DEPARTMENT OF STATISTICS
University of Wisconsin
1300 University Ave.
Madison, WI 53706

TECHNICAL REPORT NO. 1119

April 11, 2006

Kernel Regularization and Dimension Reduction

Fan Lu [1]
Departments of Statistics, University of Wisconsin, Madison, WI

Sündüz Keleş [2]
Department of Statistics and Department of Biostatistics and Medical Informatics, University of Wisconsin, Madison, WI

Yi Lin [3]
Department of Statistics, University of Wisconsin, Madison, WI

Stephen J. Wright [4]
Department of Computer Sciences, University of Wisconsin, Madison, WI

Grace Wahba [1]
Department of Statistics, Department of Computer Sciences and Department of Biostatistics and Medical Informatics, University of Wisconsin, Madison, WI

Key Words and Phrases: Kernel Regularization, Regularized Kernel Estimate (RKE), multidimensional scaling (MDS), manifold unfolding, positive definite matrices, noisy dissimilarity data, convex cone programming.

[1]Research supported in part by NSF Grant DMS0072292, NIH Grant EY09946, and ONR Grant N00014-06-1-0095.
[2]Research supported in part by Wisconsin Alumni Research Foundation.
[3]Research supported in part by NSF Grant DMS0134987.
[4]Research supported in part by NSF Grants ATM-0296033, CNS-0127857, CCF-0113051, ANI-0330538, DMS-0427689, CCF-0430504, and DOE Grant DE-FG02-04ER25627.

Kernel Regularization and Dimension Reduction

Fan Lu, Sündüz Keleş, Yi Lin, Stephen J. Wright and Grace Wahba
University of Wisconsin, Madison, 53706, USA

Abstract

It is often possible to use expert knowledge or other sources of information to obtain dissimilarity measures for pairs of objects, which serve as pseudo-distances between the objects. When dissimilarity information is available as the data, there are two different types of problems of interest. The first is to estimate full position configuration for all objects in a low dimensional space while respecting the dissimilarity information. This is usually for the purposes of visualizing the data and/or conducting further statistical analysis, such as clustering or classification. Multidimensional Scaling (MDS), which is still an active research area, has been traditionally used to tackle this problem. In the second type of problems, the high dimensional data points are assumed to lie on a low dimensional manifold and the goal is to unfold the manifold in order to recover the underlying intrinsic low dimensional structure.

We provide a novel, unified framework called Kernel Regularization to optimally solve both types of problems. Advanced optimization techniques are utilized to obtain the global solutions accurately and efficiently. The proposed method can naturally accommodate the dissimilarity information with possibly crude, noisy, incomplete, inconsistent and weighted observations. Various favorable operating characteristics and properties of the method are illustrated using both simulated and real data sets.

1 Dissimilarity Information and Regularized Kernel Estimate

Given a set of N objects, suppose we have obtained a measure of dissimilarity, d_{ij}, for certain object pairs (i, j). We introduce the class of Regularized Kernel Estimates (RKEs), which we define as solutions to optimization problems of the following form:

$$\min_{K \in S_N} \sum_{(i,j)\in\Omega} L\big(w_{ij}, d_{ij}, \hat{d}_{ij}(K)\big) + \lambda J(K), \qquad (1)$$

where S_N is the convex cone of all real nonnegative definite matrices of dimension N, Ω is the set of pairs for which we utilize dissimilarity information, and L is some reasonable loss function, convex in $\hat{d}_{ij}$, where $\hat{d}_{ij}$ is the dissimilarity induced by K. J is some reasonable convex kernel penalty (regularizing) function and λ is a tuning parameter balancing fit to the data and the penalty on K. The w_{ij} are weights that may, if desired, be associated with particular (i, j) pairs. The natural induced dissimilarity, which is a real squared distance admitting of an inner product, is $\hat{d}_{ij} = K(i,i)+K(j,j)-2K(i,j) = B_{ij}\cdot K$, where $K(i, j)$ is the (i, j) entry of K and B_{ij} is a symmetric matrix of dimension N with all elements 0 except $B_{ij}(i, i) = B_{ij}(j, j) = 1$, $B_{ij}(i, j) = B_{ij}(j, i) = -1$. The inner (dot) product of two matrices of the same dimensions is defined as: $A \cdot B = \sum_{i,j} A(i, j) \cdot B(i, j) \equiv \text{trace}(A^T B)$. There are essentially no restrictions on the set of pairs other than requiring that the graph of the objects with pairs connected by edges be connected. A pair may have repeated observations, which just yield an additional term in (1) for each separate observation. If the pair set induces a connected graph, then the minimizer of (1) will have no local minima.

