SVR-HDMR software user's guide

Introduction:

SVR-HDMR is a machine learning algorithm that combines the support vector machine (SVM) -regression and high dimensional model representation(HDMR). This software package is designed to facilitate the implementation of this core algorithm with a graphic user interface, which consists of 1) SVR-HDMR Solver and 2) SVR-HDMR Viewer. The former is responsible for input data preparation, model parameter setup, model construction and sensitivity analysis. The latter is focused on visualization, and model exploration, prediction. They used to be separate modules, but were later integrated into one package. All the functionalities could be accessed by running the solver, i.e., "svr_HDMR_main.m".

System requirement:
This software is developed using matlab (2014b) in a win64 PC. Installation of matlab(2014b or later version) and statistics and machine learning toolbox is required. It also works on a MAC, although there could be some minor display issues for the interface appearance.

What is covered in this user's guide: It will guide you through all the steps of using this software of various functionalities as an operational manual, using examples to demonstrate and show the usage of each built-in tool.

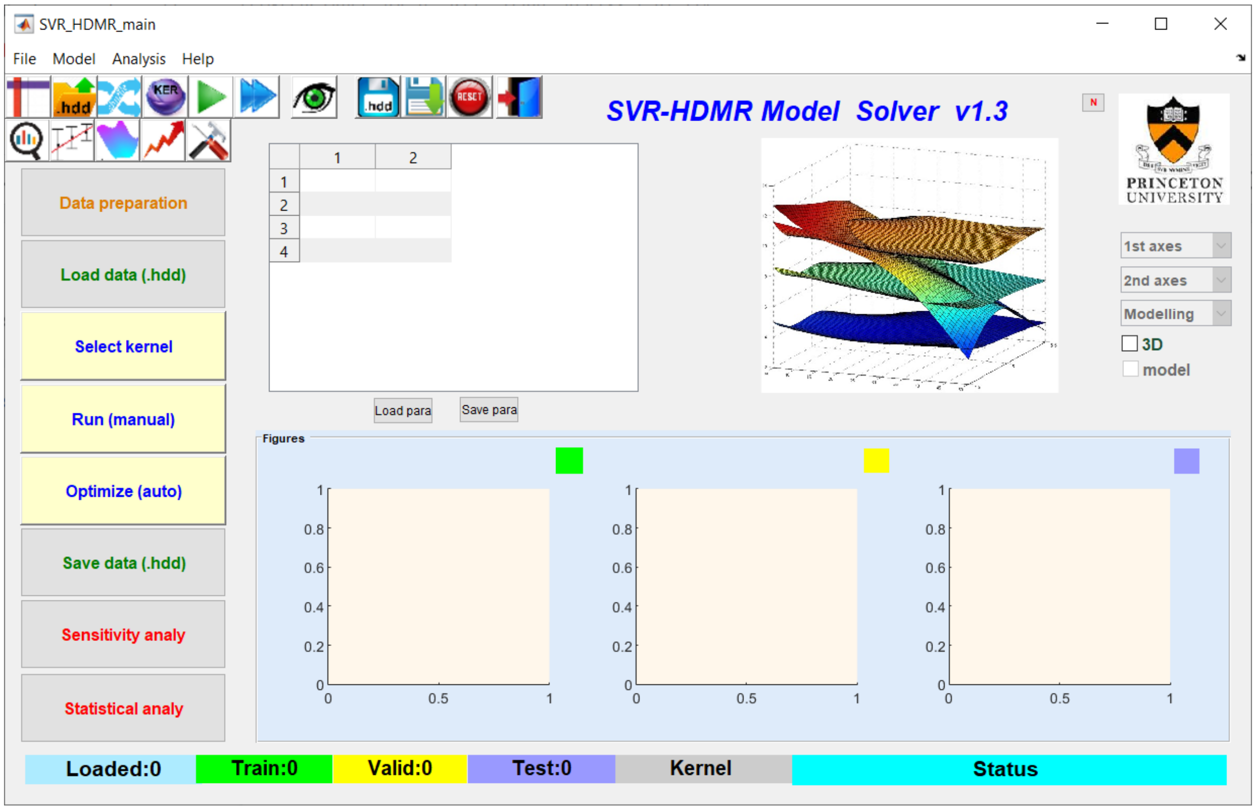
What is not covered: The coding level details are not covered. The mathematics, concepts behind SVR-HDMR and some terminologies associated with sensitivity analysis are not explained or covered. Please refer to the paper below and some earlier publications from the same group for details.

Li, Genyuan et al. "High dimensional model representation constructed by support vector regression. I. Independent variables with known probability distributions." *Journal of Mathematical Chemistry* 55 (2016): 278-303.

Part I. SVR-HDMR solver:

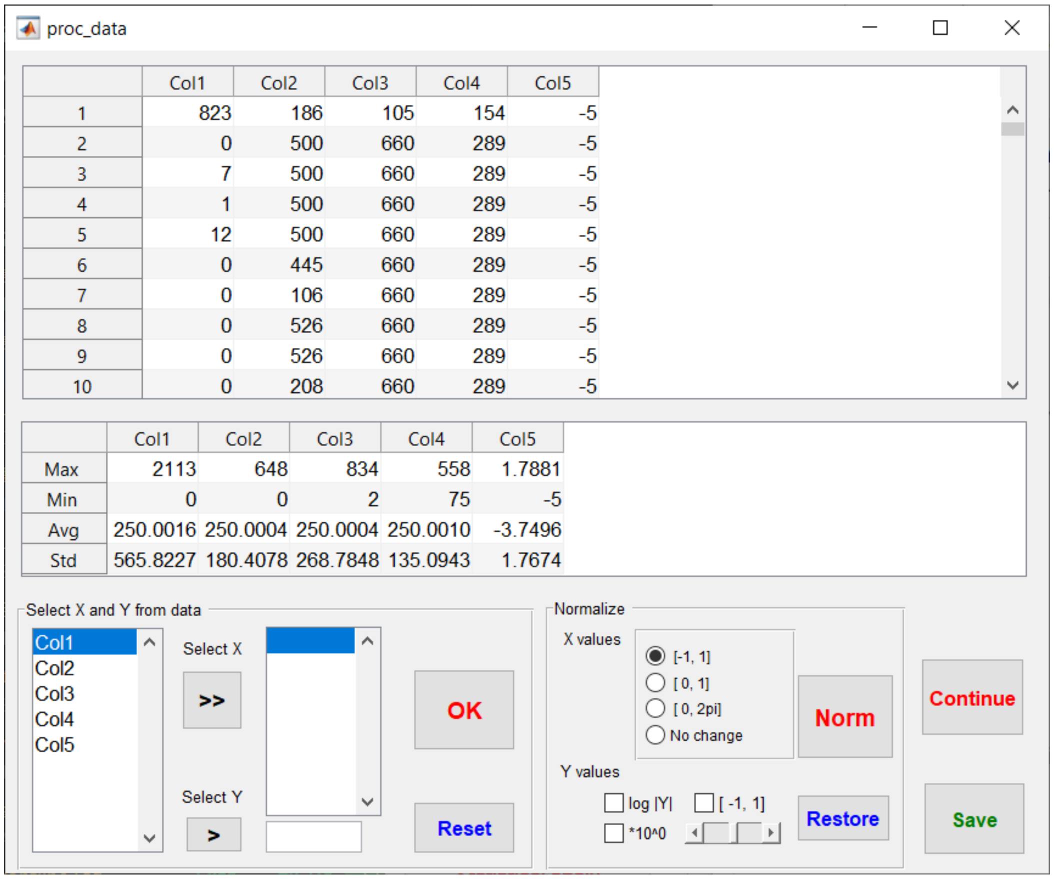
Start up:

"svr_HDMR_main.m" is the main file to run to start the program. Please run this .m file inside matlab (don't run or double click the "svr_HDMR_main.fig" file). Below shows the user interface for the initial run.



Data preparation:

The first step is to prepare the input data. This can be done by either clicking the "Data preparation" button or the button (a slightly different version) or from the "File" menu. An .txt file in tabular format will be loaded through an openFileDialog (there are several example .txt files in the folder). Column names will be included if they exist, otherwise, Col1, Col2,... will be added as the default column names. The interface will lead you through to select the dependent variables (X) and a single independent variable (Y). And also, various options for normalization of X values and Y values conversions are provided. Usually X should be normalized to [-1,1]. Whether Y should be converted depends on its distribution (google search "boxcox transformation" for ideas). A reset and a restore button makes it easy to go back and redo. After these two steps, click "Continue" to move on to the next step, or "Save" as (.hdd) file (Save is strongly recommended so that you only need to do this once per dataset). Note that, only ".hdd" files can be recognized by this software to proceed to modeling. It's in simple text format and 'hdd' does not have any specific meanings, just an arbitrary name to differentiate from ".txt" files.



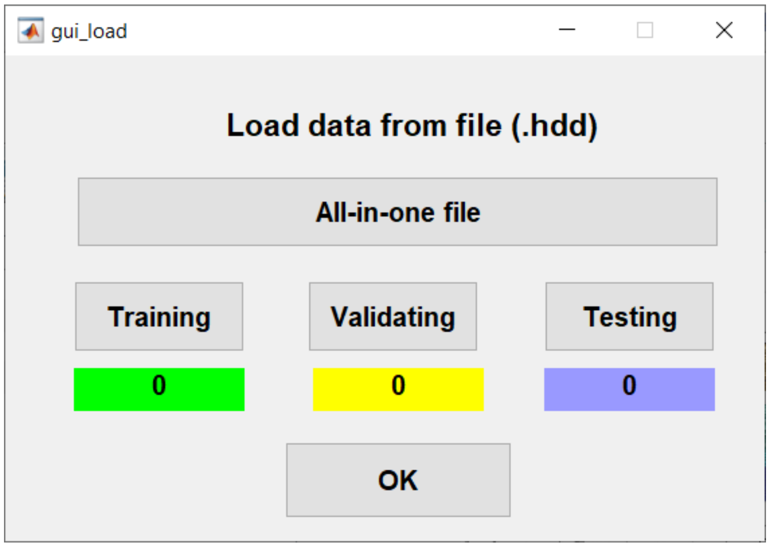
	Col1	Col2	Col3	Col4	Col5
1	823	186	105	154	-5
2	0	500	660	289	-5
3	7	500	660	289	-5
4	1	500	660	289	-5
5	12	500	660	289	-5
6	0	445	660	289	-5
7	0	106	660	289	-5
8	0	526	660	289	-5
9	0	526	660	289	-5
10	0	208	660	289	-5

	Col1	Col2	Col3	Col4	Col5
Max	2113	648	834	558	1.7881
Min	0	0	2	75	-5
Avg	250.0016	250.0004	250.0004	250.0010	-3.7496
Std	565.8227	180.4078	268.7848	135.0943	1.7674

