Gaussian Process-based Spatio-Temporal Predictor

Balázs Varga

Department of Control for Transportation and Vehicle Systems, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, Műegyetem rkp. 3, H-1111 Budapest, Hungary
e-mail: varga.balazs@kjk.bme.hu

Abstract: This paper presents a grid-based algorithm using Gaussian Processes to predict outputs using spatially and temporally dependent data. First, independent Gaussian Processes are formulated along space and time axes. Then, these processes are coupled with a common noise in the covariance kernel. This common noise acts as a smoothing parameter, trading off accuracy at knots for extrapolation capabilities. The algorithm can predict time-series at unmeasured locations. The efficiency of the algorithm is demonstrated in a traffic flow prediction problem. Results suggest that applying a common additive noise term capturing cross covariances improves prediction accuracy when extrapolating outside the dataset.

Keywords: Gaussian Process; Spatio-temporal prediction; Traffic flow prediction

1 Introduction

A Gaussian Process (GP) is a collection of random variables of which have (consistent) joint Gaussian distributions. A GP is fully specified by its mean function and a covariance kernel. The main application of GPs is function approximation. Thus, Gaussian Processes can be interpreted as distributions over functions [1].

The general framework of Gaussian Processes cannot handle vector-valued functions, i.e., it is not trivial how to formulate cross-covariances between the outputs. Within machine learning, handling multiple inputs and outputs is vital. Neural networks have this feature by default [2]. On the other hand, kernel methods require some extensions in order to handle multiple outputs.

The most important features of multivariate GPs are the cross-covariances between the outputs: they describe how the output processes are related to each other. Dependency can be tackled in several ways. For example, [3] employs shared dependency on a latent white noise process convolved with smoothing kernels.

In [4], the sum of separable kernels is employed to tackle dependent processes. [2] provides a comprehensive overview of how multivariate kernels can be constructed for GPs. In addition, multivariate GPs can be extended to handle heterogeneous data sources akin to neural networks [5] or constraints [6].

GPs are commonly used for both spatial and temporal function approximations. For example, a specialized version of GP in geostatistics called Kriging is commonly used. Kriging handles the mean separately as a generalized least squares estimate and variograms the common choice for kernels [7]. In spatial estimation, GPs can be used for meteorological forecasts [8, 9], sensor placement [10], or predicting traffic flow at unmeasured sites [11]. GPs are often used for time-series forecasting [12, 13]. Some specific applications involve fatigue analysis [14], wind speed forecasting [15], or economic predictions [16, 17]. From the above list, it is clear that GPs (despite some of their limitations) have powerful function approximation capabilities for a vast range of applications.

One area where GPs fall short is spatio-temporal approximation. The challenge in predicting both spatially and temporally correlated data is modeling covariance between space and time dimensions. The distance between a point in a Euclidean space and a time instant cannot be exactly formulated using kernels: they are different domains. [18] uses the product of spatial and temporal kernels for neuroimaging. Spatio-temporal GPs are often handled in a state-space setting, predicting real-time system dynamics [19, 20].

This manuscript presents a grid-based spatio-temporal Gaussian Process function approximator. Instead of computing the hyperparameters together, independent GPs are formulated along space and time axes. Then, the independent processes are coupled with a noise term in the kernel. The algorithm can predict time-series modeled as Gaussian Processes at unmeasured locations demonstrated through an urban road traffic example. The main benefit of the proposed algorithm is its simplicity: the approximator is achieved through sequential GP fitting.

The paper is organized as follows. Section 2 details the formulation of the predictor and the prediction procedure. Then, in Section 3, the algorithm is demonstrated through a traffic flow prediction example. Finally, the findings of the paper are summarized in the Conclusions section.

2 Spatio-Temporal Predictor

2.1 Data

Consider a dataset $S = \big(X, T, Y(X,T)\big)$, with $X = [x_i]_{i=1}^{N} \in \mathbb{R}^{N\times n}$ being the design matrix of locations in the $\mathbb{R}^n$ dimensional Euclidean space. $T \in \mathbb{R}^{1\times k}$ is the set of discrete-time samples, and $Y(X,T) \in \mathbb{R}^{N\times k}$ is the vector-valued measurement data. One row of $Y(X,T)$, $Y(x_i,T)$ represents one time series at location $x_i \in X$. Each realization is assumed to be evenly sampled in time, and k discrete time-steps long. Similarly, $Y\big(X,t_p\big)$ denotes a spatial prediction at fixed time instant $t_p \in T$: $p = 1,2,\dots,k$.

