Dimensionality Reduction with Dimension Selection

Yi Guo[1,⋆], Junbin Gao[2], and Feng Li[3]

[1] CSIRO Mathematics, Informatics and Statistics,
North Ryde, NSW 1670, Australia
yi.guo@csiro.au
[2] School of Computing and Mathematics,
Charles Sturt University, Bathurst, NSW 2795, Australia
jbgao@csu.edu.au
[3] Earth observation Technology Application Department,
Academy of Opto-Electronics, CAS, China
lifeng@aoe.ac.cn

Abstract. We propose a novel method called sparse dimensionality reduction (SDR) in this paper. It performs dimension selection while reducing data dimensionality. Different from traditional dimensionality reduction methods, this method does not require dimensionality estimation. The number of final dimensions is the outcome of the sparse component of this method. In a nutshell, the idea is to transform input data to a suitable space where redundant dimensions are compressible. The structure of this method is very flexible which accommodates a series of variants along this line. In this paper, the data transformation is carried out by Laplacian eigenmaps and the dimension selection is fulfilled by $l2/l1$ norm. A Nesterov algorithm is proposed to solve the approximated SDR objective function. Experiments have been conducted on images from video sequences and protein structure data. It is evident that the SDR algorithm has subspace learning capability and may be applied to computer vision applications potentially.

1 Introduction

Recent years have been witnessing large increase in studies on dimensionality reduction (DR). The purpose of DR is mainly to find the corresponding counterparts (or embeddings) of the input data of dimension M in a much lower dimensional space (so-called latent space, usually Euclidean) of dimension n and $n \ll M$ without incurring significant information loss. A number of new algorithms which are specially designed for nonlinear dimensionality reduction (NLDR) have been proposed such as Lapacian Eigenmaps (LE) [1], Isometric mapping (Isomap) [2], Local Tangent Space Alignment (LTSA) [3], Gaussian Process Latent Variable Model (GPLVM) [4] etc. to replace the simple linear

⋆ The author to whom all the correspondences should be addressed.

J. Pei et al. (Eds.): PAKDD 2013, Part I, LNAI 7818, pp. 508–519, 2013.

methods such as Principal Component Analysis (PCA) [5], Linear Discriminant Analysis (LDA) [6] in which the assumption of linearity is essential.

Among these NLDR methods, it is worth mentioning those which can handle highly structured or so-called *non-vectorial* data (for example proteins, which are not readily converted to vectors) directly without vectorization. This category includes the "kernelized" linear methods. Typical methods are Kernel PCA (KPCA) [7], Generalized Discriminant Analysis (GDA or KLDA) [8] and so on. The application of the kernel function not only introduces certain nonlinearity implied by the feature mapping associated with the kernel which enables the algorithms to capture the nonlinear features, but also embraces much broader types of data including the aforementioned non-vectorial data. Meanwhile, kernels can also be regarded as a kind of similarity measurements which can be used in measurement matching algorithms like Kernel Laplacian Eigenmaps (KLE) [9] and Twin Kernel Embedding (TKE) [10]. Because these methods can directly use the structured data through kernel functions and hence bypass the vectorization procedure which might be a source of bias, they are widely used in complex input patterns like proteins, fingerprints, etc.

Although DR is proven to work well for some machine learning tasks such as classification [11], an inevitable question yet to be answered in applying DR is how to estimate the dimensionality which is the so-called intrinsic dimension estimation. Various methods have been presented in machine learning literature [12]. Nevertheless, a very simple way for dimension estimation is to reduce the dimension one at a time in a suitable space until significant information loss occurs and then stop. This procedure does the reduction and dimension estimation at the same time. There are two very important ingredients in this method: the proper space of the transformed data and the stop criterion of dimension reducing. These two should be combined seamlessly.

