# Global and Local Gaussian Process for Multioutput and Treed Data

Jhouben J. Cuesta(✉), Mauricio A. Álvarez, and Álvaro Á. Orozco

Faculty of Engineering, Universidad Tecnológica de Pereira, Pereira, Colombia
{jycuesta,malvarez,aaog}@utp.edu.co

**Abstract.** We propose a novel Multi-Level Multiple Output Gaussian Process framework for dealing with multivariate and treed data.We define a two-layer hierarchical tree with parent nodes on the upper layer and children nodes on the lower layer in order to represent the interaction between the multiple outputs.Then we compute the Multiple Output Gaussian Process (MGP) covariance matrix as a linear combination of a global multiple output covariance matrix (using the total number of outputs) and a set of local matrices (only using the outputs belonging to each parent node). With this construction of the covariance matrix and the tree we are capable to do interpolation using the MGP framework. To improve the results, we also test different ways of computing the Intrinsic Model of Coregionalization covariance matrix that uses the input space. Results over synthetic data, Motion Capture data and Wireless data shows that the proposed methodology makes a better representation of treed multiple output data.

## 1 Introduction

Gaussian Processes (GP) [7] are widely used for Bayesian regression and classification. Recently, they have been used more often in different disciplines due to its powerful prediction abilities, and the availability of GP implementations in different programming languages. GP provides a framework for non-linear interpolation and uncertainty quantification for single output problems (e.g., modeling the stock exchange), and multiple output problems (e.g, modeling a temperature map over a complete area) [4]. In the latter case, the GP are usually known as multiple output Gaussian processes (MGP). A MGP makes possible to include the correlation of the outputs, improving predictions while maintaining a positive definite covariance matrix.

In this paper, we are interested in modeling multiple-output data with a hierarchical relationship between the outputs (e.g.,the relationship between femur, tibia and the foot in the skeleton) in order to keep improve the predictions by exploiting the hierarchical correlation of the outputs. There are some methodologies that have made predictions for one output treed data, either by dividing the input space and computing classification and regression Trees [9]; making partitions over the input data and defining independent GP for each partition [8], [14]; or putting a prior over the inputs [11].


V. Murino and E. Puppo (Eds.): ICIAP 2015, Part I, LNCS 9279, pp. 161–171, 2015.
DOI: 10.1007/978-3-319-23231-7_15

To the best of our knowledge, the papers [5] [10] are the only two dealing with a multiple output treed data configuration. [10] proposes a Multivariate Bayesian treed Gaussian process to model the cross-covariance function and the nonstationarity of a set of multivariate outputs defining partitions over the input space. In [5] is proposed a multiple output framework for uncertainty quantification based on a construction of a correlation tree using a multi-element method, but assuming a constant relationship between outputs for a fast computing. In this paper, we propose a MGP modification (GLMGP) that includes Global and Local relationship between multiple output data and stores it into a covariance matrix. Despite of needing a prior knowledge of the hierarchy of the data, the proposed model improves the prediction performance and conserves the classical tractability of the GP framework. We made predictions over real hierarchical multiple output data applications: motion capture data-set[1], where the angular position and the hierarchical structure of the bones and a wireless spatial network configuration with sectors located within cells. Furthermore we improve even more the predictions by changing the way of computing the covariance matrix given the application.

This paper is organized as follows: in Section 2 we explain the way to go from a GP passing through a MGP in order to define the proposed GLMGP in Section 2.3. Later in Section 3 we made a comparison of the results of the MGP and the proposed methodology; we first explain the validation measures used to compare the models (Section 3.1), then we compare and analyze the results over simulated and real data. Finally on Section 4 conclusions are made.

## 2 Materials and Methods

In this section, we remark the basics of a GP regression. Later, we explain the multiple output framework with the two common approaches for covariance matrix computation: the linear model of coregionalization and the process convolution. Finally in Section 2.3 we introduce the proposed methodology and the proposed modification to the multiple output approach.

