Sparse Inverse Kernel Gaussian Process Regression

Kamalika Das[*], Ashok N. Srivastava [†]
[*]UARC University of California, Santa Cruz,
NASA Ames Research Center, MS 269-1, Moffett Field, CA-94035
Email:Kamalika.Das@nasa.gov
[†]NASA Ames Research Center,
MS 269-3, Moffett Field, CA-94035
Email:Ashok.N.Srivastava@nasa.gov

A shorter version of this paper was published in NASA Conference on Intelligent Data Understanding 2011[4].

Abstract

Regression problems on massive data sets are ubiquitous in many application domains including the Internet, earth and space sciences, and finances. Gaussian Process regression is a popular technique for modeling the input-output relations of a set of variables under the assumption that the weight vector has a Gaussian prior. However, it is challenging to apply Gaussian Process regression to large data sets since prediction based on the learned model requires inversion of an order n kernel matrix. Approximate solutions for sparse Gaussian Processes have been proposed for sparse problems. However, in almost all cases, these solution techniques are agnostic to the input domain and do not preserve the similarity structure in the data. As a result, although these solutions sometimes provide excellent accuracy, the models do not have interpretability. Such interpretable sparsity patterns are very important for many applications. We propose a new technique for sparse Gaussian Process regression that allows us to compute a parsimonious model while preserving the interpretability of the sparsity structure in the data. We discuss how the inverse kernel matrix used in Gaussian Process prediction gives valuable domain information and then adapt the inverse covariance estimation from Gaussian graphical models to estimate the Gaussian kernel. We solve the optimization problem using the alternating direction method of multipliers that is amenable to parallel computation. We compare the performance of this algorithm to different existing methods for sparse covariance regression in terms of both speed and accuracy. We demonstrate the performance of our method in terms of accuracy, scalability and interpretability on two different satellite data sets from the climate domain.

Keywords: sparse regression, Gaussian processes, Earth science data, ADMM

I. Introduction

In many application domains, it is important to predict the value of one feature based on certain other measured features. For example, in the Earth Sciences, predicting the precipitation at one location given the humidity, sea surface temperature, cloud cover, and other related factors is an important problem in climate modeling. For such problems, simple linear regression based on minimization of the mean squared error between the true and predicted values can be used for modeling the relationship between the input and the target features. In decision support systems which use these predictive algorithms, a prediction with low confidence may be treated differently than if the same prediction was given with high-confidence. Thus, while the predicted value from the regression function is clearly important, the confidence in the prediction is equally important. A simple model such as linear regression does not provide us with that information. Also, models like linear regression, in spite of being easy to fit and being highly scalable, fail to capture nonlinear relationships in the data. Gaussian Process regression (GPR) is one regression model that can capture nonlinear relationships and outputs a distribution of the prediction where the variance of the predicted distribution acts as a measure of confidence in the prediction. Moreover, the inverse kernel (or covariance) matrix has many interesting properties along the gaussian graphical model perspective, that can be exploited for better understanding relationships within the training examples. Depending on the nature of the data, these relationships can indicate dependencies (causalities) for certain models.

However, predictions based on GPR method, requires inversion of a kernel (or covariance) matrix of size $n \times n$, where n is the number of training instances. This kernel inversion becomes a bottleneck for very large data sets. Most of the existing methods for efficient computation in GPR involve numerical approximation techniques that exploit data sparsity. While this does speed up GPR computations, one serious drawback of these approximations is that the resulting GPR model loses interpretability. Even if we get reasonably accurate predictions, we fail to unearth significant connections between the training points or identify the most influential training points for a specific set of test points. Sometimes such relationships reveal important information about the application domain. For example, in studies of climate network, we can find which locations on the Earth's grid have significant impact on a specific group of test locations. Depending on whether the regression problem is time-delayed with respect to the target variable, this method can also reveal unknown teleconnection patterns which are otherwise extremely difficult to find based on existing climate indices.

In this paper we propose a sparse GPR algorithm which not only scales to very large data sets but also allows us to construct a complete yet sparse inverse covariance matrix, thereby facilitating interpretability. The method proposed in this paper induces sparsity by introducing a regularizer in a pseudo negative log likelihood objective used for covariance selection. This forces the algorithm to seek a parsimonious model for GPR prediction having excellent interpretability. One of the highlights of the solution technique used in this paper is a completely parallelizable framework for solving the inverse covariance estimation problem using the alternating direction method of multipliers (ADMM) that allows us to exploit modern parallel and multi-core architectures. This also addresses the situation where the entire covariance matrix cannot be loaded into memory due to size limitations.

The rest of the paper is organized as follows. In the next section (Section II) we present some background material related to GPR and some existing methods of solving the GPR problems. In Section III we discuss the equivalence between inverse kernel and covariance matrices. Next we present our new sparse inverse covariance matrix using ADMM technique (Section IV). Experimental results are discussed in Section V. We conclude the paper in Section VI.

II. Background: Gaussian Process Regression

Since this paper proposes a technique of model fitting using Gaussian Process regression, we start with a brief review of it here. Rasmussen and Williams [18] provide an excellent introduction on this subject. Gaussian Process regression is a generalization of standard linear regression. If $\mathbf{X}$ is the training data set having n multidimensional observations (rows) $\mathbf{x}_1, \ldots, \mathbf{x}_n$, with each $\mathbf{x}_i \in \mathbb{R}^D$ and the corresponding target is represented by a $n \times 1$ vector $\mathbf{y}$, then the standard linear regression model is:

$$f(\mathbf{x}) = \mathbf{x}\mathbf{w}^T, \qquad y = f(\mathbf{x}) + \epsilon$$

where $\mathbf{w}$ is a D-dimensional weight vector of parameters and ϵ is additive Gaussian noise such that $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Assuming that we choose the prior distribution of the weights to be Gaussian with mean zero and covariance Σ_p, the posterior distribution of the weights, following Bayesian inferencing techniques, can be written as:

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}) \sim \mathcal{N}\left(\frac{1}{\sigma^2} A^{-1}\mathbf{X}^T y, A^{-1}\right)$$

where $A = \sigma^{-2}\mathbf{X}^T\mathbf{X} + \Sigma_p^{-1}$. Given the posterior and the likelihood, the predictive distribution of a test input $\mathbf{x}^*$ is obtained by averaging over all possible models ($\mathbf{w}$) to obtain:

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{X}, \mathbf{y}) \sim \mathcal{N}\left(\frac{1}{\sigma^2}\mathbf{x}^* A^{-1} X^T \mathbf{y}, \mathbf{x}^* A^{-1}\mathbf{x}^{*T}\right)$$

Using a kernel (covariance) function $k(\mathbf{x}_i, \mathbf{x}_j)$ in place of a mapping from input space to an N-dimensional space, and applying some algebraic manipulations, we can write the predictive mean and variance of the posterior distribution as

$$\widehat{\mathbf{y}}^* = K^*(\sigma^2 I + K)^{-1}\mathbf{y} \tag{1}$$

$$C = K^{**} - K^*(\sigma^2 I + K)^{-1} K^{*T} \tag{2}$$

where the ij^{th} entry of K is $k(\mathbf{x}_i, \mathbf{x}_j)$ and K^* and K^{**} are similarly the cross covariance matrices involving the test point $\mathbf{x}^*$. Equations 1 and 2 pose significant computational challenge due to the requirement of inverting the covariance matrix K of size n^2. If the number of observations n is large, the $O(n^3)$ operation can be a bottleneck in the process of using Gaussian Process regression.

In the next section, we discuss several techniques that have been proposed in the literature for approximating the inverse matrix for large datasets.

A. Existing methods for efficient GP computation

Approximations are introduced in the Gaussian Process literature for either finding closed-form expressions for intractable posterior distributions or for gaining computational advantage for large data sets. Here we are interested in the second goal and, therefore, briefly discuss the existing research in this area. Smola and Bartlett [19] describe a sparse greedy method that does not require evaluating the full covariance matrix K and finds an approximation to the maximum aposteriori estimate by selecting an 'active' subset of columns of K by solving an expensive optimization problem. The running time of the numerical approximation is reduced from $O(n^3)$ to $O(nm^2)$ where m ($m \ll n$) is the rank of the matrix approximation.

A related approach of low rank matrix approximation called the subset of regressors method [27] involves selecting the principal sub-matrix of the unperturbed covariance matrix K by matrix factorization. Though this method has been found to be numerically unstable, recent research by Foster *et al.* [9] has shown that if we use partial Cholesky decomposition to factorize the covariance matrix and perturb the low rank factor such that independent rows and columns form the principal sub-matrix, then the approximation we get is numerically stable. The authors report excellent accuracy using their approximation calculations when the rank of the reduced matrix is a small factor (5) times the rank of the original data matrix X.