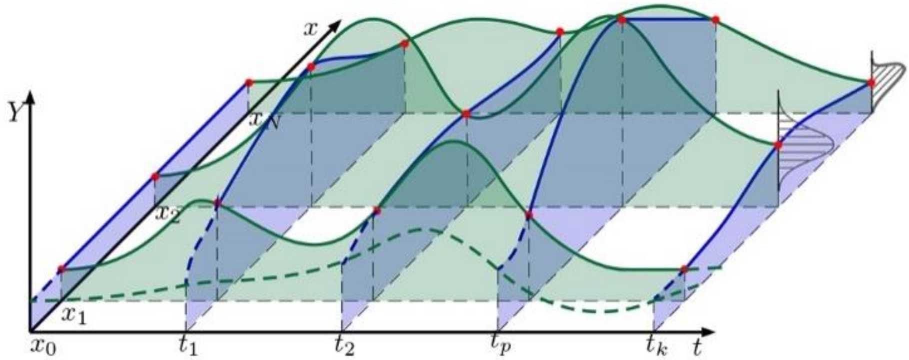

Figure 1

Predicting space x and time t dependent outputs Y via multiple GPs. Data is denoted with dots and estimates are denoted with dashed lines.

2.2 Spatial Gaussian Processes

The first goal is approximating the underlying target functions $f_p\big(x,t_p\big)$ based on data points $Y\big(X,t_p\big)$ for every $p \in 1,2,\dots,k$ time instant. Consider k independent Gaussian Processes

$$f_p\big(x,t_p\big) \sim \mathcal{GP}\left(m_p(x), k_p(x,x')\right), \tag{1}$$

with $m_p(x) \in \mathbb{R}^{N\times 1}$ being the mean function, and $k_p(x,x')$ being associated the covariance function, i.e.,

$$m_p(x) = E\left(f_p\big(x,t_p\big)\right), \tag{2}$$

$$k_p(x,x') = E\left(f_p\big(x,t_p\big) - m_p(x)\right)\left(f_p\big(x',t_p\big) - m_p(x')\right). \tag{3}$$

For the sake of simplicity, assume $m_p(x) = 0$, i.e., the process is first-order stationary. Besides, GPs are insensitive to the mean; defining the mean function is not that important [21].

The chosen kernel to model covariance is a squared exponential kernel with an additive independent identically distributed Gaussian noise term

$$k_p(x, x') = \sigma_p^2 e^{-\frac{1}{2\theta_p^2}(x-x')^T(x-x')} + a_p^2 \delta_{x,x'}, \quad (4)$$

where $\Theta_p = [\sigma_p, \theta_p, a_p]$ are the hyperparameters of the p^{th} GP model. $\delta_{x,x'}$ is the Kronecker delta. σ_p scales the variance of the function between knots (design points), θ_p is the length-scale parameter, calibrates the smoothness of the function, i.e., a large length-scale means a slowly changing function and more reliable extrapolations. Finally, the noise variance term a_p scales the overall uncertainty of the estimate. This parameter helps to reduce overfitting [1].

Hyperparameters are commonly learned by minimizing the negative marginal log-likelihood (NMLL) $-log\left(P\left(Y\left(X, t_p\right)\middle|X, \Theta_p\right)\right)$, which can be given as

$$\min_{\Theta_p} \frac{1}{2}\left(\left(Y\left(X, t_p\right) - m_p(X)\right)^T K_p(X, X)^{-1} \left(Y\left(X, t_p\right) - m_p(X)\right) + log\left|K_p(X, X)\right| + klog(2\pi)\right), \quad (5)$$

where $K_p(X, X) = \left[k_p\left(x_i, x_j\right)\right]_{i,j}, \ i, j = 1,2, \dots, N$ is the covariance matrix.

2.3 Spatial Gaussian Processes

In the same vein, GPs can be constructed for the time-series at every location. Define a temporal GP as

$$f_i(x_i, t) \sim \mathcal{GP}\left(m_i(t), k_i(t, t')\right) \quad (6)$$

for each location i. Similarly, assume zero means $m_i(t) = 0$ and squared exponential kernels with i.i.d. Gaussian noise:

$$k_i(t, t') = \sigma_i^2 e^{-\frac{1}{2\theta_i^2}(t-t')^2} + a_i^2 \delta_{t,t'}, \quad (7)$$

with hyperparameters $\Theta_i = [\sigma_i,\ \theta_i,\ a_i]$. The hyperparameters can be obtained in the same fashion as for the spatial prediction: minimizing the negative marginal log-likelihood $-log\left(P\left(Y(x_i, T)\middle|X, \Theta_p\right)\right)$ as

$$\min_{\Theta_i} \frac{1}{2}\left(\left(Y(x_i, T) - m_i(T)\right)^T K_i(T, T)^{-1}\left(Y(x_i, T) - m_i(T)\right) + log|K_i(T, T)| + klog(2\pi)\right), \quad (8)$$

with $K_i(T,T) = \left[k_i\left(t_\alpha, t_\beta\right)\right]_{\alpha,\beta}$, $\alpha, \beta = 1,2,\ldots,k$ being the temporal covariance matrix at location i [22].

