Author: Y. Guo, J. Gao, P. Kwan and K. Xinsheng Hou

Title: Visualization of protein structure relationships using constrained twin kernel embedding

Journal: Journal of Biomedical Science and Engineering ISSN: 1937-6871 1937-688X

Year: 2008

Volume: 1

Issue: 2

Pages: 133-140

Abstract: In this paper, a recently proposed dimensionality reduction method called Twin Kernel Embedding (TKE) [10] is applied in 2-dimensional visualization of protein structure relationships. By matching the similarity measures of the input and the embedding spaces expressed by their respective kernels, TKE ensures that both local and global proximity information are preserved simultaneously. Experiments conducted on a subset of the Structural Classification Of Protein (SCOP) database confirmed the effectiveness of TKE in preserving the original relationships among protein structures in the lower dimensional embedding according to their similarities.This result is expected to benefit subsequent analyses of protein structures and their functions.

Author Address: jbgao@csu.edu.au

Visualization of Protein Structure Relationships Using Constrained Twin Kernel Embedding

Yi Guo
School of Science and Technology
University of New England, Armidale, NSW 2351, Australia
yguo4@turing.une.edu.au

Junbin Gao*
School of Computer Science
Charles Sturt University, Bathurst, NSW 2795, Australia
jbgao@csu.edu.au

Paul W. Kwan
School of Science and Technology
University of New England, Armidale, NSW 2351, Australia
kwan@turing.une.edu.au

Kevin X. Hou
School of Biological, Biomedical and Molecular Sciences
University of New England, Armidale, NSW 2351, Australia
xhou@une.edu.au

March 11, 2008

Abstract

In this paper, a recently proposed dimensionality reduction method called Twin Kernel Embedding (TKE) [10] is applied in 2-dimensional visualization of protein structure relationships. By matching the similarity measures of the input and the embedding spaces expressed by their respective kernels, TKE ensures that both local and global proximity information are preserved simultaneously. Experiments conducted on a subset of the Structural Classification Of Protein (SCOP) database confirmed the effectiveness of TKE in preserving the original relationships among protein structures in the lower dimensional embedding according to their similarities. This result is expected to benefit subsequent analyses of protein structures and their functions.

1 Introduction

Recent years have seen phenomenal advances in dimensionality reduction (DR) methods that are widely applied in bioinformatics [26, 14, 6, 16], biometrics [19, 4, 15, 17], robotics [11], etc. The rapid growth of DR methods stems from the need to reduce the complexity of the problem at hand. The target of these methods is mainly to find the corresponding counterparts of the

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

input data in a much lower dimensional space without incurring significant information loss. The low dimensional representation can be used in subsequent procedures such as classification, pattern recognition, and so on. For example, a medium sized protein structure typically has a few thousands of degrees of freedom which is naturally residing in very high dimensional space. It causes the so-called "curse of dimensionality" problem which will not only drastically increase the computational complexity of the learning algorithms but also require large storage space, leading to very slow indexing and searching speed in a large scale database in the sequel. Through DR, most of the redundant dimensions can be removed and the degrees of freedom left are then input to the subsequent discriminant and classification tasks with considerable simplication.

Another advantage of using DR is the 2- or 3-dimensional mappings of the original data can be visualized in an Euclidean space that can facilitate interpreting the relationships among data by the researchers. Normally, the relationships among a set of protein structures are typically represented in the form of trees derived by hierarchical clustering. However, this representation only provides some hints on the evolutionary distances between protein structures. This limitation motivates our applying to the application of DR methods in visualizing the similarity relationships among protein structures.

In this paper, we will propose a new DR algorithm called Constrained Twin Kernel Embedding (CTKE) based on Twin Kernel Embedding (TKE) [10] to achieve the above target. To provide the necessary background knowledge, we will first give a brief review of DR methods on protein structures in the next section, followed by an introduction of TKE and related topics involved in this new algorithm. Then the CTKE algorithm will be introduced, which integrates the similarity matching by TKE with the objective function used in LS-SVM [23]. Experiments on real protein structure data will be presented and finally we summarize this paper with a conclusion.

2 The Related Works

DR methods can be categorized into Linear DR methods (LDR) such as Principal Component Analysis (PCA) [13], Linear Discriminant Analysis (LDA) [7] and NonLinear DR methods (NLDR) such as ISOMAP [25], Laplacian Eigenmaps (LE) [3], Locally Linear Embedding (LLE) [20], etc. LDR methods have been widely used in bioinformatics due to their simplicity. For example, Teodoro et al. [27] applied PCA to transform the original high dimensional protein motion data into a lower dimensional representation that captures the dominant modes of motions of the protein. However, the linearity assumption on which linear methods are constructed does not hold in most cases. In [5], Das et al. successfully projected the folding free-energy on a few relevant coordinates by using a typical nonlinear method, ISOMAP, to correctly identify the transition-state ensemble of the reaction based on the fact that empirical reaction coordinates routinely used in protein folding studies cannot be reduced to a linear combination of the Cartesian coordinates.

