Principle of Learning Metrics for Exploratory Data Analysis

SAMUEL KASKI AND JANNE SINKKONEN[1]
Neural Networks Research Centre
Helsinki University of Technology
P.O. Box 9800, FIN-02015 HUT, Finland
{*samuel.kaski,janne.sinkkonen*}*@hut.fi*

Abstract. Visualization and clustering of multivariate data are usually based on mutual distances of samples, measured by heuristic means such as the Euclidean distance of vectors of extracted features. Our recently developed methods remove this arbitrariness by learning to measure important differences. The effect is equivalent to changing the metric of the data space. It is assumed that variation of the data is important only to the extent it causes variation in *auxiliary data* which is available paired to the primary data. The learning of the metric is supervised by the auxiliary data, whereas the data analysis in the new metric is unsupervised. We review two approaches: a clustering algorithm and another that is based on an explicitly generated metric. Applications have so far been in exploratory analysis of texts, gene function, and bankruptcy. Connections for the two approaches are derived, which leads to new promising approaches to the clustering problem.

Keywords: Discriminative clustering, exploratory data analysis, Fisher information matrix, information metric, learning metrics

1 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, including the variable selection or feature extraction.

In a broader data analysis or modeling framework the model family specifies which kinds of dependencies are sought for. Typically, selection of the data variables and their preprocessing (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 choice and transformation of the data variables is important because most unsupervised methods depend heavily on the metric, i.e. the distance measure, of the input space. Such methods include at least clustering, probability density estimation, and most 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 to a large extent be replaced by an automatically learned metric,

[1]The authors contributed equally to the work.

while discoveries can still be made with unsupervised learning methods within the constraints set by the new metric.

The methods presented in this paper are applicable when there exist suitable *auxiliary data* which implicitly reveals what is relevant or important in the primary data. Such auxiliary data are available at least in the settings where supervised learning methods, regression and classification, are usually applied. The difference here is that the goal is to model and understand the *primary data* and learn what is relevant there, whereas in supervised learning the sole purpose is to predict the auxiliary data. In practical data analysis the analyst of course needs to find or choose suitable auxiliary data, and the quality of the results is determined by this choice.

Consequently, we assume that the data comes in pairs $(\mathbf{x}, c)$: the primary data vectors $\mathbf{x} \in \mathbb{R}^n$ are always paired with auxiliary data c which in this paper are discrete. Important variation in $\mathbf{x}$ is supposed to be revealed by variation in the conditional density $p(c|\mathbf{x})$.

The distance d between two close-by data points $\mathbf{x}$ and $\mathbf{x} + d\mathbf{x}$ is defined as the difference between the corresponding distributions of c, measured by the Kullback-Leibler divergence D_{KL}, i.e.

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

Bankruptcy risk is an example of auxiliary data that indicates importance in the analysis of the financial states of companies: the $c = 1$ if the company goes bankrupt and $c = 0$ if it stays alive.

Note that although the Kullback-Leibler divergence is not a metric per se, it is symmetric for small distances and therefore locally a metric (see Section 2 for details; note that locality can be relaxed by extending the metric). Proximity relations (i.e., loosely speaking, the topology) of the data space are preserved, but the arbitrariness of feature selection is removed by locally re-scaling the data space to make it reflect important variation in data.

We call the idea of measuring distances in the data space by approximations of (1) the learning metrics principle. Following the principle, we have so far developed a clustering algorithm [1] and a way to explicitly generate metrics for practical purposes such as visualization [2]. These methods are reviewed in Sections 2 and 3, and in Section 4 they are shown to have a close connection, which leads to new promising approaches to the clustering problem.

2 Learning Metrics Explicitly

The first approach is based on an explicit 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 using the Bayes rule from the Parzen and MDA2 (Mixture Discriminant Analysis; [3, 4]) estimates of the joint density $p(c, \mathbf{x})$.

The estimate $\hat{p}(c|\mathbf{x})$ can be plugged into the definition of the metric (1). Local instances $d\mathbf{x}$ are then approximated by the quadratic form

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

where $\hat{\mathbf{J}}(\mathbf{x})$ is an approximation to the Fisher information matrix, computed from $\hat{p}(c|\mathbf{x})$. For details, see [2].

The resulting metric resembles the information metric [5, 6] used to compare generative probabilistic models. The novelty here is that now 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 (Riemann metric) by integrating the local distances along minimal paths in the data space. (Note that this extension preserves the proximity relations of the data space, whereas the straightforward application of (1) to non-local comparisons would not.) The integration would, of course, be computationally demanding. We have used the local distances 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 change the metric as well, and automatic feature extraction methods are therefore an alternative for explicitly changing the metric. For instance, mutual information-like criteria have been used for feature extraction [7, 8] in classification tasks. Note, however, that the effects of all suitably regular transformations (and more) can be obtained with a change of the metric; it can be shown that our principle 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.