The generalized Bayesian committee machine [24] is another approach for reducing the computational complexity of any kernel-based regression technique, by dividing the data arbitrarily into M almost equal sized partitions, training a different estimator on each partition, and combining the estimates given by the different estimators using the inverse of the variance to ensure that least certain predictions are given the smallest weights in the final prediction. This method allows us to choose M to be equal to $K\alpha$ so that it becomes linear in K in computational complexity. The

Bayesian Committee Machine weights the training data based on the test points using a block diagonal approximation and, therefore, the model needs to be retrained every time a new test set comes in. A related method recently proposed by Das and Srivastava [3] works for multimodal data. It partitions the input space into multiple clusters, with each one corresponding to one mode of the data distribution. Then, each cluster is modeled using a normal distribution and all points which are not modeled by any of the normal distributions are grouped using a separate cluster. Each cluster learns a separate GP model and a weighted sum based prediction is used for the gating.

A recent development is the ℓ_1 penalized GPR method (GPLasso) introduced by Yan and Qi [29] in which the authors explore sparsity in the output rather than the input. They propose a GPR technique that minimizes the Kullback-Leibler divergence between the posterior distributions of the exact and the sparse solutions using a ℓ_1 penalty on the optimization. They pose this problem as a LASSO optimization [23] and solve a rank reduced approximate version of this using the Least Angle Regression (LARS) method [8]. The authors present this work as a pseudo output analogy of the work by Snelson *et al.* [20]. Quiñonero-Candela and Rasmussen [17] provide a unifying view of all sparse approximation techniques for Gaussian Process regression by analyzing the posterior and reinterpreting each algorithm as an exact inferencing method using approximate priors.

All the methods discussed in this section apply some form of numerical approximation technique to reduce the rank of the kernel matrix for efficient matrix inversion. As a result, they often lose model interpretability — a value at any position of the reduced rank inverted matrix cannot be traced back to any cell of the original kernel. In many domains, however understanding the sparsity structure is important. For example, in Earth Sciences, it is not only important to get good predictions from the GPR model, but it is also important to understand how different geographical regions are connected and how these locations influence one another. Unfortunately, none of the efficient GPR techniques allow this. Our proposed technique in the next section not only learns a sparse GP model but also allows domain scientists to draw conclusions about the sparsity structure by studying the inverse covariance matrix.

III. SPI-GP:Sparse Gaussian Process using inverse covariance estimation

Let $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ be a set of multi-dimensional gaussian observations such that

$$\mathbf{x}_i \sim \mathcal{N}(\mu, \Sigma) \in \mathbb{R}^d$$

where $\mu \in \mathbb{R}^d$ and $\Sigma \in \mathbb{R}^{d \times d}$ are the mean and covariance matrices. While the mean μ measures the center of the distribution, the covariance matrix Σ measures the pairwise (linear) relationship between the variables. It is well known that a value of 0 at any cell of Σ implies independence of the observations:

$$\Sigma_{i,j} = 0 \Rightarrow P(\mathbf{x}_i \mathbf{x}_j) = 0$$

which means $\mathbf{x}_i$ and $\mathbf{x}_j$ are independent. In many cases, we may be interested in studying how two variables influence each other when the information about the other variables are taken into consideration. One way of doing this is by studying the inverse covariance matrix, also known as the concentration matrix or precision matrix denoted by Σ^{-1}. Unlike Σ, a value of 0 in any cell of Σ^{-1} implies conditional independence among those variables [1]. For example, $\mathbf{x}_i$ and $\mathbf{x}_j$ are conditionally independent, given all the other variables, if Σ^{-1}=0. Mathematically,

$$\Sigma^{-1}_{i,j} = 0 \Rightarrow P(\mathbf{x}_i \mathbf{x}_j | \mathbf{x}_{-i,-j}) = 0$$

where $\mathbf{x}_{-i,-j}$ denotes all the variables other than $\mathbf{x}_i$ and $\mathbf{x}_j$. Note that independence of elements implies conditional independence but not vice-versa *i.e.* a value of 0 at any cell of Σ implies that the corresponding location of Σ^{-1} is also 0; but a non-zero value at any cell of Σ matrix does not imply that the corresponding cell of Σ^{-1} will also be non-zero. The reason for studying Σ^{-1} rather than Σ, is for many gaussian distributed variables, there is more sparsity in the inverse covariance matrix than in the covariance matrix and this sparsity reveals interesting data relationships. It has been shown in [10], that inverting a covariance matrix (with the additional assumption that the inverse is sparse) is equivalent to learning a graphical model, where each node in the model corresponds to a feature and the absence of an edge between any two signifies that those features are conditionally independent.

In the case of GPR, the kernel matrix between the observations (see Eqn. 1 and 2) can be viewed as a covariance matrix among the function outputs. Formally, a gaussian process is defined as a collection of random variables, any finite number of which is jointly gaussian. Hence, it is a distribution over functions, completely specified by its mean function and covariance function as,

$$f(\mathbf{x}_i) \sim GP(m(\mathbf{x}_i), k(\mathbf{x}_i, \mathbf{x}_j))$$

where $m(\mathbf{x}_i) = E[f(\mathbf{x}_i)]$ and $k(\mathbf{x}_i, \mathbf{x}_j) = E[f(\mathbf{x}_i) - m(\mathbf{x}_i)][f(\mathbf{x}_j) - m(\mathbf{x}_j)]$ are the mean function and covariance function of some real process $f(\mathbf{x}_i)$. Note that $f(\mathbf{x}_i)$ are random variables and GP fits a distribution over all possible $f(\mathbf{x}_i)$. In our case since $f(\mathbf{x}_i)$'s are linear functions $f(\mathbf{x}_i) = \mathbf{x}_i\mathbf{w}^T$, the mean and covariance of GP can be stated as,

$$m(\mathbf{x}_i) = E[f(\mathbf{x}_i)] = \mathbf{x}_i E[\mathbf{w}^T] = 0$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = E[f(\mathbf{x}_i)f(\mathbf{x}_j)] = \mathbf{x}_i E[\mathbf{w}^T\mathbf{w}]\mathbf{x}_i^T = \mathbf{x}_i \Sigma_p \mathbf{x}_i^T$$

where $\mathbf{w} \sim N(0, \Sigma_p)$ denotes the prior distribution of the weights. The covariance function k, also known as the kernel function specifies the covariance between a pair of random variables

$$\text{cov}(f(\mathbf{x}_i), f(\mathbf{x}_j)) = E[f(\mathbf{x}_i)f(\mathbf{x}_j)] = k(\mathbf{x}_i, \mathbf{x}_j)$$

Therefore, a kernel function computed over the pairwise input points is equivalent to a covariance between the outputs. There are several choices of the kernel functions available. In this paper we have used the widely used gaussian radial basis function (rbf) kernel:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$

where σ is known as the bandwidth parameter which is typically learned from the data.

In many GPR applications, it is not only important to get good prediction accuracy, but also understand the model. For example, in Earth Sciences, teleconnections [12] reveal important symmetric and sometimes causal relationships among different events observed in geographically distant locations and can be studied by exploiting sparsity in the inverse kernel in GPR. Another possible application area is the study of climate networks [21][26][22][25][6]. Fig. 1 (left) shows the observed precipitation data of the world overlaid on a 360×720 grid. Figs. 1 (center and right) show a kernel or similarity matrix generated from the data and the corresponding inverse covariance matrix. The highlighted row and column correspond to the location marked in white on the world map. Each cell in the kernel denotes the similarity between the precipitation values of a pair of grid locations. If a cell has a value of 0 in the kernel matrix, it implies independence of those two points, whereas, a 0 value in the inverse kernel matrix implies conditional independence between the pair of points, given all the other observations. Since absolute independence is a much more strict condition to satisfy for two random variables, compared to conditional independence, the inverse kernel is a much sparser matrix to study than the kernel. This is clearly observed in Figure 1. Therefore, in this paper we are interested in studying the sparsity pattern of the inverse covariance matrix, with the information that sparsity patterns in the inverse covariance matrix leads to conditional independence among the locations of interest.

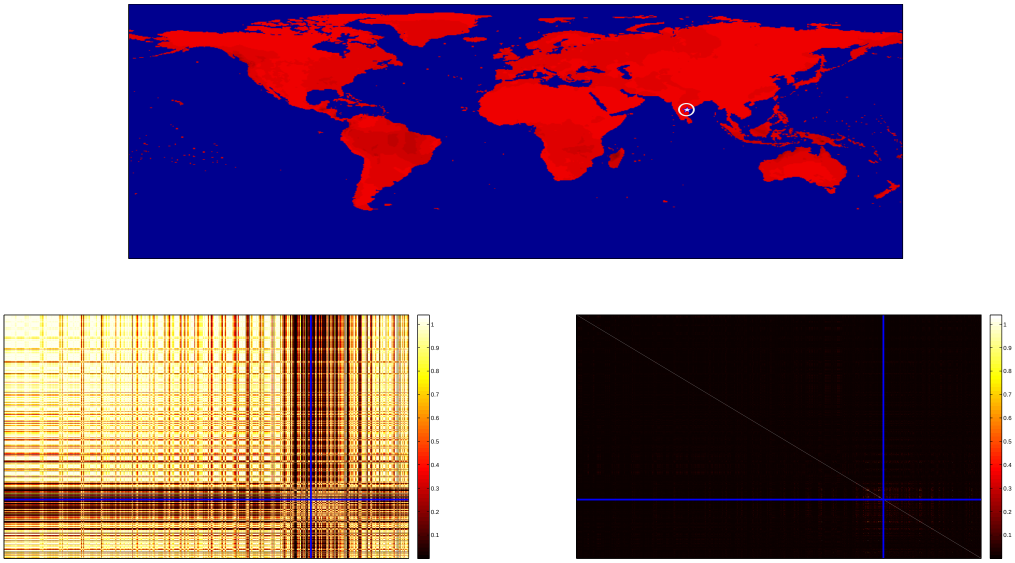


Fig. 1. Precipitation data of the world map (top figure). Note that the data is only available for land (the ocean locations have fill values of -9999). The figure in the center shows a kernel in which similarity is computed between every pair of locations from the precipitation data. Note the location marked with a circle on the left figure corresponds to the row and column in blue on the center and right figure. The right figure shows the inverse kernel matrix.

