

Inversion of a costly multivariate function in presence of categorical variables. Part I.

O. Roustant^a

Joint work with J. Cuesta-Ramirez (CEA), A. Glière (CEA),
C. Durantin (CEA), G. Perrin (CEA) and R. Le Riche (EMSE)

^a Mines Saint-Étienne (EMSE)

AIP, Grenoble, 2019 July

Outline

- 1 **Background on metamodeling and Bayesian optimization**
- 2 **Gaussian process regression with mixed inputs**
 - Building a kernel by combining 1-dimensional ones
 - Relaxation to continuous inputs with latent variables
- 3 **Bayesian optimization for mixed inputs, application to inversion**
 - → see Jhouben' slides

Outline

- 1 **Background on metamodeling and Bayesian optimization**
- 2 Gaussian process regression with mixed inputs
- 3 Bayesian optimization for mixed inputs, application to inversion

Metamodeling – Computer experiments

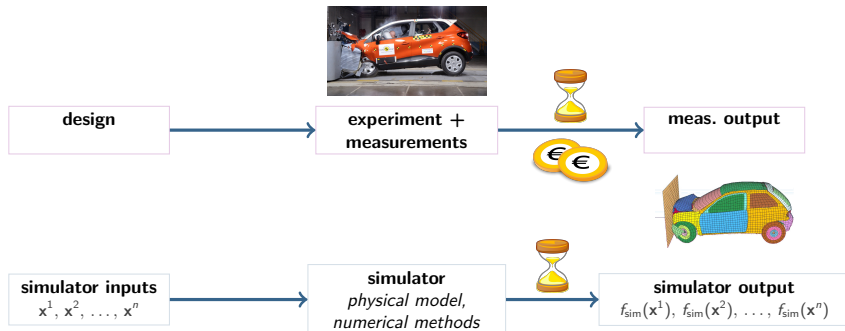
design



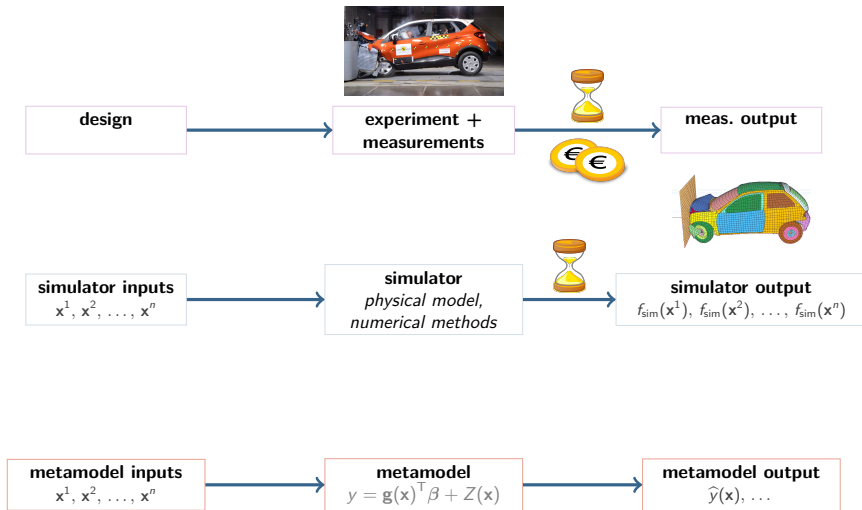
Metamodeling – Computer experiments



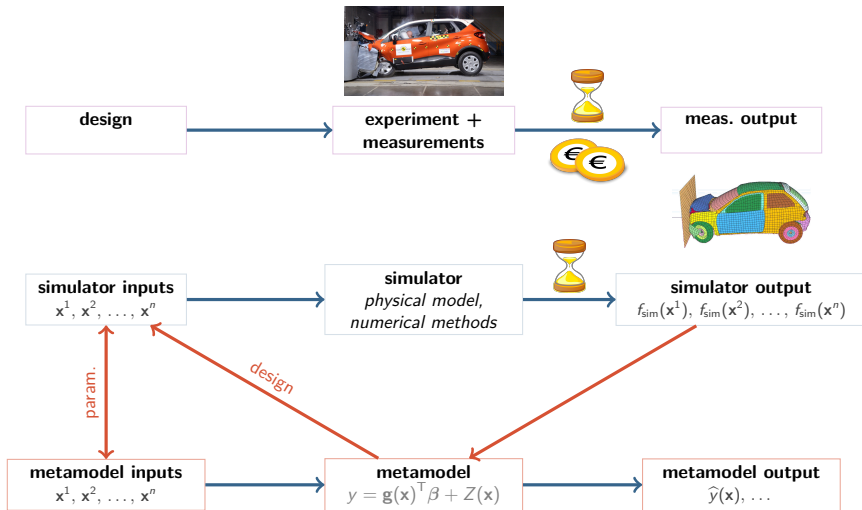
Metamodeling – Computer experiments



Metamodeling – Computer experiments



Metamodeling – Computer experiments



Gaussian processes

Gaussian processes are stochastic processes (or random fields) s.t. every finite dimensional distribution is Gaussian. → **Parameterized by two functions**

$$Z \sim GP(\underbrace{m(\mathbf{x})}_{\text{trend}}, \underbrace{k(\mathbf{x}, \mathbf{x}')}_{\text{kernel}})$$

- The trend can be any function.
- The kernel is **positive semidefinite** :

$$\forall n, \alpha_1, \dots, \alpha_n, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}, \quad \sum_{i=1}^n \alpha_i \alpha_j k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \geq 0.$$

It contains the **spatial dependence**.

