IPAM Graduate Summer School: Intelligent Extraction of Information From Graphs and High Dimensional Data

Grace Wahba

Tutorial III Regularized Kernel Estimation (RKE) From Dissimilarity Data

Based on ”A Framework for Kernel Regularization With Application to Protein Clustering”, by Fan Lu, Sündüz Keleş Stephen J. Wright and Grace Wahba. Contributed to PNAS. Available via the TRLIST link on my website:
`http://www.stat.wisc.edu/~wahba`.
as TR 1107, June 2005.
See TR1107 for references.

July 14, 2005

Regularized Kernel Estimation (RKE) From Dissimilarity Data

Abstract

In many problems approximate ”distances” or dissimilarity data may be available between objects, for example protein sequences, where various methods for coming up with similarity measures are available for implementation. The end goal of this work is to be able to take noisy, incomplete ”distance” data and obtain a ’best-fitting’ positive definite function. Since this positive definite function can provide a consistent (Euclidean) set of coordinates, the end result can be used to cluster unlabeled objects. Furthermore if a labeled training set is available, new unlabeled objects may be fit in the coordinate scheme obtained so far via a ’newbie’ algorithm to be described. Then the (multicategory) support vector machine or penalized likelihood estimate can be used to classify the new object or estimate the likelihood that it is in various classes. An application to the classification of protein sequences will be discussed.

OUTLINE

1. Motivation: To Cluster and Classify Protein Sequences, Other Applications.

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 Objects With Huge or Unconventional Descriptors

10. Multiple Sources of Information

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

In Lecture II we have seen how penalized likelihood and support vector machines work to study and classify objects that can be characterized by a vector of real numbers that can be scaled to sit in Euclidean d space. In this lecture 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 reglarized 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 $d_{ij}, i, j = 1, \cdots N$ 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. (See also L. Saul talks)

♣♣ 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:

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

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

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

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 λ 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 $x = \Big(x(1), \ldots, x(q)\Big) \in \mathcal{R}^q$ that satisfy the condition $x(1) \geq [\sum_{i=2}^{q} x(i)^2]^{1/2}$.
- S_s is the cone of symmetric positive definite $s \times s$ matrices of real numbers.

♣♣ The General Convex Cone Problem (cont).

$$\min_{X_j, x_i, z} \quad \sum_{j=1}^{n_s} C_j \cdot X_j + \sum_{i=1}^{n_q} c_i \cdot x_i + g \cdot z \qquad (3)$$

$$\text{s.t.} \sum_{j=1}^{n_s} A_{rj} \cdot X_j + \sum_{i=1}^{n_q} a_{ri} \cdot x_i + g_r \cdot z = b_r, \quad \forall_r$$

$$X_j \in S_{s_j}$$
$$x_i \in Q_{q_i}$$
$$z \in P_p.$$

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:

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

$K_N^\dagger$ is the pseudo-inverse of K_N and Ψ is a subset of $\{1, 2, \ldots, N\}$ of size t. 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}.

