Sampling Strategies for Fast Updating of Gaussian Markov Random Fields

Supplementary Material

D. Andrew Brown*
School of Mathematical and Statistical Sciences, Clemson University
and
Christopher S. McMahan
School of Mathematical and Statistical Sciences, Clemson University
and
Stella Watson Self
School of Mathematical and Statistical Sciences, Clemson University

Here the reader may find additional material, including more details about the binomial regression model for the election data, as well as Supplementary Figures discussed in the main text.

1 SUPPLEMENTARY ALGORITHMS

Input: Mean factor $\boldsymbol{b}$, precision matrix $\boldsymbol{Q}_p$.
Output: Draw $\boldsymbol{x}$ from a $N(\boldsymbol{Q}_p^{-1}\boldsymbol{b}, \boldsymbol{Q}_p^{-1})$ distribution.

1 Find the Cholesky factor $\boldsymbol{Q}_p = \boldsymbol{L}\boldsymbol{L}^T$
2 Solve $\boldsymbol{L}\boldsymbol{w} = \boldsymbol{b}$
3 Solve $\boldsymbol{L}^T\boldsymbol{\mu} = \boldsymbol{w}$
4 Sample $\boldsymbol{z} \sim N(\boldsymbol{0}, \boldsymbol{I})$
5 Solve $\boldsymbol{L}^T\boldsymbol{v} = \boldsymbol{z}$
6 Compute $\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{v}$
7 **Return** $\boldsymbol{x}$

Algorithm 1: Sampling from a typical GMRF-based full conditional encountered in block Gibbs sampling (Rue and Held, 2005).

*This material is based upon work partially supported by the National Science Foundation (NSF) under Grant DMS-1127974 to the Statistical and Applied Mathematical Sciences Institute. DAB is partially supported by NSF Grants CMMI-1563435, EEC-1744497 and OIA-1826715. CSM is partially supported

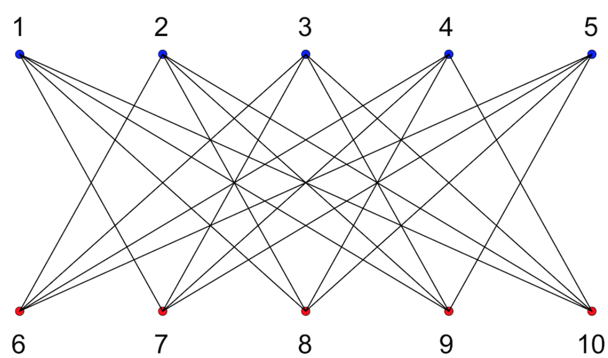
Input: MRF graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.
Output: k-coloring partition $\{\mathcal{A}_1, \mathcal{A}_2, \dots \mathcal{A}_k\}$, for some k.

```
Set j = 1 and A_0 = ∅
while V \ ∪_{l=0}^{j-1} A_l ≠ ∅ do
    I_j ← V \ ∪_{l=0}^{j-1} A_l
    A_j ← ∅
    while |I_j| > 0 do
        i ← min I_j
        A_j ← A_j ∪ {i}
        I_j ← I_j \ ({i} ∪ N(i))
    end
    j ← j + 1
end
k ← j − 1
Return {A_1, ..., A_k}
```

Algorithm 2: Greedy algorithm for $k-$coloring the nodes of an MRF graph.

1.1 Sensitivity of the Greedy Algorithm to Vertex Ordering

The following is an example to illustrate the sensitivity of the greedy algorithm to the ordering of the graph vertices.



For this graph, the greedy algorithm proceeds as follows:

1. Set $j = 1$ and $\mathcal{A}_0 = \emptyset$.
2. $\mathcal{V} \setminus \cup_{l=0}^{0} \mathcal{A}_l = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.
   (a) $\mathcal{I}_1 \leftarrow \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.
   (b) $\mathcal{A}_1 \leftarrow \emptyset$.
   (c) $|I_1| = 10$.

by National Institutes of Health Grant R01 AI121351 and NSF grant OIA-1826715.

i. $i \leftarrow 1$.

ii. $\mathcal{A}_1 \leftarrow \{1\}$.

iii. $\mathcal{I}_1 \leftarrow \{2, 3, 4, 5, 6\}$

(d) $|\mathcal{I}_1| = 5$.

i. $i \leftarrow 2$.

ii. $\mathcal{A}_1 \leftarrow \{1, 2\}$.

iii. $\mathcal{I}_1 \leftarrow \{3, 4, 5\}$

(e) $|\mathcal{I}_1| = 3$.

i. $i \leftarrow 3$.

ii. $\mathcal{A}_1 \leftarrow \{1, 2, 3\}$.

iii. $\mathcal{I}_1 \leftarrow \{4, 5\}$

(f) $|\mathcal{I}_1| = 2$.

i. $i \leftarrow 4$.

ii. $\mathcal{A}_1 \leftarrow \{1, 2, 3, 4\}$.

iii. $\mathcal{I}_1 \leftarrow \{5\}$

(g) $|\mathcal{I}_1| = 1$.

i. $i \leftarrow 5$.

ii. $\mathcal{A}_1 \leftarrow \{1, 2, 3, 4, 5\}$.

iii. $\mathcal{I}_1 \leftarrow \emptyset$

(h) $|\mathcal{I}_1| = 0$.