2.1 Self-Organizing Maps in Learning Metrics

Learning metrics can be used with several unsupervised and supervised methods. Our first experiments have been carried out with the Self-Organizing Map (SOM; [9]), an algorithm with applications reported in over 4000 publications (`http://www.cis.hut.fi/research/som-bibl/`; cf. [10]).

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 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 toward 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 sought using the local approximation (2) of the new metric. 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.)

While updating the model vectors to decrease their distance from the data vector it is essential to keep in mind that the new metric is non-Euclidean, and that therefore the direction of the steepest descent does not coincide with the ordinary gradient. Instead, the correct direction is given by the so called natural gradient [11]. When the new metric and the natural gradient are used in the SOM algorithm, certain cancellations occur and the update rule becomes that 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 usual *SOM 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 SOM is computed by iterative application of two expressions. First the best matching unit is sought with (3), where the distance is given by (2). Second, the model vectors are 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 [2].

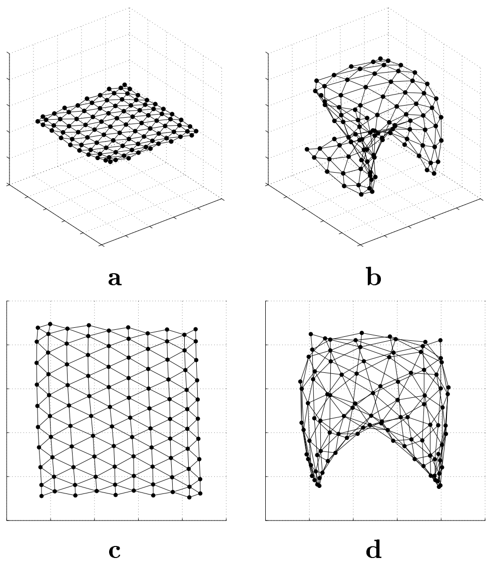


Figure 1: Two SOMs modeling data distributed uniformly within a three-dimensional cube. The class distribution of the data changes only in the horizontal direction, and the changes are assumed to indicate importance or relevance. (**a**) SOM in the new metric, (**b**) SOM in the Euclidean metric. Projections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

2.1.1 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 only the auxiliary data 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 it represents the horizontal directions in an ordered fashion. The Euclidean SOM, in contrast, folds itself while it tries to model the whole three-dimensional data distribution.

2.1.2 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. [12]) 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.

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 three years as the auxiliary data c to guide the analysis of a set of further 23 indicators [2]. 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.

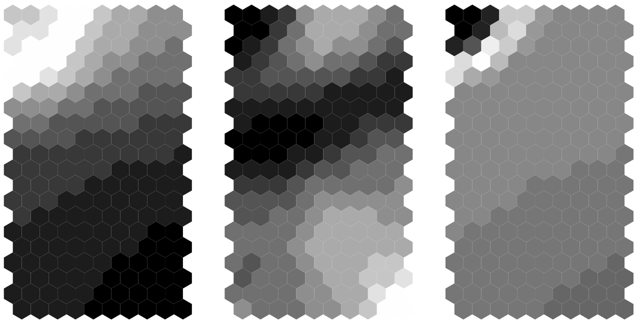


Figure 2: A SOM of companies. Each company has a certain location on the map, and each location represents different kinds of companies. **a** Estimate of the probability of bankruptcy at each map unit. **b** Distribution of the values of a sample financial indicator (liquidity) on the map. **c** The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

3 Learning Metrics Implicitly: Discriminative Clustering

In this section we describe an application of the learning metric principle to clustering.

It would be possible to apply the method of Section 2 to clustering as well, by explicitly constructing a density estimator to derive the new metric and using it with a conventional clustering algorithm. Here we apply the new distance (1) more directly, however. This is potentially more optimal than the use of a separate density estimator, for we do not know which kind of estimators perform well with the learning metrics (this question is discussed further in Section 4.4).

A general goal of clustering is to minimize within-cluster distortion or variation, and to maximize between-cluster variation. We apply the learning metrics by replacing the distortion measure within each local cluster by a kind of within-cluster Kullback-Leibler divergence. This causes the clusters to be internally as homogeneous as possible in $p(c|\mathbf{x})$—the other side of the coin is that between-cluster differences in $p(c|\mathbf{x})$ are maximized.

Below, we introduce the cost function of discriminative clustering. First, the cost of classic vector quantization is presented in a "soft" form, then it is generalized to incorporate the auxiliary variable c. Hard clustering is included in the formulation as a limit case.

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 caused if one represents data by the prototypes:

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

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

We have generalized the cost function (5) to measure the distortions of the *distributions* $p(c|\mathbf{x})$ caused by representing the conditional distribution of c at $\mathbf{x}$ by a partition-wise prototype, denoted by $\boldsymbol{\psi}_j$:

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

Note that the clusters are still defined and kept local in the primary data space, by the memberships $y_j(\mathbf{x};\mathbf{M})$. Distortion between distributions is measured by the Kullback-Leibler divergence.

