DATADVANCE
AN EADS COMPANY

MACROS

GTIVE

Generic Tool for Important Variable Extraction

Contact information

Phone	+7 (495) 781 60 88	
Web	`www.datadvance.net`	
Email	`support@datadvance.net`	Technical support, questions, bug reports
	`info@datadvance.net`	Everything else
Mail	DATADVANCE, llc Pokrovsky blvd. 3 building 1B, 4 floor 109028 Moscow Russia	

User manual prepared by Pavel Erofeev, Pavel Prikhodko, Evgeny Burnaev

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 What is GTIVE

Generic Tool for Important Variable Extraction (**GT IVE**) is a software package for performing global sensitivity analysis on user-provided data. In the [13] sensitivity analysis is defined as the study of how the variation (uncertainty) in the output of a statistical model can be attributed to different variations in the inputs of the model. In other words it is a technique for systematically changing variables (features) in a model to determine the effects of such changes.

1.2 Documentation structure

Documentation for **GTIVE** includes:

- User manual (this document) which contains:
  - A general overview of the tool's functionality;
  - Short descriptions of the algorithms;
  - Recommendations on the tool's usage;
  - Examples of applications to model problems.
- Technical reference [3] for C++ and Python API which includes:
  - Description of system requirements;
  - Installation steps;
  - Quick start guide;
  - C++ and Python API reference.

The present document has the following structure:

- Chapter 2 is an introduction to the tool's functionality. It contains an overview of relevant sensitivity analysis concepts, and explains the way the tool is applied and what results it produces.
- Chapter 3 describes the internal workflow of the tool.

- Chapter 4 describes specific sensitivity analysis techniques implemented in the tool.
- Chapter 5 describes limitations on the sample size for different techniques.
- Chapter 6 describes how the sensitivity analysis technique is selected automatically in a particular problem.
- Chapter 7 gives some examples of **GTIVE** tool use for some model and real world problems.

Chapter 2

Overview

The main goal of **GT IVE** is to estimate feature scores for the user-provided dependency [1] which can be represented as *data sample* [2] or interface to some *black box* [3]. So it solves the problem of global sensitivity analysis.

As an illustration we give the following simple example. Consider the Newton's law of universal gravitation. Say we know that every point mass attract every other point mass, but don't know what features affect that.

And say, that for some reason we think that following features may affect the force of attraction:

- m_1, m_2 - the masses of the bodies
- r - distance between bodies
- T - environment temperature
- p - atmospheric pressure
- L_1, L_2 - bodies luminosity

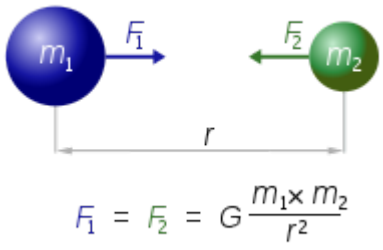

Figure 2.1: The Newton's law of universal gravitation

Also we did 30 experiments and measured all the features considered and the corresponding force of attraction. Applying **GT IVE** to this task give us the following feature ranks, see 2.1

In general the tool helps to answer the following questions:

[1] also known as function or model
[2] also known as *training data* (or *samples*)
[3] some device, system or object that provides output for a given input

m_1	m_2	r	T	p	L_1	$L2$
0.19	0.20	0.61	0.0	0.0	0.0	0.0

Table 2.1: Illustration. Scores for the Newton's law of universal gravitation problem

1. What features have no influence on the dependency and thus can be dropped in the further study?
2. If we want to reduce the number of features considered in the problem which features should we drop?
3. What features are the most influential so that they should be measured with the highest accuracy or have the highest variability in the Design of Experiments?

GTIVE calculates sensitivity indices (features scores) for each input variable (feature). That are the numbers that show relative importance of each feature in some sense. Looking at the scores one can say if one feature is more important than the others and guess to what extent. This information may be useful in the following tasks:

- In the **Surrogate Model (SM)** construction it may be beneficial to remove the least important features, because less features mean more dense sample and denser sample may provide more accurate approximation. Also many **SM** construction techniques may work better in smaller dimensions in terms of time/memory requirements.
- In the **Design of Experiment** knowing what features influence dependency the most one can plan the sample generation in a way that most important features have the highest variability. Also, if data is obtained as some physical measurements, knowing feature scores may tell what input variables should be measured with the highest accuracy.
- In the **Optimization**, when the number of allowed function calls (budget) is limited, knowing what features are less important allows for not changing them in the optimization process. Reducing number of variables by not considering features that have little effect on the dependency, one can do more optimization iterations with the same budget possibly acquiring better solution.

Examples of **GTIVE** applications to the mentioned above tasks are presented in the Chapter 7.

In this chapter the sensitivity analysis problem statement is given and short review of the state of the art methods, used in the tool, is provided.

2.1 Problem statement

The problem of the global sensitivity analysis is to estimate how variations in the output of the model can be attributed to the variations in the model inputs on all design space.

Let $Y = f(X)$, $X \in R^p$, $Y \in R^q$ be some considered dependency. $f(X)$ may be some physical experiment or a solver code. Without loss of generality only the case of $q = 1$ will be considered below. If $q > 1$ (the model has many outputs) each output is treated

independently. **GTIVE** procedure calculates score w_i for each feature x_i from a feature set $X = (x_1, \ldots, x_p)$ also known as input vector such that higher score reveals more sensitivity (higher variations) of the output Y with respect to the variations of the corresponding input. The scores are positive numbers generally between 0 and 1, higher score indicates that the variable is "more important". There are several different techniques implemented in the tool; the precise meaning of the score is technique-dependent. For a sensitivity analysis technique we wish it to share the following properties:

- If one variable is more important than the other in a technique defined way, we want it's score to be higher
- We want feature scores to be proportional to the corresponding variables influence, so that comparing scores one would get the idea of relative importance of variables

These properties allow to rank features in the order of importance and give the idea of approximately to what extent one feature is more important than other features.

2.2 Quality metrics

To compare techniques performance the following measures could be introduced. These are intuitive straightforward ways to check the variable importance, however huge amount of data or time is required to evaluate them, so these measures are not very suited for practical use and are mostly useful as reference in the benchmarking of different sensitivity analysis methods.

- **Index of variability** may be used to compare importance of the features or even feature subsets if we can calculate dependency value in a given point.

  Let features in the vector X be split into two subsets $X = (Z(X), U(X))$, where the subvector $Z(X)$ contains all important features (features with high scores) and $U(X)$ contains all unimportant features (features with low scores). Let us define by $\hat{X}(X) = (Z(X), U_0)$ some vector, where all unimportant features are fixed to some average values.

  Then the Index of Variability can be computed as follows:

  $$I(Z) = \frac{\sqrt{< (f(X) - f(\hat{X}(X)))^2 >}}{max(f(X)) - min(f(X))} \cdot 100\%, \tag{2.1}$$

  where $< .. >$, max, min are some test sample mean, maximum and minimum. The higher index of variability the less important features are chosen in Z and the more important are fixed in U.

- **Approximation error ratio**. Another way to estimate i-th feature importance is to build an approximation (surrogate model) $f_{SM_i}(Z_i(X))$ where $Z_i(X) = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_p))$, i.e. input formed from X using all features except i-th, and compare it's accuracy with approximation $f_{SM}(X)$, built using all features. So the error measure can be defined as:

  $$Err(i) = \frac{\sqrt{< (f(X) - f_{SM_i}(Z_i(X)))^2 >}}{\sqrt{< (f(X) - f_{SM}(X))^2 >}}, \tag{2.2}$$

  where $< .. >$ is the sample mean. Higher approximation error ratio means that i-th feature is more important.

2.3 Input Definition Domain Importance

It's important to note that the scores returned by **GTIVE** depend on the variation intervals of the factors. If a factor is restricted to a very narrow interval, then its score might be low even if factor is important. Moreover, the scores returned by **GTIVE** are invariant under changes of units of measurement for individual factors (as long as changes are linear). In such cases the effects of rescaled intervals are compensated by the corresponding changes in the response function.

For example, consider the case when we have a function $f(x_1, x_2) = x_1 + x_2$, with $x_1 \in [-1, 1]$ and $x_2 \in [-1, 1]$. It's obvious to expect x_1 and x_2 to have equal scores in these conditions. Now, let us expand x_1 to region $[-2, 2]$, while keeping $f(x_1, x_2)$ the same. In this case though in each point local importance of x_1 and x_2 remains similar, on the global scale x_2 would provide 4 times more variation to the output, thus rising it's feature score. It's equivalent to the case when we leave x_1 at $[-1, 1]$ and change function to $f(x_1, x_2) = 2 * x_1 + x_2$. On the contrary, consider the case when we change the measurement units of the feature. For example, x_1 and x_2 were defined in kilograms and we want to change the measurement units of x_1 to grams. In this case, though new values of rescaled x_1 would become 1000 times larger but it's feature score would remain the same.