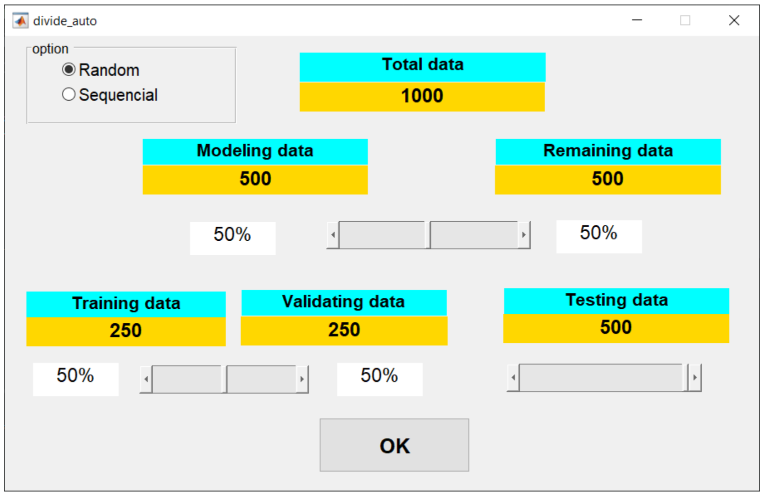
Normalize X values: [● [-1, 1]] ○ [0, 1] ○ [0, 2pi] ○ No change

Data loading, shuffle and visualization

By clicking load data or .hdd, a small window will appear like the one below. Either a) Click "All-in-one file" to load the entire setting before dividing into training, validating and testing sets. (can also leave some data as remaining, not in use) Or b) if the data preparation was done by each set separately, click the Training, validating and testing buttons individually to load them.



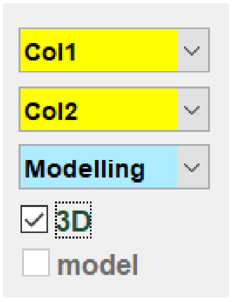
For a) load all-in-one file, after .hdd is loaded, you will be directed to another window for data division. Use slide bars to change the number of data in each set before clicking OK. The default option is random selection. If sequential is selected, the dataset will be divided in sequence as training - validating - testing - remaining. Note that, if you choose “cross-validating” in later steps (see hyper parameter optimization for details), the second division between training and validating does not matter because the modeling data which includes training and validating will be shuffled and redefined on the fly anyway.



After done with data dividing and click OK, it will go back to the previous window, which will update and show the number of data selected for each set. Click OK will return to the main window and show these updates:

1) Histogram showing the statistics of Y values for each data set
2) # of data loaded in the lower status bar
3) A default kernel parameter table
4) A popup figure showing a 2D landscape of Col1 vs. Col2 (feel free to close it)

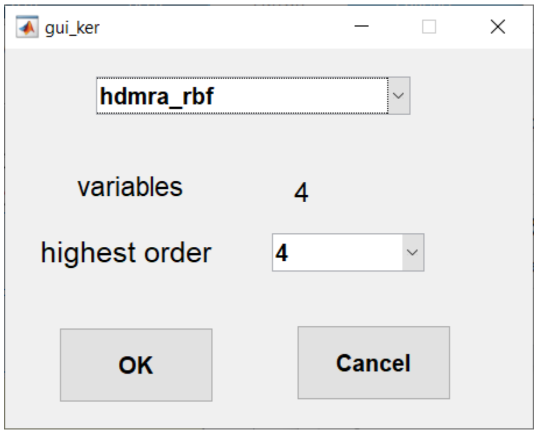
Changing the dropboxes as shown below from the main window allows you to visualize the data in 2D or 3D of different subspaces. The first dropbox defines X axis, the second defines Y axis, if 3D is checked, the output Y values will be the Z axis. The 3rd drop box allows to choose which subset of data to visualize (e.g, All, modeling, training, validating, testing) . The second checkbox “model” will be enabled only after modeling is performed, then you will have the choice of visualization either the experimental data or the model predicted data. This integrated functionality is very useful and convenient for data visualization. Alternatively, you can click for data visualization in a separate window. More visualization tools can be found in the second module of “SVR-HDMR Viewer”


Col1
Col2
Modelling
[✓ 3D]
model

This button is used for data shuffling, which will randomly shuffle the data without changing the number of data points in each set. Watching for the changes of data distribution in each histogram. This is useful especially when the data size is small and you want to include or remove certain data points from a particular set on purpose. It is usually desirable to have similar data distributions for each dataset.

Kernel selection:

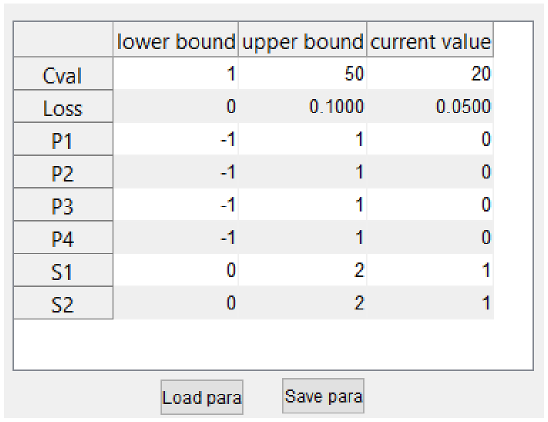
By clicking “select Kernel” button or , you will be directed to a window that allows you to select a kernel as well as kernel parameters preset. The default kernel is “hdmra_rbf”, with the highest order being equal to the number of variables. You may want to reduce it to 2 or 3 for most applications, Note that, increasing the order will also increase the number of kernel parameters, which may not necessarily lead to the improvement of model predictivity.