### 2.1 Gaussian Process

A Gaussian Process (GP) is a possible infinite collection of scalar random variables indexed by an input space such that for any finite set of inputs $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, the random variables $\mathbf{f} \triangleq [f(\mathbf{x}_1), f(\mathbf{x}_2), \cdots, f(\mathbf{x}_n)]$ are distributed according to a multivariate Gaussian distribution. A GP is completely specified by a mean function $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ and a covariance function $k_f(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))^\top]$ [12]. This formulation takes the form

$$\mathbf{f}(\mathbf{x}) \sim \mathcal{GP}\,(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')).$$

[1] CMU Graphics Lab Motion Capture Database, available on: http://mocap.cs.cmu.edu/

Without loss of generality, the mean function is assumed to be equal to zero. A covariance function is a positive semi-definite function that measures the similarity between pairs of points over the input space $\mathcal{D}$. Such functions are used to compute the so-called Gram matrix o kernel matrix. Examples of covariance functions are the Squared Exponential (RBF kernel) expressed as

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\ell^2}\right), \tag{1}$$

where $\ell$ corresponds to the length-scale; and the Màtern Class given by:

$$k(r) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}r}{\ell}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}r}{\ell}\right), \tag{2}$$

where $r = ||\mathbf{x} - \mathbf{x}'||$, $\nu$ and $\ell$ are positive parameters and $K_\nu$ is a modified Bessel function as in [7]. Common cases of $\nu$ are $\frac{1}{2}, \frac{3}{2}, \frac{5}{2}$ In the case of a $\nu = 1/2$ and $\mathcal{D} = 1$ this kernel function is called Ornstein Uhlenbeck kernel (OU-kernel).

Using $\mathcal{N}(\mathbf{0}, \mathbf{K})$ as GP prior over the functions $f(\mathbf{x})$ and a likelihood distribution given by $y(\mathbf{x}) = f(\mathbf{x}) + \epsilon$ (where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$) and using Bayes theorem, it is possible to obtain a predictive distribution for a set of new inputs $\mathbf{X}_*$,

$$\mathbf{f}_*|\mathbf{X}, \mathbf{y}, \mathbf{X}_* \sim \mathcal{N}(\bar{\mathbf{f}}_*, \mathrm{cov}(\mathbf{f}_*)), \tag{3}$$

where $\bar{\mathbf{f}}_* \triangleq \mathbb{E}[\mathbf{f}_*|\mathbf{X}, \mathbf{y}, \mathbf{X}_*] = \mathbf{K}(\mathbf{X}_*, \mathbf{X})[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1}$ and the covariance $\mathrm{cov}(\mathbf{f}_*) = \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) - \mathbf{K}(\mathbf{X}_*, \mathbf{X})[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2\mathbf{I}]^{-1}\mathbf{K}(\mathbf{X}, \mathbf{X}_*)$, here $\mathbf{K}(\mathbf{X}, \mathbf{X})$ is the covariance function evaluated on the training set $\mathbf{X}$, $\mathbf{K}(\mathbf{X}_*, \mathbf{X})$ is the covariance of the training and test sets, $\mathbf{K}(\mathbf{X}_*, \mathbf{X}_*)$ is the covariance of the new inputs and the parameter $\sigma_n^2$ represents the variance of the noise.

The estimation of the covariance function parameters is performed by maximizing the log marginal likelihood by a gradient-descent algorithm [6] [13]. The log marginal likelihood is given as in [7]

$$\log p(\mathbf{y}|\mathbf{X}, \phi) = -\frac{1}{2}\mathbf{y}^\top \Sigma^{-1}\mathbf{y} - \frac{1}{2}\log|\Sigma| - \frac{N}{2}D\log(2\pi), \tag{4}$$

where $D$ is the dimension of $\mathbf{x}$, $N$ is the number of training inputs, $\mathbf{y}$ is the vector of outputs corresponding to the total of inputs $\mathbf{X}$, $\phi$ represents the parameters, and $\Sigma = \mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2\mathbf{I}$.

### 2.2 Multiple Output Gaussian Process

The Multiple Output Gaussian Process (MGP) framework starts by defining a set of latent Gaussian processes which are then linearly combined to represent the different outputs, and thus modeling correlations between them. The key point in such framework is the definition of a suitable covariance function, in a mathematical sense, i.e., the covariance function for multiple outputs has to be positive semi-definite. See [3] for a review of different kernel functions for vector-valued data. Once the covariance funcion is defined, the predictive distribution follow a similar form to Equation (3). The parameters of the covariance funtion can be estimated by maximizing a marginal log-likelihood similar to (4).

In what follows, we briefly review the linear model of corregionalization, which is a common choice to build valid covariance functions for multiple outputs [2], and that we use for building the multi-level multi-output covariance function.

