A DYADIC IRT MODEL - SUPPLEMENTARY MATERIAL

Description of speed-dating items

The link to the original PDF speed dating survey provided in Fisman et al. (2006) is unfortunately broken. However, a copy remains publicly available on page 8 of the "data key" document at this link (also containing the data) provided by Andrew Gelman:

`http://www.stat.columbia.edu/~gelman/arm/examples/speed.dating/`

The following summarizes the survey described in that document, as relevant to our measurement of "attractiveness." Each actor was given these instructions, to be filled out about each partner immediately after their "date" during the event:

> "Circle 'Yes' or 'No' below the ID number of each person you meet to indicate whether or not you would like to see him or her again. Rate their attributes on a scale of 1-10: (1=awful, 10=great). If you haven't formed an opinion based on your conversation, fill in N/A, but please fill in all boxes. This will be TOTALLY confidential and will NOT be shared with anyone. Then, answer the remaining questions for each person you meet."

The following reproduces the scorecard that each actor was given to record their responses. The "Decision" response relates to whether the actor wanted to see their partner again, i.e. the distal outcome as described in the main text. The second "Attributes" table was used by each actor to rate their partner's characteristics. We have included only those items that pertained to the partner's attractiveness. No definition or description was given for any of the attractiveness items; this table represents exactly the choices as posed to each actor.

ID #:	1	2	3	4	5	6	7	8	9	10
Decision	yes no	yes no	yes no	yes no	yes no	yes no	yes no	yes no	yes no	yes no

Attributes (1=awful, 10=great)										
Attractive										
Sincere										
Intelligent										
Fun										
Ambitious										

