# Maximum Gradient Dimensionality Reduction

Xianghui Luo
Department of Computer Science
University of Waikato
Hamilton 3240, New Zealand
Email: uoxluo@gmail.com

Robert J. Durrant
Department of Mathematics and Statistics
University of Waikato
Hamilton 3240, New Zealand
Email: bobd@waikato.ac.nz

***Abstract*—We propose a novel dimensionality reduction approach based on the gradient of the regression function. Our approach is conceptually similar to Principal Component Analysis, however instead of seeking a low dimensional representation of the predictors that preserve the sample variance, we project onto a basis that preserves those predictors which induce the greatest change in the response. Our approach has the benefits of being simple and easy to implement and interpret, while still remaining very competitive with sophisticated state-of-the-art approaches.**

## I. Background and Introduction

Dimensionality reduction is a key component in the Statistical Analysis and Machine Learning toolbox. Modern high dimensional datasets can comprise tens of thousands of features (variables) and it is fairly common for many of these features to be irrelevant for a particular learning task – for example, gene array datasets contain large amounts of unstructured noise. When there are irrelevant features, extraction of informative variables is vital to the good performance of machine learning algorithms. Indeed, many learning algorithms involve some form of feature extraction, which is a form of dimensionality reduction, as an integral process. For example, linear regression algorithms select the variable(s) most correlated with the response and then regress onto these. Furthermore, for practical and computational reasons we frequently want to reduce the number of features before passing data to a model for learning or prediction. As well as reducing the time and space complexity of learning or prediction, and improving interpretability of models, a separate preprocessing dimensionality reduction stage essentially acts as a regularization for the machine learning model and can improve its generalization performance, especially for situations where the number of training instances are small compared to the number of features. In this paper, we propose a novel dimensionality reduction approach called *Maximum Gradient Dimensionality Reduction* (MGDR), based on the idea that a subspace of the predictor variables which induces large changes in the response contains all of the important information for a prediction task.

### A. Review of Existing Approaches

Research on dimensionality reduction has a long history, and a variety of motivations, and a wide range of learning tasks, have produced many dimensionality reduction approaches including linear and nonlinear approaches [1]. Compared to nonlinear approaches, linear dimensionality reduction methods usually have more straightforward geometric interpretations and are more stable. Moreover many commonly used nonlinear methods, such as kernel PCA, are linear approaches applied following a nonlinear transformation of the original training data. In this paper therefore we focus on linear dimensionality reduction schemes and below follows a brief review of some of the most common methods.

Perhaps the oldest and most widely used linear approach is Principal Component Analysis (PCA) [2]. PCA is motivated by preserving, in a low-rank approximation of the data, as much of the sample variance of the predictor variables as possible. However, PCA is an unsupervised approach and does not take into account information about the response, and the predictors with the largest variance may not be the most informative for a particular learning problem. In particular, the variance of the predictors depends on scaling and – for example – a simple change of units will result in a change of the variance of different predictors. Thus, PCA is sensitive to different representations of the same data and can perform very badly if used blindly. To overcome these shortcomings of PCA, Partial Least Squares (PLS) maximizes the correlation between the predictor and the response as well as the variance of the predictor. It has been shown that PLS is a compromise between Principal Component Regression and linear regression [3]. An alternative approach is Fisher's Linear Discriminant (FLD) which is a supervised method that has a nice objective, which is to maximize the ratio between the between-class variance and the within-class variance in a projected space. It tries to find a projection such that groups or classes are well separated. In principle, FLD should work better than PCA for classification tasks. However, in reality this is not always the case [4], especially for the situations where the number of observations are relatively small compared to the number of features. The reasons behind this may well be that the estimation of within-class variance and between-class variance is challenging, especially with a small sample. Independent Component Analysis (ICA)[5] assumes that data are generated by mixing some independent latent variables and it searches for projections such that the components along these projections are statistically independent. Canonical Correlation Analysis (CCA)[6], [7] searches for a pair of linear subspaces such that the cross-correlation between the predictor and response are maximized following projection to

these subspaces. However, the number of components CCA can find is limited by the dimension of the response. Slow Feature Analysis (SFA)[8] is a useful method for image data. It is built on the 'slowness principle', which says the important characteristics such as the identity of an underlying object should change very slowly, in contrast to the individual pixel values which can change very rapidly. SFA thus seeks the slow changing projections, but is not the foundation of any state-of-the-art applications and the reason may be that the slowness assumption is too strong. Sparse regularization and model selection methods such as *Lasso* [9] can be used for dimensionality reduction. However, such sparse regularization methods only works well if the given representation is a sparse representation for the underlying model – that is if the model is sparse in the given representation of the data. Distance metric learning algorithms [10], [11] learn a distance metric by maximizing the accuracy of prediction. Learning a distance metric is equivalent to learning a projection matrix and by discarding the least useful components we obtain a dimensionality reduction method.

