of **B** are *eigenvalues* of **C** and the columns of **A** are the *eigenvectors* of **C**. In general, if we fitted a model $\mathbf{ABA}'$ where **A** was full and **B** was diagonal, it would be underidentified, since there would be more parameters in the model than in the observed covariance matrix **C**. However, we can supply identifying constraints that **A** is *orthogonal*. In the Mx input file, these constraints are imposed in group 2, by setting $\mathbf{AA}' - \mathbf{I} = \mathbf{0}$. This is not an efficient method of obtaining eigenvalues and eigenvectors of a matrix, but it does highlight nonlinearly constrained optimization. For eigenvalue and eigenvector functions, see Section 3.2.2 and Table 3.5.

```
Principal components ABA' with constraints
!                    to keep A orthogonal
Data NGroups=2 NInput_vars=3 NObservations=100
Cmatrix Symm
   1.
  .6  .9
  .4  .2  .7
Matrices
A Full 3 3
B Diag 3 3
Covariances A*B*A'/
Specification A
1 2 3
4 5 6
7 8 9
Specification B
10 11 12
Start 1.0 A(1,1) A(2,2) A(3,3) B(1,1) to B(3,3)
Options LS
End

Here is the constraint A*A'=I
Data CON NInput_vars=3
Matrices
A Full 3 3 = A1
I Identity 3 3
Model A*A'-I/
Options LS
End
```

5.3.2 Fitting a PACE Model to Contingency Table Data on MZ and DZ Twins

In order to fit a model with additive genetic, common and specific environment components to categorical data collected from twins, we are faced with two possibilities. One, we could use PRELIS or similar software to estimate tetrachoric or polychoric correlation matrices and associate asymptotic weight matrices, or two, we could fit directly to the contingency tables. The advantage of the latter approach is when there is phenotypic interaction between the twins. Phenotypic interaction would lead to different variances between groups, or in the case of categorical data, to proportionate group differences in the thresholds. This example uses a simple PACE model (see the example shown in Section 5.2.5) fitted to 2×2 contingency tables obtained from MZ and DZ twin pairs. There is no information on the total variance in these data; but there is information on the relative magnitude of the variance in MZ and DZ groups. Therefore, it is necessary to constrain the total variance prior to interaction to unity. This is done in the third group.

```
Categorical data analysis.  PACE model
Data Ninput=2 NGroup=3
CTable 2 2
30  20
19  60
Matrices
A fu 2 6
B fu 2 2
I id 2 2
P Symm 6 6
T Full 2 1
Thresholds T /
Covariances (I-B)~* A*P*A' *(I-B)~ /
Specification A
1 2 3 0 0 0
0 0 0 1 2 3
Start .6 A 1 1 - A 2 6
Specification T
4 5
Specification B
0 6
6 0
Boundary -.99 .99 6
Matrix P
1
0 1
0 0 1
1 0 0 1
0 1 0 0 1
0 0 0 0 0 1
Options RSiduals
End
```

```
Test of categorical data analysis.  DZ twins in ACE model
Data Ninput=2
CTable 2 2
30  30
29  50
Matrices
A fu 2 6 =A1
B fu 2 2 =B1
I id 2 2
P Symm 6 6
T fu 2 1 =T1
Thresholds T /
Covariances (I-B)~* A*P*A' *(I-B)~ /
Matrix P
1
0 1
0 0 1
.5 0 0 1
0 1 0 0 1
0 0 0 0 0 1
Options RSiduals
End

Constraint group to ensure a*a + c*c + e*e = 1
Data Constraint Ninput=1
Matrices
S Full 1 3
I Identity 1 1
Constraint I - S*S'   /
Specification S
1 2 3
Options None Iterations=300
End
```

5.3.3 Twins and their Parents: Cultural and Genetic Transmission

The model described has been developed extensively in the univariate and multivariate case (Eaves et at, 1978; Fulker, 1982; Neale & Fulker, 1984; Vogler, 1985; Fulker, 1988; Phillips et al, 1988; Cardon, Fulker & Jöreskog, 1991). In order to preserve consistency with the ACE model presented for twin data alone, the same separation of environmental effects is made here, following the last of the referenced papers instead of the earlier treatments. A path diagram of the model is shown in Figure 5.5.

The development of a covariance structure model for this design is not simple. We describe two approaches: (i) direct use of RAM theory in which all variables in the model are represented in two matrices, and the Pearson-Aitken selection formula (see Section 5.1.3) is used to handle assortative mating, and (ii) an computationally efficient approach that progresses in a stepwise fashion from the top to the bottom of the diagram. Approach (ii) is recommended for general use except where fast hardware gives approach (i) comfortably quick turnaround.