2.4 Common Noise

After solving the nonlinear optimizations in Eq. (5) and Eq. (8) k spatial predictions (for each time step) and N time-series predictions are obtained as independent zero-mean GPs are obtained. Next, connect these processes with an additional i.i.d. Gaussian noise a_c. This common noise does not depend on spatial nor temporal distances, thus can be applied to both spatial and temporal predictions. In addition, it does not affect the structure of the kernel, i.e., a squared exponential kernel with an additive noise. Since every GP approximates functions based on data from $Y(X,T)$, it can be assumed that approximated function outputs are similar in magnitude too. Thus, it can be assumed that this noise will not corrupt some independent functions too much.

The independent noise scales the uncertainty of every GP, reducing their overfitting, resulting in better extrapolation performance at the cost of worse fit on the training data.

For every kernel $k_p(x,x')$ and $k_i(t,t')$ add a_c, i.e.,

$$K_p^*(X,X) = \left[k_p^*\left(x_i, x_j\right)\right]_{i,j} = \left[k_p\left(x_i, x_j\right) + a_c \delta_{x_i x_j}\right]_{i,j}, \tag{9}$$

$$K_i^*(T,T) = \left[k_i^*\left(t_\alpha, t_\beta\right)\right]_{\alpha,\beta} = \left[k_i\left(t_\alpha, t_\beta\right) + a_c \delta_{\alpha,\beta}\right]_{\alpha,\beta}, \tag{10}$$

and $i,j = 1,2,\ldots,N$; $\alpha, \beta, p = 1,2,\ldots,k$. With fixed hyperparameters Θ_p, Θ_i perform the following nonlinear optimization:

$$\min_{a_c} \sum_{p=1}^{k} \frac{1}{2}\Bigg(\left(Y\left(X,t_p\right) - m_p(X)\right)^T K_p^*(X,X)^{-1}\left(Y\left(X,t_p\right) - m_p(X)\right) + \log\left|K_p^*(X,X)\right| + N\log(2\pi)\Bigg) + \sum_{i=1}^{N} \frac{1}{2}\Big(\left(Y(x_i,T) - m_i(T)\right)^T K_i^*(T,T)^{-1}\left(Y(x_i,T) - m_i(T)\right) + \log\left|K_i^*(T,T)\right| + k\log(2\pi)\Big), \tag{11}$$

i.e., minimize the sum of negative marginal log-likelihoods with respect to a_c for every GP. In the proposed multi-step approach, it would make sense to omit a_p and a_i noises from the initial kernels and find one common noise for the whole dataset. On the other hand, time series come from different sensors. Thus, their noise content might be different. In addition, simulation results suggest that without these terms, predictions become significantly worse.

2.5 Prediction

Once the kernels corrupted with the common noise are constructed, both spatial and temporal predictions can be performed. One practical application is estimating time-series at an unknown location x_0. First, perform spatial predictions for every timestep $t_p \in T$ using the respective spatial GP.

Mathematically, the posterior mean $Y^*(x_0, t_p)$ and variance $\sigma^{2*}(x_0, t_p)$ are given as follows:

$$Y^*(x_0, t_p) = k_p^*(x_0, X)K_p^*(X, X)Y(X, t_p), \tag{12}$$

$$\sigma^{2*}(x_0, t_p) = k_p^*(x_0, x_0) + k_p^*(x_0, X)K_p^*(X, X)k_p^*(x_0, X)^T. \tag{13}$$

Next, connect the predicted points $[Y^*(x_0, t_p),\ \sigma^{2*}(x_0, t_p)]_{p=1}^{k}$ with a zero-mean ($m_0(x) = 0$) GP as

$$f_0(x_0, t) \sim \mathcal{GP}(m_0(x), k_0(t, t')). \tag{14}$$

Note that at every predicted point, there is a corresponding variance too. Thus, fixed variance is assumed at the knots, which can be incorporated in the kernel $k_0(t, t')$ with the help of a Kronecker delta: $\sigma^{2*}(x_0, t_p)\delta_{t,t'}$, i.e., the variance at x_0 at time t_p will be $\sigma^{2*}(x_0, t_p)$ from Eq (13). This term imposes a smoothing effect on the posterior GP, considering the uncertainty of the independent predictions. The covariance kernel is defined as follows.

$$k_0(t, t') = \sigma_0^2 e^{-\frac{1}{2\theta_0^2}(t-t')^2} + \sigma^{2*}(x_0, t_p)\delta_{t,t'}. \tag{15}$$

The remaining hyperparameters are $\Theta_0 = [\sigma_0, \theta_0]$. Learning them can be done via substituting $Y^*(x_0, T)$ into Eq. (8).

The above steps can be employed to estimate spatial dependency at an arbitrary time instant t_0 too. In that case, temporal predictions shall be made at every location to the same time instant t_0, and then the location-wise predictions shall relate to a new Gaussian Process.

The discussed Gaussian Process-based spatio-temporal function approximator algorithm is summarized in Algorithm 1 and its usage for prediction in Algorithm 2.

