Learning Metrics for Information Visualization

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

Keywords: Information visualization, learning metrics, SOM, Sammon's mapping

Abstract— The learning metrics principle shows how (nonlinear) projection and clustering methods can be made to focus on discriminative properties of data. In this paper we review and extend our earlier work on learning metrics for self-organizing maps (SOMs), compare algorithms, and introduce a new accurate distance computation algorithm. It can be used with methods that work on pairwise distances between the data samples. Its usefulness is demonstrated for Sammon's mapping, a form of multidimensional scaling.

1 Introduction

The motivation for our earlier work on SOMs that learn metrics was that in data exploration (visualization, clustering) there often is no rigorous basis for choosing a metric. The SOM algorithm [7] can use many kinds of metrics but the user has to make the choice, usually by fixing the distance measure of SOM to be Euclidean (or some other global alternative), and selecting, transforming, and scaling the input variables.

The learning metrics principle [4, 5] was developed to formalize the idea that it is possible to learn a metric from dependencies between the primary data and another set called auxiliary data. The SOM computed in learning metrics becomes *discriminative* of the auxiliary data. One example application is clustering of companies such that the clusters discriminate between companies with high and low bankruptcy risk [5].

The first work used a coarse estimate for the metric that was fast to compute but only accurate locally. We later [8] developed more accurate ways to compute longer distances. In this work the details have been finalized, and the resulting algorithms are compared to provide a recommendation of which variants to use. Comparisons with the standard Euclidean SOM and a straightforward way of incorporating class information into SOMs, the supervised SOM [7], are included.

In the SOM algorithm the metric cannot be very intensive to compute since distances need to be computed during each iteration. By contrast, in methods that require only pairwise distances between data points, the distances need to be computed only once in the beginning. We introduce a more accurate distance computation algorithm for such cases; it combines graph search with the earlier methods. The algorithm is demonstrated for Sammon's mapping [10], a variant of multidimensional scaling.

2 The learning metrics principle

The data to be analyzed are vector-valued samples $\mathbf{x} \in \mathbb{R}^n$, called here *primary data*. The difference from the standard SOM setting is that in the learning data set, each primary data vector is paired with a sample of *auxiliary data* c. Here the c in $(\mathbf{x}, c)$ is categorical, for example a class of the sample.

The fundamental assumption that makes the principle sensible is that the auxiliary data is chosen well. Changes in primary data are considered relevant only to the extent they change the auxiliary data. In a sense, choosing the auxiliary data amounts to choosing a viewpoint to the primary data. For example, changes in economic indicators of companies may be considered important only if they correlate with bankruptcy risk, or changes in gene expression only if related to the presence of a particular disease. In the latter example, an indicator of whether the disease is present is suitable auxiliary data, and the gene expression levels are the primary data.

Intuitively, the learning metrics principle suggests computing distances by how much the auxiliary data changes along a path. Differences important for auxiliary data lead to large distances, and this knowledge is extracted from data.

More formally, the squared distance d_L^2 between two close-by points $\mathbf{x}$ and $\mathbf{x} + d\mathbf{x}$ in the learning metric is defined by the Kullback-Leibler divergence between the conditional distribution of the auxiliary data at the two points, that is,

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

where $\mathbf{J}(\mathbf{x})$ is the Fisher information matrix evaluated at $\mathbf{x}$ by

$$\mathbf{J}(\mathbf{x}) = E_{p(c|\mathbf{x})}\left\{ \left(\frac{\partial}{\partial \mathbf{x}} \log p(c|\mathbf{x}) \right) \left(\frac{\partial}{\partial \mathbf{x}} \log p(c|\mathbf{x}) \right)^T \right\} . \tag{2}$$

The distance (1) is valid for close-by or 'local' points, for which $d\mathbf{x}$ is differentially small. The problem ad-

dressed in this paper appears when data points are further apart. Then the distance is defined as the minimal path integral of local distances, where the minimum is over all paths between the two points.