Stan Code

```
# clears workspace:
rm(list = ls())

library(rstan)
rstan_options(auto_write = TRUE)
options(mc.cores = 8)

library(tidyverse)

# load dataset:
load(file = "df.complete.Rdata")
load(file = "dpair.specific.Rdata")

# no gen with int stan model
modelngwi <- "
functions {
    real pcminteract(int x, real alpha, real beta, real gamma, vector delta) {
      vector[rows(delta) + 1] unsummed;
      vector[rows(delta) + 1] probs;
      unsummed = append_row(rep_vector(0.0, 1), alpha + beta + gamma - delta);
      probs = softmax(cumulative_sum(unsummed));
      return categorical_lpmf(x+1 | probs);
    }
  }
data {
  int<lower = 1> I;                 // # items
  int<lower = 1> A;                 // # actors (or partners)
  int<lower = 1> U;                 // # undirected pairs
  int<lower = 1> N;                 // # responses
  int<lower = 1> D;                 // # decisions
  int<lower = 1> B;                 // integer value for # distal regression parameters
  int<lower = 1, upper = A> aa[N];  // size N array to index actors for each response
  int<lower = 1, upper = A> pp[N];  // size N array to index partners for each response
  int<lower = 1, upper = I> ii[N];  // size N array to index items for each response
  int<lower = 0> x[N];              // size N array for responses; x = 0, 1 ... m_i
  int<lower = 1, upper = U> dd[N];  // size N array to index undirected pairs for each response
  int<lower = 1, upper = 2> mm[N];  // size N array to index match for each response
  int<lower = 1, upper = A> aaa[D]; // size D array to index actors for each decision
  int<lower = 1, upper = A> ppp[D]; // size D array to index partners for each decision
  int<lower = 1, upper = U> ddd[D]; // size D array to index undirected pairs for each decision
  int<lower = 1, upper = 2> mmm[D]; // size D array to index match for each decision
```

```
  int<lower = 0, upper = 1> zzz[D];  // size D array for decisions
}
transformed data {
  int M;                             // # parameters per item (same for all items)
    M = max(x);
}
parameters {
  vector[M] delta[I];                // length m vector for each item i
  vector[2] AB[A];                   // size 2 vector of alpha and beta for each person;
  vector[2] GG[U];                   // size 2 vector of gammas for each undirected pair;
  real<lower = 0> sigmaA;            // real sd of alpha
  real<lower = 0> sigmaB;            // real sd of beta
  real<lower = 0> sigmaG;            // real sd of gamma
  real<lower = -1, upper = 1> rhoAB; // real cor between alpha and beta (within person)
  real<lower = -1, upper = 1> rhoG;  // real cor between gammas (within pair)
  real beta[B];                      // B-dimensional array of real valued of beta
                                     // (distal regression parameters)
}
transformed parameters {
  cov_matrix[2] SigmaAB;             // 2x2 covariance matrix of alpha and beta
  cov_matrix[2] SigmaG;              // 2x2 covariance matrix of gammas
  SigmaAB[1, 1] = sigmaA^2;
  SigmaAB[2, 2] = sigmaB^2;
  SigmaAB[1, 2] = rhoAB * sigmaA * sigmaB;
  SigmaAB[2, 1] = rhoAB * sigmaA * sigmaB;
  SigmaG[1, 1] = sigmaG^2;
  SigmaG[2, 2] = sigmaG^2;
  SigmaG[1, 2] = rhoG * sigmaG^2;
  SigmaG[2, 1] = rhoG * sigmaG^2;
}
model {
  AB ~ multi_normal(rep_vector(0.0, 2), SigmaAB);
  GG ~ multi_normal(rep_vector(0.0, 2), SigmaG);
  for (n in 1:N){
    target += pcminteract(x[n], AB[aa[n],1], AB[pp[n],2], GG[dd[n], mm[n]], delta[ii[n]]);
  }
  for (d in 1:D){
    //distal logistic regression
    target += bernoulli_logit_lpmf(zzz[d] | (beta[1]
    + beta[2]*AB[aaa[d],1]
    + beta[3]*AB[ppp[d],1]
    + beta[4]*AB[aaa[d],2]
    + beta[5]*AB[ppp[d],2]
```

```
    + beta[6]*GG[ddd[d], mmm[d]]
    + beta[7]*GG[ddd[d], (3-mmm[d])]
    + beta[8]*AB[aaa[d],1]*AB[ppp[d],1]
    + beta[9]*AB[aaa[d],2]*AB[ppp[d],2]
    + beta[10]*GG[ddd[d], mmm[d]]*GG[ddd[d], (3-mmm[d])]));
  }
}
"

