Probabilistic ML:
Applications and Modeling Investigations

César Lincoln C. Mattos

Federal University of Ceará (UFC)
Department of Computer Science (DC)
Logics and Artificial Intelligence Group (LOGIA)

2021

Agenda

Agenda

1. Who are we?
2. Applications and Modeling Investigations
   - Motivation
   - Recurrent Gaussian process
   - Unscented GPLVM
   - Deep Mahalanobis GP
   - Chained GPs for Wind Turbine modeling
   - Portfolio-based Bayesian optimization
   - LS-SVR as a Bayesian RBF Network
   - Bayesian multilateration
   - Trajectory anomaly detection
3. Concluding Remarks

Fortaleza

Ceará, Brazil

- $\sim$ 2.69 millions of inhabitants.
- 5th largest city in Brazil.
- 34 Km of beaches.
- Around 25-30 $^\circ\mathrm{C}$ **all year**.
- 2nd Brazilian tourism destination.

‘Beira Mar’ Avenue.

‘Iracema guerreira’ statue.

‘Jangada’ at the sunset.

‘Dragão do Mar’ Center of Art and Culture.

Federal University of Ceará (UFC)

- 8 campi.
- $\sim$ 2,150 professors.
- $\sim$ 27,000 undergraduate students.
- $\sim$ 6,000 graduate students.
- > 110 undergraduate courses.
  - $\rightarrow$ 15 courses on CS.
- > 150 graduate courses.
  - $\rightarrow$ 2 courses on CS.

UNIVERSIDADE FEDERAL DO CEARÁ

Department of Computer Science (DC)

- Created in 1990.
- 2 Bsc degrees.
  - → Computer Science.
  - → Computer Engineering.
- Specialization degree on Information Technology.
- MSc and PhD in Computer Science.
- 30 professors.
- 8 research laboratories.
- 2 teaching laboratories.

DEPARTAMENTO DE COMPUTAÇÃO

DC/UFC

MSc and PhD programs in CS (MDCC)

- Started in 1995 (Msc) and 2004 (PhD).
- Strong academic production and fund-raising capacity.

Logics and Artificial Intelligence Group (LOGIA)

- Research in Logics, AI/ML and Computer Theory.
- AI/ML
  - → Prof. João Paulo P. Gomes, Prof. João Paulo do Vale Madeiro and Prof. César Lincoln C. Mattos.
  - → Focus on theoretical modeling and applications.
  - → International collaborations and joint projects with industry.

MDCC
Mestrado e Doutorado em Ciência da Computação

LOGIA

Agenda

Agenda

Motivation

- ML **teaching**, **research** and **applications** at UFC (and in most Brazilian universities) have focused mostly on standard ML.

Motivation

- ML **teaching**, **research** and **applications** at UFC (and in most Brazilian universities) have focused mostly on standard ML.
- There has been a growing shift towards DL (specially in applications).

Motivation

- ML **teaching**, **research** and **applications** at UFC (and in most Brazilian universities) have focused mostly on standard ML.
- There has been a growing shift towards DL (specially in applications).
- One of LOGIA's current goals is to provide some basis to overcome such Probabilistic ML limited local adoption.

Motivation

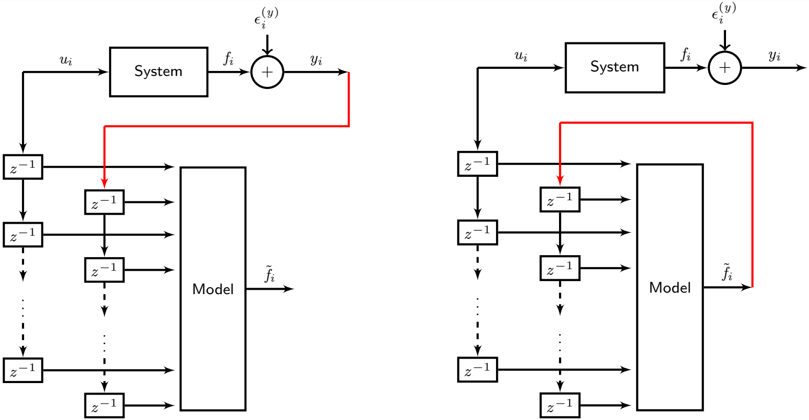
- ML **teaching**, **research** and **applications** at UFC (and in most Brazilian universities) have focused mostly on standard ML.
- There has been a growing shift towards DL (specially in applications).
- One of LOGIA's current goals is to provide some basis to overcome such Probabilistic ML limited local adoption.
- Along the way we hope to make contributions to the overall ML community.

Agenda

Dynamical modeling

System identification methodology (Ljung, 1999)

1. Collect data;
2. **Determine model structure**;
3. **Perform model selection**;
4. **Validate the model**.



(a) One-step ahead prediction. **(b) Free simulation**.

Gaussian processes for system identification

- **Models with external dynamics**: Uses measurements as regressors.
  - **Nonlinear autoregressive with exogenous inputs (NARX)**:

$$
\begin{aligned}
y_i &= f(\boldsymbol{x}_i) + \epsilon_i^{(y)}, \\
\boldsymbol{x}_i &= [\bar{\boldsymbol{y}}_{i-1}, \bar{\boldsymbol{u}}_{i-1}]^\top \\
&= [[y_{i-1}, y_{i-2}, \cdots, y_{i-L_y}], [u_{i-1}, u_{i-2}, \cdots, u_{i-L_u}]]^\top.
\end{aligned}
$$

Gaussian processes for system identification

- **Models with external dynamics**: Uses measurements as regressors.
  - **Nonlinear autoregressive with exogenous inputs (NARX)**:

$$
\begin{aligned}
y_i &= f(\boldsymbol{x}_i) + \epsilon_i^{(y)}, \\
\boldsymbol{x}_i &= [\bar{\boldsymbol{y}}_{i-1}, \bar{\boldsymbol{u}}_{i-1}]^\top \\
&= [[y_{i-1}, y_{i-2}, \cdots, y_{i-L_y}], [u_{i-1}, u_{i-2}, \cdots, u_{i-L_u}]]^\top.
\end{aligned}
$$

- **Models with internal dynamics**: Uses latent states $\boldsymbol{x}$.
  - **State-space model (SSM)**:

$$
\begin{aligned}
\boldsymbol{x}_i &= f(\boldsymbol{x}_{i-1}, u_{i-1}) + \boldsymbol{\epsilon}_i^{(x)}, \\
y_i &= g(\boldsymbol{x}_i) + \epsilon_i^{(y)}.
\end{aligned}
$$

Gaussian processes for system identification

- **Models with external dynamics**: Uses measurements as regressors.
  - **Nonlinear autoregressive with exogenous inputs (NARX)**:

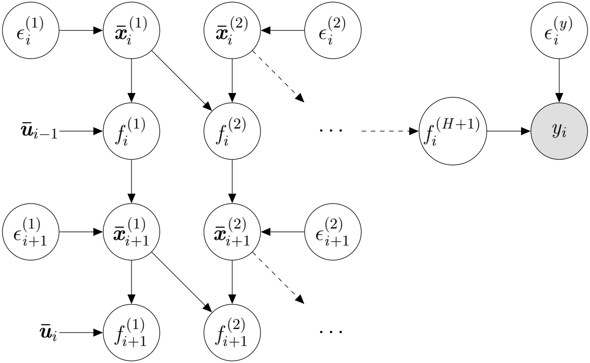
$$
\begin{aligned}
y_i &= f(\boldsymbol{x}_i) + \epsilon_i^{(y)}, \\
\boldsymbol{x}_i &= [\bar{\boldsymbol{y}}_{i-1}, \bar{\boldsymbol{u}}_{i-1}]^\top \\
&= [[y_{i-1}, y_{i-2}, \cdots, y_{i-L_y}], [u_{i-1}, u_{i-2}, \cdots, u_{i-L_u}]]^\top.
\end{aligned}
$$

- **Models with internal dynamics**: Uses latent states $\boldsymbol{x}$.
  - **State-space model (SSM)**:

$$
\begin{aligned}
\boldsymbol{x}_i &= f(\boldsymbol{x}_{i-1}, u_{i-1}) + \boldsymbol{\epsilon}_i^{(x)}, \\
y_i &= g(\boldsymbol{x}_i) + \epsilon_i^{(y)}.
\end{aligned}
$$

- **Dynamical Gaussian process models**: GP priors on functions.
  - **GP-NARX**: tractable for Gaussian observation noise.
  - **GP-SSM**: intractable due to the latent inputs.

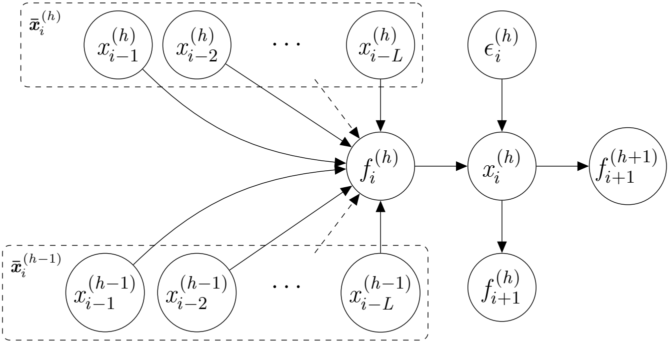
Recurrent Gaussian Processes (RGPs)[1]



RGP graphical model with H hidden layers.

- **Hierarchical structure**: Separate modeling of transition (hidden) and observation (emission) functions.
- **Latent dynamical variables**: Avoids feedback of observations.
- **REVARB (REcurrent VARiational Bayes)**: Follows mean field variational inference from Damianou and Lawrence (2013).

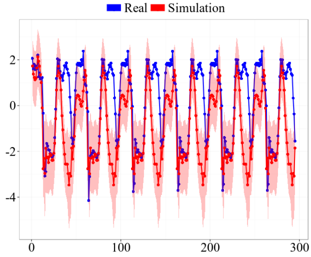
[1]Mattos et al. **Recurrent Gaussian processes**, 2016.



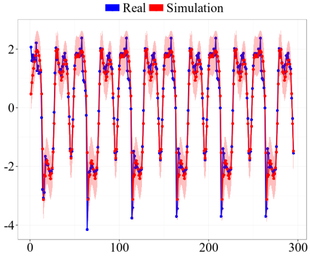
Detailing of a recurrent transition layer of the RGP model.

$$
\begin{aligned}
p\left(\boldsymbol{f}^{(h)}\middle|\hat{\boldsymbol{X}}^{(h)}\right) &= \mathcal{N}\left(\boldsymbol{f}^{(h)}\middle|\mathbf{0}, \boldsymbol{K}_f^{(h)}\right), && 1 \le h \le H+1,\\
p\left(x_i^{(h)}\right) &= \mathcal{N}\left(x_i^{(h)}\middle|\mu_{0i}^{(h)}, \lambda_{0i}^{(h)}\right), && 1 \le i \le L,\\
p\left(x_i^{(h)}\middle|f_i^{(h)}\right) &= \mathcal{N}\left(x_i^{(h)}\middle|f_i^{(h)}, \sigma_h^2\right), && L+1 \le i \le N,\\
p\left(y_i\middle|f_i^{(H+1)}, \sigma_{H+1}^2\right) &= \mathcal{N}\left(y_i\middle|f_i^{(H+1)}, \sigma_{H+1}^2\right), && L+1 \le i \le N.
\end{aligned}
$$

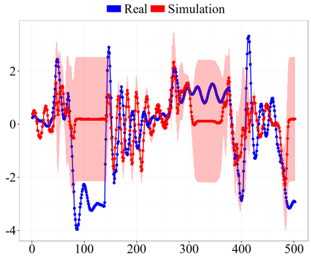
RGP for system identification (free simulation)



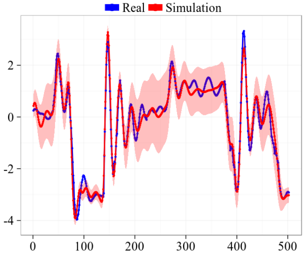
GP-NARX - RMSE = 1.9245.



RGP with 2 hidden layers - RMSE = 0.4513.



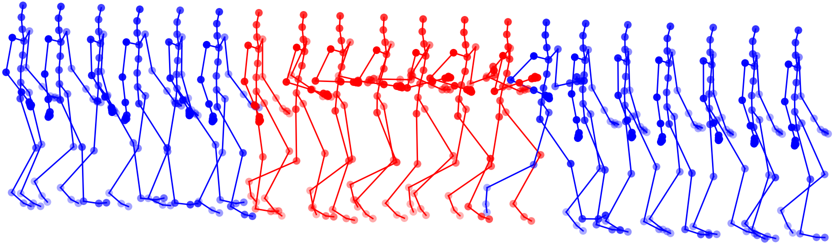
GP-NARX - RMSE = 1.5488.



RGP with 2 hidden layers - RMSE = 0.3104.

RGP for Human Motion and Avatar Control

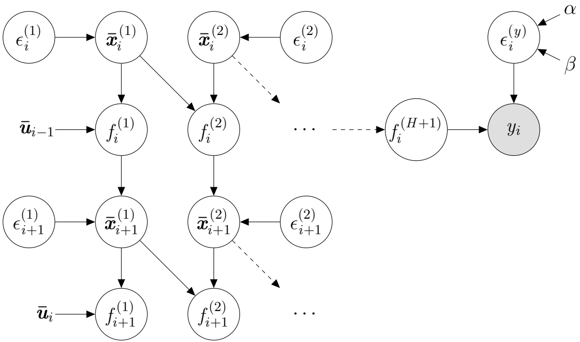
- Walking and running motions[2] with 57 output dimensions and coordinate of the left toes as input signal.
- Use velocity as a signal to control an avatar's motion.



Motion generated by the RGP model with a step control signal for the velocity.