Interestingly, we can look at this problem the other way around. Instead of dropping dimensions, we can select dimensions in a suitable space. To do this properly, we refer to the variable selection framework. The nature of the variable selection problem is combinatorial optimization and hence NP hard. Nevertheless, there is a recent trend of using sparse approximation to solve this problem which has attracted attention in statistics and machine learning society. The earliest work is from [13] called the LASSO. By adding an $l1$ norm constraint on the coefficients of a linear regression model, the original combinatorial optimization problem was converted to a convex optimization problem which is much easier to solve. The optimization is normally cast as a regularization problem solved by lots of efficient algorithms such as coordinate descent [14], iterative shrinkage-thresholding [15] and etc. Several sparsity encouraging models have been proposed afterwards with various properties. For example, the group LASSO [16] has the group selection property by applying the $l2/l1$ norm to the group coefficients.

The above discuss leads to a novel method for dimensionality reduction which selects dimensions in transformed space and the selection is carried out by sparse methods. What follows is how to choose the transformed space? The research in

this decade on dimensionality reduction provides many solutions to this question. We choose Laplacian eigenmaps in this paper because it approximates the embedded manifold through a mapping function with proximity preserving property. However, It is very convenient to extend this choice to other methods such as LLE, TKE and etc. depending on the application at hand. Another variable in this idea is the sparse method normally by using sparsity encouraging norms. Since different sparsity encouraging norms come with different features, they provide us flexibility for various machine learning tasks, for example subspace learning, feature extraction etc. We use $l2/l1$ norm here for its group selection capability which is suitable for our dimension selection purpose. In Section 3, we will briefly discuss some of its variants and show how this idea could be used for subspace learning. Since this method involves sparse models for dimensionality reduction, we call it sparse dimensionality reduction or SDR for short.

The most related work is [17] where a rank prior as a surrogate of data dimensionality was imposed on GPLVM. In [17] the transformation of data was carried out by GPLVM, which converted the data to a space where a low rank representation (measured by rank prior) was possible. The stopping criterion was the stationary point of the optimization process. The work in [18] is also similar to ours but it was built on a sparse linear regression model and a dictionary in high dimensional space is required.

The paper is organized as follows. Section 2 briefly reviews Laplacian eigenmaps. Section 3 explains the proposed SDR method in detail followed by a section for the optimization procedure. In Section 5, we present several experimental results using SDR on visualization to show its effectiveness. We conclude this paper in Section 6.

2 Laplacian Eigenmaps

In the following discussion, let $\mathbf{y}_i \in \mathbb{R}^M$ be the i-th data sample on a manifold embedded in M dimensional space. Laplacian eigenmaps (LE) [1] is a typical nonlinear method that belongs to the family of graph-based DR methods. It attempts to preserve proximity relations in the input data which is expressed by a weight matrix based on adjacency graph (or called neighborhood graph). This adjacency graph G is constructed by referring to ε neighborhood or n nearest neighbor criterion. An edge will connect $\mathbf{y}_j$ and $\mathbf{y}_i$ if $\|\mathbf{y}_i - \mathbf{y}_j\|^2 < \varepsilon$ (ε neighborhood), or if $\mathbf{y}_j$ is among n nearest neighbors of $\mathbf{y}_i$ and vice versa (n nearest neighbor is more commonly used). The adjacency graph plays an important role in dimensionality reduction which leads to a series of graph based methods [19,20].

After the construction of the adjacency graph, the weights on the edges are evaluated that stand for the proximity relations among the input data. There are also two variations of setting the weights in LE: (a) exponential decay function:

$$\text{the weight } w_{ij} = \begin{cases} e^{-\sigma\|\mathbf{y}_i - \mathbf{y}_j\|^2}, \text{ if } \mathbf{y}_i \text{ is connected with } \mathbf{y}_j; \\ 0, \text{ otherwise.} \end{cases}$$

(b) binary ($\sigma = 0$): $w_{ij} = 1$ if $\mathbf{y}_i$ and $\mathbf{y}_j$ are connected, and $w_{ij} = 0$ otherwise. This simplification avoids the need to choose σ.

