LEARNING METRICS FOR EXPLORATORY DATA ANALYSIS

Samuel Kaski
Neural Networks Research Centre
Helsinki University of Technology
P.O. Box 5400, FIN-02015 HUT, Finland
E-mail: samuel.kaski@hut.fi
Web: www.cis.hut.fi

Abstract. Visualization and cluster analysis of multivariate data is usually based on distances between samples in a data space. The distance measure is often heuristically chosen, for instance by choosing suitable features and then using a global Euclidean metric. We have developed methods that remove the arbitrariness by measuring distances only along important (local) directions. The metric is learned from auxiliary data that is paired with the primary data during the learning process. It is assumed that changes in the primary data are important or relevant if they cause changes in the auxiliary data; for example, in analysis of gene expression the auxiliary data can indicate the functional classes of the genes. The new distance measures can be used for instance in clustering and Self-Organizing Map-based data visualization. The methods have so far been applied in analysis of bankruptcy, text documents, and gene expression.

INTRODUCTION

Unsupervised learning methods search for statistical dependencies in data. They are often used as "discovery tools" or exploratory tools to reveal statistical structures hidden in large data sets. It is sometimes even hoped for that "natural properties" of the data could be discovered in a purely data-driven manner, i.e., without any prior hypotheses. Such discoveries are possible, but the findings are always constrained by the choice of the model family and the data set: the variable selection or feature extraction.

In a broader data analysis or modeling framework the choice of the model family specifies which kinds of dependencies are sought for, and the choice of the data variables and their preprocessing and transformations (feature extraction) specifies which aspects of the data are deemed interesting or important. It might even be said that unsupervised learning is always supervised by these choices.

The reason the choice and transformation of the data variables is important is that most unsupervised methods depend heavily on the metric, i.e. the distance measure, of the input space. This applies at least to clustering, probability density estimation, and many visualization methods.

The aim of the present work is to automatically *learn* metrics which measure distances along important or relevant directions, and to use the metrics in unsupervised learning. The laborious implicit supervision by manually tailored feature extraction will be replaced by an automatically learning metric, while discoveries can still be made with unsupervised learning methods within the constraints set by the new metric.

If the importance or relevance is to be learned, some auxiliary information needs to specify what is important. Assume that there exist paired data $(\mathbf{x}, c)$, where $\mathbf{x} \in \mathbb{R}^n$ is the usual (primary) data vector and c is auxiliary data. It is further assumed that changes in the auxiliary data are important. For instance, in bankruptcy analysis $\mathbf{x}$ can be a collection of financial indicators of a company, and c indicates whether the company has gone bankrupt within some years after the indicators were collected.

In the studies discussed in this paper the auxiliary data is discrete, multinomially distributed, and the changes in its distribution, $p(c|\mathbf{x})$, as $\mathbf{x}$ changes are assumed important. The distance d between two close-by data points $\mathbf{x}$ and $\mathbf{x} + d\mathbf{x}$ will be defined as the difference between the corresponding distributions, i.e.

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

where D_{KL} denotes the Kullback-Leibler divergence. The new metric preserves proximity relations (i.e., loosely speaking, the topology) of the data space but changes the originally arbitrary metric.

The definition (1) cannot be used as such in practice since the distributions $p(c|\mathbf{x})$ are not known. The metric will be learned from a finite data set $\{(\mathbf{x}_k, c_k)\}$, $k = 1, \ldots, N$. We have developed two approaches suitable for slightly different purposes; they will be discussed in the next two sections.

LEARNING METRICS EXPLICITLY

The first approach is based on an estimate $\hat{p}(c|\mathbf{x})$ of the conditional probabilities. Several kinds of useful estimators exist; in the examples of this section the probabilities will be derived by applying the Bayes rule to Parzen estimates of the joint probability $p(c, \mathbf{x})$, and to a variant of the mixture density model called Mixture Discriminant Analysis (MDA2; [4]).

The estimate can then be plugged into the definition of the metric (1). It can be shown that, still assuming $d\mathbf{x}$ is small, the metric is given by the local approximation

$$d_L^2(\mathbf{x}, \mathbf{x} + d\mathbf{x}) = d\mathbf{x}^T F(\mathbf{x}) d\mathbf{x} \ , \tag{2}$$

where $F(\mathbf{x})$ is (analogous to) the Fisher information matrix. More details can be found in [9].