(i) $j = 2$.

3. $\mathcal{V} \setminus \cup_{l=0}^{1} \mathcal{A}_l = \{6, 7, 8, 9, 10\}$.

(a) $\mathcal{I}_2 = \{6, 7, 8, 9, 10\}$.

(b) $\mathcal{A}_2 = \emptyset$.

(c) $|I_2| = 5$.

i. $i = 6$.

ii. $\mathcal{A}_2 = \{6\}$.

iii. $\mathcal{I}_2 = \{7, 8, 9, 10\}$

(d) $|I_2| = 4$.

i. $i = 7$.

ii. $\mathcal{A}_2 = \{6, 7\}$.

iii. $\mathcal{I}_2 = \{8, 9, 10\}$

(e) $|I_2| = 3$.

i. $i = 8$.

ii. $\mathcal{A}_2 = \{6, 7, 8\}$.

iii. $\mathcal{I}_2 = \{9, 10\}$

(f) $|I_2| = 2$.

i. $i = 9$.

ii. $\mathcal{A}_2 = \{6, 7, 8, 9\}$.

iii. $\mathcal{I}_2 = \{10\}$

(g) $|I_2| = 1$.

i. $i = 10$.

ii. $\mathcal{A}_2 = \{6, 7, 8, 9, 10\}$.

iii. $\mathcal{I}_2 = \emptyset$

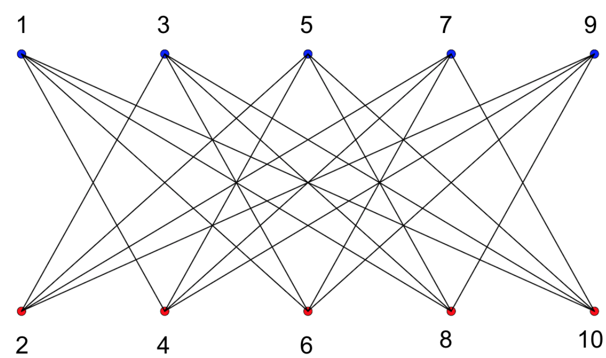
(h) $j = 3$

4. $\mathcal{V} \setminus \cup_{l=0}^{1} \mathcal{A}_l = \emptyset$.

5. $k = 2$

6. Return $\{\{1, 2, 3, 4, 5\}, \{6, 7, 8, 9, 10\}\}$.

We see that in this case, the greedy algorithm returns the optimal coloring. However, suppose that for the same graph, the vertices were ordered differently, as depicted below.



The greedy algorithm proceeds as follows:

1. Set $j = 1$ and $\mathcal{A}_0 = \emptyset$.

2. $\mathcal{V} \setminus \cup_{l=0}^{0} \mathcal{A}_l = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.

(a) $\mathcal{I}_1 \leftarrow \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.

(b) $\mathcal{A}_1 \leftarrow \emptyset$.

(c) $|I_1| = 10$.

i. $i \leftarrow 1$.

ii. $\mathcal{A}_1 \leftarrow \{1\}$.

iii. $\mathcal{I}_1 \leftarrow \{2, 3, 5, 7, 9\}$.

(d) $|I_1| = 5$.

    i. $i \leftarrow 2$.
    ii. $\mathcal{A}_1 \leftarrow \{1, 2\}$.
    iii. $\mathcal{I}_1 \leftarrow \emptyset$.
  (e) $j = 2$.

3. $\mathcal{V} \setminus \cup_{l=0}^{1} \mathcal{A}_l = \{3, 4, 5, 6, 7, 8, 9, 10\}$.
  (a) $\mathcal{I}_2 \leftarrow \{3, 4, 5, 6, 7, 8, 9, 10\}$. $\mathcal{A}_2 \leftarrow \emptyset$.
  (b) $|I_2| = 8$.
    i. $i \leftarrow 3$.
    ii. $\mathcal{A}_2 \leftarrow \{3\}$.
    iii. $\mathcal{I}_2 \leftarrow \{4, 5, 7, 9\}$.
  (c) $|I_2| = 4$.
    i. $i \leftarrow 4$.
    ii. $\mathcal{A}_2 \leftarrow \{3, 4\}$.
    iii. $\mathcal{I}_2 \leftarrow \emptyset$.
  (d) j = 3.