RAM Theory Approach

As discussed in Section 4.4.5, structural equation models may be specified very simply in terms of three matrices. The first matrix, **S**, is symmetric and specifies all the two-headed arrows between all the variables (both latent and observed) in the diagram. The second matrix, **A**, is asymmetric and specifies all the single-headed arrows between all the variables in the model. Causal paths from variable i to variable j are specified in element $\mathbf{A}_{ji}$. The third matrix, **F**, is used to filter the observed variables from the latent variables for the purpose of model fitting. This example is a relatively inefficient approach to fitting this model, but it illustrates the flexibility of Mx to implement theory-driven models explicitly.

```
!
! Rose social fears data
!
!  Full 9-Phenotype model for all pedigree types
!
! P->C transmission & P--P assortment
!
G1 - covariance in the absence of assortative mating
Data Calculation NGroups=13
Matrices
A Full 19 19
I Iden 19 19
S Symm 19 19
Compute (I-A)~*S*(I-A)~' /

Labels Row A
!1  2    3    4   5  6  7   8  9 10  11 12 13
! 14 15   16   17  18  19
PH PW PMZ1 PMZ2 PDZ AH AH' CH EH AW AW' CW EW
AMZ CE EMZ1 EMZ2 ADZ EDZ
Labels Col A
PH PW PMZ1 PMZ2 PDZ AH AH' CH EH AW AW' CW EW
```

