A Hierarchical Approach to Scalable Gaussian Process Regression for Spatial Data

Jacob Dearmon
Meinders School of Business
Oklahoma City University
jdearmon@okcu.edu

Tony E. Smith
Department of Electrical and Systems Engineering
University of Pennsylvania
tesmith@seas.upenn.edu

1. Introduction

Large scale datasets such as County Assessor's geodatabases offer novel opportunities to investigate spatial phenomena at much finer levels of resolution than in the past. Spatial spillovers, urban infill, renovation price effects and proximity to investment\amenity zones can now be examined at the parcel level, opening up new avenues for the bulk identification of specific development opportunities. The central purpose of this paper is to develop an approach to analyzing such data in an efficient manner. Our approach starts with Gaussian Process Regression (GPR), which is well known prediction tool for analyzing spatial datasets. Moreover, the smooth nature of its prediction surfaces is particularly well suited for identifying the local marginal effects (LME) of key explanatory variables [as developed in Dearmon & Smith (2016, 2017)]. It is these effects that will allow an examination of more fine-grained spatial phenomena, such as the local development opportunities mentioned above.

However, the application of such GPR methods to large data sets has thus far been limited by the need to invert large dense covariance matrices. Thus, it is not surprising that this practical limitation has led to a variety of methods for approximating GPR models by more efficiently computable versions [as reviewed for example in Chen et al. (2017)]. In the present paper, we focus on one of the most promising of these approaches, namely the development of a *hierarchical covariance approximation* to GPR by Jie Chen ([C1] = Chen et al, 2017; [C2] = Chen & Stein, 2017), which we here denote by GPR-HCA This hierarchical extension of Nyström's low-rank approximation yields dramatic improvement in both speed and accuracy of predictions. In fact, this approximation allows matrix inversions that achieve the optimal efficiency level of $O(n)$, i.e., are *linear* in the matrix dimension, *n*. Of equal importance, these approximations are guaranteed to yield positive definite matrices that generate well-defined Gaussian Processes. So, from a methodological perspective, our central objective is to extend such approximations to the analysis of local marginal effects in large-data contexts.

To do so, we begin in Section 2 with a review of the standard Gaussian Process Regression model, and in particular, its associated local marginal effects. In Section 3, we then develop the GPR-HCA method in detail. One contribution of this paper is to give an explicit probabilistic interpretation of this method, which we illustrate for two- and three-level hierarchies. In addition, we highlight some of the key auxiliary tools proposed by Chen ([C1],[C2]) which are particularly useful for our LME extensions. In Section 4, we test both the accuracy and scalability of this hierarchical approach by constructing a simple two-variable simulation model that allows for visual as well as numerical comparisons with other methods. Here we begin by comparing GPR-HCA with the standard Gaussian Process Regression model (GPR-FULL) over sample sizes small enough to allow the full version to be run. In addition, we compare GPR-HCA with two other large-scale prediction models for sample sizes up to half a million. Of particular relevance is the Nearest-Neighbor approximation of Gaussian Processes (NNGP) first introduced by Datta et al. (2016), which also yields covariance approximations that are linear in matrix dimension and generate well-defined Gaussian Processes. In addition, we also compare GPR-HCA performance with one of the standard machine learning algorithms, namely the Generalized Boosted Models (GBM) algorithm of Ridgeway (2007). In all cases we find comparable predictive performance, and much improved time costs over GPR-FULL in particular. However, while such comparative tests are important, they are not of primary interest for our present purposes. More important is the technical extension of GPR-HCA to the evaluation of LME's for large data sets. Within the same simulation framework, such estimated LME's are shown to accurately replicate the derivatives of well-behaved functions corrupted by noise. In Section 5, we turn to an empirical application and apply these HCA-tools to tackle the difficult and often ill-behaved relationship between house prices and attributes using data obtained from nearly a decade's worth of County Assessor's databases in Oklahoma County. In particular, we focus on two distinct regions of Oklahoma County; one just north of downtown where spatial spillovers appear to be present and the other a small, wealthy municipality, located further north, where spatial infill opportunities appear to exist. We investigate and analyze such phenomena using GPR-HCA, and provide confirmatory evidence of our findings using building permit data. Finally, we conclude in Section 6 with a brief discussion of several possible extensions of this work that are of both practical and technical importance.

2. Gaussian Process Regression

Given a spatial process with *response variable*, Y_l , on a domain, $S = \{x_l = (x_{l1},..,x_{ld})\} \subseteq \mathbb{R}^d$ of possible *explanatory variables* [including the spatial coordinates of location, l], we start by assuming that stochastic variations in observed values of Y_l about their common mean, μ , are governed by a underlying (latent) zero-mean Gaussian process, $f : S \to \mathbb{R}$, with observed values (measurements) corrupted by independent additive Gaussian noise,

$$(1) \qquad Y_l = \mu + f(x_l) + \varepsilon_l \quad , \quad \varepsilon_l \sim N(0,\sigma^2)$$

In essence this implies that latent responses, $f = (f_l : l = 1,..,n)$, at any finite set of locations with associated explanatory variables, $X = (x_l : l = 1,..,n)$ are *multi-normally* distributed as

$$(2) \qquad f \sim N[0_n, K(X,X)]$$

with covariance matrix, $K(X,X) = [k(x_i,x_j) : i, j = 1,..,n]$, generated by a *kernel function*, $k(x_l,x_h)$ $[\equiv \text{cov}(f_l,f_h)]$, depending only on the attribute profiles of response variates. By (1) this implies that the resulting observed responses, $Y = (Y_l : l = 1,..,n)$, are distributed as

$$(3) \qquad Y \sim N[\mu 1_n, K(X,X) + \sigma^2 I_n]$$

where 1_n and I_n denote respectively the unit vector and identity matrix of size *n*. To model spatial covariance, we here employ the standard (anisotropic) squared exponential (SE) kernel function:

$$(4) \qquad k(x_l, x_h) = v \exp\left[-\sum_{i=1}^{d} \tfrac{1}{2\tau_i^2}(x_{li} - x_{hi})^2\right]$$

where v denotes the common *variance* of all responses, i.e., $\mathrm{var}(f_l) = k(x_l, x_l) = v$, and where each *length-scale* parameter, $\tau_j > 0$, governs the degree to which variable, x_j, influences covariance.

With these assumptions, the fundamental *Gaussian Process Regression* (GPR) problem is to obtain the predictive (conditional) distribution of latent responses, $f_* = f(X_*)$, at n_t *test locations* with attributes, $X_* = (x_{*l} : l = 1,..,n_t)$, given observed responses, $Y = (Y_1,..,Y_n)$, at *n training locations* with attributes, $X = (x_l : l = 1,..,n)$. If we start with the joint distribution,

$$(5) \qquad \begin{pmatrix} f_* \\ Y \end{pmatrix} \sim N\left[\begin{pmatrix} 0_{n_t} \\ \mu 1_n \end{pmatrix}, \begin{pmatrix} K(X_*, X_*) & K(X_*, X) \\ K(X, X_*) & K(X, X) + \sigma^2 I_n \end{pmatrix}\right]$$

then the desired (conditional) *predictive distribution* is well known to be multi-normal

$$(6) \qquad f_* \mid Y \sim N\left[E(f_* \mid Y), \mathrm{cov}(f_* \mid Y)\right]$$

with conditional mean and covariance,

$$(7) \qquad E(f_* \mid Y) = K(X_*, X)[K(X, X) + \sigma^2 I_n]^{-1}(Y - \mu)$$

$$(8) \qquad \mathrm{cov}(f_* \mid Y) = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma^2 I_n]^{-1} K(X, X_*)$$

As developed in a previous paper Dearmon & Smith (2017), we also consider *local marginal effects* (LME),

$$(9) \qquad \frac{\partial E(f_* \mid Y)}{\partial x_{*l,j}} = \frac{\partial K(x_{*l}, X)}{\partial x_{*l,j}}[K(X, X) + \sigma^2 I_n]^{-1}(Y - \mu), \quad l = 1,..,n_t$$

capturing the expected impact of small changes in individual attributes, $j = 1,..,d$, such as the impact of an additional square foot on the expected sales price of a given house with a specific set of attributes. For purposes of model calibration and prediction, a key scaling issue that arises is the size of the inverse to be calculated in (7), (8) and (9).

3. Hierarchical Covariance Approximation

Assuming that n is large, the objective of Chen's procedure is to construct a hierarchical approximation to the n-square covariance matrix, K. The approach starts by partitioning domain S into a collection of *basic* subdomains, S_i, $i = 1,..,b$, where each subset of sample points, $X_i = S_i \cap X = [x_{i1},..,x_{in_i}]$, is sufficiently small to ensure that the associated covariance matrix, $K_{ii} = K(X_i, X_i)$, can easily be inverted. (Note that for notational simplicity, we have now dropped references to individual spatial locations, l). The second step is to approximate the covariances,

$$(10) \qquad K_{ij} = K(X_i, X_j) = [k(x_i, x_j) : x_i \in X_i, x_j \in X_j] , \quad i, j = 1,..,q \;\; (i \neq j)$$

between distinct subdomains in terms of their mutual covariances with smaller sets of "landmark" points.[1] These concepts are best illustrated by simple examples.

3.1 Two-Level Hierarchical Example

The simplest example involves a partitioning of S into two subdomains, S_1 and S_2, as illustrated in Figure 1 below, where for graphical convenience we show only the *spatial* coordinates ($d = 2$).

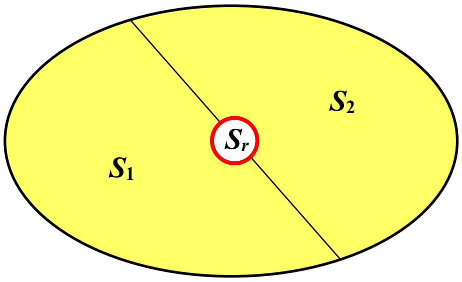

Figure 1. Two-Level Partition

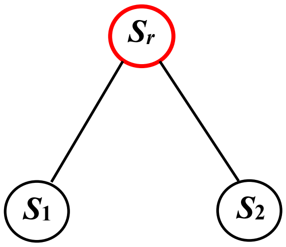

Figure 2. Tree Representation

To approximate the covariances between points in these two subdomains, one selects a small representative subset of points, $X_r = [x_{r1},..,x_{rn_r}] \subset X_1 \cup X_2$, designated as *landmark points* for X_1 and X_2. In this case, X_r, is associated with the full domain, $S = S_1 \cup S_2 \equiv S_r$. Moreover, given the hierarchical relations among these three domains (with respect to set containment, $\subseteq$), Figure 1 can also be represented as a two-level *tree structure* with *root*, S_r, and *leaves*, (S_1, S_2), as shown in Figure 2. This underlying tree structure is of fundamental importance in the recursive calculation of the covariance approximations discussed below.

In terms of these landmark points, Chen's hierarchical approximation to K_{12} in (10) is given by Nyström's (n_r-rank) approximation,

[1] These are also referred to as "inducing" points (as for example in Rasmussen & Quinonero-Candela, 2005).

(11) $$K_{12}^{H} = K_{1r}K_{rr}^{-1}K_{r2} = K(X_1, X_r)K(X_r, X_r)^{-1}K(X_r, X_2) = (K_{21}^{h})^{T}$$

where *H* denotes “hierarchical”. Note from the positive definiteness of the full covariance matrix, K, that K_{rr}^{-1} is well defined and is also positive definite. In these terms, the full hierarchical approximation of K is given by

(12) $$K^{H} = \begin{pmatrix} K_{11}^{H} & K_{12}^{H} \\ K_{21}^{H} & K_{22}^{H} \end{pmatrix} = \begin{pmatrix} K_{11} & K_{1r}K_{rr}^{-1}K_{r2} \\ K_{2r}K_{rr}^{-1}K_{r1} & K_{22} \end{pmatrix}$$

This is essentially the example in expression (4) of [C2] with only two subdomains. Note also from the positive definiteness of the block diagonal structure, that even though the off-diagonal approximations are not of full rank, it is not surprising that the overall approximation is of full rank. What is far less obvious is that this approximation is actually positive definite, i.e., is itself a full-rank covariance matrix. While the proof of positive definiteness in this two-level case is a simple consequence of Schur Complementarity ([C1], Theorem 3), the higher-level cases developed below are considerably more subtle.

Probabilistic Interpretation.

