Samuel Kaski, Janne Sinkkonen, and Arto Klami. 2005. Discriminative clustering. Neurocomputing, volume 69, numbers 1-3, pages 18-41.

Available online at www.sciencedirect.com
SCIENCE @ DIRECT®

ELSEVIER

Neurocomputing 69 (2005) 18–41

NEUROCOMPUTING

www.elsevier.com/locate/neucom

# Discriminative clustering[☆]

Samuel Kaski[a,b,*], Janne Sinkkonen[a,b], Arto Klami[a,b]

[a]Neural Networks Research Centre, Helsinki University of Technology, P.O. Box 5400,
FI-02015 HUT, Finland
[b]Department of Computer Science, P.O. Box 68, FI-00014 University of Helsinki, Finland

Received 29 March 2004; received in revised form 18 November 2004; accepted 9 February 2005
Available online 22 August 2005

**Abstract**

A distributional clustering model for continuous data is reviewed and new methods for optimizing and regularizing it are introduced and compared. Based on samples of discrete-valued auxiliary data associated to samples of the continuous primary data, the continuous data space is partitioned into Voronoi regions that are maximally homogeneous in terms of the discrete data. Then only variation in the primary data associated to variation in the discrete data affects the clustering; the discrete data "supervises" the clustering. Because the whole continuous space is partitioned, new samples can be easily clustered by the continuous part of the data alone. In experiments, the approach is shown to produce more homogeneous clusters than alternative methods. Two regularization methods are demonstrated to further improve the results: an entropy-type penalty for unequal cluster sizes, and the inclusion of a model for the marginal density of the primary data. The latter is also interpretable as special kind of joint distribution modeling with tunable emphasis for discrimination and the marginal density.

*Keywords:* Discriminative clustering; Information metric; Learning metrics; Regularization; Vector quantization

[☆] The first two authors contributed equally to this work.

[*] Corresponding author. Neural Networks Research Centre, Helsinki University of Technology, P.O. Box 5400, FI-02015 HUT, Finland. Tel.: +358 9 451 8203; fax: +358 9 451 3277.

*E-mail address:* Samuel.Kaski@hut.fi (S. Kaski).

0925-2312/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2005.02.012

## 1. Introduction

Models exist for discovering components underlying co-occurrences of nominal variables [4,5,14], and for the joint distribution $p(c, \mathbf{x})$ of continuous $\mathbf{x} \in \mathbb{X} \subset \mathbb{R}^n$ and discrete data $c$ [12,13,18]. We consider the related task of clustering the continuous primary data by conditional modeling such that the clusters become "relevant for" or "informative of" the discrete auxiliary data, i.e., capable of predicting $p(c|\mathbf{x})$. The discriminative approach is expected (and indeed found) to result in clusters more informative about $c$ than those obtained by modeling the joint distribution. The continuity of $\mathbf{x}$ distinguishes the setting from that of (classic) distributional clustering [20,22,26].

The task, coined *discriminative clustering* (DC), is different from classification in that the number of clusters is not constrained to be equal to the number of classes, which for clustering purposes may be much too high or low. In DC, the derived cluster structure of the $\mathbb{X}$-space is the primary outcome, even to the degree that the distributional parameters predicting $p(c|\mathbf{x})$ within a cluster can be integrated out.

The main application area for DC is in data exploration or mining. Alternatively, when $c$ is interpreted as an existing probabilistic partitioning of $\mathbb{X}$, DC can be used to alter the coarseness of the partitioning.

A prototypical application would be grouping of existing customers of a company on the basis of continuous covariates ($\mathbf{x}$; including, for instance, coordinates of residence, age, etc.) into clusters that are informative of the buying behavior of the customers across several product categories ($c$). New real or potential customers can then be clustered even before they have made their first purchases. Other potential applications include finding prototypical gene expression patterns to refine existing functional classifications of genes [21], clustering of financial statements to discover different ways to descend into bankruptcy, and partitional clustering in general when a variable $c$ is used to automatically guide the feature selection.

In this paper, an earlier model for DC [21] is reviewed and extended. On-line implementation of the earlier model was simple and it had interesting connections to neural computation, but for practical data analysis it had a shortcoming: it was formulated for distributions of data instead of finite data sets, which implies that it cannot take the uncertainty caused by the finiteness of the sample rigorously into account. In this paper, we formulate DC in Bayesian terms, which has the additional benefit that the parameters the earlier method included for modeling the auxiliary data $c$ can be integrated out from the cost function.

The model cannot be optimized directly by gradient-based algorithms, but we show that complementing a conjugate gradient algorithm with a smoothing of partitions gives comparable results to the much more time-consuming simulated annealing (SA). To further improve the performance, the model is additionally regularized in two alternative ways: by penalizing from unequal cluster sizes, or alternatively by adding to the cost function a term modeling the primary data. The latter is equivalent to generative modeling of the full joint distribution $p(c, \mathbf{x})$ of the primary and auxiliary data, but also interpretable as a tunable compromise between modeling $p(\mathbf{x})$ and $p(c|\mathbf{x})$.

In experiments, all the proposed models outperform alternative mixture-based models in their task, and both of the regularization methods outperform pure DC. In most cases, the new Bayesian optimization method performs better than the older stochastic on-line algorithm, and requires less time for optimization.

## 2. DC model

We will start by reviewing the basic DC model [15,21], and by simultaneously clarifying its relationship with maximum likelihood estimation. Although different from the original derivation, the perspective here makes a Bayesian extension possible.

The goal of DC is to partition the primary data space into clusters that are (i) local in the primary space and (ii) homogeneous and predictive in terms of auxiliary data. (The connection between homogeneity and predictivity of the clusters is detailed below.) Locality is enforced by defining the clusters as Voronoi regions in the primary data space: $\mathbf{x}$ belongs to cluster $j$, $\mathbf{x} \in V_j$, if $\|\mathbf{x} - \mathbf{m}_j\| \leqslant \|\mathbf{x} - \mathbf{m}_k\|$ for all $k$. The Voronoi regions are uniquely determined by the parameters $\{\mathbf{m}_j\}$.

Homogeneity is enforced by assigning a *distributional prototype* denoted by $\boldsymbol{\psi}_j \equiv p(c|\mathbf{x}, \mathbf{x} \in V_j)$ to each Voronoi region $j$, and searching for partitionings capable of predicting auxiliary data with the prototypes. The resulting model is a piecewise-constant generative model for $c$ conditioned on $\mathbf{x}$, with the log likelihood

$$L = \sum_j \sum_{\mathbf{x} \in V_j} \log \psi_{j,c(\mathbf{x})}. \tag{1}$$

The probability of class $i$ within the $j$th Voronoi region $V_j$ is predicted to be $\psi_{ji} = p(c_i|\mathbf{x}, \mathbf{x} \in V_j)$, and $c(\mathbf{x})$ denotes the class of sample $\mathbf{x}$.

In summary, the assumed data-generating mechanism is simple: The primary data $\mathbf{x}$ are covariates that determine the cluster membership $j$ (the relationship is deterministic given the parameters of the clusters). The auxiliary data $c$ are then generated by a cluster-specific multinomial having parameters $\psi_{ji}$. This generative mechanism is assumed throughout the paper.

Asymptotically for large data sets,

$$L \propto -\sum_j \int_{V_j} D_{\mathrm{KL}}(p(c|\mathbf{x}), \boldsymbol{\psi}_j) p(\mathbf{x})\, \mathrm{d}\mathbf{x} + \text{const.}, \tag{2}$$

where $D_{\mathrm{KL}}$ is the Kullback–Leibler divergence between the observed distribution of auxiliary data and the prototype. This is the cost function of $K$-means clustering or vector quantization (VQ) with the distortion measured by $D_{\mathrm{KL}}$. In this sense, maximizing the likelihood of the model therefore maximizes the distributional homogeneity of the clusters.

It can be shown [21] that maximizing (2) is equivalent to maximizing the mutual information between the auxiliary variable and the partitioning, which is a connection to models that use the empirical mutual information as a clustering criterion [2]. Asymptotically DC performs VQ in Fisher metrics, with the restriction of Voronoi regions being those of the original, usually Euclidean metric [15].

### 2.1. Optimization

When DC was introduced (although not under its current name), an on-line stochastic algorithm for optimizing the cost function (2) was derived [21]. The gradient of the cost is non-zero only at the borders of the Voronoi regions, and to overcome this difficulty the regions are softened or smoothed. The resulting algorithm is briefly reviewed here.

