

## **Statistics 860 Lecture 19** ©G. Wahba 2016

Based on “A Framework for Kernel Regularization With Application to Protein Clustering”, by Fan Lu, Sündüz Keleş Stephen J. Wright and Grace Wahba, PNAS, 102(2005). Open access link available via the TRLIST link on my website:

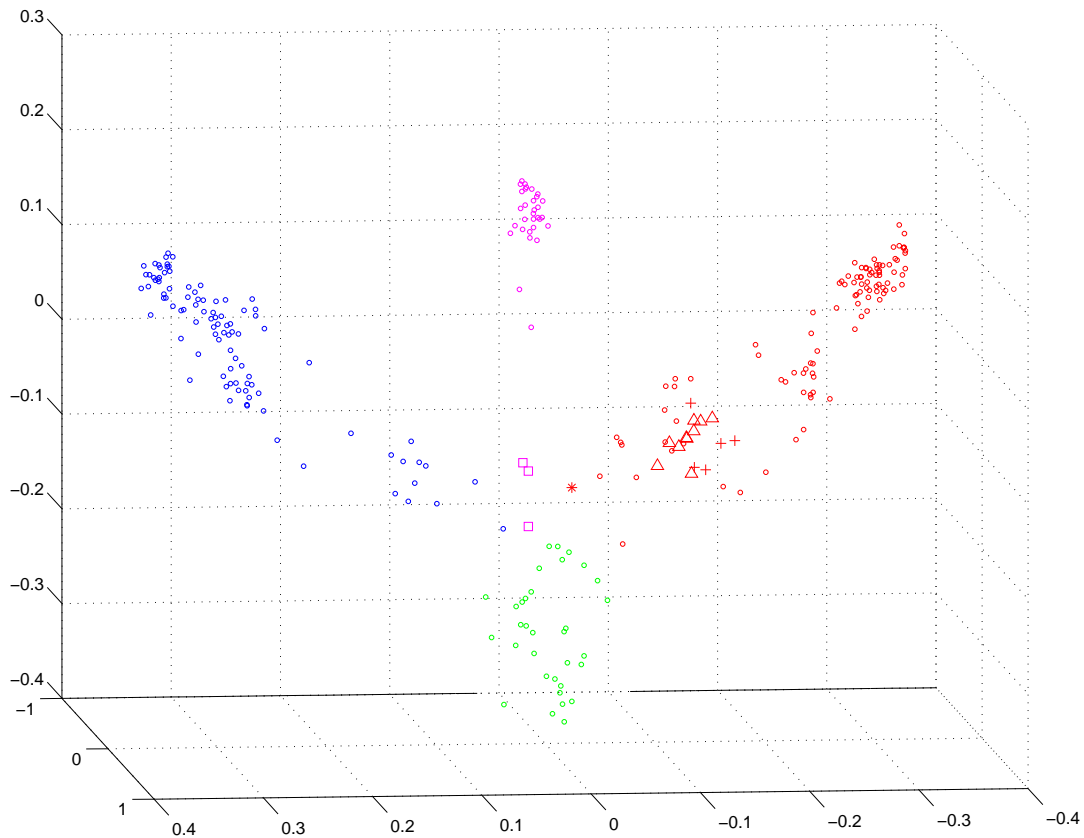
`http://www.stat.wisc.edu/~wahba.or  
lu.etal.pnas05.pdf  
and`

”Robust Manifold Unfolding With Kernel Regularization” by Fan Lu, Yi Lin and Grace Wahba, TR 1108, Oct. 2005. `tr1108rr.pdf`

An approach to supervised/semi-supervised machine learning is described, using Regularized Kernel Estimation. This can be used when only noisy dissimilarity data between objects in a training set is given. A particularly interesting example is provided by the use of BLAST scores in clustering and classifying protein sequence data; the BLAST scores give a pairwise dissimilarity score between protein sequences, labeled or not. A variant of the problem is appropriate as a dimension reduction tool for data that sits in a low dimensional nonlinear manifold in a high dimensional space. Tuning and computation issues and open questions will be mentioned. Next lecture will show how the RKE was used to analyze pedigree information.

## OUTLINE

1. Motivation: To cluster and classify objects for which only dissimilarity information is known (ex. Protein Sequences)
2. Multidimensional Scaling, other related work
3. Dissimilarity information and RKE
4. Special case:  $l_1$  loss and trace penalty.
5. The General Convex Cone Problem
6. The "Newbie" formulation.
7. A numerical trick: subsampling
8. Eigensequence plots, truncation, tuning
9. Clustering and classification of protein sequences
10. Multiple sources of information
11. Robust manifold unfolding



3D representation of the sequence space for 280 proteins from the globin family. Red:  $\alpha$ -globins, blue:  $\beta$ -globins, purple: myoglobins, green: a heterogeneous group of proteins from other small subfamilies within the globin family. Note that there are no units on the axes. The coordinate system has been derived from (noisy, scattered, incomplete) dissimilarity data (BLAST scores) via Regularized Kernel Estimation (RKE).

## ♣♣ Motivation: To Cluster and Classify Objects With Huge or Unconventional Descriptors

In the first figure we saw a multicategory support vector machine classifying clouds that could be characterized by a vector of real numbers that could be scaled to sit in Euclidean  $d$  space. ( $d = 2$  in the plot, but there were actually radiance observations at 12 bandwidths.) Here we are interested in objects like images, texts, microarray gene chips, protein sequences and other objects that do not have natural, or useful easily obtained vectors of real numbers encoding their attributes. What we do assume we have is some dissimilarity information which measures how far apart (in some sense) (a subset of) pairs of objects in the training set are. These may be crude subjective judgments, as might be obtained from a panel studying images, counts of matching words, in the case of texts, projections of extremely large real vectors in the case of gene chips, or similarity information between protein sequences as might be obtained from popular techniques as found in the BLAST algorithms. By the use of regularized kernel estimation (RKE) we can turn the given information into Euclidean vectors.