Because of the power of the NLDR methods, they have also been applied to visualizing the relationships between protein structures. Hanke and Reich [12] employed the Kohonen self organizing maps, a special form of neural networks as a visualization tool for the analyses of protein structure similarity by converting the sequences into a characteristic signal matrix. In [1], by using the pairwise similarity index between two sequences, Sammon maps projected the sequences onto a display plane in such a way that the Euclidean distances between the images approximate as closely as possible the corresponding values in the original sequence space.

The metric used to measure the similarity between two protein structures was based on the individual residue similarities derived from a series of amino acid exchange matrices. Further, a modified nonlinear Sammon projection was developed in [2] to display the relationships among protein structures based on their amino acid composition.

Recently, a new method called Stochastic Proximity Preserving (SPE) was introduced into this field by Farnum et al. in [6]. SPE preserves only the local relationships among closely related sequences to avoid a drawback of those global methods such as MDS that underestimates the proximity of sequences since all pairwise distances are included in the algorithm, leading to erroneous results. To emphasize the proximity, SPE first applies a neighborhood filtering procedure to the similarity matrix (the similarity metric used in SPE is identical to that used in [1]). Then the filtered similarity matrix is input into the Sammon's nonlinear mapping to obtain the final result.

From the discussion above, we can clearly see that there are three important components in DR: dis/similarity metric for the input data (protein structures in this paper), the objective function (the core of the algorithm) and the dis/similarity metric for images (the corresponding low dimensional representation of the protein structures) which is usually the Euclidean distance. These DR methods are trying to preserve the similarity metric of the protein structures as much as possible and reproduce it in a human interpretable space. The dis/similarity metrics used in those methods mentioned above are based on proteins and their structure-related evaluators while the objective functions are from Sammon's mapping.

In computational biology, the similarity metric known as kernel functions that are both powerful and promising is gaining much attention. An important advantage of a kernel function is that the form of the input data does not have to be vectorial. Any structured data like protein structures can be properly processed by specially designed kernel functions. This advantage avoids the information loss during vectorization. TKE is constructed on the basis of this kind of similarity metric. The objective function is totally different from Sammon's mapping and furthermore the dis/similarity metric for images is not limited to simple Euclidean distance, but a kernel function to capture the nonlinearity.

3 Twin Kernel Embedding

Without loss of generality, the following notations will be adopted. The data in the input space $\mathcal{Y}$ are denoted by $\mathbf{y}_i$ $(i = 1, \ldots, N)$ while $\mathbf{x}_i$ $(i = 1, \ldots, N)$ their embeddings in a low-dimensional space or the so-called latent space $\mathcal{X}$. Notice that here the $\mathbf{y}_i$ will be the protein structures. The term "embeddings" is from the manifold learning literature which means the images of the input data or equivalently the embedded data. In addition, $\mathbf{Y}$ and $\mathbf{X}$ will be used to denote respectively the set of input objects and the set of embedded objects. If the objects were vectorial, $\mathbf{Y}$ (and $\mathbf{X}$) would denote a matrix consisting of rows of vectors. Furthermore, $a \cdot b$ denotes the inner product of two vectors a and b.

The Twin Kernel Embedding (TKE) preserves the similarity structure of input data in the latent space by matching the similarity relations represented by two kernel Gram matrices, i.e. one for the input data and the other for their embeddings by simply minimizing the objective function

$$-\mathrm{Vec}\mathbf{K_y} \cdot \mathrm{Vec}\mathbf{K_x}, \tag{1}$$

where Vec is the vec operator on matrix (to stack all the columns of the matrix to make a long vector) and $\mathbf{K_y}$ and $\mathbf{K_x}$ are the kernel Gram matrices derived from valid Mercer kernel functions $k_y(\cdot, \cdot)$ and $k_x(\cdot, \cdot)$ [21] defined on the input data and embeddings respectively. The idea is to

preserve the similarities among the input data and reproduce them in the lower dimensional latent space expressed again in similarities among embeddings. To make this point clearer, we can simply regard $\text{Vec}\mathbf{K_y} \cdot \text{Vec}\mathbf{K_x}$ as a linear kernel (liner kernel is defined as $k(a,b) = a \cdot b$) which is a measure of similarity of the variables involved in the kernel function. The larger the value of the kernel, the more similar these two variables are. As a result, we minimize (1) to make $\mathbf{K_x}$ and $\mathbf{K_y}$ as similar as possible.

