Regularized Kernel Local Linear Embedding on Dimensionality Reduction for Non-vectorial Data

Yi Guo[1] and Junbin Gao[2]* and Paul W. Kwan[3]

[1] yg_au@yahoo.com.au
[2] School of Computing and Mathematics,
Charles Sturt University, Bathurst, NSW 2795, Australia,
jbgao@csu.edu.au
[3] School of Science and Technology,
University of New England, Armidale, NSW 2351, Australia,
kwan@turing.une.edu.au

Abstract. In this paper, we proposed a new nonlinear dimensionality reduction algorithm called regularized Kernel Local Linear Embedding (rKLLE) for highly structured data. It is built on the original LLE by introducing kernel alignment type of constraint to effectively reduce the solution space and find out the embeddings reflecting the prior knowledge. To enable the non-vectorial data applicability of the algorithm, a kernelized LLE is used to get the reconstruction weights. Our experiments on typical non-vectorial data show that rKLLE greatly improves the results of KLLE.

1 Introduction

In recent years it has been undergoing a 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 D in a much lower dimensional space (so-called latent space, usually Euclidean) of dimension d and $d \ll D$ without incurring significant information loss. A number of new algorithms which are specially designed for nonlinear dimensionality reduction (NLDR) have been proposed such as Local Linear Embedding (LLE) [1], Lapacian Eigenmaps (LE) [2], Isometric mapping (Isomap) [3], Local Tangent Space Alignment (LTSA) [4], Gaussian Process Latent Variable Model (GPLVM) [5] etc. to replace the simple linear methods such as Principal Component Analysis (PCA) [6], Linear Discriminant Analysis (LDA) [7] 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 [8] (for example video sequences, proteins etc which are not readily converted to vectors) directly without vectorization. This category includes the "kernelized" linear methods. Typical

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

methods are Kernel PCA (KPCA) [9], Generalized Discriminant Analysis (GDA or KLDA) [10]. 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 Multi-Dimensional Scaling (MDS) [11]. A typical example is Kernel Laplacian Eigenmaps (KLE) [12]. 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.

Because of its simplicity and elegant incarnation of nonlinearity from local linear patches, LLE has attracted a lot of attention. However, it has two obvious drawbacks. Firstly, it can only take vectorial data as input. Secondly, it does not exploit the prior knowledge of input data which is reflected by its somewhat arbitrary constraints on embeddings. As more and more non-vectorial data applications are emerging quickly in machine learning society, it is very desirable to endow LLE the ability to process this type of data. Fortunately, it is not difficult since only inner product is involved in LLE formulation. The "kernel trick " [13] provides an elegant solution to this problem. By introducing kernel functions, LLE can accept non-vectorial data which can be called KLLE. Moreover, another benefit from kernel approaches is its similarity measure interpretation which can be seen as a prior knowledge. We will utilize this understanding to restrict the embeddings and hence provide a solution to the second problem. This is done by incorporating kernel alignment into the current LLE as a regularizer of the embeddings which enforces the similarity contained in kernel function to be duplicated in lower dimensional space. It is equivalent to imposing a preference on the embeddings which favors such configuration that shares the same similarity relation (reflected by kernel function) as that among original input data. We conclude it as a new algorithm called regularized KLLE (rKLLE) as the main contribution of this paper. Our experiments on some typical non-vectorial data show that rKLLE greatly improves the results of KLLE.

This paper is organized as follows. Next section gives a brief introduction to LLE and we deduce the formulation of KLLE in sequel. rKLLE is developed in Section 4, followed by experimental results to show its effectiveness. Finally we conclude this paper in last section with highlight of future research.

2 Local Linear Embedding

We use following notations throughout the paper. $\mathbf{y}_i$ and $\mathbf{x}_i$ are the i-th input datum and its corresponding low-dimensional embedding, and $\mathbf{Y}$ and $\mathbf{X}$ the collection of input data and embeddings respectively. Generally, $\mathbf{X}$ is a matrix with data in rows.