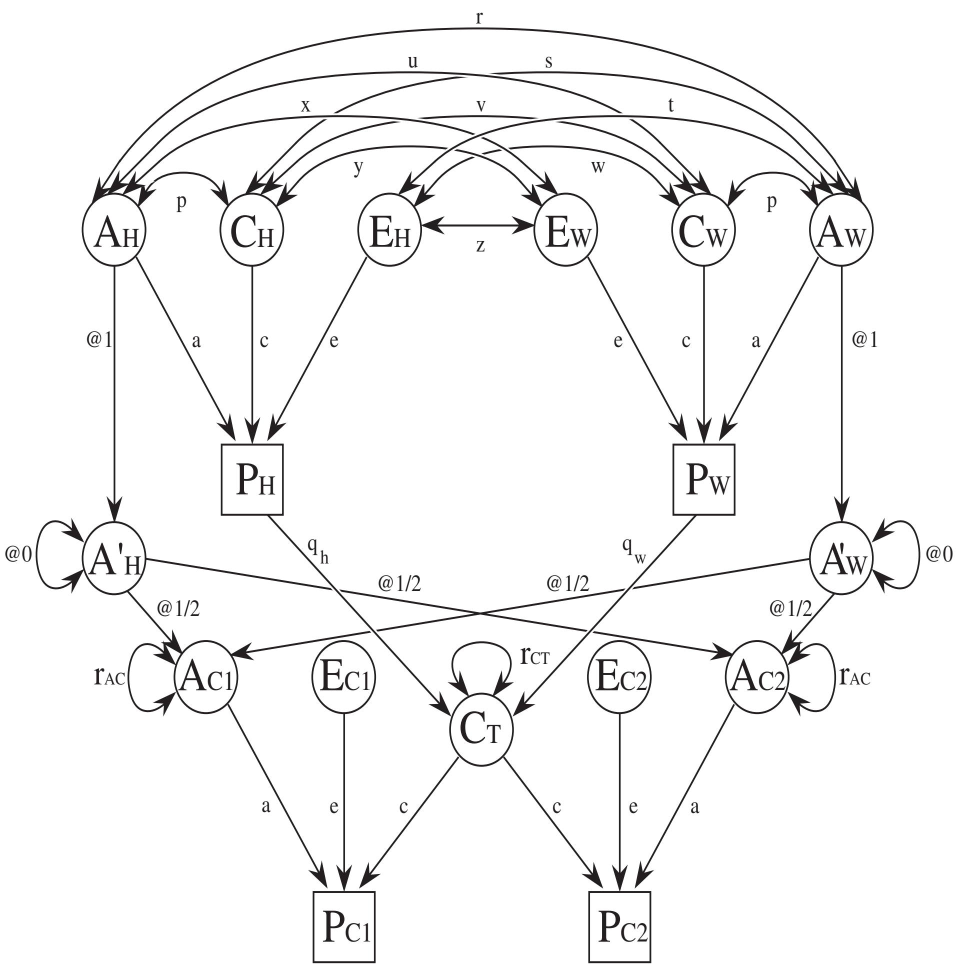

Figure 5.5: Model of mixed genetic and cultural transmission. Phenotypes of a husband and wife (P_H and P_W) directly affect the shared environment of their children (CT). Assortative mating based on phenotypes generates covariance between the latent variables of the parents (parameters r to z). The additive genetic and shared and specific environmental effects (parameters a, c and e) and the covariance of C and A (parameter p) are assumed to be equal across generations. Reduced genetic correlation between parents and offspring may be modelled if paths from A_H to A'_H and from A_W to A'_W are free parameters, with corresponding alteration of the residuals on A'_H and A'_W to maintain unit variance
.

```
AMZ CE EMZ1 EMZ2 ADZ EDZ

Specification A
0 0 0 0 0  1 0 2 3 0  0 0 0 0 0  0 0 0 0
0 0 0 0 0  0 0 0 0 1  0 2 3 0 0  0 0 0 0
0 0 0 0 0  0 0 0 0 0  0 0 0 1 2  3 0 0 0
0 0 0 0 0  0 0 0 0 0  0 0 0 1 2  0 3 0 0
0 0 0 0 0  0 0 0 0 0  0 0 0 0 2  0 0 1 3

0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0
0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0
0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0
0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0
0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0

0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0
0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0
0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0
0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0
4 4 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0

0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0
0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0
0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0
0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0

Value 1.0  A 7 6  A 11 10
Value 0.5  A 14 7   A 14 11  A 18 7   A 18 11

Labels Row S
!1  2    3    4   5  6  7   8  9 10  11 12 13
! 14 15   16   17  18  19
PH PW PMZ1 PMZ2 PDZ AH AH' CH EH AW AW' CW EW
AMZ CE EMZ1 EMZ2 ADZ EDZ

Specify S
0
0 0
0 0 0
0 0 0 0
0 0 0 0 0

0 0 0 0 0  0
```

```
0 0 0 0 0  0 0
0 0 0 0 0  8 0 0
0 0 0 0 0  0 0 0 0
0 0 0 0 0  0 0 0 0 0

0 0 0 0 0  0 0 0 0 0  0
0 0 0 0 0  0 0 0 0 8  0 0
0 0 0 0 0  0 0 0 0 0  0 0 0
0 0 0 0 0  0 0 0 0 0  0 0 0 5
0 0 0 0 0  0 0 0 0 0  0 0 0 0 6

0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0
0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0
0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 5
0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0

Labels Col S
!1  2    3    4   5  6  7   8  9 10  11 12 13
! 14 15   16   17  18  19
PH PW PMZ1 PMZ2 PDZ AH AH' CH EH AW AW' CW EW
AMZ CE EMZ1 EMZ2 ADZ EDZ
Start 1.0 S 6 6 S 8 8 S 9 9 S 10 10 S 12 12 S 13 13 S 16 16
Start 1.0 S 17 17 S 19 19
Options RSiduals
End

Group 2  Get covariance of parent phenotypes with other variables
Data Calculation
Matrices
X IZero  2 19
R Symme 19 19 = %E1
Y ZIden 19 17
Compute X*R*Y /
Options RSiduals
End

Group 3 - Get covar. of variables that are not parents' phenotypes
Data Calculation
Matrices
R Symm  19 19 = %E1
Z ZIden 17 19
Compute Z*R*Z' /
Options RSiduals
```

```
End

Group 4 - Obtaining Covariance Matrix of Parents
Data Calc
Matrices
J IZero  2 19
B Full  19 19 = %E1
Compute J*B*J' /
Options RSiduals
End

Group 5 - Handle the effects of assortative mating
Data Calculation
Matrices
A FU  2  2 =%E4
B FU  2 17 =%E2
C FU 17 17 =%E3
I ID  2  2
M SY  2  2
Compute       (A+M)|(A+M)*A~*B
_     ((A+M)*A~*B)'|C-B'*(A~*(I-(A+M)*A~))*B /
! Compute    (A+M)|(A+M)*B
! _     ((A+M)*B)'|C-B'*-M*B /     ! REDUCED FORM
Specify M
0
7 0
Start .1 M 2 1
Options RSiduals
End

Group 6 - Constraints on kids' G-E variance and covariance
Data Constraint NInput_vars=2
Matrices
E Full 19 19 =%E5
C Stan 2 2
F Full 2 19
Constraint F*E*F'-C /
Specify C
8
Matrix F
0 0 0 0 0  0 0 0 0 0  0 0 0 1 0  0 0 0 0
0 0 0 0 0  0 0 0 0 0  0 0 0 0 1  0 0 0 0
Labels Col F
```

```
Labels Row A
!1  2    3    4   5  6  7   8  9 10  11 12 13
! 14 15   16   17  18  19
PH PW PMZ1 PMZ2 PDZ AH AH' CH EH AW AW' CW EW
AMZ CE EMZ1 EMZ2 ADZ EDZ
Options RSiduals
End

Group 7 - VA phobia fear data: MZ twins & their parents
!Data NI=8 NO=340
!PMatrix File=mzphob.cor
!ACov File=mzphob.asy
!Labels
! T1SOC T1ACRO T2SOC T2ACRO
! P3SOC P3ACRO P4SOC P4ACRO
!Select P3SOC P4SOC T1SOC T2SOC /
Data NInput_vars=4 NObservations=144
CMatrix
.903
.113 1.1053
.2547 .0864 .9467
.1602 .2052 .4938 .8947
Matrices
C FU 19 19 = %E5
F IZ 4 19
Covariances F*C*F'/
Options RSiduals
End
! DZ data

Group 8 - Rose social criticism DZ  twins & their parents
!Data NI=8 NO=227
!PMatrix File=dzphob.cor
!ACov File=dzphob.asy
!Labels
! T1SOC T1ACRO T2SOC T2ACRO
! P3SOC P3ACRO P4SOC P4ACRO
!Select P3SOC P4SOC T1SOC T2SOC /
Data NInput_vars=4 NObservations=106
CMatrix
1.0811
.2111 .9063
.1443 .2398 1.2079
```

```
.1001 .1551 .3246 1.0318
Matrices
C FU 19 19 = %E5
F FU 4 19
Covariances F*C*F'/
Matrix F
1 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0
0 1 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0
0 0 1 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0
0 0 0 0 1  0 0 0 0 0  0 0 0 0 0  0 0 0 0
Options RSiduals IT=500
End

Group  9 - Standardization
Data Calc
Matrices
A Full 1 3
S Stand 3 3
Covariances A.A@(A*S*A')~ /
Specification A
1 2 3
Specification S
8
0 0
Options RSiduals
End

Group 10 Expected correlation matrix
Data Calc
Matrices
A Full 19 19 = %E5
!Compute \stnd(A) /
Compute A /
Options RSiduals NO_output
End

Group 11 Getting expected matrices for phenotypes
Data Calc
Matrices
J IZero 5 19
A Full 19 19 = %E10
Compute J*a*j' /
Options RSiduals
```

```
End

Group 12 Getting Covariances among Parents' latent variables
Data Calc
Matrices
F fu 6 19
A fu 19 19 = %E5
Compute F*A*F' /
Start 1 f 1 6  f 1 8  f 1 9  f 2 10 f 2 12 f 2 13
Options RSiduals
End

Group 13  - Parameter estimates
Data Calculation
Matrices
P Full 1 8
Compute P/
Labels COL P
A C E Q ResGs ResCt Mu  S
Specification P
1 2 3 4  5     6    7   8
Start .7 P 1 1 P 1 3
Start .5 P 1 5
Start 1 P 1 6
Start .1 P 1 4 P 1 8
Bound .001 1 P 1 1 P 1 2 P 1 3 P 1 5 P 1 6
Bound -1 1 P 1 4 P 1 7 P 1 8
Options TH=-1 Multiple
End

Save ramrose3.mxs
! No assortment
Drop 7
Options Multiple
End
! No genes
get ramrose3.mxs
Drop 1 8
Options Multiple
End
! No VCT
get ramrose3.mxs
Drop 4 8
```

```
Options Multiple
End
! No CE at all
Get ramrose3.mxs
Drop 2 4 8
Options Multiple
End
! No resemblance
Drop 1 7
End
```