IV. Sparse covariance selection

There exist several techniques in the literature for solving the inverse covariance estimation problem also known as the covariance selection problem.

Given a data set containing d features, Meinshausen *et al.* [16] infers the graphical model (and therefore the inverse covariance matrix) by taking one variable at a time and then finding all the connections of that variable with all of the other ones. For each variable d_i in the dataset, the method constructs a lasso regression problem by taking all the other variables as inputs and d_i as the target with an additional sparsity constraint on the solution weights. The non-zero entries of the weight vector signifies a connection between that feature and the target d_i. To deal with inconsistencies among the connections, the authors have proposed two schemes: (1) in the **AND** technique, an edge is established in the graphical model between any two features d_i and d_j iff both d_i and d_j have non-zero entries in the weight vector when they are each used as target in different lasso problems, and (2) in the **OR** scheme, an edge is established if either d_i or d_j has a non-zero weight when the other is taken as the target. One serious drawback of this method is the number of independent lasso problems increases linearly with the size of the feature space.

Banerjee *et al.* [1] propose a different solution to the inverse covariance selection problem. They show that based on Dempster's theory [5], estimating the inverse covariance matrix is equivalent to minimizing the pseudo negative log likelihood. The objective function takes the form:

$$\mathbf{Tr}(KS) - \log det(S)$$

where K is the empirical covariance (or kernel) matrix and S is the desired inverse of K i.e. $S = K^{-1}$, $\mathbf{Tr}(\cdot)$ is the trace of a matrix, and $det(\cdot)$ is the matrix determinant. Solution to the above equation is stable when an additional sparsity constraint is imposed on the inverse, *i.e.*

$$\mathbf{Tr}(KS) - \log det(S) + \lambda \left\| S \right\|$$

where λ controls the degree of sparsity. This is a convex optimization problem and in order to solve this, the authors propose a block-wise interior point algorithm.

Friedman *et al.* [10] generalizes both these papers and present a very efficient algorithm based on the lasso technique. Their objective function is the same as used by Banerjee *et al.* [1] *i.e.* they try to maximize the log likelihood of the model with the additional sparsity constraint. They show that the solution proposed by Meinshausen [16] is an approximation of the log likelihood estimate proposed by Banerjee *et al.* [1]. They propose a new algorithm based on coordinate descent to solve the same trace minimization problem. This algorithm is based on recursively solving lasso subproblems for each variable until convergence. The authors note that this new algorithm is at least 50 to 4000 times faster than existing techniques and therefore scales to much larger data sets.

Very recently Hsieh et al. [13] proposed a new approach for solving the maximum likelihood problem much faster than existing methods. Unlike existing techniques that use first order gradient descent for optimization, the authors resort to a Newton's method based quadratic approximation that accounts for the structure of the MLE problem. This method can scale to about 10,000 data points.

However, there is one drawback common to all these optimization techniques. All these techniques assume that the data can be loaded in computer memory for the analysis. Unfortunately, in applications such as Earth Sciences, most datasets are massive — they contain millions of observations (locations) and therefore constructing a full covariance matrix in memory is itself impossible, leaving aside the computational power necessary to run these optimization techniques for inverse estimation. To solve the large scale inverse covariance estimation problems which do not fit into the memory of one machine, in this paper we propose our SPI-GP method which works by distributing the workload among a network of machines. The technique we follow is based on the method of Alternating Direction Method of Multipliers (ADMM) which is a distributable algorithm for solving very large convex optimization problems. We give a brief overview of ADMM technique in the next section.

A. Alternating Direction Method of Multipliers for convex problems

Alternating Direction Method of Multipliers (ADMM) [11][7][2] is a decomposition algorithm for solving separable convex optimization problems of the form:

$$\min \quad G_1(x) + G_2(y) \quad \text{subject to} \quad Ax - y = 0, \quad x \in \mathbb{R}^n, \quad y \in \mathbb{R}^m$$

where $A \in \mathbb{R}^{m \times n}$ and G_1 and G_2 are convex functions. The algorithm derivation is as follows. First, the augmented Lagrangian is formed:

$$L_\rho(x, y, z) = G_1(x) + G_2(y) + z^T(Ax - y) + \rho/2 \, \|Ax - y\|_2^2$$

where ρ is a positive constant known as the penalty parameter. ADMM iterations can then be written as:

$$\begin{aligned} x^{t+1} &= \min_x \left\{ G_1(x) + z^{tT} Ax + \rho/2 \left\| Ax - y^t \right\|_2^2 \right\} && (3) \\ y^{t+1} &= \min_y \left\{ G_2(y) - z^{tT} y + \rho/2 \left\| Ax^{t+1} - y \right\|_2^2 \right\} && (4) \\ z^{t+1} &= z^t + \rho \left(Ax^{t+1} - y^{t+1} \right) && (5) \end{aligned}$$

This is an iterative technique where t is the iteration counter, and the initial vectors y^0 and z^0 can be chosen arbitrarily. ADMM can be written in a different form (known as the scaled form) by combining the linear and quadratic terms of the Lagrangian:

$$z^T(Ax - y) + \rho/2 \left\| (Ax - y) \right\|_2^2 \;=\; \rho/2 \left\| (Ax - y) + (1/\rho) z \right\|_2^2 - 1/(2\rho) \left\| z \right\|_2^2$$

Now scaling the dual variable $p = (1/\rho)z$, the iterations of ADMM become:

$$\begin{aligned} x^{t+1} &= \min_x \left\{ G_1(x) + \rho/2 \left\| Ax - y^t + p^t \right\|_2^2 \right\} && (6) \\ y^{t+1} &= \min_y \left\{ G_2(y) + \rho/2 \left\| Ax^{t+1} - y + p^t \right\|_2^2 \right\} && (7) \\ p^{t+1} &= p^t + \rho \left(Ax^{t+1} - y^{t+1} \right) && (8) \end{aligned}$$

It has been argued [11] that ADMM is very slow to converge especially when high accuracy is desired. However, ADMM converges within a few iterations when moderate accuracy is desired. This can be particularly useful for many large scale problems similar to the one we consider in this paper.

Critical to the working and convergence of the ADMM method is the termination criterion. The primal and dual residuals are:

$$r_p^{t+1} = Ax^{t+1} - y^{t+1} \quad \text{(primal residual)}$$

$$r_d^{t+1} = \rho A(y^{t+1} - y^t) \quad \text{(dual residual)}$$

A reasonable termination criterion is when either the primal or the dual residuals are below some thresholds *i.e.*

$$\left\| r_p^{t+1} \right\|_2 \leq \epsilon_p \quad \text{and} \quad \left\| r_d^{t+1} \right\|_2 \leq \epsilon_d.$$

where ϵ_p and ϵ_d are the primary and dual feasibility tolerances. Using user-defined values for ϵ_1 and ϵ_2, these tolerances can be stated as,

$$\epsilon_p = \epsilon_1 \sqrt{m} + \epsilon_2 \max\left(\left\|Ax^{t+1}\right\|_2, \left\|y^{t+1}\right\|_2\right)$$

$$\epsilon_d = \epsilon_1 \sqrt{n} + \epsilon_2 \left\|A^T p^{t+1}\right\|_2.$$

Although we have used ADMM here for solving the inverse kernel estimation problem, it has been used for many tasks such as text mining [15], classification [30], gene expression network optimization, image reconstruction and de-blurring using GPU [28] and many more.

In the next section we discuss the ADMM update rules for the sparse inverse covariance estimation problem.

B. Alternating Direction Method for sparse inverse kernel estimation

We start with the prior assumption that the inverse kernel matrix K^{-1} is sparse. This is a reasonable assumption when studying climate data, because given a location *i.e.* any row of the inverse kernel matrix, there are few major locations which influence this location.