The weight matrix $\mathbf{W}$ ($\{w_{ij}\}$) containing the proximity information is then used to construct the Laplacian of the graph $\mathbf{L} = \mathbf{D} - \mathbf{W}$ where $\mathbf{D}$ is a diagonal matrix and its ii-th element $[\mathbf{D}]_{ii} = \sum_{j=1}^{N} w_{ij}$. The reason for Laplican is that the optimal locality preserving maps of the data on manifold onto $\mathbb{R}^K$ (K is at most $M-1$ because of the removal of arbitrary translation) can be approximated by obtaining the smallest K eigen vectors (excluding the eigen vector corresponding to eigenvalue 0) from the following generalized eigendecomposition

$$\begin{aligned} &\text{min. } \mathrm{tr}[\mathbf{X}^T\mathbf{L}\mathbf{X}] \\ &\text{s.t. } \mathbf{X}^T\mathbf{D}\mathbf{X} = \mathbf{I} \end{aligned} \tag{1}$$

where $\mathbf{X}$ of size $N \times K$ is the matrix of maps of $\mathbf{y}_i$'s in $\mathbb{R}^K$ and $\mathbf{I}$ is the identity matrix with compatible dimensions. For dimensionality reduction purpose, K is selected (somewhat arbitrarily) much less than M or by dimension estimation.

LE is a local method since it is based on the adjacency graph. Several variants have been derived from original LE such as the LPP (Locality Preserving Projection) [21] and the KLE (Kernel LE) [9]. LPP introduces a linear constraint between input data and embeddings, i.e $\mathbf{x}_i = \mathbf{A}\mathbf{y}_i$ while KLE replaces the weight matrix by a sparse kernel Gram matrix.

3 Data Reduction with Dimension Selection

We interpret the dimensionality reduction as a process of space transformation under the framework of Laplacian eigenmaps. The assumption is that the data lie on or near to a dimensional manifold embedded in M dimensional ambient space. The Laplacian eigenmaps is to unfold the manifold in a K dimensional subspace. As we do not know the intrinsic dimension of the manifold, We have to resort to other methods, which may be heterogeneous totally to LE, to estimate the dimensionality in advance.

However, if the unfolded manifold is indeed low dimensional, it should be "compressible", i.e. we can drop redundant dimensions while maintain the structure of the unfolded manifold in this transformed space. As discussed in Section 1, the force of compressing can be realized by introducing sparsity encouraging norms. The suitable one is the $l2/l1$ norm [16] defined as

$$||\mathbf{X}||_{2/1} = \sum_{i=1}^{K} ||\mathbf{X}^i||_2$$

where $\mathbf{X}^i$ is the i-th column of $\mathbf{X}$ and $||\cdot||_2$ is $l2$ norm. When $\mathbf{X}$ is a row vector, $l2/l1$ norm degenerates to normal $l1$ norm. An outstanding feature of $l2/l1$ norm is its group selection capability meaning that the elements in some $l2$ norms (groups) will be compressed towards zero altogether if the norm is minimized, for example in the group variable selection in [16,22]. For our purpose of dimension

selection, we use it to vanish the whole dimension, which is regarded as group in terms of $l2/l1$ norm, if it is dispensable.

Our sparse dimensionality reduction (SDR) takes the following form

$$\begin{aligned} \text{min. } & \text{tr}[\mathbf{X}^T\mathbf{L}\mathbf{X}] \\ \text{s.t. } & \mathbf{X}^T\mathbf{D}\mathbf{X} = \mathbf{I} \\ & ||\mathbf{X}||_{2/1} \leq t \end{aligned} \tag{2}$$

where $t \in \mathbb{R}^+$ is the parameter to control the "dimension sparsity". Following the previous discussion, the rationale is quite clear, that is we unfold the manifold in such a way that some of the dimensions can be reduced or in other words the most important dimensions can be selected. The algorithm starts with a generalized eigen decomposition, i.e. (2) without $l2/l1$ norm constraint. Once the selection completes, we retain the selected dimensions only from the initialization only. Details about implementation will be presented in Section 4.

Interestingly, if we substitute the $l2/l1$ norm by the nuclear norm in (2) as follows

$$\begin{aligned} \text{min. } & \text{tr}[\mathbf{X}^T\mathbf{L}\mathbf{X}] \\ \text{s.t. } & \mathbf{X}^T\mathbf{D}\mathbf{X} = \mathbf{I} \\ & ||\mathbf{X}||_* \leq t, \end{aligned} \tag{3}$$

the idea would be unfolding the manifold such that the rank of the of the embeddings, i.e. $\mathbf{X}$, is restricted. This is equivalent to finding a subspace in $\mathbb{R}^K$ that reveals the lower dimensional structure of the manifold. It suggests that potentially SDR can be used as a tool for subspace learning [23]. More generally, we can use other sparsity encouraging norms denoted as lq, which certainly brings different interpretation to this method.

