Variance Reduction in the Robbins-Monro procedure in RSiena

Tom A.B. Snijders

University of Oxford
University of Groningen

UK-SNA 2013

Robbins-Monro procedure

The Robbins-Monro procedure, proposed originally in 1951
by Herbert Robbins and Sutton Monro,
is a procedure to solve equations of the kind

$$f(\theta) = 0$$

for functions $f(\theta)$ that cannot be calculated,
but that can be stochastically simulated with error;
for example, the median lethal dose of a poison.

Robbins-Monro procedure

The Robbins-Monro procedure, proposed originally in 1951
by Herbert Robbins and Sutton Monro,
is a procedure to solve equations of the kind

$$f(\theta) = 0$$

for functions $f(\theta)$ that cannot be calculated,
but that can be stochastically simulated with error;
for example, the median lethal dose of a poison.

That is, we can simulate random variables $X(\theta)$ for which

$$E\{X(\theta)\} = f(\theta) ;$$

and we wish to solve the equation

$$E\{X(\theta)\} = 0 .$$

The Robbins Monro procedure has been
much further developed since 1951;
it is the workhorse in the RSiena package
for computing estimates in stochastic actor-oriented models
according to the *Method of Moments*.

The Robbins Monro procedure has been
much further developed since 1951;
it is the workhorse in the RSiena package
for computing estimates in stochastic actor-oriented models
according to the *Method of Moments*.

To define it, denote the data (observed networks etc.) by x
and assume that x is the outcome of the random process X.

Denote the parameter of the probability model by

$$\theta = (\theta_1, \theta_2, \ldots, \theta_K) \; .$$

Method of Moment Estimation

For each θ_k we determine a statistic $z_k(x)$,
for which the distribution of $z_k(X)$ reflects the value of θ_k;
this means that if the value of θ_k gets higher,
then the outcomes of $z_k(X)$ tend to show higher values.

These are arranged in the vector $z(x) = \big(z_1(x), z_2(x), \ldots, z_K(x)\big)$.

The parameter estimate $\hat{\theta}$
is defined as the solution of the equation

$$E_{\hat{\theta}}\big\{z(X)\big\} \;=\; z(x) \qquad \textit{'expected = observed'}$$

Method of Moment Estimation

For each θ_k we determine a statistic $z_k(x)$,
for which the distribution of $z_k(X)$ reflects the value of θ_k;
this means that if the value of θ_k gets higher,
then the outcomes of $z_k(X)$ tend to show higher values.

These are arranged in the vector $z(x) = \big(z_1(x), z_2(x), \ldots, z_K(x)\big)$.

The parameter estimate $\hat{\theta}$
is defined as the solution of the equation

$$E_{\hat{\theta}}\big\{z(X)\big\} \;=\; z(x) \qquad \textit{'expected = observed'}$$

So the function $f(\theta)$ mentioned above is

$$f(\theta) \;=\; E_{\theta}\big\{z(X)\big\} - z(x)\;.$$

The Robbins-Monro procedure is an iterative algorithm:

if the current value of θ is $\hat{\theta}^{(N)}$,
we simulate the random process

$$X^{(N)} \sim \text{ model corresponding to } \hat{\theta}^{(N)}$$

and we update the parameter

$$\hat{\theta}^{(N+1)} = \hat{\theta}^{(N)} - a_N D^{-1} \left(z(X^{(N)}) - z(x) \right) .$$

a_N is a sequence with $a_N \downarrow 0$,
D is a matrix indicating the sensitivity of $E_\theta \{ z(X) \}$ to θ.

'If the simulated $z_k(X^{(N)})$ is too large (small),
decrease (increase) θ_k.'

So the idea is, for solving

$$f(\theta) = 0 \,,$$

to replace the function $f(\theta)$
by a random variable that has $f(\theta)$ as expected value:
we use a *Monte Carlo simulation method* to approximate $f(\theta)$.

This will go better
when the variance of the random variable is smaller.

So the idea is, for solving

$$f(\theta) = 0 \,,$$

to replace the function $f(\theta)$
by a random variable that has $f(\theta)$ as expected value:
we use a *Monte Carlo simulation method* to approximate $f(\theta)$.

This will go better
when the variance of the random variable is smaller.

The art of computer simulation knows a large variety of methods
to improve the efficiency of the simulation process
— i.e., work with a smaller error variance.

These were once known affectionately as swindles.

Swindle: regression method

Swindle: regression method

A useful swindle is the *regression method*:

When estimating an expected value $E\big(z_k(X)\big)$ by simulation,
if you can find a random variable U_k, correlated with $z_k(X)$,
and for which $E\big(U_k\big) = 0$,
then calculate the regression coefficient β_k of $z_k(X)$ on U_k
and subtract the prediction of $z_k(X)$ based on U_k:

$$E\big\{z_k(X) - \beta_k U_k\big\} = E\big\{z_k(X)\big\} = f_k(\theta)$$

so this does not affect the estimated value
and decreases the variance.

In statistical modeling,

a well-known function with expected value 0

is the *score function* with coordinates

$$J_k(x, \theta) = \frac{\partial}{\partial \theta_k} \log \left(p_\theta(x) \right) ,$$

where $p_\theta(x)$ is the probability (density) function of X

and θ_k is one of the coordinates of θ.

In statistical modeling,
a well-known function with expected value 0
is the *score function* with coordinates

$$J_k(x, \theta) = \frac{\partial}{\partial \theta_k} \log \left(p_\theta(x) \right) ,$$

where $p_\theta(x)$ is the probability (density) function of X
and θ_k is one of the coordinates of θ.

For the stochastic actor-oriented model,
the score function is too complicated to be computed.

In statistical modeling,
a well-known function with expected value 0
is the *score function* with coordinates

$$J_k(x, \theta) = \frac{\partial}{\partial \theta_k} \log \left(p_\theta(x) \right) ,$$

where $p_\theta(x)$ is the probability (density) function of X
and θ_k is one of the coordinates of θ.

For the stochastic actor-oriented model,
the score function is too complicated to be computed.

However, in RSiena we do calculate the score function for the *augmented data*, i.e., the data including all the ministeps.

The ministeps cannot be observed, but this does not matter –
they are simulated.

'Dolby' noise reduction

Denote by $\widetilde{X}$ the augmented data (i.e., including the ministeps) and by

$$J_k(\widetilde{X}, \theta)$$

the score function of the augmented data w.r.t. θ_k .

Then the modified Robbins-Monro method has update step

$$\hat{\theta}^{(N+1)} = \hat{\theta}^{(N)} - a_N D^{-1} \left(z(X^{(N)}) - \beta J(\widetilde{X}^{(N)}, \hat{\theta}^{(N)}) - z(x) \right)$$

where $\beta = (\beta_1, \beta_2, \ldots, \beta_K)$ and β_k is an estimate
for the regression coefficient of $z_k(X)$ on $J_k(\widetilde{X}, \theta)$.

The variance of this update is smaller,
which should make the algorithm more stable.

Implementation in RSiena

The correlation between $z_k(X)$ and $J_k(\widetilde{X}, \theta)$ is relatively high (because the $z_k(X)$ are indeed good statistics for the MoM).

Therefore, only the univariate regressions of $z_k(X)$ on $J_k(\widetilde{X}, \theta)$ (same k) are used.

The following steps are added to the phases of the Siena algorithm:

Phase 1: Calculate $\beta_1, \beta_2, \ldots, \beta_K$ as the regression coefficients in the sample of Phase 1 (for initial value of θ).

Phase 2: Use the modified update steps in each step of Phase 2.

Phase 3: Recalculate $\beta_1, \beta_2, \ldots, \beta_K$ in the larger sample of Phase 3, for the final value of θ,
for possible use in a next estimation run ('prevAns').

Another alteration: estimation of D

We study this together with another alteration to the algorithm, an improved matrix D.