Gaussian processes and approximation / interpolation

GPs conditional distributions are Gaussian (analytical expressions)

- The conditional mean is linear in the conditioner
- The conditional variance does not depend on it!
→ very useful for adding new points in sequential strategies

In the background, Z is conditioned on $Z(\mathbf{x}^{(1)}) = z_1, \dots, Z(\mathbf{x}^{(n)}) = z_n$.

Playing with kernels

A lot of flexibility can be obtained with kernels !

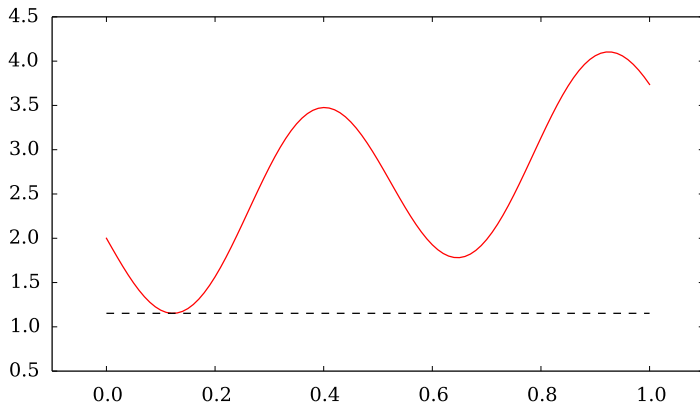
Building a kernel from other ones (basic examples)

Sum, tensor sum	$k_1 + k_2, k_1 \oplus k_2$
Product, tensor product	$k_1 \times k_2, k_1 \otimes k_2$
ANOVA	$(1 + k_1) \otimes (1 + k_2)$
Warping	$k(\mathbf{x}, \mathbf{x}') = k_1(f(\mathbf{x}), f(\mathbf{x}'))$
...	...

See examples in [[Rasmussen and Williams, 2006](#)]... and in this talk !

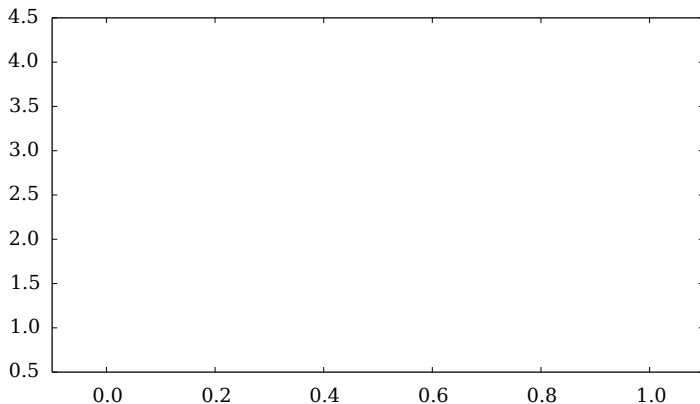
GP-based optimization

How to find the global minimum of a function... when each evaluation is costly?



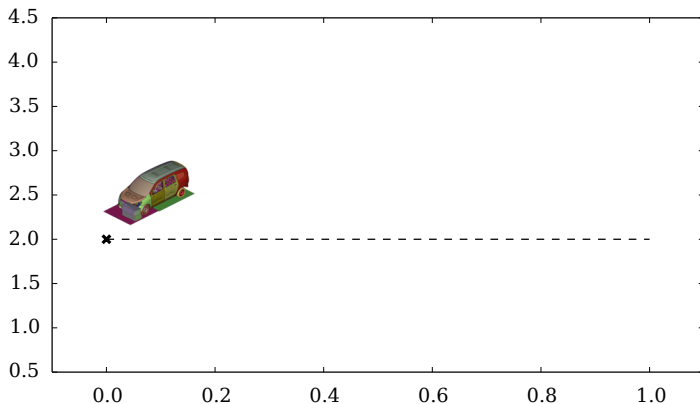
GP-based optimization

How to find the global minimum of a function... when each evaluation is costly ?



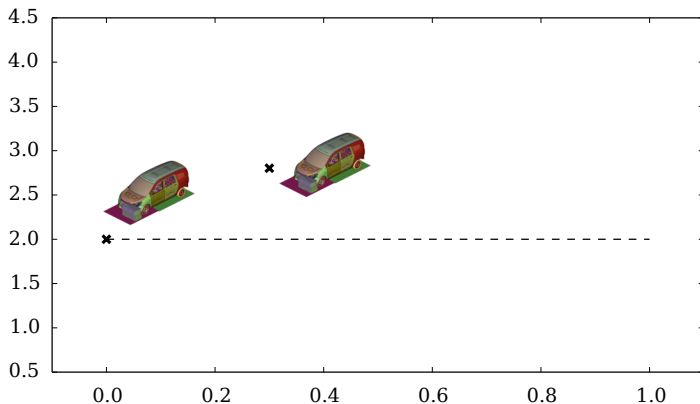
GP-based optimization

How to find the global minimum of a function... when each evaluation is costly?



