R Package UP : Universal prediction distribution for meta-models

Malek BEN SALEM [1]

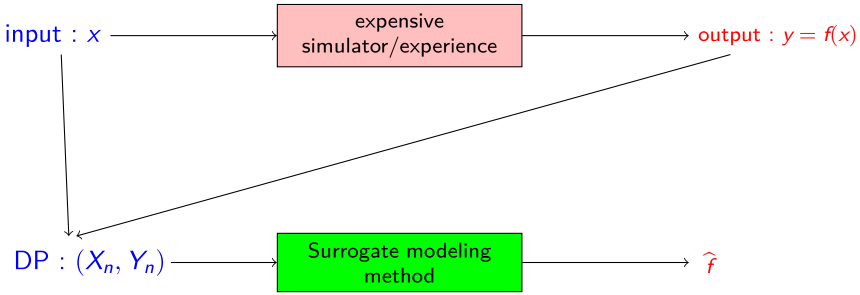
[1] Mines Saint-Etienne
Malek BEN SALEM is funded by a CIFRE grant, subsidized by the French National Association for Research and Technology (ANRT, CIFRE grant number 2014/1349).

29 June 2017

Context

input : x $\longrightarrow$ $\boxed{\text{expensive simulator/experience}}$ $\longrightarrow$ output : $y = f(x)$

Context

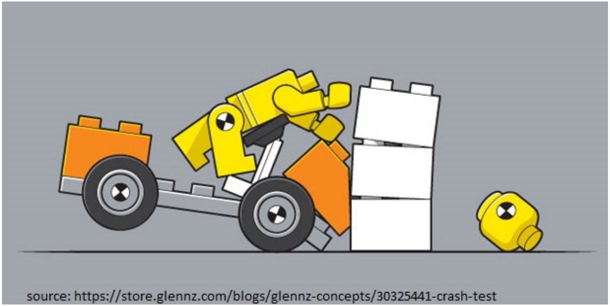

input : x
expensive simulator/experience
output : $y = f(x)$
DP : (X_n, Y_n)
Surrogate modeling method
$\widehat{f}$

Plan

Example : Engineering design

Crash test example

- Expensive crash test
- Expensive simulations



Example : a pizza experience

Objective

- A delicious pizza
- A model to predict the pizza deliciousness.

Figure – Objective

Example : a pizza experience

Objective

- A delicious pizza
- A model to predict the pizza deliciousness.

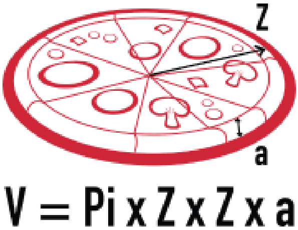
FIGURE – Objective

FIGURE – What I do

Example : a pizza experience

Notions

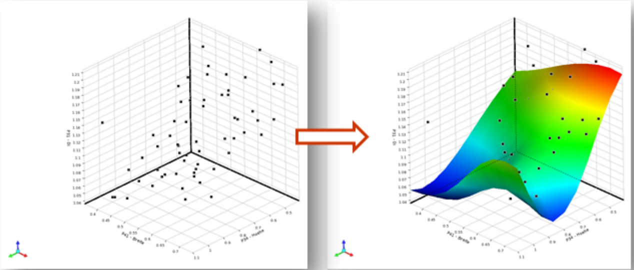
- Parameters : mass of flour $x_f \in [f, F]$, water $x_w \in [w, W]$, olive oil $x_o[o, O]$, salt $x_s \in [s, S]$, yeast $x_y \in [y, Y]$, Time to rise $T_r \in [r, R]$...
- Parametric design space $[f, F] \times [w, W] \times [o, O] \times [s, S] \times [y, Y] \times [r, R]$
- Quantities of interest : deliciousness, Final mass,..

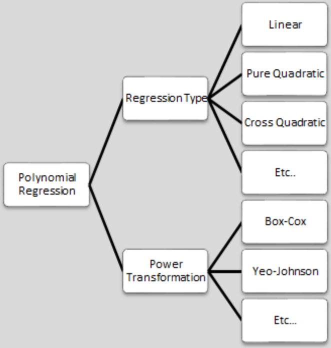


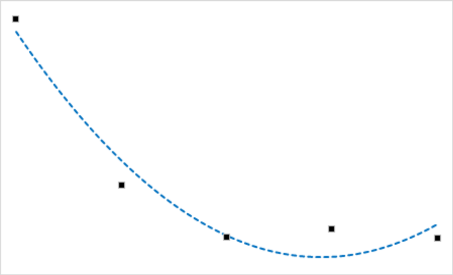
Context

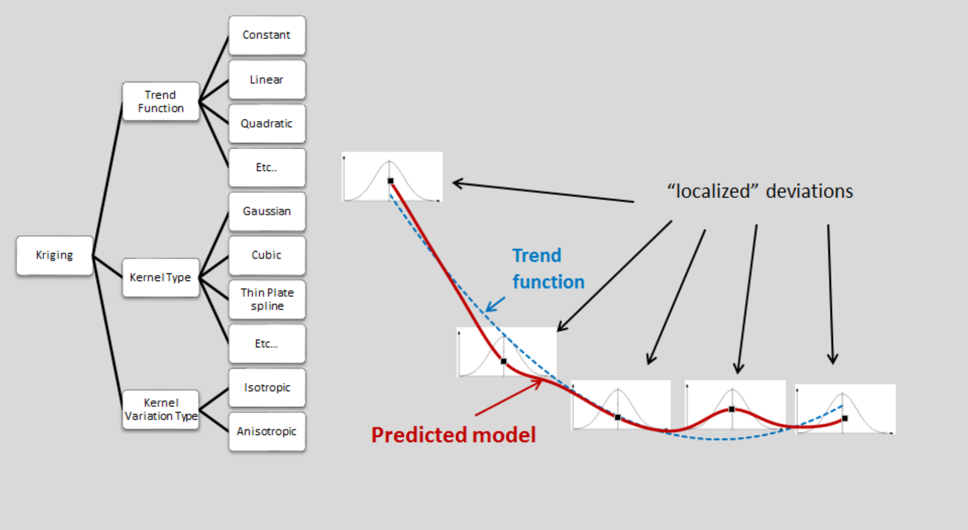
Notions

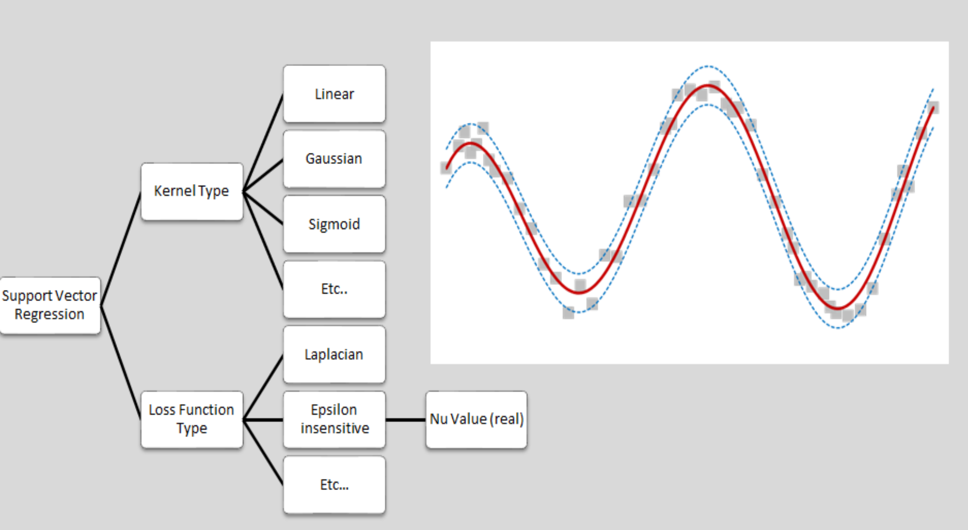
- Design points (DP) or training data : $\mathbf{Z_n} = (\mathbf{X_n}, \mathbf{Y_n})$, where $y_i = f(x_i)$, $\forall\ 1 \leq i \leq n$
- Response surface : $\widehat{f}$ (also called surrogate model or metamodel)
- Various response surfaces for various type of problems

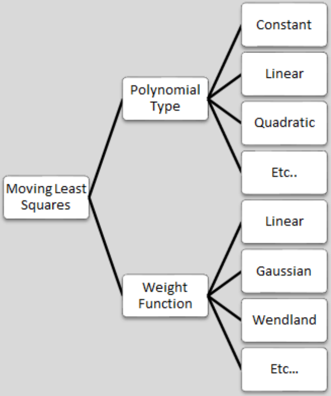


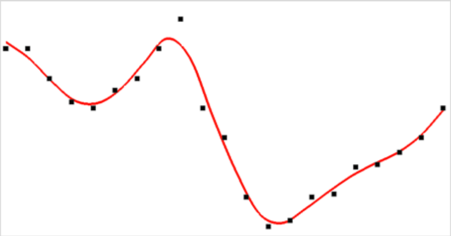

Polynomial Regression
Regression Type
Power Transformation
Linear
Pure Quadratic
Cross Quadratic
Etc..
Box-Cox
Yeo-Johnson
Etc...