Furthermore, we can extend this idea to LPP which is in line with LE. LPP has another layer on top of LE which is a linear mapping from input data to embeddings. In this case, we have the following objective in variable $\mathbf{A}$

$$\begin{aligned} \text{min. } & \text{tr}[\mathbf{A}\mathbf{Y}^T\mathbf{L}\mathbf{Y}\mathbf{A}^T] \\ \text{s.t. } & \mathbf{A}\mathbf{Y}^T\mathbf{D}\mathbf{Y}\mathbf{A}^T = \mathbf{I} \\ & ||\mathbf{A}||_q \leq t, \end{aligned} \tag{4}$$

where $\mathbf{A} \in \mathbb{R}^{K\times M}$ is the linear transformation matrix. q varies depending on the purpose of the application.

The flexibility of SDR enables it to embrace a lot of existing DR methods as well as sparse methods. In this paper, we focus only on (2) for its simplicity and direct understanding of dimension selection.

4 SDR Implementation

We proceed to obtaining the solution for SDR in (2). Direct optimization of the objective of SDR in (2) is very difficult because the nonsmooth $l2/l1$ norm

constraint and quadratic equality constraint, which makes it a quadratic programming with quadratic constraint (QPQC) problem with additional norm restriction. The quadratic equality constraint effectively excludes some popular alternating optimization schemes such as ADMM [24] since the augmented Lagrange term from the equality has no close form solution. It also throws some trouble to second order optimizer such as Newton-Raphson method because the Hessian is a tensor. To maintain the convexity of the original problem and also to make it easier to solve, we relax the equality constraint by converting it to a regularization term in the objective so that the first order algorithms are applicable. As a result, the original SDR problem has been converted to the following form

$$\begin{aligned} \text{min. } & \frac{1}{2}\text{tr}[\mathbf{X}^T\mathbf{L}\mathbf{X}] + \frac{\lambda}{4}||\mathbf{X}^T\mathbf{D}\mathbf{X} - \mathbf{I}||_F^2 + z\mathbf{e}^T\mathbf{t} \\ \text{s.t. } & ||\mathbf{X}||_{2/1} \preceq \mathbf{t} \end{aligned} \tag{5}$$

where $||\cdot||_F$ denotes the F norm of a matrix and $\mathbf{e}$ is an all one column vector. Note that we add another regularizer to the $l2/l1$ norm of $\mathbf{X}$, $z\mathbf{e}^T\mathbf{t}$ ($\mathbf{t} \in \mathbb{R}^K$), and the i-th element of $\mathbf{t}$ is responsible for $||\mathbf{X}^i||_2$. z has the same function as t in (2) controlling the dimension sparsity. The introduction of this additional regularization does not bring extra complexity; however, it enables us to use the efficient Euclidean projection explained in [25] with which (5) can be solved using Nesterov algorithm [26] easily. We will not go into too much detail of the algorithm but provide the necessary elements to make it work. Write

$$f(\mathbf{X}, \mathbf{t}) = \text{tr}[\mathbf{X}^T\mathbf{L}\mathbf{X}] + \lambda||\mathbf{X}^T\mathbf{D}\mathbf{X} - \mathbf{I}||_F^2 + z\mathbf{e}^T\mathbf{t}$$

supposing λ and z are given. We have the derivatives

$$\frac{\partial f(\mathbf{X}, \mathbf{t})}{\partial \mathbf{X}} = \mathbf{L}\mathbf{X} + \lambda\mathbf{D}\mathbf{X}(\mathbf{X}^T\mathbf{D}\mathbf{X} - \mathbf{I}) \tag{6}$$

and $\frac{\partial f(\mathbf{X},\mathbf{t})}{\partial \mathbf{t}} = z\mathbf{e}$ for this first order algorithm. The detailed optimization procedures are shown in Table 1.