With this is mind, it is instructive to develop a direct probabilistic approach to these hierarchical approximations, i.e., a full-dimensional Gaussian probability model with precisely this covariance, K^{H}. To do so, we start with the latent process, $f \sim N(0, K)$, in (2) and let $f_i = f(X_i)$, $i = 1, 2, r$. To approximate the covariance, K_{12}, between f_1 and f_2 in terms of their relations with f_r, we then consider their conditional means and covariances

(13) $$E(f_i \mid f_r) = K_{ir}K_{rr}^{-1}f_r \;\;,\; i = 1, 2$$

(14) $$\mathrm{cov}(f_i \mid f_r) = K_{ii} - K_{ir}K_{rr}^{-1}K_{ri} \;\;,\; i = 1, 2$$

which are essentially obtained from (7) and (8) by setting $\sigma^2 = 0$. A key feature of the multi-normal distribution is that while the conditional mean in (13) depends on the value of f_r, the conditional covariance in (14) does not. This plays a crucial role in the following construction. As a first step, if we now designate the following zero-mean version of $f_i \mid f_r$ as a *centered conditional*,

(15) $$Z_{i|r} \sim N(0_{n_r}, K_{ii} - K_{ir}K_{rr}^{-1}K_{ri}) \;,\; i = 1, 2$$

then since f_r does not appear in the distribution of $Z_{i|r}$, we may choose $Z_{1|r}$ and $Z_{2|r}$ to be independent not only of one another but also f_r. For notational consistency, we also let

$Z_r \sim N(0_{n_r}, K_{rr})$ denote a version of f_r that is independent of both $Z_{1|r}$ and $Z_{2|r}$, so that by construction the random vector, $Z = (Z_{1|r}, Z_{2|r}, Z_r)$, is multi-normal[2] with:

$$(16) \qquad Z = \begin{pmatrix} Z_{1|r} \\ Z_{2|r} \\ Z_r \end{pmatrix} \sim N\left[\begin{pmatrix} 0_{n_1} \\ 0_{n_2} \\ 0_{n_r} \end{pmatrix}, \begin{pmatrix} K_{11} - K_{1r}K_{rr}^{-1}K_{r1} & & \\ & K_{22} - K_{2r}K_{rr}^{-1}K_{r2} & \\ & & K_{rr} \end{pmatrix}\right]$$

The desired probability model can then be formed as linear combinations of these independent basis vectors. If we now define the coefficient matrices,

$$(17) \qquad A_{ij} = K_{ij}\,K_{jj}^{-1}\ ,\ \ i,j = 1,2,r$$

then the appropriate *hierarchical model*, $H = (H_1, H_2)$, for the present case is given by

$$(18) \qquad H_1 = Z_{1|r} + A_{1r}\,Z_r$$

$$(19) \qquad H_2 = Z_{2|r} + A_{2r}\,Z_r$$

where each vector of latent variables, $H_i = (h_{ij} : j = 1,..,n_i)$, represents a hierarchical version of the original latent responses, $(f_{ij} : j = 1,..,n)$, in the full model (1). Intuitively, it is the second terms in these expressions (both containing Z_r) that govern the covariances between random vectors H_1 and H_2. As we shall see below, the first terms then serve to maintain the desired marginal distributions of H_1 and H_2. Note also that since (18) and (19) can be written in matrix form as

$$(20) \qquad H = \begin{pmatrix} H_1 \\ H_2 \end{pmatrix} = \begin{bmatrix} I_{n_1} & 0 & A_{1r} \\ 0 & I_{n_2} & A_{2r} \end{bmatrix} \begin{pmatrix} Z_{1|r} \\ Z_{2|r} \\ Z_r \end{pmatrix}$$

it follows that H is a linear transformation of Z, and thus is also multi-normally distributed.[3] So if it can be shown that cov(H) = K^H, then since $E(Z) = 0$ by construction, we will obtain a well-defined probability model

$$(21) \qquad H \sim N(0_n, K^H)$$

[2] Note that whenever $X_i \cap X_r \neq \varnothing$, the conditional covariance matrix, $X_{ii} - X_{ir}\,X_{rr}^{-1}X_{ri}$, in (14) must be *singular*. But as will be seen in footnote 3 below, this has no substantive consequences for the model constructed.

[3] As seen in (22) below, the random vectors, H_1 and H_2, have full rank covariance matrices, and thus are properly multi-normally distributed even when $Z_{1|r}$ and $Z_{2|r}$ are singular multi-normal [see for example Anderson (1958, Theorem 2.4.5)].

with the desired covariance matrix, K^H. It is this *hierarchical model*, *H*, that will replace f in expression (2) of the original model. So, all that remains to be shown is that this hierarchical model has the desired covariance structure. These same observations will continue to hold in more complex examples, and shall not be repeated.

In the present case, we begin by observing that expressions (14) through (18), together with the independence of the *Z* components, imply that

$$(22)\qquad \mathrm{cov}(H_1) = \mathrm{cov}(Z_{1|r}) + \mathrm{cov}(A_{1r}\,Z_r) = (K_{11} - K_{1r}K_{rr}^{-1}K_{r1}) + A_{1r}\,\mathrm{cov}(Z_r)A_{1r}^T$$

$$= (K_{11} - K_{1r}K_{rr}^{-1}K_{r1}) + (K_{1r}K_{rr}^{-1})(K_{rr})K_{rr}^{-1}K_{r1} = K_{11} = K_{11}^h$$

and similarly, that $\mathrm{cov}(H_{22}) = K_{22}^h$. Moreover, the independence and zero-mean properties of the *Z* components also imply that

$$(23)\qquad \mathrm{cov}(H_1, H_2) = E[H_1H_2^T] = E[(Z_{1|r} + A_{1r}\,Z_r)(Z_{2|r} + A_{2r}\,Z_r)^T]$$

$$= E[(A_{1r}\,Z_r)(A_{2r}\,Z_r)^T] = A_{1r}\,E(Z_rZ_r^T)\,A_{2r}^T = A_{1r}\,\mathrm{cov}(Z_r)\,A_{2r}^T$$

$$= (K_{1r}K_{rr}^{-1})(K_{rr})(K_{rr}^{-1}K_{r2}) = K_{1r}K_{rr}^{-1}K_{r2} = K_{12}^h\ ,$$

which together with $\mathrm{cov}(H_2, H_1) = \mathrm{cov}(H_1, H_2)^T$ yields the desired result, $\mathrm{cov}(H) = K^H$.

3.2 Three-Level Hierarchical Example

If the full sample of locations, $X \subset S$, is extremely large, then each of the subsets, $X_i \subset S_i$, $i = 1,2$, may also be large. Suppose for example that S was partitioned into four smaller subdomains, (S_1, S_2, S_3, S_4), as shown in Figure 3 below. While one could in principle use the same set of landmark points, $X_r \subset S_r = S$, to approximate covariances among the points, $X_i = X \cap S_i$, $i = 1,..,4$, it is now possible to refine these approximations. In the present spatial setting, it is reasonable to suppose that points in adjacent domains, say S_i and S_j are more closely related (have higher covariances) than other point pairs. If so, then a better approximation to covariances between S_i and S_j is obtained by using only landmark points in $S_i \cup S_j$.

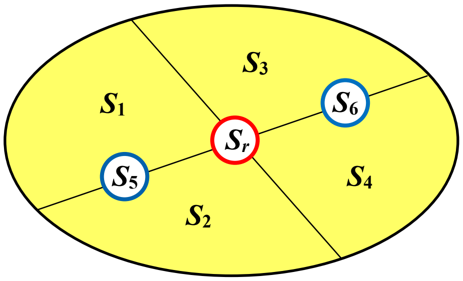

Figure 3. Three-Level Partition

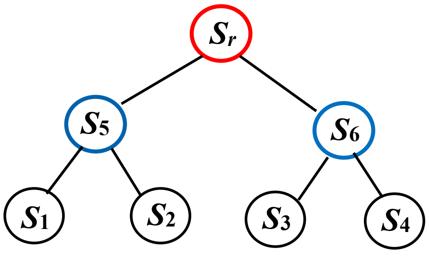

Figure 4. Tree Representation

To model such relations, we first recall from the hierarchical tree structure in Figure 2 above that subdomains S_1 and S_2 are also called *children* of the *parent* domain, S_r . In these terms, the construction in (18) and (19) can be viewed as a "parent-child" relationship. Following Chen [C1, sect. 2.2]), we refine covariance approximations by extending this type of relationship. If we let $S_5 = S_1 \cup S_2$ and $S_6 = S_3 \cup S_4$, then as seen in Figures 3, (S_1, S_2) and (S_3, S_4) are the respective children of S_5 and S_6 . If landmark points, $X_i = [x_{i1},..,x_{in_i}] \in S_i$, are chosen for $i = 5,6$, then these can in principle be used to approximate covariances between their respective children. Similarly, if we again designate the root domain by $S_r = S = S_5 \cup S_6$, then the subdomains (S_5, S_6) are themselves children of S_r . So, if we again choose landmark points for this parent domain, $X_r = [x_{r1},..,x_{rn_r}] \in S_r$, then these can also be used to approximate covariances between children in X_5 and X_6 . These nesting relationships can alternatively be represented by the tree structure in Figure 4, where the basic partition domains, (S_1, S_2, S_3, S_4) , at the lowest level again constitute the leaf nodes of the tree with root node, S_r , and intermediate nodes, S_5 and S_6 . Every link between nodes now represents a parent-child relation.

Extended Probabilistic Interpretation.

To extend the probabilistic interpretation of the two-level hierarchical covariance approximation above, we start at the upper level and define hierarchical random vectors for S_5 and S_6 [paralleling (18) and (19) above] as,

(24) $$H_{ir} = Z_{i|r} + A_{ir}\, Z_r \;,\; i = 5,6$$

where the centered conditionals, $Z_{i|r}$, and coefficients, A_{ir} , have exactly the same meaning as in (15) and (17) [with (5,6) replacing (1,2)]. So in particular, these upper-level variables are capturing relations between the n_i landmark points in X_i and the n_r landmark points in X_r . The desired *hierarchical model*, $H = (H_1, H_2, H_3, H_4)$, is then defined at the lower level by:

(25) $$H_i = Z_{i|5} + A_{i5} H_{5r} = Z_{i|5} + A_{i5}\,(Z_{5|r} + A_{5r}\, Z_r) = Z_{i|5} + A_{i5}\, Z_{5|r} + A_{i5} A_{5r}\, Z_r \,,\quad i = 1,2$$

(26) $$H_i = Z_{i|6} + A_{i6} H_{6r} = Z_{i|6} + A_{i6}\,(Z_{6|r} + A_{6r}\, Z_r) = Z_{i|6} + A_{i6}\, Z_{6|r} + A_{i6} A_{6r}\, Z_r \,,\quad i = 3,4$$

The parentheses in second equalities in (25) and (26) serve to highlight the recursive nature of these definitions, while the last equalities exhibit the linear relations between H and hierarchical family of basis vectors, $Z = \{Z_{1|5}, Z_{2|5}, Z_{3|6}, Z_{4|6}, Z_{5|r}, Z_{6|r}, Z_r\}$, shown in Figure 5 below. As an extension of the two-level model in (18) and (19), we now see from (25) for example that the second terms involving $Z_{5|r}$ reflect the covariance relations between H_1 and H_2 . Similarly, the last terms involving Z_r in both (25) and (26) reflect additional covariance relations among all four components of $H = (H_1, H_2, H_3, H_4)$.