The cost (6) is minimized with respect to both sets of prototypes, $\mathbf{m}_j$ and $\boldsymbol{\psi}_j$. This can be done by a simple, gradient-based on-line stochastic approximation algorithm [1].

For normalized von Mises-Fisher (vMF) type membership functions $y_j(\mathbf{x}) = Z^{-1}(\mathbf{x})e^{-\kappa \mathbf{x}^T\mathbf{m}_j}$ (with $Z(\mathbf{x})$ such that $\sum_j y_j(\mathbf{x}) = 1$), defined on the hypersphere where $\|\mathbf{x}\| = \|\mathbf{m}\| = 1$, the algorithm is the following. Denote the i.i.d. data pair at on-line step t by $(\mathbf{x}(t), c(t))$ and index the (discrete) value of $c(t)$ by i, that is, $c(t) = c_i$. Draw two clusters, j and l, independently with probabilities given by the membership functions $\{y_k(\mathbf{x}(t))\}_k$. Reparameterize the distributional prototypes by the "soft-max", $\log \psi_{ji} = \gamma_{ji} - \log \sum_m \exp(\gamma_{jm})$, to keep them summed up to unity. Adapt the prototypes by

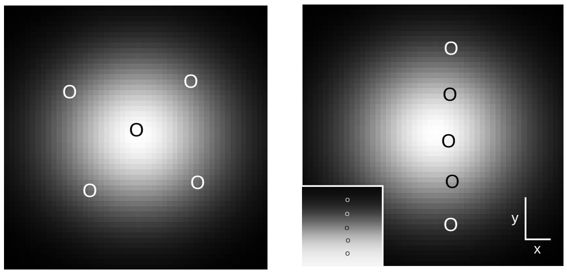
$$\begin{aligned} \mathbf{m}_l(t+1) &= \mathbf{m}_l(t) - \alpha(t)\left[\mathbf{x}(t) - \mathbf{x}(t)^T\mathbf{m}_l(t)\mathbf{m}_l(t)\right] \log \frac{\psi_{ji}(t)}{\psi_{li}(t)} && (7)\\ \gamma_{lm}(t+1) &= \gamma_{lm}(t) - \alpha(t)\left[\psi_{lm}(t) - \delta_{mi}\right] \;, && (8) \end{aligned}$$

where δ_{mi} is the Kronecker delta. Due to the symmetry between j and l, it is possible (and apparently beneficial) to adapt the parameters twice for each t by swapping j and l in (7) and (8) for the second adaptation. Note that no updating takes place if $j = l$, i.e. then $\mathbf{m}_l(t+1) = \mathbf{m}_l(t)$. During learning the parameter $\alpha(t)$ decreases gradually toward zero according to a schedule that fulfills the conditions of the stochastic approximation theory. Convergence of the algorithm for the vMF-type membership functions has been proven [13].

Similar algorithms can easily be derived for other kinds of membership functions [1]. For example, for the normalized Gaussians, $y_j(\mathbf{x}) = Z^{-1}(\mathbf{x})e^{-\|\mathbf{x}-\mathbf{m}_j\|^2/\sigma^2}$, the first adaptation step becomes

$$\mathbf{m}_l(t+1) = \mathbf{m}_l(t) - \alpha(t)\left[\mathbf{x}(t) - \mathbf{m}_l(t)\right] \log \frac{\psi_{ji}(t)}{\psi_{li}(t)} \;.$$

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 a discrete random variable.



a **b**

Figure 3: A demonstration of (**a**) the K-means and (**b**) the 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 in that one of the directions of the input space is totally meaningless, and for example a solution in which the units are ordered in two columns is almost as good in terms of the cost function.)

The algorithm defines soft clusters even though hard, non-overlapping clusters would be more suitable for data exploration-type of tasks where the data is to be reduced to a small number of prototypes in order to make a large data set more comprehensible. The reason for using soft clusters to approximate hard ones is computational: the cluster structure is most efficiently optimized by gradient-based algorithms, and at the limit of hard clusters only data exactly at the border of the clusters, sets of probability measure zero, affect the gradient. Hard clusters are therefore not optimizable by a gradient algorithm. (In the algorithm presented above, we would almost always have $j = l$, with no updating taking place.)

In practice some kind of heuristic assumptions may be applicable to replace the real gradient by an approximation which is actually computable, and we are currently investigating these possibilities.

3.1 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**.

3.2 Applications

The clustering method has already been applied to several kinds of data sets, including text documents, gene expression patterns, and company bankruptcies.