The smoothing is performed by introducing membership functions $y_j(\mathbf{x}; \{\mathbf{m}\})$ to (2). The values of the membership functions vary between 0 and 1, and $\sum_j y_j(\mathbf{x}) = 1$. The smoothed cost function is

$$E'_{\mathrm{KL}} = \sum_j \int y_j(\mathbf{x}; \{\mathbf{m}\}) D_{\mathrm{KL}}(p(c|\mathbf{x}), \boldsymbol{\psi}_j) p(\mathbf{x})\, \mathrm{d}\mathbf{x}. \tag{3}$$

One possible form for the memberships is the normalized Gaussian,

$$y_j(\mathbf{x}) = Z(\mathbf{x})^{-1} \exp^{(-\|\mathbf{x}-\mathbf{m}_j\|^2/2\sigma^2)}, \tag{4}$$

where $Z$ normalizes the sum to unity for each $\mathbf{x}$. The value of the parameter $\sigma$ controlling the smoothness can be chosen with a validation set.

The smoothed cost is then minimized with the following algorithm. Denote the i.i.d. data pair at the 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 values of the membership functions $\{y_k(\mathbf{x}(t))\}_k$. To keep the distributional parameters summed up to unity, reparameterize them by the "soft-max", $\log \psi_{ji} = \gamma_{ji} - \log \sum_m \exp(\gamma_{jm})$. Adapt the prototypes by

$$\mathbf{m}_j(t+1) = \mathbf{m}_j(t) - \alpha(t)\left[\mathbf{x}(t) - \mathbf{m}_j(t)\right] \log \frac{\psi_{li}(t)}{\psi_{ji}(t)}, \tag{5}$$

$$\gamma_{jm}(t+1) = \gamma_{jm}(t) - \alpha(t)\left[\psi_{jm}(t) - \delta_{mi}\right], \tag{6}$$

where $\delta_{mi}$ is the Kronecker delta. Due to the symmetry between $j$ and $l$, it is possible (and evidently beneficial) to adapt the parameters twice for each $t$ by swapping $j$ and $l$ in (5) and (6) for the second adaptation. Note that no updating of the $\mathbf{m}$ takes place if $j = l$; then $\mathbf{m}_j(t+1) = \mathbf{m}_j(t)$. During learning the parameter $\alpha(t)$ decreases gradually toward zero according to a schedule that, to guarantee convergence, must fulfill the conditions of the stochastic approximation theory.

For finite data, the algorithm maximizes the conditional likelihood (1)—by heuristically smoothing the clusters to get a computable gradient.

## 3. MAP estimation of clusters of DC

The earlier DC algorithm [21] was motivated by maximization of the empirical mutual information. Asymptotically, for large amounts of data, empirical mutual information is a justified measure of homogeneity or dependency. Maximization of mutual information was re-interpreted in the previous section as maximum likelihood estimation, which uses smoothed cluster memberships as an optimization trick. The new interpretation opens up the possibility of Bayesian extensions. For small data sets, an alternative, potentially better-behaving form of DC is obtained by marginalizing the likelihood (1), as introduced next. It turns out that the distributional prototypes $\{\psi_j\}$ can be analytically integrated out from the posterior distribution of $\{\mathbf{m}_j\}$ and $\{\psi_j\}$ given data, to leave only the parameters $\{\mathbf{m}_j\}$ of the Voronoi regions. This is convenient and should improve the results by taking into account the uncertainty associated with the $\{\psi_j\}$. Our goal is to partition the primary space instead of predicting the classes by the $\{\psi_j\}$, and hence the predictions are not needed.

The auxiliary data are denoted by $D^{(c)}$, and the primary data by $D^{(x)}$. We then wish to find the set of clusters $\{\mathbf{m}_j\}$ which maximizes the marginalized posterior (the integration is over all the $\psi_j$)

$$MAP_{\mathrm{DC}} = p(\{\mathbf{m}_j\}|D^{(c)}, D^{(x)}) = \int_{\{\psi_j\}} p(\{\mathbf{m}_j\}, \{\psi_j\}|D^{(c)}, D^{(x)}) d\{\psi_j\}. \tag{7}$$

In this paper, the improper prior $p(\{\mathbf{m}_j\}, \{\psi_j\}) \propto p(\{\psi_j\}) = \prod_j p(\psi_j)$ is used, where the factors $p(\psi_j) \propto \prod_i \psi_{ji}^{n_i^0-1}$ are Dirichlet priors with the parameters $n_i^0$ common to all $j$. Dirichlet distribution is the conjugate of the multinomial distribution, and is therefore convenient. By Bayes rule and marginalization, the postrior[1] is then given by

$$\begin{aligned} MAP_{\mathrm{DC}} &\propto \int_{\{\psi_j\}} p(D^{(c)}|\{\mathbf{m}_j\}, \{\psi_j\}, D^{(x)}) p(\{\psi_j\}) d\{\psi_j\} \\ &= \prod_j \int_{\psi_j} p(D_j^{(c)}|\psi_j) p(\psi_j)\, \mathrm{d}\psi_j \\ &\propto \prod_j \int_{\psi_j} \prod_i \psi_{ji}^{n_i^0+n_{ji}-1}\, \mathrm{d}\psi_j = \prod_j \frac{\prod_i \Gamma(n_i^0 + n_{ji})}{\Gamma(N^0 + N_j)}. \end{aligned} \tag{8}$$

Here $n_{ji}$ is the number of samples of class $i$ in cluster $j$, $D_j^{(c)}$ is the auxiliary data in cluster $j$, $N_j = \sum_i n_{ji}$, and $N^0 = \sum_i n_i^0$.

The final objective function (8) is thus the posterior probability of the cluster centroids given the data. Assuming the DC task, the objective is meaningful for comparing various alternative methods, and it can be used as an optimization criterion by searching the maximum a posterior (MAP) estimate. In practice, we use

[1] Also interpretable as marginalized maximum likelihood here, since the prior for $\{\mathbf{m}_j\}$ is improper.

the logarithm of the posterior

$$\log p(\{\mathbf{m}_j\}|D^{(c)}, D^{(x)}) = \sum_{ij} \log \Gamma(n_i^0 + n_{ji}) - \sum_j \log \Gamma(N^0 + N_j) + const. \tag{9}$$

for computational simplicity. Notice that the connection to mutual information is retained in the marginalization process. Maximizing the posterior asymptotically maximizes the mutual information between the clusters and auxiliary data (Appendix A).

### 3.1. Optimization

Plain marginalized DC is unsuitable for gradient-based optimization for the same reason as the infinite-data cost (2): the gradient would be affected only by samples at the (typically zero-probability) border of the clusters. This problem was earlier avoided by a smoothing approach, and similar smoothing is possible also in the marginalized DC. The smoothed "number" of samples is $n_{ji} = \sum_{c(\mathbf{x})=i} y_j(\mathbf{x})$, where $c(\mathbf{x})$ is the class of $\mathbf{x}$ and $y_j(\mathbf{x})$ is a smoothed cluster "membership function", as defined in (4). In the experiments, the smoothing is only used for optimization; no smoothing is used when evaluating the clustering results. The value for the smoothing parameter $\sigma$ is again selected by validation.

The smoothed MAP objective function (9) becomes

$$\log p(\{\mathbf{m}_j\}|D^{(c)}, D^{(x)}) = \sum_{ij} \log \Gamma\left(n_i^0 + \sum_{c(\mathbf{x})=i} y_j(\mathbf{x})\right) - \sum_j \log \Gamma\left(N^0 + \sum_{\mathbf{x}} y_j(\mathbf{x})\right) + const. \tag{10}$$

For normalized Gaussian membership functions (4), the gradient of the objective function with respect to the $j$th model vector is (Appendix B)

$$\sigma^2 \frac{\partial}{\partial \mathbf{m}_j} \log p(\{\mathbf{m}_j\}|D^{(c)}, D^{(x)}) = \sum_{\mathbf{x},l} (\mathbf{x} - \mathbf{m}_j) y_l(\mathbf{x}) y_j(\mathbf{x}) (L_{j,c(\mathbf{x})} - L_{l,c(\mathbf{x})}), \tag{11}$$

where

$$L_{ji} \equiv \Psi(n_{ji} + n_i^0) - \Psi(N_j + N^0).$$

Here $\Psi$ is the digamma function, the derivative of the logarithm of $\Gamma$. Any standard gradient-based optimization algorithm can be used to maximize (10); we used conjugate gradients.

