM algorithm can be found in [28].

After the training, there are at least two possibilities to perform imputation in the GTM. One may use the expected values $\mathbb{E}(\boldsymbol{X}_{\mathrm{mis}}|\boldsymbol{X}_{\mathrm{obs}},\boldsymbol{W},\beta)$ or impute using the maximum-a-posteriori (MAP) estimates $p_{\mathrm{MAP}}(\boldsymbol{X}_{\mathrm{mis}}|\boldsymbol{X}_{\mathrm{obs}},\boldsymbol{W},\beta)$ which takes the missing values from the most similar map unit. Additionally, multiple imputations can be conducted by sampling the posterior distribution $p(\boldsymbol{X}_{\mathrm{mis}}|\boldsymbol{X}_{\mathrm{obs}},\boldsymbol{W},\beta)$.

5.2. SOM and missing values

The SOM has been used for missing value imputation with many kinds of data, such as survey data [29,30], socioeconomic data [31], industrial data [32,33] and climate data [34]. In most of the SOM literature, the missing values are treated as was proposed in [31]. The best-matching units for the data vectors with missing values are computed by omitting the missing values. This is consistent with the procedure in the probabilistic setting. The missing values are ignored also while updating the reference vectors. This approach is implemented in the widely used SOM Toolbox [13]. After the training, missing values can be filled according to the best-matching units of the corresponding data vectors.

5.2.1. Imputation SOM

A novel approach, named the *Imputation SOM* (*impSOM*), stems from the way missing values are treated while using the GTM with an isotropic noise model (see above). The distances between data points and reference vectors are evaluated as described above, since this already corresponds to the statistical approach. While updating the reference vectors, instead of ignoring the missing

data their expected values

$$\hat{\boldsymbol{x}}_{ni,\mathrm{mis}} = \mathbb{E}[\boldsymbol{x}_{n,\mathrm{mis}}|\boldsymbol{m}_i] = \boldsymbol{m}_i \tag{11}$$

are used. Above, expectation is used in an informal sense, since the SOM is not a statistical model. This results in an update rule, where the reference vectors are updated according to (1) such that for each unobserved component of $\boldsymbol{x}_n$ the current value $\boldsymbol{m}_i$ is used. Thus, the data with missing values contribute by restraining the reference vectors in the dimensions corresponding to the missing values.

5.3. Model selection

This section demonstrates several aspects of model selection. The wine data set from UCI machine learning repository is used. It contains 13 chemical properties of 178 wines that come from three different wine regions. The data was normalized to have zero mean and unit variance and 5, 10, 30 and 50% of the values were randomly removed for validation, resulting in missing-completely-at-random (MCAR) data set.

Selecting the number of map units for the SOM is a subtle task. If the purpose of using the SOM is missing value imputation, one does not have the RMS plot of Fig. 5(b) available without first performing some kind of validation. Moreover, traditional SOM error measures, such as quantization and topological error, or combined error (3) do not give any straightforward way of determining a suitable map size. These difficulties in mind, the number of map units was deliberately increased above the number of data points in both this and the next section. This allowed experimenting the hypothesis that the excess map units interpolate the data space allowing more precise imputation (see, e.g., [34,28]).

Fig. 5 shows the combined error (3) and RMS imputation error with different map sizes K. The results are shown with 50% missing data, since the differences between the methods are emphasized when the missingness ratio is increased. The Imputation SOM obtains lower combined error with any map size ($p < 10^{-6}$), two-sample t-test was used for all statistical tests. Fig. 5(b) shows that the Imputation SOM is more robust in terms of the RMS imputation error when the map size is increased, but the batch SOM performs better with small K (significant when $K \leq 10$ and $K \geq 40$, $p < 10^{-6}$). The larger the map, the bigger the difference in the RMS imputations error in advantage of the Imputation SOM. Both methods perform best imputation when the grid size is close to 20 map units, hence $K=20$ was used in the subsequent comparison with 50% missing data.

