In 'Advances in Self-Organizing Maps', pp. 224-229, Springer, 2001

A Topography-Preserving Latent Variable Model with Learning Metrics

Samuel Kaski and Janne Sinkkonen

Helsinki University of Technology
Neural Networks Research Centre
P.O. Box 5400, FIN-02015 HUT, Finland
{samuel.kaski,janne.sinkkonen}@hut.fi

Abstract. We introduce a new mapping model from a latent grid to the input space. The mapping preserves the topography but measures local distances in terms of auxiliary data that implicitly conveys information about the relevance or importance of local directions in the primary data space. Soft clusters corresponding to the map grid locations are defined into the primary data space, and a distortion measure is minimized for paired samples of primary and auxiliary data. The Kullback-Leibler divergence-based distortion is measured between the conditional distributions of the auxiliary data given the primary data, and the model is optimized with stochastic approximation yielding an algorithm that resembles the Self-Organizing Map, but in which distances are computed by taking into account the (local) relevance of directions.

1 Introduction

Topograhy-preserving latent variable models like the Self-Organizing Map (SOM) [2,3] are valuable tools especially for descriptive data analysis tasks, creating overviews of the data. Such models form an organized mapping from the latent space, usually a two-dimensional discrete map grid, to the input space. The map grid can be used as a graphical display whereon close-by locations represent similar data. Additional properties of the data, such as the density (cluster) structure and the distribution of the values of data variables, can be visualized on the display.

The mapping characterizes the probability density $p(\mathbf{x})$ of the multivariate (vectorial) data $\mathbf{x}$, but it depends on the metric of the data space. The metric in turn depends on the feature extraction: it is usually selected by first choosing the data variables and their relative scales, and then using a simple global measure such as the Euclidean distance. Feature extraction is often far from trivial. The variables may be of diverse nature, have different units of measurement, and their relative importance may be unknown. Moreover, the relative importance may be different in different locations of the data space.

We have earlier studied methods for learning suitable local distance measures. The task is impossible unless more information is brought to the setup. Our assumption has been that there is available some auxiliary data c whose conditional distribution implicitly conveys information about the relative importance of the available variables: the more the distribution $p(c|\mathbf{x})$ of the auxiliary data changes when $\mathbf{x}$ changes, the more important the change of $\mathbf{x}$ is. The auxiliary data occurs as paired samples $\{(\mathbf{x}^k, c^k)\}_k$ with the primary data. For simplicity we have so far assumed that the auxiliary data is discrete, multinomially distributed. In our case studies the auxiliary data have indicated a future bankruptcy in bankruptcy analysis, functional classes of genes in gene expression analysis, and keywords selected by experts in content-based text clustering.

We wish to measure distances between distributions, and so a natural choice is the Kullback-Leibler divergence D. A new (local) metric d is defined by the divergence between distributions at nearby points:

$$d^2(\mathbf{x}, \mathbf{x} + d\mathbf{x}) \equiv D(p(c|\mathbf{x}) \| p(c|\mathbf{x} + d\mathbf{x})) \ . \tag{1}$$

In an earlier work [1], $p(c|\mathbf{x})$ was approximated with a conditional density estimator and the distances (1) with the Fisher information matrix, and then computed a SOM was computed in the resulting metric. Since this otherwise promising approach involves several steps of approximations and relatively demanding computation, we have started to investigate other alternatives as well.

One approach is to directly incorporate the new distance measure (1) into the cost function of the mapping, as was earlier done for clustering purposes [4]. In the present paper we generalize this work to a latent variable model, in which the cluster centers are arranged onto a SOM-like grid, and their parameters are tied to each other through hidden variables and a neighborhood function: The cluster centers are sums of the hidden variables, weighted by the neighborhood function. This forces the grid to be organized, or "topography-preserving," at the minimum of the cost function.

2 The Method

We assume that there is a set of paired data $\{(\mathbf{x}_k, c_k)\}_k$ available, where $\mathbf{x} \in \mathbb{X} \subset \mathbb{R}^n$ and c is multinomially distributed. We postulate a grid on which a set of cluster centers $\mathbf{m}_j$ is attached, and we wish to form a soft clustering of the data space that uses differences between the distributions $p(c|\mathbf{x})$ as the distortion measure. Moreover, the cluster centers that are close-by on the grid are constrained to be relatively similar.