2.4 State of the art methods

There are lots of approaches to the problem of global sensitivity analysis [5, 13, 12, 7, 14, 8]. Technique appropriate for each task depends on the problem conditions and user requirements. We've designed the **GTIVE** tool to include the most effective state of the art methods, covering different problem settings. In this section brief overview of the techniques used in the **GTIVE** is provided.

We may group sensitivity analysis techniques in two big groups:

- Methods that can work with any sample.
- Methods that require sample of a particular structure to work.

Generally, the methods of the second group are more precise, but due to the sample form requirements one usually needs to have an interface to the considered function to be able to generate required specific sample.

For each situation different techniques are implemented in the **GTIVE** and we refer to them as *sample based* and *black box based* correspondingly.

2.4.1 Sample based techniques

These techniques require some data sample $(\mathbf{X}, \mathbf{Y})$ given, where $\mathbf{X} = \{X^i, i = 1, \ldots, K\}$, $\mathbf{Y} = \{Y^i, i = 1, \ldots, K\}$, components of input vector $X^i = (x_1^i, \ldots, x_p^i)$, $Y^i = f(X^i)$, K is the total number of samples.

In the **GTIVE** the following sample based techniques are implemented:

- **RidgeFS**

  In case the sample is small and so there is no benefit in using complex approaches feature scores may be estimated with linear model.

It is assumed that $\mathbf{Y} = \mathbf{X}b + \varepsilon$, $b = (b_1, \ldots, b_p)$ are some coefficients and $\varepsilon = \{\varepsilon^i, i = 1, \ldots, K\}$ is zero mean white noise. Coefficients b are estimated as $\hat{b} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)\mathbf{X}^T\mathbf{Y}$, where $\mathbf{I} \in R^{p\times p}$ and λ is tuned using LOO CV approach, see [5].

Then feature score for i-th variable is estimated as

$$w_i = \frac{\widehat{b_i}^2 / var(x_i)}{var(\mathbf{Y})}, i = 1, \ldots, p, \tag{2.3}$$

where $var(x_i)$ is a variance of the i-th feature, estimated using sample.

Pros:

- ◇ Works fast
- ◇ Can handle very large data sets
- ◇ Best possible choice if the true model is linear

Cons:

- ◇ Not suitable for strongly non linear models

- **Mutual Information**

A group of techniques that estimate feature score by computing Mutual Information of considered feature and the output:

$$I(x_i, Y) = \int p(x_i, Y)\, log \frac{p(x_i, Y)}{p(x_i)\, p(Y)} dx_i dY. \tag{2.4}$$

The idea is to measure how far the joint distribution $p(x_i, Y)$ of the feature and the output is from the case of two independent random values where $p(x_i, Y) = p(x_i)\, p(Y)$. The greater the difference the more relevant feature is. Feature score for i-th variable is estimated as:

$$w_i = I(x_i, Y), i = 1, \ldots, p. \tag{2.5}$$

In the **GTIVE** we adopted two techniques to estimate Mutual Information (*kraskov* and *histogram* estimates). *Kraskov* estimate gives more accurate results, but becomes computationally expensive and so can't be used for large data samples. *Histogram* based estimate may be crude on small samples, but is very cheap in terms of memory and computation time, so it can be applied to a very large data sets.

In more details:

– **Kraskov estimate** is an estimation of Mutual Information technique based on nearest neighbor approach. The technique provides good accuracy for small and moderate sample sizes, but becomes very computationally expensive in case of large samples. Define a metric in space $Z = (X, Y)$ as $\rho_z(Z, Z^*) = \max(\rho_x(X, X^*), \rho_y(Y, Y^*))$, where $\rho_x(X, X^*)$ is the Euclidean norm in the X space and $\rho_y(Y, Y^*)$ is the Euclidean norm in the Y space.

Let k be the algorithm parameter setting number of nearest neighbors in the Z space, then let

$$\epsilon(j) = \rho_z(Z^j, \mathbf{k}\text{-}th\, neighbor\, of\, Z^j) \tag{2.6}$$

. We set n_x^j and n_y^j as number of points in the X and Y spaces correspondingly whos distance to X^j and Y^j is smaller than $\epsilon(j)$.

In [8] it's shown that

$$I_k\left(x_i, y\right) \approx \psi(k) - \langle \psi(n_x + 1) + \psi(n_y + 1) \rangle - \psi(K), \tag{2.7}$$

where $\langle ... \rangle$ is the sample mean, k is the number of nearest neighbors (algorithm parameter), $\psi(z)$ is Euler digamma function.

– **Histogram based estimate** is an estimate of Mutual Information technique using histogram based pdf estimation. Method may be less accurate than previous one in case of small and moderate samples, but can handle very large data sets. In this approach pdf of x_i, Y and pdf of (x_i, Y) are estimated using histograms. For example pdf of x_i is estimated as

$$\hat{p_i}(x) = \frac{\sum_{j=1}^{K} I(x_i^j \in (x_i - h/2,\, x_i + h/2))}{Kh}, \tag{2.8}$$

where h is a bin size, $I(\cdot)$ is an indicator function. In the **GTIVE** implementation the cross validation approach is used to estimate optimal histogram bin size h, see [5]. If the sample size is at least 20000 points, then accelerated optimization procedure for the bin size selection is used.

Pros:

◇ Works fast

◇ Can handle small as well as large data sets. Sample of few dozens points is sufficient to catch the most important features. As the sample size increase resolution grows.

◇ Robust to noise and outliers

Cons:

◇ Cant handle feature interdependencies

- **SMBFAST (Surrogate Model-Based FAST)**

Surrogate Model-Based FAST is a complex approach combining the surrogate modeling paradigm and the idea of black box analysis with the extended FAST method (see 2.4.2). Currently all GTApprox techniques except the Mixture of Approximators and Geostatistical Gaussian Processes are available in SMBFAST for training the internal surrogate model, and same features and restrictions apply (see the **GT Approx** manual [2] for details).

Due to the model training overhead, SMBFAST may be time consuming but it is the most accurate of all currently implemented sample-based techniques.

Pros:

◇ The most accurate of all currently implemented sample-based techniques

◇ Incorporates approximation capabilities of **GT Approx**

Cons:

◇ May take a long time (as building of **GT Approx** model inside is required)

2.4.2 Black box based techniques

These techniques generate new sample points during their work so they require connection to some black box function $Y = f(X)$. In case of black box based method term *budget*[4] is used instead of sample size.

Note that in these methods one has to specify the region (some hypercube) where points are generated.

In the **GTIVE** the following black box based techniques are implemented:

- **Elementary Effects** is a screening technique able to work with relatively small samples.

  The idea of Elementary Effects approach is to generate uniform (in terms of space-filling properties) 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:

  $$d_i(X) = \frac{Y(x_1, \ldots, x_i + \delta_i, \ldots, x_p) - Y(x_1, \ldots, x_i, \ldots, x_p)}{\delta_i}, \tag{2.9}$$

  where δ_i is a step size. Score for i-th feature is computed as

  $$w_i = \frac{\Delta_i^2 \, \mu_i}{\pi^2 \, var(\mathbf{Y})}, \; i = 1, \ldots, p, \tag{2.10}$$

  where $\mu_i = \frac{1}{r} \sum_{j=1}^{r} d_i^2(X^j)$, r is a number of steps changing i-th feature value on all trajectories, X^j is the input value at these steps; Δ_i is a range of possible values for i-th feature; $var(\mathbf{Y})$ is a sample variance of black box values on generated sample points. Actually the method gives normalized estimate of average squared partial derivatives.

  Pros:

  - ◇ Can provide reliable estimates even for very small budgets. Minimal number of black box function calls equals few times number of features which is sufficient to get estimation for not very complex cases

  Cons:

  - ◇ Generates trajectories randomly
  - ◇ Not robust to outliers

- **Extended FAST (Fourier Amplitude Sensitivity Testing)** is a technique suited for the case when cheap black box is available (like surrogate model, see 2.4.1). It requires quite many samples to estimate score.

