Discriminative Components of Data

Jaakko Peltonen and Samuel Kaski, *Senior Member, IEEE*

***Abstract*— A simple probabilistic model is introduced to generalize classical linear discriminant analysis in finding components that are informative of or relevant for data classes. The components maximize the predictability of the class distribution which is asymptotically equivalent to (i) maximizing mutual information with the classes, and (ii) finding principal components in the so-called learning or Fisher metrics. The Fisher metric measures only distances that are relevant to the classes, that is, distances that cause changes in the class distribution. The components have applications in data exploration, visualization, and dimensionality reduction. In empirical experiments the method outperformed, in addition to more classical methods, a Renyi entropy-based alternative while having essentially equivalent computational cost.**

***Index Terms*— Component model, discriminant analysis, exploratory data analysis, learning metrics, mutual information**

I. Introduction

THE goal of this work is to learn discriminative components of multivariate continuous data. Linear discriminant analysis (LDA; [1], see [2]) is a classical method for this task. The LDA components have traditionally been used for classification, that is, discriminating the classes. They construct Bayes-optimal class borders in the two-class case, assuming the classes are normally distributed and share the same covariance matrix.

Numerous alternative methods and generalizations have been developed for classification. However, the LDA components are additionally useful for describing and characterizing class separation and the contribution of original variables to it, and for visualizing the data. Our goal is to generalize LDA as a component model for these latter purposes, by removing the restrictive assumption of normal distribution with equal covariance matrices in each class.

Our view to why LDA-based visualizations are useful is that discriminant analysis finds, intuitively speaking, directions that are *relevant* to or informative of the classification. Relevance obviously needs to be defined more exactly before it can be usefully applied. Our second goal in this paper, besides generalizing LDA, is to define more rigorously what it means for components to be relevant for classes.

Mutual information is a natural measure of the (asymptotical) statistical dependency between two random variables, such as the primary data and their classes. Becker and Hinton [3], [4] suggested maximizing the mutual information between two processing modules to model common properties in their inputs; the common properties could be interpreted as the relevant ones. Tishby et al. [5] formalized relevance in a very closely related way as constrained maximization of mutual information, using rate distortion theory applicable to discrete data. A sample application is clustering documents by the common occurrences of words in them [6].

Torkkola [7], [8] optimized projections by maximizing mutual information, to produce a discriminative feature transformation. Instead of the standard mutual information based on Shannon entropy, however, he used Renyi entropy. Renyi-based formalism using formulas from [9], [10] was claimed to be more suitable than the traditional Shannon entropy since it avoids computational difficulties.

In this paper we generalize linear discriminant analysis by extending this line of work about maximizing mutual information. We introduce a very simple generative model that can be optimized using standard machinery of probabilistic inference, instead of resorting to infinite-data formalisms based on either Shannon or Renyi entropy. The obvious problem with such entropy-based formalisms is that probability distributions need either be assumed or estimated. The proposed model asymptotically maximizes Shannon mutual information, and its computational cost is essentially equivalent to the Renyi-based alternative, suggesting that switching to Renyi may be unnecessary. The relative goodness of the alternatives is investigated empirically.

In summary, the main advantages of the proposed model are that it is very simple and consistent with both of the relevant traditions: generative (predictive) probabilistic modeling and modeling relevant (discriminative) properties of data by maximizing mutual information. The components are relevant to the classes in the sense of being predictive or informative of them.

The remaining objective is to justify why the model is useful as a component model for characterizing and visualizing class separation and the contributions of the variables to it. This is done through a connection to still another formalism, the learning metrics principle [11], [12], which uses information-geometric methods to construct (Riemannian) metrics to the data space. Distances correspond to class changes; assuming the classes are here relevant, this is then precisely the metric that measures the relevant differences. We sketch a connection to show that the proposed model can be asymptotically interpreted as principal component analysis in the learning metrics.

The components are expected to be useful for reducing the dimensionality for visualization, exploration, and

The authors are with the Neural Networks Research Centre, Helsinki University of Technology, P.O. Box 5400, FIN-02015 HUT, Finland. S. Kaski is currently with the Department of Computer Science, P.O. Box 26, FIN-00014 University of Helsinki, Finland. E-mail: jaakko.peltonen@hut.fi, samuel.kaski@cs.helsinki.fi

interpretation of the primary data, or alternatively as a preprocessing transformation for further analysis.

II. The model

The learning data consists of pairs $(\mathbf{x}, c)$, where the primary data $\mathbf{x}$ are multivariate samples from the vector space $\mathbb{R}^n$. In this work the auxiliary data c are categorical (multinomial), attaining one from a set of N_c unordered values, the classes. The two key assumptions are that (i) analysis of the primary data is of the main interest, and (ii) the classification has been chosen properly such that variation in the primary data is assumed relevant or important only to the extent it causes changes in the c.

We search for dimensionality-reducing transformations of the primary data to smaller-dimensional vectors $\mathbf{y} = \mathbf{f}(\mathbf{x})$, $\mathbf{y} \in \mathbb{R}^d$. In this paper the transformation is linear, $\mathbf{f}(\mathbf{x}) = \mathbf{W}^T\mathbf{x}$, where $\mathbf{W}$ is the orthogonal transformation matrix to be optimized. The columns $\mathbf{w}_i$ of $\mathbf{W}$ are the basis vectors of the reduced-dimensional space that is a subspace of the primary data space. The basis vectors decompose the data into *components* $\mathbf{w}_i^T\mathbf{x}$. Note that the transformation as such does not depend on the auxiliary data. Hence, once it has been optimized, it can also transform new primary data without known auxiliary data.

A. Objective function

The goal is to find a transformation that makes the subspace as informative as possible of the classes. Assuming the classification defines what is interesting or important, the columns of the estimated transformation matrix $\mathbf{W}$ represent the 'informative' components of the primary data.

Informativeness will be measured by predictive power, by constructing a generative probabilistic model of c given the projected value $\mathbf{f}(\mathbf{x})$, and maximizing its log-likelihood. The model then has a well-defined criterion for fitting it to finite data.

The generative model predicts the distribution of c based on the projected value; the prediction is denoted by $\hat{p}(c|\mathbf{f}(\mathbf{x}))$. The log-likelihood of the model for the paired data $\{(\mathbf{x}, c)\}$ is

$$L = \sum_{(\mathbf{x},c)} \log \hat{p}(c|\mathbf{f}(\mathbf{x})) \tag{1}$$

where $\mathbf{f}(\mathbf{x}) = \mathbf{W}^T\mathbf{x}$ are the coordinates of the points in the linear projection subspace. The function L is to be maximized with respect to the projection matrix $\mathbf{W}$. In case parametric estimators of $\hat{p}$ are used, their parameters need to be optimized as well. Any parametric or non-parametric estimator can be used; the relative goodness of different estimators can be measured with standard methods of probabilistic model selection. The crucial thing is that the prediction is made after the projection. In this paper we use non-parametric Parzen estimators.

The basic model generates the classes c but not the $\mathbf{x}$, that is, the $\mathbf{x}$ are treated as covariates. In other words, the model predicts c based on $\mathbf{x}$. That is why we will alternatively call the model 'predictive'.

B. Optimization

To optimize the projection we need an estimator for $p(c|\mathbf{f}(\mathbf{x}))$, the conditional probabilities of auxiliary data in the projection space. Given the estimator, the likelihood (1) can then be optimized with any standard non-linear optimization algorithm; in this paper we use stochastic approximation. We will next derive the algorithm for the specific choice of a linear function $\mathbf{f}$ and a general class of estimators $\hat{p}$, including non-parametric Parzen estimators.

1) Estimation of conditional densities: In this paper we use standard Parzen estimators with Gaussian kernels for $p(c|\mathbf{f}(\mathbf{x}))$; other estimators could be used as well. Since the algorithms can be easily formulated in a way that is applicable to mixtures of Gaussians as well, we use the more general formalism here. The estimates are of the form

$$\hat{p}(c|\mathbf{f}(\mathbf{x})) = \frac{G(\mathbf{f}(\mathbf{x}), c)}{\sum_{c'} G(\mathbf{f}(\mathbf{x}, c'))} \tag{2}$$

where G is a weighted sum $G(\mathbf{f}(\mathbf{x}), c) = \sum_{m=1}^{M} \psi_{mc} g(\mathbf{f}(\mathbf{x}), m)$ of M spherical Gaussian kernels

$$g(\mathbf{f}(\mathbf{x}), m) = \frac{1}{(2\pi\sigma^2)^{d/2}} e^{(-\|\mathbf{f}(\mathbf{x})-\mathbf{f}(\mathbf{r}_m)\|^2/2\sigma^2)} . \tag{3}$$

The number of kernels M, the width σ of the Gaussians, the location parameters $\mathbf{r}_m$ and the weights ψ_{mc} are parameters to be optimized. The weights must satisfy $0 < \psi_{mc} < 1$ and $\sum_{m,c} \psi_{mc} = 1$.

Notice that the probability (2) is calculated *in the projection space*; the values of the Gaussians (3) depend only on projected coordinates $\mathbf{f}(\mathbf{x})$, and the Gaussian centers $\mathbf{f}(\mathbf{r}_m)$ are defined by projection from data space parameters $\mathbf{r}_m$.

Both Parzen-type estimators with Gaussian windows and mixtures of Gaussians can be expressed with (2). For a Parzen-type estimator the probabilities are directly based on a learning data set $\{(\mathbf{x}_i, c_i)\}_{i=1}^N$ where the $\mathbf{x}_i$ are the primary data and the c_i the auxiliary data. Parzen estimators result from setting $M = N$, weights $\psi_{mc} = \delta_{c_m,c}/N$, where $\delta_{c_m,c}$ is one if $c_m = c$ and zero otherwise, and for the locations $\mathbf{r}_m = \mathbf{x}_m$. The only free parameter is then the width σ of the Gaussians which we will optimize using a validation set.

For a mixture of Gaussians, M can be either preset or validated. The ψ_{mc} and the $\mathbf{r}_m$ are to be optimized, and σ is either optimized or validated.

An advantage of nonparametric Parzen-type estimation is that there is no need to separately re-estimate while optimizing the projection. Since the Gaussian centers are formed of projected primary data points the estimate is defined by the projection, and the optimization is capable of accounting for the changes in the estimate. While mixtures of Gaussians can be fast to compute, they need to be re-estimated when the projection changes. See Section VI for a discussion on how to combine optimization of both the density estimate and the projection. A further advantage of Parzen-type estimation is that it is a *consistent* estimator of the conditional density [13]: as the number of data points grows and σ decreases, the conditional estimate approaches the true value. The disadvantage is the long computation time for large data sets; using only a subset of data will help reduce computation time, however. In the experiments we have used Parzen-type estimation.

2) Optimization of the projection by stochastic approximation: In this paper stochastic approximation is used for optimizing the likelihood (1) of the projection $\mathbf{f}(\mathbf{x})$. Stochastic approximation is applicable to objective functions that are averages of another function. Here the average is taken of $L(\mathbf{x}, c) \equiv \log \hat{p}(c|\mathbf{f}(\mathbf{x}))$, that is, over the discrete distribution of the paired samples:

$$\frac{1}{N} L(\mathbf{W}) = \frac{1}{N} \sum_{(\mathbf{x},c)} L(\mathbf{x}, c; \mathbf{W}) \; .$$

Under certain mild assumptions [14] L can be optimized by iteratively moving towards the sample-specific gradient. At step t the update is

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \alpha(t) \frac{\partial L(\mathbf{x}, c; \mathbf{W})}{\partial \mathbf{W}} \; .$$

The step size has to fulfill the conditions $\sum \alpha(t) = \infty$ and $\sum \alpha^2(t) < \infty$. In practice the number of steps is finite and only an approximation to the optimum is obtained.