# no gen with int model
I <- max(df.complete$item)
A <- max(df.complete$actor)
U <- max(df.complete$unique.pair)
N <- nrow(df.complete)
D <- nrow(dpair.specific)
B <- 10

data <- list(I = I,
             A = A,
             U = U,
             N = N,
             D = D,
             B = B,
             aa = as.numeric(df.complete$actor),
             pp = as.numeric(df.complete$partner),
             ii = as.numeric(df.complete$item),
             x = as.numeric(df.complete$x),
             dd = as.numeric(df.complete$unique.pair),
             mm = as.numeric(df.complete$selector),
             aaa = as.numeric(dpair.specific$actor),
             ppp = as.numeric(dpair.specific$partner),
             ddd = as.numeric(dpair.specific$unique.pair),
             mmm = as.numeric(dpair.specific$selector),
             zzz = as.numeric(dpair.specific$decision))

set.seed(349)
samples <- stan(model_code=modelngwi,
                data=data,
                iter=2000,
                chains=4,
                seed = 349)

pcm_estimated_values <- summary(samples,
```

```
                                pars = c("sigmaA",
                                         "sigmaB",
                                         "sigmaG",
                                         "rhoAB",
                                         "rhoG",
                                         "beta"),
                                probs = c(.025, .975))
View(pcm_estimated_values$summary)

# no gen no int stan model
modelngni <- "
functions {
    real pcminteract(int x, real alpha, real beta, real gamma, vector delta) {
      vector[rows(delta) + 1] unsummed;
      vector[rows(delta) + 1] probs;
      unsummed = append_row(rep_vector(0.0, 1), alpha + beta + gamma - delta);
      probs = softmax(cumulative_sum(unsummed));
      return categorical_lpmf(x+1 | probs);
    }
  }
data {
  int<lower = 1> I;                 // # items
  int<lower = 1> A;                 // # actors (or partners)
  int<lower = 1> U;                 // # undirected pairs
  int<lower = 1> N;                 // # responses
  int<lower = 1> D;                 // # decisions
  int<lower = 1> B;                 // integer value for # distal regression parameters
  int<lower = 1, upper = A> aa[N];  // size N array to index actors for each response
  int<lower = 1, upper = A> pp[N];  // size N array to index partners for each response
  int<lower = 1, upper = I> ii[N];  // size N array to index items for each response
  int<lower = 0> x[N];              // size N array for responses; x = 0, 1 ... m_i
  int<lower = 1, upper = U> dd[N];  // size N array to index undirected pairs for each response
  int<lower = 1, upper = 2> mm[N];  // size N array to index match for each response
  int<lower = 1, upper = A> aaa[D]; // size D array to index actors for each decision
  int<lower = 1, upper = A> ppp[D]; // size D array to index partners for each decision
  int<lower = 1, upper = U> ddd[D]; // size D array to index undirected pairs for each decision
  int<lower = 1, upper = 2> mmm[D]; // size D array to index match for each decision
  int<lower = 0, upper = 1> zzz[D]; // size D array for decisions
}
transformed data {
  int M;                            // # parameters per item (same for all items)
    M = max(x);
}
```

```
parameters {
  vector[M] delta[I];               // length m vector for each item i
  vector[2] AB[A];                  // size 2 vector of alpha and beta for each person;
  vector[2] GG[U];                  // size 2 vector of gammas for each undirected pair;
  real<lower = 0> sigmaA;           // real sd of alpha
  real<lower = 0> sigmaB;           // real sd of beta
  real<lower = 0> sigmaG;           // real sd of gamma
  real<lower = -1, upper = 1> rhoAB; // real cor between alpha and beta (within person)
  real<lower = -1, upper = 1> rhoG;  // real cor between gammas (within pair)
  real beta[B];                     // B-dimensional array of real valued of beta