In Nesterov algorithm, there are two sequences of variables, the target in the optimization problem, $\mathbf{X}$ and $\mathbf{t}$ in this case, and assistant variables, $\mathbf{S}$ and $\mathbf{h}$ shown in Table 1 corresponding to $\mathbf{X}$ and $\mathbf{t}$ respectively. The assistant variable is a linear combination of current and previous estimation of the target, e.g. $\mathbf{S}_i = \mathbf{X}_i + \alpha_i(\mathbf{X}_i - \mathbf{X}_{i-1})$. The tentative new estimation is given by the gradient projection in Line 6, where $\mathcal{P}_\mathcal{C}(x)$ is the Euclidean projection of x onto the feasible convex set $\mathcal{C}$. In our case, $\mathcal{C}$ is the set of values satisfying $||\mathbf{X}||_{2/1} \preceq \mathbf{t}$. The Euclidean projection of the given pair $\mathbf{U}$ and $\boldsymbol{v}$ is the solution to the following

$$\begin{aligned} \min_{\mathbf{X},\mathbf{t}} \ & \frac{1}{2}||\mathbf{X} - \mathbf{U}||_2^2 + \frac{1}{2}||\mathbf{t} - \boldsymbol{v}||_2^2 \\ \text{s.t. } & ||\mathbf{X}||_{2/1} \preceq \mathbf{t}. \end{aligned} \tag{7}$$

Table 1. SDR algorithm implementation

Optimize SDR via Nesterov algorithm

Input: $\mathbf{L}$, $\mathbf{D}$, λ, z, $\mathbf{X}_0$, $\mathbf{t}_0$
Output: optimal $\mathbf{X}$ and $\mathbf{t}$
1. Initialization: $\mathbf{X}_1 = \mathbf{X}_0$, $\mathbf{t}_1 = \mathbf{t}_0$, $l_{-1} = l_0 = 1$, $\gamma_0 = 1$.
2. for $i = 1$ to ...
3. $\alpha_i = \frac{l_{i-2}-1}{l_{i-1}}$, $\mathbf{S}_i = \mathbf{X}_i + \alpha_i(\mathbf{X}_i - \mathbf{X}_{i-1})$, $\boldsymbol{h}_i = \mathbf{t}_i + \alpha_i(\mathbf{t}_i - \mathbf{t}_{i-1})$
4. for $k = 1$ to ...
5. $\gamma = 2^{k-1}\gamma_{i-1}$
6. $[\mathbf{X}_{i+1}, \mathbf{t}_{i+1}] = \mathcal{P}_{\mathcal{C}}([\mathbf{S}_i, \mathbf{t}_i] - \frac{f'([\mathbf{S}_i, \mathbf{t}_i])}{\gamma_i})$
7. if $f([\mathbf{S}_{i+1}, \mathbf{t}_{i+1}]) \leq f_{\gamma,[\mathbf{S}_i,\mathbf{t}_i]}([\mathbf{S}_{i+1}, \mathbf{t}_{i+1}])$ then
8. $\gamma_i = \gamma$, break
9. end if
10. end for
11. $l_i = (1 + \sqrt{1 + 4l_{i-1}^2})/2$
12. if convergent then stop and output $\mathbf{X}_i$ and $\mathbf{t}_i$ as the solution.
13. end for

$f_{\gamma,\mathbf{x}}(\mathbf{y})$ is the the linear approximation of the objective function $f(\mathbf{y})$ at the point $\mathbf{x}$ regularized by the proximality

$$f_{\gamma,\mathbf{x}}(\mathbf{y}) = f(\mathbf{x}) + f'(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + \frac{\gamma}{2}||\mathbf{y} - \mathbf{x}||_2^2.$$

The algorithm in Table 1 has incorporated the Nemirovski line search in Line 4 to 10 for better efficiency. The initialization of $\mathbf{X}$ can be obtained from the solution of the generalized eigendecomposition $\mathbf{LX} = \mathbf{DX}\boldsymbol{\Lambda}$ where $\boldsymbol{\Lambda}$ is the diagonal matrix of the eigenvalues. Note that the last eigenvalue is 0 and its corresponding eigenvector should be removed to obtain translation invariant as in LE. The initial $\mathbf{t}$ can simply be the $l2$ norm of each column of $\mathbf{X}$.

