Spatial Dependency Modeling Using Spatial Auto-Regression[*]

Mete Celik[1], Baris M. Kazar[2], Shashi Shekhar[1], Daniel Boley[1], David J. Lilja[3]

Abstract

Parameter estimation of the spatial auto-regression model (SAR) is important because we can model the spatial dependency, i.e., spatial autocorrelation present in the geo-spatial data. SAR is a popular data mining technique used in many geo-spatial application domains such as regional economics, ecology, environmental management, public safety, public health, transportation, and business. However, it is computationally expensive because of the need to compute the logarithm of the determinant of a large matrix due to Maximum Likelihood Theory (ML). Current approaches are computationally expensive, memory-intensive and not scalable. In this paper, we propose a new ML-based approximate SAR model solution based on the Gauss-Lanczos algorithm and compare the proposed solution with two other ML-based approximate SAR model solutions, namely Taylor's series, and Chebyshev polynomials. We also algebraically ranked these methods. Experiments showed that the proposed algorithm gives better results than the related approaches when the data is strongly correlated and problem size is large.

Keywords: Spatial Auto-Regression Model, Spatial Dependency Modeling, Spatial Autocorrelation, Maximum Likelihood Theory, Gauss-Lanczos Method.

1. Introduction

Extracting useful and interesting patterns from massive geo-spatial datasets is important for many application domains, including regional economics, ecology, environmental management, public safety, public health, transportation, and business [3, 15, 17]. Many classical data mining algorithms, such as linear regression, assume that the learning samples are *independently and identically distributed (i.i.d.)*. This assumption is violated in the case of spatial data due to spatial autocorrelation [15] and in such cases classical linear regression yields a weak model with not only low prediction accuracy [17] but also residual error exhibiting spatial dependence. Modeling spatial dependencies improves overall classification and prediction accuracies. The Spatial auto-regression (SAR) model is a generalization of linear regression to handle these concerns.

However, estimation of the SAR model parameters is computationally very expensive because of the need to compute the logarithm of the determinant (log-det) of a large matrix. For example, it can take an hour of computation for a spatial dataset with 10K observation points on a single IBM Regatta processor using a 1.3GHz pSeries 690 Power4 architecture with 3.2 GB memory. This has limited the use of SAR to small problem sizes, despite its promise to improve classification and prediction accuracy.

ML-based SAR model solutions [1, 5]can be classified into exact [6, 8, 11-13] and approximate solutions [7, 10, 16], based on how they compute certain compute-intensive terms (log-det term) in the SAR solution procedure. Exact solutions suffer from high computational complexities and memory requirements due to the computation of all the eigenvalues of a large matrix. Approximate SAR model solutions try to approach the computationally complex term of the SAR model by reducing the computation time and providing computationally feasible and scalable SAR model solutions. This study covers only ML-based approximate SAR model solutions. However, we will also include exact solution in our experiments for comparison purposes.

In this paper, we propose a new ML-based approximate SAR solution, and compare and algebraically rank approximate ML-based SAR model solutions. In contrast to the related approximate SAR model solutions, our algorithm provides better approximation when the data is strongly correlated (i.e., spatial dependency is high) and problem size gets high. The key idea of the proposed algorithm is to find only the some of the eigenvalues of a large

[*] This work was partially supported by the Army High Performance Computing Research Center (AHPCRC) under the auspices of the Department of the Army, Army Research Laboratory (ARL) under contract number DAAD19-01-2-0014 and the NSF grant IIS-0208621, and the NSF grant IIS-0534286. This work received additional support from the University of Minnesota Digital Technology Center and the Minnesota Supercomputing Institute.

[1] Computer Science Department, University of Minnesota, MN, USA, {mcelik, shekhar, boley}@cs.umn.edu

[2] Oracle Corporation, USA, baris.kazar@oracle.com

[3] Electrical and Computer Engineering Department, University of Minnesota, MN, USA, lilja@ece.umn.edu

matrix, instead of finding *all* the eigenvalues, by reducing the size of large matrix dramatically using Gauss-Lanczos (GL) algorithm [2]. Because of this property of GL algorithm, we can save huge computation costs, especially when the matrix size is quite large.

The paper compares the proposed algorithm with two related approximate approaches and the exact solution procedure. Then, we algebraically rank them to determine which method gives better approximations in what conditions. Experimental results show that the proposed algorithm saves computation time for the large problem sizes. Experiments also showed that it gives better results when the dataset is strongly correlated (i.e., spatial dependency is high).

2. **Problem Statement**

In this study, we rank the algebraic errors of three ML-based approximate SAR model solutions [7, 10, 16]. Given a spatial framework, observations on a dependent variable, a set of explanatory variables, and neighborhood relationship among the spatial data, SAR parameter estimation based on Maximum Likelihood theory [1, 5] aims to find the optimum SAR model parameters by minimizing the likelihood function of the SAR model solution. The problem is formally defined as follows.

Given:

- A spatial framework S consisting of sites $\{s_1, \ldots, s_n\}$ for the underlying spatial graph G.
- A collection of explanatory functions f_{x_k} : $S \rightarrow R^k$, k =1,….,K. R^k is the range of possible values for explanatory functions.
- A dependent function $f_y : R \rightarrow R^y$
- A family F (i.e., $\mathbf{y}=\rho\mathbf{W}\mathbf{y}+\mathbf{x}\beta+\varepsilon$) of learning model functions mapping $R^1 \times \ldots\ldots \times R^K \rightarrow R^y$.
- A neighborhood relationship R on the spatial framework

Find:

- The SAR model parameters ρ and the regression coefficient β.

Objective:

- Algebraic error ranking of approximate SAR model solutions.

