Bayesian SEM and Interaction Effects

Zachary Roman M.S.[1]

[1] Center for Research Methods and Data Analysis and Psychology Department

2018

CENTER FOR RESEARCH METHODS & DATA ANALYSIS
College of Liberal Arts & Sciences

Outline

Outline

Interaction Terms

Many meaningful hypothesis can only be tested with interaction terms. When we hypothesize that the relationship between two variables is different because of a third we are referring to interaction effects.

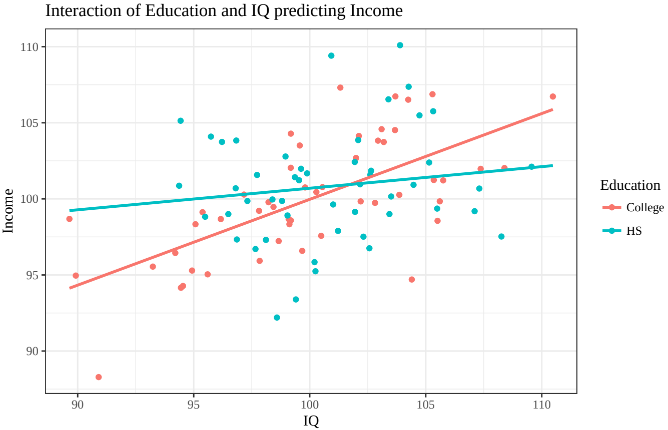
- IQ positively predicts education (in years) but the relationship varies by SES.
- GPA positively predicts salary, but the relationship differs by gender.
- Bench press weight positively predicts squat weight but the relationship differs by height.

Regression: Interaction Term

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \underbrace{\beta_3 x_1 \cdot x_2}_{\text{product term}} + \epsilon$$

Interaction Terms

- Categorical by Continuous
  - IQ predicts Income, but has a different relationship (slope) at varying levels of education



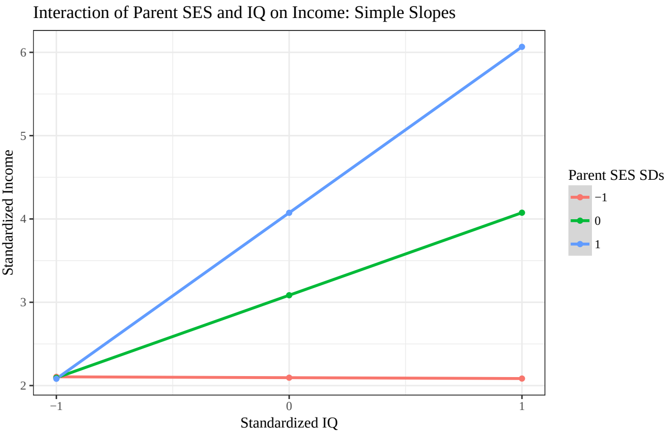
Interaction Terms

- Continuous by Continuous
  - Interpreting continuous by continuous interactions cannot be done with just a coefficient. We typically plot simple slopes to "probe" the relationship.

```
Call:
lm(formula = Income ~ IQ * PSES, data = dat)

Residuals:
    Min      1Q  Median      3Q     Max
-5.8539 -1.3270 -0.2993  1.3951  5.8935

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.084322   0.226037   13.64   <2e-16 ***
IQ          0.989595   0.021580   45.86   <2e-16 ***
PSES        0.990908   0.024113   41.09   <2e-16 ***
IQ:PSES     1.001278   0.002486  402.84   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.136 on 96 degrees of freedom
Multiple R-squared:  0.9996,  Adjusted R-squared:  0.9996
F-statistic: 7.373e+04 on 3 and 96 DF,  p-value: < 2.2e-16
```

Interaction Terms ...

- IQ predicts Income, but has differing slopes at high and low levels of parent SES



Outline

Kenny and Judd 1984

Kenny & Judd (1984) Hypothesized that voters positions (V) on an issue and her/his judgment of the candidates position (C) on that issue, should be "moderated" by the voters sentiment (S) of the candidate. In other words, they predict if people like a candidate, they perceive the candidates stance on an issue to be more similar to their own.

Kenny Judd 1984

Kenny / Judd utilized the 1968 National election survey conducted by the University of Michigan (Judd et al., 1983).