**Linear Model of Coregionalization:** In the linear model of coregionalization (LMC) the covariance function is formed by a sum of separable kernels. Under this LMC assumption, the outputs are expressed as linear combinations of independent random functions, ensuring a valid positive semi-definite covariance matrix [3]. Over a set of outputs $\{f_d(\mathbf{x})\}_{d=1}^{D}$ with $\mathbf{x} \in \mathbb{R}^p$, each component $f_d$ is expressed as

$$f_d(\mathbf{x}) = \sum_{q=1}^{Q}\sum_{i=1}^{R_q} a_{d,q}^i u_q^i(\mathbf{x}),$$

where $Q$ represents the groups of latent functions $u_q^i(\mathbf{x})$ and $R_q$ are represents the number of functions in a group that share the same covariance; and the functions $u_q^i(\mathbf{x})$, with $q = 1, ..., Q$ and $i = 1, ..., R_q$ have mean equal to zero and covariance $\mathrm{cov}[u_q^i(\mathbf{x}), u_{q'}^{i'}(\mathbf{x}')] = k_q(\mathbf{x}, \mathbf{x}')$ if $q = q'$ and $i = i'$. The cross-covariance between any two functions $f_d(\mathbf{x})$ and $f_d'(\mathbf{x}')$ is given in terms of the covariance functions for $u_q^i(\mathbf{x})$

$$\mathrm{cov}[f_d(\mathbf{x}), f_{d'}(\mathbf{x}')] = \sum_{q=1}^{Q}\sum_{q'=1}^{Q}\sum_{i=1}^{R_q}\sum_{i'=1}^{R_q} a_{d,q}^i a_{d',q'}^{i'} \mathrm{cov}[u_q^i(\mathbf{x}), u_{q'}^{i'}(\mathbf{x}')].$$

Due to independence of the latent functions, the kernel matrix can now be expressed as

$$\mathbf{K}(\mathbf{x}, \mathbf{x}') = \sum_{q=1}^{Q} \mathbf{B}_q k_q(\mathbf{x}, \mathbf{x}'), \tag{5}$$

where $Q$ represents the number of latent functions, each $\mathbf{B}_q \in \mathbb{R}^{D\times D}$ is known as a *coregionalization matrix* and the rank $R_q$ of each matrix $\mathbf{B}_q$ is determined by the number of latent functions that share the same covariance function.

When $Q = 1$ in Eq. (5), the LMC approach is known as the Intrinsic Coregionalization Model (ICM). The kernel matrix for multiple outputs becomes $\mathbf{K}(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}')\mathbf{B}$ [1], and for an entire data set $\mathbf{X}$ takes the form

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{B} \otimes k(\mathbf{X}, \mathbf{X}), \tag{6}$$

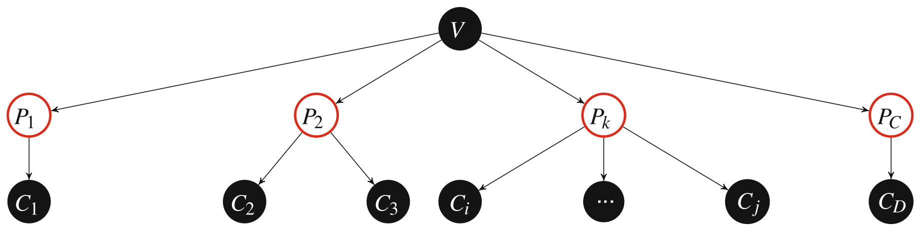


**Fig. 1.** Representation of a Global-Local or Parent-Child treed structure. $V$ represents the variable of interest. The red nodes represent the $C$ parents and the lower $D$ nodes represent the total number of children (equal to the total number of outputs of a single MGP approach).

where the operator $\otimes$ represents the Kronecker product. There are different ways in which $\mathbf{B}$ can be parameterized. One of them is using a Cholesky decomposition, $\mathbf{B} = \mathbf{L}\mathbf{L}^\top$, or using a kernel matrix computed from a valid kernel function, like the ones in Eqs. (1) and (2).

### 2.3 Global and Local Multi Output Treed GP

In this paper, we propose the Global and Local Treed Multiple Output Gaussian Process (GLMGP) as a multiple output GP that computes the correlations of multiple-output data with a parent-child- type of configuration.