Constraints:

- S is a multi-dimensional Euclidean Space,
- The values of the explanatory variables **x** and the dependent function (observed variable) **y** may not be independent with respect to those of nearby spatial sites, i.e., spatial autocorrelation exists.
- The domain of **x** and **y** are real numbers.
- The SAR parameter ρ varies in the range [0,1),
- The error is normally distributed with unit standard deviation and zero mean, i.e., $\varepsilon \sim N(0,\sigma^2\mathbf{I})$ IID
- The neighborhood matrix **W** exhibits sparsity.

2.1. **Basic Concepts**

The SAR model, also known in the literature as the spatial lag model or mixed regressive model [1, 4, 5], is an extension of the linear regression model (equation (1)).

$$\mathbf{y} = \rho\mathbf{W}\mathbf{y} + \mathbf{x}\beta + \varepsilon \tag{1}$$

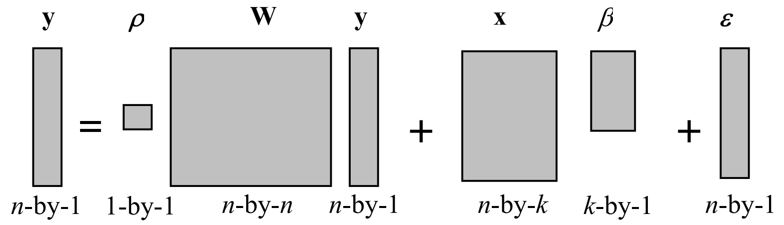


Figure 1. Data structures of the SAR model equation

The data structures of the SAR model can be seen in Figure 1. Here ρ is the spatial autocorrelation parameter, **y** is an n-by-1 vector of observations on the dependent variable, **x** is an n-by-k matrix of observations on the explanatory variable, **W** is the n-by-n neighborhood matrix that accounts for the spatial relationships (dependencies) among the spatial data, β is a k-by-1 vector of regression coefficients, and **ε** is an n-by-1 vector of unobservable error. The *spatial autocorrelation* term ρ**Wy** is added to the linear regression model in order to model the strength of the spatial dependencies among the elements of the dependent variable **y**. Moran's Index [5] can be used to see whether there is significant spatial dependency in the given dataset.

The log-likelihood function (i.e., the logarithm of the ML function) to be optimized for the ρ parameter is given in equation 2. The function contains two parts, such as log-det term and *SSE* term.

$$\min_{|\rho|<1} \frac{-2}{n} \underbrace{\ln\left|\mathbf{I}-\rho\mathbf{W}\right|}_{\log-\det} + \underbrace{\ln((\mathbf{I}-\rho\mathbf{W})\mathbf{y})^T(\mathbf{I}-\mathbf{x}(\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T)^T(\mathbf{I}-\mathbf{x}(\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T)(\mathbf{I}-\rho\mathbf{W})\mathbf{y})}_{SSE} \tag{2}$$

The log-likelihood function optimized is using nonlinear optimization techniques, such as, golden section search, to find the best estimate for the SAR model parameters. Rather than optimizing for both SAR parameters ρ and β, it is faster and easier to optimize one unknown (i.e., ρ) since both parameters are dependent on each other.

3. **ML-based Approximate SAR Model Solutions**

The exact SAR model solutions suffer high computational complexity and memory requirements even in parallel form [6]. These limitations have led us to investigate approximate solutions for SAR model parameter estimation with the main objective of scaling the SAR model for large spatial data analysis problems. We inspected two approximate SAR model solutions Taylor's series expansion and Chebyshev coefficients, and developed a new approximate SAR model solution based on the Gauss-Lanczos algorithm. Then we compared all the approximate SAR model solutions.

3.1. **Approximation by Taylor's Series Expansion**

[9] suggests an approximation of the log-det of a matrix by means of the traces of the powers of the neighborhood matrix, **W** (equation 3). It basically finds the trace of the matrix logarithm, which is equal to the log-det of the matrix. In this approach, the Taylor's series expansion is used to approximate the $\sum_{i=1}^{n}\ln(1-\rho\lambda_i)$ where λ_i represents the i^{th} eigenvalue that lies in the interval [-1,+1] and ρ is the scalar parameter from the interval (-1,+1). The term $\sum_{i=1}^{n}\ln(1-\rho\lambda_i)$ can be expanded as $\sum_{i=1}^{n}(\rho\lambda_i)^k/k$ provided that $|\rho\lambda_i|<1$, which will hold for all i if $|\rho|<1$. Equation 3, which states the approximation used for the log-det term of log-likelihood function, is obtained using the relationship between the eigenvalues and the trace of a matrix, i.e., $\sum_{i=1}^{n}\lambda_i^k = tr(\mathbf{W}^k)$.

$$\ln|\mathbf{I}-\rho\mathbf{W}| = tr(\ln(\mathbf{I}-\rho\mathbf{W})) = \sum_{k=0}^{\infty}\frac{\rho^k tr(\mathbf{W}^k)}{k} \tag{3}$$

The approximation comes into the picture when we sum up to a finite value, r, instead of infinity. Therefore, equation 3 is relatively much faster because it eliminates the need to calculate the compute-intensive eigenvalue estimation when computing the log-det term (Figure 2).

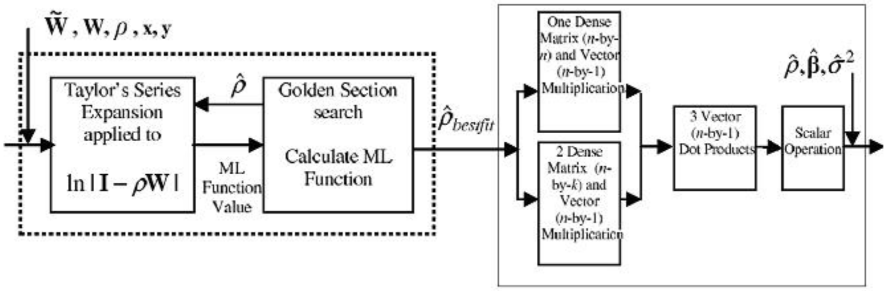

$\tilde{\mathbf{W}}, \mathbf{W}, \rho, \mathbf{x}, \mathbf{y}$
Taylor's Series Expansion applied to $\ln|\mathbf{I}-\rho\mathbf{W}|$
$\hat{\rho}$
Golden Section search
Calculate ML Function
ML Function Value
$\hat{\rho}_{bestfit}$
One Dense Matrix (n-by-n) and Vector (n-by-1) Multiplication
2 Dense Matrix (n-by-k) and Vector (n-by-1) Multiplication
3 Vector (n-by-1) Dot Products
Scalar Operation
$\hat{\rho}, \hat{\beta}, \hat{\sigma}^2$

Figure 2. The system diagram of the Taylor's series approximation for the SAR model solution.