Recall the Robbins-Monro update :

$$\hat{\theta}^{(N+1)} = \hat{\theta}^{(N)} - a_N D^{-1} \left(z(X^{(N)}) - z(x) \right) .$$

Another alteration: estimation of D

We study this together with another alteration to the algorithm, an improved matrix D.

Recall the Robbins-Monro update or the modified update:

$$\hat{\theta}^{(N+1)} = \hat{\theta}^{(N)} - a_N D^{-1} \left(z(X^{(N)}) - \beta J(\widetilde{X}^{(N)}, \hat{\theta}^{(N)}) - z(x) \right)$$

Another alteration: estimation of D

We study this together with another alteration to the algorithm, an improved matrix D.

Recall the Robbins-Monro update

$$\hat{\theta}^{(N+1)} = \hat{\theta}^{(N)} - a_N D^{-1} \left(z(X^{(N)}) - \beta J(\tilde{X}^{(N)}, \hat{\theta}^{(N)}) - z(x) \right)$$

where D is a matrix approximating

$$\frac{\partial E_\theta \{ z(X) \}}{\partial \theta} .$$

Another alteration: estimation of D

We study this together with another alteration to the algorithm, an improved matrix D.

Recall the Robbins-Monro update

$$\hat{\theta}^{(N+1)} = \hat{\theta}^{(N)} - a_N D^{-1} \Big(z(X^{(N)}) - \beta J(\widetilde{X}^{(N)}, \hat{\theta}^{(N)}) - z(x) \Big)$$

where D is a matrix approximating

$$\frac{\partial E_\theta \{ z(X) \}}{\partial \theta} .$$

Asymptotically, by estimating θ not by the last $\hat{\theta}^{(N)}$ but by a "tail average" of this sequence, a wide range of D will yield good results.

The current algorithm uses a Monte Carlo estimate $\widehat{D}_{(1)}$ of

$$\frac{\partial E_\theta\{z(X)\}}{\partial \theta}$$

calculated in Phase 1 of the algorithm for the initial value of θ.

To achieve stability of the algorithm,

D is then calculated as the diagonal matrix of this matrix of derivatives:

$$D = \operatorname{diag} \widehat{D}_{(1)} .$$

Diagonalizing sacrifices some efficiency for stability.

The second alteration to the algorithm is

using only a partial diagonalization:

$$D = \frac{1}{2} \widehat{D}_{(1)} + \frac{1}{2} \operatorname{diag} \widehat{D}_{(1)} .$$

Simulation study

To investigate the properties of these variance reduction techniques, a simulation study was made:

- ⇒ Only network dynamics, with one covariate.
- ⇒ Only 2 waves.
- ⇒ n (number of actors): 30 and 100.
- ⇒ A simple and a more complex model specification.
- ⇒ Start with a random network; then simulate the model with rate parameter = 20; collect wave 1; then simulate the model again, and collect wave 2.
- ⇒ Estimation under the correct model specification.
- ⇒ Repeated estimations, using 'prevAns', until convergence is good as indicated by a maximal t-ratio for convergence of 0.10.

So we have 4 studies ($n = 30, 100$; 2 model specifications);

So we have 4 studies ($n = 30, 100$; 2 model specifications);

each study has a 2×2 design:
Dolby yes/no $\times$ Half-diagonalize yes/no.

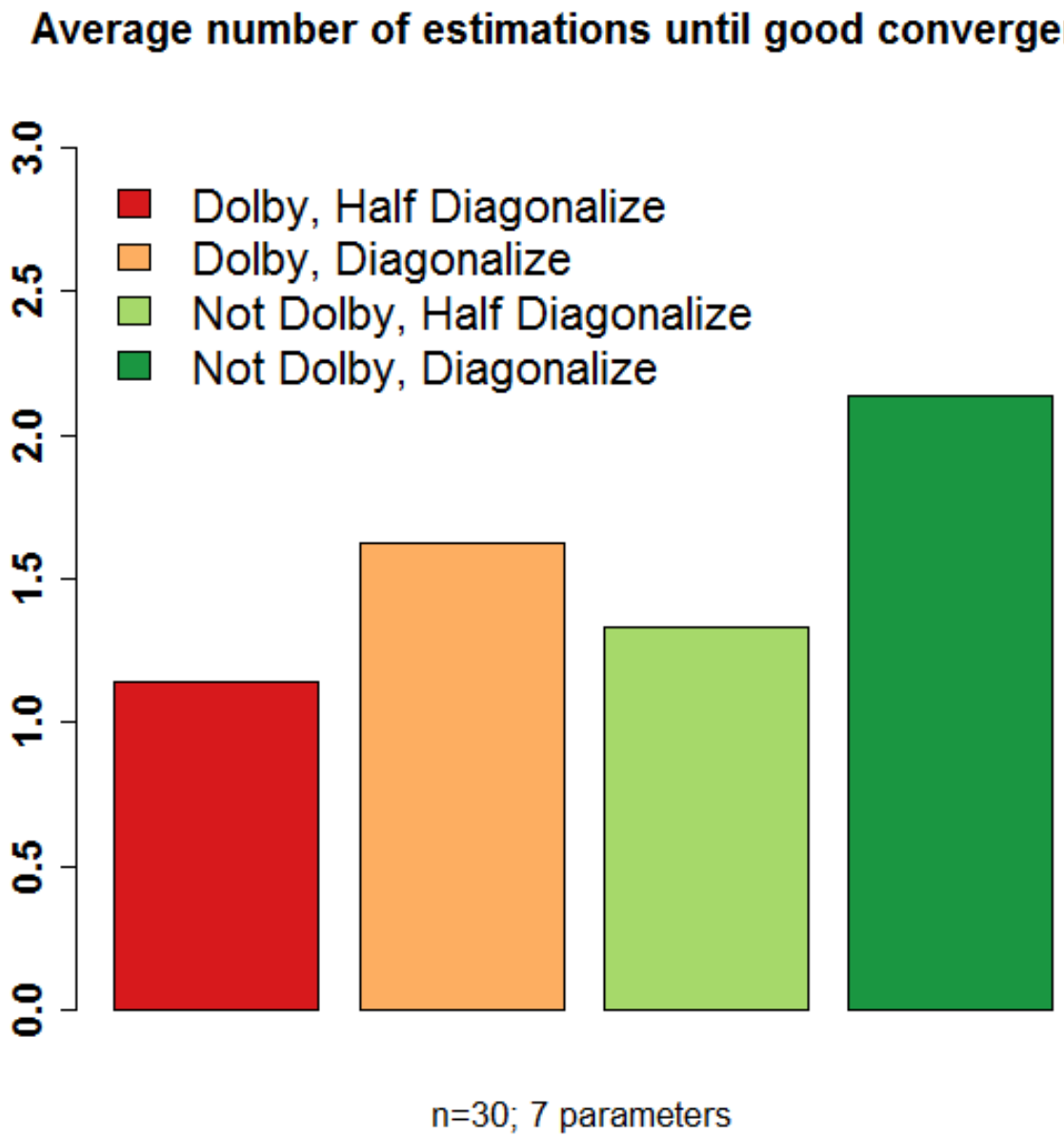
For each of these 16 combined specifications
we make 750 – 1,000 estimation runs for simulated data sets.

All models use an actor covariate V,
distributed between –2 and +2, mean 0.