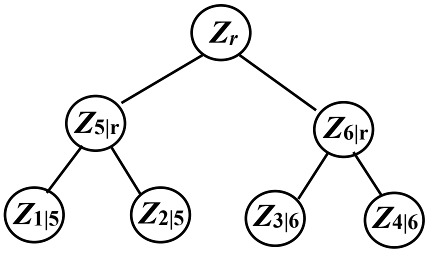

Figure 5. Random Basis Vectors

$$\boldsymbol{K}^H = \begin{pmatrix} \boldsymbol{K}_{11}^H & \boldsymbol{K}_{12}^H & \boldsymbol{K}_{13}^H & \boldsymbol{K}_{14}^H \\ & \boldsymbol{K}_{22}^H & \boldsymbol{K}_{23}^H & \boldsymbol{K}_{24}^H \\ & & \boldsymbol{K}_{33}^H & \boldsymbol{K}_{34}^H \\ & & & \boldsymbol{K}_{44}^H \end{pmatrix}$$

Figure 6. Hierarchical Covariance Matrix

For this three-level example, the hierarchical approximation, K^H, to $K = K(X,X)$, can be defined by specifying the matrix cells shown in Figure 6 (together with symmetry). Following expression (16) in [C1], there are only three types of covariance expressions to be considered, namely within domains (first-level interactions), between adjacent domains (second-level interactions) and between non-adjacent domains (higher-level interactions), as can be illustrated by K_{11}^H, K_{12}^H, and K_{13}^H:

(27) $K_{11}^H = \text{cov}(H_1, H_1) = K_{11}$

(28) $K_{12}^H = \text{cov}(H_1, H_2) = K_{15}\, K_{55}^{-1}\, K_{52}$

(29) $K_{13}^H = \text{cov}(H_1, H_3) = K_{15}\, K_{55}^{-1} K_{5r}\, K_{rr}^{-1}\, K_{r6}\, K_{66}^{-1}\, K_{63}$

But (27) follows from the argument in (22) together with the recursive nature of (25). A first application of (22) [to expression (24)] yields $\text{cov}(H_{5r}) = K_{55}$. But the independence of $Z_{1|5}$ and $H_{5r} (= Z_{5|r} + A_{5r} Z_r)$ together with a second application of (22) [to the first equality in (25)] shows that

(30) $\text{cov}(H_1) = \text{cov}(Z_{1|5}) + A_{15}\,\text{cov}(H_{5r}) = (K_{11} - K_{15}\, K_{55}^{-1}\, K_{51}) + (K_{15}\, K_{55}^{-1}) K_{55}\, (K_{55}^{-1}\, K_{51}) = K_{11}$

Moreover, since $Z_{1|5}$, $Z_{2|5}$ and $H_{5r} (= Z_{5|r} + A_{5r} Z_r)$ are mutually independent, it also follows that

(31) $\text{cov}(H_1, H_2) = \text{cov}[(Z_{1|5} + A_{15}\, H_{5r}), (Z_{1|5} + A_{25}\, H_{5r})] = A_{15}\,\text{cov}(H_{5r})\, A_{52}$

$$= (K_{15} K_{55}^{-1}) K_{55} (K_{55}^{-1} K_{52}) = K_{15} K_{55}^{-1} K_{52}$$

Finally, since all components of Z are independent, the same argument shows that

(32) $\text{cov}(H_1, H_3) = \text{cov}[(Z_{1|5} + A_{15}\, Z_{5|r} + A_{15} A_{5r} Z_r), (Z_{3|6} + A_{36}\, Z_{6|r} + A_{36} A_{6r} Z_r)]$

$$= A_{15} A_{5r}\,\text{cov}(Z_r)\, A_{r6} A_{63} = (K_{15} K_{55}^{-1})(K_{5r} K_{rr}^{-1}) K_{rr}\, (K_{rr}^{-1} K_{r6})(K_{66}^{-1} K_{63})$$

$$= K_{15}\, K_{55}^{-1} K_{5r}\, K_{rr}^{-1}\, K_{r6}\, K_{66}^{-1}\, K_{63}$$

So again, we see that $\text{cov}(H) = K^H$.

3.3 General Modeling Scheme

The above examples should make it sufficiently clear that the general *hierarchical model* consists of a family of random vectors, $H = (H_i : i = 1,..,b)$, where for each basic subdomain, S_i , of S (i.e., leaf of the associated tree), the random vector, H_i, is a nested linear combination of the basis vectors, *Z*, such as in Figure 5 above. In particular, if for each node, i_1, in the tree we now designate the unique path, $i_1 \to i_2 \to \cdots \to i_{m-1} \to i_m \to r$, of successive parents (ancestors) up to the root node, r, as the *root path* for i_1, then the appropriate form of H_i for each leaf node, i , with root path, $i \to i_1 \to i_2 \to \cdots \to i_{m-1} \to i_m \to r$, now takes the form:

$$(33) \qquad H_i = Z_{i|i_1} + A_{i i_1} \left(Z_{i_1|i_2} + A_{i_1 i_2} \left(\cdots \left(Z_{i_{m-1}|i_m} + A_{i_{m-1} i_m} \left(Z_{i_m|r} + A_{i_m r} Z_r \right) \right) \cdots \right) \right)$$

In terms of this notation, the desired covariance for H_i [given by the top half of expression (14) in [C1] for a representative point pair, (x, x'), in X_i] is simply the kernel covariance,

$$(34) \qquad \mathrm{cov}(H_i) = k(X_i, X_i) = K_{ii}$$

In addition, the desired covariance between any pair of leaf vectors, H_i and H_j, with least common ancestor, s [possibly root, *r*, itself] and root paths

$$(35) \qquad i \to i_1 \to i_2 \to \cdots \to i_{p-1} \to i_p \to s \to h_1 \cdots \to h_m \to r$$

$$(36) \qquad j \to j_1 \to j_2 \to \cdots \to j_{q-1} \to j_q \to s \to h_1 \cdots \to h_m \to r$$

is given [in terms of expression (16) in [C1] for point pairs, $x \in X_i$ and $x' \in X_j$]

$$(37) \qquad \mathrm{cov}(H_i, H_j) = K_{i\,i_1} K_{i_1 i_1}^{-1} K_{i_1 i_2} K_{i_2 i_2}^{-1} \cdots K_{i_p s} K_{ss}^{-1} K_{s\,j_q} K_{j_q j_q}^{-1} \cdots K_{j_2 j_1} K_{j_1 j_1}^{-1} K_{j_1 j}$$

Note in particular that hierarchical covariances in (12) for our two-level example and in (27) through (29) for our three-level example are both instances of (34) and (37). In Appendix 1 it is shown that the hierarchical model in (33) continues to exhibit this covariance structure in all cases, and thus provides a general probabilistic formulation of hierarchical covariance matrices. This is of particular importance in that such matrices are themselves *exact* covariance matrices (as observed in [C2, p.5]), and need not themselves be interpreted as "approximations".

3.4 Efficient Algorithms and Storage

While the probabilistic development above provides a more concrete interpretation of hierarchical covariance matrices, it cannot be overemphasized that the real power of these hierarchical structures is their *computational efficiency*, which allows Gaussian Process Regression models to be extended to large data sets. Rather than storing the entire kernel matrix in memory, much smaller block diagonal matrices (covariances of leaves, H_i), are stored along with even smaller matrices found at the parent nodes of the space partitioning tree. Omitting off-diagonal blocks of the covariance matrix (covariances between leaf pairs, H_i and H_j), generates significant gains in scalability since these omitted blocks are only calculated on an as-needed basis using the appropriate tree traversal.

This may appear to simply trade the problem of storage with that of drastically increased calculation requirements. But careful inspection shows that this computation issue is not as serious as one might expect. Referring back to equation (37), suppose that leaves j and k share the same parent node, j_1. Then the covariance between H_i and H_k is given by

$$(38) \qquad \mathrm{cov}(H_i, H_k) = K_{i\,i_1} K_{i_1 i_1}^{-1} K_{i_1\,i_2} K_{i_2 i_2}^{-1} \cdots K_{i_p\,s} K_{ss}^{-1} K_{s\,j_q} K_{j_q\,j_q}^{-1} \cdots K_{j_2\,j_1} K_{j_1\,j_1}^{-1} K_{j_1 k}$$

which is seen to differ from (37) by only the last element, $K_{j_1 k}$. This type of overlap suggests that computational procedures can be recursively structured to avoid repeated calculations of common products such as in (37) and (38). Such recursive procedures are formalized in [C1] and [C2].

While the full set of procedures can be found in these references, the three most basic operations are *matrix-vector products* (O.1), *matrix inversion* (O.2), and *determinant calculations* (O.3). For our present purposes, the application these operations is best illustrated in terms of the log likelihood function,

$$(39) \qquad L(\theta \mid y) = -\tfrac{1}{2}\log[\det(C_\theta)] - \tfrac{1}{2} y' C_\theta^{-1} y - \tfrac{n}{2}\log(2\pi)$$

for a multinormal random vector, $y \sim N(0, C_\theta)$ with hierarchical covariance matrix, C_θ parameterized by θ. Such likelihood calculations are performed many times in the estimation of θ, and require efficient methods for large scale datasets. Having constructed and stored the matrix, C_θ, within the HCA framework, one calculates $\det(C_\theta)$ by the determinant operation (O.3) [which actually calculates the log determinant directly]. One then constructs C_θ^{-1} by the inverse operation (O.2). Finally, this is followed by the calculation of $C_\theta^{-1} y$ using matrix-vector product operation (O.1), which in turn reduces the quadratic form, $y' C_\theta^{-1} y$, to a simple inner product of n-vectors.

In addition to these three main operations which are used exclusively for calculations with training data (y, X), there are also more specialized operations designed for calculations involving covariances, $K(X_*, X)$ with n_t prediction points, X_*. In particular, there is a matrix-vector product operation (O.4) for calculating expressions such as the conditional means in (7) and local marginal effects in (9), while a quadratic form operation (O.5) is used for calculating the conditional covariances in (8). Here it should be noted that while these operations were originally developed in [C2] for the vector case of single prediction points ($n_t = 1$), such procedures are readily extendable to matrices. Hierarchical procedures such, as O.4 and O.5, avoid the need to form full $n_t \times n$ covariance matrices, $K(X_*, X)$.

To make matters more concrete, we conduct a series of simple experiments (using Matlab R2018b and GPStuff (Vanhatalo et al. (2013)). Results of these experiments are displayed in Figure 1 below (with HCA = GPR-HCA and FULL = GPR-FULL). For HCA (where we use 150 landmark points and a maximum of 1000 observations per leaf) we consider sample sizes ranging from 2,000 to 128,000 observations. For FULL we cap the number of observations to 32,000 for Storage and 16,000 for the Matrix Inverse Operation (which uses an efficient mex file[4]). As shown in Figure 7, FULL has a dramatic acceleration of costs with increasing sample size, while HCA's storage and operations are linear in the number of samples. These findings are consistent with [C1] and [C2] where it is shown that *as long as the maximum number of landmark points on each level of the hierarchy is held constant, the overall costs of both computation and storage are linear in the number of samples, n.*

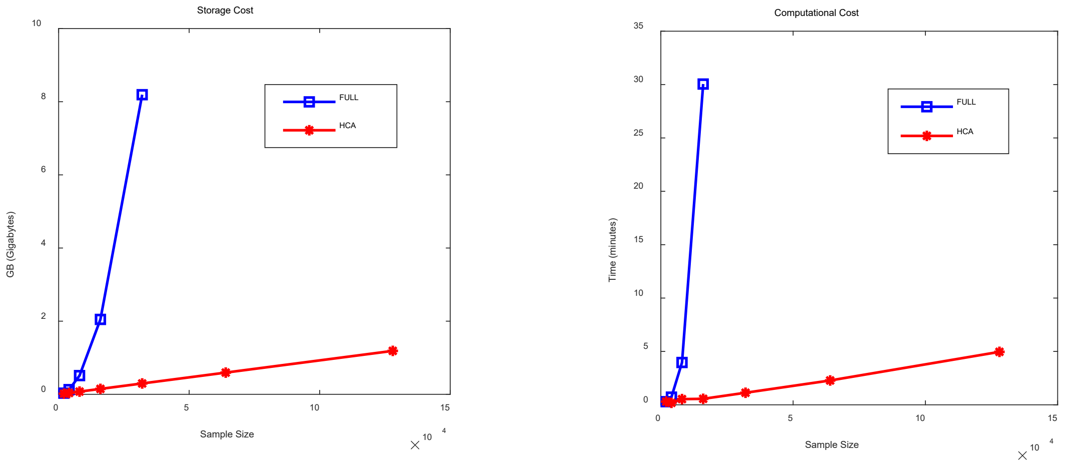

Figure 7. Computation and Storage Comparisons of Matrix Inversion for HCA versus FULL

Finally it should be noted that large kernel matrices tend to be ill-conditioned, and in particular, may lose their positive definiteness when inverted. In expression (3) above, the addition of measurement-error variance, $\sigma^2 I_n$, to matrix K tends to counteract this ill-conditioning for the diagonal blocks of hierarchical covariance matrices, K^H, such as K_{11} and K_{22} in expression (12) above. But this is not true of off-diagonal "landmark" covariance matrices such as K_{rr} in the same expression. So following Chen [C1, section 4.3] we add small regularizing effects to these matrices (which are similar in form to $\sigma^2 I_n$).