Computationally Efficient Approach

It is possible to get much faster turnaround with computationally efficient approaches to structural modelling of twins and their parents. The constraints are calculated in group 1, to generate the (3×3) block of correlations between husband latent variables and wife latent variables.

$$\begin{array}{cccc} & AH & CH & EH \\ AW & r & s & t \\ CW & u & v & w \\ EW & x & y & z \end{array}$$

In the absence of sex differences in genetic or environmental effects, this matrix is symmetric.

The second group constructs the 6×6 (symmetric) correlation matrix between the parents' latent variables by concatenating the within-person parts (matrix `D` of group 1) with the off-diagonal across-parents part calculated in group 1 (matrix `%E1`).

$$\begin{array}{ccccccc} & AH & CH & EH & AW & CW & EW \\ AH & 1 & & & & & \\ CH & p & 1 & & & & \\ EH & 0 & 0 & 1 & & & \\ AW & r & s & t & 1 & & \\ CW & u & v & w & p & 1 & \\ EW & x & y & z & 0 & 0 & 1 \end{array}$$

This matrix is used in group 3 (matrix `P`) to calculate the covariance between the parents' phenotypes and the latent variables of the offspring. Group 4 introduces the cultural transmission parameters. Of note here is the matrix `S`, which contains residual parameters to equate the genetic (and environmental) variance in children to those of the parents. Also present is additional covariance between the environments of the children (parameter 9, element `S 5 5`), due to sources other than parental phenotypes. The fifth group specifies the nonlinear constraints, keeping the variances of AC1, AC2, EC1 and EC2 at unity, and equating the covariance of AC2 and EC2 to that in the parental generation. The sixth group keeps the variance of parents at unity. Standardization of the model in this way helps to keep optimization well conditioned. As shown, group 7 is useless and could be omitted. However, it is instructive to include it to show how the model could be altered to specify a model with reduced genetic correlation between parents and children. Variance parameters are supplied in the data groups (8 and 9) where the observed data are supplied and the fit function is calculated. Note the key difference between MZ and DZ twins is that in the former, genetic variance comes from the same genotype (elements `L 8 3 3` and `L 8 4 3`), whereas in the latter genetic variance is caused by two different (though correlated) genotypes via elements `L 9 3 3` and `L 9 4 4`). The tenth group simply prints out parameter estimates.

