



MCMC estimation for the p_2 network regression model with crossed random effects

Bonne J. H. Zijlstra^{1*}, Marijtje A. J. van Duijn² and Tom A. B. Snijders^{3,4}

¹Department of Educational Sciences/IOPS, University of Amsterdam, Amsterdam, The Netherlands

²Department of Sociology/ICS/IOPS, University of Groningen, Groningen, The Netherlands

³Department of Sociology/ICS, University of Groningen, Groningen, The Netherlands

⁴Department of Statistics, University of Oxford, Oxford, UK

The p_2 model is a statistical model for the analysis of binary relational data with covariates, as occur in social network studies. It can be characterized as a multinomial regression model with crossed random effects that reflect actor heterogeneity and dependence between the ties from and to the same actor in the network. Three Markov chain Monte Carlo (MCMC) estimation methods for the p_2 model are presented to improve iterative generalized least squares (IGLS) estimation developed earlier, two of which use random walk proposals. The third method, an independence chain sampler, and one of the random walk algorithms use normal approximations of the binary network data to generate proposals in the MCMC algorithms. A large-scale simulation study compares MCMC estimates with IGLS estimates for networks with 20 and 40 actors. It was found that the IGLS estimates have a smaller variance but are severely biased, while the MCMC estimates have a larger variance with a small bias. For networks with 20 actors, mean squared errors are generally comparable or smaller for the IGLS estimates. For networks with 40 actors, mean squared errors are the smallest for the MCMC estimates. Coverage rates of confidence intervals are good for the MCMC estimates but not for the IGLS estimates.

1. Introduction

The p_2 model (Van Duijn, Snijders, & Zijlstra, 2004) is applied in social network analysis. A social network consists of a set of actors and the ties between them. The p_2 model is an extension of the p_1 model (Holland & Leinhardt, 1981) which models binary tie

*Correspondence should be addressed to Bonne J. H. Zijlstra, Nieuwe Prinsengracht 130, 1018 VZ Amsterdam, The Netherlands (e-mail: B.J.H.Zijlstra@uva.nl).

variables using actor-specific sender and receiver parameters and overall density and reciprocity parameters. In the p_2 model, these parameters are regressed on covariates, and correlated random effects are added to the sender and receiver parameters, which considerably reduces the number of model parameters. The p_2 model is formally introduced in Section 2. In this paper, new estimation methods for the p_2 model are proposed and compared to the earlier proposed estimation method in a large-scale simulation study.

The p_2 model can be viewed as a generalized linear mixed model (McCullagh & Nelder, 1989; Skrondal & Rabe-Hesketh, 2004). Until recently, generalized linear mixed models were usually estimated with iterative generalized least squares (IGLS) algorithms using a first- or second-order Taylor approximation of the likelihood function (see Goldstein, 2003), often resulting in biased estimates (Rodríguez & Goldman, 1995, 2001).

An alternative to IGLS is numerical integration (see Gibbons & Hedeker, 1997). However, with the crossed random effects in the p_2 model, standard numerical integration is not feasible and complex approximations are required (Shun & McCullagh, 1995; Shun, 1997). Another alternative is Markov chain Monte Carlo (MCMC) estimation, which has been shown to give nearly unbiased estimates in other generalized linear mixed models (Browne, 1998; Browne & Draper, 2000). Gill and Swartz (2004) used MCMC estimation for a random effects extension of the p_1 model formulated as a log-linear model (Fienberg & Wasserman, 1981). A disadvantage of this approach is that possible covariates can only be categorical and there have to be sufficiently many observations in each category.

MCMC is often applied within a Bayesian framework where, unlike the frequentist IGLS approach, prior distributions for the parameters are specified. Based on the data and priors, the MCMC procedure generates a sample from the posterior distributions. Inference is typically based on a credibility interval of this sample. In this paper, three MCMC estimation algorithms for the p_2 model are proposed and explored by simulations.

In the simulation study, estimates are compared on the basis of their bias and variance. Preferably, both should be small. They are combined in a single measure by computing mean squared errors. To evaluate the quality of inferences about the parameter estimates, coverage of the true parameter value by IGLS confidence intervals and MCMC credibility intervals is investigated. Coverage rates of credibility intervals based on normal approximations of the MCMC estimates are explored as well.

2. The p_2 model

The dependent variable in the p_2 model is a social network or digraph. A digraph represents the directed lines or ties between a fixed set of nodes or actors. A pair of actors and their observed ties is called a dyad. Here, observed ties are coded as binary, resulting in four possible outcomes for each dyad: (0,0), (1,0), (0,1), and (1,1). They are modelled as follows in the p_1 model (Holland & Leinhardt, 1981):

$$P(Y_{ij} = y_1, Y_{ji} = y_2)$$

$$= \frac{\exp\{y_1(\mu + \alpha_i + \beta_j) + y_2(\mu + \alpha_j + \beta_i) + y_1 y_2 \rho\}}{1 + \exp\{\mu + \alpha_i + \beta_j\} + \exp\{\mu + \alpha_j + \beta_i\} + \exp\{\mu + \mu + \alpha_i + \beta_j + \alpha_j + \beta_i + \rho\}},$$

$$y_1, y_2 \in \{0, 1\}, 1 \leq i < j \leq n,$$

(1)

where Y_{ij} is the directed tie variable (with value 0 or 1, where 1 indicates the presence of a tie) from actor i to actor j . Parameters α_i and β_i are the sender and receiver parameters, respectively, of actor i . Parameter μ is called the density parameter. It represents the log odds of a tie in the case of zero sender and receiver effects and no reciprocity. Parameter ρ represents the log odds ratio of a symmetric over a non-symmetric dyad. Therefore, ρ is called the reciprocity parameter. For identification purposes, a restriction needs to be imposed on the sender and receiver parameters, usually $\sum \alpha_i = \sum \beta_i = 0$.

In the p_2 model (Van Duijn *et al.*, 2004), α , β , μ , and ρ are further extended to include covariates. Density and reciprocity parameters μ and ρ have additional subscripts i and j to indicate that they are dyad-specific,

$$\begin{aligned}\alpha_i &= \mathbf{X}_{1i}\boldsymbol{\gamma}_1 + A_i, & \beta_i &= \mathbf{X}_{2i}\boldsymbol{\gamma}_2 + B_i, & \mu_{ij} &= \mu + \mathbf{Z}_{1ij}\boldsymbol{\delta}_1, \\ \rho_{ij} &= \rho + \mathbf{Z}_{2ij}\boldsymbol{\delta}_2,\end{aligned}\tag{2}$$

where A_i and B_i are normal independent and identically distributed random effects with $E(A_i) = E(B_i) = 0$, variances σ_A^2 and σ_B^2 , and covariance σ_{AB} . The random effects for actors $i = 1, \dots, n$ are collected in vector \mathbf{C} ,

$$\mathbf{C}_i^T = (A_i, B_i),$$

with covariance matrix Σ ,

$$\Sigma = \begin{pmatrix} \sigma_A^2 & \sigma_{AB} \\ \sigma_{AB} & \sigma_B^2 \end{pmatrix}.\tag{3}$$

Because the random actor effects occur in all ties sent or received by the same actor, modelling the dependence between these relations, the random effects in the p_2 model have a crossed structure.

Matrices \mathbf{X}_1 and \mathbf{X}_2 contain actor-specific covariates and \mathbf{Z}_1 and \mathbf{Z}_2 are matrices with dyad-specific covariates. Vectors $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\delta}_1$, and $\boldsymbol{\delta}_2$ contain fixed regression parameters for the sender, receiver, density, and reciprocity effects, respectively. All fixed parameters are collected in the k -dimensional parameter vector $\boldsymbol{\theta}$,

$$\boldsymbol{\theta}^T = (\mu, \rho, \boldsymbol{\gamma}_1^T, \boldsymbol{\gamma}_2^T, \boldsymbol{\delta}_1^T, \boldsymbol{\delta}_2^T).$$

The p_2 model can be formulated as a generalized linear mixed model for a multinomial outcome variable with four categories, where the data \mathbf{Y} are represented by $n(n-1)/2$ stacked three-dimensional vectors \mathbf{d}_{ij} ,

$$\mathbf{d}_{ij} = \begin{pmatrix} Y_{ij}(1 - Y_{ji}) \\ Y_{ji}(1 - Y_{ij}) \\ Y_{ij}Y_{ji} \end{pmatrix} = \begin{pmatrix} d_{1ij} \\ d_{2ij} \\ d_{3ij} \end{pmatrix}.$$

Using this representation, the numerator in (1) can be expressed as $\exp(\xi_{ij}^T \mathbf{d}_{ij})$ with

$$\xi_{ij} = \begin{pmatrix} \alpha_i + \beta_j + \mu \\ \alpha_j + \beta_i + \mu \\ \xi_{1ij} + \xi_{2ij} + \rho \end{pmatrix} = \begin{pmatrix} \xi_{1ij} \\ \xi_{2ij} \\ \xi_{3ij} \end{pmatrix}.$$

The probability of \mathbf{Y} under the p_1 model is then rewritten as an exponential family distribution,

$$P(\mathbf{Y} = \mathbf{y}) = \exp(\xi^T \mathbf{d} - b(\xi)), \quad (4)$$

with $b(\xi) = \sum_{i < j} k(\xi_{ij})$, and $k(\xi_{ij}) = \ln(1 + \exp(\xi_{1ij}) + \exp(\xi_{2ij}) + \exp(\xi_{3ij}))$.

The p_1 model is thus a generalized linear model, expressed here as a multinomial model with (0,0) as the reference category. The p_2 model is obtained by substituting ξ_{ij} using (2), resulting in a generalized linear mixed model (4) with

$$\xi_{ij} = \begin{pmatrix} A_i + \mathbf{X}_{1i}\gamma_1 + B_j + \mathbf{X}_{2j}\gamma_2 + \mu_{ij} + \mathbf{Z}_{1ij}\delta_1 \\ A_j + \mathbf{X}_{1j}\gamma_1 + B_i + \mathbf{X}_{2i}\gamma_2 + \mu_{ji} + \mathbf{Z}_{1ji}\delta_1 \\ \xi_{1ij} + \xi_{2ij} + \rho_{ij} + \mathbf{Z}_{2ij}\delta_2 \end{pmatrix} = \begin{pmatrix} \xi_{1ij} \\ \xi_{2ij} \\ \xi_{3ij} \end{pmatrix}. \quad (5)$$

The likelihood function of the p_2 model can be written as

$$P(\mathbf{Y}|\boldsymbol{\theta}, \Sigma) = \int P(\mathbf{Y}|\boldsymbol{\theta}, \mathbf{C})P(\mathbf{C}|\Sigma)d\mathbf{C}, \quad (6)$$

where the integral is taken over the random effects \mathbf{C} and $P(\mathbf{Y}|\boldsymbol{\theta}, \mathbf{C})$ is the p_1 probability (4) with ξ defined as in (5). One way of dealing with the integral is through IGLS, an estimation procedure based on Taylor approximations to generalized linear mixed models, which is discussed in the next section. However, IGLS does not always work well for non-linear models (Rodríguez & Goldman, 1995, 2001). An alternative is MCMC estimation, discussed in Sections 4–8.