Although it is usually natural to require the observed dissimilarity information $\{d_{ij}\}$ to satisfy $d_{ij} \geq 0$ and $d_{ij} = d_{ji}$, the general formulation above does not require these properties to hold. The observed dissimilarity information may be incomplete (with the restriction noted), it may not satisfy the triangle inequality, or it may be noisy. It also may be crude, as for example when it encodes a small number of coded levels such as "very close", "close", "distant", and "very distant".

2 Procrustes Measures

A reasonable measure of the distance/dissimilarity between two kernel matrices is needed to check convergence and characterize the goodness of fit for different estimates. In some related literature, such a measure is called Procrustes measure.

A suitable measure proposed in [1] is based on the positional differences after matching two gram matrices under translation, rotation, and reflection. Suppose A and B are two centered gram matrices, then the measure is calculated as follows:

$$G(A, B) = trace(A) + trace(B) - 2\, trace(A^{\frac{1}{2}} B A^{\frac{1}{2}})^{\frac{1}{2}}.$$

The normalized version of this measure is simply:

$$\gamma_p(A, B) = G(A, B)/(trace(A)trace(B))^{\frac{1}{2}}. \qquad (2)$$

Alternatively, if we care only about the pairwise distance information, we can introduce another normalized measure:

$$\gamma_d(A, B) = \sum_{i<j} |\hat{d}_{ijA} - \hat{d}_{ijB}| / \sum_{i<j} \frac{1}{2}(\hat{d}_{ijA} + \hat{d}_{ijB}), \qquad (3)$$

where $\hat{d}_{ijA}$ and $\hat{d}_{ijB}$ are pairwise squared distance between object i and j, induced by A and B, respectively. Both γ_p and γ_d will be close to zero if kernels A and B represent close configurations.

3 General Convex Cone Problem

In the next two sections, we will introduce two specific formulations of (1) to solve the two types of problems introduced in the Abstract. Both formulations can be converted into a so-called general convex cone programming problem, which we now specify here. This problem, which is central to modern optimization research, involves some unknowns that are vectors in Euclidean space and others that are symmetric matrices. These unknowns are required to satisfy certain equality constraints and are also required to belong to cones of a certain type. The cones have the common feature that they all admit a self-concordant barrier function, which allows them to be solved by interior-point methods that are efficient in both theory and practice [2].

To describe the cone programming problem, we define some notation. Let $\mathcal{R}^p$ be Euclidean p-space and let P_p be the nonnegative orthant in $\mathcal{R}^p$, that is, the set of vectors in $\mathcal{R}^p$ whose components are all nonnegative. We let Q_q be the second-order cone of dimension q, which is the set of vectors $x = \big(x(1), \ldots, x(q)\big) \in \mathcal{R}^q$ that satisfy the condition $x(1) \geq [\sum_{i=2}^{q} x(i)^2]^{\frac{1}{2}}$. We define S_s to be the cone of symmetric positive semidefinite $s \times s$ matrices of real numbers. Inner products between two vectors are defined in the usual way and we use the dot notation for consistency with the matrix inner product notation. The general convex cone problem is then:

$$\begin{aligned} \min_{X_j, x_i, z} \quad & \sum_{j=1}^{n_s} C_j \cdot X_j + \sum_{i=1}^{n_q} c_i \cdot x_i + g \cdot z \qquad (4) \\ \text{s.t.} \quad & \sum_{j=1}^{n_s} A_{rj} \cdot X_j + \sum_{i=1}^{n_q} a_{ri} \cdot x_i + g_r \cdot z = b_r, \quad \forall_r \\ & X_j \in S_{s_j} \ \forall_j; \quad x_i \in Q_{q_i} \ \forall_i; \quad z \in P_p. \end{aligned}$$

Here, C_j, A_{rj} are real symmetric matrices (not necessarily positive semidefinite) of dimension s_j and c_i, $a_{ri} \in \mathcal{R}^{q_i}$, g, $g_r \in \mathcal{R}^p$, $b_r \in \mathcal{R}^1$.