The convergence is guaranteed by Nesterove algorithm. The computational complexity is mainly from matrix multiplications in the evaluation of the gradient in (6). The Euclidean projection in (7) is linear time. So the dominant complexity is $\mathcal{O}(K^3N^4)$ since $\mathbf{D}$ in (6) is a diagonal matrix. It may look very high. But as it iterates, many columns of $\mathbf{X}$ become zero, which effectively brings down K. Our experience with computation time is that it completes in several minutes on uptodate personal computer for $N = 2000$, $K = 600$.

5 Experimental Results

We applied the SDR to several data sets: COIL data set, Frey faces and SCOP protein structure data where LE has difficulties. They are widely used for machine learning and image processing tests. We mainly reduced the dimensionality

to 2 so that we can plot the embeddings in 2D plane for interpretation. z is selected by bisection so that only required dimensions were selected. To construct LE initialization for the SDR algorithm and LE itself, we set the number of nearest neighbors to be 5 and used simple binary weight for images data. We used KLE with Mammoth kernel for protein data where the number of nearest neighbors was 8. All these parameters were frequently used or reported to be somewhat optimal. For other methods in protein experiment, we also set their parameters to their reported optimal if any. In SDR optimization procedure, we set the update tolerance to be 1e-10 and maximum number of iterations to be 100, whichever reaches first stops the algorithm. It turned out that these settings worked well on the data sets we have tested in this section.

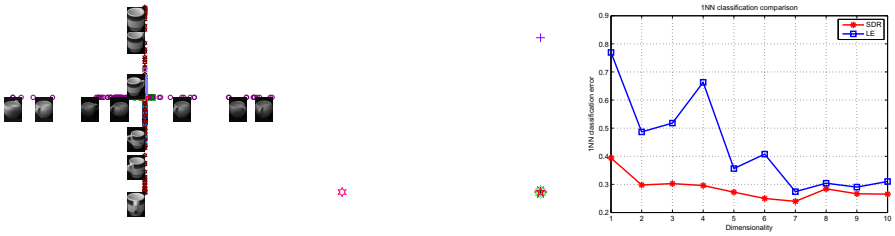
5.1 COIL 3D Images

We demonstrate SDR's dimension selection capability against LE in this experiment. We took the first 20 objects from Columbia Object Image Library (http://www1. cs.columbia.edu/CAVE/software/softlib/coil-100.php). Each object has 72 greyscale images of size 128×128 taken from video frames. Since all the images have been perfectly aligned and noise free, traditional methods like PCA can achieve good embedding. However, we focus on dimension selection capability of SDR here. In regard to 2D display, we expected that the objects to line up in the 2D plane somehow with some overlap. As we can see from Fig. 1 (b) where shapes stand for objects, LE's result is 3 isolated islands with heavy overlapping. However the 2D space selected by SDR reveals two cups classes clearly with overlap in the middle with other objets, which is closer to our expectation. This suggests SDR's subspace learning capability, which is further confirmed in the following experiment.

We further extended the target dimension from 1 to 10 and used the 1 nearest neighbor (1NN) classification errors rate as in [4] to compare the results quantitatively. The smaller the error rate, the better the method. The 1NN classification error rates are plotted in Figure 1 (c). It turned out that dimensions selected by SDR are consistently better although they are not optimized for classification task.

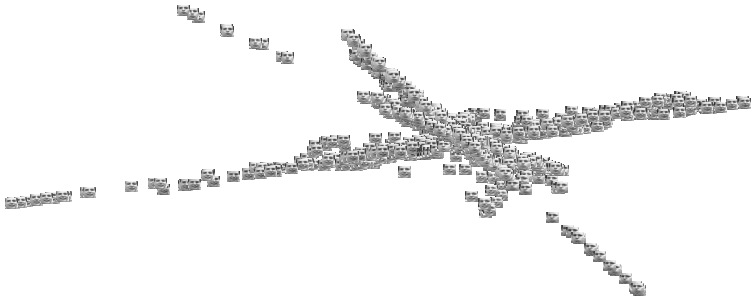
5.2 Frey Faces

In this subsection, the input data is 1,965 images (each 28×20 grayscale) of a single person's face taken from a video sequence which was also used in [27]. It is widely accepted that two parameters control the images, the face direction and facial expression. Intuitively, there should be two axes in 2D plane for these two parameters fused together somehow. However, the understanding like this is somewhat artificial. This may not even close to the truth. But we hope our algorithms can shed some light on these two parameters. Very interestingly as shown in Figure 2 (a) corresponding to SDR results, three axes for happy, sad and plain expressions respectively with continuously changing face direction can be clearly observed. It turns out that SDR identified the major dimensions as