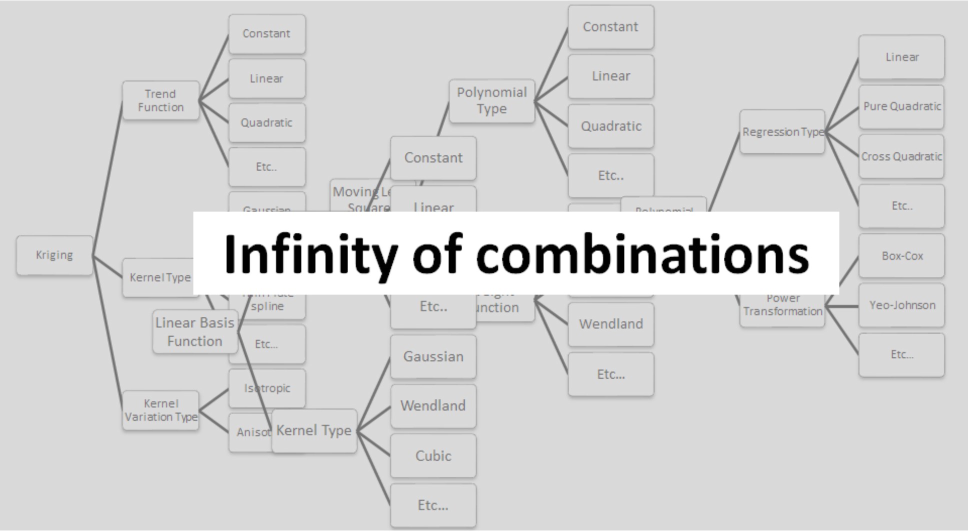



Kriging
Trend Function
Kernel Type
Kernel Variation Type
Constant
Linear
Quadratic
Etc..
Gaussian
Cubic
Thin Plate spline
Etc...
Isotropic
Anisotropic
"localized" deviations
Trend function
Predicted model


Support Vector Regression
Kernel Type
Linear
Gaussian
Sigmoid
Etc..
Loss Function Type
Laplacian
Epsilon insensitive
Nu Value (real)
Etc...


Moving Least Squares
Polynomial Type
Constant
Linear
Quadratic
Etc..
Weight Function
Linear
Gaussian
Wendland
Etc...




Infinity of combinations
Kriging
Trend Function
Constant
Linear
Quadratic
Etc..
Kernel Type
Linear Basis Function
Etc...
Kernel Variation Type
Isotropic
Kernel Type
Gaussian
Wendland
Cubic
Etc...
Constant
Etc..
Polynomial Type
Constant
Linear
Quadratic
Etc..
Wendland
Etc...
Regression Type
Linear
Pure Quadratic
Cross Quadratic
Etc..
Power Transformation
Box-Cox
Yeo-Johnson
Etc...

Gaussian Distribution

Gaussian Distribution

A random variable follows a Gaussian (or normal) distribution with mean μ and variance σ^2 if its probability density function is

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \text{ for } x \in R. \quad (1)$$

Multivariate Gaussian Distribution

The probability density function of a multivariate normal random vector writes :

$$f_Y(x) = \frac{1}{|2\pi\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^t\Sigma^{-1}(x-\mu)\right) \quad (2)$$

Gaussian Distribution

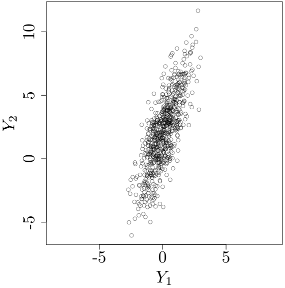


FIGURE – $\mu = (0, 2),\ \Sigma = \begin{pmatrix} 1 & 2 \\ 2 & 7 \end{pmatrix}$

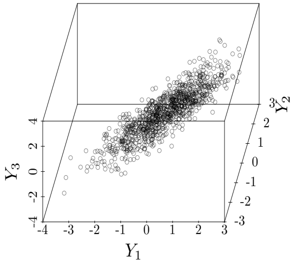


FIGURE – $\mu = (0, 0, 0),\ \Sigma = \begin{pmatrix} 1 & 0.4 & 0.8 \\ 0.4 & 0.5 & 0.3 \\ 0.8 & 0.3 & 1 \end{pmatrix}$

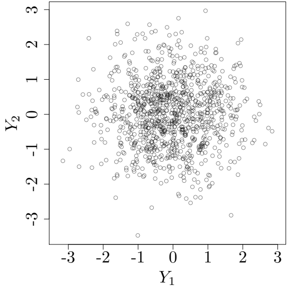


FIGURE – $\mu = (0, 0),\ \Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

Gaussian Process Regression

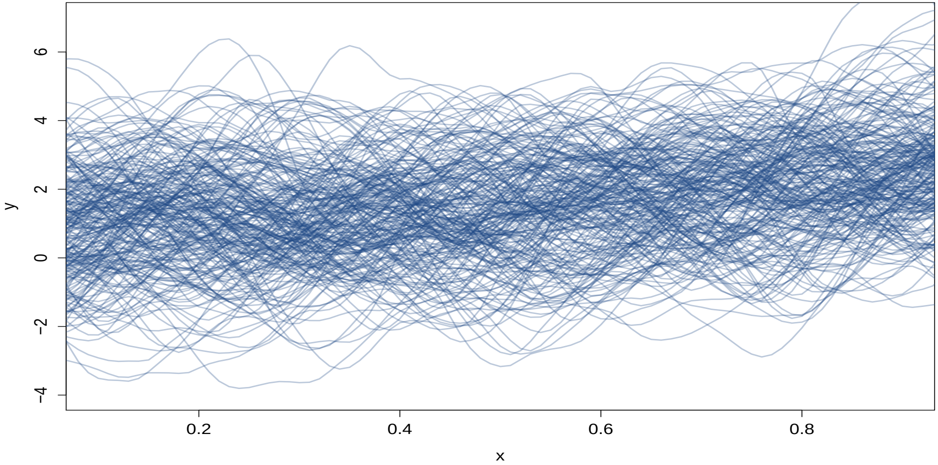
Conditional distribution of a Gaussian vector

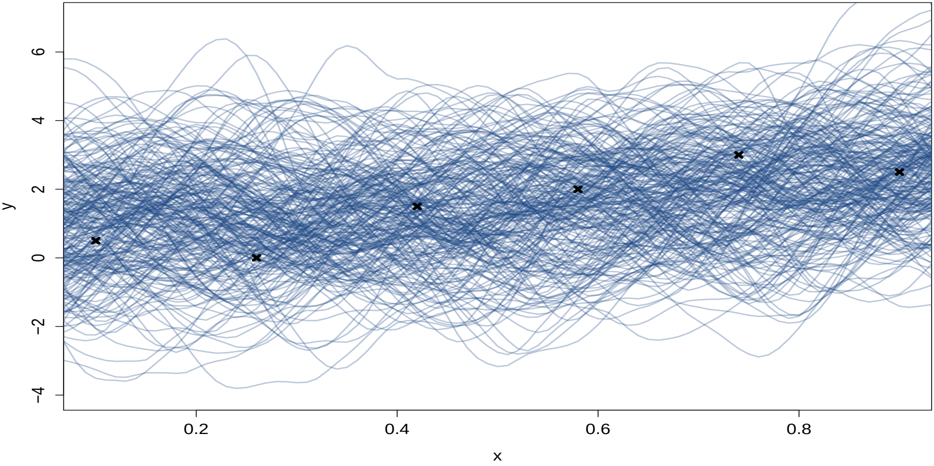
Let (Y_1, Y_2) be a Gaussian vector (Y_1 and Y_2 may both be vectors) with mean $\mu = (\mu_1, \mu_2)^t$ and covariance matrix