The global solution of a convex cone programming problem can be obtained numerically using publicly available software such as SDPT3 [3] and DSDP5 [4].

4 RKE for Multidimensional Scaling

MDS Problem. The goal of this task (see [5]) is to estimate full position configuration for all objects in a preferably low dimensional space while respecting all dissimilarity information available, whatever "local" or "global" (corresponding to dissimilar or similar pairs). More details on the material presented in this section can be found in our paper [6].

Formulation. We describe a specific formulation of (1), based on a linearly weighted l_1 loss, and use the trace function in the regularization term to promote dimension reduction. The resulting problem is as follows:

$$\min_{K \succeq 0} \sum_{(i,j) \in \Omega} w_{ij} |d_{ij} - B_{ij} \cdot K| + \lambda \, trace(K). \qquad (5)$$

Both this formulation and the variant of it, in which a quadratic loss function is used in place of the l_1 loss function can be posed as convex conic optimization problems (see [6]).

The reason that we use trace as the kernel regularization function is very intuitive. (There turned out to be some theoretical evidence, which we won't discuss here, to support this choice). We want to obtain low rank kernels as RKE solutions. But rank is not a nice function to optimize with since it is discontinuous and nonconvex. So we use trace, which is a continuous and linear (thus convex) function of the kernel, as a simple approximation to the rank. Another inspiration is from the idea of LASSO [7]. Since we want to promote the sparsity among eigenvalues of the estimated kernel, the LASSO idea suggests to regularize the l_1 norm of the eigensequence vector (i.e., trace of the kernel) instead of the l_0 norm (i.e., rank of the kernel).

"Newbie" Problem (out-of-sample extension). We now consider the situation in which a solution K_N of (5) is known for some set of N objects. We wish to augment the optimal kernel (by one row and column), without changing any of its existing elements, to account for a new object (a newbie). That is, we wish to find a new "pseudo-optimal" kernel $\tilde{K}_{N+1}$ of the form:

$$\tilde{K}_{N+1} = \begin{bmatrix} K_N & b^T \\ b & c \end{bmatrix} \succeq 0, \qquad (6)$$

(where $b \in \mathcal{R}^N$ and c is a scalar) that solves the following optimization problem:

$$\begin{aligned} \min_{c \geq 0, b} \ & \textstyle\sum_{i \in \Psi} w_i \ |d_{i,N+1} - B_{i,N+1} \cdot K_{N+1}| \qquad (7) \\ \text{s.t.} \ & \ b \in \text{Range}(K_N), \quad c - b^T K_N^+ b \geq 0, \end{aligned}$$

where K_N^+ is the pseudo-inverse of K_N and Ψ is a subset of $\{1, 2, \ldots, N\}$ of size t. The quantities w_i, $i \in \Psi$ are the weights assigned to the dissimilarity data for the new point. The constraints in this problem are the necessary and sufficient conditions for $\tilde{K}_{N+1}$ to be positive semidefinite. Again, this constrained optimization problem and its quadratic loss variant can be formulated as convex cone programming problems (see [6]). Their global solutions can be obtained in polynomial time.

Choosing Elements of Ω and Tuning λ. See [6].

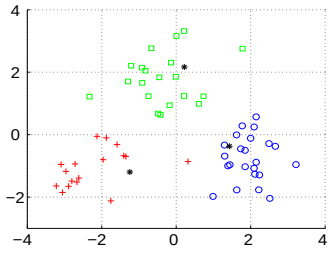

Figure 1: *Noisy Clusters: Original data.* Black stars are three left-out "newbies" from three clusters respectively.

An Example of Simulated Clusters. We simulated three clusters in the two-dimensional Euclidean space. The data points, 63 of them in total, are random samples from three distinct bivariate normal distributions. To check the ability of our method for recovering the clustering structure under noise, we obtained the dissimilarity data using the following procedure. We first added

two noisy coordinates to each data point. These two noisy coordinates follow two independent normal distributions with relatively small variances. The squared Euclidean distances between all pairs of data points, i.e., $d_{ij}s$ in our notation, were then binned into 10 equal sized bins over the interval from the minimum to the maximum of those positive $d_{ij}s$. The value of each d_{ij} was then replaced by the center value of the bin which it belongs to. This is an analog of the scenario where only ranks are provided as the distance/dissimilarity measure. The noisy $d_{ij}s$ were then treated as observed dissimilarity data. Note that the binning procedure here can introduce very none-Euclidean noise.

