Neurocomputing 80 (2012) 31–37

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Dimensionality reduction based on non-parametric mutual information

Lev Faivishevsky *, Jacob Goldberger

School of Engineering, Bar-Ilan University, Ramat-Gan 52900, Israel

ARTICLE INFO

Available online 6 November 2011

Keywords:
Dimensionality reduction
Semisupervised learning
Classification
Regression

ABSTRACT

In this paper we introduce a supervised linear dimensionality reduction algorithm which finds a projected input space that maximizes the mutual information between input and output values. The algorithm utilizes the recently introduced MeanNN estimator for differential entropy. We show that the estimator is an appropriate tool for the dimensionality reduction task. Next we provide a nonlinear regression algorithm based on the proposed dimensionality reduction approach. The regression algorithm achieves comparable to state-of-the-art performance on the standard data sets but is three orders of magnitude faster. In addition we describe applications of the proposed dimensionality reduction algorithm to reduced-complexity supervised and semisupervised classification tasks.

1. Introduction

Many real world regression problems deal with analysis of high-dimensional data. The goal of supervised dimensionality reduction is to reduce the dimensionality of the input space while preserving the information about the output values. One typical method is canonical correlation analysis (CCA) [1] which is a way of measuring the linear relationship between two multidimensional variables. It finds two bases, one for each variable, that are optimal with respect to correlations and, at the same time, it finds the corresponding correlations. Partial least squares (PLS) was developed in econometrics in the 1960s by Herman Wold. PLS is basically the singular-value decomposition of a between-sets covariance matrix; for an overview, see e.g. [2]. In the PLS regression, the principal vectors corresponding to the largest principal values are used as a new, lower dimensional, basis for the signal. A regression of y onto x is then calculated in this new space, see [3] for a recent study in dimensionality reduction using reproducing Hilbert kernel spaces. Other relevant studies include [4,5].

Recently, methods for feature extraction that are based on mutual information maximization have been proposed, see e.g. [6]. These methods, however, perform dimensionality reduction by choosing a subset of given dimensions only and applying the k NN based mutual information estimators. Torkkola [7] suggested using mutual information for supervised dimensionality reduction in a classification setup by learning a Parzen window approximation for the joint input-target distribution. One of the main difficulties with this kernel-based estimation lies in the need for a correct choice of kernel width.

Another approach known as Maximum Mutual Information Projection (MMIP) [8] proposed dimensionality reduction based on histogram estimation of mutual information. A significant drawback of this approach is the treatment of the multidimensional result subspace. The histogram estimation of mutual information is of exponential complexity with respect to the result subspace dimension. Therefore the MMIP finds the subspace in an iterative manner, performing an optimal one-dimensional projection on each step. The next iteration then searches for the optimal 1D subspace in the orthogonal complement subspace of the previously found 1D projections. This approach is suboptimal because the mutual information is a nonlinear function and the orthogonal projection does not remove the dependence on the previously found 1D subspaces. This means that the resulting output multidimensional subspace does not necessarily correspond to the maximal information between the projected inputs and targets. The recently proposed Maximal Mutual Information Feature Extractor (MMIFE) [9] is based on a nonlinear entropy estimation of the 1D random variable. The MMIFE has a similar drawback as MMIP in the case of multidimensional resulting subspaces since the MMIFE applies the same process of iterative 1D projections and orthogonal complements.

In this paper we present an algorithm for supervised linear dimensionality reduction that uses mutual information as a criterion. The advantage of this method is that it preserves the information contained in the input space by searching for optimal linear combinations of existing features. This optimization is efficiently accomplished by conjugated gradient methods applied to the recently introduced MeanNN estimator [10] for the mutual information, which has the advantage of an analytical expression for the gradient. Based on this estimator we define an efficient

* Corresponding author.
E-mail addresses: levtemp@gmail.com (L. Faivishevsky), goldbej@eng.biu.ac.il (J. Goldberger).

0925-2312/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2011.07.028

nonlinear regression in the extracted linear subspace. The performance of the proposed regression is comparable to state-of-the-art methods but is three orders of magnitude faster on the test stage. The same dimensionality reduction concept is then applied to classification tasks.

The rest of the paper is organized as follows. Section 2 reviews non-parametric k NN and MeanNN estimators for differential entropy. Section 3 introduces the Mutual Information Dimensionality Reduction (MIDR) method. Section 4 presents the nonlinear regression method. Section 5 presents an application to classification problems. Section 6 describes an application to semisupervised classification. Section 7 reports experiment results on standard data sets.

2. Non-parametric estimators for differential entropy

Our dimensionality reduction method is based on a smooth non-parametric approximation of differential entropy which is reviewed below. The differential entropy of X is defined as

$$H(X) = -\int f(x)\log f(x)\,dx \tag{1}$$

We describe a derivation of the Shannon differential entropy estimate. Our aim is to estimate $H(X)$ from a random sample $(x_1,\ldots,x_n)$ of n random realizations of a d-dimensional random variable X with an unknown density function $f(x)$. The approach estimates the mutual information based on k-nearest neighbor statistics. The approach was originally suggested by Kozachenko and Leonenko [11], see also [12,13]. A nice property of these estimators is that they can be easily utilized for high dimensional random vectors and no parameters need to be predefined or separably tuned (other than determining the value of k).