Algorithm 1

Gaussian Process-based spatio-temporal function predictor

Inputs: $X, T, Y(X, T), x_0, t_0$
Outputs: ${\Theta_p}_{p=1}^{k}, {\Theta_i}_{i=1}^{N}, a_c$

for $p = 1,2, \dots, k$ **do**
    Construct spatial GPs $f_p(x, t_p)$.
    Find hyperparameters Θ_p.
end
for $i = 1,2, \dots, N$ **do**
    Construct temporal GPs $f_i(x_i, t)$.
    Find hyperparameters Θ_i.
end
Add a common noise a_c to every kernel $K_p(X, X)_{p=1}^{k}$ and $K_i(T, T)_{i=1}^{N}$.
Optimize sum of NMLLs w.r.t. a_c.

Algorithm 2

Gaussian Process-based spatio-temporal prediction

Inputs: $X, T, Y(X, T), x_0, t_0, [\Theta_p], [\Theta_i]_{i=1}^{N}, a_c$
Outputs: $Y^*(x_0, t_0), \sigma^{2*}(x_0, t_0)$

for $p = 1,2, \dots, k$ **do**
    Predict posteriors $[Y^*(x_0, t_p)]_{p=1}^{k}$ and $\sigma^{2*}(x_0, t_p)\}_{p=1}^{k}$.
end
Construct a temporal GP $f_0(x_0, t)$ with $[Y^*(x_0, t_p), \sigma^{2*}(x_0, t_p)]_{p=1}^{k}$.
Find hyperparameters Θ_0.
Predict $Y^*(x_0, t_0)$ and $\sigma^{2*}(x_0, t_0)$ with $f_0(x_0, t)$.

3 Numerical Example

This section presents a numerical example to demonstrate the proposed algorithm. The input data is a set of one day-long traffic flow log with hourly sampling (in vehicles/hour) from 20 locations in Turin, Italy (Figure 2). Detector locations are not given with their geographical positions because it does not accurately describe their spatial dependency. For example, a detector is likely to be more correlated to another a few hundred meters downstream than to one that is next to it but measuring traffic in the opposite direction. Instead, the spatial dependency is given in a higher-dimensional Euclidean space. It is obtained by transforming an arbitrary similarity measure into a higher dimensional feature space via multidimensional scaling [23] as in [24].

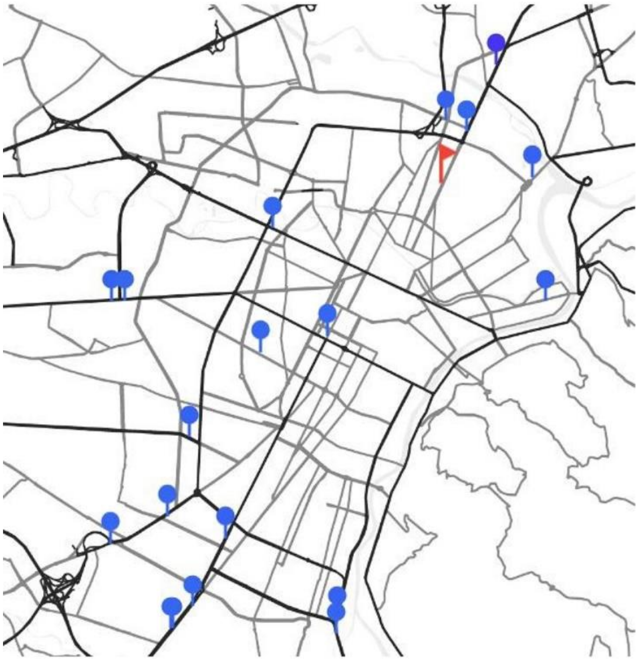

Figure 2
Detector locations in Turin, Italy. GPS coordinates of the predicted road section: *lat: 45.095, lon: 7.6954*. Source: https://snazzymaps.com

The goal is to estimate the hourly traffic flow for one day (and also give short-term prediction) based on the logs of the other 20 traffic flow detectors.

First, with the proposed algorithm, 24 independent spatial GPs are constructed (one for each hour), describing the functional relationship between detector locations in the Euclidean space. Then, temporal GPs are fitted on the traffic logs at every traffic flow detector location. Finally, the common noise parameter is computed. Figure 3 and Figure 4 depict the results of the approximated traffic flows at each location at a given hour without and with the common noise parameter, respectively. Similarly, one day's traffic flow at a fixed location is given in Figure 5 and Figure 6. Results suggest that both the independent and the dependent GPs can predict the traffic flow accurately. However, the independent GPs perform better in terms of variance and Root Mean Square Error (RMSE). In the spatial case, the RMSE is 30.47% higher when considering a_c (in that particular time instant). It is slightly better for the spatial case: the dependent GP has only 14.54% higher RMSE. This result is expected: the independent GPs were trained for the approximated data. In contrast, an extra noise term will generalize the predictor for the whole dataset. When predicting outside of the dataset, better results are expected with the dependent solution.