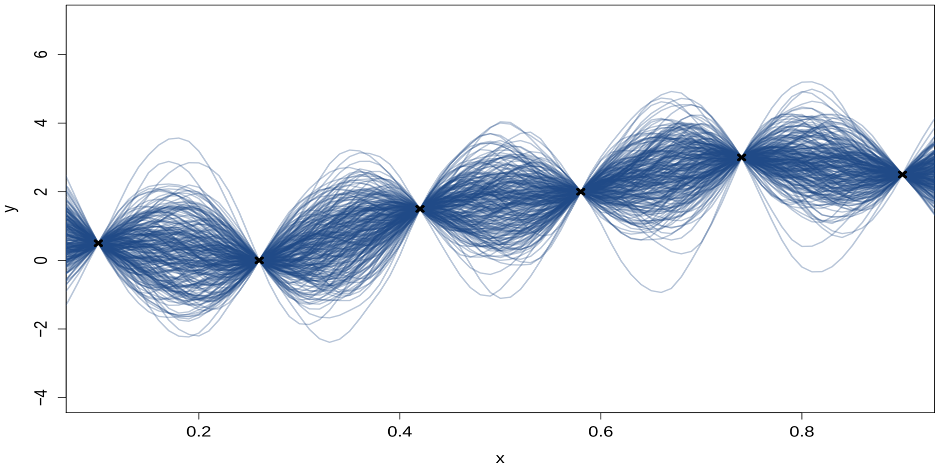
$$\Sigma = \begin{pmatrix} \Sigma_{1,1} & \Sigma_{1,2} \\ \Sigma_{2,1} & \Sigma_{2,2} \end{pmatrix}. \tag{3}$$

The conditional distribution of Y_1 knowing Y_2 is still multivariate Gaussian :

$$\begin{aligned} Y_1|Y_2 \sim \mathcal{N}(\mu_c, \Sigma_c) \text{ with } \mu_c &= \mathrm{E}[Y_1|Y_2] = \mu_1 + \Sigma_{1,2}\Sigma_{22}^{-1}(Y_2 - \mu_2) \\ \Sigma_c &= [Y_1, Y_1|Y_2] = \Sigma_{1,1} - \Sigma_{1,2}\Sigma_{2,2}^{-1}\Sigma_{2,1}, \end{aligned} \tag{4}$$


y
6
4
2
0
−2
−4
0.2
0.4
0.6
0.8
x


6
4
2
0
−2
−4
y
0.2
0.4
0.6
0.8
x


6
4
2
0
−2
−4
y
0.2
0.4
0.6
0.8
x

Focus on Kriging and EGO

Context

- DOE : (X, Y)
- A covariance function $k_\theta(x, y)$

Simple Kriging

$$
\begin{aligned}
\mu(x) &= \mathrm{E}[Z(x)|Z(X) = Y] = k_\theta(x, X) k_\theta(X, X)^{-1} Y \\
\sigma^2(x) &= \mathrm{var}[Z(x)|Z(X) = Y] = k_\theta(x, x) - k_\theta(x, X) k_\theta(X, X)^{-1} k_\theta(X, x)
\end{aligned} \tag{5}
$$

Kernels

Name	Expression
squared exponential	$k(x, y) = \sigma^2 \exp\left(-\frac{(x-y)^2}{2\theta^2}\right)$
Matern 5/2	$k(x, y) = \sigma^2 \left(1 + \frac{\sqrt{5}\lvert x-y\rvert}{\theta} + \frac{5\lvert x-y\rvert^2}{3\theta^2}\right) \exp\left(-\frac{\sqrt{5}\lvert x-y\rvert}{\theta}\right)$
Matern 3/2	$k(x, y) = \sigma^2 \left(1 + \frac{\sqrt{3}\lvert x-y\rvert}{\theta}\right) \exp\left(-\frac{\sqrt{3}\lvert x-y\rvert}{\theta}\right)$
exponential	$k(x, y) = \sigma^2 \exp\left(-\frac{\lvert x-y\rvert}{\theta}\right)$
Brownian	$k(x, y) = \sigma^2 \min(x, y)$
white noise	$k(x, y) = \sigma^2 \delta_{x,y}$
constant	$k(x, y) = \sigma^2$
linear	$k(x, y) = \sigma^2 xy$
cosine	$k(x, y) = \sigma^2 \cos\left(\frac{x-y}{\theta}\right)$
sinc	$k(x, y) = \sigma^2 \frac{\theta}{x-y} \sin\left(\frac{x-y}{\theta}\right)$

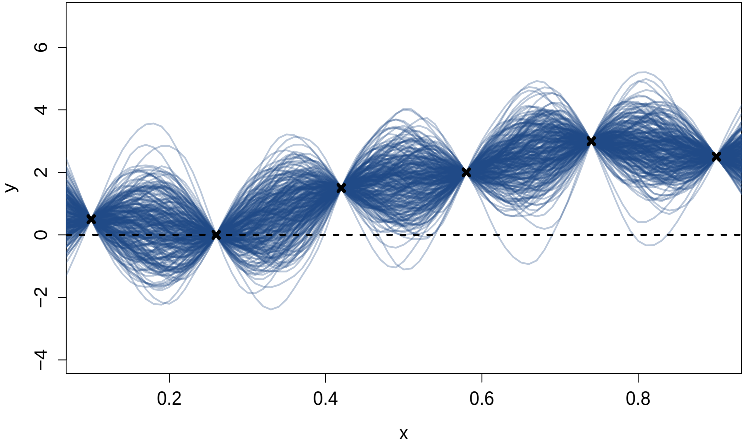
Kernels

Name	Expression
squared exponential[1]	$k(x,y) = \sigma^2 \exp\left(-\frac{1}{2}x - y\theta^2\right)$
Matérn 5/2	$k(x,y) = \sigma^2 \left(1 + \sqrt{5}x - y\theta + \frac{5}{3}x - y\theta^2\right) \exp\left(-\sqrt{5}x - y\theta\right)$
Matérn 3/2	$k(x,y) = \sigma^2 \left(1 + \sqrt{3}x - y\theta\right) \exp\left(-\sqrt{3}x - y\theta\right)$
exponential	$k(x,y) = \sigma^2 \exp\left(-x - y\theta\right)$
white noise	$k(x,y) = \sigma^2 \delta_{x,y}$
constant	$k(x,y) = \sigma^2$
linear	$k(x,y) = \sum_{i=1}^{d} \sigma_i^2 x_i y_i$

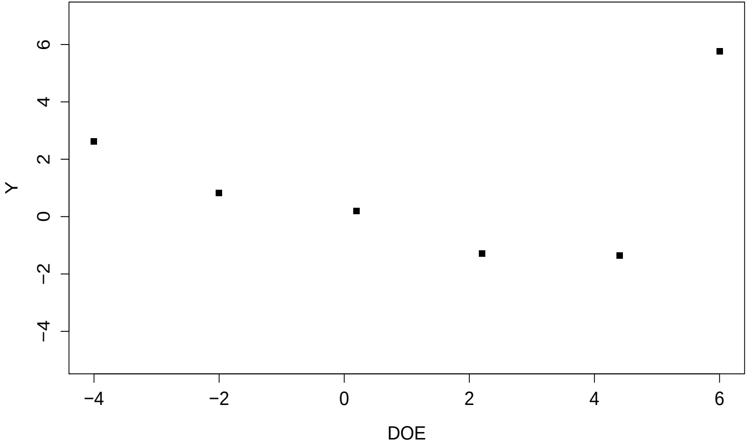
TABLE – Examples of common d-dimensional kernels.

Expected Improvement

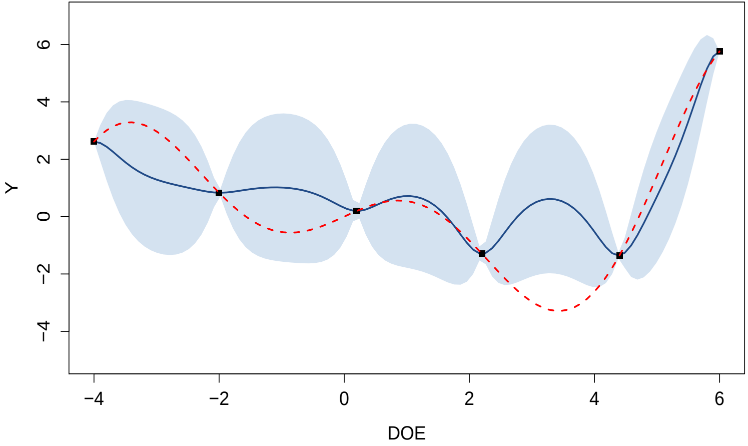
$$EI_n(\mathbf{x}) = \mathbb{E}[\max(y_n^\star - y(\mathbf{x}), 0) \mid \mathbf{Z_n}]$$