Another problem is evident in (2): the distribution $p(c|\mathbf{x})$ is usually not known. In this paper the metric is computed by estimating $p(c|\mathbf{x})$ explicitly. An alternative, not discussed here, is to define for the whole system a cost function so that the method can be asymptotically shown to use the learning metric (1) [3, 4].

3 Computation of distances

In practice we need to resort to approximations since only a finite data set and limited computational resources are available. In this section we review approximation methods proposed earlier, and introduce a new one.

3.1 Density estimation

Given a finite data set, we wish to estimate the densities $p(c|\mathbf{x})$ of auxiliary data, to be used for computing the Fisher information matrix (2). In [8] the following estimator was the best in the majority of experiments.

Take a standard mixture of Gaussians, where each generator generates a Gaussian density in the primary data space and a multinomial distribution in the auxiliary space.

The center of the Gaussian of generator j is $\boldsymbol{\theta}_j$ and the common variance σ^2. The multinomial probability of generating class c is parameterized by ψ_{jc}. The probability of choosing generator j is π_j. Hence $\sum_j \pi_j = 1$ and $\sum_c \psi_{jc} = 1$ for all j. The conditional density generated by such a model is

$$\hat{p}(c|\mathbf{x}) = \frac{\sum_j \psi_{jc}\pi_j \exp(-\|\mathbf{x}-\boldsymbol{\theta}_j\|^2/2\sigma^2)}{\sum_j \pi_j \exp(-\|\mathbf{x}-\boldsymbol{\theta}_j\|^2/2\sigma^2)} \,. \quad (3)$$

Both standard Parzen estimators and a joint density model called MDA2 [2] belong to the model family. However, instead of optimizing the model to maximize the joint likelihood of primary and auxiliary data, as in the above-mentioned algorithms, we directly optimize the conditional probabilities needed for the Fisher information matrix, with π_j restricted to be same for all generators, by maximizing the conditional likelihood of the auxiliary data with conjugate gradients.

3.2 Approximations to path integrals

Even when the parametric form of the conditional density is known, the distance defined as a minimal path integral (cf. Section 2) cannot usually be computed in a closed form.

Local distances. The easiest approximation is to simply use the local metric (1) even for longer intervals [5], that is, to compute the distance between the points $\mathbf{x}$ and $\mathbf{m}$ by

$$d_1^2(\mathbf{x},\mathbf{m}) = (\mathbf{x}-\mathbf{m})^T\mathbf{J}(\mathbf{x})(\mathbf{x}-\mathbf{m}) \,, \quad (4)$$

where $\mathbf{J}(\mathbf{x})$ is estimated from (3), replacing $p(c|\mathbf{x})$ by its approximation $\hat{p}(c|\mathbf{x})$. This is called here the '1-point' approximation, since it evaluates the local metric at one point. Since the 1-point approximation ignores changes in the local metric along the path, it is not expected to be accurate over long intervals.

Distances along straight lines. The local approximation can be improved by still assuming that the minimal path between $\mathbf{x}$ and $\mathbf{m}$ is a straight line, but by computing the distance more accurately along it [8]. The line is divided into T evenly spaced segments, and the local approximation is used for each segment. This 'T-point' approximation is computed by

$$d_T(\mathbf{x},\mathbf{m}) = \frac{1}{T}\sum_{i=0}^{T-1}\left(\mathbf{r}^T\mathbf{J}\left(\mathbf{x}+\frac{i}{T}\mathbf{r}\right)\mathbf{r}\right)^{1/2} \,, \quad (5)$$

where $\mathbf{r} = \mathbf{m} - \mathbf{x}$.

In [8] it was shown that the T-point distance approximation improved SOM performance. However, it is computationally more intensive. In Section 4 we discuss a speedup for SOM training and in Section 6.3 we empirically test how much the computation can be sped up while still getting good results.

