MACROS®

by DATADVANCE

Generic Tool for Sensitivity and Dependency Analysis

Motivation: Simple Example

If the friction from brakes, wind resistance and all such factors remain same, which will stop first: a heavier car or a lighter car?

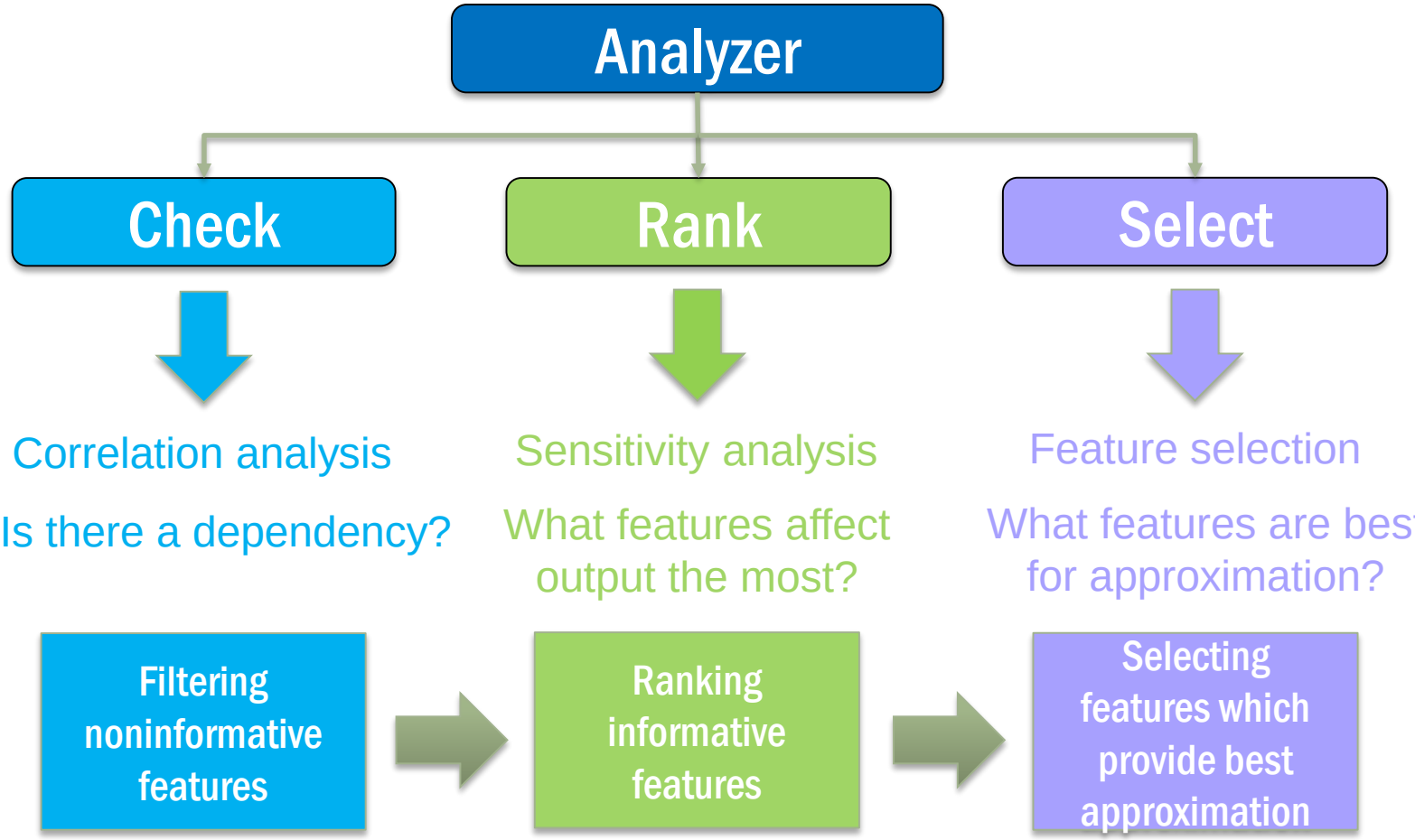
?

Motivation: Simple Example

- **Physics tells:**
  - breaking distance significantly depends on speed and friction coefficient (road and tires parameters), but does not depend on car mass

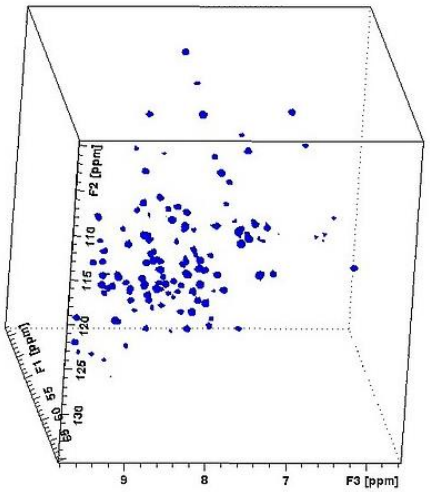
- If we didn't know physics and only have made a number of experiments this **answer could be obtained with a tool such as GT SDA.**

GT SDA structure

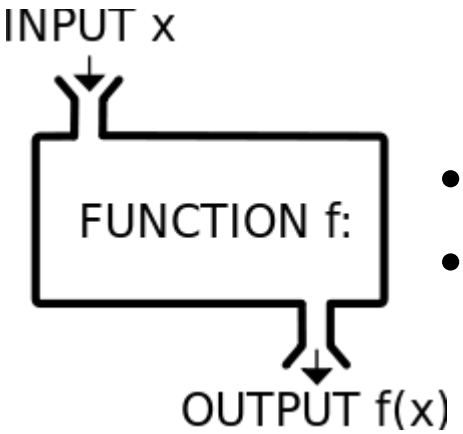


Considered Initial Conditions

- **Sample**
  - It's not possible to get new sample points



- **Black Box**
  - We can compute output in arbitrary points



- **Expensive**
- **Cheap**

Introduction

- $\boldsymbol{X}$ – whole input data sample
- $\boldsymbol{Y}$ – whole output data sample
- x_i - input data for i —th feature
- **Budget** – number of maximum allowed calls to the black box function from GT SDA

Preparations

- To use GT SDA Ranker in MacrosForPython, you need to import corresponding package:

```
from da.macros import gtsda
```

- After importing GT SDA package, we can create Analyzer object to run all GT SDA methods:

```
analyzer = gtsda.Analyzer()
```