3. IGLS estimation of the p_2 model

For non-linear models, several iterative generalized least squares (see Goldstein, 2003) algorithms are available, which alternate two basic steps until convergence. In the first step, the model is approximated around its current estimates, and estimates of the fixed parameters are obtained. In the second step, the variance of the residuals is linearly regressed on indicator variables for the variance components, yielding estimates for the variance and covariance parameters of the random effects (see Goldstein, 2003, Appendix 2.1).

The marginal quasi-likelihood (MQL-1) algorithm uses a first-order Taylor approximation around the current estimates of the fixed parameters (see Goldstein & Rasbash, 1996). The MQL-2 algorithm applies a second-order Taylor approximation. In both MQL algorithms, random parameters are set to zero in the first step. Rodríguez and Goldman (1995) showed that the MQL algorithm in other generalized linear mixed models for binary data produces biased estimates for fixed and random effect parameters. In response, Goldstein and Rasbash (1996) proposed the PQL-2

algorithm which performs considerably better. The penalized quasi-likelihood (PQL) algorithm uses an approximation around the current estimates including estimates of random effects, with a first- (PQL-1) or second-order (PQL-2) Taylor expansion. However, even with PQL-2, some estimates are still biased (Goldstein & Rasbash, 1996). Therefore, the IGLS algorithms for the p_2 model are also expected to yield biased estimates, the extent of which is examined in a simulation study, presented in Section 9.

Two IGLS algorithms for the p_2 model are briefly discussed here, both of which are comparable to the MQL-1 algorithm. The first one (Van Duijn *et al.*, 2004) formulates the probability of the two observed ties in a dyad as the product of the two binomial probabilities of the first and the second tie conditional on the first,

$$P(Y_{ij} = y_1, Y_{ji} = y_2) = P(Y_{ij} = y_1)P(Y_{ji} = y_2 | Y_{ij} = y_1), \quad y_1, y_2 \in \{0, 1\}, \quad (7)$$

where both binomial probabilities are derived from the p_2 model. Because of the two 'observations' for each dyad, this algorithm is called IGLS-2.

The approximation of (7) depends on which of the two tie variables is taken to be first, but allows incomplete dyadic data, i.e. Y_{ij} observed but Y_{ji} not. Results from the IGLS-2 procedure change, usually only slightly, if the first and second tie variables are reversed. This can be avoided by expressing the p_2 model as a multinomial model in which the dyads have four possible outcomes, as in (4) using (5). The resulting algorithm, for obvious reasons called IGLS-3, is insensitive to data ordering, but requires complete dyadic data.

The IGLS-3 algorithm is similar to IGLS-2, except that instead of the independent binomial variance structure, the probabilities in (4), using (5), have a multinomial covariance matrix,

$$\text{diag}(E(d_{ij})) - E(d_{ij})E(d_{ij})^T. \quad (8)$$

The IGLS algorithm can be refined by taking into account the sampling variation of the regression parameters θ . This is done with a restricted IGLS (RIGLS) algorithm (Goldstein & Rasbash, 1992). In the simulation study, results from the IGLS-2, IGLS-3, and RIGLS-3 algorithms are compared, along with the results of MCMC algorithms.

4. MCMC estimation of the p_2 model

In Markov chain Monte Carlo estimation, a Markov chain is used to obtain a sample from a (multivariate) distribution from which Monte Carlo estimates are calculated. Here, the p_2 model is formulated as a Bayesian model and MCMC is applied to obtain a sample of the posterior distributions of the model parameters (see Gilks, Richardson, & Spiegelhalter, 1996). Applying Bayes' theorem, posterior distributions for the model parameters conditional on the data and the prior distributions are derived. An advantage of MCMC is that it provides conceptually straightforward methods to set up simulation algorithms for the posterior distributions. These generate samples for which convergence is achieved, after a sufficiently long initial burn-in period and for large enough samples.

4.1. The p_2 model including prior distributions

In the full multivariate distribution of the p_2 model, (6) is extended with the prior distributions for the parameters θ and Σ , $P_\theta(\theta)$ and $P_\Sigma(\Sigma)$. Assuming that Σ and θ are *a priori* independent, the factorization

$$P(\mathbf{Y}, \mathbf{C}, \Sigma, \theta) = P_Y(\mathbf{Y}|\mathbf{C}, \theta)P_C(\mathbf{C}|\Sigma)P_\Sigma(\Sigma)P_\theta(\theta) \quad (9)$$

results, where

$$P_Y(\mathbf{Y}|\mathbf{C}, \theta) = \prod_{i < j}^n \{f_1(Y_{ij}, Y_{ji}|\mathbf{C}_i, \mathbf{C}_j, \theta)\},$$

with f_1 the p_2 probabilities (conditional on θ and \mathbf{C}) given in (4) using (5). The second factor of (9) is the product over all actors of the bivariate normal distribution of the random effects,

$$P_C(\mathbf{C}|\Sigma) = \prod_{i=1}^n f_2(\mathbf{C}_i|\Sigma),$$

with

$$f_2(\mathbf{C}_i|\Sigma) = (2\pi)^{-1} |\Sigma|^{-1/2} \exp \left\{ -\frac{1}{2} (\mathbf{C}_i^T \Sigma^{-1} \mathbf{C}_i) \right\}.$$

Given the assumption of normally distributed random effects, a convenient choice is to take the prior distribution of the covariance matrix for the random effects, $P_\Sigma(\Sigma)$, as inverse Wishart (see Johnson & Kotz, 1972, pp. 158–159). This is the conjugate prior distribution. Here, we assume for Σ a prior inverse Wishart distribution with $\nu_\sigma = 3$ degrees of freedom and prior covariance matrix $\Sigma_\sigma = \mathbf{I}$, the identity matrix, which represents little prior information for Σ .

The prior distribution for the fixed model parameters, $P_\theta(\theta)$, is chosen to be multivariate normal with means $\mu_\theta = \mathbf{0}$ and a diagonal covariance matrix Σ_θ depending on effect size considerations in multinomial regression. Our aim is to have a slightly informative prior, which is applicable in most social network data sets. The p_2 probability function resembles the logistic function $e^x/(1 + e^x)$, where $x = \pm 10$ corresponds to extremely large or small probabilities. Therefore, it is assumed that the products $\gamma_1 \mathbf{X}_1$, $\gamma_2 \mathbf{X}_2$, $\delta_1 \mathbf{Z}_1$, $\delta_2 \mathbf{Z}_2$ in the p_2 model are unlikely to exceed 10 in absolute value. In more general notation, $|\theta_i \mathbf{r}_i| = 10$ is considered to be a very large value for the product of a certain parameter θ_i with corresponding covariate \mathbf{r}_i . Thus, a normal prior distribution with zero means and standard deviation $10/\sigma(r)$ is assigned to each parameter θ_i :

$$\theta_i \sim N\left(0, \frac{100}{\sigma^2(\mathbf{r}_i)}\right), \quad \theta_i \in \{\gamma_1^T, \gamma_2^T, \delta_1^T, \delta_2^T\}.$$

The parameters μ and ρ are assumed to have a normal prior with zero mean and variance 100.

4.2. A hybrid Metropolis–Hastings algorithm for the p_2 model

A convenient starting-point to set up a transition kernel for a Markov chain that generates samples from the posterior distributions of the p_2 model parameters is the Gibbs sampler (Geman & Geman, 1984), defined by consecutively sampling from the conditional distributions of each parameter, given the data, all other parameters, and the prior distributions. The Gibbs sampler is a special case of the Metropolis–Hastings algorithm (see Chib & Greenberg, 1995). If the conditional distribution has an unfamiliar distributional form, it can be approximated using the Metropolis–Hastings algorithm which, instead of directly drawing from the conditional distribution, either accepts or rejects a proposal as a next value in the chain. For the p_2 model, proposals are generated in two ways: as independent proposals and as random perturbations from the current value of the parameters. This is elaborated below.

From the factorization of the parameter sets in (9), three conditional distributions required for the sampler are derived, $P(\mathbf{C}|\mathbf{Y}, \boldsymbol{\theta}, \boldsymbol{\Sigma})$, $P(\boldsymbol{\Sigma}|\mathbf{C})$, and $P(\boldsymbol{\theta}|\mathbf{Y}, \mathbf{C}, \boldsymbol{\Sigma})$. The conditional distribution of the random effects is proportional to the product of the p_2 distribution, (4) with (5), and the bivariate normal for the random effects,

$$P(\mathbf{C}|\mathbf{Y}, \boldsymbol{\theta}, \boldsymbol{\Sigma}) \propto \prod_{i < j}^n \{f_1(Y_{ij}, Y_{ji}|\mathbf{C}_i, \mathbf{C}_j, \boldsymbol{\theta})\} P_C(\mathbf{C}|\boldsymbol{\Sigma}),$$

which cannot be sampled from directly.

The conditional distribution of the covariance matrix $\boldsymbol{\Sigma}$ is proportional to the product $P_C(\mathbf{C}|\boldsymbol{\Sigma})P_{\Sigma}(\boldsymbol{\Sigma})$ of the density of the random effects and the inverse Wishart prior for $\boldsymbol{\Sigma}$, resulting in an inverse Wishart distribution with $\nu_{\sigma} + n$ degrees of freedom and covariance matrix $(\sum_{i=1}^n \mathbf{C}_i \mathbf{C}_i^T + \boldsymbol{\Sigma}_{\sigma}^{-1})^{-1}$ (see Box & Tiao, 1973, p. 427). Draws from the conditional distribution of $\boldsymbol{\Sigma}$ are obtained by inverting a draw from the Wishart distribution for the conditional distribution of $\boldsymbol{\Sigma}^{-1}$ (using the algorithm by Odell & Feiveson, 1966).