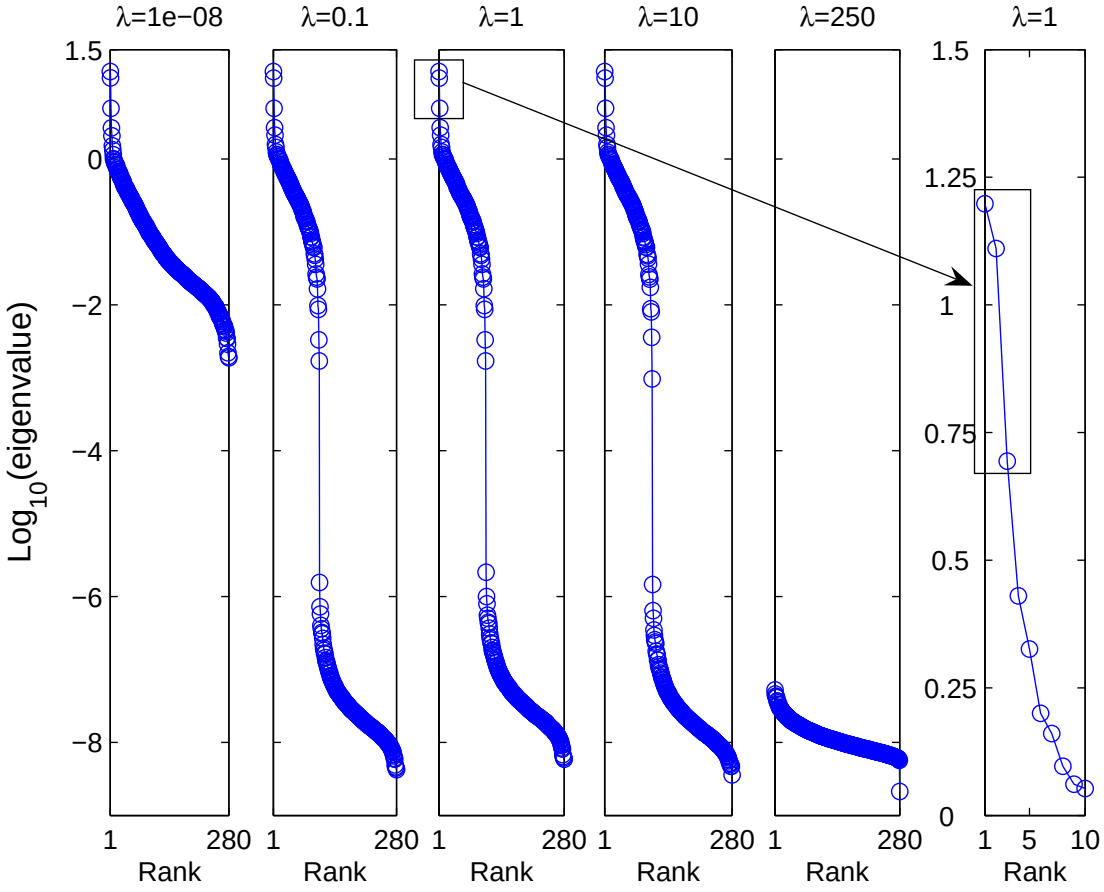
Present algorithms require $O(m^2)$ storage, where m is the number of d_{ij}. To save on computer storage, even when a full set is available, a random subset of the available dissimilarity data is chosen such that each object i appears with roughly the same frequency among the (i,j) pairs of Ω. For each 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 $\widehat{d}_{ij}$ from several different random subsets does not vary noticeably.

♣♣ Eigensequence Plots, Tuning and Truncation

Increasing λ will decrease the trace of the estimated K, it is expected that the effective rank, that is, the rank after truncating negligible engenvalues will decrease.

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



Log scale eigensequence plots for 5 values of λ. 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.

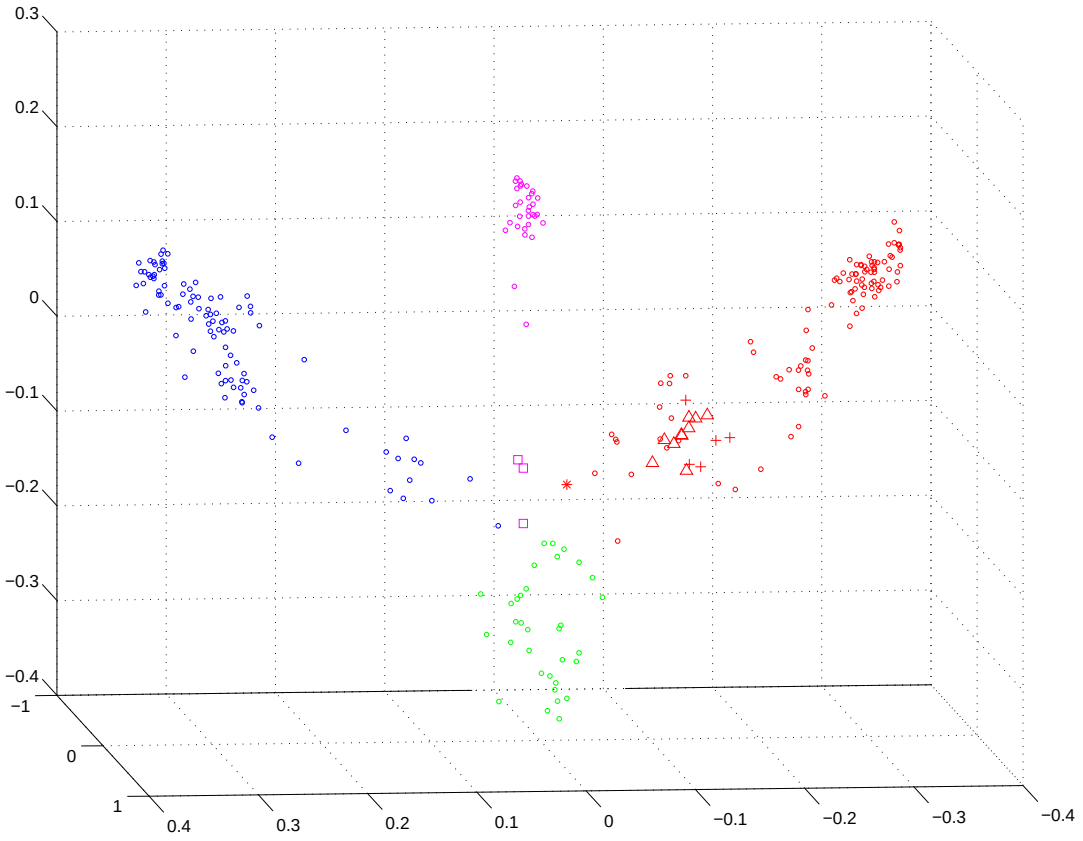
Setting all the eigenvalues of K after the largest p to 0 results in the νth coordinates of the jth object as $x_j(\nu) = \sqrt{\lambda_\nu}\phi_\nu(j), \nu = 1, 2, \ldots, p$, where the λ_ν, ϕ_ν are the first p eigenvalues and eigenvectors of K and $\phi_\nu(j)$ is the j component of ϕ_ν.