Alternatively, the objective function (9) can be optimized directly by simulated annealing (SA). The above-described smoothed optimization method is compared with SA in the experimental section of this paper. In each iteration of SA, a candidate step is generated by making small random displacements to the prototype vectors. The step is accepted if it increases the value of the objective function. Even if it decreases the objective function, it is accepted with a probability that is a decreasing function of the change in the objective function.

We used Gaussian displacements with the covariance matrix $\sqrt{T}\sigma^2\mathbf{I}$. Here $\mathbf{I}$ is the identity matrix and $T$ is the temperature parameter that was decreased linearly from 1 to 0.1. The parameter $\sigma$ was chosen in preliminary experiments using a validation set. A displacement step that decreases the objective function by $\Delta E$ is accepted with the probability $\exp(-\Delta E/T)$.

## 4. DC produces optimal contingency tables

A large number of methods for analyzing statistical dependencies between discrete-valued (nominal or categorical) random variables on the basis of co-occurrence frequencies or *contingency tables* exist, many of which are classical (see, for example, [1,9–11]). An old example, due to Fisher, is to measure whether the order of adding milk and tea affects the taste. The first variable indicates the order of adding the ingredients, and the second whether the taste is better or worse. In medicine, one variable could indicate health status and the other demographic groups. The resulting contingency table is tested for dependency of the row and column variables.

Given discrete-valued auxiliary data, the result of any clustering method can be analyzed as a contingency table: The possible values of the auxiliary variable correspond to columns and the clusters to rows of a two-dimensional table. Clustering compresses the potentially large number of multivariate continuous-valued observations into a manageable number of categories, and the contingency table can be tested for dependency. Note that the difference from the traditional use of contingency tables is that the row categories are not fixed; instead, the clustering method tries to find a suitable categorization. The question here is, *is discriminative clustering a good way of constructing such contingency tables?* The answer is that it is optimal in a sense introduced below. First, however, we have to consider the problem of finite sample sizes.

For large sample sizes the sampling variation of the cell frequencies in the table becomes negligible. Then empirical mutual information, approaching the real mutual information as more data become available, would be a natural measure of dependency between the margins of the contingency table.

The various ways to take into account the effects of small sample sizes and/or small cell frequencies of contingency tables have been a subject of much research. Bayesian methods cope well with small data sets; below we will derive a connection between a simple Bayesian approach (a special case of [11]), and our DC method. The classical results were derived for contingency tables with fixed margin categories, while we optimize the categories.

A type of Bayesian test for dependency in contingency tables is based on computing the *Bayes factor* against the hypothesis $H$ of statistical independence of the row and column categories [11]

$$\frac{P(\{n_{ji}\}|\bar{H})}{P(\{n_{ji}\}|H)}.$$

Here $\bar{H}$ is the negation of $H$, that is, it is the alternative hypothesis that the margins are dependent. In practice, the hypotheses will be formulated as (Dirichlet) priors, either as a product of marginal priors (independency) or over all cells (dependency).

In the special case of one fixed margin (the auxiliary data) in the contingency table, and the prior defined in Section 3 with $n_i^0 \equiv n^0$ for all $i$, the Bayes factor is proportional to (8) (Appendix C). MAP estimation of discriminative clusters is thus equivalent to constructing a dependency table that results in a maximal Bayes factor, under the constraints of the model.

## 5. Regularization

A problem with pure DC is that the categories may overfit to apparent dependencies in a small data set. Two regularization methods for the marginalized DC (8) are introduced in this section to reduce overfitting. The first is a straightforward attempt to improve optimization, while the latter is interpretable as joint distribution modeling. Such an explicit modeling of the "covariates" (**x**) may improve discrimination, especially with small data sets (cf. [19]).

### 5.1. Emphasizing equal cluster sizes

In the first, rather non-parametric regularization method, equal distribution of data into the clusters is favored, which is useful at least in avoiding "dead clusters" after bad initialization. The "equalized" or "penalized" objective function is

$$C_{\mathrm{EQ}}(\{\mathbf{m}_j\}) = \sum_{ij} \log \Gamma(n_i^0 + n_{ji}) - (1 + \lambda_{\mathrm{EQ}}) \sum_j \log \Gamma(N^0 + N_j), \tag{12}$$

where $\lambda_{\mathrm{EQ}} > 0$ is a parameter governing the amount of regularization. As the number of data samples increases, (12) divided by $N$ approaches mutual information plus $\lambda_{\mathrm{EQ}}$ times the entropy of the clusters (plus a term that does not depend on the parameters; see Appendix A). Hence, the larger $\lambda_{\mathrm{EQ}}$ is, the more solutions with roughly equal numbers of samples in the clusters are favored.

An alternative to equalization would be to use the prior $n_i^0$ also in place of $N^0$. The effect would be similar to that of (12) in the sense that the second part of the cost function would become more important. Such a prior would be inconsistent from the viewpoint of the derivation of (8) in Section 3, but the prior is wholly justified in the Bayes factor interpretation (Section 4).

### 5.2. Modeling the marginal density of primary data

Discriminative methods that model the conditional probability $p(c|\mathbf{x})$ may benefit from the regularizing effects of modeling the marginal $p(\mathbf{x})$. To investigate whether this is the case with DC, we complemented it to the full joint distribution model

$$p(c, \mathbf{x}|\{\mathbf{m}_j\}, \{\boldsymbol{\psi}_j\}) = p(c|\mathbf{x}, \{\mathbf{m}_j\}, \{\boldsymbol{\psi}_j\}) p(\mathbf{x}|\{\mathbf{m}_j\}) \tag{13}$$

with a generative Gaussian mixture-type model for $p(\mathbf{x})$. Note that both factors of (13) are parameterized by the same centroids $\{\mathbf{m}_j\}$. As will be made explicit in (14) and (15), the special kind of parameterization makes it possible to interpret (13) as an adjustable compromise between modeling $p(\mathbf{x})$ and $p(c|\mathbf{x})$.

Here we use a standard mixture of Gaussians to model $p(\mathbf{x})$, with isotropic Gaussians with covariances $\sigma^2_{\mathrm{MoG}}\mathbf{I}$ and centers $\{\mathbf{m}_j\}$.

### 5.2.1. MAP estimation of clusters of the joint model

With the (improper) prior $p(\{\mathbf{m}_j\}, \{\boldsymbol{\psi}_j\}) \propto p(\{\boldsymbol{\psi}_j\}) = \prod_j p(\boldsymbol{\psi}_j)$, the posterior (8) gets the extra factor

$$\prod_{\mathbf{x}\in D^{(x)}} \sum_j \rho_j \exp(-\lambda_{\mathrm{MoG}}\|\mathbf{x} - \mathbf{m}_j\|^2),$$

where $\lambda_{\mathrm{MoG}} = \frac{1}{2\sigma^2_{\mathrm{MoG}}}$, and $\rho_j$ are the weights of the Gaussians. The parameter $\lambda_{\mathrm{MoG}}$ is used instead of the variances of Gaussians to better illustrate the regularizing nature of the term: $\lambda_{\mathrm{MoG}} = 0$ means no regularization.

Correspondingly, the log posterior of the joint model becomes

$$\begin{aligned}\log p(\{\mathbf{m}_j\}|D^{(c)}, D^{(x)}) \propto & \sum_{ij} \log \Gamma(n_i^0 + n_{ji}) - \sum_j \log \Gamma(N^0 + N_j) \\ & + \sum_{\mathbf{x}\in D^{(x)}} \log \sum_j \rho_j \exp(-\lambda_{\mathrm{MoG}}\|\mathbf{x} - \mathbf{m}_j\|^2), \\ & \equiv MAP_{\mathrm{DC}} - E_{\mathrm{MoG}}(\lambda_{\mathrm{MoG}}) \equiv C_{\mathrm{MoG}}(\{\mathbf{m}_j\}), \end{aligned} \tag{14}$$

where $-E_{\mathrm{MoG}}(\lambda_{\mathrm{MoG}})$ is proportional to the log-likelihood of the mixture of Gaussians, and $C_{\mathrm{MoG}}(\{\mathbf{m}_j\})$ is the final objective function. The model for $p(\mathbf{x})$ can be interpreted as an additive regularization term of the cost function. A change in the value of $\lambda_{\mathrm{MoG}}$ makes the focus of the clustering shift between DC and traditional mixture-based clustering. In practice, the value of $\lambda_{\mathrm{MoG}}$ will be chosen using a validation set to maximize the unregularized cost (8).