The above mentioned methods all have extensions and variants. For example, there are many supervised variants of PCA, such as [12], [13]. In the past two decades the statistics community has quite some novel methods based on the idea of Sufficient Dimension Reduction (SDR) [14]. A projection of the original data is said to be *sufficient* if it is as informative as the original data. In other words, the response only depends on a subspace of the original data space. Such a subspace is call the *effective dimension reduction* (EDR) space. This assumption indeed seems reasonable in real world applications where only some of the variables are important (or relevant) for the level of the response. The important work in SDR includes sliced inverse regression (SIR)[15], sliced average variance estimation (SAVE)[16], minimum average variance estimation (MAVE)[17], and kernel dimension reduction (KDR)[18]. Most related to our paper is a line of work based on derivatives [19], [20], [21], [22]. These are based on the idea that the derivatives of the regression functions or the conditional densities lie in the EDR space. After doing an eigendecomposition on the sum of the cross-products of the derivatives at each data point, a projection matrix is obtained by retaining the most important eigenvectors as columns.

### B. Our Approach

In this paper, we propose a novel dimensionality reduction approach called *Maximum Gradient Dimensionality Reduction* (MGDR). It is based on the idea that the directions in the original data space that induce large changes in the response are the most informative directions for a prediction task. Thus these directions represent the most informative features, and by projecting to these informative features, we obtain dimension-reduced data that retain most of the information relevant to the response. MGDR differs from the gradient-based SDR approaches in the sense that the SDR approaches try to obtain an EDR, while MGDR tries to retain the most informative features. They are also different in the sense that gradient-based SDR approaches require an eigendecomposition of the sum of the cross-products of the gradients in obtaining the projection matrix.

## II. Preliminaries

All learning models basically try to learn a function which maps the input to the output from a finite sample, such that the error on unlabelled data is small. The output is typically one-dimensional: a (real) scalar value for regression tasks and a (categorical) class label for classification tasks. The (predictor) input is usually a high dimensional variable, the components of which can consist of numerical or categorical quantities. For simplicity and concreteness, here we focus on a multi-variable linear regression model where the input is a real-valued vector to gain some insight into ways of finding the most informative features.

Let $x \in \mathbb{R}^D$ be the input vector. Let $y \in \mathbb{R}$ be the output. A linear regression model assumes the output is a linear function of the input. That is

$$E[y|x] = f(x) = w^T x + b, \tag{1}$$

where $E[y|x]$ is the expected value of $y$ given $x$. The regression task is to learn the function $f(x) = w^T x + b$ for accurate prediction. Given $N$ observations $(x_i, y_i), i = 1, \cdots, N$, the *Least Squares* method minimizes the squared loss between predictions and known outputs to obtain the following closed form for an optimal solution:

$$\hat{w} = (X^T X)^{-1} X^T (Y - \hat{b}\mathbf{1}), \quad \hat{b} = \bar{y} - \bar{x}^T \hat{w}, \tag{2}$$

where $X = (x_1, \cdots, x_N)^T$ is the *design matrix*, $Y = (y_1, \cdots, y_N)^T$, and $\bar{x}$ and $\bar{y}$ are the means of $\{x_i\}_{i=1}^N$ and $\{y_i\}_{i=1}^N$ respectively.