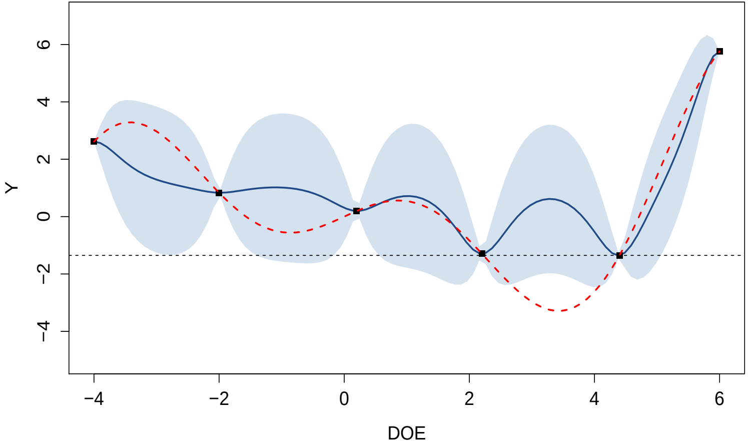
EGO Example



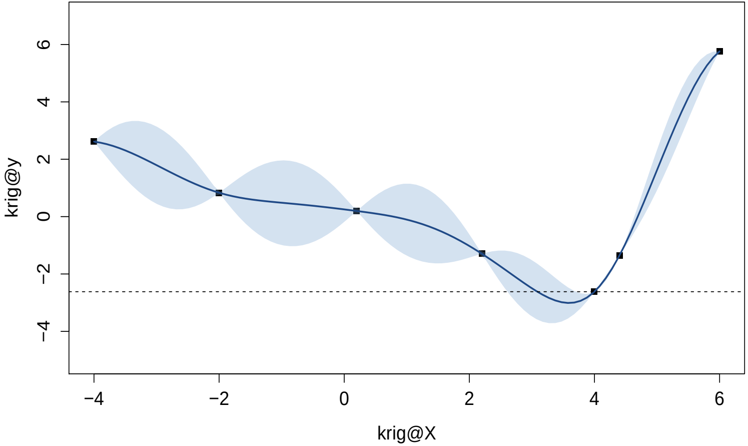
EGO Example



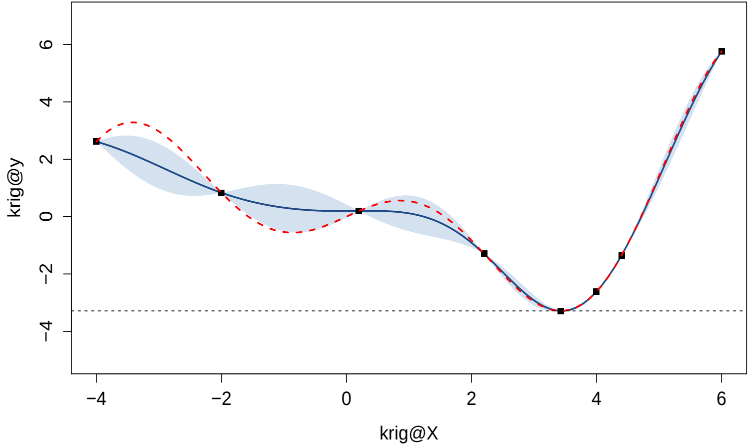
EGO Example



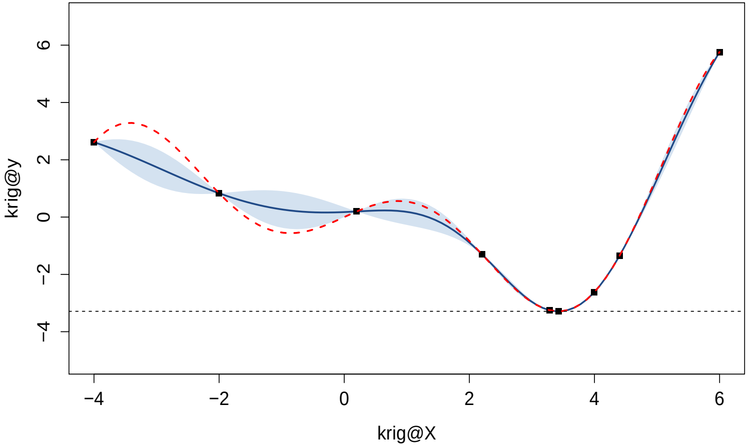
EGO Example



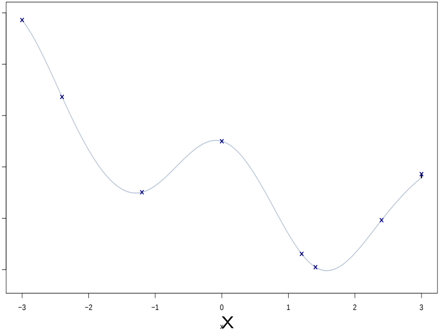
EGO Example



EGO Example



Cross-validation : example



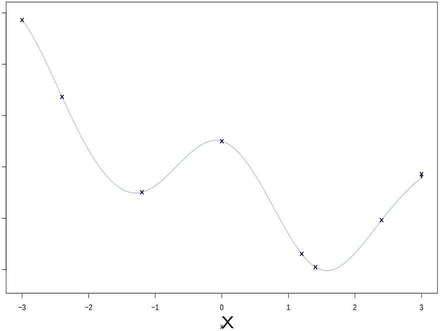
Cross-Validation

- The errors are computed based on the "sub-models" predictions.
- It is a universal method.
- The errors of the "sub-models" are used for the master model prediction assessment.

Prediction uncertainty

The idea is to use the sub-models to assess the master model prediction uncertainty.

Cross-validation : example



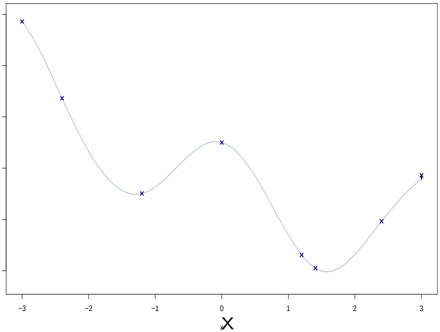
Cross-Validation

- The errors are computed based on the “sub-models” predictions.
- It is a universal method.
- The errors of the “sub-models” are used for the master model prediction assessment.

Prediction uncertainty

The idea is to use the sub-models to assess the master model prediction uncertainty.

Cross-validation : example



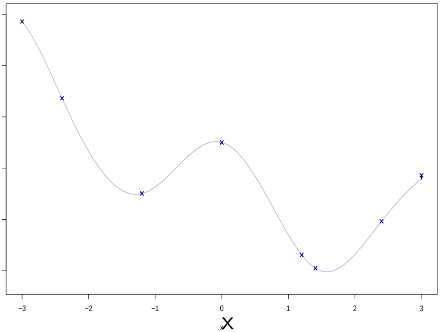
Cross-Validation

- The errors are computed based on the "sub-models" predictions.
- It is a universal method.
- The errors of the "sub-models" are used for the master model prediction assessment.

Prediction uncertainty

The idea is to use the sub-models to assess the master model prediction uncertainty.

Cross-validation : example



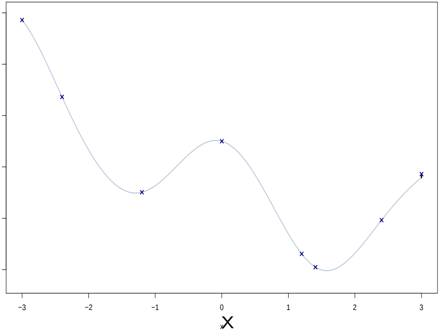
Cross-Validation

- The errors are computed based on the "sub-models" predictions.
- It is a universal method.
- The errors of the "sub-models" are used for the master model prediction assessment.

Prediction uncertainty

The idea is to use the sub-models to assess the master model prediction uncertainty.

Cross-validation : example



Cross-Validation

- The errors are computed based on the "sub-models" predictions.
- It is a universal method.
- The errors of the "sub-models" are used for the master model prediction assessment.

Prediction uncertainty

The idea is to use the sub-models to assess the master model prediction uncertainty.

Sub-models predictions variance

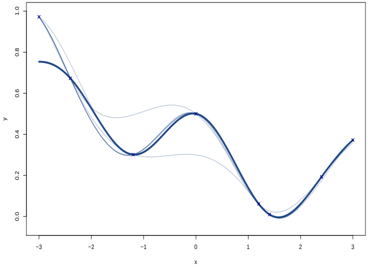


FIGURE – Cross-Validation sub-models predictions

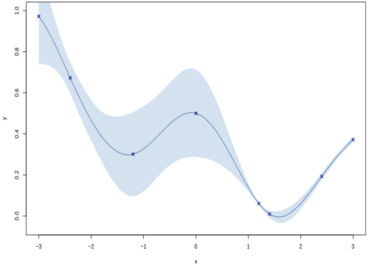


FIGURE – Blue color : master model predictions $\pm 3\widehat{\sigma}(x)$

Local aspect

The predictions vary according to the design space.