The conditional distribution of the fixed parameters $\boldsymbol{\theta}$ is again proportional to the product of the p_2 distribution, (4) with (5), and a normal prior distribution,

$$P(\boldsymbol{\theta}|\mathbf{Y}, \mathbf{C}, \boldsymbol{\Sigma}) \propto \left\{ \prod_{i < j}^n f_1(Y_{ij}, Y_{ji}|\mathbf{C}_i, \mathbf{C}_j, \boldsymbol{\theta}) \right\} P_{\theta}(\boldsymbol{\theta}).$$

Because neither the conditional distributions of $\boldsymbol{\theta}$ nor the conditional distribution of \mathbf{C} can be sampled from directly, draws for these variables are obtained by applying the Metropolis–Hastings algorithm.

The Metropolis–Hastings step for the fixed parameters in $\boldsymbol{\theta}$ accepts a proposal $\boldsymbol{\theta}^*$ with a probability that depends on the ratio of the conditional distribution for the proposal and for $\boldsymbol{\theta}^{(t-1)}$, the current value of $\boldsymbol{\theta}$. It further depends on the probability of the current value and the proposal defined by a generating function $g(\boldsymbol{\theta})$. The probability can be expressed as

$$\min \left(1, \frac{P(\boldsymbol{\theta}^*|\mathbf{Y}, \mathbf{C}, \boldsymbol{\Sigma})g(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)}{P(\boldsymbol{\theta}^{(t-1)}|\mathbf{Y}, \mathbf{C}, \boldsymbol{\Sigma})g(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})} \right). \quad (10)$$

If the proposal $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^*$ is rejected, $\boldsymbol{\theta}^{(t-1)}$ is retained.

The Metropolis-Hastings step for sampling the random effects \mathbf{C}_i , with proposal generating function $b(\mathbf{C}_i)$, accepts the proposal $\mathbf{C}_i^{(t)} = \mathbf{C}_i^*$ with probability

$$\min\left(1, \frac{P(\mathbf{C}_i^*|\mathbf{Y}, \boldsymbol{\theta}, \boldsymbol{\Sigma})b(\mathbf{C}_i^{(t-1)}|\mathbf{C}_i^*)}{P(\mathbf{C}_i^{(t-1)}|\mathbf{Y}, \boldsymbol{\theta}, \boldsymbol{\Sigma})b(\mathbf{C}_i^*|\mathbf{C}_i^{(t-1)})}\right). \quad (11)$$

The following sections present three MCMC sampling procedures, distinguished by their proposal distributions. The first is an adaptive algorithm that evaluates random walk proposals (see Gilks *et al.*, 1996, for the random walk). The second is an independence chain sampler, evaluating independent proposals, which gives rise to the third algorithm, also using random walk proposals.

5. Proposal distributions for an adaptive random walk algorithm

Based on the work by Browne (1998) for multi-level models (see also Browne & Draper, 2006), a procedure for an adaptive random walk algorithm for the p_2 model was developed. In a random walk algorithm, proposals are evaluated in Metropolis steps (Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953), a particular case of the Metropolis-Hastings kernel where the probabilities of getting to and from a proposal are equal. For the p_2 model, this leads to acceptance probabilities as in (10) and (11) with $g(\cdot)$ and $b(\cdot)$ cancelling out because they are equal in the numerator and denominator.

In the random walk algorithm, the proposed value $\boldsymbol{\theta}^*$ is a random perturbation from the current value,

$$\boldsymbol{\theta}^* = \boldsymbol{\theta}^{(t-1)} + \boldsymbol{\zeta}_1, \quad \boldsymbol{\zeta}_1 \sim N(\mathbf{0}, \boldsymbol{\Phi}_\theta)$$

where $\boldsymbol{\zeta}_1$ is a perturbation from a multivariate normal distribution with zero means and covariance matrix $\boldsymbol{\Phi}_\theta$, or equivalently, $g(\boldsymbol{\theta}^*) = N(\boldsymbol{\theta}^{(t-1)}, \boldsymbol{\Phi}_\theta)$.

Likewise, for evaluating proposals for the random effects, $b(\mathbf{C}_i^*) = N(\mathbf{C}_i^{(t-1)}, \boldsymbol{\Phi}_C)$. Each new set \mathbf{C}^* of n proposals is composed of the current values plus a random perturbation from a multivariate normal distribution with zero means and covariance matrix $\boldsymbol{\Phi}_C$:

$$\mathbf{C}_i^* = \mathbf{C}_i^{(t-1)} + \boldsymbol{\zeta}_{2i}, \quad \boldsymbol{\zeta}_{2i} \sim N(\mathbf{0}, \boldsymbol{\Phi}_C).$$

In the adaptive random walk algorithm, proposals for the random effects are evaluated for all n actors in a single Metropolis step, using the same covariance matrix for the random perturbations, $\boldsymbol{\Phi}_C$.

The hard part in the random walk algorithm is to get good working values for $\boldsymbol{\Phi}_\theta$ and $\boldsymbol{\Phi}_C$. For ill-chosen values, the random walk algorithm performs very poorly up to the point where it does not move at all. For the p_2 model, the random walk algorithm is made adaptive following the method proposed by Browne (1998) and Browne and Draper (2006). In the initial stages, variances of the random perturbation are adapted to some optimal acceptance rate in the Metropolis steps, after obtaining a batch of observations of the parameters from the Markov chain. With each batch, the covariance matrices of $\boldsymbol{\theta}$ and \mathbf{C} are updated using the additional observations and multiplied by a scale factor (S) which depends on the acceptance rate of the Metropolis steps. The acceptance rate is monitored by the number of accepted proposals in a batch (N_a). S adjusts N_a to achieve the required number of accepted proposals in a batch (N_o).

If $N_a > N_o$ then

$$S = S \left(1 + \left(1 - \frac{N_b - N_a}{N_b - N_o} \right) F \right), \quad \text{and otherwise} \quad S = \frac{S}{1 + (1 - N_a/N_o)F}, \quad (12)$$

where N_b is the size of the batch and F is a certain factor, which we take to be 1. Thus, a simplified version of ‘sampler 2’ from Browne (1998) is applied to perturbations from multivariate distributions.

The current random walk algorithm has two adaptive sequences (12), one for θ and one for C , both using 20 batches of 125 iterations. Because plausible starting configurations of the random perturbations are needed for the adaptive sequences to work well, RIGLS-3 covariance matrices of the fixed and random parameters are used as starting configurations for the random perturbations. If the RIGLS-3 estimation fails, only the first two iterations of the RIGLS-3 algorithm are performed. These fail much less frequently, while still providing a reasonable starting-point.

Gelman, Roberts, and Gilks (1995) state that optimal acceptance rates for random walk proposal distributions are between 44% for a single parameter and decline towards 25% for multidimensional distributions. The aim here is an acceptance rate of 33%, for which the efficiency of random walk algorithms of all dimensions proved to be near optimal (Gelman *et al.*, 1995). After the adaptive sequence, the covariance matrix for the random perturbations is fixed and used in the proposal generation throughout the remainder of the sampling procedure.

Because in this adaptive random walk algorithm the entire matrix C is evaluated in a single step, this algorithm is called the ‘random walk simultaneous’, or RW Sim, algorithm.

6. Proposal distributions applying normal approximations to the data

Considering a model in which the data have a multivariate normal distribution with approximately the same mean vector and covariance matrix, as in the p_2 model, leads to conditional distributions of θ and C which also are normal. These approximated conditional distributions for θ and C , which are easily sampled from, appear to yield reasonably good proposal distributions. These are applied in an independence chain algorithm and a related random walk algorithm, which are derived below.

6.1. Independence chain

In the independence chain sampler, the proposal θ^* is drawn from a distribution g , independent of the current value, $\theta^{(t-1)}$; similarly, the proposal C_i^* is drawn from h , independent of $C_i^{(t-1)}$. To obtain the proposal generating functions g and h , the distribution of Y is approximated by a normal distribution, where a first-order Taylor approximation is used for the mean, m , and a block diagonal structure according to the multinomial distribution, as given in (8), for the covariance matrix T . The approximations of Y for the fixed parameters θ and for the random effects C are different because two Taylor approximations are used with respect to each set of parameters,

$$Y \sim N(m(\theta^0, C^0) + D_0(\theta^0, C^0)(\theta - \theta^0), T(\theta^0, C^0))$$

$$Y \sim N(m(\theta^0, C^0) + F_0(\theta^0, C^0)(C - C^0), T(\theta^0, C^0)),$$

where the superscript zero denotes the current estimates. The Taylor approximation is centred on the mean $m(\boldsymbol{\theta}^0, \mathbf{C}^0) = E_{\theta}(Y|\mathbf{C}^0, \boldsymbol{\theta}^0)$, the expectation of \mathbf{Y} given the current values of the parameters $\boldsymbol{\theta}$ and \mathbf{C} as in (4) using (5). $D_0(\boldsymbol{\theta}^0, \mathbf{C}^0)$ has elements $\partial m(\boldsymbol{\theta}^0, \mathbf{C}^0)/\partial \boldsymbol{\theta}$, the vector of partial derivatives of $m(\boldsymbol{\theta}^0, \mathbf{C}^0)$ with respect to the fixed parameters in $\boldsymbol{\theta}$. $F_0(\boldsymbol{\theta}^0, \mathbf{C}^0)$ contains the partial derivatives with respect to the random effects in \mathbf{C} .

The approximated conditional normal distribution of $\boldsymbol{\theta}$ based on the normal approximation of \mathbf{Y} is derived as follows. After orthogonalization, the approximate distribution of \mathbf{Y} is

$$\mathbf{T}^{-1/2}(\mathbf{Y} - \mathbf{m}) \sim N(\mathbf{T}^{-1/2}\mathbf{D}_0(\boldsymbol{\theta} - \boldsymbol{\theta}^0), \mathbf{I}),$$

with $\mathbf{m} = m(\boldsymbol{\theta}^0, \mathbf{C}^0)$, $\mathbf{D}_0 = D_0(\boldsymbol{\theta}^0, \mathbf{C}^0)$ and $\mathbf{T} = T(\boldsymbol{\theta}^0, \mathbf{C}^0)$, and $\mathbf{T}^{-1/2}$ is obtained from the eigenvalue decomposition $\mathbf{T} = \mathbf{K}\mathbf{\Lambda}\mathbf{K}^T$ giving $\mathbf{T}^{-1/2} = \mathbf{K}\mathbf{\Lambda}^{-1/2}\mathbf{K}^T$. Further transformation to a linear model gives

$$\mathbf{T}^{-1/2}((\mathbf{Y} - \mathbf{m}) + \mathbf{D}_0\boldsymbol{\theta}^0) \sim N(\mathbf{G}\boldsymbol{\theta}, \mathbf{I}),$$

with $\mathbf{G} = \mathbf{T}^{-1/2}\mathbf{D}_0$. Denoting $\mathbf{T}^{-1/2}((\mathbf{Y} - \mathbf{m}) + \mathbf{D}_0\boldsymbol{\theta}^0)$ by \mathbf{W} , an expression for a linear model for \mathbf{W} with parameter $\boldsymbol{\theta}$ and predictor \mathbf{G} is obtained. The linear model and the normal prior distribution of $\boldsymbol{\theta}$ with mean $\boldsymbol{\mu}_{\theta}$ and variance Σ_{θ} result in a conditional normal distribution for $\boldsymbol{\theta}$ (for a similar type of derivation, see Box & Tiao, 1973, p. 74):

$$\boldsymbol{\theta}|\mathbf{W} \sim N\left((\mathbf{G}^T\mathbf{G} + \Sigma_{\theta}^{-1})^{-1}(\mathbf{G}^T\mathbf{W} + \Sigma_{\theta}^{-1}\boldsymbol{\mu}_{\theta}), (\mathbf{G}^T\mathbf{G} + \Sigma_{\theta}^{-1})^{-1}\right). \quad (13)$$

