To appear in Neural Computation

Clustering based on conditional distributions in an auxiliary space

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

Abstract

We study the problem of learning groups or categories that are local in the continuous *primary* space, but homogeneous by the distributions of an associated auxiliary random variable over a discrete *auxiliary* space. Assuming variation in the auxiliary space is meaningful, categories will emphasize similarly meaningful aspects of the primary space. From a data set consisting of pairs of primary and auxiliary items, the categories are learned by minimizing a Kullback–Leibler divergence-based distortion between (implicitly estimated) distributions of the auxiliary data, conditioned on the primary data. Still, the categories are defined in terms of the primary space. An on-line algorithm resembling the traditional Hebb-type competitive learning is introduced for learning the categories. Minimizing the distortion criterion turns out to be equivalent to maximizing the mutual information between the categories and the auxiliary data. In addition, connections to density estimation and to the distributional clustering paradigm are outlined. The method is demonstrated by clustering yeast gene expression data from DNA chips, with biological knowledge about the functional classes of the genes as the auxiliary data.

1 Introduction

Clustering algorithms and their goals vary, but it is common to aim at clusters that are relatively homogeneous while data in different clusters is dissimilar. The results depend totally on the criterion of similarity. The difficult problem of selecting a suitable criterion is commonly addressed by feature extraction and variable selection methods. Recently, metrics have additionally been derived by fitting a generative model to the data and using information-geometric methods for extracting a metric from the model (Hofmann, 2000; Jaakkola and Haussler, 1999; Tipping, 1999).

We study the related case in which additional useful information exists about the data items during the modeling process. The information is available as *auxiliary samples* c_k; they form pairs $(\mathbf{x}_k, c_k)$ with the primary samples $\mathbf{x}_k$. In this paper $\mathbf{x}_k \in \mathbb{R}^n$ and the c_k is multinomial. The extra information may, for example, be labels of functional classes of genes, like in our case study.

It is assumed that differences in the auxiliary data indicate what is important in the primary data space. More precisely, the difference between samples $\mathbf{x}_k$ and $\mathbf{x}_l$ is significant if the corresponding values c_k and c_l are different. The usefulness of this assumption depends, of course, on the choice of the auxiliary data.

Since the relationship between the auxiliary data and the primary data is stochastic, we get a better description of the difference between values $\mathbf{x}$ and $\mathbf{x}'$ by measuring differences between the *distributions* of c, given $\mathbf{x}$ and $\mathbf{x}'$. The conditional densities $p(c|\mathbf{x})$ and $p(c|\mathbf{x}')$ are not known, however. Only the set of sample pairs $\{(\mathbf{x}_k, c_k)\}_k$ is available. Because our aim is to minimize within-cluster dissimilarities, the clusters should be homogeneous in terms of the (estimated) distributions $p(c|\mathbf{x})$.

In order to retain the potentially useful structure of the primary space, we only use the auxiliary data to indicate importance, and define the clusters in terms of localized basis functions within the primary space. Such a clustering can then be used later for new samples from the primary data space even when the corresponding auxiliary samples are not available.

Very loosely speaking, our aim is to preserve the topology of the primary space, but measure distances by similarity in the auxiliary space.

Clearly almost any kind of paired data is applicable, but only good auxiliary data improves clustering. If the auxiliary data is closely related to the goal of the clustering task, as for example a performance index would be, then the auxiliary data guides the clustering to emphasize the important dimensions of the primary data space, and to disregard the rest. This automatic relevance detection is the main practical motivation for the present work.

We have earlier constructed a local metric into the primary space that measures distances in the primary space by approximating the corresponding differences between conditional distributions $p(c|\mathbf{x})$ in the auxiliary space (Kaski et al., 2001). The metric can be used for clustering, and then maximally homogeneous clusters in terms of the conditional distributions appear. In the present work we introduce an alternative method which, contrary to the approach generating an explicit metric, does not need an estimate of the conditional distributions $p(c|\mathbf{x})$ as an intermediate step.

We will additionally show that minimizing the within-cluster distortion is equivalent to maximizing the mutual information between the basis functions used for defining the clusters (interpreted as a multinomial random variable) and the auxiliary data. Maximization of mutual information has been previously used for constructing neural representations (Becker, 1996; Becker and Hinton, 1992). Other related works and paradigms include learning from (discrete) dyadic data (Hofmann, 1998), and distributional clustering (Pereira et al., 1993) with the information bottleneck (Tishby et al., 1999) principle.

2 Clustering based on the Kullback-Leibler divergence

We seek to cluster items $\mathbf{x}$ of the data space by utilizing the information within a set of pairs $(\mathbf{x}_k, c_k)$ of data. The set consists of paired samples of two random variables. The vector-valued random variable X takes values $\mathbf{x} \in \mathbb{X} \subset \mathbb{R}^n$, and the c_k (or sometimes just c) are values of the multinomial random variable C. We wish to keep the clusters local with respect to $\mathbf{x}$, but measure similarities between the samples $\mathbf{x}$ by the differences of the corresponding conditional distributions $p(c|\mathbf{x})$. These distributions are unknown and will be implicitly estimated from the data.

Vector quantization (VQ) or, equivalently, K-means clustering, is one approach to categorization. In VQ the goal is to minimize the average distortion E between the data and the prototypes or codebook vectors $\mathbf{m}_j$, defined by

$$E = \sum_j \int y_j(\mathbf{x}) D(\mathbf{x}, \mathbf{m}_j)\, p(\mathbf{x})\, d\mathbf{x} \; . \tag{1}$$

Here $D(\mathbf{x}, \mathbf{m}_j)$ denotes the measure of distortion between $\mathbf{x}$ and $\mathbf{m}_j$, and $y_j(\mathbf{x})$ is the cluster membership function that fulfills $0 \leq y_j(\mathbf{x}) \leq 1$ and $\sum_j y_j(\mathbf{x}) = 1$. In the classic "hard" vector quantization the membership function is binary valued: $y_j(\mathbf{x}) = 1$ if $D(\mathbf{x}, \mathbf{m}_j) \leq D(\mathbf{x}, \mathbf{m}_i)$ for all i, and $y_j(\mathbf{x}) = 0$ otherwise. Such functions define a partitioning of the space into discrete cells, Voronoi regions, and the goal of learning is to find the partitioning that minimizes the average distortion. If the membership functions $y_j(\mathbf{x})$ may attain any values between zero and one, the approach may be called soft vector quantization (Nowlan, 1990, has studied a maximum likelihood solution).

We measure distortions D as differences between the distributions $p(c|\mathbf{x})$ and model distributions. The measure of the differences will be the Kullback–Leibler divergence, defined for two discrete-valued distributions with event probabilities $\{p_i\}$ and $\{\psi_i\}$ as $D_{KL}(p, \psi) \equiv \sum_i p_i \log(p_i/\psi_i)$. In our case, the first distribution is the multinomial distribution in the auxiliary space that corresponds to the data $\mathbf{x}$, namely $p_i \equiv p(c_i|\mathbf{x})$. The second distribution is the prototype; let us denote the jth prototype by $\boldsymbol{\psi}_j$.

When the Kullback–Leibler distortion measure is plugged into equation (1), the error function of VQ, the average distortion becomes

$$E_{KL} = \sum_j \int y_j(\mathbf{x}) D_{KL}(p(c|\mathbf{x}), \boldsymbol{\psi}_j)\, p(\mathbf{x})\, d\mathbf{x} \; . \tag{2}$$

Instead of the distortions between the vectorial samples and vectorial prototypes as in (1), we now compute point-wise distortions between the distributions $p(c|\mathbf{x})$ and the prototypes $\boldsymbol{\psi}_j$. The prototypes are distributions in the auxiliary space.