The entropy is the average of $-\log f(x)$. If there were unbiased estimators for $\log f(x_i)$, this would yield an unbiased estimator for the entropy. We estimate $\log f(x_i)$ by considering the probability density function $P_{ik}(\epsilon)$ for the distance between x_i and its k-th nearest neighbor (the probability is computed over the positions of all other $n-1$ points, with x_i kept fixed). The probability $P_{ik}(\epsilon)d\epsilon$ is equal to the chance that there is one point within distance $r\in[\epsilon,\epsilon+d\epsilon]$ from x_i, that there are $k-1$ other points at smaller distances, and that the remaining $n-k-1$ points have larger distances from x_i. Denote the mass of the ϵ-ball centered at x_i by $p_i(\epsilon)$, i.e. $p_i(\epsilon)=\int_{\|x-x_i\|<\epsilon} f(x)\,dx$. Applying the trinomial formula we obtain

$$P_{ik}(\epsilon) = \frac{(n-1)!}{1!(k-1)!(n-k-1)!}\frac{dp_i(\epsilon)}{d\epsilon}p_i^{k-1}(1-p_i)^{n-k-1} \tag{2}$$

It is easy to verify that $\int P_{ik}(\epsilon)\,d\epsilon = 1$. Hence, the expected value of the function $\log p_i(\epsilon)$ according to the distribution $P_{ik}(\epsilon)$ is

$$E_{P_{ik}(\epsilon)}(\log p_i(\epsilon)) = \int_0^\infty P_{ik}(\epsilon)\log p_i(\epsilon)\,d\epsilon = k\binom{n-1}{k}\int_0^1 p^{k-1}(1-p)^{n-k-1} \times \log p\,dp = \psi(k)-\psi(n) \tag{3}$$

where $\psi(x)$ is the digamma function (the logarithmic derivative of the gamma function). To verify the last equality, differentiate the identity $\int_0^1 x^{a-1}(1-x)^{b-1} = \Gamma(a)\Gamma(b)/\Gamma(a+b)$ with respect to parameter a and recall that $\Gamma'(x)=\psi(x)\Gamma(x)$. The expectation is taken over the positions of all other $n-1$ points, with x_i kept fixed. Assuming that $f(x)$ is almost constant in the entire ϵ-ball around x_i, we obtain

$$p_i(\epsilon) \approx c_d\epsilon^d f(x_i) \tag{4}$$

where d is the dimension of x and c_d is the volume of the d-dimensional unit ball ($c_d=\pi^{d/2}/\Gamma(1+d/2)$ for the Euclidean norm). Substituting Eq. (4) into Eq. (3), we obtain

$$-\log f(x_i) \approx \psi(n)-\psi(k)+\log(c_d)+dE(\log(\epsilon)) \tag{5}$$

which leads to the unbiased k NN estimator for the entropy [11]:

$$H_k(X) = \psi(n)-\psi(k)+\log(c_d)+\frac{d}{n}\sum_{i=1}^{n}\log\epsilon_i \tag{6}$$

where ϵ_i is the distance from x_i to its k-th nearest neighbor. An alternative proof of the asymptotic unbiasedness and consistency of the k NN estimator can be found in [14]. This estimation thus links between information theory and nearest-neighbor concepts. Unlike previously suggested approximations, there are no parameters to be tuned.

Because it is non-parametric, the k NN estimator (6) relies on order statistics. This makes the analytical calculation of the gradient virtually impossible. Furthermore it leads to a certain lack of smoothness of the estimator value as a function of the sample coordinates. Finally, finding the k-nearest neighbor is a computationally intensive problem. It becomes practically obligatory to use involved approximate nearest neighbor techniques for large data sets.

Recently we proposed a new smooth estimator for the entropy evaluation as a function of sample coordinates [10]. It is based on the fact that the k NN estimator (6) is valid for every k. Therefore the differential entropy can also be extracted from the mean of several estimators corresponding to different values of k. Next we consider all the possible values of order statistics k from 1 to $n-1$:

$$H_{mean} = \frac{1}{n-1}\sum_{k=1}^{n-1}H_k = \log(c_d)+\psi(n)+\frac{1}{n-1}\sum_{k=1}^{n-1}\left(-\psi(k)+\frac{d}{n}\sum_{i=1}^{n}\log\epsilon_{i,k}\right) \tag{7}$$

where $\epsilon_{i,k}$ is the k-th nearest neighbor of x_i. Consider the double-summation last term in Eq. (7). Exchanging the order of summation, this sum adds the sum of the log of its distances to all its nearest neighbors in the sample for each sample point x_i. It is of course equivalent to the sum of the logs of its distances to all other points in the sample set. Hence the mean estimator (7) for the differential entropy can be written as

$$H_{mean} = \frac{d}{n(n-1)}\sum_{i\neq j}\log\|x_i-x_j\|+\text{const} \tag{8}$$

Note that, unlike the k NN based estimator, this entropy estimator is a smooth function of the given data points and is not sensitive to small perturbations in the values of $x_1,\ldots,x_n$.