We first define the tree $T$ as a vector of parent indexes for every child output. With this information we compute a global covariance matrix $\mathbf{K}_g$ defined as a Multi GP covariance for all the outputs (using, for example, the LMC) and later we compute the contribution of the children as a block-diagonal matrix formed by a set of $C$ local matrices $\mathbf{K}_\ell^i$ (where $C$ is equal to the number of parents of the global layer and $i = 1, ..., C$). Each local matrix $\mathbf{K}_\ell^i$ is computed again as a multiple output covariance matrix, but in this computation the outputs are equivalent to the group of children associated to the $i$th parent (again this covariance is computed using, for example, LMC). The resulting covariance matrix $\mathbf{K}$ takes the form

$$\mathbf{K} = \mathbf{K}_g + \text{blockdiag}(\{\mathbf{K}_\ell^i\}_{i=1}^C),$$

where $C$ is the number of parents, and each matrix $\mathbf{K}_g$ and $\mathbf{K}_\ell$ is computed as in Equations 5 or 6.

The Figure 1 shows a general treed structure with a two layer (parent-child) representation. A real-life example of a tree structured data is Mocap data. Here we have measures of the spatial position of different body parts while a subject is performing a motion. Here, the tree is represented by the body that has a hierarchical inner structure i.e the femur, tibia and the foot of the left leg. We can exploit a parent-children relation where the femur is the parent and the tibia and the foot that will include information of the skeletal structure. Now, the position data of the leg can be interpreted as multiple output because of the influence of the femur, tibia and foot on the motion of a subject.

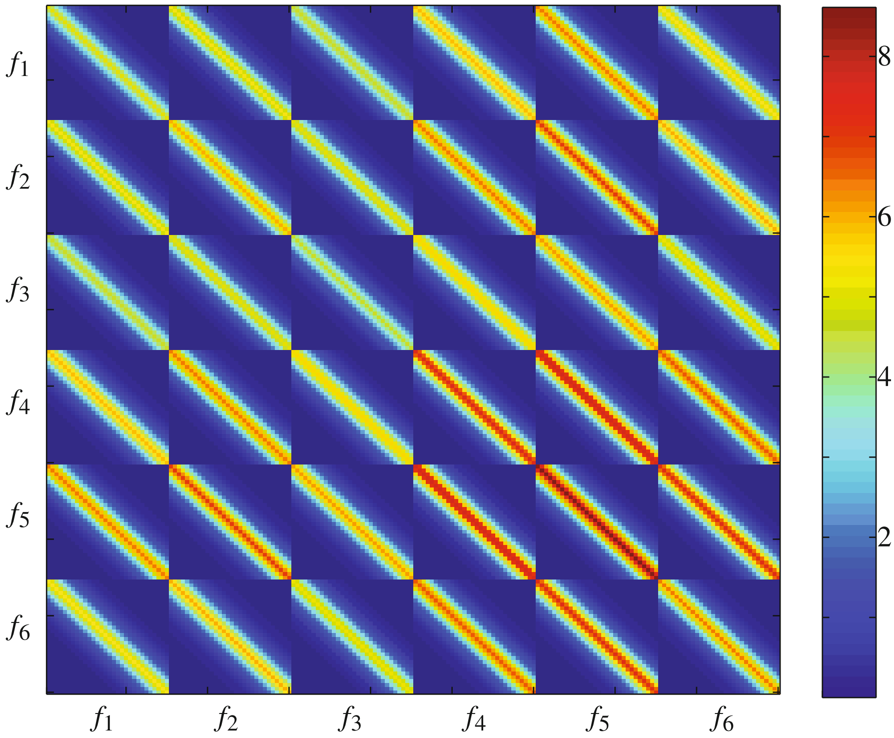


**Fig. 2.** MultiGP Covariance Matrix Block Structure for a set of 6 outputs $f$

Figures 2 and 3 show both a MGP and GLMGP covariance matrices for a configuration of six outputs and the same six outputs plus a tree structure conformed by three parents for the GLMGP (these two configuration will be explained in detail on Section 3.2). We see that our proposed covariance represents the children contribution as a non uniform increment of the values of the diagonal block. The increment value depends on the contribution that the children do to each parent. The proposed modification of the covariance matrix is flexible in the way that we can define different covariance functions, number of latent functions or rank for the coregionalization matrices for each part of the tree, this means we can define a different configuration per each Multi-GP covariance matrix to be computed. Asides this modification does not represent any change on the marginal likelihood expressed before in Equation 4.