Next, focus on the unmeasured location x_0. The aim is to estimate the hourly traffic flow at that location with a technique that can predict future traffic flow at unmeasured (intermittent or future) time instants. The input is the one-day-long traffic logs at every measured site. Traffic flow is estimated at location x_0 with three methods: i) using independent spatial GPs (Figure 7), ii) using spatial GPs with the same additive noise a_c (Figure 8), and iii) by fitting a temporal GP on the predicted data (Figure 9).

In every case, spatial GPs are formulated every hour and predictions are made for the traffic flow at x_0. When extrapolating with the independent GPs (because of their overfitting) the variance of the prediction becomes extremely high (Figure 7). On the other hand, the mean is accurately found with a relatively small RMSE. Series of spatial predictions with the common noise reduce the process variance significantly. In addition, the RMSE is reduced slightly, by 2.31%. This result validates the benefit of smoothing the independent GPs with an additive noise term.

To be able to predict both spatially and temporally outside the dataset (i.e., an arbitrary time instant t_0 at location x_0), a new Gaussian Process is fit on the predicted data points in Figure 8. This GP imposes additional smoothing on the predicted daily traffic flow log at x_0 and increases the prediction error, see Figure 9. The variances at each knot match the predicted variance computed by the dependent spatial GPs, prescribed in Eq. (15). However, the means are offset by the smoothing. In addition, the rapidly growing variance after the last time instant (the 23^{th} hour) suggests that the proposed algorithm is only capable of short-term temporal predictions.

Experiments were run on a computer with a 7^{th} Generation Intel Core i7 processor running at 2.9 GHz using 8 GB RAM. Optimization was carried out in Python 3.7, using COBYLA solver from the SciPy library [25].