Locally Linear Embedding (LLE) preserves the local linear relations in the input data which is encapsulated in a weight matrix $\mathbf{W}$. The algorithm starts with constructing a neighborhood graph by n nearest neighboring and then $\mathbf{W}$ ($[\mathbf{W}]_{ij} = w_{ij}$) is obtained by

$$\mathbf{W} = \arg\min_{\mathbf{W}} \sum_{i=1}^{N} \|\mathbf{y}_i - \sum_{j=1}^{n} w_{ij}\mathbf{y}_{i_j}\|^2 \tag{1}$$

subject to $\sum_j w_{ij} = 1$ and $w_{ij} = 0$ if there is no edge between $\mathbf{y}_i$ and $\mathbf{y}_j$ in the neighborhood graph. $\mathbf{y}_{i_j}$ is the j-th neighbor of $\mathbf{y}_i$.

Finally, the lower-dimensional embeddings are estimated by minimizing

$$\sum_i \|\mathbf{x}_i - \sum_j w_{ij}\mathbf{x}_j\|^2. \tag{2}$$

with respect to $\mathbf{x}_i$'s under the constraints $\sum_i \mathbf{x}_i = \mathbf{0}$ and $\frac{1}{N}\sum_i \mathbf{x}_i\mathbf{x}_i^\top = \mathbf{I}$ to remove arbitrary translations of the embeddings and avoid degenerate solutions. By doing this, the local linearity is reproduced in latent space.

3 Kernelized LLE

Actually, because of the quadratic form (1), w_{ij}'s are solved for each $\mathbf{y}_i$ separately in LLE. So we minimize $||\mathbf{y}_i - \sum_j w_{ij}\mathbf{y}_{i_j}||^2$ with respect to w_{ij}'s which is

$$\sum_j \sum_k w_{ij}(\mathbf{y}_{i_k} - \mathbf{y}_i)^\top(\mathbf{y}_{i_j} - \mathbf{y}_i)w_{ik} = \mathbf{w}_i^\top \mathbf{K}_i \mathbf{w}_i \tag{3}$$

subject to $\mathbf{e}^\top\mathbf{w}_i = 1$ where $\mathbf{e}$ is all 1 column vector. $\mathbf{w}_i$ is the vector of the reconstruction weights of $\mathbf{x}_i$, i.e. $\mathbf{w}_i = [w_{i1}, \ldots, w_{in}]$ and $\mathbf{K}_i$ is the local correlation matrix whose jk-th element is $(\mathbf{y}_{i_k} - \mathbf{y}_i)^\top(\mathbf{y}_{i_j} - \mathbf{y}_i)$.

Apparently, only inner product of input data is involved in (3). By using the "kernel trick" [13], the inner product can be replaced by any other positive definite kernel functions. Hence we substitute every inner product by a kernel $k_y(\cdot,\cdot)$ in the formation of $\mathbf{K}_i$ and have

$$[\mathbf{K}_i]_{jk} = k_y(\mathbf{y}_{i_k}, \mathbf{y}_{i_j}) - k_y(\mathbf{y}_{i_k}, \mathbf{y}_i) - k_y(\mathbf{y}_{i_j}, \mathbf{y}_i) + k_y(\mathbf{y}_i, \mathbf{y}_i).$$

Because the kernel function implies a mapping function ϕ from input data space to feature space and $k_y(\mathbf{y}_i, \mathbf{y}_j) = \phi(\mathbf{y}_i)^\top\phi(\mathbf{y}_j)$, we are actually evaluating the reconstruction weights in feature space. After we get the reconstruction weights, we can go further to solve (2) to obtain the embeddings in latent space.

We proceed to minimize (3) with equality constraint using Lagrange multiplier (we ignore the subscript i for simplicity):

$$J = \mathbf{w}^\top\mathbf{K}\mathbf{w} - \lambda(\mathbf{w}^\top\mathbf{e} - 1)$$

in which stationary point is the solution. $\frac{\partial J}{\partial \mathbf{w}} = 2\mathbf{K}\mathbf{w} - \lambda\mathbf{e} = 0$ leads to $\mathbf{w} = \frac{1}{2}\lambda\mathbf{K}^{-1}\mathbf{e}$. With $\mathbf{w}^\top\mathbf{e} = 1$, we have $\lambda = \frac{2}{\mathbf{e}^\top K^{-\top}\mathbf{e}}$. What follows is

$$\mathbf{w} = \frac{\mathbf{K}^{-1}\boldsymbol{e}}{\boldsymbol{e}^\top\mathbf{K}^{-\top}\mathbf{e}}.$$

The last thing is the construction of the neighborhood graph. It is quite straightforward in fact. Since kernel function represents similarity, we can just simply choose n most similar input data around $\mathbf{y}_i$. This corresponds to searching n nearest neighbors of $\phi(\mathbf{y}_i)$ in feature space because distance can be converted to inner product easily.