4. $\mathcal{V} \setminus \cup_{l=0}^{2} \mathcal{A}_l = \{5, 6, 7, 8, 9, 10\}$.
  (a) $\mathcal{I}_3 \leftarrow \{5, 6, 7, 8, 9, 10\}$. $\mathcal{A}_3 \leftarrow \emptyset$.
  (b) $|I_3| = 6$.
    i. $i \leftarrow 5$.
    ii. $\mathcal{A}_3 \leftarrow \{5\}$.
    iii. $\mathcal{I}_3 \leftarrow \{6, 7, 9\}$.
  (c) $|I_3| = 3$.
    i. $i \leftarrow 6$.
    ii. $\mathcal{A}_3 \leftarrow \{5, 6\}$.
    iii. $\mathcal{I}_3 \leftarrow \emptyset$.
  (d) j = 4.

5. $\mathcal{V} \setminus \cup_{l=0}^{2} \mathcal{A}_l = \{7, 8, 9, 10\}$.
  (a) $\mathcal{I}_4 \leftarrow \{7, 8, 9, 10\}$. $\mathcal{A}_4 \leftarrow \emptyset$.
  (b) $|I_4| = 4$.
    i. $i \leftarrow 7$.
    ii. $\mathcal{A}_4 \leftarrow \{7\}$.
    iii. $\mathcal{I}_4 \leftarrow \{8, 9\}$.
  (c) $|I_4| = 1$.

i. $i \leftarrow 8$.

ii. $\mathcal{A}_4 \leftarrow \{7, 8\}$.

iii. $\mathcal{I}_4 \leftarrow \emptyset$.

(d) j = 5.

6. $\mathcal{V} \setminus \cup_{l=0}^{2} \mathcal{A}_l = \{9, 10\}$.

(a) $\mathcal{I}_5 \leftarrow \{9, 10\}$.

(b) $\mathcal{A}_5 \leftarrow \emptyset$.

(c) $|I_5| = 2$.

i. $i \leftarrow 9$.

ii. $\mathcal{A}_5 \leftarrow \{9\}$.

iii. $\mathcal{I}_5 \leftarrow \{10\}$.

(d) $|I_5| = 1$.

i. $i \leftarrow 10$.

ii. $\mathcal{A}_5 \leftarrow \{9, 10\}$.

iii. $\mathcal{I}_5 \leftarrow \emptyset$.

(e) j = 6.

7. $k \leftarrow 5$.

8. Return $\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}, \{9, 10\}\}$.

Under this vertex labeling, the greedy algorithm returns a coloring consisting of 5 colors, over twice as many as the optimal solution. The sensitivity was recognized by Culberson (1992), who proposes iterated versions that repeatedly apply the greedy algorithm to permutations of the vertices, leading to solutions that are closer to optimal.

2 SUPPLEMENTARY FIGURES

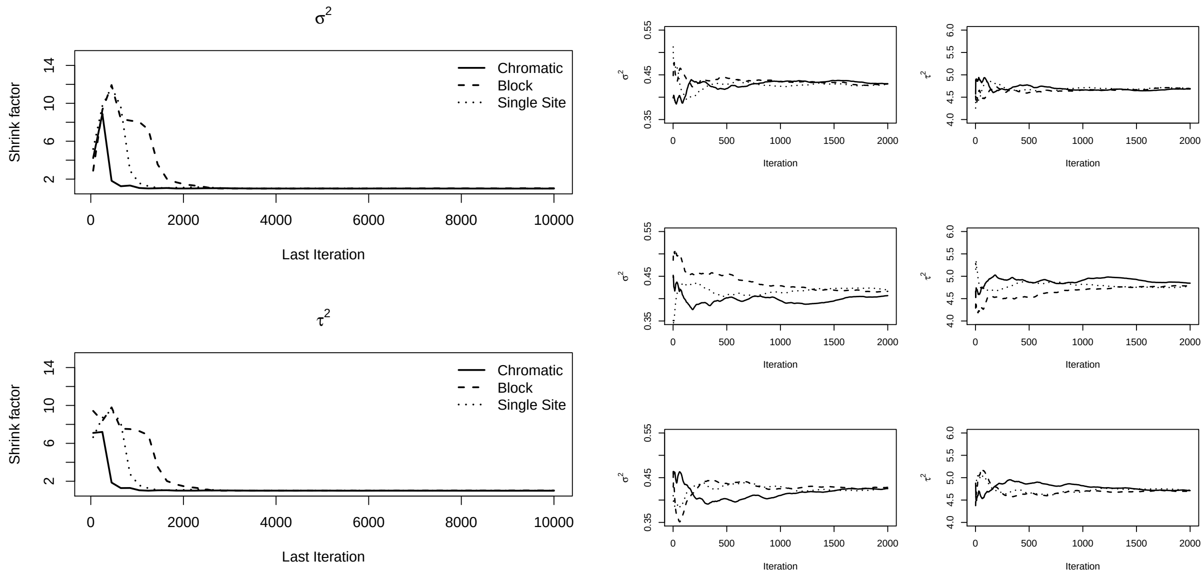


Figure 1: Left panel: Gelman plots (Brooks and Gelman, 1998) of the potential scale reduction factors versus chain length for the 50×50 *regular array example. Right panel: Cumulative averages* $\widehat{\sigma^2}^{(k)}$ *and* $\widehat{\tau^2}^{(k)}$, $k = 1, \ldots, 2,000$ *calculated from three independent chains. In the right panel, the top, middle, and bottom rows correspond to chromatic, block, and single-site sampling, respectively.*

