SLAC Computation Research Group
Stanford, California

CGTM No. 157
October 1974

HOLMES -- Computer Program for Two-Dimensional Projection Pursuit

J. H. Friedman

Abstract

A computer program that implements the Projection Pursuit algorithm of Friedman and Tukey is described. This routine seeks to find two-dimensional linear projections of multivariate data that are relatively highly revealing.

INTRODUCTION

HOLMES is a collection of subroutines for finding those linear two-dimensional projections of multivariate data that reveal interesting structure. The algorithm and its properties are detailed elsewhere[1] and are only discussed briefly here as they relate to its implementation.

Projections onto two dimensions are characterized by two directions $\hat{k}$ and $\hat{\ell}$ (conveniently taken to be orthogonal with respect to the initially given coordinates and their scales). These directions define a two-dimensional plane embedded in a multidimensional space.

The Projection Pursuit algorithm assigns to each orientation of such a plane in the multidimensional space, a numerical index, $I(\hat{k},\hat{\ell})$, that corresponds to the degree of structuring present when the data is projected onto it. The more structure present in the projection, the larger $I(\hat{k},\hat{\ell})$ becomes. The essence of the algorithm is to find those projection orientations (defined by the directions of $\hat{k}$ and $\hat{\ell}$) that maximize $I(\hat{k},\hat{\ell})$. For complex data structures, several solutions may exist, and for each of these, the solution projections can be visually inspected by the researcher for interpretation and judgment as to their usefulness. This multiplicity is often important.

The input to the algorithm consists, principally, of the multivariate data point set and starting directions, $\hat{k}_s$ and $\hat{\ell}_s$, in the multidimensional data space. Starting with the plane defined by $\hat{k}_s$ and $\hat{\ell}_s$, the algorithm finds the directions $\hat{k}^*$ and $\hat{\ell}^*$ that represent the first local maximum of $I(\hat{k},\hat{\ell})$ uphill from the starting directions $\hat{k}_s$ and $\hat{\ell}_s$. Various useful starting directions include the larger principal axes[3] of the data set, the original coordinate axes, and even directions chosen at random. From these various searches, several quite distinct solutions may result. Each of these projections can then be examined to determine their usefulness in data interpretation.

THE PROJECTION INDEX

The intent of the projection index is to assign to each two-dimensional planar orientation $(\hat{k},\hat{\ell})$ in the multivariate space, a numerical value that closely corresponds to the degree of data structuring present when the multivariate data is projected onto the plane. It was found[2] that those projections that were most interesting to researchers were those that tended to produce many very small interpoint distances while, at the same time, maintaining the overall spread of the data in the projection. Such projections will, for instance, tend to concentrate the points into clusters while, at the same time, separating the clusters.

The spread of the data as projected onto the plane, $s(\hat{k},\hat{\ell})$, is estimated by taking the product of the trimmed standard deviations of the data from the means along each direction, $\hat{k}$ and $\hat{\ell}$. That is

$$s(\hat{k},\hat{\ell}) = s(\hat{k}) \cdot s(\hat{\ell})$$

where

$$s(\hat{k}) = \left[\sum_{i=pN}^{(1-p)N} (\vec{X}_i \cdot \hat{k} - \bar{X}_k)^2 / (1-2p)N \right]^{1/2}$$

and (1)

$$\bar{X}_k = \sum_{i=pN}^{(1-p)N} \vec{X}_i \cdot \hat{k} / (1-2p)N$$

Here N is the total number of data points, and $\vec{X}_i$ (i=1,N) are the multivariate vectors representing each of the data points ordered according to their projections $\vec{X}_i \cdot \hat{k}$. A small fraction, p, of the points that lie at each of the extremes of the projections are omitted from both sums. Thus, extreme values of $\vec{X}_i \cdot \hat{k}$ do not contribute to $s(\hat{k})$, which is thus robust against extreme outliers. The expression for $s(\hat{\ell})$ is the same as eqn 1 with $\hat{\ell}$ replacing $\hat{k}$.

For a "small interpoint distance" measure, an average nearness function is used which has the form

$$d(\hat{k},\hat{\ell}) = \sum_{i=1}^{N} \sum_{j=1}^{N} (R^2 - r_{ij}^2)\, 1(R^2 - r_{ij}^2) \qquad (2)$$

where

$$r_{ij}^2 = (\vec{X}_i \cdot \hat{k} - \vec{X}_j \cdot \hat{k})^2 + (\vec{X}_i \cdot \hat{\ell} - \vec{X}_j \cdot \hat{\ell})^2$$

and $1(\eta)$ is unity for positive valued arguments and zero for negative values. Thus, the double sum is confined to pairs with $0 \leq r_{ij} \leq R$.