Fig. 6 shows the behavior of the RMS imputation error with the different GTM imputation techniques. Error bars are omitted in order to avoid clutter. The GTM with $M=9$ (3×3) RBFs was selected using cross-validation. In Fig. 6(a) with 10% missing data, all the GTM imputation techniques improve as the map size is increased. However, maps with PCA initialization obtain relatively good results (no statistical difference to largest maps) already at $K=9$, hence this map size was chosen for subsequent comparison. In Fig. 6(b) with 50% missing data, the behavior is very different. It is notable, that the best imputation results are obtained using the smallest reasonable map size, three map units. When the GTM with 3 map units is used, the SOM initialization was not feasible, since only rectangular grid sizes are implemented in the SOM toolbox. In both figures, difference between MAP and expected

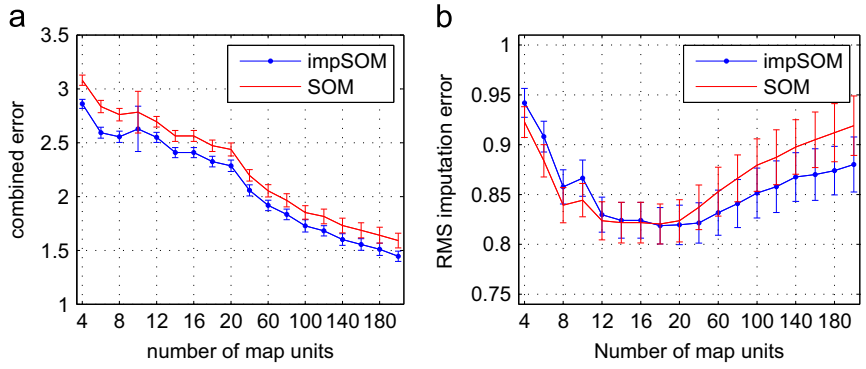


Fig. 5. (a) The mean combined error and (b) the mean RMS imputation error with respect to number of map units with wine data and 50% of missing data (note the nonlinear x-axis). The error bars show the standard deviation of the data. In (a), the methods are significantly different for all map sizes, K, ($p < 10^{-6}$) and in (b), the difference in means is significant when $K \leq 10$ and $K \geq 40$ ($p < 10^{-6}$). In both figures, the Imputation SOM is more robust when the grid size is increased. (a) Combined error. (b) RMS imputation error.

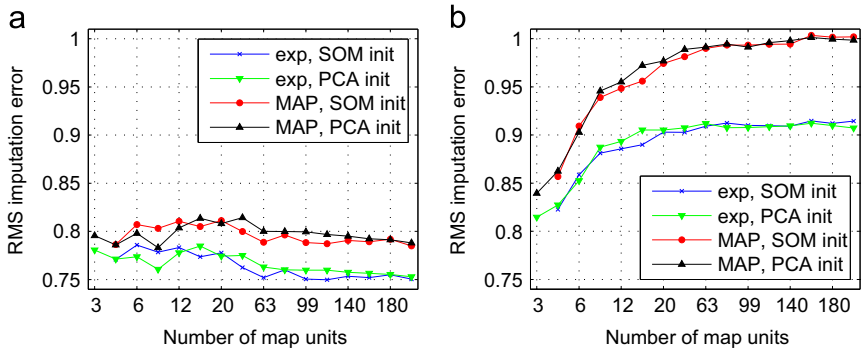


Fig. 6. The RMS imputation error for GTM with (a) 10% missing data and (b) 50% missing data in wine data with different map sizes. Better results are obtained when using expected values (exp) rather than maximum-a-posteriori estimates (MAP) for imputation ($p < 10^{-15}$). The SOM initialization is significantly better compared to PCA initialization only with 50% missing data and map size $K = 15$ $p < 0.005$). Note the nonlinear x-axis. (a) 10% of data missing. (b) 50% of data missing.

value imputation is significant for all K ($p < 10^{-15}$). In Fig. 6(b), there is still some evidence supporting the initialization with the SOM reference vector; when $K=15$, the RMS imputation error is slightly better using the SOM initialization ($p < 0.005$).

Table 1 summarizes the results of the experiments with the wine data set. One hundred randomly generated data sets with each missingness ratio were imputed using all the methods. For each missingness ratio, the smallest map size whose result was not statistically significantly worse than the best RMS imputation error, was chosen. The resulting map sizes were 63 (5%), 63 (10%), 40 (20%), 40 (30%), 20 (50%) map units for the SOM and 99 (5%), 99 (10%, SOM init), 9 (10%, PCA init), 4 (30%), 3 (50%, PCA init), 4 (50%, SOM init) map units for the GTM. Topographic mappings are compared with naive mean imputation and Variational Bayesian PCA (VBPCA) [35], which can be used as a generic black box imputation even with extremely sparse data. Comparison with the VBPCA is used to evaluate the general usability of topographic mappings in missing value imputation tasks. VBPCA can perform automatic relevance detection (ARD), hence no model selection is needed. We emphasize that more than two principal components are used in VBPCA. The mean result and its standard deviation in parentheses for each method and missingness ratio are listed. The best result(s) for each missingness ratio are in bold face.