[2] Data available at `http://mocap.cs.cmu.edu/`

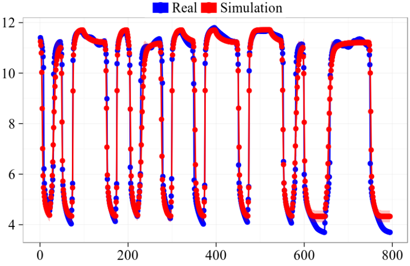
RGP-t/REVARB-t[3]: RGP + Student-t likelihood



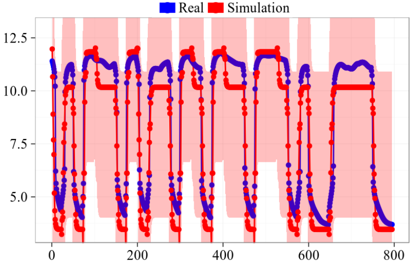
$$
\begin{aligned}
p\left(\boldsymbol{f}^{(h)} \middle| \hat{\boldsymbol{X}}^{(h)}\right) &= \mathcal{N}\left(\boldsymbol{f}^{(h)} \middle| \boldsymbol{0}, \boldsymbol{K}_f^{(h)}\right), && 1 \leq h \leq H+1,\\
p\left(x_i^{(h)}\right) &= \mathcal{N}\left(x_i^{(h)} \middle| \mu_{0i}^{(h)}, \lambda_{0i}^{(h)}\right), && 1 \leq i \leq L,\\
p\left(x_i^{(h)} \middle| f_i^{(h)}\right) &= \mathcal{N}\left(x_i^{(h)} \middle| f_i^{(h)}, \sigma_h^2\right), && L+1 \leq i \leq N,\\
p\left(y_i \middle| f_i^{(H+1)}, \tau_i\right) &= \mathcal{N}\left(y_i \middle| f_i^{(H+1)}, \tau_i^{-1}\right), && L+1 \leq i \leq N,\\
p(\tau_i) &= \Gamma(\tau_i | \alpha, \beta), && L+1 \leq i \leq N.
\end{aligned}
$$

[3] Mattos et al., **Deep RGP for outlier-robust system identification**, 2017.

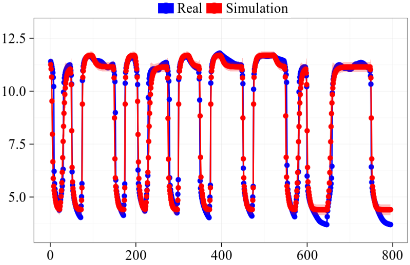
RGP-t for robust system identification



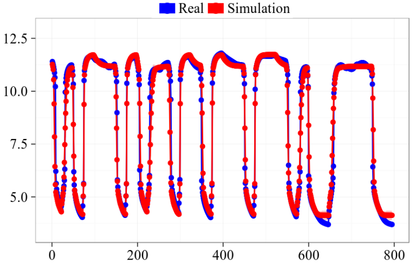
(a) RGP $(H = 2)$ without outliers.



(b) RGP $(H = 2)$ with 30% of outliers.



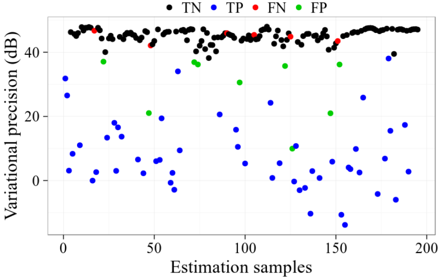
(c) RGP-t $(H = 2)$ without outliers.



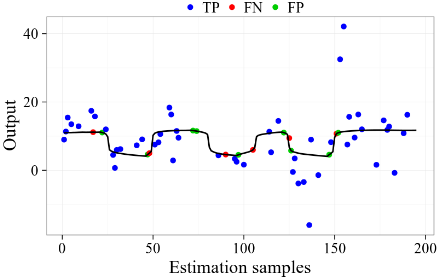
(d) RGP-t $(H = 2)$ with 30% of outliers.

Free simulation on test data after estimation on the **pH dataset**.

RGP-t for robust system identification



(a) Variational precisions after optimization.



(b) Detected outliers.

Outlier detection by the RGP-t model with 2 hidden layers and REVARB-t inference for the *pH* estimation data in the scenario with 30% of outliers.

Scaling inference with RGP models[4]

S-REVARB

SVI framework adapted to the RGP model and the REVARB method, following Hensman et al. (2013) for better scaling.

- **Local S-REVARB**: More directly derived, but preserves all the variational parameters.
- **Global S-REVARB**: Avoids the growth of the variational parameters with recognition models.

$$
\begin{tikzcd}
\boldsymbol{u} \arrow[r] \arrow[dr] & \mathbf{RNN}_{\mu}^{(1)} \arrow[r] & \boldsymbol{\mu}^{(1)} \arrow[l, bend right] \arrow[r] \arrow[dl] \arrow[dr] & \mathbf{RNN}_{\mu}^{(2)} \arrow[r] & \boldsymbol{\mu}^{(2)} \arrow[l, bend right] \arrow[r, dashed] \arrow[dl] \arrow[dr, dashed] & \cdots \arrow[r, dashed] \arrow[dr, dashed] & \mathbf{RNN}_{\mu}^{(H)} \arrow[r] & \boldsymbol{\mu}^{(H)} \arrow[l, bend right] \arrow[dl] \\
 & \mathbf{MLP}_{\lambda}^{(1)} \arrow[r] & \boldsymbol{\lambda}^{(1)} & \mathbf{MLP}_{\lambda}^{(2)} \arrow[r] & \boldsymbol{\lambda}^{(2)} & \cdots & \mathbf{MLP}_{\lambda}^{(H)} \arrow[r] & \boldsymbol{\lambda}^{(H)}
\end{tikzcd}
$$

Diagram for the recognition models of the Global S-REVARB framework.

[4] Mattos and Barreto, **A stochastic variational framework for Recurrent Gaussian processes models**, 2019.

System identification with large datasets

Wiener-Hammerstein (95,000/84,000 training/testing samples)	RMSE	NLPD
RNN (1 hidden layer)	1.222×10^{-2}	-
RNN (2 hidden layers)	8.247×10^{-3}	-
Variational Sparse GP-NARX ($N = 5000$)	3.584×10^{-2}	-1.883
REVARB ($H = 1, N = 5000$)	2.037×10^{-2}	-2.406
REVARB ($H = 2, N = 5000$)	1.547×10^{-2}	-2.544
Local S-REVARB ($H = 1$)	1.295×10^{-2}	-2.609
Local S-REVARB ($H = 2$)	2.372×10^{-2}	-2.308
Global S-REVARB ($H = 1$)	8.369×10^{-3}	-2.606
Global S-REVARB ($H = 2$)	$\mathbf{5.664 \times 10^{-3}}$	$\mathbf{-2.643}$

Summary of free simulation results after estimation from large dynamical datasets.

	Size
RNN (1 hidden layer)	2201
RNN (2 hidden layers)	4402
Local S-REVARB ($H = 1$)	194,206
Local S-REVARB ($H = 2$)	386,574
Global S-REVARB ($H = 1$)	8608
Global S-REVARB ($H = 2$)	15,378

Comparison of the number of adjustable parameters (RNNs) or hyperparameters and variational parameters (S-REVARB variants) in the *Wiener-Hammerstein* benchmark.

Agenda

Gaussian Process Latent Variable Model (GPLVM)

- Variational inference for GPLVMs only has exact solutions for a limited set of kernels (Titsias and Lawrence, 2010).