- Voters Position (2 Items): Likert 0 - 7
  1. How strong is your position on crime in the US?
  2. How much do you support the US inclusion in the Vietnam War?
- Candidate Position (2 Items): Likert 0 - 7
  1. How strong do you think the candidate's stance on crime is?
  2. How much do you think the candidate supports the US inclusion in the Vietnam War?
- Sentiment (2 Items) Thermometer 0 - 100
  1. How much do you like the candidate as a potential president?
  2. How much do you like the candidate in general?

See (Judd et al., 1983) for detailed explanation of the items and hypothesis.

Import the data

```r
dat<- read.table("data/data_kj.dat", header =
    TRUE)
head(dat)
```

```
     v1    v2    s1    s2    c1    c2
1 -0.74 -1.20  0.05 -1.75 -1.02  0.20
2  3.16  1.73 -1.64  0.75  0.34 -1.46
3 -0.05 -0.97 -3.10 -0.64  1.71  0.12
4  1.49 -0.87 -0.75  0.92 -0.75  0.28
5  0.44 -0.89 -0.21  0.27  0.40  0.48
6  0.11 -0.53  0.84  0.78  2.85  1.82
```

Descriptive Information

The data have been transformed to z-score (standardized) metric.

```
library(psych)
dat<- data.frame(scale(dat))
describe(dat,fast = T)
```

```
   vars    n mean sd   min  max range   se
v1    1 1160    0  1 -2.77 2.87  5.64 0.03
v2    2 1160    0  1 -3.52 3.09  6.61 0.03
s1    3 1160    0  1 -2.93 2.92  5.85 0.03
s2    4 1160    0  1 -3.41 3.81  7.22 0.03
c1    5 1160    0  1 -3.00 3.14  6.14 0.03
c2    6 1160    0  1 -3.28 3.81  7.09 0.03
```

Univariate Plots

I am reshaping the data from wide to long to assist in plotting.

```
library(ggplot2)
library(ggridges)
library(GGally)
library(reshape2)
dat_long <- melt(dat)
head(dat_long)
```

```
  variable      value
1       v1 -0.55257993
2       v1  2.22672276
3       v1 -0.06085715
4       v1  1.03661110
5       v1  0.28833730
6       v1  0.05316553
```

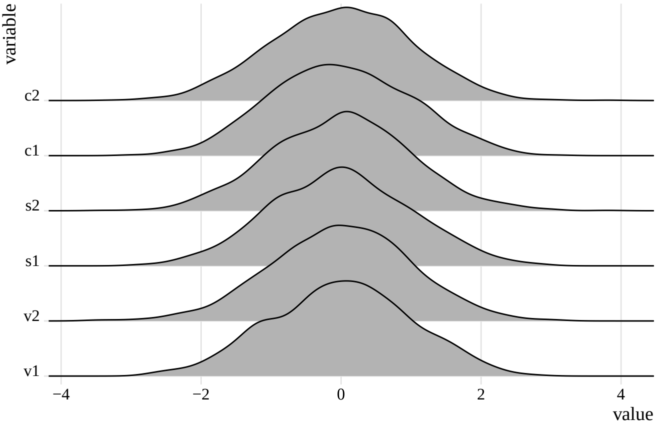
```
tail(dat_long)
```

Univariate Plots ...

```
     variable      value
6955       c2 -0.9926458
6956       c2 -1.4027629
6957       c2 -0.3120260
6958       c2  0.8659699
6959       c2 -1.2020673
6960       c2 -0.6348841
```

```
ggplot(data = dat_long, aes(y = variable, x =
    value)) +
   geom_density_ridges() + theme_ridges()+
   scale_y_discrete(expand = c(0.01, 0))+
   scale_x_continuous(expand = c(0, 0))
```

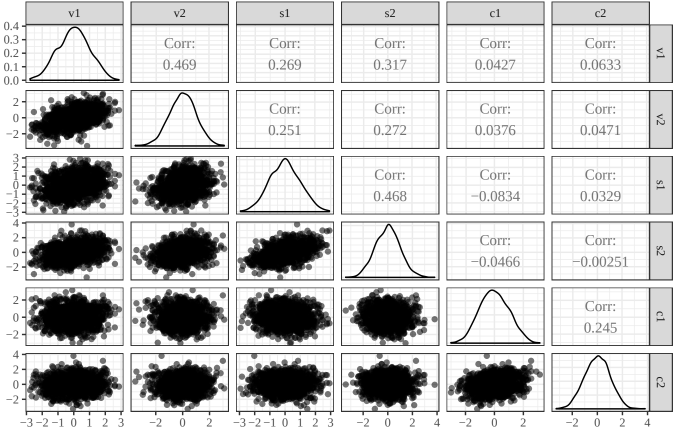
Univariate Plots ...