variables 4

highest order 4

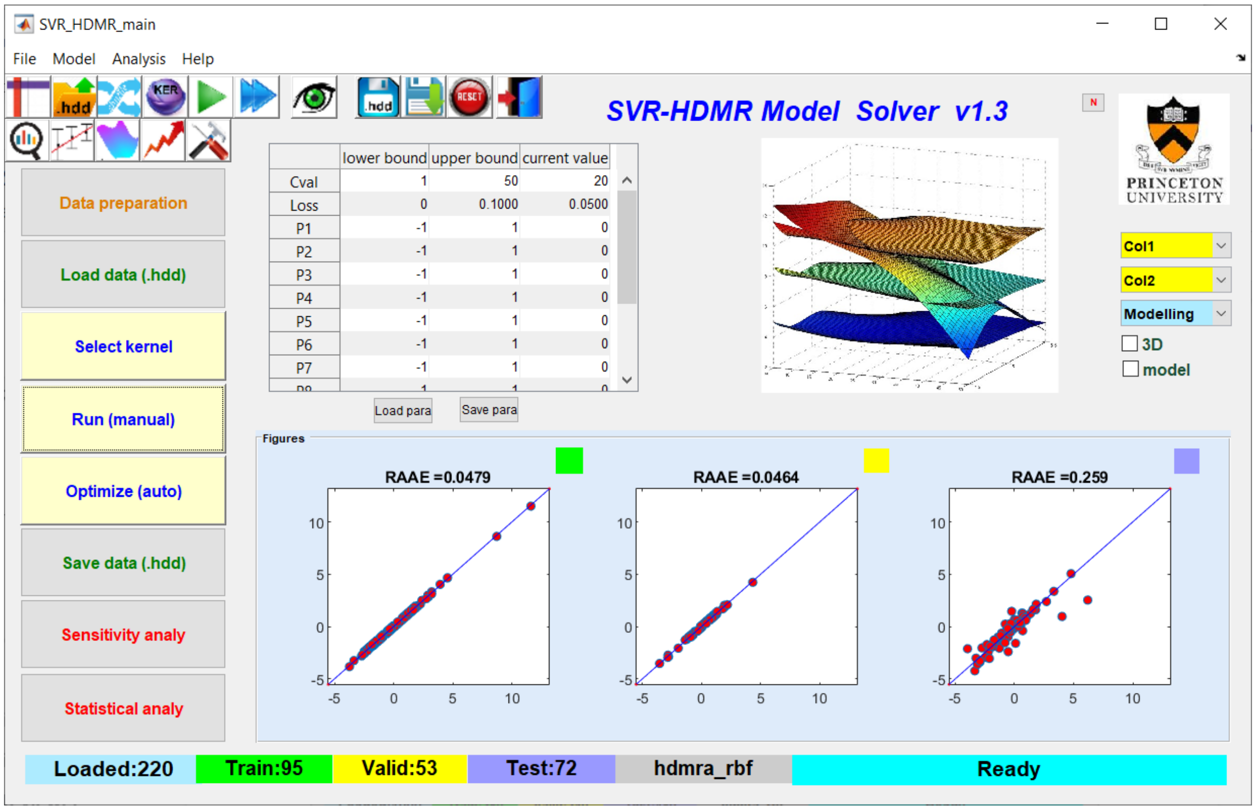
The kernel parameter table in the main window will update accordingly, inside which, you can manually set each value in “current value”, and also set the lower/upper bound for each parameter for hyper parameter optimization. The two buttons below the table are for load and save parameters. Note that, for “load para”, it must be ensured to exactly match the current kernel setting, which is taken care of by the user.



	lower bound	upper bound	current value
Cval	1	50	20
Loss	0	0.1000	0.0500
P1	-1	1	0
P2	-1	1	0
P3	-1	1	0
P4	-1	1	0
S1	0	2	1
S2	0	2	1

Model training:

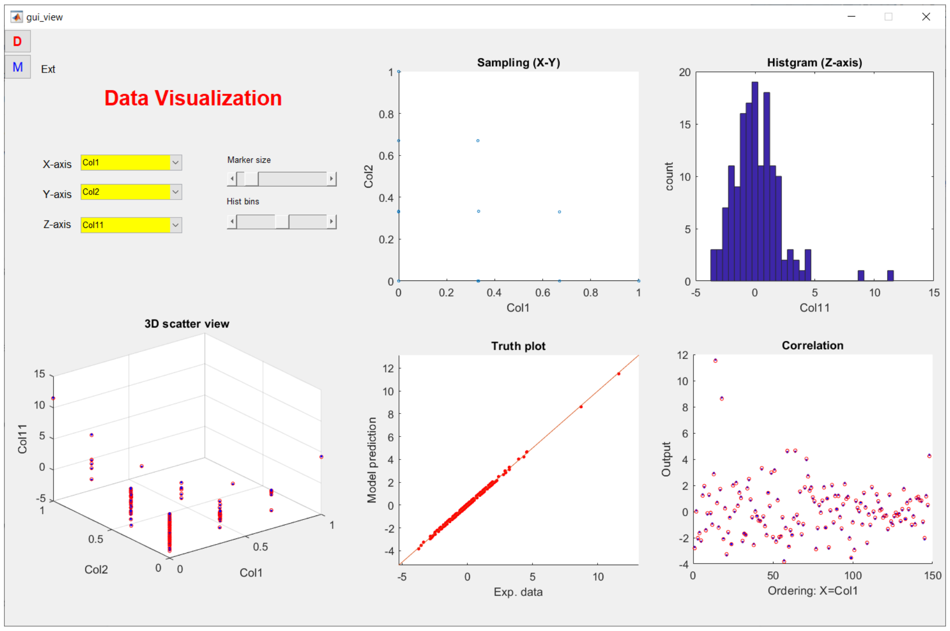
Click Run(manual) or ▶ to perform a single run of SVR using the kernel settings specified in the current value. The right bottom status bar will turn from red (SVR Kernel) to green (training) to yellow(validating) to purple (testing) to cyan (Ready) that completes one machining leaning cycle. Kernel parameters are not optimized in this mode. You are allowed to change parameters manually and rerun as many times as you want. Therefore it’s called the manual mode.
The three figures shown as histograms before will now display the truth plots for each dataset as shown as an example below. By Clicking the color square button on the right top side of each figure, a separate window containing the plot will popup, allowing you to save the figure. You need to close the popup window before you can click the color square again for another one to popup.



	lower bound	upper bound	current value
Cval	1	50	20
Loss	0	0.1000	0.0500
P1	-1	1	0
P2	-1	1	0
P3	-1	1	0
P4	-1	1	0
P5	-1	1	0
P6	-1	1	0
P7	-1	1	0