Application-specific 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 keywords provided by the authors of the documents. The clustering then automatically utilizes the manual work of the authors to infer 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 and the clusters. The clusters obtained with our method convey more information about

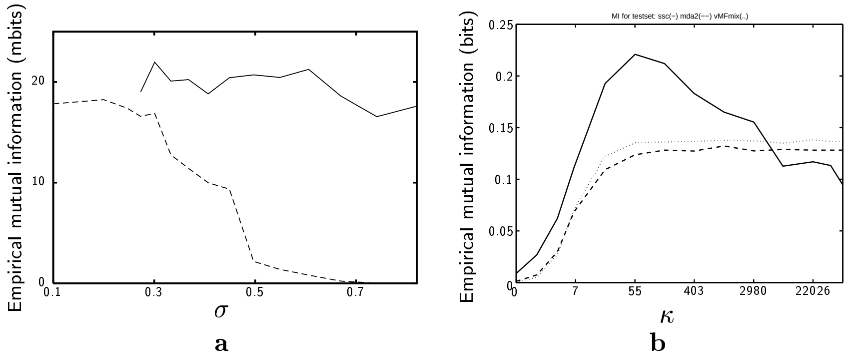


Figure 4: Mutual information between the clusters and the auxiliary variables for (**a**) bankruptcy data and (**b**) gene expression data. Solid line: clustering in learning metrics; dotted line: plain mixture model; dashed line: MDA2; σ: standard deviation of (normalized) Gaussian membership functions; and κ: parameter governing the width of vMF basis functions, cf. [1]. 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 improve the results further.

bankruptcy than an alternative "baseline" method that can be used for the same purpose (Mixture Discriminant Analysis 2 or MDA2; [3, 4]).

In earlier studies of gene expression data (measures of gene "activity" in various experimental settings), the expression profiles have either been clustered with traditional clustering methods or classified to *a priori* known categories. Discriminative clustering combines the good sides of both approaches: prior knowledge in the form of a known classification can be incorporated to an exploratory cluster analysis. The knowledge can be about e.g. gene functions or on properties of the proteins produced by the genes. In Figure 4**b**, the performance of the discriminative clustering algorithm is compared to an alternative method in the analysis of yeast genes and their functional classes; for details, see [1].

3.3 Related Methods

Mutual information has been used earlier as a cost function to construct neural representations and clusterings [14]. Compared to that work, our main contributions in the discriminative clustering algorithm are the simple stochastic approximation algorithm and its interpretation as a distortion-minimizing VQ-like algorithm.

Interestingly, the on-line 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 also bears resemblance to the Learning Vector Quantization (LVQ; [9]) algorithms.

Another related line of work is distributional clustering of discrete co-occurrence data [15, 16, 17], where both the primary and the auxiliary data spaces have been discrete. In contrast, we cluster a continuous vector space, or effectively try to find the best possible quantization of the space. If distributional clustering was generalized to continuous spaces the clusters would become non-local, defined simply as any collections of the points $\mathbf{x}$ of the primary data space with homogeneous class distributions $p(c|\mathbf{x})$. Our goal in this work has been to explore and mine the primary data space, and therefore we have preferred *local* clusters or partitions.

Asymptotically, the shape of the local regions has certain "metric" properties which the non-restricted partitions lack (details in Section 4).

From the practical viewpoint, another difference is that a distributional clustering solution defines cluster identities only for pairs $(\mathbf{x}, p(c|\mathbf{x}))$ where the distribution $p(c|\mathbf{x})$ needs be known or estimated. After all, this is why it is called *distributional* clustering. Our clusters are defined into the primary data space, and hence the cluster identities are defined for single samples $\mathbf{x}$. Our goal has been to quantize the primary data space $\mathcal{X}$ in a way which discriminates different distributions $p(c|\mathbf{x})$ well, hence we have coined the term *discriminative* clustering.

We will return to distributional clustering briefly in Section 4.5 where optimal partition shapes of discriminative clustering are considered.

4 Connections Between the Two Alternative Ways of Learning Metrics

The two methods presented in Sections 2 and 3 are alternative approaches to the same problem of deriving informative representations with the help of auxiliary data. Besides having a common goal, the methods have theoretical connections that will be derived in this section for the asymptotic case of local clusters, under some simplifying assumptions. We will concentrate on "hard" forms of clustering or quantization, meaning that partition cells are disjoint: The membership functions $y_j(\mathbf{x})$ of Section 3 get only values zero and one. This holds asymptotically for the soft Gaussian or vMF partitions used in the practical algorithms when $\sigma \to 0$ or $\kappa \to \infty$, respectively. Hence, the results are asymptotic in the sense that the number of clusters increases and they become harder.