- Loading data (one output) for sample input

```
data = np.loadtxt('data.csv', delimiter=',')
X = data[:, :-1]
Y = data[:, -1]
```

Preparations: Blackbox usage

- To use **black-box based techniques** we first need to prepare black-box

```
class ExampleBlackbox(da.macros.blackbox.Blackbox):
    def prepare_blackbox(self):
        self.add_variable((0, 1))
        self.add_variable((0, 1))
        self.add_variable((0, 1))
        self.add_response()

    def evaluate(self, queryx):
        result = []
        for x in queryx:
            result.append(model_func(x))
        return result

bbox = ExampleBlackbox()
```

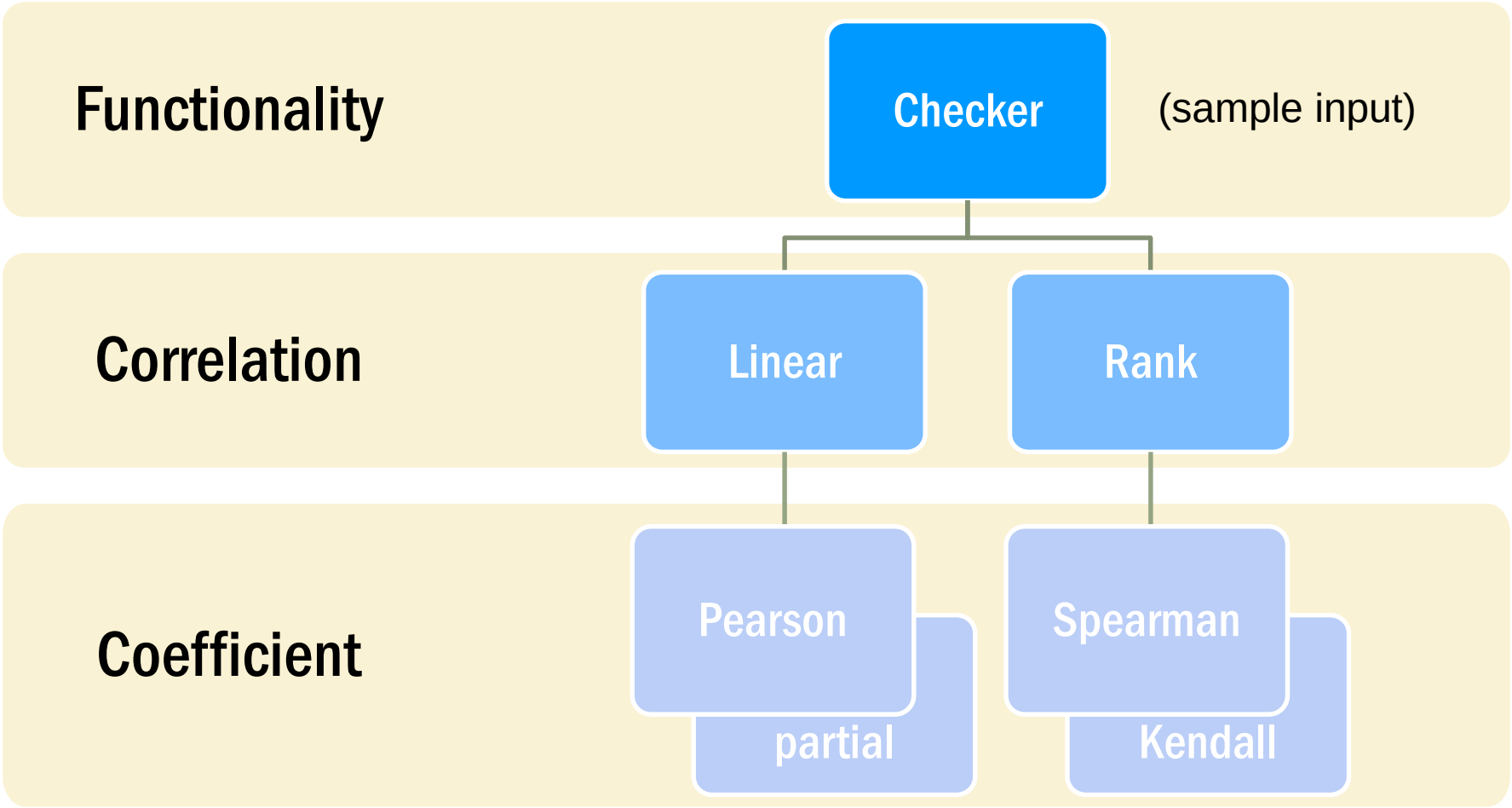
add 3 variables in problem

add new response in problem

model function to analyze

create black-box

GT SDA: Checker



Using Checker

- **Correlation type**

```
analyzer.options.set({'gtsda/checker/scorestype':
'pearsoncorrelation‘})
```

or

```
analyzer.options.set({'gtsda/checker/scorestype':
'pearsonpartialcorrelation‘})
```

or

```
analyzer.options.set({'gtsda/checker/scorestype':
‘spearmancorrelation‘})
```

- **Runnining Checker**

```
result = analyzer.check(x=X, y=Y)
```

Pearson correlation *(linear, univariate)*

- Pearson correlation is a popular of estimating a linear dependency.

- Pearson correlation between output y and feature x_i:
  - 1 if and only if $y = ax_i$ if $a > 0$
  - -1 if and only if $y = ax_i$ if $a < 0$ correspondingly.

- Mathematicaly Pearson correlation coefficient between output y and feature x_i may be defined as:

$$\rho = \frac{\sum(x_i - \hat{x_i})(y - \hat{y})}{\sqrt{\sum(x_i - \hat{x_i})^2}\sqrt{\sum(y - \hat{y})^2}}$$

Pearson correlation *(linear, univariate)*

- **Pros:**
  - Very simple and reliable measure
- **Cons:**
  - Degrades if output depends on many inputs
  - May not catch some non linear dependencies

Partial pearson correlation *(linear, multivariate)*

- The method is a **generalization of pearson correlation** for the multivariate analysis problems.
- **The difference is** that the method substract from the considered input and output possible infuence of other inputs.
- After that It computes **pearson correlation coefficient** between each substracted input and output.
- The method provides ± 1 if and only if y is a linear function of considered feature x_i *(even if* y *lineary depends on other features* X*)*.