## ♣♣♣ Multidimensional Scaling, Other Related Work

The work we will talk about is related to Multidimensional Scaling (MDS), a procedure that has been around for many years. However, the regularized kernel estimate we will obtain will work differently. In MDS pairs of distances  $d_{ij}, i, j = 1, \dots, N$  between  $N$  objects are given, generally in a higher dimension than one is interested in. In MDS a prior choice of dimension is selected, and one searches for a set  $\hat{d}_{ij}$  of Euclidean distances in the chosen number of dimensions that best fits the data, typically in a least squares sense, that is, minimize

$$\sum_{i \leq j} (d_{ij} - \hat{d}_{ij})^2.$$

In two or three dimensions, the results can be used to plot and visualize the  $N$  objects and see their relations. Note that the orientation of the plot is not determined. MDS is widely used in psychological studies, an example might be a plot beginning with the percent of agreement between judges.

## ♣♣ Dissimilarity Information and RKE

Given a set of  $N$  objects, suppose we have obtained a measure of dissimilarity,  $d_{ij}$ , for certain object pairs  $(i, j)$ . Regularized Kernel Estimation (RKE): Finds  $K$ :

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

$S_N$  is the convex cone of all real nonnegative definite matrices of dimension  $N$ ,  $\Omega$  is the set of pairs with dissimilarity information  $d_{ij}$ , and the induced dissimilarity  $\hat{d}_{ij}$  is

$$\hat{d}_{ij} = K(i, i) + K(j, j) - 2K(i, j)$$

where  $K(i, j)$  is the  $(i, j)$  entry of  $K$ .  $L$  measures the discrepancy between the observed and induced dissimilarity.  $L$  and  $J$  are convex in  $K$  and  $\lambda$  is a tuning parameter balancing fit to the data and the penalty or complexity on  $K$ .

No restrictions on the set of pairs other than requiring that the graph of the objects with pairs connected by edges be connected.

Observed dissimilarity information may be incomplete, may not satisfy the triangle inequality, 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”. A dimension is not specified in advance.

The literature has many examples of methods for estimating  $K$  from observations for the purpose of clustering and classification. The closest in spirit to the approach here might be Lancriet *et al* (2004) in *Machine Learning* who assume that  $K$  is a linear combination of prespecified kernels and then estimate the coefficients by semidefinite programming.



### ♣♣ Special Case: $l_1$ loss and trace penalty

The special case used here the  $l_1$  loss function and trace penalty:

$$\min_{K \succeq 0} \sum_{(i,j) \in \Omega} |d_{ij} - \hat{d}_{ij}(K)| + \lambda \text{trace}(K). \quad (2)$$

This formulation can be posed as a special case of a general convex cone optimization problem for which efficient software is available. The sum of squares loss function with trace penalty can also be solved with convex cone software.

## ♣♣ The General Convex Cone Problem.

Notation:

- $\mathcal{R}^p$  is Euclidean  $p$ -space
- $P_p$  is the nonnegative orthant in  $\mathcal{R}^p$ , that is, the set of vectors in  $\mathcal{R}^p$  whose components are all nonnegative.
- $Q_q$  is the second-order cone of dimension  $q$ , which is the set of vectors  $x = (x(1), \dots, x(q)) \in \mathcal{R}^q$  that satisfy the condition  $x(1) \geq [\sum_{i=2}^q x(i)^2]^{1/2}$ .
- $S_s$  is the cone of symmetric positive definite  $s \times s$  matrices of real numbers.

### ♣♣ The General Convex Cone Problem (cont).

$$\begin{aligned} \min_{\mathbf{X}_j, \mathbf{x}_i, z} \quad & \sum_{j=1}^{n_s} C_j \cdot \mathbf{X}_j + \sum_{i=1}^{n_q} c_i \cdot \mathbf{x}_i + g \cdot z \quad (3) \\ \text{s.t.} \quad & \sum_{j=1}^{n_s} A_{rj} \cdot \mathbf{X}_j + \sum_{i=1}^{n_q} a_{ri} \cdot \mathbf{x}_i + g_r \cdot z = b_r, \quad \forall_r \\ & \mathbf{X}_j \in S_{s_j} \\ & \mathbf{x}_i \in Q_{q_i} \\ & z \in P_p. \end{aligned}$$

$C_j, A_{rj}$  are real symmetric matrices (not necessarily positive semidefinite) of dimension  $s_j$

$$c_i, a_{ri} \in \mathcal{R}^{q_i},$$

$$g, g_r \in \mathcal{R}^p,$$

$$b_r \in \mathcal{R}^1.$$

The solution of a general convex cone problem can be obtained numerically using publicly available software such as SDPT3 or DSDP5.

SDPT3: Tütüncü, R. H., Toh, K. C. & Todd, M. J. (2003) *Mathematical Programming* **95**, 189–217.

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

## ♣♣ The "Newbie" Formulation

Suppose a solution  $K_N$  has been found for a "training" 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. That is, find a new "pseudo-optimal" kernel  $\tilde{K}_{N+1}$  of the form

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

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

$$\begin{aligned} \min_{b, c \geq 0} & \sum_{i \in \Omega_{\text{newbie}}} \left| d_{i, N+1} - \hat{d}_{i, N+1}(K_{N+1}) \right| \\ \text{s.t.} & \quad b \in \text{Range}(K_N), \quad c - b^T K_N^\dagger b \geq 0. \end{aligned}$$

$K_N^\dagger$  is the pseudo-inverse of  $K_N$  and  $\Omega_{newbie}$  is a subset of  $\{1, 2, \dots, N\}$ . The constraints in this problem are the necessary and sufficient conditions for  $\tilde{K}_{N+1}$  to be positive semidefinite. Note that only a subset of the  $d_{i,N+1}$  are required.

## ♣♣ Missing Data, Subsampling

The algorithm does not require a full set of  $d_{ij}$ . With large data sets it can work quite well with a fraction of the  $N(N - 1)/2$  possible  $d_{ij}$ .