3.5 Parameter Estimation

Given this hierarchical covariance approximation structure, together with a set of observed responses, $y = (y_1,..,y_n)'$, and associated attributes, $X = [x_l = (x_{l1},..,x_{ld}) : l = 1,..,n]$, the estimation of mean and covariance parameters for GPR-HCA proceeds along standard lines. First, given that our primary interest is in covariance estimation, we employ the simple kriging

[4] Blake, Eric (2015). Fast and Accurate Symmetric Positive Definite Matrix Inverse, Matlab Central File Exchange.

conventions of estimating the common mean, μ, of responses in (1) by their sample mean, $\overline{y} = \frac{1}{n}\Sigma_i y_i$. In this way, we can focus on response deviations about this sample mean, and proceed to estimate the covariances kernel parameters, $(v, \tau_1, .., \tau_d)$ in (4), together with measurement variance, σ^2, in (1). Thus by letting $\theta = (v, \tau_1, .., \tau_d, \sigma^2)$ denote the full vector of parameters to be estimated, and making the parameter dependency of K explicit by writing K_θ in (3), we now treat Y in (3) as a deviation vector with distribution, $N[0_n, K_\theta(X, X) + \sigma^2 I_n]$, so that the log likelihood function in (37) takes the more explicit form,

$$(40) \quad L(\theta \mid X, y) = -\tfrac{1}{2}\log(\det[K_\theta(X,X) + \sigma^2 I_n]) - \tfrac{1}{2} y'[K_\theta(X,X) + \sigma^2 I_n]^{-1} y - \tfrac{n}{2}\log(2\pi)$$

In these terms our (positive) parameters, $\theta = (\theta_i : i = 1, .., d+2)$, are postulated to have independent log-Gaussian priors, $p(\ln \theta_i)$, yielding a log posterior density of the form:

$$(41) \quad \log p(\theta \mid y, X) = \log p(y \mid X, \theta) + \sum\nolimits_{i=1}^{d+2} p(\ln \theta_i) - \log p(y)$$

$$= L(\theta \mid y, X) + \sum\nolimits_{i=1}^{d+2} p(\ln \theta_i) - \log p(y)$$

It is this *energy function* that is maximized to obtain *maximum a-posterior*i (MAP) estimates, $\hat{\theta}$, of all parameters. In the numerical simulations and applications to follow, all parameter priors are assumed to have the common form, $\log \theta_i \sim N(2, 9)$, which essentially yields vague priors with conservative mean values for both length scales and variances.

The estimation procedure for this GPR-HCA model was programmed in Matlab, and optimized using Matlab's *fmincon* routine. Matlab code is available from the authors.

4. Simulation Analyses

To investigate the behavior of GPR-HCA, we begin with a simple simulation model that allows us to explore the computational efficiency of this method, well as its predictive accuracy. To do so, we employ the following two-variable model with Gaussian noise,

$$(42) \quad y = \cos(8x_2 - 3.5) + .8\left[\sin(4x_1x_2) + \cos(2x_1 + 6.66)\right] + \varepsilon \, , \; \varepsilon \sim N(0, \gamma)$$

defined over the unit square $(x_1, x_2) \in [0,1]^2$. Unless otherwise noted, the noise variance, γ, is set to 0.25. For our later purposes, the associated local marginal effects for this model are given by:

$$(43) \; \frac{\partial E[y]}{\partial x_1} = 3.2\, x_2 \cos(4x_1x_2) - 1.6\sin\left(2x_1 + 6.66\right)$$

$$(44) \; \frac{\partial E[y]}{\partial x_2} = -8 \, \sin(8x_2 - 3.5) + 3.2\, x_1 \cos(4x_1x_2)$$

This two-variable setup allows the model mean, $E(y)$, to be displayed visually as in Figure 8(a). This not only provides a contextual feel for the underlying relationship, but also allows a direct comparison with the estimated mean, $\hat{E}(y)$, in Figure 8(b) [to be discussed later].

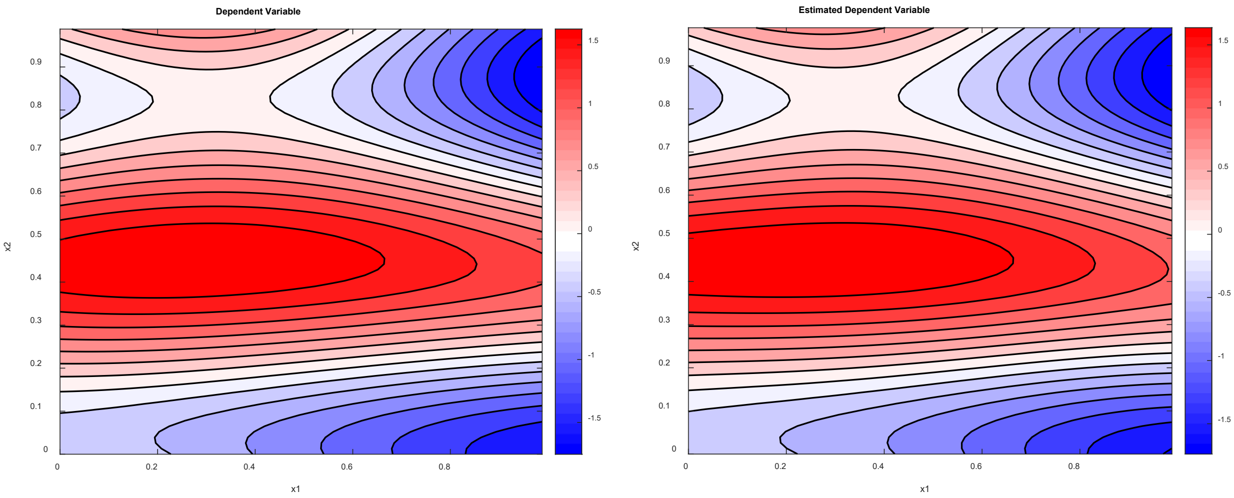

Figure 8. Contour Plots of the (a) True Mean Dependent Variable and (b) GPR-HCA Estimated Mean Dependent Variable

While the most natural benchmark for comparison is in terms of the full model (GPR-FULL), this estimation procedure is constrained to small sample sizes (at most 15,000 samples) rendering such comparisons infeasible on larger datasets. Consequently, we also employ the more scalable algorithms, NNGP and GBM, as mentioned in the in the introductions. This allows scalability comparisons at much larger sample sizes.

In Section 4.1, we begin by comparing the scalability and accuracy of GPR-HCA with GPR-FULL (again denoted as HCA and FULL) over a limited range of simulated sample sizes from model (42), and then consider more extended-range comparisons with GBM and NNGP in Section 4.2. Finally, we examine both the scalability and accuracy of Local Marginal Effects for GPR-HCA in Section 4.3.

4.1 Limited-Range Comparisons with GPR-FULL

Here it is instructive to compare these two methods both with respect to parameter estimation and out-of-sample predictions.

Parameter Estimation. Turning first to the relative scalability of parameter estimation procedures, the computation times for estimating parameters, $\theta = (v, \tau_1, \tau_2, \sigma^2)$, by both HCA and FULL are shown in Figure 9 for a selected range of sample sizes up to 15000. Here (as in all examples to follow) HCA is parameterized using leaves of maximum size 1000 together with 150 landmark points at each hierarchical level. From this figure it is evident that even for sample sizes as small as a few hundred, HCA is already orders of magnitude faster than FULL.

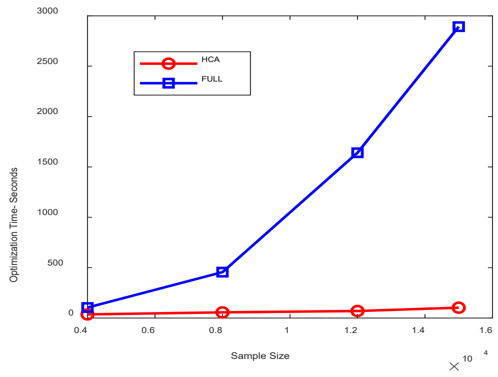

Figure 9. Comparison of Computation Times
for GPR-HCA and GPR-FULL

To gauge the similarity of parameter estimates for these two methods, it is instructive to compare the energy functions (41) generated by HCA and FULL for a specific case (using a training set of 2117 points). Following Chen and Stein [C2, Figure 5], we focus on a subplot of the (τ_1, τ_2) plane, holding all other parameters at their optimal values. Results for these two key parameters are shown for FULL and HCA in Figures 10(a) and 10(b), respectively, where length scales are plotted in terms of their log values, $\ln(\tau_1)$ and $\ln(\tau_2)$, and where the large dot in each figure denotes the optimal parameter values. Here it is clear that with only 150 landmark points, the energy functions are virtually identical in shape and size. More generally, it appears from further simulations with this model (as well as those in Chen & Stein [C2]) that there is little to be gained by further increases in the number of landmark points.

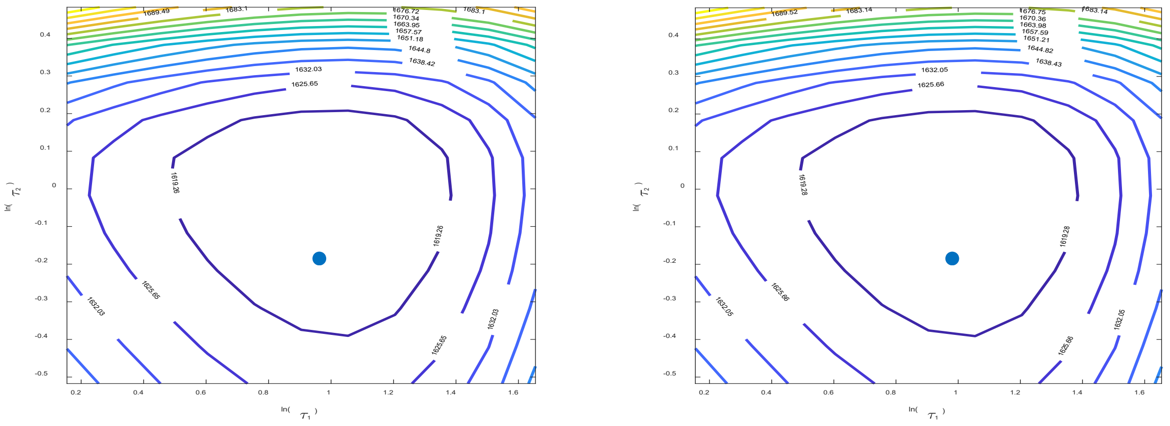

Figure 10. Energy Function for (a) FULL and (b) HCA

Out-of-Sample Prediction. With respect to predictions, comparable batch-sample procedures were carried out for a range of random test samples up to 350K. Computation times for HCA and FULL are shown in Figure 11(a) [where the linearity of computation times for FULL as well as HCA results from the batch nature of such computations]. While such times are seen to be about twice as large for FULL in the present illustration, such times depend critically on the size of the training set used (and for large training sets are of course infeasible for FULL).

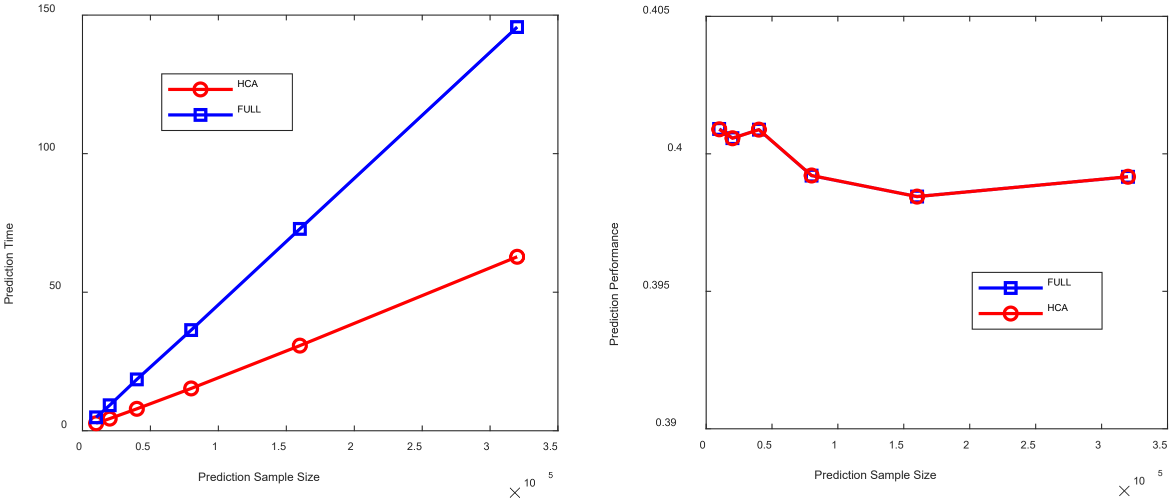

Figure 11. Comparisons of Batch-Sample Predictions for FULL and HCA with respect to (a) Computation Times, and (b) Accuracy in terms of MAE

Turning to the comparison of mean absolute errors in Figure 11(b), these errors are in fact so close in values that the blue curve for FULL cannot even be seen. So for predictions as well as parameter estimates, the key point is again that even in the range where the full version of GPR is feasible, the present hierarchical covariance approximation is not only dramatically faster, but also appears to exhibit no substantial loss of accuracy. In the present example it is also of interest to note that these mean absolute errors are remarkably small. In fact, for the model in (42) with normal errors, the absolute deviations of similated values about the mean are well known to be distributed as a "folded normal" with mean $\gamma\sqrt{2/\pi}$, which for the present case of $\gamma = 0.5$ yields 0.3989. So it should be clear that the prediction errors above are almost entirely due to fluctuations generated by the model error term itself.