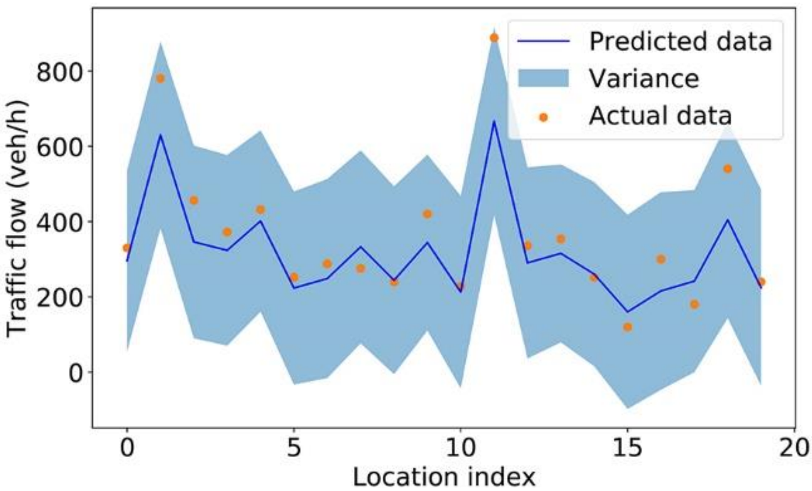

Figure 3
Spatial prediction at a fixed time without a_c. $RMSE = 81.91$

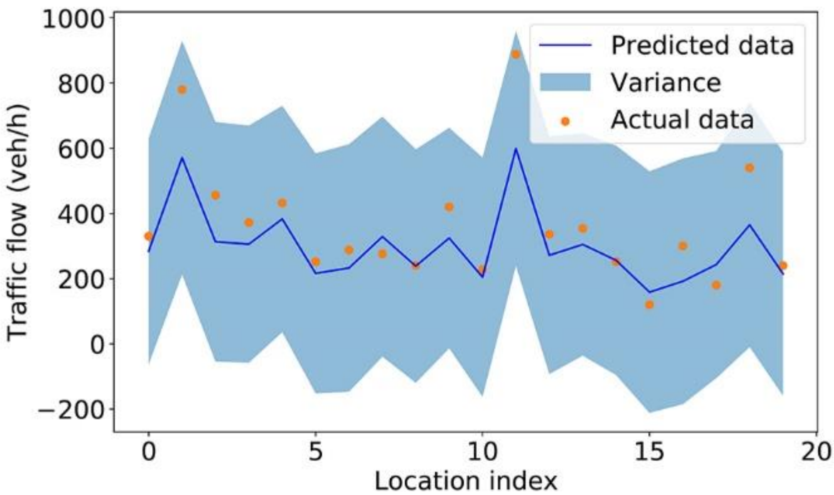

Figure 4
Spatial prediction at a fixed time with a_c. $RMSE = 106.87$

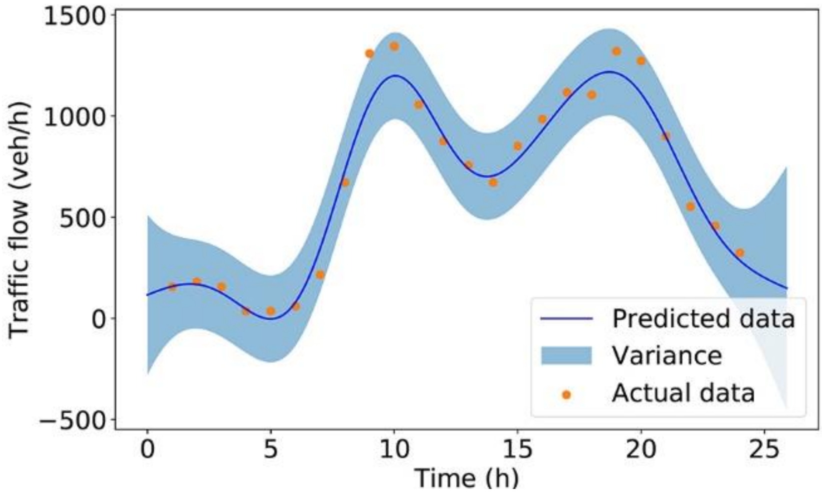

Figure. 5
Temporal prediction at a fixed location without a_c. $RMSE\ =\ 84.85$

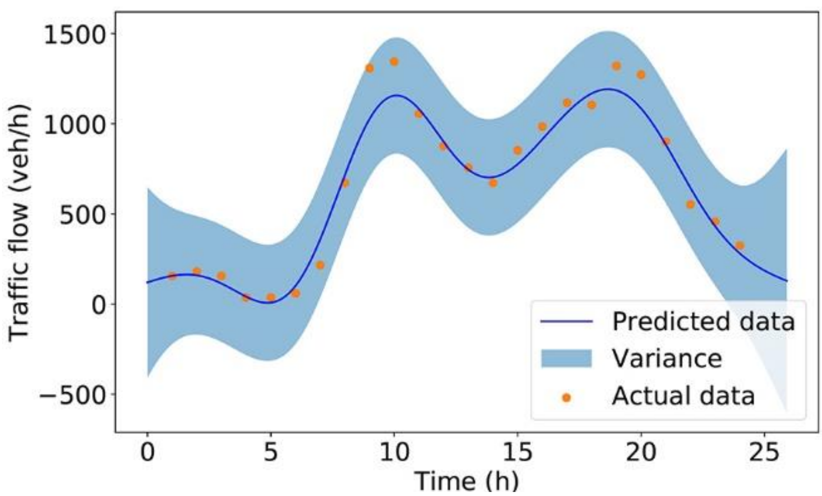

Figure 6
Temporal prediction at a fixed location with a_c. $RMSE\ =\ 97.19$

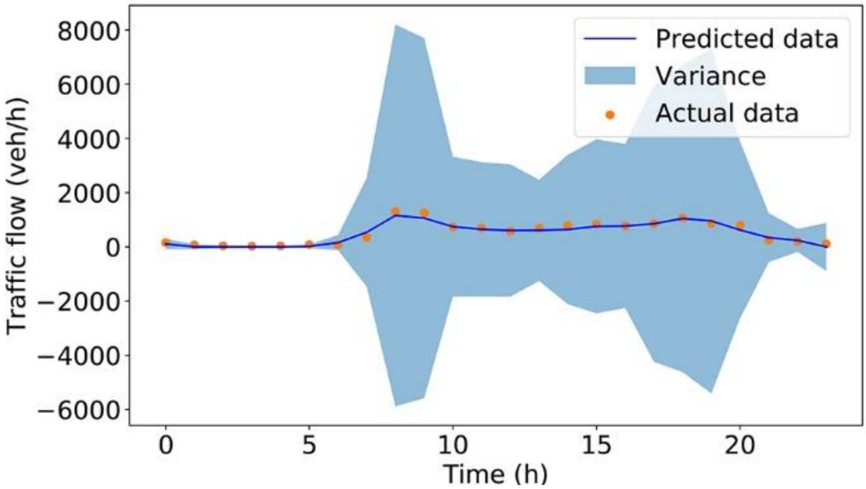

Figure 7
Spatio-temporal prediction using independent spatial GPs. $RMSE = 99.94$