Next assume that in addition to input vectors $x_1,\ldots,x_n\in\mathcal{R}^D$ we also have target values $y_1,\ldots,y_n\in\mathcal{R}$. We can express the mutual information between X and Y by means of joint and marginal entropies. Using the MeanNN entropy estimator we get a MeanNN estimator for the mutual information:

$$I_{mean}(X;Y) = H_{mean}(X)+H_{mean}(Y)-H_{mean}(X,Y) \tag{9}$$

3. MI linear dimensional reduction

In this study we address the problem of supervised dimensionality reduction. Our goal is to utilize the smooth entropy estimator, described in the previous section, to form an information-theoretic criterion that can be easily optimized. Given n vectors $X=\{x_1,\ldots,x_n\}$ in $\mathcal{R}^D$ and the corresponding target values $Y=\{y_1,\ldots,y_n\}$ in $\mathcal{R}$ we want to find a linear transformation $A:\mathcal{R}^D\to\mathcal{R}^d$ that maximizes the mutual information $I(AX;Y)$. Since we want to predict the target in the projected space, we search for features that are most correlated with the target. The mutual

information criterion is a way to quantify this correlation. We search for a matrix A that maximizes the mutual information between the targets and the transformed inputs.

To estimate the mutual information between a one-dimensional random variable Y and the d-dimensional random vector AX, with no prior information about their joint distribution, we apply the MeanNN estimator of mutual information (see Section 2). We want to maximize the mutual information as a function of the matrix A. An information theory relation reveals that: $I(AX;Y)=H(AX)+H(Y)-H(AX,Y)$. Since Y does not depend on A, to maximize the mutual information we need to compute:

$$I_{mean}(AX;Y)=\text{const}+\frac{d}{n(n-1)}\sum_{i\neq j}\log(\|A(x_i-x_j)\|^2)-\frac{d+1}{n(n-1)}\sum_{i\neq j}\log(\|A(x_i-x_j)\|^2+\|y_i-y_j\|^2) \quad (10)$$

To find the best linear dimensionality reduction we have to solve the optimization problem:

$$\hat{A}=\arg\max_A I_{mean}(AX;Y)$$

Such an optimization can be done using conjugate gradient techniques. The smoothness of the MeanNN entropy estimator enables its gradient to be analytically computed. Differentiating $I_{mean}(AX;Y)$ with respect to the transformation matrix A yields a gradient rule which we can use for learning:

$$\frac{\partial I_{mean}(AX;Y)}{\partial A}=\frac{d}{n(n-1)}\sum_{i\neq j}\frac{A(x_i-x_j)(x_i-x_j)^\top}{\|A(x_i-x_j)\|^2}-\frac{d+1}{n(n-1)}\sum_{i\neq j}\frac{A(x_i-x_j)(x_i-x_j)^\top}{\|A(x_i-x_j)\|^2+\|y_i-y_j\|^2} \quad (11)$$

The learning algorithm therefore is: maximize the above objective (10) using a gradient-based optimizer such as delta-bar-delta or conjugate gradients. Of course, as the cost function above is not convex, some care must be taken to avoid local maxima during training. We dub the proposed method Mutual Information Dimensionality Reduction (MIDR). It is summarized in Fig. 1. A standard information theory exercise reveals that $I(AX;Y)$ is invariant to any invertible transformation on either AX or Y. Since $\|A(x_i-x_j)\|^2=(x_i-x_j)^\top A^\top A(x_i-x_j)$, our optimization criterion depends only on $A^\top A$. Hence, every orthogonal matrix $\mathbf{R}_{d\times d}$ yields a solution $\mathbf{R}\cdot A$ that is completely equivalent to A. Therefore our cost function $I_{mean}(AX;Y)$ is rotation invariant. We note in passing that we can make the MI approximation scale invariant by applying it to $I(AX;\|A\|Y)=I(AX;Y)$ such that $\|\cdot\|$ is the Frobenius norm. However, by using this approach the cost we optimize is no longer rotation invariant since the Frobenius norm is not rotation invariant. Thus we opt for the MIDR approach that yields a rotation invariant score. In our approach, to control the matrix scale we can penalize large-norm transformations A by adding a regularization term $-\lambda\|A\|^2$ to the cost function we are maximizing such that λ is a pre-specified positive constant that can be set in a cross-validation step.

4. An application to nonlinear regression

To demonstrate the level of performance of the proposed dimensionality reduction method we next apply it to the problem of nonlinear regression. Consider a fixed sample of n input points $X=\{x_1,\ldots,x_n\}$ in R^D, along with target values $Y=\{y_1,\ldots,y_n\}$ in R. Our goal is to estimate a functional dependence: $y=f(x)$ in a way, that allows efficient computation of a predicted output $y_{test}=f(x_{test})$ for an input x_{test} at the testing stage.

Here we consider dimensionality reduction as a way to cope with the challenges of the high dimensional input space of regression tasks. The basic idea is to perform the dimensionality reduction of the input space as a preprocessing step that preserves specific information about the target function. A proper dimensionality reduction results in a low dimensional space such that the target function still may be predicted based on the features lying in that subspace. Such a prediction should then be performed by a regression technique. This prediction will be computationally efficient because it runs in the low dimensional subspace. However a high accuracy can be achieved by combining a dimensionality reduction and a regression method. For instance, linear dimensionality reduction can be done in step 1 to achieve an informative low dimensional subspace in which a more involved nonlinear prediction technique can be applied. This type of two step algorithm constitutes a nonlinear regression method that benefits from fast linear operation in the initial input space and a precise regression in the reduced subspace that still performs fast and accurately.