                                    // (distal regression parameters)
}
transformed parameters {
  cov_matrix[2] SigmaAB;            // 2x2 covariance matrix of alpha and beta
  cov_matrix[2] SigmaG;             // 2x2 covariance matrix of gammas
  SigmaAB[1, 1] = sigmaA^2;
  SigmaAB[2, 2] = sigmaB^2;
  SigmaAB[1, 2] = rhoAB * sigmaA * sigmaB;
  SigmaAB[2, 1] = rhoAB * sigmaA * sigmaB;
  SigmaG[1, 1] = sigmaG^2;
  SigmaG[2, 2] = sigmaG^2;
  SigmaG[1, 2] = rhoG * sigmaG^2;
  SigmaG[2, 1] = rhoG * sigmaG^2;
}
model {
  AB ~ multi_normal(rep_vector(0.0, 2), SigmaAB);
  GG ~ multi_normal(rep_vector(0.0, 2), SigmaG);
  for (n in 1:N){
    target += pcminteract(x[n], AB[aa[n],1], AB[pp[n],2], GG[dd[n], mm[n]], delta[ii[n]]);
  }
  for (d in 1:D){
    //distal logistic regression
    target += bernoulli_logit_lpmf(zzz[d] | (beta[1]
    + beta[2]*AB[aaa[d],1]
    + beta[3]*AB[ppp[d],1]
    + beta[4]*AB[aaa[d],2]
    + beta[5]*AB[ppp[d],2]
    + beta[6]*GG[ddd[d], mmm[d]]
    + beta[7]*GG[ddd[d], (3-mmm[d])]));
  }
}
"
```

```
# no gen with int model
I <- max(df.complete$item)
A <- max(df.complete$actor)
U <- max(df.complete$unique.pair)
N <- nrow(df.complete)
D <- nrow(dpair.specific)
B <- 7

data <- list(I = I,
             A = A,
             U = U,
             N = N,
             D = D,
             B = B,
             aa = as.numeric(df.complete$actor),
             pp = as.numeric(df.complete$partner),
             ii = as.numeric(df.complete$item),
             x = as.numeric(df.complete$x),
             dd = as.numeric(df.complete$unique.pair),
             mm = as.numeric(df.complete$selector),
             aaa = as.numeric(dpair.specific$actor),
             ppp = as.numeric(dpair.specific$partner),
             ddd = as.numeric(dpair.specific$unique.pair),
             mmm = as.numeric(dpair.specific$selector),
             zzz = as.numeric(dpair.specific$decision))

set.seed(349)
samples <- stan(model_code=modelngni,
                data=data,
                iter=2000,
                chains=4,
                seed = 349)

pcm_estimated_values <- summary(samples,
                                pars = c("sigmaA",
                                         "sigmaB",
                                         "sigmaG",
                                         "rhoAB",
                                         "rhoG",
                                         "beta",
                                         "delta"),
                                probs = c(.025, .975))
View(pcm_estimated_values$summary)
```

```
# with gen no int stan model
modelwgni <- "
functions {
    real pcminteract(int x, real alpha, real beta, real gamma, vector delta) {
      vector[rows(delta) + 1] unsummed;
      vector[rows(delta) + 1] probs;
      unsummed = append_row(rep_vector(0.0, 1), alpha + beta + gamma - delta);
      probs = softmax(cumulative_sum(unsummed));
      return categorical_lpmf(x+1 | probs);
    }
  }
data {
  int<lower = 1> I;                  // # items
  int<lower = 1> A;                  // # actors (or partners)
  int<lower = 1> U;                  // # undirected pairs
  int<lower = 1> N;                  // # responses
  int<lower = 1> D;                  // # decisions
  int<lower = 1> B;                  // integer value for # distal regression parameters
  int<lower = 1, upper = A> aa[N];   // size N array to index actors for each response
  int<lower = 1, upper = A> pp[N];   // size N array to index partners for each response
  int<lower = 1, upper = I> ii[N];   // size N array to index items for each response
  int<lower = 0> x[N];               // size N array for responses; x = 0, 1 ... m_i
  int<lower = 1, upper = U> dd[N];   // size N array to index undirected pairs for each response
  int<lower = 1, upper = 2> mm[N];   // size N array to index match for each response
  int<lower = 0, upper = 1> gg[N];   // size N array to index gender for each response