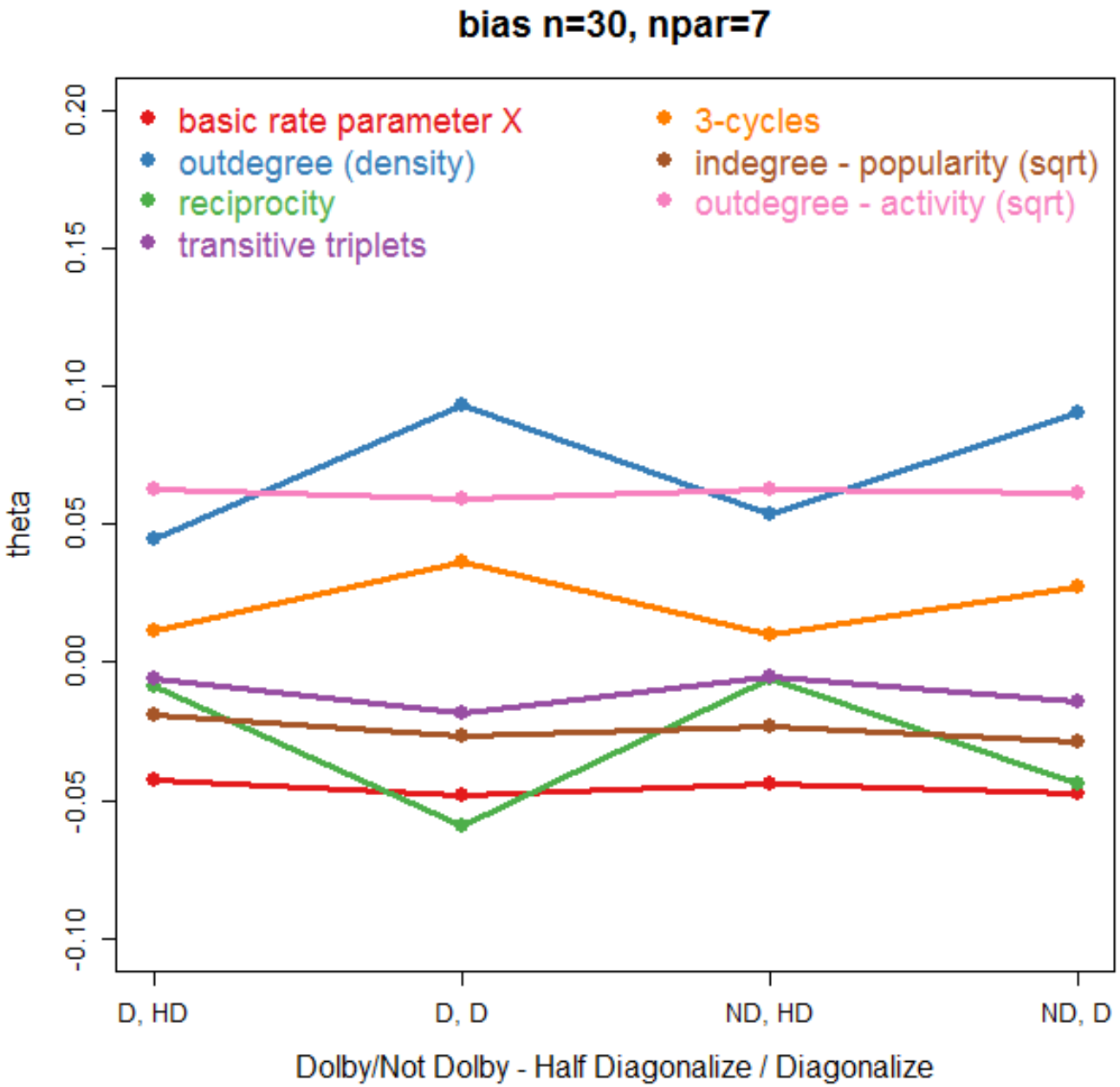
Study 1: $n = 30$ actors, model specification:

1. basic rate parameter	5.0
2. outdegree	–1.4
3. reciprocity	2.0
4. transitive triplets	0.4
5. 3-cycles	–0.4
6. indegree - popularity (sqrt)	–0.2
7. V similarity	0.6

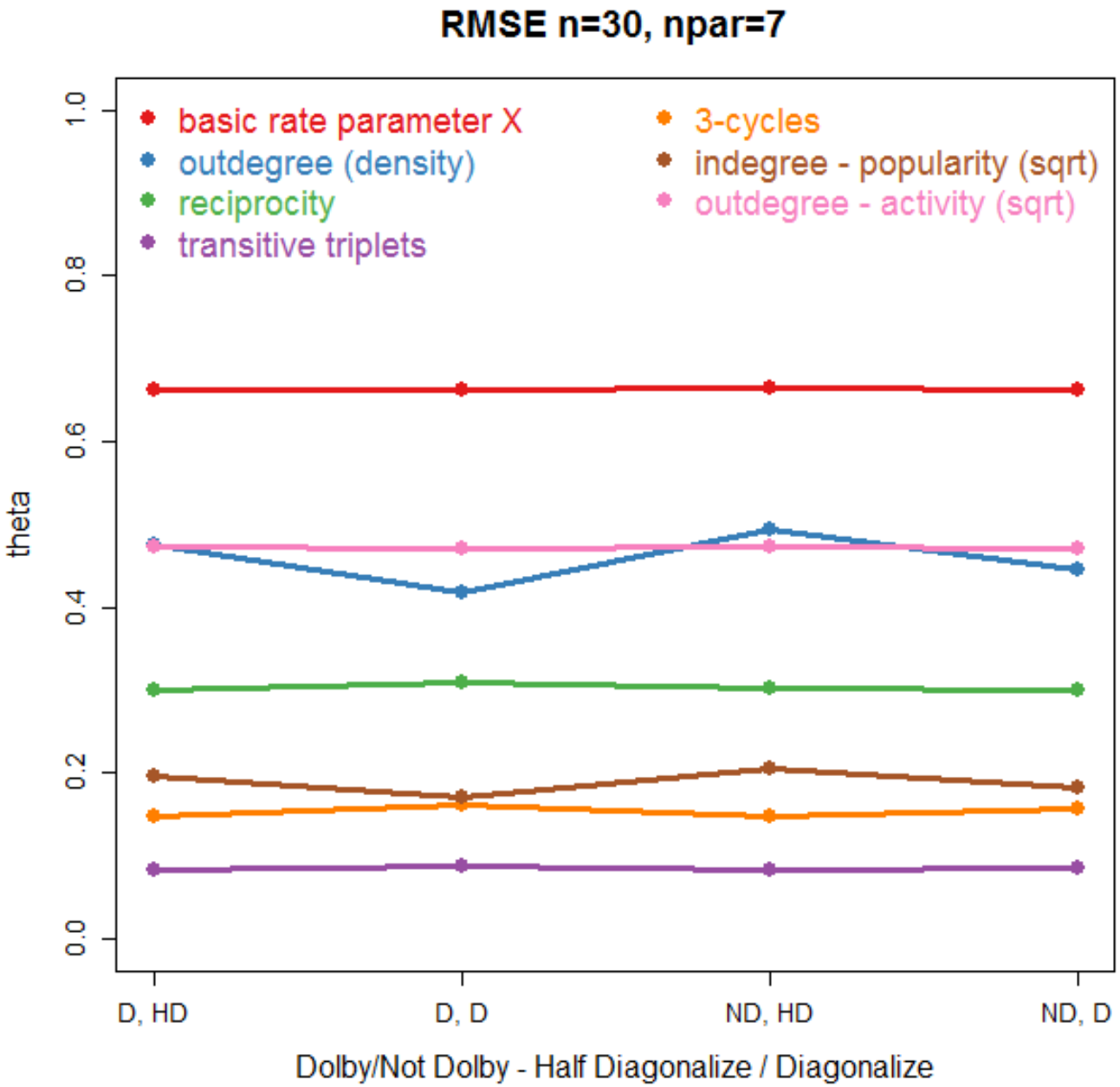
Average degrees 5.8 (wave 1) and 6.7 (wave 2).



When using Dolby together with Half Diagonalization, average number of estimation runs required is halved.



Bias is smaller for Half Diagonalization.