Gaussian Process Latent Variable Model (GPLVM)

- Variational inference for GPLVMs only has exact solutions for a limited set of kernels (Titsias and Lawrence, 2010).
  - → This restriction is due to the integrals, named Ψ-statistics, that appear in the evidence lower bound:

$$
\begin{aligned}
2\ln p(\boldsymbol{y}_{:d}) \geq \quad & \ln|\boldsymbol{K}_u| - n\ln\left(2\pi\sigma_y^2\right) - \ln|\,\boldsymbol{W}\,| \\
& - \frac{\boldsymbol{y}_{:d}^\top \boldsymbol{y}_{:d}}{\sigma_y^2} + \frac{\boldsymbol{y}_{:d}^\top \boxed{\boldsymbol{\Psi}_1}\, \boldsymbol{W}^{-1} \boxed{\boldsymbol{\Psi}_1}^\top \boldsymbol{y}_{:d}}{\sigma_y^2} \\
& - \frac{\boxed{\psi_0}}{\sigma_y^2} + \frac{\mathrm{Tr}\left(\boldsymbol{K}_u^{-1} \boxed{\boldsymbol{\Psi}_2}\right)}{\sigma_y^2}
\end{aligned}
$$

Unscented GPLVM[5]

- Popular GP frameworks, by default, solve this issue by using the Gauss-Hermite (GH) quadrature.

[5] de Souza *et al*. **Learning GPLVM with arbitrary kernels using the unscented transformation**, 2021.

Unscented GPLVM[5]

- Popular GP frameworks, by default, solve this issue by using the Gauss-Hermite (GH) quadrature.
- However, GH is not viable on problems with modest input dimensions D due to cost proportional to H^D (for a chosen H).

[5]de Souza *et al*. **Learning GPLVM with arbitrary kernels using the unscented transformation**, 2021.

Unscented GPLVM[5]

- Popular GP frameworks, by default, solve this issue by using the Gauss-Hermite (GH) quadrature.
- However, GH is not viable on problems with modest input dimensions D due to cost proportional to H^D (for a chosen H).
- Monte Carlo (MC) integration could also be used, but due to its stochasticity, efficient optimizers (eg L-BFGS) cannot be used.

[5]de Souza *et al*. **Learning GPLVM with arbitrary kernels using the unscented transformation**, 2021.

Unscented GPLVM[5]

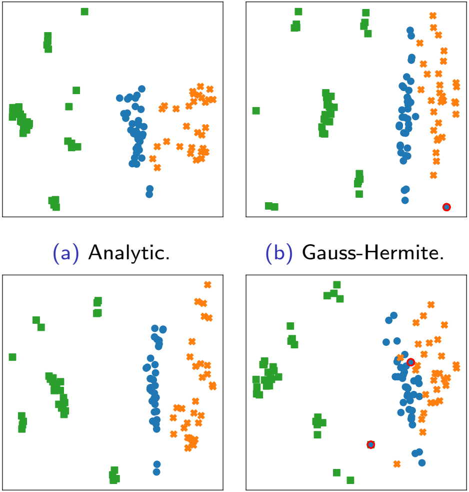
- Popular GP frameworks, by default, solve this issue by using the Gauss-Hermite (GH) quadrature.
- However, GH is not viable on problems with modest input dimensions D due to cost proportional to H^D (for a chosen H).
- Monte Carlo (MC) integration could also be used, but due to its stochasticity, efficient optimizers (eg L-BFGS) cannot be used.
- The unscented transformation (UT) presents itself as a parameterless, deterministic, and linearly scaling alternative.

[5] de Souza *et al.* **Learning GPLVM with arbitrary kernels using the unscented transformation**, 2021.

Unscented GPLVM - dimensionality reduction

Results for the Oil flow dataset. Note that the UT managed to achieve better results while using $\frac{1}{3}$ of the evaluations of the GH.

Method	**# evaluations**	**Kernel**	**Accuracy**
PCA	-	-	79.0 ± 6.5
Analytic	-	RBF	98.0 ± 2.7
Gauss-Hermite	32	Matérn 3/2	95.0 ± 6.1
Unscented	10	Matérn 3/2	100.0 ± 0.0
Monte Carlo	10	Matérn 3/2	85.6 ± 8.7
	32	Matérn 3/2	87.9 ± 5.4
	200	Matérn 3/2	95.4 ± 3.0



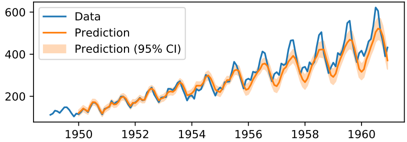
(a) Analytic. (b) Gauss-Hermite.

(c) UT. (d) MC(32).

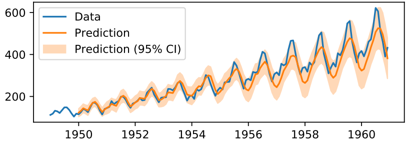
Unscented GPLVM - time series prediction

Results for the Airline dataset. Comparing UT with GH, a 170 fold increase in number of evaluations resulted in only a 0.06 decrease in NLPD.

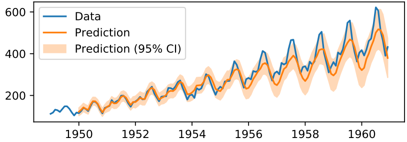
Method	# evaluations	Kernel	NLPD
GP-NARX	-	Per.+RBF+Lin.	7.46
GPLVM - Analytic	-	RBF+Linear	7.08
GPLVM - GH	4096	Per.+RBF+Lin.	5.20
GPLVM - UT	24	Per.+RBF+Lin.	5.26
GPLVM - MC	24	Per.+RBF+Lin.	5.41 ± 0.17
	200	Per.+RBF+Lin.	5.19 ± 0.06
	4096	Per.+RBF+Lin.	5.19 ± 0.01



(a) GP-NARX



(b) GPLVM (GH).



(c) GPLVM (UT).

Agenda

Deep Mahalanobis Gaussian Process[6]

- Most widely used kernels are stationary, which hinders the modeling of functions with input dependent smoothness.

[6] Work in progress by Daniel de Souza.