Our cost function should therefore have two properties: it shoud minimize a distortion measure, measured between the conditional distributions $p(c|\mathbf{x})$, and it should promote the smoothness of the map. We look at the distortion first, by starting from a generalized version of vector quantization, and impose a smoothing condition later.

A cost function measuring distortion between distributions. In "soft" vector quantization the goal is to find prototypes or codebook vectors $\mathbf{m}_j$ which minimize the average distortion E between the data and the codebook vectors. The average distortion is

$$E = \sum_j \int y_j(\mathbf{x}; \mathbf{m}_j) D(\mathbf{x}, \mathbf{m}_j)\ p(\mathbf{x})\ d\mathbf{x}\ , \tag{2}$$

where $D(\mathbf{x}, \mathbf{m}_j)$ is the distortion made when representing $\mathbf{x}$ by $\mathbf{m}_j$, and $y_j(\mathbf{x}; \mathbf{m}_j)$ is the "soft" cluster membership function that fulfills $0 \leq y_j(\mathbf{x}) \leq 1$ and $\sum_j y_j(\mathbf{x}) = 1$ for all $\mathbf{x}$. (Note that we have simplified the notation here for clarity: y_j may actually depend on all the $\mathbf{m}_j$.)

Because we are interested in differences between the distributions $p(c|\mathbf{x})$, we attach a distributional prototype $\boldsymbol{\psi}_j$ to each cluster j, and replace the usual squared Euclidean distortion measure with the Kullback–Leibler divergence $D_{KL}(p(c|\mathbf{x}), \boldsymbol{\psi}_j) \equiv \sum_i p(c_i|\mathbf{x}) \log(p(c_i|\mathbf{x})/\psi_{ji})$. Note that unlike the ordinary prototype of the standard vector quantization, $\boldsymbol{\psi}_j$ is not directly tied to a certain point of the input space (we will later point out an analytical form for it at the optimum of the cost function).

With this distortion measure the cost function of the vector quantization becomes

$$E_{KL} = \sum_j \int y_j(\mathbf{x}; \mathbf{m}_j) D_{KL}(p(c|\mathbf{x}), \boldsymbol{\psi}_j)\ p(\mathbf{x})\ d\mathbf{x}\ . \tag{3}$$

Instead of computing the distortions between the vectorial samples and vectorial prototypes as in (2), we now compute distortions between the distributions $p(c|\mathbf{x})$ and the distributional prototypes $\boldsymbol{\psi}_j$. The cost function (3) is to be minimized with respect to both the $\mathbf{m}_j$ and the $\boldsymbol{\psi}_j$.

Parameterization and imposing smoothness. As in the earlier work [4] with independently parameterized clusters, the cluster membership functions $y_j(\mathbf{x})$ will be of the form $y_j(\mathbf{x}) = G_j(\mathbf{x}) / \sum_k G_k(\mathbf{x})$, where the $G_j(\mathbf{x}) = e^{-\|\mathbf{x}-\mathbf{m}_j\|^2/2\sigma^2}$ are Gaussians. The dispersion parameter σ, implicit in the cost function 3, is selected *a priori* or by using a validation set. Note that because $\sum_j y_j(\mathbf{x}) = 1$, the membership functions may be interpreted as the conditional pdf of a random variable, here denoted by V and the values denoted by v_j.

In the present work our intention is to generate a solution where the clusters j reside on a grid and the nearby clusters are relatively similar by their parameters. To achieve such a smoothness, we force nearby clusters to be similar by making the cluster location parameters $\mathbf{m}_j$ dependent on each other through a set of *hidden variables* $\mathbf{n}_k$: $\mathbf{m}_j = \sum_k h_{jk}\mathbf{n}_k$. The coefficients h_{jk} are positive, fulfilling $\sum_k h_{jk} = 1$, and they define the topology of the grid; they resemble the neighborhood function of the SOM. We have used (normalized) Gaussian neighbourhoods, with the standard deviation denoted by β, and a rectangular grid.