4 Regularized KLLE

Because of introducing kernel functions, not only can we process non-vectorial data in LLE, but also we are provided similarity information among input data which can be further exploited. The constraints used in the KLLE force the embeddings to have standard deviation. Apparently, this preference is imposed artificially on the embeddings which may not reflect the ground truth. This raises a question: can we use something more "natural" instead? Combining these points gives rise to the idea of replacing current constraint in KLLE by similarity matching.

The idea is implemented in following steps. Firstly, we pick up a similarity measure ($k_x(\cdot,\cdot)$, another kernel) in latent space which is matched to its counterpart in input space i.e. $k_y(\cdot,\cdot)$. Secondly, the similarity matching is implemented by kernel alignment. Thirdly, we turn the constrained optimization problem to regularization and therefore, the new regularized KLLE (rKLLE) minimizes the following objective function:

$$\begin{aligned} L = &\sum_i ||\mathbf{x}_i - \sum_j w_{ij}\mathbf{x}_{i_j}||^2 - \alpha \sum_{ij} k_x(\mathbf{x}_i, \mathbf{x}_j) k_y(\mathbf{y}_i, \mathbf{y}_j) \\ &+ \beta \sum_{ij} k_x^2(\mathbf{x}_i, \mathbf{x}_j). \end{aligned} \qquad (4)$$

The second and third regularization terms are from similarity matching[4]. α and β are positive coefficients which control the strength of the regularization. What is expressed in (4) is that the embedded data will retain the same local linear relationships as input data under the constraint that they should also exhibit the same similarity structure as that in input space.

In rKLLE, the prior knowledge provided by $k_y(\cdot,\cdot)$ is fully used in latent space and hence avoids from introducing other "rigid" assumptions which may be far away from the truth. There is also much room to accommodate additional priors due to its flexible algorithmic structure. For example if we know that the

[4] The second term is kernel alignment and last term is designed to avoid trivial solution such as infinity.

embeddings are from Gaussian distribution, we can add another regularizer on $\mathbf{X}$ (e.g. $\sum_i \mathbf{x}_i^\top \mathbf{x}_i$) at the end to incorporate this. An important issue related to the similarity match is the selection of the $k_x(\cdot,\cdot)$. In practice, we can choose RBF kernel, $k_x(\mathbf{x}_i, \mathbf{x}_j) = \gamma \exp(-\sigma||\mathbf{x}_i - \mathbf{x}_j||^2)$, because it has strong connection with Euclidean distance and this connection can be fine tuned by choosing appropriate hyper-parameters. Fortunately, the optimization of hyper-parameters can be done automatically as shown below.

The computational cost of rKLLE is higher than KLLE since (4) does not have close form solution. The above objective function can be written in simpler matrix form

$$\begin{aligned} L &= \mathrm{tr}[(\mathbf{X} - \mathbf{W}\mathbf{X})^\top (\mathbf{X} - \mathbf{W}\mathbf{X})] - \alpha \mathrm{tr}[\mathbf{K}_X^\top \mathbf{K}_Y] + \beta \mathrm{tr}[\mathbf{K}_X^\top \mathbf{K}_X] \\ &= \mathrm{tr}[(\mathbf{X}^\top \mathbf{M}\mathbf{X})] - \alpha \mathrm{tr}[\mathbf{K}_X^\top \mathbf{K}_Y] + \beta \mathrm{tr}[\mathbf{K}_X^\top \mathbf{K}_X] \end{aligned}$$