Deep Mahalanobis Gaussian Process[6]

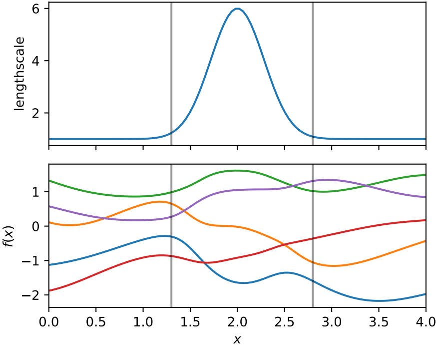
- Most widely used kernels are stationary, which hinders the modeling of functions with input dependent smoothness.
- Based on the work of Gibbs (1997), Paciorek (2003) shows that any stationary kernel k can be transformed into a non-stationary kernel k_{NS} through the following transformation:

$$
\begin{aligned}
k(\boldsymbol{a}, \boldsymbol{b}) &= \phi\big((\boldsymbol{a} - \boldsymbol{b})\boldsymbol{\Delta}^{-1}(\boldsymbol{a} - \boldsymbol{b})^\top\big), \\
k_{\text{NS}}(\boldsymbol{a}, \boldsymbol{b}) &= \sqrt{2}\, \frac{|\boldsymbol{\Delta}(\boldsymbol{a})|^{\frac{1}{4}} |\boldsymbol{\Delta}(\boldsymbol{b})|^{\frac{1}{4}}}{|\boldsymbol{\Delta}(\boldsymbol{a}) + \boldsymbol{\Delta}(\boldsymbol{b})|^{\frac{1}{2}}} \\
&\quad \cdot \phi\left((\boldsymbol{a} - \boldsymbol{b})\left(\frac{\boldsymbol{\Delta}(\boldsymbol{a}) + \boldsymbol{\Delta}(\boldsymbol{b})}{2}\right)^{-1}(\boldsymbol{a} - \boldsymbol{b})^\top\right).
\end{aligned}
$$

[6]Work in progress by Daniel de Souza.

Deep Mahalanobis GP

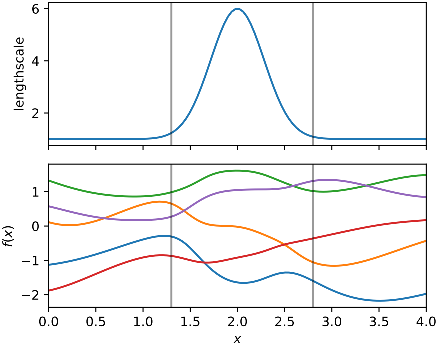
- As noted by Gibbs (1997), the varying lengthscales lose their interpretability. For example:



- Note that the function is wavier when the lengthscale is higher.

Deep Mahalanobis GP

- As noted by Gibbs (1997), the varying lengthscales lose their interpretability. For example:



- Note that the function is wavier when the lengthscale is higher.

- Paciorek (2003) showed that, unlike stationary kernels, this kernel does not induce a metric space on the inputs.

- In stationary kernels, the mapping of $\boldsymbol{x}$ to this metric space is $\boldsymbol{x}\left(\boldsymbol{\Delta}^{-1/2}\right)^{\top}$

- Due to the replacement of $\boldsymbol{\Delta}$ by $\frac{\boldsymbol{\Delta}(\boldsymbol{a})+\boldsymbol{\Delta}(\boldsymbol{b})}{2}$, the dependence of $\boldsymbol{a}$ in the projection of $\boldsymbol{b}$ (and *vice versa*) means that the triangle inequality can be violated.

Deep Mahalanobis GP

- Is it possible to define non-stationary kernels and preserve at least one of these properties?

Deep Mahalanobis GP

- Is it possible to define non-stationary kernels and preserve at least one of these properties?
- Yes! At least for squared exponential kernels. First we rewrite:

$$
\begin{aligned}
\mathrm{RBF}(\boldsymbol{a}, \boldsymbol{b}) &= \exp\left(-\frac{1}{2}(\boldsymbol{a}-\boldsymbol{b})\boldsymbol{\Delta}^{-1}(\boldsymbol{a}-\boldsymbol{b})^\top\right) \\
&= \exp\left(-\frac{1}{2}(\boldsymbol{a}\boldsymbol{\Delta}^{-\frac{1}{2}}{}^\top - \boldsymbol{b}\boldsymbol{\Delta}^{-\frac{1}{2}}{}^\top)(\boldsymbol{a}\boldsymbol{\Delta}^{-\frac{1}{2}}{}^\top - \boldsymbol{b}\boldsymbol{\Delta}^{-\frac{1}{2}}{}^\top)^\top\right) \\
&= \exp\left(-\frac{1}{2}(\boldsymbol{a}\boldsymbol{W}^\top - \boldsymbol{b}\boldsymbol{W}^\top)(\boldsymbol{a}\boldsymbol{W}^\top - \boldsymbol{b}\boldsymbol{W}^\top)^\top\right).
\end{aligned}
$$

Now we just need to add an input dependency on $\boldsymbol{W}$.

Deep Mahalanobis GP

- The non-stationary kernel becomes:

$$k_{\mathrm{NS}}(\boldsymbol{a}, \boldsymbol{b}) = \exp\left(-\frac{1}{2}(\boldsymbol{a}\boldsymbol{W}(\boldsymbol{a})^\top - \boldsymbol{b}\boldsymbol{W}(\boldsymbol{b})^\top)(\boldsymbol{a}\boldsymbol{W}(\boldsymbol{a})^\top - \boldsymbol{b}\boldsymbol{W}(\boldsymbol{b})^\top)^\top\right).$$

Deep Mahalanobis GP

- The non-stationary kernel becomes:

$$k_{\mathrm{NS}}(\boldsymbol{a}, \boldsymbol{b}) = \exp\left(-\frac{1}{2}(\boldsymbol{a}\boldsymbol{W}(\boldsymbol{a})^\top - \boldsymbol{b}\boldsymbol{W}(\boldsymbol{b})^\top)(\boldsymbol{a}\boldsymbol{W}(\boldsymbol{a})^\top - \boldsymbol{b}\boldsymbol{W}(\boldsymbol{b})^\top)^\top\right).$$

- There is still no notion of lengthscales as before, but we kept the property that this kernel induces a metric space on the input.

Deep Mahalanobis GP

- The non-stationary kernel becomes:

$$k_{\mathrm{NS}}(\boldsymbol{a}, \boldsymbol{b}) = \exp\left(-\frac{1}{2}(\boldsymbol{a}\boldsymbol{W}(\boldsymbol{a})^\top - \boldsymbol{b}\boldsymbol{W}(\boldsymbol{b})^\top)(\boldsymbol{a}\boldsymbol{W}(\boldsymbol{a})^\top - \boldsymbol{b}\boldsymbol{W}(\boldsymbol{b})^\top)^\top\right).$$

- There is still no notion of lengthscales as before, but we kept the property that this kernel induces a metric space on the input.
- By placing a GP prior on $\boldsymbol{W}(\boldsymbol{x})$, we obtain a deep Gaussian process model where each layer connects not through outputs to inputs, but outputs to kernel hyperparameters.

Deep Mahalanobis GP

We chose a two-layer model as a starting point:

$$p(\boldsymbol{W} \mid \boldsymbol{X}) = \prod_{q,d}^{Q,D} \mathcal{N}\left(\boldsymbol{w}_{:qd} \mid \mathbf{0}, \boldsymbol{K}_w^{(q)}\right),$$

$$p(\boldsymbol{f} \mid \boldsymbol{W}, \boldsymbol{X}) = \mathcal{N}(\boldsymbol{f} \mid \mathbf{0}, \boldsymbol{K}_f),$$

where:

$$\left[\boldsymbol{K}_w^{(q)}\right]_{ij} = {\sigma_w^{(q)}}^2 \exp\left(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{x}_j){\boldsymbol{\Delta}_w^{(q)}}^{-1}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top\right)$$

$$[\boldsymbol{K}_f]_{ij} = \sigma_f^2 \exp\left(-\frac{1}{2}(\boldsymbol{x}_i \boldsymbol{W}_i^\top - \boldsymbol{x}_j \boldsymbol{W}_j^\top)(\boldsymbol{x}_i \boldsymbol{W}_i^\top - \boldsymbol{x}_j \boldsymbol{W}_j^\top)^\top\right)$$

Variational inference in this model is an extension of the methods by Titsias and Lázaro-Gredilla (2013), which deals with the stationary case.