In the proposed methodology, we compute the covariance matrices using the ICM. The ICM matrix was computed using two approaches: using a Cholesky decomposition (called ICM-L) and a covariance function (called ICM-K). The use of ICM-L or ICM-K depends on the context and will be explained further in the results on Sec. 3.

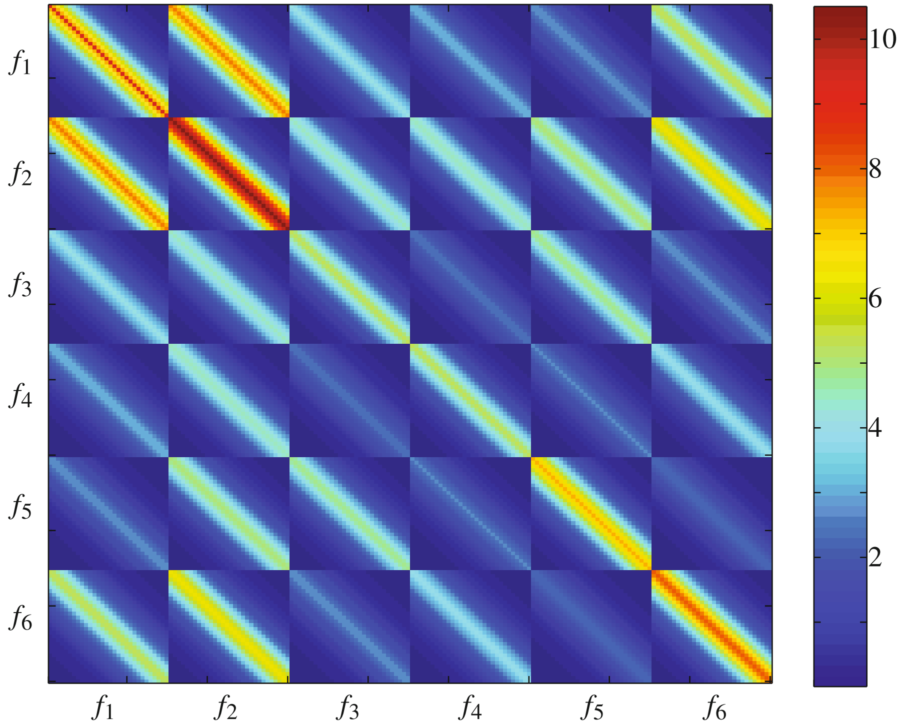


**Fig. 3.** GMLGP Covariance Matrix Block Structure for a set of 6 outputs $f$

# 3 Results

In this section we compare the results obtained after comparing the MGP LMC methods against the GLMGP ICM-L and ICM-K over synthetic and real data. First in Section 3.1 we describe the Validation and Error Measures. Later we report the results over synthetic and real data in Sections 3.2 and 3.3 respectively. All the algorithms were trained using 70% of the data and validated with remaining 30%. We used the the kernels OU, Squared Exponential and Màtern 3/2 covariance functions as expressed on Equations 1 and 2 on section 2.2. We repeated the experiment 10 times in order to report a standard deviation of the error measures.

## 3.1 Validation and Error Measures

The proposed GLMGP methodology and the MGP LMC were compared using the Standardized Mean Square Error (SMSE) and the Mean Standardized Log Loss (MSLL) measures [7] via hold-out validation (see [6]) and computed as in [2]. It is important to remark that a lower SMSE error implies a better interpolator and a lower negative MSLL implies that the model is more adequate to the data.

### 3.2 Results over Simulated Data

We generate a first synthetic data-set (Sy1) from a Multiple Output Configuration using a MGP model (composed by a one-dimensional input, 6 outputs, 30 points per output, a màtern 3/2 kernel, $Q = 1$ latent functions and a rank of coregionalization matrix of $R_q = 1$). We generate a second synthetic data-set (Sy2) from a Global Local Treed Configuration model with the same 6 outputs, a hierarchical tree $T = [1, 1, 1, 2, 2, 3]$ (this tree means that there are 3 nodes in the Global layer and the first node have 3 outputs associated, the second node have 2 and the third node just 1 output), a $Q = 1$ a màtern 3/2 kernel for the Global layer and [OU,OU,màtern32] respectively to each group of outputs on the Local layer. The results over Sy1 and Sy2 are summarized in tables 1 and 2. The first table shows that on the MGP data the GLMGP is as better interpolator and model than MGP, but in the second table we see that the proposed GLMGP performs and model better the tree structured data. It is important to remark that both methods were trained using just a màtern32 kernel and using $Q = 1$ and $R_q = 1$.