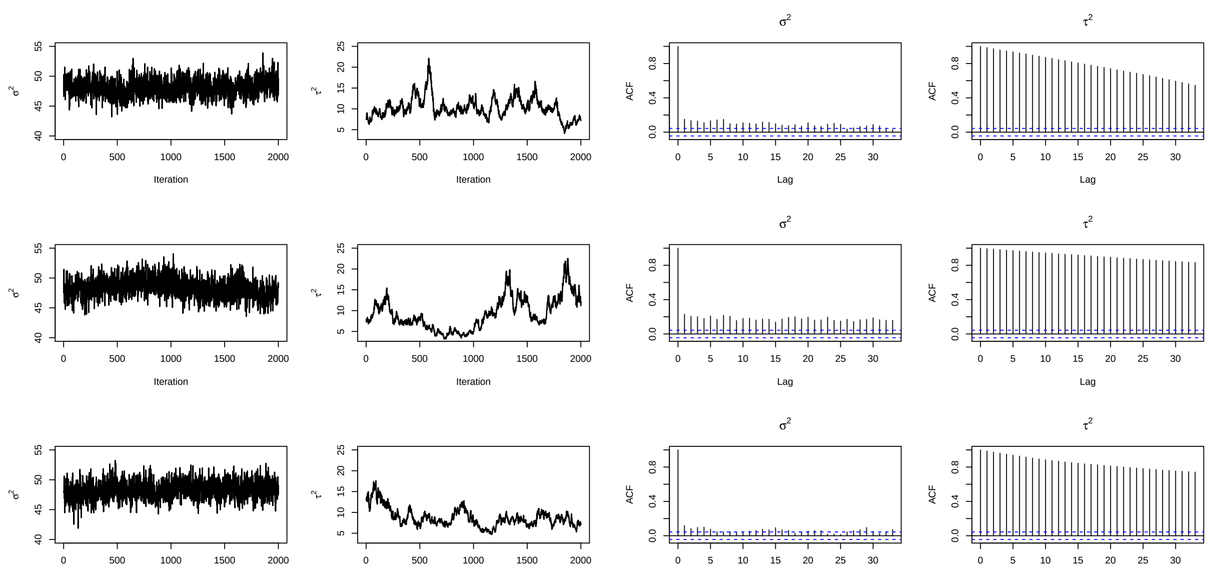


Figure 2: Left panel: MCMC Trace plots of single chains each for σ^2 *and* τ^2 *for the noisy* 50×50 *regular array example. Right panel: Empirical ACF plots for these chains. The top, middle, and bottom rows are from the chromatic, block, and single-site chains, respectively.*

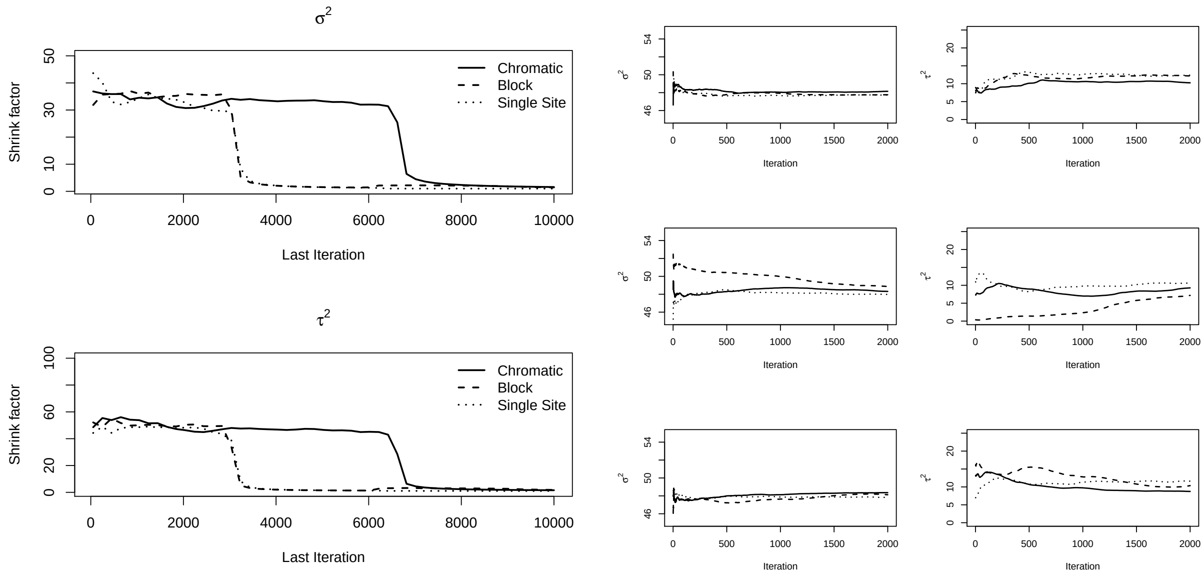


Figure 3: Left panel: Gelman plots (Brooks and Gelman, 1998) of the potential scale reduction factors versus chain length for the noisy 50×50 *regular array example. Right panel: Cumulative averages* $\widehat{\sigma^2}^{(k)}$ *and* $\widehat{\tau^2}^{(k)}$, $k = 1, \ldots, 2000$ *calculated from three independent chains. In the right panel, the top, middle, and bottom rows correspond to chromatic, block, and single-site sampling, respectively.*