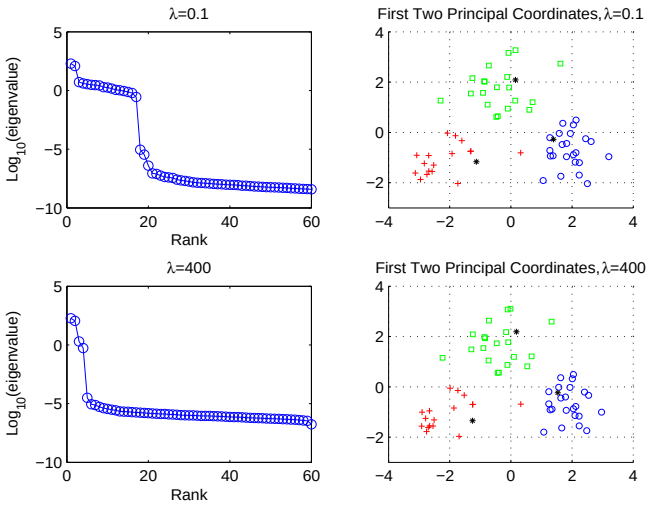

Figure 2: *Noisy Clusters: Effect of λ on the Regularized Kernel Estimate.* The two upper plots are RKE results with $\lambda = 0.1$. The upper left one is the eigensequence plot. The upper right one is the plot of the first two principal coordinates for recovered configuration and "newbies". The two lower plots are RKE results with $\lambda = 400$.

In this simulation, we used all distinct pairwise squared distances. We also saved aside one data point from each cluster to test our "newbie" algorithm. The RKE and newbie formulations we used for this example are quadratic-loss formulations described in the Appendix of [6]. The original clusters are displayed in Figure 1, with different colors and symbols for different clusters. The true newbie positions are marked with black stars. The same colors and marks are used for the RKE recovered configurations in the upper right and lower right plots of Figure 2. In Figure 2, the upper two plots are RKE results with $\lambda = 0.1$, while the lower two are RKE results with $\lambda = 400$. As we can see from Figure 2, both the recovered configurations and the newbie positions are fairly close to the truth. However, the estimated kernel with $\lambda = 0.1$ has many eigenvalues besides the most significant two, corresponding to small noisy dimensions; whereas for the estimated kernel with $\lambda = 400$, most of these noisy eigenvalues "dropped" to machine zero. This clearly shows the desired effect of the trace regularization term promoting dimension reduction. Moreover, when $\lambda = 0.1$, we get the Procrustes measures as defined in Section 2

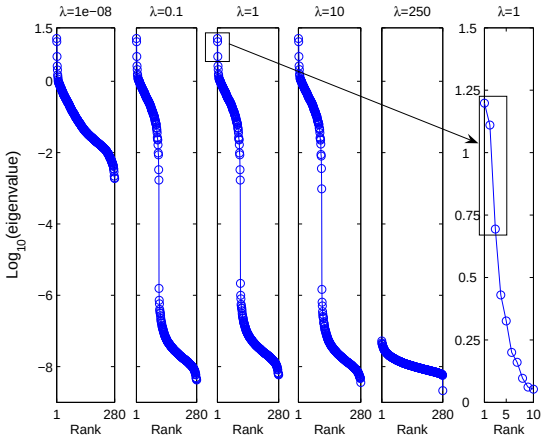

Figure 3: *Left five panels: log scale eigensequence plots for five values of λ.* As λ increases, smaller eigenvalues begin to shrink. Right panel: first ten eigenvalues of the $\lambda = 1$ case displayed on a larger scale.

$\gamma_p = 0.090$ and $\gamma_d = 0.0309$, while when $\lambda = 400$, we have $\gamma_p = 0.0089$ and $\gamma_d = 0.0269$. This suggests that tuning λ in a proper way can improve the accuracy of the estimated kernel.