The details of deriving the gradient are given in Appendix A. The result is

$$\begin{aligned} \frac{\partial}{\partial \mathbf{W}} L(\mathbf{x}, c; \mathbf{W}) &= \big(E_{\xi(m|\mathbf{f}(\mathbf{x}))}\{(\mathbf{x} - \mathbf{r}_m)(\mathbf{x} - \mathbf{r}_m)^T\} \\ &\quad - E_{\xi(m|\mathbf{f}(\mathbf{x}),c)}\{(\mathbf{x} - \mathbf{r}_m)(\mathbf{x} - \mathbf{r}_m)^T\}\big) \frac{\mathbf{W}}{\sigma^2} \\ &= \frac{1}{\sigma^2} \big(E_{\xi(m|\mathbf{f}(\mathbf{x}))}\{(\mathbf{x} - \mathbf{r}_m)(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{r}_m))^T\} \\ &\quad - E_{\xi(m|\mathbf{f}(\mathbf{x}),c)}\{(\mathbf{x} - \mathbf{r}_m)(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{r}_m))^T\}\big) \; . \end{aligned} \tag{4}$$

This is a difference between two cross-correlation-like matrices, one conditioned on the auxiliary data and the other without it. If the auxiliary variable brings no extra information the matrices are equal and the gradient is zero. Above, the operators $E_{\xi(m|\mathbf{f}(\mathbf{x}))}$ and $E_{\xi(m|\mathbf{f}(\mathbf{x}),c)}$ denote weighted sums over mixture components m, with respective weights

$$\xi(m|\mathbf{f}(\mathbf{x})) = \frac{\sum_{c'} \psi_{mc'} g(\mathbf{f}(\mathbf{x}), m)}{\sum_k \sum_{c'} \psi_{kc'} g(\mathbf{f}(\mathbf{x}), k)} \quad \text{and} \tag{5}$$

$$\xi(m|\mathbf{f}(\mathbf{x}), c) = \frac{\psi_{mc} g(\mathbf{f}(\mathbf{x}), m)}{\sum_k \psi_{kc} g(\mathbf{f}(\mathbf{x}), k)} \; . \tag{6}$$

The weighted sums are functionally similar to expectations over conditional distributions of m. We do not, however, assume a generative model for the primary data, and the weights need not correspond to a maximum likelihood probability estimate.

For Parzen-type estimation, the stochastic samples $(\mathbf{x}, c)$ and the Gaussian components may be derived from the same dataset. We have additionally incorporated a minor improvement: if the stochastic sample has index m', exclude this index from the sums over m and k in (4), (5) and (6). This results in a kind of implicit leave-one-out validation during learning: the class prediction at sample m' is based on the other samples and their class labels, but not on the class label of m' itself. That is, m' is considered 'new data' in the prediction. Such leave-one-out prediction within the learning data set partially prevents overfitting the projection; without it, the classes of isolated samples would be 'perfectly predicted', i.e., fully overfitted. The similar adjustment does not affect the update step of the comparison method MRMI.

3) Orthogonality by reparameterization: The gradient update rules of the previous section do not yet enforce the projection matrix to remain orthonormal.

Theoretically, whether the projection is orthonormal or not has no effect on predictive power: if two projections span the same subspace, conditional probabilities after both projections converge to the same values at the large sample limit (for consistent estimators).

In practice, if the objective function were based on an overly rigid predictor, it might not predict well from a non-orthonormal projection where the data might be 'stretched' along some directions. Moreover, orthonormality is desirable to avoid unnecessary free parameters, and possible bad local minima (such as all components converging to the same value).

A straightforward way to enforce orthonormality is to reparameterize the matrix by the so-called Givens rotations. A similar reparameterization was used in [7]. In an orthonormal projection there are $(n - d)d$ rotation parameters (angles), where n is the original dimensionality and d is the projection dimensionality. This reduces the number of parameters compared to optimizing the nd elements of the projection matrix directly.

The reparameterization is

$$\mathbf{W} = \mathbf{W}_0 \left(\prod_{i=1}^{d} \left(\prod_{j=d+1}^{n} \mathbf{G}_{ij} \right) \right) \mathbf{W}_1$$

where $\mathbf{G}_{ij}$ is a rotation matrix in the ij plane by an angle λ_{ij}, that is, its elements (i,i), (i,j), (j,i) and (j,j) form a standard two-dimensional rotation matrix by

$$\begin{bmatrix} G_{ij}(i,i) & G_{ij}(i,j) \\ G_{ij}(j,i) & G_{ij}(j,j) \end{bmatrix} = \begin{bmatrix} cos(\lambda_{ij}) & sin(\lambda_{ij}) \\ -sin(\lambda_{ij}) & cos(\lambda_{ij}) \end{bmatrix}$$

and the other elements of $\mathbf{G}_{ij}$ are from an identity matrix. The matrix products are written out first term leftmost, i.e. $\prod_{j=d+1}^{n} \mathbf{G}_{ij} = \mathbf{G}_{i,d+1} \cdot \ldots \cdot \mathbf{G}_{in}$. The Givens reparameterization ensures that $\mathbf{W}$ is orthogonal at all times. The angles are initially zero, and $\mathbf{W}_0$ is an initial rotation matrix. The last matrix $\mathbf{W}_1$ simply selects the first d components after the rotations.

The gradient of a single rotation matrix $\mathbf{G}_{ij}$ with respect to λ_{ij} is a zero matrix except for elements (i,i), (i,j), (j,i) and (j,j), for which

$$\frac{\partial}{\partial \lambda_{ij}} \begin{bmatrix} G_{ij}(i,i) & G_{ij}(i,j) \\ G_{ij}(j,i) & G_{ij}(j,j) \end{bmatrix} = \begin{bmatrix} -sin(\lambda_{ij}) & cos(\lambda_{ij}) \\ -cos(\lambda_{ij}) & -sin(\lambda_{ij}) \end{bmatrix} \; .$$

For brevity, let us use a single rotation index for the angles and corresponding matrices: $\mathbf{G}_m = \mathbf{G}_{ij}$ and $\lambda_m = \lambda_{ij}$, where $m = (i-1)(n-d) + j - d$. With this notation, the (stochastic) gradient of a rotation angle can be shown to be

$$\begin{aligned}\frac{\partial}{\partial \lambda_m} L(\mathbf{x}, c; \mathbf{W}) &= \sum_{k,l} \frac{\partial L}{\partial w_{kl}} \frac{\partial w_{kl}}{\partial \lambda_m} \\ &= \sum_{k,l} \left[\frac{\partial}{\partial \mathbf{W}} L(\mathbf{x}, c; \mathbf{W}) \right]_{kl} \\ &\cdot \left[\mathbf{W}_0 \left(\prod_{i=1}^{m-1} \mathbf{G}_i \right) \left(\frac{\partial \mathbf{G}_m}{\partial \lambda_m} \right) \left(\prod_{i=m+1}^{d(n-d)} \mathbf{G}_i \right) \mathbf{W}_1 \right]_{kl} \qquad (7)\end{aligned}$$

where $[\ldots]_{kl}$ denotes the element (k, l) of the matrix inside the brackets.

A simple alternative to Givens rotations is to simply optimize the elements of the projection matrix directly, and orthonormalize the projection after each update. This technique has been used in [8]. However, this method does not reduce the number of parameters to optimize.

Note that the projection can be easily extended by adding scaling after rotation: simply replace $\mathbf{W}_1$ by $\mathbf{W}_1\mathbf{A}$ where the diagonal matrix $\mathbf{A}$ contains the scaling parameters. This leads to $(n - d + 1)d$ parameters to optimize, still less than nd when $d > 1$.

As discussed above, non-orthonormality like scaling does not affect predictive power. However, it can affect practical optimization with the Parzen estimator, since scaling effectively changes its resolution in different directions. In some situations this might help but the extra parameters might complicate optimization. We did not use scaling in the experiments (scaling parameters were set to 1). Instead, the resolution of the Parzen estimator was controlled with the σ parameter; in the empirical tests (Section V), this was sufficient to yield good results.

4) The algorithm: The parameters of the method can be simply optimized with standard gradient methods, such as conjugate gradient or stochastic gradient. Here we use the latter method. We present its details here for completeness.

The stochastic approximation update step is as follows: at iteration t, pick a sample $(\mathbf{x}, c)$, and adjust the rotation angles by

$$\lambda_m(t+1) = \lambda_m(t) + \alpha(t) \frac{\partial}{\partial \lambda_m} L(\mathbf{x}, c; \mathbf{W}) \,. \qquad (8)$$

We used piecewise linear schedules for the $\alpha(t)$.

The on-line algorithm for optimizing the projection is summarized in Fig. 1. Steps 3(b) and 3(c) and Equation (4) are the ‘core’ of the algorithm; the more complicated equation (7) simply accounts for the reparameterization. The time complexity of each iteration is $\mathcal{O}(N)$ with respect to the number of samples N.

5) Initialization: The algorithm is based on (stochastic) gradient optimization, and hence may find only a local optimum, depending on the initialization (and the random presentation order of data). The Renyi-based comparison method in Section III also has this problem. Multiple restarts from different initializations could be used to avoid local maxima. In this paper, we used a simple alternative: we initialized the first components of $\mathbf{W}_0$ by orthonormalizing an LDA projection (LDA does not give a full basis; the remaining components were filled with orthonormal vectors).

```
1) Choose the initial rotation $\mathbf{W}_0$, for example by
   orthonormalizing an LDA projection.
2) Choose the width $\sigma$ for the Gaussians (ultimately
   with a validation set), and a schedule (piecewise-
   linear decreasing function) for the learning rate.
3) Repeat the following steps (for a set number of
   iterations):
    a) Sample an input $(\mathbf{x}, c)$ from the data.
    b) Compute component weights $\xi(m|\mathbf{x})$ by (5) and
       $\xi(m|\mathbf{x}, c)$ by (6).
    c) Compute the stochastic gradient $\frac{\partial}{\partial \mathbf{W}} L(\mathbf{x}, c; \mathbf{W})$
       for the projection matrix by (4).
    d) Compute the gradients for the Givens rotation
       angles by (7).
    e) Adjust the angles by (8).
```

Fig. 1. Algorithm for optimizing the cost function (1) in Section II-A with Parzen estimators or other Gaussian mixture-based probability estimators.

In contrast, LDA and PCA find the global optima of their cost functions. Nevertheless, in the empirical tests in Section V the proposed method with the on-line algorithm achieves better results.

III. Comparison method: transformation that maximizes Renyi entropy-based mutual information

Torkkola and Campbell [7] have introduced a method, denoted here by MRMI for Maximization of Renyi Mutual Information, that is closely related to ours. Both methods search for a linear projection that maximizes mutual information between the features and class labels. The main difference is that instead of Shannon entropy, Torkkola and Campbell use Renyi quadratic entropy in defining the mutual information. The work is based on earlier ideas about Renyi entropy-based feature extraction [9], [10].

The second difference is in the estimation of the projection. We define the model for finite data as a generative (conditional) probability density model, which makes it possible to rigorously use the machinery of probabilistic inference. The connection to mutual information is asymptotic, which in our opinion is natural since mutual information is defined in terms of the (unknown) distributions.

