Chapter 6
Fitting a Factor Model with the Two-Stage Approach

Abstract In this chapter I will illustrate fitting a factor model within a MASEM analysis using the metaSEM package. The data come from a meta-analysis performed by Fan et al. (Personality and Individual Differences 48(7):781–785, 2010), who collected correlation matrices of the 8 subscales of a test to measure "Emotional intelligence" from 19 studies. The preparation of the data, and the fixed and random effects Stage 1 analyses are explained step by step. Next, the Stage 2 factor model is fit to the pooled correlation matrix from the random effects Stage 1 analysis. All steps that have to be taken to perform the analyses are discussed, as well as the relevant output.

Keywords Meta-analytic structural equation modeling · MetaSEM · Factor model · Emotional intelligence · MSCEIT · Fixed effects · Random effects

6.1 Introduction

Fan et al. (2010) used meta-analytic factor analysis to investigate the factor structure of a measurement instrument of emotional intelligence, the Mayer-Salovey-Caruso Emotional Intelligence Test Version 2.0 (MSCEIT). Emotional intelligence is defined as a set of skills hypothesized to contribute to the accurate appraisal and expression of emotion, the effective regulation of emotion, and the use of feelings to motivate, plan, and achieve in one's life (Salovey and Mayer 1989). The MSCEIT consists of 8 subscales. Previous research on the factor structure of the MSCEIT lead to contradictory results, and a MASEM made it possible to compare the fit of several proposed factor models on the aggregated data across 19 studies. Based on these analyses, a three-factor model was found to have the best fit. In this section I will replicate the fixed effects analysis of Fan et al. and additionally run a random effects MASEM. The data and script to replicate the analyses can be found on my website (http://suzannejak.nl/masem).

S. Jak, *Meta-Analytic Structural Equation Modelling*,
SpringerBriefs in Research Synthesis and Meta-Analysis,
DOI 10.1007/978-3-319-27174-3_6

6.2 Preparing the Data

Fan et al. collected 19 correlation matrices from different studies. Most of the studies reported all correlations between the 8 scales of the MSCEIT, for some studies the correlation had to be deduced from other information (see Fan et al.) and for two studies one and two variables were missing. The correlation matrices are collected in a text file, “fan_msceit.dat”, which contains the lower triangular of the matrix in each study. This is a part of the file:

```
1
.34    1
.36    .33    1
.24    .22    .17    1
.23    .12    .32    .31    1
.14    .11    .26    .19    .43    1
.14    -.06   .22    .18    .23    .34    1
.11    -.11   .16    .31    .43    .48    .47    1

1
.406   1
.312   .376   1
.373   .450   .375   1
.258   .297   .189   .227   1
 0      0      0      0      0     NA
.309   .372   .270   .324   .361   0      1
.322   .388   .282   .337   .377   0      .511   1
```

The function readLowTriMat() can be used to store these matrices in a list that can serve as input for the analysis. The function takes the filename and the number of variables per study as arguments, and then creates a list of correlation matrices. If variables are missing in some studies, this should be indicated by NA on the diagonal. The second matrix shown above does not contain information about the sixth variable, the NA on the diagonal ensures that the associated rows and columns will be filtered out during the analysis (so it does not matter what values are given for the missing correlations). The next two lines of code create the list of matrices and a vector with the associated sample sizes. The argument skip = 1 is needed because the first line of the file contains copyright information, and should be skipped by the function.

```
cordat <- readLowTriMat(file = "fan_msceit.dat", no.var = 8,
                        skip = 1)

N <- c(5000,457,412,655,150,450,138,237,314,405,
       375,239,260,266,209,84,192,523,198)
```

6.3 Fixed Effects Analysis

The `tssem()` function is used to estimate the pooled correlation matrix under the fixed effects model.

```
stage1fixed <- tssem1(my.df = cordat, n = N,
                      method = "FEM")
summary(stage1fixed)
```