Partial pearson correlation *(linear, multivariate)*

- **Pros:**
  - In case of multivariate linear dependency tool provides accurate estimates *(does not degrade as much as pearson correlation when the problem dimensionality increases)*
- **Cons:**
  - Can have long working time in case when the total number of features is big.
  - If dependency is not linear *(and especially there are strong cross-feature interactions)* method estimates may degrade

Spearman correlation *(monotonic, univariate)*

- Spearman correlation is also a **generalization of pearson correlation** but this time for the case when dependency is not linear but still monotonic.
- **The difference is** that the method computed rank transformation of all inputs and outputs.
- After that It computes **pearson correlation coefficient** between each transformed input and output.
- The method provides ± 1 if and only if y is a monotonic function of considered feature x_i.

Spearman correlation *(monotonic, univariate)*

- **Pros:**
  - May catch not only linear but also any monotonic dependency
- **Cons:**
  - Degrades if output depends on many inputs
  - May not catch some non monotonic dependencies

Kendall Tau *(linear, univariate, discrete)*

- Kendall correlation is a method to calculate correlation between discrete or categorial samples.
- It can also detect monotonic dependency.
- The score is computed by:

$$\tau = \frac{2T}{n(n-1)},$$

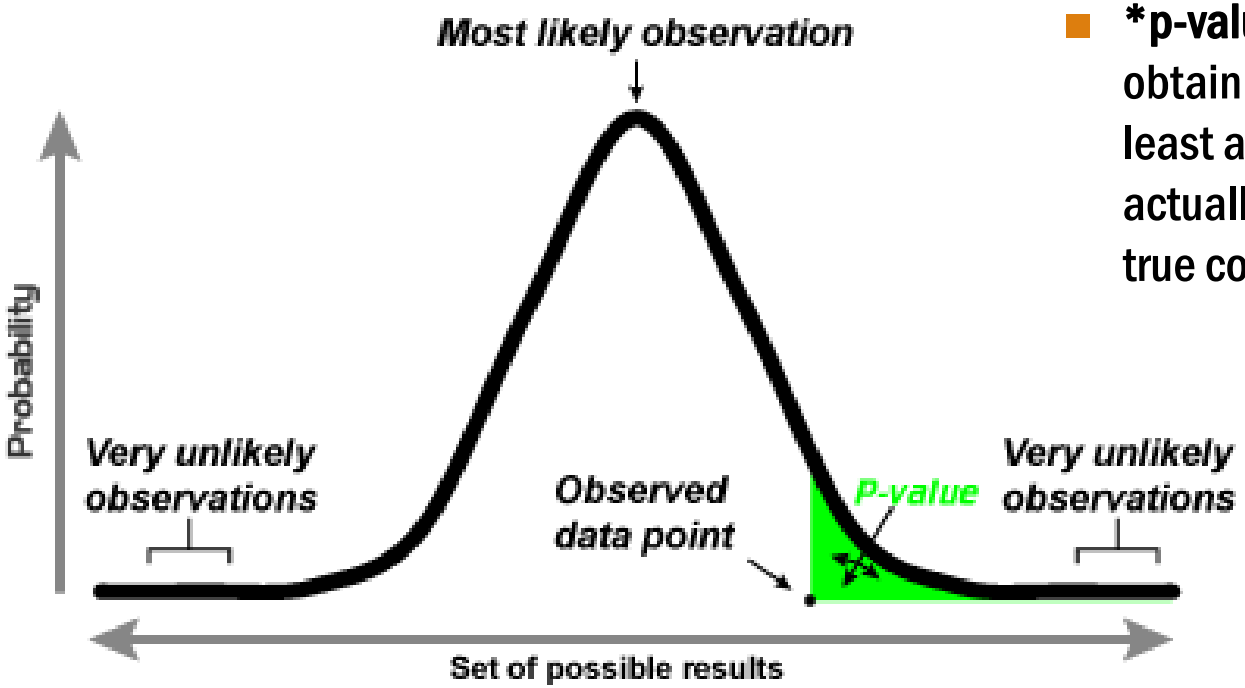
where $T = \sum_{\{i<j\}} sign(x_i < x_j)\, sign(y_i < y_j)$

Kendall Tau *(linear, univariate, discrete)*

- **Pros:**
  - May catch not only linear but also any monotonic dependency
  - Is effective for categorial and discrete variables
- **Cons:**
  - Degrades if output depends on many inputs
  - May not catch some non monotonic dependencies
  - Computational time is $O(n^2)$

Significance testing

- For each correlation coefficient GT SDA Checker also may compute corresponding **p-value*** that allows to tell if correlation is statistically significant.