### 5.2.2. K-means regularization

This interpretation suggests a simpler, partly heuristic regularization: replacing the log-cost of a mixture of Gaussians by the (negative) cost function of Euclidean $K$-means clustering, that is,

$$E_{\mathrm{VQ}}(\lambda_{\mathrm{VQ}}) = \sum_{j;\mathbf{x}\in V_j} \lambda_{\mathrm{VQ}}\|\mathbf{x} - \mathbf{m}_j\|^2.$$

Here $\lambda_{\mathrm{VQ}}$ has a similar role as the $\lambda_{\mathrm{MoG}}$ in the mixture of Gaussians.

While using $K$-means instead of a mixture of Gaussians is not probabilistically rigorous,[2] it is intuitively meaningful (we can think of it as making a compromise

[2] $K$-means clustering can be derived probabilistically from the so-called "classification mixture" [6], but the final log-posterior would end up having an extra term proportional to $\log Z(\{\mathbf{m}_j\})$, where $Z(\{\mathbf{m}_j\})$ is a sum of Gaussian integrals over the Voronoi regions. Computing $Z(\{\mathbf{m}_j\})$ is infeasible.

between the Kullback–Leibler divergence in (2) and the Euclidean distance in $E_{\mathrm{VQ}}$) and computationally simple. The tunable compromise between DC and $K$-means clustering is apparent if the cost is written as

$$C_{\mathrm{VQ}}(\{\mathbf{m}_j\}) = MAP_{\mathrm{DC}} - E_{\mathrm{VQ}}(\lambda_{\mathrm{VQ}}) = MAP_{\mathrm{DC}} - \lambda_{\mathrm{VQ}} E_{\mathrm{VQ}}(1). \tag{15}$$

## 6. Related methods

Below, connections to some related problems and approaches of data analysis, including feature selection and various clustering criteria, are briefly discussed.

### 6.1. Automatic feature extraction

Proper manual feature selection and extraction is an indispensable but laborious first step in data analysis. Automated methods have been developed for complementing it, especially in pattern recognition applications.

If feature extraction does not change the topology of the input space it is a less general operation than a change of the metric, and DC can be (asymptotically) interpreted as a change of metric [15]. Nevertheless, it may be advisable to use automatic feature extraction methods as preprocessing for DC for two reasons: (i) The clusters of DC are defined to be Euclidean Voronoi regions in the data space. Their shape could, in principle at least, be tuned by transforming the feature space. (ii) Dimensionality reduction reduces the number of parameters and regularizes the solution. Additionally, any desired changes in the topology by discontinuous transformations can be included as preprocessing steps before DC.

### 6.2. DC by pre-estimating densities

A possible alternative approach to DC would be to first find a density estimate $\hat{p}(c|\mathbf{x})$ for the data and then apply more or less standard clustering algorithms on a metric that is based on the estimated densities (see [16]). The most straightforward proximity measure between two points $\mathbf{x}$ and $\mathbf{y}$ would be $D_{\mathrm{KL}}(\hat{p}(c|\mathbf{x}), \hat{p}(c|\mathbf{y}))$. This does not keep the clusters local in the primary data space, however. Instead, a metric locally equivalent to the KL divergence should be generated with the help of the Fisher information matrix for two close-by points.

The problem of the approach is that it involves two unrelated criteria: one for density estimation and another one for clustering. It is hard to see how the two costs could be made commensurable in a principled way. Still, the approach works as a practical data engineering tool, and has been applied [16] to self-organizing maps [17].

### 6.3. Generative co-occurrence models

The term *co-occurrence model* refers to a model of the joint occurrences of nominal variables. For example, in document clustering, the two nominal variables could be the documents and the words, and the documents could be clustered by comparing the occurrences of words within the documents.

From the statistical point of view, the most straightforward method of modeling co-occurrence data of $x$ and $y$ would be to postulate a parameterized probabilistic model $p(x, y|\theta)$ and estimate its parameters $\theta$ by using a conventional criterion such as the maximum likelihood. Based on this approach, Hofmann [14] has introduced a class of mixture models, both for the marginals and the joint distribution. In the field of text document analysis, he coined the joint distribution model probabilistic latent semantic indexing (PLSI).

Conceptually, DC can be seen as a co-occurrence model, or, more exactly, as a special kind of a distributional clustering model for the conditional distributions $p(c|\mathbf{x})$ of the continuous margin $\mathbf{x}$, with the clusters restricted to be local.

### 6.4. Classic distributional clustering and the information bottleneck (IB)

*Distributional clustering* is another term for clustering one margin of co-occurrence data, introduced by Pereira et al. [20]. The IB principle [26,22] gives a deeper justification for the classic distributional clustering.

Although the IB was originally introduced for categorical variables, the principle itself has commonalities with the theory of DC. We therefore discuss the bottleneck in some length here.

Tishby et al. [26] get their motivation from the rate distortion theory of Shannon and Kolmogorov (see [8] for a textbook account). In the rate distortion theory framework, one finds an optimal representation—or conventionally, 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 $X$. The optimality of the representation is measured by its capability to represent another random variable $C$, possibly after being distorted by a noisy transmission channel such as lossy compression. The representation $v$ for an input sample $x$ could in the deterministic case be given by a function $v(x)$, but in general the relationship is stochastic and described by the density $p(v|x)$. The overall frequency of the codes is described by the marginal density $p(v)$.

In the rate distortion theory, the real-valued distortion function $d(x, v)$ is assumed to be known, and the mutual information $I(X; V)$ is minimized with respect to the representation $p(v|x)$, subject to the constraint $E_{X,V}\{d(x, v)\} < k$ (this is made more intuitive below). At the optimum the conditional distributions defining the codebook are

$$p(v_l|x) = \frac{p(v_l)\, \exp[-\beta d(x, v_l)]}{\sum_j p(v_j)\, \exp[-\beta d(x, v_j)]}, \tag{16}$$

where $\beta$ is a constant that depends on $k$. In the information bottleneck, the negative mutual information $-I(C; V)$ is used as the average distortion $E_{X,V}\{d(x, v)\}$.

In more intuitive terms and equivalently to the procedure above, the mutual information $I(C;V)$, earlier presented as the negative distortion, is *maximized*, that is, the representation $V$ is made as informative about $C$ as possible, given a limited value for $I(X;V)$, which is now interpretable as a kind of resource limitation on the representation $V$.

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 (16) with

$$d(x, v_j) = D_{\mathrm{KL}}(p(c|x), p(c|v_j)). \tag{17}$$

The result is self-referential through $p(c|v)$, and therefore does not constitute an algorithm for finding the $p(v|x)$ and $p(c|v)$. An explicit solution can be obtained by an iterative algorithm that resembles the Blahut–Arimoto algorithm (cf. [8]).

In order to clarify the connection to DC, consider a continuous data space $\mathbb{X}$. The bottleneck principle of defining partitions (16) can at least informally be extended to this case: For a continuous $\mathbf{x}$ and the asymptotic case of a large enough (de)regularization parameter $\beta$, the bottleneck clusters in (17) become Voronoi regions of the Kullback–Leibler distortion, were $\mathbb{X}$ categorical or not. The Kullback–Leibler Voronoi regions would be non-local in the $\mathbb{X}$-space. In practice, IB for continuous data would require additional parameterization of the clusters. In DC, clusters have been parameterized as Voronoi regions in the $\mathbb{X}$-space, giving the additional bonus of local clusters. Locality eases interpretation and may be important in some applications.

The cost functions of IB and (asymptotical) DC have a common term, the mutual information $I(C;V)$. The bottleneck has an additional term for keeping the complexity of the representation low, whereas the complexity of discriminative clusters is restricted by their number, parameterization, and in practice by regularization.

Like the original mutual information or KL-distortion cost of DC, the cost of IB is defined for distributions instead of data sets. The straightforward way of applying such a cost function to finite data sets is to approximate the densities by the empirical distributions (see Section 2).

At the limit of crisp clusters, or for uniform distribution of $\mathbf{x}$, IB is equivalent to finding a maximum likelihood solution of a certain multinomial mixture model [23]. To our knowledge, no marginalization procedures similar to marginalized DC have been proposed.

In summary, DC can be interpreted to extend the original distributional clustering paradigm by introducing a continuous variable. The concept of local clusters or the asymptotic connection to metrics described in [15] are not even meaningful in the discrete co-occurrence setup. In practical applications the continuity makes parameterization of the partitions necessary, and the implementation of discriminative clustering becomes very different from the classic co-occurrence models.