With such an assumption, the ADMM algorithm is as follows. Let K be the observed kernel matrix between the grid locations. For a moderate sized K, one can search over all sparsity patterns, since for a fixed sparsity pattern the log likelihood estimate of K is a tractable problem. However, this becomes very challenging for large K. One technique which has been used earlier for sparse covariance selection problem [1] is to minimize the negative log likelihood of $S = K^{-1}$ with respect to the observed data with a penalty term added to induce sparsity. This resulting objective function can be written as

$$\min \quad \mathbf{Tr}(KS) - \log det(S) + \lambda \left\|S\right\|_1$$

where $\left\|\cdot\right\|_1$ is the ℓ_1-norm or the sum of the absolute values of the entries of a matrix and λ is a constant which determines the amount of sparsity. Larger the value of λ, sparser is the solution S. The ADMM version of this problem can be written as follows:

$$\min \quad \mathbf{Tr}(KS) - \log det(S) + \lambda \left\|Y\right\|_1 \quad \text{subject to} \quad S - Y = 0$$

By constructing the augmented Lagrangian and using the derivations given in Section IV-A for the scaled version of the problem, the ADMM updates for the above estimation problem are:

$$\begin{aligned} S^{t+1} &= \min_x(\mathbf{Tr}(KS) - \log det(S) + \rho/2 \left\|S - Y^t + P^t\right\|_F) && (9) \\ Y^{t+1} &= \min_y \left(\lambda \left\|Y\right\|_1 + \rho/2 \left\|S^{t+1} - Y + P^t\right\|_F\right) && (10) \\ P^{t+1} &= P^t + \left(S^{t+1} - Y^{t+1}\right) && (11) \end{aligned}$$

with $\left\|\cdot\right\|_F$ denoting the Frobenius norm of a matrix. These updates can be simplified further. Taking the derivative of Eqn. 9 and setting it to 0 we get,

$$\begin{aligned} & K - S^{-1} + \rho(S - Y^t + P^t) = 0 \\ \Rightarrow\ & \rho S - S^{-1} = \rho(Y^t - P^t) - K \end{aligned}$$

Now let $Q\Lambda Q^T$ be the eigen decomposition of $\rho(Y^t - P^t) - K$. Therefore, continuing from the previous step,

$$\begin{aligned} & \rho S - S^{-1} = \rho(Y^t - P^t) - K \\ \Rightarrow\ & \rho S - S^{-1} = Q\Lambda Q^T \\ \Rightarrow\ & \rho Q^T S Q - Q^T S^{-1} Q = Q^T Q \Lambda Q^T Q \\ \Rightarrow\ & \rho\widehat{S} - \widehat{S}^{-1} = \Lambda \quad [\text{since } Q^T Q = Q Q^T = I] && (12) \end{aligned}$$

where $\widehat{S} = Q^T S Q$. Solution to Eqn. 12 can easily be found noting that the right hand side is a diagonal matrix of the eigenvalues λ_i's. For each diagonal entry of $\widehat{S}_{ii}$, $\forall i = 1 : n$, we have

$$\rho\widehat{S}_{ii} - \widehat{S}_{ii}^{-1} = \lambda_i$$

which, using the formula of finding the roots of a quadratic equation is

$$\widehat{S}_{ii} = \frac{\lambda_i + \sqrt{\lambda_i^2 + 4\rho}}{2\rho}$$

Therefore, $S = Q\widehat{S}Q^T$ is the optimal value of the S minimization step. In our studies we have used the full eigen decomposition $Q\Lambda Q^T$. To reduce complexity, we can use power method to extract the top few eigenvalue-eigenvector pairs (based on a threshold of how much of variance is captured) and set the other pairs to 0. This would set those $\hat{S}_{ii} = 1/\sqrt{\rho}$, while keeping the others the same. We plan to study the effect of the accuracy and running time of SPI-GP on the percent of variance captured by the eigen decomposition method in our future work.

Eqn. 10 can also be simplified further and can be written as the element-wise soft thresholding operation:

$$Y_{ij}^{t+1} = \Im_{\lambda/\rho}\left(S_{ij}^{t+1} + P_{ij}^{t}\right)$$

Once the sparse inverse kernels are constructed, they can be plugged back into Eqn. 1 and 2 to compute the final prediction mean and variance. Note that, the other kernel matrices K^* and K^{**} are computed among the test points and hence they are fairly small. Moreover, these matrices do not require matrix inversion.

In the next section we describe the SPI-GP algorithm in details.

C. SPI-GP: algorithm description

The SPI-GP algorithm is based on the ADMM technique described in the previous section. Alg. 1 presents the pseudo-code of the algorithm. The inputs are the kernel K, algorithm parameters λ and ρ, number of iterations $numIter$ and the error tolerances ϵ_1 and ϵ_2. The output of the algorithm is the estimated inverse of K in $S = K^{-1}$. The algorithm proceeds in an iterative fashion. In every iteration, an eigen decomposition is performed of the matrix

$$[Q \quad \Lambda] = \rho(Y^{t-1} - P^{t-1}) - K.$$

The eigenvalues Λ and eigenvectors Q are used to update the S variable. The Y-update is a soft thresholding operation of $(S^t + P^{t-1})$ with threshold λ/ρ. Finally, the P-update is a linear dual variable update. Also during each iteration, the primal and dual residuals r_p and r_d are computed along with the corresponding error thresholds. Whenever the residuals become less than the error thresholds, the algorithm stops. The result is returned in the matrix S. In our experiments we have chosen $\rho = 1$

Computational complexity of ADMM: Since the algorithm requires eigen decomposition for every S update, and the Y and P updates are constant time operations, the runtime complexity is $O(mn^3)$, where m is the number of iterations and n is the size of the dataset (training points).

Algorithm 1: SPI-GP: ADMM for Sparse Kernel Inversion

```
Input: $K$, $\rho$, $\lambda$, $numIter$, $\epsilon_1$, $\epsilon_2$
Output: $S = K^{-1}$
Initialization: $Y^1 = 0, P^1 = 0$
begin
    for t=2 to numIter do
        $[Q \quad \Lambda] = \textbf{evd}[\rho(Y^{t-1} - P^{t-1}) - K]$;
        for i=1 to n do
            $\widehat{S}_{ii} = \frac{\lambda_i + \sqrt{\lambda_i^2 + 4\rho}}{2\rho}$;
        $S^t = Q\widehat{S}Q^T$;
        $Y^t = \textbf{softThreshold}[(S^t + P^{t-1}), \lambda/\rho]$;
        $P^t = P^{t-1} + (S^t - Y^t)$;
        $r_p = \|S^t - Y^t\|_F$;
        $r_d = \|-\rho(S^t - Y^{t-1})\|_F$;
        $\epsilon_p = \epsilon_1\sqrt{n} + \epsilon_2 \max(||S^t||_F, ||Y^t||_F)$;
        $\epsilon_d = \epsilon_1\sqrt{n} + \epsilon_2||\rho P^t||_F$;
        if $(r_p < \epsilon_p)$ AND $(r_d < \epsilon_d)$ then
            break;
```

Convergence of ADMM: In order to ensure convergence of ADMM, two basic assumptions are necessary: (1) the functions G_1 and G_2 are closed, proper and convex, and (2) the unaugmented Lagrangian has a saddle point. Based on these two conditions, it can be shown that [2]:

- primal residual approaches 0 i.e. $r^t \to 0$ as $t \to \infty$
- the objective function approaches the optimal value
- dual variable P approaches feasibility

In practice however, ADMM may be slow to converge. This type of algorithms, are therefore, more useful when moderate accuracy is necessary within a relatively few iterations. Although this algorithm is slow and sometimes has convergence issues, it is the only method that is amenable to parallel computing which is essential for many large data sets that do not fit in the main memory of a single machine.

D. SPI-GP: distributed implementation

As we have discussed earlier, ADMM is amenable to distributed computation in a network of machines. This becomes particularly important when the data does not fit into the memory of one machine. This form of ADMM is known as consensus optimization. In this form, the objective function G_1 needs to be decomposable across ℓ nodes $M_1, \ldots, M_\ell$ as follows:

$$\min \quad \textstyle\sum_{i=1}^{\ell} G_1(x_i) + G_2(y) \text{ subject to } \quad Ax_i - y = 0, \quad x_i \in \mathbb{R}^n, \quad y \in \mathbb{R}^m$$

where x_i is the i-th block of data and is stored at machine M_i. The solution to this optimization is the same as given in Section IV-A. The update rules can be written as,

$$\begin{aligned}
x_i^{t+1} &= \min_{x_i} \left\{ G_1(x_i) + z^{tT} A x_i + \rho/2 \left\| Ax_i - y^t \right\|_2^2 \right\} \\
y^{t+1} &= \min_{y} \left\{ G_2(y) + \sum_{i=1}^{\ell} \left(-z_i^{tT} y + \rho/2 \left\| Ax_i^{t+1} - y \right\|_2^2 \right) \right\} \\
z_i^{t+1} &= z_i^t + \rho \left(Ax_i^{t+1} - y^{t+1} \right)
\end{aligned}$$

Unfortunately, the above method cannot be applied for the optimization of the inverse covariance matrix in our case. This is because $\log det(S)$ is not a decomposable function.

Therefore, to solve this problem for large kernel matrices, we use the ScaLAPACK routine of Matlab. It allows the kernel matrix to be distributed across different machines, but still compute the eigen decomposition correctly. For a Matlab implementation, this is done using the co-distributed array data structure and an overloaded *eig* function. It should be noted here that this method *does not* attempt to speed up the GPR process. Instead, it makes GPR possible for extremely large data sets where the entire kernel matrix cannot be loaded in the main memory due to size limitations.