Minimizing the cost function. The gradient of the cost function with respect to the hidden variables is

$$\nabla_{\mathbf{n}_j} E_{KL} = -\frac{1}{\sigma^2} \sum_{i,k,l} h_{lj} \log \frac{\psi_{li}}{\psi_{ki}} \int (\mathbf{x} - \mathbf{m}_l) p(v_l, v_k, c_i, \mathbf{x}) d\mathbf{x} \ , \tag{4}$$

where the random variable interpretation of the membership functions was used to write $y_l(\mathbf{x}) y_k(\mathbf{x}) p(c_i, \mathbf{x})$ as $p(v_l, v_k, c_i, \mathbf{x})$.

A very simple algorithm follows by applying stochastic approximation to the optimization problem. The (factorizable) joint distribution $p(v_l, v_k, c_i, \mathbf{x})$ can be used as the sampling function: an input $\{(\mathbf{x}(t), c_i)\}$ is drawn, and then two clusters k and l from the $\mathbf{x}$-conditioned multinomial distribution $\{y_k(\mathbf{x}(t))\}$. The adaptation rule can be formulated on the basis of the cluster centers $\mathbf{m}_j$ alone, without referring to the hidden variables:

$$\mathbf{m}_j(t+1) = \mathbf{m}_j(t) + \alpha(t) \bar{h}_{lj} \log \frac{\psi_{li}}{\psi_{ki}} (\mathbf{x} - \mathbf{m}_l) \ , \tag{5}$$

where $\bar{h}_{lj} \equiv \sum_s h_{ls} h_{js}$.

The other free parameters besides the membership function locations are the prototype distributions ψ_{ji}. After the reparameterization $\log \psi_{ji} \equiv \gamma_{ji} - \log \sum_r \exp \gamma_{jr}$, to keep ψ_{ji} positive and summed up to unity, the distributional prototype can be optimized with the stochastic approximation step

$$\gamma_{lr}(t+1) = \gamma_{lr}(t) - \alpha(t)(\psi_{lr} - \delta_{ri}) \ , \tag{6}$$

where δ_{ri} is the Kronecker delta. (For more details cf. [4].)

It is possible to show that the optimal values for the prototypes are $\psi_{ji} = p(c_i|v_j)$, i.e., they converge to the cluster-wise averages of the conditional distributions $p(c_i|\mathbf{x})$. Plugging this into the distortion function (3) gives a connection to mutual information: minimizing the Kullback-Leibler based distortion function is in fact equivalent to maximizing the mutual information between the random variables C and V, i.e. essentially between the guiding auxiliary variable and the cluster membership functions.

3 Demonstration

We demonstrate the method by mapping a seemingly simple toy data set, in which we know what the solution should approximately be. The data is uniformly distributed inside a 3D cube (the thin lines in Fig. 1**a**). The class distribution changes only in the direction of the horizontal xy-plane, approximately uniformly everywhere. Assuming that the changes of the class distribution are the important aspects in the data, the vertical z-direction is irrelevant and resources will be wasted if it is modelled.

Data. The data consists of four equally likely classes c_i, of which two change linearly along the one side (x-direction) of the horizontal square, and the other two along the other side (y-direction). If the length of the side of the cube is 1.6 and the origo is in the middle of the cube, the class distributions are $p(c_0|\mathbf{x}) = 0.1x + 0.25$, $p(c_1|\mathbf{x}) = 0.5 - p(c_0|\mathbf{x})$, $p(c_2|\mathbf{x}) = 0.1y + 0.25$, and $p(c_3|\mathbf{x}) = 0.5 - p(c_2|\mathbf{x})$.

Optimization. During preliminary simulations it turned out that the algorithm described in Section 2 is fairly sensitive to the initial conditions, and hence we optimized the cost function in two phases. The first phase ensures the organization of the map and the second is a fine-tuning phase.