4.1 Asymptotically, Discriminative Clustering is Vector Quantization in Fisher Metrics

At the limit of hard clusters the cost function of discriminative clustering (6) becomes

$$E_{KL} = \sum_j \int_{V_j} D_{KL}(p(c|\mathbf{x}), \boldsymbol{\psi}_j)\, p(\mathbf{x})\, d\mathbf{x} \,. \tag{9}$$

At the limit of zero dispersion (σ or κ^{-1}), the membership of (almost) all points $\mathbf{x}$ is one for one partition cell, indexed below by $j(\mathbf{x})$, and zero for the others. In the Euclidean case with normalized Gaussian membership functions the partitions become *Voronoi regions*: $j(\mathbf{x})$ points to the cluster for which the distance $d(\mathbf{x}, \mathbf{m}_j)$ from the centroid $\mathbf{m}_j$ is the smallest. We denote these regions by V_j; formally, $\mathbf{x} \in V_j$ if $d(\mathbf{x}, \mathbf{m}_j) \leq d(\mathbf{x}, \mathbf{m}_k)$ for all k. The borders of the regions are assumed to have zero probability mass and therefore they are negligible.

It is assumed that almost all Voronoi regions become increasingly local when their number increases. (In singular cases such as in Figure 3**b** we identify data samples with their equivalence classes having zero mutual distance.) There are always some non-compact Voronoi regions at the borders of the data manifold, but it is assumed that the probability mass within them can be made arbitrarily small by increasing the number of regions. Assume further that the densities $p(c|\mathbf{x})$ are differentiable. Then the class distributions $p(c|\mathbf{x})$ can be made arbitrarily close to linear within each region V_j by increasing the number of Voronoi regions.

Let E_{V_j} denote the expectation over the Voronoi region V_j with respect to the probability density $p(\mathbf{x})$. At the optimum of the cost E_{KL}, we have $\boldsymbol{\psi}_j = E_{V_j}[p(c|\mathbf{x})]$, i.e. the parameters $\boldsymbol{\psi}_j$ are equal to the means of the conditional distribution within the Voronoi regions (see [1]; this holds even for the soft clusters).

Since $p(c|\mathbf{x})$ is linear within each Voronoi region, there exists a linear operator L_j for each V_j, for which $p(c|\mathbf{x}) = L_j\mathbf{x}$. The distributional prototypes then become

$$\boldsymbol{\psi}_j = E_{V_j}[p(c|\mathbf{x})] = E_{V_j}[L_j\mathbf{x}] = L_j E_{V_j}[\mathbf{x}] \equiv L_j\tilde{\mathbf{m}}_j = p(c|\tilde{\mathbf{m}}_j) \; ,$$

and the cost function becomes

$$E_{KL} = \sum_j \int_{V_j} D_{KL}(p(c|\mathbf{x}), p(c|\tilde{\mathbf{m}}_{j(\mathbf{x})}))\; p(\mathbf{x})\; d\mathbf{x} \; .$$

That is, given a locally linear $p(c|\mathbf{x})$, there exists a point $\tilde{\mathbf{m}}_j = E_{V_j}[\mathbf{x}]$ for each Voronoi region such that the Kullback-Leibler divergence appearing in the cost function can be measured with respect to the distribution $p(c|\tilde{\mathbf{m}}_{j(\mathbf{x})})$ instead of the average over the whole Voronoi region.

Because the Kullback-Leibler divergence is locally equal to a quadratic form of the Fisher information matrix (see e.g. [18]), we may expand the divergence around $\tilde{\mathbf{m}}_j$ to get

$$E_{KL} = \sum_j \int_{V_j} (\mathbf{x} - \tilde{\mathbf{m}}_{j(\mathbf{x})})^T \mathbf{J}(\tilde{\mathbf{m}}_{j(\mathbf{x})})(\mathbf{x} - \tilde{\mathbf{m}}_{j(\mathbf{x})})\; p(\mathbf{x})\; d\mathbf{x} \; , \tag{10}$$

where $\mathbf{J}(\tilde{\mathbf{m}}_{j(\mathbf{x})})$ is the Fisher information matrix evaluated at $\tilde{\mathbf{m}}_{j(\mathbf{x})}$.

Note that the Voronoi regions V_j are still defined by the parameters $\mathbf{m}_j$ and in the original, usually Euclidean metric.

In summary, discriminative clustering or maximization of mutual information, as presented in Section 3, asymptotically finds a partitioning from the family of local Euclidean Voronoi partitionings, for which the within-cluster distortion in the Fisher metric is minimized. In other words, discriminative clustering asymptotically performs vector quantization in the Fisher metric by Euclidean Voronoi regions: Euclidean metrics define the family of Voronoi partitionings $\{V_j\}_j$ over which the optimization is done, and the Fisher metric is used to measure distortion inside the regions.

4.2 Optimal Voronoi Regions are Approximately Spherical in the Fisher Metric

The partitions of discriminative clustering have so far been constrained to consist of Euclidean Voronoi regions. Let us next characterize the optimal shape of a single Voronoi region. Supposedly, a constellation of Voronoi regions tries to approximate this shape by setting the location parameters $\mathbf{m}_j$ suitably, i.e., within the limits the Voronoi condition allow.