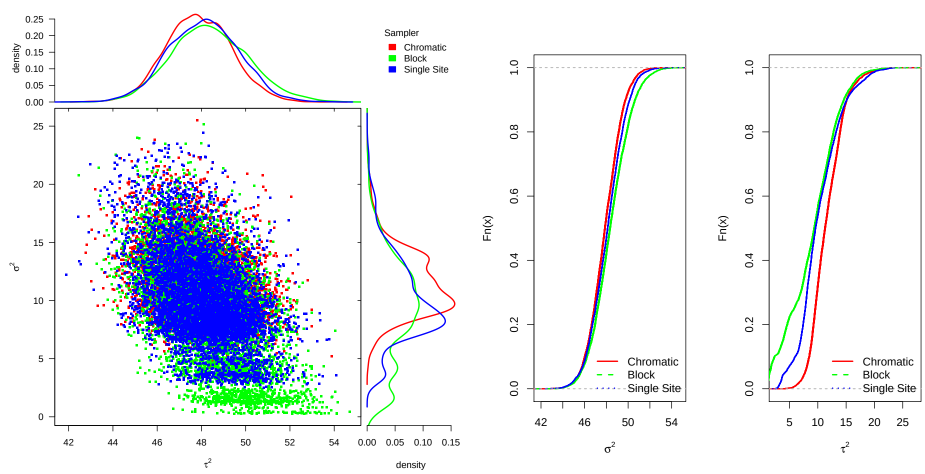


Figure 4: Left panel: Scatterplot and approximate marginal posterior densities estimated from the three sampling approaches for the noisy 50×50 *regular array example. Right panel: Empirical CDFs based on the output. The left panel was created using code available at* `https://github.com/ChrKoenig/R_marginal_plot`.

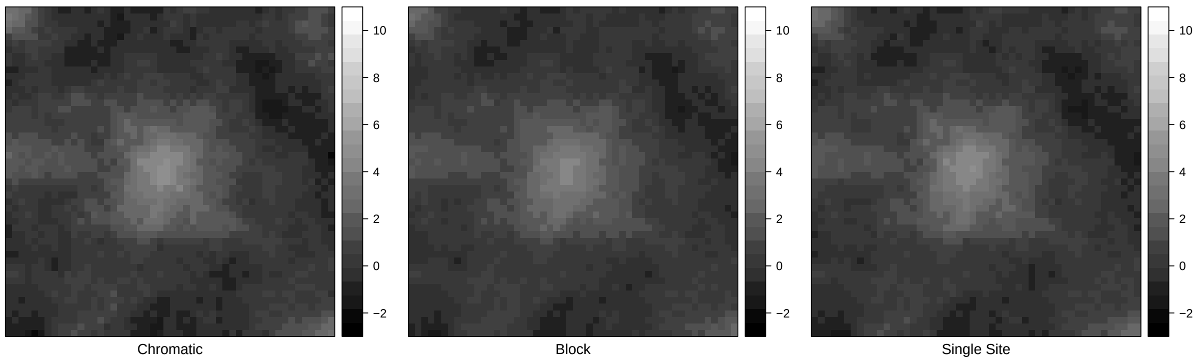


Figure 5: Posterior mean estimates of the the true underlying image obtained from each sampling approach in the noisy 50×50 regular array example.

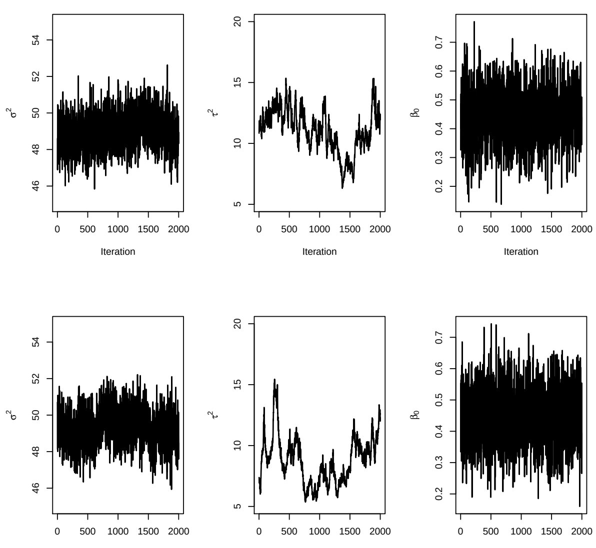


Figure 6: MCMC Trace plots of single chains each for σ^2, τ^2, and β_0 chains in the noisy 80×80 regular array example. The top and bottom rows are from the chromatic and block samplers, respectively.