Table 1
The means and the standard deviations (in parentheses) of the RMS imputation errors for different imputation methods obtained by imputing a hundred data sets with randomly generated missing data and four missingness proportions using the wine data set. The best results for each missingness proportion is in bold face (including the results which do not differ from the best statistically significantly). SOM methods perform better compared to all GTM methods ($p < 0.01$), when missingness proportion is 30%. (*) GTM expectation imputation performs better compared to MAP imputation for all missingness proportions (two last rows).

Method	5% Missing	10% Missing	30% Missing	50% Missing
Mean imputation	1.01 (0.06)	1.01 (0.05)	1.01 (0.03)	1.01 (0.02)
VBPCA	**0.71** (0.06)	**0.72** (0.04)	**0.77** (0.03)	**0.82** (0.02)
SOM	0.74 (0.07)	0.75 (0.05)	0.78* (0.03)	**0.83** (0.02)
ImpSOM	0.75 (0.07)	0.75 (0.05)	0.78* (0.03)	**0.82** (0.02)
GTM exp, SOM init	0.75 (0.07)	0.75 (0.06)	0.79 (0.03)	**0.82** (0.02)
GTM exp, PCA init	0.75 (0.08)	0.76 (0.06)	0.79 (0.03)	**0.82** (0.02)
GTM MAP, SOM init	0.78 (0.07)	0.78 (0.06)	0.82 (0.03)	0.86 (0.03)
GTM MAP, PCA init	0.79 (0.08)	0.80 (0.06)	0.82 (0.03)	0.84 (0.02)

The SOM methods and GTM with expectation imputation perform similarly except for 30% missingness ratio, when the difference between the SOM methods and all GTM methods is statistically significant ($p < 0.01$). The GTM using the expected values for imputation performs better compared to the MAP estimation of the missing values ($p < 10^{-4}$). The difference between VBPCA and the topographic methods is statistically significant except for the 50% missingness ratio.

We also present the visualizations provided by these methods when operating with 50% missing data in the Figs. 7 and 8. The gray-scale coloring behind the SOM Figs. 7(a) and (b) show U-Matrices of the maps. The three colors—blue, green and red—represent wines from three different wine regions, and the size of the colored markers in the SOMs is proportional to the number of data vectors mapped to the corresponding map unit.

For both SOMs in Fig. 7, the RMS imputation errors are relatively close: 0.812 for the batch SOM and 0.803 for the

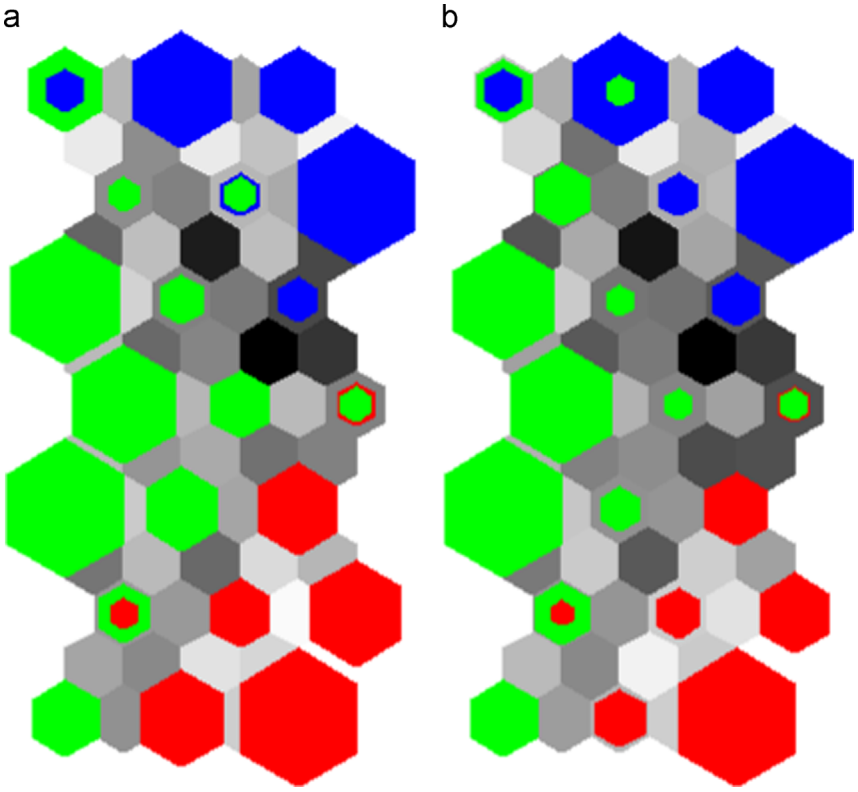


Fig. 7. Clustering of the wine data set with 50% missing data using (a) the batch SOM and (b) the Imputation SOM. A SOM with 21(7×3) map was used. The size of the colored markers is proportional to the number of data vectors mapped to the corresponding map unit. (a) SOM. (b) impSOM. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