In general, distortion within a Voronoi region V with the parameters $\mathbf{m}$ of the center can be written as

$$E_V \equiv \int_V D(\mathbf{x}, \mathbf{m}) p(\mathbf{x}) d\mathbf{x} = \int_{\mathbf{u}} \int_{r=0}^{R_0(\mathbf{u})} D(\mathbf{m} + r\mathbf{u}, \mathbf{m})\; r^\rho\; p_{\mathbf{x}}(\mathbf{m} + r\mathbf{u})\; dr\; d\mathbf{u} \; ,$$

where ρ is the dimensionality of the space, the outer integral goes over all unit vectors $\mathbf{u}$, $p_{\mathbf{x}}$ is the density before changing the variables, and $R_0(\mathbf{u})$ is the distance of the centroid $\mathbf{m}$ from the border of the Voronoi region V in the direction $\mathbf{u}$.

Imagine next that there is some freedom to set the borders $R_0(\mathbf{u})$. A necessary condition for the optimal form of a Voronoi region of a fixed size

$$p(V) \equiv \int_V p(\mathbf{x}) d\mathbf{x} = \int_{\mathbf{u}} \int_{r=0}^{R_0(\mathbf{u})} r^\rho\; p_{\mathbf{x}}(\mathbf{m} + r\mathbf{u})\; dr\; d\mathbf{u}$$

is obtained from the variation

$$\frac{\delta E_V}{\delta R(\mathbf{u})}(R_0(\mathbf{u})) + \lambda \frac{\delta p(V)}{\delta R(\mathbf{u})}(R_0(\mathbf{u})) = (\lambda + D(\mathbf{m} + R_0\mathbf{u}, \mathbf{m}))\, R_0^{\rho}\, p_{\mathbf{x}}(\mathbf{m} + R_0\mathbf{u})$$

by setting it to zero. Here λ is a Lagrange multiplier. The solutions are of the form

$$D(\mathbf{m} + R_0\mathbf{u}, \mathbf{m}) = \text{const.},$$

which implies that at the optimum, the distortion is constant at the borders of the Voronoi region.

For a Euclidean distortion the optimal shape is a sphere, and for a Fisher distortion the optimal Voronoi region would be the sphere of the Fisher metric. We may conjecture that the actual Voronoi regions approximate spheres in the Fisher metric, within the limits of the constraint that they must be Euclidean Voronoi regions.

4.3 Optimal Partitioning

The discriminative clustering algorithm was shown above, in Section 4.1, to (asymptotically) carry out vector quantization in the Fisher metric, under the additional constraint that the partitioning consists of Euclidean Voronoi regions. Let us next relax this constraint.

We define the goal of discriminative clustering as follows: Find a partitioning of the primary data space under two constraints: (1) The partitioning minimizes the within-cell distortion of the auxiliary information or, equivalently, maximizes mutual information, under the constraint that (2) the cells of the partitioning are "local" in the primary data space.

We consider a partitioning local if it consists of *Voronoi regions in any metric* M that generates the same topology as the original metric (i.e., does not tear the space). The goal of discriminative clustering in this most general form is then to *find a local partitioning* $\{V_j^M\}_j$ *that minimizes (9).*

Asymptotically, discriminative clustering is equivalent to minimizing (10). It is straightforward to show that *the optimal partitioning* $\{V_j^M\}_j$ *consists of Voronoi regions of the Fisher metric.* An $\mathbf{x}$ belongs to the Voronoi regions V_j^F of the Fisher metric, $\mathbf{x} \in V_j^F$, if

$$(\mathbf{x} - \tilde{\mathbf{m}}_j)^T \mathbf{J}(\tilde{\mathbf{m}}_j)(\mathbf{x} - \tilde{\mathbf{m}}_j) \leq (\mathbf{x} - \tilde{\mathbf{m}}_k)^T \mathbf{J}(\tilde{\mathbf{m}}_k)(\mathbf{x} - \tilde{\mathbf{m}}_k) \qquad (11)$$

for all k. Assume tentatively that the partitioning is not $\{V_j^F\}_j$, and that the cluster centroids $\tilde{\mathbf{m}}_k$ are at their optimal locations. Then any modification of the partitioning toward $\{V_j^F\}_j$ in the sense of changing points to belong to their "correct" Fisherian Voronoi regions V_j^F decreases the distortion (10).

4.4 Optimal Discriminative Clustering by Optimal Density Estimation

The new metric of Section 2 was based on the Fisher information matrix computed from an explicit estimate $\hat{p}(c|\mathbf{x})$ of the conditional density $p(c|\mathbf{x})$. The metric was used in connection with the Self-Organizing Maps.

In a similar fashion, the metric could be used for clustering. The cost function would then be (10), where $\mathbf{J}$ would be the Fisher information matrix computed from the estimate $\hat{p}(c|\mathbf{x})$, and V_j would be Voronoi regions spanned by the estimate and its Fisher metric. (A slight, asymptotically vanishing difference exists: we have so far computed the Fisher information matrix at $\mathbf{x}$ instead of at $\mathbf{m}_j(\mathbf{x})$; it is an open question which is better in practice.)