  The idea here is to measure what portion of output variance is described by the variance of the feature. To do so for each feature main indices are estimated as

  $$S_i = \frac{V_{x_i}[E_{\sim x_i}(Y|x_i)]}{V(Y)}, \tag{2.11}$$

[4] number of function calls allowed for method

where $V_{x_i}[\cdot]$ is a variance with respect to x_i, $E_{\sim x_i}(\cdot|x_i)$ is a conditional mean with respect to all features except x_i. Instead of computing multivariate Monte Carlo estimates, method uses space filling one-dimensional curves of the form

$$x_i(s) = \frac{1}{2} + \frac{1}{\pi} arcsin(sin(v_i s + \phi_i)) \tag{2.12}$$

to generate sample points. Here each feature have some frequency v_i assigned from some incommensurate set v_i, s is the coordinate on one-dimensional curve and ϕ_i is a some random constant phase shift. Using Fourier decomposition in case of (2.12) we may say that

$$f(X) = \sum_{j=-\infty}^{\infty} (A_j \cos(js) + B_j \sin(js)),$$

$$A_j = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(s) cos(js) ds,$$

$$B_j = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(s) sin(js) ds.$$

These integrals can be estimated using points generated on the curve (2.12). In this case, e.g. conditional variance can be estimated as

$$V_{x_i}[E_{\sim x_i}(Y|x_i)] = 2 \sum_{j=1}^{K} (A_{jv_i}^2 + B_{jv_i}^2), \quad jv_i \text{ is an integer}, \tag{2.13}$$

where K is some predefined number.

Another appealing property of this approach is it's ability to accurately estimate total indices. In this case all cross-variable interactions that include i-th feature are taken into account in the corresponding scores, i.e. the score is estimated as follows:

$$S_i = 1 - \frac{V_{\sim x_i}[E_{x_i}(Y|x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_p)]}{V(Y)}. \tag{2.14}$$

To do this estimation unique frequency v_i is given to x_i and the same frequency v is given to all other features, then the same procedure as above is performed.

The score for i-th feature is

$$w_i = S_i, i = 1, \ldots, p. \tag{2.15}$$

Pros:

- ◇ Can give main effect as well as total effect estimations
- ◇ Needs less samples than for most of other variance based approaches (about 72 points per feature is recommended)

Cons:

- ◇ Still requires relatively large samples

What technique to choose in each case is decided by the initial problem conditions (we have sample or black box) and best practice. For details see Chapter 6.

2.5 Scores variance estimation

It's possible to compute score estimation variances to check how reliable obtained scores values are.
When one obtains score and estimation of variance one may expect that there is high probability (usually estimated at 99.99966%) of the true score value lying inside the

$$[score - 3 \cdot \sqrt{variance},\ score + 3 \cdot \sqrt{variance}]$$

range. So if zero is outside of this range one may decide that score is significantly larger than zero. It means that corresponding feature has significant influence on the function value, and this feature can be treated as important.

Actually, estimation of true confidence intervals for scores is quite a complicated problem. However, we consider that our approximation for confidence intervals is sufficiently accurate to help in selection of the important features.

2.6 Remark on other sensitivity analysis methods

In this section we will discuss **GTIVE** methods with respect to well-known Pearson's and Spearman's correlation coefficients.

Let us consider the limitations of these correlation coefficients:

- Pearson's correlation coefficient is suitable only for using with linear functional dependencies. There is an analog of such a technique in **GTIVE**, namely RidgeFS.
- Spearman's correlation coefficient is suitable only for monotonic functions. In **GTIVE** we do not make such assumptions for *nonlinear* techniques (i.e. for all except RidgeFS).

To clarify these points, we will give an example. Let us consider the sensitivity analysis problem for a function $f = x^2 + 2y^2$, $x, y \in [-1, 1]$. In this case, nonlinear **GTIVE** techniques are supposed to identify correctly the presence of dependency and the influence of each variable on the output. The results are summarized in the table 2.2 (for uniformity, **GTIVE** scores are given after taking the square root). As expected, since the function is not linear and monotonic, the first three techniques gave inaccurate results.

2.7 Remark on the selection of techniques for GTIVE

The selection of techniques for **GTIVE** was associated with different factors.

1. The need to provide basic modes of operation:
   - reliable linear solution on a small sample: RidgeFS
   - medium-size sample, from 50 to 500 points: Mutual Information (kraskov)
   - large sample, from 200 to several hundred thousand points: Mutual Information (histogram)
   - black box with small budget, from $2 \cdot (\text{inputDimension} + 1)$ to ≈ 2000: Elementary Effects

Technique	X	Y
Pearson	59%	41%
Spearman	78%	22%
RidgeFS	74%	26%
Mutual Inf (kraskov)	33%	67%
Mutual Inf (hist)	34%	66%
Elementary Effects	35%	65%
FAST	34%	66%

Table 2.2: Pearson's and Spearman's correlation coefficients and **GTIVE** techniques.

- black box with large budget, from $65 \cdot \text{inputDimension}$ to hundreds of thousands: FAST

2. The popularity of techniques:

- RidgeFS is a standard linear estimate.
- Mutual Information is a widely used technique for feature selection in biology, medicine, image processing (e.g. see [17], [11], [10], [16]).
- Elementary Effects is a standard *screening* technique based on computation of average partial derivatives and recommended in [13].
- FAST is a common way to calculate so-called *global sensitivity indexes*. The efficient calculation of such indexes with FAST is described in [13] and [9]. Examples of usage of this approach are given in [15] and [19].

Chapter 3

Internal workflow

3.1 General workflow

As described in Section 2.4, **GTIVE** includes two types of techniques: blackbox- and sample-based. Main difference, regarding the tool's internal workflow, is that there is no preprocessing step in the blackbox-based mode since in this mode **GTIVE** generates the sample itself and ensures it has a correct structure and does not contain any degenerate data. Conversely, in sample-based mode the sample analysis is essential because in general there are no guarantees for the sample quality.

Thus **GTIVE** internal workflow generally consists of the following steps:

1. **Preprocessing.** Only in sample-based mode. In this step, redundant data is removed from the training set and the sample is normalized — see Section 3.2.

2. **Analyzing training data and options, selecting technique.** In this step, training sample properties and options specified by user are analyzed for compatibility, and the most appropriate estimation technique is selected — see Chapter 6.

3. **Estimating feature scores and scores standard deviation.** In this step, feature scores are estimated using the technique selected in the previous step.

   If the `VarianceEstimateRequired` option is on, the result also includes score standard deviation (std calculation is off by default). For vector functions (functions with multidimensional output), feature scores and scores standard deviation are estimated for each component independently — see Section 3.3 for the results description.

   For individual technique descriptions, see Chapter 4 and Section 2.4.

3.2 Preprocessing

As we work with initial training dataset some reasonable preprocessing must be applied to it in order to remove possible degeneracies in the data. Let $(\mathbf{X}, \mathbf{Y})$ be the $N \times (p+q)$ matrix of the training data, where the rows are $(p+q)$-dimensional training points, and the columns are individual scalar components of the input or output. The matrix $(\mathbf{X}, \mathbf{Y})$ consists of the sub-matrices $\mathbf{X}$ and $\mathbf{Y}$. We perform the following operations with the matrix $(\mathbf{X}, \mathbf{Y})$:

1. Remove all exact duplicates: search for rows in $(\mathbf{X}, \mathbf{Y})$ containing the same data and, if two or more matches are found, delete every row except one (since repeated data

points do not add any information). A warning is sent to log if there were any rows removed.

2. Remove all constant columns in sub-matrices $\mathbf{X}$ and $\mathbf{Y}$. A constant column means that all the training vectors have the same value of one of the input components. In particular for $\mathbf{X}$, this means that the training DoE is degenerate and covers only a certain section of the original design space. Column removals also produce a warning to log.

   As a result, we obtain a reduced matrix $\left(\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}}\right)$ consisting of the submatrices $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$. Accordingly, we define *effective input dimension* ($\tilde{p}$) as the number of columns in $\widetilde{\mathbf{X}}$, and *effective sample size* ($\tilde{N}$) as the number of rows in $\left(\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}}\right)$.

3. Next, sample values in the $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$ matrices are normalized so that for each component of the input and output its mean equals 0 and standard deviation equals 1:

$$x_i = \frac{x_i - \overline{x_i}}{\sigma(x_i)},\ y_i = \frac{y_i - \overline{y_i}}{\sigma(y_i)} \tag{3.1}$$

   This is the last sample preprocessing step if not using the Mutual Information technique. This means that for RidgeFS and SMBFAST techniques the scores are estimated using the normalized reduced matrix rather than the original matrix $\left(\mathbf{X}, \mathbf{Y}\right)$. Mutual Information technique includes one more preprocessing step below.