The average distortion will be minimized by parameterizing the functions $y_j(\mathbf{x})$, and optimizing the distortion with respect to the cluster membership parameters and the prototypes. When the parameters of $y_j(\mathbf{x})$ are denoted by $\boldsymbol{\theta}_j$, the average distortion can be written

$$E_{KL} = -\sum_{i,j} \int [y_j(\mathbf{x}; \boldsymbol{\theta}_j) \log \psi_{ji}] p(c_i, \mathbf{x})\, d\mathbf{x} + \text{const.} \; , \tag{3}$$

where the constant is independent of the parameters.

The membership functions $y_j(\mathbf{x}; \boldsymbol{\theta}_j)$ can be interpreted as conditional densities $p(v_j|\mathbf{x}) \equiv y_j(\mathbf{x})$ of a multinomially distributed random variable V that indicates the cluster identity. The value of the random variable V will be denoted by $v \in \{v_j\}$ and the value of the random variable C corresponding to the multinomially distributed auxiliary distribution will be denoted by $c \in \{c_i\}$. Given $\mathbf{x}$, the choice of the cluster v does not depend on the c. In other words, C and V are conditionally independent: $p(c, v|\mathbf{x}) = p(c|\mathbf{x})p(v|\mathbf{x})$. It follows that $p(c, v) = \int p(c|\mathbf{x})p(v|\mathbf{x})p(\mathbf{x})d\mathbf{x}$.

It can be shown (see Appendix 1) that if the membership distributions are of the normalized exponential form

$$y_j(x; \boldsymbol{\theta}_j) = \frac{\exp f(\mathbf{x}; \boldsymbol{\theta}_j)}{\sum_l \exp f(\mathbf{x}; \boldsymbol{\theta}_l)} , \tag{4}$$

then the gradient of E_{KL} with respect to the parameters $\boldsymbol{\theta}_l$ becomes

$$\frac{\partial E_{KL}}{\partial \boldsymbol{\theta}_l} = \sum_{i,j} \int \frac{\partial f(\mathbf{x}; \boldsymbol{\theta}_l)}{\partial \boldsymbol{\theta}_l} \log \frac{\psi_{ji}}{\psi_{li}} \, p(c_i, v_j, v_l, \mathbf{x}) \, d\mathbf{x} \, . \tag{5}$$

The prototypes $\boldsymbol{\psi}_j$ are probabilities of multinomial distributions, and therefore they must fulfill $0 \leq \psi_{ji} \leq 1$ and $\sum_i \psi_{ji} = 1$. We will incorporate these conditions into our model by reparameterizing the prototypes as follows:

$$\log \psi_{ji} \equiv \gamma_{ji} - \log \sum_m e^{\gamma_{jm}} \, . \tag{6}$$

The gradient of the average distortion (3) with respect to the new parameters of the prototypes is

$$\frac{\partial E_{KL}}{\partial \gamma_{lm}} = \sum_i \int (\psi_{lm} - \delta_{mi}) \, p(c_i, v_l, \mathbf{x}) \, d\mathbf{x} \, , \tag{7}$$

where the Kronecker symbol $\delta_{mi} = 1$ when $m = i$ and $\delta_{mi} = 0$ otherwise.

The average distortion can be minimized with stochastic approximation, by sampling from $y_j(\mathbf{x})y_l(\mathbf{x})p(c_i, \mathbf{x}) = p(v_j, v_l, c_i, \mathbf{x})$. This leads to an on-line algorithm in which the following steps are repeated for $t = 0, 1, \ldots$ with $\alpha(t)$ gradually decreasing towards zero:

1. At the step t of stochastic approximation, draw a data sample $(\mathbf{x}(t), c(t))$. Assume that the value of $c(t)$ is c_i; this defines the value of i in the following steps.

2. Draw two basis functions, j and l, from the multinomial distribution with probabilities $\{y_k(\mathbf{x}(t))\}_k$.

3. Adapt the parameters $\boldsymbol{\theta}_l$ and γ_{lm}, $m = 1, \ldots, N_c$, by

$$\boldsymbol{\theta}_l(t+1) = \boldsymbol{\theta}_l(t) - \alpha(t) \left[\frac{\partial f(\mathbf{x}; \boldsymbol{\theta}_l)}{\partial \boldsymbol{\theta}_l} \right]_{\boldsymbol{\theta}_l = \boldsymbol{\theta}_l(t)} \log \frac{\psi_{ji}}{\psi_{li}} \tag{8}$$

$$\gamma_{lm}(t+1) = \gamma_{lm}(t) - \alpha(t)(\psi_{lm} - \delta_{mi}) \, , \tag{9}$$

   where N_c is the number of possible values of the random variable C. 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 (8) and (9) for the second adaptation. Note that $\boldsymbol{\theta}_l(t+1) = \boldsymbol{\theta}_l(t)$ if $j = l$.

In stochastic approximation the α should fulfill the conditions $\sum_t \alpha(t) = \infty$ and $\sum_t (\alpha(t))^2 < \infty$. In practice we have used piecewise-linear decreasing schedules.

Below we will consider two special cases. In the demonstrations in Figure 1 the basis functions are normalized Gaussians in the Euclidean space $\mathbb{X} = \mathbb{R}^n$. In the

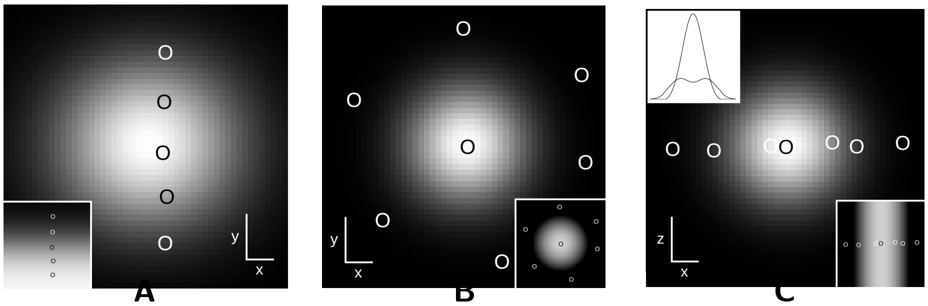

Figure 1: The location parameters $\boldsymbol{\theta}_j$ (small circles) of the Gaussian basis functions of two models, optimized for two-class ($n(c) = 2$), two-dimensional and three-dimensional data sets. The shades of gray at the background depict densities from which the data was sampled. **a**: 2D data, with roughly Gaussian $p(\mathbf{x})$. The inset shows the conditional density $p(c_0|\mathbf{x})$ that is monotonically decreasing as a function of the y-dimension. The model has learned to represent only the dimension on which $p(c|\mathbf{x})$ changes. **b** and **c**: Two projections of 3D data, with a symmetric Gaussian $p(\mathbf{x})$ (ideally, the form of this distribution should not affect the solution). The insets show the conditional density $p(c_0|\mathbf{x})$, which decreases monotonically as a function of a two-dimensional radius, and stays constant with respect to the orthogonal third dimension z. The one-dimensional cross section describing $p(c_0, \mathbf{x})$ and $p(c_1, \mathbf{x})$ as a function of the two-dimensional radius is shown in the inset of **c**. The model has learned to represent only variation in the direction of the radius and along the dimensions x and y, and discards the dimension z as irrelevant.

second case in Section 3, gene expression data mapped onto a hypersphere, $\mathbb{X} = \mathbb{S}^n$ are clustered by using normalized von Mises–Fisher distributions (Mardia, 1975) as the basis functions.