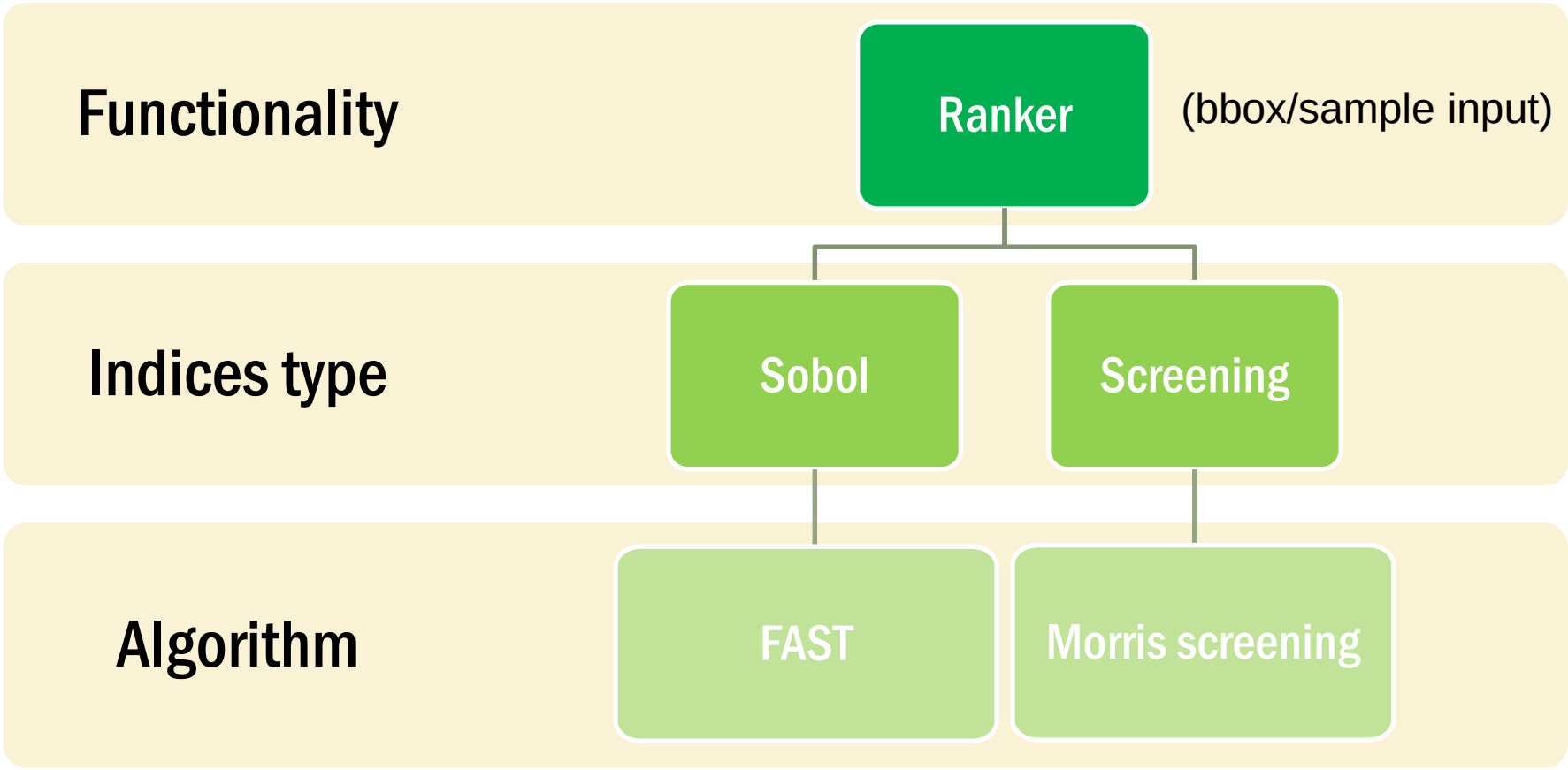
A p-value (shaded green area) is the probability of an observed (or more extreme) result arising by chance

- ***p-value** - is the probability of obtaining correlation score value at least as extreme as the one that was actually observed, assuming that the true correlation is 0

To be added soon…

- In MACROS 4.0 final release we would add
  - **(partial) Distance correlation** technique that is a generalization of pearson correlation for general type of dependencies

GT SDA: Ranker



Using ranker

- Selecting indices type

```
analyzer.options.set({"GTSDA/Ranker/IndicesType": "screening"})
```

or

```
analyzer.options.set({"GTSDA/Ranker/IndicesType": "sobol"})
```

- Runnining Ranker in sample mode

```
result = analyzer.rank(x=X, y=Y)
```

or blackbox mode

```
result = analyzer.rank(blackbox=blackbox, budget=budget)
```

Sobol indicies

- Idea of Sobol indices is to measure what portion of output variance is described by the variance of the input feature.

- In GT SDA we may simultaneously estimate:

  - **Total indices** - tells what portion of output variance would be lost if we fix considered feature to it's mean value, while still vary other inputs *(takes into account all cross-feature interactions)*

  - **Main indices** - tells what portion of output variance would be decribed by considered input provided all other inputs are fixed at their mean values *(ignores cross-feature interactions).*

  - **Interaction indices** - difference between Total and Main, that represents strength of variables interactions.

Sobol indicies

- For each feature **main** indices are estimated as

$$S_i = \frac{V_{x_i}[E_{\sim x_i}(Y|x_i)]}{V(Y)},$$

- **total** indices are estimated as

$$T_i = 1 - \frac{V_{\sim x_i}\left[E_{x_i}\left(Y\middle|x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_p\right)\right]}{V(Y)},$$

where $E_{x_i}[\cdot], V_{x_i}[\cdot]$ is a mean and variance with respect to x_i,

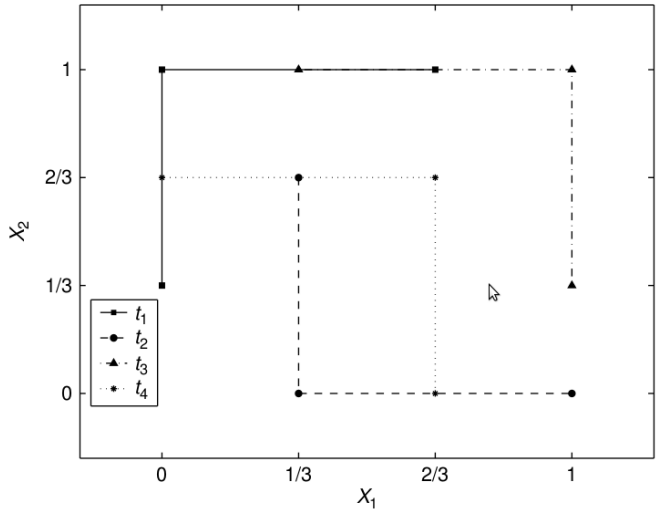
$E_{\sim x_i}(\cdot\,|x_i), V_{\sim x_i}(\cdot\,|x_i)$ is a conditional mean and variance with respect to all features except x_i.

Sobol indicies

- **Pros:**
  - Can measure the strength of non-linear dependency
  - Provided it has enough budget can measure precise values of main, interaction and total effects
- **Cons:**
  - Requires significant budget to obtain meaningful result
  - Needs cheap blackbox to work (in case of sample input needs to construct surrogate model which may take significant time for large samples)

Screening indicies

- Idea of Morris Screening approach is to generate uniform set of trajectories in the design space.



- On each step of trajectory only one component x_i of input vector X is changed, and the following function is estimated:

$$d_i(X) = \frac{Y(x_1,\ldots,x_i+\delta_i,\ldots,x_p)-Y(x_1,\ldots,x_i,\ldots,x_p)}{\delta_i},$$

where δ_i is a step size.

Screening indicies

- **Score for i-th feature is computed as**

$$W[i] = \frac{1}{r}\sum_{j=1}^{r}\left|d_i\left(X^j\right)\right|,$$

  - **r – is a number of steps changing i-th feature value on all trajectories and X^j is the input value at these steps.**

    - For linear functions scores returned by method coincide with coefficients for linear regression.
    - For non-linear functions they may be sought as crude estimates of average absolute value of partial derivatives.