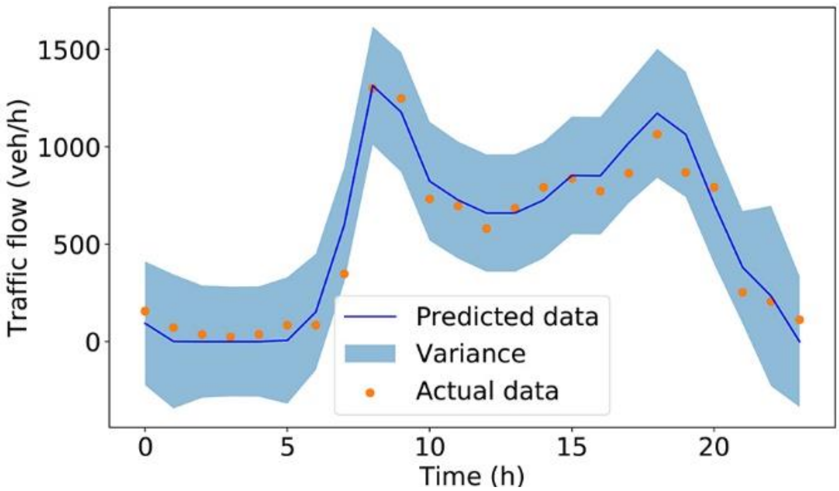

Figure 8
Spatio-temporal prediction with a_c. $RMSE = 97.68$

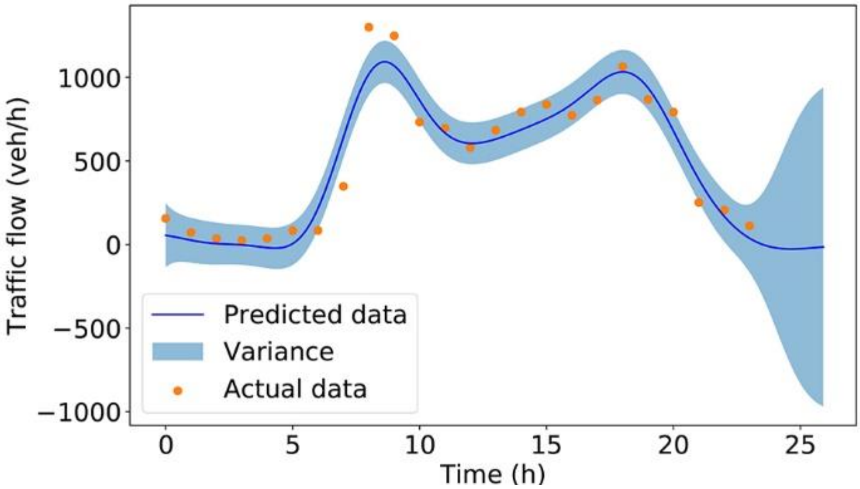

Figure 9

Fitted GP on the posterior at location x_0. $RMSE = 118.35$

Conclusions

The paper presented a method for approximating spatially and temporally dependent data. The method is based on constructing a set of independent Gaussian Processes and then combining them with a common noise term in the covariance function. The efficiency of the algorithm was demonstrated in a traffic flow prediction problem. Results show that the added noise term smoothens the predictions and reduces variance when extrapolating outside of the dataset. RMSE of the predictions increases slightly: in the example demonstrated in Section 3, it was reduced by 2.31%. It is also possible to fit an additional GP on the predicted data to predict both spatially and temporally (for the short-term) outside the dataset. The proposed methodology is universal and computationally light. The sole heuristic in the proposed approach is the selection of the covariance kernel. A different selection of the kernel might yield slightly improved results.

Acknowledgements

This research was supported by the ÚNKP-20-3 New National Excellence Program of the Ministry for Innovation and Technology from the Source of the National Research, Development, and Innovation Fund.

The author would like to thank the Department of Transportation Engineering, University of Naples Federico II for the data used in the numerical example.

References

[1] Rasmussen, C. E. "Gaussian processes in machine learning", Springer, Berlin, Germany, 2003; https://doi.org/10.1007/978-3-540-28650-9_4

[2] Alvarez, M. A., Rosasco, L., Lawrence, N. D., "Kernels for vector-valued functions: A review", arXiv preprint arXiv:1106.6251, 2011; Available at: https://arxiv.org/abs/1106.6251 Accessed: 15. 01. 2021

[3] Boyle, P., Frean, M. "Dependent gaussian processes" In: Proceedings of the 17th International Conference on Neural Information Processing Systems (NIPS), Vancouver, Canada, 2004, pp. 217-224; https://doi.org/10.5555/2976040.2976068

[4] Carmeli, C., Vito, D. E., Toigo, A. "Vector valued reproducing kernel Hilbert spaces of integrable functions and mercer theorem", Analysis and Applications, 4(04), pp. 377-408, 2006; https://doi.org/10.1142/S0219530506000838

[5] Moreno-Muñoz, P., Artés-Rodríguez, A., Álvarez, M. A. "Heterogeneous multi-output gaussian process prediction". In: Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS), Vancouver, Canada, 2018, pp. 6711-6720; Available at: https://arxiv.org/pdf/1805.07633.pdf, Accessed: 07. 01. 2022

[6] López-Lopera, A. F., John, S. T., Durrande, N. "Gaussian process modulated Cox processes under linear inequality constraints". In: Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS, PMLR), Okinawa, Japan, 2019, pp. 1997-2006; Available at: http://proceedings.mlr.press/v89/lopez-lopera19a/lopez-lopera19a.pdf, Accessed: 07. 01. 2022

[7] Kleijnen, J. P. C. "Kriging metamodeling in simulation: A review", European Journal of Operational Research, 192(3), pp. 707-716, 2009; https://doi.org/10.1016/j.ejor.2007.10.013

[8] Wilks, D. S. "A gridded multisite weather generator and synchronization to observed weather data", Water Resources Research, 45(10), W10419, 2009; https://doi.org/10.1029/2009WR007902