Protein Clustering and Visualization with RKE for MDS. One of the challenging problems of contemporary biology is inferring molecular functions of unannotated proteins. A widely used successful method of protein function prediction is based on sequence similarity. Statistically significant sequence similarity, which is typically based on a pairwise alignment score between two proteins, forms the basis for inferring the same function. Two major related problems exist for predicting function from sequence. The first problem is the clustering of large number of unlabeled protein sequences into subfamilies for the purpose of easing database searches and grouping similar proteins together. The second problem is concerned with assigning new unannotated proteins to the closest class, given the labeled or clustered training data. We show here that RKE methodology provide an efficient way to represent each protein sequence with a feature vector in an appropriate coordinate system by utilizing the pairwise dissimilarity between protein sequences.

We illustrate the utility of RKE methodology using a dataset of globins that was first analyzed in [8] by a profile HMM approach. The dataset, distributed with the HMMER2 software package [9], has a total of 630 globin sequences. The globin family is a large family of heme-containing proteins with many sub-families. It is mainly involved in binding and/or transportation of oxygen. For illustration purposes, we randomly choose 280 sequences from these data so that three large sub-classes of the globin family (alpha chains, beta chains, myoglobins) are included along with a heterogeneous class containing various types of chains. This selection resulted in a total of 112 "alpha-globins", 101 "beta-globins", 40 "myoglobins", and 27 "globins" (the heterogeneous class). The proportion of sequences in each class were taken to be proportional to the class sizes in the original dataset. We used the RKE formulations (5) and (7) for this application. The `Bioconductor` pack-

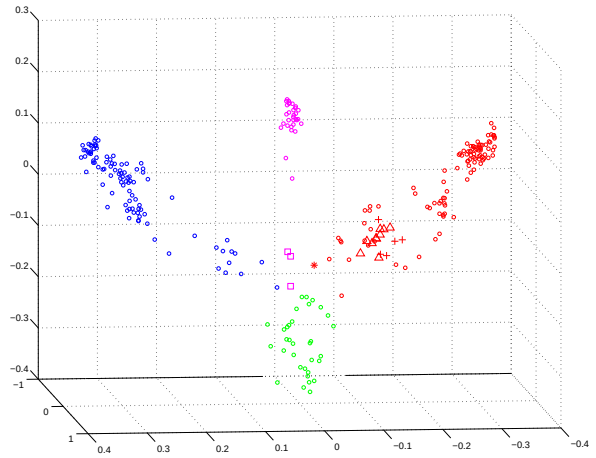

Figure 4: *3D representation of the sequence space for 280 proteins from the globin family.* Different subfamilies are encoded with different colors: Red symbols are alpha-globin subfamily, blue symbols are beta-globins, purple symbols represent myoglobin subfamily, and green symbols, scattered in the middle, are a heterogeneous group encompassing proteins from other small subfamilies within the globin family. Here, hemoglobin zeta chains are represented by the symbol +, fish myoglobins are marked by the symbol □, and the diverged alpha-globin HBAM_RANCA is shown by the symbol *. Hemoglobin alpha-D chains, embedded within the alpha-globin cluster, are highlighted using the the symbol △.

age `pairseqsim` [10] was used to obtain global pairwise alignment (BLAST [11]) scores for all pairs of $N = 280$ sequences. For more implementation details and interesting findings through our RKE results, see [6]. Note that Figure 3 clearly shows the desired dimension reduction effect of the chosen kernel regularization function (trace).

5 RKE for Manifold Unfolding

Manifold Unfolding Problem. One special case of the dimension reduction problem arises often when the goal is to find a meaningful/expected low-dimensional structure behind high-dimensional observations, or more precisely, to recover a low-dimensional parameterization of high-dimensional data assuming all the data lie on a low-dimensional manifold. In several recent papers (see [12] and its references), a large family of algorithms has been proposed to solve this particular type of dimension reduction problem (hereinafter, manifold-unfolding problem), in the spirit of reconstructing the manifold structure globally, but respecting only local information from the observed data. More details on the material presented in this section can be found in our paper [12].

Formulation. We refer to [12] for a detailed derivation of the formulation and emphasize the central ideas here. First, fitting "locally", i.e., constructing Ω or choosing w_{ij} appropriately so that only observed dissimilarities between neighbors (close pairs) appear in the sum of loss, is essential. Second, choosing the kernel regularization function to be negative average squared Euclidean distances among all objects, which after simple calculations can be shown to be proportional to $(NI - E) \cdot K$, where N is again the number objects, I is N-dimensional identity matrix and E is N by N matrix with all elements being 1, is also crucial. The RKE formulation for manifold-unfolding problem is then:

$$\sum_{(i,j)\in\Omega} w_{ij}|d_{ij} - B_{ij} \cdot K| - 2\lambda(NI - E) \cdot K. \qquad (8)$$

Again, this formulation can be posed as a convex cone programming problem (see [12]) which can be solved globally in polynomial time.