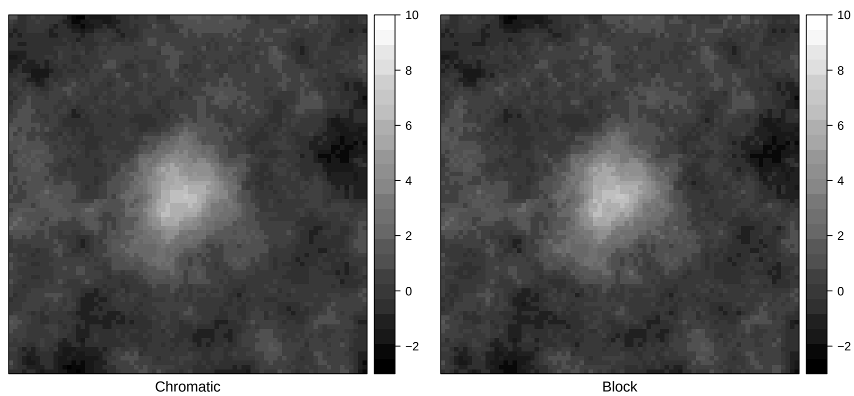


Figure 7: Posterior mean estimates of the the true underlying image obtained from the chromatic and block sampling approaches in the noisy 80×80 *regular array example.*

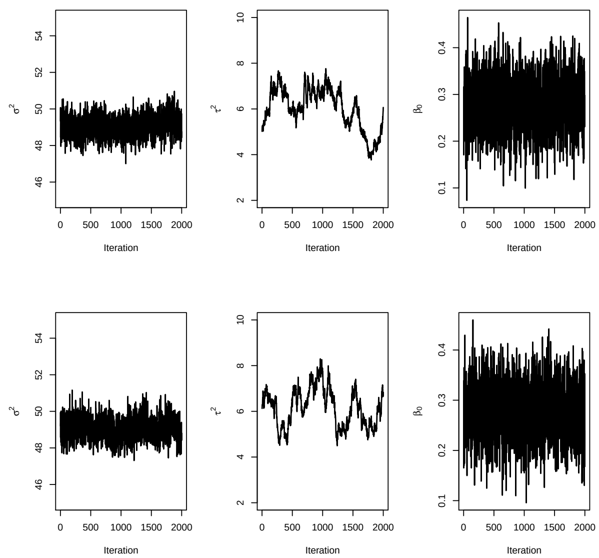


Figure 8: MCMC Trace plots of single chains each for σ^2, τ^2, and β_0 chains in the noisy 128×128 regular array example. The top and bottom rows are from the chromatic and block chains, respectively.

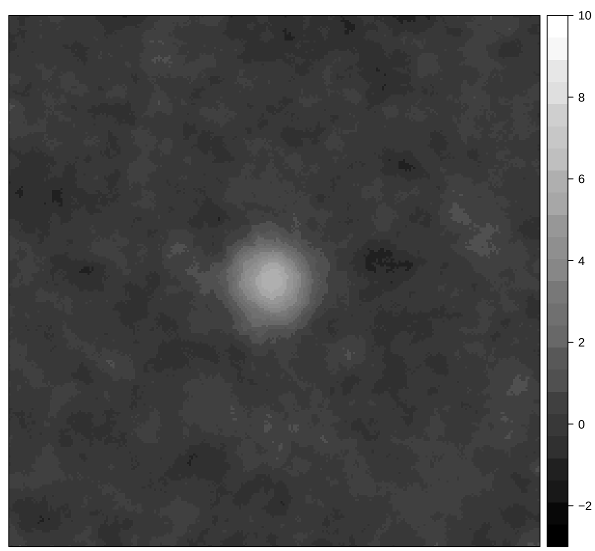


Figure 9: Posterior mean estimate of the the true underlying image obtained from chromatic sampling in the noisy 256×256 regular array example. The estimate is based on the last 2,000 of 10,000 iterations of the Markov chain.

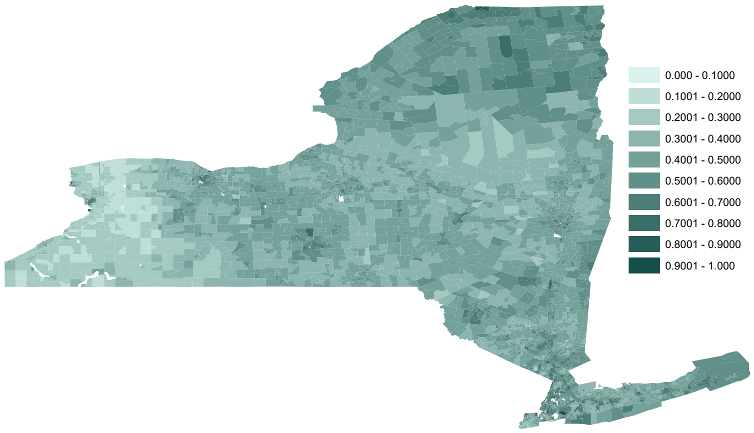


Figure 10: The raw data from the New York election example. The figure displays the total number of votes for the Democratic candidate in each precinct divided by the total number of votes from that precinct.