Loaded:220 Train:95 Valid:53 Test:72 hdmra_rbf Ready

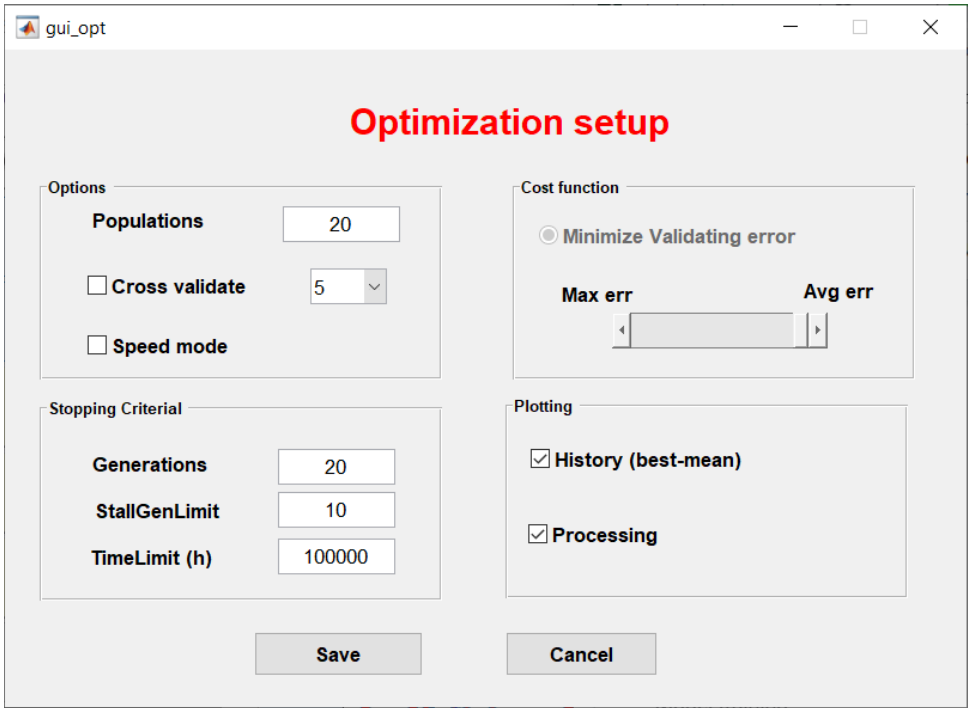
Also, please note that the “model” check box is now enabled and you can visualize the model predictions that overlay the measured ones by clicking it and selecting the desired columns to view. Alternatively, click for comprehensive data visualization. This data visualization gui is standalone, you can run “gui_view.m” in matlab without the need of running svr_HDMR and use the left side buttons ‘D’ to load .hdd data, and ‘M’ to load model data saved by (see Data I/O section)



Hyper parameter optimization:

By clicking Optimize(auto) or , you are ready to setup for a kernel parameter optimization, which might take much longer depending on the data size, kernel choices and optimization setup. Below is the optimization setup table defining

1) the number of populations in each generation (a global optimization GA is used here),
2) if cross validation is enabled, and # of fold
3) For speed mode, it does not make predictions for the testing set until the optimization is completed, so you won't be able to see the real-time update of the true performance (by examining the testing data truth plot), but saves time.
4) Stopping criterion defining the maximum generations, stallgenLimit (best not changing over ? generations), and maximum time in hours
5) Cost function definition, by default it's minimizing the mean err of the validation data set, this can be changed to minimizing the max err as another extreme, and use the sliding bar to choose a balance.
6) The progressing plot showing the GA optimization outcome


gui_opt

Optimization setup

Options

Populations: 20

☐ Cross validate: 5

☐ Speed mode

Cost function

● Minimize Validating error

Max err — Avg err

Stopping Criterial

Generations: 20

StallGenLimit: 10

TimeLimit (h): 100000

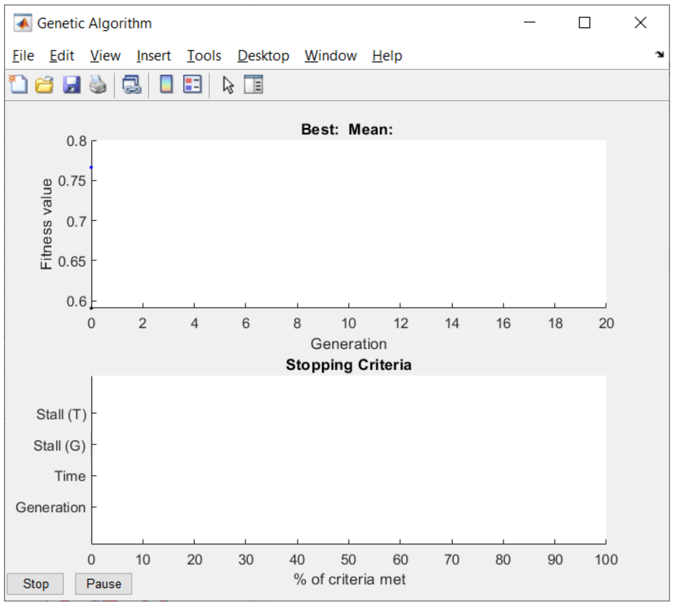
Plotting

☑ History (best-mean)

☑ Processing

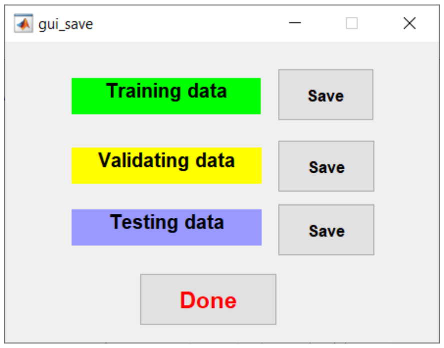
After clicking save and confirming (a dialog box), you are set to go for optimization. The progress window will appear after running one generation. Wait until it completes or click pause or stop to interrupt and finish the run (also need to wait one more generation before it fully stops).