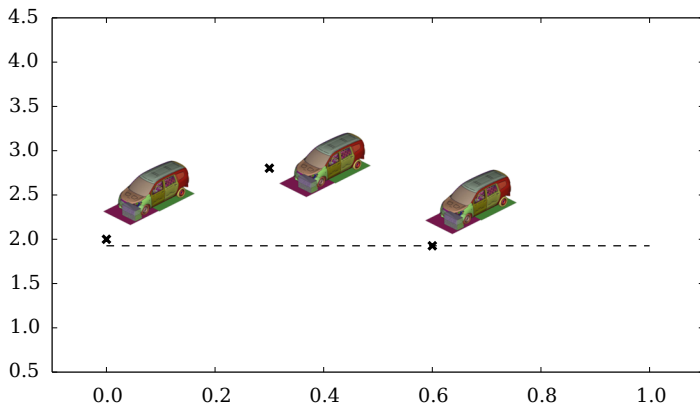
GP-based optimization

How to find the global minimum of a function... when each evaluation is costly?



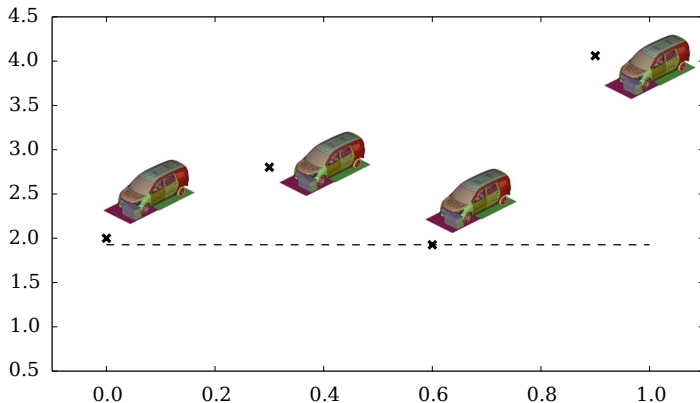
GP-based optimization

How to find the global minimum of a function... when each evaluation is costly?



GP-based optimization

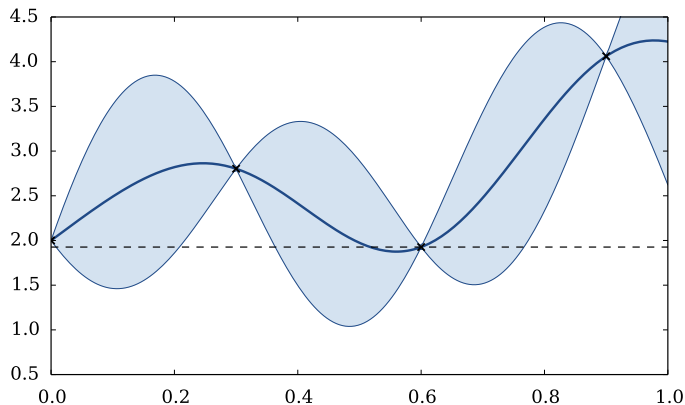
How to find the global minimum of a function... when each evaluation is costly?



GP-based optimization

A solution : **GP-based (or "Bayesian") optimization** [Moćkus, 1975, Jones et al., 1998]

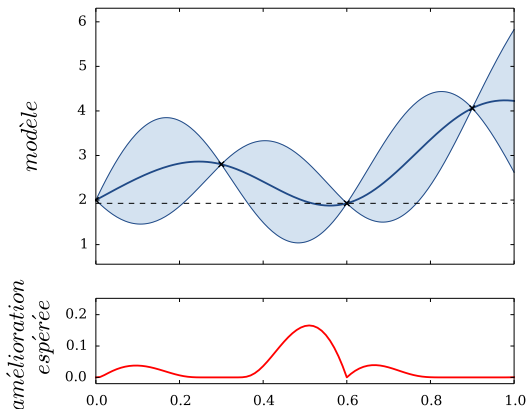
First ingredient : a GP model Y



GP-based optimization

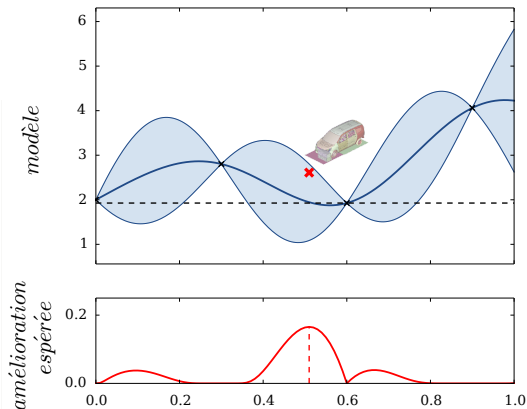
Second ingredient : an **easy-to-compute** criterion **accounting for uncertainty at unknown regions**, e.g. here “expected improvement”

$$EI(x) = E([f_0 - Y(x)]^+ | Y(x_1), \dots, Y(x_n)) \quad f_0 : \text{current minimum}$$



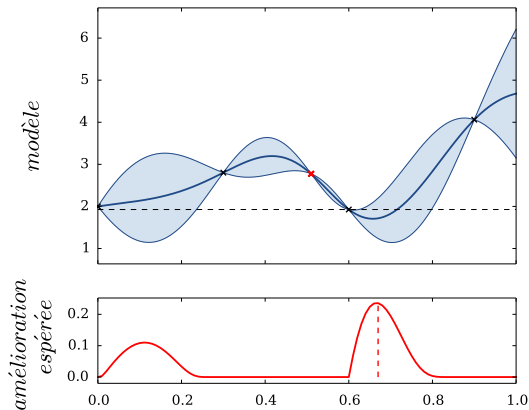
GP-based optimization

The algorithm (here “EGO”) : (1) Find the next point by maximizing the criterion
 → (2) Evaluate the function → (3) Update the GP model ↑