Deep Mahalanobis GP

Our initial experiments against doubly stochastic DGP (Salimbeni and Deisenroth, 2017) shows that DMGP has equivalent or better performance, with DMGP having a more significant bias for dimensionality reduction.

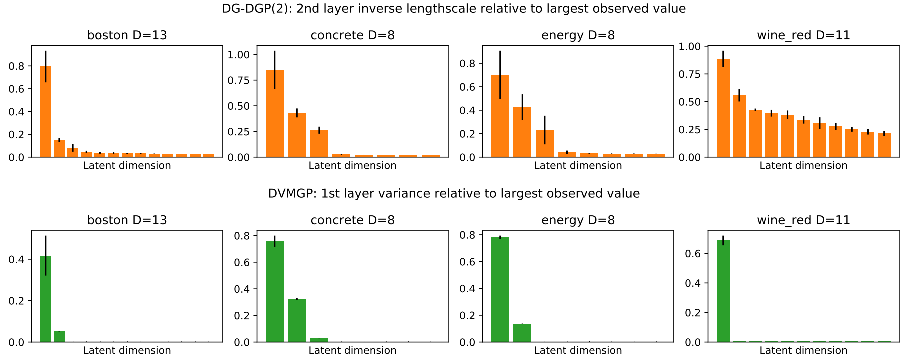


Figure 3: Most relevant latent dimension for each model.

Deep Mahalanobis GP

Our initial experiments against doubly stochastic DGP (Salimbeni and Deisenroth, 2017) shows that DMGP has equivalent or better performance, with DMGP having a more significant bias for dimensionality reduction.

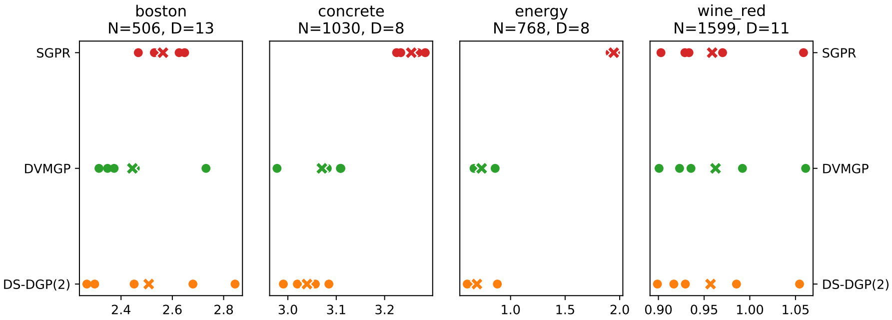


Figure 4: NLPD for each of the 5-fold splits.

Agenda

Wind Turbine Power Curve (WTPC) Modeling

Problem Description: Model the distribution of the normalized power p, given the wind speed v.

Data Peculiarities

- Sigmoidal shape limited in the interval $[0, 1]$;
- Heteroscedastic noise;
- Presence of outliers, whose location can be input-dependant.

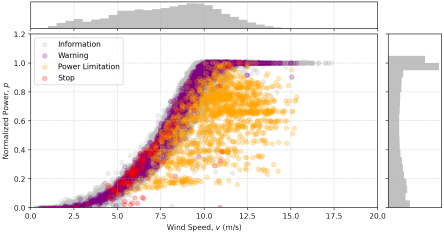


Figure 5: Normalized power p vs. wind speed v data used for WTPC modeling. Color-coding represents operational status derived from event logs.

Chained GPs applied to WTPC Modeling[7]

- We follow a Chained GP approach (Saul et al, 2016).
- **Likelihood**: We choose a Student-t likelihood whose parameters depend on $x = v$ through $L = 3$ independent GPs $f^{(1)} = f, f^{(2)} = g, f^{(3)} = h$:

$$p(y_i|f_i, g_i, h_i) = \mathcal{T}(y_i|\mu_y = f_i, \sigma_y = t(g_i), \nu = t'(h_i)),$$

where $t(g) = \exp(g)$, and $t'(h) = 3 + \exp(h)$.

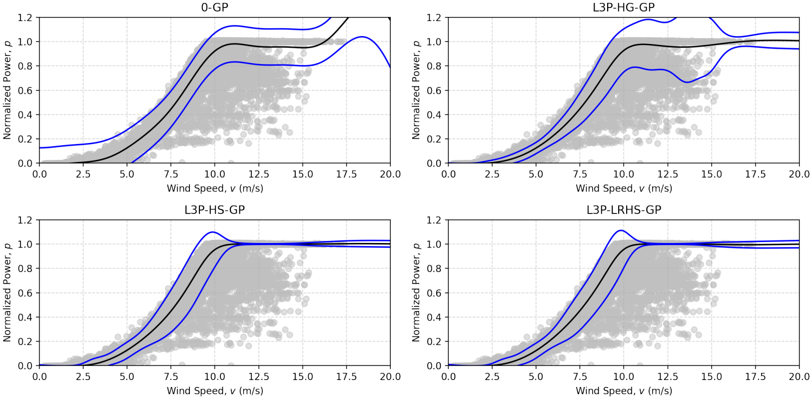
- **Domain knowledge**: We consider a sigmoidal-shaped mean function $\mu_f(\cdot)$ for the GP prior on f:

$$\mu_f(x) = \left[1 + \exp\left(-\left(\frac{v - v_0}{s}\right)\right)\right]^{-1/\gamma}.$$

[7] Virgolino, **Wind Turbine Power Curve Modeling with Gaussian Processes**, 2020.

Chained GPs applied to WTPC

- **Inference**: Variational approach with ELBO that can be factorized to enable SVI.



Experiments with different regression models. 0-GP: standard GP; L3P: Logistic 3-Parameter; HS: Gaussian; HS: Student-t; LRHS: Locally Robust Heteroscedastic Student-t.

Agenda

Portfolio-based Bayesian optimization

- Bayesian optimization has been an effective entry point to sell GPs and Bayesian methods in practical applications.

Portfolio-based Bayesian optimization

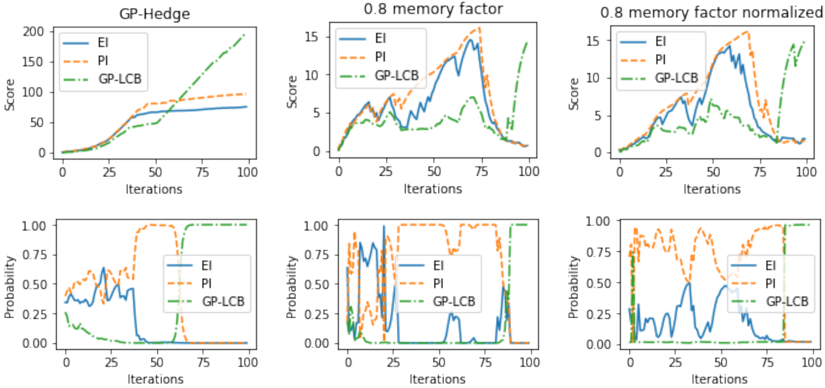
- Bayesian optimization has been an effective entry point to sell GPs and Bayesian methods in practical applications.
- Portfolio-based strategies (Hoffman et al., 2011; Shahriari et al., 2014) have been a straightforward approach to alleviate the need to choose an acquisition function and to improve results.
- GP-Hedge (Hoffman et al., 2011) adopts a portfolio of acquisition functions governed by a multi-armed bandit strategy.
  - $\rightarrow$ All past measures of each acquisition function are considered to choose the next query point.