where $\mathbf{K}_X$ and $\mathbf{K}_Y$ are the kernel Gram matrices of $k_x(\cdot,\cdot)$ and $k_y(\cdot,\cdot)$ respectively, $\mathbf{M} = (\mathbf{I} - \mathbf{W})^\top (\mathbf{I} - \mathbf{W})$ and $\mathbf{I}$ is the identity matrix. We have to employ gradient descent based solver here. For the derivative, we first obtain

$$\frac{\partial L_{2,3}}{\partial \mathbf{K}_X} = -2\alpha \mathbf{K}_Y + 2\beta \mathbf{K}_X,$$

where $L_{2,3}$ is the second and third term of L. Then we get $\frac{\partial L_{2,3}}{\partial X}$ by chain rule (it depends on the form of k_x). The derivative of the first term of L is $2\mathbf{M}\mathbf{X}$. By putting them together, we can obtain $\frac{\partial L}{\partial X}$. The derivative of L with respect to hyper-parameters of k_x, denoted by $\mathbf{\Theta}$, can be calculated in the same way. Once we have the current version of $\mathbf{X}$, the gradient can be evaluated. Therefore, optimization process can be initialized by a guess of $\mathbf{X}$ and $\mathbf{\Theta}$. The initial $\mathbf{\Theta}$ can be arbitrary while starting $\mathbf{X}$ can be provided by other DR methods as long as non-vectorial data are applicable. From the candidates of gradient descent solvers, we choose SCG (scaled conjugate gradient) [14] because of its fast speed.

5 Experimental Results

To demonstrate the effectiveness of the proposed rKLLE algorithm, the experiments of visualizing non-vectorial data (the target latent space is a normal 2-D plane) were conducted on images (MNIST handwritten digits[5] and Frey faces[6]) and proteins (from SCOP database, Structural Classification Of Protein[7]). Proteins are recognized as typical highly structured data. The results of other algorithms are also shown for comparison.

5.1 Parameters Setting

rKLLE has some parameters to be determined beforehand. Through empirical analysis (performing batches of experiments on different data sets varying only

[5] MNIST digits are available at http://yann.lecun.com/exdb/mnist/

[6] Available at http://www.cs.toronto.edu/∼roweis/data/.

[7] SCOP data is available at http://scop.mrc-lmb.cam.ac.uk/scop/.

the parameters), we found the proposed algorithm is not sensitive to the choice of the parameters, as long as the conjugate gradient optimization can be carried out without immature early stop. So we use the following parameters throughout the experiments which are determined by experiments: $\alpha = 1e-3$ and $\beta = 5e-5$ and $n = 6$ in rKLLE neighborhood graph construction. The minimization will stop after 1000 iterations or when consecutive update of the objective function is less than 10^{-7}. $k_x(\cdot, \cdot)$ is RBF kernel and initialization is done by KPCA[8].

5.2 Handwritten Digits

A subset of handwritten digits images is extracted from the MNIST database. The data set consists of 500 images with 50 images per digit. All images are in grayscale and have a uniform size of 28×28 pixels. It is easy to convert them to vectors. So we can also present the results of other DR algorithms for comparison. However, in rKLLE, they were treated as non-vectorial data as bags of pixels [15] and use the shape context based IGV (SCIGV) kernel [16] which is specially designed for shapes in images and robust to the translation of shapes in images.

The experimental results are presented in Figure 1 and legend is in panel (c). Visually, the result of rKLLE is much better than others. The 2D representations of rKLLE reveal clearer clusters of digits than others. To give a quantitative analysis on the quality of the clusters, we use the leave-one-out 1 nearest neighbor (1NN) classification errors as in [5]. The smaller the number of errors, the better the method. The 1NN classification errors of different methods are collected in Table 1 and the result of each method is the best it can achieve by choosing the optimal parameters of the method (as shown in Figure 1) according to this standard. It is interesting to observe that rKLLE is the best regarding this standard. It shows clearly that rKLLE improves the result of KLLE both visually and quantitatively.