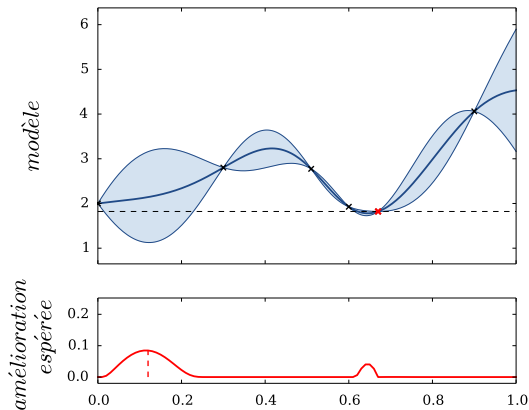
GP-based optimization

Iteration 2 :



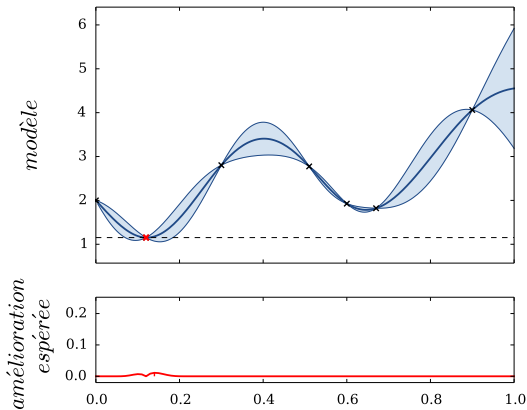
GP-based optimization

Iteration 3 :



GP-based optimization

Theory shows that **EGO algorithm provides a dense sequence of points**, up to a slight condition on the kernel used for GPs [[Vazquez and Bect, 2010](#)].



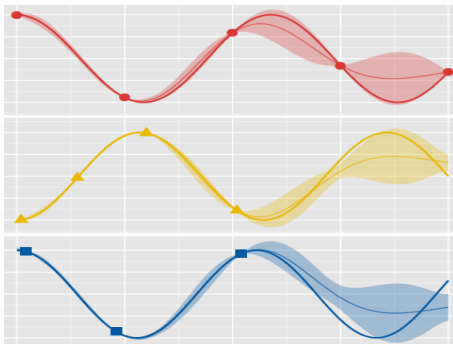
Outline

- 1 Background on metamodeling and Bayesian optimization
- 2 Gaussian process regression with mixed inputs**
 - Building a kernel by combining 1-dimensional ones
 - Relaxation to continuous inputs with latent variables
- 3 Bayesian optimization for mixed inputs, application to inversion

GP interpretation when no distance is available

A GP for $(x, u) \in [0, 1] \times \{"red", "yellow", "blue"\}$ can be defined with :

- a kernel on $[0, 1]$, i.e. a **covariance function**
- a kernel on $\{"red", "yellow", "blue"\}$, i.e. a **covariance matrix**
- a valid operation between them, such as $*$, $+$, ...



Example : $\text{Cov}(Y(x, "blue"), Y(x', "red")) = k(x, x') \times 0.8$

One way to construct kernels for mixed inputs

What is a kernel for one categorical variable u on $\{1, \dots, m\}$?

A positive semidefinite matrix \mathbf{T} of size m

One way to construct kernels for mixed inputs

What is a kernel for one categorical variable u on $\{1, \dots, m\}$?

A positive semidefinite matrix \mathbf{T} of size m

Combining 1D kernels for $\mathbf{w} = (\mathbf{x}, \mathbf{u})$

Examples of valid operations :

$$\text{(Product)} \quad k(\mathbf{w}, \mathbf{w}') = k_{\text{cont}}(\mathbf{x}, \mathbf{x}') k_{\text{cat}}(\mathbf{u}, \mathbf{u}')$$

$$\text{(Sum)} \quad k(\mathbf{w}, \mathbf{w}') = k_{\text{cont}}(\mathbf{x}, \mathbf{x}') + k_{\text{cat}}(\mathbf{u}, \mathbf{u}')$$

$$\text{(ANOVA)} \quad k(\mathbf{w}, \mathbf{w}') = (1 + k_{\text{cont}}(\mathbf{x}, \mathbf{x}'))(1 + k_{\text{cat}}(\mathbf{u}, \mathbf{u}'))$$

One way to construct kernels for mixed inputs

What is a kernel for one categorical variable u on $\{1, \dots, m\}$?