For this linear regression problem, there is one direction in the original data space that captures all of the information relevant to the response $y$, namely the feature along the direction of $w$. If $D$ is large, then running regression on the original data space is a waste of computational resources since – in principle – we need only search in this one dimensional space (or indeed any proper subspace of $\mathbb{R}^D$ containing $w$) to find this feature. Moreover working with the $D$-dimensional inputs, the prediction accuracy also may not be satisfactory due to the presence of irrelevant variables. Thus it would be desirable to first discard the variables that are unlikely to be informative and narrow down to the potentially useful ones before applying our learning algorithm. A variety of usefulness criteria have led to a variety of dimensionality reduction approaches, some of which we reviewed earlier, but here we simply start with the straightforward observation that any irrelevant variables do not cause any change in the response, so they should not affect the predicted response either. In the linear regression model, all the directions orthogonal to $w$ do not induce any change in $y$. To extract the informative features, we propose to use the gradient in that direction as a usefulness measure. Hence we devise the *Maximum Gradient Dimensionality Reduction* (MGDR) approach.

## III. ALGORITHM

In order to devise a dimensionality reduction method based on the gradients, we begin with the assumption that the response is a linear function of the predictor variables $y = w^T x + b$, where $y \in \mathbb{R}$ is the response, $x \in \mathbb{R}^D$ is the predictor variable, $w \in \mathbb{R}^D$, $b \in \mathbb{R}$. Suppose we have the training observations $\{x_i, y_i\}_{i=1}^N$. We consider the following vectors

$$\frac{x_i - x_j}{y_i - y_j}, \quad i = 1, \cdots, N, \quad j = 1, \cdots, N, \quad i < j. \tag{3}$$

These vectors are in the $D$-dimensional data space and represent inverse gradients between particular predictor-response pairs in the training data. Note that the smaller $\|\frac{x_i - x_j}{y_i - y_j}\|$ is, the larger the rate of change in the response in that direction. According to our linear model, these vectors are equal to

$$\frac{x_i - x_j}{w^T(x_i - x_j)} = \frac{1}{\|w\| \cos\theta(w, x_i - x_j)} \frac{x_i - x_j}{\|x_i - x_j\|}, \tag{4}$$

where $\theta(w, x_i - x_j)$ is the (unknown) angle between the vectors $w$ and $x_i - x_j$. Note that the first term in (4) is a scalar, while the second is a unit norm vector, hence the magnitudes of the vectors (3) are determined by $\cos\theta(w, x_i - x_j)$. Furthermore, with a smaller $\theta(w, x_i - x_j)$, $x_i - x_j$ is more aligned with $w$, and $w$ is the (unknown) direction that captures all of the information relevant to the response $y$. Thus for our projection we will select vectors of the form $\frac{x_i - x_j}{y_i - y_j}$ with the smallest magnitude, since these are the most aligned with $w$. We construct the projection matrix in an iterative way, similar to Gram-Schmidt orthogonalization: First, we order the vectors (3) in increasing magnitude, select the one with least magnitude, and take the normalized sum of them as the first projection direction. Next we project the training data onto that vector and we regress $y$ on the components of $x$ in that direction to obtain the residual as the new $y$. Continuing, we subtract the projection of the whole training data from the original data to get our new predictor values $x$; Finally, the new $x$ and $y$ values are then used for the next iteration to obtain the next projection direction. The MGDR algorithm is presented in Fig. 1 in detail. The algorithm takes training data and outputs a dimensionality reduction projection matrix, which can be applied on training data and test data to reduce the dimensionality.

It is a straightforward analysis to obtain the time complexity of MGDR, which is $O(N^2D) + O(ND^2)$, that is, the same time complexity as for vanilla PCA.

### A. MGDR for binary classification

MGDR stems from the authors' further consideration of their M-PCA2 algorithm in [23]. In that work we consider a binary classification problem. Let $\{(x_i, y_i)\}_{i=1}^N$ be a set of labeled training data points, where for convenience we assume $x_i$ is a point in $\mathbb{R}^D$, and $y_i \in \{-1, +1\}$ is the class label, $\forall i \in \{1, \cdots, N\}$. Let $C_-$ and $C_+$ be the sets of indices of the data points that belong to class -1 and class +1 respectively, i.e. $C_- = \{i : y_i = -1\}, C_+ = \{j : y_j = +1\}$.

M-PCA2 constructs the vectors $z_{ij} = x_i - x_j, \forall i \in C_+, \forall j \in C_-$, such that $i \in \arg\min_{k \in C_+} \|x_k - x_j\|$ or $j \in \arg\min_{k \in C_-} \|x_i - x_k\|$ as a proxy for the margin between the two classes. M-PCA2 then runs eigendecomposition on these vectors to obtain a projection matrix for reducing dimension that approximately preserves the margin between classes.