### 6.5. Generative models for the joint density of mixed-type variables

It is popular to use finite mixture models for the so-called model-based clustering. In the models, each data sample $\mathbf{x}$ is generated by one of a finite number of generators, identified with the clusters, and the whole density $p(\mathbf{x})$ is a mixture of the densities. The models can be fitted to data by, e.g., maximizing the likelihood with the EM algorithm.

In a model for paired data it has been assumed that each generator generates both the discrete $c$ and the continuous $\mathbf{x}$, from a multinomial and a Gaussian distribution, respectively [13,18]. This model for the joint density $p(c, \mathbf{x})$ is called mixture discriminant analysis (MDA2).

DC models only the conditional density $p(c|\mathbf{x})$. While conditional densities can be derived from models of the joint density by the Bayes rule, it is possible that conditional models perform better because they focus resources more directly on the conditional density. Section 7 provides empirical support for this hypothesis.

On the other hand, regularization by joint modeling (Section 5.2) turns DC towards the traditional joint density models, and we believe that making a compromise between the two extremes provides better generalization ability.

## 7. Experiments

The experiments are divided into three parts. First, DC is demonstrated on a simple toy data set to illustrate its discriminative properties and the effect of regularization. Second, the new finite-data (marginalized) variant is compared with the older stochastic algorithm using standard machine learning data sets. The two optimization algorithms (Section 3.1) for the marginalized DC are additionally compared. Finally, the two regularization principles (Section 5) are tested to find out how much they help when there are few training samples. The closest alternative mixture-based methods are included for reference in all comparisons to demonstrate that DC solves a problem not addressed by standard clustering methods.

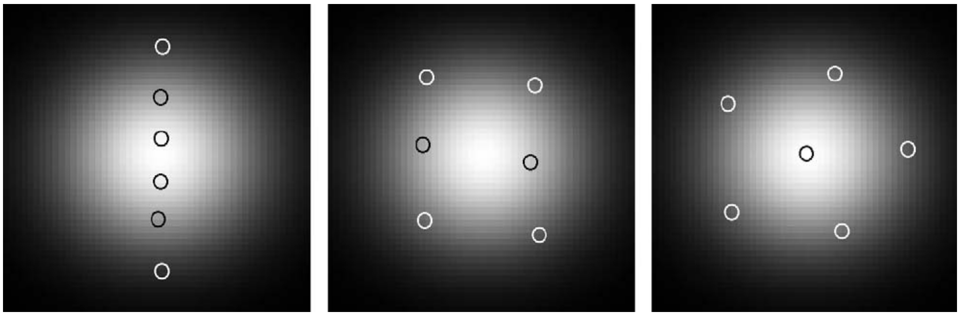

Fig. 1. The VQ-regularized DC model (15) makes a compromise between the plain DC and ordinary $K$-means (VQ). From the viewpoint of plain DC ($\lambda_{\mathrm{VQ}} = 0$; left), only the vertical dimension is relevant as the distribution of the binary auxiliary data $c$ was made to change monotonically and only in that direction. A compromise representation for the data is found at $\lambda_{\mathrm{VQ}} = 0.02$ (middle). The algorithm turns into ordinary VQ when $\lambda_{\mathrm{VQ}} \to \infty$ (right). Circles denote the Voronoi region parameters $\{\mathbf{m}_j\}$ and gray shades the density $p(\mathbf{x})$.

### 7.1. Toy demonstration

The Voronoi region centers $\{\mathbf{m}_j\}$ of the VQ-regularized model (15) are shown in Fig. 1 for three different values of the regularization parameter $\lambda_{\mathrm{VQ}}$. Data (10,000 samples) were from an isotropic 2D Gaussian with a vertically varying $p(c|\mathbf{x})$. Samples come from two classes, 5000 samples each, and $p(c_1|\mathbf{x})$ increases monotonically from bottom to top. Naturally, $p(c_2|\mathbf{x}) = 1 - p(c_1|\mathbf{x})$ then increases from top to bottom. For small values of $\lambda_{\mathrm{VQ}}$, the original cost function of DC is minimized, and the clusters represent only the vertical direction of the $\mathbb{X}$-space, where the conditional distribution $p(c|\mathbf{x})$ changes. When $\lambda_{\mathrm{VQ}}$ increases, the clusters gradually start to represent all variation in $\mathbf{x}$, converging to the $K$-means solution for large $\lambda_{\mathrm{VQ}}$. Similar compromise would be found with the mixture of Gaussians regularization (14) as well.

### 7.2. Comparison with the stochastic DC

The performance of the older stochastic on-line algorithm presented in Section 2.1 and the two optimization algorithms for maximizing the marginalized posterior given in Section 3.1 are compared here on three real-life data sets. Two standard mixture models, a mixture of Gaussians modeling $p(\mathbf{x})$ and MDA2 modeling $p(c, \mathbf{x})$ (see Section 6.5), are included for reference. Thus, the comparisons also show whether discriminative modeling outperforms ordinary clustering methods in its task.

#### 7.2.1. Materials and methods

The algorithms were compared on three data sets: the Landsat satellite data (36 dimensions, six classes, and 6435 samples) and the Letter Recognition data (16 dimensions, 26 classes and 20,000 samples) from the UCI Machine Learning Repository [3], and speech data from the TIMIT collection [25]. Altogether, 14,994 samples were picked up from the TIMIT material, classified into 41 groups of phones (phonetic sounds), and encoded into 12 cepstral components.

First, the best values for the smoothing parameter $\sigma$ were sought in a series of preliminary runs. The data sets were partitioned into 2, 5, and 10 clusters, using the class indicators as the auxiliary data. For each number of clusters, solutions were computed at 30 logarithmically spaced values of the smoothing parameter $\sigma$ of DC, and at 30 similar values of the spread of the Gaussians of the mixture models. Another set of 30 logarithmically spaced values was tried for the width of the jumping kernel in SA.

The cluster prototypes (centers) of all models were initialized to random draws from data. A conservatively large number of iterations was chosen: 100 EM iterations for the mixture models, and 100,000 times the number of clusters for the stochastic iterations with the non-marginalized DC and SA with marginalized DC. The maximal number of iterations for the marginalized DC optimized with conjugate gradients was set to 29, but the algorithm converged well before that in most cases.

All the prior parameters $n_i^0$ were set to unity. The adaptation coefficient $\alpha$ in (5) of the old stochastic algorithm decreased piecewise-linearly from 0.05 to zero, and the coefficient in (6) was two times larger.

The performance of the methods is here compared using the posterior probability (9) of the cluster prototypes, computed from held-out data. Note that the auxiliary parts of the held-out data are not used in any way in computing the cluster identities (which are a function of the primary data alone).

The posterior probability is a justified measure for the goodness of discriminative clusters, irrespective of how the clusters are generated. It is somewhat problematic, however, that it is also the cost function of some of our own methods. Therefore, the main conclusion to be drawn from the comparisons is that DC does what it promises. An alternative goodness measure would be the empirical mutual information of the cluster identities and the nominal auxiliary data labels. This produces practically identical results (not shown).

### 7.2.2. Results

The significance of the performance differences was tested with two-tailed $t$-tests over 10-fold cross-validation runs (Table 1). For each combination of a model and a cluster count, the smoothing parameter of the model was fixed to its best value found in the preliminary phase of the experiments.

The best DC variant was always significantly better than either of the non-DC methods. On Landsat and TIMIT data sets, the marginalized MAP variant is the best regardless of the number of clusters. In the more interesting cases, the five- and ten-cluster solutions, the best result is obtained with the conjugate gradient algorithm. On the Letter Recognition data, the old stochastic algorithm produces the best results, but the difference to the marginalized DC optimized with conjugate gradients is insignificant.