A positive semidefinite matrix \mathbf{T} of size m

Combining 1D kernels for $\mathbf{w} = (\mathbf{x}, \mathbf{u})$

Examples of valid operations :

$$\text{(Product)} \quad k(\mathbf{w}, \mathbf{w}') = k_{\text{cont}}(\mathbf{x}, \mathbf{x}') k_{\text{cat}}(\mathbf{u}, \mathbf{u}')$$

$$\text{(Sum)} \quad k(\mathbf{w}, \mathbf{w}') = k_{\text{cont}}(\mathbf{x}, \mathbf{x}') + k_{\text{cat}}(\mathbf{u}, \mathbf{u}')$$

$$\text{(ANOVA)} \quad k(\mathbf{w}, \mathbf{w}') = (1 + k_{\text{cont}}(\mathbf{x}, \mathbf{x}'))(1 + k_{\text{cat}}(\mathbf{u}, \mathbf{u}'))$$

Notice $*$ one of them. Examples of valid kernels for \mathbf{w} :

$$k(\mathbf{w}, \mathbf{w}') = k_{\text{cont}}^1(x_1, x'_1) * \dots * k_{\text{cont}}^I(x_I, x'_I) * [\mathbf{T}_1]_{u_1, u'_1} * \dots * [\mathbf{T}_J]_{u_J, u'_J}$$

One way to construct kernels for mixed inputs

What is a kernel for one categorical variable u on $\{1, \dots, m\}$?

A positive semidefinite matrix \mathbf{T} of size m

Combining 1D kernels for $\mathbf{w} = (\mathbf{x}, \mathbf{u})$

Examples of valid operations :

$$(\text{Product}) \quad k(\mathbf{w}, \mathbf{w}') = k_{\text{cont}}(\mathbf{x}, \mathbf{x}') k_{\text{cat}}(\mathbf{u}, \mathbf{u}')$$

$$(\text{Sum}) \quad k(\mathbf{w}, \mathbf{w}') = k_{\text{cont}}(\mathbf{x}, \mathbf{x}') + k_{\text{cat}}(\mathbf{u}, \mathbf{u}')$$

$$(\text{ANOVA}) \quad k(\mathbf{w}, \mathbf{w}') = (1 + k_{\text{cont}}(\mathbf{x}, \mathbf{x}'))(1 + k_{\text{cat}}(\mathbf{u}, \mathbf{u}'))$$

Notice $*$ one of them. Examples of valid kernels for \mathbf{w} :

$$k(\mathbf{w}, \mathbf{w}') = k_{\text{cont}}^1(x_1, x'_1) * \dots * k_{\text{cont}}^I(x_I, x'_I) * [\mathbf{T}_1]_{u_1, u'_1} * \dots * [\mathbf{T}_J]_{u_J, u'_J}$$

Not the most general way, but recovers the usual models of the literature.

→ Alternatives : Use a d-dim. continuous kernel, use $*_i, *_j$, and so on...

Kernels for ordinal variables

Warping ([Qian et al., 2007])

- When the levels of u are ordered : $1 \leq 2 \leq \dots \leq m$, define :

$$T_{\ell, \ell'} = k_{\text{cont}}(F(\ell), F(\ell')), \quad \ell, \ell' = 1, \dots, m.$$

where k_{cont} is a 1-dim. continuous kernel, and $F : \{1, \dots, m\} \rightarrow \mathbb{R}$ is \uparrow .

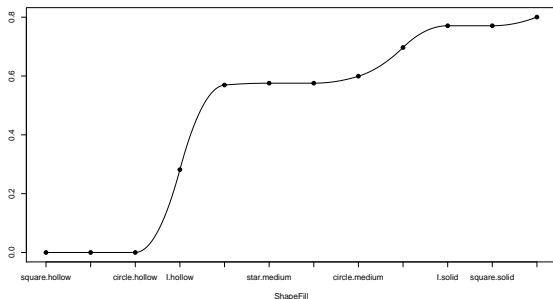


Figure – An example of warping as a spline of degree 2, coming soon on kergp.

Kernels for nominal variables

- **General**

- ▶ Spectral param. $\mathbf{T} = \mathbf{PDP}^\top$
- ▶ Spherical param. $\mathbf{T} = \mathbf{LL}^\top$ (*correlation case*)

- **Compound symmetry** ([Pinheiro and Bates, 2009])

$$T_{\ell, \ell'} = \begin{cases} v & \text{if } \ell = \ell' \\ c & \text{if } \ell \neq \ell' \end{cases}$$

- **Group kernels**, such as ([Qian et al., 2007, Roustant et al., 2018]) :

$$T_{\ell, \ell'} = \begin{cases} v_g & \text{if } \ell = \ell' \\ c_{g(\ell), g(\ell')} & \text{if } \ell \neq \ell' \end{cases}$$

- **Low “rank” approaches** ([Rapisarda et al., 2007], [Zhang et al., 2018])

Low-rank $\mathbf{T} = \mathbf{FF}^\top$, with $\mathbf{F} : L \times q$

Latent-variable : $T_{\ell, \ell'} = k_{\text{cont}}(F(\ell), F(\ell'))$, with $F : \{1, \dots, m\} \rightarrow \mathbb{R}^q$

Latent variables and low-rank approaches

Interpretation of latent variable kernels ([Zhang et al., 2018])

The underlying Gaussian process for a latent variable kernel is

$$Z(u) = Y(F_1(u), \dots, F_q(u))$$

where each F_i is a mapping from $\{1, \dots, m\} \rightarrow \mathbb{R}$, called “**latent variable**”.

- Example : u : type of lubricant, F_1 : viscosity, F_2 : boiling point, ...
- Only F_i 's values at $1, \dots, m$ are used :
the kernel is parameterized by the $F_i(\ell)$, $\ell = 1, \dots, m$, $i = 1, \dots, q$.
[up to simplifications, e.g. $F_1(1) = 0$]

Latent variables and low-rank approaches

Interpretation of latent variable kernels ([Zhang et al., 2018])

The underlying Gaussian process for a latent variable kernel is

$$Z(u) = Y(F_1(u), \dots, F_q(u))$$

where each F_i is a mapping from $\{1, \dots, m\} \rightarrow \mathbb{R}$, called “**latent variable**”.