After the optimization is completed, the main window will update the kernel parameters of its current values which are the optimized ones and also the final truth plots.



Data I/O:

Clicking the “Save data” button allows you to save the current division of the dataset into each set in “hdd”. So that you can always come back and reproduce the same result by loading them using method b) in the “Data loading” section. Note that, in order to reproduce the results, kernel parameters should also be saved to avoid rerun the optimization. (Rerun of the optimizations won’t end up with the same results even for the same data set)

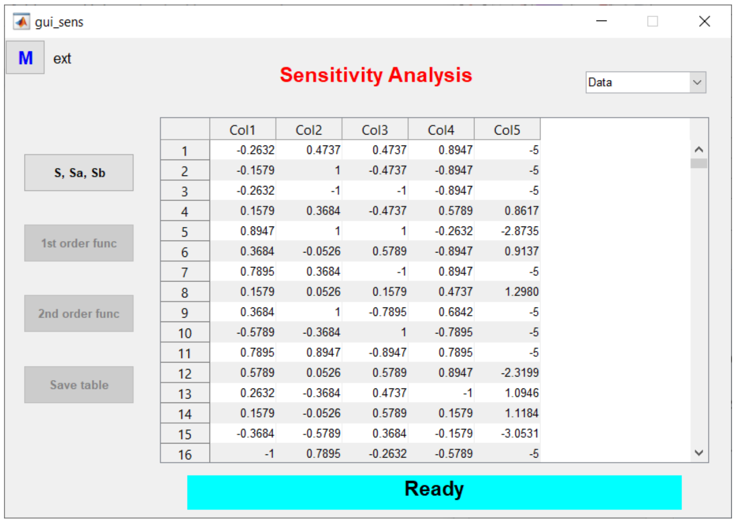


Use button to save the model, which includes info of input data, kernel type, kernel parameters, predicted values, all in one .mat file. This .mat can be loaded by several standalone gui, such as “data visualization” and “sensitivity analysis”, wherever there’s a blue ‘M’ button on the top-left of the gui.

Sensitivity analysis:

Clicking the Sensitivity analysis button or will load the sensitivity analysis gui. You can also run “gui_sens.m” as a standalone and load in the previously saved model using the blue ‘M’ button on the top-left corner (‘ext’ will turn into the name of the model).

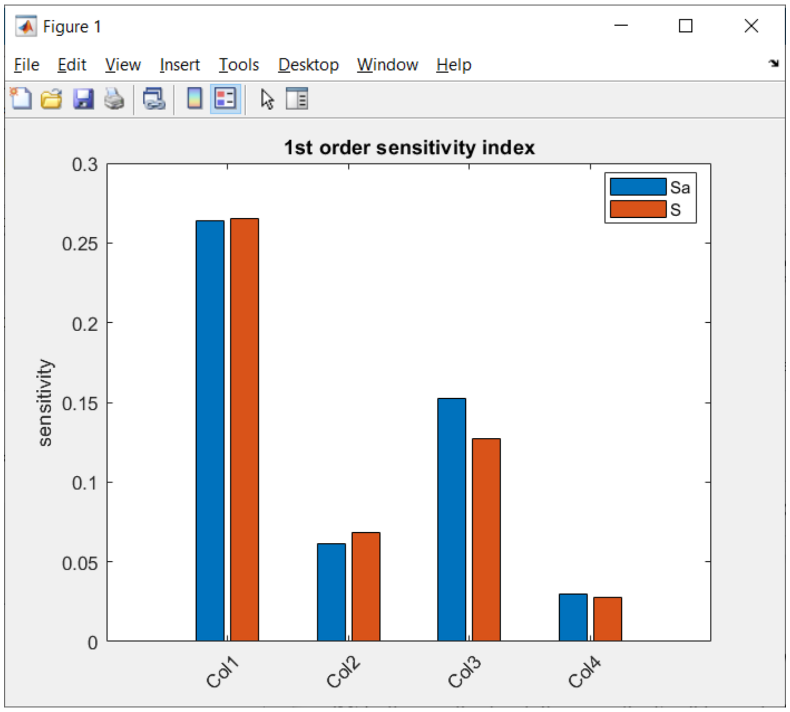
1) Initially, it only shows the input data of multi-X and Y
2) Click the first button “S,Sa,Sb”, it will calculate the sensitivity indices S, Sa, and Sb and summarize them in three different tables. Use the dropdown box on the top right side to switch between 1st order, 2nd order and sum. In the meantime, a new window will popup showing the 1st order sensitivity index of Sa and S for each variable. (Note that, S=Sa+Sb, so Sb is the difference between S and Sa, which is not displayed)
3) Clicking the second button “1st order func” plots the 1st order component functions. The plot data can be saved if you want to plot it in your own way.
4) Clicking the third button “2nd order func” plots the 2nd order component functions. The plot data can be saved if you want to plot it in your own way.
5) The tables can be saved in .txt by clicking the last button