**Maximum Gradient Dimensionality Reduction**

**input**: Training data $\{x_i, y_i\}_{i=1}^N$ and target dimension $K$
**for** $k = 1, \cdots, K$
  construct the vectors $\frac{x_i - x_j}{y_i - y_j}$ for $1 \le i < j \le N$
    and retain $N$ of them with the smallest magnitudes
    and denote these vectors by $\{z_i\}_{i=1}^N$
  let $R_k = \sum_{i=1}^N z_i / \| \sum_{i=1}^N z_i \|$
  regress $\{y_i\}_{i=1}^N$ on $\{x_i^T R_k\}_{i=1}^N$ to obtain $\{\hat{y}_i\}_{i=1}^N$
  let $y_i = y_i - \hat{y}_i$, $i = 1, \cdots, N$
  let $x_i = x_i - R_k R_k^T x_i$, $i = 1, \cdots, N$
**end**
$P = [R_1, \cdots, R_K]$
**output**: the projection matrix $P$

Fig. 1. The MGDR Algorithm

The relation between MGDR and M-PCA2 lies in the following fact. $z_{ij} = x_i - x_j$ is the difference vector of two nearest neighbors of different classes. For this classification problem, the label is discrete and can only take on two values, i.e. $\{+1, -1\}$. Thus, $y_i - y_j$ is always 2. By using these vectors $z_{ij} = x_i - x_j$, we are in fact choosing $\frac{x_i - x_j}{y_i - y_j}$ with the smallest magnitudes, just like MGDR.

### B. Interpretation of MGDR

It has been noted that the least squares estimate of the estimated weight vector satisfies the following identity [24]:

$$\hat{w}^{ls} = \frac{\sum_{i \neq j} (y_i - y_j)(x_i - x_j)}{\sum_{i \neq j} \|x_i - x_j\|_2^2}.$$

With a little algebraic manipulation, we obtain

$$\hat{w}^{ls} = \sum_{i \neq j} \frac{(y_i - y_j)^2}{\sum_{i \neq j} \|x_k - x_l\|_2^2} \frac{x_i - x_j}{y_i - y_j} = \sum_{i \neq j} \hat{w}_{ij} \frac{x_i - x_j}{y_i - y_j},$$

where $\hat{w}_{ij} := \frac{(y_i - y_j)^2}{\sum_{i \neq j} \|x_i - x_j\|_2^2}$. In this identity, the least squares estimate of the weight coefficient is a weighted sum of the "inverse gradient" $\frac{x_i - x_j}{y_i - y_j}$ and the weights $\hat{w}_{ij}$ are proportional to the squared change in the corresponding $y_i$ and $y_j$. In our maximum gradient method, the projection components are a sum of the unweighted "inverse gradients" of smallest magnitudes. Thus MGDR is equivalent to partially carrying out a least-squares regression with the additional constraint of uniformly weighting the gradients. Thus MGDR can be viewed intuitively as a regularization of the original least-squares problem.

## IV. EXPERIMENTS

In this section, we compare MGDR to PCA, PLS, Lasso and the state-of-the-art SDR approach LSGDR[22]. We measure the performance by the Mean Squared Error (MSE) on test

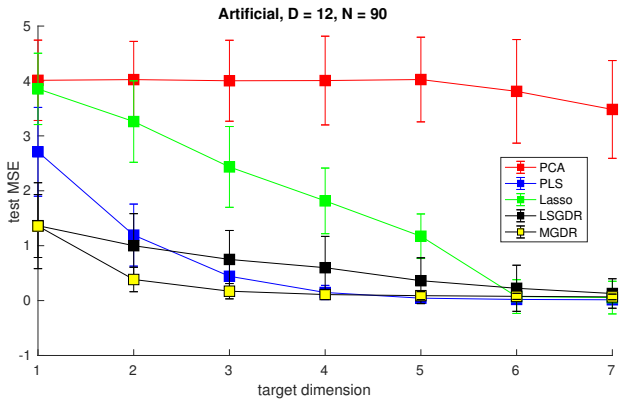


Fig. 2. Data model: $y = x_1 + \cdots + x_6 + \epsilon$, $x \in \mathbb{R}^{12}$, $x \sim \mathcal{N}(0, I)$, $\epsilon \sim \mathcal{N}(0, 0.01)$. Number of data points $N = 90$.

data for regression datasets and by classification test error for classification datasets. We test the performance of our approach on both synthetically generated datasets as well as on several real world datasets.