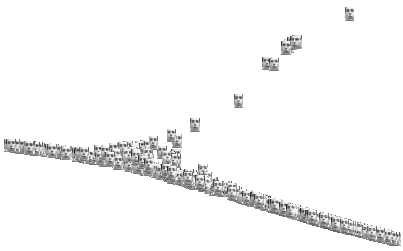


(a) SDR (b) LE (c) 1NN error

Fig. 1. COIL 3D images



(a) SDR



(b) LE

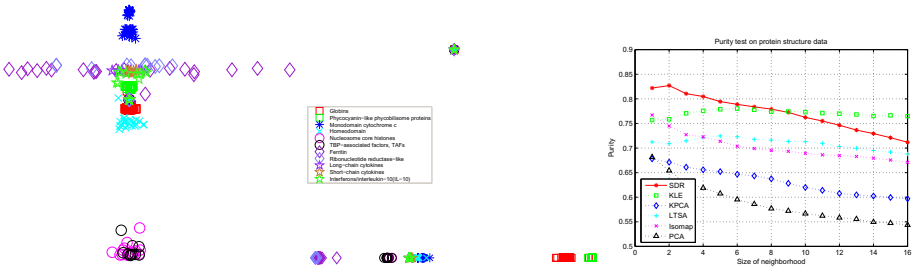
Fig. 2. Frey faces

facial expressions. The way that SDR axes are lined up once again pronounces its potential in subspace learning. The LE's result smeared out the facial expression and direction, which is not really informative.

5.3 Protein Structure Data

We move from image data (mainly video sequences) to protein structure data where KLE is more suitable because of the non-vectorial property of the protein data. Experiment was conducted on the SCOP (Structural Classification Of Protein). This database is available at http://scop.mrc-lmb.cam.ac.uk/scop/. It provides a detailed description of the structural and evolutionary relationships of the proteins of known structure. 292 protein sequences from different superfamilies and families were extracted for the test. The kernel we used is the Mammoth kernel, a so-called alignment kernel [28].

Only the results of SDR and KLE are plotted in Figure 3 for limited space. However other methods were tested. Each point (denoted as a shape in the figure) represents a protein. The same shapes with the same colors are the proteins from the same families (shown in legend) while the same shapes with different colors represent the proteins from different families but the same superfamilies. Except for better scattered clusters in SDR result, one noticeable difference is that one family dominates (diamonds) the horizontal axis in the middle in SDR result and others are projected to the other axis as 2D space is apparently not enough for this data set.



(a) SDR (b) KLE (c) Purity comparison

Fig. 3. 2D visualization of protein structure data

We used "purity" [10] to quantify the clusters produced by different methods. It uses the fraction of the number of samples from the same class as given point in a neighborhood of size n. The purity is the average of the fraction over all points. The higher the purity, the better the quality of the clusters. As we can see from Figure 3 (c), SDR has the purest clusters when $n < 9$. Although it drops below KLE when $n \geq 9$, it is still better than other methods in this test. Note that for linear methods like PCA we used the vectorization method derived form the kernel introduced in [10].

6 Conclusion

We proposed a novel method called sparse dimensionality reduction (SDR) in this paper along with a practical optimization algorithm to minimize an approximated SDR objective. SDR projects input data to a space where the redundant dimensions are compressible, and therefore it is not necessary to specify the dimensionality of the target space. The dimension sparsity parameter z in (5) is determined empirically. Bisection can be used if the target dimensionality is clear as shown in the experiments. If the final dimensionality is tied up with some quantitative standard such as MSE in regression, we can optimize z against it. It exhibits subspace learning property and the interesting results in images from video sequences suggested that SDR may be suitable for, and not confined to, computer vision applications such as subspace identification etc.