We recognize that this specification in matrices is not as simple as might be obtained if a specification in terms of expected correlations was implemented in a FORTRAN program. However, modifications to the Mx code are relatively simple to make. For example, to change the model of assortative mating the first group could be altered. In the example code the necessary components to allow for reduced parent-child genetic covariance are present but commented out. Additionally, facilities for dropping parameters to fit reduced models or for adding different data summaries make this example a convenient starting point for comprehensive analysis of data from all types of nuclear family and twin and parent data.

```
!
! Rose Fear data:  Social phobia on twins and parents
!
! Twins and parents: Genetic and cultural transmission model.
!                    P--P assortment
!                    Reduced P-C genetic correlation is
!                              commented out group 7
!
Group 1 - Assortative mating constraints
Data Calculation NGroups=10
Matrices
A Full 1 3
D Standardized 3 3
M Full 1 1
Compute (A*D)'*M*(A*D) /
Labels Column A
A C E
Labels Row A
P
Specification A
1 2 3
Matrix A
```

```
.7 .5 .5
Labels Row D
A C E
Labels Col D
A C E
Specification D
4
0 0
Specification M
5
End

Group 2 - construct the overall covariance matrix of parents'
!         latent variables
Data Calculate
Matrices
D Stan 3 3 = D1
E Full 3 3 = %E1
Compute
  (D|E
  _E|D) /
Labels row E
Ah Ch Eh
Labels col E
Aw Cw Ew
End

Group 3 - precalculate parents model
Data Calculate
Matrices
P Symmetric 6 6 = %E2
G Full 4 6
R Diagonal 4 4
Compute G*P*G' + R/
! For reduced genetic correlation between parent & child model
! parts commented out with ! are needed
Specification G
1 2 3 0 0 0
0 0 0 1 2 3
0 0 0 0 0 0 !99 0 0 0 0 0
0 0 0 0 0 0 !0 0 0 99 0 0
Labels row G
PH PW AH' AW'
```

```
Labels column G
AH CH EH AW CW EW
Value 1 G 3 1 G 4 4
! Specification r
! 0 0 100 100
End

Group 4 - precalculate middle of model
Data Calculate
Matrices
P Symmetric 4 4 = %E3
F Full 5 4
S Symmetric 5 5
Compute F*P*F'+S /
Labels row F
PH PW AC1 AC2 CT
Labels col F
PH PW AH' AW'
Specification F
 0 0 0 0
 0 0 0 0
 0 0 0 0
 0 0 0 0
 6 7 0 0
Value 1 F 1 1 F 2 2
Value .5 F 3 3  F 3 4  F 4 3  F 4 4
Labels row S
PH PW AC1 AC2 CT
Labels col S
PH PW AC1 AC2 CT
Specification S
0
0 0
0 0 8
0 0 0 8
0 0 0 0 9
Start .5 S 3 3   S 4 4
Start 1 s 5 5
End

G5 - Constraints on genetic environmental variance and
!    covariance in children
Data Constraint NInput_vars=2
```

```
Matrices
D Standardized 3 3 = D1
!Correlation matrix of AH CH EH
H Full 5 5 = %E4
!Correlation matrix of PH PW AC1 AC2 CE
Y IZero 2 3
!Filter matrix to get corr matrix AH CH
Z ZIden 2 5
!Filter matrix to get corr matrix AC2 CE
Constraint Y*D*Y' - Z*H*Z' /
Options NO_output
End

G6 - Fourth constraint for standardization
Data Constraint NInput_vars=1
Matrices
H Full 5 5 = %E4
Z IZ 1 5
I Identity 1 1
Constraint I - Z*H*Z' /
Options NO_output
End

Group 7 - new constraint for genetic transmission parameter
! The commented out parts of this group are needed if reduced
! genetic correlation between parent and child model is to be
! fitted, and the covariance statement should be commented out
Data Calculation ni=1
! Change the above to Data Constraint ni=1 for Rg model
Matrices
A Symm 1 1
B Symm 1 1
I Iden 1 1
Constraint a /
! Delete the above line for Rg model and uncomment
! the next five lines:
!constraint (a*a+b)-i /
!Specification a
!99
!Specification b
!100
Options None
End
```

```
Group 8 - Rose fear data: MZ twins & their parents
Data NInput_vars=4 NObservations=144
Labels
Husband Wife MZ_Twin1 MZ_Twin2
CMatrix
.903
.113 1.1053
.2547 .0864 .9467
.1602 .2052 .4938 .8947
Matrices
C Full 5 5 = %E4
L Full 4 5
V Diagonal 4 4  !Matrix of standard deviations
E Diagonal 4 4  !Matrix to add random environment for children
Covariances V*(L*C*L'+E*E)*V'/
Specification V
10 10 10 10
Matrix V
1 1 1 1
Labels row V
PH PW PT1 PT2
Labels column V
PH PW PT1 PT2
Labels row E
Dummy Dummy Et1 Et2
Labels column E
Dummy Dummy Et1 Et2
Specification E
0 0 3 3
Boundary 1 5 v 1 1 v 2 2 v 3 3 V 4 4
Labels row L
Ph Pw Pmzt1 Pmzt2
Labels column L
Ph Pw At1 At2 Ct
Specification L
0 0 0 0 0
0 0 0 0 0
0 0 1 0 2
0 0 1 0 2
Value 1 L 1 1 L 2 2
Options RSiduals
End
```

```
Group 9 - Rose social criticism DZ  twins & their parents
Data NInput_vars=4 NObservations=106
Labels
Husband Wife DZ_Twin1 DZ_Twin2
CMatrix
1.0811
.2111 .9063
.1443 .2398 1.2079
.1001 .1551 .3246 1.0318
Matrices
C Full 5 5 = %E4
L Full 4 5
V Diag 4 4 = V8
E Diag 4 4 = E8
Covariances V*(L*C*L'+E*E)*V'/
Specification L
0 0 0 0 0
0 0 0 0 0
0 0 1 0 2
0 0 0 1 2
Value 1 L 1 1 L 2 2
Labels row E
Dummy Dummy Et1 Et2
Labels column E
Dummy Dummy Et1 Et2
Labels row L
Ph Pw Pdzt1 Pdzt2
Labels column L
Ph Pw At1 At2 Ct
Start 1.5 v 1 1 v 2 2 v 3 3 V 4 4
Boundary .5 2 v 1 1 v 2 2 v 3 3 V 4 4
Options RSiduals IT=500
End

Group 10 - summary printout of parameter estimates
Data Calculation
Matrices
P Full 1 10
Compute P/
Specification P
1 2 3 4 5 6 7 8 9 10
Labels c p
```

```
a c e s mu zf zm res(a) res(c) vp
Boundary -.99 .99 4 5 6 7
Boundary 0 1 1 2 3 8 9
Options Multiple
End
!
! Re-fit the model with Father-child and Mother-child
!  cultural transmission set equal
!
Equate f 4 5 1 f 4 5 2
End
```

5.4 Fitting Models to Raw Data

When models are fitted to raw data, it is normal to provide a model for the means as well as for the covariances. Otherwise, there is little difference in the approaches. Mx computes minus twice the log-likelihood of the data, with an arbitrary constant that is a function of the data. Thus there is no overall measure of fit, but there are relative measures of fit, since differences in fit function between submodels are distributed as χ^2.