Bivariate plots

```
ggpairs(dat,aes(alpha = 0.9),upper =
    list(continuous = wrap("cor", color =
    "black"))) +
    theme_bw()
```

Bivariate plots ...



Outline

Traditional Methods: Brief Look [1]

- Product Indicators (Kenny and Judd 1984)
    - Multiply latent factor score for 2 variables to produce interaction factor
    - Constrain Factor loadings
    - Positively biased, inflated type 1 error rate, unrealistic assumptions
- Latent Moderated Structures (Klein & Moosbrugger, 2000)
    - Conditionally derived variance and covariances
    - Only assumes normality of the observed data
    - Still positively biased and inflated type 1 error rate
- Two Stage Method of Moments (Wall & Amemiya, 2003)
    - Two stage estimation, first factor model, then structural regressions
    - Extracts factor scores from first stage to create a regression model with product indicator
    - Less biased for polynomial effects, but interactions still see high type 1 error rates and parameter inflation

[1] see Brandt et al. (2014), for a detailed simulation investigating these methods.

Why Bayesian SEM?

- Today, we will employ a Bayesian method of estimating a latent interaction term in a structural model.
- With current software options Bayesian methods have the most flexibility we can specify a more realistic likelihood function.
- Frequentist methods don't allow the general user to manipulate the likelihood function, this results in unrealistic assumptions about the data.

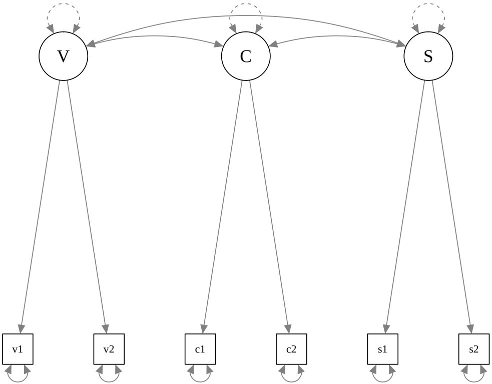
Outline

Measurement Model: CFA

We will now build the measurement model for the Kenny and Judd 1984 data

```r
library(lavaan)
library(semPlot)
syntax1 <- '
 V =~ v1 + v2
 C =~ c1 + c2
 S =~ s1 + s2'
mod1 <- cfa(model = syntax1,
             data = dat,
             std.lv = TRUE)
semPaths(mod1, layout = "tree2")
```

Measurement Model: CFA ...



Measurement Model: Summary

```
summary(mod1)
```

```
lavaan (0.5-23.1097) converged normally after  32 iterations

  Number of observations                          1160

  Estimator                                         ML
  Minimum Function Test Statistic                9.043
  Degrees of freedom                                 6
  P-value (Chi-square)                           0.171

Parameter Estimates:

  Information                                 Expected
  Standard Errors                             Standard

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  V =~
    v1                0.727    0.040   18.026    0.000
    v2                0.645    0.038   16.858    0.000
  C =~
    c1                0.787    0.265    2.969    0.003
    c2                0.311    0.108    2.876    0.004
```