Screening indicies

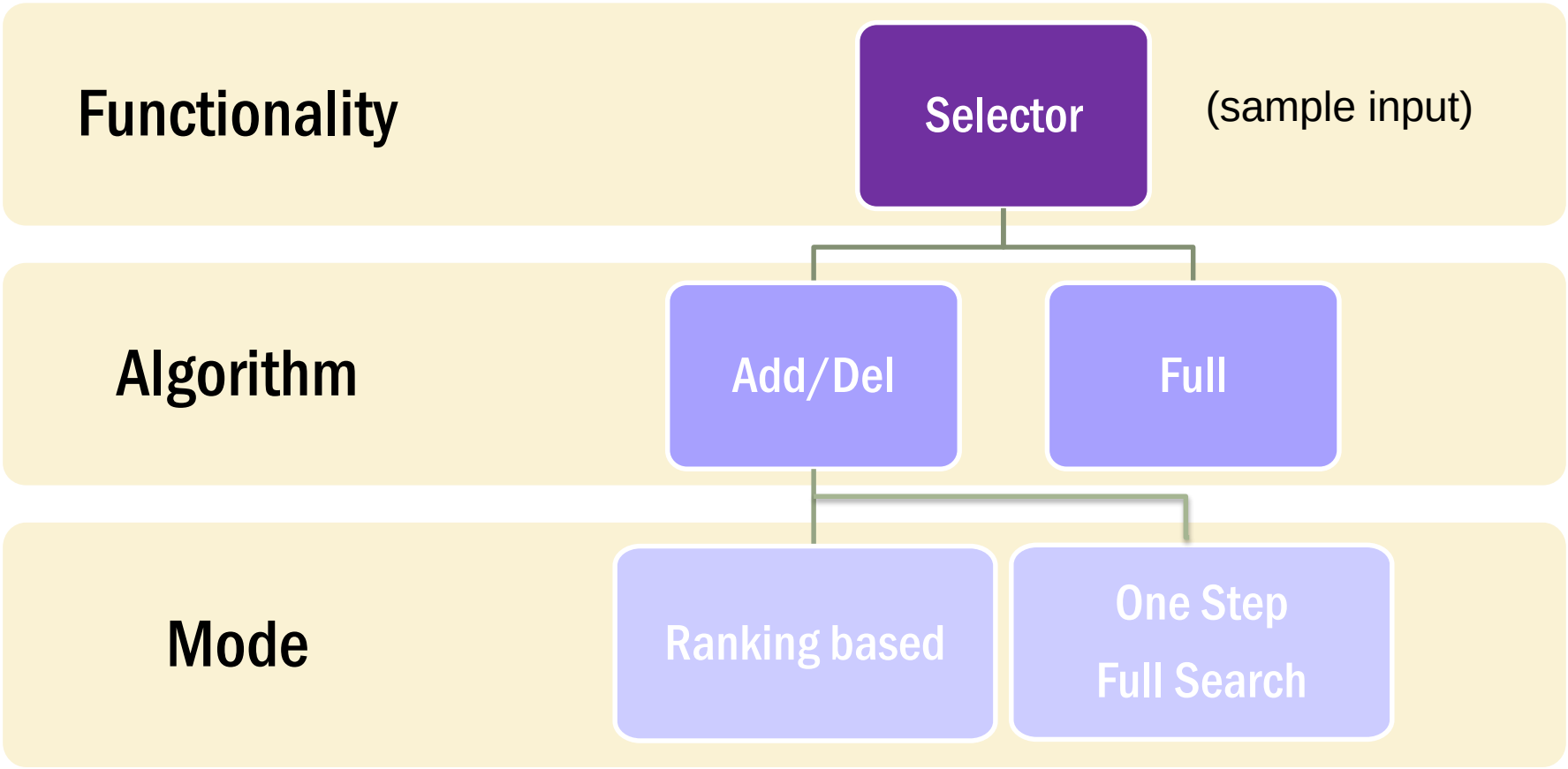
- **Pros:**
  - May work even with very small budgets *(so viable option even for expensive blackbox)*
  - Very effective at finding features that have no influence
  - Can measure the level of nonlinearity and monotonicity of dependency

- **Cons:**
  - Scores do not have a strict physical meaning (though d_i may be thought as crude estimation of derivatives)

To be added soon…

- In MACROS 5.0 we would add:
  - **Taguchi scores** technique that would allow to rank variables with respect to their influence being sensitive to noise

GT SDA: Selector



Using Selector

- Selecting search strategy

```
analyzer.options.set({"GTSDA/Ranker/IndicesType": "screening")
```

- Selecting validation type

```
analyzer.options.set({'gtsda/selector/validationtype': 'testsample'})
```

- Runnining Selector

with no test data available

```
result = analyzer.select(x=X, y=Y)
```

or with test data available

```
result = analyzer.select(x=X, y=Y, x_test=X_test,
y_test=Y_test)
```

Selector

- Selector functionality of GT SDA is a tool for feature selection on user-provided data.
- The tool is intended to select the feature subset which allows to construct the most accurate approximation model.
- Tool iteratively searches through different feature subsets in some order to determine which features are important for the quality of approximation.
- User may flexibly configure search options as well as models quality metrics.
- Feature ranking from GT SDA Ranker may be used to improve search speed

Selector search strategies

- At the moment user may select following models validation approaches:
  - *Use train data* - which is fastest and easisest, but is the least reliable, so should be avoided if possible
  - *Use internal validation* - good accuracy estimate but requires more time
  - *Use test sample* - best option, but requires independent test sample to be available

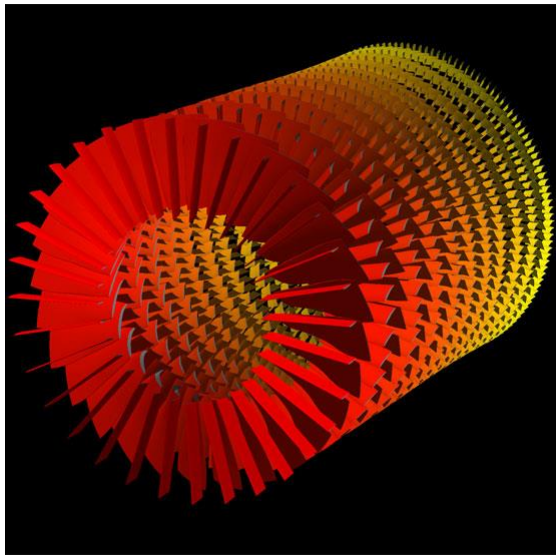
Selector search strategies

- Selector implements number of search strategies:
  - ***Full-search*** - checks all possible feature combinations *(may take very long time!)*,
  - ***Add*** - starts with empty feature list and tries to add features by one *(good choice if you have lots in inputs)*,
  - ***Del*** - starts with all features selected and tries to remove features by one *(good choice if you expect that you have dependent or redundant inputs)*,
  - ***AddDel*** - trade-off between previous two strategies