Approximation by Chebyshev Polynomials

This approach uses the symmetric equivalent of the neighborhood matrix **W** (i.e., $\tilde{\mathbf{W}}$). The eigenvalues of the symmetric matrix $\tilde{\mathbf{W}}$ are the same as those of the neighborhood matrix **W**. The lemma 3.1 leads to a very efficient and accurate approximation of the log-det term of the log-likelihood function shown in equation 2.

Lemma 3.1: *The Chebyshev solution tries to approximate the log-det of* (**I**-ρ**W**) *involving a symmetric neighborhood matrix* $\tilde{\mathbf{W}}$ *as in equation 4, which is the relationship of the Chebyshev polynomial to the log-det of* (**I**-ρ **W**) *matrix. The three terms are enough for approximating the log-det term with an accuracy of 0.03%* [7].

$$\ln|\mathbf{I}-\rho\tilde{\mathbf{W}}| \equiv \ln|\mathbf{I}-\rho\tilde{\mathbf{W}}| \approx \sum_{j=0}^{q+1} c_j(\rho)tr(T_{j-1}(\tilde{\mathbf{W}})) - \frac{1}{2}c_1(\rho) \tag{4}$$

Proof: The proof of this equality is available in [14]. □

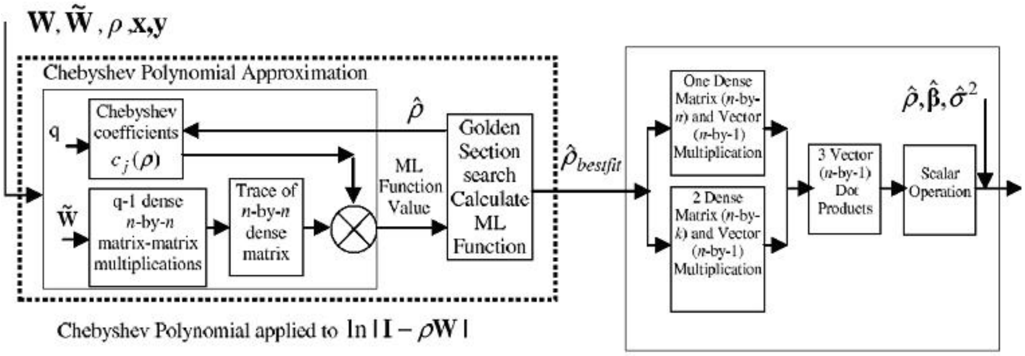
The value of "q" is 3, which is the highest degree of the Chebyshev polynomials. Therefore, only $T_0(\tilde{\mathbf{W}})$, $T_1(\tilde{\mathbf{W}})$, and $T_2(\tilde{\mathbf{W}})$ have to be computed where:

$$T_{k+1}(\tilde{\mathbf{W}}) = 2\tilde{\mathbf{W}}T_{k+1}(\tilde{\mathbf{W}}) - T_{k-1}(\tilde{\mathbf{W}}) \tag{5}$$

The Chebyshev polynomial coefficients $c_j(\rho)$ are given in equation 6.

$$c_j(\rho) = \frac{2}{q+1}\sum_{k=1}^{q+1}\ln(1-\rho\cos(\frac{\pi(k-\frac{1}{2})}{q+1}))\cos(\frac{\pi(j-1)(k-\frac{1}{2})}{q+1}) \tag{6}$$

In Figure 3, the ML function is determined by computing the maximum of the sum of the log-det of a large matrix and the *SSE* term. The SAR parameter ρ that achieves this maximum value is the desired value that makes the classification most accurate. The parameter "q" is the degree of the Chebyshev polynomial, which is used to approximate the log-det term. The pseudocode of the Chebyshev polynomial approximation is presented in Figure 3. Lemma 3.2 reduces the computational complexity of the Chebyshev polynomial from *$O(n^3)$* to approximately *$O(n^2)$*.



$\mathbf{W}, \tilde{\mathbf{W}}, \rho, \mathbf{x}, \mathbf{y}$
Chebyshev Polynomial Approximation
q
Chebyshev coefficients $c_j(\rho)$
$\hat{\rho}$
$\tilde{\mathbf{W}}$
q-1 dense n-by-n matrix-matrix multiplications
Trace of n-by-n dense matrix
ML Function Value
Golden Section search Calculate ML Function
$\hat{\rho}_{bestfit}$
One Dense Matrix (n-by-n) and Vector (n-by-1) Multiplication
2 Dense Matrix (n-by-k) and Vector (n-by-1) Multiplication
3 Vector (n-by-1) Dot Products
Scalar Operation
$\hat{\rho}, \hat{\boldsymbol{\beta}}, \hat{\sigma}^2$
Chebyshev Polynomial applied to $\ln|\mathbf{I}-\rho\mathbf{W}|$