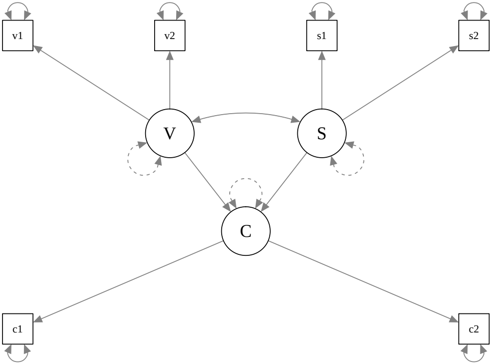
Measurement Model: Summary ...

```
  S =~
    s1              0.648    0.038   16.959    0.000
    s2              0.721    0.040   18.012    0.000

Covariances:
                 Estimate  Std.Err  z-value  P(>|z|)
  V ~~
    C               0.086    0.052    1.648    0.099
    S               0.591    0.038   15.370    0.000
  C ~~
    S              -0.104    0.055   -1.891    0.059

Variances:
                 Estimate  Std.Err  z-value  P(>|z|)
   .v1              0.471    0.050    9.460    0.000
   .v2              0.583    0.043   13.438    0.000
   .c1              0.380    0.416    0.915    0.360
   .c2              0.902    0.075   12.040    0.000
   .s1              0.579    0.043   13.315    0.000
   .s2              0.479    0.049    9.752    0.000
    V               1.000
    C               1.000
    S               1.000
```

Extension to SEM

```
syntax2 <- '
 V =~ v1 + v2
 C =~ c1 + c2
 S =~ s1 + s2
 C ~ V + S'
mod2 <- sem(model = syntax2,
             data = dat,
             std.lv = TRUE)
semPaths(mod2, layout = "tree")
```

Extension to SEM ...



SEM: Summary

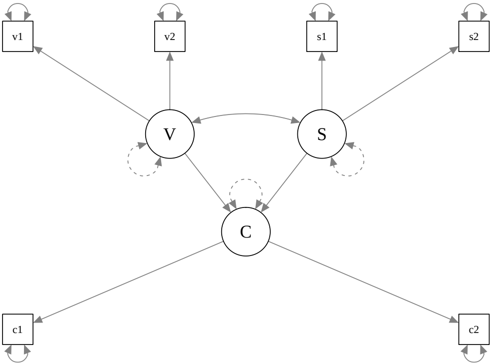
```
lavaan (0.5-23.1097) converged normally after  35 iterations

  Number of observations                          1160

  Estimator                                         ML
  Minimum Function Test Statistic                9.043
  Degrees of freedom                                 6
  P-value (Chi-square)                           0.171

Parameter Estimates:

  Information                                 Expected
  Standard Errors                             Standard

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  V =~
    v1                0.727    0.040   18.026    0.000
    v2                0.645    0.038   16.858    0.000
  C =~
    c1                0.769    0.270    2.853    0.004
    c2                0.304    0.102    2.984    0.003
  S =~
    s1                0.648    0.038   16.959    0.000
    s2                0.721    0.040   18.012    0.000
```

SEM: Summary ...

```
Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  C ~
    V                 0.232    0.101    2.293    0.022
    S                -0.243    0.104   -2.344    0.019

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  V ~~
    S                 0.591    0.038   15.369    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .v1                0.471    0.050    9.460    0.000
   .v2                0.583    0.043   13.438    0.000
   .c1                0.380    0.416    0.915    0.360
   .c2                0.902    0.075   12.040    0.000
   .s1                0.579    0.043   13.315    0.000
   .s2                0.479    0.049    9.752    0.000
    V                 1.000
   .C                 1.000
    S                 1.000
```

Outline

SEM: Interaction Term



SEM: Interaction Term ...

Goal: Create product term equivalent for latent variables

$$\xi_1 \cdot \xi_2$$

With our working example:

$$S \cdot V$$

Outline

Bayesian SEM: Kenny and Judd Data

Now we will conduct the *pre-processing* stage of the analysis

1. First we import the data and center the "predictors"

```
dat.kj <- read.table("data/data_kj.dat",
    header=T)
dat.kj$v1 <- dat.kj$v1-mean(dat.kj$v1)
dat.kj$v2 <- dat.kj$v2-mean(dat.kj$v2)
dat.kj$s1 <- dat.kj$s1-mean(dat.kj$s1)
dat.kj$s2 <- dat.kj$s2-mean(dat.kj$s2)
```

Bayesian SEM: Kenny and Judd Data ...

2. The next stage is to generate the appropriate dimensions of our input matrices and create a list of their elements

```
N <- nrow(dat.kj)
x <- cbind(dat.kj$v1,
           dat.kj$v2,
           dat.kj$s1,
           dat.kj$s2)
y <- cbind(dat.kj$c1,
           dat.kj$c2)
Kx <- ncol(x)
Ky <- ncol(y)
datstan <- list(N, Kx, Ky, y, x)
```

Bayesian SEM: Kenny and Judd Data ...