T-AXI: problem description

Description:

The T-C_DES (Turbomachinery Compressor DESign) code is a meanline axial flow compressor design tool



Task:

Determine what features affect Compressor Pressure Ratio (with IGV) output

Result:

GT SDA with **Pearson Partial Correlation** coefficient **on ~20000 points** budget determined that only **22 of 163 features significantly affect output**

T-AXI: source data

Initial input point $X^0 = (x^0_1, \ldots, x^0_{163})$, see tables 5,6 and 7

Table 5. Stage data for 10 stage design (stage.e3c-des).

Parameter	Stage 1	2	3	4	5	6	7	8	9	10
Stage rotor inlet angle [deg]	10.3	13.5	15.8	18	19.2	19.3	16.3	15	13.6	13.4
Stage rotor inlet Mach no.	0.59	0.51	0.475	0.46	0.443	0.418	0.402	0.383	0.35	0.313
Total Temperature Rise [K]	52.70	52.30	51.12	49.74	49.14	43.62	45.69	47.27	48.26	47.57
Rotor loss coef.	0.053	0.0684	0.0684	0.0689	0.069	0.069	0.069	0.069	0.069	0.07
Stator loss coef.	0.07	0.065	0.065	0.06	0.06	0.065	0.065	0.065	0.065	0.1
Rotor Solidity	1.666	1.486	1.447	1.38	1.274	1.257	1.31	1.317	1.326	1.391
Stator Solidity	1.353	1.277	1.308	1.281	1.374	1.474	1.379	1.276	1.346	1.453
Stage Exit Blockage	0.963	0.956	0.949	0.942	0.935	0.928	0.921	0.914	0.907	0.9
Stage bleed [%]	0	0	0	0	1.3	0	2.3	0	0	0
Rotor Aspect Ratio	2.354	2.517	2.33	2.145	2.061	2.028	1.62	1.417	1.338	1.361
Stator Aspect Ratio	3.024	2.98	2.53	2.21	2.005	1.638	1.355	1.16	1.142	1.106
Rotor Axial Velocity Ratio	0.863	0.876	0.909	0.917	0.932	0.947	0.971	0.967	0.98	0.99
Rotor Row Space Coef.	0.296	0.4	0.41	0.476	0.39	0.482	0.515	0.58	0.64	0.72
Stator Row Space Coef.	0.32	0.35	0.45	0.45	0.9	0.46	0.89	0.52	0.58	0.55
Stage Tip radius [m]	0.351	0.336	0.328	0.321	0.315	0.308	0.304	0.300	0.297	0.295

Table 6. Initial data for 10 stage design (init.e3c-des).

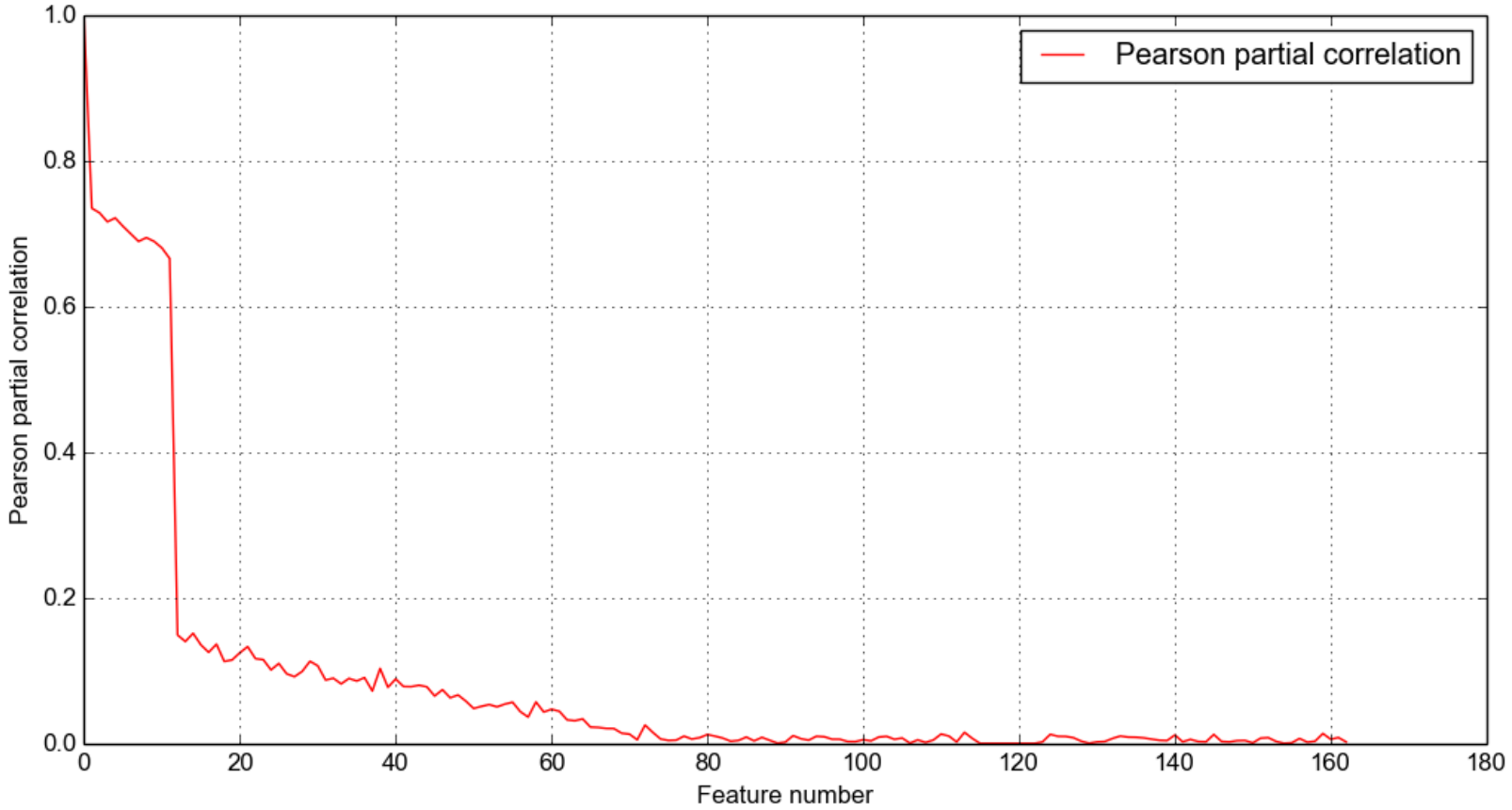
Number of Stages	10	fix
Mass Flow Rate [kg/s]	54.4	
Rotor Angular Velocity [rpm]	12,299.49	
Inlet Total Pressure [Pa]	101,325	
Inlet Total Temperature [K]	288.15	
Alpha 3 - Last Stage [deg]	0	fix
Mach 3 - Last Stage	0.272	
Ratio of Specific Heats	1.37836	fix
Gas Constant [kJ/kg*K]	0.287	fix
Clearance Ratio	0.0015	