The projection index, $I(\hat{k},\hat{\ell})$, which defines the degree of structuring present when the data is projected onto the plane is taken to be

$$I(\hat{k},\hat{\ell}) = d(\hat{k},\hat{\ell}) \cdot s(\hat{k},\hat{\ell}) \quad . \qquad (3)$$

The cutoff radius, R, in eqn 2 is a parameter of the algorithm whose value is set by the user. Its value establishes the distance in the projected subspace over which the local density is averaged, and thus its value establishes the minimum scale of density variation detectable by the algorithm. A choice for its value can be influenced by the global scale of the data, as well as any information that may be known about the nature of the variations in the multivariate density of the points. The sample size is also an important consideration since the radius should be large enough to include, on the average, enough points (in each projection) to obtain a reasonable estimate of the local density. These considerations usually result in a compromise, making R as small as possible, consistent with the sample size requirement.

PURSUIT STRATEGY

The projection index, $I(\hat{k},\hat{\ell})$, for a two-dimensional plane embedded in an n-dimensional space is defined by eqns 1, 2 and 3. This index is a function of the $2n-4$ parameters that define such a plane. A straightforward

pursuit strategy would seek the maximum of $I(\hat{k},\hat{\ell})$ with respect to these parameters. Although completely general, this straightforward strategy is computationally very expensive. The strategy employed by HOLMES is somewhat less general but much faster computationally. This strategy holds one of the directions defining the plane, for example $\hat{k}$, fixed and seeks the maximum of $I(\hat{k},\hat{\ell})$ with respect to $\hat{\ell}$ in the (n-1)-dimensional subspace orthogonal to $\hat{k}$, $\tilde{E}^{n-1}(\hat{k})$. This reduces the number of search parameters to n-2. Additional computational economy is achieved in computing $I(\hat{k},\hat{\ell})$ by knowing that $\hat{k}$ is constant and that $\hat{k}\cdot\hat{\ell} = 0$.

When a solution direction, $\hat{\ell}^*$, is found, $I(\hat{k},\hat{\ell})$ is further maximized with respect to $\hat{k}$ in $\tilde{E}^{n-1}(\hat{\ell})$ while holding $\hat{\ell}$ fixed at the solution value $\hat{\ell} = \hat{\ell}^*$. This process of alternately fixing one direction and varying the other in the orthogonal subspace of the first is repeated until the solution becomes stable. The final directions $\hat{k}^*$ and $\hat{\ell}^*$ are then regarded as defining the solution plane.

CONSTRAINT DIRECTIONS

In order to encourage the algorithm to find as many distinct solutions as possible, it is useful to be able to reduce the dimensionality of the space to be searched. This can be done by choosing an arbitrary set of directions, $\{\hat{o}_i\}_{i=1}^{m}$, $m < n-2$, which need not be mutually orthogonal, and applying the constrants

$$\hat{k}^* \cdot \hat{o}_i = 0 \text{ and } \hat{\ell}^* \cdot \hat{o}_i = 0 \qquad (l= 1,m) \qquad (4)$$

on the solution directions $\hat{k}^*$ and $\hat{\ell}^*$. Possible choices for constaint directions might be solution directions found on previous searches, or directions that are known in advance to contain considerable, but well understood, structure. Also, when the choice of scales for the several coordinates is guided by considerations outside the data, one might wish to remove directions with small variance about the mean since these directions often provide little

information about the structure of the data. The introduction of each such constraint direction reduces by one the number of search variables, and thus increases the computational efficiency of the algorithm. HOLMES allows for the introduction of an arbitrary number, m < NDIM-2 of non-parallel constraint directions.

IMPLEMENTATION

Two-dimensional projection pursuit is invoked from a FORTRAN program by a subroutine call

CALL HOLMES(RADIUS,NDIM,NPTS,X,KEY,YB,XB,KEYSCL,SCALE,D).