5.4.1 Estimating Means and Covariances

This section demonstrates maximum likelihood estimation using complete, balanced raw data. A cholesky decomposition (see Figure 5.2) is used for the covariance structure, and the means are estimated separately. Alternative models for covariances or means or both could be used if desired.

```
!
! ML fitting to raw data simulated
! with SAS, whose PROC COR COV gave:
!
! VARIABLE         N         MEAN        STANDARD
!                                        DEVIATION
! P1            1000     0.00182388     0.98499439
! P2            1000    -0.98608262     1.40083917
! P3            1000     2.05400385     1.79139557
!
! COVARIANCE MATRIX
!               P1       P2        P3
! P1       0.970214 0.506058 0.620529
! P2       0.506058  1.96235 0.807754
! P3       0.620529 0.807754   3.2091
```

```
!
! Cholesky for covariance structure
!
ML example, calculation of likelihood for each observation.
Data NInput_vars=3 NObservations=1000 NGroups=1
RA File=[NEALE.MX.EXAMPLES]MLPED.RAW
Matrices
M Full  1 3 Free
S Lower 3 3 Free
Means M /
Covariances S*S' /
Matrix_start_values S
1 0 0
.6 .8 0
.6 .0 .8
Options RM
End
```

The relevant part of the output from this job shows good agreement with the SAS results, calculating means and covariances in the usual fashion, as would be expected for a sample size of 1000. The estimates of the variances are slight underestimates since the ML estimate of a variance is biased, having denominator n instead of $n-1$. Multiplying the ML estimates by 1000/999 we recover the calculated covariances precisely.

```
  ML EXAMPLE, CALCULATION OF LIKELIHOOD...

  MATRIX M
 This is a FULL matrix of order    1 by    3
    0.0018   -0.9861    2.0536

  MATRIX S
 This is a constrained FULL matrix of order    3 by    3
    0.9692    0.5056    0.6199
    0.5056    1.9604    0.8069
    0.6199    0.8069    3.2059
```

5.4.2 Variable Pedigree Sizes

When there are many different possible configurations of data, it is most convenient to use a variable length data file (see Section 2.3.3). This information can be read by Mx and the likelihood of the data may be calculated for any structural model for the covariances and the means. In this example, a Cholesky decomposition (see Figure5.2) of the expected covariance matrix is specified in Group 1.

```
!
! ML fitting to raw variable length data.
!
! Cholesky decomposition for the expected covariance matrix
!
! Also matrix expression for means,
!  - in this case just a simple vector with free parameters.
!
Variable pedigree size ML example.
Data NInput_vars=3 NObservations=1000 NGroups=1
VL File=[neale.mx.examples]UNBALANCED.RAW
Matrices
M Full 1 3 Free
S Lower 3 3 Free
Means M /
Covariances S*S' /
Start .7 s 1 1 - s 3 3
Options RM
End
```

Typical lines of the data file 'unbalanced.raw' look like this:

```
2
1 2   0.5550   -1.1114
3
1 2 3   1.6442   -0.1319  3.609508
3
1 2 3  -0.2145   -1.2193  5.011667
3
1 2 3   2.2274   -1.9423  0.714351
```

There can be problems when beginning to fit models to VL data. One commonly encountered difficulty is that at the starting values, the likelihood is effectively zero for one or more pedigrees. Since Mx is going to try to take the logarithm of the likelihood, some corrective action is required. During iteration, a penalty function is used, but this rarely helps the case where there are poor starting values. To help guide the user to the problem with the starting values, Mx prints out 3 columns of information: the observed and expected means, and then the standardized difference between them. Now if these differences are large (say more than 3 for any variable), it would be a good idea to change starting values of either the means (to make the expected ones closer to the observed) or the variances (to make the standardized difference less). If the starting values of the means are very bad, then it would make sense to change them; however, if they are not, the error may occur with another vector in the dataset, in which case modifying the starting values to increase the expected variances should help. If not, examine the expected covariance matrix; sometimes large expected covariances can make a particular pairings of scores rather unlikely. It is usually better to supply starting values that specify a diagonal variance-covariance matrix, since the overall likelihood is simply the product of the likelihoods of the individual variables. If each of these likelihoods is reasonable, e.g. a standardized difference of less than two, then the overall likelihood will not produce problems unless the number of variables in the vector is very large.

5.5 User-defined Fit Functions

This is a simple example to illustrate the use of a user-defined fit function, in this case least-squares. The model statement evaluates to a scalar which is minimized by Mx. Note that this approach is generally less efficient than using built-in formulae available in Mx.

```
!
! User defined fit function - least squares
!
User defined function to fit to a correlation matrix
!                    by least squares
Data NInput_vars=3 NGroupies=1
CMatrix Symm
1
.2 1
.3 .4 1
Matrices
A Symm 3 3 = %O1
B Stan 3 3 Free
Covariances \tr((a-b)*(a-b))/
Options USer RSiduals
End
```

5.6 Correction for Ascertainment

In Section 4.2.3, we described how proband-ascertained ordinal data could be used to estimate population covariances or genetically informative parameters. This same truncate selection might be applied using a screening instrument, so that only individuals above a certain threshold value are sampled. If such an ascertainment scheme was applied in a pairwise fashion, such that only pairs in which at least one of the pair was above threshold, the likelihood of these observations would require correction for the necessarily missing pairs concordant for being below threshold.