Uncertainty on design points

The uncertainty is maximal on design points.

Sub-models predictions variance

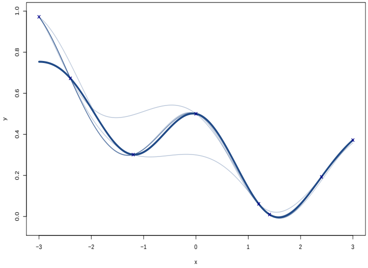


FIGURE – Cross-Validation sub-models predictions

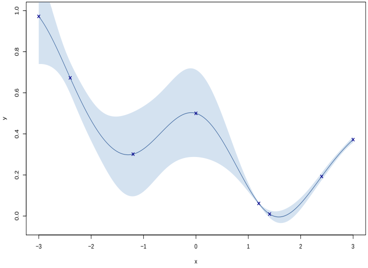


FIGURE – Blue color : master model predictions $\pm 3\widehat{\sigma}(x)$

Local aspect

The predictions vary according to the design space.

Uncertainty on design points

The uncertainty is maximal on design points.

Sub-models predictions variance

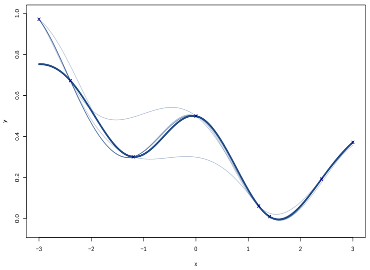


FIGURE – Cross-Validation sub-models predictions

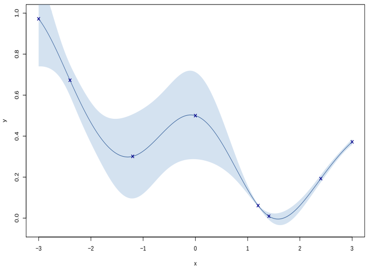


FIGURE – Blue color : master model predictions $\pm 3\widehat{\sigma}(x)$

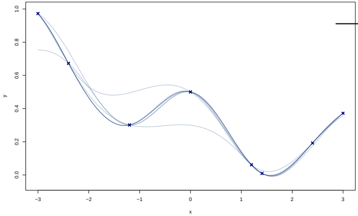
Local aspect

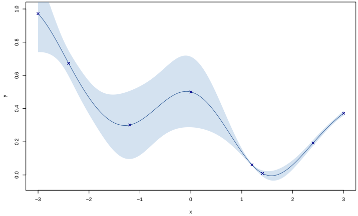
The predictions vary according to the design space.

Uncertainty on design points

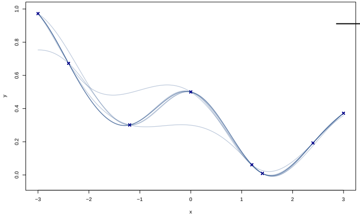
The uncertainty is maximal on design points.

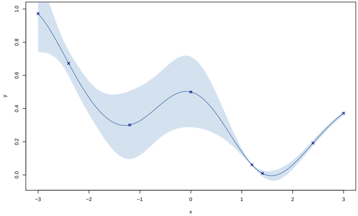
Sub-models variance

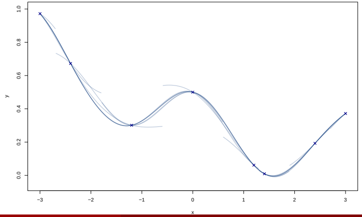




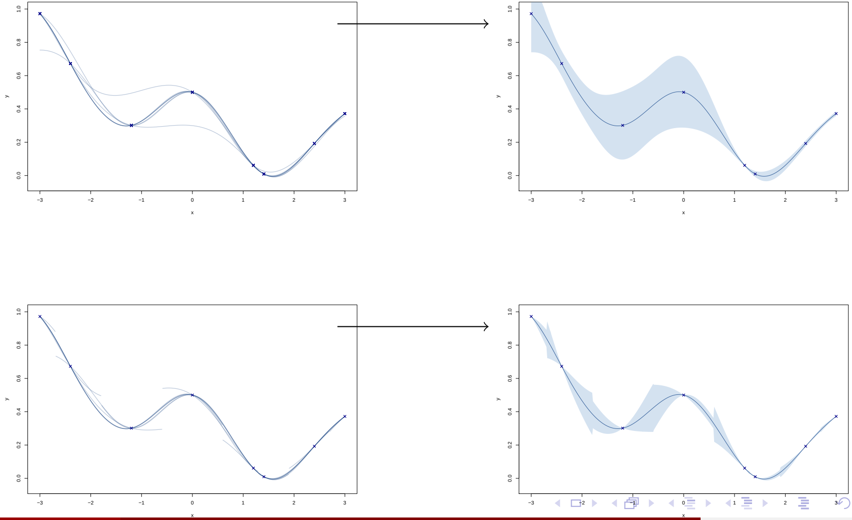
Sub-models variance







Sub-models variance



Universal Prediction distribution

Definition : UP-distribution

The Universal Prediction distribution (UP-distribution) is the weighted empirical distribution

$$\mu_{(n,\mathbf{x})}(dy) = \sum_{i=1}^{n} w_{i,n}(\mathbf{x}) \delta_{\widehat{y}_{-i}(\mathbf{x})}(dy)$$

where the weights are : $w_{i,n}(\mathbf{x}) = \dfrac{1 - e^{-d((\mathbf{x},\mathbf{x_i}))^2 \rho^{-2}}}{\sum_{j=1}^{n} \left(1 - e^{-d(\mathbf{x},\mathbf{x_j})^2 \rho^{-2}}\right)}$

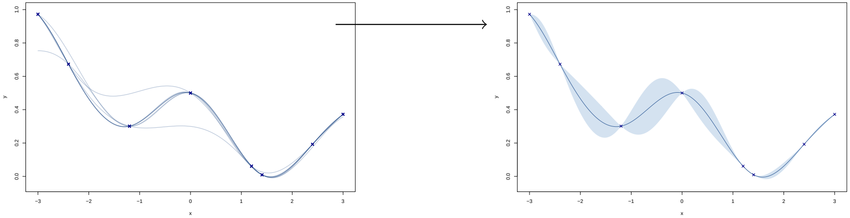
Universal Prediction distribution

Definition : UP-distribution

The Universal Prediction distribution (UP-distribution) is the weighted empirical distribution

$$\mu_{(n,\mathbf{x})}(dy) = \sum_{i=1}^{n} w_{i,n}(\mathbf{x})\delta_{\hat{y}_{-i}(\mathbf{x})}(dy)$$

where the weights are : $w_{i,n}(\mathbf{x}) = \dfrac{1-e^{-d((\mathbf{x},\mathbf{x_i}))^2\rho^{-2}}}{\sum_{j=1}^{n}\left(1-e^{-d(\mathbf{x},\mathbf{x_j})^2\rho^{-2}}\right)}$



Universal Prediction distribution

Definition : UP-distribution

The Universal Prediction distribution (UP-distribution) is the weighted empirical distribution

$\mu_{(n,\mathbf{x})}(dy) = \sum_{i=1}^{n} w_{i,n}(\mathbf{x}) \delta_{\hat{y}_{-i}(\mathbf{x})}(dy)$

where the weights are : $w_{i,n}(\mathbf{x}) = \frac{1-e^{-d((\mathbf{x},\mathbf{x_i}))^2 \rho^{-2}}}{\sum_{j=1}^{n}\left(1-e^{-d(\mathbf{x},\mathbf{x_j})^2 \rho^{-2}}\right)}$

Universal Prediction distribution

Definition : UP-distribution

The Universal Prediction distribution (UP-distribution) is the weighted empirical distribution

$$\mu_{(n,\mathbf{x})}(dy) = \sum_{i=1}^{n} w_{i,n}(\mathbf{x})\delta_{\hat{y}_{-i}(\mathbf{x})}(dy)$$

where the weights are : $w_{i,n}(\mathbf{x}) = \frac{1-e^{-d((\mathbf{x},\mathbf{x_i}))^2\rho^{-2}}}{\sum_{j=1}^{n}\left(1-e^{-d(\mathbf{x},\mathbf{x_j})^2\rho^{-2}}\right)}$

