Learning More Accurate Metrics for Self-Organizing Maps

Jaakko Peltonen, Arto Klami, and Samuel Kaski

Neural Networks Research Centre, Helsinki University of Technology,
P.O. Box 9800, FIN-02015 HUT, Finland
{Jaakko.Peltonen, Arto.Klami, Samuel.Kaski}@hut.fi
http://www.cis.hut.fi

Abstract. Improved methods are presented for learning metrics that measure only important distances. It is assumed that changes in primary data are relevant only to the extent that they cause changes in auxiliary data, available paired with the primary data. The metrics are here derived from estimators of the conditional density of the auxiliary data. More accurate estimators are compared, and a more accurate approximation to the distances is introduced. The new methods improved the quality of Self-Organizing Maps (SOMs) significantly for four of the five studied data sets.

1 Introduction

Variable selection or feature extraction is a burning problem especially for exploratory (descriptive) data analysis. The quality of the results is determined by the selection since there is no other supervision. Poor features may emphasize uninteresting properties of the data.

An alternative view to feature extraction is that the topology and the metric of the data space need be chosen. We study the choice of the metric; if the topology need be changed it can be done as a preprocessing step.

Assume that there exists auxiliary data c paired with the primary data $\mathbf{x}$. Here $\mathbf{x}$ is vector-valued and c categorical (finite number of possible values). Assume further that the goal is to study the $\mathbf{x}$, explore or describe them, but that changes in $\mathbf{x}$ are only relevant to the extent they cause changes in c. An example is analysis of the causes of bankruptcy, where the $\mathbf{x}$ contains features of the financial state of a company and c denotes whether the company goes bankrupt or not.

In the learning metrics principle ([5, 9]; see [4, 5] for more detailed discussion) the distance d between two close-by points $\mathbf{x}$ and $\mathbf{x} + d\mathbf{x}$ of the primary data space is measured by approximations to the distance between the important things, the distribution of c:

$$d^2(\mathbf{x}, \mathbf{x} + d\mathbf{x}) \equiv D_{KL}(p(c|\mathbf{x}), p(c|\mathbf{x} + d\mathbf{x})) = d\mathbf{x}^T \mathbf{J}(\mathbf{x}) d\mathbf{x} \; . \tag{1}$$

Here D_{KL} is the Kullback-Leibler divergence and $\mathbf{J}(\mathbf{x})$ is the Fisher information matrix having $\mathbf{x}$ as its parameters,

$$\mathbf{J}(\mathbf{x}) = E_{p(c|\mathbf{x})}\left[(\nabla_{\mathbf{x}} \log p(c|\mathbf{x}))\,(\nabla_{\mathbf{x}} \log p(c|\mathbf{x}))^T\right] \,. \tag{2}$$

In earlier studies the metric has either been incorporated into the cost function of a method [9], or the density $p(c|\mathbf{x})$ has been estimated and the Fisher information matrix computed from the estimate [5]. Here we extend the latter approach by more accurate density estimators and distance approximation. Since different kinds of data sets may require different kinds of estimators, we suggest choosing the estimator using a validation set.

2 Self-Organizing Maps in Learning Metrics

We will apply the learning metrics to Self-Organizing Maps (SOMs) [7] to improve our earlier results [5].

A SOM is a regular lattice of units i. Each unit contains a model $\mathbf{m}_i$, a representation of particular kinds of data in the data space. The model vectors are adapted with an iterative training algorithm to follow the distribution of the training data. For brevity, we call a SOM trained in learning metrics SOM-L and a SOM in Euclidean metrics SOM-E.

The training algorithm repeats two steps, winner search and adaptation. At each iteration t, an input sample $\mathbf{x}(t)$ is picked randomly from the data, and a winner SOM unit $w(t)$ is selected by

$$w(t) = \arg\min_i d^2(\mathbf{x}(t), \mathbf{m}_i(t)) \,, \tag{3}$$

where d^2 is the distance function. Here the distance is not in the traditional Euclidean metric but the learning metric (1) derived from the auxiliary data.

When the winner has been selected, the model vectors are all adapted towards the input sample in the steepest descent direction. For learning metrics the direction is given by the natural gradient. For the local approximation (1) this leads to the familiar update rule

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t) h_{wi}(t)(\mathbf{x}(t) - \mathbf{m}_i(t)) \tag{4}$$

which we have used in this paper. Here $\alpha(t)$ is the learning rate and $h_{wi}(t)$ is the neighborhood function, a decreasing function of the distance between i and $w(t)$ on the SOM lattice.

In practical SOM-L training we use two approximations for calculating learning metric distances. Firstly, the matrix $\mathbf{J}(\mathbf{x})$ is computed from an estimate of the conditional density $p(c|\mathbf{x})$. The investigated alternatives are introduced in Section 3. Secondly, the global distance between two points $\mathbf{x}$ and $\mathbf{m}$ is actually defined as the minimal path integral of the local distances, where the minimum is taken over all paths between $\mathbf{x}$ and $\mathbf{m}$. Finding exact minimal path integrals is computationally prohibitive, so in Section 4 we consider several approximations. The approximations are compared empirically in Section 5.