In the independence chain sampler, the function $g(\boldsymbol{\theta}|\mathbf{W})$ generates proposals from (13). Similarly, an approximated normal conditional distribution for \mathbf{C}_i is derived as

$$\mathbf{C}_i|\mathbf{Z} \sim N\left((\mathbf{H}^T\mathbf{H} + \Sigma_C^{-1})^{-1}(\mathbf{H}^T\mathbf{Z} + \Sigma_C^{-1}\boldsymbol{\mu}_C), (\mathbf{H}^T\mathbf{H} + \Sigma_C^{-1})^{-1}\right), \quad (14)$$

with

$$\mathbf{H} = \mathbf{T}^{-1/2}\mathbf{F}_0,$$

and

$$\mathbf{Z} = \mathbf{T}^{-1/2}((\mathbf{Y} - \mathbf{m}) + \mathbf{F}_0\mathbf{C}^0),$$

from which the function $b(\mathbf{C}_i|\mathbf{Z})$ generates proposals in the independence chain. In the Metropolis-Hastings step (10) of the independence chain sampler, $g(\boldsymbol{\theta}^*)$ represents the probability of obtaining $\boldsymbol{\theta}^*$ given the current values of all parameters, including $\boldsymbol{\theta}^{(t-1)}$. This can be written as $g(\boldsymbol{\theta}^*|\mathbf{W}^{(t-1)})$.

If $\boldsymbol{\theta}^*$ is accepted, the proposal distribution g is replaced with the normal density (13) based on all current parameter estimates, including $\boldsymbol{\theta}^*$ and \mathbf{Y} . In that case, in the independence chain sampler, $g(\boldsymbol{\theta}^{(t-1)})$ in (10) becomes $g(\boldsymbol{\theta}^{(t-1)}|\mathbf{W}^*)$, where \mathbf{W}^* is \mathbf{W} based on all current values and $\boldsymbol{\theta}^*$. For the random effects \mathbf{C}_i , similar Metropolis-Hastings steps are taken. This algorithm is called the 'independence chain' (IC) algorithm.

Note that it is also possible to maintain a certain proposal generating function $g(\boldsymbol{\theta}^*|\mathbf{W}^{(t-1)})$ for multiple Metropolis-Hastings steps. In that case, the proposal generating function is not replaced when $\boldsymbol{\theta}^*$ is accepted. This saves computation time for a cycle, but

more cycles are required for convergence of MCMC. In Section 7, an optimum between these two opposite mechanisms is sought.

6.2. Random walk algorithm

From the normal approximation a second random walk algorithm is derived, not using adaptation, but with the covariance matrices of the normal approximations for the conditional distributions of θ and C_i as Φ_0 and Φ_{C_i} .

In the burn-in sequence, the initial stage before obtaining the MCMC sample, these matrices are updated every 100 iterations using Metropolis–Hastings steps (10) and (11). In these steps, the normal approximations (13) and (14) are recalculated, providing new values for Φ_0 and Φ_{C_i} . Because the computationally very intensive update of Φ_0 and Φ_{C_i} is performed far less frequently, it requires considerably fewer computational resources per iteration than the IC algorithm. Note that since in (13) the C_i variables are independent, this random walk algorithm evaluates the random effects for all actors separately. It is simply called the ‘random walk’ (RW) algorithm.

7. Implementation of the MCMC algorithms

This section investigates the behaviour of the three MCMC algorithms proposed in the previous sections. The question is whether they converge to similar posterior distributions and how efficiently they converge computationally. The study is based on one simulated data set with 40 actors and parameter values $\sigma_A^2 = 1$, $\sigma_B^2 = 1$, $\sigma_{AB} = 0$, $\mu = -2$, $\rho = 2$, $\gamma_1 = 0.05$, and $\delta_1 = 0.5$, where the covariate for γ_1 is an integer ranking the 40 actors (taking values 1–40) and the covariate for δ_1 is a network generated with the stated values of the parameters σ_A^2 , σ_B^2 , σ_{AB} , μ and ρ . (The model is also used as Model 2 in the simulation study, presented in the next section.) The reason for using this model is that we know from experience that this is a realistic type of data set, but the estimation of parameter γ_1 for the actor rank number can be troublesome because of its large negative correlation with μ in the MCMC simulation chain. All simulation results presented are based on 200,000 iterations of the MCMC algorithms, following a burn-in of 10,000 for the RW Sim algorithm and 2,000 for the IC and RW algorithms.

In Figure 1, displaying the posterior means and standard errors (posterior standard deviations) of parameters μ , ρ , γ_1 , and δ_1 , it can be seen that all three algorithms converge reasonably quickly to the posterior means.

Figure 2 shows that the standard errors (posterior standard deviations) provided by the three algorithms have not converged as closely as the posterior means, particularly for μ . The RW and RW Sim algorithms appear to show rather arbitrary variation, but the IC algorithm systematically underestimates the standard errors. This is true for all parameters, also for γ_1 where this is not immediately clear from the figure.

In the right-hand panel of Figure 2, a remedy for the smaller standard errors from the independence chain sampler can be observed. The figure shows the standard errors for μ , ρ , δ_1 , and γ_1 for runs of the independence chain when the variances of the proposal distributions are multiplied by 3, 5, and 10. For all parameters, the independence chain now seems to converge to similar posterior standard errors as the random walk algorithms. Apparently, the variation in the true conditional distribution is underestimated by the normal approximation. Multiplying the covariance matrices of the proposal distributions by a factor of at least 3 appears to be a satisfactory remedy.

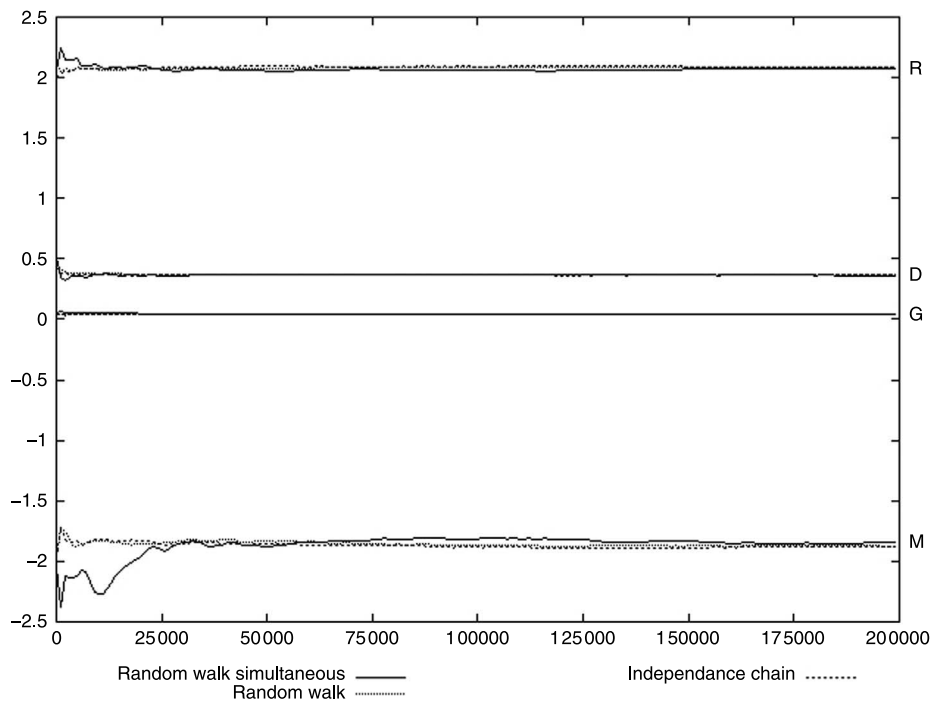


Figure 1. Posterior means for ρ (R), δ_I (D), γ_I (G), and μ (M).

The independence chain sampler is computationally much more demanding than the random walk algorithms. It took approximately 1, 8, and 22 hours for the RW Sim, RW, and IC algorithms, respectively, to obtain the samples of Figure 1 on the same standard 1 GHz PC performing no other major tasks. If the same proposal distribution in the independence chain sampler is maintained for a number of iterations, the computational efficiency of this algorithm could be improved by up to a factor of 3 per iteration. However, maintaining a proposal distribution for several iterations will decrease the acceptance rate of the proposals and consequently also decrease the efficiency of the independence chain. After some experimenting, multiplication of the

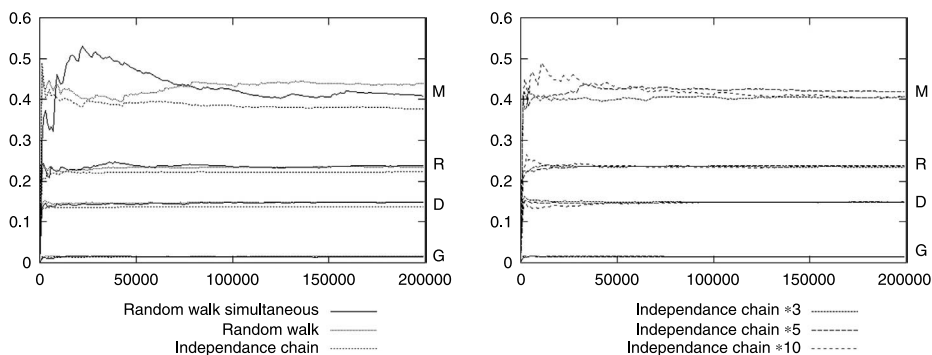


Figure 2. Posterior standard errors for μ (M), ρ (R), δ_I (D), and γ_I (G).

proposal variance by 5 and maintaining the same proposal generating distribution for 10 iterations appears to be a reasonable choice. Of course, it is not guaranteed that these choices work for different parameter values. The proof of the pudding is in the eating: the algorithm with this implementation will be used in the simulation studies treated later in this paper. The quality of convergence will be one of the aspects evaluated to determine the quality of the resulting MCMC estimators.