Table 1
Average cost (negative log posterior) of the algorithms over ten-fold cross-validation trials

| Data | $N_c$ | CG MAP | SA MAP | sDC | MoG | MDA2 |
|---|---|---|---|---|---|---|
| Landsat | 2 | 833.86 | **828.75** | 953.78 | 913.87 | 918.86 |
| Landsat | 5 | **472.40** | 493.03 | 701.10 | 623.09 | 649.77 |
| Landsat | 10 | **432.81** | 455.07 | 550.52 | 504.74 | 494.64 |
| TIMIT | 2 | 4651.1 | **4537.8** | 4577.3 | 4577.8 | 4550.6 |
| TIMIT | 5 | **4514.9** | 4516.6 | 4548.4 | 4584.4 | 4579.9 |
| TIMIT | 10 | **4860.3** | 4874.9 | 4886.0 | 5085.6 | 4953.4 |
| Letter | 2 | **5904.2** | 5911.2 | 5928.3 | 6451.6 | 6126.7 |
| Letter | 5 | 5109.4 | 5050.0 | **5046.3** | 6330.4 | 5436.8 |
| Letter | 10 | 4704.8 | 4821.5 | **4632.3** | 6183.9 | 5186.0 |

Best performance for each cluster number ($N_c$) is shown in bold, and results with $p$-value under 0.01 (pairwise $t$-test) have been doubly underlined. Single underlining denotes $p$-value under 0.05. *CG MAP*: MAP estimation of smoothed DC by a conjugate gradient algorithm; *SA MAP*: MAP estimation by simulated annealing; *sDC*: DC by old stochastic algorithm; *MoG*: mixture of Gaussians; *MDA2*: a mixture model for joint probabilities.

Three conclusions can be drawn. (i) The DC algorithms perform clearly better in their task than standard clustering algorithms, which is expected as they solve a problem not directly addressed by standard clustering. (ii) The new marginalized version gives results better than (two of the three data sets) or comparable to the old stochastic version. (iii) The two optimization algorithms of marginalized DC are comparable, the conjugate gradient algorithm having a clear edge on ten-cluster solutions. Because of its clearly faster computation, it is therefore the preferred choice in most cases.

## 7.3. *Effect of regularization*

### 7.3.1. *Methods*

Next we compared the plain marginalized DC model and its regularized variants on two data sets, with the final performance of the models measured by the pure DC objective function (9), that is, without any regularization. The closest alternative mixture models were again included for reference, as was DC optimized with the stochastic on-line algorithm. To keep the experiment set manageable, no regularization methods were applied to the stochastic algorithm. This time also classical Euclidean VQ ($K$-means) was included for completeness, as it is used for regularization in one of the models. The marginalized DC models were optimized by the conjugate gradient algorithm, based on the results of the previous section.

Since the effects of regularization were expected to be most apparent for small data sets, the data were split into a number of smaller subsets on which a set of independent tests were made. The Landsat data were left out, because it had too few samples for the test setup.

The Letter Recognition data were split into five subsets. Two-fold modeling and testing for each subset gave a total of ten repetitions of ten-cluster solutions. The width parameter of the mixture components and smoothing, and the regularization parameters were selected by five-fold cross-validation within each learning set. The parameters $\{\mathbf{m}_j\}$ were initialized to a random set of training samples. (Results with the $K$-means initialization appearing in Table 2 are from experiments that will be described later in the paper.)

A larger subset (99,983 samples) of the TIMIT collection was used, and it allowed us to use ten subsets, resulting in 20 repetitions (with parameters within each repetition selected by three-fold cross-validation).

### 7.3.2. *Results*

The best regularized methods were significantly better than plain marginalized DC, which in turn produced better discriminative clusters than the reference methods. The results (columns “Letter rand” and “TIMIT rand” in Table 2) are clear for the TIMIT data, where the old stochastic algorithm is also significantly worse than the regularized marginalized DC. On the Letter Recognition data, the old stochastic algorithm was the best, as in the previous set of experiments (Section 7.2), but the difference to the best marginalized DC is insignificant. Note that DC

Table 2
Comparison of marginalized DC and its regularized versions DC-VQ (15), DC-MoG (14), and DC-EQ (12) on two data sets, Letter Recognition and TIMIT

| Method | Letter rand | Letter VQ | TIMIT rand | TIMIT VQ |
|---|---|---|---|---|
| sDC | **4769.1** | 4830.8 | 13231 | 12792 |
| DC | 4961.9 | 4816.9 | 12981 | 12780 |
| DC-VQ | 4933.4 | 4779.5 | 12905 | 12767 |
| DC-MoG | 4843.0 | 4763.7 | **12876** | **12718** |
| DC-EQ | 4864.1 | **4699.8** | 12942 | 12757 |
| MoG | 6174.9 | 6210.8 | 13515 | 13494 |
| VQ | 6194.9 | 6194.9 | 13487 | 13487 |
| MDA2 | 5206.4 | 5280.8 | 13012 | 12989 |

The first line (sDC) comes from the old stochastic algorithm. Mixture of Gaussians (MoG), plain *K*-means (VQ), and joint density model MDA2 [13] have been included for reference. The results are presented for both random and *K*-means (VQ) initialization. Key: see Table 1, and note that *p*-values are here computed along columns, not rows.

regularized with the mixture of Gaussians is significantly better than plain marginalized DC also on the Letter Recognition data, although the difference is not visible in Table 2. Combined, these results show that regularization helps marginalized DC, but the performance compared to the old stochastic algorithm still depends on the data. The heuristic regularization with *K*-means seems to have slightly lower performance compared to the probabilistically justified mixture of Gaussians, but the difference is not significant.

In Fig. 2, the effect of tuning the compromise between *K*-means and DC in VQ-regularization is demonstrated. As expected, increasing $\lambda_{\mathrm{VQ}}$ shifts the solution from optimizing the posterior probability (9) towards optimizing the *K*-means error. The new finding is the slanted L-form: slight regularization improves the predictive power of the clusters for the test set. Replacing *K*-means with a mixture of Gaussians gives a similar curve.

Finally we studied, by repeating the ten-cluster experiments of Table 2, whether replacing the random initialization with *K*-means would improve results and reduce variation between data sets. The results of almost all DC variants improved significantly (columns "Letter VQ" and "TIMIT VQ" in Table 2). The only exception was the old stochastic algorithm on the Letter Recognition data; against expectation, its performance decreased. Regularized versions were still the best, but their relative goodness varied.

The results from the regularization experiments can be summarized into three main points. (i) Regularization improves the performance of marginalized DC on small data sets. (ii) The relative performance of the two regularization principles depends on the data. (iii) Initialization by *K*-means significantly improves the performance, and should be used instead of random initialization.

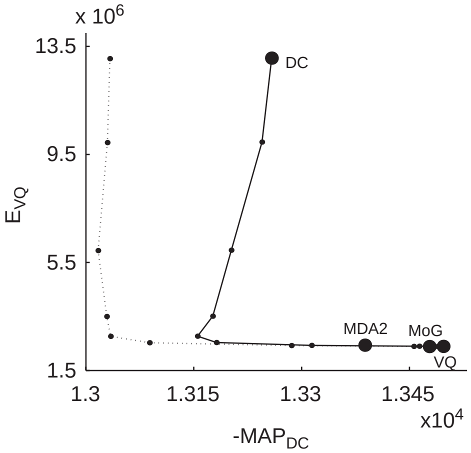

Fig. 2. The effect of tuning the VQ-regularization on TIMIT data. The curves show how the two components of the cost change as the amount of regularization is tuned. The two components are: $K$-means cost ($E_{\mathrm{VQ}}$) and predictive power ((9); $MAP_{\mathrm{DC}}$). Small dots on the curves: VQ-regularized DC with varying parameter $\lambda_{\mathrm{VQ}}$; large dots from left to right: plain DC, MDA2, mixture of Gaussians (MoG), plain $K$-means (VQ). Solid line: test set; dashed line: learning set. Results are averages over cross-validation runs, and for computational reasons the parameter $\sigma$ of the DC runs was not cross-validated but kept constant.

## 8. Discussion

An algorithm for distributional clustering of continuous data, interpreted as covariates of discrete data, was reviewed and extended. With prototype distributions of the discrete variable associated to Voronoi regions of the continuous data space, the regions are optimized to “predict” the discrete data well. In experiments, the method produced better-discriminating clusters than other common methods.

The core DC model for $p(c|\mathbf{x})$ is very close to models proposed earlier for classification (e.g. RBF [18]). In DC, however, the main outcome are clusters of $\mathbf{x}$, which enabled us to marginalize out the parameters producing predictions of $c$. The new cost function takes into account the finite amount of data, and its maximization is equivalent to maximizing a Bayesian measure for statistical dependency in contingency tables.

Optimization of the new cost function leads to clustering results that are comparable to or better than those produced by the previously presented stochastic on-line algorithm. We also augmented the new cost function by two regularization methods. Similar kinds of regularization approaches are applicable also for the old infinite-data cost functions.