3 Estimating the Auxiliary Distribution

Learning of the metric is based on the Fisher information matrix of a conditional density estimate. Here we discuss alternative kernel estimators.

We have previously derived the conditional densities from estimators of the joint density of $\mathbf{x}$ and c. Two standard estimators, the nonparametric Parzen kernel estimate and a version of Mixture Discriminant Analysis (MDA2) were used. Here we compare other estimators to MDA2; Parzen was too computationally intensive to be included as such.

Since only the conditional densities $p(c|\mathbf{x})$ are needed here, directly estimating them should improve the results. We consider two alternatives. The first is a kind of a mixture of experts (see [3]):

$$\hat{p}_{MoE}(c_i|\mathbf{x}) = \sum_{j=1}^{N_U} y_j(\mathbf{x})\psi_{ji} \ . \tag{5}$$

Here N_U is the number of mixture components. The ψ_{ji} are the parameters of the multinomial distribution generated by the expert j. Their sum is fixed to unity by softmax-reparameterization (not shown). The $y_j(\mathbf{x})$ form the gating network; we used Gaussians normalized to sum to unity for each $\mathbf{x}$.

The second method for conditional density estimation is a product of experts [2], here

$$\hat{p}_{PoE}(c_i|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \sum_{j=1}^{N_U} \exp\left(y_j(\mathbf{x}) \log \psi_{ji}\right) \tag{6}$$

where $Z(\mathbf{x})$ normalizes the density to sum to one.

The MDA2 is fitted to data by maximizing the joint log-likelihood with respect to the ψ_{ji} and the parameters of the gating network by the EM algorithm. For the other models the mean conditional log-likelihood of the auxiliary data is maximized by conjugate gradient algorithms. All estimators include a free dispersion parameter, the variance of the Gaussians. This parameter is chosen to maximize the conditional likelihood of auxiliary data on a validation set.

4 Distance Approximations

The true learning metric distances are minimal path integrals which must be approximated. A simple approximation $\hat{d}_1^2$ of the squared distance, used earlier in e.g. [5], is to evaluate the metric at the input sample $\mathbf{x}$ and to extend the local distance to the whole space. In winner search the distance becomes

$$\hat{d}_1^2(\mathbf{x}, \mathbf{m}) = (\mathbf{m} - \mathbf{x})^T \mathbf{J}(\mathbf{x})(\mathbf{m} - \mathbf{x}) \ . \tag{7}$$

We call this the '1-point approximation'. It is accurate when the model vectors are close to $\mathbf{x}$. Note that for winner selection we need not know the exact distances but only which one is the smallest.

Table 1. The Data Sets

Data set	Dimensions	Classes	Samples
Landsat Satellite Data *	36	6	6435
Letter Recognition Data *	16	26	20000
Phoneme Data from LVQ_PAK [8]	20	14	3656
TIMIT Data from [10]	12	41	14994
Bankruptcy Data used in [5]	23	2	6195

* from UCI Machine Learning Repository [1]

Our earlier conditional density estimates have been smooth, obtained with a small number of wide kernels. Such estimates may fail to notice some detail in the density but the simple local approximation (7) may be reasonably accurate because of the smoothness. However, for more accurate estimators that potentially change more rapidly the local approximation may hold only very locally.

A more accurate but still computable approximation is obtained by assuming that the minimal path is a line but that the metric may change along the line. When the metric along the line connecting $\mathbf{x}$ and $\mathbf{m}$ is evaluated at T points, the distance becomes

$$\hat{d}_T^2(\mathbf{x},\mathbf{m}) = \frac{1}{T^2}\left(\sum_{t=1}^{T}\left((\mathbf{m}-\mathbf{x})^T\mathbf{J}\left(\mathbf{x}+\frac{t-1}{T}(\mathbf{m}-\mathbf{x})\right)(\mathbf{m}-\mathbf{x})\right)^{1/2}\right)^2 . \quad (8)$$

We call the above the 'T-point approximation'.

The T-point approximations involve more computation. The computational complexity of a single SOM-L training iteration becomes $\mathcal{O}(N_{DIM}N_CN_UN_{SOM}T)$ for N_{SOM} model vectors with dimensionality N_{DIM}, N_C classes, and N_U mixture components. By comparison, the complexity of the 1-point approximation is $\mathcal{O}(N_{DIM}N_C(N_U+N_{SOM}))$.

The T-point winner search may be speeded up by first using 1-point distances to winnow the set of winner candidates; e.g. the W model vectors that are closest according to $\hat{d}_1^2$ are selected and the winner is chosen from these by $\hat{d}_T^2$. In the empirical tests of Section 5 we have used $T = 10$ evaluation points and $W = 10$ winner candidates, resulting in a 20-fold speed-up compared to the unwinnowed T-point approximation, but computational time compared to the 1-point approximation was still about 100-fold.

5 Empirical Testing

The methods were compared on five different data sets (Table 1). The class labels were used as the auxiliary data and the data sets were preprocessed by removing the classes with only a few samples.