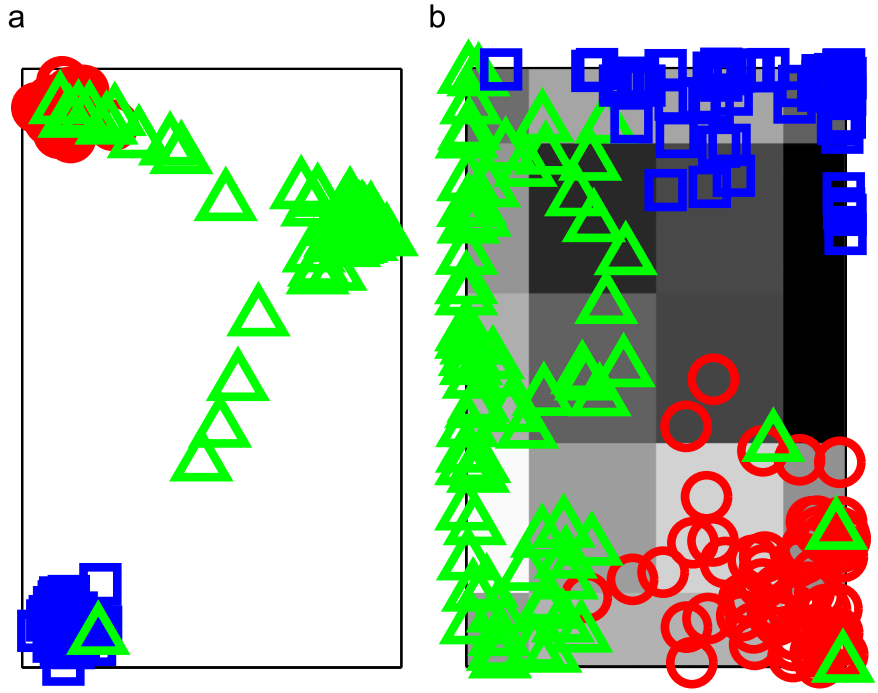


Fig. 8. Clustering of the wine data set with 50% missing data using the GTM with (a) 3, and (b) 20 (5×4) map units. Gaussian jitter is added in (a). (a) 3 map units. (b) 20 map units. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

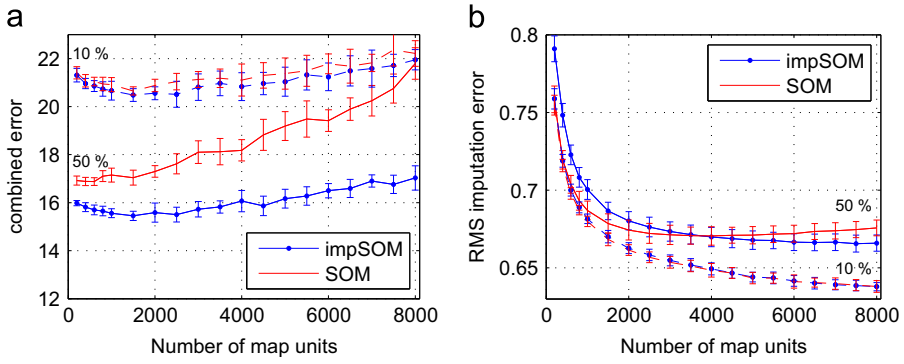


Fig. 9. A comparison between the batch SOM and Imputation SOM algorithms with 10% (dashed line) and 50% (solid line) of MAR missing values in ISOLET data. Errors are computed using left-out validation data in 10-fold cross-validation and error bars show the standard deviation of results. (a) Combined error. (b) RMS imputation error.

Imputation SOM. There are only minor differences between the maps produced by the batch SOM and the Imputation SOM.