Below we describe a simple nonlinear regression approach which exhibits high accuracy in terms of the testing set error while remaining highly computationally efficient. This is done by running the MIDR to reduce the input dimensionality so as to obtain a smaller subspace, that still retains maximal information about the target values.

Then, function g is approximated by a multinomial function P of degree l:

$$g(x)\approx P(Ax)=P(w)$$

Input: A set of n pairs: $(x_i,y_i)\in R^D\times R$, reduced dimension d.

Output: A linear projection A of input space $X\subset R^D\rightarrow AX\subset R^d$ that preserves maximal information about target Y based on the cost function $C(A)=I_{mean}(AX;Y)$

Method:

- Set an initial value for A.
- Apply a Conjugate-Gradient optimization to find the maximum of $C(A)$:

$$C(A)=\frac{d}{n(n-1)}\sum_{i\neq j}\log(\|A(x_i-x_j)\|^2)-\frac{d+1}{n(n-1)}\sum_{i\neq j}\log(\|A(x_i-x_j)\|^2+\|y_i-y_j\|^2)$$

using an analytical expression for the gradient:

$$\frac{\partial C(A)}{\partial A}=\frac{d}{n(n-1)}\sum_{i\neq j}\frac{A(x_i-x_j)(x_i-x_j)^\top}{\|A(x_i-x_j)\|^2}-\frac{d+1}{n(n-1)}\sum_{i\neq j}\frac{A(x_i-x_j)(x_i-x_j)^\top}{\|A(x_i-x_j)\|^2+\|y_i-y_j\|^2}$$

Fig. 1. The Mutual Information Dimensionality Reduction (MIDR) algorithm.

$$P(w)=\sum_{i_1+i_2+\cdots+i_d\le l} c_{i_1 i_2 \ldots i_d} w_1^{i_1} w_2^{i_2} \ldots w_d^{i_d}$$

The approximation is done by fitting coefficients in the L_2 norm:

$$\hat{P}=\arg\min \|Y-P(w)\|_2$$

This minimization problem is a linear problem as a function of the polynomial coefficients and therefore it can be easily solved. Values of the intended intrinsic dimension d and the multinomial degree l are determined by the cross-validation.

Of course, the regression in the resulting linear subspace may be performed by approaches other than polynomial regression. More powerful techniques such as GPR or SVM may be applied. However they are usually more computationally hard. In this performance vs. complexity tradeoff we chose the polynomial regression because it performs well and may be computed fast. In particular it does not require storing any subset from the training set for the test stage computations. Below we empirically demonstrate that in the output subspace of MIDR a simple polynomial regression achieves roughly the same or better results as an involved GPR technique.

The MIDR in this case can be viewed as a generalization of projection pursuit [15]. In fact, in the projection pursuit regression (PPR) the residual variance is brought to a minimum, whereas MIDR maximizes the mutual information between the predictor function and the target values. Therefore the PPR method is best suited for Gaussian variables, whereas MIDR has no inherent limitations for the explored variables distributions. We denote the nonlinear regression algorithm based on the MIDR dimensionality reduction method followed by a polynomial regression model as the 'Mutual Information Polynomial Regression' (MIPR) algorithm.

5. An application to classification

In this paper until now we focused on efficient nonlinear regression algorithms. The proposed mutual information based linear dimensionality reduction can be also utilized for classification. The problem of classification differs from regression in terms of the domain of the target values. Namely, consider for a fixed sample of n input points $X=\{x_1,\ldots,x_n\}$ in R^D: their target values are $C=\{c_1,\ldots,c_n\}$ in a discrete set Ω, as opposed to R in the case of regression. Here also our goal is to estimate a functional dependence: $c=f(x)$ in a way, that allows for efficient computation of a predicted output $c_{test}=f(x_{test})$ for an input x_{test} at the testing stage. A standard method for classification is the k NN technique, in which a test input x_{test} is classified by finding its k-th nearest neighbor out of the train inputs $\hat{x}_{train}$ and assigning $c_{test}=c_{train}(\hat{x}_{train})$. However, if the inputs belong to high-dimensional space the nearest neighbor search becomes computationally prohibitive. A useful technique to overcome the curse of dimensionality is to perform a dimensionality reduction in the input subspace prior to the NN classification. In fact, the LDA technique does exactly the same thing by projecting the inputs in the subspace spanned by eigenvectors of a combination of intra–inter covariance matrices [16]. In order to apply MIDR to the task of classification we need to estimate the mutual information between the continuous inputs and discrete target values. Fortunately, this can be done easily by means of conditional entropies:

$$I(X;C)=H(X)-H(X|C)=H(X)-\sum_{c\in\Omega} p(c)H(X|c) \tag{12}$$

In the above equation c denotes each possible value of the target variable, $p(c)$ is the probability for the target variable to have value c, and $H(X|c)$ is the entropy of the input random vector restricted to values such that their output equals c, the so-called in-class entropy. Thus the MeanNN estimator can be applied to estimate the mutual information between the continuous and the discrete variables. Therefore the optimal linear dimensionality reduction matrix A is found by maximizing the MeanNN estimate of the mutual information.