  int<lower = 1, upper = A> aaa[D];  // size D array to index actors for each decision
  int<lower = 1, upper = A> ppp[D];  // size D array to index partners for each decision
  int<lower = 1, upper = U> ddd[D];  // size D array to index undirected pairs for each decision
  int<lower = 1, upper = 2> mmm[D];  // size D array to index match for each decision
  int<lower = 0, upper = 1> zzz[D];  // size D array for decisions
}
transformed data {
  int M;                             // # parameters per item (same for all items)
    M = max(x);
}
parameters {
  vector[M] delta[I];                // length m vector for each item i
  vector[2] AB[A];                   // size 2 vector of alpha and beta for each person;
  vector[2] GG[U];                   // size 2 vector of gammas for each undirected pair;
  real<lower = 0> sigmaA;            // real sd of alpha
  real<lower = 0> sigmaB;            // real sd of beta
  real<lower = 0> sigmaG;            // real sd of gamma
```

```
  real<lower = -1, upper = 1> rhoAB; // real cor between alpha and beta (within person)
  real<lower = -1, upper = 1> rhoG;  // real cor between gammas (within pair)
  real mu;                           // real value of mean of theta for males
  real beta[B];                      // B-dimensional array of real valued of beta
                                     // (distal regression parameters)
}
transformed parameters {
  cov_matrix[2] SigmaAB;             // 2x2 covariance matrix of alpha and beta
  cov_matrix[2] SigmaG;              // 2x2 covariance matrix of gammas
  SigmaAB[1, 1] = sigmaA^2;
  SigmaAB[2, 2] = sigmaB^2;
  SigmaAB[1, 2] = rhoAB * sigmaA * sigmaB;
  SigmaAB[2, 1] = rhoAB * sigmaA * sigmaB;
  SigmaG[1, 1] = sigmaG^2;
  SigmaG[2, 2] = sigmaG^2;
  SigmaG[1, 2] = rhoG * sigmaG^2;
  SigmaG[2, 1] = rhoG * sigmaG^2;
}
model {
  AB ~ multi_normal(rep_vector(0.0, 2), SigmaAB);
  GG ~ multi_normal(rep_vector(0.0, 2), SigmaG);
  for (n in 1:N){
    target += pcminteract(x[n], AB[aa[n],1] - mu*gg[n], AB[pp[n],2], GG[dd[n], mm[n]], delta[ii[n]]);
  }
  for (d in 1:D){
    //distal logistic regression
    target += bernoulli_logit_lpmf(zzz[d] | (beta[1]
    + beta[2]*AB[aaa[d],1]
    + beta[3]*AB[ppp[d],1]
    + beta[4]*AB[aaa[d],2]
    + beta[5]*AB[ppp[d],2]
    + beta[6]*GG[ddd[d], mmm[d]]
    + beta[7]*GG[ddd[d], (3-mmm[d])]));
  }
}
"

# no gen with int model
I <- max(df.complete$item)
A <- max(df.complete$actor)
U <- max(df.complete$unique.pair)
N <- nrow(df.complete)
D <- nrow(dpair.specific)
```

```
B <- 7

data <- list(I = I,
             A = A,
             U = U,
             N = N,
             D = D,
             B = B,
             aa = as.numeric(df.complete$actor),
             pp = as.numeric(df.complete$partner),
             ii = as.numeric(df.complete$item),
             x = as.numeric(df.complete$x),
             dd = as.numeric(df.complete$unique.pair),
             mm = as.numeric(df.complete$selector),
             gg = as.numeric(df.complete$male),
             aaa = as.numeric(dpair.specific$actor),
             ppp = as.numeric(dpair.specific$partner),
             ddd = as.numeric(dpair.specific$unique.pair),
             mmm = as.numeric(dpair.specific$selector),
             zzz = as.numeric(dpair.specific$decision))

set.seed(349)
samples <- stan(model_code=modelwgni,
                data=data,
                iter=2000,
                chains=4,
                seed = 349)

pcm_estimated_values <- summary(samples,
                                pars = c("sigmaA",
                                         "sigmaB",
                                         "sigmaG",
                                         "rhoAB",
                                         "rhoG",
                                         "mu",
                                         "beta"),
                                probs = c(.025, .975))
View(pcm_estimated_values$summary)
```