### A. Experiments on Synthetic Data

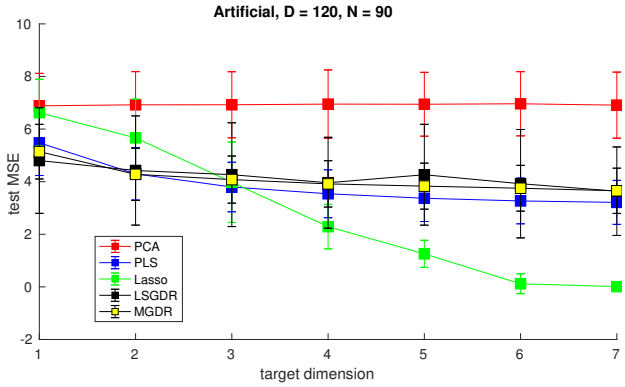


Fig. 3. Data model: $y = x_1 + \cdots + x_6 + \epsilon$, $x \in \mathbb{R}^{120}$, $x \sim \mathcal{N}(0, I)$, $\epsilon \sim \mathcal{N}(0, 0.01)$. Number of data points $N = 90$.

We generate two sets of artificial regression data. The first set of data is generated as follows. The dimension of the predictor is set to be $D = 12$. The true relation between $x$ and $y$ is $y = x_1 + \cdots + x_6 + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 0.01)$ is a random noise and only the first six features of the explanatory variable are informative for the response. We generate $N = 90$ data points according to the distribution $x \sim \mathcal{N}(0, I)$. This is the situation where the number of instances is much larger than the number of features. The second set of artificial data is generated in a similar way with $D = 120$ and all other parameters remaining the same. This is the case when the number of instances is a lot smaller than the number of features.

To test the performance of our approach, we generate 50 random splits of these artificial data into 60 training instances and we reserve 30 instances for testing. For a given target dimension $K$ and a given split, for each split we first run each dimensionality reduction algorithm on the training set to obtain a projection matrix and we reduce the dimension of the original dataset to $K$, or in the case of Lasso we instead restrict the number of learned weight coefficients to $K$. Then a linear regression model is built using the reduced training data and its MSE is obtained on the corresponding reduced test data. In this way, we obtain 50 MSE's for each combination of a dimensionality reduction approach and a target dimension $K$. We then compare the performance by the mean and standard deviation of MSE. In the experiment, we set the target dimensions to be one of $1, \cdots, 7$. The results are presented in Fig. 2 and Fig. 3.

For the dataset with $D = 12$ and $N = 90$, the experimental results show that MGDR significantly outperforms other approaches for small target dimensions $K = 1, 2, 3, 4, 5$, which are of the most practical interest. For larger target dimensions, there is no practical or statistically significant differences between MGDR and PLS, Lasso, LSGDR, and our approach is competitive with other approaches. These results clearly show the superiority of our approach MGDR for dimensionality reduction.

**Remark.** *It is interesting to observe the results for Lasso in Fig. 2. Lasso is a sparse model selection method. Its weakness is that its performance depends on the coordinate representation of the data. If the data is not sparse in a given representation, Lasso will not perform that well. This fact is indicated by our experimental results. Since the number of informative features is 6 in the given representation, Lasso performs significantly worse than our approach MGDR for target dimensions $K = 1, 2, 3, 4, 5$. Once the target dimension $K$ has reached 6, Lasso becomes the best performing approach.*

### B. Experiments on Public Regression Datasets

We further test the performance of our approach MGDR on several real world regression datasets: Spectra [25], Concrete Slump [26], Combined Cycle Power Plant [27], Body Fat [28]. The characteristics of these datasets are shown in Table I.