Leading to this output:

```
Coeffıcıents.
        Estimate Std.Error z value  Pr(>|z|)
S[1,2] 0.3690033 0.0084335  43.755 < 2.2e-16 ***
S[1,3] 0.3164490 0.0088628  35.706 < 2.2e-16 ***
S[1,4] 0.3291198 0.0087740  37.511 < 2.2e-16 ***
S[1,5] 0.1857085 0.0094337  19.686 < 2.2e-16 ***
S[1,6] 0.1944074 0.0094758  20.516 < 2.2e-16 ***
S[1,7] 0.2135400 0.0093234  22.904 < 2.2e-16 ***
S[1,8] 0.2235946 0.0092685  24.124 < 2.2e-16 ***
S[2,3] 0.3643780 0.0085434  42.650 < 2.2e-16 ***
S[2,4] 0.3263959 0.0087876  37.143 < 2.2e-16 ***
S[2,5] 0.2337159 0.0092278  25.327 < 2.2e-16 ***
S[2,6] 0.2018553 0.0094436  21.375 < 2.2e-16 ***
S[2,7] 0.2428587 0.0092039  26.387 < 2.2e-16 ***
S[2,8] 0.2215793 0.0092865  23.860 < 2.2e-16 ***
S[3,4] 0.3655632 0.0085451  42.781 < 2.2e-16 ***
S[3,5] 0.3333873 0.0087821  37.962 < 2.2e-16 ***
S[3,6] 0.2779142 0.0091839  30.261 < 2.2e-16 ***
S[3,7] 0.3418394 0.0087278  39.167 < 2.2e-16 ***
S[3,8] 0.3173547 0.0088668  35.791 < 2.2e-16 ***
S[4,5] 0.2492657 0.0092489  26.951 < 2.2e-16 ***
S[4,6] 0.2572274 0.0093092  27.632 < 2.2e-16 ***
S[4,7] 0.3242843 0.0088399  36.684 < 2.2e-16 ***
S[4,8] 0.3189749 0.0088592  36.005 < 2.2e-16 ***
S[5,6] 0.4907931 0.0075245  65.226 < 2.2e-16 ***
S[5,7] 0.3641924 0.0085319  42.686 < 2.2e-16 ***
S[5,8] 0.3272308 0.0087472  37.410 < 2.2e-16 ***
S[6,7] 0.3211214 0.0088663  36.218 < 2.2e-16 ***
S[6,8] 0.3489128 0.0086744  40.223 < 2.2e-16 ***
S[7,8] 0.5065221 0.0072959  69.425 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’
1
```

```
Goodness-of-fit indices:
                                              Value
Sample size                              10564.0000
Chi-square of target model                1818.8709
DF of target model                         484.0000
p value of target model                      0.0000
Chi-square of independence model        19130.4290
DF of independence model                   512.0000
RMSEA                                        0.0704
SRMR                                         0.1267
TLI                                          0.9242
CFI                                          0.9283
AIC                                        850.8709
BIC                                      -2665.4894
OpenMx status1: 0 ("0" or "1": The optimization is considered
fine.
 Other values indicate problems.)
```

The degrees of freedom are equal to the number of observed correlation coefficients minus the number of estimated correlation coefficients. There are 17 observed complete correlation matrices with 8 * 7/2 = 28 correlation coefficients each. One study missed one variable, and has 7 * 6/2 = 21 coefficients, and one study missed 2 variables and has 6 * 5/2 = 15 observed coefficients. So, in total there are 17 * 28 + 21 + 15 = 512 observed correlation coefficients. The model has 28 parameters, which are the correlation coefficients that are assumed to be equal across studies. Hence, degrees of freedom are 512 − 28 = 484. This calculation leads to the correct number of degrees of freedom, but in reality the diagonal elements of the observed correlation matrices are also counted as observed statistics, and a diagonal matrix is also estimated for each observed matrix (see Eq. 2.7 in Chap. 2). Because the number of observed diagonal elements is equal to the number of estimated diagonal elements, degrees of freedom do not change by evaluating the diagonal elements.

The chi-square is significant ($\chi2_{(484)} = 1818.87$, $p < 0.05$), exact fit of the Stage 1 model does not hold, indicating that exact homogeneity of the correlation coefficients across studies is rejected. The RMSEA of 0.07 however shows acceptable approximate fit, which could serve as an indication that homogeneity holds

Table 6.1 Pooled correlation matrix of the research variables from the fixed effects analysis

	1	2	3	4	5	6	7	8
1. Faces	1							
2. Pictures	0.37	1						
3. Facilitation	0.32	0.36	1					
4. Sensations	0.33	0.33	0.37	1				
5. Changes	0.19	0.23	0.33	0.25	1			
6. Blends	0.19	0.20	0.28	0.26	0.49	1		
7. Emotional management	0.21	0.24	0.34	0.32	0.36	0.32	1	
8. Emotional relations	0.22	0.22	0.32	0.32	0.33	0.35	0.51	1

approximately, and the pooled correlation matrix from the fixed effects analysis could be used to fit the structural model. Table 6.1 shows the rounded parameter estimates in matrix form. These coefficients can be extracted from the output with `coef(stage1fixed)`.