Asymptotically, for a large number of prototypes, the distortion of such a clustering is equal to

$$\hat{E}_{KL} = \sum_j \int_{V_j} D_{KL}(\hat{p}(c|\mathbf{x}), \hat{p}(c|\mathbf{m}_{j(\mathbf{x})}))\; p(\mathbf{x})\; d\mathbf{x} = \sum_j \int_{V_j} D_{KL}(\hat{p}(c|\mathbf{x}), \hat{\boldsymbol{\psi}}_j)\; p(\mathbf{x})\; d\mathbf{x}\ , \tag{12}$$

by the same reasoning as in Section 4.1. Here $\hat{\boldsymbol{\psi}}_j = E_{V_j}[\hat{p}(c|\mathbf{x})]$. This is just the cost function of discriminative clustering with the conditional densities replaced by their estimates $\hat{p}(c|\mathbf{x})$.

Although conceptually simple and relatively easy to implement, the approach of estimating the densities separately is problematic in that density estimators are optimized in the sense of maximum likelihood or some other *criteria of density estimation*, whereas here we are interested in the estimation of metrics, or ultimately the estimation of partitions. The criteria of density estimation are not necessarily optimal for discriminative clustering.

Equation (12), however, suggests a new alternative approach for discriminative clustering: Given a density estimator $\hat{p}(c|\mathbf{x})$, define the Voronoi regions by (11) in the Fisher metric induced by the estimator. Then, deviating from (12), measure the distortion by the true Kullback-Leibler distortion. Effectively, the cost function would be that of discriminative clustering (9), but the Voronoi regions would be defined by the density estimator.

Assuming our goal is discriminative clustering we could then state a criterion for the optimality of conditional density estimates: *The estimate $\hat{p}(c|\mathbf{x})$ is optimal if the Voronoi regions defined by it minimize the cost function (9) of discriminative clustering.*

4.5 Comparison to the Information Bottleneck Principle

The cost function of the Information Bottleneck, a distributional clustering framework for discrete co-occurrence data [17], is (in our notation) $I(X;V) + \beta I(C;V)$. This is equal to the cost function of discriminative clustering of Section 3, i.e. the mutual information $I(C;V)$, combined with the 'bottleneck' term $I(X;V)$ which restricts the representation efficiency of the partitioning V. (See Section 3.3 for a general description of the differences of these two approaches.)

If we apply the Information Bottleneck principle to a continuous-valued random variable X, the optimal form of the membership functions $y(\mathbf{x})$ becomes [17]

$$y_j(\mathbf{x}) = \frac{p(v_j)}{Z(\mathbf{x})} \exp\left(-\beta D_{\mathrm{KL}}(p(c|\mathbf{x}), \boldsymbol{\psi}_j)\right)\ ,$$

where $p(v_j) = \int y(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$ is the size of the partition V_j, $\boldsymbol{\psi}_j = E_{V_j}[p(c|\mathbf{x})]$ is the average of $p(c|\mathbf{x})$ over the partition, and $Z(\mathbf{x})$ is such that $\sum_j y_j(\mathbf{x}) = 1$ for all $\mathbf{x}$.

The partitions are not directly computable (for $p(v_j)$ and $\boldsymbol{\psi}_j$ depend on $y(\mathbf{x})$), and the iterative, nonparametric approach of [17] useful in the discrete case fails for a continuous X. It is, however, interesting to look at the limit of the partitions as the cost function approaches our cost function, i.e. when $\beta \to \infty$. The partitioning then approaches Voronoi regions in the Kullback-Leibler distortion: $\mathbf{x} \in V_j^{\mathrm{KL}}$ if $D_{\mathrm{KL}}(p(c|\mathbf{x}), \boldsymbol{\psi}_j) \leq D_{\mathrm{KL}}(p(c|\mathbf{x}), \boldsymbol{\psi}_k)$ for all $k \neq j$.

Although the V^{KL}-regions are potentially nonlocal, the result can still be used to characterize the optimal shapes of *local* Voronoi regions. One V^{KL}-region may become divided into several local regions in the primary data space. Asymptotically, under the assumptions of Section 4.1, the distribution $p(c|\mathbf{x})$ becomes linear in each region. As the KL-divergence asymptotically approaches the Fisher metric, the subregions approach Fisherian Voronoi regions.

4.6 Connection to Maximum Likelihood Estimation

Note that for finite data minimizing the cost function (9) is equivalent to maximizing

$$L = \sum_j \sum_{\mathbf{x} \in V_j} \log \psi_{j,c(\mathbf{x})} , \tag{13}$$

where $c(\mathbf{x})$ is the index of the class of sample $\mathbf{x}$. This is the log likelihood of a piecewise constant conditional density estimator. The estimator predicts the distribution of C to be $\boldsymbol{\psi}_j$ within the Voronoi region j. The likelihood is maximized with respect to both the $\boldsymbol{\psi}_j$ and the partitioning, under the defined constraints.