By contrast, in MRMI an estimator of the projected joint density of the data is constructed, and the estimated mutual information of the projection and the class distribution is maximized. Parzen and Gaussian mixture estimates were used in [15]. The possible theoretical problem with this approach seems to be that the cost function used for estimating the density is not directly related to the overall modeling goal, that is, maximization of mutual information. For Parzen estimators this problem does not occur, however.

In the experiments of Section V, our method is compared to other projection methods including MRMI. We optimized both the proposed method and MRMI in the same way to make sure differences in optimization algorithms do not cause differences in results. Parzen estimators are used to estimate the densities, and the projection is parameterized by Givens rotations. For MRMI, this leads to an algorithm similar to the one in Fig. 1. The only difference is that the gradient $\frac{\partial}{\partial \mathbf{W}} L(\mathbf{x}, c; \mathbf{W})$ in steps 3(b) and 3(c) of the algorithm will be replaced by the following gradient (see Appendix B for details):

$$\frac{\partial}{\partial \mathbf{W}} L_{MRMI}(\mathbf{x}, c; \mathbf{W}) = \frac{-1}{\sigma^2 N} \sum_{k=1}^{N} G(\mathbf{f}(\mathbf{x})|\mathbf{f}(\mathbf{x}_k), \sigma^2 \mathbf{I}) \cdot \left[\delta_{c,c_k} + \sum_{c'} \hat{p}(c')^2 - 2\hat{p}(c) \right] (\mathbf{x} - \mathbf{x}_k)(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x}_k))^T . \tag{9}$$

Here $(\mathbf{x}_k, c_k)$ are the data samples used in the Parzen estimator, $\hat{p}(c) = \frac{1}{N} \sum_{k=1}^{N} \delta_{c,c_k}$, and $G(\mathbf{f}(\mathbf{x})|\mathbf{f}(\mathbf{x}_k), \sigma^2 \mathbf{I})$ is the value at $\mathbf{f}(\mathbf{x})$ of a d-dimensional Gaussian distribution with mean $\mathbf{f}(\mathbf{x}_k)$ and covariance matrix $\sigma^2 \mathbf{I}$. The other quantities are defined as in Section II-B. Like the algorithm for the proposed method, this algorithm for MRMI also has a time complexity of $\mathcal{O}(N)$ per iteration with respect to the number of samples N.

Note that all terms in the gradient (9) are directly proportional to absolute values of Gaussian functions. These can get very low values if the sample $\mathbf{x}$ is projected far from any neighboring points, especially when the projection subspace is high-dimensional. Therefore, the (initial) learning rates α and values of σ need to be chosen carefully to ensure meaningful parameter updates. To ensure this, in the experiments the σ and the initial learning rate were validated from a wide range of values; see Section V-C for details.

IV. Relationships to other methods

When described as a predictive model of the class distribution the proposed model is extremely simple, which we consider to be one of its main assets. It is relatively straightforward to apply the machinery of probabilistic inference to it. The usefulness of the method, however, comes from its relationships to other theories and models which suggest ways of using it. These connections will be presented in this section: Relationships to the theory of maximization of mutual information and learning metrics, interpretation as generalized linear discriminant analysis, and relationships to a few other component models.

A. Regression

Regression methods aim to model the variation in a regressed function with a specified group of regressors. The proposed method can be viewed from a regression point of view as well. However, in this context it is an unconventional solution since we are interested in finding components of data, not merely predicting. The likelihood cost function is also unconventional for standard regression, compared to for instance squared error minimization. Many regression methods such as projection pursuit regression and two-layer neural networks make restrictive assumptions about the relation between the regressors and the regressed function. By contrast, we use a nonparametric estimator which leads to less restrictions.

Several linear methods such as canonical correlation analysis (CCA) and partial least squares (PLS) have been used in this task. When the values to be regressed are classes, the most appropriate of these linear methods is LDA.

Recent regression methods include sliced inverse regression (SIR; [16]), principal Hessian directions (pHd; [17]) and sliced average variance estimation (SAVE; [18]). When classes are used for the regressed function, SIR is effectively equivalent to linear discriminant analysis LDA, except for predictor scaling [19]. In Section V we compare the proposed method with other methods including LDA. For pHd the regressed function must be one-dimensional, and we are not aware of multivariate pHd extensions. Therefore pHd is not suitable for cases where there are over two (unordered) classes. SAVE has been included in the comparisons. It effectively searches for the subspace corresponding to quadratic discriminant analysis (QDA) [19]. Its problem is that it may miss linear trends [19], [20].

B. Maximization of mutual information

It is straightforward to show that the objective function (1) has an asymptotic connection to the mutual information. This connection is not specific to the present model and is probably already known. As the amount of data N increases (here $\mathbf{y} = \mathbf{f}(\mathbf{x})$),

$$\begin{aligned} \frac{1}{N} L \underset{N \to \infty}{\longrightarrow} & \sum_c \int p(c, \mathbf{x}) \log \hat{p}(c|\mathbf{f}(\mathbf{x})) d\mathbf{x} \\ &= \sum_c \int p(c, \mathbf{y}) \log \frac{p(c, \mathbf{y})}{p(c) p(\mathbf{y})} d\mathbf{y} \\ &\quad - \sum_c \int p(c, \mathbf{y}) \log \frac{p(c|\mathbf{y})}{\hat{p}(c|\mathbf{y})} d\mathbf{y} - H(C) \\ &= I(C, Y) - E_{p(\mathbf{y})} [D_{KL}(p(c|\mathbf{y}), \hat{p}(c|\mathbf{y}))] - H(C) . \end{aligned} \tag{10}$$

(The constant $1/N$ on the first line has no effect on optimization.) The first term on the last line is the (true) mutual information between the auxiliary variable C having the values c and the projected primary variable Y having the values $\mathbf{y}$. The second term is the average estimation error of the auxiliary data after the projection. The term $H(C)$, the entropy of C, is constant.

Hence, maximization of the objective function of the generative model (1) is asymptotically equivalent to maximizing the mutual information and simultaneously minimizing the estimation error. The estimation error vanishes asymptotically for consistent estimators such as the Parzen estimators. For non-consistent estimators straightforward maximization of mutual information would neglect the estimation error.

The implication is that for large data sets and consistent density estimators the proposed model follows the tradition of maximizing mutual information [3]–[5]. For small data sets the infinite-data formalisms do not ensure good generalization, however, whereas the likelihood formulation makes it possible to apply all the machinery of probabilistic inference.

C. Linear discriminant analysis and canonical correlation analysis

In classical linear discriminant analysis (LDA [1]; see e.g. [2]), each class is assumed multinormally distributed with the same covariance matrix in each class. For a two-class problem the direction in the data space that maximizes the within-class variance while minimizing the between-class variance is sought. The solution can be found by estimating the within- and between-class covariance matrices, and it is asymptotically optimal for classification if the assumptions hold.

The solution can be generalized to multiple classes, by still maximizing the between-class variance and minimizing within-class variance. The first 'canonical component' corresponds to the direction in which the multiple correlation with the groups is the largest. The second component has the next largest correlation while being uncorrelated with the first, and so on. This method is sometimes called multiple discriminant analysis or canonical discriminant analysis. When the assumptions of LDA hold, only the $N_c - 1$ first components help in discriminating the classes, where N_c is the total number of classes.

There is a direct connection between LDA and canonical correlations, a method that has been shown to maximize mutual information for multinormal data. In canonical correlation analysis there are two multivariate samples and pairs of directions, one of each pair in each space, are sought. The goal is to maximize the correlation between the projections. The first pair maximizes it, the second maximizes the correlation constrained to being uncorrelated with the first, and so on. It can be shown that linear discriminant analysis is equivalent to canonical correlation analysis if the class variable is encoded in the 1-out-of-N_c manner, that is, each class has its own dimension, and the value of the dimension is 1 if the sample belongs to the class, and 0 otherwise [21].

Canonical correlations have a close relationship to mutual information. If both paired samples are multinormally distributed—actually even elliptical symmetry suffices—canonical correlation analysis maximizes the mutual information between the paired variables [22].

Although canonical correlations are closely related to linear discriminant analysis, to our knowledge no general relationship between LDA and mutual information is known.

In summary, LDA is a well-established way of finding directions that best discriminate multinormally distributed classes. In this paper we discard the parametric assumptions about the distribution of the classes, and estimate the densities nonparametrically. The second advantage is that density estimation in the reduced-dimensional space suffices.

Relationship to the proposed method. If the classes are multinormally distributed, our method finds the same projection as LDA, at least under two additional assumptions. In addition to the assumption that all classes have the same covariance matrix, it is required that (i) the class centers reside within a d-dimensional subspace of the original space, if a d-dimensional projection is sought, and (ii) there is enough data, i.e., the result is asymptotical.

The proof is simple with these assumptions; a sketch is presented here. It is known (see for example [23]) that LDA is equivalent to maximizing the likelihood of a joint density model for the data and the classes, in the original primary data space. Each class is modeled by a separate Gaussian density. It is then straightforward to show that the conditional class density $p(c|\mathbf{x})$ of the optimal LDA model (and asymptotically of the data as well) is constant in all directions orthogonal to the d-dimensional subspace containing the class centers. This can be seen by factoring the density into two terms; the first term depends only on the important d dimensions and the second only on the other dimensions. Our method, by comparison, builds a model $\hat{p}(c|\mathbf{f}(\mathbf{x}))$ for the conditional distribution that only varies within d dimensions, and the optimal solution clearly is to match them to the dimensions where the densities really vary. The correct solution is reached if the probability estimator $\hat{p}(c|\mathbf{f}(\mathbf{x}))$ is asymptotically capable of finding the true distribution within the projection space, which holds at least for the nonparametric estimator we have used.

Incidentally, since our method asymptotically maximizes mutual information, the proof implies that classical LDA maximizes it as well under the restrictive assumptions above. Note that the proposed new method does not need the assumptions.

Furthermore, this means canonical correlations maximize mutual information also in the non-multinormal case, with 1-out-of-N_c class encoding and the assumptions above.

Several generalizations of LDA have been proposed, including heteroscedastic discriminant analysis (see, e.g., [24], [25]) in which the classes are allowed to have different covariance matrices. There do not exist as close connections between our proposed method and these generalizations. The key difference, in our opinion, is that whereas LDA-based methods make parametric assumptions on the *joint density* of the data and the classes, the proposed method only requires a (nonparametric) estimate of the conditional density.

D. Principal component analysis

Principal component analysis (PCA; see, e.g., [2]) is discussed here because it has a close connection to the proposed method. PCA searches for a linear transformation $\mathbf{y} = \mathbf{W}^T\mathbf{x}$ to a lower-dimensional space, such that the average Euclidean reconstruction error

$$\int d_E^2(\mathbf{x}, \mathbf{W}\mathbf{W}^T\mathbf{x})p(\mathbf{x})d\mathbf{x} \tag{11}$$

resulting from representing data in the projection space is minimized. Here d_E^2 is the squared distance between the reconstruction $\mathbf{W}\mathbf{W}^T\mathbf{x}$ and the original point $\mathbf{x}$ and $\mathbf{W}$ is the $n \times d$ matrix consisting of orthonormal basis vectors. The cost is minimized by calculating the covariance matrix of data and choosing the basis vectors to be its eigenvectors corresponding to the largest eigenvalues.

E. Learning metrics

The learning metrics principle ([11], [12], [26]; see also [27], [28]) suggests using information-geometric [29], [30] methods to learn metrics to a *data space*. Information geometry defines so-called Fisher metrics to the *parameter space* of a generative probabilistic model. When learning metrics, the coordinates of the primary data space are regarded as parameters of a generative model that predicts the distribution of auxiliary data; in this paper the classes are the auxiliary data.