6.4 Random Effects Analysis

Stage 1: A random effects analysis also seems appropriate for these data. If the heterogeneity of the correlation coefficients is not substantial, the results will not be very different from the fixed effects analysis. The following code will run the random effects Stage 1 analysis. As it was not possible to estimate the study-level covariance, the random effects type “`Diag`” is used.

```
stage1random <- tssem1(my.df = cordat, n = N, method = "REM",
                       RE.type = "Diag")

summary(stage1random)
```

To save space, the raw output is not shown here. The Q-statistic is significant ($Q_{(484)} = 2061.08$), so homogeneity is rejected based on this test. The I^2 of the correlation coefficients range between 0.19 and 0.88 indicating substantial heterogeneity. Table 6.2 shows the pooled correlation matrix from the random effects analysis (with the I^2 values above the diagonal).

The correlation coefficients are somewhat different from the fixed effects estimates. Another difference is in the asymptotic variance covariance matrix of these correlation coefficients that will be used as a weight matrix in the Stage 2 analysis. The asymptotic variance from the random effects analysis will be larger, leading to larger confidence intervals around the Stage 2 estimates.

Stage 2: I am going to fit the structural model to the pooled random effects matrix from Stage 1. Figure 6.1 shows the 3-factor structure that will be fitted to these data. The specification of the parameter matrices for the Stage 2 model does not differ between the random or fixed approach. In the illustration of the path model in Chap. 5, I already introduced the A-matrix with regression coefficients and the S-matrix with variances and covariances. These matrices feature in the factor model as well. The A-matrix contains the factor loadings (λ’s in Fig. 6.1), and matrix S contains the residual variances (θ’s in Fig. 6.1) as well as the factor variances and covariances (φ’s in Fig. 6.1). For factor analysis, a third matrix is needed, which is a matrix that indicates which variables are observed and which variables are latent. This is matrix F. In the current example, we have 8 observed variables and 3 factors. Therefore both the A-matrix and the S-matrix will have 11 rows and 11 columns. The F-matrix will have 8 rows and 11 columns. Matrix F is a selection matrix that filters out the latent variables, it is an identity matrix with the rows associated with the latent variables removed. In the current

Table 6.2 Pooled correlations (below diagonal) and I^2 (above the diagonal) of the research variables from the random effects analysis

	1	2	3	4	5	6	7	8
1. Faces	1	0.82	0.68	0.19	0.47	0.41	0.41	0.42
2. Pictures	0.37	1	0.59	0.37	0.30	0.46	0.57	0.46
3. Facilitation	0.31	0.32	1	0.65	0.78	0.78	0.84	0.68
4. Sensations	0.32	0.31	0.33	1	0.77	0.86	0.74	0.77
5. Changes	0.22	0.21	0.27	0.27	1	0.88	0.86	0.84
6. Blends	0.20	0.20	0.24	0.25	0.45	1	0.76	0.74
7. Emotional management	0.21	0.21	0.30	0.28	0.28	0.28	1	0.87
8. Emotional relations	0.22	0.19	0.27	0.31	0.31	0.32	0.45	1