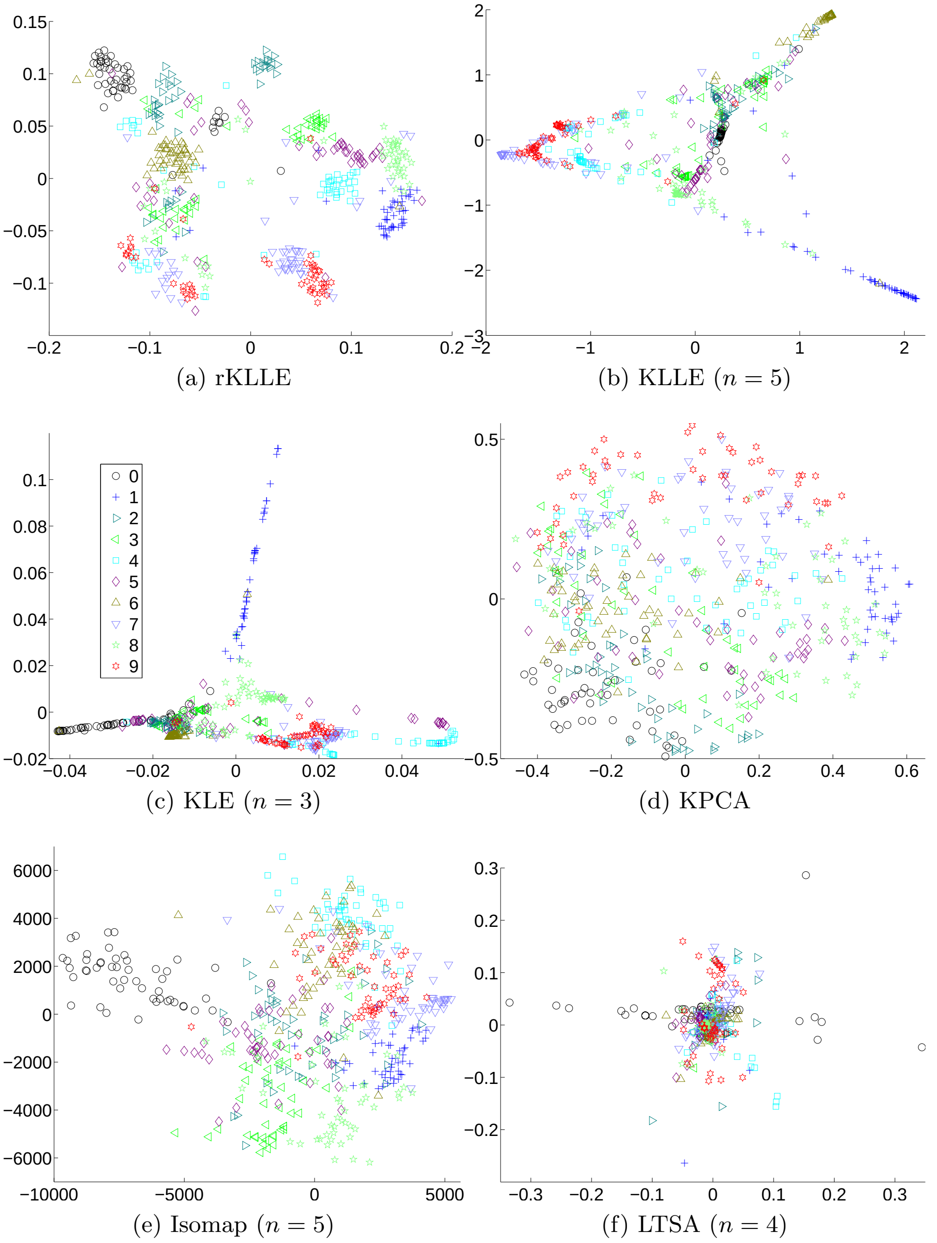
Algorithm	rKLLE	KLLE	KLE	KPCA	Isomap	LTSA
error	**110**	171	157	333	222	232

Table 1: Comparison of leave-one-out 1NN classification errors of different algorithms.

5.3 Proteins

Another promising application of DR is in bioinformatics. Experiments were conducted on the SCOP database. This database provides a detailed and comprehensive description of the structural and evolutionary relationships of the

[8] We choose KPCA instead of KLLE because this configuration yields better results in terms of leave-one-out 1NN classification errors.