"Newbie" Problem. The formulation is similar to (7) with the only exception that Ψ is constructed locally.

Choosing Elements of Ω and Tuning λ. See [12].

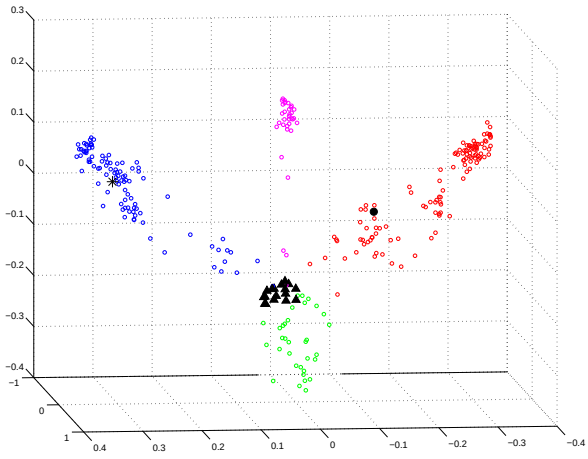

Figure 5: *Positioning test globin sequences in the coordinate system of 280 training sequences from the globin family.* The newbie algorithm is used to locate one Hemoglobin zeta chain (black circle), one Hemoglobin theta chain (black star), and seventeen Leghemoglobins (black triangles) into the coordinate system of the training globin sequence data.

Unfolding the Noisy Wisconsin Roll. This example is specially designed to show the robustness of our method, when compared to the method recently proposed in [13], which has a basic idea similar to ours. We consider two types of noise, which are imposed on the pairwise distances between neighbors after the all neighbors are selected. In this example, the data points are sampled on a "Wisconsin roll" (see Figure $6(a)$), which is a Swiss roll except there is a window in the shape of letter 'W' punched out (thus no points will be sampled within 'W') which can be seen clearly if the roll is flatten out. To impose the first type of noise, twenty percent of the selected pairwise distances are multiplied by a uniform random number over the interval from 0.85 to 1.15. The second type of noise is introduced to all chosen $d_{ij}s$ (between chosen