V. Experimental results

For the performance study of SPI-GP, the experimental results are reported on a synthetic multivariate Gaussian distribution data set, two relatively small benchmark data sets from the GPR literature, and two different real life climate domain data sets. For generating the multivariate Gaussian, we fix the number of dimensions and samples. We then generate a sparse inverse covariance matrix with all zeros and ones along the diagonal. We randomly insert ones at certain locations in our inverse covariance. We make this inverse matrix symmetric and positive definite (by making the minimum eigenvalue positive). Finally we invert this matrix and draw Gaussian samples with zero mean which becomes the covariance matrix input to our algorithm. Using this data set we demonstrate the scalability of the distributed SPI-GP method on a cluster of computing nodes. The benchmark data sets that we have used, are used in the Gaussian Process Regression literature for performance validation [29]. The first data set (Snelson) is a small synthetic data set used by Snelson *et al.* [20] in the sparse GP paper and is being used only to illustrate model quality of SPI-GP compared to other methods. The other data set, pumadyne-8nm, has approximately 7000 points and is a realistic simulation of the dynamics of a Puma 560 robot arm. The regression task in this data set is to predict the angular acceleration of one of the robot arm's links using angular positions, velocities and torques of the robot's arm as input.

The real world data sets used in this paper are both from the Earth Science domain. The first one is a historical data set consisting of NCEP/NCAR features available at `http://www.cdc.noaa.gov/data/gridded/data.ncep.reanalysis.html`[14] and cross-matched normalized difference vegetation index (NDVI) data (NDVI) from the National Oceanic and Atmospheric Administrations Advanced Very High Resolution Radiometer (NOAA/AVHRR). The climate variables in this data set are include pressure, sea surface temperature, temperature, and precipitation from 1982 till 2002. Each variable is observed at a $0.5°$ resolution over the entire grid. The data used here are composites of observations over a month. Thus there are 360×720=259200 values for each variable vectorized and stored as a single row corresponding to a time point (a month). Therefore, each variable has 12×21 = 252 rows in the data set, each having 259200 columns. Note that some variables are observed only in land while others only in ocean. For any variable, the locations which do not contain any meaningful data has a fill value of -9999.0.

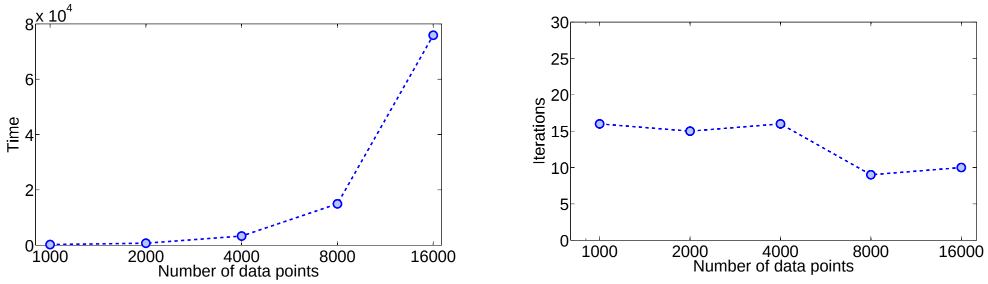
The second real world data set that we have used in our experiments is the MODerate-resolution Imaging Spectroradiometer (MODIS) data providing 500-meter surface reflectance data for the state of California adjusted using a bidirectional reflectance distribution function (BRDF). The data is collected at intervals of every 8 days and stored as 1203×738 image file. Each image data is recorded for seven different wavelengths corresponding to seven different channels. Since these channels observe the same spatial location at the same time instances, there is a high correlation among the different bandwidths. Therefore, Gaussian Process regression can be used to model the relationship between the channels for creating *Virtual* Sensors and detecting changes in land cover. Based on careful exploratory analysis and domain expert feedback, three features (Band1 620 - 670 nm, Band4 545 - 565 nm and Band5 1230 - 1250 nm) have been chosen to model the target (Band6 1628 - 1652 nm). The data set contains nine years worth of data (2001-2009) arranged at the top level by the number of years where each year contains forty six (collected every 8 days) images and has approximately 15 million observations per year.

Experimental setup: The algorithms have been run using a 64-bit 2.66 GHz Intel Xeon dual quad core Dell Precision 690 desktop running Red Hat Enterprise Linux version 5.7 having 24 GB of physical memory. The SPI-GP algorithm is parallelizable and has been executed on a 64-bit Linux cluster consisting of 16 slave nodes where each node is a dual processor 1-U server containing two quad-core Intel Xeon 2.66GHz processors totaling 128 cores and 128GB Ram (1Gb/Core). All centralized algorithms are implemented and run in MATLAB R2010a. The distributed SPI-GP code uses the Parallel Toolbox in MATLAB R2010a.

We report three different sets of experiments here to summarize the performance of our algorithm in comparison to existing algorithms. We demonstrate the scalability of our algorithm for both the distributed and centralized versions. For accuracy, we compare the performance of our algorithm with an existing state-of-the-art sparse inverse covariance computation technique. Finally, for the climate data set we also look at interpretability of results in terms of the sparsity structure obtained from the penalized maximum likelihood computation.

A. *Study 1: Scalability study on synthetic data*

In this study we report the scalability of the distributed SPI-GP algorithm with respect to two different scenarios. In the first scenario, we keep the number of distributed computing nodes constant and increase the size of the data set. This increases the portion size of the covariance matrix per node and we study how our algorithm performs in terms of both running time and convergence. In our first experiment we fix the number of cores on which we run our experiment and vary the size of the training data. Figure 2(a) reports the running time for SPI-GP for different sizes of the data set on 10 cores. We vary the size of the data set from 1000 to 160000 samples, each set having a dimension of 5000. The kernel matrix in the last case is 16000×16000, which when partitioned columnwise for 10 jobs makes the data set size for each job 1600×16000. Because of the eigen decomposition required in every iteration of the algorithm, the algorithm is $O(n^3)$ per iteration, where n is the number of data points. However, due to the distributed computation, we see that the growth in the running time is of the order of $O(nr^2)$, where r is the rank of the kernel matrix partition available to each distributed job. It is evident from this



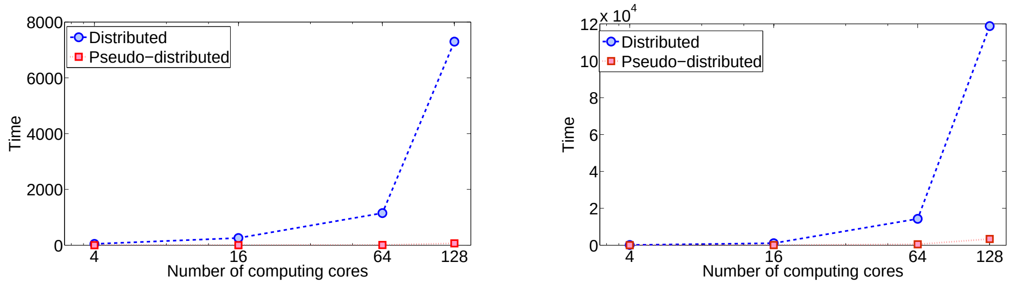
(a) Running time (in secs) for different dataset sizes on 10 cores.

(b) Iterations needed for distributed SPI-GP to converge

Fig. 2. Scalability study of SPI-GP on synthetic data. Time and iterations required for convergence are reported for distributed jobs on 10 cores with increasing data set size.

that since the computation is distributable, we can do the sparse inverse estimation for Gaussian Process regression for very large data sets, provided we have access to cluster computation environments. Figure 2(b) reports the number of iterations that are required for each of the problem sizes to converge with error tolerance of the order of 10^{-3}. We see that the number of iterations vary from 16 in the worst case to 9 in the best case. This number is significantly low, indicating that for reasonable tolerance ranges, the ADMM-based algorithm can converge quite fast.

In our second scalability experiment, we report the running time of the SPI-GP algorithm as we increase the number of processors keeping the points per processor constant. We experiment with two different sizes of the data set. For our first experiment the number of data points per processor is 16384 ($O(10^5)$) while for the second experiment the number of points per node is 262144 ($O(10^6)$). The results are shown in Figure 3(a) and Figure 3(b) respectively. In each case we vary the number of computing nodes from 4 to 128. In both cases we see that there is an almost quadratic increase in the running time for the distributed SPI-GP implementation, shown in blue lines in Figure 3. The red lines in Figure 3 indicate the running times for a pseudo-distributed implementation of the SPI-GP algorithm, where the parallel jobs within an iteration are executed sequentially in a single processor. We see that the running time in this case increases negligibly. This is understandable since the size of the problem remains the same in each experiment, and the slight increase in running time is due to the increased number of sequential operations for each iteration, as we increase the number of partitions in the data exponentially. This experiment is done to illustrate that the parallel eigen decomposition method



(a) (b)

Fig. 3. Running time (in secs) for increasing number of computing cores, with data points per code being constant. Fig. 3(a) reports running time for O(10^5) points per core and 3(b) shows running time for O(10^6) points per core.

in Matlab being a fully synchronized operation, the network synchronization in a distributed cluster computing environment takes up the bulk of the computation time. The optimization routine takes only a very small fraction of the time for execution. However, the advantage of distributed implementation becomes evident when we need to deal with data sets that cannot be loaded on to the memory of a single workstation.