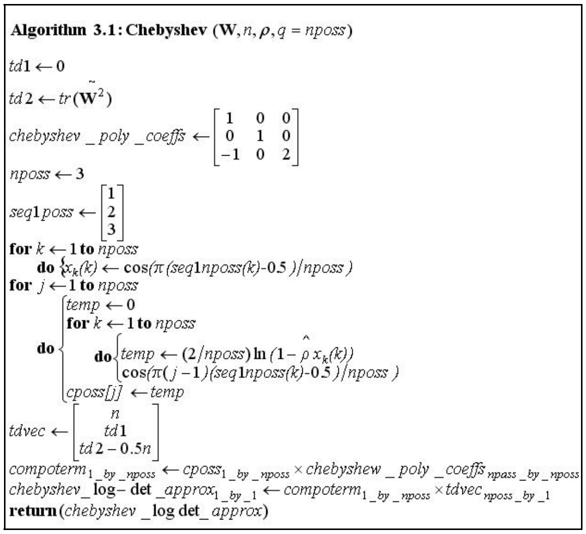
Figure 3. The system diagram of Chebyshev polynomial approximation for the SAR model solution

Lemma 3.2: *For regular grid-based nearest-neighbor symmetric neighborhood matrices, the relationship shown in equation 6 holds.*

$$tr(\tilde{\mathbf{W}}^2) = \sum_{i=1}^{n}\sum_{j=1}^{n}\tilde{w}_{ij}^2 \qquad \textit{where the (i,j)}^{th} \textit{ element of } \tilde{\mathbf{W}} \textit{ is } \tilde{w}_{ij}. \tag{6}$$

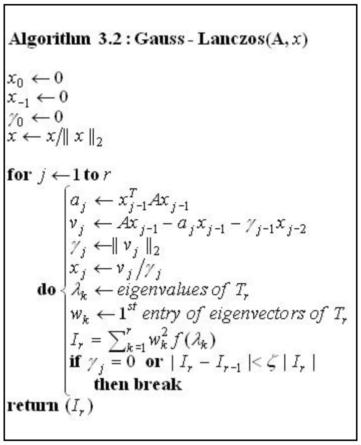
Proof: The equality property given in equation 6 follows from the symmetry property of the symmetrized neighborhood matrix. In other words, this is valid for all symmetric matrices. The trace operator sums the diagonal elements of the square of the symmetric matrix $\tilde{\mathbf{W}}$. This is the equivalent of saying that the trace operator first multiplies and adds the i^{th} column with the i^{th} row of the symmetric matrix, where the i^{th} column and the i^{th} row of the matrix are the $\tilde{\mathbf{W}}$ entries in a symmetric matrix. □

In the pseudocode of the Chebyshev approximation (Figure 4 (a)), the powers of the **W** matrices, whose traces are to be computed, go up to 2. The parameter "q" is the degree of the Chebyshev polynomial which is used to approximate the term ln|**I**-ρ**W**|. The ML function is computed by calculating the maximum of the log-likelihood functions (i.e. the log-det term and the *SSE term*).



```
**Algorithm 3.1: Chebyshev** $(\mathbf{W}, n, \rho, q = nposs)$
$td1 \leftarrow 0$
$td2 \leftarrow tr(\tilde{\mathbf{W}}^2)$
$chebyshev\_poly\_coeffs \leftarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 2 \end{bmatrix}$
$nposs \leftarrow 3$
$seq1poss \leftarrow \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$
for $k \leftarrow 1$ to $nposs$
    do $\{x_k(k) \leftarrow \cos(\pi(seq1nposs(k)\text{-}0.5)/nposs)$
for $j \leftarrow 1$ to $nposs$
    do
        $temp \leftarrow 0$
        for $k \leftarrow 1$ to $nposs$
            do
                $temp \leftarrow (2/nposs)\ln(1-\hat{\rho}\, x_k(k))$
                $\cos(\pi(j-1)(seq1nposs(k)\text{-}0.5)/nposs)$
        $cposs[j] \leftarrow temp$
$tdvec \leftarrow \begin{bmatrix} n \\ td1 \\ td2 - 0.5n \end{bmatrix}$
$compoterm_{1\_by\_nposs} \leftarrow cposs_{1\_by\_nposs} \times chebyshew\_poly\_coeffs_{npass\_by\_nposs}$
$chebyshev\_\log\text{-}\det\_approx_{1\_by\_1} \leftarrow compoterm_{1\_by\_nposs} \times tdvec_{nposs\_by\_1}$
return $(chebyshev\_\log\det\_approx)$
```

(a) Pseudocode of Chebyshev Algorithm



```
**Algorithm 3.2: Gauss - Lanczos** $(\mathbf{A}, x)$
$x_0 \leftarrow 0$
$x_{-1} \leftarrow 0$
$\gamma_0 \leftarrow 0$
$x \leftarrow x/\| x \|_2$
for $j \leftarrow 1$ to $r$
    do
        $a_j \leftarrow x_{j\text{-}1}^T A x_{j\text{-}1}$
        $v_j \leftarrow A x_{j\text{-}1} - a_j x_{j\text{-}1} - \gamma_{j\text{-}1} x_{j\text{-}2}$
        $\gamma_j \leftarrow \| v_j \|_2$
        $x_j \leftarrow v_j / \gamma_j$
        $\lambda_k \leftarrow$ eigenvalues of $T_r$
        $w_k \leftarrow 1^{st}$ entry of eigenvectors of $T_r$
        $I_r = \sum_{k=1}^{r} w_k^2 f(\lambda_k)$
        if $\gamma_j = 0$ or $| I_r - I_{r\text{-}1} | < \zeta | I_r |$
            then break
return $(I_r)$
```

(b) Pseudocode of Gauss-Lanczos Algorithm

Figure 4: The pseudocodes of: (a) Chebyshev and (b) Gauss-Lanczos algorithms

3.2. A New Approximation Based on Gauss-Lanczos

We developed a new ML-based approximate SAR model solution based on the Gauss-Lanczos algorithm (Figure 5). [2] suggests the GL method to approximate the eigenvalue problem of ln|**I**-ρ**W**| (Figure 4(b)). First, the problem is transformed to quadratic form $u^T f(\mathbf{A}) u$ (in our case **A** equals the symmetric positive definite matrix **(**ln|**I**-ρ**W**|**))**, where **A**, *u*, and *f* represent a matrix, a vector, and a function, respectively. In our case, the function *f* represents the logarithm of a matrix. Then, the quadratic form is converted to a Riemann-Stieltjes integral problem (detailed information can be found in [2]).To approximate the integral, gauss-type quadrature rules are applied using the Lanczos procedure (equation 7).