Whereas the independence chain sampler can be improved by increasing the covariance matrices from the normal approximations of the conditional distributions of C_i and θ , the random walk algorithm, RW, can be improved by reducing them, i.e. multiplying the covariance matrices by a scalar c with $0 < c < 1$. As in the adaptive random walk algorithm, RW Sim, the acceptance rates in the RW algorithm can be increased by decreasing the covariance matrix of the random perturbations, Φ_θ and Φ_{C_i} . From experience, we know that multiplying Φ_θ and Φ_{C_i} by $2^{-x/5}$, where x is the number of dimensions of the parameter vectors, keeps the ratio of accepted proposals roughly between 1/3 and 2/3 for $x \leq 25$, the most efficient acceptance rates for this random walk algorithm. Without adjusting the covariance matrices of the random perturbations, the acceptance rates steadily drop with an increasing number of parameters. The three implementations of the MCMC algorithms used in the simulation study are summarized in Table 1.

Table 1. Summary of the implementation of the MCMC algorithms

	RW Sim (adaptive simultaneous random walk)	IC (independence chain)	RW (random walk)
Pre burn-in	RIGLS-3, adaptive sequences (12) with 20 batches of 125 iterations determining Φ_C and Φ_θ		
Burn-in	Fixed Φ_C and Φ_θ from pre burn-in	Every 10th iteration, new proposal distributions for C and θ from normal approximations (13) and (14) with Φ_{C_i} and Φ_θ steps (10) and (11)	Every 100th iteration updated Φ_{C_i} and Φ_θ from normal approximations, with Φ_{C_i} and Φ_θ times $2^{-0.2(\# \text{ par})}$, using steps (10) and (11)
Sampling sequence	Length: 10,000 iterations Fixed Φ_C and Φ_θ from pre burn-in	Length: 2,000 iterations Every 10th iteration, new proposal distributions as in burn-in	Length: 2,000 iterations Fixed Φ_{C_i} and Φ_θ from burn-in
Evaluation of random actor effects proposals	Length: 40,000 iterations For all n actors in a single Metropolis step	Length: 8,000 iterations Metropolis–Hastings step for each actor separately	Length: 4,000 iterations Metropolis–Hastings step for each actor separately

8. Simulation study

8.1. Design

In the simulation study, three different model specifications are used for networks with 20 and 40 actors. A network with 20 actors can be considered as small and a network with 40 actors as medium-sized. For each of the six combinations of model and network size, 1,000 data sets are simulated. The models are estimated with the three MCMC algorithms and the three IGLS algorithms.

Model 1 has parameters $\mu = -2$ and $\rho = 2$ with independent standard normal random effects and no covariates (this model is called 'empty'). Model 2 has one dyadic density covariate \mathbf{Z}_1 and one sender covariate \mathbf{X}_1 in addition to the parameters of Model 1. The dyadic density covariate with regression parameter 0.5 is a network (*net1*) generated according to Model 1. The sender covariate is the actor's rank number (1, . . . , n) with regression parameter 0.05. Model 3 has negatively correlated random effects, $\sigma_{AB} = -0.5$, with the sender variance, $\sigma_A^2 = 1.5$, twice as large as the receiver variance, $\sigma_B^2 = 0.75$, a receiver covariate \mathbf{X}_2 , two density covariates \mathbf{Z}_1 , and one reciprocity covariate \mathbf{Z}_2 . The receiver covariate is binary (0,1), drawn from a distribution with equal probabilities for both outcomes and has coefficient -0.1 . The first density covariate is *net1*, with coefficient 0.5. The second density covariate (*fc*) is constructed from an actor covariate with five discrete outcomes (1, 2, 3, 4, and 5) from a multinomial distribution with equal probabilities. The dyadic covariate *fc* contains the absolute differences of the outcomes of the multinomial distribution for all pairs of actors. Model 3 includes this absolute difference dyadic covariate (*fc*) as a density covariate with regression coefficient 0.2 and as a reciprocity covariate with coefficient 0.05.

8.2. Simulation results

We first discuss the results of the simulation study in terms of bias and variance for the three models separately for IGLS and MCMC. Then we compare both estimation methods by studying the mean squared errors and coverage rates of confidence and credibility intervals.

8.2.1. IGLS

Results of the IGLS simulations are shown in Table 2, confirming the findings of Rodríguez and Goldman (1995). The average parameter estimates are all biased towards zero and the magnitude of the bias is roughly proportional to the size of the parameters. Notice that compared to the networks with 20 actors, the bias is not reduced for networks with 40 actors. The standard errors, however, are reduced for larger samples.

Where in a linear model random effects can be provisionally set to zero because their contributions cancel, in the IGLS algorithms for non-linear models a different approximation is used for each observed value. It is unlikely that across all approximations the random effects will cancel out. This could well be the reason for the biased IGLS estimates.

The estimates for Model 2 show that a non-trivial number of the IGLS-3 estimates failed to converge, particularly for 40 actors. This is due to numerical overflow or the algorithm being unable to reach the convergence criterion. The linear approximation of the model is less accurate when the probabilities modelled deviate further from 0.5, causing more estimation problems. In particular with the multinomial formulation in (R)IGLS-3, the probabilities modelled are smaller than in IGLS-2 and therefore the deviation from a linear

Table 2. IGLS mean parameter estimates and mean standard errors over 1,000 replications of models 1, 2, and 3

	IGLS-2		IGLS-3		RIGLS-3		IGLS-2		IGLS-3		RIGLS-3	
	Mean	SE	Mean	SE	Mean	SE	Mean	SE	Mean	SE	Mean	SE
<i>Model 1</i>												
$\sigma_A^2 = 1$	0.809	0.255	20 actors		0.833	0.263	0.860	0.269	0.671	0.131	0.715	0.139
$\sigma_B^2 = 1$	0.807	0.254	0.832	0.263	0.860	0.269	0.680	0.132	0.722	0.140	0.732	0.141
$\sigma_{AB} = 0$	-0.102	0.183	-0.189	0.190	-0.165	0.193	-0.047	0.093	-0.149	0.100	-0.140	0.101
$\mu = -2$	-1.519	0.320	-1.612	0.308	-1.612	0.317	-1.498	0.197	-1.611	0.190	-1.611	0.192
$\rho = 2$	1.546	0.397	1.802	0.370	1.802	0.370	1.523	0.190	1.837	0.180	1.837	0.180
Converged runs	1,000		1,000		1,000		1,000		1,000		1,000	
<i>Model 2</i>												
$\sigma_A^2 = 1$	0.705	0.220	20 actors		0.740	0.236	0.830	0.256	0.610	0.118	0.670	0.137
$\sigma_B^2 = 1$	0.742	0.228	0.783	0.246	0.813	0.252	0.626	0.120	0.698	0.138	0.708	0.139
$\sigma_{AB} = 0$	-0.115	0.162	-0.179	0.173	-0.150	0.182	-0.053	0.084	-0.123	0.096	-0.116	0.098
$\mu = -2$	-1.527	0.504	-1.625	0.512	-1.608	0.536	-1.509	0.312	-1.638	0.321	-1.632	0.328
$\rho = 2$	1.559	0.345	1.781	0.352	1.781	0.351	1.527	0.173	1.824	0.185	1.825	0.184
$\gamma_1 = 0.05$	0.038	0.036	0.037	0.038	0.035	0.040	0.037	0.012	0.035	0.012	0.035	0.012
$\delta_1 = 0.5$	0.373	0.253	0.363	0.272	0.359	0.273	0.376	0.125	0.374	0.151	0.373	0.151
Converged runs	1,000		984		989		1,000		876		878	
<i>Model 3</i>												
$\sigma_A^2 = 1.5$	1.084	0.308	20 actors		1.015	0.291	1.061	0.301	0.958	0.173	0.929	0.168
$\sigma_B^2 = 0.75$	0.631	0.206	0.572	0.191	0.639	0.206	0.527	0.105	0.489	0.098	0.508	0.101
$\sigma_{AB} = -0.5$	-0.381	0.193	-0.272	0.174	-0.246	0.181	-0.327	0.104	-0.259	0.096	-0.253	0.097
$\mu = -2$	-1.510	0.435	-1.441	0.434	-1.482	0.449	-1.480	0.242	-1.418	0.242	-1.415	0.246
$\rho = 2$	1.480	0.596	1.258	0.557	1.261	0.557	1.475	0.289	1.290	0.272	1.290	0.272
$\gamma_2 = -0.1$	-0.073	0.378	-0.075	0.383	-0.095	0.404	-0.077	0.227	-0.078	0.229	-0.083	0.234
$\delta_1(\hat{c}) = 0.2$	0.161	0.171	0.165	0.165	0.164	0.166	0.155	0.081	0.155	0.079	0.155	0.079
$\delta_1(\text{net}) = 0.5$	0.382	0.254	0.394	0.257	0.389	0.260	0.375	0.123	0.384	0.123	0.383	0.123
$\delta_2(\hat{c}) = 0.05$	0.006	0.371	0.023	0.300	0.023	0.301	0.005	0.149	0.019	0.142	0.019	0.142
Converged runs	1,000		1,000		1,000		1,000		1,000		1,000	

model is more severe. The actor rank number with regression parameter γ_1 in Model 2 is a 'strong' covariate, even more so if it runs up to 40. The strong effect of this covariate will result in some estimated probabilities close to zero (or one).

Although still underestimated (in an absolute sense), in Models 1 and 2 the means of σ_A^2 , σ_B^2 , μ , and ρ , obtained with (R)IGLS-3, are slightly closer to the true values than the IGLS-2 means. In all models, the bias of the covariance parameter is larger for (R)IGLS-3. The coefficients of the covariates in Model 3 are slightly less negatively biased for (R)IGLS-3, whereas the other parameters show slightly more underestimation.