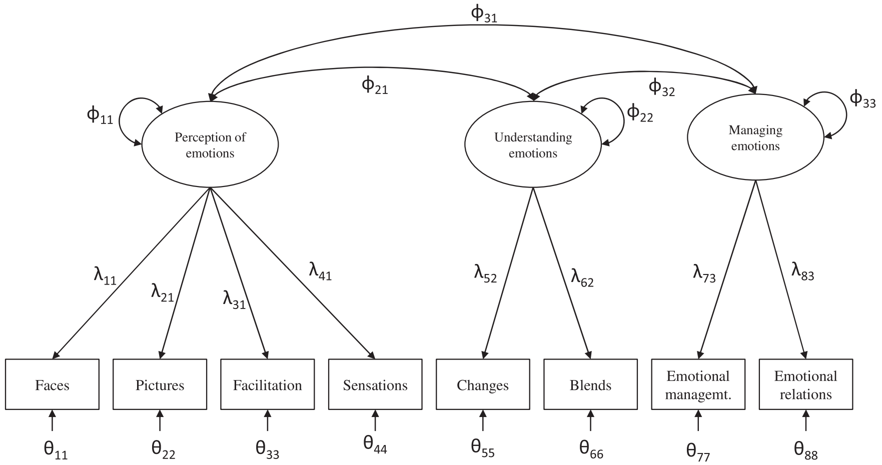

Fig. 6.1 Three factor model on the subscales of the MSCEIT

example, we put the observed variables first, the F matrix can be created using the `create.Fmatrix()` function directly:

```
F <- create.Fmatrix(c(1,1,1,1,1,1,1,1,0,0,0), name="F")
```

Next, we need the A-matrix. I am going to create the A-matrix in steps. First I will create a 8 by 3 matrix lambda, which has the factor loadings.

```
lambda <- matrix(
             c("0.3*L11",0,0,
               "0.3*L21",0,0,
               "0.3*L31",0,0,
               "0.3*L41",0,0,
                0,"0.3*L52",0,
                0,"0.3*L62",0,
                0,0,"0.3*L73",
                0,0,"0.3*L83"),
             nrow=8,
             ncol=3,
             byrow = TRUE)
```

Like the matrices in the path model, if a number is specified in the lambda matrix, it indicates that the factor loading is not estimated but fixed (fixed at the given number, zero in this case). If it is not a number, but for example “0.3 * L11”, the parameter is given a starting value of 0.3 and it gets the label “L11”. To correctly fix and free elements it may help to think of the columns of lambda as being associated with the common factors and the rows as being associated with the indicators. For example, if indicator number three loads on the first factor (or, the third indicator variable regresses on the first factor), we specify a free parameter for the element in the third row, first column (“0.3 * L31”). Note that the `matrix()` function fills in the values column wise by default, so we use the argument `byrow` = `TRUE`. The object lambda looks like shown below.

```
> lambda
     [,1]      [,2]      [,3]
[1,] "0.3*L11" "0"       "0"
[2,] "0.3*L21" "0"       "0"
[3,] "0.3*L31" "0"       "0"
[4,] "0.3*L41" "0"       "0"
[5,] "0"       "0.3*L52" "0"
[6,] "0"       "0.3*L62" "0"
[7,] "0"       "0"       "0.3*L73"
[8,] "0"       "0"       "0.3*L83"
```

The A-matrix should be an 11 by 11 matrix, in which the factor loadings are in rows 1–8 (associated with the observed variables) and columns 9–11 (associated with the factors). The rest of the matrix should consist of zeros, as there are no other regression coefficients than factor loadings in the model. The zeros can be added to the A matrix by adding a 8 by 8 matrix to the left of lambda and consequently a 3 by 11 matrix with zeros below using the `cbind()` and `rbind()` functions. Next, the `as.mxMatrix()` function is used to create the matrices that are used by OpenMx, which are a matrix indicating the labels of the parameters, a matrix with the starting values of the parameters and a matrix indicating whether the parameter is freely estimated (indicated by `TRUE`) or not (indicated by `FALSE`).

```
A <- rbind(cbind(matrix(0,ncol=8,nrow=8),lambda),
             matrix(0, nrow=3, ncol=11))

A <- as.mxMatrix(A)

# not required but it helps to provide labels
dimnames(A) <- list(
c("face","pict","faci","sens","chen","blen","emma","emre","F1
","F2","F3"),
c("face","pict","faci","sens","chen","blen","emma","emre","F1
","F2","F3"))
```