RMSE =

Root

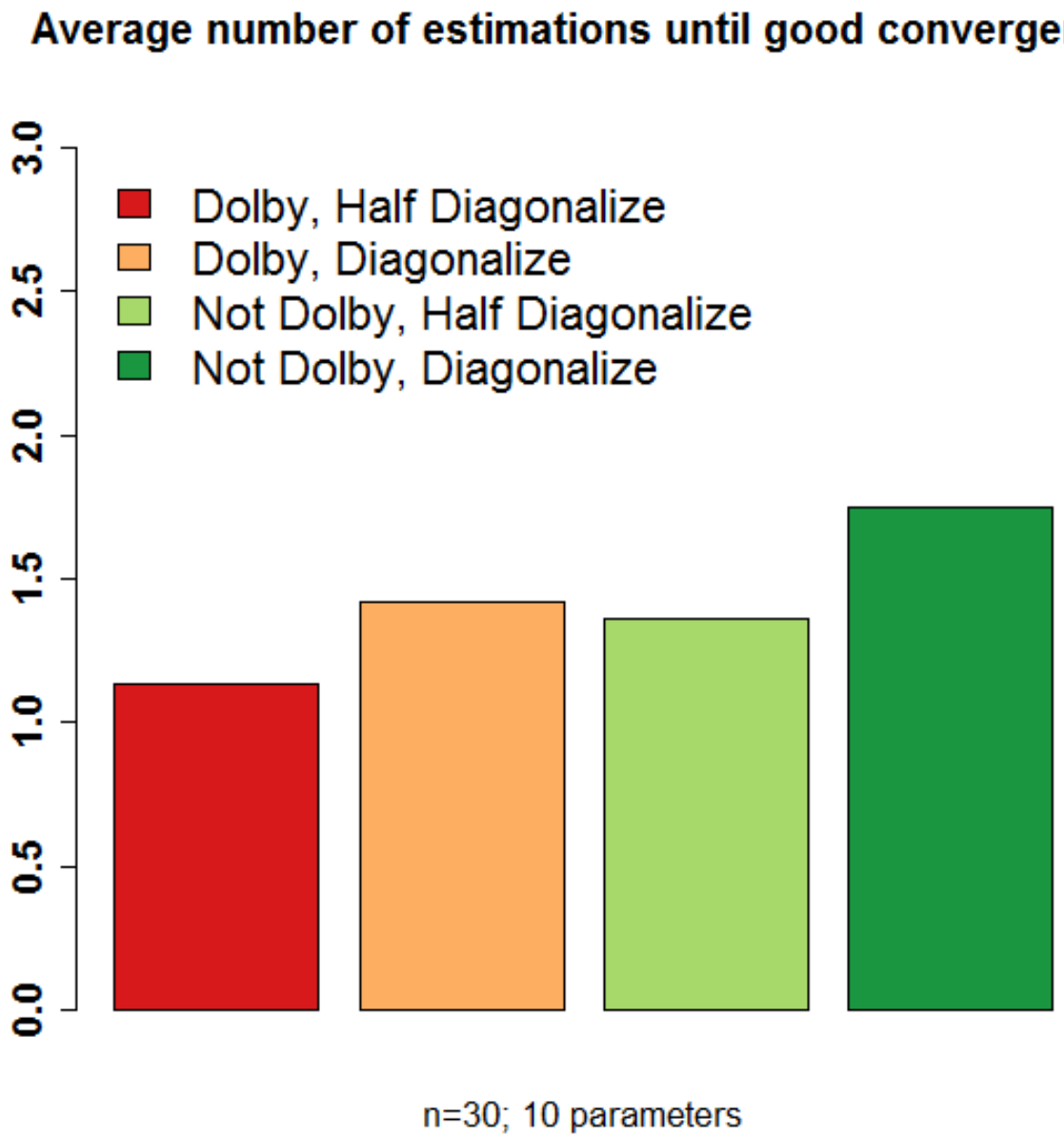
Mean Squared Error

Half Diagonalization gives a small increase in RMSE for outdegree and indegree-pop. parameters.

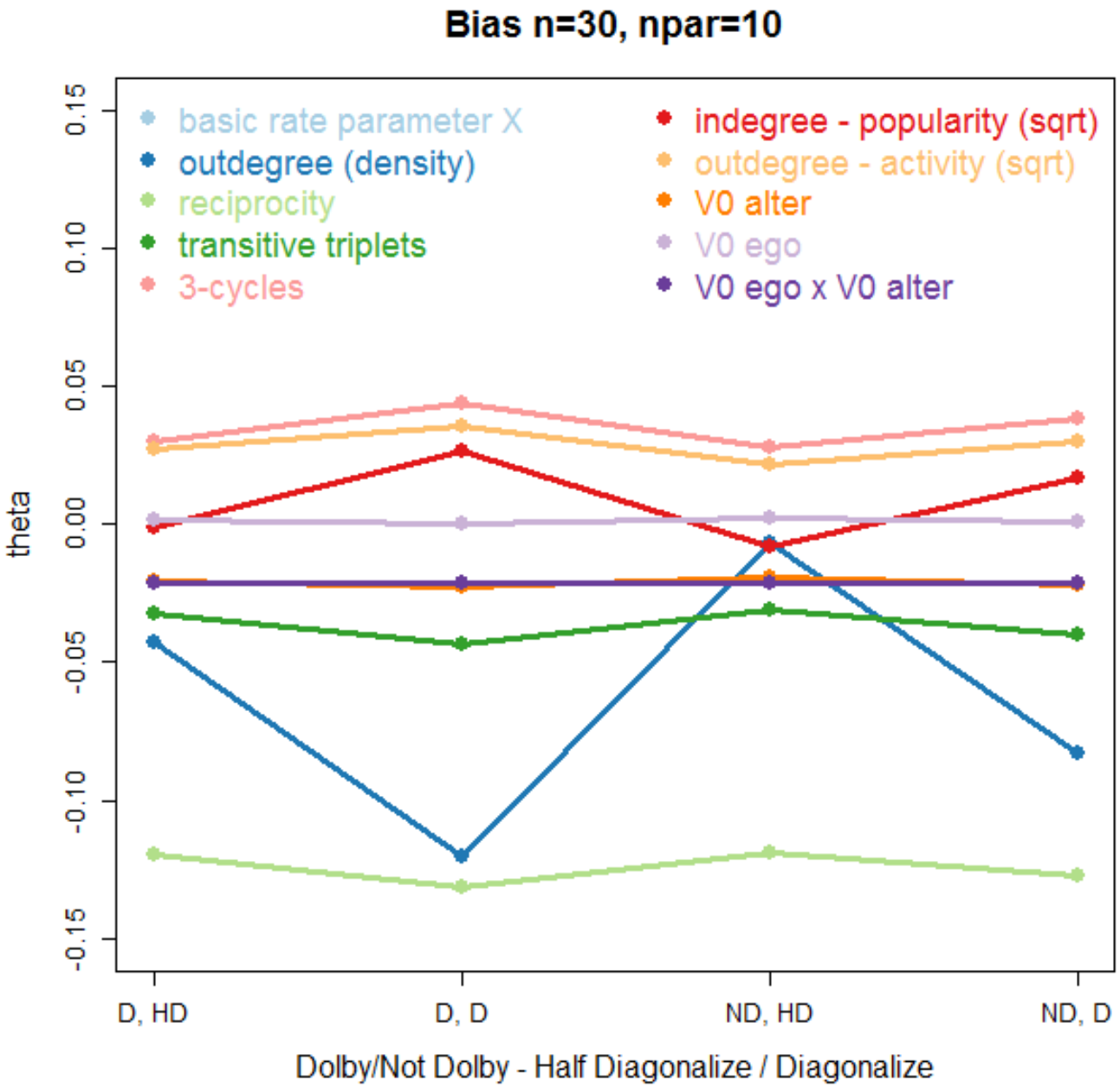
Study 2: $n = 30$ actors, model specification:

1.	basic rate parameter	5.0
2.	outdegree	–0.8
3.	reciprocity	2.0
4.	transitive triplets	0.3
5.	3-cycles	–0.35
6.	indegree - popularity (sqrt)	–0.2
7.	outdegree - activity (sqrt)	–0.1
8.	V alter	0.2
9.	V ego	0.0
10.	V ego $\times$ V alter	0.2