$$\ln\left|\mathbf{I} - \rho\,\tilde{\mathbf{W}}\right| = tr(\ln(\mathbf{I} - \rho\,\tilde{\mathbf{W}})) \approx \frac{1}{m}\sum_{i=1}^{n} I_r^{(i)} \tag{7}$$

In equation 7, the quadrante formula is represented by I_r, which is approximated by the GL method. The parameter *m* represents the number of runs of the GL method. To find a satisfactory estimation of the quantity of trace function *tr*, the GL algorithm is applied *m* times and the average of the I_r 's are taken. The GL algorithm (Figure 5) takes two inputs, a real *n*-by-*n* symmetric positive definite matrix **A** and a real *n*-by-1 vector with $x^T x$=1. First, in the "for" loop (Figure 4(b)), GL computes *r*-by-*r* symmetric tri-diagonal matrix T_r until a convergence criterion (γ_j=0 or $|I_r - I_{r-1}| < \zeta |I_r|$) is satisfied, or GL computes *r* times, which can be specified by the user by transforming the **A** matrix to the quadrature form, where *r*<<*n*. Then, GL computes eigenvalues λ_k and first elements w_k of eigenvectors of matrix T_r. Finally, I_r is calculated, as given in equation 8.

$$I_r = \sum_{k=1}^{r} w_k^2 f(\lambda_k) \tag{8}$$

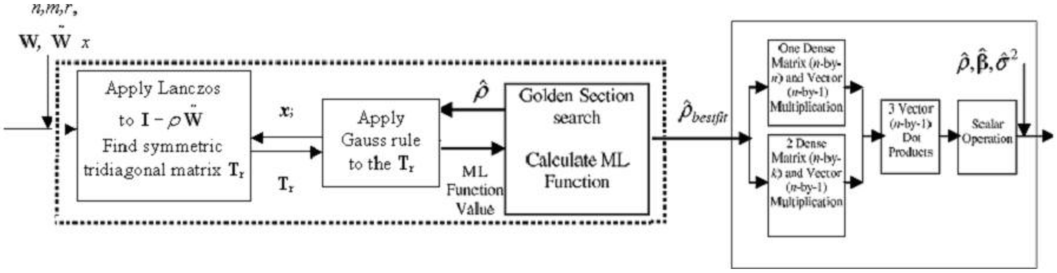

$n, m, r,$
$\mathbf{W}, \ddot{\mathbf{W}}\ x$
Apply Lanczos to $\mathbf{I}-\rho\ddot{\mathbf{W}}$ Find symmetric tridiagonal matrix $\mathbf{T}_r$
$\mathbf{x}_i$
$\mathbf{T}_r$
Apply Gauss rule to the $\mathbf{T}_r$
$\hat{\rho}$
ML Function Value
Golden Section search
Calculate ML Function
$\hat{\rho}_{bestfit}$
One Dense Matrix (n-by-n) and Vector (n-by-1) Multiplication
2 Dense Matrix (n-by-k) and Vector (n-by-1) Multiplication
3 Vector (n-by-1) Dot Products
Scalar Operation
$\hat{\rho}, \hat{\boldsymbol{\beta}}, \hat{\sigma}^2$

Figure 5. Gauss-Lanczos approximation method

4. **Error Ranking**

This section formulates the relative error ranking of the approximations to log-det and hence the effect on the estimation of the parameter ρ.

Using the log-likelihood function $\ell(\theta \mid \mathbf{y})$ given in equation 2, we can write ρ as a function of the $\ell(\theta \mid \mathbf{y})$, such that $\rho = f^{-1}\,\ell(\theta \mid \mathbf{y})$ Thus the change (error) in the log-likelihood due to the approximation is reflected into the estimation of the parameter ρ as follows where the operator Δ denotes the difference between the exact (i.e., true) and the approximated values:

$$\Delta\rho = \frac{df^{-1}(\ell(\theta \mid \mathbf{y}))}{d\ell(\theta \mid \mathbf{y})}\Delta\ell(\theta \mid \mathbf{y}) \tag{9}$$

The quantity $\Delta\rho$ is the error in ρ obtained from the approximate method. The quantity $\Delta\,\ell(\theta \mid \mathbf{y})$ is the error in the log-likelihood function from the approximate method, which we can compute algebraically.

The derivation part will be the same for the different approximations since the initial ρ parameter is fixed and other variables are the same for each approximate solution. The error in the ρ parameter can be estimated by multiplying the error in the log-det by a derivative term.

Since we assume that we have the same *SSE* term for all SAR model solutions, we do not approximate it (i.e., Δ*SSE*=0). The term $\Delta\,\ell(\theta \mid \mathbf{y})$ corresponds directly to the error in the log-det approximation i.e., $\Delta\ln|\mathbf{I}\text{-}\rho\mathbf{W}|$.

5. **Experimental Design and System Setup**

In the experiments synthetic datasets were generated for different problem sizes, such as n=400, 1600, 2500 and for different spatial auto-regression parameters. We took 4-neighbors (i.e., North, South, East, and West neighbors) (Appendix I) of the interested cell (location) and all experiments were run on the same platform. All the experiments were carried out using the same common experimental setup summarized in Table 1.