The class labels then implicitly define what is important, and the analysis is focused on the important aspects. For instance, if an input variable has no effect on the class distribution, it does not affect the distance in the new metric.

Distances d_L in the learning metric are defined in terms of the conditional distributions of auxiliary data: local distances are Kullback-Leibler divergences D_{KL} between the distributions, expressible as quadratic forms with the Fisher information matrix $\mathbf{J}(\mathbf{x})$,

$$\begin{aligned} d_L^2(\mathbf{x}, \mathbf{x} + d\mathbf{x}) &\equiv D_{KL}(p(c|\mathbf{x}), p(c|\mathbf{x} + d\mathbf{x})) \\ &= d\mathbf{x}^T\mathbf{J}(\mathbf{x})d\mathbf{x} \end{aligned} \tag{12}$$

where

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

and global distances are minimal path integrals of the local distances.

The learning metrics principle is defined in (12) in terms of known distributions $p(c|\mathbf{x})$. In practical data analysis there is seldom enough data to estimate the distributions accurately and the principled way of applying learning metrics, taken in this paper as well, is slightly more complicated. We define a generative model that can be optimized with well-defined machinery of probabilistic inference. We then show that the model asymptotically has a connection to the learning metrics principle.

The connection to learning metrics here is that the proposed method has an (asymptotic) interpretation as a component model, with the cost function expressed in learning metrics. The cost function is analogous to the cost of principal components analysis, discussed in Section IV-D.

The cost is the average reconstruction error. The reconstruction $\mathbf{r}(\mathbf{f}(\mathbf{x}))$ of $\mathbf{f}(\mathbf{x})$ is defined as a point in the primary data space that projects to $\mathbf{f}(\mathbf{x})$ and where the class distribution best matches that of the projection, by the Kullback-Leibler divergence. The reconstruction error is measured between it and the original data $\mathbf{x}$. The reconstruction is defined only to provide an interpretation of the method; in practice data can be projected without knowing the reconstructions.

To be precise the connection is that the mutual information in (10) equals (with certain approximations; see Appendix C)

$$\begin{aligned} I(C, Y) &= \sum_c \int p(c, \mathbf{y}) \log p(c|\mathbf{y})d\mathbf{y} + \text{const.} \\ &\approx -\int p(\mathbf{x})d_L^2(\mathbf{x}, \mathbf{r}(\mathbf{f}(\mathbf{x})))d\mathbf{x} + \text{const.} \end{aligned}$$

where the right-hand side is an average squared distance d_L from samples $\mathbf{x}$ to their reconstructions $\mathbf{r}(\mathbf{f}(\mathbf{x}))$. There are two differences from the PCA cost function (11). The main difference is that the error is measured in the so-called learning metrics. An additional difference is that the reconstruction is defined implicitly as a kind of missing value problem: the coordinates orthogonal to the projection subspace are reconstructed by finding the projected point with the smallest Kullback-Leibler divergence between the auxiliary distributions at the point and the projection.

Another difference is that PCA components can be computed one by one using an on-line algorithm. By contrast, our method searches for the whole set of components at the same time, and the components change if their number is changed.

For good enough (that is, consistent) estimators $\hat{p}$ the second term in (10) asymptotically vanishes. Hence, *optimizing the proposed cost function (1) is asymptotically approximately equivalent to minimizing the reconstruction error of principal components analysis, measured in learning metrics.*

V. Experiments

In this section we compare the proposed new projection method to the two other most closely related linear projections, the Renyi entropy-based MRMI and linear discriminant analysis. PCA is also included to provide a baseline; unlike the other methods, PCA does not use auxiliary data.

The aim of the experiments is threefold: to measure quantitatively how well the new method performs; to compare the projections qualitatively; and to demonstrate how to use the components in practical data analysis. Quantitative comparison of visualization and exploratory capability of methods is very hard and we have to resort

TABLE I
STATISTICS OF THE TEST DATASETS.

Dataset	Dimensionality	Number of classes	Number of samples
Landsat	36	6	4435
LVQ_PAK	20	13	3656
Isolet	30	26	3742
MFeat	76	10	1500
TIMIT	12	41	14994

to indirect measures; we will make sure, however, that the comparisons between the proposed method and the Renyi-based alternative are not biased.

A. Data

We compared the methods on five real-world datasets whose properties are summarized in Table I. The Landsat, Isolet, and Multiple Features (MFeat) data sets are from UCI Machine Learning Repository [31], LVQ_PAK refers to the Finnish acoustic phoneme data distributed with the LVQ-PAK [32], and TIMIT refers to phoneme data from the Darpa TIMIT acoustic phoneme database [33].

The datasets were first preprocessed. The dimensionality of the Isolet data was reduced to 30 by a PCA projection to reduce the computational load of the proposed algorithm and the MRMI. The Multiple Features data contained several different types of features, of which Fourier coefficients were selected. The LVQ_PAK data were used as is.

For four of the datasets, we sought a projection to 5 dimensions. As an exception, a three-dimensional projection was sought for the Landsat dataset since it contains only six classes. In general, the projection dimensionality should be chosen large enough to make interesting findings possible, and yet small enough to give understandable results. Here we chose (arbitrarily) a 5-dimensional projection; the effect of the dimensionality on the results will be studied in Fig. 3, and methods for choosing 'optimal' dimensionality will be studied later.

B. Quality measure

A fair performance measure is needed to compare the methods. It cannot of course be the objective function of any of the methods. Since all aim to be discriminative we chose the classification error, measured for the simple non-parametric *K nearest neighbor* (KNN) classifier, working in the projected space. Since both the proposed method and the MRMI comparison use non-parametric kernel-based estimates, the measure does not favor either method. Note that minimization of the classification error is not the primary goal of any of the methods; the results give only indirect evidence.

To be precise, results were evaluated by projecting the test set and calculating a KNN classification for each test sample from $K = 5$ neighbors selected from the projected learning set. The classifier predicted the class having a majority within the neighbors. The resulting classification error rates of all models were then compared.

Ties were broken by assigning *equal portions* of the tied sample to all tied classes. If the sample belongs to one of the tied classes, this yields 'partial' classification error, for instance 4/5 if there were 5 tied classes. If the sample is from none of the tied classes, a 'full' classification error of 1 occurs. The result equals the expected classification error when the class is picked randomly from the tied classes.

Note that in principle the internal density estimates of the proposed method and MRMI could be used for classification; however, such nonlinear classifiers cannot be derived from PCA and LDA. Using a single classification method ensures that performance differences result from the quality of the projection, not of the classifier.

C. Experimental set-up

The experimental set-up for the quantitative comparison required three steps. Ultimately, we compared the methods by cross-validation experiments on the 5 data sets. In order to reduce the computation time we chose the width of the Gaussians σ beforehand, in preliminary experiments, using a validation set. The same was done for the initial learning rate α. Of the four comparison methods, the new algorithm and MRMI have these parameters; LDA and PCA have no adjustable parameters.

It turned out that MRMI is very sensitive to the choice of these parameters. Hence, to make sure that we did not give unjust advantage to our method, we had to start by first choosing a proper *range* for the parameter values. The best value from this range was then chosen using the validation set.

In all of the experiments, both our method and MRMI were computed for 20,000 stochastic approximation steps, starting from an LDA initialization. During the computation, the learning rate was decreased piecewise linearly to zero.

1) Choosing the range of parameter values for validation: Each database was initially divided into a training set and a validation set (of about a third of the data) that were used to choose the range for the parameters.

Again, to save computation time, the range for the initial learning rate α was chosen based on only one of the data sets, the LVQ_PAK data. Each potential value was tried with a range of σ values, and the minimum α-specific classification error on the validation set was sought. New (smaller or larger) α values were then tried until a local minimum of classification error was found. The resulting set of α values was then used in all the subsequent experiments, for all data sets.

For our method, a less elaborate scheme was used: the range of α values was simply hand-picked for LVQ_PAK data instead of validated.

For each data set, the range of σ values was logarithmic, starting from roughly the root mean squared distance from a point to its nearest neighbor, and extending to the average distance to the furthest neighbor. For some data sets more (larger or smaller) values were added approximately logarithmically, in case a local maximum was not found within the default set.

2) Choosing parameter values by validation: When the ranges of α and σ had been validated, the values within the range that gave the best values for the validation set were chosen.

3) Cross-validation tests: Given the values of α and σ, the methods were then compared with a cross-validation test.

The data was re-divided into ten sets, and in each fold of cross-validation one of the ten was used as test data and the other nine as learning data. The projection matrix was optimized with the learning data, and the performance was measured on the respective test set.

D. Quantitative comparison

The statistical significance of the difference between the best and the second best method was evaluated for each data set by a t-test of the 10-fold cross-validation results. Table II shows the average performances across the folds, and for completeness the p-values of the differences between all methods.

Our method achieved the best average result for four of the five data sets. The difference between our method and the second best method was significant for three of the sets.

For Landsat data, our method was surprisingly bad, possibly due to overlearning resulting from a poorly chosen parameter value. MRMI was the best here but the difference to the second best (LDA) is not significant. For the LVQ_PAK data set, our method was significantly better than the second best (PCA), and the rest had similar performance. For Isolet data our method achieved significantly better results than the second best (LDA). LDA was closely followed by MRMI and PCA had the worst results. For the Multiple Features (MFeat) data set, our method was significantly better than PCA. The others did not seem to outperform PCA. For TIMIT data, our method was the best but the difference was not significant.

SAVE was the worst on all of the five data sets: the average error was 68.2 percent on Landsat, 18.5 on LVQ_PAK, 73.6 on Isolet, 62.0 on MFeat and 62.8 on Timit. The p-values of t-tests of the difference to other methods were less than 0.01 on all sets.

Our hypothesis for the poor Landsat performance of our method is that the σ and α parameters were badly chosen: their good performance in the parameter validation stage was likely due to random variation in the stochastic optimization. We tested this hypothesis by re-running the cross-validation with the second-best values found in the parameter validation (larger σ and α). This decreased the error rate to 13.72 percent, and differences to MRMI, LDA and PCA became non-significant. Reducing α further improved the cross-validation performance to 12.62 percent—the best result for Landsat.

TABLE II

Difference of performance of the methods. The leftmost figures in each table are average classification error rates over the ten folds, in percentages. The methods are listed in the order of increasing average error rate, and the best method is shown in boldface. The figures in the last three columns of each table are p-values for a two-tailed t-test that the difference between the methods on the row and the column is nonzero.

Landsat

	Error	LDA	PCA	New
MRMI	13.34	0.49	0.05	0.04
LDA	13.62	-	0.46	0.06
PCA	13.96		-	0.27
New	14.70			-

LVQ_PAK

	Error	PCA	MRMI	LDA
New	8.51	0.04	$< 10^{-4}$	$< 10^{-3}$
PCA	9.60	-	0.20	0.09
MRMI	10.25		-	0.33
LDA	10.51			-

Isolet

	Error	LDA	MRMI	PCA
New	17.74	$< 10^{-4}$	$< 10^{-5}$	$< 10^{-7}$
LDA	28.79	-	0.56	$< 10^{-6}$
MRMI	29.44		-	$< 10^{-6}$
PCA	40.15			-

MFeat

	Error	PCA	MRMI	LDA
New	17.06	0.02	$< 10^{-2}$	$< 10^{-2}$
PCA	19.60	-	0.37	0.05
MRMI	20.89		-	0.90
LDA	21.08			-

TIMIT

	Error	MRMI	LDA	PCA
New	59.58	0.95	0.93	$< 10^{-6}$
MRMI	59.59	-	0.99	$< 10^{-6}$
LDA	59.60		-	$< 10^{-7}$
PCA	64.10			-

E. Qualitative comparison

For the purpose of easy visual comparison, we computed two-dimensional projections of the MFeat data. The projections that gave best performance on the validation set were selected.