4. The Mutual Information technique is known to possibly show some performance degradation when feature values are distributed over a uniform grid (which is the case after the normalization). Due to this, in case of using the Mutual Information technique (whether Kraskov or histogram estimate), a small scale uniform noise in range $[-10^{-10}, 10^{-10}]$ is applied to all input and output components. If rank transform is on (see option `RankTransform`), the noise is applied after the transform. Thanks to its small scale, it does not have any significant effect on the final results, while the robustness of the Mutual Information technique is notably improved.

3.3 Results

The resulting output of **GTIVE** contains a feature score matrix $\mathbf{S}$ and, if std calculation is on (see option `VarianceEstimateRequired`), a score standard deviation matrix $\mathbf{D}$. The size of both matrices is $q \times p$: the number of rows is equal to the output dimension q, the number of columns is equal to the number of features, or the input dimension p (the original input dimension, not the effective input dimension $\tilde{p}$).

3.3.1 Feature scores

Each element s_{ij} of the $\mathbf{S}$ matrix is the sensitivity of the i-th output component to the j-th feature. In general, s_{ij} is a positive real number, except some special cases:

- In the sample-based mode, if the value of the j-th feature in the sample is constant (the $\mathbf{X}$ matrix contains a constant column), all scores of this feature (j-th column in $\mathbf{S}$) are set to NaN (special not-a-number value) since there is no way to estimate the sensitivity of the output to a constant component.

- In the sample-based mode, if the value of the i-th response component in the sample is constant (the $\mathbf{Y}$ matrix contains a constant column), the scores of all features *vs* this output (i-th row in $\mathbf{S}$) are set to 0.0 — it is assumed that this output is insensitive to all features since its value is constant.
- The first of the above rules has priority: if the sample contains both a constant feature x_j and a constant output y_i, the s_{ij} score is NaN.
- In the blackbox-based mode, if the generation region (see 2.4.2) is defined in such a way that the lower and upper bounds of some feature are equal, this feature is interpreted as a constant input, so its resulting score will be NaN, similarly to the sample-based mode with a constant column.

Note that **GTIVE** can't handle features collinearity. For instance, if the values of two features are always equal, they are assigned equal scores, while in reality it is possible that the output is totally insensitive to the first feature and changes its value only due to the change of the second feature. This is one of the examples of a degenerate data sample, and such features have to be filtered out before passing data to **GTIVE**.

3.3.2 Standard deviation

Standard deviation matrix $\mathbf{D}$ is structurally similar to the score matrix: each element σ_{ij} is the standard deviation of the s_{ij} score. In general, σ_{ij} is a non-negative real number, except the case than s_{ij} score is NaN. In this case, σ_{ij} is also set to NaN.

Note that standard deviation is calculated only when `VarianceEstimateRequired` is on, else the $\mathbf{D}$ matrix is empty.

Chapter 4

User configurable options

GTIVE combines a number of scores estimation techniques of different types. By default, the tool selects the optimal technique compatible with the user–specified options and in agreement with the best practice experience. Alternatively, the user can directly specify the technique through advanced options of the tool. This section describes the available techniques and it's options; selection of the technique in a particular problem is described in Chapter 6.

4.1 RidgeFS

Short name: LR

General description: Estimation of feature scores as normalized coefficients of regularized linear regression. Regularization coefficient is estimated by minimization of generalized cross-validation criterion [5]. Also, see Section 2.4.1.

Variance estimation: Yes

Restrictions: Can be applied to data sample only.

Strengths and weaknesses: A very robust and fast technique with a wide applicability in terms of the input space dimensions and amount of the training data. It is, however, usually rather crude, and the estimation can hardly be significantly improved by adding new training data.

Options: No options.

4.2 Mutual Information (Kraskov estimate)

Short name: Kraskov

General description: Mutual information estimate of feature scores based on the nearest neighbors information [8]. Also, see Section 2.4.1.

Variance estimation: Yes

Strengths and weaknesses: Is a robust nonlinear estimation technique, however can be applied only to small moderate samples due to memory limitations. Method tends to underscore features in case of heavy cross-feature interactions.

Restrictions: Can be applied to data sample only.

Options:

- `NumberOfNeighbors`

  Values: integer in range $[1, 0.8 \cdot (\text{effective sample size}) - 1]$.

  Default: 0 (auto).

  Short description: number of nearest neighbors used to estimate mutual information

  Description: Option specifies number of nearest neighbors used in estimation of mutual information if 'kraskov' technique is selected (manually or automatically). Increasing this value gives smaller variance of score estimation at the cost of higher systematic errors and vice versa. Best practice recommend to set it as a small integer value of around 5 in most cases.

- `RankTransform`

  Values: on, off

  Default: on

  Short description: Apply rank transform (copula transform) before computing mutual information.

  Description: If this option is on (True), rank transform is applied to the input sample before computing mutual information. In most cases, it allows for a more accurate mutual information estimate.

4.3 Mutual Information (Histogram based estimate)

Short name: Hist

General description: Mutual information estimate of feature scores based on the histogram construction. Also, see Section 2.4.1.

Strengths and weaknesses: Too crude for small samples, but have very low memory requirements so can be applied in the case of very large data sets. If the sample size is at least 20000, then accelerated optimization of histogram parameters is used. Tends to underscore features in case of heavy cross-feature interactions.

Variance estimation: Yes

Restrictions: Can be applied to data sample only.

Options:

- `RankTransform`

  Values: on, off

  Default: on

  Short description: Apply rank transform (copula transform) before computing mutual information.

  Description: If this option is on (True), rank transform is applied to the input sample before computing mutual information. In most cases, it allows for a more accurate mutual information estimate.

4.4 SMBFAST (Surrogate Model-Based FAST)

Short name: SMBFAST

General description: Surrogate Model-Based FAST combines the surrogate modelling and usage of extended FAST method. Also, see Section 2.4.1.

Strengths and weaknesses: SMBFAST may be time consuming but it is the most accurate of all currently implemented sample-based techniques.

Variance estimation: Yes

Restrictions: Can be applied to data sample only.

Options:

- `Accelerator`

  Values: integer in range [1, 5], or 0 (auto)

  Default: 0 (automatically set by the approximator)

  Short description: Five-position switch to control trade-off between speed and accuracy for the internal approximator used in SMBFAST.

  Description: Since SMBFAST builds a surrogate model (to be used as a FAST black-box), it actually uses **GT Approx** internally and makes certain options of this internal approximator available as **GTIVE** options. This option is essentially the same as `GTApprox/Accelerator`, except that 0 is also a valid value, meaning that the setting will be automatically selected by the internal approximator.

- `NumberOfCVFold`

  Values: integer in range [2, 2^{31}-2], or 0 (auto)

  Default: 0 (auto select)

  Short description: The number of cross-validation subsamples to estimate the variance of scores.

Description: In order to estimate the variance of scores, the principle of cross validation is used. Cross validation involves dividing the input sample into a number of subsamples (cross-validation subsets). This option sets the number of subsamples to divide in.

- `SensitivityIndexesType`

  Values: enumeration: total, main

  Default: total

  Short description: Select the type of score index to be computed.

  Description: This option is a switch selecting the type of index computed by the FAST procedure used internally in SMBFAST. Main index estimate is usually more reliable, but this index takes into account only the influence of the considered feature on the output, ignoring the influence of cross-feature interactions. Total index estimates total influence of the variable on the output, taking into account all possible interactions between the considered feature and other input features, but its estimate is generally less reliable.

- `SurrogateModelType`

  Values: enumeration: LR, SPLT, HDA, GP, HDAGP, SGP, GeoGP, TA, iTA, RSM, or Auto

  Default: Auto

  Short description: Specify the algorithm for the internal approximator used in SMBFAST.

  Description: Since SMBFAST builds a surrogate model (to be used as a FAST blackbox), it actually uses **GT Approx** internally and makes certain options of this internal approximator available as **GTIVE** options. This option is essentially the same as `GTApprox/Technique`. Default (Auto) selects a technique according to the GTApprox decision tree, with a single difference: HDAGP is never selected automatically, and where GTApprox would select HDAGP, the GP technique is used instead.

4.5 Elementary Effects

Short name: EE

General description: A screening technique estimating feature scores as an average of the function partial derivatives [13]. Also, see Section 2.4.2.

Strengths and weaknesses: Can work with very small budgets and still give reliable estimates in most cases, however may take time if the budget is big, due to complex problem of selecting appropriate set of trajectories. Note that method actually allows some randomization, so one can get different estimates by varying global `Seed` parameter.