(a) rKLLE

(b) KLLE ($n = 5$)

(c) KLE ($n = 3$)

(d) KPCA

(e) Isomap ($n = 5$)

(f) LTSA ($n = 4$)

Fig. 1: The result of different algorithms on MNIST handwritten digits database. The parameters of algorithms are chosen to achieve lowest 1NN classification errors.

proteins of known structure. 292 proteins from different superfamilies and families are extracted for the test. The kernel for proteins is MAMMOTH kernel which is from the family of the so-called alignment kernels whose thorough analysis can be found in [17]. The corresponding kernel Gram matrices are available on the website of the paper and were used directly in our experiments.

Visualizing proteins on the 2D plane is of great importance to facilitate researchers to understand the biological meaning. The representation of proteins on the 2D plane should reflect the relational structure among proteins, that is, proteins having similar structures should be close while those with different structures should be far away.

The results are plotted in Figure 2. The results of other non-vectorial data applicable algorithms are also presented for comparison. Each point (denoted as a shape in the figure) represents a protein. The same shapes with the the same color are the proteins from same families while the same shapes with different colors represent the proteins from different families but from the same superfamilies.

rKLLE reveals the fact that proteins from the same families congregate together as tight clusters and hence gains better interpretability. Interestingly, it also reveals the truth that the proteins from the same superfamily but different families are similar in structure, which is reflected by the fact that the corresponding groups (families) are close if they are in the same superfamily (same shape). Others fail to uncover these.

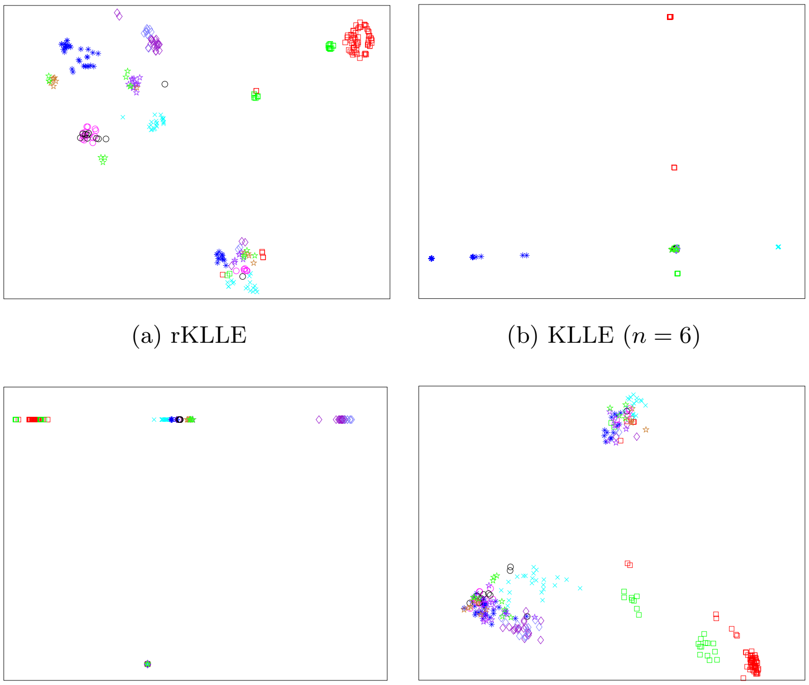
5.4 rKLLE on Image Manifold Learning

Lastly, we present the result of rKLLE on image manifold learning. The objects are 1965 images (each image is 20×28 grayscale) of a single person's face extracted from a digital movie which are also used in [1]. Since the images are well aligned, we simply use the linear kernel as $k_y(\cdot, \cdot)$ in this case.

Two facts can be observed from the result shown in Figure 3. First, it demonstrates group property of the faces. The faces with similar expressions and poses congregated as clusters. Second, The embeddings of the faces in 2D latent space indicate the possible intrinsic dimensionality of the images: expression and pose. From left to right and top to bottom, we can read the natural transition of the expression and pose. Therefore we can conjure that the horizontal axis might roughly correspond to the poses and vertical one to expressions and highly nonlinearity is apparently coupled in the axes. However, the real parametric space underpinning those faces images still needs further investigation.