Present algorithms require  $O(m^2)$  storage, where  $m$  is the number of  $d_{ij}$ . To save on computer storage, even when a full set is available, a random subset of the available dissimilarity data is chosen such that each object  $i$  appears with roughly the same frequency among the  $(i, j)$  pairs of  $\Omega$ . For each  $i$ , a fixed number  $k$  of pairs  $(i, j)$  with  $j \neq i$  is chosen.

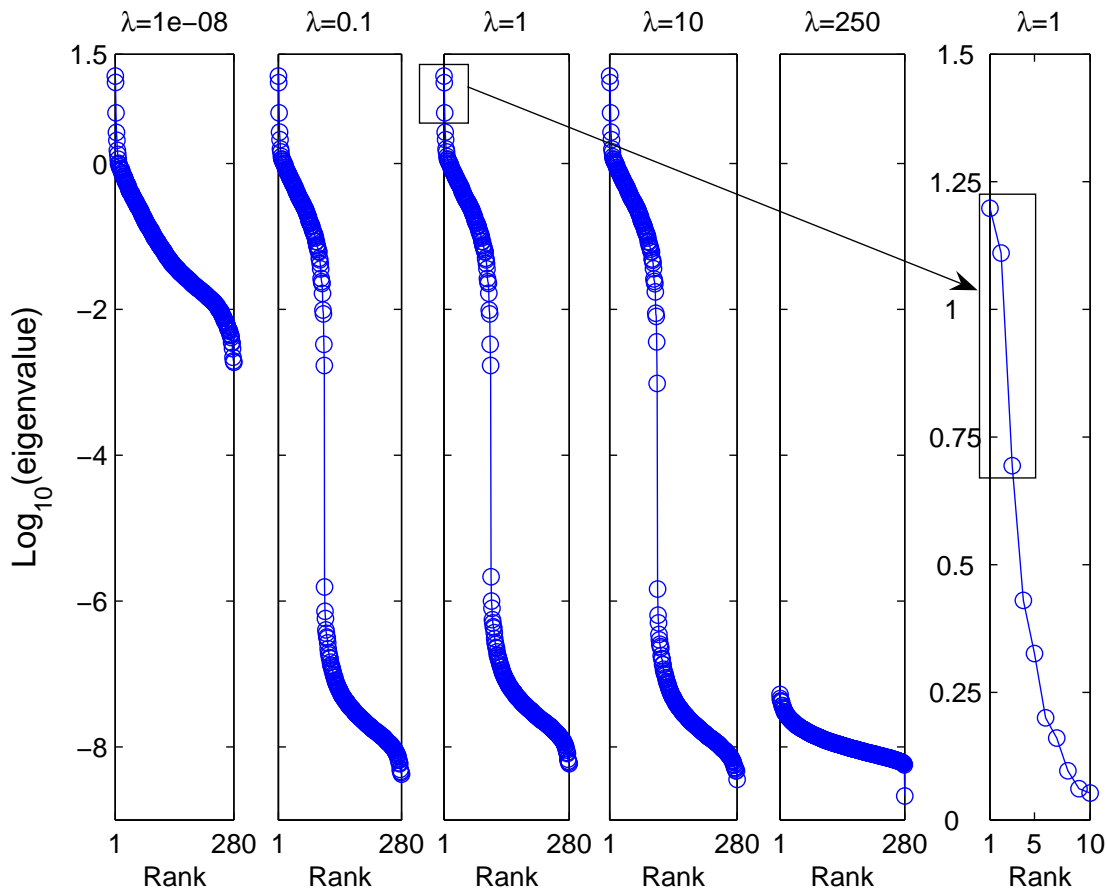
The parameter  $k$  is verified to be sufficiently large when the estimated  $\hat{d}_{ij}$  from several different random subsets does not vary noticeably.

## ♣♣ Eigensequence Plots, Tuning and Truncation

Increasing  $\lambda$  will shrink the eigenvalues of the estimated  $K_\lambda$ . Ideally it will shrink the smaller eigenvalues while keeping the larger ones intact, thus providing clues to the minimum number of dimensions needed to retain most of the dissimilarity information.



## ♣♣ Eigensequence Plots, Tuning and Truncation (cont.)



**Log scale** eigensequence plots for 5 values of  $\lambda$ . Protein sequence example with  $N = 280$ ,  $k = 55$ . Right panel is the  $\lambda = 1$  case on an expanded scale. Note natural breaks appear after both the second and the third eigenvalues. Eyeball method tuning choice:  $\lambda = 1$ , Dimension  $p = 3$ .

Setting all the eigenvalues of  $K$  after the largest  $p$  to 0 results in the  $\nu$ th coordinates of the  $i$ th object as  $x_i(\nu) = \sqrt{\lambda_\nu} \phi_\nu(j)$ ,  $\nu = 1, 2, \dots, p$ ,  $i = 1, \dots, N$ , where the  $\lambda_\nu, \phi_\nu$  are the first  $p$  eigenvalues and eigenvectors of  $K$  and  $\phi_\nu(j)$  is the  $j$  component of  $\phi_\nu$ .