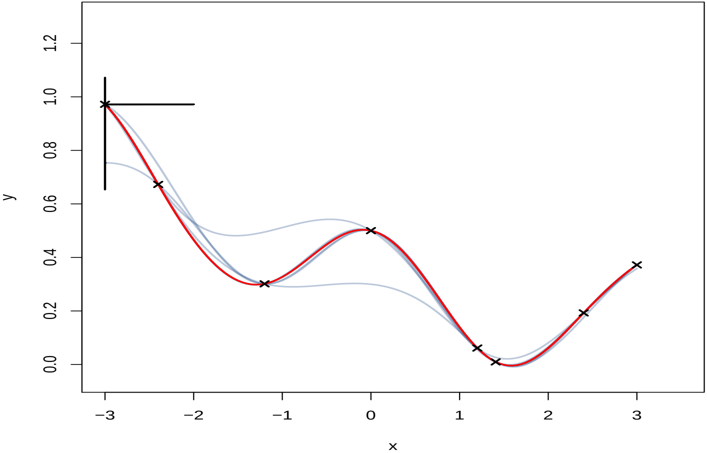
UP mean

$$\begin{aligned}\widehat{m}_n(x) &= \int y\, d\widehat{\mu}_{n,x}(y) \\ &= \sum_{i=1}^{n} w_i(x)\widehat{y}_{-i}(x)\end{aligned} \tag{6}$$

UP variance

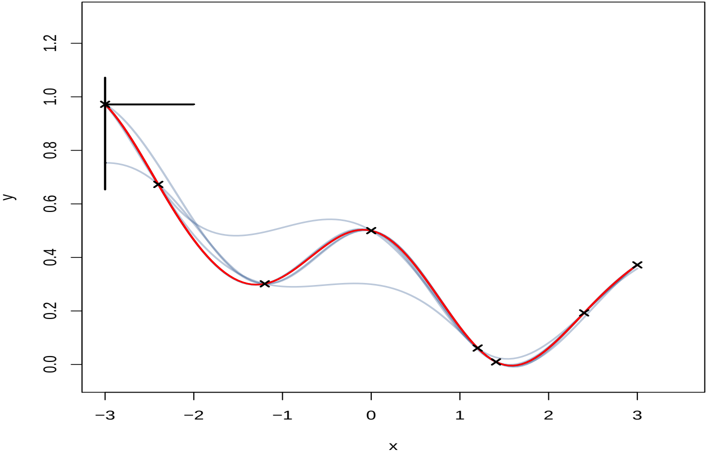
$$\begin{aligned}\widehat{\sigma}_n^2(x) &= \int (y - \widehat{m}_n(x))^2\, d\widehat{\mu}_{n,x}(y) \\ &= \sum_{i=1}^{n} w_i(x)(\widehat{y}_{-i}(x) - \widehat{m}_n(x))^2\end{aligned} \tag{7}$$

UP-Distribution example (1/2)



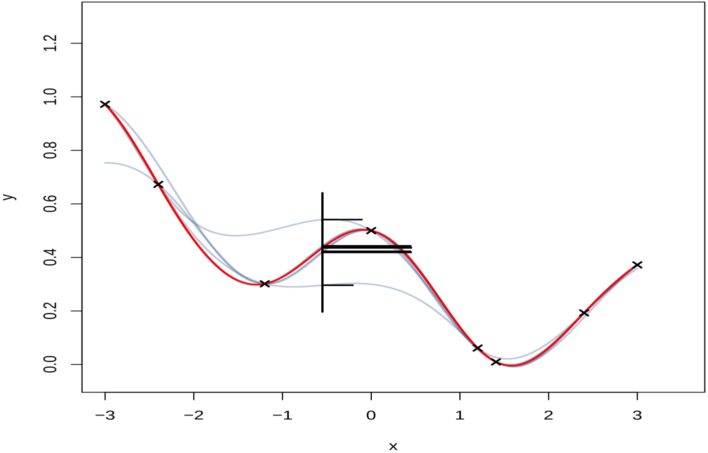
Horizental bars : UP distribution in x

UP-Distribution example (1/2)



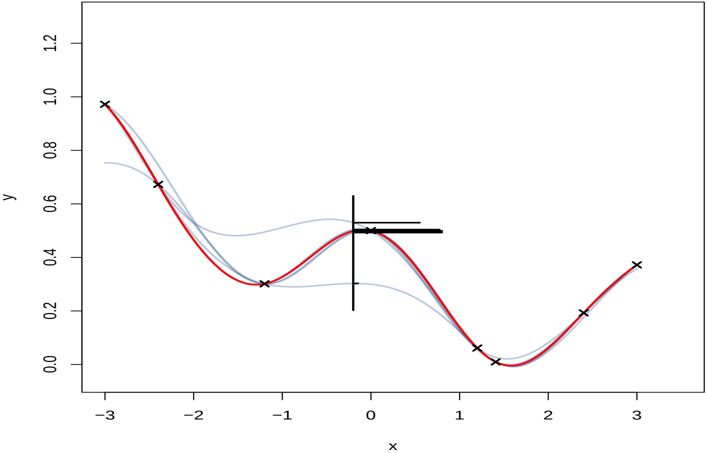
Horizental bars : UP distribution in x

UP-Distribution example (1/2)



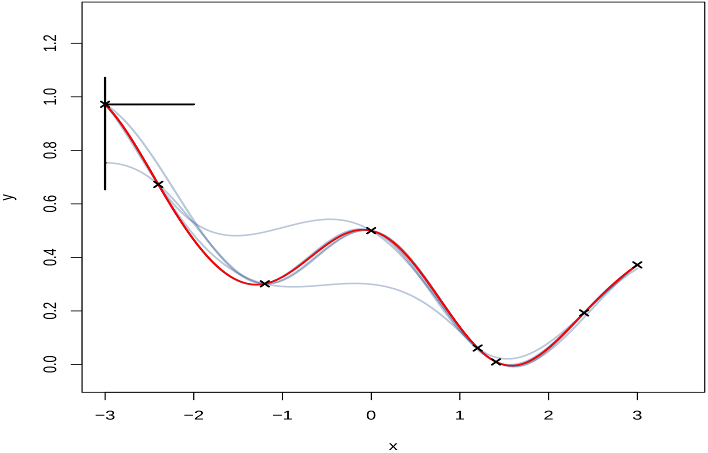
Horizental bars : UP distribution in x

UP-Distribution example (1/2)



Horizental bars : UP distribution in x

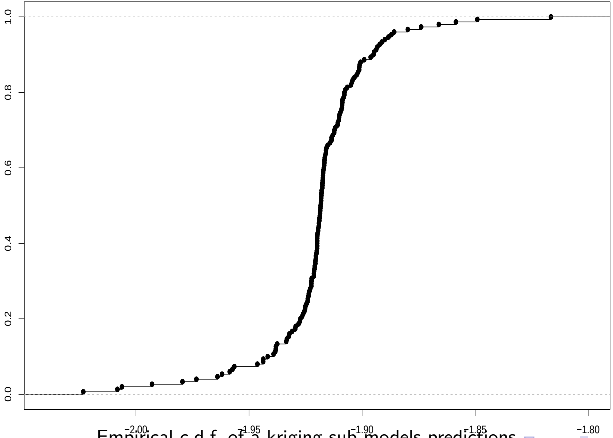
UP-Distribution example (1/2)



Horizental bars : UP distribution in x

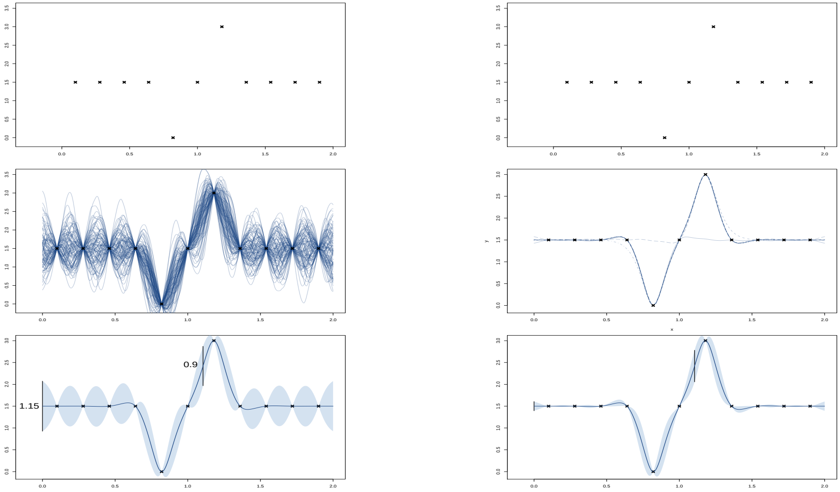
UP-Distribution example (2/2)

Welch function (20 dimension, 150 design points)



Empirical c.d.f. of a kriging sub-models predictions

UP-distribution vs Kriging Prediction distribution



Kriging variance

Kriging UP-distribution

UP distribution applications

UP-distribution enables

- The quantification of a local uncertainty assessment.
- The definition of a sequential design of experiments algorithms.
- The adaptation of most the kriging-based algorithms to all the surrogate models.