All quantities are input to the routine except XB,YB and SCALE which serve a dual purpose as both input and output arrays. These quantities have the following meanings:

RADIUS is the cutoff radius, R, defined in eqn 2 and discussed above. Its value is set by the user, based on his knowledge of the input data. The user may also ask HOLMES to automatically determine a cutoff radius value based on its calculations on the input data. This is signalled by setting RADIUS to 'DFLT' in the call to HOLMES. For this case, the cutoff radius is calculated as

$$R = f_o * s(\hat{e}_1) \tag{5a}$$

where

$$f_o = RADSCL/\sqrt{NPTS} \qquad (NPTS \leq NPTSO) \tag{5b}$$

$$f_o = RADSCL*(\sqrt{NPTSO} / \log(NPTSO)) * (\log(NPTS)/(NPTS)) \quad (NPTS > NPTSO)$$

In HOLMES, RADSCAL is set to 2.5 and NPTSO is set to 1000. (See below on how to change these values.) The quantity $s(\hat{e}_1)$ is the standard deviation about the mean of the data as projected onto the principal axis[3] with the largest eigenvalue. (The trimming factor, p, (Eqn 1) is the zero for this calculation only; i.e., $s(\hat{e}_1)$ is the square root of the largest eigenvalue of the total

sample covariance matrix.) The values for f_o and R are claculated so that the average number of points within the local cutoff radius grows as the square root of the sample size for NPTS $\leq$ NPTSO, and grows as the logarithm for NPTS > NPTSO.

The quantities NDIM, NPTS and X refer to the input data.

NDIM = number of attributes per data point (dimensionality of data space)

NPTS = number of data points (sample size)

X = array, dimensioned X(NDIM,NPTS) in the calling program, that contains the coordinates of the data points.

The quantities KEY,XB and YB refer to the starting directions, $\hat{k}_s$ and $\hat{\ell}_s$, for the search.

KEY = axis flag =
- 'ORIG': use original axes for starting directions
- 'EIGN': use principal axes[3] as starting directions (ordered in decreasing value of corresponding eigenvalue)
- 'USER': use user supplied vectors as starting directions

XB = array, dimensioned XB(NDIM) in the calling programs that contain the starting direction $\hat{k}_s$.

YB = array, dimensioned YB(NDIM) in the calling program that contains the starting direction $\hat{\ell}_s$.

For KEY = 'USER', these arrays must contain the components of the two vectors that define the starting plane. These vectors need not be normalized or orthogonal. However, they must not be exactly parallel.

For KEY = 'EIGN' or 'ORIG', only the first element of these two arrays are input to HOLMES. (The arrays must still be dimensioned to NDIM.) The first element of each array contains the number of the principal (KEY='EIGN') or original (KEY='ORIG') axis that defines the starting direction. For

example,

XB(1) = 1.0

YB(1) = 2.0

defines axes one and two as the starting directions $\hat{k}_s$ and $\hat{\ell}_s$.

Upon return from HOLMES, the arrays XB and YB contain the solution directions $\hat{k}^*$ and $\hat{\ell}^*$ found in the projection pursuit.

The quantities KEYSCL and SCALE refer to the scaling of the input data along the original axes before the projection pursuit.

KEYSCL = scaling flag =
- 'NONE' do not scale data
- 'USER' use user provided scales
- 'ORIG' employ automatic scaling
- 'SMAX' automatic scaling with user provided lower limits for scales

SCALE = array, dimensioned SCALE(NDIM). For KEYSCL = 'NONE' or 'ORIG' this is a scratch array for HOLMES. For KEYSCL = 'USER', it contains the user provided scales and for KEYSCL = 'SMAX', it contains the user provided lower limits for the axis scales.

The automatically calculated scales for each original axis are the standard deviations of the data about their means, as projected onto each axis. Under the KEYSCL = 'SMAX' option, the axis scale is set to the maximum of the calculated scale and the user provided minimum scale for that axis. (The standard deviation calculation for the scales is trimmed by the global trimming factor IPER, described below. IPER is set to zero in HOLMES but may be changed by the user.) Upon return from HOLMES to the calling program, SCALE contains the scales that were used to divide the coordinate values of each data point along each of the axes.

D = Real*8 scratch array for HOLMES, dimensioned

Real*8 D(NDIM,NDIM,3).

Labeled Common Scratch Arrays

Four labeled common scratch arrays must be declared in the calling program as working storage for HOLMES. These are listed here along with their various dimensions:

```
COMMON // V(1750) /POINTS/ Z(2*NPTS+2) /DATA/P(NDIM*NPTS+1) /IORD/MORD(NEV/2)
```

Constraint Directions

As discussed above, HOLMES allows for the introduction of an arbitrary number ($<$ NDIM-2) of constraint directions. The solution directions, $\hat{k}^*$ and $\hat{\ell}^*$, will be constrained to be simultaneously perpendicular to each of these directions. The constraint directions need not themselves be mutually orthogonal, but no two may be exactly paralled (to machine accuracy). The constraint directions are entered into a labeled common whose label and dimensions are given below:

```
COMMON / NORMAL / NORM / NORMV / VEC(NDIM,NORM)
```

NORM = number of user supplied constraint directions

VEC = coordinates of constraint vectors

If not referenced by the user, the default value for NORM is zero.