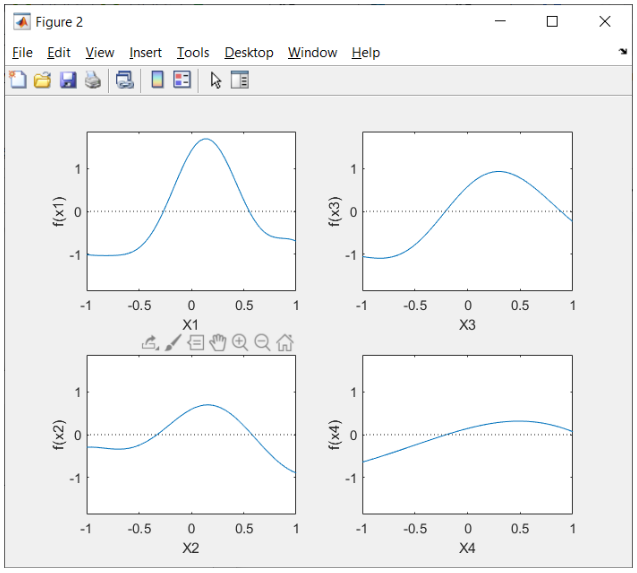


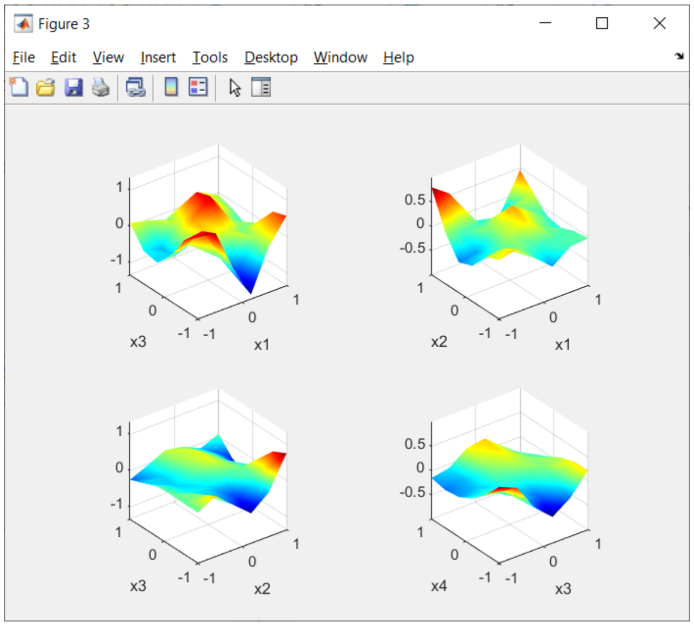
Sensitivity Analysis

	Col1	Col2	Col3	Col4	Col5
1	-0.2632	0.4737	0.4737	0.8947	-5
2	-0.1579	1	-0.4737	-0.8947	-5
3	-0.2632	-1	-1	-0.8947	-5
4	0.1579	0.3684	-0.4737	0.5789	0.8617
5	0.8947	1	1	-0.2632	-2.8735
6	0.3684	-0.0526	0.5789	-0.8947	0.9137
7	0.7895	0.3684	-1	0.8947	-5
8	0.1579	0.0526	0.1579	0.4737	1.2980
9	0.3684	1	-0.7895	0.6842	-5
10	-0.5789	-0.3684	1	-0.7895	-5
11	0.7895	0.8947	-0.8947	0.7895	-5
12	0.5789	0.0526	0.5789	0.8947	-2.3199
13	0.2632	-0.3684	0.4737	-1	1.0946
14	0.1579	-0.0526	0.5789	0.1579	1.1184
15	-0.3684	-0.5789	0.3684	-0.1579	-3.0531
16	-1	0.7895	-0.2632	-0.5789	-5