The following STAN code was adapted with permission from Brandt et al. (2014).

3. Now we will look at the STAN syntax file starting with the data stanza

```
data {
  int<lower=0> N;
  int<lower=0> Kx;
  int<lower=0> Ky;
  matrix[N,Kx] x;
  matrix[N,Ky] y;
}
```

Bayesian SEM: Kenny and Judd Data ...

The parameter stanza provides information that we wish to sample, it is our list of coefficients of interest

```
parameters {
  real       b0;                          // Int eta
  vector[5]  b1;                          // Reg coef
  vector<lower=0>[Kx] sigmax;   // Res var X
  vector<lower=0>[Ky] sigmay;   // Res var Y
  real<lower=0>      sigmaeta; // Res var Eta
  vector<lower=0>[2] sigmaxi;   // Var xi
  vector[N]   eta;                        // Lat var mat
  cholesky_factor_corr[2] L1;   // Cholesky speed
  matrix[N,2] zi;                         // Z mat for Chol
}
```

Bayesian SEM: Kenny and Judd Data ...

The transformed parameter stanza includes the measurement and structural specifications

```
transformed parameters {
  matrix[N,Kx] mux;       // E[x|xi]
  matrix[N,Ky] muy;       // E[y|eta]
  vector[N]   mueta;      // E[eta|xi]
  matrix[N,2] xi;         // Xi
  xi = zi*diag_pre_multiply(sigmaxi,L1)'; // Cholesky version of Xi
```

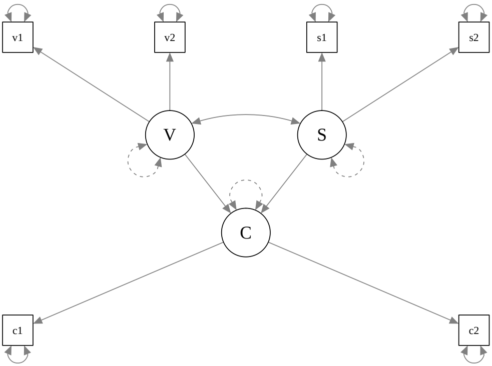
Bayesian SEM: Kenny and Judd Data ...

This is the measurement model denoting the factor loadings

```
  for (i in 1:N){
    mux[i,1] = xi[i,1]; // item 1 on F1
    mux[i,2] = xi[i,1]; // item 2 on F1
    mux[i,3] = xi[i,2]; // item 3 on F2
    mux[i,4] = xi[i,2]; // item 4 on F2
    muy[i,1] = eta[i];  // item 5 on Eta
    muy[i,2] = eta[i];  // item 6 on Eta
```

Bayesian SEM: Kenny and Judd Data ...

```
semPaths(mod2)
```



Bayesian SEM: Kenny and Judd Data ...

This is the structural portion of the model, specifying the latent regressions

```
    mueta[i] = b0
             +b1[1]*xi[i,1]
             +b1[2]*xi[i,2]
             +b1[3]*xi[i,1]*xi[i,1]
             +b1[4]*xi[i,1]*xi[i,2]
             +b1[5]*xi[i,2]*xi[i,2];
   }
}
```

Bayesian SEM: Kenny and Judd Data ...

Finally, we specify the likelihood and parameter priors in the model stanza

```
model {
  for(z in 1:4){x[,z] ~ normal(mux[,z],sigmax[z]);}
  for(z in 1:2){y[,z] ~ normal(muy[,z],sigmay[z]);}
  eta ~ normal(mueta,sigmaeta); //latent
  to_vector(zi) ~ normal(0,1);  //Cholesky
  b0 ~ normal(0,1);             //Reg coefs
  b1 ~ normal(0,1);
  sigmax   ~ cauchy(0,2.5);     //Prior SDs
  sigmay   ~ cauchy(0,2.5);
  sigmaeta ~ cauchy(0,2.5);
  sigmaxi  ~ cauchy(0,2.5);
  L1 ~ lkj_corr_cholesky(2);}   // Cholesky prior
```

Bayesian SEM: Kenny and Judd Data ...