Variance estimation: Yes

Restrictions: Can be applied to the black box only.

Options:

- `Deterministic`

  Values: boolean.

  Default: on.

  Short description: require IVE process to be deterministic.

  Description: If this switch is turned on, then all random processes in all algorithms are started with some fixed seed ensuring result to be the same on every run. In the current version the switch affects only black-box based techniques (FAST and Elementaty Effects).

- `Seed`

  Values: integer [1, 2147483647].

  Default: 100.

  Short description: change fixed seed when `Deterministic` is on.

  Description: Enables user to use different fixed seeds for IVE process. In the current version the switch affects only black-box based techniques (FAST and Elementaty Effects).

- `MinCurveNum`

  Values: integer [1, 2147483647].

  Default: 200.

  Short description: number of space filling curves tested to compute elementary effects. Also, see Section 2.4.2.

  Description: Option specifies number of curves to be used in estimation of elementary effects. The more curves is used the better parameter space is explored, resulting in more accurate scores estimation, however it takes additional time.

4.6 Extended FAST (Fourier Amplitude Sensitivity Testing)

Short name: FAST

General description: Variance based estimation of feature scores. Methods can estimate cross variable interactions as well as isolated (main) variable indices (which can be useful to some additional manual dependency analysis) [12]. Also, see Section 2.4.2.

Strengths and weaknesses: Needs large enough computational budget (number of function calls): at least 65 calls per feature to get stable estimate, — however is very precise (if the budget is enough) even in the case of strong variables inter-dependencies. Note that method actually allows some randomization, so one can get different estimates by varying global `Seed` parameter.

Variance estimation: Yes

Restrictions: Can be applied to the black box only.

Options:

- `Deterministic`

  Values: boolean.

  Default: on.

  Short description: require IVE process to be deterministic.

  Description: If this switch is turned on, then all random processes in all algorithms are started with some fixed seed ensuring result to be the same on every run. In the current version the switch affects only black-box based techniques (FAST and Elementaty Effects).

- `Seed`

  Values: integer [1, 2147483647].

  Default: 100.

  Short description: change fixed seed when `Deterministic` is on.

  Description: Enables user to use different fixed seeds for IVE process. In the current version the switch affects only black-box based techniques (FAST and Elementaty Effects).

- `SensitivityIndexesType`

  Values: enum: total, main .

  Default: total.

  Short description: selects type of score index to be computed

  Description: Switch selects if the FAST procedure should compute 'main' or 'total' score index. 'Main' index takes into account only isolated influence of the considered feature on the output ignoring the influence of cross-features interactions. 'total' index estimates total influence of the variable on the output, taking into account all possible interactions between the considered feature and other input features, but it's estimate is generally less reliable.

- `NumberOfSearchCurves`

  Values: integer [0, 2147483647].

  Default: 0 ("0" means auto selection: 4, if the budget is sufficient, and less otherwise).

  Short description: adds random multistart to FAST curves used for estimation of sensitivity indexes

  Description: Option allows performing multistart when building FAST space filling curves. It can potentially increase accuracy at the cost of increasing the budget requirements `NumberOfSearchCurves` times. Minimal allowable budget is equal to $65 \cdot \tilde{p} \cdot$ `NumberOfSearchCurves`, where $\tilde{p}$ is the effective dimension of input vector (the number of not-constant input factors).

Chapter 5

Limitations

The maximum size of the training sample, which can be processed by **GTIVE**, is primarily determined by the user's hardware. Necessary hardware resources depend significantly on the specific technique — see descriptions of individual techniques. Accuracy of estimation tends to improve as the sample size increases.

Technique	Input type	Performance on huge training sets	Other restrictions
RidgeFS	sample		linear dependencies only
Kraskov	sample	limited by available RAM	
Histogram	sample		
SMBFAST	sample	potentially long runtime	
EE	blackbox	potentially long runtime	
FAST	blackbox		

Table 5.1: Technique summary

Contrary to the maximum size, there is a certain minimum for the size of the training set (or for the available number of blackbox calls), which depends on the technique used. As explained in Section 3.2, this condition refers to the *effective* values, i.e. the ones obtained after preprocessing. An error with the corresponding error code will be returned if this condition is violated.

The requirements on minimum sample size (budget) are summarized in Table 5.2. For most techniques there are two different limits, depending on whether the calculation of scores standard deviation is required by user or not (see option `VarianceEstimateRequired`).

Table 5.2 denotes the following:

- $\tilde{p}$: the effective input dimension after the sample preprocessing.
- s: the `GTIVE/SMBFAST/NumberOfCVFold` option value.
- NN: the `GTIVE/MutualInformation/NumberOfNeighbors` option value. Corresponding limit is in effect only if the option is set by user.
- NR: the `GTIVE/FAST/NumberOfSearchCurves` option value. Corresponding limit is in effect only if the option is set by user.
- $\lceil x \rceil$ is the value of x rounded up (to the next integer).

Technique	Minimum size (bugdet)	
	std calculation on	std calculation off
RidgeFS	$\tilde{p}+2$	$\tilde{p}+1$
SMBFAST	$\left\lceil \frac{2\tilde{p}+3}{s-1} \right\rceil \cdot s$	$2\tilde{p}+3$
Mutual Information (Kraskov)	20, or $\left\lceil \frac{NN+1}{0.8} \right\rceil$	20, or $NN+1$
Mutual Information (histogram)	3	3
EE	$2(\tilde{p}+1)$	$\tilde{p}+1$
FAST	$65\tilde{p} \cdot 3$, or $65\tilde{p} \cdot NR$, $NR \geq 3$	$65\tilde{p}$, or $65\tilde{p} \cdot NR$

Table 5.2: Minimum sample size (blackbox budget) for GTIVE techniques

Chapter 6

Selection of technique

This section details manual and automatic selection of one of the techniques described in Chapter 4.

6.1 Selection of the technique by the user

The user may specify the technique by setting the option `Technique`, which may have the following values:

- `Auto` — best technique will be determined automatically (default)
- `RidgeFS`
- `Mutual Information` — to select specific estimation type additional parameter `/MutualInformation/Algorithm` may be specified having possible values 'kraskov' for Kraskov estimation or 'hist' for histogram based approach. If none is specified than 'kraskov' estimate is used if there is < 500 sample points and 'hist' estimate is used otherwise. If 'hist' estimate is used and the sample size is at least 20000, then accelerated optimization of 'hist' parameters is used.
- `SMBFAST`
- `ElementaryEffects`
- `FAST`

6.2 Default automatic selection

The decision tree, describing the default selection of the estimation technique is shown in Figure 6.1. The factors influencing the choice are:

- Input type, i.e. sample or blackbox.
- Sample size (for blackbox, budget) K and effective input dimension $\tilde{p}$ of the training sample.

The result is the estimated feature scores. The selection is performed in agreement with properties of individual technique as described in Chapter 4. In particular for the sample input:

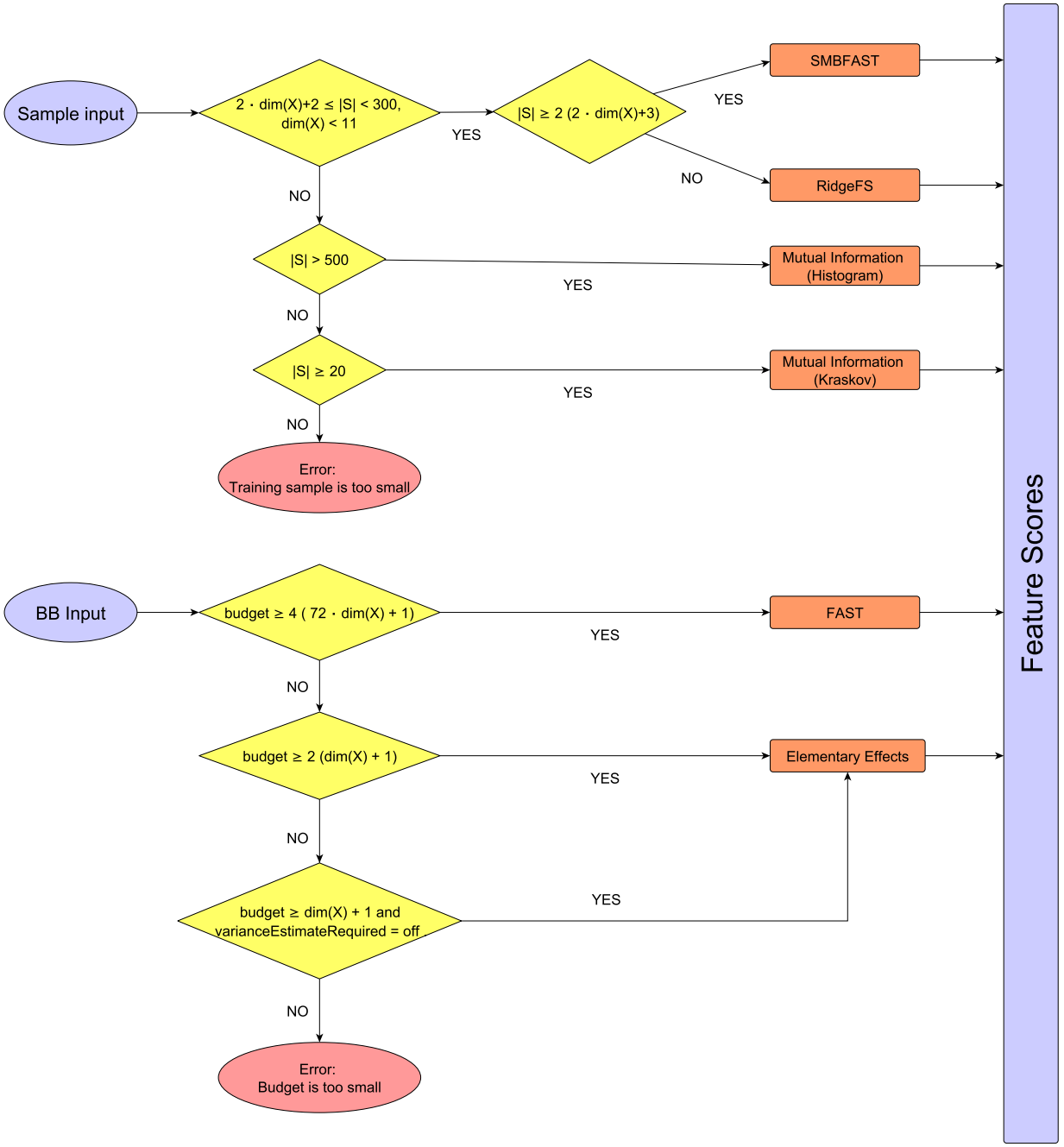

Figure 6.1: The **GTIVE** internal decision tree for the choice of default estimation method

- If $\tilde{p} \leq 10$, $K < 300$, and $2\tilde{p} + 2 \leq K < 2 \cdot (2\tilde{p} + 3)$, RidgeFS is selected.
- If $\tilde{p} \leq 10$, $K < 300$, but $K \geq 2 \cdot (2\tilde{p} + 3)$, SMBFAST is selected.
- In other cases, Mutual Information is selected, which uses the Histogram technique if $K > 500$ (and accelerated histogram estimate if $K > 20000$), and Kraskov if $20 \leq K \leq 500$.

For the black box input:

- If $K \geq 4 \cdot (72\tilde{p} + 1)$ then the FAST technique is chosen.
- If $2 \cdot (\tilde{p} + 1) \leq K < 4 \cdot (72\tilde{p} + 1)$ then the Elementary Effects is used.
- If $\tilde{p} + 1 \leq K < 2 \cdot (\tilde{p} + 1)$, the tool will start only if score variance estimation is not required by user (see option `VarianceEstimateRequired`). Otherwise, if variance estimation is required or $K < \tilde{p} + 1$, the tool will not start.

Chapter 7

Usage Examples

In this section we will apply **GTIVE** to some artificial model functions and some real world data sets to demonstrate method properties.

7.1 Artificial Examples

In this section we will demonstrate performance of various techniques implemented in the **GTIVE** on some known artificial functions.

7.1.1 Example 1: simple function, no cross-feature interaction

In this example we will consider the function:

$$f(x_1, x_2, x_3, x_4, x_5) = x_1^2 + 2x_2^2 + 3x_3^2 + 4x_4^2 + 5x_5^2,\ x_i \in [-1,\ 1],\ i = 1, \ldots, 5 \tag{7.1}$$

.

In this case we have no cross-feature interactions. So we can approximately estimate that true scores should have ratio $1 : 4 : 9 : 16 : 25$. In this example, we will refer to these scores as True.

We've calculated feature scores with all methods for different sample sizes and presented comparison with our expectations of what true features might be in this problem in the tables below.

Results for RidgeFS are presented in the Table 7.1. As expected, RidgeFS assumes linear dependency, so methods fails to estimate correct scores.

Sample size	x_1	x_2	x_3	x_4	x_5
True	0,0181	0,0727	0,1636	0,2909	0,4545
30	0,1847	0,1628	0,1113	0,2425	0,2983
100	0,2164	0,2074	0,1687	0,2128	0,1944
500	0,1449	0,1876	0,2312	0,1505	0,2855

Table 7.1: Example 1. RidgeFS scores

Results for Elementary Effects are presented in the Table 7.2. Elementary Effects gives satisfactory close to True results on 30 points sample already and very close results on 100 points.

Sample size	x_1	x_2	x_3	x_4	x_5
True	0,0181	0,0727	0,1636	0,2909	0,4545
30	0.0152	0,0754	0,1782	0,2811	0,4213
100	0,0193	0,0721	0,1691	0,2952	0,4415

Table 7.2: Example 1. Elementary Effects scores

Results for Mutual Information (Kraskov estimate) are presented in the Table 7.3. Kraskov estimate gives satisfactory results on 30 points and quite close to True on 500 points.

Sample size	x_1	x_2	x_3	x_4	x_5
True	0,0181	0,0727	0,1636	0,2909	0,4545
30	0,1058	0,1051	0,0963	0,26478	0,4279
100	0,0867	0,0785	0,1220	0,2562	0,4563
500	0,0366	0,0774	0,1375	0,2772	0,4711

Table 7.3: Example 1. Mutual Information (Kraskov estimate) scores

Results for Mutual Information (histogram estimate) are presented in the Table 7.4. As expected, Histogram based estimation of Mutual Information is inferior to Kraskov estimate on small samples, but still manages to do close to True estimation.

Sample size	x_1	x_2	x_3	x_4	x_5
True	0,0181	0,0727	0,1636	0,2909	0,4545
30	0,0622	0	0,0656	0,2988	0,5733
100	0	0,0856	0,1486	0,2513	0,5142
500	0,0059	0,0315	0,1287	0,2914	0,5422
750	0,0084	0,0585	0,1501	0,2958	0,4868
1000	0	0,0513	0,1725	0,3020	0,4740
2000	0,0039	0,0609	0,1791	0,2967	0,4591

Table 7.4: Example 1. Mutual Information (histogram estimate) scores

Results for FAST are presented in the Table 7.5. FAST needs at least $65 \times 6 = 390$ points to work on this sample. It gives satisfactory results on 500 points, and good on 1000 points.

Sample size	x_1	x_2	x_3	x_4	x_5
True	0,0181	0,0727	0,1636	0,2909	0,4545
500	0,0339	0,0963	0,2589	0,2744	0,3362
750	0,0442	0,0824	0,1638	0,2370	0,4723
1000	0,0273	0,0808	0,1681	0,2697	0,4538

Table 7.5: Example 1. FAST scores

7.1.2 Example 2: usage of confidence intervals to determine redundant variables

In this example we will demonstrate how knowing confidence intervals can tell us whether the function depends on the feature or not.

For simplicity let us consider the function:

$$f(x_1, x_2, x_3, x_4, x_5) = x_1^2 + x_1 x_2^2 + 0.01 x_3^2,\ x_i \in [-1,\ 1],\ i = 1, 2, 3. \tag{7.2}$$

Here the function depends very weakly on x_3.

We generate 200 points random sample for this function and apply **GTIVE** (in this case Mutual Information kraskov algorithm will be used). Results for scores and the standard deviation of scores (the square root of estimated variance of scores) are provided in the table 7.6.

Sample size	x_1	x_2	x_3
Scores	0,7494	0,2506	0,0
stdScores	0,1019	0,0745	0,0516

Table 7.6: Example 2. GT IVE scores and the standard deviation of scores

Using confidence intervals one may additionally check whether we can trust obtained score values. Score value for third feature is zero so it's contribution was not detected on this sample size. To check if scores for the first and the second features are significantly larger than zero one should check if for i-th feature zero belongs to the interval $(Score_i - 3 \cdot stdScore_i,\ Score_i + 3 \cdot stdScore_i)$. For the first feature:

$$Score_1 - 3 \cdot stdScore_1 = 0.4437 > 0$$

For the second feature:

$$Score_2 - 3 \cdot stdScore_2 = 0.0272 > 0$$

which means that both scores with very high probability are significantly larger than zero. And obviously this value is negative for the third feature.