For Gaussians parameterized by their locations $\boldsymbol{\theta}_l$ and having a diagonal covariance matrix in which the variance σ^2 is equal in each dimension, $f(\mathbf{x}; \boldsymbol{\theta}_l) = -\|\mathbf{x} - \boldsymbol{\theta}_l\|^2 / 2\sigma^2$, and

$$\frac{\partial f(\mathbf{x}; \boldsymbol{\theta}_l)}{\partial \boldsymbol{\theta}_l} = \frac{1}{\sigma^2}(\mathbf{x} - \boldsymbol{\theta}_l) \; . \tag{10}$$

The von Mises–Fisher (vMF) distribution is an analogue of the Gaussian distribution on a hypersphere (Mardia, 1975) in that it is the maximum entropy distribution when the first two moments are fixed. The density of an n-dimensional vMF distribution is

$$\mathrm{vMF}(\mathbf{x}; \boldsymbol{\theta}) = \frac{1}{Z_n(\kappa)} \exp \kappa \frac{\mathbf{x}^T \boldsymbol{\theta}}{\|\boldsymbol{\theta}\|} \; . \tag{11}$$

Here the parameter vector $\boldsymbol{\theta}$ represents the mean direction vector. The normalizing coefficient $Z_n(\kappa) \equiv (2\pi)^{\frac{1}{2}n} I_{\frac{1}{2}n-1}(\kappa) / \kappa^{\frac{1}{2}n-1}$ is not relevant here but it will be used in the mixture model of Section 3. The function $I_r(\kappa)$ is the modified Bessel function of the first kind and order r.

In the clustering algorithm described above we will use vMF basis functions

$$f(\mathbf{x}; \boldsymbol{\theta}_l) = \kappa \mathbf{x}^T \boldsymbol{\theta}_l / \|\boldsymbol{\theta}_l\| \tag{12}$$

with a constant dispersion κ. Then,

$$\frac{\partial f(\mathbf{x}; \boldsymbol{\theta}_l)}{\partial \boldsymbol{\theta}_l} = \kappa(\mathbf{x} - \mathbf{x}^T \boldsymbol{\theta}_l \boldsymbol{\theta}_l / \|\boldsymbol{\theta}_l\|^2) / \|\boldsymbol{\theta}_l\| \; .$$

The norm $\|\boldsymbol{\theta}_l\|$ does not affect f, and we may normalize $\boldsymbol{\theta}_l$, whereby the gradient becomes

$$\frac{\partial f(\mathbf{x};\boldsymbol{\theta}_l)}{\partial \boldsymbol{\theta}_l} = \kappa(\mathbf{x} - \mathbf{x}^T\boldsymbol{\theta}_l\boldsymbol{\theta}_l)\ . \tag{13}$$

It can be shown (Kaski, 2000) that the stochastic approximation algorithm defined by eqns (8), (9), and (13) converges with probability one when the $\boldsymbol{\theta}_j$ are normalized after each step.

2.1 Connections to mutual information and density estimation

It can be easily shown (using the information inequality) that at the minimum of the distortion E_{KL} the prototype $\boldsymbol{\psi}_l$ takes the form

$$\psi_{li} = p(c_i|v_l)\ . \tag{14}$$

Hence, the distortion (3) can be expressed as

$$E_{KL} = -\sum_{i,j} p(c_i, v_j) \log \frac{p(c_i, v_j)}{p(c_i)p(v_j)} + \text{const.} = -I(C;V) + \text{const.}\ , \tag{15}$$

$I(C;V)$ being the mutual information between the random variables C and V. Thus, minimizing the average distortion E_{KL} is equivalent to maximizing the mutual information between the auxiliary variable C and the cluster memberships V.

Minimization of the distortion has a connection to density estimation as well; the details are described in Appendix 2. It can be shown that minimization of E_{KL} minimizes an upper limit of the mean Kullback-Leibler divergence between the real distribution $p(c|\mathbf{x})$ and a certain estimate $\hat{p}(c|\mathbf{x})$. The estimate is

$$\hat{p}(c_i|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \sum_j y_j(\mathbf{x};\boldsymbol{\theta}_j) \log \psi_{ji} \tag{16}$$

where $Z(\mathbf{x})$ is a normalizing coefficient selected such that $\sum_i \hat{p}(c_i|\mathbf{x}) = 1$.

For Gaussian basis functions, the upper limit becomes tight when σ approaches zero and the cluster membership functions become binary. Then the cluster membership functions approach indicator functions of Voronoi regions. Note that the solution cannot be computed in practice for $\sigma = 0$; smooth vector quantization with $\sigma > 0$ results in a compromise in which the solution is tractable but only an upper limit of the error will be minimized.

Furthermore, the mean divergence can be expressed in terms of the divergence of joint distributions as follows:

$$E_X\{D_{KL}(p(c|\mathbf{x}), \hat{p}(c|\mathbf{x}))\} = D_{KL}(p(c,\mathbf{x}), \hat{p}(c|\mathbf{x})p(\mathbf{x}))\ . \tag{17}$$

Here $E_X\{\cdot\}$ denotes the expectation over values of X and the latter D_{KL} is the divergence of the joint distribution. The expression (17) is a cost function of an estimate of the conditional probability $p(c|\mathbf{x})$. Intuitively speaking, by minimizing (17) resources are not wasted in estimating the marginal $p(\mathbf{x})$ but all resources are concentrated on estimating $p(c|\mathbf{x})$.

It can further be shown (Appendix 2) that maximum likelihood estimation of the model $\hat{p}(c|\mathbf{x})$ using a finite data set is asymptotically equivalent to minimizing (17).

2.2 Related works

2.2.1 Competitive learning

It is well known that the early works on competitive learning or adaptive feature detectors (e.g., Didday, 1976; Grossberg, 1976; Nass and Cooper, 1975; Pérez, et al., 1975) have a close connection to vector quantization (Gersho, 1979; Gray, 1984; Makhoul et al., 1985) and K-means clustering (Forgy, 1965; MacQueen, 1967). The neurons in a competitive-learning network are parameterized by vectors describing the synaptic weights, denoted by $\mathbf{m}_j$ for neuron j. In the simplest models the activity of a neuron due to external inputs is a nonlinear function f of the inputs $\mathbf{x}$ multiplied by the synaptic weights, $f(\mathbf{x}^T\mathbf{m}_j)$. The activities of the neurons compete; the activity of each neuron reduces the activity of the others via negative feedback.

If the competition is of the winner-take-all type (Kaski and Kohonen, 1994) only the neuron with the largest $f(\mathbf{x}^T\mathbf{m}_j)$ remains active. Each neuron therefore functions as a feature detector that detects whether the input comes from a particular domain of the input space. During Hebbian-type learning the neurons gradually specialize in representing different types of domains. For recent more detailed accounts see (Kohonen, 1984; Kohonen, 1993; Kohonen and Hari, 1999).

Although in the above discussion the winner was defined in terms of inner products, it is possible to generalize the model to an arbitrary metric. If the usual Euclidean metric is used, the learning corresponds to minimization of a mean-squared vector quantization distortion or, equivalently, minimization of the distance to the closest cluster center in the K-means clustering. The domain of the input space that a neuron detects can hence be interpreted as a Voronoi region in vector quantization.