- Example : u : type of lubricant, F_1 : viscosity, F_2 : boiling point, ...
- Only F_i 's values at $1, \dots, m$ are used :
the kernel is parameterized by the $F_i(\ell)$, $\ell = 1, \dots, m$, $i = 1, \dots, q$.
[up to simplifications, e.g. $F_1(1) = 0$]

Links with low-rank kernels

If $k_{\text{cont}}(\mathbf{f}, \mathbf{f}') = \langle \mathbf{f}, \mathbf{f}' \rangle$ is the dot product on \mathbb{R}^q , then the latent variable kernel is a low-rank kernel $\mathbf{T} = \mathbf{F}\mathbf{F}^\top$, with $F_{\ell,i} = F_i(\ell)$, for $\ell = 1, \dots, m$, $i = 1, \dots, q$.

→ Latent variables kernels are extending low-rank kernels for general k_{cont}

Outline

- 1 Background on metamodeling and Bayesian optimization
- 2 Gaussian process regression with mixed inputs
- 3 **Bayesian optimization for mixed inputs, application to inversion**
 - → see Jhouben' slides

Acknowledgements

- This work has been funded by the **Chair in Applied Mathematics OQUAIDO**, gathering partners in technological research (BRGM, CEA, IFPEN, IRSN, Safran, Storengy) and academia (CNRS, Ecole Centrale de Lyon, Mines Saint-Etienne, University of Grenoble, University of Nice, University of Toulouse) around advanced methods for Computer Experiments.
- Many thanks to several slides co-writers : M. Binois, Y. Deville, N. Durrande, D. Ginsbourger, R. Le Riche, A. Lopez Lopéra, E. Padonou.



Références I



Jones, D. R., Schonlau, M., and Welch, W. J. (1998).
Efficient global optimization of expensive black-box functions.
Journal of Global Optimization, 13(4) :455–492.



Močkus, J. (1975).
On Bayesian methods for seeking the extremum.
In Marchuk, G. I., editor, *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*, pages 400–404, Berlin, Heidelberg. Springer Berlin Heidelberg.



Pinheiro, J. and Bates, D. (2009).
Mixed-effects models in S and S-PLUS.
Statistics and Computing. Springer New York.



Qian, P. Z. G., Wu, H. C. F., and Wu, J. (2007).
Gaussian process models for computer experiments with qualitative and quantitative factors.
Technical report, Department of statistics, University of Wisconsin.



Rapisarda, F., Brigo, D., and Mercurio, F. (2007).
Parameterizing correlations : a geometric interpretation.
IMA Journal of Management Mathematics, 18(1) :55–73.

Références II



Rasmussen, C. E. and Williams, C. K. (2006).

Gaussian processes for machine learning.

the MIT Press.



Roustant, O., Padonou, E., Deville, Y., Clément, A., Perrin, G., Giorla, J., and Wynn, H. (2018).

Group kernels for Gaussian process metamodels with categorical inputs.

ArXiv e-prints.



Vazquez, E. and Bect, J. (2010).

Convergence properties of the expected improvement algorithm with fixed mean and covariance functions.

Journal of Statistical Planning and Inference, 140(11) :3088 – 3095.



Zhang, Y., Tao, S., Chen, W., and Apley, D. (2018).

A latent variable approach to Gaussian process modeling with qualitative and quantitative factors.

Technical report, Northwestern University.



Inversion of a costly multivariate function in presence of categorical variables - Part II

Jhouben Ramirez^{1,3}

Joint work with: O. Roustant³, A. Glière¹, R. Le Riche³, C. Durantin², G. Perrin²

AIP 2019 - Grenoble | July, 2019

¹ CEA, LETI, Grenoble, France ² CEA, DAM, Paris, France

³ Mathématiques appliquées, Mines, Saint-Etienne, France

- 1 Bayesian optimization for mixed inputs, application to inversion
 - Relaxation to continuous inputs : LV-EGO algorithm
 - Other suitable algorithms
- 2 Computer Experiments
 - Constructing a test-case
 - Desing of Experiment
 - Benchmark Results
- 3 Conclusions and Perspectives

Outline for section 1

- 1 Bayesian optimization for mixed inputs, application to inversion
 - Relaxation to continuous inputs : LV-EGO algorithm
 - Other suitable algorithms
- 2 Computer Experiments
 - Constructing a test-case
 - Desing of Experiment
 - Benchmark Results
- 3 Conclusions and Perspectives

LV-EGO algorithm

Let $u \rightarrow F(u) = (F_1(u), \dots, F_q(u))$, be the latent variable mapping

- 1: **Generate** the initial design (DoE) for (x, u)
- 2: **Estimate** F parameters by MLE from the DoE
- 3: **while** budget is not consumed **do**
- 4: **Perform** EGO in the *image space* $(x, F(u)) \rightarrow (x^*, f^*)$.
- 5: **Recover** the *pre-image* component u^* as :

$$u^* = \underset{u}{\operatorname{argmin}} \quad |El(x^*, f^*) - El(x^*, F(u))|$$
- 6: **Update** the DoE : (x^*, f^*) (*best point*) and $(x^*, F(u^*))$ (*best feasible*) with output value $y(x^*, u^*)$.
- 7: **end while**
- 8: **Return** the best (x^*, u^*) over all the budget iterations.