The final stage is to deploy the model using Rstan. We use 2000 iterations disregarding half as warm up. We use 2 chains in this analysis.

```
rt1  <- stanc("STAN/sem1b.stan")
sm1  <- stan_model(stanc_ret = rt1, verbose=FALSE)
fit1 <- sampling(sm1, data=datstan,
                 chains = 2, itter = 2000,
                 warmup = 1000)
```

Convergence Estimates

Stan objects can be very large, and computationally time consuming. For today's purpose we will import an already fit stan model.

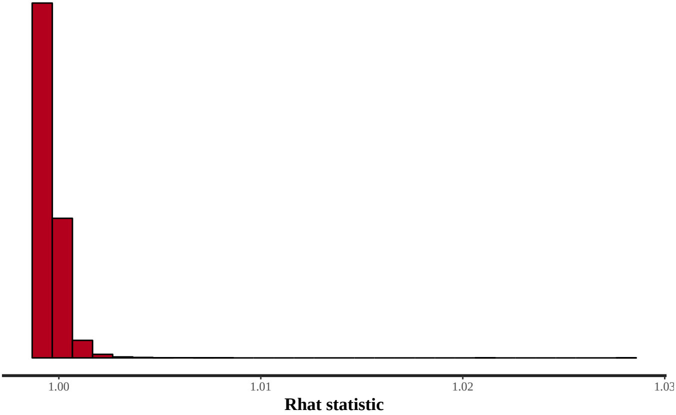
```
fit1 <- readRDS("stan/KJ_stan_model.rds")
names(fit1)[1:6]  <- c("Intercept",
                       "Voter",
                       "Sentiment",
                       "Voter^2",
                       "VoterXSent",
                       "Sentiment^2")
```

Convergence Estimates ...

First we must assess convergence of the Markov Chains. Rhat is a common statistic for assessing convergence, Rhat values Greater than 1.1 are typically thought to be non-converged.

```
rstan::stan_rhat(fit1)
```

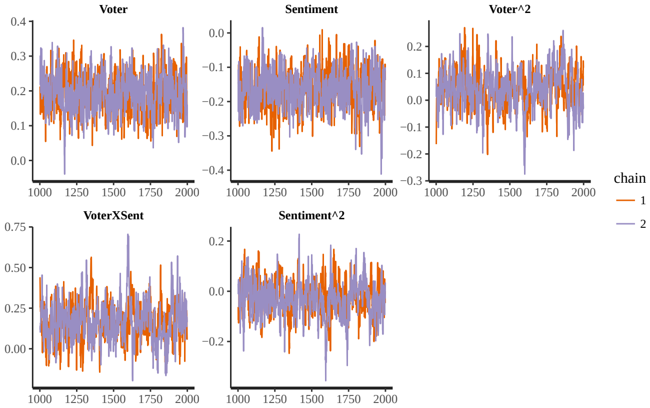
Convergence Estimates ...



Convergence Estimates ...

The mixing of the chains in the trace plots suggest good convergence

```
rstan::stan_trace(fit1,"b1")
```

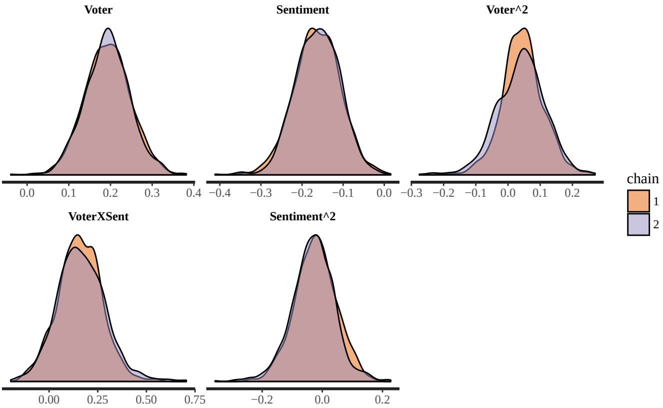


Convergence Estimates ...

Now that we have determined proper convergence of the solution we can assess the parameter estimates

```
rstan::stan_dens(fit1,"b1",separate_chains =T)
```

Convergence Estimates ...



Convergence Estimates ...