The resulting A-matrices look as follows.

```
FullMatrix 'A'

$labels
     face pict faci sens chen blen emma emre F1    F2    F3
face NA   NA   NA   NA   NA   NA   NA   NA   "L11" NA    NA
pict NA   NA   NA   NA   NA   NA   NA   NA   "L21" NA    NA
faci NA   NA   NA   NA   NA   NA   NA   NA   "L31" NA    NA
sens NA   NA   NA   NA   NA   NA   NA   NA   "L41" NA    NA
chen NA   NA   NA   NA   NA   NA   NA   NA   NA    "L52" NA
blen NA   NA   NA   NA   NA   NA   NA   NA   NA    "L62" NA
emma NA   NA   NA   NA   NA   NA   NA   NA   NA    NA    "L73"
emre NA   NA   NA   NA   NA   NA   NA   NA   NA    NA    "L83"
F1   NA   NA   NA   NA   NA   NA   NA   NA   NA    NA    NA
F2   NA   NA   NA   NA   NA   NA   NA   NA   NA    NA    NA
F3   NA   NA   NA   NA   NA   NA   NA   NA   NA    NA    NA

$values
     face pict faci sens chen blen emma emre  F1  F2  F3
face    0    0    0    0    0    0    0    0 0.3 0.0 0.0
pict    0    0    0    0    0    0    0    0 0.3 0.0 0.0
faci    0    0    0    0    0    0    0    0 0.3 0.0 0.0
sens    0    0    0    0    0    0    0    0 0.3 0.0 0.0
chen    0    0    0    0    0    0    0    0 0.0 0.3 0.0
blen    0    0    0    0    0    0    0    0 0.0 0.3 0.0
emma    0    0    0    0    0    0    0    0 0.0 0.0 0.3
emre    0    0    0    0    0    0    0    0 0.0 0.0 0.3
F1      0    0    0    0    0    0    0    0 0.0 0.0 0.0
F2      0    0    0    0    0    0    0    0 0.0 0.0 0.0
F3      0    0    0    0    0    0    0    0 0.0 0.0 0.0

$free
      face  pict  faci  sens  chen  blen  emma  emre    F1    F2    F3
face FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
pict FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
faci FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
sens FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
chen FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
blen FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
emma FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
emre FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
F1   FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
F2   FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
F3   FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

$lbound: No lower bounds assigned.

$ubound: No upper bounds assigned.
```

The S-matrix with variances and covariances will also be created in steps. It actually consists of a variance covariance matrix of the observed variables and a variance covariance matrix of the factors. First, I am going to create the matrix with the residual variances of the observed variables. These are represented by θ's in Fig. 6.1. The matrix theta is an 8 by 8 matrix, with freely estimated parameters on its diagonal. As there are no residual covariances in the model, all off-diagonal elements are fixed at zero. First, I create an 8 by 8 matrix with zero's, and then I add the vector with the information about the residual variance on its diagonal.

```
theta <- matrix(0,nrow = 8,ncol = 8)
diag(theta) <- c("0.1*t11","0.1*t22","0.1*t33","0.1*t44",
                 "0.1*t55","0.1*t66","0.1*t77","0.1*t88")
```

The phi matrix contains the variances and covariances of the factors. For identification, the factor variances are fixed at 1. The correlations between the factors are specified off-diagonal.

```
phi <- matrix(
            c(1,"0.1*phi21","0.1*phi31",
               "0.1*phi21",1,"0.1*phi32",
               "0.1*phi31","0.1*phi32",1),
             nrow = 3,
             ncol = 3)
```

The function `bdiagMat()` creates the larger S-matrix from the theta and phi matrices. By using the `as.MxMatrix()` function on this S-matrix, the matrices with labels, starting values and free/fixed elements to be used by OpenMx are created.

```
S <- bdiagMat(list(theta, phi))
S <- as.mxMatrix(S)

dimnames(S) <- list(
c("face","pict","faci","sens","chen","blen","emma","emre","F1
","F2","F3"),
c("face","pict","faci","sens","chen","blen","emma","emre","F1
","F2","F3"))
```