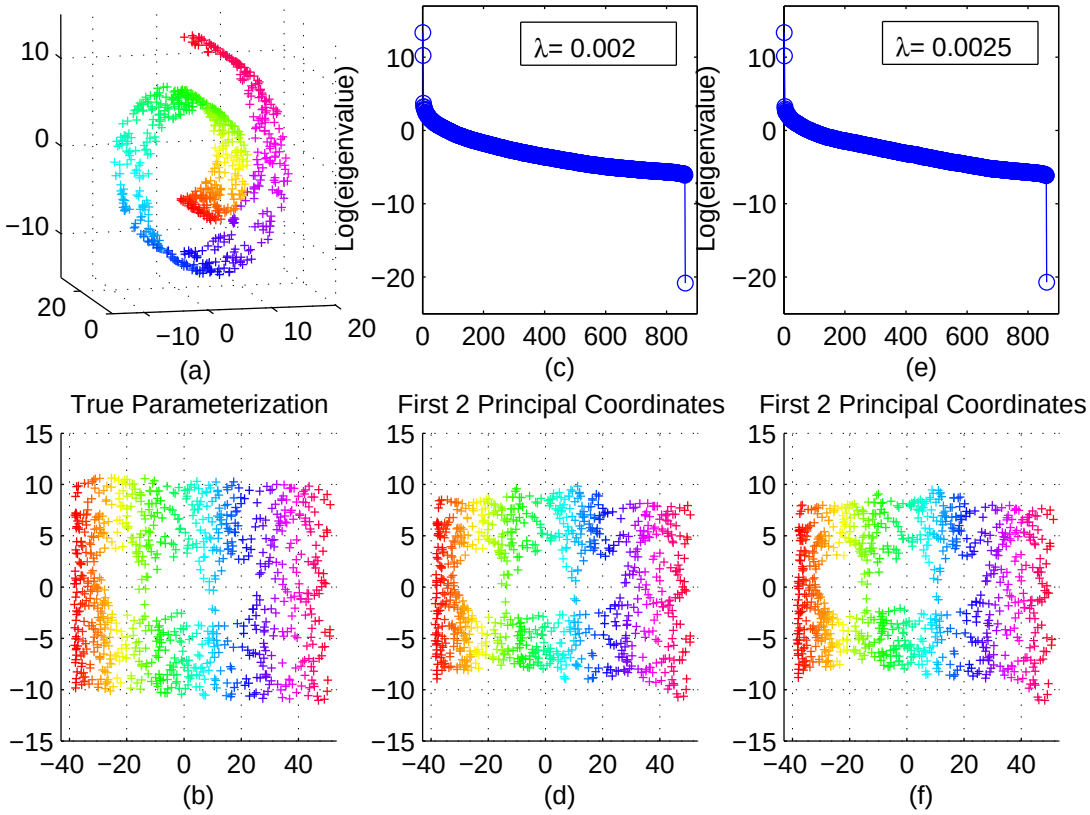

Figure 6: *Wisconsin Roll*. (a) Scatter plot of original data points; (b) True parameterization; (c) & (d) Eigensequence of the solution kernel and Regularized Kernel Embedding of the Wisconsin Roll with first type of noise and $\lambda = 0.002$; (e) & (f) Similar to (c) & (d) but for Wisconsin Roll with second type of noise and $\lambda = 0.0025$.

neighbors) using the binning procedure as used in the simulated example in Section section:mds.

A random sample of 861 points was used for this example with the neighborhood size set to be $k = 6$. In both of these noisy situations, our method successfully (with λ in a proper range) converges to a global optimum with only two significant dimensions (see eigensequences plots displayed in Figures $6(c)$ and $6(e)$). The Procrustes measure in Table 1 below shows our solution is very close to the truth, although the recovered embeddings shown in Figures $6(d)$ and $6(f)$ are slightly distorted from the truth as in Figure $6(b)$ due to the imposed noise.

Table 1: Procrustes Measure between Estimate and Truth

	1st type of noise case	2nd type of noise case
γ_p	0.0055	0.0030
γ_d	0.0154	0.0112

On the contrary, the algorithm in [13] fails to converge because it tries to solve an infeasible primal problem for which the dual is unbounded. For the solvers we used, DSDP5 reported " DSDP: Dual Unbounded, Primal Infeasible" and SDPT3 reported "Stop: primal problem is suspected of being infeasible". These results are expected because when a certain level of noise is directly imposed on the distance information, it is very likely that no Euclidean metric can fit the noisy distance data (for instance if the triangle inequality is violated somewhere). Then problem set-up in [13] is infeasible in the sense that no solution can satisfy all the constraints simultaneously.

Fixing a Broken Stick. Here, we describe a toy example for the purpose of highlighting the difference between our method and the method proposed in [13]. The primary difference between the two methods is that for the method in [13] local distances are enforced rigidly whereas we relax this requirement. We want to show that this relaxation can be very important for manifold-unfolding problems even in the cases without noise.

The data points are randomly sampled on two branches of a 'broken stick' (see top right plot in Figure 7). One branch is from the origin to the point $(1, 1)$ and the other is from $(1, 1)$ to $(2, 0)$. We force the sample to include the point $(1, 1)$. The manifold-unfolding goal here is to flatten out the stick. If any of the pairs for which the squared distance is selected to fit, has one member from the left branch and the other member from the right branch, then

the method in [13] will not be able to flatten out the stick. For our method, a small λ will not flatten out the stick either, but a sufficiently large λ will. The result from employing the method in [13] with $k = 5$ is almost visually indistinguishable from the plot in Fig 10. With $k = 5$ and λ too small ($\lambda = 1e - 5$) , our method also fails to flatten out the stick but recovers the original broken stick. As can be seen in the upper left corner of Figure 7, two outstanding eigenvalues are obtained. However, with λ sufficiently large ($\lambda = 0.3$), we see only one outstanding eigenvalue thus we obtain the one dimensional flattened stick on the lower right corner of Figure 7. As expected, within our regularized kernel embedding framework, the smoothness/dimensionality is controlled by the smoothing/tuning parameter λ.