Here we can print the parameters of interest in the model

```
params <-
    c("b0","b1","sigmay","sigmax","sigmaeta","phi")
print(fit1,pars=params)
```

```
Inference for Stan model: sem1b.
2 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=2000.

             mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
Intercept    0.01    0.00 0.06 -0.10 -0.03  0.01  0.04  0.12   242 1.00
Voter        0.19    0.00 0.05  0.09  0.16  0.19  0.23  0.30   378 1.00
Sentiment   -0.16    0.00 0.05 -0.27 -0.20 -0.16 -0.12 -0.06   406 1.00
Voter^2      0.04    0.00 0.07 -0.10  0.00  0.04  0.09  0.18   213 1.00
VoterXSent   0.16    0.01 0.12 -0.06  0.08  0.16  0.24  0.40   195 1.00
Sentiment^2 -0.03    0.00 0.07 -0.17 -0.07 -0.02  0.02  0.11   196 1.01
sigmay[1]    1.00    0.00 0.03  0.95  0.98  1.00  1.02  1.06   719 1.00
sigmay[2]    1.00    0.00 0.03  0.95  0.98  1.00  1.02  1.06   519 1.00
sigmax[1]    1.01    0.00 0.03  0.95  0.99  1.01  1.03  1.08  1238 1.00
sigmax[2]    1.05    0.00 0.03  0.99  1.03  1.05  1.07  1.11  1309 1.00
sigmax[3]    1.04    0.00 0.03  0.98  1.02  1.04  1.06  1.10  1604 1.00
sigmax[4]    0.98    0.00 0.03  0.92  0.96  0.98  1.00  1.04  1274 1.01
sigmaeta     0.51    0.00 0.04  0.41  0.49  0.51  0.54  0.59    90 1.03
```

Convergence Estimates ...

```
phi[1,1]    0.93    0.00 0.07  0.80  0.88  0.92  0.97  1.06   780 1.00
phi[1,2]    0.54    0.00 0.05  0.45  0.51  0.54  0.57  0.63   740 1.01
phi[2,1]    0.54    0.00 0.05  0.45  0.51  0.54  0.57  0.63   740 1.01
phi[2,2]    0.88    0.00 0.06  0.75  0.84  0.88  0.92  1.01   564 1.01

Samples were drawn using NUTS(diag_e) at Thu May 17 12:50:44 2018.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

Convergence Estimates ...

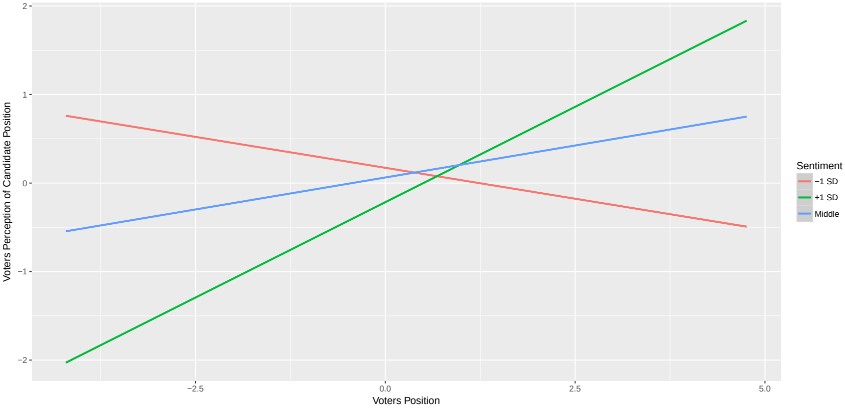


Figure: Latent Interaction: breaking down the complicated effect

??latent1)

Outline

Comparison of Methods

This table contains the estimates from a the methods we discussed today

Regression Path	K&J 1984	Mplus LMS	Lavaan	Bayesian
Voter	0.180	0.219	0.232	0.192
Sentiment	-0.111	-0.237	-0.243	-0.163
Voter^2	0.009	0.048	NA	0.042
Sentiment^2	-0.019	-0.032	NA	-0.031
Voter x Sentiment	0.207	0.193	NA	0.16

References

Brandt, H., Kelava, A., & Klein, A. (2014). A simulation study comparing recent approaches for the estimation of nonlinear effects in sem under the condition of nonnormality. *Structural equation modeling: a multidisciplinary journal*, 21(2), 181–195.

Judd, C. M., Kenny, D. A., & Krosnick, J. A. (1983). Judging the positions of political candidates: Models of assimilation and contrast. *Journal of Personality and Social Psychology*, 44(5), 952.

Kenny, D. A. & Judd, C. M. (1984). Estimating the nonlinear and interactive effects of latent variables. *Psychological bulletin*, 96(1), 201.

Klein, A. & Moosbrugger, H. (2000). Maximum likelihood estimation of latent interaction effects with the lms method. *Psychometrika*, 65(4), 457–474.

References ...

Wall, M. M. & Amemiya, Y. (2003). A method of moments technique for fitting interaction effects in structural equation models. *British Journal of Mathematical and Statistical Psychology*, 56(1), 47–63.

Session