Table 7. IGV data for 10 stage design (igv.e3c-des).

Soldity	0.6776	
Aspect ratio	5.133	
Phi Loss Coef.	0.039	
Inlet Angle	0	fix
Inlet Mach	0.47	
Lambda	0.97	
IGV Row Space Coef.	0.4	
IGV Tip Radius [m]	0.36211	

163 inputs = 15·10 inputs (table 5) + 6 inputs (table 6) + 7 inputs (table 7)

T-AXI results ($\pm 20\%$)



T-AXI results (±20%)

Table 5. Stage data for 10 stage design (stage.e3c-des).

	Stage									
Parameter	1	2	3	4	5	6	7	8	9	10
Stage rotor inlet angle [deg]	10,3	13,5	15,8	18	19,2	19,3	16,3	15	13,6	13,4
Stage rotor inlet Mach no.	**0,59**	0,51	0,475	0,46	0,443	0,418	0,402	0,383	0,35	0,313
Total Temperature Rise [K]	**52,696**	**52,301**	**51,117**	**49,736**	**49,144**	**43,617**	**45,69**	**47,269**	**48,255**	**47,565**
Rotor loss coef.	0,053	0,0684	0,0684	0,0689	0,069	0,069	0,069	0,069	0,069	0,07
Stator loss coef.	0,07	0,065	0,065	0,06	0,06	0,065	0,065	0,065	0,065	0,1
Rotor Solidity	1,666	1,486	1,447	1,38	1,274	1,257	1,31	1,317	1,326	1,391
Stator Solidity	1,353	1,277	1,308	1,281	1,374	1,474	1,379	1,276	1,346	1,453
Stage Exit Blockage	**0,963**	0,956	0,949	0,942	0,935	0,928	0,921	0,914	0,907	0,9
Stage bleed [%]	0	0	0	0	1,3	0	2,3	0	0	0
Rotor Aspect Ratio	2,354	2,517	2,33	2,145	2,061	2,028	1,62	1,417	1,338	1,361
Stator Aspect Ratio	3,024	2,98	2,53	2,21	2,005	1,638	1,355	1,16	1,142	1,106
Rotor Axial Velocity Ratio	**0,863**	0,876	0,909	0,917	0,932	0,947	0,971	0,967	0,98	0,99
Rotor Row Space Coef.	0,296	0,4	0,41	0,476	0,39	0,482	0,515	0,58	0,64	0,72
Stator Row Space Coef.	0,3	0,336	0,438	0,441	0,892	0,455	0,886	0,512	0,583	0,549
Stage Tip radius [m]	**0,3507**	**0,3358**	0,3283	0,3212	0,3151	0,3084	0,3042	0,2995	0,2970	0,2946

Table 6. Initial data for 10 stage design (init.e3c-des).

Mass Flow Rate [kg/s]	**54,4**
Rotor Angular Velocity [rpm]	**12299,494**
Inlet Total Pressure [Pa]	**101325**
Inlet Total Temperature [K]	**288,15**
Mach 3 - Last Stage	0,272
Clearance Ratio	0,0015

Table 7. IGV data for 10 stage design (igv.e3c-des).

Soldity	0.6776
Aspect ratio	5.133
Phi Loss Coef.	0.039
Inlet Mach	**0.47**
Lambda	**0.97**
IGV Row Space Coef.	0.4
IGV Tip Radius [m]	**0.36211**

The most important feature is filled with dark green, next 10 important ones are filled with light green color, and next 11 less important variables are filled with orange color

Eurocopter test case: data

Modelling of helicopter loads for different maneuver types

- 66 load parameters
- 2 outputs (static and dynamic load)
- 32 maneuver types
- In total, 66 x 2 x 32 = 4 224 models to build
- The training sample size from 0 to 108
- Result (CHAMALO project): about 50% of missing loads may be calculated with a sufficient accuracy (<20%)

Eurocopter test case: data

Data is quite challenging:

- High input dimension (35)
- Low sample size (< 108)

Hint:

- Expert feature selection (about 8-10 features)

Could we do better then expert?

Eurocopter test case: data

Our approach:

- Rank features by GT SDA Ranker
- Select best features by GT SDA Selector
- Validate result by GTApprox IV (LOO)

Subset of 7 pairs maneuver-load was selected to test the approach.