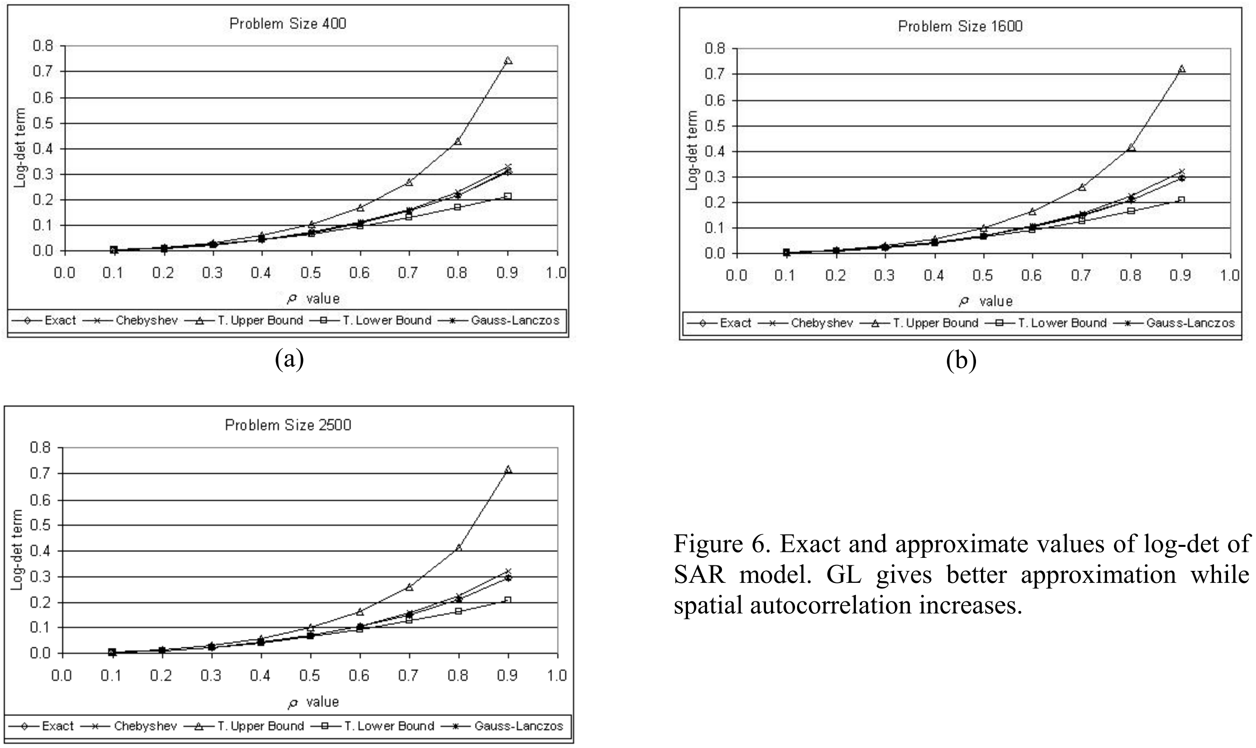
Table 1. The experimental design

Factor Name	Parameter Domain
Problem Size (n)	400, 1600, 2500 observation points
Neighborhood Structure	2-D with 4-neighbors
Candidates	Exact Approach (Eigenvalue Computation Based) Taylor's Series Approximation Chebyshev Polynomial Approximation Gauss-Lanczos Approximation
Dataset	Synthetic Dataset for ρ=0.1, 0.2, ….., 0.9
SAR Parameter ρ	[0,1)
Programming Language	Matlab

6. Results and Discussion

ML-based solutions of the SAR require computing the log-det of a large matrix (**I**-ρ**W**), which is computationally expensive. Approximate SAR model solutions try to approximate the log-det of a large matrix by reducing computation cost of this term. It is observed that exact SAR model solution takes approximately 2 orders of magnitude of more time than approximate solutions. In this study, we algebraically ranked ML-based approximate SAR model solutions.

In the experiments we tried to identify the behavior of the candidate algorithms for different problem sizes and for different spatial autocorrelation values (thus different spatial dependencies). Exact and approximated results for the log-det term of the SAR model are given in Figure 6 and 7. The results of the GL approximation are the average of several runs. We generated synthetic datasets for different ρ parameters. Figure 6 shows the approximation results for the log-det term of the SAR model of the candidate methods. We ran the experiments for three different problem sizes, such as 400, 1600, and 2500. It is observed that Taylor's series approximation gives upper and lower bounds of the approximation log-det term of the SAR model for all problem sizes. Chebyshev approximation gives the optimum results when the spatial autocorrelation parameter ρ is close to zero for all problem sizes. In contrast, for all problem sizes, GL approximation gives better results than Chebyshev approximation when the autocorrelation is high such as spatial autocorrelation parameter is close to 1. This behavior of the GL approximation can be explained by the fact that many cancellations occur while the GL calculates the logarithms of all the eigenvalues of matrix T_r when the spatial autocorrelation low (ρ is close to zero).



(a)