Applications

- Optimization
- Refinement
- Inverse problems

UP distribution applications

UP-distribution enables

- The quantification of a local uncertainty assessment.
- The definition of a sequential design of experiments algorithms.
- The adaptation of most the kriging-based algorithms to all the surrogate models.

Applications

- **Optimization**
- Refinement
- Inverse problems

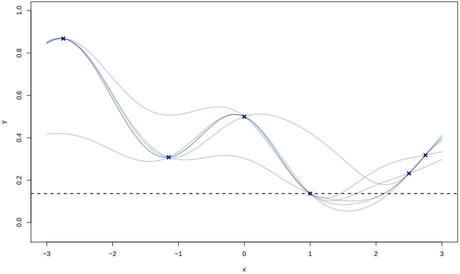
Optimization

Empirical Expected improvement

$$EEI(x) = \int (f_{min} - y)_+ \, d\widehat{\mu}_{n,x}(y)$$

UP Efficient Global Optimization

UP-EGO consists in sampling the point that maximizes $EEI + \delta \underline{d}_{\mathbf{X_n}}$.



Proposition :(BEN SALEM, ROUSTANT, GAMBOA, TOMASO 2015)

Let s be a real function defined on $\mathbb{X}$ and let $\mathbf{x}^\star \in \arg\min\{s(\mathbf{x}), \mathbf{x} \in \mathbb{X}\}$. If $\widehat{s}$ is an interpolating continuous metamodel bounded on $\mathbb{X}$, then $\mathbf{x}^\star$ is adherent to the sequence of points S generated by UP-EGO($\widehat{s}$).

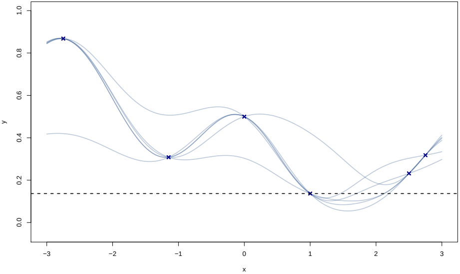
Optimization

Empirical Expected improvement

$$EEI(x) = \int (f_{min} - y)_+ \, d\widehat{\mu}_{n,x}(y)$$

UP Efficient Global Optimization

UP-EGO consists in sampling the point that maximizes $EEI + \delta \underline{d}_{\mathbf{X_n}}$.



Proposition :(BEN SALEM, ROUSTANT, GAMBOA, TOMASO 2015)

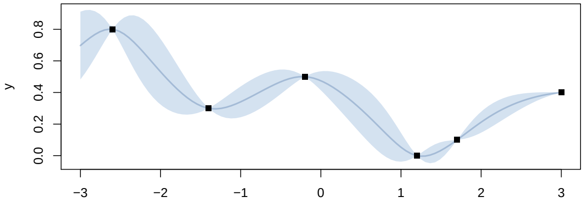
Let s be a real function defined on $\mathbb{X}$ and let $\mathbf{x}^\star \in \arg\min\{s(\mathbf{x}), \mathbf{x} \in \mathbb{X}\}$. If $\widehat{s}$ is an interpolating continuous metamodel bounded on $\mathbb{X}$, then $\mathbf{x}^\star$ is adherent to the sequence of points S generated by UP-EGO($\widehat{s}$).

Installation

```
# Install devtools, if you haven't already.
install.packages("devtools")

library(devtools)
install_github("malekbs/UP")
```

First example

```
library(UP)
x           <- as.matrix(c(-2.6,-0.2, 1.7,-1.4,1.2,3))
y           <- c(0.8, 0.5, 0.1, 0.3, 0, 0.4)
xverif      <- seq(-3, 3, length.out =300)
krig        <- krigingsm$new()
resampling  <- UPClass$new(x, y, Scale =TRUE)
upsm        <- UPSM$new(sm= krig, UP= resampling)
prediction  <- upsm$uppredict(xverif)
plotUP1D(xverif, prediction, x, y)
```



First example

```
names(predictions)
```

```
## [1] "master_prediction" "sub_predictions"   "coeff"
## [4] "upsd"              "mindist"           "mean"
## [7] "unsc_subpred"
```

UPSM

Requirement

- A Surrogate model (metamodel)
- A resampling strategy

Meta-models

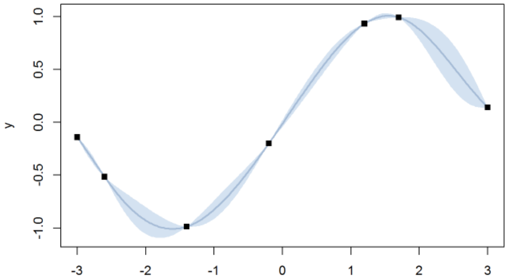
- Kriging
- SVM
- Customized

Resampling techniques

- Leave-One-Out CV
- *k*-Fold CV
- Customized

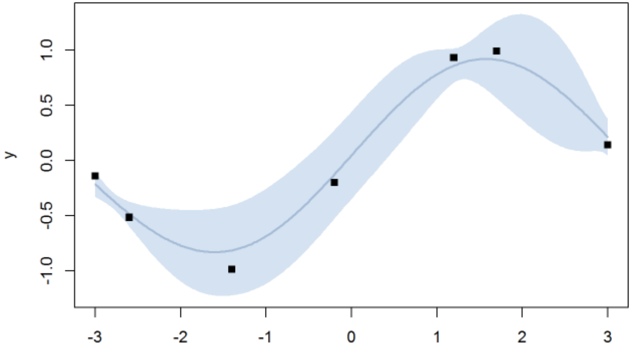
Second example : other features

```
library(UP)

x              <- as.matrix(c(-3,-2.6,-0.2, 1.7,-1.4,1.2,3))
y              <- sin(x)
xverif         <- seq(-3, 3, length.out =300)

###### first metamodel kriging ######
krig           <- krigingsm$new()
resampling     <- UPClass$new(x, y, Scale = TRUE)
upsm           <- UPSM$new(sm= krig, UP= resampling)
prediction     <- upsm$uppredict(xverif)
plotUP1D(xverif, prediction, x, y)
```



Second example : other features

```
###### SVM accroding to different cost value #####
upsvm          <- UPSM$new(sm = svmsm$new(parameters = list(cost=7)), UP= resampling)
prediction2    <- upsvm$uppredict(as.matrix(xverif))
plotUP1D(xverif, prediction2, x, y)
```

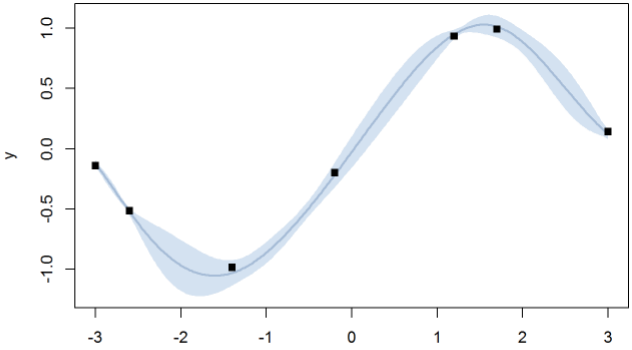


Second example : other features

```
######### aggregation ############################

######### fitness function #######################
listCrit          <- list()
listCrit[[1]]     <- MSE$new()
listCrit[[2]]     <- Resampling_Error$new()
listCrit[[3]]     <- penlrm$new(x, y)
fit               <- custom_fit$new(c(4,2,1), listCrit)

listsm            <- list()
listsm[[1]]       <- upsm
listsm[[2]]       <- upsvm
parameters        <- list(listsm =listsm)
ens               <- aggregation$new(fit, x, y, parameters = parameters)
ens$train()
prediEns          <- ens$predict(xverif)

prediction_ens    <- ens$uppredict(as.matrix(xverif))
```

Second example : other features

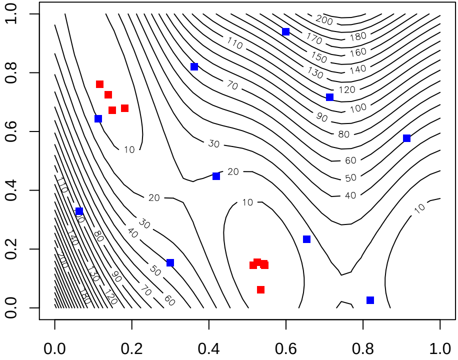
```
plotUP1D(xverif, prediction_ens, x, y)
```