Mathematically, this can be expressed as a double integral of the bivariate normal distribution:

$$L_{\tilde{A}} \quad = \quad \int_{-\infty}^{t}\int_{-\infty}^{t}\phi(v_1, v_2)\, dv_2 dv_1$$

where t is the ascertainment threshold, v_1 and v_2 are the liability values of individuals 1 and 2, and ϕ is the multinormal probability density function. The likelihood of a pair of observations x_1 and x_2 given the ascertainment scheme is therefore:

$$L(x_1, x_2|A) \quad = \quad \frac{\phi(x_1, x_2)}{1 - \int_{-\infty}^{t}\int_{-\infty}^{t}\phi(v_1, v_2)\, dv_2 dv_1}$$

If we use twice the negative log-likelihood as a function to minimize, then the ascertainment correction becomes more clearly distinct:

$$\begin{aligned} -2\ln L(x_1, x_2|A) &= -2\left\{\ln\left(\phi(x_1, x_2)\right) - \ln\left(1 - \int_{-\infty}^{t}\int_{-\infty}^{t}\phi(v_1, v_2)\, dv_2 dv_1\right)\right\} \\ &= -2\ln\left(\phi(x_1, x_2)\right) + 2\ln\left(1 - \int_{-\infty}^{t}\int_{-\infty}^{t}\phi(v_1, v_2)\, dv_2 dv_1\right) \end{aligned}$$

when we have m pairs, the likelihoods are summed over $j = 1 \ldots m$, giving

$$= \quad -2\ln\left(\phi(x_{1j}, x_{2j})\right) + 2m\ln\left(1 - \int_{-\infty}^{t}\int_{-\infty}^{t}\phi(v_1, v_2)\, dv_2 dv_1\right)$$

The first term is the function value calculated by Mx when fitting to raw data. The second term may be calculated by obtaining the value of the integral from a dummy categorical data group using zero observed frequencies in each cell. The threshold in this group should be fixed at the population value, and the correlation should be constrained to equal the correlation of these two variables as estimated from the ascertained data. The expected proportions under the bivariate normal distribution are passed as a matrix using the `%P` constraint as described in Section 3 on page 22. Note that this example does not follow the uppercase convention of uppercase for minimum input needed.

```
Simulated twin data.  Raw ML estimation
Data Ni=2 Ngroups=3 No=1000
Raw_data file=[neale.sas]mzasc.dat
Matrices
M Full 1 2
R Stan 2 2 free
Mean M /
Covariance R /
Matrix M
```

```
0 0
Bound -.99 .99 R 1 2
Options RSiduals
End

Dummy group to calculate expected cell proportions
Data Ni=2
CTable 2 2
0 0          ! It's full of zeros so it
0 0          ! contributes zero to the function
Matrices
T Full 2 1
R Stan 2 2 =R1
Thresholds T /
Covariance R /
Matrix T
1.282
1.282
Options RSiduals
End

Calculate ascertainment correction
Data Ni=0
Matrices
I iden 1 1
J izero 1 2
P full 2 2 = %P1
T full 1 1
Compute T*\ln(I-J*P*J') /
\index{matrix types!constrained!}
Matrix T
2000      ! twice the sample size of group 1
Options User-defined RSiduals Multiple
End
```

Appendix A

Using Mx under different operating systems

Currently Mx is available for VAX VMS, several unix systems (e.g. Vax BSD4.3, Sun-OS, IBM AIX, Dec Ultrix, HP), and MSDOS 3.0 (or higher, with or without Microsoft Windows) or OS2, for IBM compatible 386SX and above. Porting to other environments presents little difficulty, since the code is written in standard FORTRAN. Versions for most operating systems can be downloaded via binary ftp to sapphire.gen.vcu.edu

A.1 Using Mx under VAX VMS

Mx for VMS is distributed with command file (MX.COM) which deals with file extensions and checks to see whether the user has sufficient memory resources. Typically one user at a site, the administrator, will keep the command and executable files with read and execute permission set for all users. Then if users define the symbol mx with a command such as

```
$ MX :== "@DISK1:[BOSS.MX]MX.COM DUMMY"
```

where DISK1 is the name of the diskdrive, BOSS is the name of the administrator, and BOSS.MX is the name of the subdirectory in which the mx.com and mx.exe files are stored. The above symbol definition will permit users to run Mx either interactively or as a batch job. The only difference is that the former will produce a mesmerizing display of the Mx logo and the latter will free up your terminal to do something else while it runs. The output file may be read while the program is running, though emptying the print buffer has been reduced to improve performance.

We recommend that input files have the naming convention `CUTENAME.MX` where `CUTENAME` is a name of your choice. It allows you run Mx with the following syntax:

```
$ MX CUTENAME
```

or alternatively, you can run the job in batch with

```
$ BAT/Q=whateveryoulike MX CUTENAME
```

for short jobs, ignore the /Q= bit. With this syntax, the output will be in a file called `CUTENAME.MXO`; if you wish to use other extensions or names for input and output files, you could, for example, create a file called `BADNAME.ABC`, and use the syntax:

```
$ MX BADNAME.ABC AWFULNAME.XYZ
```

which would create an output file called `AWFULNAME.XYZ`. The command

```
$ MX BADNAME.ABC
```

would generate an output file called `BADNAME.MXO`

We can extend the idea of filename extensions a little further to include: covariance matrices, .COV; correlation matrices, .COR; contingency tables, .CTG; matrices, .MAT; variable length files, .VL; weight matrices, .ASY; inverted weight matrices, .ASI; vectors of means, .MEA; mx save files, .MXS. Of course, it doesn't make any difference to the program what you call the files, but some widely-used conventions such as these help you and other users to understand the content of directory when you (or your colleagues) look at it six months later.

A.2 Using Mx under UNIX

Mx may be used very simply in UNIX by typing

```
mx <inputfile >outputfile
```

and the parameter `&` may be added to run jobs in batch. For very cpu intensive applications the command

```
nice mx <inputfile >outputfile
```

may be used to run the job at lower priority to avoid overtaxing the system.