The relationship of competitive learning to our work is that the "cluster membership functions" $y_j(\mathbf{x};\boldsymbol{\theta}_j)$ in Section 2 may be interpreted as the outputs of a set of neurons, and in the limit of crisp membership functions (for Gaussians, $\sigma \to 0$) only one neuron, the one having the largest external input, is active and can be interpreted as the winner. After learning our algorithm therefore corresponds to a traditional competitive network. The learning procedure makes the difference by making the network detect features that are as homogeneous as possible with regard to the auxiliary data.

The learning algorithm has a potentially interesting relation to Hebbian or competitive learning as well. Assume that at most two of the neurons may become active at a time, with probabilities $y_j(\mathbf{x};\boldsymbol{\theta}_j)$. Then the learning algorithm (8) for vMF kernels reads

$$\boldsymbol{\theta}_l(t+1) = \boldsymbol{\theta}_l(t) + \alpha(t)(\mathbf{x} - \mathbf{x}^T\boldsymbol{\theta}_l\boldsymbol{\theta}_l)(\log\psi_{li} - \log\psi_{ji})$$

(neuron j is adapted at the same time, swapping j and l). If the activity of the neurons is binary-valued, i.e., the neurons j and l have activity value one and the others value zero, then the adaptation rule for any neuron k can be expressed by

$$\boldsymbol{\theta}_k(t+1) = \boldsymbol{\theta}_k(t) + \alpha(t)\eta_k(t)(\mathbf{x} - \mathbf{x}^T\boldsymbol{\theta}_k\boldsymbol{\theta}_k)(\log\psi_{ki} - \log\psi_{ji}) \ . \tag{18}$$

Here $\eta_k(t)$ denotes the activity of the neuron k . Here the term $\eta_k(t)\mathbf{x}$ is Hebbian whereas $\eta_k(t)\mathbf{x}^T\boldsymbol{\theta}_k\boldsymbol{\theta}_k$ is a kind of a forgetting term (cf. Kohonen, 1984; Oja, 1982). The difference from common competitive learning then lies within the last parentheses in (18): The parameter vector of the neuron of the active pair (j, l) that represents better the class i has a larger value of $\log\psi$ and is moved towards the current sample, whereas the other neuron is moved away from the sample. (Note also the similarity to the LVQ algorithms; e.g. Kohonen, 1995.)

Note that for normalized Gaussian membership functions $y_j(\mathbf{x};\boldsymbol{\theta}_j)$ and $\sigma \neq 0$ our model is a kind of a variant of Gaussian mixture models or soft vector quantization. At the limit of crisp feature detectors (for Gaussians, $\sigma \to 0$), the output of the network reduces to a 1-of-C-coded discrete value. Similarly, the outputs of the soft version can be interpreted as probability density functions of a multinomial random variable. Such an interpretation has already been made by Becker in some of her work, discussed in Section 2.2.2.

2.2.2 Multinomial variables

Becker et al. (Becker and Hinton, 1992; Becker, 1996) have introduced a learning goal for neural networks called *Imax*. Their networks consist of two separate modules having different inputs, and the learning algorithms aim at maximizing the mutual information between the outputs of the modules. For example, if the inputs are two-dimensional arrays of random dots with stereoscopic displacements simulating the views of two eyes, the networks are able to infer depth from the data.

The variant called discrete Imax (Becker, 1996) is closely related to the clustering algorithm of the present paper. In Imax, the outputs of the neurons in each module are interpreted as the probabilities of a multinomial random variable, and the goal of learning is to maximize the mutual information between the variables of the two modules.

Our model differs from Becker's in two ways. First, Becker uses (normalized) $\exp(\mathbf{x}^T \boldsymbol{\theta}_j)$ as basis functions, whereas our parameterization makes the basis functions invariant to the norms of $\boldsymbol{\theta}_j$ (cf. eqn 11). Without such invariance, the units with largest norms may dominate the representation, a phenomenon noted by Becker as well.

The other difference is that Becker optimizes the model using gradient descent based on the whole batch of input vectors, whereas we have a simple on-line algorithm (18) adapting on the basis of one data sample at a time.

The gradient of the discrete Imax with respect to the parameters $\boldsymbol{\theta}_l$ is, after simplification and in our notation,

$$\frac{\partial I}{\partial \boldsymbol{\theta}_j} = \sum_{i,j} \int \mathbf{x} \log \frac{p(c_i|v_j)}{p(c_i|v_l)} \, p(v_j, v_l, c_i, \mathbf{x}) \, d\mathbf{x} \; . \tag{19}$$

It would be possible to apply stochastic approximation here by sampling from $p(v_j, v_l, c_i, \mathbf{x})$, which leads to a different adaptation rule than ours.

Becker (1996) has also used Gaussian basis functions, but with some approximations and ending up having a different formula for the gradient.

2.2.3 Continuous variables

The mutual information between continuous-valued outputs of two neurons can be maximized as well (Becker and Hinton, 1992; Becker, 1996). Some assumptions about the continuously-valued signals and the noise has to be made, however. In (Becker and Hinton, 1992) the outputs were assumed to consist of Gaussian signals corrupted by independent, additive Gaussian noise.

In this paper the multinomial Imax has been reinterpreted as (soft) vector quantization in the Kullback-Leibler "metric" in which the distance is measured in an auxiliary space. In neural terms, the model builds a representation of the input space: each neuron is a detector specialized to represent a certain domain. In contrast, the continuous version tries to represent the input space by generating a parametric transformation to a continuous, lower-dimensional output space. If the parameterization and the assumptions about the noise are correct, the continuous representations are potentially more accurate. The advantage of the quantized representation is that no such assumptions need to be made; the model[1] is almost purely data-driven (semi-parametric), and of course very useful for clustering-type applications.

It is particularly difficult to maximize the mutual information if each module has several continuous-valued outputs. In some recent works (Fisher and Principe, 1998; Torkkola and Campbell, 2000) the Shannon entropy has been replaced by the quadratic Renyi entropy, yielding simpler formulae for the mutual information.

2.2.4 Information bottleneck and distributional clustering

In distributional clustering works (Pereira et al., 1993) with the *information bottleneck principle* (Tishby et al., 1999), mutual information between two discrete variables has been maximized.

Tishby et al. get their motivation from the rate distortion theory of Shannon and Kolmogorov (see Cover and Thomas, 1991, for a review). In the rate distortion theory, the aim is to find an optimal codebook for a set of discrete symbols when a 'cost' in the form of a distortion function describing the effects of a transmission line is given.

In our notation, the authors consider the problem of building an optimal representation V for a discrete random variable X. In the rate distortion theory, a real-valued distortion function $d(x,v)$ is assumed known, and $I(X;V)$ is minimized with respect to the representation (or conventionally, the codebook) $p(v|x)$ subject to the constraint $E_{X,V}\{d(x,v)\} < k$. At the minimum the conditional distributions defining the codebook are

$$p(v_l|x) = \frac{p(v_l)\exp\left[-\beta d(x,v_l)\right]}{\sum_j p(v_j)\exp\left[-\beta d(x,v_j)\right]} \ , \tag{20}$$

where β depends on k. The authors realized that if the average distortion $E_{X,V}\{d(x,v)\}$ is replaced by the mutual information $-I(C;V)$, then the rate distortion theory gives a solution that captures as much information of the "relevance variable" C as possible. Here the multinomial random variable C has the same role as our auxiliary data.