$$\begin{aligned}\arg\max_A &\left(H(AX)-\sum_{c\in\Omega} p(c)H(AX|c)\right)\\ &\approx \arg\max_A \left(\frac{1}{n(n-1)}\sum_{i\ne j}\log(\|A(x_i-x_j)\|^2)\right.\\ &\left.-\sum_{c\in\Omega}\frac{p(c)}{n_c(n_c-1)}\sum_{i_c\ne j_c}\log(\|A(x_{i_c}-x_{j_c})\|^2)\right)\end{aligned} \tag{13}$$

where n_c is the number of samples having target value c. Clearly $n=\sum_{c\in\Omega} n_c$. The gradient of the MeanNN estimator of the mutual information with respect to A is given by

$$\begin{aligned}\frac{\partial I}{\partial A}=&\left(\frac{2}{n(n-1)}\sum_{i\ne j}\frac{A(x_i-x_j)(x_i-x_j)^\top}{\|A(x_i-x_j)\|^2}\right.\\ &\left.-\sum_{c\in\Omega}\frac{2p(c)}{n_c(n_c-1)}\sum_{i_c\ne j_c}\frac{A(x_{i_c}-x_{j_c})(x_{i_c}-x_{j_c})^\top}{\|A(x_{i_c}-x_{j_c})\|^2}\right)\end{aligned}$$

A 1NN classification constitutes the last stage of the classification approach, here dubbed 'Mutual Information Classification' (MIC).

6. An application to semisupervised classification

In the previous section we showed how the mutual information based dimensionality reduction uses train points x_{train} along with their labels c_{train} to obtain a reduced complexity classification for the test data x_{test}. In some cases, however, it is beneficial to take into account the test data points during dimensionality reduction stage [17]. Methods that make use of the unlabeled test data for training belong to the domain of semisupervised learning. In this section we show how the mutual information dimensionality reduction works in the semisupervised setting.

Obviously, we cannot directly apply the optimization process (13) to find the reduction matrix A because we do not know the labels c_{test} of the unlabeled data. To overcome this obstacle we need to estimate these missing labels in a reliable manner first, and then find the reduction matrix by the above optimization procedure. In the framework of the reduced complexity classification we work only with projected data; therefore the quality of labels estimation depends on the reduction matrix chosen.

These considerations naturally lead to an iterative alternating optimization scheme of the proposed algorithm of semisupervised mutual information based classification. Each iteration has two stages. First we estimate the missing labels c_{test} for the projected test data Ax_{test} using the current estimate of the reduction matrix A. Second, we find an update to the projection matrix A by maximization (13) using all the data X. In this second stage we do not distinguish between train data points x_{train} labeled by their given labels c_{train} and the test points x_{test} labeled by their previously estimated labeled c_{test}. The iterations continue until a stopping criterion is satisfied.

The only question left is how we estimate the missing labels of the projected test data in the first stage of each iteration. There are obviously many ways to do this. For example, we could just obtain a simple classification of the projected test data based on the projected train data, by, say, 1NN classification. This, however, is suboptimal in the typical setting where semisupervised

methods are advantageous, namely in situations in which we have only a few train data and a lot of test points.

In these situations we would like to exploit the intrinsic structure of the test data to infer their labels better. For example, if the test data are composed of clusters, test points belonging to the same cluster will tend to have same labels, even if some trivial classification based on distance to the closest train data points would classify them differently. In fact, test label assignment can be seen as a clustering in which each cluster is associated with one of the classes. Practically, this corresponds to the clustering of both train and test data points at once, where train data points of same labels are preassigned to be in the same clusters.

There are a variety of clustering algorithms that can be casted into our framework. We found a non-parametric information clustering (NIC) [18] to be highly appropriate to the proposed scheme. This algorithm produces clustering that maximizes the mutual information between data points and assigned labels. The NIC does not require any additional information about data distribution since it uses the non-parametric MeanNN estimator for entropy estimation as a score function

$$I(X;C) \approx S_{NIC}(C) = \sum_j \frac{1}{n_j - 1} \sum_{i \neq l | c_i = c_l = j} \log \|x_i - x_l\| \tag{14}$$

where the summation runs for possible cluster numbers j and n_j is the number of points assigned to each cluster j. We can easily modify the approach for our needs of clustering all the data by fixing the labels of the train points and only updating the labels of test points. We apply this modified clustering on all the projected data $AX = \{AX_{train}, AX_{test}\}$ to get labels as a maximizer of the mutual information:

$$\arg\max_C \sum_j \frac{1}{n_j - 1} \sum_{i \neq l | c_i = c_l = j} \log \|Ax_i - Ax_l\| \tag{15}$$

where the maximization is done over all the label sets C which coincide with the given labels on the training subset. Hence the whole algorithm is composed of two-stage iterations of mutual information maximization. In the first stage of an iteration, we perform a discrete optimization of the mutual information between current labels and projected data to obtain a reliable guess about possible labels of test points (15). In the second stage we calculate the mutual information based linear dimensionality reduction between labels obtained in the first stage and data points in order to obtain optimal projection matrix A (13). We continue these two-stage iterations until a stopping criterion is satisfied. The algorithm outputs an optimal dimensionality reduction matrix A and optimal test labels c_{test}. On each iteration the algorithm monotonically increases the score $I(AX;C)$. In the first stage A is kept fixed and we optimize $I(AX;C)$ as a function of C. In the second stage C is kept fixed and we optimize $I(AX;C)$ as a function of A. This monotonic score improvement guarantees converges (to a local optimum) of the algorithm. We dub the proposed algorithm Semisupervised Mutual Information based Classification (SemiMIC).