Eurocopter test case: results

Problem/ Algorithm		Expert features	Selector full	Selector expert
AMOAR1_TC 42 points	RRMS	0,237	0,104	0,169
	Features	[0, 1, 2, 3, 16, 20, 23, 24, 25]	[0, 7, 9, 11, 17, 30]	[0, 1, 16, 20, 25]
AMDAR1_TC 87 points	RRMS	0,278	0,195	0,184
	Features	[0, 1, 2, 3, 16, 20, 23, 24, 25]	[0, 7, 9, 12, 14, 17, 18, 26, 30]	[0, 1, 2, 20, 23, 25]
AMDAR1_L 61 points	RRMS	0,488	0,309	0,354
	Features	[0, 1, 16, 23, 24, 25, 31, 32]	[0, 1, 6, 7, 22, 27, 29, 30]	[24, 25, 31, 32]

Eurocopter test case: results

Problem/ Algorithm		Expert features	Selector full	Selector expert
PT291J_TD 37 points	RRMS	0,326	0,271	0,268
	Features	[0, 1, 2, 3, 16, 20, 23, 24, 25]	[0, 2, 3, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 27, 29, 30]	[0, 2, 20, 23, 25]
PB371J_TD 85 points	RRMS	0,172	0,119	0,163
	Features	[0, 1, 2, 3, 16, 20, 23, 24, 25]	[2, 4, 8, 9, 11, 21, 24, 25, 30]	[0, 1, 2, 20, 25]
CFIX_F 107 points	RRMS	0,179	0,133	0,148
	Features	[0, 1, 2, 3, 16, 23, 24, 25]	[0, 1, 2, 3, 7, 10, 20, 21, 25]	[0, 1, 3, 25]

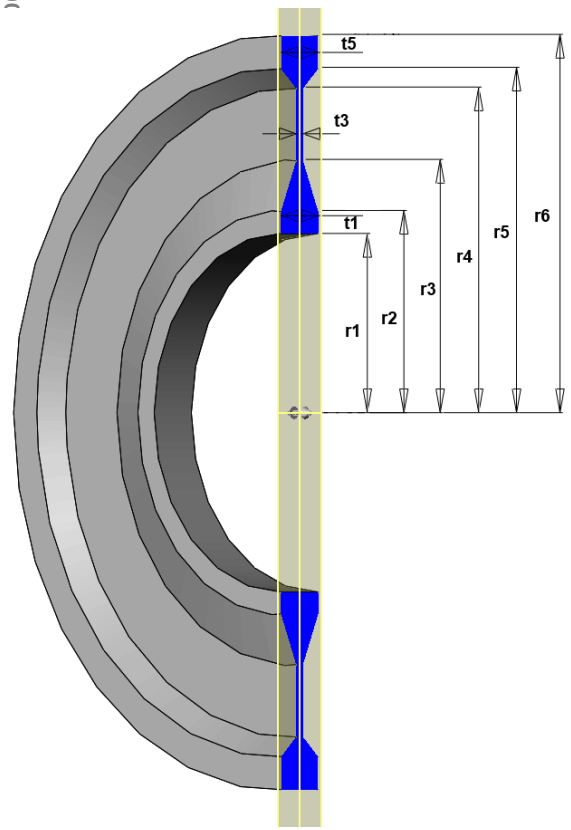
Eurocopter test case: results

Problem/Algorithm	RRMS	Features
Expert features	0,436	[0, 1, 2, 3, 16, 22, 23, 24, 25, 29]
Selector full, AddDel	0,286	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 26, 27]
Selector expert, AddDel	0,226	[0, 1, 3, 16, 24, 29]
Selector full, AddDel, OneStepFullSearch=on	0,0756	[3, 4, 8, 12, 23, 27, 28]
Selector expert, AddDel, OneStepFullSearch=on	0,182	[0, 3, 23, 24, 25, 29]

Eurocopter test case: conclusions

- GTSDA allows to intelligently select subset of features with smaller error then expert subset.
- Subset of features selected is normally better then expert choice.
- Some of expert features come to selected subset.
- Playing with feature selection technique allows to improve results significantly.

Rotating disk example

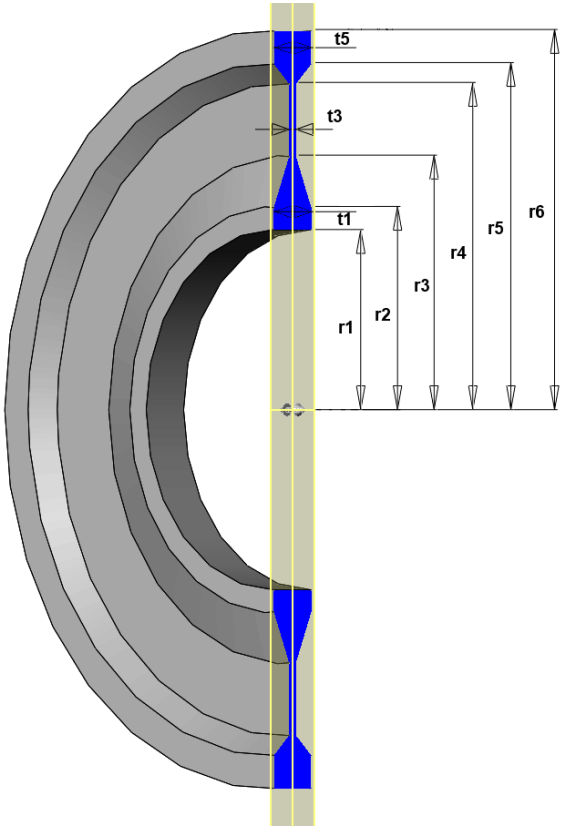


Here we provide an example of how GT SDA can be of help in solving an optimization problem.

Consider a problem of designing geometry of rotating disk shown on picture.

Our goal is to create disk of minimum mass that still satisfies given mechanical and stability constraints.

Rotating disk example



We already have our geometry parametrized for us and the disk may be represented as a vector of 9 numbers (3 thickneses and 6 radiuses)

Let us set values for initial configuration and fix several of them:

$r1 = 109\ mm$
$r2 = 123\ mm$
$r3 = 154\ mm$
$r4 = 198\ mm$
$r5 = 210\ mm$
$r6 = 230\ mm$

$t1 = 32\ mm$
$t3 = 6\ mm$
$t5 = 32\ mm$

$10 \le r1 \le 110$
$4 \le t1 \le 50$
$120 \le r2 \le 140$
$150 \le r3 \le 168$
$4 \le t3 \le 50$
$170 \le r4 \le 200$

So we have dependency on 6 parameters

Rotating disk example

The goal of optimization is to

- **Minimize mass of disk** $\min M$

While satisfying constraints on

- **maximal tension** $s_{eqv} \leq 600\ MPa$
- **Radial displacement on outer disk radius** $u2 \leq 0.3\ \text{mm}$

Conclusion

- GT SDA implements number of state of the art techniques that allow to address the following questions:
  - Is there a dependency between corresponding pair of features? *(with ability to check statistical significance)*
  - What features affect output the most?
  - What feature subset is best to construct surrogate model with?
- The tool is in active development and new possibilities for data analysis is being