Jan. 8, 1997

User's Manual for the GTREE Program to Fit Additive Trees

James E. Corter
Teachers College, Columbia Uinversity

CONTENTS:

AUTHOR'S ADDRESS:
Box 118, Teachers College
Columbia University
New York, NY 10027 (email: jec34@tc.columbia.edu)

==========

I. Introduction

GTREE is a program written in the PASCAL language, that implements the "generalized triples" or "GT" algorithm described by Corter (1997; see also Corter, 1992). The algorithm looks through all triples of objects x,y,z, using the precomputed summed row distances of each object to all other objects to decide if y or z is x's neighbor in the tree structure. The algorithm is therefore nearly order-n faster than the original Sattath and Tversky or ST (1977) algorithm (algorithm ST) and Corter's (1982) variant of the ST procedure (algorithm STC), as implemented in the ADDTREE/P program (Corter, 1982), though for small to moderate data sets (n<40) there is little difference in speed (the algorithms are indistinguishable in terms of fit of the obtained solutions).

Differences between the ST and STC algorithms are described more fully in Corter (1996).

Because the GTREE code is adapted from the ADDTREE/P code, the input file format and useage of the program is identical to that ofthe ADDTREE/P program.

II. Description of Input File Format

Input files should contain the following lines:

****

(1) first line: a list of analysis option commands, zero or more of the following keywords. Note that lowercase letters may be used, and that all keywords may be abbreviated to 3 characters.

DATA Requests printing of the raw data matrix.

TRANSFORMED Requests printing of the data, after transformation into distance-like numbers.

MODELDISTANCES Requests printing of the model (tree) distances.

RESIDUALS Requests printing of the residuals matrix.

HEIGHTS Requests printing of the distance of each node from the root.

NOTREE Suppresses printing of the tree graph.

NONUMBER Suppresses numbering of objects in the tree graph.

SPECIFYORDER If this is included, line 1 of the input file must be followed by a line containing a list of object numbers. The leaves of the tree graph are then ordered to conform as closely as possible to this list. (NOTE: Care must be taken not to omit or duplicate numbers in this list. Also, note that not all orderings of the leaves are possible, since the tree structure imposes constraints on the possible orderings.)

NOSUBTRACTCONSTANT The default assumption is interval-scale data; this implies complete freedom in choosing an additive constant. Therefore the primary approach is to either add OR subtract an additive constant to exactly satisfy the triangle inequality:
d(i,j) + d(i,k) <= d(j,k) for all i,j,k, AND d(i,j) + d(i,k) =

d(j,k) for some i,j,k. But if "nosubtractconstant" is specified, strict inequality is allowed, i.e. if d(i,j) + d(i,k) > d(j,k) holds for all i,j,k in the data, no constant is subtracted.

MINVARROOT The default rooting is to combine the last few remaining clusters into the root node; if "minvarroot" is specified, the program will search for the root that minimizes the variance of the distances from the root to the leaves.

ROOTAT Specify where the root is to be placed. If this option is requested, this keyword must be followed by two node numbers, e.g. "rootat 12 13 ". Note that the two nodes specified must be contiguous in the tree structure. A previous run will generally be necessary to determine the correct node numbers.

LINESIZE Sets the length of lines in the output file. The current default value is 80 characters (so that output may be conveniently read on a video terminal). Example: "linesize 66 ".

Note that any of the keywords mentioned above may be abbreviated to three (or more) characters; also, capital or lower-case letters are ok. Keywords may be separated by either blanks or commas. However, when a keyword above specifies that a parameter is to be read in following the keyword (e.g. ("rootat 12 13 "), it is best to follow the numerical parameter with a <space> (" ") instead of a <comma> (",") since some PASCALs will object to finding a comma while attempting to read a number.

If the first line is blank, the standard analysis is done and the default output is obtained (estimates, tree graph, and statistics).

****

(1.5) (optional line: see SPECIFYORDER above)

****

(2) second line: a comment line for labeling of the output (up to 80 characters in length).