B. *Study 2: Accuracy study on benchmark and real data*

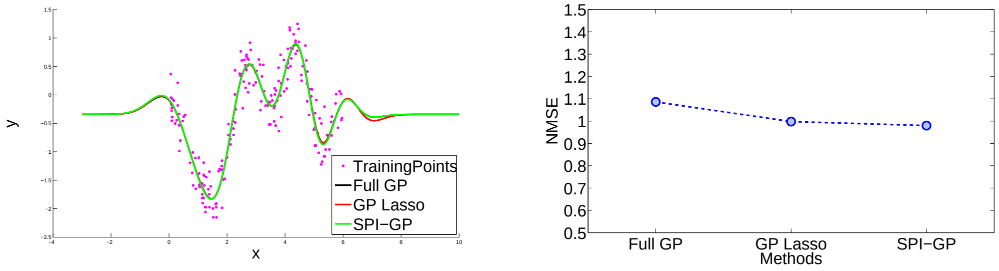
In this study, we report the accuracy of the SPI-GP method on two benchmark data sets used in the literature, and two real-life application data sets in earth science. For the first experiment, we compare the accuracy of SPI-GP to a state-of the-art sparse Gaussian Process regression method [29], using a couple of data sets that the authors have used in their paper. For this experiment we also compare our results to the regular Gaussian Process regression without penalty. The metric used for our accuracy study is the normalized mean squared error, defined as:

$$NMSE = \frac{\sum_{i=1}^{n} (\widehat{y}_i^* - y_i)^2}{n \times var(\widehat{\mathbf{y}}^*)}.$$

where y_i is the observed value of the target y_i having variance $var(\widehat{\mathbf{y}}^*)$, $\hat{y}_i$ is the prediction of y_i and n is the size of the test set.

Figure 4 shows the quality of prediction of SPI-GP compared to Full GP and GPLasso on the two benchmark data sets *Snelson* [29] and pumadyne-8nm[1]. Figure 4(a) shows the plot of the

[1] `http://www.cs.toronto.edu/~delve/`



(a) Prediction for *Snelson* dataset

(b) Prediction for pumadyne-8nm dataset

Fig. 4. Quality of prediction on benchmark data sets *Snelson* (left) and pumadyne-8nm (right). For left figure, black line represents predictions using Full GP (no approximations), the red and green lines are for GP-Lasso and SPI-GP respectively. Right figure shows the normalized mean squared error for each of these three methods on the pumadyne-8nm data set

predicted values of *Snelson* overlayed on the training set (plotted in magenta dots). The black line represents predictions using Full GP, the red and green lines are for GP Lasso and SPI-GP respectively. It can be observed that prediction quality of SPI-GP is comparable to both Full GP and GPLasso in this case. The black line (plotted first) which represents the method with no approximation is completely obscured by the green line for SPI-GP, whereas the red line for GP-Lasso is partially obscured by the fit of SPI-GP. This indicates that SPI-GP performs as good as Full GP, whereas GP-Lasso does well in most places except at points where the red line is visible, where it deviates slightly from the true fit. Figure 4(b) shows the normalized mean squared error for each of these three methods on the pumadyne-8nm data set. It is a difficult data set to model and all three methods have high errors. NMSE for full GP is 1.086, while NMSE for a 300-rank reduced GPLasso is 0.996 and SPI-GP with $\lambda = 0.01$ is 0.983. This indicates that SPI-GP performs exactly the same way as state-of-the-art sparse Gaussian Process regression in terms of accuracy. The additional benefit is an interpretable model that explains relations among the regressors and the target with respect to the samples. To verify the performance of the ADMM-based optimization solution, we have also performed experiments where the ADMM-based inversion in SPI-GP has been replaced by the graphical lasso [1] method. Since both methods converge to the exact same optimum for both the Snelson and the pumadyne data sets, their plots obscure each other in the figures and are therefore not included in the graphs shown in Figure 4.

We also report NMSE of SPI-GP on the two prediction problems on the two real world data sets. For the first data set we predict precipitation on a region of the Indian peninsula based on

Training years	Test months							
	March		June		September		December	
	Full-GP	SPI-GP	Full-GP	SPI-GP	Full-GP	SPI-GP	Full-GP	SPI-GP
1982	0.237	0.283	0.454	0.439	0.426	0.426	0.371	0.361
1983	0.258	0.292	0.492	0.492	0.658	0.658	0.374	0.374
1984	0.261	0.273	0.451	0.451	0.818	0.819	0.374	0.368
1985	0.196	0.208	0.475	0.450	0.396	0.396	0.385	0.385

TABLE I

NMSE of GPR for 1986 for quarterly GP models built from 1982-1985.

historical data from the entire grid. If a set of points are very similar to the points representing rainfall in the Indian subcontinent, then it is intuitive that those points should be very good predictor of precipitation in India. In this study we verify this intuition by choosing the top k locations of the world that are most similar to the precipitation in the Indian subcontinent, based on the sparse inverse covariance estimation, and then build a prediction model based on only those points. The value of k is chosen to be $n/2$ where n is kernel dimension. In most cases we see that the smaller model is more accurate than the entire data set.

For our study we choose a time scale of three months for the precipitation prediction problem. The data set we are considering has 20 years of precipitation data. For any year in this data set, we build models on the quarterly precipitation information and test the model on a one-month time delay. In our first experiment we test our quarterly models from 1982 to 1985 on 1986 data and the NMSE values are reported in Table I. We repeat the experiment for a training period of 1982 to 1995 to test on 1996 and the results are shown in Table II. For baseline, we use results obtained by using the entire data set (Full-GP) rather than the chosen subset. It should be noted here that Full-GP refers to the standard Gaussian Process regression method. Since the data set contains approximately 250K points, it is not possible to build a Gaussian Process model on this entire data set and we instead sample 8000 points randomly from this data set to run the optimization for choosing the model parameters and then build the kernel on only those sample points. The results reported for Full GP in the table are the best over 10 runs of this experiment. However, the variance for the runs is high, indicating that such a uniform sampling based approach may not always produce the desired model.

Training years	Test months							
	March		June		September		December	
	Full-GP	SPI-GP	Full-GP	SPI-GP	Full-GP	SPI-GP	Full-GP	SPI-GP
1982	0.311	0.293	0.554	0.563	0.706	0.706	1.23	1.22
1986	0.325	0.295	0.587	0.595	0.81	0.809	1.301	1.3
1991	0.281	0.278	0.564	0.586	0.782	0.781	1.15	1.15

TABLE II

NMSE OF GPR FOR 1996 FOR QUARTERLY GP MODELS BUILT FROM 1982-1991.

Tables I and II document the NMSE values for predicting precipitation in India for months March, June, September and December for the years 1986 and 1996 respectively. The experimental setup is as follows: We build models on the first two months of each quarter for all the training years combined and test it on the third month of the same quarter. In our first experiment, we study the prediction for the year 1986 based on the models built on observations from the years 1982 to 1985. In the second experiment, the quarterly models are built on all years from 1982 to 1991 combined. For the experiments reported here, we look at a 2° resolution of the observation grid, which makes the kernel size 16200×16200. This is done to keep the running time reasonable for the experiments, although in theory the method can handle larger kernel sizes. NMSE values in the table range from as low as .19 to as high as 1.3 for different prediction scenarios. For example year 1986 has reasonably good predictability and has lower variation in the NMSE values than year 1996.

Although March, June and September have reasonably low NMSE values for 1996, the month of December does not have that indicating that the model prediction is working as poorly as random for the different training years. For this study, the NMSE value for the GPR model of top k values from SPI-GP is better than the best Full-GP model prediction. This happens because the most similar points capture more information and less of noise as has been verified earlier in [3]. It can be noted the improvement in NMSE observed is not significant. This is partly due to the fact that the precipitation prediction problem that we are studying is a difficult problem in climate science since the data lacks reasonable quality. The linear correlations for different data subsets and different test sets can vary from -0.2 (very poor) to 0.88 (high correlation) accounting for the high variability in the NMSE values observed for the different test scenarios.

The exceptionally poor performance of the prediction model for the last two quarters of 1996 could also be attributed to changing climate during that period of the year as compared to historical observations from more than a decade ago.

The last data set that we have experimented with is the MODIS data. The regression problem for the data set is to predict frequency band 6, given bands 1, 4, and 5. Here we report the prediction results for three different days for the year 2001. We choose a location in the California central valley region as our test point. We have divided the days of the observations into three groups representing winter (from January to April), summer (from May to August) and fall (September to December). The results shown here are predictions for one day each from the three groups. Similar to the climate data set, here also we see that using SPI-GP to create a smaller and less noisy data set improves the prediction of the Gaussian Process regression model. The variation in the absolute values of NMSE for the three different days can be attributed to data quality. Since this is satellite data, seasonal variations in cloud cover, haze, etc. can affect the quality of the data being collected. Gaussian Process regression can be used in such cases to create a stable and less noisy data set for such observations.