4.7 New Potential Approaches to Discriminative Clustering

According to the suggestion of Section 4.3, the most general (theoretical) form of discriminative clustering would be to find a metric in which Voronoi regions are local and maximize the mutual information, or equivalently the distortion (9). Asymptotically, the optimal partitioning was shown to consist of Voronoi regions in the Fisher metric, and we suggest using such regions for non-asymptotic cases as well.

The remaining problem is the computation of the partitions in practice, for the results of this section apply only to hard clusters, in the asymptotic case and with known densities $p(c|\mathbf{x})$.

One of the most promising routes would be to use the soft discriminative clustering algorithm of Section 3 in an estimated Fisher metric, for example by replacing the inverse covariance matrix of the Gaussian kernels by the Fisher information matrix of a suitable density estimator.

Another path was already sketched in Section 4.4: Given a family of density estimators $\hat{p}(c|\mathbf{x})$, define the Voronoi regions by the Fisher metric derived from the estimators, and minimize the average distortion with respect to both the partitioning and the parameters of the density estimators.

Given a suboptimal partitioning with good estimates of the prototypes $\boldsymbol{\psi}_j$, we could iteratively improve the partitioning by re-estimating the Fisher matrices directly from the relationships of the Voronoi regions with their neighbors.

The parameterization of the kernels of the soft discriminative clustering could also be directly relaxed, in principle. The many extra parameters would, however, require extra data, presumably much more than the alternative solutions.

5 Discussion

We have introduced the concept of learning metrics. It is used to measure important variation of the data, and to disregard meaningless variation. It is assumed that the data of primary interest is available paired with so called auxiliary data, and that variation in (the conditional distributions of) the auxiliary data indicates importance of the associated variation of the primary data.

Natural candidate data sets are those used in the common supervised tasks, i.e., regression and classification. Methods based on learning metrics have, however, a more general scope: that of exploring statistical dependencies between data sets and within the primary data.

Two approaches utilizing the principle were reviewed. First, a metric was derived from a density estimator of the conditional distributions of the auxiliary data.

The relative performance of different density estimators is still an open question. The second approach, discriminative clustering, incorporates the learning metric principle directly into the cost function of clustering. Both approaches have been successfully applied in data analysis.

Finally, we presented a relationship between the two approaches: Asymptotically, discriminative clustering performs vector quantization in Fisher metrics, but with the Voronoi regions defined in the original, usually Euclidean metric. This and other results point to new approaches toward practical algorithms.

Acknowledgments

This work was supported by the Academy of Finland, in part by the grants 50061 and 1164349. We would like to thank Jaakko Peltonen and Janne Nikkilä for their contributions to the earlier works reviewed in this paper.

References

[1] J. Sinkkonen and S. Kaski, "Clustering based on conditional distributions in an auxiliary space," *Neural Computation*, vol. 14, 2002, pp. 217–239.

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

[3] 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.

[4] D. J. Miller and H. S. Uyar, "A mixture of experts classifier with learning based on both labelled and unlabelled data," in M. Mozer, M. Jordan, and T. Petsche, eds., *Advances in Neural Information Processing Systems 9*, Cambridge, MA: MIT Press, 1997, pp. 571–577.

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

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

[7] J. W. Fisher III and J. Principe. "A methodology for information theoretic feature extraction," in *Proceedings of IJCNN'98, International Joint Conference on Neural Networks*, Piscataway, NJ: IEEE Service Center, vol. 3, 1998, pp. 1712–1716.

[8] K. Torkkola and W. Campbell, "Mutual information in learning feature transformations," in *Proceedings of the 17th International Conference on Machine Learning*, Stanford, CA: Morgan Kaufmann, 2000, pp. 1015–1022.

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

[10] S. Kaski, J. Kangas, and T. Kohonen, "Bibliography of self-organizing map (SOM) papers: 1981–1997," *Neural Computing Surveys*, vol. 1, 1998, pp. 1–176.

[11] S.-I. Amari. "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, 1998, pp. 251–276.

[12] 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, 1998, pp. 373–380.

[13] S. Kaski, "Convergence of a stochastic semisupervised clustering algorithm," *Technical Report A62*, Helsinki University of Technology, Espoo, Finland: Publications in Computer and Information Science, 2000.

[14] S. Becker, "Mutual information maximization: models of cortical self-organization," *Network: Computation in Neural Systems*, vol. 7, 1996, pp. 7–31.

[15] 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, 1998, pp. 466–472.

[16] 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*, 1993, pp. 183–190.

[17] 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.

[18] S. Kullback, *Information Theory and Statistics*, New York: Wiley, 1959.