****

(3) third line: the following parameters (separated by commas or blanks):

<number of objects> (integer)

SIMILARITIES or DISSIMILARITIES (specifies similarity or distance (dissimilarities) data)

FULL or LOWERHALF (specifies shape of matrix)

<fieldsize> (integer; width of OUTPUT data fields)

<number of digits after the decimal point> (integer; number of digits after the decimal place in the OUTPUT matrices)

NOTE: the last two parameters here refer only to the OUTPUT. Also note that you should try to leave at least two spaces between numbers: e.g. if your data is numbers between 0 and 99, and you want to see one digit after the decimal place, you should specify 6 and 1 for these parameters. This is to leave room for possible minus signs and the decimal point itself.

****

(4) fourth, etc. lines: the data matrix itself

NOTE: some Pascals require real format, that is, the data must have

a decimal point, and digits both before and after (e.g. 45.0, not 45 or 45. or .45) NOTE: the data matrix can be in a variety of formats, as long as the order of entries corresponds to the order of entries in a (full or lowerhalf) matrix. So for example, the input data file might contain only one number to a line, as long as the order of the lines corresponds to the order of entries in a lowerhalf (or full) matrix. However, one thing that must NOT occur is trailing blanks on a line, since the program then expects another number on that line.

****

(5) (optional) lines immediately following the data: a list of object names, one to a line, maximum length 20 characters.

III. Example Input and Output Files

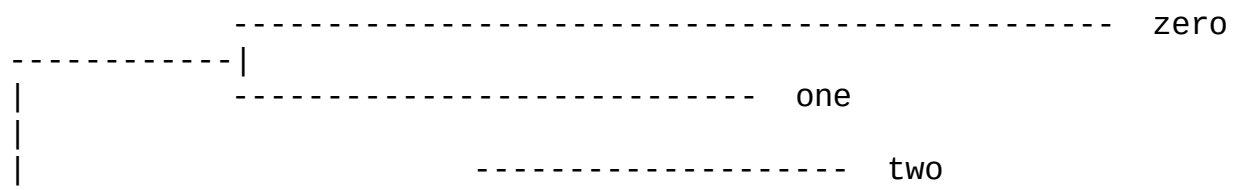
EXAMPLE INPUT FILE:

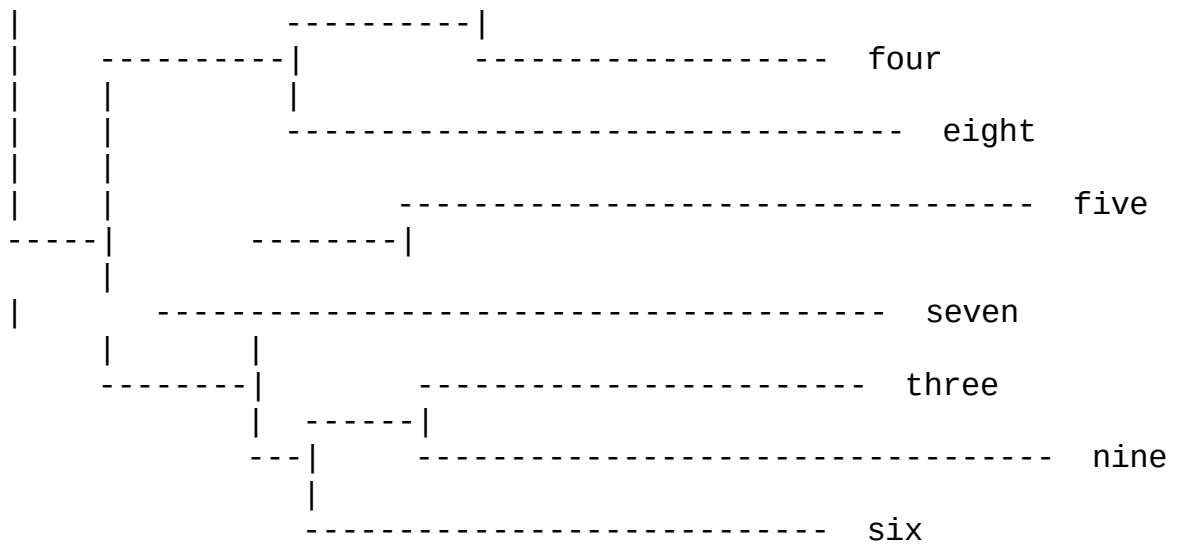
```
nonumber,residuals,minvarroot,linesize 70
numbers: abstract concept (Shepard, Kilpatric, & Cunningham; 1975)
10 dis low 7 1
421
584 284
709 346 354
684 646 059 413
804 588 671 429 409
788 758 421 300 388 396
909 630 796 592 742 400 417
821 791 367 804 246 671 350
 400
850 625 808 263 683 592 296
 459 392
zero
one
two
three
four
five
six
seven
eight
nine
```