4.2 Extended-Range Comparisons with NNGP and GBM

As should be evident from Figure 9, the linear scalability properties of GPR-HCA allow for the analysis of data sets vastly larger than those feasible for GPR-FULL.[5] So for larger data sets, it is appropriate at this point to compare HCA with other well-known linearly scalable prediction models, namely NNGP and GBM, as mentioned above. Computation times (averaged across 3 different runs) over a selected range of sample sizes up to n =500,000 are shown for HCA, NNGP, and GBM in Figure 12(a).[6] These times also include predictions for a randomly selected

[5] In fact, the temperature application of Chen and Stein [C2] involves more than 2 million observations.

[6] The explicit samples sizes shown are [10,000, 20,000, 40,000, 80,000, 160,000, 320,000, 500,000].

set of 10,000 out-of-sample points.[7] Using these points, the relative prediction accuracy is then compared in terms of mean absolute errors (MAE), as shown in Figure 12(b).

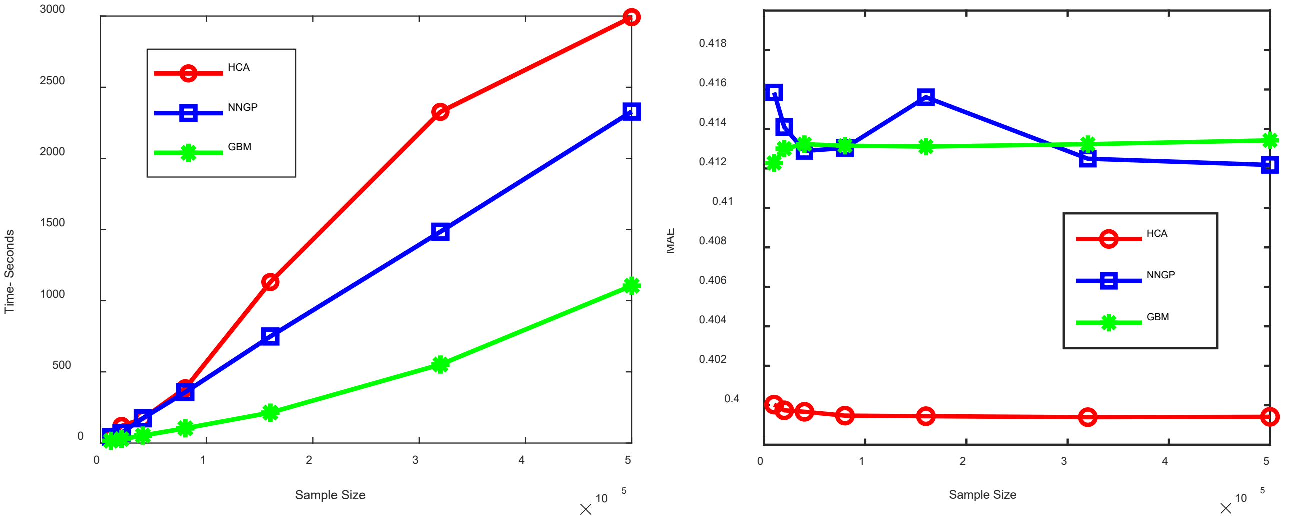

Figure 12. Comparisons of GPR-HCA with both NNGP and GBM in terms of (a) Computing Times, and (b) Mean Absolute Errors

The key feature of the computing-time plots in Figure 12(a) is their approximate linearity, which underscores the demonstrable fact that all three procedures have complexity of order $O(n)$. However, it should be stressed that the relative magnitudes of these computation times are more difficult to compare. On the one hand, both the both the NNGP and GBM models involve many alternative specifications, as well as tuning parameters that have not been fully optimized. In particular, the present version of NNGP used is the conjugate version (spConjNNGP) in the R package, spNNGP, with default settings including an exponential specification of the kernel function [as documented in Finley et al., 2020]. With respect to GBM, the cross-validation method used to gauge iteration numbers involves many repetitions of model estimations (and can be replaced by faster but less accurate methods). On the other hand, it should be emphasized that both NNGP and GBM have been written in optimized C\C++ code, which is well known to be dramatically faster than the Matlab code used here for HCA.

Turning next to the relative accuracy of such predictions, it is clear from Figure 12(b) that for this simulation example, HCA is uniformly more accurate than both NNGP and GBM. Moreover, while the MAE values exhibited by HCA are almost identical to the model fluctuations themselves (as mention at the end of Section 4.1 above), those of both NNGP and GBM are noticeably higher. However, it must again be emphasized that there is a speed-accuracy tradeoff here, especially for GBM. We elected to use 10,000 trees for GBM, which is a typical size in practice. The results for 100,000 trees (not shown) yield predictions very close to

[7] Computation times for HCA automatically include calculations of Local Marginal Effects at each prediction point (which are not directly relevant for either NNGB or GBM). But these add little in the way of time differences.

HCA, though with computing times that are actually slower than HCA. For NNGP, we have increased the default value of k=2 to k=5 in the k-fold cross-validation procedure for estimating covariance parameters. But even larger values appear to have little effect on prediction accuracy. However, it should also be noted that, unlike expression (4) above, the kernel functions employed in NNGP are isotropic, and thus somewhat less flexible than (4) for prediction purposes. In summary, the essential message of Figure 12 from our point of view is that the present hierarchical covariance approximation method is competitive with existing alternative models both in terms its scalability and prediction accuracy.

To gain further appreciation for the accuracy of this method, the two-dimensional nature of our present simulation model allows a direct visual comparison of the contours of $E(y)$ in Figure 8(a) above with the estimated contours, $\hat{E}(y)$, for GPR-HCA as shown in Figure 8(b) above (for a training sample of 12,569 observations and 150 landmark points).[8] Here the remarkable similarity of these contours underscores the ability of GPR-HCA to faithfully capture the full structure of the underlying model.

4.3 Scalability and Accuracy of Local Marginal Effects

As detailed in Dearmon & Smith (2017), a key attractive feature of GPR-FULL is its ability to predict not only $E(y)$ values at out-of-sample points but also to estimate the local rates of change of these values with respect to key explanatory variables, i.e., the Local Marginal Effects (LMEs) given by expression (9) above. In particular, for the specific squared exponential kernel in expression (4), it follows by direct calculation that for prediction points, $X_* = [x_{*_i} : i = 1,..,q]$,

$$(45) \qquad \frac{\partial K(x_{*l}, X)}{\partial x_{*l,j}} = \quad = -\tfrac{1}{\tau_j^2}[k(x_l, x_{*1})(x_{lj} - x_{*1j}),..,k(x_l, x_{*q})(x_{lj} - x_{*qj})] \, , \; j = 1,..,d$$

Here we consider how well the present more scalable GPR-HCA version captures these same effects. With respect to computation times for LME predictions, it is enough to note that these times now depend not only on the particular batch scheme employed, but also on the number of explanatory variables considered. Other than these more complex dependencies, the results for our simulation model (not shown) continue to be linear, and are qualitatively similar to the linear graph for HCA predictions in Figure 11(a).

Of more interest for our present purposes is the accuracy of these LME predictions. As with the comparisons of $E(y)$ and $\hat{E}(y)$ in Figure 8 above, the quality of LME predictions is best seen visually. In Figures 13 and 14 below we compare contour plots of the exact LMEs for this simulation model with their associated predictions based on GPR-HCA. If the true partial derivatives with respect to each variable, x_j , [given in (43) and (44)] are denoted by

[8] These predictions are computed for a regular grid of points in $[0,1]^2$ and, in a manner similar to Figure 8(a), contours are then interpolated and plotted using the Matlab program, *contour.m*. Similar procedures are used to obtain Figures 13(b) and 14(b) below.

$LME(y\,|\,x_j)$, and if the associated estimates based on HCA [obtained from (9) together with (45)] are denoted by $\widehat{LME}(y\,|\,x_j)$, then the contour plots for $LME(y\,|\,x_1)$ and $\widehat{LME}(y\,|\,x_1)$ are shown in Figure 13, and those of $LME(y\,|\,x_2)$ and $\widehat{LME}(y\,|\,x_2)$ are shown in Figure 14.

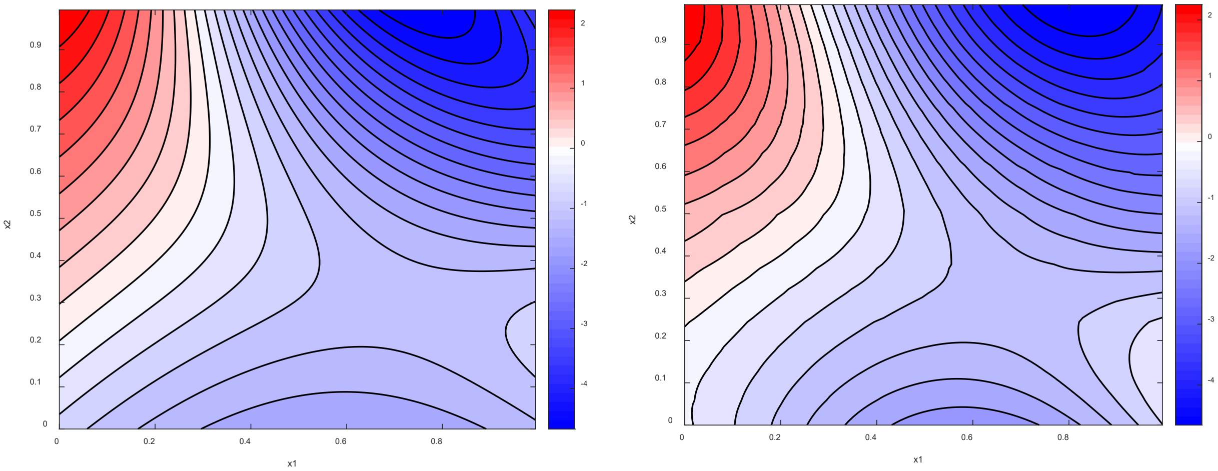

Figure 13. Contour plot of (a) $LME(y\,|\,x_1)$, and (b) $\widehat{LME}(y\,|\,x_1)$

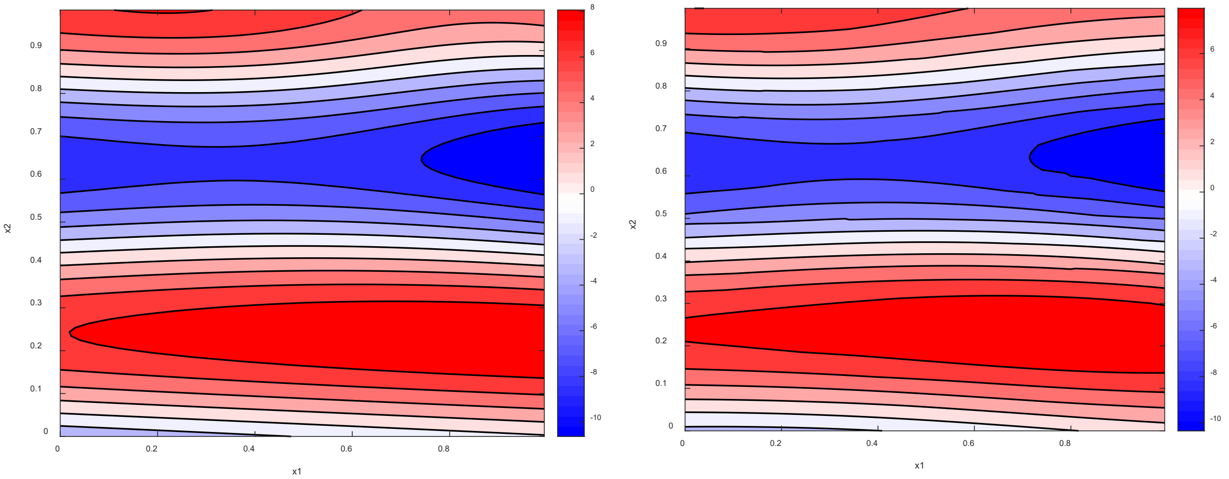

Figure 14. Contour plot of (a) $LME(y\,|\,x_2)$, and (b) $\widehat{LME}(y\,|\,x_2)$

These two figures suggest that the ability of GPR-FULL to capture local marginal effects is indeed well preserved by GPR-HCA (even with only 150 landmark points at each hierarchy level). Moreover, while the smooth nature of our present two-dimensional example allows these derivatives to be easily plotted and visualized, the empirical example developed in the next section shows that such local marginal effects can also be identified in more realistic multi-dimensional applications.