Alternate CALLS

HOLMES may be called repeatedly from the calling program, each call initiating a projection pursuit. If subsequent calls do not change the input data or its scaling, an alternate shortened calling sequence may be used

```
CALL SHERLK(RADIUS,KEY,XB,YB) .
```

The arguments in the calling sequence have the same meanings as described above. This call may not be the first call to HOLMES. All quantities that are absent from this shortened calling sequence have the values assigned to them on the last call to HOLMES, using the full calling sequence.

Random Direction Generator

To facilitate starting the projection pursuit algorithm at random directions in the multidimensional space, HOLMES provides a random direction generator

```
CALL RDAXES(YB,XB,NDIM)
```

Upon return to the calling program, YB and XB [dimensioned to NDIM in the calling program] contain the coordinates of two random directions in the NDIM-dimensional space. Repeated calls to RDAXES generates new pairs of random directions for each call.

Optional User Control

There are several parameters internal to HOLMES that the user may change at his discretion. Some of these parameters control the verboseness of the output. Others are parameters of the projection pursuit algorithm to which it is reasonably insensitive, and thus, they need not be changed frequently. These parameters are stored in the labeled common blocks listed below, along with their default values (set in a BLOCK DATA subprogram internal to HOLMES).

```
COMMON / DONE / EPSLN,NITER
                (0.02)  (6)
```

These parameters control the number of iterations in the pursuit strategy discussed above. The iteration procedure stops when

$$(I(\hat{k},\hat{\ell}) - I_{-1}(\hat{k},\hat{\ell})) \,/\, I(\hat{k},\hat{\ell}) \leq \text{EPSLN}$$

or when NITER iterations have been completed. Here $I(\hat{k},\hat{\ell})$ is the value of the projection index at the current iteration and $I_{-1}(\hat{k},\hat{\ell})$ is the value at the previous iteration.

```
COMMON/COMST/ IPER,PEROUT
              (0)  (.01)
```

IPER is a global trimming parameter for the evaluation of both $d(\hat{k},\hat{\ell})$ and $s(\hat{k},\hat{\ell})$ (Eqn's 1 and 2). That is, IPER projected points are deleted from

each extreme of the projection before both $d(\hat{k},\hat{\ell})$ and $s(\hat{k},\hat{\ell})$ are evaluated for the projection. PEROUT is the parameter p in Eqn 1 and is a fractional trimming factor for the evaluation of $s(\hat{k},\hat{\ell})$ only.

```
COMMON /EIT/ EITMIN,RADSCL,NEVO
             (.01)  (2.5) (1000)
```

If the variance of the data, as projected onto a principal axis, is sufficiently small (as compared to the largest eigenvalue of the covarience matrix), then HOLMES will automatically consider that axis to be a constraint direction. EITMIN is the minimum value of the square root of the ratio of the axis variance to the largest eigenvalue, for the principal axis not to be considered a constraint direction. The parameters RADSCL and NEVO are discussed above under the default option for the local cutoff radius.

```
COMMON/ CNTRL/ NBX,NBY,CONV1,CONV2,NSTPS,NPRNT
               (100)(50)(0.05)(-.02) (2)   (1)
```

NBX is the number of horizontal characters (cells) in pictorial scatterplot ($\leq$ 100)

NBY is the number of vertical characters (cells) in pictorial scatterplot ($\leq$ 50)

CONV1 and CONV2 control the convergence criteria of the maximization algorithm used to maximize $I(\hat{k},\hat{\ell})$ within each HOLMES iteration. The search converges when the change in every parameter is less than .1*CONV1, or when $|[I(\hat{k},\hat{\ell}) - I(\hat{k}',\hat{\ell}')]/I(\hat{k},\hat{\ell})| \leq$ -CONV2, where $I(\hat{k},\hat{\ell})$ is the projection index value at the current maximizer iteration and $I(\hat{k}',\hat{\ell}')$ is the value at the previous iteration. NSTPS pertains to the maximum number of iterations allowed for the maximizer. This maximum number of iterations is NSTPS, plus the number of search variables. NPRNT controls the verbosity of the printed output from the maximizer.

NPRNT
- ≤ 0: no maximizer printed output
- > 0: printed output for every iteration number that is an integral multiple of NPRNT

PRINTED OUTPUT

As indicated above, the nature of the printed output from HOLMES depends upon the options chosen by the user. Figure 2 illustrates the printed output from the FORTRAN program listed in Figure 1.