To avoid any trivial solutions to (1), two regularization terms on the kernel and embeddings are introduced and the objective function in (1) becomes

$$L = -\text{tr}(\mathbf{K_y}\mathbf{K_x}) + \lambda_k \text{tr}(\mathbf{K_x}\mathbf{K_x}) + \lambda_x \text{tr}(\mathbf{X}\mathbf{X}^\top), \tag{2}$$

where we use the fact that $\text{Vec}\mathbf{K_y} \cdot \text{Vec}\mathbf{K_x} = \text{tr}(\mathbf{K_y}\mathbf{K_x})$. The second term is a ridge regularizer on the kernel to make sure that the norm of the kernel is controlled. This can avoid solutions that simply let the elements in $\mathbf{K_x}$ go to infinity. The third term imposes a heavy penalty on too large a norm for the embeddings which ensure that their coordinates are relatively small. λ_k and λ_x are tunable parameters to control the strength of the regularization and are assumed to be positive.

In order to capture the nonlinear structure, $k_x(\cdot,\cdot)$ should be chosen to be nonlinear. Normally, we use the RBF kernel

$$k_x(\mathbf{x}_i, \mathbf{x}_j) = \gamma \exp\left(-\sigma \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2}\right). \tag{3}$$

where γ and σ are positive hyperparameters. Nonetheless, other kernels can also be applied here. The selection of RBF kernel is for its connection to the Euclidean distance which has a geometric understanding of the embeddings. There is no closed form solution for $\mathbf{X}$ and hence an optimization procedure like gradient descent based algorithm for optimization should be employed provided that $k_x(\cdot,\cdot)$ is differentiable. The initialization of $\mathbf{X}$ is also required to start the optimization process. KLE [9], KPCA [22] and other methods that can work with kernels could be utilized here for initializing $\mathbf{X}$. The dimension of the embeddings which is normally 2 is assigned according to the application. A by-product of this optimization process is that we can get the optimal hyper-parameters (such as γ and σ if RBF kernel is used) of the kernel function $k_x(\cdot,\cdot)$ as well. It ensures that the kernel we pick up is well adjusted.

TKE is designed to preserve locality and non-locality at the same time. This is done by the filtering of the entries in $\mathbf{K_y}$. Not all the entries remain the same in the optimization process but those that convey most of the similarity information of the input data. This filtering process is fulfilled by performing k-nearest neighbor selecting procedure on $\mathbf{K_y}$. Given an object $\mathbf{y}_i$ in the input space, only those objects whose similarities (in the sense of kernel values) to $\mathbf{y}_i$ are in the k nearest neighbors that are selected to retain their original values while all others are set to 0. The variable $k(> 1)$ in k-nearest neighboring controls the neighborhood that the algorithm will preserve. It can be interpreted as constructing a weighted adjacency graph which performs in feature space and the weights on edges are evaluated by a kernel as that in KLE. Similar procedure is involved in SPE as well which uses the neighborhood radius to ensure the proximity preserving property. Because TKE tries to match $\mathbf{K_x}$ to $\mathbf{K_y}$ and RBF kernel (3) cannot give a value of 0 except that two points in the latent space are very far apart, TKE seeks a solution that keeps the points in the same neighborhood close while makes the points not in the same neighborhood be very far. However, TKE also works without filtering in which case TKE will preserve all pairwise similarities and will become a global approach simply.

In addition to the fact that TKE outperforms other methods such as KPCA, KLE etc, an elegant feature of TKE is that it uses only the pairwise similarities since in its objective

function, only the kernel Gram matrix of the input data is required. Through TKE, any kind of data can be visualized in lower dimensional space as long as an appropriate kernel is defined for them. As a result, a kernel Gram matrix on the input data $\mathbf{K_y}$ and an initialization for $\mathbf{X}$ will be adequate for TKE to find the optimal embeddings.

4 Constrained Twin Kernel Embedding

It is clear from observing (2) that there is no explicit connections between input data and their corresponding embeddings in TKE except the similarity preserving. As such, the information hidden in the input data is neglected. For example, if the input data actually stay on or near to a smooth manifold embedded in the ambient space, the location of the input data can be explored to predict the coordinates of the embeddings on the manifold. Furthermore, TKE can only find the optimal embeddings for currently presented data as we can see from its objective function. To address these problems, we introduce the constraints reflecting the relationship between input data and embeddings into TKE by following steps. We first define a mapping function $f : \mathcal{Y} \to \mathcal{X}$ and then incorporate it into the objective functional of TKE as regularization terms. Finally the optimal embeddings $\mathbf{X}$ and the f will be searched via conjugate gradients algorithm.