The map consisting of 7 by 7 units was initialized randomly by setting the model vectors to the first 49 data points. In the *first learning phase* of 2 million iterations the width of the neighborhood β was gradually decreased from 4.0 to its final value of 1.0. The width decreased first with a piecewise-linear approximation to an exponential curve, and then linearly to one in the end. The learning coefficient $\alpha(t)$ was simultaneously decreased from 0.5 to zero with the same schedule. The cluster membership functions were rather wide, with $\sigma = 0.3$.

During this organization phase we applied another rule in addition to (5) to reduce exessive degrees of freedom in the mapping. Note that the sole purpose of this first learning phase is to force the mapping to a favourable, organized initial state. All clusters in the neighborhood of the cluster s closest to the data point, $s = \arg\min_j \|\mathbf{x} - \mathbf{m}_j\|$, were adapted towards $\mathbf{m}_s$:

$$\mathbf{m}_j(t+1) = \mathbf{m}_j(t) + (\alpha(t)/C)^2 \bar{h}_{sj}(\mathbf{m}_s(t) - \mathbf{m}_j(t)) \; . \tag{7}$$

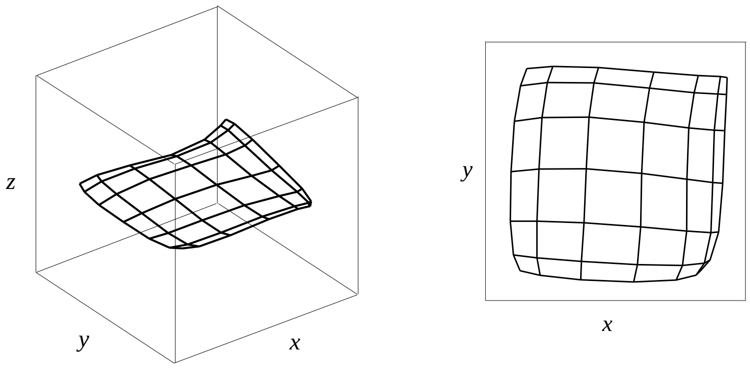
In our simulations $C = 4$. Note that as α goes to zero, α^2 goes to zero much faster, and hence (7) has an effect only in the beginning of the learning process.

In the second, fine-tuning phase of another 2 million iterations, the width of the neighborhood remained at 1.0, and $\alpha(t)$ decreased with the same piecewise-linear schedule from 0.05 to zero. The cluster membership functions were sharpened during the first 100,000 iterations by changing σ linearly from 0.3 to 0.1. The adaptation step (7) was not used.

Results. As can be seen in Figure 1**a**, the converged map represents only the two directions x and y that are relevant in the sense of the classes. The map represents these two dimensios in an ordered fashion (Fig. 1**b**). By contrast, a SOM would try to fold itself to fill the whole 3D cube (cf. [3]).

4 Conclusions

We have introduced a new mapping method that preserves the topography of the input space but implicitly measures the similarity of close-by data



a **b**

Fig. 1. **a** In a three-dimensional cube filled with uniformly distributed data, the map represents only the relevant two-dimensional xy-subspace, in which the conditional distribution of the auxiliary data changes. **b** In the relevant subspace the representation is ordered ("topography-preserving"). Model vectors that are neighbors on the map grid have been connected by lines, and the model vectors are represented by the nodes of the resulting "network."

points by differences in the conditional distributions of associated auxiliary data. The auxiliary data can therefore be used to define what is important in the primary data. Assuming such auxiliary information is available, the new method solves the problem of how the (originally arbitrary) distances should be computed, common in many exploratory multivariate analysis tasks.

The proposed optimization method is still fairly sensitive to the initial state and details of the optimization procedure. We will next concentrate on improving the optimization method.

References

1. S. Kaski, J. Sinkkonen, and J. Peltonen. Bankruptcy analysis with self-organizing maps in learning metrics. *IEEE Transactions on Neural Networks*, 2001. Accepted for publication.
2. T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.
3. T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, 1995. (Third, extended edition 2001).
4. J. Sinkkonen and S. Kaski. Clustering based on conditional distributions in an auxiliary space. *Neural Computation*, 2001. Accepted for publication.