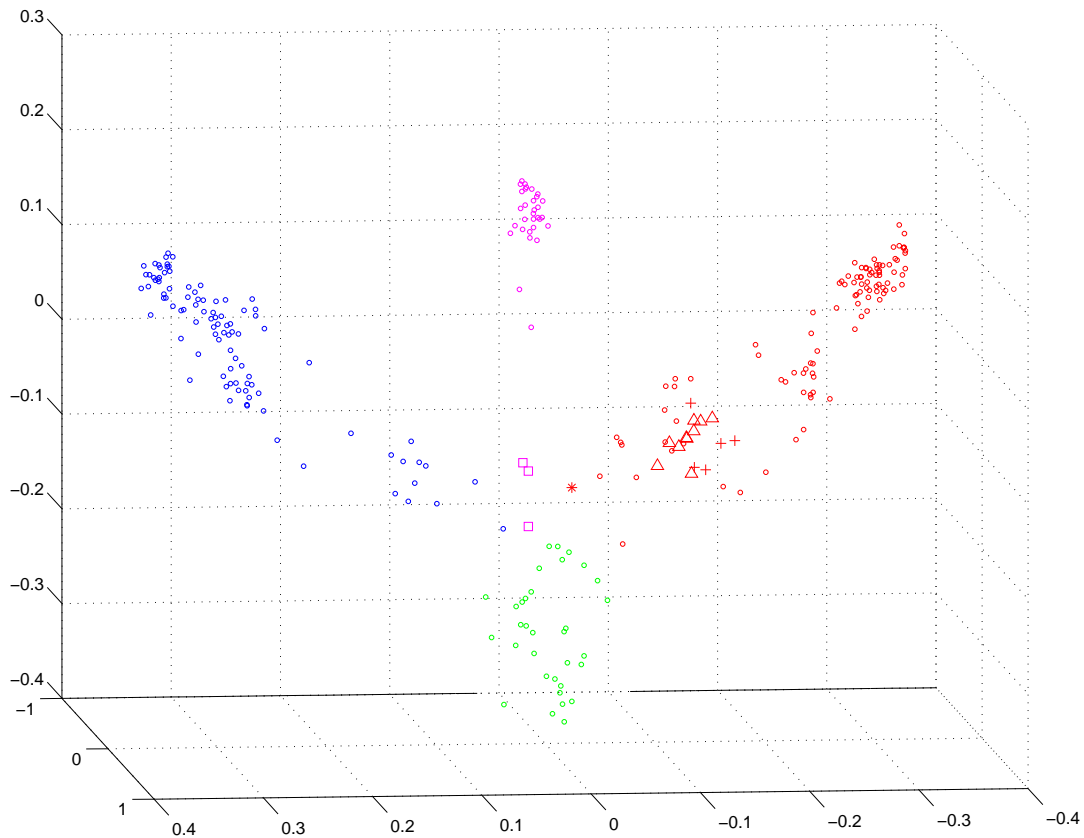
For a newbie, once  $\hat{d}_{i,N+1}$  are found for sufficiently many  $i$ , the coordinates of the newbie are easily found.

The coordinates of the original set of  $N$  objects are always centered at the origin since the RKE estimate of  $K$  always has the constant vector as a 0 eigenvector.

## ♣♣ Clustering and Classification of Protein Sequences

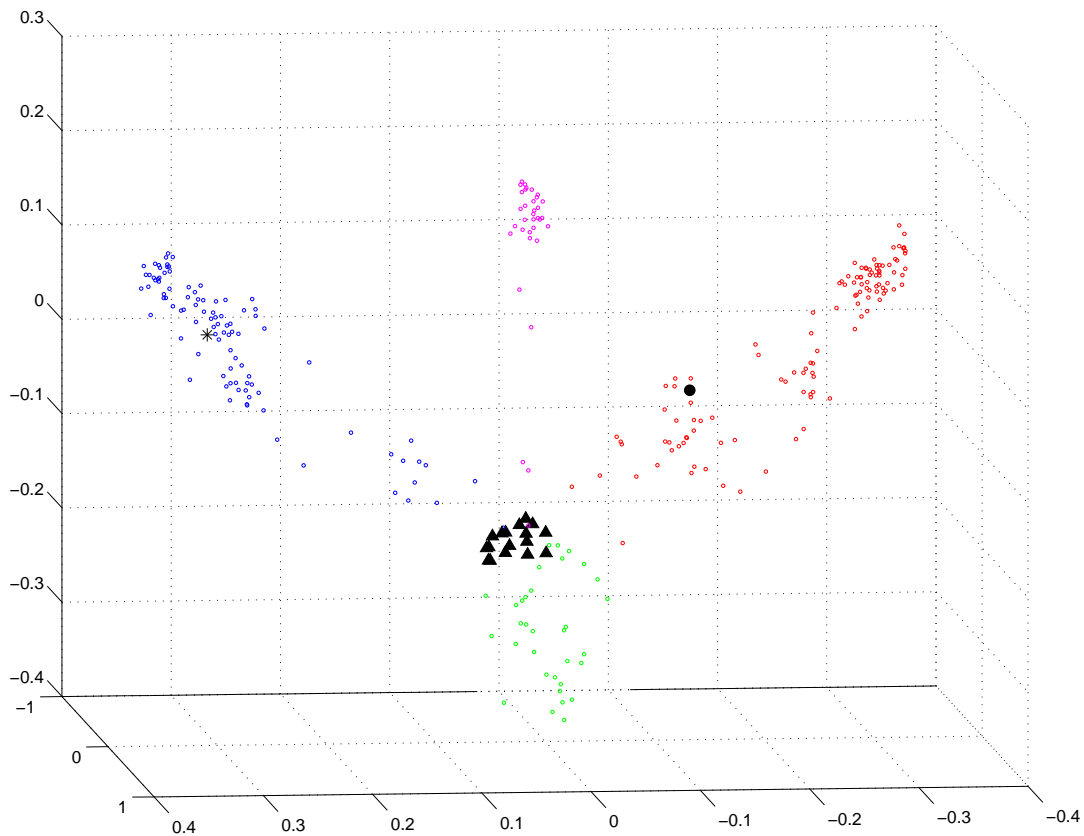
Challenging problem in biology is inferring molecular functions of unannotated proteins. One method for doing this is to examine the sequence similarity between the unannotated protein and a set of annotated proteins - those whose function is understood. The first problem is the clustering of large numbers of protein sequences into subfamilies to group similar proteins together. The second problem is to assign new unannotated proteins to the closest class, given labeled or clustered training data. (Much literature).

Example: 630 Globin sequences. Chose 280 sequences including three large sub-classes of the globin family (112 alpha globin, 101 beta globins, 40 myoglobins, 27 globins (a heterogeneous category)). The `Bioconductor` package `pairseqsim` was used to obtain global pairwise alignment scores for all pairs of the  $N = 280$  sequences.  $k = 55$ , or about 36% of the total possible.