We can start from the kernel feature mapping directly and incorporate it as the core part of the LS-SVM (the dual form of the objective function of LS-SVM) into TKE as what has been done in [24]. We minimize the following objective function similar to that of LS-SVM with equality constraints

$$J = \frac{\upsilon}{2} \sum_{j=1}^{d} \omega_j^\top \omega_j + \frac{\eta}{2} \sum_{ij} e_{ij}^2 \tag{4}$$

$$\text{s.t. } x_{ij} = \omega_j^\top \varphi_j(\mathbf{y}_i) + e_{ij} \tag{5}$$

corresponding to the maximum margin (the first term) and least square errors (the second term) in (4) where υ and η are adjustable parameters. φ_j's are the feature mappings which map the input $\mathbf{y}_i$ into Hilbert space where the inner product is defined. ω_j is a column vector having the same dimension as the Hilbert space. We can regard the $\mathbf{x}_i$ in (5) as the projection of $\varphi(\mathbf{y}_i)$ onto a subspace parameterized by ω_j's. Because the difference of the constraints, it does not have the same geometrical interpretation of LS-SVM because the original constraints are inequalities reflecting the correct classification hyperplanes while here they are just feature mapping which builds the relation between input data $\mathbf{y}_i$ and its counterpart $\mathbf{x}_i$ in low dimensional space. In (4), we are minimizing the error e_{ij} in reconstruction of $\mathbf{x}_i$ essentially however it should be done with ω_{ij} properly constrained. This happens to have the form of the LS-SVM objective function. To solve (4) with the equality constraints (5), we can use Lagrange multipliers as

$$L = \frac{\upsilon}{2} \sum_{j=1}^{d} \omega_j^\top \omega_j + \frac{\eta}{2} \sum_{ij} e_{ij}^2 + \sum_{ij} \alpha_{ij}(x_{ij} - \omega_j^\top \varphi_j(\mathbf{y}_i) - e_{ij}) \tag{6}$$

From the saddle points, $\frac{\partial L}{\partial \omega_j} = 0$, $\frac{\partial L}{\partial e_{ij}} = 0$ we have

$$\frac{\partial L}{\partial \omega_j} = \upsilon \omega_j - \sum_i \alpha_{ij} \varphi_j(\mathbf{y}_i) = 0 \;\Rightarrow\; \omega_j = \frac{1}{\upsilon} \sum_i \alpha_{ij} \varphi_j(\mathbf{y}_i)$$

$$\frac{\partial L}{\partial e_{ij}} = \eta e_{ij} - \alpha_{ij} = 0 \;\Rightarrow\; e_{ij} = \frac{1}{\eta} \alpha_{ij}.$$

Substitute them back into (6) and eliminate ω_j and e_{ij} we have the dual problem to be maximized according to the min-max duality

$$\begin{aligned} L =& \frac{\upsilon}{2}\sum_{j=1}^{d}(\frac{1}{\upsilon}\sum_{i}\alpha_{ij}\varphi_j(\mathbf{y}_i))^\top(\frac{1}{\upsilon}\sum_{i}\alpha_{ij}\varphi_j(\mathbf{y}_i)) + \frac{\eta}{2}\sum_{ij}(\frac{1}{\eta}\alpha_{ij})^2 \\ &+ \sum_{ij}\alpha_{ij}\{x_{ij} - (\frac{1}{\upsilon}\sum_{i}\alpha_{ij}\varphi_j(\mathbf{y}_i))^\top\varphi_j(\mathbf{y}_i) - \frac{1}{\eta}\alpha_{ij}\} \\ =& -\frac{1}{2\upsilon}\sum_{j}\boldsymbol{\alpha}_j^\top\mathbf{K}_j\boldsymbol{\alpha}_j - \frac{1}{2\eta}\sum_{ij}\alpha_{ij}^2 + \sum_{ij}\alpha_{ij}x_{ij} \end{aligned} \tag{7}$$

where $\boldsymbol{\alpha}_j = (\alpha_{1j}, \ldots, \alpha_{2j})^\top$, $\varphi(\mathbf{y}_m)^\top\varphi(\mathbf{y}_i) = k_j(\mathbf{y}_m, \mathbf{y}_i)$ to which the "kernel trick" applies. $k_j(\cdot,\cdot)$ is the kernel associated with the kernel mapping and $\mathbf{K}_j$ is the Gram matrix derived from $k_j(\cdot,\cdot)$ accordingly. We then maximize the above dual problem with respect to α_{ij} instead of ω_j and e_{ij}. From the discussion above we see that the x_{ij}'s are free variables. To limit the choice of them, we combine (7) with the objective functional of TKE to incorporate the similarity preserving as

$$\begin{aligned} L =& -\sum_{i,j}k_y(\mathbf{y}_i,\mathbf{y}_j)k_x(\mathbf{x}_i,\mathbf{x}_j) + \lambda_k\sum_{ij}k_x(\mathbf{x}_i,\mathbf{x}_j)^2 + \lambda_x\sum_{i}\mathbf{x}_i\mathbf{x}_i^\top \\ &+ \frac{1}{2\upsilon}\sum_{j}\boldsymbol{\alpha}_j^\top\mathbf{K}_j\boldsymbol{\alpha}_j + \frac{1}{2\eta}\sum_{ij}\alpha_{ij}^2 - \sum_{ij}\alpha_{ij}x_{ij} \end{aligned} \tag{8}$$