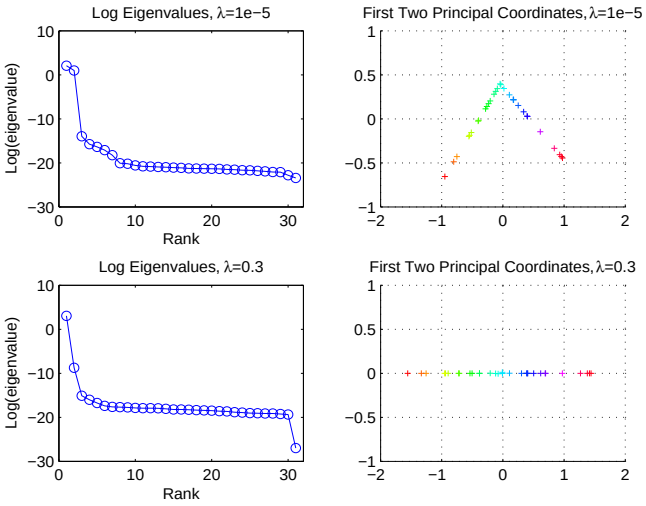

Figure 7: *Broken Stick: Effect of λ on the Regularized Kernel Embedding using (8).* Small λ does not flatten out the stick, but a larger λ does.

6 Discussion

In this paper, we developed a general framework called kernel regularization. We also described two special formulations of the framework for solving two different problems where dimension reduction is promoted through regularizing the kernel estimates. Our methods are robust against noise and provide global solution via modern convex cone programming techniques. It is worth mentioning that if we choose to impose the centering constraint $E \cdot K = 0$ (although we can do without this) in problem (8), the kernel regularization function for manifold unfolding becomes $J(K) = -2(NI - E) \cdot K = -2NI \cdot K = -2N\text{trace}(K)$. Interestingly, in problem (5), the kernel regularization function we use to promote dimension reduction is trace instead of the negative trace (with a constant multiplier). So, the trace regularization function with different signs in front of it both actually promote dimension reduction but only in different scenarios.

Current work in progress includes extensions of both the methodology and the applications including the clustering of proteins at the top level of the protein hierarchy and genome sequence recovering through manifold unfolding. We are also working on developing systematic tuning methods for the RKEs. Future work of interest includes exploring the properties of the alternatives provided here and their applications in other contexts.

References

[1] R. Sibson. Studies in the robustness of multidimensional scaling: Procrustes statistics. *Journal of the Royal Statistical Society Series, B*, 40:234–238, 1978.

[2] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM Studies in Applied Mathematics, v. 13, 1993.

[3] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming*, 95(2):189–217, 2003.

[4] S. J. Benson and Y. Ye. DSDP5: A software package implementing the dual-scaling algorithm for semidefinite programming. Technical Report ANL/MCS-TM-255, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, June 2004.

[5] A. Buja and D. Swayne. Visualization methodology for multidimensional scaling. *Journal of Classification*, 19:7–43, 2002.

[6] **F. Lu**, S. Keles, S. Wright, and G. Wahba. Framework for kernel regularization with application to protein clustering. *Proceedings of the National Academy of Sciences*, 102:12332–12337, 2005.

[7] R. J. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series, B*, 58:267–288, 1996.

[8] A. Krogh, M. Brown, I. S. Mian, K. Sjlander, and D. Haussler. Hidden markov models in computational biology: Applications to protein modeling. *Journal of Molecular Biology*, 235:1501–1531, 1994.

[9] S. R. Eddy. Profile hidden markov models. *Bioinformatics*, 14:755–763, 1998.

[10] R. C. Gentleman, V. J. Carey, D. J. Bates, B. M. Bolstad, M. Dettling, *et al*. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, 5(10):doi:10.1186/gb–2004–5–10–r80, 2004.

[11] S. F. Atschul, W. Gish, W. Miller, E. W. Myers, and D.J. Lipman. A basic local alignment search tool. *Journal of Molecular biology*, 215:403–410, 1990.

[12] **F. Lu**, Y. Lin, and G. Wahba. Robust manifold unfolding with kernel regularization. Technical Report 1108, Department of Statistics, University of Wisconsin-Madison, Madison, WI, Oct 2005.

[13] K. Q. Weinberger, F. Sha, and L. K. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. *Proceedings of the Twenty First International Conference on Machine Learning (ICML-04)*, pages 839–846, 2004.