The resulting metric resembles the information metric [1, 10] used to measure the distances between generative probabilistic models. The novelty here is that the Fisher information matrix defines a metric *in the data space*. The $\mathbf{x}$ is used as the "parameters" of a model that generates a probability distribution for the auxiliary c values.

The definition (2) gives a local distance measure which could in principle be extended to non-local distances by integrating the local distances along minimal paths in the data space[1]. The integration would, of course, be computationally demanding. In this paper the local distances will be used as approximations. They are feasible for models such as several clustering methods and the Self-Organizing Maps that rely mostly on local distances.

Note that transformations of the data space into a new space change the metric as well, and automatic feature extraction methods could therefore be used as an alternative for explicitly changing the metric. For instance mutual information-like criteria have been used for feature extraction [3, 16] in classification tasks. Note, however, that if the transformation is suitably regural then the same effect can be obtained with a change of the metric; it can be shown that our method is in principle invariant to diffeomorphisms. If, on the other hand, a change of topology is desirable then a suitable transformation can be applied before the metric is learned.

Self-Organizing Maps in Learning Metrics

The learning metrics can be used with several unsupervised and supervised methods. Our first experiments have been made with the Self-Organizing Map (SOM; [12]), an algorithm with applications reported in over 4000 publications (`http://www.cis.hut.fi/research/som-bibl/`; cf. [8]).

The SOM is a method for organizing data on a usually two-dimensional graphical map display which preserves similarity relationships in the data: close-by locations of the display represent similar data. The mapping is defined by a set of model vectors that are attached to discrete grid points on the display, and the model vectors learn to represent the probability distribution of the data in an orderly fashion. The SOM display is especially useful for data visualization and exploratory analysis: It provides an overview of the similarity relationships in large data sets, and additional properties of the data such as cluster structures and distribution of the values of data variables can be visualized on the same display.

The original on-line version of the SOM algorithm consists of two steps that are applied iteratively. Denote the input vector at time step t by $\mathbf{x}(t)$; for a finite data set it is chosen randomly. First the best matching SOM unit

[1]Note that this extension preserves the proximity relations of the data space, whereas the straightforward extension of (1) to non-local distances would not.

w is sought by

$$w = \arg\min_j d^2(\mathbf{x}, \mathbf{m}_j) \ , \tag{3}$$

where $\mathbf{m}_j$ denotes the model vector attached to the map unit j and d is the distance measure, usually the Euclidean distance. The model vectors are then updated towards the negative gradient of their (squared) distance from the data point.

When computing the SOM in the new metric, the best matching unit will be searched for using the local approximation of the new metric (2). It is assumed that replacing the true (global) distance with the approximation does not change the winner, which is sensible since the potential winners are usually close to the data point in both metrics. The ultimate test of the assumption is, of course, in the applications.

When updating the model vectors to decrease their distance from the data vector it is essential to note that the new metric is non-Euclidean. Hence, the direction of the steepest descent is given by the natural gradient [6]. It is straightforward to show that when the new metric and the natural gradient are used in the SOM algorithm, its update rule becomes the usual update rule of the Euclidean SOM, i.e.,

$$\mathbf{m}_j(t+1) = \mathbf{m}_j(t) + h_{wj}(t)(\mathbf{x}(t) - \mathbf{m}_j(t)) \ . \tag{4}$$

Here h_{wj} is the *neighborhood function*, a decreasing function of the distance between the units w and j on the map grid. The height and width of the neighborhood function are decreased during learning.

In summary, the best matching unit is sought with (3), where the distance is given by (2). The model vectors are then updated according to (4). The computational complexity is somewhat higher than that of the Euclidean SOM; in the bankruptcy case study discussed below it is about doubled. For more details see [9].

A Demonstration. The difference between SOMs computed in the learning metric and in the Euclidean metric is demonstrated with a seemingly simple data set in Figure 1. The data is three-dimensional but the distribution of the auxiliary data changes only in two dimensions. Hence, assuming changes in the auxiliary data only are important, the third (vertical) dimension is irrelevant and resources are wasted if it is modeled. It can be seen in Figure 1 that the SOM in the new metric (derived from Parzen estimates) does not model the vertical direction at all while representing the horizontal directions in an ordered fashion. The Euclidean SOM, in contrast, folds itself while trying to model the whole three-dimensional data distribution.