where we turn the maximization of (7) to minimization aligned with the TKE objective functional. Here we see that the terms related to $\mathbf{y}_i$ are expressed by kernel $k_y(\cdot,\cdot)$. Therefore, this revised objective function is still non-vectorial data applicable. Again we express (8) into matrix form to facilitate the differentiation and let $k_j(\cdot,\cdot) = k_y(\cdot,\cdot)$ for simplicity

$$\begin{aligned} L =& -\operatorname{tr}[\mathbf{K}_\mathbf{x}\mathbf{K}_\mathbf{y}] + \lambda_k\operatorname{tr}[{\mathbf{K}_\mathbf{x}}^2] + \lambda_x\operatorname{tr}[\mathbf{X}\mathbf{X}^\top] \\ &+ \frac{1}{2\upsilon}\operatorname{tr}[\mathbf{A}^\top\mathbf{K}_\mathbf{y}\mathbf{A}] + \frac{1}{2\eta}\operatorname{tr}[\mathbf{A}^\top\mathbf{A}] - \operatorname{tr}[\mathbf{A}^\top\mathbf{X}] \end{aligned} \tag{9}$$

where

$$\mathbf{A} = \begin{bmatrix} \alpha_{11} & \ldots & \alpha_{1d} \\ \vdots & & \vdots \\ \alpha_{N1} & \ldots & \alpha_{Nd} \end{bmatrix}.$$

Hence we minimize L in (8) with respect to $\mathbf{A}$, $\mathbf{X}$ and kernel hyperparameters of $k_x(\cdot,\cdot)$. If we substitute the saddle point solution back into the equality constraints (5), the following mapping function is handy to predict new input samples

$$x_{ij} = \frac{1}{\upsilon}\sum_{m}\alpha_{mj}k_y(\mathbf{y}_m,\mathbf{y}_i) + \frac{1}{\eta}\alpha_{ij}. \tag{10}$$

The errors $\frac{1}{\eta}\alpha_{ij}$ can be neglected since the values of the errors are very small compared with $\mathbf{X}$ after optimization.

To apply the conjugate gradient algorithm, the derivatives of L with respect to $\mathbf{X}$ and $\mathbf{A}$ are given by

$$\frac{\partial L}{\partial \mathbf{X}} = \frac{\partial L}{\partial \mathbf{K_x}} \frac{\partial \mathbf{K_x}}{\partial \mathbf{X}} - \mathbf{A}, \text{ and, } \frac{\partial L}{\partial \mathbf{K_x}} = 2\lambda_k \mathbf{K_x} - \mathbf{K_y}, \tag{11}$$

$$\frac{\partial L}{\partial \mathbf{A}} = \frac{1}{\upsilon} \mathbf{K_y} \mathbf{A} + \frac{1}{\eta} \mathbf{A} - \mathbf{X}. \tag{12}$$

$\mathbf{X}$ is still initialized by KPCA or KLE to obtain pure non-vectorial data applicability. Initial $\mathbf{A}$ is from the solution of $\frac{\partial L}{\partial \mathbf{A}} = 0$ after $\mathbf{X}$ is known. We have $\mathbf{A} = (\frac{1}{\upsilon}\mathbf{K_y} + \frac{1}{\eta}\boldsymbol{I})^{-1}\mathbf{X}$. It implies that we could alternately update $\mathbf{X}$ and $\mathbf{A}$ in optimization, however we still use conjugate gradient algorithm to update them at the same time. Because the mapping function defined between input space and latent space acts as constraints, we call this algorithm Constrained TKE (CTKE). It is noticeable that in CTKE, we use $\mathbf{K_y}$ to construct the mapping function, so we do not have to filter $\mathbf{K_y}$ before commencing the algorithm.

5 Experimental Results

Experiments were conducted on the SCOP (Structural Classification Of Protein). This database is available at http://scop.mrc-lmb.cam.ac.uk/scop/. The database comes with a detailed and comprehensive description of the structural and evolutionary relationships of the proteins of known structure. 292 protein structures from different protein superfamilies and families are extracted for the test. The kernels we used are from the family of the so-called alignment kernels whose thorough analyses can be found in [18]. The corresponding kernel Gram matrices are available on the website of the paper as supplements and were used directly in our experiments.