EXAMPLE OUTPUT FILE:

```
additive tree analysis (GTREE version 1.0):
numbers: abstract concept (Shepard, Kilpatric, & Cunningham; 1975)



(    0.0  needed for positivity of distances )
   303.0  added to exactly satisfy triangle inequality



var. of dis. from root to leaves =116529.7842
node 14  score=61715.504
node 15  score=116529.784
```

```
node 16  score=177332.083
node  7  score=177332.083
changeroot( 17 => 14)
node 11  score=43795.738
node 13  score=61831.331
node 15  score=116529.784
node 18  score=116529.784
changeroot( 14 => 11)
node  1  score=147886.744
node  2  score=147886.744
node 13  score=61831.331
node 17  score=61715.504

var. of dis. from root to leaves =43795.7375



node     length    children
   1      454.6                     zero
   2      269.4                     one
   3      185.6                     two
   4      235.6                     three
   5      176.4                     four
   6      327.4                     five
   7      264.1                     six
   8      375.6                     seven
   9      325.3                     eight
  10      330.4                     nine
  11      127.5       1  2
  12      103.2       3  5
  13       97.5      12  9
  14       53.1      13 17
  15       81.8       6  8
  16       53.9       4 10
  17       77.4      15 18
  18       37.9      16  7
  19        0.0      11 14
```





stress formula 1 = 0.0926
stress formula 2 = 0.4958
r(monotonic) squared=0.7542
r-squared (p.v.a.f.)=0.6249

residual distances:

0.0								
-134.5	-249.4							
-28.1	-206.0	-134.2						
-25.2	121.9	0.0	-66.0					
-14.8	-45.7	101.0	-4.7	-151.7				
76.4	231.5	-41.7	49.4	-65.5	-12.2			
42.0	-51.8	177.9	110.2	133.1	0.0	-39.3		
66.1	221.2	55.9	279.3	-55.9	64.6	-149.2	-254.6	
18.2	-21.7	225.0	0.0	109.3	63.6	-49.4	-117.6	-227.4

IV. Installing GTREE on your System

The provided version of GTREE has input/output statements written for use with Turbo PASCAL (Borland). Previous versions have been successfully compiled on UNIX systems. Because the PASCAL language as originally defined does not contain specifications for certain I/O and file definition operations, statements pertaining to these operations may vary across systems. Please notify James Corter of problems or comments regarding portability.

For users without access to a PASCAL compiler, a DOS-executable version of GTREE may be retrieved via FTP from a site at Teachers College, Columbia University. To retrieve the DOS-executable files, FTP to ftp.ilt.columbia.edu, log in as "anonymous", then connect ("cd") to the directory "users/corter". A list of available files may be viewed by typing "dir", and individual files may be retrieved by typing "get <filename>". To retrieve executable files, the FTP file type must be set to binary before using "get".

Current source code and executable versions are limited to a maximum of 80 objects. With certain compilers, it is possible to increase this ceiling, simply by increasing the values of the constants "maxobject","twiceobject", and "maxsize" specified in the program declarations section of the source code.

V. References

Corter, J.E. (1982). ADDTREE/P: A PASCAL program for fitting additive trees based on Sattath & Tversky's ADDTREE program. Behavior Research Methods and Instrumentation, 14, 353-354.

Corter, J.E. (1992). An order-N**3 combinatoric algorithm for fitting additive trees. Presented at annual meeting of the Classification Society of North America, East Lansing MI, June.

Corter, J.E. (1996). Tree Models of Similarity and Association. (Sage University Papers series: Quantitative Applications in the Social Sciences, series no. 07-112). Thousand Oaks CA: Sage.

Corter, J. E. (1998). An efficient metric combinatorial algorithm for fitting additive trees. Multivariate Behavioral Research, 33, 249-272.

Sattath, S., & Tversky, A. (1977). Additive similarity trees. Psychometrika, 42, 319-345.