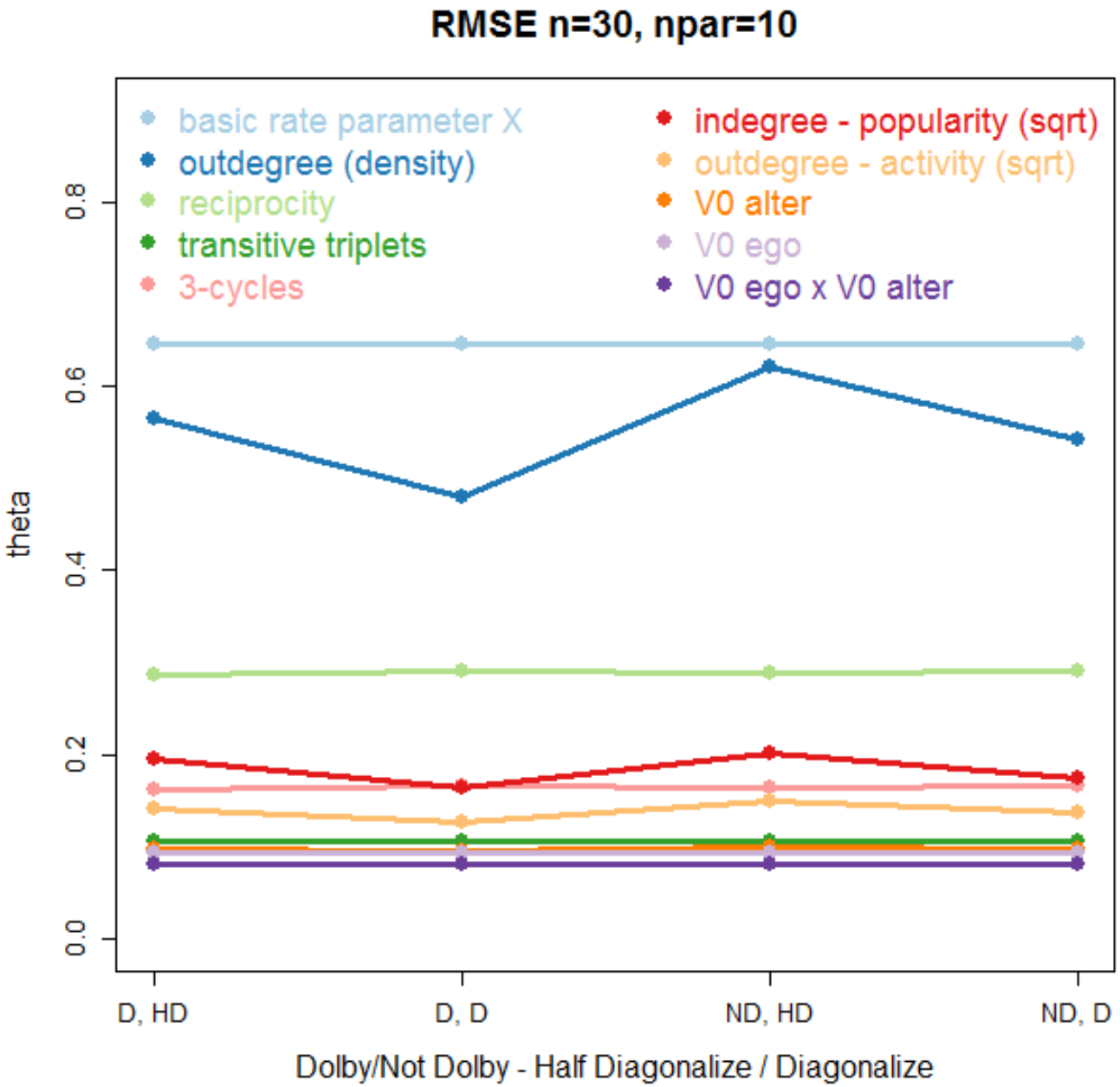
Average degrees 5.0 (wave 1) and 5.5 (wave 2).



When using Dolby together with Half Diagonalization, average number of estimation runs required is decreased by factor 1.5.



Bias is smaller for Half Diagonalization.



RMSE =

Root

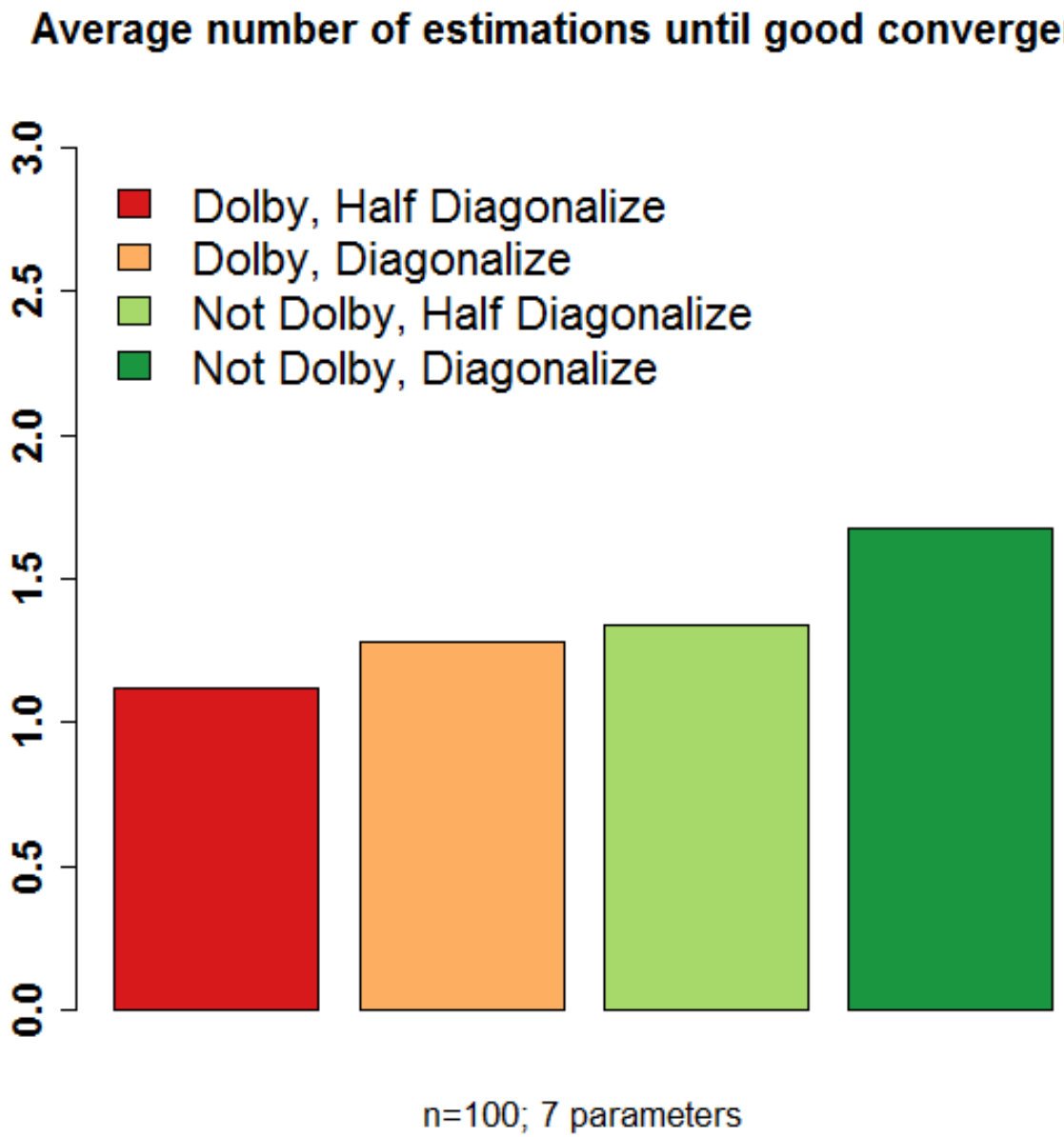
Mean Squared Error

Half diagonalization gives an increase in RMSE for outdegree and indegree-pop. parameters.