In addition to the new cost function, this paper contains three new empirical results. (i) The fast optimization of smoothed Voronoi regions by conjugate gradients produces clusters comparable to those obtained by the considerably more time-consuming simulated annealing (SA). (ii) The two regularization methods, equalization of the cluster sizes and shifting towards a joint distribution model, improve the results compared to plain DC. No conclusion could be drawn of the relative goodness of the two methods, however. (iii) Initialization is important: $K$-means is superior to initialization by random data.

Regularization by joint distribution modeling is interpretable as the inclusion of a term modeling the primary data in the cost function. The number of parameters in the regularized models is independent of the regularization parameter $\lambda$, and in this sense the model complexity is fixed. A regularized model therefore makes a compromise, tunable by $\lambda$, in representing variation of $\mathbf{x}$ associated with changes in $p(c|\mathbf{x})$ (the DC task), and in representing all variation isotropically (the classical clustering task). In the experiments with regularization, performance on learning data is not impaired, while test set performance improves significantly. For some reason, therefore, allocating resources to modeling $p(\mathbf{x})$ improves generalization with respect to $p(c|\mathbf{x})$.

An adjustable combination of two mixture models was recently proposed for joint modeling of terms and hyperlinks in text documents [7]. Here a similar combination improved a discriminative (conditional-density) model. The joint distribution modeling approach also makes it possible to treat primary data samples lacking the corresponding auxiliary part as partially missing data, along the lines of "semisupervised learning" proposed for classification tasks [24].

Finally, the improvement obtained by $K$-means initialization hints at a practical optimization strategy that starts with standard clustering and tunes it gradually towards DC.

## Acknowledgements

This work was supported in part by the Academy of Finland, grant 52123, and by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views. The authors would like to thank Janne Nikkilä for insights on the contingency tables, and acknowledge that access rights to the materials produced in this project are restricted due to other commitments.

## Appendix A. Connection of the marginalized likelihood to mutual information

Consider the objective function of the penalized clustering algorithm,

$$C_{\mathrm{EQ}}(\{\mathbf{m}_j\}) = \sum_{ij} \log \Gamma(n_i^0 + n_{ji}) - (1+\lambda) \sum_j \log \Gamma(N^0 + N_j),$$

which (up to a constant) reduces to (9) if $\lambda = 0$. The Stirling approximation $\log \Gamma(s+1) = s \log s - s + \mathcal{O}(\log s)$ applied to (9) yields

$$C_{\mathrm{EQ}}(\{\mathbf{m}_j\}) = \sum_{ij} (n_{ji} + k_{ji}) \log(n_{ji} + k_{ji}) - (1+\lambda) \sum_j (N_j + k_j) \log(N_j + k_j) + \mathcal{O}(\log N),$$

where $k_{ji}$ and $k_j$ are constants that depend on the prior. Note that $N \geqslant N_j \geqslant n_{ji}$.

The zeroth-order Taylor expansion $\log(s+k) = \log s + \mathcal{O}(k/s)$ gives

$$C_{\mathrm{EQ}}(\{\mathbf{m}_j\}) = \sum_{ij} n_{ji} \log n_{ji} - (1+\lambda) \sum_j N_j \log N_j + \mathcal{O}(\log N).$$

Division by $N$ then gives

$$\frac{C_{\mathrm{EQ}}(\{\mathbf{m}_j\})}{N} = \sum_{ij} \frac{n_{ji}}{N} \log \frac{n_{ji}/N}{N_j/N} - \lambda \sum_j \frac{N_j}{N} \log \frac{N_j}{N} - \lambda \log N + \mathcal{O}\left(\frac{\log N}{N}\right),$$

where $n_{ji}/N$ approaches $p_{ji}$, that is, the probability of class $i$ in cluster $j$, and $N_j/N$ approaches $p_j$ as the number of data samples increases. Hence,

$$\frac{C_{\mathrm{EQ}}(\{\mathbf{m}_j\})}{N} \to \sum_{ij} p_{ji} \log \frac{p_{ji}}{p_j p_i} - \sum_i p_i \log \frac{1}{p_i} + \lambda \sum_j p_j \log \frac{1}{p_j} - \lambda \log N,$$

where the first term is the mutual information, the second term is a constant (with respect to $\{\mathbf{m}_j\}$), the third term is $\lambda$ times the entropy of $p_j$, and the last term does not depend on the parameters.

For $\lambda = 0$ the result is equal to mutual information added by a constant.

## Appendix B. Gradient of the marginalized likelihood

Denote for brevity $t_{ji} = n_{ji} + n_i^0$ and $T_j = \sum_i t_{ji}$. The gradient of (10) with respect to $\mathbf{m}_j$ is

$$\begin{aligned} \frac{\partial}{\partial \mathbf{m}_j} \log p(\{\mathbf{m}_j\}|D^{(c)}, D^{(x)}) &= \sum_{il} \Psi(t_{li}) \sum_{c(\mathbf{x})=i} \frac{\partial}{\partial \mathbf{m}_j} y_l(\mathbf{x}) - \sum_{\mathbf{x},l} \Psi(T_l) \frac{\partial}{\partial \mathbf{m}_j} y_l(\mathbf{x}) \\ &= \sum_{\mathbf{x},l} [\Psi(t_{l,c(\mathbf{x})}) - \Psi(T_l)] \frac{\partial}{\partial \mathbf{m}_j} y_l(\mathbf{x}). \end{aligned}$$

It is straightforward to show that for normalized Gaussian membership functions

$$\frac{\partial}{\partial \mathbf{m}_j} y_l(\mathbf{x}) = \frac{1}{\sigma^2} (\mathbf{x} - \mathbf{m}_j)(\delta_{lj} - y_l(\mathbf{x})) y_j(\mathbf{x}).$$

Substituting this to the gradient gives

$$\sigma^2 \frac{\partial}{\partial \mathbf{m}_j} \log p(\{\mathbf{m}\}|\mathbf{D}^{(c)}, \mathbf{D}^{(x)}) = \sum_{\mathbf{x},\mathbf{l}} (\mathbf{x} - \mathbf{m}_j)(\delta_{lj} - \mathbf{y}_l(\mathbf{x}))\mathbf{y}_j(\mathbf{x})[\Psi(\mathbf{t}_{l,c(\mathbf{x})}) - \Psi(\mathbf{T}_l)]. \tag{B.1}$$

The final form (11) for the gradient results from applying the identity

$$\sum_l (\delta_{lj} - y_l) y_j L_l = \sum_l y_l y_j (L_j - L_l)$$

to (B.1).

## Appendix C. Connection of marginalized likelihood to contingency tables

A connection between the posterior probability (8) and a Bayesian measure for statistical dependency in contingency tables is derived below. Note that in contrast to a connection with mutual information, which could in principle be used to measure statistical dependency, this connection is non-asymptotic.

Denote the number of samples in the $i$th auxiliary category, that is, the number of entries in the $i$th column margin of the contingency table, by $n(c_i)$. In our application of contingency tables, only this margin is fixed—the other consists of the discriminative clusters and therefore depends on the cluster centroids $\{\mathbf{m}\}$.

As explained in Section 4, evidence for dependence of the margin variables can be quantified with the Bayes factor, implicating the relative strength of evidence for dependence (see [11] for an advanced treatment with infinite mixtures). For computing the Bayes factor, assumptions of dependence and independence are encoded into the joint prior distribution of data over the cells: Under the hypothesis $H$ of independence between the rows and columns, the prior probability of data in each cell is the product of margin (Dirichlet) prior probabilities, whereas under the hypothesis of dependence, the prior is simply one Dirichlet distribution over all the cells.