The functional to be minimized becomes $I(X;V) - \beta I(C;V)$, and its variational optimization with respect to the conditional densities $p(v|x)$ leads to the solution (20) with

$$d(x,v_j) = D_{KL}(p(c|x), p(c|v_j)) \ . \tag{21}$$

Together these two equations give a characterization of the optimal representation V once we accept the criterion $I(X;V) - \beta I(C;V)$ for the goodness of the representation. The characterization is self-referential through $p(c|v)$ and therefore does not in itself present an algorithm for finding the $p(v|\mathbf{x})$ and $p(c|v)$, but Tishby et al. introduced an algorithm for finding a solution in the case of a multinomial X.

Like the method presented in this paper, the bottleneck aims at revealing nonlinear relationships between two data sets by maximizing a mutual information-based cost function. The relation between the two approaches is that although we started from the clustering viewpoint, our error criterion E_{KL} turned out to be equivalent to the (negative) mutual information $I(C;V)$. The bottleneck has an additional term in its error function for keeping the complexity of the representation low, while the complexity of our clusters is restricted by their number and their parameterization.

---

[1]We have made some assumptions by parameterizing the $f(\mathbf{x};\boldsymbol{\theta})$ in (4). However, the model becomes semi-parametric as a scale parameter, similar to the σ for Gaussians and $1/\kappa$ for the vMF kernels, approaches zero.

The most fundamental difference between our clustering approach and the published bottleneck works, however, arises from the continuity of our random variable X. The theoretical form of the bottleneck principle, equation (21), is not limited to discrete or finite spaces. According to our knowledge, however, no continuous applications of the principle have so far been published. For a continuous X, the distortion $d(x, v)$ in (21) cannot be readily evaluated without some kinds of additional assumptions, such as restrictions to the form of the cluster memberships $p(v|\mathbf{x})$. Our solution is to parameterize $p(v|\mathbf{x})$, which allows us to optimize the partitioning of the data space $\mathbb{X}$ into (soft) clusters.[2]

3 Case Study: Clustering of gene expression data

We tested our approach by clustering a large, high-dimensional data set, expressions of the genes of the budding yeast *Saccharomyces cerevisiae* in various experimental conditions. Such measurements, obtained from so-called DNA chips, are used in functional genomics to infer similarity of function of different genes. There are currently two popular approaches for analyzing expression data: traditional clustering methods (see, eg., Eisen et al., 1998) and supervised classification methods (Support Vector Machines, Brown et al., 2000). In this case study we intend to show that our method has the good sides of both of the approaches.

For the majority of yeast genes there exists a functional classification that is based on biological knowledge. The goal of the supervised classifiers is to learn this classification, in order to predict functions for new genes. The classifiers may additionally be useful in that the errors they make on the genes having known classes may suggest that the original functional classification has errors.

The traditional unsupervised clustering methods group solely on the basis of the expression data and do not utilize the known functional classes. Hence they are applicable to sets of genes without known classification, and they may additionally generate new discoveries: There may be hidden similarities between the classes in the hierarchical functional classification, and there may even exist new subclasses that become revealed as more experimental data is collected. The clustering methods can therefore be used as hypothesis generating machines.

The disadvantage of the currently used clustering algorithms is that the results are determined by the metric used for measuring similarity of the expression data. The metric is always somewhat arbitrary unless it is based on a considerable amount of knowledge about the functioning of the genes.

Our goal is to use the known functional classification to implicitly define which aspects of the expression data are important in the data. The clusters are local in the expression data space, but the prototypes are placed to minimize the average distortion (2) in the space of the functional classes. The difference from supervised classification methods is that while classification methods cannot surpass the original classes the (supervised) clusters are not tied to the classification and may reveal substructures within and relations between the known functional classes.

In this case study we compare our method empirically with alternative methods, and demonstrate its convergence properties and the potential usefulness of the results. More detailed biological interpretation of the results will be presented in subsequent papers.

We compared our model with two standard state-of-the-art mixture density models. The first is a totally unsupervised mixture of von Mises–Fisher distri-

[2]Alternatively, instead of solving a continuous problem, $\mathbb{X}$ could be (suboptimally) partitioned into predefined clusters, after which the standard distributional clustering algorithms are applicable.

butions. The model is analogous to the usual mixture of Gaussians; the Gaussian mixture components are simply replaced by the von Mises–Fisher components. The model is

$$p(\mathbf{x}) = \sum_j p(\mathbf{x}|v_j)p_j \tag{22}$$

where $p(\mathbf{x}|v_j) = \mathrm{vMF}(\mathbf{x}; \boldsymbol{\theta}_j)$, and vMF is defined in (11). The p_j are the mixing parameters.

In the second model (Hastie et al., 1995) the joint distribution between the functional classes c and the expression data $\mathbf{x}$ is modeled by a set of additive components denoted by u_j:

$$p(c_i, \mathbf{x}) = \sum_j p(c_i|u_j)p(\mathbf{x}|u_j)p_j \ , \tag{23}$$

where $p(c_i|u_j)$ and p_j are parameters to be estimated, and $p(\mathbf{x}|u_j) = \mathrm{vMF}(\mathbf{x}; \boldsymbol{\theta}_j)$. Both models are fitted to the data by maximizing their log-likelihood with the EM algorithm (Dempster et al., 1977).

3.1 The data

Temporal expression patterns of 2,476 genes of the yeast were measured with DNA chips in nine experimental settings (for more details see Eisen et al., 1998; the data is available at `http://rana.Stanford.edu/clustering/`). Each sample measures the expression level of a gene compared to the expression in a reference state. There were altogether 79 time points for each gene, represented below by the feature vector $\mathbf{x}$.

The data was preprocessed in the same way as in (Brown et al., 2000), i.e., by taking logarithms of the individual values and by normalizing the length of $\mathbf{x}$ to unity. The data were then divided into a training set containing two thirds of the samples and a test set containing the remaining third. All the reported results except Table 1 are computed for the test set.

The functional classification was obtained from the Munich Information Center for Protein Sequences Yeast Genome Database (MYGD[3]). The classification system is hierarchic, and we chose to use the 16 highest-level classes to supervise the clustering. Sample classes include metabolism, transcription, and protein synthesis. Some genes belonged to several classes. Seven genes were removed because of a missing classification at the highest level of the hierarchy.

3.2 The experiments

We first compared the performance of the three models, the mixture of Gaussians, MDA2, and our own, after the algorithms had converged. All models had 8 clusters and they were run until there was no doubt on convergence: the mixture of Gaussians and MDA2 for 150 epochs through the whole data and our model for 4,500,000 stochastic iterations ($\alpha(t)$ decreased first with a piecewise-linear approximation to an exponential curve, and then linearly to zero in the end). All models were run three times with different randomized initializations, and the best of the three results was chosen.

We measured the quality of the resulting clusterings by the average distortion error or, equivalently, the empirical mutual information. When estimating the empirical mutual information the table of the joint distributions $p(c_i, v_j)$ is first estimated. In our model the ith row of the table is updated by $p(v_j|\mathbf{x}) = y_j(\mathbf{x}; \boldsymbol{\theta}_j)$ (eqn

[3]`http://www.mips.biochem.mpg.de/proj/yeast`

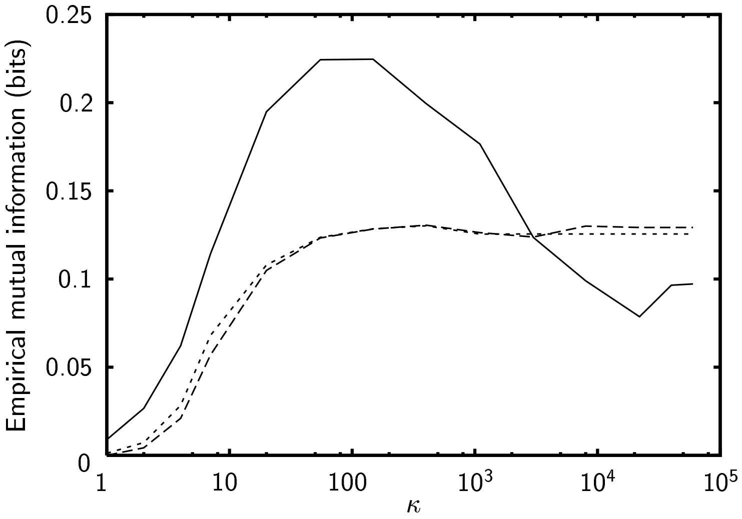

Figure 2: Empirical mutual information between the generated gene expression categories and the functional classes of the genes, as a function of parameter κ which governs the width of the basis functions. Solid line: our model, dashed line: mixture of vMFs, dotted line: MDA2.