5. Empirical Application: Housing Prices in Oklahoma County

In this final section, GPR-HCA is applied to residential parcels found in the Oklahoma County Assessor's Database. As seen in Panel (a) of Figure 15 below, Oklahoma County is centrally located within the state and contains several cities including Edmond, Bethany, and Nichols Hills as well as the largest portion of the state capital, Oklahoma City. From a real estate perspective, Oklahoma City represents a secondary or tertiary investment market; significant heterogeneity exists across parcels, more than would typically be present in a dense tier-one urban corridor.

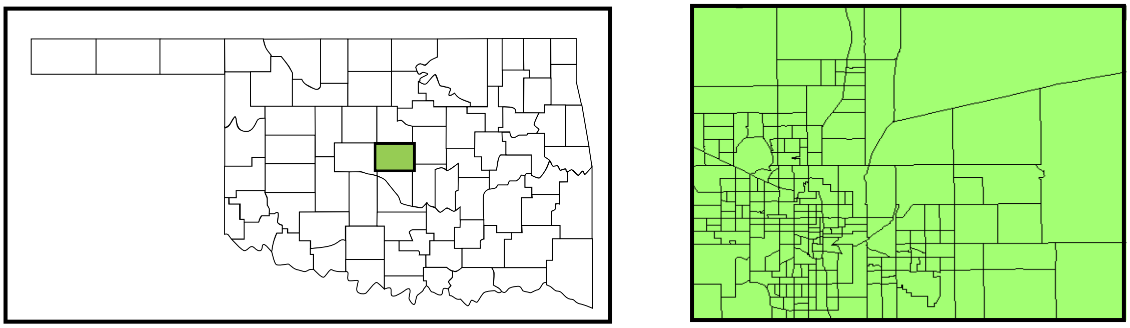

Figure 15. (a) Map of Counties in Oklahoma, (b) Map of Census Tracks in Oklahoma County.

The spatially varying size of census tracts seen in Panel (b) is suggestive of this heterogeneity. The downtown core is found in the densest area of tracks. But this density dissipates as one moves outwards towards the borders of the county, especially to the North and East. This heterogeneity presents some unique modeling challenges; challenges that are very much absent from our well-behaved simulation above.

Technical Considerations. Consequently, some key enhancements are necessary to improve HCA's effectiveness for this real-world application. Of particular concern is the stability of the optimization routine– an issue previously noted by Chen and Stein (2017). After much investigation, we find that Matlab's *inv* function is not sufficiently accurate for our exercise; replacing *inv* by Matlab's more robust *backslash* operator in all HCA algorithms drastically improves the stability (and reproducibility) of the optimization routine. As a secondary enhancement, a more judicious and careful selection of landmark points is conducted. Rather than using a random draw of points, we opt for k-means clustering on our *x* values scaled by preliminary lengthscale estimates.[9] The number of clusters is set to the desired number of landmark points, and the training point nearest the centroid of each cluster is selected as the landmark point for that cluster. This ensures a diverse spread of points across the appropriate region of the tree. [10]

[9] The use of k-means for choosing landmark (or inducing) points is quite common in the literature (Park and Choi, 2010; Hensman et al., 2015).

[10] Changes of a more indirect nature are to allow tree indexing on different sets of attributes. For our present purposes, we partitioned on all variables, except sale year, which made the grouping more spatial than temporal.

When higher numbers of landmark points are desired for increased prediction accuracy, a two-stage estimation procedure may be warranted. Here we have found that by using a small number of landmark point in a first stage to obtain initial parameter estimates, convergence times for a larger number of landmark points in a second stage can be substantially reduced.

Housing Data. Data for this application are taken from the Oklahoma County Assessor's database from 2010 to 2018 (as well as 2019 for building permit information); most of these are certification databases required for assessments. To minimize data errors and outlier events, we focus on residential sales greater than $20,000 and involving house of more than 100 square feet. The training dataset consists of 110,837 residential sales, and is used to predict sales prices for 220,030 parcels if sold in 2018. For purposes of this exercise, just eight explanatory variables are used for price prediction: sale date, locational coordinates, lot size, square feet, year built, neighborhood code, and subdivision id.[11] Summary statistics for these datasets are provided in Table 1. Prediction data are, on average, associated with older, smaller homes located in more established neighborhoods. Sales data are consistent with the idea of suburban residential development where substantial numbers of new, large homes are developed and sold, pushing up the square feet and year built.

Training Dataset

	Price	Sale Date	Cx	Cy	Square Feet	Lot Size	Year Built	Neighborhood Code	Subdivision ID
Mean	173,140	201383	2111552	199294	1,839	0.291	1978	3146	15132
Std. Dev.	158,939	259	29468	33302	863	0.33	27	1117	5063
Min	20,500	200906	2065510	137578	188	0.02	1889	1001	1160
Max	5,200,000	201806	2224110	264539	20,021	3	2018	4944	26724

Prediction Dataset

	Sale Date	Cx	Cy	Square Feet	Lot Size	Year Built	Neighborhood Code	Subdivision ID
Mean	201806	2113115	191884	1715	0.302	1971	2905	14100
Std. Dev.	0	29491	32029	886	0.353	25	1089	4859
Min	201806	2065510	137578	502	0.034	1889	1001	1160
Max	201806	2224110	264490	20021	2.999	2018	4944	26724

Table 1. Summary Statistics

Turning next to model results, we begin in Figure 16 with a spatial comparison of GPR_HCA predictions and corresponding Assessor assigned market values for each of the 220,030 parcels.

[11] Assessor data also provides estimates of market value for each parcel. These are only used in the assessment of predictive model performance.

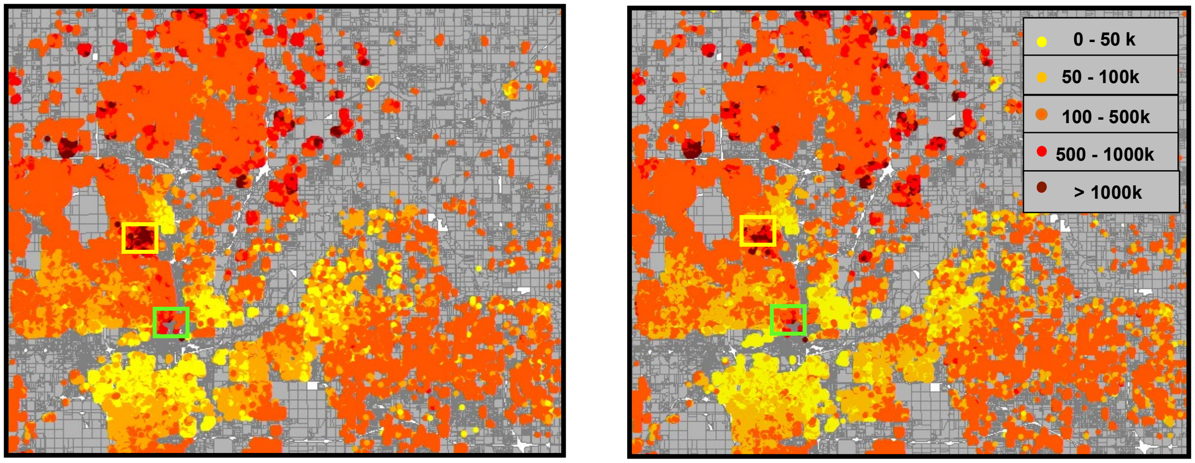

Figure 16. (a) Sales Values predicted by GPR_HCA, (b) Oklahoma County Assessor Market Values [the yellow and green boxes are discussed below]

From a visual perspective, the GPR_HCA predictions are seen to match quite well with County Assessor's Market Values (with mean predicted value, $167,000, slightly higher than the Assessor values ($163,000). This is also seen by a comparison with GBM in Figure 17 below. Here the blue histogram plots error frequencies for GPR_HCA and the red histogram plots those of GBM (with 100K trees) for the same data. Here it is clear that GPR_HCA is much more concentrated around zero, with a Mean Absolute error of $22,958 versus $30,732 for GBM.

However, the plot of Assessed versus Predicted Values in Panel (a) [again with blue denoting GPR_HCA prediction] shows that for very expensive homes (above $2 million) GPR_HCA exhibits some underestimation errors that are noticeably more extreme than GBM. This is even more dramatic at the low end where a few GPR_HCA estimates are actually negative. We return to this issue in the concluding remarks. But for the present we simply note that these outliers involve less than 0.1% of the entire sample.

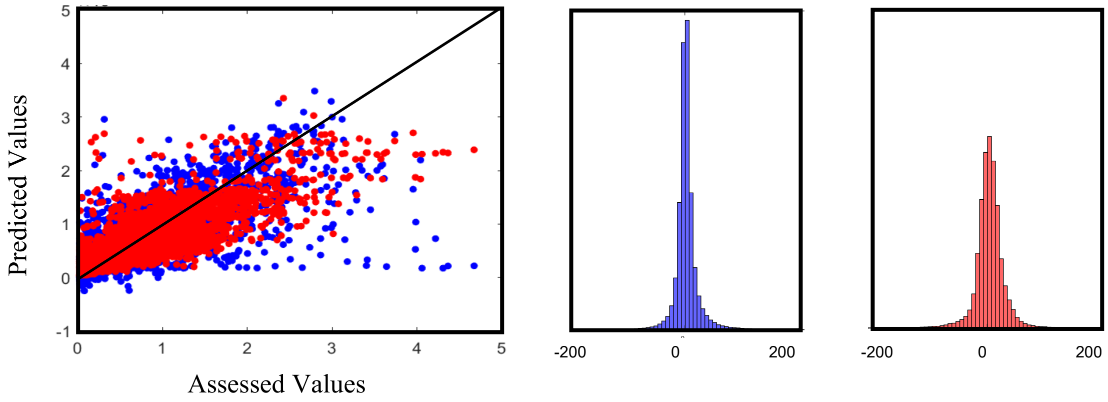

Figure 17. (a) Predicted versus Assessed Values in millions of dollars, where blue points denote GPR_HCA predictions and red points denote GBM predictions, (b) Histogram of GPR_HCA prediction errors in thousands of dollars, and (c) GBM prediction errors in thousands of dollars.

Local Marginal Effects. Finally, we turn to the analysis of local marginal effects, which represents a key contribution of this current work. For this empirical exercise we focus on the Local Marginal Effect of Square Feet (LME_sqft), which represents the estimated impact of an additional square foot on sales price, given the other attributes of a house. We obtained estimates of LME_sqft for all 220,030 prediction parcels. The overall distribution of these values, shown in Panel (a) of Figure 18, is seen to be roughly normally distributed about a mean value of $64.55 (which is just below the low-end of per square foot remodeling costs of adding new square feet[12]). We also show the spatial distribution of LME_sqft values in Panel (b). A comparison with Figure 16 above suggests that such magnitudes are sensitive to location, and that larger magnitudes of LME_sqft are roughly associated with higher home prices.

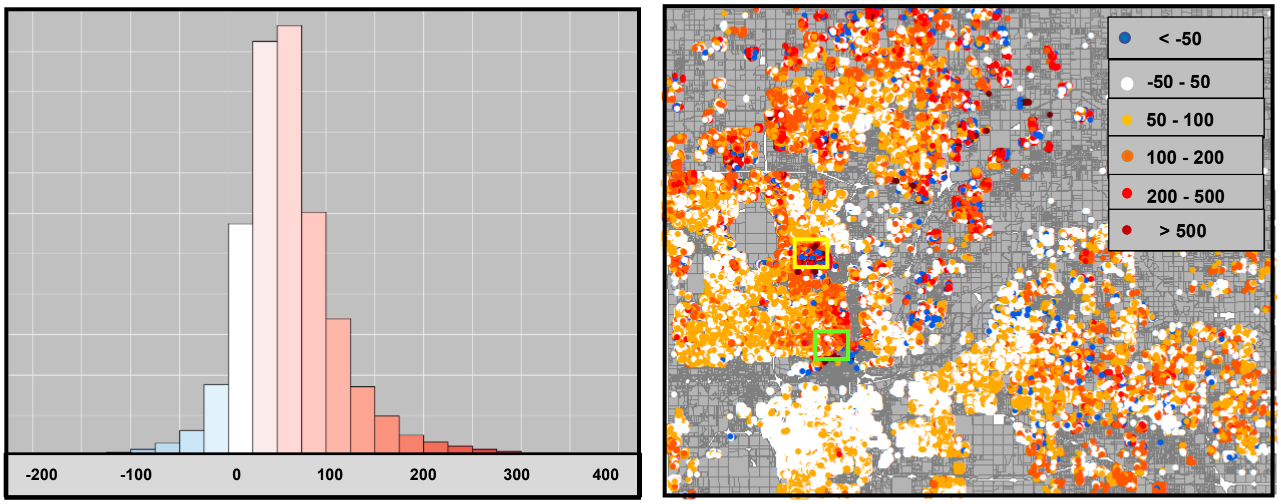

Figure 18. (a) Frequency distribution of LME_Sqft for the 220,030 prediction parcels in Oklahoma County (with positive values in shades of red, and negative values in blue). (b) Spatial distribution of LME_Sqft for these parcels (boxes are repeated from Figure 16)

More importantly, with the finer resolution implicit in LME analysis, we can now uncover more nuanced and detailed economic phenomena that would have been obscured by less granular methods. As examples, we focus on two smaller areas within Oklahoma County which appear to involve somewhat different aspects of economic development. In view of space limitations, we provide only an informal examination of these aspects.