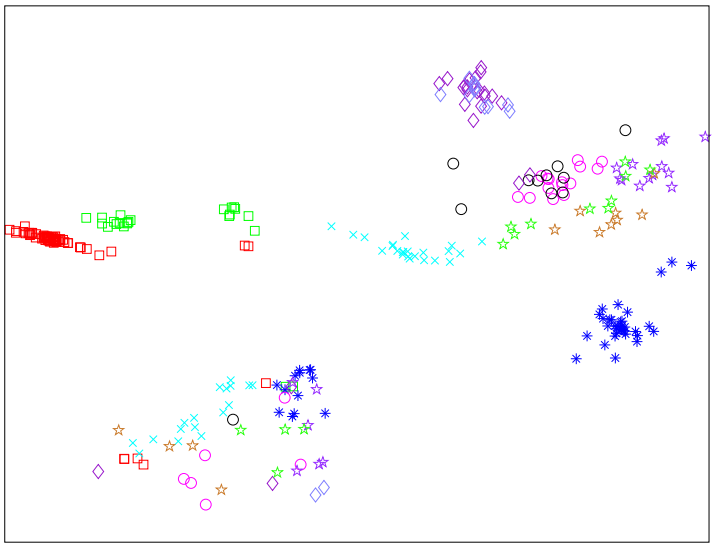
5.1 Parameters Setting

Both CTKE and TKE have parameters to be determined beforehand. Through empirical analyses (performed a set of experiments on the same data set varying only the parameters), we found these algorithms are not sensitive to the choice of the parameters, so long as the conjugate gradient optimization can be carried out without premature early stop. So we use the following parameters in the experiments. For TKE, $\lambda_k = 0.005$, $\lambda_x = 0.001$ and $k = 10$ in k nearest neighboring; for CTKE, $\lambda_k = 0.05$, $\lambda_x = 0.01$, $\upsilon = \eta = 0.5$. 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 the RBF kernel and initialization is done by KPCA for both CTKE and TKE.

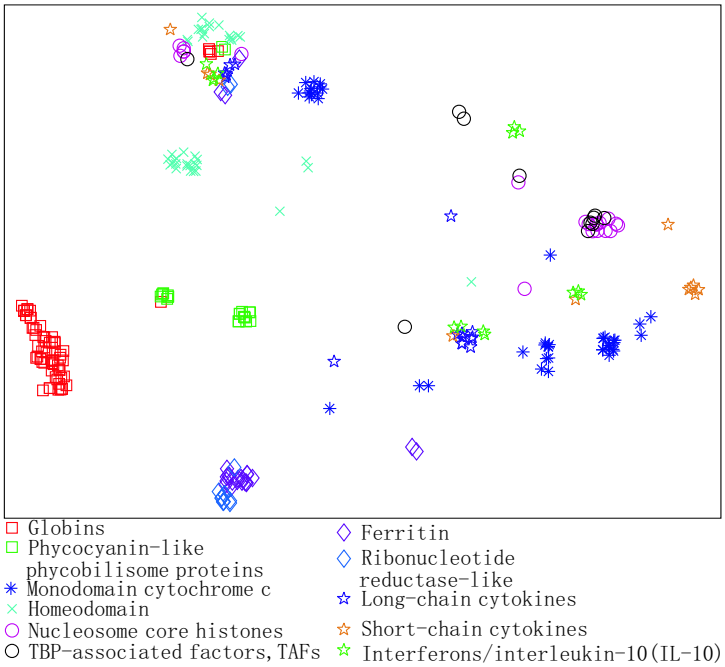
5.2 Proteins on 2-D Plane

The proteins from the same families are expected to be close in the 2-dimensional space with overlappings indicating similar proteins but actually from different families. The results are plotted in Figure 1. The results of other methods (kernel applicable methods) are also presented for comparison in Figure 2. Each point (denoted as a shape in the figures) represents a protein. The same shapes with the same colors are the proteins from same families while the same shapes with different colors represent the proteins from different families but of the same superfamilies. All the figures in this paper share the same legends as those in Figure 1 (b).

Both TKE and CTKE reveal the fact that proteins from the same families congregate together as clusters while KPCA and KLE fail to reveal it. For example, in Figure 1 (a) and (b), almost all of the proteins from the globin family gathered at the bottom left corner indicating



(a) CTKE with MAMMOTH kernel



(b) TKE with MAMMOTH kernel

Figure 1: Visualization results of protein structures

similar structures. Interestingly, CTKE and TKE also reveals the fact that the proteins from the same superfamily but different families are similar in structure, which is reflected in the 2-dimensional plane that the corresponding groups (families) are close if they are in the same superfamily. For instance, note that the proteins from ferritin and ribonucleotide reductase-like families (blue diamonds and violet diamond respectively in the figure), they are close in the group level. SPE has comparable performance visually. But the overlapping in the middle shows that it cannot distinguish some families clearly.