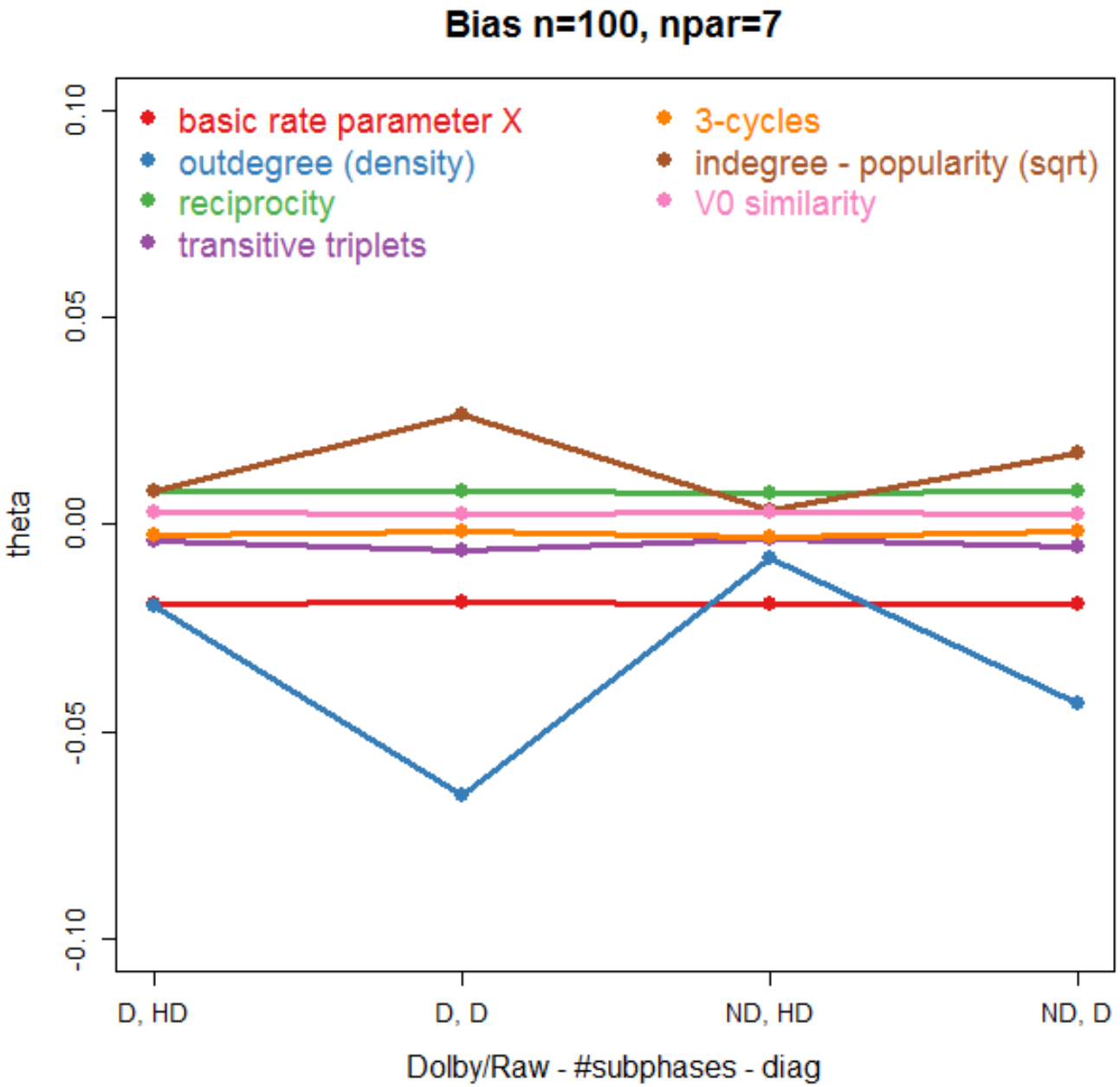
Study 3: $n = 100$ actors, model specification:

1. basic rate parameter	5.0
2. outdegree	–1.5
3. reciprocity	2.0
4. transitive triplets	0.4
5. 3-cycles	–0.4
6. indegree - popularity (sqrt)	–0.2
7. *V* similarity	0.6

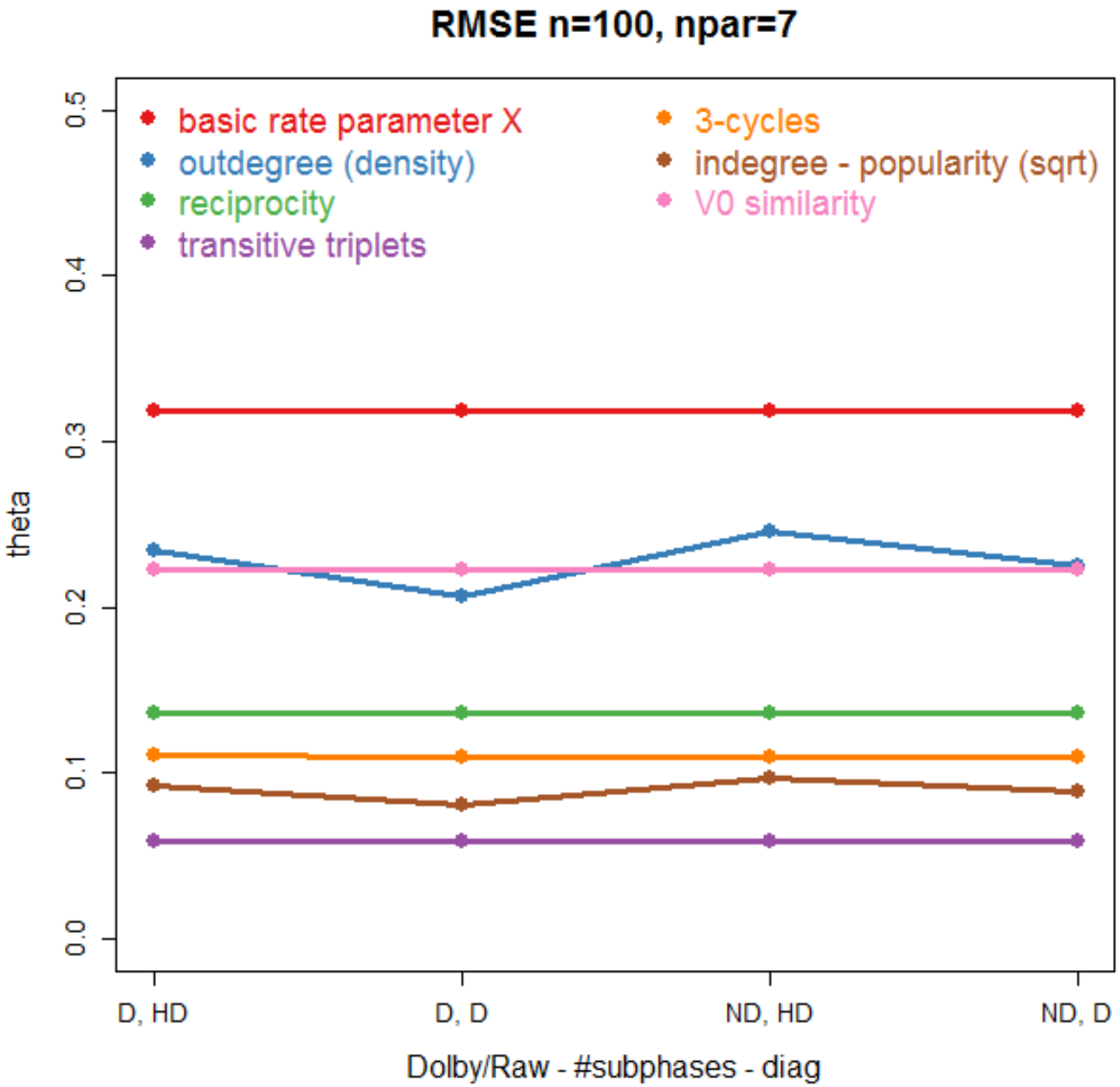
Average degrees 5.7 (wave 1) and 6.0 (wave 2).