**Table 1.** Results Over Sy1 Data. Methods with a $*$ used both the same training points

| Model | SMSE | MSLL |
|---|---|---|
| MGP LMC | 0.2185 ±0.0813 | -1.0499 ±0.0839 |
| GLMGP ICM-L | 0.2192 ±0.0878 | -1.0335 ±0.1132 |
| MGP LMC* | 0.1208 ±0.0176 | -1.1398 ±0.1046 |
| **GLMGP IMC-L*** | **0.1173±0.0183** | **-1.1406±0.1310** |

**Table 2.** Results Over Sy2 treed Data. Methods with a $*$ used both the same training points

| Model | SMSE | MSLL |
|---|---|---|
| MGP LMC | 0.2156 ±0.0357 | -1.3623 ±0.1061 |
| GLMGP ICM-L | 0.1245 ±0.0302 | -1.4918 ±0.0916 |
| MGP LMC* | 0.1480 ±0.0419 | -1.4175 ±0.0948 |
| **GLMGP ICM-L*** | **0.0881±0.0143** | **-1.6910±0.0718** |

### 3.3 Results over Real Data

In this section we show examples of interpolation for real data. Firstly we use Mocap data-set (Online Available http://mocap.cs.cmu.edu/) as a time series regression example. Later a Colombian Network Wireless data is used for a spatio-temporal interpolation (this database is not available due to copyright).

**Mocap Dataset.** For Mocap data we worked with the Subject eight - trial two of a walk motion. We selected the bone structures of the right leg and left leg as Global layer parents; and the femur, tibia, foot and toes as the children for each parent (with reference to Figure 1). The time interval $t$ of the motion was taken as the input $\mathbf{X}$ while the angles $\theta_x, \theta_y, \theta_z$ were taken as the outputs. There was a total of 8 outputs because we remove 4 angles that had no significant variation ($1e-7$) and a tree $T = [1,1,1,1,2,2,2,2]$. With this data we did interpolation using the MGP LMC and the GLMGP. Table 3 shows the results of the MGP against the two proposed methodologies. We see that GLMGP ICM is slightly better than MGP on SMSE, but has a better interpretation for Mocap treed data. In this one dimensional case we used the Cholesky decomposition for computing the coregionalization matrix (ICM-L) instead of a covariance function (ICM-K). This is done because because of using ICM-K was not ensuring a positive definite covariance matrix.

**Table 3.** Results over Mocap Data

| Model | SMSE | MSLL |
|---|---|---|
| MGP ICM | 0.2880 ±0.0186 | -0.8839 ±0.0551 |
| **GLMGP ICM-L** | **0.2533±0.0166** | **-1.2894±0.0780** |

**Wireless Treed Data.** The Wireless data used in this paper is conformed by 30 daily measurements of a Traffic Key Performance Indicator (KPI) of a network of 32 sectors placed in 11 different spatially located cells. Using the information of the cell-sector relationship we treated the cells as a parent and the sectors as the children and defined the tree as $T = [1,1,2,2,2,3,3,3,\cdots,11,11,11]$. We take the spatial coordinates $[x_{x_i}, x_{y_i}]$ and the day $t_i$ as the inputs, and the KPI value as the output $y$. We tested different configurations of MGP LMC and GLMGP and reported the best results on Table 4. The best LMC model was a configuration of a màtern32 kernel with $Q = 2$ and $R_q = 1$; the best ICM-L was a configuration of màtern32 for Global layer and OU for local layer with $R_q = 1$ for all the MGP covariances computed. The best ICM-K model had the same kernel configuration of ICM-L. Despite of this we see on the results table that ICM-K improves the results considerably, even for the SMSE.

**Table 4.** Results over Wireless Treed (Cell-Sector) Data.

| Model | SMSE | MSLL |
|---|---|---|
| MGP LMC | 0.2885 ±0.0390 | -0.9147 ±0.0789 |
| GLMGP ICM-L | 0.4744 ±0.2720 | -0.6850 ±0.4064 |
| **GLMGP ICM-K** | **0.1602±0.0149** | **-1.1726±0.3011** |

## 4 Conclusions

We have presented GLMGP as a model that takes into account a parent-child relationship between data outputs and represents it in a covariance matrix. Instead of modifying the prior, we defined a tree with global indexes over the inputs that remains unchanged the tractability of the model. This model proved to be better than the MGP in capturing the information and interpretation of the structured data . Despite of the fact that the model can not learn or define a proper tree by its own, it is very useful in applications when we know the interaction of the output variables previously like the cell-sector relationship, skeletal structures of the body, etc.