The resulting S-matrices look like below.

```
FullMatrix 'S'

$labels
     face  pict  faci  sens  chen  blen  emma  emre  F1      F2      F3
face "t11" NA    NA    NA    NA    NA    NA    NA    NA      NA      NA
pict NA    "t22" NA    NA    NA    NA    NA    NA    NA      NA      NA
faci NA    NA    "t33" NA    NA    NA    NA    NA    NA      NA      NA
sens NA    NA    NA    "t44" NA    NA    NA    NA    NA      NA      NA
chen NA    NA    NA    NA    "t55" NA    NA    NA    NA      NA      NA
blen NA    NA    NA    NA    NA    "t66" NA    NA    NA      NA      NA
emma NA    NA    NA    NA    NA    NA    "t77" NA    NA      NA      NA
emre NA    NA    NA    NA    NA    NA    NA    "t88" NA      NA      NA
F1   NA    NA    NA    NA    NA    NA    NA    NA    NA      "phi21" "phi31"
F2   NA    NA    NA    NA    NA    NA    NA    NA    "phi21" NA      "phi32"
F3   NA    NA    NA    NA    NA    NA    NA    NA    "phi31" "phi32" NA

$values
     face pict faci sens chen blen emma emre  F1  F2  F3
face  0.1  0.0  0.0  0.0  0.0  0.0  0.0  0.0 0.0 0.0 0.0
pict  0.0  0.1  0.0  0.0  0.0  0.0  0.0  0.0 0.0 0.0 0.0
faci  0.0  0.0  0.1  0.0  0.0  0.0  0.0  0.0 0.0 0.0 0.0
sens  0.0  0.0  0.0  0.1  0.0  0.0  0.0  0.0 0.0 0.0 0.0
chen  0.0  0.0  0.0  0.0  0.1  0.0  0.0  0.0 0.0 0.0 0.0
blen  0.0  0.0  0.0  0.0  0.0  0.1  0.0  0.0 0.0 0.0 0.0
emma  0.0  0.0  0.0  0.0  0.0  0.0  0.1  0.0 0.0 0.0 0.0
emre  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.1 0.0 0.0 0.0
F1    0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0 1.0 0.1 0.1
F2    0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0 0.1 1.0 0.1
F3    0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0 0.1 0.1 1.0

$free
      face  pict  faci  sens  chen  blen  emma  emre    F1    F2    F3
face  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
pict FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
faci FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
sens FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
chen FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
blen FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
emma FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
emre FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
F1   FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE
F2   FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE
F3   FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE

$lbound: No lower bounds assigned.

$ubound: No upper bounds assigned.
```

Now the required matrices for the Stage 2 analysis are created, the model can be fit to the pooled matrix from Stage 1. As the heterogeneity seems to be substantial, I will fit the model to the Stage 1 matrix from the random effects analysis. The `tssem()` function distils the averaged correlation matrix and the asymptotic variance covariance matrix from the Stage 1 object `stage1random`. As with the path model I used the `diag.constraints` = `TRUE` and I asked for likelihood based confidence intervals around the parameter estimates.

```
stage2_random <- tssem2(stage1random, Amatrix=A, Smatrix=S,
               Fmatrix=F, diag.constraints=TRUE, intervals="LB")
```

The output can be viewed using the `summary()` function.

```
95% confidence intervals: Likelihood-based statistic
Coefficients:
      Estimate Std.Error  lbound  ubound
L11    0.53025        NA 0.49697 0.56407
L21    0.51982        NA 0.48654 0.55357
L31    0.57671        NA 0.53907 0.61460
L41    0.58797        NA 0.55268 0.62363
L52    0.67185        NA 0.61658 0.72768
L62    0.63164        NA 0.57877 0.68495
L73    0.65046        NA 0.60409 0.69766
L83    0.68395        NA 0.63626 0.73287
t11    0.71883        NA 0.68182 0.75302
t22    0.72979        NA 0.69356 0.76327
t33    0.66741        NA 0.62227 0.70940
t44    0.65429        NA 0.61108 0.69455
t55    0.54862        NA 0.47047 0.61983
t66    0.60103        NA 0.53084 0.66503
t77    0.57691        NA 0.51325 0.63508
t88    0.53222        NA 0.46289 0.59518
phi21  0.60974        NA 0.55321 0.67299
phi31  0.62987        NA 0.57794 0.68595
phi32  0.66528        NA 0.59029 0.74944

Goodness-of-fit indices:
                                                Value
Sample size                                10564.0000
Chi-square of target model                    42.2013
DF of target model                            17.0000
p value of target model                        0.0006
Number of constraints imposed on "Smatrix"     8.0000
DF manually adjusted                           0.0000
Chi-square of independence model            2486.1537
DF of independence model                      28.0000
RMSEA                                          0.0118
SRMR                                           0.0257
TLI                                            0.9831
CFI                                            0.9897
AIC                                            8.2013
BIC                                         -115.3073
OpenMx status1: 0 ("0" or "1": The optimization is considered
fine.
Other values indicate problems.)
```

The 8 by 8 pooled correlation matrix on which the model is fitted contains 28 correlation coefficients. The model contains 8 factor loadings, 8 residual variances, and 3 factor covariances (factor variances were fixed at 1), which sums up to 19 parameters. However, because during estimation the 8 diagonal elements of the estimated covariance are constrained to be 1, this reduces the number of parameters by 8. Degrees of freedom are therefore equal to $28 - 19 + 8 = 17$. The model does not fit exactly, as the chi-square is significant ($\chi^2_{(17)} = 42.20$, $p < 0.05$).