Portfolio-based Bayesian optimization

- Bayesian optimization has been an effective entry point to sell GPs and Bayesian methods in practical applications.
- Portfolio-based strategies (Hoffman et al., 2011; Shahriari et al., 2014) have been a straightforward approach to alleviate the need to choose an acquisition function and to improve results.
- GP-Hedge (Hoffman et al., 2011) adopts a portfolio of acquisition functions governed by a multi-armed bandit strategy.
  - $\rightarrow$ All past measures of each acquisition function are considered to choose the next query point. Is this a desirable behavior?

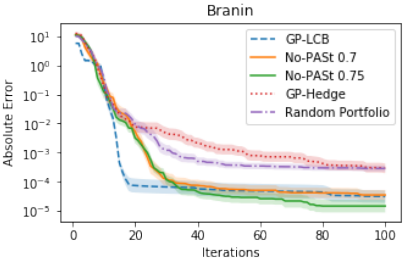
Normalized Portfolio Allocation Strategy BO[8]

- No-PASt-BO aims to overcome GP-Hedge limitations by
  - → reducing the influence of far past evaluations;
  - → presenting a built-in normalization step that avoids similar probabilities in the portfolio.

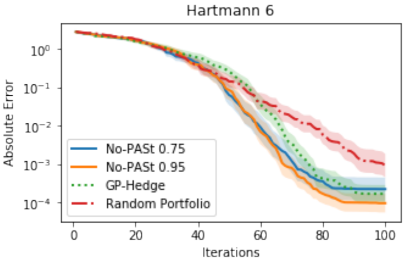


[8]Vasconcelos et al., **No-PASt-BO: Normalized Portfolio Allocation Strategy for Bayesian Optimization**, 2019.

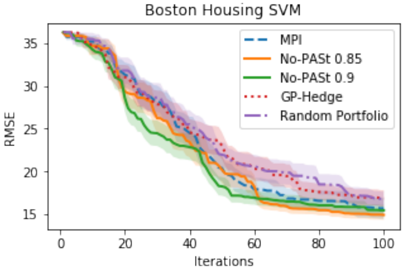
No-PASt-BO



(a) Portfolios with 3 acquisition functions.



(b) Portfolios with 9 acquisition functions.



(c) SVR hyperparameter optimization task.

Agenda

LS-SVR as Bayesian RBF networks

LS-SVR

- Least squares support vector machine (LS-SVM) is simplification of classical SVMs using all points as support vectors;
- Solved in the dual space using least squares;
- Very popular, with $\approx$10k citations, and people use the same formulation for regression (LS-SVR).

LS-SVR as Bayesian RBF networks

Our take on it[9]

- LS-SVR is a point estimate (MAP) for a Bayesian GLM;
- We show how to encode the SV constraints as a Gaussian prior;
- Notably, our prior is conjugate and we can go Bayesian "for free".

Given $\epsilon > 0$, a Bayesian ϵ-LS-SVR is a Bayesian RBF network with all training points as centroids in the hidden layer:

$$\begin{aligned} y_n &\sim \mathcal{N}\left(\sum_{i=1}^{N} \alpha_i k(\boldsymbol{x}_n, \boldsymbol{x}_i) + b,\, \sigma^2 \right), \\ [b, \boldsymbol{\alpha}]^\top &\sim \mathcal{N}(\mu, \Sigma), \\ \Sigma^{-1} &= \begin{bmatrix} \epsilon & \gamma^{-1}\mathbf{1}^\top \\ \gamma^{-1}\mathbf{1} & \mathbf{1}\mathbf{1}^\top + 2\gamma^{-1}\Omega + \gamma^{-2} I \end{bmatrix}, \quad \mu = \gamma^{-1}\Sigma \begin{bmatrix} 0 \\ y \end{bmatrix}. \end{aligned}$$

[9] Mesquita et al, LS-SVR as Bayesian RBF networks, IEEE TNNLS, 2020.

LS-SVR as Bayesian RBF networks

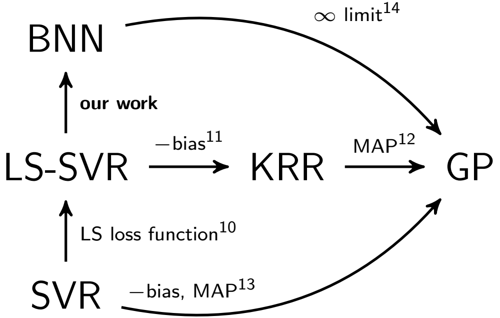


Figure 7: Relations between the regression learning models considered in the present study. Some relevant references are highlighted in each edge.

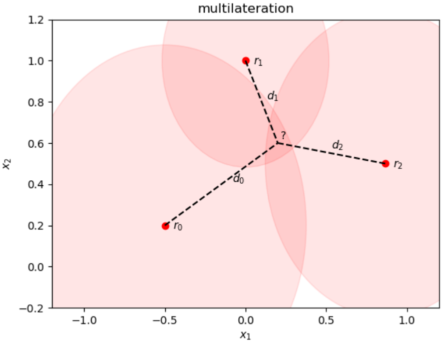
---

[10] Saunders et al. (1998), Suykens et al. (2002)
[11] Saunders et al. (1998), Cristianini et al. (2000)
[12] Rasmussen and Williams (2006)
[13] Gao et al. (2002), Chu et al. (2004)
[14] Neal, (1995), Williams, (1997)

Agenda

Multilateration

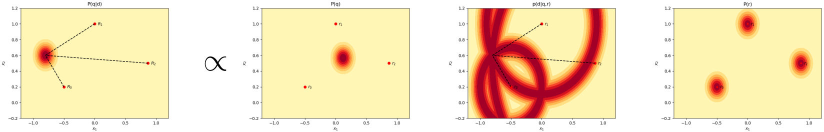
- Multilateration is a general technique to determine the position of an object based on measures from other known objects.
  - $\rightarrow$ Input:
    - a set of K reference points $\boldsymbol{r}_k \in \mathbb{R}^D$;
    - the estimated distances $\boldsymbol{d} \in \mathbb{R}^K$ from the query point $\boldsymbol{q} \in \mathbb{R}^D$ to the reference points
  - $\rightarrow$ Output: the position of the query point $\boldsymbol{q}$.