The metric was estimated from training data with the methods presented in Sections 3 and 4. The number of mixture components (10, 30, or 100) and the dispersion parameter were selected using a validation set. The SOM-E and the SOM-L were trained in the resulting metrics, using both the 1-point and T-point ($T = 10$, $W = 10$) distance approximations for SOM-L.

The accuracy of the resulting SOMs in representing the important auxiliary data was measured by the conditional likelihood evaluated at the winner SOM units [5]. The quality of the SOM visualizations was monitored visually.

The significance of the difference between the best SOM-L and SOM-E was tested using 10-fold cross-validation. The dispersion (Gaussian variance) was validated anew in each fold to maximize map accuracy, using part of the training set for validation. The accuracy (likelihood) for the best map was then calculated for the test set.

To reduce the consumption of computational resources, we selected a suboptimal density estimator for the SOM-L having T-point distances: the estimator optimal for 1-point SOM-L was chosen with 30 components.

6 Results

The learning metrics improved the accuracy of the SOMs on all data sets; the improvements were significant by t-test ($p < 0.05$) between SOM-E and SOM-L with the T-point approximation, except for the Bankruptcy data ($p = 0.07$).

The mixture of experts (5) was best for SOM-L in three sets, the product of experts in one and MDA2 in one.

The more accurate distance approximation (8) is crucial. The SOM-L with the 1-point approximation was only comparable or worse than SOM-E on two data sets, while SOM-Ls trained with the improved approximation are on average better on all sets.

Figure 1 shows the performance of the SOM-L and SOM-E on one data set in the dispersion validation phase, averaged over the cross-validation folds. The SOM-L with the 1-point approximation is here roughly equal to SOM-E, but the SOM-L with the T-point approximation is clearly better for all dispersion values.

7 Discussion

More accurate estimation of the learning metrics still improves the SOM results from the earlier results. The computational complexity is higher but manageable with suitable approximations.

Based on preliminary results it seems that both of the new elements, more accurate density estimation and more accurate distance approximation, are required. The earlier one-point distance approximations are not accurate enough for the new density estimators capable of following more accurately the details of the conditional density.

Acknowledgment

This work was supported by the Academy of Finland, in part by the grant 52123.

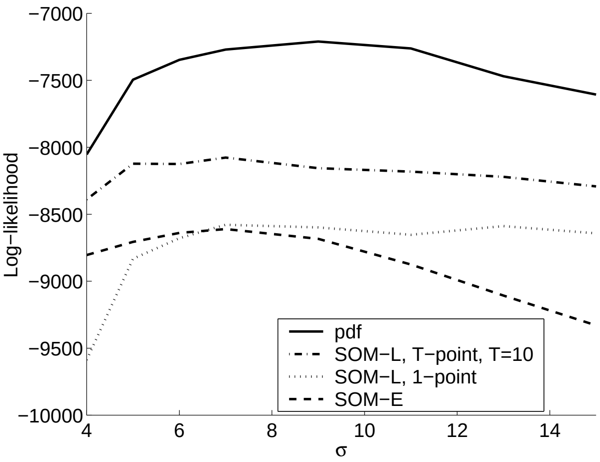

Fig. 1. Average accuracy of SOM-L vs. SOM-E for the TIMIT data over the 10-fold validation sets. The likelihood given by the best pdf estimate (mixture of 100 experts) is included for reference; it is the approximate upper limit.

References

1. Blake, C.L., and Merz C.J. UCI Repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998.
2. Hinton, G.E. Products of Experts. In *Proceedings of ICANN'99, the Ninth International Conference on Artificial Neural Networks*, 1–6, IEE, London, 1999.
3. Jordan, M., and Jacobs, R. Hierarchical Mixtures of Experts and the EM Algorithm. *Neural Computation*, 6:181–214, 1994.
4. Kaski, S., Sinkkonen, J. Principle of learning metrics for exploratory data analysis. Submitted to a journal.
5. Kaski, S., Sinkkonen, J., and Peltonen, J. Bankruptcy Analysis with Self-Organizing Maps in Learning Metrics. *IEEE Transactions on Neural Networks*, 12:936–947, 2001.
6. Kaski, S., and Venna, J. Neighborhood preservation in nonlinear projection methods: An experimental study. In G. Dorffner, H. Bischof, and K. Hornik, editors, *Artificial Neural Networks–ICANN 2001*, 458–491, Springer, Berlin, 2001.
7. Kohonen, T. Self-Organizing Maps. Springer, Berlin, 1995 (Third, extended edition 2001).
8. Kohonen T, Kangas J, Laaksonen J, and Torkkola K. LVQ_PAK: A program package for the correct application of Learning Vector Quantization algorithms. In *Proceedings of IJCNN'92, International Joint Conference on Neural Networks*, I:725-730, 1992.
9. Sinkkonen, J., and Kaski, S. Clustering based on conditional distributions in an auxiliary space. *Neural Computation*, 14:217–239, 2002.
10. TIMIT 1998. CD-ROM prototype version of the DARPA TIMIT acoustic-phonetic speech database.