The IBM RISC 6000 series of computers has good facilities for editing files and executing programs. The INED editor, which is invoked with the command e, offers a user-defined menu system. Thus while editing an Mx input file, batch execution can be initiated simply by selecting an option on the menu. The menu file should be edited to have something like

```
|Batch run Mx on file being edited  |screen|
    (~neale/bin/mx <$FILE >"$FILE"o ; cat ~neale/bin/job.done) &|
```

where neale/bin would be changed to reflect the path where Mx is to be found, and job.done is an ascii file containing a suitable message such as"I have just finished another Mx job for you". This menu option should be configured to save files. It will then begin background Mx execution with a couple of keystrokes. Copies of the necessary profile file for the editor are supplied with the Mx program file; additional copies may be obtained from the author. The only gripe I have with the 6000 series INED editor is that somehow it will place an unrecognizable but invisible character at the start of the next line if the last character on the previous line is underscore (_). Thus matrix formulae spanning several lines should be written, e.g.,

```
 A*B*C' | C
_     C | D*E*D'
```

instead of

```
 A*B*C' | C_
      C | D*E*D'
```

which will not always work under the INED system. I don't have time to investigate this system-dependent problem, so it is likely to stay for a while — unless anyone has any suggestions?

A.3 Using Mx under MSDOS

MX is written in about 12,000 lines of FORTRAN, however it links to a further 20,000 lines of NPSOL code for optimization, so the resultant .exe file does not run within the 640K limit. Fortunately, the Lahey compiler allows binding of a loader to the code which permits Mx to use memory beyond the 640K limit in 386 and 486 machines. Naturally the presence of RAM disks etc. would decrease normally available machine memory, so a utility whatmem.exe is supplied with the program to allow you to check what is actually available. If the program runs out of memory, it will use virtual memory (disk space) instead, but obviously this can drastically decrease performance. The use of Stacker (file compression software) also seems to slow things down a lot, especially for the first run in a multiple fit file.

Note the difference between your computer running out of memory and Mx running out of workspace. Currently, the PC version is configured with 100,000 double precision words of workspace; larger workspace versions are available by special request.

MX now no longer requires that the output files be specified on the command line. This means that under windows and similar environments, .mx files can be associated with the mx.exe program. Double clicking on a .mx file will then run mx and create an output file of the same name but with .mxo as its extension.

A.4 Using Mx under VM

VM is a commonly encountered operating system on IBM mainframes such as the 3081 and 3090 series. It is not my favorite operating system. Porting Mx to this environment was more difficult than to any other, and the `Save` and `Get` functions for binary files are not yet operational. The saving grace of these machines is their speed, although they are now matched by cheaper and easier to use unix workstations.

Appendix B

Error Messages

B.1 General Input Reading

If, while looking for a number, a (non-comment) input line with only non-numeric characters on it (e.g. `MATRIX P`) is found, the program issues a warning:

```
**WARNING** Non-numeric characters found when trying to read a
number
```

This is to alert the user to what is probably an error of not reading enough numbers, either because:

- input lines are too long (120 column maximum)
- the matrix was given the wrong dimensions in the Matrices command
- too few numbers have been supplied

This is not a fatal error, but normally some other problem will arise. Similarly, if too many numbers have been provided for a matrix, (or a mistake was been made when defining the type or dimensions of the matrix) the program will usually try to read a number as a keyword, ask something like "Just what is this keyword supposed to mean?" and stop.

Sometimes an integer overflow will occur if too many digits have been read in free format for a number. Try to keep the number of digits to 9 or less. If you really need more precision, use the exponential format (D+00) or read data from a file.

B.2 Error Codes

This is a list of the error codes reported by Mx. The error messages are supposed to be self-explanatory, but they are usually quite brief.

1 *First input line after title must be DATA.* Will occur if the Title line has been forgotten.

2 *Data line must have NI and NO specified.* (Data groups only)

3 *End of file while trying to read title.*

4 *Must specify Matrices at this point...*

5 *Not a legal matrix name.* Use A-Z, one letter only

6 *Illegal matrix type...* Check for typos.

7 *Sorry, I seem to be at the end of your input file.* Check NG is correct.

8 *Number of selected variables must be less than or equal to NI ...* Otherwise you won't be analyzing something sensible and positive definite.

9 *Error - no variables selected for analysis.*

10 *Terribly sorry, I don't have enough workspace.*

11 *Please try not to refer to matrices that you haven't specified.*

12 *Please use only PATTERN or SPECIFICATION throughout.* Pattern and Specification cannot be used in the same input file.

13 *Error - no matrix specified...*

14 *More than 3 dimensions specified.* After a matrix name, there should be a maximum of 3 numbers to identify the element.

15 *Error - Matrix has not been specified.*

16 *Matrix has been specified not to have free elements.* See Table 3.1 to figure out which types of matrix can have free elements.

17 *You cannot alter off-diagonal elements of a diagonal matrix.*

18 *Incorrect element of a Subdiagonal matrix specified.* You can't alter elements on or above the diagonal of as Subdiagonal matrix

19 *Incorrect element of a Standardized matrix specified.* You can't alter elements on the diagonal of a Standardized matrix.

20 *Please use 2 or 3 dimensional format for elements.* Array references should be group,row,col, with group optional.

21 *Sorry, I can't find your matrix element.* I really should tidy up sometime