We also tested two ways of computing the coregionalization matrix in order make a more flexible model the ICM-L and the ICM-K. In the case of the ICM-L it had a good performance for synthetic and mocap data and the ICM-K performed better on the wireless data-set. In the future, we expect to improve this model in order to include more than a Global and Local layers and also a modification that can estimate a tree structure that improves the interpolation results. In addition, for the one-dimensional case, we expect to find a parametrization of the outputs under the ICM-K framework to ensure a positive definite covariance matrix.

**Acknowledgments.** JJC thanks to the research project *Análisis de Fallas en Redes de Telefonía Móvil usando reconocimiento estadístico de patrones* in the program of Jóvenes Investigadores "Virginia Gutiérrez de Pineda". JJC and MAA thank to Universidad Tecnológica de Pereira for the support under the research projects *Procesos Gaussianos Jerárquicos de Múltiples Salidas Para el Análisis y Expansión de Redes de Telefonía Móvil*, and the research project *Human motion synthesis through physically-inspired machine learning models*, with code $6 - 15 - 3$. The authors would also like to acknowledge the project *Estimación de los parámetros de neuro modulación con terapia de estimulación cerebral profunda en pacientes con enfermedad de Parkinson a partir del volumen de tejido activo planeado*, financed by Colciencias with code $1110 - 657 - 40687$.

## References

1. Álvarez, M.A.: Convolved Gaussian Process Priors for Multivariate Regression With Applications to Dynamical Systems. Ph.D. thesis, School of Computer Science, University of Manchester (2011)
2. Álvarez, M.A., Lawrence, N.D.: Computationally efficient convolved multiple output Gaussian processes. Journal of Machine Learning Research **12**, 39 (2011)
3. Álvarez, M.A., Rosasco, L., Lawrence, N.: Kernels for Vector-Valued Functions: A Review. Now Publishers Incorporated (2012). http://books.google.com.co/books?id=2A5hLwEACAAJ
4. Álvarez, M.A., Lawrence, N.D.: Sparse convolved multiple output gaussian processes for multi-outptut regression. In: NIPS 2008, pp. 57–64 (2009)

5. Bilionis, I., Zabaras, N.: Multi-output local gaussian process regression: Applications to uncertainty quantification. J. Comput. Physics **231**(17), 5718–5746 (2012). http://dblp.uni-trier.de/db/journals/jcphy/jcphy231.html#BilionisZ12
6. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer (2007). http://www.openisbn.com/isbn/9780387310732/
7. Carl Edwards Rasmussen, C.W.: Gaussian Processes for Machine Learning. The MIT Press (2006)
8. Fox, E.B., Dunson, D.B.: Multiresolution gaussian processes. In: Bartlett, P.L., Pereira, F.C.N., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) NIPS, pp. 746–754 (2012). http://dblp.uni-trier.de/db/conf/nips/nips2012.html#FoxD12
9. Gramacy, R.B., Lee, H.K.H.: Bayesian treed gaussian process models with an application to computer modeling. Journal of the American Statistical Association (2007)
10. Konomi, B., Karagiannis, G., Sarkar, A., Sun, X., Lin, G.: Bayesian treed multivariate gaussian process with adaptive design: Application to a carbon capture unit. Technometrics, 145–158, May 2014
11. Lawrence, N.D., Moore, A.J.: Hierarchical gaussian process latent variable models. In: Proceedings of the 24th International Conference on Machine Learning, pp. 481–488 (2007)
12. Liutkus, G., Badeau, R.: Multi-dimensional signal separation with gaussian processes. In: IEEE Statistical Signal Processing Workshop (SSP), pp. 401–404 (2011)
13. Murphy, K.P.: Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning Series). The MIT Press (2012). http://www.openisbn.com/isbn/9780262018029/
14. Park, S., Choi, S.: Hierarchical gaussian process regression. In: JMLR: Workshop and Conference Proceedings 13, pp. 95–110 (2010)