Figure 11: 7-Coloring of the $14,926$ *precincts in New York found via Algorithm 2.*

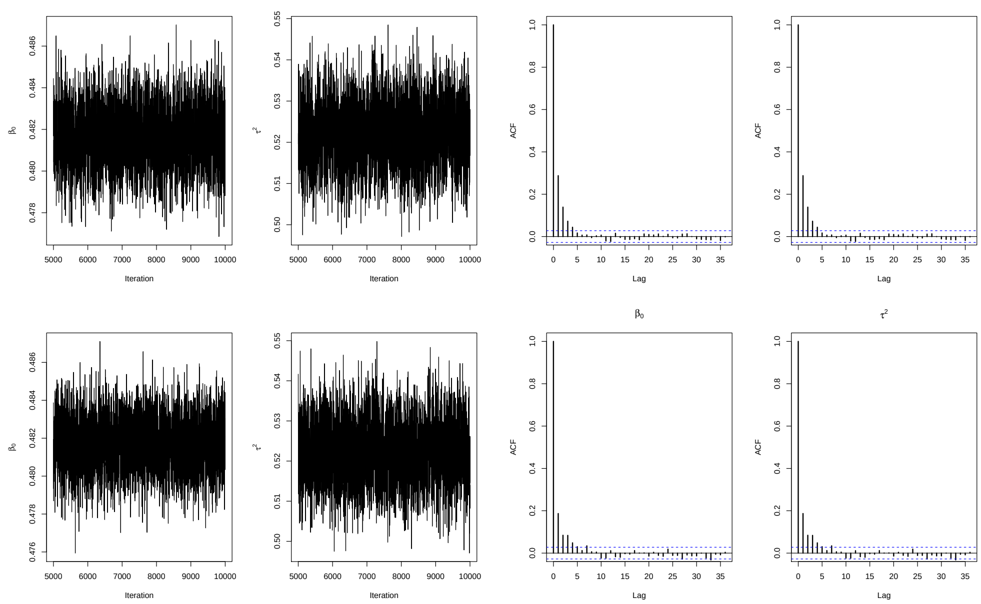


Figure 12: Left panel: MCMC trace plots of single chains for the β_0 and τ^2 chains from the New York election data example. Right panel: Empirical ACF plots for these chains. The top and bottom rows are from the chromatic and block chains, respectively.

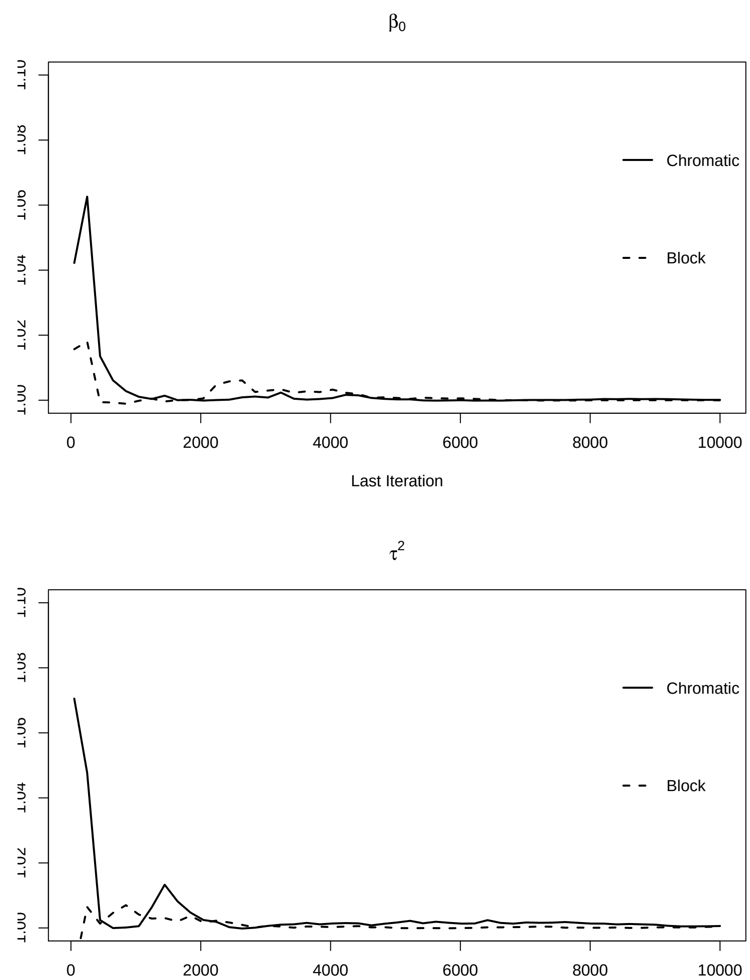


Figure 13: Gelman plots (Brooks and Gelman, 1998) of the potential scale reduction factors versus chain length for the New York election example.