According to the validation, an optimal number of map units for the GTM with 50% missingness ratio equals 3, hence the resulting visualization, shown in Fig. 8(a), differs from the ones obtained using the SOM. Gaussian jitter is added to distinguish points in the figure. The latent points $\boldsymbol{u}_i$ are assigned such that they form an equilateral triangle in the latent space; a configuration resembling the array of the hexagonal SOM. In the visualizations, the distances between the units are proportional to their distances in the original data space, that is, $d(\boldsymbol{u}_i, \boldsymbol{u}_j) \propto d(\boldsymbol{m}_i, \boldsymbol{m}_j)$. The resulting RMS imputation error is 0.808. It is notable, that the GTM is able to provide results comparable with the SOM, with only 3 map units. However, this is understandable since the data actually consists of three different clusters, wines from three distinct regions. Fig. 8(b) shows a GTM visualization using 20 (5×4) map units. In this example, the RMS imputation error equals 0.798, which is slightly lower compared to the GTM with three map units. The same realization of missing values was used in both figures. Furthermore, for this data the GTM with both PCA and the SOM initialization resulted in similar clustering. One possible explanation to this is, that the data lies close to the linear manifold spanned by the two principal components which makes learning the GTM model from the PCA initialization a feasible task.

To conclude this section, there seems to be small pieces of evidence—more robust imputation with increased grid size and lower combined error—supporting the Imputation SOM over the batch SOM. Regarding the GTM imputation, using the expectation of missing values proved to be the superior over the MAP estimates. This is natural, since using the MAP estimates discard information and is rarely a wise choice when dealing with multimodal distributions.

5.4. ISOLET data

Finally, the methods are compared using the ISOLET data introduced in Section 4. Fig. 9 illustrates differences between the Imputation SOM and the batch SOM. The values are based

Table 2
The means and the standard deviations (in parentheses) of the RMS imputation errors for different imputation methods obtained by imputing 10 data sets with randomly generated missing data and four missingness proportions using the ISOLET data set. Differences between the SOM methods are significant with 30 and 50% missing data (*). The VBPCA obtains the best result for each missingness ratio.

Method	5% Missing	10% Missing	30% Missing	50% Missing
Mean imputation	1.00 (0.03)	1.00 (0.02)	1.00 (0.01)	1.000 (0.01)
VBPCA	**0.42** (0.001)	**0.42** (0.001)	**0.44** (0.001)	**0.47** (0.0003)
SOM	0.64 (0.002)	0.64 (0.001)	0.652* (0.001)	0.669* (0.001)
ImpSOM	0.64 (0.002)	0.64 (0.001)	0.654* (0.001)	0.665* (0.001)

on 10-fold cross-validation, where both error measures are evaluated using left-out validation data, and the means and standard deviations (error bars) over validation folds are shown. Differences in combined error (3) increase when the map size is grown. The higher the proportion of missing data, the larger the difference in terms of the combined error. Differences are highly significant ($p < 10^{-5}$ for all K) when missingness proportion is 30% or 50% but is not significant ($p > 0.05$) when 10% or 5% missing data is used.

There are significant differences error between the SOM methods, with respect to RMS imputation error, only when missingness proportion is 50%. In that case, the batch SOM performs significantly ($p < 10^{-3}$) better imputation with small map sizes, when $K \leq 1000$, but the difference turns in favor of the Imputation SOM when $K \geq 7000$.

Table 2 summarizes the results of the experiments with the ISOLET data set. Ten randomly generated data sets with each missingness ratio were imputed using the SOM methods, VBPCA and mean imputation. For each missingness ratio, the smallest SOM map size, whose result was not statistically significantly worse than the best RMS imputation error, was chosen. The resulting map sizes were $K=8000$ for all SOMs except the batch SOM with 50% missing data, in which case $K=4000$. For all GTMs, the best map size was $K=3500$. The results are reported on data normalized to zero mean and unit variance.

The results with ISOLET data reveal more substantial differences between the methods compared to the results in Table 1 with the low-dimensional wine data. The GTM is excluded from the table, since we did not manage to thoroughly validate GTM parameters (K and M) for all missingness ratios, due to increasing computational burden when increasing the number of RBFs, M. However, the best results we obtained using the GTM with $M=400$ (20×20) RBF centers were significantly worse than the SOM results in Table 2. For the GTM with SOM initialization the lowest obtained RMS imputation errors were greater than 0.7 and with PCA initialization over 0.8. Differences between the SOM methods are statistically significant with 30 and 50% missing data, but in practice the differences are negligible compared to the differences to VBPCA and the GTM. The best results were obtained using VBPCA, which determines the number of components by ARD and more than two components (latent dimensions) were used. This suggests that topographic mappings are not particularly suitable for missing value imputation and it might be beneficial to impute the data with any robust imputation method before the SOM or the GTM visualization.

6. Conclusions and discussion

In this paper, we have studied convergence properties of the SOM and the GTM and their behavior in the presence of missing data. We also showed that initializing the GTM with the SOM may be beneficial in some cases where the GTM with the conventional PCA initialization fails to fit the data. This was demonstrated using the ISOLET, MNIST and ENCODE data sets. The initialization seems to have very little effect with the wine data set, the data with lowest dimensionality used in our experiments.