Application: Bankruptcy Analysis. Quantitative analyses of bankruptcy commonly aim at either prediction of bankruptcies, or rating of companies based on the probability of bankruptcy. Kiviluoto and Bergius (see e.g. [11]) have used SOM-based analyses and visualizations to pursue a more comprehensive quantitative picture of corporate behavior than that provided by a single figure of merit.

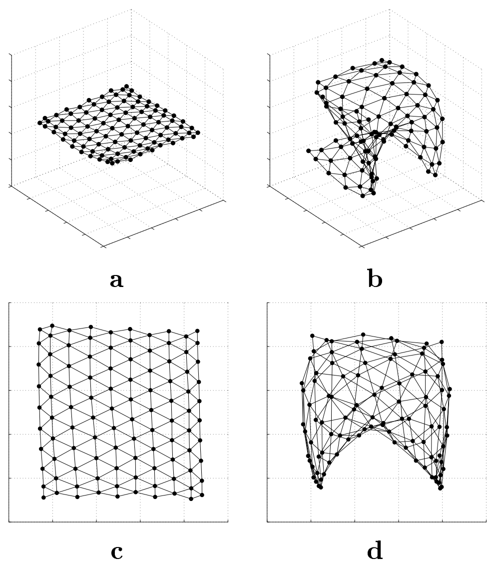


Figure 1: SOMs that model data distributed uniformly within a three-dimensional cube. The class distribution of the data changes only in the horizontal direction, and the changes are assumed to indicate importance or relevance. (**a**) SOM in the new metric, (**b**) SOM in the Euclidean metric. Projections from above: **c**: New metric, **d**: Euclidean metric. Details omitted for brevity.

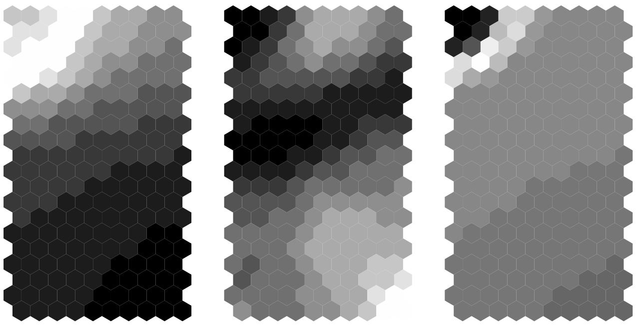
Since the bankruptcy risk is, after all, perhaps the most important single indicator, we have used the knowledge of whether the company has gone bankrupt within 3 years as the auxiliary data c to guide the analysis of a further set of 23 indicators [9]. The SOM computed in the learning metric was more accurate than the Euclidean SOM in estimating the bankruptcy risk ($P < 0.002$ for estimates computed at the best-matching units). The resulting SOM was well-organized, and it can be used for visualizing both the probability of bankruptcy (Fig. 2**a**) and the distribution of the financial indicators (a sample indicator is shown in Fig. 2**b**).

Moreover, the *contributions* of the indicators to the metric can be visualized on the map (Fig. 2**c**) to assess which indicators are important for the companies occupying different locations of the map.

LEARNING METRICS IMPLICITLY: CLUSTERING BY SIMILARITY IN AN AUXILIARY SPACE

The method presented in the previous section is a viable data analysis tool. However, since the approach requires a separate probability estimation stage involving some approximations, we have additionally pursued alternative methods.

The method presented below is applicable to clustering and vector quantization tasks. It will be introduced by generalizing the cost function of



a **b** **c**

Figure 2: A SOM of companies. Each company has a certain location on the map, and each location represents different kinds of companies. **a** Estimate of the probability of bankruptcy at each map unit. **b** Distribution of the values of a sample financial indicator (liquidity) on the map. **c** The (relative) contribution of the liquidity indicator to the change in bankruptcy sensitivity. White: largest values, black: smallest values.

"soft" vector quantization, and then presenting a stochastic approximation algorithm for minimizing it.