6 Conclusion

In this paper, we proposed the regularized KLLE in which the original constraint of embeddings is replaced by a more natural similarity matching. It exploits the given information of the input data through regularization. It is a showcase of incorporating prior knowledge into dimensionality reduction process. Although the



(a) rKLLE

(b) KLLE ($n = 6$)

(c) KLE ($n = 6$)

(d) KPCA

Fig. 2: The results of different algorithms with MAMOTH kernel. The parameters of algorithms are chosen to achieve lowest 1NN classification errors.

computational cost is higher than KLLE or LLE, the improvement is significant from the results of experiments on typical non-vectorial data.

Due to the flexible structure of rKLLE, it is possible to handle other prior knowledge like class information. So it is likely to extend it to supervised or semisupervised learning setting. This will be our future research.

References

1. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(22) (2000) 2323–2326
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation **15**(6) (2003) 1373–1396
3. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(22) (2000) 2319–2323
4. Zhang, Z., Zha, H.: Principal manifolds and nonlinear dimensionality reduction via tangent space. SIAM Journal on Scientific Computing **26**(1) (2005) 313–338

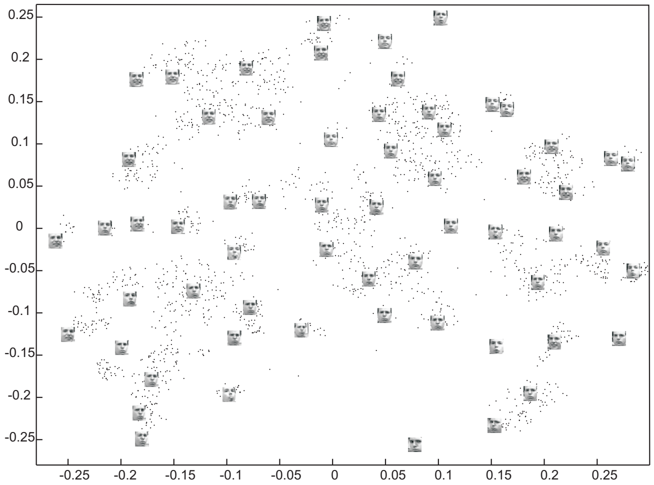


Fig. 3: The embeddings of faces in 2D latent space estimated by rKLLE.

5. Lawrence, N.: Probabilistic non-linear principal component analysis with gaussian process latent variable models. Journal of Machine Learning Research **6** (2005) 1783–1816
6. Jolliffe, M.: Principal Component Analysis. Springer-Verlag, New York (1986)
7. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of Eugenics **7** (1936) 179–188
8. Gärtner, T.: A survey of kernels for structured data. ACM SIGKDD Explorations Newsletter **5**(1) (2003) 49–58
9. Schölkopf, B., Smola, A.J., Müller, K.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation **10** (1998) 1299–1319
10. G., B., F., A.: Generalized discriminant analysis using a kernel approach. Neural Computation **12**(10) (2000) 2385–2404
11. Davison, M.L.: Multidimensional Scaling. Wiley series in probability and mathematical statistics, Applied probability and statistics. Wiley, New York (1983)
12. Guo, Y., Gao, J., Kwan, P.W.: Kernel Laplacian eigenmaps for visualization of non-vectorial data. In: Lecture Notes on Artificial Intelligence. Volume 4304. (2006) 1179–1183
13. Schölkopf, B., Smola, A.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. The MIT Press, Cambridge, MA (2002)
14. Nabney, I.T.: NETLAB: Algorithms for Pattern Recognition. Advances in Pattern Recognition. Springer, London (2004)
15. Jebara, T.: Images as bags of pixels. In: Ninth IEEE International Conference on Computer Vision (ICCV'03). Volume 1. (2003) 265–272
16. Guo, Y., Gao, J.: An integration of shape context and semigroup kernel in image classification. In: International Conference on Machine Learning and Cybernetics. (2007)
17. Qiu, J., Hue, M., Ben-Hur, A., Vert, J.P., Noble, W.S.: An alignment kernel for protein structures. In: Bioinformatics. Volume 23. (2007) 1090–1098