4 with f defined by 12) for each sample $(c_i, \mathbf{x})$. In the Gaussian mixture model the update is $p(v_j|\mathbf{x}) = p(\mathbf{x}|v_j)p_j/p(\mathbf{x})$ and in MDA2 it is $p(u_j|\mathbf{x}) = p(\mathbf{x}|u_j)p_j/p(\mathbf{x})$. After the table $p(c_i, v_j)$ is computed, the performance criterion is obtained from eqn. (15) without the constant.[4]

The results are shown in Figure 2 for different values of the parameter κ that governs the width of the vMF kernels (11). Our model clearly outperforms the other models for a wide range of widths of the kernels and produces the best overall performance, i.e. the clusters of our model convey more information about the functional classification of the genes than the alternative models.

There is a somewhat surprising sideline in the results: The Gaussian mixture model is about as good as the MDA2, although it does not utilize the class information at all. The reason is probably in some special property of the data since for other data sets MDA2 has outperformed the plain mixture model.

Next we demonstrate the number of iterations required for convergence. The empirical mutual information is plotted in Figure 3 as a function of the number of iterations. In our model the schedule for decreasing the coefficient $\alpha(t)$ was the same in each run, stretched to cover the number of iterations and decaying to zero in the end. The number of complete data epochs for the MDA2 was made comparable to the number of stochastic iterations by multiplying it with the number of data and the number of kernels, divided by two (in our algorithm, two kernels are updated at each iteration step).

The performance of MDA2 attains its maximum quickly, but our model surpasses MDA2 well before 500,000 iterations (Fig. 3).

[4]Note that the empirical mutual information is an upward-biased estimate of the real mutual information, and the bias grows with decreasing data. As the size of our sample is rather large and constant across the compared distributions, and the number of values of the discrete variables is small, the bias does not markedly affect the results.

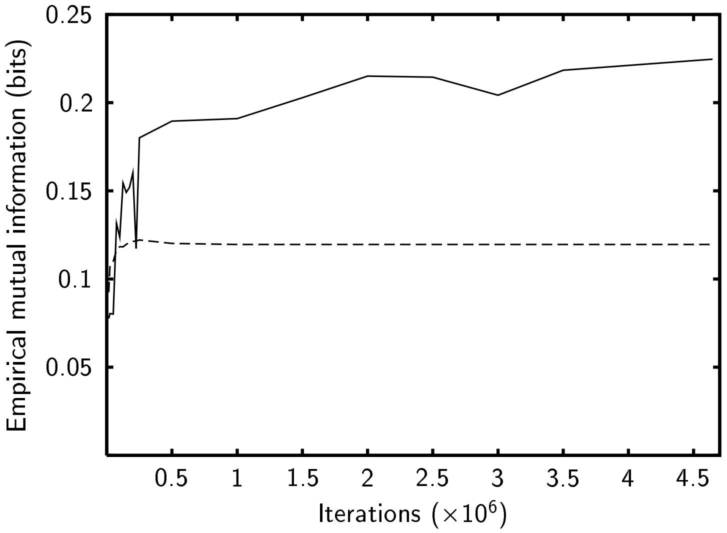

Figure 3: Empirical mutual information as a function of the number of iterations. Solid line: our model, dashed line: MDA2. $\kappa = 148$.

	Cluster number							
Class	1	2	3	4	5	6	7	8
a	0	1	6	0	1	0	0	0
b	0	1	16	0	0	0	0	0
c	1	5	39	1	1	4	14	3
d	0	3	8	1	0	0	0	2
e	122	1	0	2	0	2	44	2
f	3	3	1	20	0	46	2	12
g	0	1	0	21	0	0	0	7

Table 1: Distribution of genes (learning and test set combined) of sample functional subclasses into the 8 clusters obtained with our method. These subclasses were not used in supervising the clustering. a: the pentose-phosphate pathway, b: the tricarboxylic-acid pathway, c: respiration, d: fermentation, e: ribosomal proteins, f: cytoplasmic degradation, g: organization of chromosome structure.

Finally, we demonstrate the quality of the clustering by showing the distribution of the genes into the clusters from a few functional *sub*classes, known to be regulated in the experimental conditions like those of our data (Eisen et al., 1998). Note that these subclasses were not included in the values of the auxiliary variable C. Instead, they were picked from the second and third level of the functional gene classification. To characterize all the genes, the learning and the test sets were now combined. In Table 1 each gene is assigned to the cluster having the largest value of the membership function for that gene. The table reveals that many subclasses are concentrated in one of the clusters found by our algorithm. The first four (a-d) subclasses belong to the same first-level class and are placed in the same cluster, number 2.

For comparison, the distribution of the same subset of genes into clusters formed by the mixture of vMFs and MDA2 is shown in Table 2. The concentratedness of the classes in different clusters can be summarized by the empirical mutual information within the table; the mutual information is 1.2 bits for our approach, and 0.92 for the other two.

Class	Cluster number 1	2	3	4	5	6	7	8
a	0	3	1	0	0	2	1	1
b	1	1	0	0	0	14	0	1
c	3	16	2	5	23	14	4	1
d	0	9	0	2	0	0	2	1
e	0	6	1	4	32	1	125	4
f	42	12	6	8	0	4	4	11
g	3	1	10	5	0	0	2	8

Table 2: Distribution of genes (learning and test set combined) of sample functional subclasses into the 8 clusters obtained by the mixture of vMFs model and MDA2. Both methods yield the same table for these subclasses. For explanation of the classes see Table 1.

In Table 1, produced by our method, two of the subclasses (c and e) have been clearly divided into two clusters, suggesting a possible biologically interesting division. Its relevance will be determined later by futher biological inspection; in this paper our goal is to demonstrate that the semisupervised clustering approach can be used to explore the data set and to provide potential further hypotheses about its structure.

4 Conclusions

We have described a soft clustering method for continuous data which minimizes the within-cluster distortion between distributions of associated, discrete auxiliary data. The approach was inspired by our earlier work in which an explicit density estimator was used to derive an information-geometric metric for similar kinds of clustering tasks (Kaski et al., 2001). The present method is conceptually simpler and does not require explicit density estimation which is known to be difficult in high-dimensional spaces.

The task is analogous to that of distributional clustering (Pereira et al., 1993) of multinomial data with the information bottleneck method (Tishby et al., 1999), or learning from dyadic data (Hofmann, 1998). The main difference from our method is that these works operate on a discrete and finite data space while our data is continuous. Our setup and cost function have connections to the information bottleneck method but the approaches are not equivalent.