The classification error rates were 28.20% for our method, 32.88% for MRMI, 34.76% for LDA, and 34.88% for PCA.

Fig. 2 shows scatterplots of the (out-of-sample) validation data. The MFeat data are Fourier coefficients of handwritten numerals.

It is clearly visible in the projections that LDA and PCA both separate the classes 0, 8, 5, and 2 well from the other numerals. The others are overlapping, grouped together at the bottom of the projection images. LDA has kept the classes more tightly clustered than PCA.

The new method and MRMI achieve the best results. The main difference is that MRMI lets numerals 6 and 9 overlap 3 and 4, while the new method separates them.

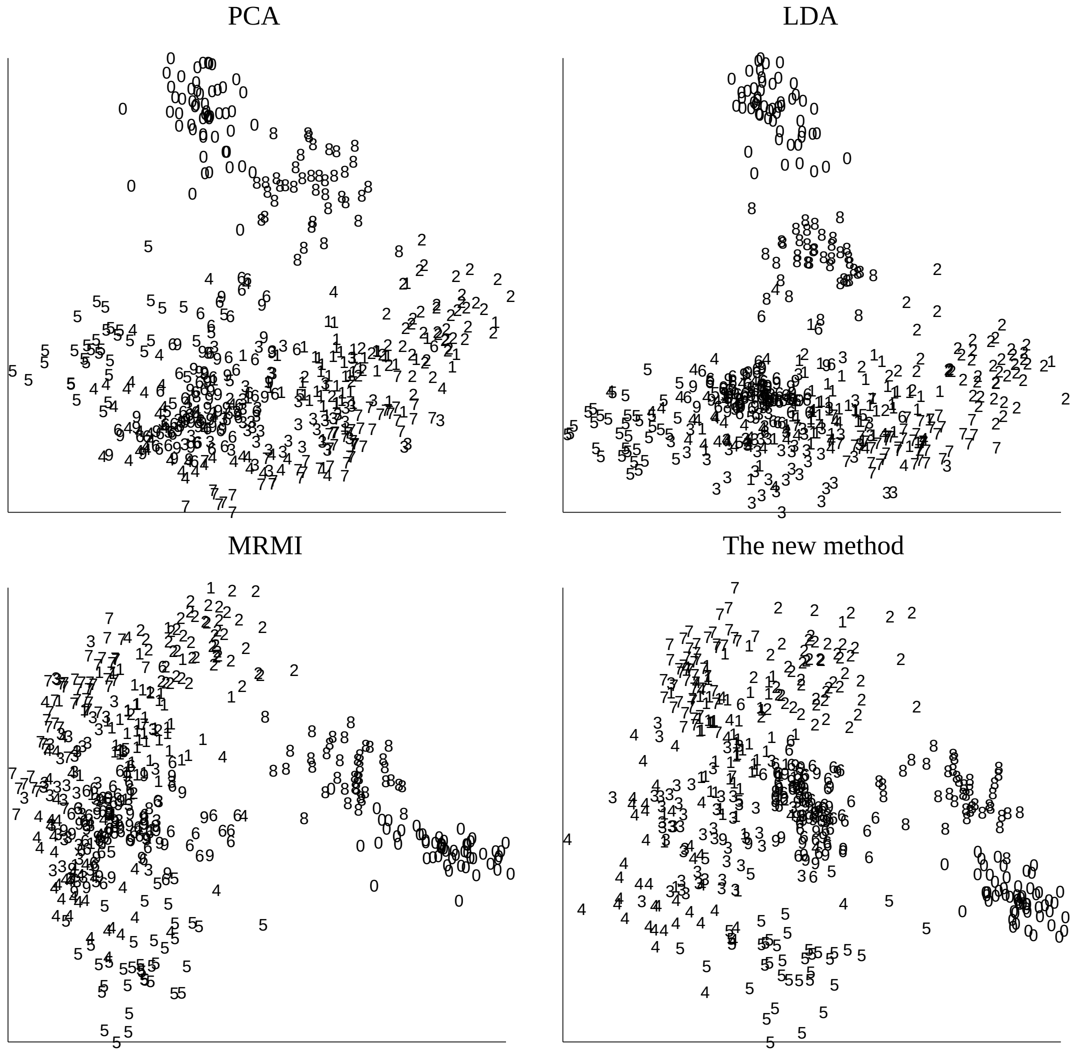

Fig. 2. Two-dimensional projections of MFeat leave-out data containing 10 classes of handwritten numerals. PCA=principal components analysis, LDA=linear discriminant analysis, MRMI=maximization of Renyi mutual information.

The new method also separates 5 slightly better from 4. The LDA initialization seems to be still visible in the MRMI result, while the new method has discovered a projection clearly differing from LDA.

F. Complexity control

The complexity of the predictions depends on two main factors: the kernel width parameter σ and the number of components (projection dimensionality d).

1) Kernel width: In our experiments the parameter σ has been chosen by preliminary experiments to maximize performance on a preliminary division of data, as described in Section V-C. In general, low values of σ increase the resolution of the Parzen windowing; in the limit $\sigma \to 0$ the predictor predicts the class of the nearest neighbour after projection. High values soften the prediction over an increasing neighbourhood, and in the limit $\sigma \to \infty$ the overall class proportions are predicted. Since distances between samples tend to increase with dimensionality, larger σ may be required for high-dimensional projections.

2) Dimensionality: In general choosing the dimensionality is a difficult issue where for instance Bayesian methods may help. We do not study such extensions here but measure the results as a function of dimensionality.

We computed projections of increasing dimensionality (from 2 to 25) for the Isolet data set by the proposed method. As a comparison we also computed LDA projections; LDA cannot find more than 25 components here since there are 26 classes.

The goodness of the projections was again evaluated by the indirect measure in Section V-B, that is, classification error for test data, classified based on training data with KNN after the projection.

To preserve computation time, we did not use cross-validation here. Instead, the projections were computed on a single training set, and their goodness was evaluated on a separate test set.

For the proposed method, the σ and α parameters were first validated for each dimensionality, by dividing the training set into learning and validation subsets. The

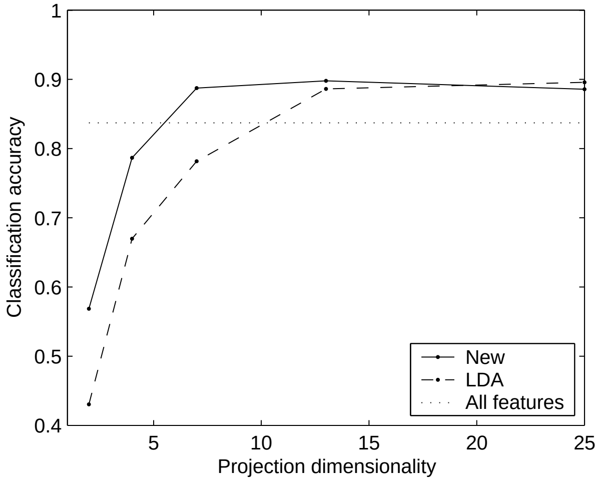

Fig. 3. Classification performance (rate of correct classification) in projection spaces of varying dimensionality for leave-out Isolet data. Solid line: proposed method, dashed line: LDA, dotted line: all features (no projection).

parameters were chosen to maximize performance on the validation subset; to avoid excess noise in the validation, a moving average filter was applied to the results matrix along the σ axis. The final projections were then computed on the whole training set.

The results are shown in Fig. 3. Increasing the dimensionality improves the performance of both methods. The proposed method consistently outperforms LDA at low dimensionalities; for high dimensionalities the results converge. The performance seems to saturate after dimensionality 7 for the proposed method and 13 for LDA. Note that at high dimensionalities both methods yield better classification than using all features; this suggests that there may be irrelevant directions in the original data.

A note about ordering of components: The proposed method does not give an ordering of the components that it finds, or a successive embedding of the projection subspaces. The lack of ordering is a property of the task, not the method: there exist data sets where the successive projection subspaces may even be orthogonal![1]

In many practical cases an ordering may exist; then one can extract the components by successively finding lower-dimensional subspaces, with the previous projected data as input. In each projection, some directions are left out of the subspace. The directions left out first are the least important.

G. Data analysis demonstration

In this section we demonstrate one way of using the extracted components for exploratory data analysis. As an example we will explore the LVQ_PAK data.

The LVQ_PAK data consists of features of short-time samples of spoken phonemes. The primary data are the feature values and the auxiliary data the phonemes (classes). We know how the features have been computed and what they represent, but we will not use this knowledge in the analysis. This prior knowledge will be used only after the data-driven component analysis to verify and interpret the findings.

The goals of the analysis are (i) to visualize similarities of the classes, and (ii) to perform *feature exploration* to discover how the features differentiate between phonemes.

We chose again the projection dimension of 5, and of the projections computed in Section V-C we chose the one that gave the best validation result in the parameter selection phase.

1) Visualization of classes: Scatterplots are the simplest means of visualization. Fig. 4 shows three scatterplots for different pairs of the 5 components.

It is clear based on the visualizations that similar phonemes are usually close-by in the projection space: for example, A is adjacent to Ä, O is adjacent to U, and M is adjacent to N. Moreover, the 2nd component (x-axis in the lower figures) arranges vowel sounds in the order A, Ä, O, U, E, I, where both the front vowels Ä, E, I and the back vowels A, O, U are arranged in the order of height (defined roughly as the pitch of the first formant).

2) Feature exploration to characterize the classes: Further insight is gained by investigating how particular classes are differentiated in the projection. For example, the second component suffices to separate A from the other phonemes. By contrast, I does not seem to be well separated by any single component. For instance, along component 2 it overlaps with S and along component 1 with the other vowels (see the topmost projection in Fig. 4). The best single projection for I is perhaps the plane formed of components 1 and 2.

The components can be interpreted in terms of the 'contributions' of the original variables in them, assuming the original variables have intuitive interpretations. A simple way of interpreting the components is to list which primary variables have the largest (absolute) component values in them.

Luckily, in this case study we actually do know that the features are so-called cepstral components of the original speech waveforms, and we can study the meaning of the projection directions in terms of sound spectra. For example, we can study what spectral characteristics the projections use to discriminate between A and S. We stress that this information is not used by the method, and this additional study is included merely to gain insight into what the method discovers.

In the LVQ_PAK data, the original variables are *MEL-frequency cepstral components*, corresponding to linear filters of the logarithmic power spectrum of a sound. The feature extraction process is described in [34].

Since the projection is a linear operation, each projection direction can be represented as a single linear filter of the logarithmic power spectrum. Fig. 5 shows the shape of the combined filter for each component. In other words, the subfigures show what kinds of spectral characteristics

[1] For example, in a toy example where a torus (class 1) is wrapped around the middle of a cylinder (class 2), the correct one- and two-dimensional solutions are orthogonal to each other.