We have also proposed a novel way of treating missing values in the SOM training called the Imputation SOM and showed that this revision makes the SOM more robust in terms of the combined error (3) when missing values are present. The difference between the batch SOM and the Imputation SOM is emphasized when the map size is increased. However, when the main goal is to impute missing data, more well-founded ways, such as *multiple imputation by chained equations* (MICE) [36,37] or VBPCA, ought to be considered. In MICE, which is widely used in missing value imputation tasks, each variable with missing data is characterized by a separate conditional linear model. In the case, where one aims at visualizing the data, one option (not investigated here) would be to first impute the data using another method, and use the SOM or the GTM for the visualization task afterwards. It was also shown that if the GTM is used for missing value imputation, expected values rather than MAP estimates of missing values ought to be used. In the light of the imputation results, it seems that the SOM initialization of the GTM improves the learning, but the GTM performs worse in terms of RMS imputation error compared to the SOM when using high-dimensional ISOLET data set.

Our experiments were not suitable for comparing the CPU time consumed by SOM and GTM since the GTM algorithm used in our experiments was not implemented having computational efficiency in mind. However, it is claimed in [3] that for both algorithms, the dominant computational cost arises from the evaluation of the Euclidean distances between data points and reference vectors. This was verified to be true for GTMs with relative small number of RBFs ($M < 100$) using the Matlab Profiler, which measures where a program spends computational time. However, when the number of RBFs is increased, which was the case in ISOLET imputation experiments in Section 5.4, the computational cost becomes greater compared to the SOM.

In the future, it might be interesting to study whether the self-organization of the GTM benefits from sequential training. In our initial experiments, we have found that mini-batch training speeds up the convergence, as proposed by [21]. Additionally, the improvements developed to enhance the self-organization of the batch SOM may be applied for the GTM, as well. The number of RBFs, M, roughly corresponds to the width of the neighborhood function in the SOM. The smaller M, i.e. less RBFs, the more rigid the mapping. Thus, the effect of annealed neighborhood may be achieved by increasing the number of RBFs during the learning. It is also possible to use regularization, as was shown in [28], in order to control the rigidness of the GTM.

References

[1] T. Kohonen, Self-organized formation of topologically correct feature maps, Biol. Cybern. 43 (1982) 59–69.
[2] T. Kohonen, Self-Organizing Maps, 3rd Edition, Springer-Verlag, Inc., New York, Secaucus, NJ, USA, 2001.
[3] C.M. Bishop, M. Svensén, C.K.I. Williams, GTM: the generative topographic mapping, Neural Comput. 10 (1) (1998) 215–234.
[4] J. Venna, S. Kaski, Local multidimensional scaling, Neural Netw. 19 (6–7) (2006) 889–899.
[5] S. Kaski, J. Kangas, T. Kohonen, Bibliography of self-organizing map (SOM) papers: 1981–1997, Neural Comput. Surv. 1 (3&4) (1998) 1–176.
[6] M. Oja, S. Kaski, T. Kohonen, Bibliography of self-organizing map (SOM) papers: 1998–2001 addendum, Neural Comput. Surv. 3 (1) (2003) 1–156.
[7] M. Pöllä, T. Honkela, T. Kohonen, Bibliography of Self-Organizing Map (SOM) Papers: 2002–2005 Addendum, Technical Report TKK-ICS-R23, Helsinki University of Technology, 2009.
[8] An integrated encyclopedia of DNA elements in the human genome, Nature 489 (7414) (2012) 57–74. URL http://dx.doi.org/10.1038/nature11247.