Bayesian Multilateration[15]

- We make the following assumptions:
    - $p(\boldsymbol{q})$: A normal prior with mean given by the mean of the reference points.
    - $p(\boldsymbol{r}_k)$: A normal distribution with mean given by the measured position of the reference point.
    - $p(d_k|\boldsymbol{q}, \boldsymbol{r}_k)$: A Nakagami likelihood with the mode at the measured distance.
- Bayesian Multilateration formulation:

$$p(\boldsymbol{q}|\boldsymbol{d}) \propto p(\boldsymbol{q}) \prod_{k=1}^{K} \int p(d_k|\boldsymbol{q}, \boldsymbol{r}_k) p(\boldsymbol{r}_k) \mathrm{d}\boldsymbol{r}_k.$$



[15]Work in progress by Alisson Alencar.

Agenda

Trajectory anomaly detection using NFs[16]

Problem Statement

- Let $\mathcal{T} = \{\mathbf{T}_n\}_{n=1}^{N}$ be a set of trajectories such that
$$\mathbf{T}_m \triangleq \left(\boldsymbol{q}_1^{(m)}, \boldsymbol{q}_2^{(m)}, \cdots, \boldsymbol{q}_l^{(m)}, \cdots, \boldsymbol{q}_{L_m}^{(m)}\right),$$
where $\boldsymbol{q}_l^{(m)} = \left(q_{l,1}^{(m)}, q_{l,2}^{(m)}, q_{l,3}^{(m)}\right)$ is a *location point*.
- We want to create a density estimation model to evaluate the anomaly degree of any given trajectory.

[16] Dias, M. L. D., *et. al.*; **Anomaly Detection in Trajectory Data with Normalizing Flows**, 2019

Trajectory anomaly detection using NFs[16]

Problem Statement

- Let $\mathcal{T} = \{\mathbf{T}_n\}_{n=1}^{N}$ be a set of trajectories such that
$$\mathbf{T}_m \triangleq \left(\boldsymbol{q}_1^{(m)}, \boldsymbol{q}_2^{(m)}, \cdots, \boldsymbol{q}_l^{(m)}, \cdots, \boldsymbol{q}_{L_m}^{(m)}\right),$$
where $\boldsymbol{q}_l^{(m)} = \left(q_{l,1}^{(m)}, q_{l,2}^{(m)}, q_{l,3}^{(m)}\right)$ is a *location point*.
- We want to create a density estimation model to evaluate the anomaly degree of any given trajectory.

Proposed methodology

- Segment-based anomaly detection with Normalizing Flows.
- Trajectory segments:
$$\mathbf{S}_i^{(m)} \triangleq \left(\boldsymbol{q}_i^{(m)}, \boldsymbol{q}_{i+1}^{(m)}, \cdots, \boldsymbol{q}_{i+W}^{(m)}\right).$$
where $W \leqslant L_m$ and $1 \leqslant i \leqslant L_m - W + 1$.

[16] Dias, M. L. D., *et. al.*; **Anomaly Detection in Trajectory Data with Normalizing Flows**, 2019

Trajectory anomaly detection using NFs

Aggregated anomaly detection with NFs (GRADINGS)

1. Create trajectory segments:

$$\mathcal{X} = \bigcup_{m=1}^{M} \left\{ \boldsymbol{x}_n = \delta\left(\mathbf{S}_i^{(m)}\right) \Big|_{i=1}^{L_m - W + 1} \right\},$$

2. Estimate distribution of trajectory segments using Normalizing Flows:

$$\alpha\left(\mathbf{S}_i^{(m)}\right) = -\log p\left(\delta\left(\mathbf{S}_i^{(m)}\right)\right).$$

3. Aggregate anomaly scores:

$$A\left(\mathbf{T}_m\right) = \varphi\left(\left\{\alpha\left(\mathbf{S}_i^{(m)}\right)\right\}_{i=1}^{L_m - W + 1}\right).$$

Trajectory anomaly detection using NFs

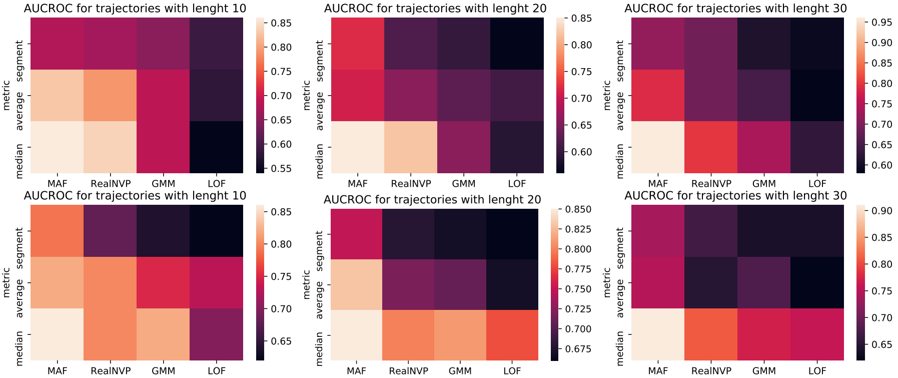


Figure 9: AUCROC for (top row) CAR $\times$ BUS (bottom row) BUS $\times$ CAR.

Trajectory anomaly detection using NFs

Table 1: FP rates obtained when we fix a true positive rate of 80%.

			Model			
Scenario	*Variant*	*Length*	MAF	RealNVP	GMM	LOF
CAR $\times$ BUS	segment	10	**0.423**	0.643	0.698	0.719
		20	**0.498**	0.640	0.653	0.688
		30	**0.608**	0.652	0.699	0.727
	average	10	0.342	**0.335**	0.376	0.465
		20	**0.272**	0.435	0.500	0.550
		30	**0.361**	0.577	0.556	0.622
	median	10	**0.245**	0.375	0.308	0.481
		20	**0.247**	0.335	0.353	0.419
		30	<u>**0.201**</u>	0.361	0.315	0.462
BUS $\times$ CAR	segment	10	0.603	**0.592**	0.597	0.684
		20	**0.510**	0.633	0.682	0.692
		30	**0.489**	0.517	0.631	0.689
	average	10	**0.252**	0.310	0.482	0.712
		20	**0.529**	0.601	0.635	0.704
		30	**0.311**	0.555	0.622	0.732
	median	10	**0.226**	0.330	0.761	0.771
		20	**0.190**	0.294	0.744	0.819
		30	<u>**0.055**</u>	0.328	0.564	0.747

Trajectory anomaly detection using NFs

Ongoing work

- Expand current work for general multivariate time-series
  - $\rightarrow$ Motion Glow[17] + Flow Gaussian Mixture Model[18].
  - $\rightarrow$ Automatic anomaly threshold selection using Extreme Value Theory (EVT)[19].

[17] Henter et al., 2020
[18] Izmailov et al., 2019
[19] Siffer et al., 2017

Agenda

Concluding Remarks

- Probabilistic ML presents plenty of application possibilities.
- Several model extensions and theoretical aspects to be pursued.
- Great for small(ish) data.
- We still need:
  - → more researchers trained on probabilistic modeling.
  - → more seamless integration with DL methods.
  - → faster 'notebook draft to deployed solution' pipeline.
  - → time to revisit/apply previous ideas (versus 'trendy' topics).
- Collaborative efforts are always welcomed!

Questions?

César Lincoln C. Mattos
@cesarlincoln
cesarlincoln@dc.ufc.br