Season	NMSE	
	Full-GP	SPI-GP
Winter	0.018699	0.017698
Summer	0.541981	0.423473
Fall	0.425959	0.395641

TABLE III
NMSE FOR BAND 6 PREDICTION IN MODIS DATA SET FOR 2001.

C. *Interpretability of results on real-world data sets*

In many real-life applications, we are often not only interested in the accuracy of the prediction model, but we also want to understand the model in order to better explain the underlying physical phenomenon driving the model predictability. In the case of climate studies, model interpretability can provide much coveted understanding of which geographical regions are most similar to other regions of interest, even when the relevant regions are located far apart on the

(a) Network representing a sub-matrix of the inverse covariance matrix

(b) Network representing a sub-matrix of the sparse inverse covariance matrix estimated using SPI-GP

Fig. 5. Interpretability of the model is much higher in the case of a parsimonious model.

earth's grid. In this section, we discuss how the SPI-GP models can be interpreted to uncover geographical relationships among different regions.

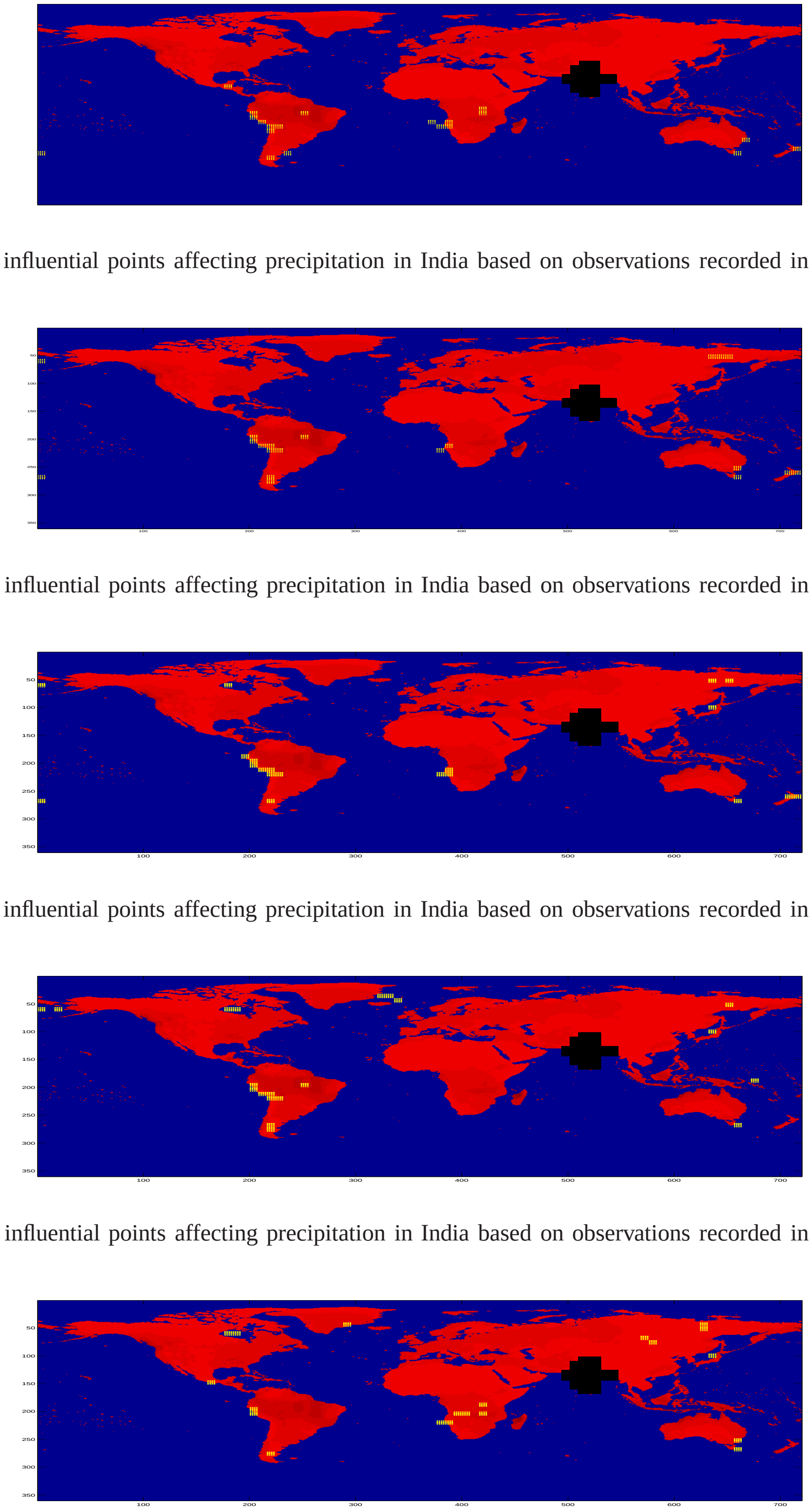
To illustrate how the SPI-GP method can be used for studying climate networks, we take a small number of locations on the grid and compute the inverse covariance matrix for the climate variable precipitation. We represent each point on the earth's grid as a node in a network and the edges represent non-zero entries in the inverse covariance matrix. We want to identify the most similar nodes, given a reference node, highlighted in red in Figure 5. As can be seen in Figure 5(a), the true inverse covariance is difficult to understand or interpret, given the huge amount of network connections for any particular node in the graph. It is important to note that a large number of these connections are very small non-zero values, indicating almost no connection between the corresponding pair of nodes. Figure 5(b) is a sparse variant of the same graph and shows only the important connections to the relevant node. Thus SPI-GP increases the interpretability of a Gaussian Process model.

For the NCEP/NCAAR data set, we study precipitation in the Indian peninsula. We want to identify points that have similar pattern as our test set and study how, if at all, these points change over a course of 20 years. Since all climatic connections change very slowly with time, we construct the relevant network connections for Indian precipitation every 5 years. Fig. 6 shows the results. Each plot in Fig. 6 is for the average of one year's data. The variable shown in the figures is precipitation. The black markers are the locations in India. The yellow markers indicate the top 10 areas which influence India. These are the points which have the highest values in the estimated inverse kernel matrix corresponding to test points for India. As Figure 6 shows, there are certain regions which remain similar to our test set for the entire period of 20 years, while others appear and disappear over time. Some locations which show consistent influence pattern include the west coast of South America, west coast of Africa, and east coast of Australia. This shows that there is a clear climatic connection between the precipitation patterns of these regions. Some less consistent locations include areas in China which are evolving connections which might become consistent over time.

For the MODIS data set, we look for vegetation patterns that are most similar to a chosen area of the California Central valley. In this study we look for seasonal variation in similarity of vegetation. We coarsen the MODIS data of 1 km resolution to 5 km and study this variation for the year 2001. Fig. 7 shows the results. The area of the central valley that we are interested in, is highlighted using a black square marker on the map of California. The color map shows the observed values of vegetation index for a particular composite in 2001. The red circled markers indicate 5 locations in California whose vegetation index is similar to our region of interest. We notice that there is some overlap between these similar points in winter and summer and overlap between a different set of locations for the summer and fall. This indicates changes in the vegetation pattern in the state of California across different seasons and such observation may be very significant in decision support systems in with agriculture and planning.

VI. Conclusion

In this paper we discuss a method for sparse inverse Gaussian Process regression that allows us to compute a parsimonious model while preserving the interpretability of the sparsity structure in the data. We discuss how the inverse kernel matrix used in Gaussian Process prediction gives valuable information about the regression model and then adapt the inverse covariance estimation from Gaussian graphical models to estimate the Gaussian kernel. We solve the optimization problem using the alternating direction method of multipliers that is amenable to parallel computation. This sparsity exploiting GPR technique achieves two goals: (i)it provides valuable insight into the regression model and (ii)it allows for parallelization so that the entire kernel matrix need not be loaded into one memory, thereby removing size related constraints plaguing large scale analysis. We perform extensive experiments on both synthetic and real-world data sets and report various computational aspects of the algorithm, namely scalability and accuracy. We also illustrate how this method produces an interpretable model thats aids



(a) Most influential points affecting precipitation in India based on observations recorded in the year 1982