We begin with the densely populated area of Oklahoma City shown in Figure 19 (corresponding to the green box in Figures 16). Here residences are characterized by small lots laid out on a fairly uniform grid. The top two panels show predicted and assessed values in this area, again reflecting the goodness of fit seen at the county-wide level Figure 16 [where the smaller prices in the legend of panel (b) here reflect the sparsity of homes above $1 million in this area]. The most

[12] A cursory search suggests that the lower bound on a room addition is about $80 per square foot (as for example in https://www.ownerly.com/home-improvement/home-addition-cost/, https://www.homeadvisor.com/cost/additions-and-remodels/build-an-addition/ , and https://www.homelight.com/blog/room-addition-cost/) .

expensive homes in the southeast corner are just north of downtown, and consist of the two historic neighborhoods, Heritage Hills and Mesta Park. (The red neighborhoods further north, Edgemere and Crown Heights, are also historic areas). Turning to estimates of LME_sqft in panel (c) [corresponding to the green box Figure 18] we see that within the highest price Heritage Hills area (denoted by the yellow ellipse), there are a number of negative LME_sqft values shown in blue. This is indicative of the large homes found in this historic neighborhood, where further expansion is evidently less attractive. In fact, the largest house in our training dataset, at a size of over 20,000 square feet, is located in this neighborhood.

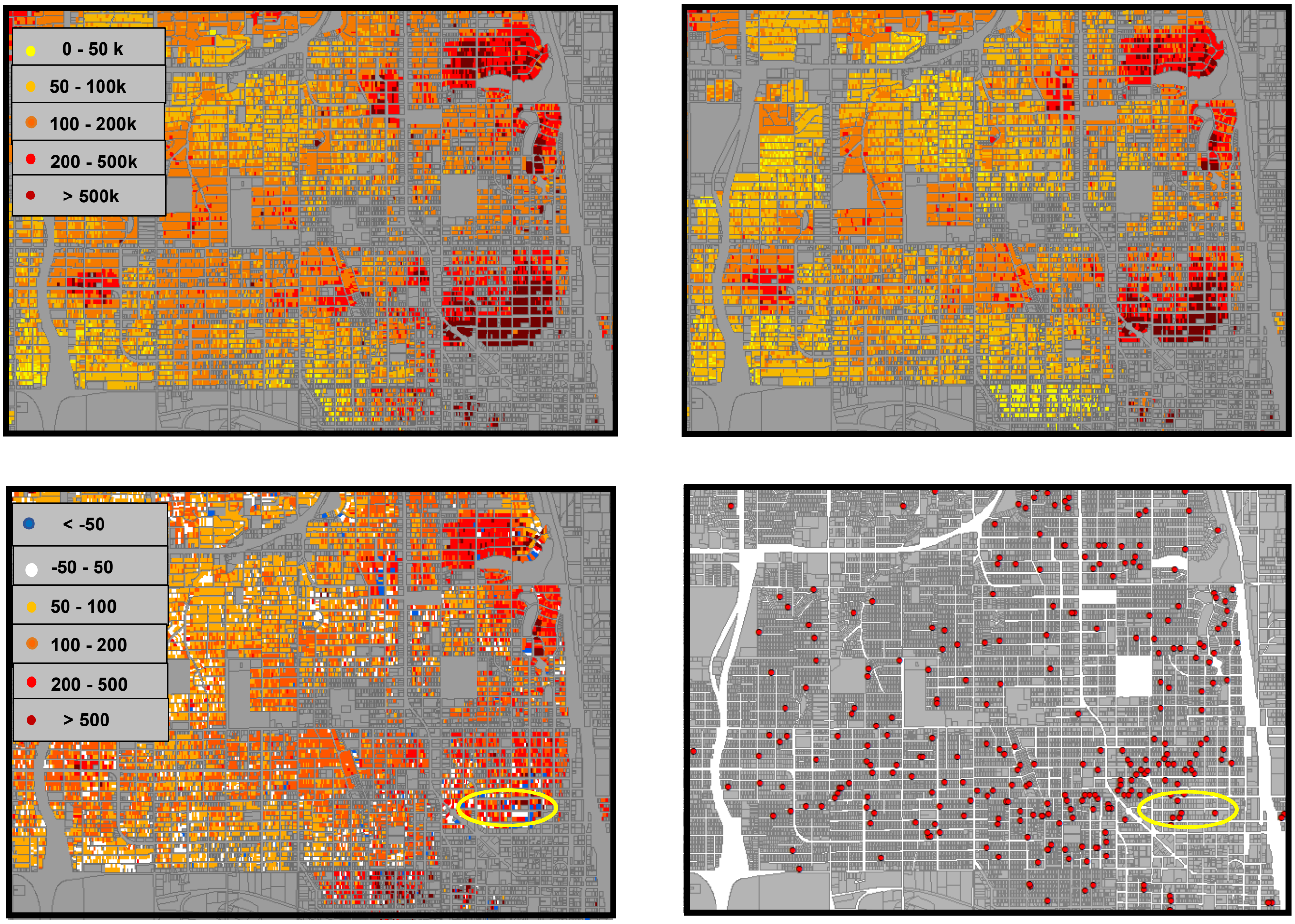

Figure 19. (a) Sales Values predicted by GPR_HCA, (b) Oklahoma County Assessor Market Values, (c) LME_sqft estimates (with yellow ellipse denoting the highest priced area), and (d) Building Permits issued in 2018-2019.

But on the north and west peripheries of this area one sees more uniform positive values of LME_sqft, where proximity to both this higher priced housing area and downtown appear to offer attractive expansion opportunities. This is further supported by data on building permits for the same period[13] [panel (d)] which show that such permits are most highly concentrated in the same area. As one moves further away to the north and west, both LME_sqft values and the density of building tend to decrease. Taken together, the results are strongly suggestive of the *spatial-spillover* effects widely studied in the housing literature [see for example, in Defusco et al. (2018)]. But while such effects are typically analyzed at a broader regional scale,[14] the present results suggest that LME analysis can provide meaningful information at the local neighborhood level.

While spatial spillovers are associated with trends in LME_sqft values at the neighborhood level and higher, there are also more localized development opportunities associated with individual homes or parcels. One type of local development, referred to in the planning literature as *spatial-infill* development (Landis et al., 2006; Daisa and Parker, 2009; McConnell and Wiley, 2010), includes both the development of vacant land in nearly built-up areas and the redevelopment of underutilized parcels. Such development is driven less by spatial trends in housing prices than by local variation in such prices. Adjacent parcels exhibiting a high degree of price variation may have significant differences that can be exploited by developers for profit. For commercial properties, an empty parcel of land sandwiched between two urban high rises is the most obvious example.[15] For residential properties, such differences can be more subtle. For example, older and smaller homes might actually be demolished to make room for more stately homes, provided their locations are in highly desirable areas. Here one might expect smaller homes to exhibit positive LME_sqft effects on price, especially when in close proximity to larger more expensive homes. Moreover, if these larger homes themselves tend to be overbuilt, the effect of an additional square foot might in fact be *negative*, leading to high local variation in such values.

In our present data, a good example is provided by the small city of Nichols Hills just north of Oklahoma City as shown in panel (b) of Figure 21 below (the slightly larger region shown in the other three panels corresponds to the yellow boxes in Figures 16 and 18). The median housing value ($686,300) in this wealthy community is more than four times that of Oklahoma City. The highest priced homes (over $1 million) in Panel (a) are seen from Panel (b) to be clustered around the golf course on the left and the smaller park on the right. The corresponding values of LME_sqft in Panel (c) exhibit much more extreme volatility than those of Figure 20 above, and in particular, contain many more negative values. The large size of these homes is also evident from the large lots seen in this area. Finally, the building permits shown in Panel (d) are seen to be clustered in and around this same area. So, as indicated the discussion above, the presence of such price volatility may indeed be creating new opportunities for development.

While such conjectures clearly require further analysis, the purpose of these examples is mainly to illustrate how this GPR-HCA model and its corresponding LME estimates can in principle be used to quickly identify possible areas for new development in large data sets. Finally, while we have here focused explicitly on the identification of development opportunities in a real estate context, it should be clear that a wide range of additional spatial applications are possible.

[13] This building permit data is taken from the County Assessor's database, with dates issued in 2018 or later. Building costs for permits in our data set all exceed $5,000.

[14] One exception is the recent paper by Cohen and Zabel (2020) which analyzes such spillover effects at the census tract level in the Greater Boston Area.

[15] Such situations do not usually occur in tier-one markets.

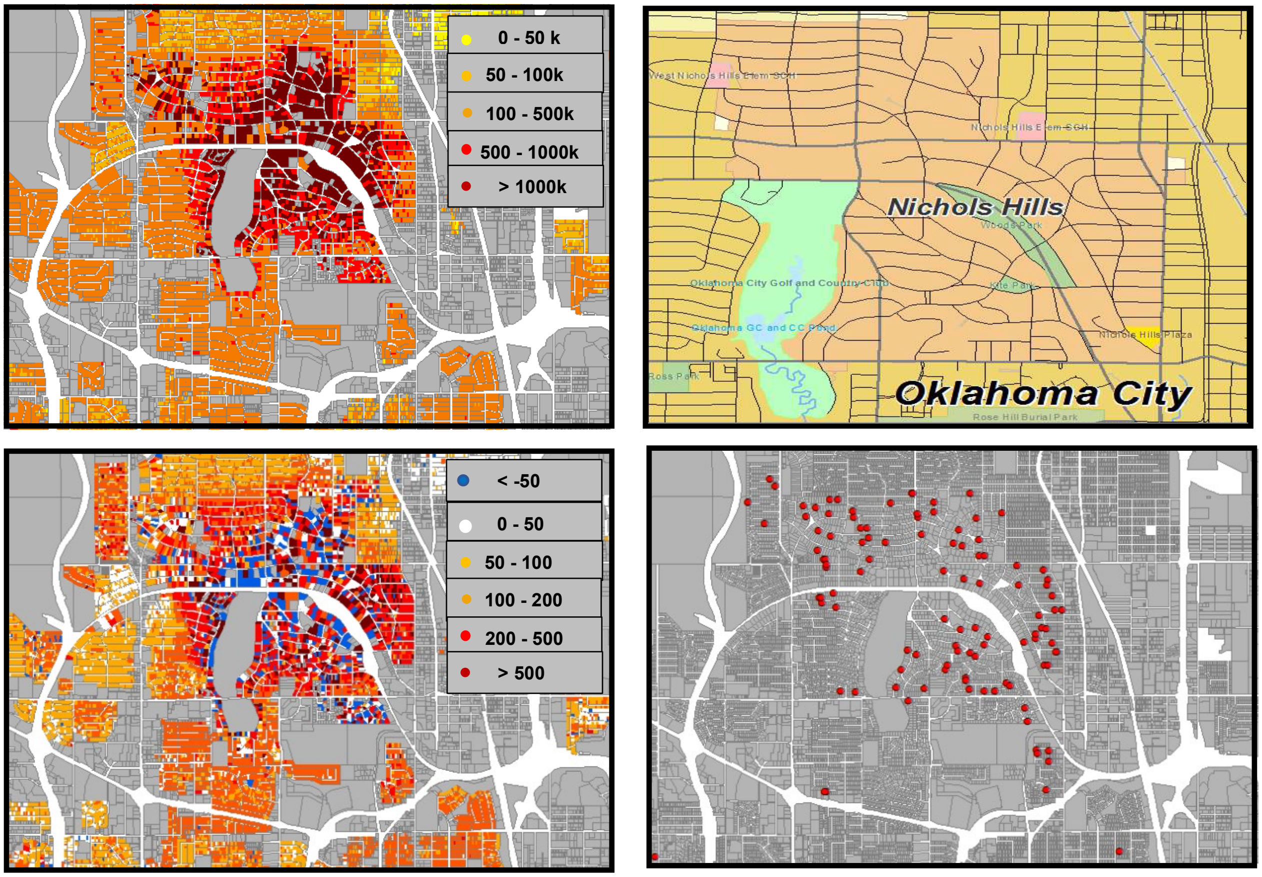

Figure 21. (a) Sales Values predicted by GPR_HCA, (b) Street Map of Nichols Hills, (c) LME_sqft estimates, and (d) Building Permits issued in 2018-2019.