8.2.2. MCMC

Table 3 shows the average posterior means for the three MCMC algorithms, implemented as indicated in Table 1. In contrast to the IGLS estimation methods, the MCMC algorithms perform rather well with respect to bias, producing means of the

Table 3. MCMC average posterior means over 1,000 replications of Models 1, 2, and 3

	RW Sim	IC	RW	RW Sim	IC	RW
<i>Model 1</i>		20 actors			40 actors	
$\sigma_A^2 = 1$	1.019	1.003	1.011	1.002	0.990	0.998
$\sigma_B^2 = 1$	1.012	0.996	1.006	1.019	1.008	1.017
$\sigma_{AB} = 0$	0.029	0.076	0.030	-0.002	0.033	-0.001
$\mu = -2$	-1.968	-1.917	-1.967	-1.992	-1.938	-1.992
$\rho = 2$	1.927	1.868	1.926	1.988	1.942	1.985
Runs	1,000	1,000	1,000	1,000	1,000	1,000
Acceptance rate \mathbf{C}	0.36	0.29	0.59	0.35	0.29	0.60
Acceptance rate $\mathbf{\theta}$	0.35	0.27	0.60	0.35	0.26	0.60
<i>Model 2</i>		20 actors			40 actors	
$\sigma_A^2 = 1$	1.059	1.041	1.056	1.028	1.016	1.027
$\sigma_B^2 = 1$	1.068	1.053	1.065	1.036	1.025	1.034
$\sigma_{AB} = 0$	0.020	0.074	0.021	-0.002	0.031	-0.003
$\mu = -2$	-1.968	-1.933	-1.965	-1.991	-1.960	-1.991
$\rho = 2$	1.955	1.895	1.953	1.996	1.954	1.995
$\gamma_1 = 0.05$	0.049	0.049	0.049	0.049	0.049	0.050
$\delta_1 = 0.5$	0.499	0.504	0.500	0.509	0.510	0.508
Runs	1,000	1,000	1,000	981	1,000	1,000
Acceptance rate \mathbf{C}	0.36	0.29	0.60	0.35	0.29	0.60
Acceptance rate $\mathbf{\theta}$	0.35	0.12	0.49	0.35	0.11	0.49
<i>Model 3</i>		20 actors			40 actors	
$\sigma_A^2 = 1.5$	1.607	1.579	1.607	1.522	1.508	1.520
$\sigma_B^2 = 0.75$	0.852	0.826	0.851	0.791	0.775	0.788
$\sigma_{AB} = -0.5$	-0.463	-0.392	-0.464	-0.496	-0.462	-0.494
$\mu = -2$	-2.010	-1.971	-2.007	-1.994	-1.968	-1.998
$\rho = 2$	1.902	1.815	1.902	1.987	1.943	1.981
$\gamma_2 = -0.1$	-0.103	-0.099	-0.102	-0.105	-0.102	-0.101
$\delta_1(fc) = 0.2$	0.213	0.214	0.213	0.203	0.203	0.202
$\delta_1(netI) = 0.5$	0.515	0.524	0.517	0.504	0.506	0.504
$\delta_2(fc) = 0.05$	0.028	0.031	0.028	0.020	0.022	0.023
Runs	1,000	1,000	1,000	1,000	1,000	1,000
Acceptance rate \mathbf{C}	0.36	0.29	0.60	0.35	0.29	0.60
Acceptance rate $\mathbf{\theta}$	0.35	0.07	0.46	0.35	0.07	0.46

fixed parameters close to the true values. The variances of the random effects are somewhat overestimated, but as the number of actors increases, the MCMC estimates improve. For Model 2 with 40 actors, due to non-convergence of the RIGLS-3 algorithm, the RW Sim algorithm failed to converge for 19 simulated data sets (see also Table 1). These failures still occurred after reducing the number of RIGLS-3 iterations to two. (R)IGLS-3 was applied instead of IGLS-2 because both the RW Sim algorithm and (R)IGLS-3 use the multinomial data representation as in (4).

The acceptance rates of the RW Sim algorithm are close to the required 33%, while results in Table 3 show that with increasing dimensions of θ , the acceptance rates of the proposals drop for the IC and RW algorithms. Assuming that the random walk algorithm is most efficient roughly in the range of acceptance rates between 1/3 and 2/3, the RW Sim and RW algorithms perform reasonably well. Because the efficiency of an independence chain decreases with any decrease in the acceptance rates, the IC algorithm clearly becomes increasingly inefficient with more parameters in θ . A convenient way to inspect whether estimates in Table 3 converge to similar distributions is by comparing the average posterior standard deviations with the standard deviations of the posterior means, displayed in Table 4. These should be approximately equal, which is true for most estimates.

Table 4. MCMC average posterior standard deviations and standard deviations of the posterior means over 1,000 replications of Models 1, 2, and 3

	RW Sim		IC		RW		RW Sim		IC		RW	
	Mean SD	SD Mean	Mean SD	SD Mean	Mean SD	SD Mean	Mean SD	SD Mean	Mean SD	SD Mean	Mean SD	SD Mean
<i>Model 1</i>	20 actors						40 actors					
σ_A^2	0.535	0.507	0.527	0.491	0.529	0.495	0.299	0.282	0.300	0.280	0.299	0.279
σ_B^2	0.532	0.456	0.525	0.447	0.528	0.453	0.305	0.298	0.305	0.297	0.305	0.298
σ_{AB}	0.390	0.350	0.385	0.336	0.387	0.348	0.216	0.214	0.217	0.211	0.216	0.214
μ	0.399	0.391	0.387	0.380	0.391	0.390	0.247	0.263	0.242	0.246	0.243	0.259
ρ	0.514	0.504	0.517	0.496	0.511	0.499	0.251	0.253	0.256	0.251	0.251	0.256
<i>Model 2</i>	20 actors						40 actors					
σ_A^2	0.537	0.503	0.531	0.490	0.533	0.502	0.299	0.296	0.299	0.292	0.299	0.294
σ_B^2	0.529	0.486	0.525	0.473	0.525	0.478	0.269	0.291	0.297	0.291	0.297	0.291
σ_{AB}	0.394	0.364	0.390	0.349	0.392	0.363	0.215	0.210	0.216	0.212	0.215	0.213
μ	0.629	0.658	0.616	0.652	0.616	0.666	0.397	0.418	0.391	0.412	0.391	0.424
ρ	0.464	0.467	0.471	0.459	0.463	0.472	0.230	0.229	0.236	0.227	0.230	0.231
γ_1	0.044	0.046	0.043	0.046	0.043	0.047	0.014	0.016	0.014	0.015	0.014	0.016
δ_1	0.300	0.307	0.309	0.309	0.299	0.308	0.148	0.151	0.153	0.151	0.148	0.151
<i>Model 3</i>	20 actors						40 actors					
σ_A^2	0.765	0.772	0.762	0.761	0.764	0.769	0.422	0.410	0.423	0.413	0.422	0.410
σ_B^2	0.455	0.417	0.450	0.394	0.454	0.415	0.239	0.229	0.239	0.227	0.239	0.228
σ_{AB}	0.461	0.446	0.457	0.425	0.462	0.445	0.249	0.243	0.249	0.242	0.249	0.242
μ	0.522	0.519	0.535	0.525	0.515	0.523	0.291	0.296	0.296	0.296	0.285	0.302
ρ	0.693	0.706	0.709	0.716	0.689	0.713	0.338	0.344	0.347	0.348	0.334	0.349
γ_2	0.439	0.457	0.447	0.457	0.433	0.458	0.269	0.280	0.273	0.280	0.266	0.283
δ_1	0.184	0.188	0.190	0.194	0.182	0.189	0.088	0.087	0.091	0.089	0.087	0.088
δ_1	0.299	0.318	0.315	0.322	0.298	0.318	0.143	0.148	0.151	0.149	0.143	0.149
δ_2	0.327	0.327	0.340	0.340	0.325	0.331	0.155	0.156	0.162	0.159	0.154	0.158

Even though the means of the MCMC estimates in Table 3 appear to be nearly unbiased compared to the means of the IGLS estimates in Table 2, the standard deviations in the MCMC samples in Table 4 are sometimes more than twice as large as the IGLS standard errors in Table 2. This raises the question as to which of the estimation methods produces the best estimates in terms of mean squared error (MSE, the mean of the squared difference between the estimates and the true parameter values). The MSE is equal to the sum of the squared bias, i.e. the difference between the average parameter estimate and the true value, and the variance of the parameter estimate. The root mean squared errors (RMSEs) are displayed in Table 5. Observe that the posterior means from the MCMC algorithms have a small bias and thus the RMSEs are almost equal to the standard deviations of the estimates.

In Table 5, the IGLS RMSEs for networks with 20 actors are either similar to or smaller than the MCMC RMSEs. Even though the IGLS parameter estimates are biased, small RMSEs are obtained because of the small standard errors.

For the networks with 40 actors, MCMC RMSEs are smaller except for σ_{AB} , and for γ_2 and δ_2 in Model 3. Since the MCMC fixed parameters are virtually unbiased and the IGLS bias is roughly proportional to the size of the parameters and insensitive to network size, the effect of the bias of the IGLS estimates on the RMSEs becomes relatively larger for larger networks.

Considering the RMSEs, the IGLS estimates can be preferred for networks of size 20 because these generally are at least as close to the true values as the MCMC estimates. Conversely, for networks with 40 actors, the MCMC estimates are preferred.

Another important way to evaluate the estimation methods is by looking at the confidence and credibility intervals of the parameter estimates. Confidence intervals for the IGLS estimates can be constructed around the IGLS parameter estimates covering the central 95% and 99% of a normal distribution with the standard deviations equal to the IGLS standard errors. Credibility intervals from the Bayesian MCMC results are defined by the central 95th or 99th percentiles of the posterior. This procedure does not make assumptions about the shape of the posterior distribution. In Figures 3 and 4 the coverage percentages are shown.

The coverage rates of the IGLS estimates in Figures 3 and 4 are mostly far below the nominal percentages, except for some of the regression parameters. Whereas in the RMSEs the contributions of the bias and the variance of the IGLS estimates can compensate for each other, for the coverage rates of confidence intervals these tend to reinforce each other negatively. In IGLS, with increasing sample size the bias remains similar, while the standard errors decrease. Therefore, in networks with 40 actors, the IGLS coverage rates are even worse than for those with 20 actors.

Most MCMC coverage percentages appear to be satisfactory. In general, better coverage rates are obtained with larger networks. Coverage rates for μ and the sender and receiver covariates are relatively low when compared with the desired rates. These terms always appear together in the p_2 probability, (4) with (5), and are therefore the most difficult parameters to estimate. It should be noted, however, that μ gives a baseline for the probability of a tie and is seldom of substantive interest. Relatively low coverage rates are also observed for σ_{AB} in Model 3 for 20 actors.

A convenient way of testing MCMC results is just to focus on the estimated posterior means and standard errors instead of coverage rates of credibility intervals. Assuming that the parameters are normally distributed, credibility intervals can be constructed based on estimated posterior means and standard errors of the MCMC samples. Figures 5 and 6 contain coverage rates of these credibility intervals.