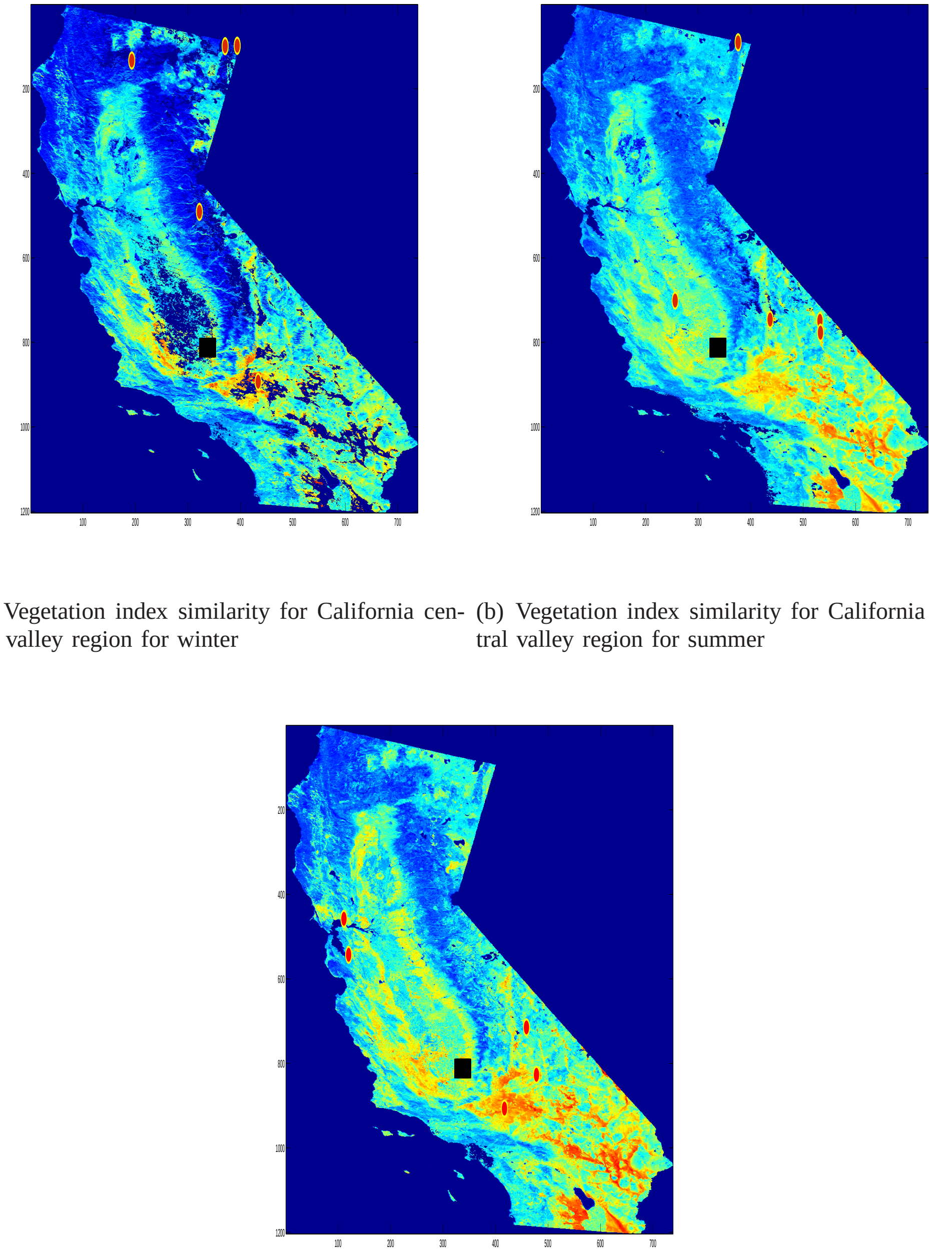
(b) Most influential points affecting precipitation in India based on observations recorded in the year 1986

(c) Most influential points affecting precipitation in India based on observations recorded in the year 1991

(d) Most influential points affecting precipitation in India based on observations recorded in the year 1996

(e) Most influential points affecting precipitation in India based on observations recorded in the year 2001

Fig. 6. Evolution of the precipitation network over 20 years for the Indian peninsula



(a) Vegetation index similarity for California central valley region for winter

(b) Vegetation index similarity for California central valley region for summer

(c) Vegetation index similarity for California central valley region for fall

Fig. 7. Seasonal variation of the vegetation index similarity for California central valley region

understanding the underlying physical phenomenon responsible for model predictability. For future work, we want to explore these models in details with help from domain scientists to discover new relationships and explain current observations in climate. In terms of the algorithm, we want to develop an approximate version of the optimization problem that is decomposable and, therefore, amenable to distributed computing in a more loosely coupled computing scenario.

ACKNOWLEDGEMENTS

This work was supported by the Systemwide Safety and Assurance Technologies project of Aviation Safety Program under the NASA Aeronautics Mission Directorate.

REFERENCES

[1] O. Banerjee, L. Ghaoui, A. d'Aspremont, and G. Natsoulis. Convex optimization techniques for fitting sparse Gaussian graphical models. In *Proceedings ICML-06*, pages 89–96, 2006.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 2011.

[3] K. Das and A. Srivastava. Block-GP: Scalable Gaussian Process Regression for Multimodal Data. In *The 10th IEEE International Conference on Data Mining, ICDM 2010*, pages 791–796, 2010.

[4] K. Das and A. Srivastava. Sparse inverse gaussian process regression with application to climate network discovery. In *Proceedings of 2011 Conference on Intelligent Data Understanding*, pages 233–247, 2011.

[5] A. P. Dempster. Covariance Selection. *Biometrics*, 28:157–175, 1972.

[6] J. Donges, Y. Zou, N. Marwan, and J. Kurths. Complex networks in climate dynamics. *EPJ Special Topics*, 174:157–179, 2009.

[7] J. Eckstein and D. Bertsekas. An alternating direction method for linear programming. Technical Report LIDS-P ; 1967, MIT, 1990.

[8] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

[9] L. Foster, A. Waagen, N. Aijaz, M. Hurley, A. Luis, J. Rinsky, C. Satyavolu, M. Way, P. Gazis, and A. Srivastava. Stable and Efficient Gaussian Process Calculations. *JMLR*, 10:857–882, 2009.

[10] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics Journal*, 9(3):432–441, 2008.

[11] M. Fukushima. Application of the alternating direction method of multipliers to separable convex programming problems. *Computational Optimization and Applications*, 1:93–111, 1992.

[12] M. H. Glantz, R. W. Katz, and N. Nicholls. *Teleconnections linking worldwide climate anomalies : scientific basis and societal impact*. Cambridge University Press, 1991.

[13] C. Hsieh, M. Sustik, I. Dhillon, and P. Ravikumar. Sparse inverse covariance matrix estimation using quadratic approximation. In *Advances in Neural Information Processing Systems 24*, 2011.

[14] E. Kalnay, M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, L. Candin, S. Saha, G. White, J. Woollen, Y. Zhu, M. Chelliah, W. Ebisuzaki, W. Higgins, J. Janowiak, K. C. Mot, C. Ropelewski, A. Leetmaa, R. Reynolds, R. Jenne, and D. Joseph. The NCEP/NCAR 40-Year Reanalysis Project. *B. Am Metrolo. Soc.*, 77(3):437–471, 1996.

[15] S. Kasiviswanathan, P. Melville, A. Banerjee, and V. Sindhwani. Emerging topic detection using dictionary learning. In *Proceedings of CIKM 2011*, pages 745–754, 2011.

[16] N. Meinshausen, P. Bhlmann, and E. Zrich. High Dimensional Graphs and Variable Selection with the Lasso. *Annals of Statistics*, 34:1436–1462, 2006.

[17] J. Quiñonero-Candela and C. E. Rasmussen. A Unifying View of Sparse Approximate Gaussian Process Regression. *JMLR*, 6:1939–1959, 2005.

[18] C. E. Rasmussen. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[19] A. J. Smola and P. Bartlett. Sparse Greedy Gaussian Process Regression. In *Proc. of NIPS 13*, pages 619–625, 2000.

[20] E. Snelson and Z. Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Proceedings of NIPS 18*, 2005.

[21] K. Steinhaeuser, N. Chawla, and A. Ganguly. An exploration of climate data using complex networks. *SIGKDD Explorations Newsletter*, 12:25–32, November 2010.

[22] K. Steinhaeuser, N. Chawla, and A. Ganguly. Complex networks as a unified framework for descriptive analysis and predictive modeling in climate science. *SADM J.*, 4(5):497–511, 2011.

[23] R. Tibshirani. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[24] V. Tresp. The generalized bayesian committee machine. In *Proc. of KDD*, pages 130–139, 2000.

[25] A. Tsonis, K. Swanson, and P. Roebber. What do networks have to do with climate? *BAMS*, 87:585–595, 2006.

[26] A. A. Tsonis, K. L. Swanson, and P. J. Roebber. What do networks have to do with climate? *Bulletin of the American Meteorological Society*, 87(5):585–595, 2006.

[27] G. Wahba. *Spline Models for Observational Data*. SIAM, 1990.

[28] P. Weiss, V. Lobojois, and D. Kouame. Alternating direction method of multipliers applied to 3d light sheet fluorescence microscopy image deblurring using gpu hardware. In *Proceedings of IEEE EMBC 2011*, pages 4872–4875, 2011.

[29] F. Yan and Y. Qi. Sparse Gaussian Process Regression via ℓ_1 Penalization. In *Proceedings of ICML-10*, pages 1183–1190, 2010.

[30] G. Ye, Y. Chen, and X. Xie. Efficient variable selection in support vector machines via the alternating direction method of multipliers. *Journal of Machine Learning Research - Proceedings Track*, 15:832–840, 2011.