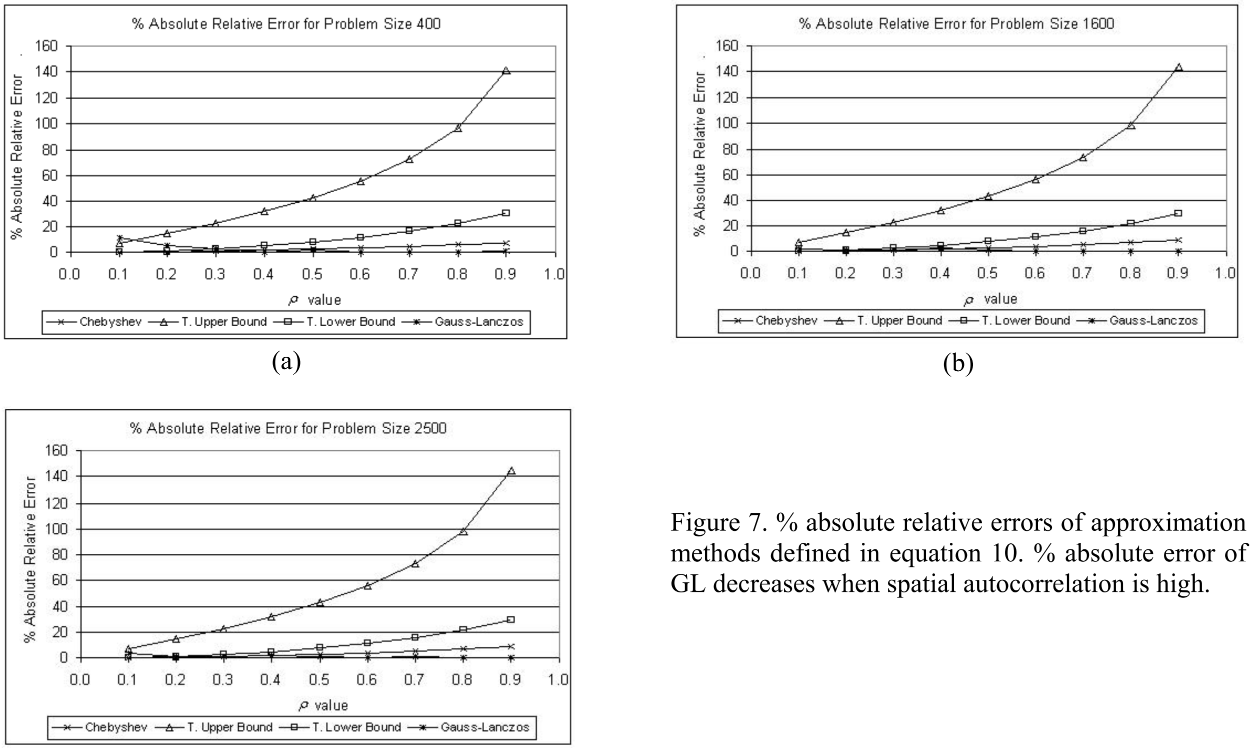
(b)

(c)

Figure 6. Exact and approximate values of log-det of SAR model. GL gives better approximation while spatial autocorrelation increases.

Figure 7 gives the difference in the accuracy of the results by approximation methods where the difference in the accuracy is defined by the absolute relative error defined in equation 10. It is observed that absolute relative error (% accuracy) of Taylor's series and Chebyshev approximations increase while ρ parameter is increasing but Chebyshev approximation gives better results than Taylor's series approximation. In contrast, absolute relative error of GL algorithm decreases while ρ parameter is increasing. We can conclude that GL algorithm gives more accurate results than the other methods. Therefore, GL is better than the other candidate solutions when the spatial autocorrelation is high (ρ is close to 1).

$$100 \times abs\left(\frac{(\frac{-2}{n})\ln|\mathbf{I}-\rho\mathbf{W}|_{exact} - (\frac{-2}{n})\ln|\mathbf{I}-\rho\mathbf{W}|_{approximate}}{(\frac{-2}{n})\ln|\mathbf{I}-\rho\mathbf{W}|_{exact}} \right) \tag{10}$$



(a)

(b)

(c)

Figure 7. % absolute relative errors of approximation methods defined in equation 10. % absolute error of GL decreases when spatial autocorrelation is high.

The computational cost of the Chebyshev approximation is $O(n^3)$. Using lemma 3.2 the cost of the Chebyshev approximation can be reduced to approximately $O(n^2)$. In contrast, the cost of the GL approximation is $2mrO(n^2)$, which includes $2mr$ matrix-vector multiplications of the rank-n matrix. Thus, GL is slightly more expensive than Chebyshev and Taylor's series approximations. In the GL procedure, m represents the number of iterations. In our experiments, m was fixed (i.e., m=400) for each problem size. If the problem size is large enough, the effect of m will be less in the computation cost. In GL, r represents the size of tri-diagonal symmetric matrix **T** where $r<<n$. The size of the **T** matrix changes during the GL procedure according to various the problem sizes and ρ parameters. In our experiments the value of r varies between 5 and 8 where $r<<n$ for problem sizes 400, 1600, 2500. The effect of the r parameter will also be less for the larger problem sizes. Results show that GL approximation is one of the candidate solutions for large problem sizes, especially when the spatial autocorrelation is high, and m and r parameters are smaller than the problem size.

It is also observed that the quality of the results of the GL algorithm depends on the number of iterations, as discussed before, and the initial Lanczos vector which is selected randomly. In our experiments, the initial Lanczos vector is selected as a discrete random vector where values of components are either -1 or 1 with the probability of 0.5. Finally, it is also observed that increasing the number of iterations can decrease the effect of the random number generator. However, increasing the number of iterations may lead to the increase in the computation cost of the GL approximation.

7. Conclusion and Future Work

In this study we algebraically compared three approximate solution procedures of the SAR model and explained which method is better in what conditions and proposed a new Maximum Likelihood Theory based approximate SAR model solution based on Gauss-Lanczos algorithm. The key idea of the proposed algorithm is to find only some of the eigenvalues of a large matrix, instead of finding all the eigenvalues, by reducing the size of large matrix dramatically using Gauss-Lanczos algorithm [2]. In the experiments, Cheyshev polynomial approximation provides better approximation when the spatial autocorrelation is low. Gauss-Lanczos approximation gives better approximation than the other methods when the problem size is large and spatial autocorrelation is high.

Our future work will examine how to parallelize the Gauss-Lanczos algorithm to decrease computation cost.

Acknowledgements

The authors would like to thank the University of Minnesota Digital Technology Center and Minnesota Supercomputing Institute for use of their computing resources. The authors are grateful to the members of the Spatial Database Group, ARCTiC Labs Group, Professor Vladimir Cherkassky, Dr. Birali Runesha and Dr. Shuxia Zhang at the Minnesota Supercomputing Institute for valuable discussions. The authors also thank Kim Koffolt for helping improve the readability of this paper.