In (soft) vector quantization the goal is to find a set of prototypes or codebook vectors $\mathbf{m}_j$ that minimizes the average distortion E made by representing samples by the codebook vectors. The average distortion is

$$E = \sum_j \int y_j(\mathbf{x}; \mathbf{M}) D(\mathbf{x}, \mathbf{m}_j)\, p(\mathbf{x})\, d\mathbf{x} \ , \tag{5}$$

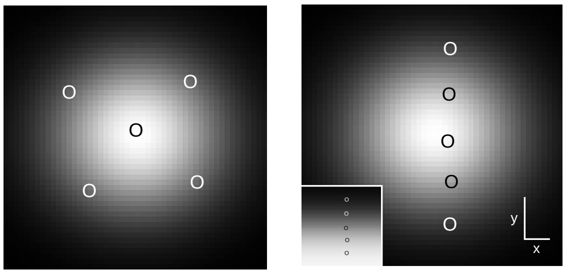
where $D(\mathbf{x}, \mathbf{m}_j)$ is the distortion made by representing $\mathbf{x}$ by $\mathbf{m}_j$, and $y_j(\mathbf{x}; \mathbf{M})$ is the "soft" cluster membership function that fulfills $0 \leq y_j(\mathbf{x}) \leq 1$ and $\sum_j y_j(\mathbf{x}) = 1$ for all $\mathbf{x}$. The $\mathbf{M}$ denotes the matrix consisting of all the model vectors.

Vector quantization will next be generalized to measure the distortions in an auxiliary space, while still keeping the clusters local in the primary data space. This can be done by introducing an additional set of distributional prototypes $\boldsymbol{\psi}_j$, one for each cluster j, and by replacing the distortion measure with the Kullback–Leibler divergence in the auxiliary space. With the new distortion measure the cost function of vector quantization becomes

$$E_{KL} = \sum_j \int y_j(\mathbf{x}; \mathbf{M}) D_{KL}(p(c|\mathbf{x}), \boldsymbol{\psi}_j)\, p(\mathbf{x})\, d\mathbf{x} \ . \tag{6}$$

The clusters are still defined in terms of the primary data space, by the memberships $y_j(\mathbf{x}; \mathbf{M})$.

The cost function (6) is to be minimized with respect to both sets of prototypes, the $\mathbf{m}_j$ and the $\boldsymbol{\psi}_j$. Below a simple stochastic approximation algorithm [14] will be presented for membership functions of the normalized von Mises-Fisher form, i.e. spherical analogues of Gaussians.



a **b**

Figure 3: Demonstration of (**a**) K-means and (**b**) learning metrics clustering of a simple data set. The pdf of the data is shown in shades of gray, the cluster centers as circles, and the distribution of one of the two classes (auxiliary data) as shades of gray in the inset. (Note that the data on the right is singular, and for example a solution in which the units are ordered in two columns is almost as good in terms of the cost function.)

Denote the i.i.d. data pair at step t by $(\mathbf{x}(t), c(t))$, and the index of the value of $c(t)$ by i. Draw two clusters, j and l, from a multinomial distribution with probabilities given by the membership functions $\{y_k(\mathbf{x}(t))\}_k$. Reparameterize the distributional prototypes by a "soft max" of the new parameters γ_{lm}, defined by $\log \psi_{ji} = \gamma_{ji} - \log \sum_m \exp(\gamma_{jm})$, to keep them summed up to unity. Adapt the prototypes by

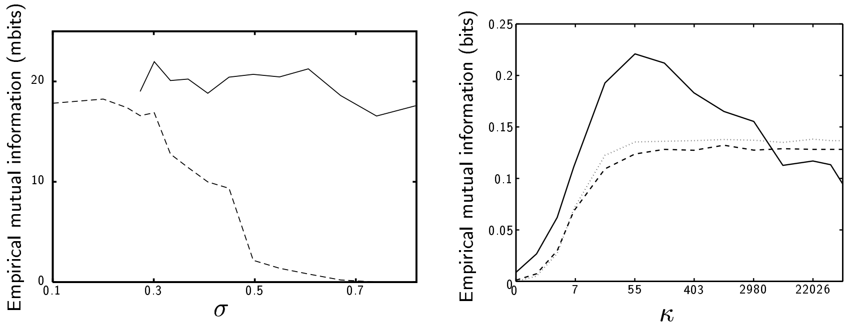
$$
\begin{aligned}
\mathbf{m}_l(t+1) &= \mathbf{m}_l(t) - \alpha(t)\left[\mathbf{x}(t) - \mathbf{x}(t)^T \mathbf{m}_l(t)\mathbf{m}_l(t)\right] \log \frac{\psi_{ji}(t)}{\psi_{li}(t)} && (7)\\
\gamma_{lm}(t+1) &= \gamma_{lm}(t) - \alpha(t)\left[\psi_{lm}(t) - \delta_{mi}\right] , && (8)
\end{aligned}
$$