Global distances by graph approximation. The 1-point and T-point distance approximations assume that the minimum-distance path between two points is (close to) a line. This assumption can be relaxed by choosing a set of points for which pairwise distances are known, and computing the minimal path through the graph formed by connecting the points.

In principle the set of points could be chosen to form a regularly spaced lattice, but the curse of dimensionality would prevent practical computation. We suggest choosing the known data points. It makes sense to compute distances more accurately where there is data. The pairwise distances are computed with the T-point approximation, and finally the minimal path distances are computed with Floyd's algorithm. The time complexity of this new 'graph' approximation is $O(N^3)$ with respect to the number of samples N.

Analogous graph computation has earlier been suggested for distances computed from unsupervised generative models [9].

Choosing the approximation. The 1-point and T-point approximations are suited for algorithms for which accurate approximation of local distances is

more important than of long ones. The SOM which searches for nearest model vectors is an example. The approximations are relatively fast and hence attractive when distances cannot be computed beforehand.

The graph approximation is feasible for algorithms where the set of points (and hence minimal paths) do not change during learning. Methods that use only the matrix of pairwise distances are ideal candidates. In contrast, for algorithms like the SOM where the prototype vectors change, the minimum paths would need to be re-evaluated at each iteration.

The application to SOMs in Section 4 uses the 1-point and T-point approximations, and the application to Sammon's mappings in Section 5 uses the T-point and graph approximations.

4 SOMs with learning metrics

SOMs with learning metrics (called here SOM-L) have been applied to analysis of bankruptcies [5] and gene expression. The algorithm is reviewed here briefly.

Learning. After the density estimator has been constructed, the SOM-L is computed as usual by iterating a winner selection and an adaptation step. Here we consider only the on-line (sequential) SOM algorithm. The SOM computation as such is unsupervised; auxiliary data is not required for the samples. Only the metric has been supervised with the auxiliary data.

The winner (best-matching) SOM unit $w(t)$ at iteration t is chosen by

$$w(t) = \arg\min_i \hat{d}_L^2(\mathbf{x}(t), \mathbf{m}_i(t)) \;, \qquad (6)$$

where $\mathbf{x}(t)$ is the input sample at iteration t, $\mathbf{m}_i$ is the prototype vector corresponding to unit i at iteration t, and $\hat{d}_L^2$ is either of the distance approximations d_1 or d_T discussed in Section 3.2.

The T-point approximation can be sped up by *winnowing* the set of winner candidates by the faster 1-point approximation. The T-point distances are computed only for the W most promising candidates.

After the winner has been chosen, the prototype vectors are adjusted to decrease the distance to the data. Since the new metric is a so-called Riemannian metric, steepest descent direction is not given by the gradient but the so-called natural gradient. For the 1-point estimate the update coincides with the standard SOM adaptation rule

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t) h_{w(t),i}(t)(\mathbf{x}(t) - \mathbf{m}_i(t)) \;, \qquad (7)$$

where $\alpha(t)$ is the learning rate and $h_{w(t),i}(t)$ is the neighborhood function at iteration t.

Improved adaptation. For the T-point approximation the natural gradient in principle changes the update rule slightly; the relative lengths of the T segments adjust the update length. However, the adjustments did not improve performance in preliminary experiments, and we chose to use the simpler adaptation method (7) in further experiments in Section 6.

5 Sammon's mapping by learning metrics

Theory. In methods that start from the *distance matrix* of pairwise (global) distances, it is not necessary to be able to compute distances between arbitrary points. Computing only the distance matrix by learning metrics is sufficient.

As in SOM-L, the aim still is to study the primary data. The distances are based on primary-space differences, and the auxiliary data only guides which differences are relevant. Hence analysis based on the distance matrix is meaningful for and informative of the primary data. By contrast, the tentative alternative of defining the distance based on class index only would disregard the primary data completely.