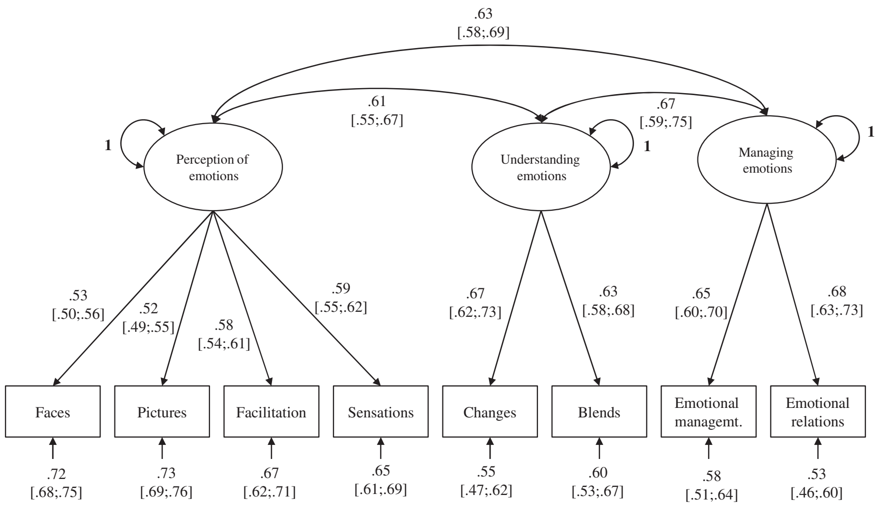

Fig. 6.2 Factor model on the MSCEIT with parameter estimates and 95 % confidence intervals

The RMSEA value of 0.012 indicates close approximate fit, and the CFI of 0.99 also indicated satisfactory fit of the model. The parameter estimates with the confidence intervals could therefore be interpreted. All factor loadings are positive, larger than 0.50, and significantly larger than zero. The correlations between the three factors is substantial (0.62, 0.64 and 0.67), but not so large that some factors may be redundant. Figure 6.2 shows the graphical model with the parameter estimates.

As long as there are no mediating variables in the model, an alternative to using the argument `diag.constraints` = `TRUE` in the `tssem2()` function is to use `diag.constraints` = `FALSE` (or to leave out this argument). This will lead to the same fit results and parameter estimates, but the way the analysis is performed is different. Without the diagonal constraints, the diagonals are totally left out of the analysis (the diagonal entries are not counted as observations), and no residual variances ($\mathbf{\Theta}$) are estimated. Because a correlation matrix is analyzed, we know that the total variance of each indicator equals 1. The residual variances can therefore be calculated from the matrix with estimated factor loadings ($\mathbf{\Lambda}$) and matrix with estimated factor variances and covariances ($\mathbf{\Phi}$) using $\mathbf{\Theta} = \mathbf{I} - \text{diag}(\mathbf{\Lambda}\mathbf{\Phi}\mathbf{\Lambda}^{\mathrm{T}})$, where $\mathbf{I}$ is an 8 by 8 identity matrix.

Acknowledgment I am grateful to Dr. Huiyong Fan and the co-authors of his study for sharing the collected correlation matrices with me. Indirectly, I am also grateful to the authors of the primary studies, who were willing to share their correlation matrices with them. I would like to stress that the ownership of these data belongs to the primary authors. I put the file with the correlation matrices on my Web site for educational purposes. Any part of these data should not be used for other purpose than research or education, without prior written permission from these primary authors or Dr. Huiyong Fan.

References

Fan, H., Jackson, T., Yang, X., Tang, W., & Zhang, J. (2010). The factor structure of the Mayer-Salovey-Caruso Emotional Intelligence Test V2.0 (MSCEIT): A meta-analytic structural equation modeling approach. *Personality and Individual Differences, 48*(7), 781–785.

Salovey, P., & Mayer, J. D. (1989). Emotional intelligence. *Imagination, Cognition and Personality, 9*(3), 185–211.