where δ_{mi} is the Kronecker delta. Due to the symmetry between j and l, it is possible to adapt the parameters twice for one t by swapping j and l in (7) and (8) for the second adaptation. Note that $\mathbf{m}_l(t+1) = \mathbf{m}_l(t)$ if $j = l$. During learning the parameter $\alpha(t)$ decreases gradually towards zero according to a schedule that fulfills the conditions of the stochastic approximation theory. Convergence of the algorithm has been proven [7].

Similar algorithms can be easily derived for other kinds of membership functions, such as Gaussians [14].

It can be shown that the cost function is equal to the *mutual information* (plus a constant) between the auxiliary data and the clusters, interpreted as multinomially distributed random variables.

A Demonstration. Figure 3 shows a simple example in which the data is Gaussian and the auxiliary distribution changes linearly in the y-direction (see the inset in Fig. 3**b**). The optimal local clusters that have homogeneous class distribution are horizontal slices, and hence the optimal configuration for the cluster centers is along a line in the y-direction (Fig. 3**b**). The solution found by the familiar K-means clustering is shown for reference in Figure 3**a**.



a **b**

Figure 4: Mutual information between the clusters and the auxiliary variables for (**a**) bankruptcy data and (**b**) gene expression data. Solid line: clustering in learning metrics; dotted line: plain mixture model; dashed line: MDA2; σ: standard deviation of (normalized) Gaussian membership functions; and κ: parameter governing the width of vMF basis functions, cf. [14]. There were 10 clusters in **a** and 8 clusters in **b**. In **a** the learning algorithm had numerical problems for small σ; we are currently investigating alternatives to still improve the results.

Applications. The clustering method has already been applied to several kinds of data sets including text document collections, gene expression data, and financial indicators of companies. Suitable auxiliary data has been used to "guide" the clustering to concentrate on the important aspects of the primary data.

For the text documents we have used as the auxiliary information the keywords provided by the authors of the documents. The clustering then automatically utilizes the manual work of the authors for inferring what is important in the texts.

In bankruptcy analysis the companies were clustered using the bankruptcy risk as the auxiliary data; details have been provided in the previous section. Figure 4**a** shows the mutual information between the bankruptcy risk and the clusters. The clusters obtained with our method convey more information about bankruptcy than an alternative "baseline" method that can be used for the same purpose, Mixture Discriminant Analysis (MDA2; [4]).

In earlier studies on expression profiles of genes, i.e. measures of their expression in various experimental settings, the profiles have either been clustered with traditional clustering methods, or classified according to known classes. The new method aims at combining the good sides of both: Prior knowledge, embedded for example in the known functional classes of the proteins the genes encode, can be utilized as auxiliary data while still retaining the possibility for new discoveries. The mutual information between clusters of yeast genes and the functional classes is shown in Figure 4**b**; it is larger for learning metrics-based clustering than alternative methods. For more details see [14].

Related Methods. Mutual information has been used previously as a cost function in a closely related method for constructing neural representations and for clustering [2]. Our main contributions are the simple stochastic approximation algorithm and the interpretation as a VQ-like algorithm. Interestingly, the algorithm resembles Hebbian learning: Since only "active" units are updated, the term $\mathbf{x}(t)$ within the parentheses in (7) can be considered a Hebbian term, and $\mathbf{x}(t)^T\mathbf{m}_l(t)\mathbf{m}_l(t)$ a forgetting term. The difference from traditional Hebbian learning is the multiplier $\log(\psi_{ji}/\psi_{li})$ which guides the clustering by taking the auxiliary information into account. The resulting learning algorithm bears a resemblance to the Learning Vector Quantization (LVQ; [12]) algorithms.