Joint Estimation with Interactions in Distal Outcome Model

Using the joint MCMC estimation approach described in Section 3, the results of estimating the basic dyadic partial credit model (2) together with the distal regression in (6) without constraining $b_7 = b_8 = b_9 = 0$ are presented in Tables S1 and S2 below.

TABLE S1.

Estimates of Standard Deviations and Correlations of Individual and Dyadic Latent Traits (Joint Approach)

	Mean	MC Err	SD	95% CI	Eff N	$\hat{R}$
σ_α	1.03	0.0011	0.0368	(0.96,1.10)	1,168	1.0014
σ_β	0.63	0.0012	0.0262	(0.58,0.68)	517	1.0027
σ_γ	0.98	0.0011	0.0179	(0.95,1.02)	283	1.0108
$\rho_{\alpha\beta}$	-0.06	0.0020	0.0537	(-0.17,0.04)	725	1.0024
ρ_γ	0.46	0.0010	0.0233	(0.42,0.51)	544	1.0070

TABLE S2.

Estimates for Distal Outcome Regression (Joint Approach)

	Mean	MC Err	SD	95% CI	Eff N	$\hat{R}$
$b_0[1]$	-0.87	0.0040	0.0848	(-1.03,-0.71)	454	1.0075
$b_1[\alpha_a]$	0.15	0.0050	0.0978	(-0.04, 0.33)	378	1.0093
$b_2[\alpha_p]$	-0.02	0.0013	0.0566	(-0.13, 0.09)	1,978	1.0117
$b_3[\beta_a]$	-3.03	0.0166	0.2642	(-3.62,-2.58)	252	1.0133
$b_4[\beta_p]$	3.56	0.0106	0.2148	(3.17, 4.01)	413	1.0066
$b_5[\gamma_{a,p}]$	3.50	0.0144	0.2478	(3.06, 4.06)	295	1.0096
$b_6[\gamma_{p,a}]$	0.17	0.0038	0.0879	(0.00, 0.35)	523	1.0089
$b_7[\alpha_a\alpha_p]$	-0.01	0.0115	0.0557	(-0.13, 0.09)	1,351	1.0021
$b_8[\beta_a\beta_p]$	0.45	0.0084	0.2099	(0.06, 0.87)	627	1.0046
$b_9[\gamma_{a.p}\gamma_{p,a}]$	-0.28	0.0050	0.1292	(-0.53,-0.02)	667	1.0034

Sequential Estimation

Using the sequential estimation approach with multiple imputation described in Section 3, the results of first estimating the dyadic partial credit model (ignoring the distal outcome), and subsequently estimating the distal regression are presented in Tables S3 and S4. We report estimates as means of draws, and the values in parentheses represent the 2.5^{th} and the 97.5^{th} quantiles of the parameter estimates.

MCMC estimates for the standard deviations and correlations of the individual and dyadic latent traits are shown in Table S3. The estimates are qualitatively similar to the estimates from the joint approach reported in Table 1 of the main text.

Estimates for the distal regression based on multiple draws of the latent traits from their posterior distribution are shown in Table S4. While the sign of the coefficient estimates are the same as for the joint approach in Table 2 of the main text, their magnitudes differ substantially. Overall, the estimates using the sequential approach are smaller in absolute value, particularly for b_3, b_4 and b_5. This may be because the joint approach effectively treats the distal outcome as an item in the measurement model, and therefore makes the latent traits highly predictive of the distal outcome.

TABLE S3.

Sequential Estimation Approach: Estimates of Standard Deviations and Correlations of Individual and Dyadic Latent Traits

	without gender				with gender	
	with interactions		without interactions		without interactions	
μ_{male}					-0.15	(-0.36,0.06)
σ_α	1.05	(0.98,1.13)	1.05	(0.98,1.13)	1.05	(0.98,1.13)
σ_β	0.71	(0.66,0.76)	0.71	(0.66,0.76)	0.71	(0.66,0.76)
σ_γ	0.89	(0.86,0.92)	0.89	(0.86,0.92)	0.89	(0.86,0.91)
$\rho_{\alpha\beta}$	0.03	(-0.06,0.13)	0.03	(-0.06,0.13)	0.04	(-0.06,0.13)
ρ_γ	0.35	(0.30,0.40)	0.35	(0.30,0.40)	0.35	(0.30,0.40)

TABLE S4.

Sequential Estimation Approach: Estimates for Distal Outcome Regression

	without gender				with gender	
	with interactions		without interactions		without interactions	
b_0	-0.36	(-0.43,-0.29)	-0.36	(-0.43,-0.29)	-0.36	(-0.43,-0.28)
b_1	0.40	(0.36, 0.44)	0.40	(0.36, 0.44)	0.38	(0.34, 0.43)
b_2	-0.03	(-0.07, 0.00)	-0.03	(-0.07, 0.00)	-0.02	(-0.06, 0.02)
b_3	-0.35	(-0.44,-0.26)	-0.34	(-0.42,-0.25)	-0.32	(-0.42,-0.23)
b_4	1.29	(1.19, 1.38)	1.28	(1.19, 1.37)	1.26	(1.16, 1.36)
b_5	0.82	(0.76, 0.89)	0.82	(0.77, 0.89)	0.82	(0.76, 0.88)
b_6	0.06	(0.01, 0.11)	0.06	(0.01, 0.11)	0.06	(-0.01, 0.11)
b_7	-0.01	(-0.04, 0.01)				
b_8	0.18	(0.09, 0.28)				
b_9	-0.02	(-0.07, 0.04)				