7.1.3 Example 3: difference between 'main' and 'total' scores in FAST

In this example we will consider FAST performance for the function:

$$f(x_1, x_2, x_3) = x_1^2 + 2x_1x_2 + x_3^2,\ x_i \in [-1,\ 1],\ i = 1, 2, 3, \tag{7.3}$$

that on the one hand is still simple enough to form some expectations of what true scores should be, but on the other hand it already has some feature interactions.

So in this example one may expect to see x_1 having the largest score, x_2 be on the second place and x_3 be the least important feature.

We will use this example to demonstrate the difference between main and total FAST scores. Main scores take into account only isolated variable contribution to the variance of output, meaning that main scores would ignore influence of the $x_1 \cdot x_2$ term. Total scores on the other side should account all feature interactions. In the manual dependency analysis comparison of these two indices allows for some investigation of the dependency nature. We've estimated these scores using 500 and 1000 points samples to show the difference in the results.

Total scores are presented in the Table 7.7.

Sample size	x_1	x_2	x_3
500	0,4449	0,3985	0,1503
1000	0,4965	0,4125	0,0869

Table 7.7: Example 2. FAST (total) scores

Main scores are presented in the Table 7.8.

Sample size	x_1	x_2	x_3
500	0,3353	0,0019	0,6604
1000	0,5016	0,0033	0,4910

Table 7.8: Example 2. FAST (main) scores

Let S_{T1}, S_{T2}, S_{T3} be total indexes of variables and S_{M1}, S_{M2}, S_{M3} be main indices.

One may see that $S_{M2} \approx 0$, $S_{T2} \gg S_{M2}$, it gives one a hint that x_2 feature appears only in interaction with some other. Also one may remember that $S_{Ti} = S_{Mi} + interaction\,terms$, i.e. say $S_{T1} = S_{M1} + S_{12} + S_{13}$, where for the example S_{12} - is a term accounting for x_1 and x_2 interaction. Notice also that $S_{M1} \approx S_{M3}$, $S_{M2} \approx 0$ and $S_{T1} \approx S_{T2} + S_{T3} \Rightarrow S_{12} + S_{13} \approx S_{12} + S_{23} + S_{13} + S_{23} \Rightarrow S_{23} \approx 0$. As a result we can make an educated guess that our function has the following form $f(x_1, x_2, x_3) = f_1(x_1) + f_2(x_3) + f_3(x_1, x_2) + f_4(x_1, x_3)$.

7.2 Real world data examples

In this section we will show application of **GTIVE** to some real world data problems.

7.2.1 T-AXI problem

- **Problem description:**
  In this problem we consider The T-C_DES (Turbomachinery Compressor DESign) code (meanline axial flow compressor design tool), which is the first step of T-AXI (an axisymmetric method for a complete turbomachinery geometry design [18]).

  Program tcdes.e3c-des.exe is used for calculation of outputs $f(X)$ for new generated inputs X. Program can be downloaded from the link:
  http://gtsl.ase.uc.edu/T-AXI/.

  Program uses a 163 dimensional feature vector describing geometry and the working condition as an input.

  The task is to determine subset of the most important features for the Compressor Pressure Ratio (With IGV) output. The dependency is considered only for $X \in V(X^0) = \{X : x^i \in [(1-\alpha)x_i^0, (1+\alpha)x_i^0]\}$, $i = 1, \ldots, 163$ where $\alpha = 0.1$, $X^0 = (x_1^0, \ldots, x_{163}^0)$ is given in Tables 7.9 – 7.11.

	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	0,297	0,2946

Table 7.9: Stage data for 10 stage design (stage.e3c-des)

Mass Flow Rate [kg/s]	54,4
Rotor Angular Velocity [rpm]	12299,5
Inlet Total Pressure [Pa]	101325
Inlet Total Temperature [K]	288,15
Mach 3 - Last Stage	0,272
Clearance Ratio	0,0015

Table 7.10: Initial data for 10 stage design (init.e3c-des)

- **Solution workflow:**
  We perform the following steps to make the analysis:

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

Table 7.11: IGV data for 10 stage design (igv.e3c-des)

1. We generate data sample of 10^4 points. One may use available code as a black box as well, but we didn't do it because code fails to compute outputs in many points.
2. On a given sample, feature scores are estimated using **GTIVE** with default settings. By default, in this case histogram based estimate is used, see 4.3.
3. Estimated feature scores are plotted on the picture 7.1. Looking at the picture one may see that there are clearly 12 most influential features. So it's natural to perform preliminary optimization of compressor varying only this 12 features instead of all 163.
4. To validate the results of the **GTIVE** we estimated the Index of Variability (2.1) of different important feature subsets Z adding features one by one starting from the ones with higher **GTIVE** scores and from the lower scores. Results are presented on the Figure 7.2.

- **Results:**
  In the Tables 7.12 – 7.13 the most important feature is filled with dark green, next 11 important ones are filled with light green color.

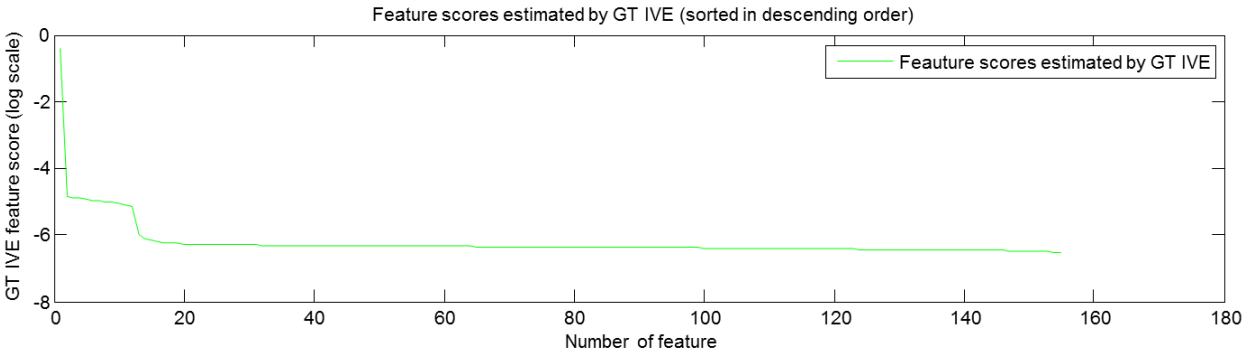

Figure 7.1: T-AXI. Feature scores estimated by **GTIVE**. **Note:** This image was obtained using an older MACROS version. Actual results in the current version may differ.

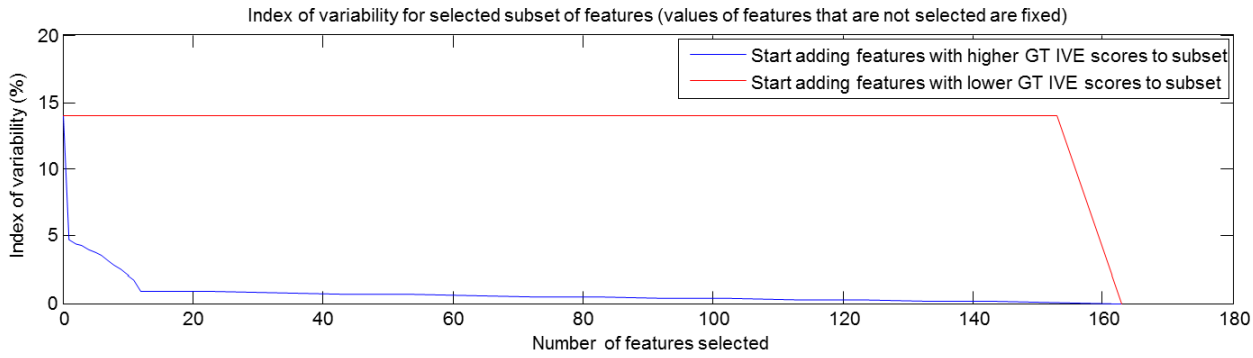

Figure 7.2: T-AXI. Index of Variance. **Note:** This image was obtained using an older MACROS version. Actual results in the current version may differ.