3 DETAILS PERTAINING TO THE SPATIAL BINOMIAL REGRESSION MODEL

It is assumed that $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_n)^T$ obeys a CAR model, which is given by $N(\mathbf{0}, \tau^2(\boldsymbol{D} - \rho\boldsymbol{W})^{-1})$, with ρ being known. The model is completed by specifying the following priors: $\beta_0 \sim N(0, \sigma_0^2)$ and $\tau^2 \sim IG(\alpha_{\tau^2}, \beta_{\tau^2})$, where the hyperparameters are chosen so these priors are vague. Thus, the unknown parameters that are to be sampled via Markov chain Monte Carlo (MCMC) are β_0, $\boldsymbol{\psi}$, $\boldsymbol{\gamma}$, and τ^2, with the only difference between the full block Gibbs and chromatic sampler being how the $\boldsymbol{\gamma}$ are sampled; i.e., the former samples all of these elements in a single block while the latter samples independent blocks of these elements from their univariate full conditionals.

To create a posterior sampling algorithm, it is first noted that the full conditional distribution of β_0 is given by $\beta_0|\boldsymbol{\psi}, \boldsymbol{\gamma}, \boldsymbol{Y} \sim N(\mu_{\beta_0}, \sigma^2_{\beta_0})$, where $\mu_{\beta_0} = \sigma^2_{\beta_0}(\mathbf{1}^T\boldsymbol{\kappa} - \mathbf{1}^T\boldsymbol{D}_{\boldsymbol{\psi}}\boldsymbol{\psi})$ and $\sigma^2_{\beta_0} = (\mathbf{1}^T\boldsymbol{D}_{\boldsymbol{\psi}}\mathbf{1} + \sigma_0^{-2})^{-1}$. Next, the full conditional distribution of τ^2 is given by $\tau^2|\boldsymbol{\gamma} \sim IG(\alpha_{\tau^2} + n/2,\ \beta_{\tau^2} + \boldsymbol{\gamma}^T(\boldsymbol{D} - \rho\boldsymbol{W})\boldsymbol{\gamma}/2)$. The full conditional distribution of the latent ψ_i are again Pólya-Gamma; i.e., $\psi_i|\beta_0, \gamma_i \sim PG(m_i, \eta_i)$, for $i = 1, ..., n$. For further details, see Polson et al. (2013). The full conditional distribution of $\boldsymbol{\gamma}$ is $\boldsymbol{\gamma}|\boldsymbol{Y}, \beta_0, \tau^2, \boldsymbol{\psi} \sim N(\boldsymbol{\mu}_{\boldsymbol{\gamma}}, \boldsymbol{\Sigma}_{\boldsymbol{\gamma}})$, where $\boldsymbol{\mu}_{\boldsymbol{\gamma}} = \boldsymbol{\Sigma}_{\boldsymbol{\gamma}}(\boldsymbol{\kappa} - \boldsymbol{D}_{\boldsymbol{\psi}}\mathbf{1}\beta_0)$ and $\boldsymbol{\Sigma}_{\boldsymbol{\gamma}} = \{\boldsymbol{D}_{\boldsymbol{\psi}} + \tau^{-2}(\boldsymbol{D} - \rho\boldsymbol{W})\}^{-1}$. Through similar arguments, it is easy to show that the univariate full conditional distribution of γ_i is given by $\gamma_i|\boldsymbol{\gamma}_{(-i)}, Y_i, \beta_0, \tau^2, \psi_i \sim N(\mu_{\gamma_i}, \sigma^2_{\gamma_i})$, where $\mu_{\gamma_i} = \sigma^2_{\gamma_i}(\tau^{-2}\rho\sum_{j\in\mathcal{N}(i)}\gamma_j - \psi_i\beta_0 + \kappa_i)$ and $\sigma^2_{\gamma_i} = (\psi_i + \tau^{-2}\boldsymbol{D}_{ii})^{-1}$.

References

Brooks, S. P. and Gelman, A. (1998), "General methods for monitoring convergence of iterative simulations," *Journal of Computational and Graphical Statistics*, 7, 434–455.

Culberson, J. C. (1992), "Iterated greedy graph coloring and the difficulty landscape," Technical Report, University of Alberta.

Polson, N. G., Scott, J. G., and Windle, J. (2013), "Bayesian Inference for Logistic Models Using Pólya Gamma Latent Variables," *Journal of the American Statistical Association*, 108, 1339–1349.

Rue, H. and Held, L. (2005), *Gaussian Markov Random Fields*, Boca Raton: Chapman & Hall/CRC.