Listed first (SCALE =) are the scale factors that were used to scale (divide) the data along each of the NDIM original axes. The following line lists the value for f_o of Eqn 5b (RADIUS SCALE =), R of Eqn 2 and 5a (LOCAL RADIUS =), and EITMIN (see optional user control) times the first (largest) eigen square root value (CUTOFF EIGEN STD =). This latter value is the smallest allowable eigenvalue square root for the corresponding principal axis not to be considered to be a constraint direction. Shown next (EIGENSTDS =) are the square roots of the eigenvalues of the total sample (no trimming) covariance matrix listed in decreasing order.

Following this general information is the printed output from each HOLMES iteration in the projection pursuit. First, a pictorial scatterplot of the data, as projected onto the plane defined by the starting axes $\hat{k}_s, \hat{\ell}_s$ (XB, YB), is displayed. The coordinates of these starting axes are listed above the plot (Y-AXIS = and X-AXIS =). As discussed above, the pursuit strategy holds one of these axes, $\hat{\ell}$, fixed (Y-AXIS) and varies the other, $\hat{k}$, (X-AXIS), in the orthogonal subspace of the first. The printed output from the maximizer follows next (if not surpressed by the user). For each iteration in the search for the maximum of the projection index, $I(\hat{k},\hat{\ell})$, the iteration number, the cumulative number of evaluations of $I(\hat{k},\hat{\ell})$, the value of

$-I(\hat{k},\hat{\ell})$ (F =), and the values of the search parameters (X =), are listed. Iteration zero corresponds to the starting direction for this HOLMES iteration. The parameters of the search are those of the solid angle transform, $SAT(\hat{k})$, of the corresponding direction $\hat{k}$ in the NDIM-dimensional space (see Ref. 1, pg. 7 and Appendix), and have no direct interpretation.

After a solution, $\hat{k}^*$, is found a pictorial scatterplot of the data as projected onto the $\hat{k}^*,\hat{\ell}$ plane is displayed, headed by a listing of the coordinates of the axes $\hat{\ell}$ (Y-AXIS=) and $\hat{k}^*$ (X-AXIS=). Following the scatterplot, the HOLMES iteration number (HOLMES ITERATION NUMBER) and the current value of the P-index (P-INDEX=) are listed.

This output sequence is repeated for each HOLMES iteration until the iteration procedure converges or exceeds the maximum number of iterations. The starting directions for each successive iteration are simply the solution directions for the previous iteration reversed, that is, $\hat{k}_s = \hat{\ell}$ and $\hat{\ell}_s = \hat{k}^*$. Thus, the initial scatterplot for an iteration is just the solution scatterplot of the previous iteration with the axes reversed.

For each call to HOLMES, this complete output sequence is repeated. If, however, subsequent calls are made to SHERLK using the shortened calling sequence, then the purely data dependent information is not repeated.

REFERENCE

1. J. H. Friedman and J. W. Tukey, "A Projection Pursuit Algorithm for Exploratory Data Analysis", IEEE Trans. Computers, Vol. C-23, pp 881-890 (1974).

2. M. A. Fisherkeller, J. H. Friedman and J. W. Tukey, "PRIM-9, An Interactive Multidimensional Data Display and Analysis System", Proceedings of the Second Annual AEC Scientific Computer Information Exchange Meeting, pp 3-33, May 2-3, 1974; also Stanford Linear Accelerator Center Report, SLAC-PUB-1408, April 1974.

3. J. H. Friedman, "Data Analysis Techniques for High Energy Particle Physics", SLAC Report 176, pp 54-56, September 1974.

```
C   TWO-DIMENSIONAL PROJECTION PURSUIT EXAMPLE. 700 POINTS, 6 DIMENSIONS.
C
      COMMON /POINTS/MM(2),Z(2,700) /DATA/NM,P(6,700) / /V(1750)
      REAL X(6,700),SCALE(6),XB(6),YB(6)
      REAL*8 D(6,6,3)
      COMMON /IORD/MORD(350)
C                                                  READ DATA INTO X.
      CALL DATAPT(X,700,6)
C                                                SET UP STARTING DIRECTIONS.
      XB(1)=1.0
      YB(1)=2.0
C                                                INITIATE PROJECTION PURSUIT.
C
      CALL HOLMES('DFLT',6,700,X,'FIGN',YB,XB,'NONE',SCALE,D)
C
      STOP
      END
?
```

FIGURE 1