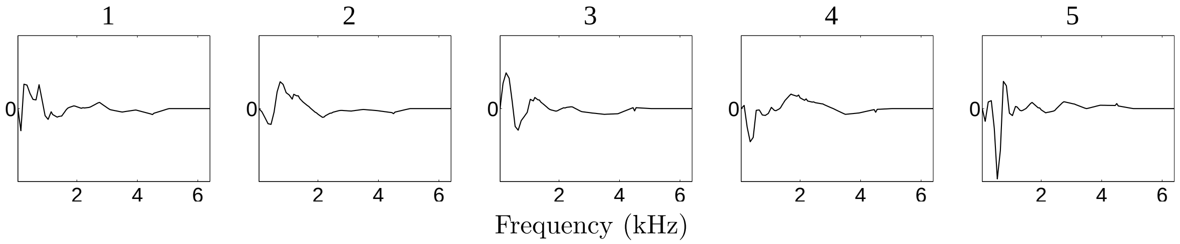

Fig. 5. Response sensitivity (weighted filter) of each component found for LVQ_PAK data. The value of the component is the inner product of the speech spectrum with this filter. X-axes: frequency, Y-axes: filter value.

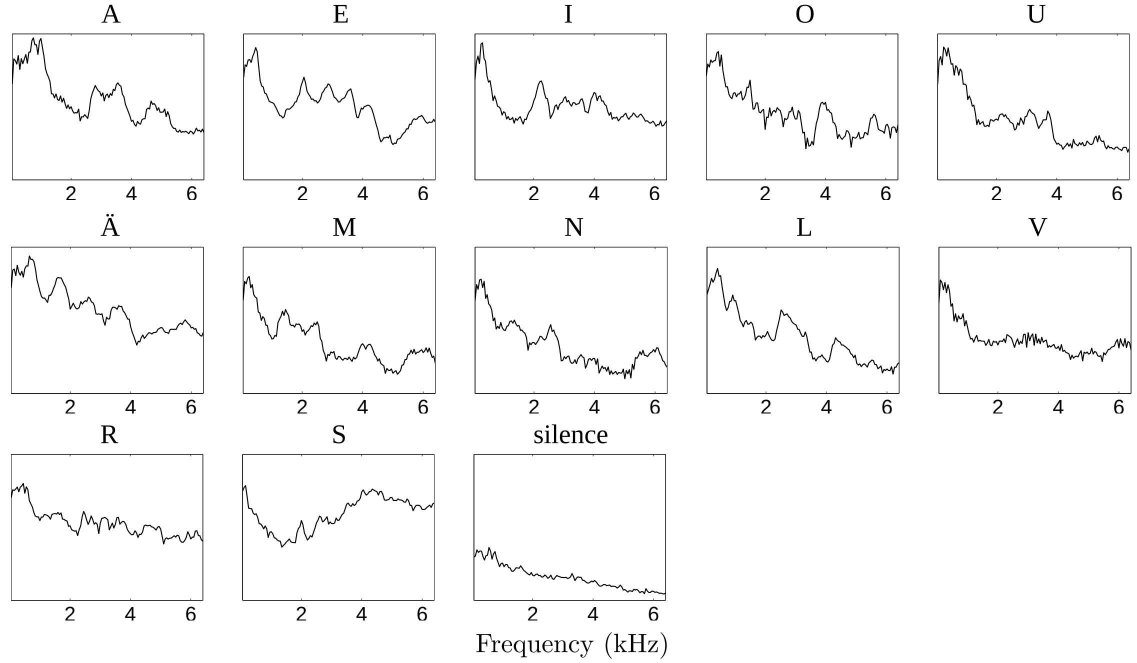

Fig. 6. Sample logarithmic power spectra for spoken (Finnish) phonemes present in LVQ_PAK data. X-axes: frequency, Y-axes: log of power. Original LVQ_PAK power spectra were not available, and these images are computed from similar Finnish speech data.

each component is sensitive to.

It was noted above that component 2 suffices to distinguish A from the other phonemes. The first *formant frequency* of A, shown by the leftmost peak in the spectrum of Fig. 6, matches the location of the upward spike in the second filter in Fig. 5. The filter gives a strong positive response to the peak in A, and hence values of component 4 for A are large. This is seen clearly in the topmost scatterplot of Fig. 4. On the other hand, the vowel in the other end of the continuum, I, has only small values at the formant frequency of A, and the first and second formant frequencies of I are near mild downward spikes in the filter. Hence, the filter gives a small response to I and the coordinates along component 2 are small in Fig. 4. The filter is not attuned just to either A or I separately; instead, it has found a shape that responds to both phonemes but in the opposite fashion—this makes it discriminate the two. The other vowels have responses between those of A and I; for example, the first formant frequency of U is about the same as that of I, but U also has large values near the upward peak, and low ones near the downward peak of the filter near 2 kHz.

The phoneme I was an example that cannot be discriminated based on only one component. The lower end of the spectrum resembles that of S; indeed, filter 2 gives S and I about the same response. Much of the power in the spectrum of S is concentrated at high frequencies (Fig. 6). None of the filters have large peaks there. Filter 1 has a mild downward spike near the start of the spectrum, and a broad two-tipped upward spike after it. Hence it gives similar responses to the vowels, since their peaks are near the upward spike. However, notice that filter 2 does separate I from the other vowels, which receive larger responses, and filter 1 separates S from I, since it has a downward spike at the first peak in the spectrum of S and the large upward spike starts at the first formant frequency of I. The filters then compensate each other when separating I. The projection is informative as a whole.

VI. Discussion

We studied data-driven learning of components of primary data that are informative of or relevant for auxiliary data. The auxiliary or side data is used to define what is relevant, and the components disregard the rest. We suggest coining this task *relevant component analysis.*

Such models are applicable to dimensionality reduction, visualization, data exploration, and feature exploration. Moreover, the specific model that was proposed is applicable as a linear model of conditional probabilities. The model complements (unsupervised) projections that aim at preserving distances or cluster structures within the data. The components model class changes, and can be used to visualize the class structure and find which components are relevant to the class changes. Such 'feature exploration' was demonstrated for speech data in this paper, for gene expression data in [35], and the model was applied for assessing and visualizing convergence of MCMC chains in [36].

In this paper a family of very simple generative models was introduced for finding linear relevant components when the auxiliary data are categorical. The models were shown to have asymptotic connections to maximization of mutual information and the so-called learning metrics; for the specific model family the concepts predictivity, maximization of mutual information, and principal component analysis in learning metrics are equivalent. In a sense, the model is a generalization of classical linear discriminant analysis. The asymptotic theoretical connections were proven for consistent estimators such as Parzen estimators; the results do not in general hold for nonconsistent estimators.

In experiments, the model outperformed both the classical linear discriminant analysis and a Renyi entropy-based projection method while having essentially equivalent computational cost with the latter. LDA is much faster. The proposed method is based on Shannon entropy, and the result may suggest a more general conjecture about the need to switch to Renyi entropy.

Visualizations of different models were additionally compared with the same conclusion, and practical data analysis was demonstrated on phoneme data.

The proposed method involves estimation of the conditional density of auxiliary data in the projection subspace. In this paper this was done with a non-parametric Parzen estimator; other estimators could also be used. The Parzen estimator has the advantage that it need not be separately re-estimated while the projection is optimized. However, it can be computationally demanding for large data sets. An alternative would be to use a parametric estimator and optimize both the projection and the estimator simultaneously. For now, we did not try this since the speed-up is not expected to be great for data sets of moderate size.

The projection was optimized by stochastic approximation. Any other standard optimization method could have been used instead; stochastic approximation has the advantage that it can be easily extended to on-line learning.

The presented method requires labeled learning data. Unlabeled data can be analyzed with the components, but only labeled data affect the learning. The model can be extended in a straightforward way to utilize unlabeled data

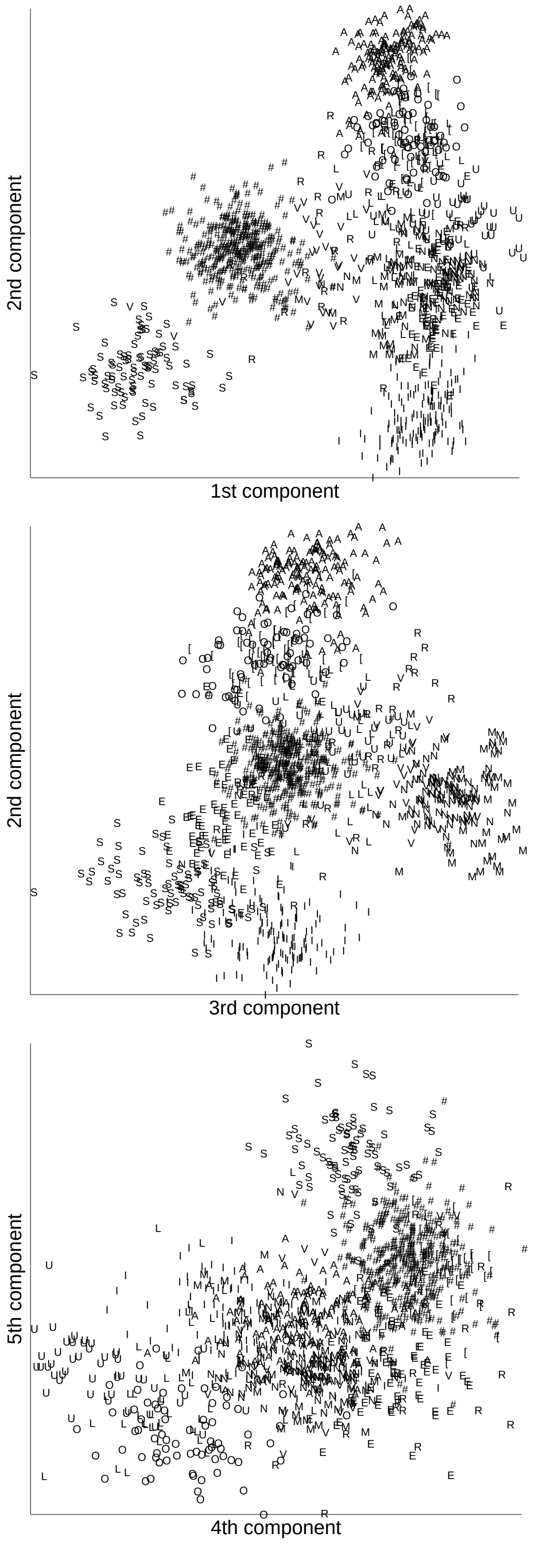

Fig. 4. Scatterplots of LVQ_PAK leave-out data on component pairs of the 5-dimensional projection. Ä is denoted by '[', silence by '#', and other phoneme classes by their respective letters.

for learning as well, along the lines of [37]. The present objective function, $L = L(C|\mathbf{f}(X))$ (cf. 1), will be replaced by a compromise parameterized by $0 \leq \beta \leq 1$: $L' = \beta L(C|\mathbf{f}(X)) + (1-\beta)L(\mathbf{f}(X))$. Here C is the label, $\mathbf{f}(X)$ is the projected primary data, and $L(\mathbf{f}(X))$ is the likelihood of a suitable generative model of $\mathbf{f}(X)$. When $\beta = 0$ the model is unsupervised, when $\beta = 1/2$ the projection models the joint density, and setting $\beta = 1$ gives the present model. Changing β will naturally change the goal of the projection, and some of the useful interpretations given in Section IV will be lost. However, there is evidence that unlabeled data may help even in a discrimination task [38].