Acknowledgement. This project is sponsored by the Australian Research Council (ARC) under Discovery Project grant DP130100364 and also partially supported by the Commonwealth of Australian under the Australian-China Science and Research Fund (ACSRF01222).

References

1. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15(6), 1373–1396 (2003)
2. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290(22), 2319–2323 (2000)
3. Zhang, Z., Zha, H.: Principal manifolds and nonlinear dimensionality reduction via tangent space. SIAM Journal on Scientific Computing 26(1), 313–338 (2005)
4. Lawrence, N.: Probabilistic non-linear principal component analysis with gaussian process latent variable models. Journal of Machine Learning Research 6, 1783–1816 (2005)
5. Jolliffe, M.: Principal Component Analysis. Springer, New York (1986)
6. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of Eugenics 7, 179–188 (1936)
7. Schölkopf, B., Smola, A.J., Müller, K.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10, 1299–1319 (1998)
8. Baudat, G., Anouar, F.: Generalized discriminant analysis using a kernel approach. Neural Computation 12(10), 2385–2404 (2000)
9. Guo, Y., Gao, J., Kwan, P.W.H.: Kernel laplacian eigenmaps for visualization of non-vectorial data. In: Sattar, A., Kang, B.-H. (eds.) AI 2006. LNCS (LNAI), vol. 4304, pp. 1179–1183. Springer, Heidelberg (2006)
10. Guo, Y., Gao, J., Kwan, P.W.: Twin kernel embedding. IEEE Transaction of Pattern Analysis and Machine Intelligence 30(8), 1490–1495 (2008)
11. Maillard, O.A., Munos, R.: Compressed least-squares regression. In: Advances in Neural Information Processing Systems 2011 (2011)
12. Farahmand, A.M., Szepesvári, C., Audibert, J.Y.: Manifold-adaptive dimension estimation. In: Proceedings of the 24th International Conference on Machine Learning (2007)

13. Tibshirani, R.: Regression shrinkage and selection via the Lasso. Journal of Royoal Statistical Society 1(58), 267–288 (1996)
14. Friedman, J.H., Hastie, T., Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software 33(1), 1–22 (2010)
15. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences 2(1), 183–202 (2009)
16. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68(1), 49–67 (2006)
17. Geiger, A., Urtasun, R., Darrell, T.: Rank priors for continuous non-linear dimensionality reduction. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 880–887 (2009)
18. Gkioulekas, I., Zickler, T.: Dimensionality reduction using the sparse linear model. In: Advances in Neural Information Processing Systems 2011 (2011)
19. Saul, L.K., Weinberger, K.Q., Sha, F., Ham, J., Lee, D.D.: Spectral methods for dimensionality reduction. In: Chapelle, O., Schölkopf, B., Zien, A. (eds.) Semi-Supervised Learning. MIT Press, MA (2006)
20. Yan, S., Xu, D., Zhang, B., Zhang, H.J., Yang, Q., Lin, S.: Graph embedding and extensions: A general framework for dimensionality reduction. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(1), 40–51 (2007)
21. He, X., Niyogi, P.: Locality preserving projections. In: Thrun, S., Saul, L., Schölkopf, B. (eds.) Advances in Neural Information Processing Systems 16. MIT Press, Cambridge (2004)
22. Guo, Y., Gao, J., Hong, X.: Constrained grouped sparsity. In: Thielscher, M., Zhang, D. (eds.) AI 2012. LNCS, vol. 7691, pp. 433–444. Springer, Heidelberg (2012)
23. Lin, Z., Liu, R., Su, Z.: Linearized alternating direction method with adaptive penalty for low rank representation. In: Advances in Neural Information Processing Systems (2011)
24. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)
25. Liu, J., Ji, S., Ye, J.: Multi-task feature learning via efficient $l2,1$-norm minimization. In: UAI, pp. 339–348 (2009)
26. Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course. Kluwer Academic Publishers (2003)
27. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290(22), 2323–2326 (2000)
28. Qiu, J., Hue, M., Ben-Hur, A., Vert, J.P., Noble, W.S.: An alignment kernel for protein structures 23(9), 1090–1098 (2007)