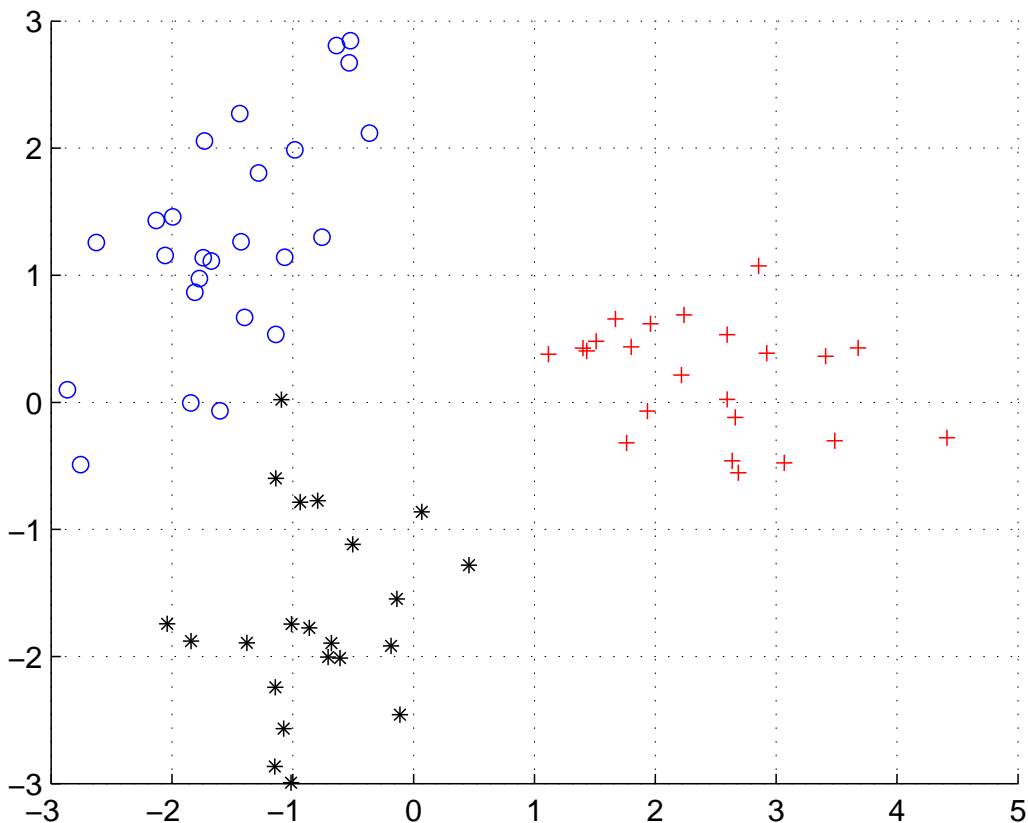
3D representation of the sequence space for 280 proteins from the globin family. Different subfamilies are encoded with different colors: Red symbols are alpha-globin subfamily, blue symbols are beta-globins, purple symbols represent myoglobin subfamily, and green symbols, scattered in the middle, are 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 purple box symbol 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 **△**.



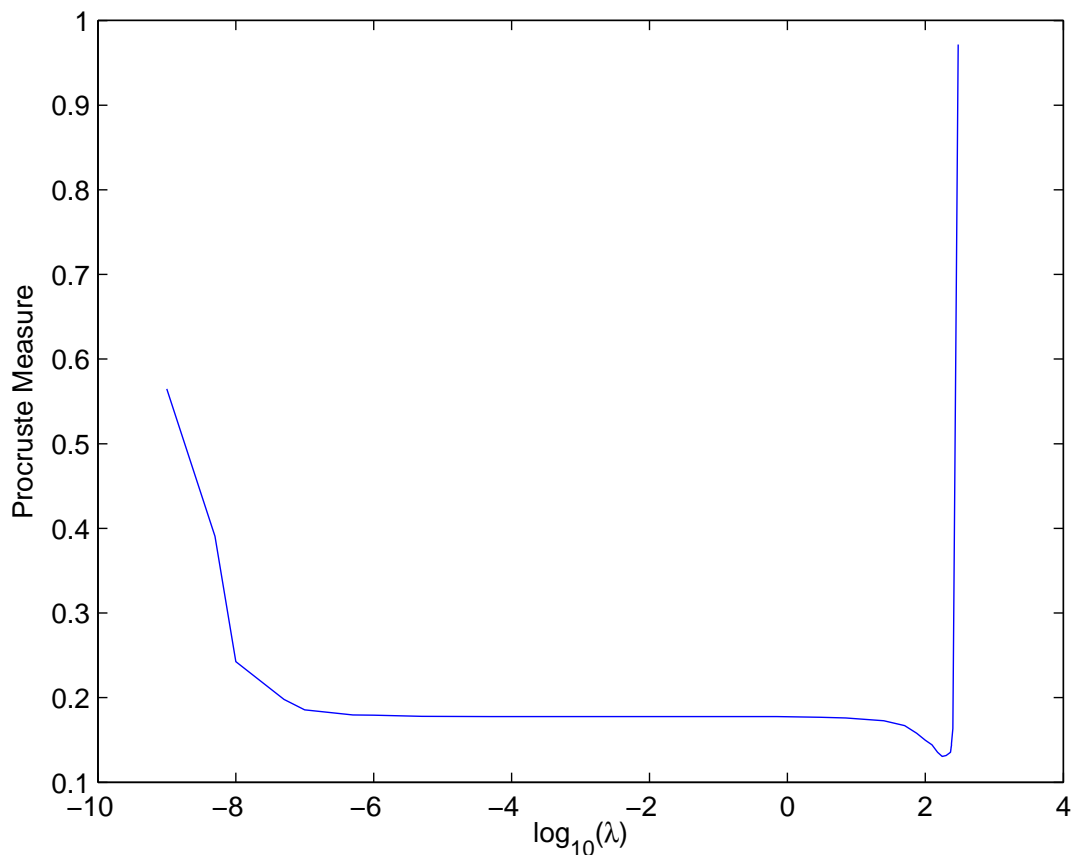
Positioning test globin sequences in the coordinate system of 280 training sequences from the globin family. The newbie algorithm is used to locate one Hemoglobin zeta chain (black circle), one Hemoglobin theta chain (black star), and seventeen Leghemoglobins (black triangles) into the coordinate system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

♣♣ CV2, pairwise cross validation. **WORK IN PROGRESS**



Plot of toy (signal) data for test of CV2, pairwise cross-validation. Noisy data was generated by adding two dimensions of small Gaussian noise, ( $\sigma \approx .3$ ) then binning the observed pairwise (four dimensional) distances into 10 equally spaced bins.