Another related line of work is the distributional clustering paradigm [5, 13, 15]. From the practical viewpoint there seem to be two main differences. First, in the distributional clustering works the primary data space has always been discrete whereas in the present work it is a continuous vector space—in effect the best possible (smooth) quantization of the continuous space is sought. Second, the computation of the distributional clusters requires an estimate of the joint probability distribution of the primary and the auxiliary data. Hence, a new sample cannot be clustered without knowing the corresponding distribution of the auxiliary variable.

DISCUSSION

We have introduced methods for deriving (local) metrics for data spaces such that the metrics concentrate on measuring important or relevant differences and disregard the irrelevant ones. The metrics are learned from a data set in which the primary data is paired with auxiliary data. It is assumed that changes in the auxiliary data are important, and changes in the primary data space are relevant if they cause changes in the auxiliary data.

The methods of course rely on the availability of suitable auxiliary data. Natural candidates are the signals used for supervision in usual supervised tasks, i.e., regression and classification. The methods can, however, be used more generally for exploring the statistical dependencies between two data sets.

Two approaches were introduced for learning metrics for data exploration. The first alternative is to begin by constructing an estimator of the probability distribution of the auxiliary data given the primary data, and then to derive the metric based on the estimator. The problem of which kinds of estimators are best for this purpose is still open.

The second approach is to incorporate the new metric directly into the cost function of, for instance, clustering algorithms. Both approaches have already been used successfully in data-analysis applications.

Acknowledgements

The development of the methods has been a joint work with Janne Sinkkonen. I would also like to thank Jaakko Peltonen and Janne Nikkilä for their contributions.

REFERENCES

[1] S.-I. Amari, **Differential-Geometrical Methods in Statistics**, New York: Springer, 1990.

[2] S. Becker, "Mutual Information Maximization: Models of Cortical Self-Organization," **Network: Computation in Neural Systems**, vol. 7, pp. 7–31, 1996.

[3] J. W. Fisher III and J. Principe, "A Methodology for Information Theoretic Feature Extraction," in **Proc. IJCNN'98, International Joint Conference on Neural Networks**, Piscataway, NJ: IEEE Service Center, vol. 3, pp. 1712–1716, 1998.

[4] 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.

[5] T. Hofmann, J. Puzicha and M. I. Jordan, "Learning from Dyadic Data," in M. S. Kearns, S. A. Solla and D. A. Cohn (eds.), **Advances in Neural Information Processing Systems 11**, San Mateo, CA: Morgan Kaufmann Publishers, pp. 466–472, 1998.

[6] S.-I. Amari, "Natural Gradient Works Efficiently in Learning," **Neural Computation**, vol. 10, pp. 251–276, 1998.

[7] S. Kaski, "Convergence of a Stochastic Semisupervised Clustering Algorithm," Techn. Report A62, **Helsinki University of Technology, Publications in Computer and Information Science**, Espoo, Finland, 2000.

[8] S. Kaski, J. Kangas and T. Kohonen, "Bibliography of Self-Organizing Map (SOM) Papers: 1981–1997," **Neural Computing Surveys**, vol. 1, no. 3&4, pp. 1–176, 1998.

[9] 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.

[10] R. E. Kass and P. W. Vos, **Geometrical Foundations of Asymptotic Inference**, New York: Wiley, 1997.

[11] K. Kiviluoto and P. Bergius, "Exploring Corporate Bankruptcy with Two-Level Self-Organizing Maps. Decision Technologies for Computational Management Science," in **Proceedings of Fifth International Conference on Computational Finance**, Boston: Kluwer Academic Publishers, pp. 373–380, 1998.

[12] T. Kohonen, **Self-Organizing Maps**, Berlin: Springer, 1995, (third, extended edition 2001).

[13] F. Pereira, N. Tishby and L. Lee, "Distributional Clustering of English Words," in **Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics**, pp. 183–190, 1993.

[14] J. Sinkkonen and S. Kaski, "Clustering Based on Conditional Distributions in an Auxiliary Space," **Neural Computation**, 2001, accepted for publication.

[15] N. Tishby, F. C. Pereira and W. Bialek, "The Information Bottleneck Method," in **37th Annual Allerton Conference on Communication, Control, and Computing**, Urbana, Illinois, 1999.

[16] K. Torkkola and W. Campbell, "Mutual Information in Learning Feature Transformations," in **Proceedings of the 17th International Conference on Machine Learning**, Stanford, CA, USA: Morgan Kaufmann, pp. 1015–1022, 2000.