	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	0,297	0,2946

Table 7.12: T-AXI. Features that influence Compressor Pressure Ratio the most (a)

Mass Flow Rate [kg/s]	54,4
Rotor Angular Velocity [rpm]	12299,5
Inlet Total Pressure [Pa]	101325
Inlet Total Temperature [K]	288,15
Mach 3 - Last Stage	0,272
Clearance Ratio	0,0015

Table 7.13: T-AXI. Features that influence Compressor Pressure Ratio the most (b)

7.2.2 Stringer (Super-Stiffener) Stress Analysis problem

- **Problem description**
  Special tool for Stress Analysis build upon a physical model computes Reserve Factors (RFs) constraints for a side panel (of an airplane) defined by its geometry ($\mathbf{G}_j,\ j = 1,\dots,5$) and applied forces ($\mathbf{F}_i,\ i = 1,2,3$) [1, 4].

  Our task here is to check whether all inputs equally influence the output RFs. In particular, the case of stringer RF (RF STR) is considered.

- **Solution workflow**

  1. We have a code that can compute RFs for the given point, so we may use black box technique.
  2. We estimate feature scores with default settings and various budget and `Seeds` (see 4.5 for details) to check what size of budget for **GTIVE** gives reliable estimates and how stable the estimates are (Elementary Effects technique was taken by default, see Section 4.5).
  3. Results for different budget sizes are presented in the Table 7.14. For each budget size 10 runs with different seeds were made to estimate standard deviation of results. One can see that mean estimates are already quite reliable on 50 points and variance of the results reduces as sample size increase. Also one may notice that RF STR is independent from feature $\mathbf{F}_1$.
  4. To validate the results of the **GTIVE** we used approximation error ratio measure (2.2) of RF STR. Results of this experiment are presented in Table 7.15 and show that error of approximation are in agreement with the values of feature scores, estimated by **GTIVE**.

GT IVE	$\mathbf{F}_1$		$\mathbf{F}_2$		$\mathbf{F}_3$		$\mathbf{G}_1$	
	mean	std	mean	std	mean	std	mean	std
50 pnts	0	0	0,0477	0,0207	0,2713	0,0704	0,0732	0,0377
300 pnts	0	0	0,0602	0,0107	0,2889	0,0216	0,0703	0,008
1000 pnts	0	0	0,0624	0,0041	0,286	0,0129	0,0715	0,0049

GT IVE	$\mathbf{G}_2$		$\mathbf{G}_3$		$\mathbf{G}_4$		$\mathbf{G}_5$	
	mean	std	mean	std	mean	std	mean	std
50 pnts	0,062	0,0197	0,1056	0,0354	0,3323	0,0836	0,1079	0,0233
300 pnts	0,0673	0,0074	0,1001	0,0123	0,3135	0,0296	0,0998	0,0157
1000 pnts	0,0674	0,0037	0,1043	0,0062	0,3089	0,0136	0,0996	0,0058

Table 7.14: Stringer stress analysis. Feature scores estimated by **GTIVE**

- **Results**

  – **GTIVE** showed that RF_STR value is independent of the feature $\mathbf{F}_1$
  – **GTIVE** using as few points as possible was able to estimate reliably relative importance of each feature

	$\mathbf{F}_1$	$\mathbf{F}_2$	$\mathbf{F}_3$	$\mathbf{G}_1$
GT IVE Score	0	0,0624	0,286	0,0715
Approx error if fixing feature / full model error	0,98	22,83	126,04	29,85

	$\mathbf{G}_2$	$\mathbf{G}_3$	$\mathbf{G}_4$	$\mathbf{G}_5$
GT IVE Score	0,0674	0,1043	0,3089	0,0996
Approx error if fixing feature / full model error	29,83	45,72	142,46	43,63

Table 7.15: Stringer stress analysis. Approximation error ratio

7.2.3 Fuel System Analysis problem

- **Problem description**
  The objective of the Research into Fuel Systems project is to deliver application that can predict pressures and mass flows for gravity feed aircraft fuel systems [6]. The desktop application comprises a two phase flow (air and fuel) analysis engine that is derived from experimental observations.

  One of the task the MACROS models are used for in this project is to approximate pressure loss coefficient and volume flow quality of the fuel flow on the diaphragm section of the pipe using experimental data.

  Experimental data is a 244 points sample with 6 features describing fuel flow (flow velocity ($\mathbf{V}$), pressure after the diaphragm ($\mathbf{P}$), temperature ($\mathbf{T}$), densities of fuel (ρ_{fuel}) and air (ρ_{air}), ratio of diaphragm diameters ($\mathbf{r}_i$)) and two outputs pressure loss coefficient ($\mathbf{C}_p$) and volume flow quality ($\mathbf{Q}$).

  We will use **GTIVE** to determine which features should be measured with the most accuracy. This is very important for experimental design: if the feature is unimportant then we shouldn't do additional expensive experiments in order to explore the dependence of the outputs ($\mathbf{C}_p$ and $\mathbf{Q}$) on this feature, and we can measure this feature with less precision in the experiments.

- **Solution workflow**

  1. We have a sample of experimental data, so sample based technique is going to be used.
  2. **GTIVE** scores were computed with the default settings (Mutual information Kraskov estimate was used in this case, see Section 4.2).
  3. To validate results we've calculated the Approximation error ratio measure 2.2 for both outputs. Table shows that **GTIVE** scores are in good agreement with feature scores, see Table 7.16.

$\mathbf{Q}$	$\mathbf{V}$	$\mathbf{P}$	$\mathbf{T}$	ρ_{air}	ρ_{fuel}	$\mathbf{r}_i$
GT IVE Score	0.7204	0.925	0.2697	0.0688	0.1731	0.6628
Approximation error if fixing feature / full model error	1.37	3.09	1,07	1.04	1.07	1,26
$\mathbf{C}_p$	$\mathbf{V}$	$\mathbf{P}$	$\mathbf{T}$	ρ_{air}	ρ_{fuel}	$\mathbf{r}_i$
GT IVE Score	0.1888	0.1166	0.0843	0.0773	0.0944	0.4383
Approximation error if fixing feature / full model error	1.04	1.19	1.12	1.12	1.03	4.45

Table 7.16: Fuel System Analysis. Features scores and Approximation error ratio

- **Results**

  – It can be seen that values of scores are in good correspondence with errors of approximation.

Bibliography

[1] E. Burnaev. Construction of the metamodels in support of stiffened panel optimization. In *Proceedings of the conference MMR 2009 Mathematical Methods in Reliability*, 2009.

[2] Datadvance. *Generic Tool for Approximation: User manual.*

[3] DATADVANCE, llc. *MACROS Generic Tool for Important Variable Extraction*, 2011.

[4] S. Grihon. Application of response surface methodology to stiffened panel optimization. In *Proceedings of 47th conference on AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2006.

[5] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction.* Springer, 2008.

[6] E. Kitanin. Air Evolution Research in Fuel Systems 4. Technical report, IRIAS, 2010.

[7] I. Kononenko. An adaptation of relief for attribute estimation in regression. 1997.

[8] A. Kraskov. Estimating mutual information. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 69:40–79, 2004.

[9] H. Liu. Relative entropy-based probabilistic sensitivity analysis methods for design under uncertainty , aiaa-2004-4589. *10-th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2004.

[10] F. Maes. Multimodality image registration by maximization of mutual information. *IEEE transactions on Medical Imaging (16)*, pages 187–198, 1998.

[11] P. Qiu. Fast calculation of pairwise mutual information for gene regulatory network reconstruction. *Computer Methods and Programs in Biomedicine*, 94(2):177–180, 2009.

[12] A. Saltelli. A quantitative model-independent method for global sensitivity analysis of model output. *Technometrics*, 41:39–56, 1999.

[13] A. Saltelli. *Global Sensitivity Analysis The Primer.* Wiley, 2008.

[14] B. Schulkopf. Measuring statistical dependence with hilbert-schmidt norms. *In Algorithmic Learning Theory*, 3734:63–74, 2005.

[15] V. Schwieger. Variance-based sensitivity analysis for model evaluation in engineering surveys. *Data Processing*, pages 1–10, 2004.

[16] H. Sundar. Robust computation of mutual information using spatially adaptive meshes. *Proceeding MICCAI'07. Proceedings of the 10-th international conference on Medical image computing and computer-assisted intervention*, Part I:950–958, 2007.

[17] G. Tourassia. Application of the mutual information criterion for feature selection in computer-aided diagnosis. *Med. Phys. 28*, 2001.

[18] M. Turner. A turbomachinery design tool for teaching design concepts for axial-flow fans compressors and turbines. *Proceedings of GT2006*, 2006.

[19] S. Vallaghe. A global sensitivity analysis of three- and four-layer eeg conductivity models. *Biomedical Engineering, IEEE Transactions (56)*, pages 988–995, 2009.

Index

Index: Options