7. Experimental results

First, we compared MIPR performance with other alternatives for the regression in the resulting subspace, such as Gaussian process regression, considered today to be the state-of-the-art in regression. The GPR technique models the outputs as Gaussian processes with a few hyperparameters, see e.g. [19]. We also used the GPR on the full input space to evaluate the possible loss of accuracy due to dimensionality reduction. We applied canonical correlation analysis as a representative alternative supervised linear dimensionality reduction method. The CCA method finds an optimal subspace by maximization of the correlation between the input and output spaces [1]. In addition we compared the proposed approach with the PLS regression.

We tested the method on a number of standard data sets from the *Data for evaluating learning in valid experiments* (*Delve*) collection.[1] We used eight different data sets from three different data set families. The first family, called *bank*, describes queues of customers in a series of banks. The other two families *kin* and *pumadyn* were generated by two synthetic robotic arms. Each family contains eight-dimensional and 32-dimensional input spaces and the output space is one-dimensional. In our experiments we used training sets of size 1000. For each run we used different splits of the data in the training, validating (1000 samples) and testing (1000 samples) data sets. The results for the test set appear in Fig. 2.

The MIPR algorithm produces comparable results to the GPR technique using the full input space and in general performs better than other methods. In particular, mutual information linear dimensionality reduction leads to a smaller error than the CCA method followed by the same polynomial regression (the method is referenced in the graph as CCA PR). On the other hand, application of a more elaborate and computationally intensive GPR technique after the MIDR stage does not improve the results.

We next present various performance aspects and a parameter sensitivity analysis of the proposed MIPR algorithm on the 'puma-8nh' data set. The optimal dimensional reduction was $d=4$, see Fig. 3 (top). The optimal multinomial degree was 3 in all runs, see Fig. 3 (bottom). Note that the computational advantage of the MIPR method is significant. For the size 1000 test set the running time for the proposed scheme was approximately three orders of magnitude faster than the GPR: 0.0021 s per run for the method vs. 1.83 s per run for the GPR. It is worth pointing out that we compared the running times using our matlab non-optimized code vs. the GPR function from the standard optimized gpml package.[2] The training time for all the methods we examined was comparable: the average training time for our method was 1 min and for GPR it was 6 min. We ran our comparison on a computer with an Intel(R) Xeon(R) 2.67 GHz processor, 3 GB RAM. It is clear that by making a set of routine optimizations even faster execution times can be achieved in the case of the MIPR method and its computational advantage over GPR will be even more significant.

In addition, we conducted numerical experiments in the field of reduced-complexity classification. We evaluated the performance of the MIC algorithm on the standard data sets from the UCI repository [20]. To assess the contribution of the MIDR to the classification accuracy we applied two other mutual information based dimensionality reduction techniques MMIP [8] and MMIFE [9] followed by 1NN classification. We also applied the LDA algorithm. The dimensionality reduction to two-dimensional subspace was carried out ($d=2$). Finally we applied 1NN classification in the full input space to provide a baseline for method accuracy assessment. To illustrate the ability of the proposed algorithm to utilize fully the information contained in the training set we used a relatively small portion (10%) of the data for training, and the testing was carried out on the rest of points. The results appear in Fig. 4.

The proposed algorithm MIC outperforms other classification schemes based on dimensionality reduction such as LDA, MMIP, MMIFE for all the data sets. Moreover, for most of the data sets MIC did not perform less well than 1NN classification that benefits from the full input space data. All the above emphasizes the fact that the MeanNN estimator for the entropy (and hence for the mutual information) produces a qualitative measure of the differential entropy leading to optimal linear dimensionality reduction.

[1] http://www.cs.toronto.edu/~delve.
[2] http://www.gaussianprocess.org/gpml/code/.

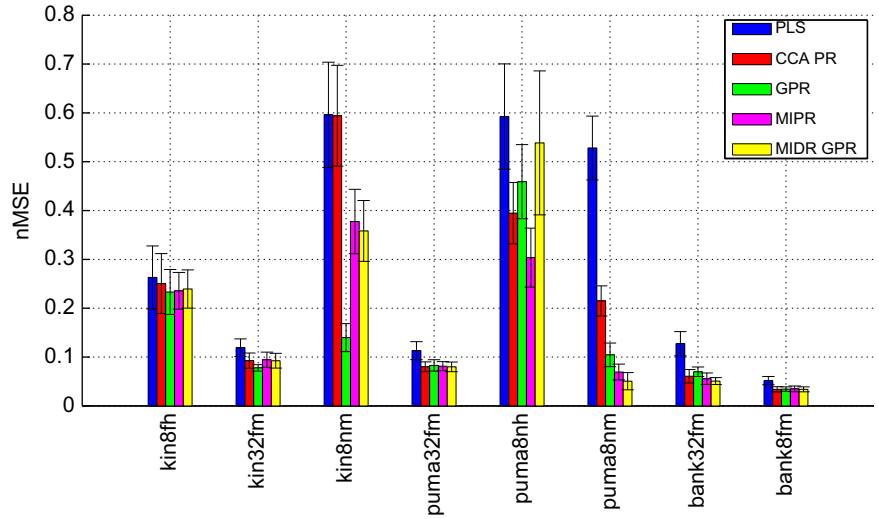

Fig. 2. Performance of several regression methods on Delve data sets. One thousand training samples, 1000 validation samples, 1000 testing samples. Statistics are shown for 10 repetitions.