## ♣♣♣ CV2, pairwise cross validation (cont.)



Normalized “Procrustes Measure” of the distance of  $K_\lambda$  from  $K$ , the Gram matrix of the (toy) signal data:

$$d_{K, K_\lambda} = \frac{\text{tr}K + \text{tr}K_\lambda - 2\text{tr}[K^{1/2}K_\lambda K^{1/2}]^{1/2}}{\sqrt{\text{tr}K \text{tr}K_\lambda}}.$$

Note minimum around  $\log_{10}(\lambda) = 2.25$ .

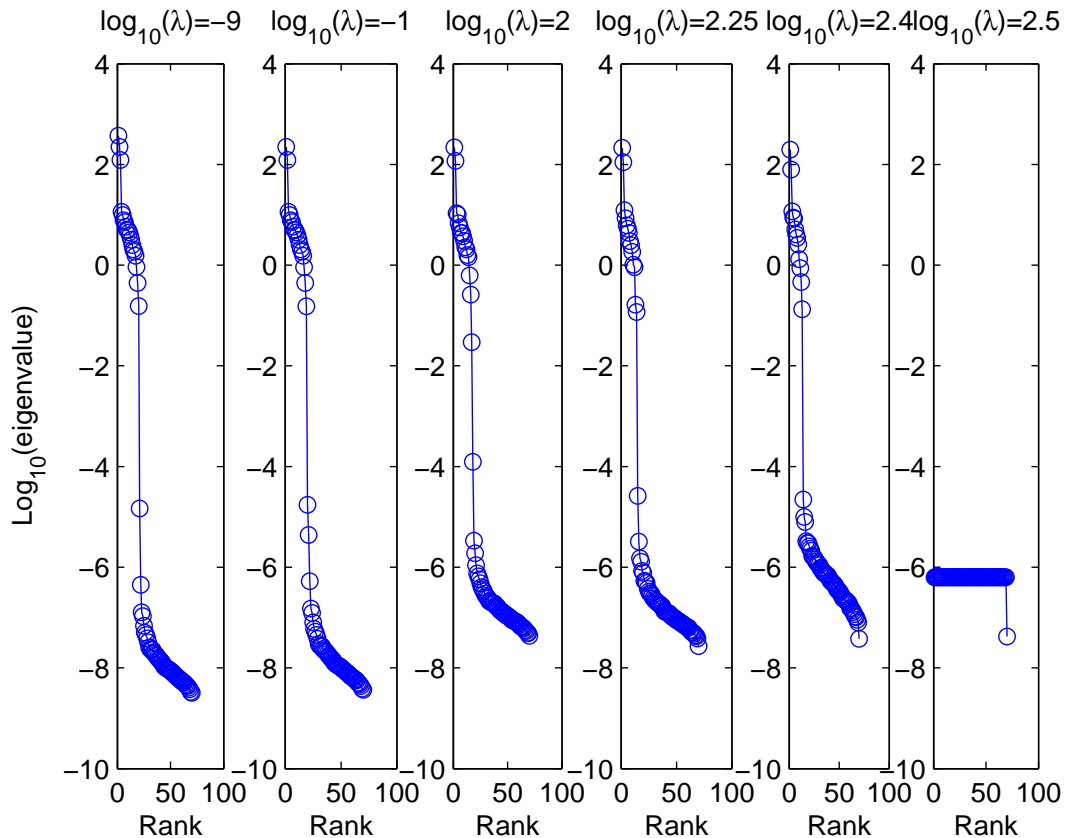


An object space measure of distance between two kernels is

$$d_{K_\alpha, K_\beta} = \min_{M_{\alpha\beta} \in \mathcal{O}} \|X_\alpha - M_{\alpha\beta} X_\beta\|^2$$

where  $\mathcal{O}$  is the class of rotation matrices and the  $X'_s$  are arrays of the fitted coordinates:  $X'_\alpha X_\alpha = K_\alpha$ , similarly for  $X_\beta$ . Interestingly enough, this gives the same result as the unnormalized "Procrustes Measure", (See "Wahba's Problem", SIAM Review 8, 384-386 (1966))  
wahbasproblem.pdf

## ♣♣ CV2, pairwise cross validation.



Eigensequences as a function of  $\lambda$  for the  $K_\lambda$  for the noisy toy data. Recall that the minimum of the procrustes measure is  $\log_{10}(\lambda) \approx 2.25 - 2.4$ .