Only linear components were considered; extensions to non-linear projections will be studied later. Linearity is a fairly restrictive constraint for a projection, and it is expected that the differences between alternative methods will be larger for non-linear methods. Another obvious direction for extensions is towards more general auxiliary data. The presented method was an extension of linear discriminant analysis; canonical correlations could be generalized in the same fashion.

Zhu and Hastie have recently presented a different kind of related work [39]. They extend classical LDA by maximizing the likelihood ratio between class-specific and class-independent models. For non-parametric estimators the method is very close to ours. More generally, however, the difference is that we do not need an estimator of primary data densities—the conditional class probabilities suffice.

Another very recent related work [40] defines a new objective function for dimensionality-reducing projections by making a suitable conditional independence assumption and using Gaussian-based kernels. The relative metrits of this promising work should be assessed empirically later. It shares some of the good properties of our method: Parametric assumptions about the distribution of classes are not needed, and the covariates need not be modeled. Both methods are ultimately optimized by standard gradient techniques. A possible problem in [40] is that a seemingly arbitrary measure of matrix size needs to be chosen. By contrast, the likelihood in our method is a well-defined finite-data criterion.

In this paper we did not consider the important, much studied issues of how to select the projection dimensionality and the kernel. Since the model conforms to the standard probabilistic framework, Bayesian methods could be applied in principle.

Appendix A

The on-line (stochastic) gradient for optimizing the objective function L in (1) with regard to the projection matrix $\mathbf{W}$ of the linear projection $\mathbf{f}(\mathbf{x}) = \mathbf{W}^T\mathbf{x}$ will be derived in this Appendix.

Assume the parameters of the conditional probability estimate $\hat{p}(c|\mathbf{f}(\mathbf{x}))$, defined by (2), are fixed with respect to $\mathbf{W}$. Then

$$\frac{\partial}{\partial \mathbf{W}} L(\mathbf{x}, c; \mathbf{W}) = \frac{\partial}{\partial \mathbf{W}} \log \hat{p}(c|\mathbf{f}(\mathbf{x})) = \frac{\frac{\partial}{\partial \mathbf{W}} \hat{p}(c|\mathbf{f}(\mathbf{x}))}{\hat{p}(c|\mathbf{f}(\mathbf{x}))} \ .$$

We have

$$\frac{\partial}{\partial \mathbf{W}} \hat{p}(c|\mathbf{f}(\mathbf{x})) = \frac{\frac{\partial}{\partial \mathbf{W}} G(\mathbf{f}(\mathbf{x}), c)}{\sum_{c'} G(\mathbf{f}(\mathbf{x}), c')} - \hat{p}(c|\mathbf{f}(\mathbf{x})) \frac{\sum_{c'} \frac{\partial}{\partial \mathbf{W}} G(\mathbf{f}(\mathbf{x}), c')}{\sum_{c'} G(\mathbf{f}(\mathbf{x}), c')}$$

and hence

$$\frac{\partial}{\partial \mathbf{W}} L(\mathbf{x}, c; \mathbf{W}) = \frac{\frac{\partial}{\partial \mathbf{W}} G(\mathbf{f}(\mathbf{x}), c)}{G(\mathbf{f}(\mathbf{x}), c)} - \frac{\sum_{c'} \frac{\partial}{\partial \mathbf{W}} G(\mathbf{f}(\mathbf{x}), c')}{\sum_{c'} G(\mathbf{f}(\mathbf{x}), c')} \ . \quad (13)$$

Since

$$\begin{aligned}\frac{\partial}{\partial \mathbf{W}} g(\mathbf{f}(\mathbf{x}), m) &= \frac{\exp(-\|\mathbf{W}^T(\mathbf{x} - \mathbf{r}_m)\|^2/(2\sigma^2))}{(2\pi\sigma^2)^{d/2}} \cdot \frac{-1}{\sigma^2}(\mathbf{x} - \mathbf{r}_m)(\mathbf{x} - \mathbf{r}_m)^T \mathbf{W} \\ &= \frac{-1}{\sigma^2} g(\mathbf{f}(\mathbf{x}), m)(\mathbf{x} - \mathbf{r}_m)(\mathbf{x} - \mathbf{r}_m)^T \mathbf{W} \\ &= \frac{-1}{\sigma^2} g(\mathbf{f}(\mathbf{x}), m)(\mathbf{x} - \mathbf{r}_m)(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{r}_m))^T\end{aligned}$$

we furthermore have

$$\frac{\partial}{\partial \mathbf{W}} G(\mathbf{f}(\mathbf{x}), c) = \frac{-1}{\sigma^2} \sum_{m=1}^{M} \psi_{mc} g(\mathbf{f}(\mathbf{x}), m)(\mathbf{x} - \mathbf{r}_m)(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{r}_m))^T \ .$$

Substituting the above into (13) the stochastic gradient of the objective function becomes

$$\begin{aligned}\sigma^2 \frac{\partial}{\partial \mathbf{W}} L(\mathbf{x}, c; \mathbf{W}) = &- \frac{\sum_m \psi_{mc} g(\mathbf{f}(\mathbf{x}), m)(\mathbf{x} - \mathbf{r}_m)(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{r}_m))^T}{\sum_k \psi_{kc} g(\mathbf{f}(\mathbf{x}), k)} \\ &+ \frac{\sum_m (\sum_{c'} \psi_{mc'}) g(\mathbf{f}(\mathbf{x}), m)(\mathbf{x} - \mathbf{r}_m)(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{r}_m))^T}{\sum_k (\sum_{c'} \psi_{kc'}) g(\mathbf{f}(\mathbf{x}), k)} \\ = &- \sum_m \frac{\psi_{mc} g(\mathbf{f}(\mathbf{x}), m)}{\sum_k \psi_{kc} g(\mathbf{f}(\mathbf{x}), k)} (\mathbf{x} - \mathbf{r}_m)(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{r}_m))^T \\ &+ \sum_m \frac{(\sum_{c'} \psi_{mc'}) g(\mathbf{f}(\mathbf{x}), m)}{\sum_k (\sum_{c'} \psi_{kc'}) g(\mathbf{f}(\mathbf{x}), k)} (\mathbf{x} - \mathbf{r}_m)(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{r}_m))^T \ .\end{aligned}$$

Defining the expectation-like operators $E_{\xi(m|\mathbf{x})}$ and $E_{\xi(m|\mathbf{x},c)}$ as weighted summation over m, with weights given by (5) and (6), respectively, finally yields (4).

Appendix B

Torkkola and Campbell [7] gave two alternative definitions for the Renyi entropy-based quadratic mutual information; the second is

$$I_T(C, Y) = \sum_c \int p(c, \mathbf{y})^2 d\mathbf{y} + \sum_c \int p(c)^2 p(\mathbf{y})^2 d\mathbf{y} - 2 \sum_c \int p(c, \mathbf{y}) p(c) p(\mathbf{y}) d\mathbf{y} \quad (14)$$

where $\mathbf{y} = \mathbf{f}(\mathbf{x}) = \mathbf{W}^T\mathbf{x}$ are the projection coordinates. This form was used in later papers [15]. When densities in the projection space are estimated with Parzen estimators based on data $\{(\mathbf{x}_k, c_k)\}_{k=1}^N$, the integral (14) can be evaluated analytically. The result, the objective function L_{MRMI} of MRMI, is ([7] with a slightly different notation)

$$\begin{aligned}
L_{MRMI} &= \frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{N}\delta_{c_k,c_l}G(\mathbf{f}(\mathbf{x}_k)|\mathbf{f}(\mathbf{x}_l),\sigma^2\mathbf{I}) \\
&+ \frac{1}{N^2}\left(\sum_c\left(\sum_{k=1}^{N}\frac{\delta_{c_k,c}}{N}\right)^2\right)\sum_{k=1}^{N}\sum_{l=1}^{N}G(\mathbf{f}(\mathbf{x}_k)|\mathbf{f}(\mathbf{x}_l),\sigma^2\mathbf{I}) \\
&\quad - \frac{2}{N^2}\sum_c\left[\left(\sum_{k=1}^{N}\frac{\delta_{c_k,c}}{N}\right)\right. \\
&\quad \left.\cdot\sum_{k=1}^{N}\sum_{l=1}^{N}\delta_{c_k,c}G(\mathbf{f}(\mathbf{x}_k)|\mathbf{f}(\mathbf{x}_l),\sigma^2\mathbf{I})\right] \qquad (15)
\end{aligned}$$

where $G(\mathbf{f}(\mathbf{x})|\mathbf{f}(\mathbf{x}_k),\sigma^2\mathbf{I})$ is the value at $\mathbf{f}(\mathbf{x})$ of a d-dimensional Gaussian distribution with mean $\mathbf{f}(\mathbf{x}_k)$ and covariance matrix $\sigma^2\mathbf{I}$. Denoting $\hat{p}(c) = \frac{1}{N}\sum_{k=1}^{N}\delta_{c_k,c}$, (15) can be rewritten in a simpler form as

$$L_{MRMI} = \frac{1}{N}\sum_{k=1}^{N}L_{MRMI}(\mathbf{x}_k, c_k; \mathbf{W})$$

where

$$\begin{aligned}
L_{MRMI}(\mathbf{x}, c; \mathbf{W}) = \sum_{l=1}^{N}&\frac{G(\mathbf{f}(\mathbf{x})|\mathbf{f}(\mathbf{x}_l),\sigma^2\mathbf{I})}{N} \\
&\cdot\left[\delta_{c,c_l} + \sum_{c'}\hat{p}(c')^2 - 2\hat{p}(c)\right] .
\end{aligned}$$

In stochastic approximation we pick $(\mathbf{x}, c)$ from the data (with equal probabilities $1/N$) and compute the gradient of $L_{MRMI}(\mathbf{x}, c; \mathbf{W})$. Since

$$\begin{aligned}
&\frac{\partial}{\partial\mathbf{W}}G(\mathbf{f}(\mathbf{x})|\mathbf{f}(\mathbf{x}_l),\sigma^2\mathbf{I}) \\
&\quad = -G(\mathbf{f}(\mathbf{x})|\mathbf{f}(\mathbf{x}_l),\sigma^2\mathbf{I})(\mathbf{x}-\mathbf{x}_l)(\mathbf{x}-\mathbf{x}_l)^T\frac{\mathbf{W}}{\sigma^2} \\
&\quad = -\frac{1}{\sigma^2}G(\mathbf{f}(\mathbf{x})|\mathbf{f}(\mathbf{x}_l),\sigma^2\mathbf{I})(\mathbf{x}-\mathbf{x}_l)(\mathbf{f}(\mathbf{x})-\mathbf{f}(\mathbf{x}_l))^T
\end{aligned}$$

the gradient of $L_{MRMI}(\mathbf{x}, c; \mathbf{W})$ becomes (9).

Appendix C

A connection between the proposed algorithm and learning metrics is sketched in this appendix. The mutual information after the projection approximately equals a certain average reconstruction error. The connection is actually not restricted to only linear projections $\mathbf{f}(\mathbf{x})$.

The assumptions are rather strong, which makes the connection more a justification than a proof. It is assumed that local approximations to the metrics are sufficient, and that good reconstruction points exist. Justification for these assumptions is given at the end of the appendix.