```
sessionInfo()
```

```
R version 3.4.4 (2018-03-15)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 18.04 LTS

Matrix products: default
BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.7.1
LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.7.1

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
     LC_TIME=en_US.UTF-8
 [4] LC_COLLATE=en_US.UTF-8     LC_MONETARY=en_US.UTF-8
     LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C                  LC_ADDRESS=C
[10] LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8
    LC_IDENTIFICATION=C

attached base packages:
[1] stats     graphics  grDevices utils     datasets  base

other attached packages:
```

Session ...

```
[1] semPlot_1.1         lavaan_0.5-23.1097 reshape2_1.4.3
     GGally_1.4.0        ggridges_0.5.0
[6] psych_1.7.8         ggplot2_2.2.1      MASS_7.3-49
     stationery_0.80

loaded via a namespace (and not attached):
 [1] nlme_3.1-137         pbkrtest_0.4-7       RColorBrewer_1.1-2
      rstan_2.17.3
 [5] rprojroot_1.3-2      mi_1.0               tools_3.4.4
      backports_1.1.2
 [9] rpart_4.1-13         d3Network_0.5.2.1    Hmisc_4.1-1
      lazyeval_0.2.1
[13] mgcv_1.8-23          colorspace_1.3-2     nnet_7.3-12
     gridExtra_2.3
[17] mnormt_1.5-5         compiler_3.4.4       qgraph_1.4.4
     fdrtool_1.2.15
[21] quantreg_5.35        htmlTable_1.11.2     SparseM_1.77
     network_1.13.0
[25] labeling_0.3         scales_0.5.0         checkmate_1.8.5
     quadprog_1.5-5
[29] sem_3.1-9            StanHeaders_2.17.2   stringr_1.2.0
     digest_0.6.15
[33] pbivnorm_0.6.0       foreign_0.8-69       minqa_1.2.4
     rmarkdown_1.8
[37] base64enc_0.1-3      jpeg_0.1-8           pkgconfig_2.0.1
     htmltools_0.3.6
```

Session ...

```
[41] lme4_1.1-17          lisrelToR_0.1.4      htmlwidgets_1.0
     rlang_0.1.6
[45] rstudioapi_0.7       huge_1.2.7           gtools_3.5.0
     statnet.common_4.0.0
[49] acepack_1.4.1        car_2.1-6            inline_0.3.14
     magrittr_1.5
[53] OpenMx_2.8.3         Formula_1.2-2        Matrix_1.2-14
     Rcpp_0.12.15
[57] munsell_0.4.3        abind_1.4-5          rockchalk_1.8.111
     whisker_0.3-2
[61] stringi_1.2.2        plyr_1.8.4           matrixcalc_1.0-3
     grid_3.4.4
[65] parallel_3.4.4       methods_3.4.4        lattice_0.20-35
     kutils_1.45
[69] splines_3.4.4        pander_0.6.1         sna_2.4
     knitr_1.19
[73] pillar_1.1.0         igraph_1.1.2         boot_1.3-20
     rjson_0.2.15
[77] corpcor_1.6.9        BDgraph_2.44         stats4_3.4.4
     XML_3.98-1.9
[81] evaluate_0.10.1      latticeExtra_0.6-28  data.table_1.10.4-3
     png_0.1-7
[85] nloptr_1.0.4         MatrixModels_0.4-1   gtable_0.2.0
     reshape_0.8.7
[89] openxlsx_4.0.17      xtable_1.8-2         semTools_0.4-14
     coda_0.19-1
```

Session ...

```
[93] survival_2.41-3       glasso_1.8           tibble_1.4.2
    arm_1.9-3
[97] ggm_2.3               ellipse_0.4.1        cluster_2.0.6
```