## ♣♣♣ CV2, pairwise cross validation (cont.)

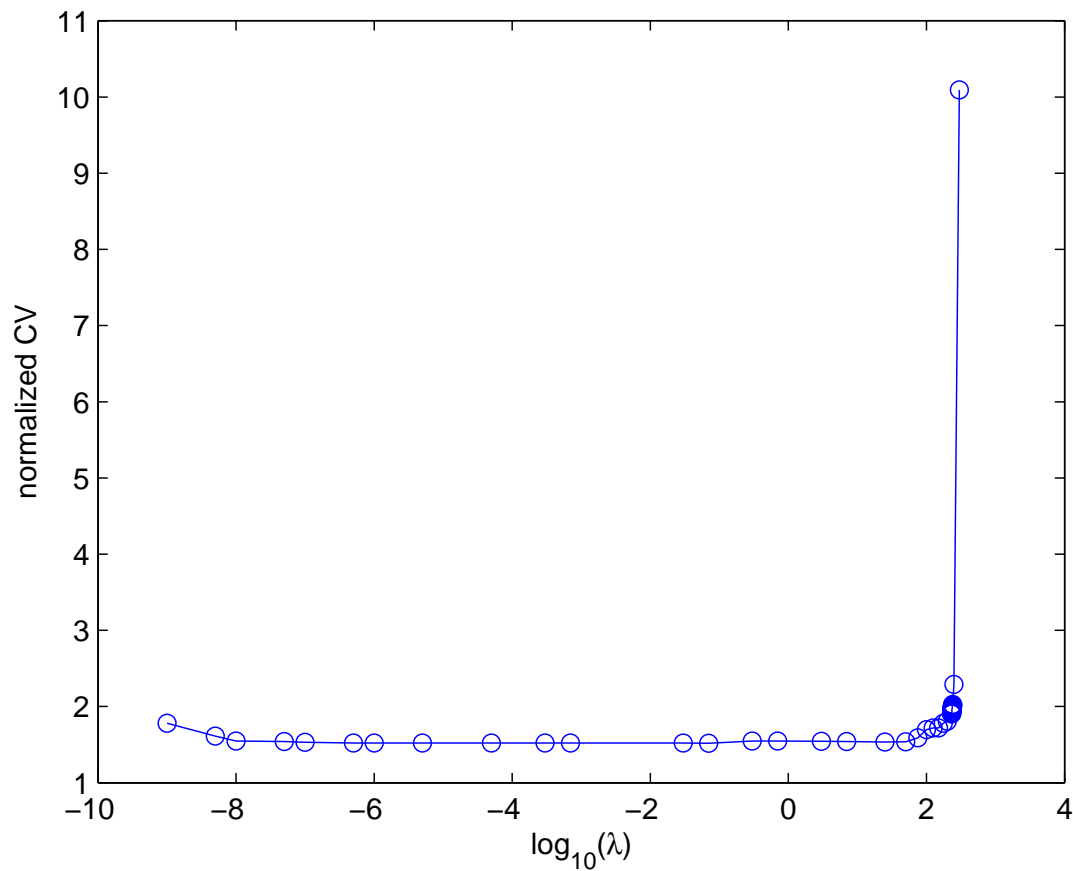
To do CV2, the pairwise crossvalidation, a separate tuning set with dissimilarity information is needed.

1. Fix  $\lambda$ . Solve the newbie problem separately for objects  $r$  and  $s$  in the tuning set to get  $\hat{d}_{ir}$  and  $\hat{d}_{is}$  for selected  $i = i_1, \dots, i_L$ .
2. Using  $\hat{d}_{ir}$  and  $\hat{d}_{is}$  for  $i_1, \dots, i_L$  obtain the newbie coordinate vectors  $x_r = x_r(\lambda)$  and  $x_s = x_s(\lambda)$ .
3. Compare the predicted distances  $|x_r - x_s|$  to the observed dissimilarities  $d_{rs}$  to get  $CV(\lambda)$ :

$$CV(\lambda) = \sum_{r,s} | |x_r(\lambda) - x_s(\lambda)| - d_{rs} |.$$

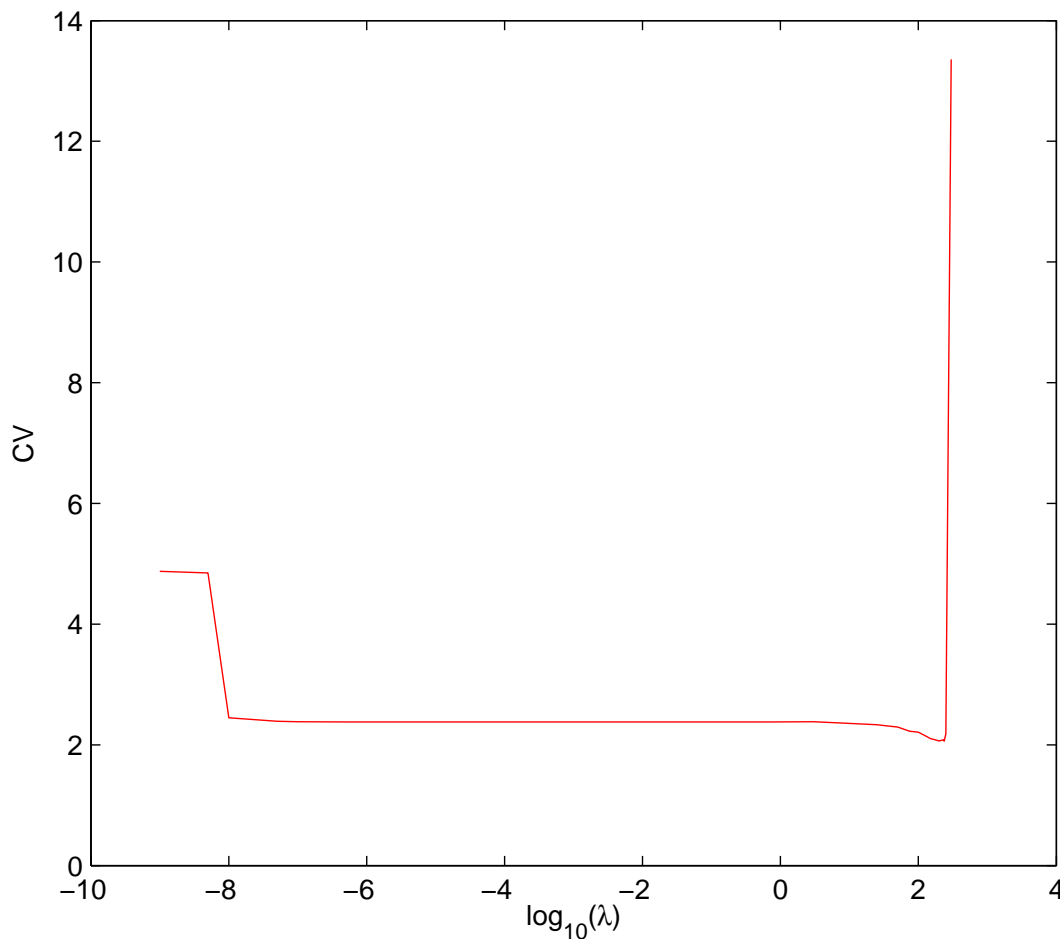
4. Increment  $\lambda$ .