Learning. The distance matrix can be computed by approximating the global distances as in Section 3.2; both the T-point and the graph approximation are applicable. After the matrix has been computed, we may summarize and visualize the data by any method, such as hierarchical clustering, that works on pairwise distances. The methods can be used as such, irrespective of how the distances were computed.

Visualization. Metric multidimensional scaling (MDS) methods are widely used for visualizing similarities of data samples based on a distance matrix. They construct a (low-dimensional) representation for the data that aims to preserve the distance matrix as Euclidean distances in the output space.

Here we chose Sammon's mapping [10], a variant of MDS whose cost function emphasizes representing small distances accurately. The Sammon's mapping computed in learning and Euclidean metrics will for brevity be denoted Sammon-L and Sammon-E, respectively. We test the algorithms empirically in Section 7.

6 Empirical SOM comparison

6.1 Goodness measure

Previously [8] we have measured SOM performance by conditional likelihood estimates of auxiliary data, at the winner units of test samples. This measure has a slightly unintuitive corollary: since it requires a density estimator, even though traditional SOMs are not

Data set	n	C	N
Landsat Satellite Data [1]	36	6	6435
Letter Recognition Data [1]	16	26	20000
LVQ_PAK (Phoneme) [6]	20	13	3656
TIMIT Data [11]	12	41	14994

Table 1: The data sets and their dimensionality (n), number of classes (C) and samples (N).

trained with such estimators, different estimators yield different results for the same SOM.

We propose to measure SOM accuracy by, in a sense, how 'unmixed' the auxiliary values are in each SOM unit. The measure is smoothed over neighboring units to reward homogeneity of SOM neighborhoods.

To be precise, the accuracy of a SOM is computed as the conditional log-likelihood of N_{TEST} test samples $\{\mathbf{x}_i, c_i\}_{i=1}^{N_{TEST}}$ projected onto the map. We define

$$Accuracy = \sum_{i=1}^{N_{TEST}} \log \frac{\sum_{j=1}^{N_{SOM}} a_{jw_i} \frac{N_{jc_i}}{N_j}}{\sum_{j=1}^{N_{SOM}} a_{jw_i}} , \qquad (8)$$

where N_{ji} is the number of test samples with auxiliary value $c = i$ and winner unit j, and $N_j = \sum_i N_{ji}$. The winner unit of the ith sample is denoted by w_i.

The projected distributions N_{jc_i}/N_j are smoothed over neighboring SOM units by weights a_{jw_i} given by

$$a_{jw_i} = \exp\left(-D^2(j, w_i)/2\lambda^2\right) N_j , \qquad (9)$$

where $D(j, w_i)$ is the distance between units j and w_i on the SOM lattice. We set the smoothing parameter to $\lambda = 1$, equal to the neighborhood function radius at the end of SOM computation. The weights a_{jw_i} are weighted by the number of samples to emphasize more reliable estimates.

According to empirical tests (not shown), both the old and new indicators measure goodness similarly and result in similar choices in parameter and method validation.

6.2 SOM-L vs. SOM-S and SOM-E

We compared SOM-L with the 1-point and T-point distance approximations to supervised SOM (SOM-S) [7] and Euclidean SOM (SOM-E) on four real world data sets (Table 1), used in [8] as well. Class labels were used as the auxiliary values c.

A 10-fold cross-validation was computed and the significance of the difference between SOM-L and the other methods was tested by a paired t-test. The parameter σ and the amount of supervision in SOM-S (the relative weight of the 1-out-of-C encoded class vector) were validated anew for each fold to maximize the accuracy (8) on a validation set sampled from each learning set. The accuracy of the best SOM was then computed on the corresponding test set.