Table 5. Root mean squared errors (RMSEs) for IGLS and MCMC algorithms for Models 1, 2, and 3

	IGLS-2	IGLS-3	RIGLS-3	RW Sim	IC	RW	IGLS-2	IGLS-3	RIGLS-3	RW Sim	IC	RW
Model 1												
$\sigma_A^2 = 1$	0.341	0.316	0.315	0.508	0.491	0.494	0.361	0.322	0.316	0.281	0.279	0.279
$\sigma_B^2 = 1$	0.324	0.305	0.302	0.456	0.446	0.453	0.356	0.319	0.313	0.298	0.298	0.298
$\sigma_{AB} = 0$	0.266	0.230	0.212	0.351	0.344	0.349	0.138	0.167	0.158	0.214	0.212	0.214
$\mu = -2$	0.567	0.490	0.490	0.392	0.389	0.391	0.540	0.436	0.436	0.263	0.255	0.259
$\rho = 2$	0.618	0.499	0.499	0.509	0.513	0.504	0.518	0.305	0.305	0.253	0.257	0.257
Model 2												
$\sigma_A^2 = 1$	0.370	0.342	0.300	0.506	0.492	0.505	0.411	0.362	0.336	0.297	0.292	0.295
$\sigma_B^2 = 1$	0.336	0.310	0.298	0.491	0.475	0.482	0.395	0.338	0.332	0.293	0.292	0.293
$\sigma_{AB} = 0$	0.243	0.217	0.202	0.363	0.356	0.363	0.122	0.145	0.138	0.210	0.212	0.214
$\mu = -2$	0.687	0.627	0.638	0.659	0.656	0.667	0.579	0.472	0.476	0.418	0.414	0.424
$\rho = 2$	0.578	0.479	0.480	0.471	0.470	0.474	0.506	0.307	0.307	0.228	0.232	0.230
$\gamma_1 = 0.05$	0.032	0.032	0.032	0.045	0.045	0.045	0.017	0.018	0.018	0.015	0.015	0.016
$\delta_1 = 0.5$	0.263	0.272	0.274	0.307	0.310	0.308	0.167	0.182	0.182	0.152	0.152	0.152
Model 3												
$\sigma_A^2 = 1.5$	0.522	0.554	0.522	0.778	0.765	0.776	0.571	0.608	0.593	0.411	0.412	0.411
$\sigma_B^2 = 0.75$	0.251	0.253	0.230	0.429	0.401	0.427	0.253	0.279	0.263	0.232	0.228	0.230
$\sigma_{AB} = -0.5$	0.281	0.255	0.285	0.447	0.438	0.446	0.221	0.249	0.255	0.243	0.245	0.243
$\mu = -2$	0.623	0.695	0.706	0.520	0.526	0.522	0.569	0.632	0.636	0.297	0.298	0.302
$\rho = 2$	0.760	0.995	0.994	0.713	0.740	0.720	0.592	0.792	0.792	0.345	0.352	0.349
$\gamma_2 = -0.1$	0.339	0.349	0.349	0.456	0.457	0.458	0.207	0.212	0.212	0.281	0.279	0.283
$\delta_1(f_c) = 0.2$	0.155	0.167	0.167	0.190	0.195	0.190	0.084	0.089	0.089	0.089	0.089	0.089
$\delta_1(\text{netl}) = 0.5$	0.263	0.272	0.274	0.318	0.322	0.318	0.167	0.164	0.164	0.148	0.148	0.148
$\delta_2(f_c) = 0.05$	0.266	0.321	0.321	0.329	0.341	0.332	0.130	0.155	0.155	0.158	0.161	0.161

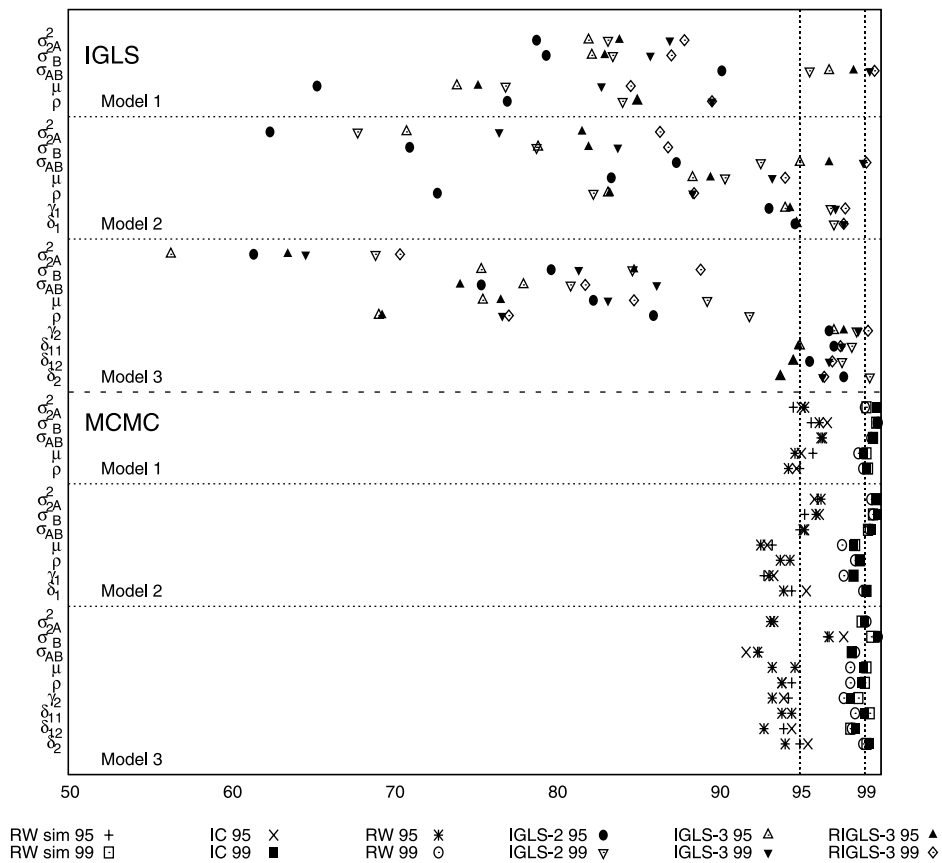


Figure 3. Coverage rates for 95% and 99% intervals for the IGLS and MCMC algorithms for networks with 20 actors.

In Figures 5 and 6, the coverage rates for the fixed parameters are slightly less accurate than the coverage rates for the credibility intervals based on the MCMC samples. (Obviously, the variance parameters cannot be expected to be normally distributed here and are only added here for illustrative purposes.) Thus, for the MCMC results, it is preferred to report the credibility intervals based on MCMC samples instead of the posterior means and standard errors. However, for networks with many actors, the results are expected to be more comparable.

Ultimately, the estimation method preferred depends on the purpose of the estimation. For networks with 40 actors, MCMC is preferred because of the small RMSEs and accurate credibility intervals. For networks with 20 actors, generally the estimates with the smallest RMSEs are obtained with IGLS, while for credibility intervals MCMC is preferred.

Small RMSEs indicate that an estimate can be expected to be close to the true value. Thus, for networks with 20 actors, the IGLS point estimates are preferred, while for hypothesis testing MCMC estimates are preferred. As the number of actors in a network increases above 20, the advantage of the smallest RMSEs is expected to shift quickly to MCMC, because the number of observed ties increases almost quadratically with the number of actors.

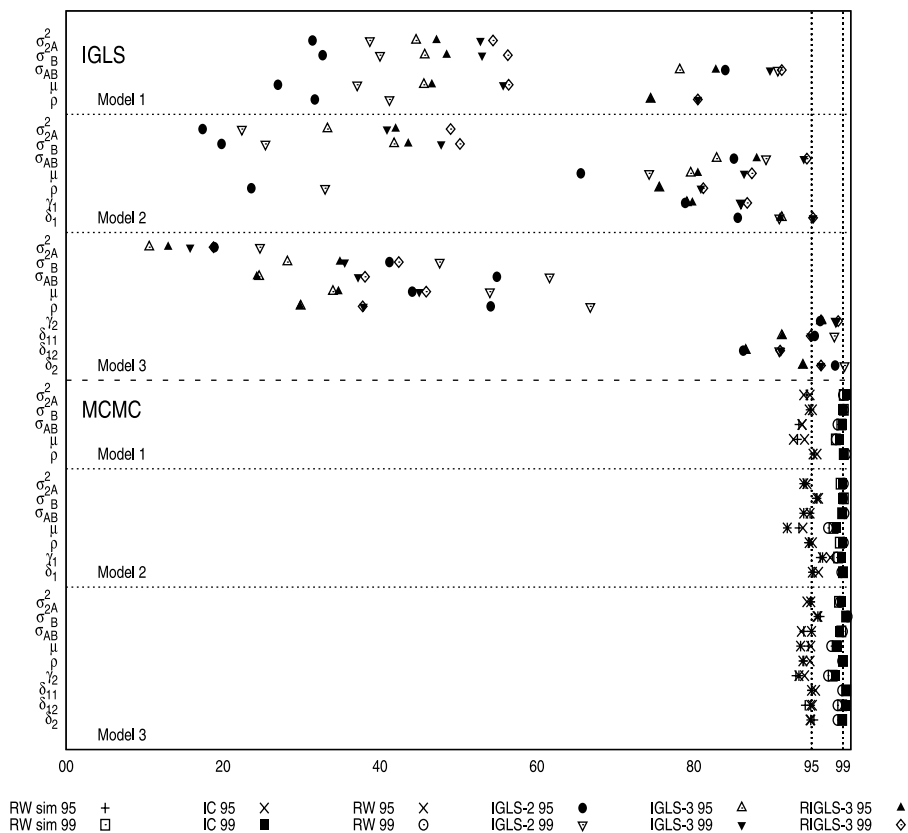


Figure 4. Coverage rates for 95% and 99% intervals for the IGLS and MCMC algorithms for networks with 40 actors.

9. Concluding remarks

Until recently, only the IGLS-2 algorithm was available for the p_2 model. The availability of the MCMC estimation algorithms for the p_2 model provides a large improvement, resulting in largely unbiased estimates with good coverage rates. The MCMC estimation methods, however, are not without drawbacks. The MCMC estimates are less efficient (have larger RMSEs) than the IGLS estimates for networks with 20 actors, and the MCMC algorithms are more time-consuming than the IGLS algorithms for the network sizes used here (taking minutes instead of seconds). Interestingly, with large networks (150 actors or more), the relative advantage in speed of computation of the IGLS algorithm is seriously reduced (both methods taking hours).