The coordinates of the 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

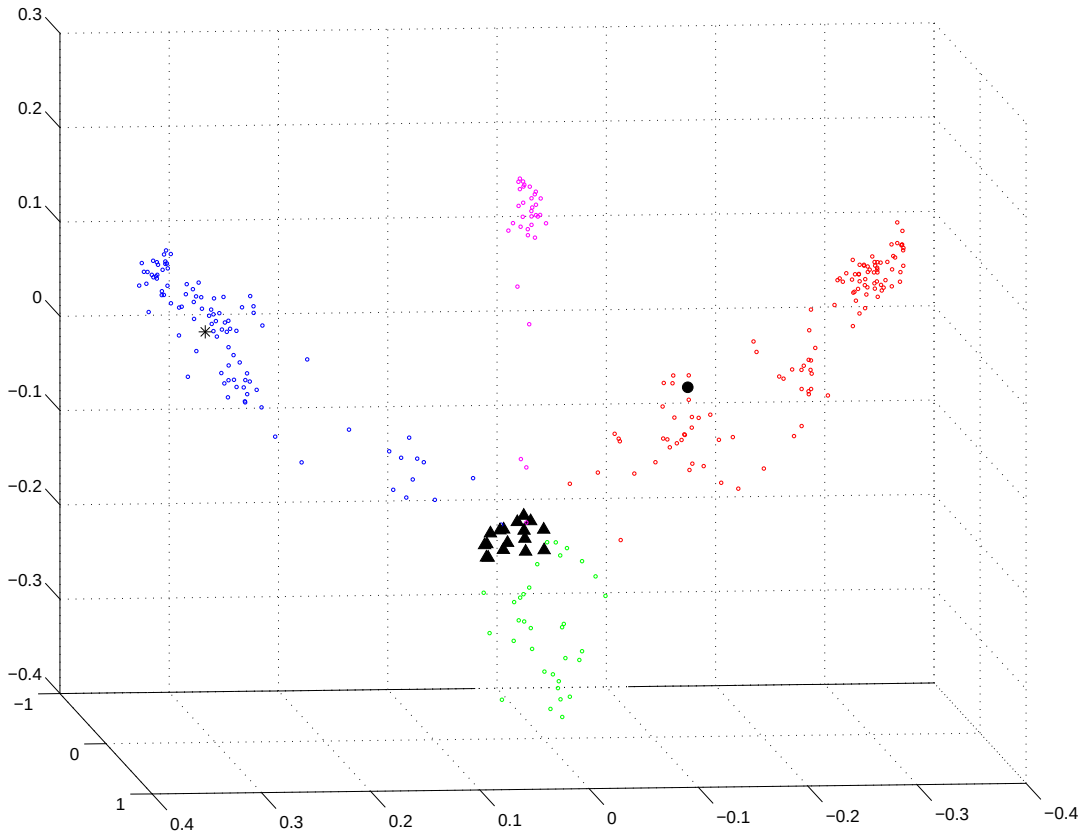
Challenging problem in biology is inferring molecular functions of unanotated proteins. One method for doing this is to examine the sequence similarity between the unannotated protein and a set of annotated proteins - those whose function is understood. The first problem is the clustering of large numbers 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-classis of the globin family (112 alpha globin, 101 beta globins, 40 myoglobins, 27 globins (a heterogenous 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 myglobin 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 myglobins 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.

♣♣ 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? Since only (relative) distance information is available, a measure of correlation between the two sets of (induced) dissimilarity data might be: A simple example is a measure of correlation:

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

where α and β index the different sources of information and s is a real number to be chosen. With labeled data, these kernels can further be examined and combined in an optimal way for classification, as for example in Lancriet *et al*. Alternatively, given a fitted kernel for one source of data, an interesting question would be 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.