♣♣ CV2, pairwise cross validation.



CV2 plot. Tuning set of 30 objects to give  $\binom{30}{2}$  pairs. Using 95% of the trace to evaluate the CV2.

♣♣♣ CV2, pairwise cross validation.



CV2 plot. Tuning set of 30 objects to give  $\binom{30}{2}$  pairs. Using only the rank 2 approximation to  $K_\lambda$  to evaluate CV2. (In this toy problem we know that the "right" answer is 2.)

## ♣♣ Multiple Sources of Information

In the case of proteins, there are other sources of information than sequence data, namely various kinds of structure information. The first question that arises when there are different sources of information might be: Are they telling you the same thing? When only (relative) distance information is available, the "Procrustes" measure provides pairwise dissimilarities between RKE's. Alternatively a simple measure of correlation is:

$$\sum_{ij} \hat{d}_{ij\alpha}^{s/2} \hat{d}_{ij\beta}^{s/2} / ((\sum_{ij} \hat{d}_{ij\alpha}^s)^{1/2} (\sum_{ij} \hat{d}_{ij\beta}^s)^{1/2})$$

where  $\alpha$  and  $\beta$  index the different sources of information and  $s$  is a real number to be chosen.

## ♣♣♣ Combining Multiple Sources of Information

With labeled data, these kernels can further be examined and combined in an optimal way for classification. One way to do that follows by letting

$$\mathcal{K} = \sum_{\alpha=1}^p \mu_{\alpha} K_{\alpha} \quad (4)$$

and tuning the  $\mu$  to optimize the classification in a Support Vector machine:

G. Lanckriet, N. Cristianini, P. Bartlett, and ElGahoui. A learning the kernel matrix with semidefinite programming. *J. Mach. Learning Res*, 5:27–72, 2004.

Numerous proposals have been made for in-sample tuning of SVM's, along with those in Lanckriet et al, see

G. Wahba, Y. Lin, Y. Lee, and H. Zhang. On the relation between the GACV and Joachims'  $\xi_{\alpha}$  method for tuning support vector machines, with extensions to the non-standard case. Technical Report 1039, Statistics Department University of Wisconsin, Madison WI, 2001. [tr1039.pdf](#)

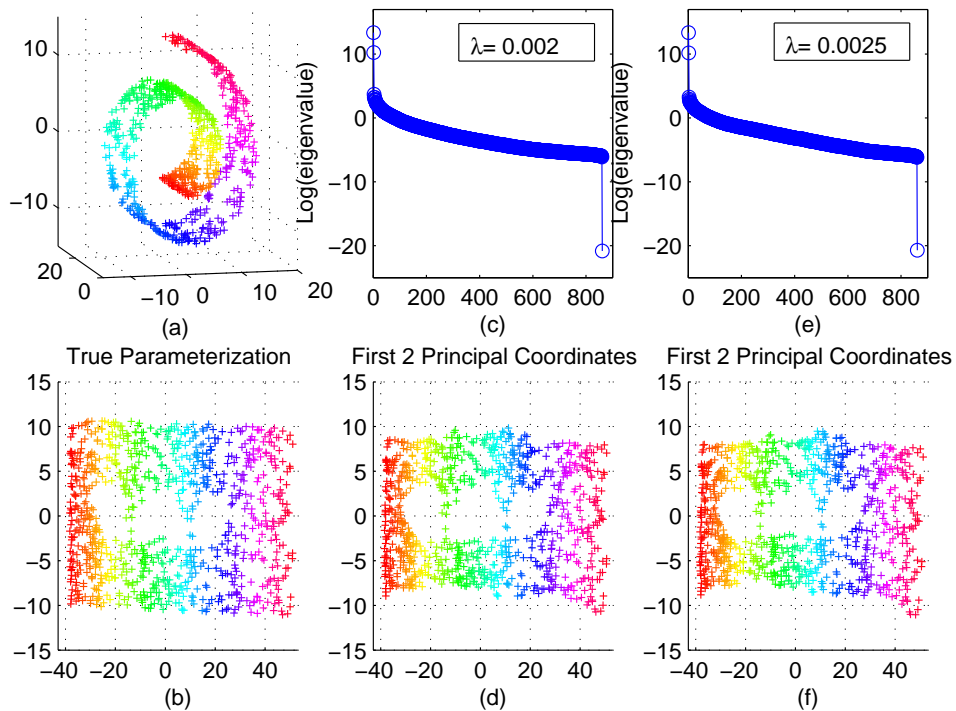
and elsewhere for references.

Alternatively, given a fitted kernel for one source of data, an interesting question is how new information from a different source is best incorporated. The answer can be expected to be different according to whether classification (given a labeled training set) or clustering (with unlabeled data) is the goal.

Numerous open questions.



## ♣♣♣ Robust Manifold Unfolding



Unrolling the Wisconsin Roll (a.k.a. "Swiss Roll") (a), (b) "truth" and truth unrolled, (c), (d) eigenvalues and flattened manifold, noise case 1, (e), (f) noise case 2.

$$\min_{K \succeq 0} \sum_{6-nn \text{ pairs}(i,j)} |d_{ij} - \hat{d}_{ij}(K)| - \lambda \text{trace}(K).$$

F. Lu, Y. Lin, and G. Wahba. Robust manifold unfolding with kernel regularization. TR 1008, Department of Statistics, University of Wisconsin, Madison WI, October, 2005.  
tr1108rr.pdf.