6. Conclusions and Directions for Further Research

In this paper we have systematically developed the hierarchical covariance approximation to Gaussian process regression (GPR-HCA) created by Chen and his co-workers ([C1], [C2]), and have extended this method to include analyses of the local marginal effects (LMEs) generated by this model. Our main objective has been to show how this scalable extension of GPR can be applied to large spatial data sets, such as county assessor data. In particular, we have applied this model to county assessor data for three adjacent counties in Oklahoma, where it was shown that the estimates of both price predictions and local marginal effects generated by GPR-HCA can be used to analyze such data at scales never before possible with standard GPR.

However, the present analysis leaves certain important questions unanswered. A first issue relates to the apparent instability of predictions for extreme values. Investigations with smaller subsets of the Oklahoma data show that this is a problem with GPR_FULL itself, and is not simply a feature of GPR_HCA. In the case of negative predictions, it should be noted that (as with ordinary regression) the fundamental Gaussian assumption itself necessarily allows negative predictions. The standard approach here is to analyze the log of the dependent variable, and convert back to make predictions. But conditional means of log-normal variates do not exhibit the same scalability properties as those of normal variates, and would require extensive parallel computing in order to be implemented for large data sets. However, for housing prices in particular, one possible alternative is to replace standard conditional-mean prediction with predictors more closely related the common real estate practice of forming offer prices based on weighted averages of recent similar sales (known as "comps"). Initial results using GPR covariances as "similarity weights" appears to be promising, and will be reported in a subsequent paper.

A more fundamental issue that has important consequences for both practitioners and researchers is the treatment of *uncertainty* in statistical decision making. For example, with respect the parcel-level investment decisions discussed in our Oklahoma application, measures of uncertainty could help individuals sift through thousands of parcels to identify investment opportunities with higher risk-adjusted rates of return.

But while the GPR model itself does allow for some degree of uncertainty in terms of the predictive distribution in expressions (6) – (8) above, no corresponding posterior distributions are available for either the derivatives of these predictions [i.e., the LMEs effects in expression (9)], or for the basic parameter estimates, $\hat{\theta}$, underlying the model itself. While it is in principle possible to use GPR-HCA to approximate posterior distributions for all such quantities in terms of Markov Chain Monte Carlo methods, such an approach currently requires extensive use of parallel computing across many servers. Thus, a key task remaining for desktop applications is to develop direct approximations to the posterior distributions of both parameter estimates and LMEs. One possibility here is the following two-stage approach. First, by applying standard asymptotic likelihood approximations to the joint posterior distribution of $\hat{\theta}$ (and employing certain extensions of the computational procedures sketched in Section 3.4), it is possible to obtain scalable approximations of this distribution. Second, by employing the Delta method to LMEs (as continuously differentiable functions of θ), it is possible to obtain corresponding scalable approximation of LME posteriors as well. This approach will be developed in detail in a subsequent paper.

A final question relates to *model* uncertainty itself. In the present paper, we have implicitly assumed that all key explanatory variables are known, and that only their relative contributions remain to be determined. However, in a previous paper (Dearmon & Smith, 2016), the GPR model was combined with Bayesian model averaging (BMA) to allow both predictions and LMEs to be averaged over sub-models involving different possible subsets of variables. Such GPR-BMA models are of course even more limited in terms of scalability. But the present GPR-HCA model is directly extendable to this BMA framework, and will be developed in a subsequent paper.

References

Anderson, T.W. (1958) *Introduction to Multivariate Statistical Analysis*, Wiley, New York.

Chen, Jie, Haim Avron, Vikas Sindhwani (2017) “Hierarchically Compositional Kernels for Scalable Nonparametric Learning”, *Journal of Machine Learning Research*, 18:1-42.

Chen, Jie and Michael L. Stein (2017) “Linear-Cost Covariance Functions for Gaussian Random Fields”, *arXiv:1711.05895*.

Cohen, J.P. and J. Zabel (2020) “Local house price diffusion”, *Real Estate Economics*, 48: 710-743.

Daisa, J. M., and T. Parker (2009). Trip Generation rates for urban Infill land uses in California. *ITE Journal*, *79*(6), 30-39.

Datta, A., S. Banerjee, A.O. Finley & A.E. Gelfand (2016) “Hierarchical Nearest-Neighbor Gaussian Process Models for Large Geostatistical Datasets”, *Journal of the American Statistical Association*, 111:514, 800-812.

Dearmon, J. and T.E. Smith (2016) “Gaussian Process Regression and Bayesian Model Averaging: An Alternative Approach to Modeling Spatial Phenomena", *Geographical Analysis*, 48: 82-111

Dearmon, J. and T.E. Smith (2017) “Local Marginal Analysis of Spatial Data: A Gaussian Process Regression Approach with Bayesian Model and Kernel Averaging” *Spatial Econometrics: Qualitative and Limited Dependent Variables*,Vol.37, ed. Batalgie, et al., pp. 297 - 342.

DeFusco, A.; Ding,W.; Ferreira, F.; Gyourko, J. (2018) “The role of price spillovers in the American housing boom”, Journal of Urban Economics, 108: 72–84.

Finley, A.O., D. Abhirup, and S. Banerjee (2020), “spNNGP R package for Nearest Neighbor Gaussian Process models”, *arXiv:2001.09111v1 [stat.CO].*

Hensman, James, Alexander G. Matthews, Maurizio Filippone, and Zoubin Ghahramani. (2015), "MCMC for variationally sparse Gaussian processes." In *Advances in Neural Information Processing Systems*, pp. 1648-1656.

Landis, J. D., Hood, H., Li, G., Rogers, T., & Warren, C. (2006). “The future of infill housing in California: Opportunities, potential, and feasibility”, *Housing Policy Debate*, *17*(4), 681-725.): 681-725.

Long, E., and M. Snead (2019). Oklahoma City Maps Projects Metropolitan Area Projects Economic Impact Study. *https://online.flippingbook.com/view/348632/*

McConnell, V. and Wiley, K., 2010. "Infill development: Perspectives and evidence from economics and planning", *Resources for the Future*, *10*, DP 13, pp.1-34.

Park, Sunho, and Seungjin Choi. "Hierarchical Gaussian process regression." (2010) *Proceedings of 2nd Asian Conference on Machine Learning*, pp. 95-110.

Rasmussen, C. and J. Quinonero-Candela (2005) "A Unifying View of Sparse Approximate Gaussian Process Regression", *Journal of Machine Learning Research*, 6: 1939–1959.

Rasmussen, C., and C. Williams (2006). *Gaussian process for machine learning.* MIT Press.

Ridgeway, G. (2007). Generalized Boosted Models: A guide to the gbm package. *Update*, *1*(1), 2007.

Vanhatalo, J., Riihimäki, J., Hartikainen, J., Jylänki, P., Tolvanen, V., & Vehtari, A. (2013). GPstuff: Bayesian modeling with Gaussian processes. *Journal of Machine Learning Research*, *14*(Apr), 1175-1179

APPENDIX 1

To show that expressions (34) and (37) are indeed the actual covariances of the random vectors in expression (33) of the text, it is convenient to introduce further simplifying notation. For each possible root path, $i_1 \to i_2 \to \cdots \to i_{m-1} \to i_m \to r$, let $H_{i_1 i_2 \cdots i_m r}$ be defined recursively for paths of length one by

(A.1) $H_{i_1 r} = Z_{i_1|r} + A_{i_1 r} Z_r$

[as in (18) of the text] and for longer paths by

(A.2) $H_{i_1 i_2 \cdots i_m r} = Z_{i_1|i_2} + A_{i_1 i_2} H_{i_2 \cdots i_m r}$

Then, (A.1) together with the argument in (14) through (17) of the text again shows that for paths of length one,

(A.3) $\text{cov}(H_{i_1 r}) = K_{i_1 i_1}$

So if it hypothesized that

(A.4) $\text{cov}(H_{i_1 \cdots i_m r}) = K_{i_1 i_1}$

holds for all paths of length *m*, then for paths of length $m+1$ it follows from the independence of $Y_{i_1|i_2}$ and $H_{i_2\cdots i_{m+1}r}$, together with (A.4) that [again from the argument in (14) through (17) in the text],

$$\text{(A.5)}\quad \text{cov}(H_{i_1 i_2\cdots i_m i_{m+1} r}) = \text{cov}(Y_{i_1|i_2} + A_{i_1 i_2} H_{i_2\cdots i_{m+1} r})$$

$$= \text{cov}(Y_{i_1|i_2}) + A_{i_1 i_2}\,\text{cov}(H_{i_2\cdots i_{m+1} r})\,A_{i_2 i_1}$$

$$= K_{i_1 i_1} - K_{i_1 i_2} K_{i_2 i_2}^{-1} K_{i_2 i_1} + (K_{i_1 i_2} K_{i_2 i_2}^{-1})[K_{i_2 i_2}](K_{i_2 i_2}^{-1} K_{i_2 i_1})$$

$$= K_{i_1 i_1} - K_{i_1 i_2} K_{i_2 i_2}^{-1} K_{i_2 i_1} + K_{i_1 i_2} K_{i_2 i_2}^{-1} K_{i_2 i_1}$$

$$= K_{i_1 i_1}$$

So by induction, (A.4) must hold for all *m*. But for any leaf, i, with root path, $i \to i_1 \to \cdots \to i_m \to r$, this implies at once that

$$\text{(A.6)}\quad \text{cov}(H_i) = \text{cov}(H_{i\, i_1\cdots i_m r}) = K_{ii}$$

and thus that expression (39) in the text must hold.

It remains to establish expression (37) in the text for any distinct leaves, i and j with root paths as in (35) and (36) (where again this taken to include the case, $s = r$). To do so, we first expand $H_i = H_{i\, i_1\cdots i_p s h_1\cdots h_m r}$ and $H_j = H_{j\, j_1\cdots j_q s h_1\cdots h_m r}$ as follows:

$$\text{(A.7)}\ H_i = Y_{i|i_1} + A_{i i_1} Y_{i_1|i_2} + (A_{i i_1} A_{i_1 i_2}) Y_{i_2|i_3} + \cdots + (A_{i\, i_1} A_{i_1 i_2} \cdots A_{i_{p-1 p}}) Y_{i_p s} + (A_{i i_1} A_{i_1 i_2} \cdots A_{i_p s}) H_{s h_1\cdots h_m r}$$

$$\text{(A.8)}\ H_j = Y_{j|j_1} + A_{j j_1} Y_{j_1|j_2} + (A_{j j_1} A_{j_1 j_2}) Y_{j_2|j_3} + \cdots + (A_{j j_1} A_{j_1 j_2} \cdots A_{j_{q-1 q}}) Y_{j_q s} + (A_{j j_1} A_{j_1 j_2} \cdots A_{j_q s}) H_{s h_1\cdots h_m r}$$

Next recall that since the random variables $(Y_{i|i_1}, Y_{i_1|i_2}, ..., Y_{i_p|s}, Y_{j|j_1}, Y_{j_1|j_2}, ..., Y_{j_q|s}, H_{s h_1\cdots h_m r})$ are all independent, it follows [as for example in (31) of the text] that all covariance terms between H_i and H_j are zero except for the shared term involving $H_{s h_1\cdots h_m r}$, so that,

$$\text{(A.9)}\quad \text{cov}(H_i, H_j) = \text{cov}[(A_{i i_1} A_{i_1 i_2} \cdots A_{i_p s}) H_{s h_1\cdots h_m r}, (A_{j j_1} A_{j_1 j_2} \cdots A_{j_q s}) H_{s h_1\cdots h_m r}]$$

$$= (A_{i i_1} A_{i_1 i_2} \cdots A_{i_p s})\,\text{cov}(H_{s h_1\cdots h_m r})(A_{s j_q} \cdots A_{j_2 j_1} \cdots A_{j_1 j})$$

But this implies at once from (A.5) that

$$\text{(A.10)} \quad \operatorname{cov}(H_i, H_j) = A_{i i_1} A_{i_1 i_2} \cdots A_{i_p s} (K_{ss}) A_{s j_q} \cdots A_{j_2 j_1} \cdots A_{j_1 j}$$

and thus that expression (37) must hold.