Other suitable algorithms

Bayesian Optimization

- EGO as implemented in DiceOptim package
- Random forest (RF) surrogate, as implemented in mlrMBO package and RandomForest package

Non Bayesian

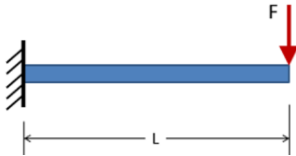
- Evolutionary Strategies (ES) as implemented in CEGO package

See ([Cauwet et al., 2019]) for more on mixed inputs algorithms!!

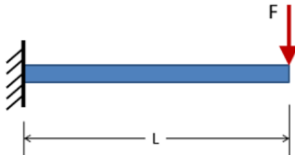
Outline for section 2

- 1 Bayesian optimization for mixed inputs, application to inversion
 - Relaxation to continuous inputs : LV-EGO algorithm
 - Other suitable algorithms
- 2 Computer Experiments
 - Constructing a test-case
 - Desing of Experiment
 - Benchmark Results
- 3 Conclusions and Perspectives

Beam bending test-case

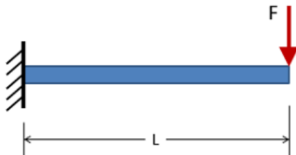


Beam bending test-case



$$y(L, S, \tilde{I}) = \frac{L^3}{3S^2\tilde{I}} + \alpha LS,$$

Beam bending test-case

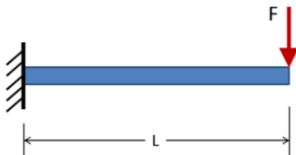


$$y(L, S, \tilde{I}) = \frac{L^3}{3S^2\tilde{I}} + \alpha LS,$$

where :

- \tilde{I} correspond to a *latent variable* $F(u)$,
- u is the categorical variable representing the beam feasible profiles (with 12 levels).

Beam bending test-case



$$y(L, S, \tilde{I}) = \frac{L^3}{3S^2\tilde{I}} + \alpha LS,$$

where :

- \tilde{I} correspond to a *latent variable* $F(u)$,
- u is the categorical variable representing the beam feasible profiles (with 12 levels).

In the following, we consider that the latent variable F is known or estimated

Beam bending test-case

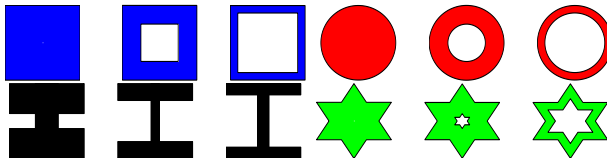


FIGURE – Representation of the values of the categorical variable u .

Beam bending test-case

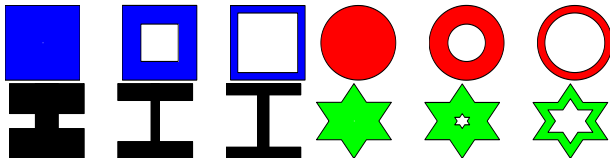


FIGURE – Representation of the values of the categorical variable u .

As an inverse problem :

For a fixed y_0 , design (L^*, S^*, u^*) is obtained from :

$$(L^*, S^*, u^*) \doteq \underset{L, S, u}{\operatorname{argmin}} |y_0 - y(L, S, u)|$$

Design of Experiment

	GP	RF
EGO on the mixed space (x, u)		X
LV-EGO : EGO on the continuous space $(x, F(u))$, with F estimated	X	
TLV-EGO : EGO on the continuous space $(x, F(u))$, with F known	X	X

Bayesian Optimization Performance Comparison (TLV stands for True Latent Variable)

	GP	RF
EGO on the mixed space (x, u)		X
LV-EGO : EGO on the continuous space $(x, F(u))$, with F estimated	X	
TLV-EGO : EGO on the continuous space $(x, F(u))$, with F known	X	X

Bayesian Optimization Performance Comparison (TLV stands for True Latent Variable)

Benchmark Design

- Latin hypercube design (LHD) with 3 points value of the categorical variable, $n = 36$.
- Budget = 50.
- Performance comparison for 100 designs.

Analyzing Experiments - DoE #1

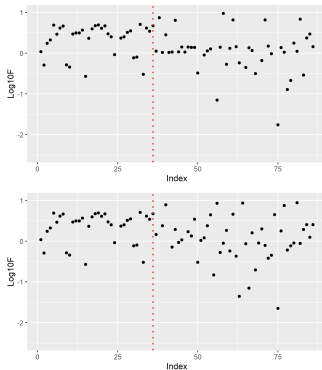


FIGURE – RF-EGO (up) and RF-TLV-EGO (bottom)

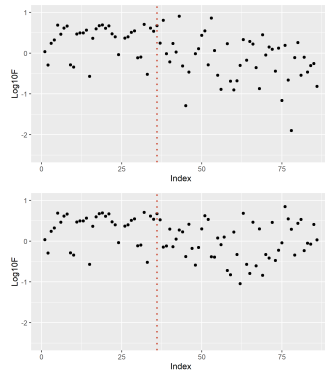


FIGURE – LV-EGO (up) and GP-TLV-EGO (bottom)