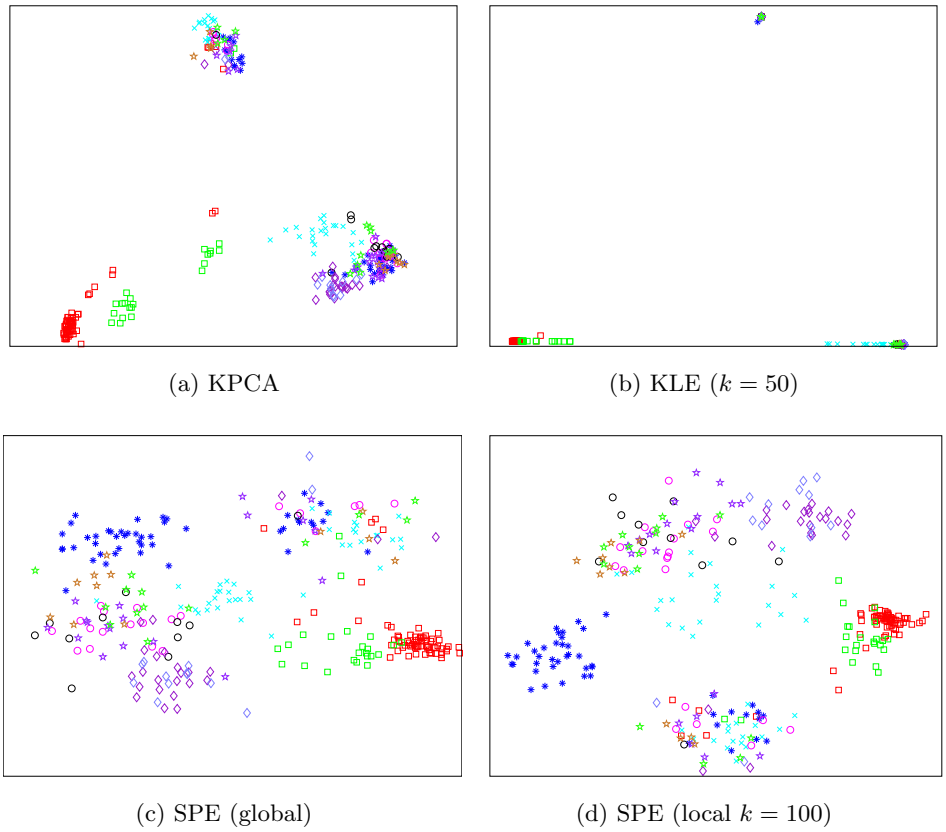
In order to further quantify the results, we use the "purity" [8] to evaluate some of these methods. It uses the fraction of the number of samples from the same class as given point in a neighborhood with size n. The purity is the average of the fraction over all points. The higher the purity, the better the quality of the clusters. Intuitively, this standard provides an objective judgement from the classification point of view and the method with better purity has the potential to achieve better classification rate. It is noticeable that for SPE and KLE with parameter k, we did multiple experiments to choose the optimal value of k corresponding to the largest purity when the size of neighborhood is 1. We found that for SPE the larger the k, the better the result. Specifically, when k equals the number of the data, i.e. no filtering at all, SPE turns out to be an nonlinear MDS. As we can see from Figure 3, CTKE and TKE have higher purity than others. The average purity of CTKE is 0.5946 and TKE 0.6135. It shows that CTKE has very close performance to TKE. However, the advantage of the CTKE is its ability to predict novel samples because of the mapping function defined explicitly between input space and latent space. This mechanism broadens the applicability of this algorithm greatly to classification, identification etc. It also provides a tool to explore the manifold formed by data.

6 Conclusions

In this paper, we visualized the similarity relationships among protein structures using constrained Twin Kernel Embedding (CTKE) which is constructed on the original TKE [10]. It has comparable performance to that of TKE but possesses the ability to predict the embeddings for novel samples. Because CTKE implements a mapping function from input space to latent space, the information among input data is further exploited. CTKE also has the similarity preserving property as TKE does and is purely non-vectorial data applicable since the mapping function comes from the feature mapping and expressed as kernel function eventually. Moreover, there is no k-nearest neighbor filtering in CTKE and hence avoid choosing another parameter which is common across other DR algorithms. From the experiments on proteins, we have seen that CTKE preserves the similarity relationships among the protein structures and reproduces them in a much lower dimensional space, allowing easy interpretation by researchers. This algorithm is promising as it can be further applied to the study of the evolution of protein structure and the prediction of proteins functions.

Acknowledgements

This work is supported by the National Science Foundation of China (NSFC 60373090), the ARC DP Development Grant from Charles Sturt University and the University Research Grant from the University of New England.



(a) KPCA (b) KLE ($k = 50$)

(c) SPE (global) (d) SPE (local $k = 100$)

Figure 2: The result of other methods with MAMMOTH kernel. Like TKE, SPE can work globally and locally depending on whether the kernel Gram matrix is filtered first by k-nearest neighboring. SPE global worked with the whole kernel Gram matrix and SPE local filtered $\mathbf{K_y}$ with 100-nearest neighboring in this experiment.