Landsat

	1-point	SOM-S	SOM-E
T-point	$\underline{4 \times 10^{-4}}$	$\underline{10^{-5}}$	$\underline{3 \times 10^{-8}}$
1-point	-	0.04	$\underline{8 \times 10^{-5}}$
SOM-S	-	-	$\underline{10^{-4}}$

Letter

	1-point	SOM-S	SOM-E
T-point	$\underline{6 \times 10^{-8}}$	$\underline{10^{-9}}$	$\underline{< 10^{-10}}$
1-point	-	$\underline{2 \times 10^{-8}}$	$\underline{< 10^{-10}}$
SOM-S	-	-	$\underline{< 10^{-10}}$

LVQ_PAK

	SOM-S	SOM-E	1-point
T-point	$\underline{0.003}$	$\underline{2 \times 10^{-6}}$	$\underline{9 \times 10^{7}}$
SOM-S	-	$\underline{0.008}$	$\underline{0.001}$
SOM-E	-	-	0.20

TIMIT

	SOM-S	1-point	SOM-E
T-point	$\underline{2 \times 10^{-5}}$	$\underline{3 \times 10^{-6}}$	$\underline{3 \times 10^{-9}}$
SOM-S	-	0.02	$\underline{2 \times 10^{-5}}$
1-point	-	-	$\underline{0.004}$

Table 2: P-values of paired t-tests. Presence of an entry means that the method in that row is on the average better than the method in that column. Significant differences ($P < 0.01$) are underlined for convenience.

For SOM-L with T-point distances (5), we used 30 generators for density estimation, and distances were computed with $T = 10$ segments to the $W = 10$ closest model vectors. For SOM-L with the 1-point distance approximation (4), the best number of generators was chosen from 10, 30, and 100 by preliminary experiments.

On all data sets, SOM-L with T-point distances is significantly better than the other methods (Table 2). SOM-L with 1-point distances outperforms SOM-S significantly on the Letter Recognition set and SOM-E on three sets.

Note that SOM-S is significantly more accurate than SOM-E on all sets. This was to be expected, as SOM-S uses more information for training than SOM-E. This finding gives support to the accuracy measure (8).

6.3 Parameter selection

For the T-point distance approximation (5), the accuracy of computed distances depends on the parameter T (number of metric evaluations over a line). The performance of SOM-L is also affected by the speedup parameter W (Section 4). Since the computational burden of SOM-L increases linearly with respect to both T and W, one should use 'minimal' values that produce 'sufficiently' good results.

We empirically tested a range of values for T and W on two datasets (Fig. 1) with preliminary validation data. The accuracy seems to increase roughly linearly

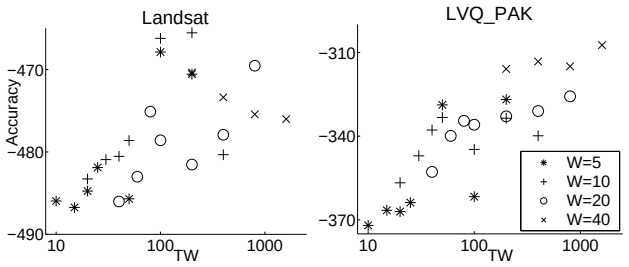

Figure 1: Validation accuracy for Landsat (left) and LVQ_PAK (right) data as a function of the distance parameters T and W, maximized over σ. The SOM-E results (not shown) are -588 for Landsat and -420 for LVQ_PAK. The respective SOM-S results, maximized over class weighting, are -541 and -361.

with the logarithm of TW. The choice $T = 10, W = 10$ used in Section 6.2 is not the best for either dataset. However, since the accuracy gain is slow and small compared to the difference to SOM-E (and to SOM-S for Landsat data), the choice seems acceptable.

7 Empirical comparison for Sammon's mapping

To our knowledge, there exists no "supervised" variant of Sammon's mapping. [1] Hence we need not compare our version to alternatives; it is a new algorithm for a new task. We will, however, carry out a sanity check to ensure the Sammon-L finds class structure from the data better than the normal unsupervised Sammon's mapping.

7.1 Goodness measure

A Sammon's mapping for a given data set (or distance matrix) does not generalize to new data, although heuristic generalizations can and have been developed. Thus, unlike a learned SOM, we cannot test a learned Sammon's mapping easily with validation data.