The Dirichlet distribution has a sharpness parameter which may be interpreted as the amount of "prior data". For the *hypothesis of dependence* $\bar{H}$, the Dirichlet prior with the same amount of prior data, denoted here by $n^0$, in each cell of the whole contingency table has been used. Under the *hypothesis of dependence* $H$, we need a Dirichlet prior for the columns and the rows. For the row margin, a Dirichlet distribution with an equal amount of prior data for each row has been used. In contrast to Good [11], we assume the same total amount of "prior data" under both hypotheses. Then the prior sample size of rows under $H$ is $N^0 = \sum_i n^0$, the prior for $\bar{H}$ marginalized. The prior for the column margin follows similarly from consistency (detailed below). The Bayes factor against $H$, conditioned on the column margin, is then

$$\frac{P(\{n_{ji}\}|\{n(c_i)\}, \bar{H})}{P(\{n_{ji}\}|\{n(c_i)\}, H)}. \tag{C.1}$$

The denominator is

$$\begin{aligned} P(\{n_{ji}\}|H) &= P(\{n_{ji}\}, \{N_j\}, \{n(c_i)\}|H) \\ &= P(\{n_{ji}\}|\{N_j\}, \{n(c_i)\}, H)P(\{N_j\}|H)P(\{n(c_i)\}|H). \end{aligned}$$

The first factor, the frequencies of data in the table given the margins, follows the hypergeometric distribution, and the second factor $P(\{N_j\}|H)$ is

$$P(\{N_j\}|H) = \int_{\boldsymbol{\theta}} P(\{N_j\}|\boldsymbol{\theta})p(\boldsymbol{\theta}|H)\,\mathrm{d}\boldsymbol{\theta}, \tag{C.2}$$

where $p(\boldsymbol{\theta}|H)$ is the Dirichlet prior and $\boldsymbol{\theta}$ are the parameters of the multinomial distribution. For multinomial data $N_j$ with $K$ nominal values and the Dirichlet prior with an equal amount $N^0 = \sum_i n^0$ of prior data for each of the $K$ values (formula 2.5 in [11]),

$$P(\{N_j\}|H) = \frac{\Gamma(KN^0)N!\prod_j \Gamma(N_j + N^0)}{\Gamma(N^0)^K \Gamma(N + KN^0)\prod_j N_j!}. \tag{C.3}$$

We get a similar expression for the third factor $P(\{n(c_i)\}|H)$, but as it is independent of the cluster solution and only depends on the column margin $\{n(c_i)\}$ which is fixed from the viewpoint of DC, we have omitted the derivation. (For consistency, the columns would have the amount $Kn^0$ of prior data.)

The numerator of (C.1) is

$$P(\{n_{ji}\}|\{n(c_i)\}, \bar{H}) = \frac{P(\{n_{ji}\}|\bar{H})}{P(\{n(c_i)\}|\bar{H})},$$

where $P(\{n_{ji}\}|\bar{H})$ is similar to (C.3) but the products go over all cells of the table and the prior data are $n^0$ for each cell. We have $P(\{n_{ji}\}|\bar{H}) \propto \prod_{i,j} \Gamma(n_{ji} + n^0)$, with irrelevant factors omitted. The margin $P(\{n(c_i)\}|\bar{H})$, on the other hand, is identical to $P(\{n(c_i)\}|H)$ and again a constant for DC.

After these considerations, the Bayes factor can be written as

$$\frac{P(\{n_{ji}\}|\{n(c_i)\}, \bar{H})}{P(\{n_{ji}\}|\{n(c_i)\}, H)} = \frac{\prod_{i,j}\Gamma(n_{ji} + n^0)}{\prod_j (N_j + N^0)} \times const. = p(\{\mathbf{m}_j\}|D^{(c)}, D^{(x)}) \times const.,$$

where the constant depends on neither $N_j$ nor $n_{ji}$.

## References

[1] A. Agresti, A survey of exact inference for contingency tables, Statist. Sci. 7 (1992) 131–153.

[2] S. Becker, Mutual information maximization: models of cortical self-organization, Netw. Comput. Neural Syst. 7 (1996) 7–31.

[3] C.L. Blake, C.J. Merz, UCI repository of machine learning databases, http://www.ics.uci.edu/~mlearn/MLRepository.html (1998).

[4] D. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.

[5] W. Buntine, Variational extensions to EM and multinomial PCA, in: T. Elomaa, H. Mannila, H. Toivonen (Eds.), Proceedings of the ECML'02, 13th European Conference on Machine Learning, Springer, Berlin, 2002, pp. 23–34.

[6] G. Celeux, G. Govaert, A classification EM algorithm for clustering and two stochastic versions, Comput. Statist. Data Anal. 14 (1992) 315–332.

[7] D. Cohn, T. Hofmann, The missing link—a probabilistic model of document content and hypertext connectivity, in: T. Leen, T. Dietterich, V. Tresp (Eds.), Advances in Neural Information Processing Systems 13, MIT Press, Cambridge, MA, 2001.

[8] T.M. Cover, J.A. Thomas, Elements of Information Theory, Wiley, New York, 1991.

[9] R.A. Fisher, On the interpretation of $\chi^2$ from the contingency tables, and the calculation of $P$, J. Roy. Statist. Soc. 85 (1922) 87–94.

[10] G.H. Freeman, J.H. Halton, Note on an exact treatment of contingency, goodness of fit and other problems of significance, Biometrika 38 (1951) 141–149.

[11] I.J. Good, On the application of symmetric Dirichlet distributions and their mixtures to contingency tables, Ann. Statist. 4 (1976) 1159–1189.

[12] T. Hastie, R. Tibshirani, Discriminant analysis by Gaussian mixtures, J. Roy. Statist. Soc. B 58 (1996) 155–176.

[13] T. Hastie, R. Tibshirani, A. Buja, Flexible discriminant and mixture models, in: J. Kay, D. Titterington (Eds.), Neural Networks and Statistics, Oxford University Press, Oxford, 1995.
[14] T. Hofmann, Unsupervised learning by probabilistic latent semantic analysis, Machine Learn. 42 (2001) 177–196.
[15] S. Kaski, J. Sinkkonen, Principle of learning metrics for data analysis, The J. VLSI Signal Process. Syst. Signal Image Video Technol. Special Issue Machine Learn. Signal Process. 37 (2004) 177–188.
[16] S. Kaski, J. Sinkkonen, J. Peltonen, Bankruptcy analysis with self-organizing maps in learning metrics, IEEE Trans. Neural Netw. 12 (2001) 936–947.
[17] T. Kohonen, Self-Organizing Maps, third ed., Springer, Berlin, 2001.
[18] D.J. Miller, H.S. Uyar, A mixture of experts classifier with learning based on both labelled and unlabelled data, in: M. Mozer, M. Jordan, T. Petsche (Eds.), Advances in Neural Information Processing Systems 9, MIT Press, Cambridge, MA, 1997, pp. 571–577.
[19] A.Y. Ng, M.I. Jordan, On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes, in: T.G. Dietterich, S. Becker, Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems 14, MIT Press, Cambridge, MA, 2002.
[20] F. Pereira, N. Tishby, L. Lee, Distributional clustering of English words, in: Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics, ACL, Columbus, OH, 1993, pp. 183–190.
[21] J. Sinkkonen, S. Kaski, Clustering based on conditional distributions in an auxiliary space, Neural Comput. 14 (2002) 217–239.
[22] N. Slonim, N. Tishby, Agglomerative information bottleneck, in: S.A. Solla, T.K. Leen, K.-R. Müller (Eds.), Advances in Neural Information Processing Systems 12, MIT Press, Cambridge, MA, 2000, pp. 617–623.
[23] N. Slonim, Y. Weiss, Maximum likelihood and the information bottleneck, in: S. Becker, S. Thrun, K. Obermayer (Eds.), Advances in Neural Information Processing Systems 15, MIT Press, Cambridge, MA, 2003, pp. 335–342.
[24] M. Szummer, T. Jaakkola, Kernel expansions with unlabeled examples, in: T. Leen, T. Dietterich, V. Tresp (Eds.), Advances in Neural Information Processing Systems 13, MIT Press, Cambridge, MA, 2001, pp. 626–632.
[25] TIMIT, CD-ROM prototype version of the DARPA TIMIT acoustic-phonetic speech database, 1998.
[26] N. Tishby, F.C. Pereira, W. Bialek, The information bottleneck method, in: 37th Annual Allerton Conference on Communication, Control and Computing, Urbana, IL, 1999, pp. 368–377.

**Samuel Kaski** received the D.Sc. (Ph.D.) degree in Computer Science from Helsinki University of Technology, Espoo, Finland, in 1997. He is currently Professor of Computer Science at University of Helsinki, Finland. His main research areas are statistical machine learning and data mining, bioinformatics, and information retrieval.

**Janne Sinkkonen** received an M.A. degree in psychology from University of Helsinki in 1996, and the Ph.D. degree on machine learning from Helsinki University of Technology (HUT) in 2004. He has worked as a researcher in a Helsinki University brain research group during 1990s, and as a researcher at the HUT Neural Networks Research Centre during 2000s. He is currently at Xtract Ltd.

**Arto Klami** received the M.Sc. degree in computer and information science from Helsinki University of Technology, Espoo, Finland, in 2003. He is currently working toward the Ph.D. degree on modeling dependencies between several data sources at the Department of Computer Science and Engineering, Helsinki University of Technology, Espoo, Finland. His research interests include machine learning and exploratory data analysis.