[9] A. Mortazavi, S. Pepke, C. Jansen, G.K. Marinov, J. Ernst, M. Kellis, R.C. Hardison, R.M. Myers, B.J. Wold, Integrating and mining the chromatin landscape of cell-type specificity using self-organizing maps, Genome Res. 23 (12) (2013) 2136–2148, http://dx.doi.org/10.1101/gr.158261.113.
[10] T. Kohonen, P. Somervuo, How to make large self-organizing maps for nonvectorial data, Neural Netw. 15 (2002) 945–952.
[11] B. Hammer, A. Gisbrecht, A. Schulz, How to visualize large data sets?, in: P. A. Estévez, J.C. Príncipe, P. Zegers (Eds.), Advances in Self-Organizing Maps of Advances in Intelligent Systems and Computing, vol. 198, Springer, Berlin, Heidelberg, 2013, pp. 1–12.
[12] Y. Koren, The BellKor Solution to the Netflix Grand Prize, 2009.
[13] J. Vesanto, J. Himberg, E. Alhoniemi, J. Parhankangas, Self-organizing map in matlab: the SOM toolbox, in: The Matlab DSP Conference, 2000, pp. 35–40.
[14] NETLAB: Algorithms for Pattern Recognition, Springer-Verlag New York, Inc., New York, NY, USA, 2002.
[15] P. Koikkalainen, E. Oja, Self-organizing hierarchical feature maps, in: IJCNN, vol. 2, 1990, pp. 279–285.
[16] B. Fritzke, Growing cell structures—a self-organizing network for unsupervised and supervised learning, Neural Netw. 7 (9) (1994) 1441–1460.
[17] M. Dittenbach, D. Merkl, A. Rauber, The growing hierarchical self-organizing map, in: IJCNN, 2000, pp. 15–19.
[18] T. Villmann, R. Der, J.M. Herrmann, T. Martinetz, Topology preservation in self-organizing feature maps: exact definition and measurement, IEEE Trans. Neural Netw. 8 (2) (1997) 256–266.
[19] L. Zhang, E. Merényi, Weighted differential topographic function: a refinement of topographic function, in: ESANN, 2006, pp. 13–18.
[20] S. Kaski, K. Lagus, Comparing self-organizing maps, in: Artificial Neural Networks (ICANN) 1996, vol. 1112, Springer, Berlin/Heidelberg, 1996, pp. 809–814.
[21] C.M. Bishop, C.K.I. Williams, Developments of the generative topographic mapping, Neurocomputing 21 (1998) 203–224.
[22] S. Haykin, Neural Networks and Learning Machines, 3rd Edition, Prentice Hall, Upper Saddle River, New Jersey, USA, 2008.
[23] K. Kiviluoto, E. Oja, S-Map: A network with a simple self-organization algorithm for generative topographic mappings, in: Advances in Neural Information Processing Systems, Morgan Kaufmann Publishers, San Francisco, CA, USA, 1998, pp. 549–555.
[24] J.-C. Fort, P. Letrémy, M. Cottrell, Advantages and drawbacks of the Batch Kohonen algorithm, in: ESANN, 2002, pp. 223–230.
[25] A. Frank, A. Asuncion, UCI Machine Learning Repository, 2010. URL ⟨http://archive.ics.uci.edu/ml⟩.
[26] M.A. Fanty, R.A. Cole, Spoken letter recognition, in: NIPS, 1990, pp. 220–226.
[27] T. Vatanen, I. Nieminen, T. Honkela, T. Raiko, K. Lagus, Controlling self-organization and handling missing values in SOM and GTM, in: P.A. Estévez, J.C. Príncipe, P. Zegers (Eds.), Advances in Self-Organizing Maps of Advances in Intelligent Systems and Computing, vol. 198, Springer, Berlin, Heidelberg, 2013, pp. 55–64.
[28] T. Vatanen, Missing Value Imputation Using Subspace Methods with Applications on Survey Data, Master's thesis, Aalto University, Espoo, Finland, 2012. URL ⟨http://users.ics.tkk.fi/tvatanen/online-papers/mthesis_vatanen.pdf⟩.
[29] F. Fessant, S. Midenet, Self-organising map for data imputation and correction in surveys, Neural Comput. Appl. 10 (4) (2002) 300–310.
[30] S. Wang, Application of self-organising maps for data mining with incomplete data sets, Neural Comput. Appl. 12 (2003) 42–48.
[31] M. Cottrell, P. Letrémy, Missing values: processing with the Kohonen algorithm, in: Applied Stochastic Models and Data Analysis, 2005, pp. 489–496.
[32] R. Rustum, A.J. Adeloye, Replacing outliers and missing values from activated sludge data using Kohonen self-organizing map, J. Environ. Eng. 133 (9) (2007) 909–916.
[33] P. Merlin, A. Sorjamaa, B. Maillet, A. Lendasse, X-SOM and L-SOM: a double classification approach for missing value imputation, Neurocomputing 73 (7–9) (2010) 1103–1108.
[34] A. Sorjamaa, Methodologies for Time Series Prediction and Missing Value Imputation (Ph.D. thesis), Aalto University School of Science and Technology, Espoo, Finland, 2010.
[35] A. Ilin, T. Raiko, Practical approaches to principal component analysis in the presence of missing values, J. Mach. Learn. Res. 99 (2010) 1957–2000.
[36] S. van Buuren, H.C. Boshuizen, D.L. Knook, Multiple imputation of missing blood pressure covariates in survival analysis, Stat. Med. 18 (1999) 681–694.
[37] S. van Buuren, K. Groothuis-Oudshoorn, MICE: multivariate imputation by chained equations in R, J. Stat. Softw. 45 (3) (2011) 1–67.