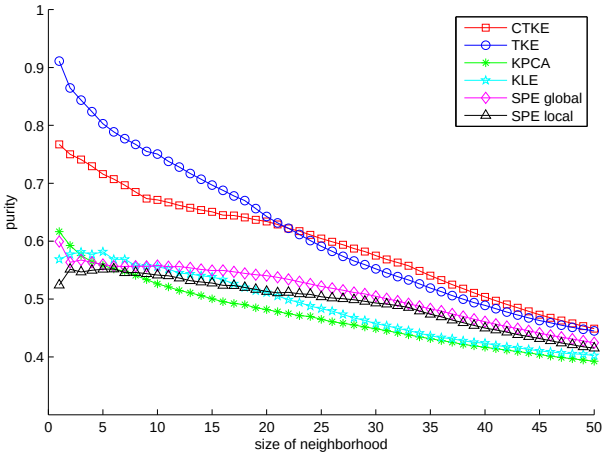


Figure 3: Comparison of different methods using purity.

References

[1] Dimitris K. Agrafiotis. A new method for analyzing protein sequence relationships based on sammon maps. *Protein Science*, 6(2):287–293, 1997.

[2] Izydor Apostol and Wojciech Szpankowski. Indexing and mapping of proteins using a modified nonlinear sammon projection. *Journal of Computational Chemistry*, 20(10):1049–1059, 1999.

[3] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[4] Samarasena Buchala, Neil Davey, Ray J. Frank, and Tim M. Gale. Dimensionality reduction of face images for gender classification. In *Proceedings of 2nd International IEEE Conference on Intelligent Systems*, volume 1, pages 88–93, June 2004.

[5] Payel Das, Mark Moll, Hernán Stamati, Lydia E. Kavraki, and Cecilia Clementi. Low-dimensional free energy landscapes of protein folding reactions by nonlinear dimensionality reduction. In *Proceedings of the National Academy of Sciences*, volume 103, pages 9885–9890, USA, 2006.

[6] Michael A. Farnum, Huafeng Xu, and Dimitris K. Agrafiotis. Exploring the nonlinear geometry of protein homology. *Protein Science*, 12(1604-1612), 2003.

[7] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[8] Amir Globerson, Gal Chechik, Fernando Pereira, and Naftali Tishby. Euclidean embedding of co-occurrence data. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 497–504. MIT Press, Cambridge, MA, 2005.

[9] Yi Guo, Junbin Gao, and Paul W. Kwan. Kernel Laplacian eigenmaps for visualization of non-vectorial data. In *Lecture Notes on Artificial Intelligence*, volume 4304, pages 1179–1183, 2006.

[10] Yi Guo, Junbin Gao, and Paul W. Kwan. Twin kernel embedding. *IEEE Transaction of Pattern Analysis and Machine Intelligence*, submitted, 2008.

[11] Jihun Ham, Yuanqing Lin, and Daniel. D. Lee. Learning nonlinear appearance manifolds for robot localization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2971– 2976, August 2005.

[12] Jens Hanke and Jens G. Reich. Kohonen map as a visualization tool for the analysis of protein sequences: multiple alignments, domains and segments of secondary structures. *Bioinformatics*, 12(6):447–454, 1996.

[13] M. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

[14] Philip M. Kim and Bruce Tidor. Subsystem identification through dimensionality reduction of large-scale gene expression data. *Genome Research*, 13:1706–1718, 2003.

[15] Nathan Mekuz, Christian Bauckhage, and John K. Tsotsos. Face recognition with weighted locally linear embedding. In *Proceedings of the 2nd Canadian Conference on Computer and Robot Vision*, pages 290–296, May 2005.

[16] Oleg Okun, Helen Priisalu, and Alexessander Alves. Fast non-negative dimensionality reduction for protein fold recognition. In *ECML*, pages 665–672, 2005.

[17] Zhengjun Pan, Rod Adams, and Hamid Bolouri. Dimensionality reduction of face images using discrete cosine transforms for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[18] Jian Qiu, Martial Hue, Asa Ben-Hur, Jean-Philippe Vert, and William Stafford Noble. An alignment kernel for protein structures. In *Bioinformatics*, to appear in 2007.

[19] Bisser Raytchev, Ikushi Yoda, and Katsuhiko Sakaue. Multi-view face recognition by nonlinear dimensionality reduction and generalized linear models. In *FGR '06: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pages 625–630, Washington, DC, USA, 2006.

[20] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(22):2323–2326, Dec. 2000.

[21] B. Schölkopf and A.J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, Cambridge, MA, 2002.

[22] Bernhard Schölkopf, Alexander J. Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[23] J.A.K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9:293–300, 1999.

[24] Johan A.K. Suykens. Data visualization and dimensionality reduction using kernel maps with a reference point. Technical Report 07-22, K.U. Leuven, ESAT-SCD/SISTA, 2007.

[25] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(22):2319–2323, Dec. 2000.

[26] Miguel L. Teodoro, George N. Phillips Jr, and Lydia E. Kavraki. A dimensionality reduction approach to modeling protein flexibility. In *International Conference on Computational Molecular Biology (RECOMB)*, pages 299–308, April 2002.

[27] Miguel L. Teodoro, George N. Phillips Jr, and Lydia E. Kavraki. Understanding protein flexibility through dimensionality reduction. *Journal of Computational Biology*, 10(3-4):617–634, June 2003.