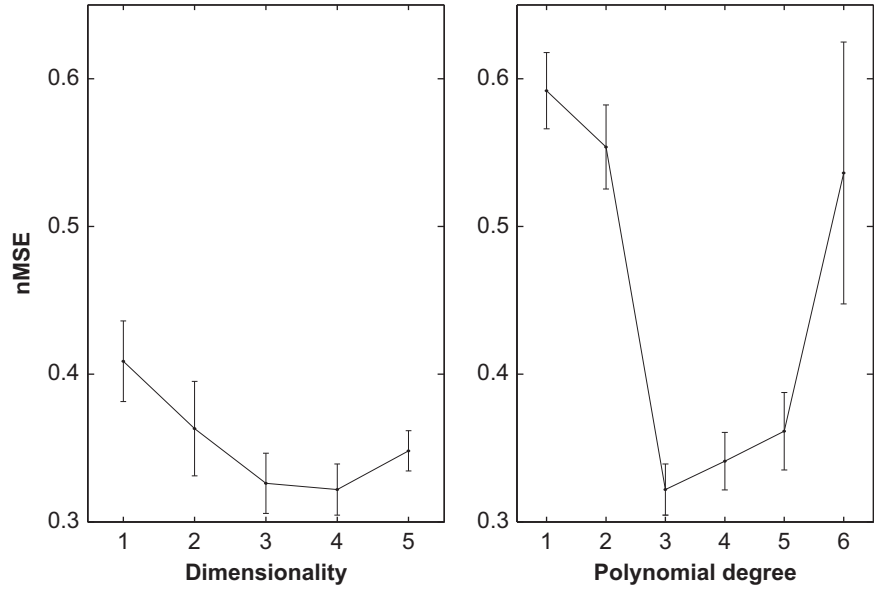

Fig. 3. MIPR parameters sensitivity. Variability of nMSE on the test set due to dimensionality of the reduced subspace (left), multinomial degree (right). Puma-8nh data, 1000 training samples, 1000 testing samples. Statistics are shown for 10 repetitions.

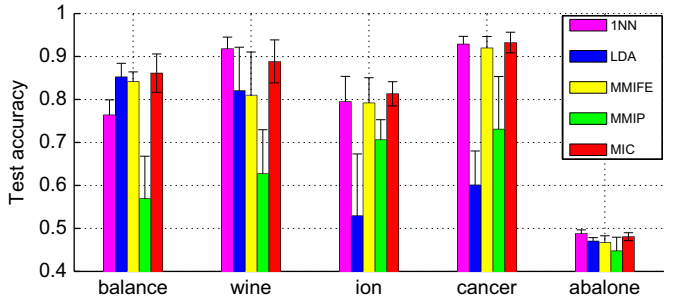

Fig. 4. Comparison of classification test accuracy. UCI classification data sets: balance, wine, ionosphere, cancer (breast) and abalone. Training is 10% of a data set, testing is 90%. Statistics are shown for 100 random repetitions. Reduction to two-dimensional subspace ($d=2$).

Finally we numerically evaluated the performance of the proposed reduced complexity semisupervised classification scheme SemiMIC. We used standard UCI repository data sets for benchmarks. We compared the performance of the proposed method SemiMIC and a number of supervised classification schemes. First we applied the 1NN method that runs in the full space. Then we used two methods that run in the reduced subspace. For this purpose we used the NCA method [21], that is considered to be a state-of-the-art reduced complexity algorithm, and our novel supervised method MIC. To demonstrate maximally the effect of addition of semisupervised learning we initialized the SemiMIC by the reduction matrix and labels obtained by MIC. We used a very small portion (5%) of data as labeled training points and the rest of points were used as unlabeled test data. The results appear in Fig. 5.

In these experiments the proposed semisupervised learning scheme SemiMIC achieved better performance than the other supervised methods running in the reduced space. In the most difficult classification task of the 'abalone' data set the SemiMIC advantage was very significant. For most of the cases the SemiMIC method even outperformed the supervised 1NN method that makes use of data in full dimensional space. These results

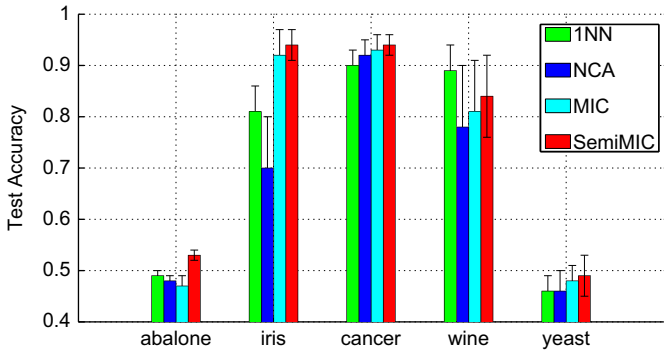

Fig. 5. Comparison of semisupervised classification test accuracy. UCI classification data sets: abalone iris, cancer, wine, yeast. Training is 5% of a data set, testing is 95%. Statistics are shown for 10 random repetitions. Reduction to two-dimensional subspace ($d=2$).

illustrate the positive impact of the addition of semisupervised capabilities to the proposed mutual information based classification schemes.