We showed that minimizing our Kullback-Leibler divergence-based distortion criterion is equivalent to maximizing the mutual information between (neural) representations of the inputs and a discrete variable studied by Becker (Becker and Hinton, 1992; Becker, 1996). The distortion was additionally shown to be bounded by a conditional likelihood, which it approaches in the limit where the clusters sharpen towards Voronoi regions.

We derived a simple on-line algorithm for optimizing the distortion measure. The convergence of the algorithm is proven for vMF basis functions in (Kaski, 2000). The algorithm was shown to have a close connection to competitive learning.

We applied the clustering method to a yeast gene expression data set that was augmented with an independent, functional classification for the genes. The algorithm performs better than other algorithms available for continuous data, the mixture of Gaussians and MDA2, a model for the joint density of the expression data and the classes. Our method turned out to be relatively insensitive to the (*a priori* set) width parameter of the Gaussian parameterization, outperforming the competing methods for a wide range of parameter values.

It was shown that the obtained clusters mediate information about the function of the genes and, although the results have not yet been biologically analyzed, potentially suggest novel cluster structures for the yeast genes.

Topics of future work include investigation of more flexible parameterizations for the clusters, relationship of the method to the metric defined in our earlier work, and variations of the algorithm towards visualizable clusterings and continuous auxiliary distributions.

Appendix 1: Gradient of the error criterion with respect to the parameters of the basis functions

If we denote

$$K_{il}(\mathbf{x}) \equiv \sum_j \frac{\partial y_j(\mathbf{x};\boldsymbol{\theta}_j)}{\partial \boldsymbol{\theta}_l} \log \psi_{ji} \ , \tag{24}$$

then

$$\frac{\partial E_{KL}}{\partial \boldsymbol{\theta}_l} = -\sum_i \int K_{il}(\mathbf{x})\ p(c_i,\mathbf{x})\ d\mathbf{x} \ . \tag{25}$$

Since the basis functions $y_j(x;\boldsymbol{\theta}_j)$ are of the normalized exponential form (4) the gradient of y_j is[5]

$$\frac{\partial y_j(\mathbf{x};\boldsymbol{\theta}_j)}{\partial \boldsymbol{\theta}_l} = \delta_{lj} y_j(\mathbf{x};\boldsymbol{\theta}_j) \frac{\partial f_j(\mathbf{x};\boldsymbol{\theta}_j)}{\partial \boldsymbol{\theta}_l} - y_j(\mathbf{x};\boldsymbol{\theta}_j) y_l(\mathbf{x};\boldsymbol{\theta}_l) \frac{\partial f_l(\mathbf{x};\boldsymbol{\theta}_l)}{\partial \boldsymbol{\theta}_l} \ . \tag{26}$$

Substituting the result into (24) gives

$$\begin{aligned} K_{il}(\mathbf{x}) &= y_l(\mathbf{x};\boldsymbol{\theta}_l) \left[\log \psi_{li} - \sum_j y_j(\mathbf{x};\boldsymbol{\theta}_j) \log \psi_{ji} \right] \frac{\partial f_l(\mathbf{x};\boldsymbol{\theta}_l)}{\partial \boldsymbol{\theta}_l} \\ &= y_l(\mathbf{x};\boldsymbol{\theta}_l) \left[\sum_j y_j(\mathbf{x};\boldsymbol{\theta}_j) \log \frac{\psi_{li}}{\psi_{ji}} \right] \frac{\partial f_l(\mathbf{x};\boldsymbol{\theta}_l)}{\partial \boldsymbol{\theta}_l} \ . \end{aligned} \tag{27}$$

Since the V and C are conditionally independent with respect to X, we may write

$$y_j(\mathbf{x};\boldsymbol{\theta}_j) y_l(\mathbf{x};\boldsymbol{\theta}_l) p(c_i,\mathbf{x}) = p(c_i, v_j, v_l, \mathbf{x}) \ . \tag{28}$$

Substituting this and eqn (27) into (25) we arrive at (5).

Appendix 2: Connection to conditional density estimation

Let us denote $y_j(\mathbf{x};\boldsymbol{\theta}_j) = y_j(\mathbf{x})$ for brevity, and note that the conditional entropy of C given X, or $H(C|X) = -\int \sum_i p(c_i,\mathbf{x}) \log p(c_i|\mathbf{x})\ d\mathbf{x}$, is independent of the

[5]Note that the normalized versions of the densities of the so called exponential family are included—if the $y_j(\mathbf{x};\boldsymbol{\theta}_j)$ are interpreted as densities of the random variables $p(v_j|\mathbf{x})$.

parameters $\boldsymbol{\theta}_j$ and ψ_{ji}. The distortion of eqn 3 can then be expressed as

$$\begin{aligned} E_{KL} &= -\sum_{i,j} \int y_j(\mathbf{x}) \log \psi_{ji}\; p(c_i, \mathbf{x})\; d\mathbf{x} - H(C|X) + \text{const.} &(29)\\ &= \int \sum_i \left[\log p(c_i|\mathbf{x}) - \sum_j y_j(\mathbf{x}) \log \psi_{ji} \right] p(c_i, \mathbf{x})\; d\mathbf{x} + \text{const.} &(30)\\ &= \int \sum_i p(c_i|\mathbf{x}) \log \frac{p(c_i|\mathbf{x})}{\exp \sum_j y_j(\mathbf{x}) \log \psi_{ji}}\; p(\mathbf{x})\; d\mathbf{x} + \text{const.} &(31)\\ &= \int \sum_i p(c_i|\mathbf{x}) \log \frac{p(c_i|\mathbf{x})}{q_i(\mathbf{x})}\; p(\mathbf{x})\; d\mathbf{x} + \text{const.}\;, &(32) \end{aligned}$$

where

$$q_i(\mathbf{x}) \equiv \exp \sum_j y_j(\mathbf{x}) \log p(c_i|v_j)\;. \tag{33}$$

The $\{q_i(\mathbf{x})\}_i$ is not a proper density. However, $\sum_i q_i(\mathbf{x}) \leq 1$ based on Jensen's inequality. Hence, for all $\mathbf{x}$,

$$\sum_i p(c_i|\mathbf{x}) \log \frac{p(c_i|\mathbf{x})}{q_i(\mathbf{x})}\; p(\mathbf{x}) \geq \sum_i p(c_i|\mathbf{x}) \log \frac{p(c_i|\mathbf{x})}{\hat{p}(c_i|\mathbf{x})}\; p(\mathbf{x})\;, \tag{34}$$

where

$$\hat{p}(c_i|\mathbf{x}) \equiv \frac{q_i(\mathbf{x})}{\sum_i q_i(\mathbf{x})}\;. \tag{35}$$

Therefore, minimizing the clustering criterion E_{KL} minimizes an upper limit of

$$\int \sum_i p(c_i|\mathbf{x}) \log \frac{p(c_i|\mathbf{x})}{\hat{p}(c_i|\mathbf{x})}\; p(\mathbf{x})\; d\mathbf{x} = E_X\{D_{KL}(p(c|\mathbf{x}), \hat{p}(c|\mathbf{x}))\}\;, \tag{36}$$

the expected KL-divergence between the conditional density $p(c|\mathbf{x})$ and its estimate $\hat{p}(c|\mathbf{x})$. (Here $E_X\{\cdot\}$ denotes the expectation over $\mathbf{x}$.) One can also write

$$\begin{aligned} E_X\{D_{KL}(p(c|\mathbf{x}), \hat{p}(c|\mathbf{x}))\} &= \int \sum_i p(c_i, \mathbf{x}) \log \frac{p(c_i, \mathbf{x})}{\hat{p}(c_i|\mathbf{x}) p(\mathbf{x})} d\mathbf{x}\\ &= D_{KL}(p(c, \mathbf{x}), \hat{p}(c|\mathbf{x}) p(\mathbf{x}))\;. &(37) \end{aligned}$$