[9] Chen, N., Qian, Z., Nabney, I. T. Meng, X. "Wind power forecasts using gaussian processes and numerical weather prediction", IEEE Transactions on Power Systems, 29(2), pp. 656-665, 2013; https://doi.org/10.1109/TPWRS.2013.2282366

[10] Du, W., Xing, Z., Li, M., He, B., Chua, L. H. C., Miao, H. "Optimal sensor placement and measurement of wind for water quality studies in urban reservoirs". In: IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks, Berlin, Germany, 2014, pp. 167-178; http://doi.org/10.1109/IPSN.2014.6846750

[11] Bae, B., Kim, H., Kim, H., Lim, H., Liu, Y., Han, L. D., Freeze, P. B. "Missing data imputation for traffic flow speed using spatio-temporal cokriging", Transportation Research Part C: Emerging Technologies, 88, pp. 124-139, 2018; https://doi.org/10.1016/j.trc.2018.01.015

[12] Girard, A., Rasmussen, C. E., Candela, J. Q., Murray-Smith, R. "Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting", In: Proceedings of the 15th International Conference on Neural Information Processing Systems (NIPS), Vancouver, Canada, 2002, pp. 545-552; Available at: https://papers.nips.cc/paper/2002/file/f3ac63c91272f19ce97c7397825cc15f-Paper.pdf Accessed: 15. 01. 2021.

[13] Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N., Aigrain, S. "Gaussian processes for time-series modelling", Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 371(1984), 2013; https://doi.org/10.1098/rsta.2011.0550

[14] Mohanty, S., Das, S., Chattopadhyay, A., Peralta, P. "Gaussian process time series model for life prognosis of metallic structures", Journal of Intelligent Material Systems and Structures, 20(8), pp. 887-896, 2009; https://doi.org/10.1177/1045389X08099602

[15] Zhang, C., Peng, T., Nazir, M. S. (2022) „A novel hybrid approach based on variational heteroscedastic Gaussian process regression for multi-step ahead wind speed forecasting". International Journal of Electrical Power & Energy Systems, 136(107717), pp. 1-17, 2022; https://doi.org/10.1016/j.ijepes.2021.107717

[16] Crépey, S., Dixon, M. F. "Gaussian Process Regression for Derivative Portfolio Modeling and Application to Credit Valuation Adjustment Computations". Journal of Computational Finance, 24(1), pp. 47-81, 2019; https://doi.org/10.21314/JCF.2020.386, Accessed: 07. 01. 2022.

[17] Hauzenberger, N., Huber, F., Marcellino, M., Petz, N. "Gaussian Process Vector Autoregressions and Macroeconomic Uncertainty". 2021; Available at: https://arxiv.org/pdf/2112.01995.pdf

[18] Hyun, J. W., Li, Y., Huang, C., Styner, M., Lin, W., Zhu, H. "STGP: Spatio-temporal Gaussian Process models for longitudinal neuroimaging data. Neuroimage, 134, pp. 550-562, 2016; http://doi.org/10.1016/j.neuroimage.2016.04.023

[19] Sarkka, S., Solin, A., Hartikainen, J. "Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing: A look at gaussian process regression through Kalman filtering", IEEE Signal Processing Magazine, 30(4), pp. 51-61, 2013; http://doi.org/0.1109/MSP.2013.2246292

[20] Foreman-Mackey, D., Agol, E., Am-bikasaran, S., Angus, R. "Fast and scalable Gaussian Process modeling with applications to astronomical timeseries", The Astronomical Journal, 154(6) pp. 220, 2017; https://doi.org/10.3847/1538-3881/aa9332

[21] Micchelli, C. A., Xu, Y., Zhang, H. “Universal kernels”, Journal of Machine Learning Research, 7(Dec), pp. 2651-2667, 2006; Availableat: at: https://surface.syr.edu/cgi/viewcontent.cgi?article=1003&context=mat Accessed: 15. 01. 2021.

[22] Pensoneault, A., Yang, X., Zhu, X., “Nonnegativity-enforced gaussian process regression”, arXiv preprint arXiv:2004.04632, 2020; Available at: https://arxiv.org/abs/2004.04632 Accessed: 15. 01. 2021.

[23] Cox, M. A. A., Cox, T. F. “Multidimensional scaling”. In: Handbook of data visualization, Springer, Berlin, Germany, 2008, pp. 315-347; https://doi.org/10.1007/978-3-540-33037-0_14

[24] Zou, H., Yue, Y., Li, Q., Yeh, A. G. O. “An improved distance metric for the interpolation of link-based traffic data using kriging: a case study of a large-scale urban road network”, International Journal of Geographical Information Science, 26(4), pp. 667-689, 2012; https://doi.org/10.1080/13658816.2011.609488

[25] Virtanen, P., Gommers, R., Oliphant, T. E, Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J. “SciPy 1.0: fundamental algorithms for scientific computing in python”, Nature Methods, 17(3), pp. 261-272, 2020; https://doi.org/10.1038/s41592-019-0686-2