8. Conclusion

This paper makes several contributions. First, we introduced a supervised linear dimensionality reduction algorithm MIDR based on maximization of mutual information between the subspace of inputs and the outputs values that produces optimal multidimensional result subspaces. We demonstrated a simple nonlinear regression algorithm MIPR that is based on MIDR. The regression method achieves essentially the same performance as the state-of-the-art regression algorithm GPR and in general performs better than other methods. The MIPR algorithm has a significant computational advantage in the test stage, in that it is three orders of magnitude faster than GPR. The MIPR structure lends itself well to fast implementation. We also provided an application for reduced-complexity classification. This leads to superior results compared to the standard methods such as LDA. The classification framework highlights the advantage of MIDR over other mutual information based dimensionality reduction schemes such as MMIP and MMIFE. Finally, we defined a semi-supervised version for reduced complexity classification. The resulting semisupervised method showed an improved performance compared to tested supervised methods.

References

[1] H. Hotelling, Relations between two sets of variates, Biometrika (1936) 321–377.
[2] P. Geladi, B.R. Kowalski, Partial least-squares regression: a tutorial, Anal. Chim. Acta 185 (1986) 1–17.
[3] K. Fukumizu, F. Bach, M.I. Jordan, Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces, J. Mach. Learn. Res. 5 (2004) 73–99.
[4] A. Globerson, S. Roweis, Metric learning by collapsing classes, in: Advances in Neural Information Processing Systems, vol. 18, 2006.
[5] K. Weinberger, J. Blitzer, L. Saul, Distance metric learning for large margin nearest neighbor classification, in: Advances in Neural Information Processing Systems, vol. 18, 2006.
[6] F. Rossi, A. Lendasse, D. François, V. Wertz, M. Verleysen, Mutual information for the selection of relevant variables in spectrometric nonlinear modelling, Chemometrics Intell. Lab. Syst. (2007) 215–226.
[7] K. Torkkola, Feature extraction by non-parametric mutual information maximization, J. Mach. Learn. Res. (2003) 1415–1438.
[8] K.D. Bollacker, J. Ghosh, Mutual information feature extractors for neural classifiers, in: Proceedings of the International Conference on Neural Networks (ICNN '96), 1996, pp. 1528–1533.
[9] J.M. Leiva-Murillo, A. Artes-Rodriguez, Maximization of mutual information for supervised linear feature extraction, IEEE Trans. Neural Networks (2007) 1433–1441.
[10] L. Faivishevsky, J. Goldberger, ICA based on a smooth estimation of the differential entropy, in: Advances in Neural Information Processing Systems, vol. 21, 2009.
[11] L. Kozachenko, N. Leonenko, On statistical estimation of entropy of random vector, Probl. Inf. Transm. 23 (2) (1987) 95–101.
[12] J.D. Victor, Binless strategies for estimation of information from neural data, Phys. Rev. E (2002) 051903.
[13] Q. Wang, S.R. Kulkarni, S. Verdu, Divergence estimation for multidimensional densities via k-nearest-neighbor distances, IEEE Trans. Inf. Theory (2009) 2392–2405.
[14] H. Singh, N. Misra, V. Hnizdo, A. Fedorowicz, E. Demchuk, Nearest neighbor estimates of entropy, Am. J. Math. Manage. Sci. (2003) 301–321.
[15] J.H. Friedman, J.W. Tukey, A projection pursuit algorithm for exploratory data analysis, IEEE Trans. Comput. (1974) 881–890.
[16] R.A. Fisher, The use of multiple measurements in taxonomic problems, Annu. Eugen. (1936) 179–188.
[17] J. Peltonen, J. Goldberger, S. Kaski, Fast semi-supervised discriminative component analysis, in: IEEE Workshop on Machine Learning for Signal Processing, 2007.
[18] L. Faivishevsky, J. Goldberger, A nonparametric information theoretic clustering algorithm, in: Proceedings of the 27th International Conference on Machine Learning, 2010.
[19] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, The MIT Press, Cambridge, MA, USA, 2006.
[20] A. Asuncion, D.J. Newman, UCI Machine Learning Repository, 2007.
[21] J. Goldberger, S. Roweis, G. Hinton, R. Salakhutdinov, Neighbourhood components analysis, in: Advances in Neural Information Processing Systems, vol. 16, 2004.

Lev Faivishevsky received his B.Sc. in Applied Mathematics and Physics from the Moscow Institute of Physics and Technology (1998), M.Sc. in Applied Mathematics and Computer Science from the Weizmann Institute (2002) and MBA from the Technion, Haifa (2002). From 2007 he is a Ph.D. student the Bar-Ilan University, School of Engineering.

Jacob Goldberger received B.Sc. in Computer Science from Bar-Ilan University, M.Sc. in Mathematics and Ph.D. in Electrical Engineering both from the Tel-Aviv University (1998). He was a postdoctoral fellow at the Weizmann Institute (2000) and at the University of Toronto (2003). In 2004, Jacob Goldberger joined the Bar-Ilan University, School of Engineering.