A. Preliminaries

Denote the set of points that are projected to $\mathbf{y}$ by $S(\mathbf{y})$, that is,

$$S(\mathbf{y}) = \{\mathbf{x}|\mathbf{f}(\mathbf{x}) = \mathbf{y}\} .$$

Given a projected point $\mathbf{y}$, the conditional auxiliary distribution is an expectation over $S(\mathbf{y})$ given by

$$p(c|\mathbf{y}) = p(c,\mathbf{y})/p(\mathbf{y}) = \frac{\int_{\mathbf{x}\in S(\mathbf{y})}p(c,\mathbf{x})d\mathbf{x}}{\int_{\mathbf{x}\in S(\mathbf{y})}p(\mathbf{x})d\mathbf{x}} .$$

Let the *reconstruction* $\mathbf{r}(\mathbf{y}) \in S(\mathbf{y})$ of a projection $\mathbf{y}$ be the point that minimizes $D_{KL}(p(c|\mathbf{y}), p(c|\mathbf{r}(\mathbf{y})))$.

B. The connection

When a consistent estimator is used for the auxiliary distributions, the second term in (10) asymptotically vanishes, and the objective function is then equivalent to

$$\begin{aligned}
I(C,Y) &= \int p(\mathbf{y})\sum_c p(c|\mathbf{y})\log p(c|\mathbf{y})d\mathbf{y} + H(C) \\
&= \int p(\mathbf{y})\sum_c p(c|\mathbf{y})\log p(c|\mathbf{r}(\mathbf{y}))d\mathbf{y} \\
&+ \int p(\mathbf{y})\sum_c p(c|\mathbf{y})\log\frac{p(c|\mathbf{y})}{p(c|\mathbf{r}(\mathbf{y}))} + H(C) \\
&= \int p(\mathbf{y})\sum_c p(c|\mathbf{y})\log p(c|\mathbf{r}(\mathbf{y}))d\mathbf{y} \\
&+ E_{p(\mathbf{y})}[D_{KL}(p(c|\mathbf{y}), p(c|\mathbf{r}(\mathbf{y})))] + H(C) \\
&= \int p(\mathbf{x})\sum_c p(c|\mathbf{x})\log p(c|\mathbf{r}(\mathbf{f}(\mathbf{x})))d\mathbf{x} \\
&+ E_{p(\mathbf{y})}[D_{KL}(p(c|\mathbf{y}), p(c|\mathbf{r}(\mathbf{y})))] + H(C) \\
&= -E_{p(\mathbf{x})}[D_{KL}(p(c|\mathbf{x})||p(c|\mathbf{r}(\mathbf{y})))] \\
&+ E_{p(\mathbf{y})}[D_{KL}(p(c|\mathbf{y}), p(c|\mathbf{r}(\mathbf{y})))] + H(C) - H(C|X) .
\end{aligned}$$

Here X denotes the random variable having values $\mathbf{x}$ in the primary data space. For two close-by points the learning metric is given by the Kullback-Leibler divergence in (12). We will assume that this local approximation is approximately correct (justification at the end); the last line can then be rewritten as

$$\begin{aligned}
I(C,Y) &\approx -E_{p(\mathbf{x})}[d_L^2(\mathbf{x}, \mathbf{r}(\mathbf{f}(\mathbf{x})))] \\
&\quad + E_{p(\mathbf{y})}[D_{KL}(p(c|\mathbf{y}), p(c|\mathbf{r}(\mathbf{f}(\mathbf{x}))))] + I(C,X) \qquad (16)
\end{aligned}$$

where the first term on the right is an average *reconstruction error*, squared distance from a point to its reconstruction, computed in the learning metric. The term $I(C,X)$ is constant with respect to the projection. If the middle term is approximately constant (with certain assumptions it is close to zero; justification at the end), maximizing $I(C,Y)$ is equivalent to minimizing the average reconstruction error.

If the middle term is not (nearly) constant, the proposed algorithm does not minimize the reconstruction error. If the goal is not to maximize mutual information but to minimize the reconstruction error, it can in principle be done by minimizing

$$E_{p(\mathbf{x})}[d_L^2(\mathbf{x},\mathbf{r}(\mathbf{f}(\mathbf{x})))]$$
$$\approx -I(C,Y)+E_{p(\mathbf{y})}[D_{KL}(p(c|\mathbf{y}),p(c|\mathbf{r}(\mathbf{y})))]+\text{const.}$$

C. Note on sufficiency of local distances

The local distance approximation used in the proof is reasonable when the learning metric distances in (almost) all $S(\mathbf{y})$ are small, or alternatively when the Fisher information matrix $\mathbf{J}(\mathbf{x})$ that defines the local metric in (12) is nearly constant in $S(\mathbf{y})$. The former holds at least when the projection subspace already explains most variation in the auxiliary data (that is, the projection dimensionality is large enough and the projection matrix is close to the optimum). Even if the distances are not small, the latter holds if the auxiliary data in $S(\mathbf{y})$ changes at a constant 'rate' (as measured by the Kullback-Leibler divergence).

D. Note on the reconstruction points

The divergence term (middle term in (16)) is zero trivially if the distribution of auxiliary data is constant in the direction orthogonal to the subspace. This holds approximately if the data is local in the orthogonal direction, which is likely to hold is the projection dimensionality is large.

Additionally, the divergence term is zero at least if the auxiliary distributions $p(c|\mathbf{x})$ in $S(\mathbf{y})$ form a *convex set* in the distribution space, that is, any weighted average of two auxiliary distributions in $S(\mathbf{y})$ also exists in $S(\mathbf{y})$. The simplest example is a weighted combination of two distributions by a continuous function of $\mathbf{x}$.

Acknowledgment

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

References

[1] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, pp. 179–188, 1936.

[2] N. H. Timm, *Applied Multivariate Analysis.* New York: Springer, 2002.

[3] S. Becker and G. E. Hinton, "Self-organizing neural network that discovers surfaces in random-dot stereograms," *Nature*, vol. 355, pp. 161–163, 1992.

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

[5] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *37th Annual Allerton Conference on Communication, Control, and Computing*, Urbana, Illinois, 1999, pp. 368–377.

[6] F. Pereira, N. Tishby, and L. Lee, "Distributional clustering of English words," in *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics.* Columbus, OH: ACL, 1993, pp. 183–190.

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

[8] K. Torkkola, "Feature extraction by non-parametric mutual information maximization," *Journal of Machine Learning Research*, vol. 3, pp. 1415–1438, 2003.

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

[10] J. C. Principe, J. W. Fisher III, and D. Xu, "Information-theoretic learning," in *Unsupervised Adaptive Filtering*, S. Haykin, Ed. New York: Wiley, 2000.

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

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

[13] L. P. Devroye and T. J. Wagner, "Distribution-free consistency results in nonparametric discrimination and regression function estimation," *The Annals of Statistics*, vol. 8, no. 2, pp. 231–239, March 1980.

[14] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications.* New York: Springer, 1997.

[15] K. Torkkola, "Learning discriminative feature transforms to low dimensions in low dimentions," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Cambridge, MA: MIT Press, 2002.

[16] K.-C. Li, "Sliced inverse regression for dimension reduction," *Journal of the American Statistical Association*, vol. 86, no. 414, pp. 316–327, June 1991.

[17] ——, "On principal Hessian directions for data visualization and dimension reduction: Another application of Stein's lemma," *Journal of the Americal Statistical Association*, vol. 87, no. 420, pp. 1025–1039, December 1992.

[18] R. D. Cook and S. Weisberg, "Sliced inverse regression for dimension reduction: Comment," *Journal of the American Statistical Association*, vol. 86, no. 414, pp. 328–332, June 1991.

[19] R. D. Cook and X. Yin, "Dimension reduction and visualization in discriminant analysis," *Australian & New Zealand Journal of Statistics*, vol. 43, no. 2, pp. 147–199, 2001.

[20] S. Weisberg, "Dimension reduction regression in R," *Journal of Statistical Software*, vol. 7, no. 1, pp. 1–22, 2002.

[21] B. D. Ripley, *Pattern Recognition and Neural Networks.* Cambridge, UK: Cambridge University Press, 1996.

[22] J. Kay, "Feature discovery under contextual supervision using mutual information," in *Proceedings of IJCNN'92, International Joint Conference on Neural Networks.* Piscataway, NJ: IEEE, 1992, pp. 79–84.

[23] T. Hastie and R. Tibshirani, "Discriminant analysis by Gaussian mixtures," *Journal of the Royal Statistical Society B*, vol. 58, pp. 155–176, 1996.

[24] N. Kumar and A. G. Andreou, "Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition," *Speech Communication*, vol. 26, 1998.

[25] G. Saon, M. Padmanabhan, R. Gopinath, and S. Chen, "Maximum likelihood discriminant feature spaces," in *ICASSP 2000*, 2000.

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

[27] J. Sinkkonen, S. Kaski, and J. Nikkilä, "Discriminative clustering: Optimal contingency tables by learning metrics," in *Proceedings of the ECML'02, 13th European Conference on Machine Learning*, T. Elomaa, H. Mannila, and H. Toivonen, Eds. Berlin: Springer, 2002, pp. 418–430.

[28] J. Peltonen, A. Klami, and S. Kaski, "Learning more accurate metrics for self-organizing maps," in *Artificial Neural Networks—ICANN 2002*, J. R. Dorronsoro, Ed. Berlin: Springer, 2002, pp. 999–1004.

[29] S. Amari and H. Nagaoka, *Methods of Information Geometry.* American Mathematical Society and Oxford University Press, 2000, vol. 191.

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

[31] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases," `http://www.ics.uci.edu/~mlearn/MLRepository.html`, 1998.

[32] T. Kohonen, J. Hynninen, J. Kangas, J. Laaksonen, and K. Torkkola, "LVQ_PAK: The learning vector quantization program package," Helsinki University of Technology, Laboratory of Computer and Information Science, FIN-02150 Espoo, Finland, Tech. Rep. A30, 1996.

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

[34] K. Torkkola, J. Kangas, P. Utela, S. Kaski, M. Kokkonen, M. Kurimo, and T. Kohonen, "Status report of the Finnish phonetic typewriter project," in *Artificial Neural Networks*, T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, Eds. Amsterdam: North-Holland, 1991, pp. 771–776.

[35] S. Kaski and J. Peltonen, "Informative discriminant analysis," in *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*. Menlo Park, CA: AAAI Press, 2003, pp. 329–336.

[36] J. Venna, S. Kaski, and J. Peltonen, "Visualizations for assessing convergence and mixing of MCMC," in *Proceedings of the 14th European Conference on Machine Learning (ECML 2003)*, N. Lavrac, D. Gamberger, H. Blockeel, and L. Todorovsk, Eds. Berlin: Springer, 2003, pp. 432–443.

[37] S. Kaski, J. Sinkkonen, and A. Klami, "Regularized discriminative clustering," in *Neural Networks for Signal Processing XIII*, C. Molina, T. Adali, J. Larsen, M. Van Hulle, S. Douglas, and J. Rouat, Eds. New York, NY: IEEE, 2003, pp. 289–298.

[38] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Cambridge, MA: MIT Press, 2002.

[39] M. Zhu and T. Hastie, "Feature extraction for non-parametric discriminant analysis," *Journal of Computational and Graphical Statistics*, vol. 12, pp. 101–120, 2003.

[40] K. Fukumizu, F. R. Bach, and M. I. Jordan, "Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces," University of California, Berkeley, Department of Statistics, Tech. Rep. 641, 2003.

Jaakko Peltonen reveived the M.Sc. degree in computer and information science from Helsinki University of Technology, Espoo, Finland, in 2001.

He is currently pursuing the Ph.D. degree in learning metrics at the Laboratory of Computer and Information Science (Neural Networks Research Centre), Helsinki University of Technology. His research interests include machine learning, exploratory data analysis and information retrieval.

Samuel Kaski (M'96–SM'02) 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.