Analyzing Experiments - DoE #2

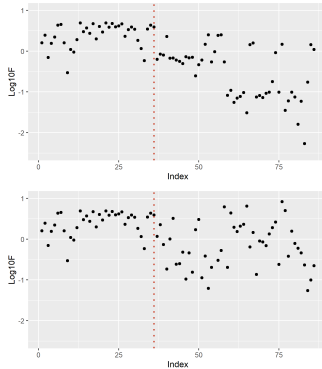


FIGURE – RF-EGO (up) and RF-TLV-EGO (bottom)

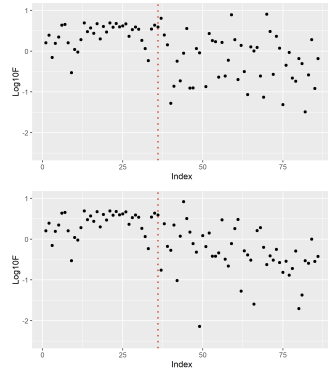


FIGURE – LV-EGO (up) and GP-TLV-EGO (bottom)

Analyzing Experiments - DoE #3

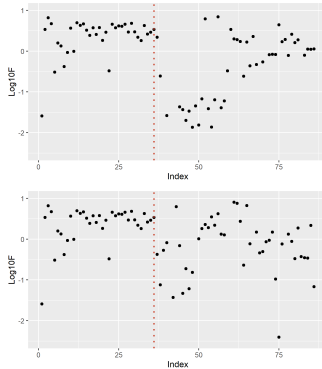


FIGURE – RF-EGO (up) and RF-TLV-EGO (bottom)

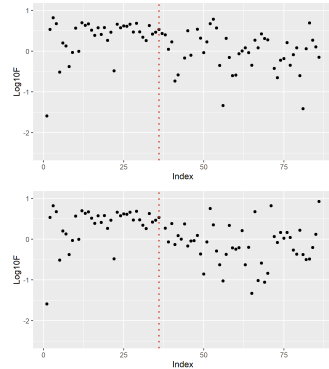
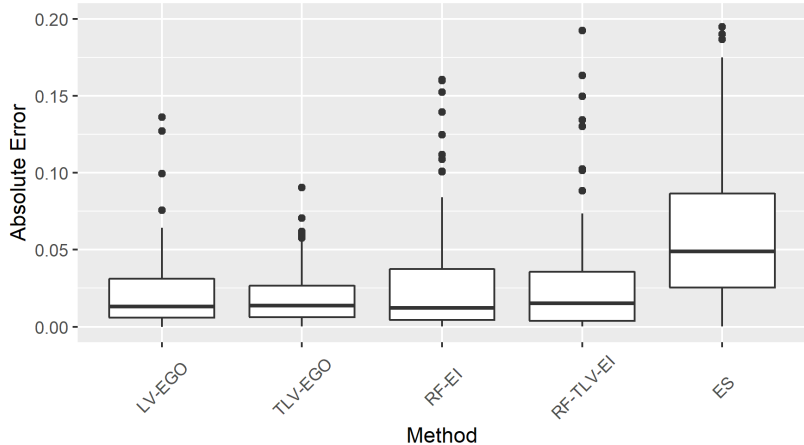


FIGURE – LV-EGO (up) and GP-TLV-EGO (bottom)

Results



Outline for section 3

- 1 Bayesian optimization for mixed inputs, application to inversion
 - Relaxation to continuous inputs : LV-EGO algorithm
 - Other suitable algorithms
- 2 Computer Experiments
 - Constructing a test-case
 - Desing of Experiment
 - Benchmark Results
- 3 Conclusions and Perspectives

Conclusions and perspectives

- Despite of being an infinite solutions inverse problem, all methods are capable to improve, most of the time the points in the original DoE.

Conclusions and perspectives

- Despite of being an infinite solutions inverse problem, all methods are capable to improve, most of the time the points in the original DoE.
- Finding the optimum in both pre image space (x, u) (random forest methods) and image space $(x, F(u))$ (EGO based methods) are suitable for finding a solution (mapping simplifies computing and arises to slightly better results).

Conclusions and perspectives

- Despite of being an infinite solutions inverse problem, all methods are capable to improve, most of the time the points in the original DoE.
- Finding the optimum in both pre image space (x, u) (random forest methods) and image space $(x, F(u))$ (EGO based methods) are suitable for finding a solution (mapping simplifies computing and arises to slightly better results).
- Learning a proper mapping, and trying to analyze its properties could lead to improve LV-EGO performance

Conclusions and perspectives

- Despite of being an infinite solutions inverse problem, all methods are capable to improve, most of the time the points in the original DoE.
- Finding the optimum in both pre image space (x, u) (random forest methods) and image space $(x, F(u))$ (EGO based methods) are suitable for finding a solution (mapping simplifies computing and arises to slightly better results).
- Learning a proper mapping, and trying to analyze its properties could lead to improve LV-EGO performance
- Perform experiments including correlations in the pre-image space are still required ([Roustant et al., 2018]).



Thanks for your attention

- Cauwet, M.-L., Le Riche, R., and Roustant, O. (2019). Mixed global optimization by algorithms composition : an empirical study with a focus on Bayesian approaches. In *30th European Conference on operational research*, Dublin, Ireland.
- Roustant, O., Padonou, E., Deville, Y., Clément, A., Perrin, G., Giorla, J., and Wynn, H. P. (2018). Group kernels for Gaussian process metamodels with categorical inputs. working paper or preprint.