Optimization UP-EGO (Looking for the best pizza recipe)

```
library("UP")
library("lhs")

d           <- 2
n           <- 10
X           <- optimumLHS(n,d)
Y           <- apply(X, 1, branin)
upsm        <- UPSM$new(sm= krigingsm$new(), UP=UPClass$new(X,Y,Scale =TRUE))

plotContour2D(branin,40,40)
lines(x =X[,1], y = X[,2], type = "p",pch=15,col = "blue")
nsteps      <- 10
upego_res   <- upego(upsm, fun = branin, nsteps = nsteps, lower= c(0,0),upper = c(1,1),
EEIcontrol=list(up_method="UP_reg"))

lines(upego_res$last$get_DOE()$x[(n+1):(n+nsteps),1],
      upego_res$last$get_DOE()$x[(n+1):(nsteps+n),2],
      "p", pch=15, col = "red")
```

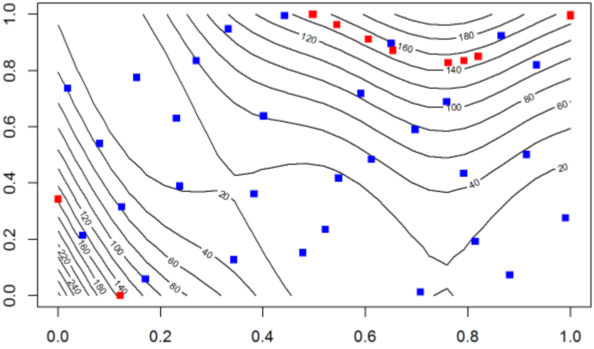
Optimization UP-EGO (Looking for the best pizza recipe)



Inversion

```
library("UP")
library("lhs")

d           <- 2
n           <- 30
X           <- optimumLHS(n,d)
Y           <- apply(X, 1, branin)

plotContour2D(branin,30,11)
lines(x =X[,1], y = X[,2], type = "p",pch=15,col = "blue")
nsteps      <- 15
upsm        <- UPSM$new(sm= krigingsm$new(), UP=UPClass$new(X,Y,Scale =TRUE))
upinv_res   <- upinverse(upsm, fun = branin, target = 150,
                   nsteps = nsteps, lower= c(0,0),upper = c(1,1))
```

Inversion


0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
20
40
60
80
100
120
140
160
180
220
240
120
140
160
180
100
80
60
40
20

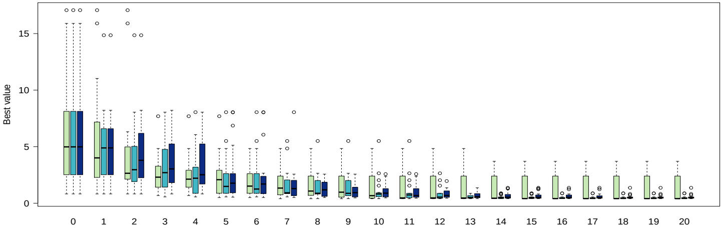
Optimization tests

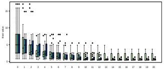
Table – Optimization test functions

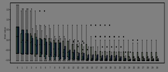
function $f^{(i)}$	Dimension $d^{(i)}$	Number of initial points $n_0^{(i)}$	Number of iterations $N_{max}^{(i)}$
Branin	2	5	40
Ackley	2	10	30
Six-hump Camel	2	10	30
Hartmann6	6	20	50

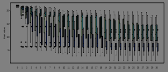
UP-EGO results

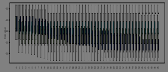
EGO(Kriging) UP-EGO(Kriging) UP-EGO(PPS-GA)




Branin

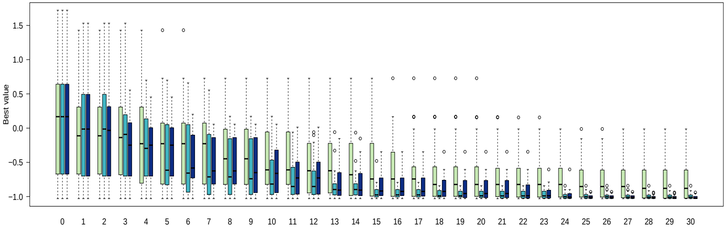

Camel

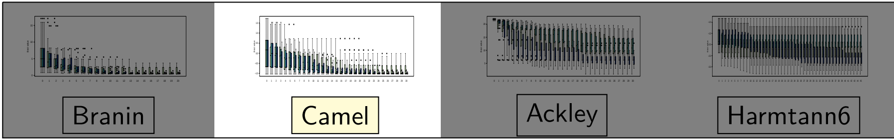

Ackley


Harmtann6

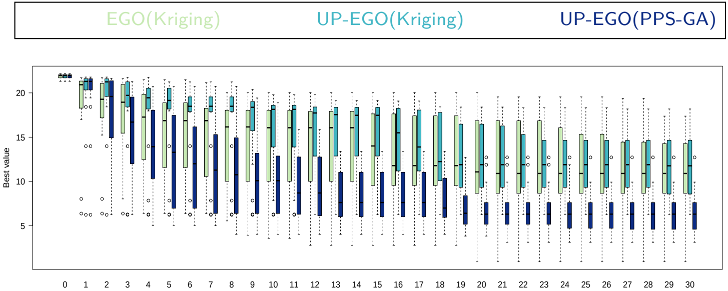
UP-EGO results

EGO(Kriging) UP-EGO(Kriging) UP-EGO(PPS-GA)





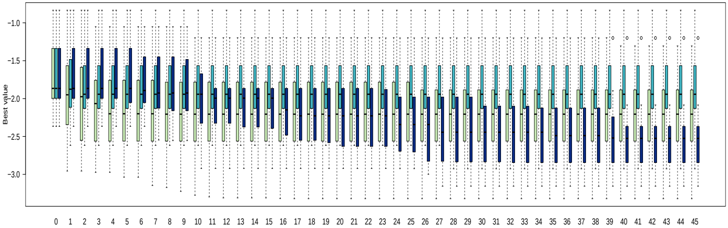
UP-EGO results

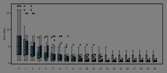


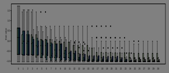
Branin | Camel | Ackley | Harmtann6

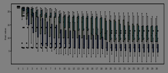
UP-EGO results

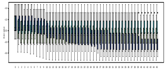
EGO(Kriging) UP-EGO(Kriging) UP-EGO(PPS-GA)




Branin

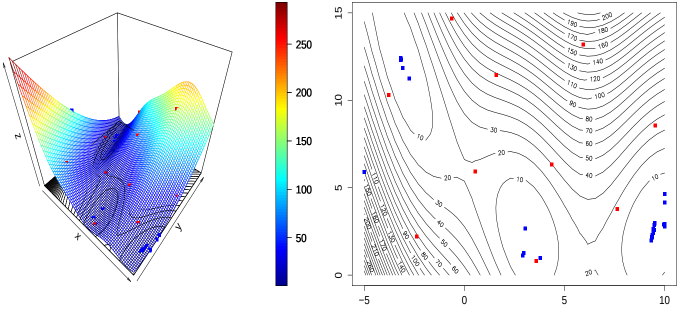

Camel


Ackley


Harmtann6

Global Optimization

Branin function : generated points



Initial design points are in red and the points which are generated by the UP-EGO are in blue

Package summary

Features

- Add a prediction distribution to a surrogate model.
- Surrogate models assessment and aggregation construction.
- Various sequential algorithms (UP SMART, UP-EGO, UP-Inverse).
- Various customization options.

Package summary

Features

- Add a prediction distribution to a surrogate model.
- Surrogate models assessment and aggregation construction.
- Various sequential algorithms (UP SMART, UP-EGO, UP-Inverse).
- Various customization options.

And especially it helps in making good pizza.

Conclusion

UP package

- Provides a universal prediction distribution.
- Can be enriched, various customized options.
- Optimization, refinement and inversion algorithms are provided.

Drawbacks

- The UP-distribution support is a finite number of points (theoretical drawback).
- Computing time is not a known strength of resampling methods such as leave-one-out (computational drawback).

Conclusion

UP package

- Provides a universal prediction distribution.
- Can be enriched, various customized options.
- Optimization, refinement and inversion algorithms are provided.

Drawbacks

- The UP-distribution support is a finite number of points (theoretical drawback).
- Computing time is not a known strength of resampling methods such as leave-one-out (computational drawback).

Merci

Further Reading

M. Ben Salem, O. Roustant, F. Gamboa, L. Tomaso
Universal Prediction Distribution for Surrogate Models
arXiv preprint arXiv :1512.07560, 2015.

install_github("malekbs\UP")

Thank you ! Any question ?
I think Nicolas Durrande for his slides about kriging.