TABLE I
PUBLIC REGRESSION DATASETS

| name | source | #instances | #features |
|---|---|---|---|
| Spectra | [25] | 60 | 400 |
| Concrete Slump | [26] | 103 | 7 |
| Combined Cycle Power Plant | [27] | 9568 | 4 |
| Body Fat | [28] | 252 | 13 |

The experiments on these public datasets are carried out in a similar way. We generate 50 random partitions of the dataset into a training set and a test set, where two thirds are used for training and the remaining for testing. As before we choose a target dimension $K$, run a dimensionality reduction approach

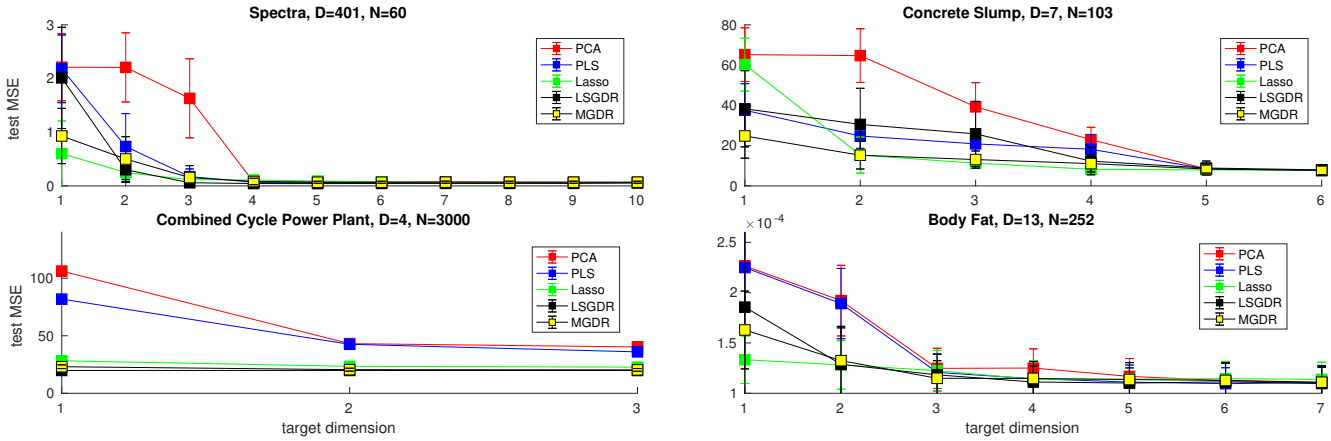


Fig. 4. Results on regression data

on the training data and then learn a linear regression model from this reduced training data and test on the reduced test data to obtain the test MSE. In this way, we obtain 50 MSE's for each combination of a dimensionality reduction method and a target dimension. The mean and standard deviation of MSE are then calculated.

For the Combined Cycle Power Plant data, since the number of instances is too large that LSGDR will be out of memory, we randomly choose 3000 instances for the experiment instead of the whole data. The Concrete Slump data have seven input features and three output variables. Here we choose one of output variable – 28-day Compressive Strength – as the response. The target dimensions are set to be $1, \cdots, 7$ for the Spectra data, $1, \cdots, 4$ for the Concrete Slump data, $1, 2$ for the Combined Cycle Power Plant data, $1, \cdots, 7$ for the Body Fat data. The detailed experimental results are shown in Fig. 4.

The results indicate that our approach is often the best method, is always competitive with and never significantly worse than the state-of-the-art methods.

### C. Experiments on Public Classification Datasets

We also test our maximum gradient method on several real world classification datasets. These are *colon* [29], *prostate* [30], *ovarian* [31], *leukemia* [32]. The characteristics of these datasets are shown in TABLE II. We cast the binary classification tasks as regression tasks with discrete targets. Since LSGDR does not work on these data, we compare MGDR to PCA, Lasso, PLS, in terms of test errors. The learning algorithm used is the $\ell_2$-regularized $\ell_2$-loss SVM implemented by *liblinear* [33] and we fitted the parameter using the whole datasets to provide a consistent baseline across all splits.

Each combination of a dimensionality reduction method, a target dimension $K$, and a dataset is fed 50 independent partitions of the dataset into a training set and a test set, where four fifths of the data was used for training and the remainder for testing, and the sampling was stratified to preserve class membership proportions. Hence 50 independent test errors are produced for each combination. These are then used to compute the mean and the standard deviation of the test errors for that combination. For each loop iteration for a particular dataset, the data splits were held constant. The target dimension is chosen to be one of $\frac{R}{4}$, $\frac{R}{2}$, $\frac{3R}{4}$, where $R$ is the rank of the training data matrix, which is roughly four fifths of the number of instances.