However, the learned metric can be used for new data. We can validate the usefulness of the metric by using it to compute a distance matrix for validation data; Sammon's mapping for new data is then computed from the distances.

We measure sensibility of the metric by an indirect measure, the performance of a nonparametric classifier in the output space of the Sammon's mapping. We computed the error rate of a K-nearest-neighbor (KNN) leave-one-out classifier for the Sammon's mapping of the validation data. Each sample was classified by majority vote of its K nearest neighbors. In case of a 'tie', each tied class got an equal weight, yielding e.g. 2/3 error if the correct class and two others are tied.

We computed the average performance over 20 mappings with different random initializations.

[1] While e.g. concatenating a 1-out-of-C encoding of the class label to the primary data, similarly to the SOM-S, would help discriminate learning data, a generalization is needed for new (unlabeled) data. The SOM-S uses learned model vectors for generalization, but no obvious method is known to generalize from a learned distance matrix.

Data set	Graph	T-point	Sammon-E
Landsat	88.95	87.49	82.69
Letter	59.26	56.15	14.29
LVQ_PAK	90.77	90.48	80.38
TIMIT	40.00	39.96	30.40

Table 3: Average percentage of correct KNN-classifications for two Sammon-L variants and Sammon-E over cross-validation folds. The Sammon-L variants are both significantly better than Sammon-E on all data sets. 'Graph': Sammon-L with graph search; 'T-point': Sammon-L without graph search.

7.2 Sammon-L vs. Sammon-E

We empirically compared the Sammon-L to the Sammon-E, on the four data sets in Table 1. To keep the computation feasible, a random subset of 5000 samples were picked from the data sets that contained more than 5000 samples.

For Sammon-L we evaluated both the T-point and graph approximations. Both approximations used $T = 10$ segments for distance computation, and the kernel width σ in the 30-kernel density estimate was validated in preliminary experiments to minimize the error rate. The KNN classifier neighborhood K was validated for all methods from the range 1 to 100.

After the parameters had been chosen, a 10-fold cross-validation was computed. In each fold, the metric for Sammon-L was learned from 4500 samples and the Sammon's mappings were computed for a separate 500 samples. The significance of the difference was then tested by paired t-test.

The resulting Sammon-L projections outperformed Sammon-E significantly ($P < 0.01$) by KNN-classification accuracy on all data sets (Table 3). Computing accurate global distances with the graph search (column 'Graph' in Table 3) further improved performance significantly on the Landsat and Letter recognition data sets. On the two other sets the difference between the Sammon-L variants was not significant.

The difference between Sammon-L with the graph approximation and Sammon-E is illustrated in Figure 2 on Letter Recognition data. The learning metric has emphasized differences where the class distributions change, leading to increased class separation and more distinctive clustering of classes.