Maximizing the likelihood of the model $\hat{p}(c_k|\mathbf{x}_k)$ for data $\{c_k, \mathbf{x}_k\}_k$ sampled from $p(c, \mathbf{x})$ is asymptotically equivalent to minimizing the Kullback-Leibler divergence (37). This is because for the N i.i.d. samples $\{c_k, \mathbf{x}_k\}_k$ the scaled log-likelihood

$$\frac{1}{N} \sum_k \log \hat{p}(c_k|\mathbf{x}_k)$$

converges to

$$\begin{aligned} \int \sum_i p(c_i, \mathbf{x}) \log \hat{p}(c_i|\mathbf{x}) d\mathbf{x} &= -\int \sum_i p(c_i|\mathbf{x}) \log \frac{p(c_i|\mathbf{x})}{\hat{p}(c_i|\mathbf{x})} p(\mathbf{x}) d\mathbf{x} + \text{const.}\\ &= -E_X\{D_{KL}(p(c|\mathbf{x}), \hat{p}(c|\mathbf{x}))\} + \text{const.} &(38) \end{aligned}$$

with probability 1 when $N \to \infty$, and because maximizing (38) minimizes (37).

In the special case when the $y_j(\mathbf{x})$ are binary-valued, the $q_i(\mathbf{x}) = \hat{p}(c_i|\mathbf{x})$ is a proper density and the equality in (34) holds for almost every $\mathbf{x}$. Therefore, the minimization of (an approximation of) the average distortion E_{KL} on a finite data set is equivalent to maximizing the likelihood of $\hat{p}(c|\mathbf{x})$ on the same sample.

Acknowledgments

This work was supported by the Academy of Finland, in part by the grant 50061. We wish to thank Janne Nikkilä, Petri Törönen, and Jaakko Peltonen for their help with the simulations and processing of the data.

References

Becker, S. (1996). Mutual information maximization: models of cortical self-organization. *Network: Computation in Neural Systems, 7*, 7–31.

Becker, S. & Hinton, G. E. (1992). Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature, 355*, 161–163.

Brown, M. P., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C. W., Furey, T. S., Ares, Jr., M., and Haussler, D. (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences, USA*, 97:262–267.

Cover, T. M. & Thomas, J. A. (1991). *Elements of Information Theory.* New York: Wiley.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38.

Didday, R. L. (1976). A model of visuomotor mechanisms in the frog optic tectum. *Mathematical Biosciences, 30*, 169–180.

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences, USA*, 95:14863–14868.

Fisher III, J. W., & Principe, J. (1998). A methodology for information theoretic feature extraction. In *Proc. IJCNN'98, International Joint Conference on Neural Networks* (pp. 1712–1716). Piscataway, NJ: IEEE Service Center.

Forgy, E. W. (1965). Cluster analysis of multivariate data: Efficiency vs interpretability of classifications. *Biometrics, 21*, 768–769.

Gersho, A. (1979). Asymptotically optimal block quantization. *IEEE Transactions on Information Theory, 25*, 373–380.

Gray, R. M. (1984). Vector quantization. *IEEE ASSP Magazine*, April, 4–29.

Grossberg, S. (1976). On the development of feature detectors in the visual cortex with applications to learning and reaction-diffusion systems. *Biological Cybernetics, 21*, 145–159.

Hastie, T., Tibshirani, R., & Buja, A. (1995). Flexible discriminant and mixture models. In Kay, J. and Titterington, D., editors, *Neural Networks and Statistics.* Oxford University Press.

Hofmann, T., Puzicha, J., & Jordan, M. I. (1998). Learning from dyadic data. In Kearns, M. S., Solla, S. A., Cohn, D. A. (Eds.), *Advances in Neural Information Processing Systems, 11* (pp. 466–472). San Mateo, CA: Morgan Kauffman Publishers.

Hofmann, T. (2000). Learning the similarity of documents: An information-geometric approach to document retrieval and categorization. In Solla, S. A., Leen, T. K., & Müller, K.-R. (Eds.), *Advances in Neural Information Processing Systems, 12* (pp. 914–920). Cambridge, MA: MIT Press.

Jaakkola, T. S. & Haussler, D. (1999). Exploiting generative models in discriminative classifiers. In Kearns, M. S., Solla, S. A., and Cohn, D. A. (Eds.), *Advances in Neural Information Processing Systems, 11* (pp. 487–493). San Mateo, CA: Morgan Kaufmann.

Kaski, S. (2000). Convergence of a stochastic semisupervised clustering algorithm. Technical Report A62, Helsinki University of Technology, Publications in Computer and Information Science, Espoo, Finland.

Kaski, S. & Kohonen, T. (1994). Winner-take-all networks for physiological models of competitive learning. *Neural Networks, 7*, 973–984.

Kaski, S., Sinkkonen, J., and Peltonen, J. (2001). Bankruptcy analysis with self-organizing maps in learning metrics. *IEEE Transactions on Neural Networks*. Accepted for publication.

Kohonen, T. (1984). *Self-Organization and Associative Memory* (3rd ed. 1989). Berlin: Springer-Verlag.

Kohonen, T. (1993). Physiological interpretation of the self-organizing map algorithm. *Neural Networks, 6*, 895–905.

Kohonen, T. (1995). *Self-Organizing Maps*. Springer, Berlin. (Second, extended edition 1997).

Kohonen, T. & Hari, R. (1999). Where the abstract feature maps of the brain might come from. *TINS, 22*, 135–139.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In Le Cam, L. M., & Neyman, J. (Eds.), *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Vol. I: Statistics* (pp. 281–297). Berkeley, CA: University of California Press.

Makhoul, J., Roucos, S., & Gish, H. (1985). Vector quantization in speech coding. *Proceedings of the IEEE, 73*, 1551–1588.

Mardia, K. V. (1975). Statistics of directional data. *Journal of the Royal Statistical Society B 37*, 349–393.

Nass, M. M. & Cooper, L. N. (1975). A theory for the development of feature detecting cells in visual cortex. *Biological Cybernetics, 19*, 1–18.

Nowlan, S. J. (1990). Maximum likelihood competitive learning. In Touretzky, D. S. (Ed.), *Advances in Neural Information Processing Systems, 2* (pp. 574–582). San Mateo, CA: Morgan Kaufmann.

Oja, E. (1982). A simplified neuron model as a principal component analyzer. *J. Math. Biology*, 15:267–273.

Pereira, F., Tishby, N., & Lee, L. (1993). Distributional clustering of English words. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics* (pp. 183–190).

Pérez, R., Glass, L., & Shlaer, R. J. (1975). Development of specificity in cat visual cortex. *Journal of Mathematical Biology, 1*, 275–288.

Tipping, M. E. (1999). Deriving cluster analytic distance functions from Gaussian mixture models. In *Proc. ICANN99, Ninth International Conference on Artificial Neural Networks* (pp. 815–820). London: IEE.

Tishby, N., Pereira, F. C., & Bialek, W. (1999). The information bottleneck method. In *37th Annual Allerton Conference on Communication, Control, and Computing*, Urbana, Illinois.

Torkkola, K. & Campbell, W. (2000). Mutual information in learning feature transformations. In *Proc. ICML'2000, the 17th International Conference on Machine Learning* (pp. 1015–1022). San Mateo, CA: Morgan Kaufmann.