Comparing the MCMC procedures, the adaptive random walk algorithm, RW Sim, has the disadvantage that it requires IGLS estimates to obtain starting values. This means that for any further development of the p_2 model, a new implementation of the IGLS algorithm (or other approximating algorithm) is needed as well. The independence chain, IC, does not have this disadvantage, but this algorithm has an acceptance rate that drops relatively quickly with the number of dimensions of the parameter vector, as can be seen in Table 3. The random walk algorithm, RW, with perturbations based on the normal approximations of the fixed and random

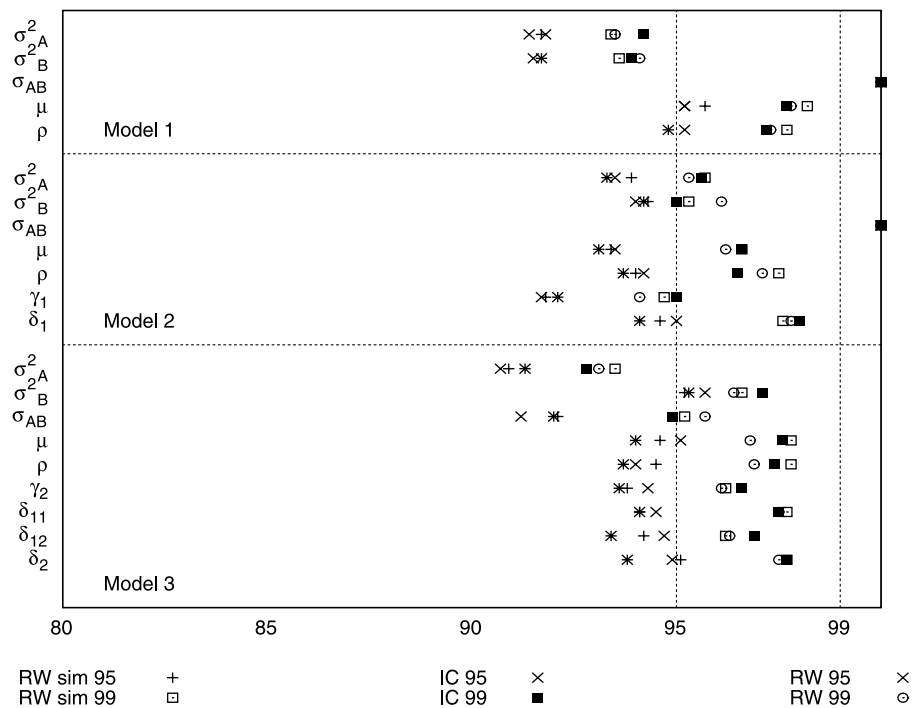


Figure 5. Coverage rates for 95% and 99% credibility intervals from the MCMC posterior means and standard errors for networks with 20 actors.

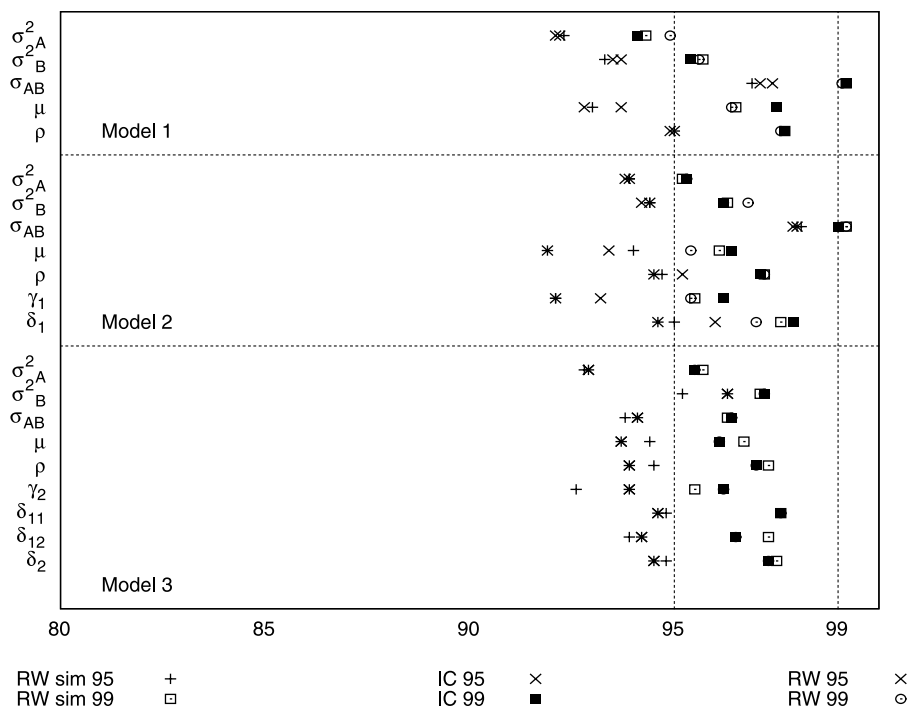


Figure 6. Coverage rates for 95% and 99% credibility intervals from the MCMC posterior means and standard errors for networks with 40 actors.

parameters performs much better in this sense. Furthermore, taking into account that the RW Sim algorithm needs more iterations to reach a converged solution, the RW and the RW Sim algorithms are comparable in terms of speed, while the IC algorithm is much slower. The p_2 model with the (R)IGLS algorithms, and the RW Sim and RW algorithms has been made available in StOCNET (Boer *et al.*, 2006), an open software system for the statistical analysis of social network data. Because of its adaptability and speed, the RW algorithm has been chosen as the default algorithm in StOCNET.

For all MCMC algorithms in the simulation study, no really good estimates for μ are obtained when covariates for the sender and receiver effects are present. Because these parameters co-occur in the p_2 probability, (4) with (5), they are inherently dependent on each other. This may also explain why the independence chain algorithm did not clearly outperform the RW algorithm in terms of efficiency (by requiring smaller samples). For the same reason, the accept-reject Metropolis-Hastings (ARMH) algorithm (Tierney, 1994) did not increase the efficiency of the MCMC algorithms. The ARMH algorithm starts with an accept-reject step, where proposals are accepted with a probability inversely related to the degree of domination of an importance function, which includes a proportionality constant for the conditional distribution sampled. Because of the dependence of the parameter sets in the p_2 model, the proportionality constant had to be set so small that the importance function hardly ever dominated the conditional probability function, which eliminates any efficiency gain from the ARMH algorithm.

Further developments of the p_2 model are in progress. They deal with modelling multiple network observations, such as friendship ties observed in multiple classrooms, as well as dealing with multiple types of relations within the same set of actors.

References

- Boer, P., Huisman, M., Snijders, T. A. B., Steglich, C., Wichers, L., & Zeggelink, E. P. H. (2006). *StOCNET user's manual*, Version 1.7. Groningen: ProGAMMA/ICS. <http://stat.gamma.rug.nl/stocnet>.
- Box, G. E. P., & Tiao, G. C. (1973). *Bayesian inference in statistical analysis*. Reading, MA: Addison-Wesley.
- Browne, W. J. (1998). *Applying MCMC methods to multi-level models*. PhD thesis, University of Bath.
- Browne, W. J., & Draper, D. (2000). Implementation and performance issues in the Bayesian and likelihood fitting of multilevel models. *Computational Statistics*, 15, 391–420.
- Browne, W. J., & Draper, D. (2006). A comparison of Bayesian and likelihood-based methods for fitting multilevel models (with discussion). *Bayesian Analysis*, 1, 473–550.
- Chib, S., & Greenberg, E. (1995). Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49, 327–335.
- Fienberg, S. E., & Wasserman, S. (1981). Comment on Holland and Leinhardt (1981). *Journal of the American Statistical Association*, 77, 54–57.
- Gelman, A., Roberts, G. O., & Gilks, W. R. (1995). Efficient Metropolis jumping rules. In J. M. Bernardo, J. O. Berger, A. P. David, & A. F. M. Smith (Eds.), *Bayesian statistics 5* (pp. 599–607). Oxford: Oxford University Press.
- Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721–741.
- Gibbons, R. D., & Hedeker, D. (1997). Random effects probit and logistic regression models for three-level data. *Biometrics*, 53, 1527–1537.

- Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. (1996). Introducing Markov chain Monte Carlo. In W. R. Gilks, S. Richardson, & D. J. Spiegelhalter (Eds.), *Markov chain Monte Carlo in practice* (pp. 1–19). London: Chapman & Hall.
- Gill, P. S., & Swartz, T. B. (2004). Bayesian analysis of directed graphs data with applications to social networks. *Journal of the Royal Statistical Society, Series C*, 53, 249–260.
- Goldstein, H. (2003). *Multilevel statistical models* (3rd ed.). London: Edward Arnold.
- Goldstein, H., & Rasbash, J. (1992). Efficient computational procedures for the estimation of parameters in the multilevel models based on iterative generalized least squares. *Computational Statistics and Data Analysis*, 13, 63–71.
- Goldstein, H., & Rasbash, J. (1996). Improved approximations for multilevel models with binary responses. *Journal of the Royal Statistical Society, Series A*, 159, 505–513.
- Holland, P. W., & Leinhardt, S. (1981). An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, 77, 33–50.
- Johnson, N. L., & Kotz, S. (1972). *Distributions in statistics: Continuous multivariate distributions*. New York: Wiley.
- McCullagh, P., & Nelder, J. A. (1989). *Generalized linear models*. London: Chapman & Hall.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equations of state calculations by fast computing machine. *Journal of Chemical Physics*, 21, 1087–1092.
- Odell, P. L., & Feiveson, A. H. (1966). A numerical procedure to generate a sample covariance matrix. *Journal of the American Statistical Association*, 61, 199–203.
- Rodríguez, G., & Goldman, N. (1995). An assessment of estimation procedures for multilevel models with binary responses. *Journal of the Royal Statistical Society, Series A*, 158, 73–89.
- Rodríguez, G., & Goldman, N. (2001). Improved estimation procedures for multi-level models with binary response: A case study. *Journal of the Royal Statistical Society, Series A*, 164, 339–355.
- Shun, Z. (1997). Another look at the Salamander mating data: A modified Laplace approximation approach. *Journal of the American Statistical Association*, 92, 341–349.
- Shun, Z., & McCullagh, P. (1995). Laplace approximation of high dimensional integrals. *Journal of the Royal Statistical Society, Series B*, 57, 749–760.
- Skrondal, A., & Rabe-Hesketh, S. (2004). *Generalized latent variable modeling*. Boca Raton, FL: Chapman & Hall/CRC.
- Tierney, L. (1994). Markov chains for exploring posterior distributions (with discussion). *Annals of Statistics*, 22, 1701–1762.
- Van Duijn, M. A. J., Snijders, T. A. B., & Zijlstra, B. J. H. (2004). p_2 : A random effects model with covariates for directed graphs. *Statistica Neerlandica*, 58, 234–254.