TABLE II
PUBLIC CLASSIFICATION DATASETS

| name | source | #instances | #features |
|---|---|---|---|
| colon | [29] | 22+40 | 2000 |
| prostate | [30] | 50+52 | 6033 |
| ovarian | [31] | 24+30 | 1536 |
| leukemia | [32] | 47+25 | 3571 |

The results on these data are presented in Fig. 5. We see that MGDR is again very competitive with the best approaches for these datasets.

## V. Discussion and Future Work

In this paper, we introduced a novel dimensionality reduction method MGDR. Unlike some competing methods it is straightforward to implement, with the same time complexity as PCA. Meanwhile compared to state-of-the-art methods, MGDR has very competitive performance and in our trials is often the best method. This makes it, we believe, a more practically appealing approach than some competing methods.

We are currently working on developing data-dependent theoretical guarantees for MGDR, although this is not straightforward. We are also examining ways in which to implement a non-linear variant of MGDR.

## References

[1] C. J. C. Burges, "Dimension Reduction: A Guided Tour," *Foundations and Trends in Machine Learning*, vol. 2, no. 4, pp. 275–365, 2009.
[2] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. Springer-Verlag New York, 2002.

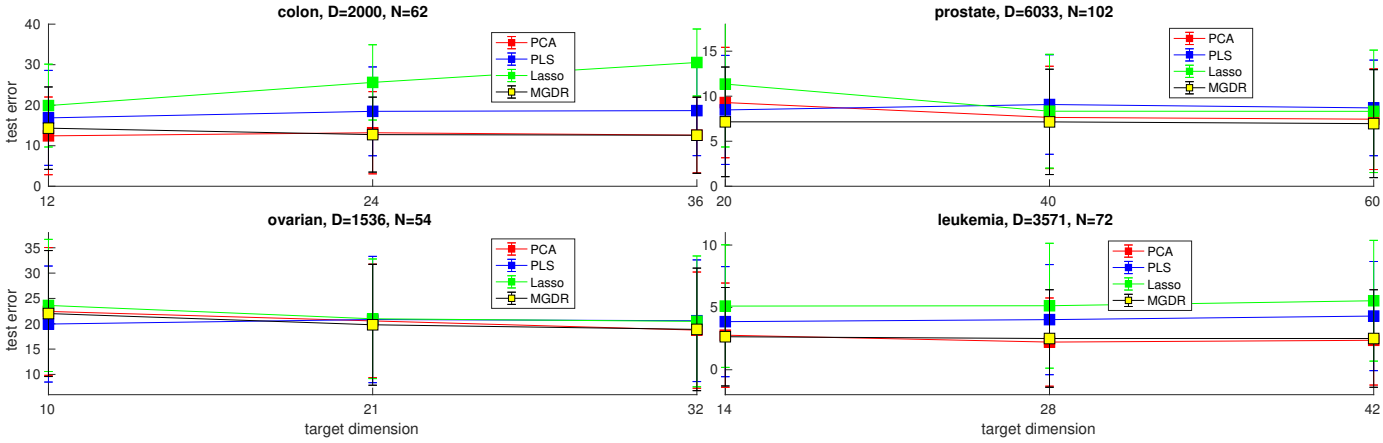


Fig. 5. Results on classification data


[3] M. Stone and R. J. Brooks, "Continuum regression: cross-validated sequentially constructed prediction embracing ordinary least squares, partial least squares and principal components regression," *Journal of the Royal Statistical Society. Series B*, vol. 52, no. 2, pp. 237–269, 1990.

[4] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228–233, 2001.

[5] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. JOHN WILEY & SONS, INC., 2001.

[6] H. Hotelling, "Relations Between Two Sets of Variates," *Biometrika*, 1936.

[7] D. R. Hardoon and J. Shawe-Taylor, "Convergence analysis of kernel Canonical Correlation Analysis: theory and practice," *Mach Learn*, vol. 74, pp. 23–38, 2009.

[8] L. Wiskott and T. J. Sejnowski, "Slow Feature Analysis: Unsupervised Learning of Invariances," *Neural Computation*, 2002.

[9] R. Tibshirani, "Regression Selection and Shrinkage via the Lasso," *Journal of the Royal Statistical Society B*, vol. 58, no. 1, pp. 267–288, 1996.

[10] J. Goldberger, S. T. Roweis, G. E. Hinton, and R. Salakhutdinov, "Neighbourhood Components Analysis," *Advances in Neural Information Processing Systems*, pp. 513–520, 2004.