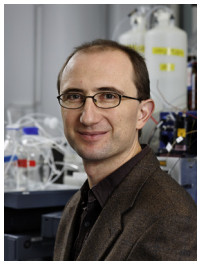
Tommi Vatanen is a Ph.D. student at Aalto University, Finland. He received his M.Sc. in bioinformation technology in 2012 from Aalto University School of Electrical Engineering. He is also affiliated with Broad Institute of MIT and Harvard in Cambridge, USA, where he is currently conducting research on human gut microbiome as a visiting graduate student.

Maria Osmala received the M.Sc. degree in bioinformation technology from Aalto University School of Electrical Engineering, Finland, in 2011. She is currently a doctoral student in Department of Information and Computer Science, Aalto University School of Science, Espoo, Finland. Her research interests include ChIP-seq data analysis in epigenetics studies, regulatory sequence prediction in human genome and data fusion in bioinformatics and computational systems biology.

Tapani Raiko received his D.Sc. degree in Computer Science in 2006 from Helsinki University of Technology. He is an Assistant Professor (tenure track) and an Academy Research Fellow at Aalto University School of Science. His research focus is deep learning.

Krista lagus is a senior researcher, group leader, and Ph.D. with experience in starting and leading research and new innovation-oriented activities. She has written over 70 scientific publications. Developed jointly two successful language technology methods and sofware applications, namely Websom and Morfessor. Started, planned and led EIT ICT Labs Wellbeing Innovation Camp 2010 and 2012.

Marko Sysi-Aho is a principal scientist and team leader of the Biosystems modelling team at the Technical research centre of Finland (VTT). He completed his Ph. D. in Computational Sciences at Aalto University in 2005. His current research focus is on medical applications of biosystems modelling, mainly related to development of methods for analysis and integration of metabolomics data with other data including environmental and life style factors.



Matej Orešič holds a PhD in biophysics from Cornell University. Since 2014 he is Principal Investigator at Steno Diabetes Center (Gentofte, Denmark), where he leads a Department of Systems Medicine. He is also an affiliated group leader at the Turku Centre for Biotechnology (Turku, Finland) and a principal investigator in the Academy of Finland Centre of Excellence in Molecular Systems Immunology and Physiology Research. His main research areas are metabolomics applications in biomedical research and integrative bioinformatics. He is particularly interested in the identification of disease vulnerabilities associated with different metabolic phenotypes and the underlying mechanisms linking these vulnerabilities with the development of specific disorders or their co-morbidities, with specific focus on obesity and diabetes and their co-morbidities. He has also initiated the popular MZmine open source project, leading to popular software for metabolomics data processing. Prior to joining Steno Diabetes Center, Dr. Orešič was research professor at VTT Technical Research Centre of Finland (Espoo, Finland), head of computational biology and modeling at Beyond Genomics, Inc. (Waltham/MA) and bioinformatician at LION Bioscience Research in Cambridge/MA.

Timo Honkela, Ph.D., is professor at the Department of Modern Languages at University of Helsinki. He has conducted research on several areas related to knowledge engineering, cognitive modeling and natural language processing. This includes a central role in the development of the Websom method for visual information retrieval and text mining based on the Kohonen self-organizing map algorithm. Honkela is a former long-term chairman of the Finnish Artificial Intelligence Society.

Harri Lähdesmäki received the M.Sc. and D.Sc. degrees from Tampere University of Technology in 2001 and 2005, respectively. Between 09/2002 and 03/2003, he was a visiting researcher at the Cancer Genomics Laboratory, The University of Texas M.D. Anderson Cancer Center, and from 2005 to 2007 he worked as a postdoctoral fellow at the Institute for Systems Biology, Seattle, WA, USA. From 2007 to 2008, he worked as an Assistant Professor in the Department of Signal Processing at Tampere University of Technology, Finland, followed by a pro term Professor position in Helsinki University of Technology until 2012. In autumn 2012 he was appointed to Assistant Professor (tenure track) and Academy Research Follow positions in the Department of Information and Computer Science at Aalto University School of Science (formerly known as Helsinki University of Technology). He is also an affiliated group leader at the Turku Center for Biotechnology, University of Turku. His research interests include computational and systems biology, regulatory genomics, statistical modeling and machine learning, with applications to immunology, stem cell and T1D.