References

[1] L. Anselin, *Spatial Econometrics: Methods and Models*. Dorddrecht, Kluwer Academic Publishers, 1988.

[2] Z. Bai and G. H. Golub, Some unusual Matrix Eigenvalue Problems, *Proceedings of VECPAR'98 - Third International Conference for Vector and Parallel Processing*, vol. 1573(4-19, 1999.

[3] S. Chawla, S. Shekhar, W. Wu, and U. Ozesmi, Modeling Spatial Dependencies for Mining Geospatial Data, *1st SIAM International Conference on Data Mining*, 2001.

[4] N. A. Cressie, *Statistics for Spatial Data*. New York, Wiley, 1993.

[5] D. A. Griffith, *Advanced Spatial Statistics*, Kluwer Academic Publishers, 1998.

[6] B. M. Kazar, S. Shekhar, D. J. Lilja, and D. Boley, A Parallel Formulation of the Spatial Auto-Regression Model for Mining Large Geo-Spatial Datasets, *SIAM International Conf. on Data Mining Workshop on High Performance and Distributed Mining (HPDM2004)*, April 2004.

[7] B. M. Kazar, S. Shekhar, D. J. Lilja, R. R. Vatsavai, and R. K. Pace, Comparing Exact and Approximate Spatial Auto-Regression Model Solutions for Spatial Data Analysis, *Third International Conference on Geographic Information Science (GIScience2004)*, October 2004.

[8] B. Li, Implementing Spatial Statistics on Parallel Computers, in *Practical Handbook of Spatial Statistics*: CRC Press, pp. 107-148, 1996.

[9] R. J. Martin, Approximations to the determinant term in Gaussian maximum likelihood estimation of some spatial models, *Statistical Theory Models*, vol. 22(1), pp. 189-205, 1993.

[10] R. K. Pace and J. P. LeSage, Chebyshev Approximation of Log-Determinant of Spatial Weight Matrices, *Computational Statistics and Data Analysis*, 2004.

[11] R. K. Pace and J. P. LeSage, Closed-form maximum likelihood estimates for spatial problems (MESS), http://www.spatial-statistics.com, 2000.

[12] R. K. Pace and J. P. LeSage, Semiparametric Maximum Likelihood Estimates of Spatial Dependence, *Geographical Analysis*, vol. 34(1), pp. 76-90, 2002.

[13] R. K. Pace and J. P. LeSage, Simple bounds for difficult spatial likelihood problems, http://www.spatial-statistics.com, 2003.

[14] W. Press, S. A. Teukulsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran 77*, Cambridge University Press, 1992.

[15] S. Shekhar and S. Chawla, *Spatial Databases: A Tour*, Prentice Hall, 2003.

[16] S. Shekhar, B. M. Kazar, and D. J. Lilja, Scalable Parallel Approximate Formulations of Multi-Dimensional Spatial Auto-Regression Models for Spatial Data Mining, *24th Army Science Conference*, November 2004.

[17] S. Shekhar, P. Schrater, R. Raju, and W. Wu, Spatial Contextual Classification and Prediction Models for Mining Geospatial Data, *IEEE Transactions on Multimedia*, vol. 4(2), pp. 174-188, 2002.

Appendix I

Formation of Neighborhood Matrix W

The neighborhood matrices used by the SAR model are the neighborhood relationships on one-dimensional regular and irregular grid spaces with two neighbors and two-dimensional regular or irregular grid space with "s" neighbors, where "s" is four, eight, sixteen, twenty-four and so on neighbors. The rows of the neighborhood matrix **W** sum to 1, which means that **W** is row-standardized i.e., row-normalized, row-stochastic, or Markov matrix (Figure 8(b)). A non-zero entry in the *j*th column of the *i*th row indicates that the *j*th observation will be used to adjust the prediction of the *i*th row where *i* is not equal to *j*. Thus, the ML theory estimated SAR model solutions used in our study accept neighborhood matrices from both regular and irregular grid spaces, which is a very important feature.

In Figure 8, we illustrate the formation of the neighborhood matrix on a 4-by-4 regular grid space. As noted before, modeling spatial dependency improves the overall classification (prediction) accuracy. Spatial dependency can be defined by the relationships among spatially adjacent pixels in a small neighborhood within a spatial framework that is a regular or irregular grid space. For the four-neighborhood case, the neighbors of the *(i,j)*th pixel of the regular grid are defined as below.

$$neighbors(i,j) = \begin{cases} (i-1,i) & 2 \geq i \geq \phi, 1 \geq j \geq q \quad \textbf{North} \\ (i,j+1) & 1 \geq i \geq \phi, 1 \geq j \geq q-1 \quad \textbf{East} \\ (i+1,j) & 2 \geq i \geq \phi-1, 1 \geq j \geq q \quad \textbf{South} \\ (i,j-1) & 1 \geq i \geq \phi, 2 \geq j \geq q \quad \textbf{West} \end{cases}$$

To form row-normalized neighborhood matrix **W**, a non-row-normalized neighborhood matrix **C** and diagonal matrix D are used, such that **W=D**$^{-1}$**C**. is formed by putting "1"s for neighborhoods of *(i,j)*th pixel of the spatial framework and by putting zeros for the rest of the entries. Values of **D** matrix can be formed as $d_{ij} = \sum_{i=1}^{n} c_{ij}$ d_{ij}=0. In other words, **W** matrix is formed by dividing non-zero elements of **C** by corresponding diagonal element of **D**. Figure 8(a) illustrates the spatial framework and Figure 8(b) shows **W** matrix for the problem size 16.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

(a)

$$\begin{bmatrix}
0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{3} & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & \frac{1}{3} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0
\end{bmatrix}$$

(b)

Figure 8. 4-by-4 (ϕ-by-q) spatial framework and row-normalized neighborhood matrix W with 4-neighbors