[11] K. Q. Weinberger and G. Tesauro, "Metric Learning for Kernel Regression," in *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, 2007, pp. 612–619.

[12] N. Karampatziakis and P. Mineiro, "Discriminative Features via Generalized Eigenvectors," *Proceedings of The 31st International Conference on Machine Learning*, pp. 494–502, 2014.

[13] B. Kang, J. Lijffijt, R. Santos-Rodríguez, and T. De Bie, "Subjectively Interesting Component Analysis," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, 2016, pp. 1615–1624.

[14] K. P. Adragni and R. D. Cook, "Sufficient dimension reduction and prediction in regression," *Phil. Trans. R. Soc. A*, vol. 367, pp. 4385–4405, 2009.

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

[16] R. D. Cook, "SAVE: A method for dimension reduction and graphics in regression," *Communications in Statistics-Theory and Methods*, vol. 29, no. 9-10, pp. 2109–2121, 2000.

[17] Y. Xia, H. Tong, W. K. Li, and L.-X. Zhu, "An adaptive estimation of dimension reduction space," *J. R. Statist. Soc. B*, vol. 64, no. 3, pp. 363–410, 2002.

[18] K. Fukumizu, F. R. Bach, and M. I. Jordan, "Kernel dimension reduction in regression," *Annals of Statistics*, 2009.

[19] A. M. Samarov, "Exploring Regression Structure Using Nonparametric Functional Estimation," *Journal of the American Statistical Association*, vol. 88, no. 423, pp. 836–847, 1993.

[20] M. Hristache, A. Juditsky, J. Polzehl, and V. Spokoiny, "Structure adaptive approach for dimension reduction," *Annals of Statistics*, vol. 29, no. 6, pp. 1537–1566, 2001.

[21] K. Fukumizu and C. Leng, "Gradient-Based Kernel Dimension Reduction for Regression," *Journal of the American Statistical Association*, vol. 109505, pp. 359–370, 2014.

[22] H. Sasaki, V. Tangkaratt, and M. Sugiyama, "Sufficient Dimension Reduction via Direct Estimation of the Gradients of Logarithmic Conditional Densities," *JMLR: Workshop and Conference Proceedings*, vol. 45, pp. 33–48, 2015.

[23] X. Luo and R. J. Durrant, "Maximum Margin Principal Components," *arXiv preprint arXiv:1705.06371*, 2017.

[24] A. Gelman and D. K. Park, "Splitting a Predictor at the Upper Quarter or Third and the Lower Quarter or Third," *The American Statistician*, vol. 63, no. 1, pp. 1–8, 2009.

[25] J. H. Kalivas, "Two data sets of near infrared spectra," *Chemometrics and Intelligent Laboratory Systems*, vol. 37, pp. 255–259, 1997.

[26] I. C. Yeh, "Modeling slump flow of concrete using second-order regressions and artificial neural networks," *Cement and Concrete Composites*, vol. 29, pp. 474–480, 2007.

[27] P. Tüfekci, "Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods," *International Journal of Electrical Power & Energy Systems*, vol. 60, pp. 126–140, 2014.

[28] K. Penrose, A. Nelson, and A. Fisher, "Generalized Body Composition Prediction Equation for Men Using Simple Measurement Techniques," *Medicine and Science in Sports and Exercise*, vol. 17, no. 2, p. 189, 1985.

[29] U. Alon, N. Barkai, D. a. Notterman, K. Gish, S. Ybarra, D. Mack, and a. J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 96, no. 12, pp. 6745–6750, 1999.

[30] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers, "Gene expression correlates of clinical prostate cancer behavior," *Cancer Cell*, vol. 1, no. 2, pp. 203–209, 2002.

[31] M. Schummer, W. V. Ng, R. E. Bumgarner, P. S. Nelson, B. Schummer, D. W. Bednarski, L. Hassell, R. L. Baldwin, B. Y. Karlan, and L. Hood, "Comparative hybridization of an array of 21 500 ovarian cDNAs for the discovery of genes overexpressed in ovarian carcinomas," *Gene*, vol. 238, no. 2, pp. 375–385, 1999.

[32] T. R. Golub, "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.

[33] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A Library for Large Linear Classification," *The Journal of Machine Learning*, vol. 9, no. 2008, pp. 1871–1874, 2008.