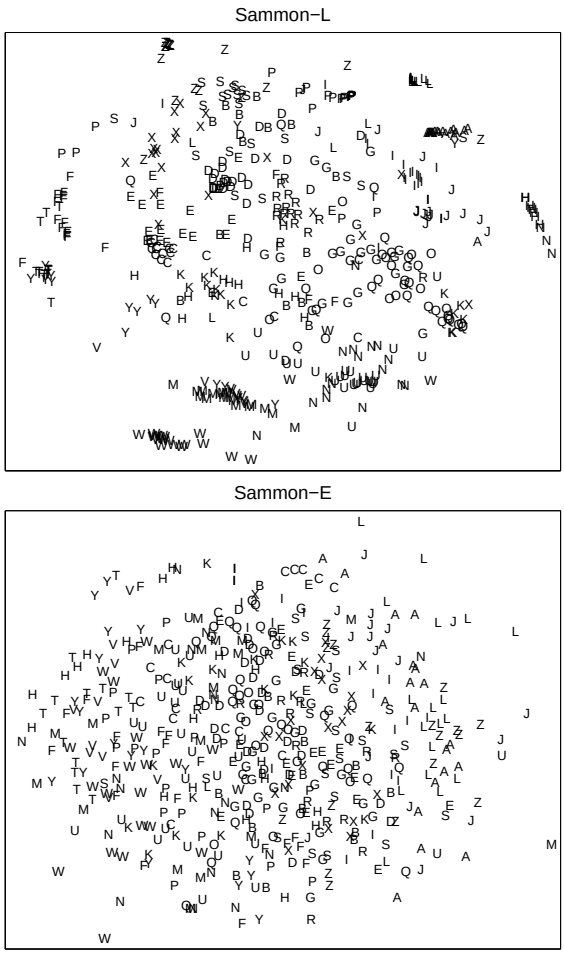

Figure 2: Illustration of the effect of learning metrics for Sammon's mapping on Letter Recognition data. Samples are marked by the letter of their respective class. The class separation is clearly increased in the Sammon-L projection (upper) in contrast to the Sammon-E (lower).

8 Conclusions

The learning metrics principle can be used to construct methods that focus on modeling interesting differences. By choosing suitable auxiliary data the analyst can specify what aspects of the data are interesting. The metric is applicable to several application areas and methods; here we have compared self-organizing maps and Sammon's mappings on standard data sets.

The proposed methods work well in practice, although they are relatively computation-intensive. The main theoretical drawback is that the complexity of the density estimator (smoothness parameter in Section 3.1) is chosen heuristically. A fix is to incorporate the metric to the model [3, 4]. However, this must be done for each model separately, losing the generality of the presented algorithms.

The goodness of visualizations depends on the overall analysis goal, which is hard to formalize as an explicit cost function. We applied simple indirect measures of how much the auxiliary data was mixed in the visualizations. Assuming the auxiliary data is well-chosen, this reveals how well important differences are discernible in the visualizations.

In empirical tests the SOM-L gave significant improvements over the supervised SOM. The improved distance calculation was crucial. Sammon-L constructed more informative visualizations than an unsupervised variant.

Acknowledgements

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

References

[1] C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998.

[2] T. Hastie, R. Tibshirani, and A. Buja. Flexible discriminant and mixture models. In J. Kay and D. Titterington, eds., *Neural Networks and Statistics*. Oxford University Press, 1995.

[3] S. Kaski and J. Peltonen. Informative discriminant analysis. In *Proc. ICML-2003, The Twentieth International Conference on Machine Learning*. Accepted for publication.

[4] S. Kaski and J. Sinkkonen. Principle of learning metrics for data analysis. *The Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology, Special Issue on Data Mining and Biomedical Applications of Neural Networks*, accepted for publication.

[5] S. Kaski, J. Sinkkonen, and J. Peltonen. Bankruptcy analysis with self-organizing maps in learning metrics. *IEEE Transactions on Neural Networks*, 12:936–947, 2001.

[6] T. Kohonen, J. Kangas, J. Laaksonen, and K. Torkkola. LVQ_PAK: A program package for the correct application of Learning Vector Quantization algorithms. In *Proc. of IJCNN'92, International Joint Conference on Neural Networks*, volume I, pp. 725–730, 1992.

[7] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, 2001. Third edition.

[8] J. Peltonen, A. Klami, and S. Kaski. Learning more accurate metrics for self-organizing maps. In J. R. Dorronsoro, ed., *Artificial Neural Networks - ICANN 2002*, pp. 999–1004. Springer, Berlin Heidelberg, 2002.

[9] M. Rattray. A model-based distance for clustering. In *Proc. of IJCNN-2000, International Joint Conference on Neural Networks*, pp. 4013–4016, Piscataway, NJ, 2000. IEEE Service Center.

[10] J. W. Sammon, Jr. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18:401–409, 1969.

[11] CD-ROM prototype version of the DARPA TIMIT acoustic-phonetic speech database, 1998.