When using Dolby together with Half Diagonalization, average number of estimation runs required is reduced by factor 1.5.



Bias is smaller for Half Diagonalization.



RMSE =

Root

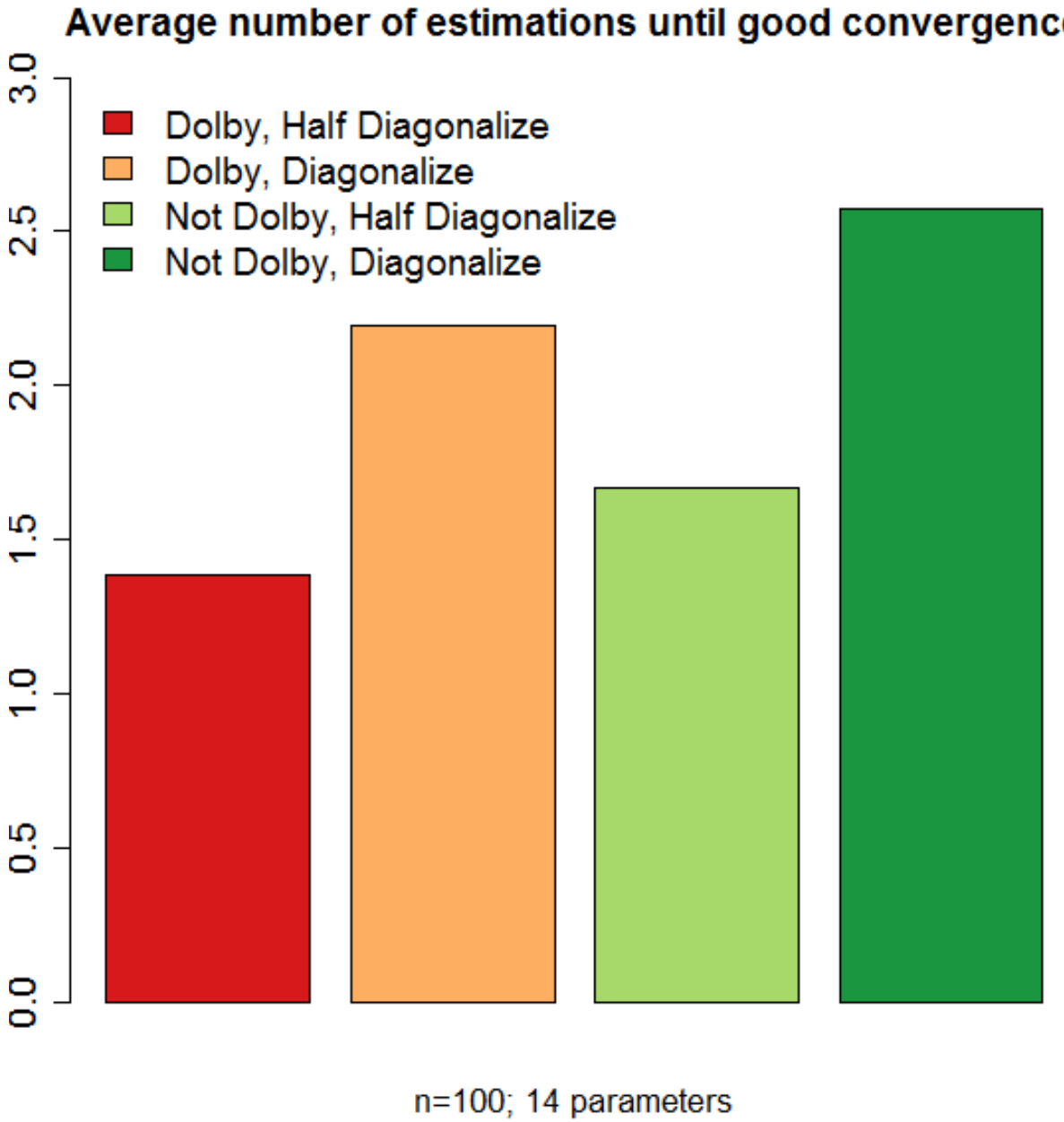
Mean Squared Error

Half diagonalization gives a small increase in RMSE for outdegree and indegree-pop. parameters.

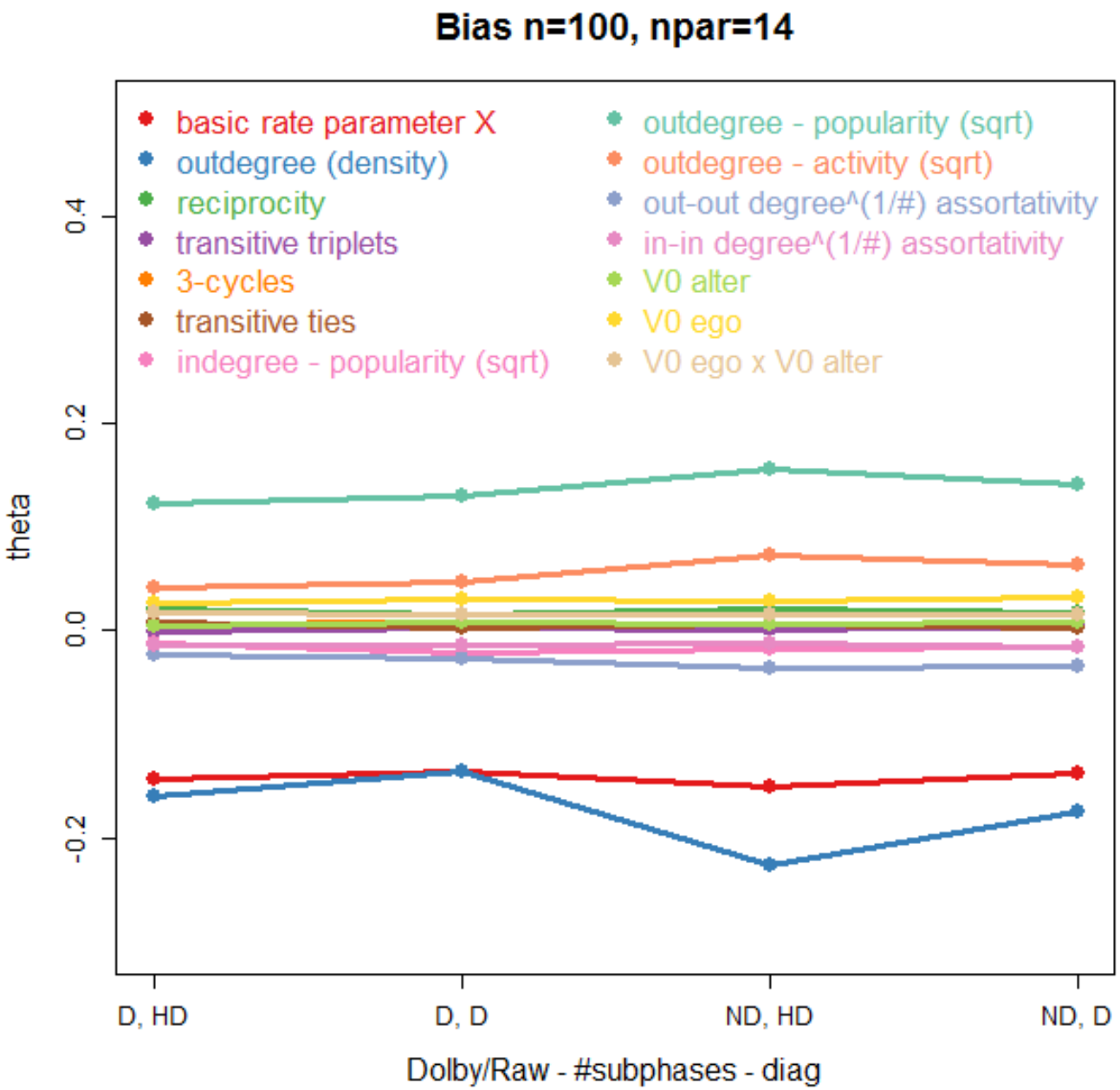
Study 4: $n = 100$ actors, model specification:

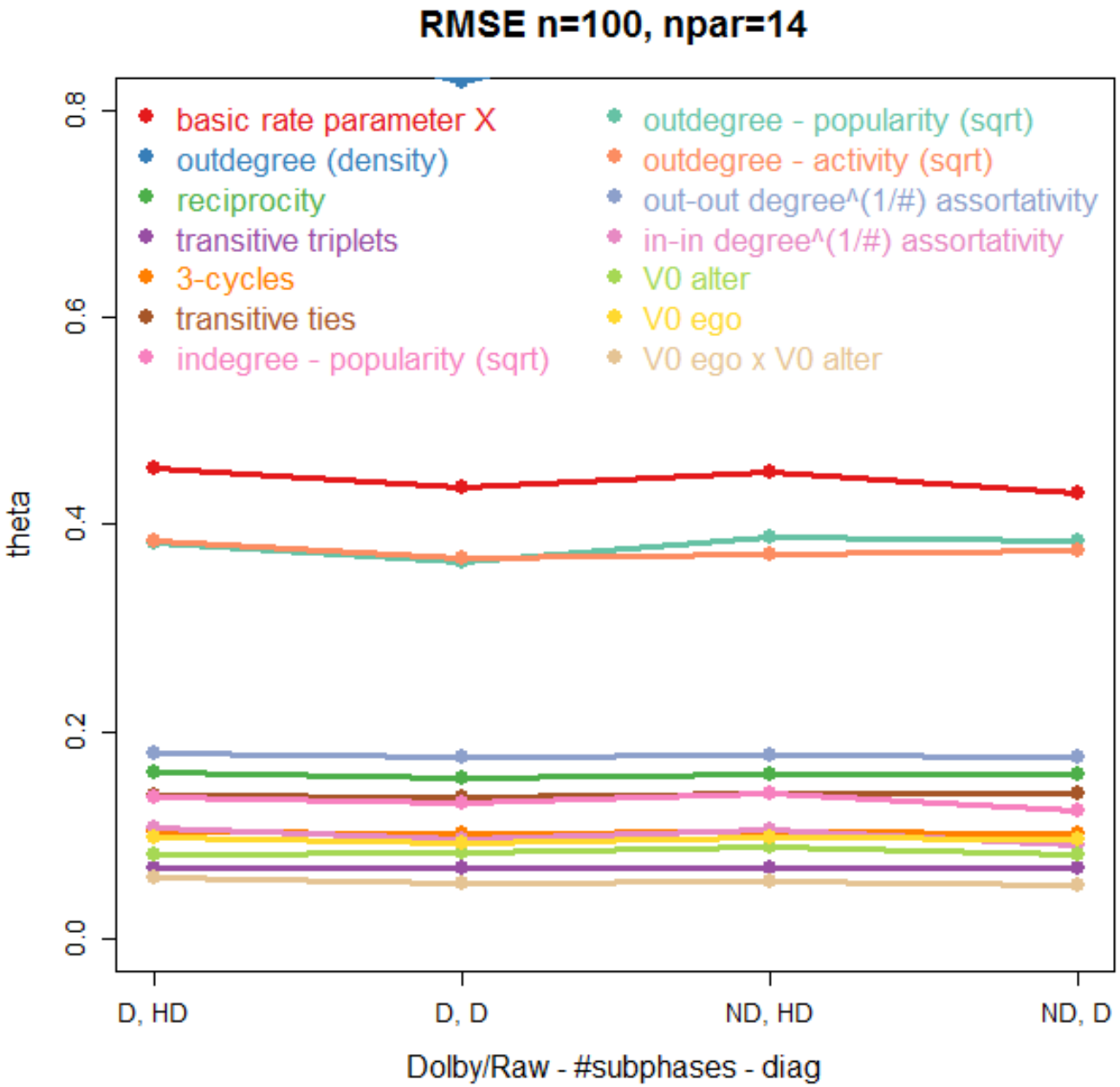
1.	basic rate parameter	6.0
2.	outdegree	–1.6
3.	reciprocity	2.0
4.	transitive triplets	0.2
5.	3-cycles	–0.2
6.	transitive ties	0.8
7.	indegree - popularity (sqrt)	–0.05
8.	outdegree - popularity (sqrt)	–0.2
9.	outdegree - activity (sqrt)	–0.2
10.	out-out degree assortativity (sqrt)	–0.0
11.	in-in degree assortativity (sqrt)	–0.0
12.	V alter	0.2
13.	V ego	0.0
14.	V ego $\times$ V alter	0.2

Average degrees 5.1 (wave 1) and 5.6 (wave 2).



When using Dolby together with Half Diagonalization, average number of estimation runs required is decreased by factor 1.7.


Bias n=100, npar=14
basic rate parameter X
outdegree (density)
reciprocity
transitive triplets
3-cycles
transitive ties
indegree - popularity (sqrt)
outdegree - popularity (sqrt)
outdegree - activity (sqrt)
out-out degree^(1/#) assortativity
in-in degree^(1/#) assortativity
V0 alter
V0 ego
V0 ego x V0 alter
theta
0.4
0.2
0.0
-0.2
D, HD
D, D
ND, HD
ND, D
Dolby/Raw - #subphases - diag



RMSE =
Root
Mean Squared Error

For several parameters, Dolby with Half Diagonalization gives a small increase in RMSE.

Discussion – summary

This study investigated the effects of two modifications
of the Robbins-Monro algorithm used for
Method of Moments parameter estimation in RSiena :

1. Variance reduction by regression on the score function;
2. Greater efficiency by not completely, but only half diagonalizing the matrix of derivatives D.

These are modification of the update in the algorithm,
and are implemented without requiring additional computations.

The consequences were investigated by 4 simulation studies.

Discussion – conclusions

$\Rightarrow$ The necessity to conduct additional estimation runs to achieve convergence was strongly reduced.

Discussion – conclusions

⇒ The necessity to conduct additional estimation runs to achieve convergence was strongly reduced.

⇒ There was a minor reduction of bias for some parameters.

Discussion – conclusions

⇒ The necessity to conduct additional estimation runs to achieve convergence was strongly reduced.

⇒ There was a minor reduction of bias for some parameters.

⇒ There was a minor increase of RMSE for some parameters.

Discussion – conclusions

- ⇒ The necessity to conduct additional estimation runs to achieve convergence was strongly reduced.
- ⇒ There was a minor reduction of bias for some parameters.
- ⇒ There was a minor increase of RMSE for some parameters.
- ⇒ The latter two points suggest that convergence still is incomplete, in spite of the stopping rule.

Discussion – conclusions

- ⇒ The necessity to conduct additional estimation runs to achieve convergence was strongly reduced.
- ⇒ There was a minor reduction of bias for some parameters.
- ⇒ There was a minor increase of RMSE for some parameters.
- ⇒ The latter two points suggest that convergence still is incomplete, in spite of the stopping rule.
- ⇒ Further investigations are planned, directed first at the gain factor a_N, and for generalization.

Discussion – conclusions

- ⇒ The necessity to conduct additional estimation runs to achieve convergence was strongly reduced.
- ⇒ There was a minor reduction of bias for some parameters.
- ⇒ There was a minor increase of RMSE for some parameters.
- ⇒ The latter two points suggest that convergence still is incomplete, in spite of the stopping rule.
- ⇒ Further investigations are planned, directed first at the gain factor a_N, and for generalization.
- ⇒ Implemented in RSiena; ⇒ new default settings.