Ready

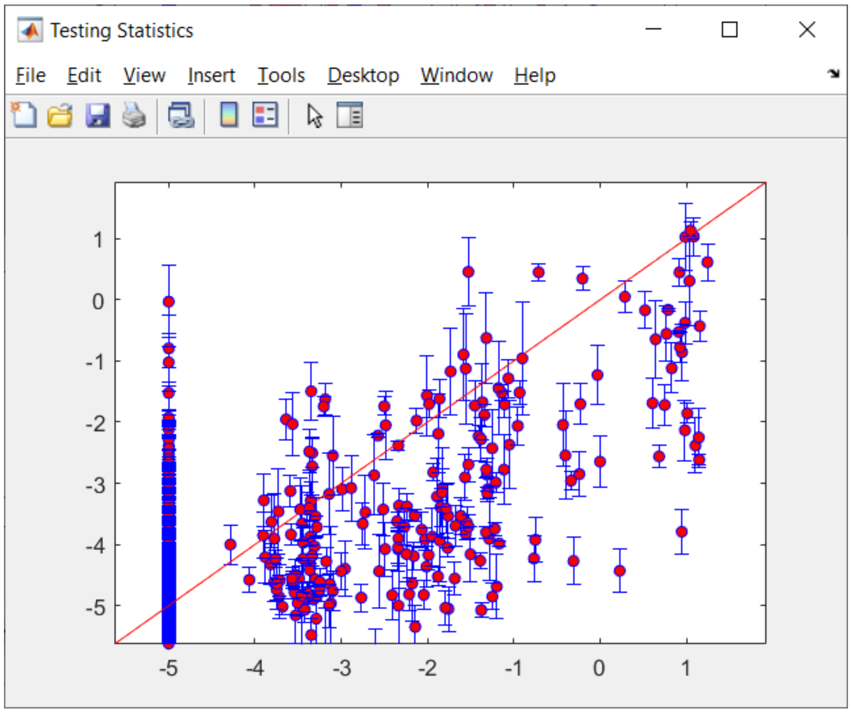




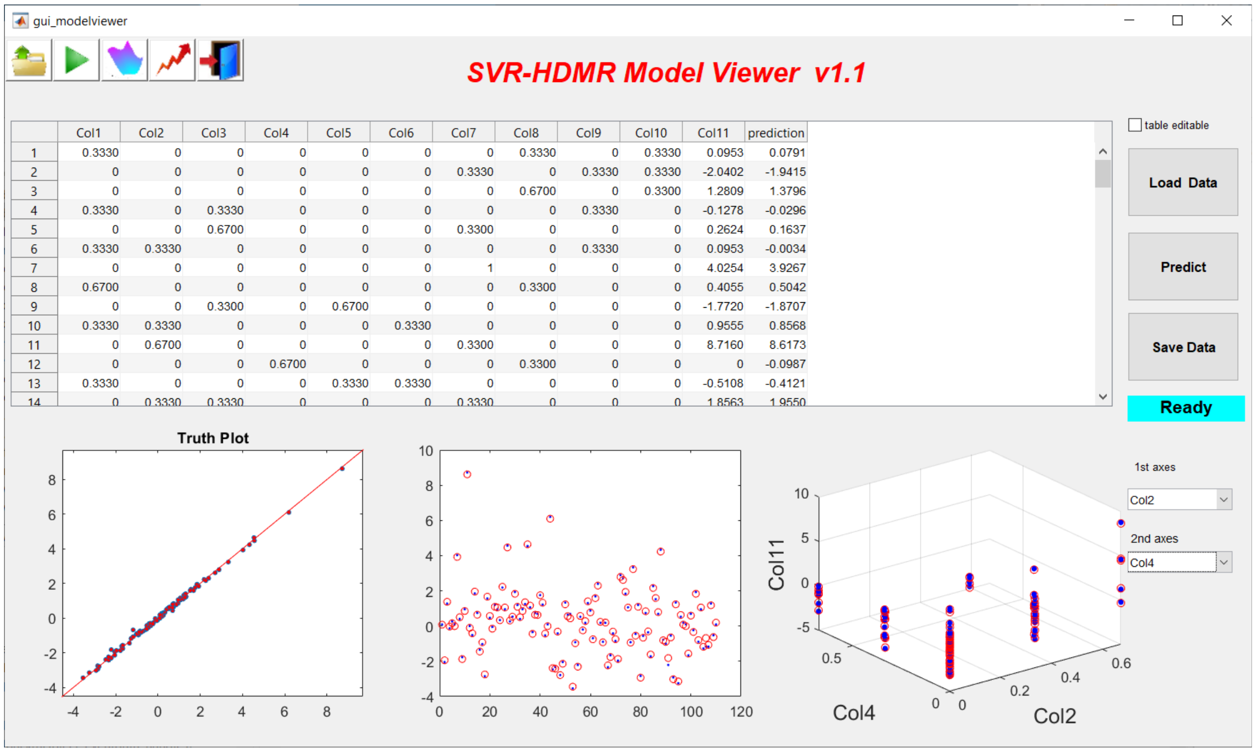


Statistical analysis:

By clicking the “Statistical analysis” button or , you will be able to input a number of iterations to perform the modeling multiple times and get the truth plots with mean and vertical error bars.



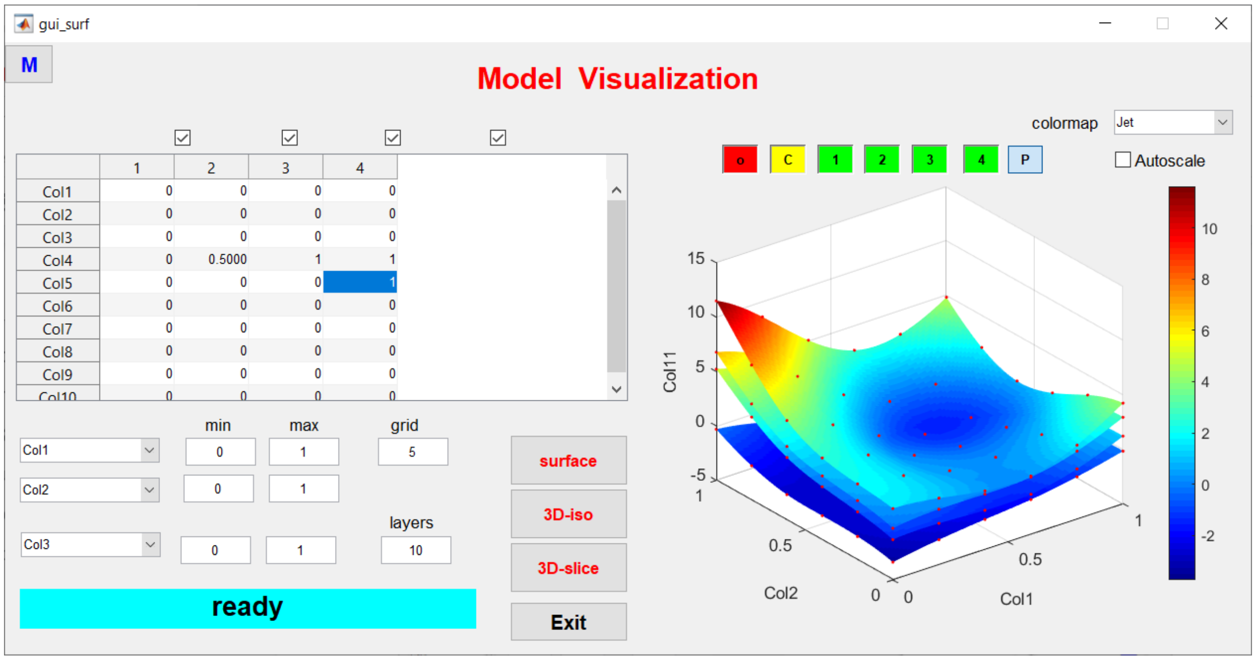
Part II. SVR-HDMR viewer:



	Col1	Col2	Col3	Col4	Col5	Col6	Col7	Col8	Col9	Col10	Col11	prediction
1	0.3330	0	0	0	0	0	0	0.3330	0	0.3330	0.0953	0.0791
2	0	0	0	0	0	0	0.3330	0	0.3330	0.3330	-2.0402	-1.9415
3	0	0	0	0	0	0	0	0.6700	0	0.3300	1.2809	1.3796
4	0.3330	0	0.3330	0	0	0	0	0	0.3330	0	-0.1278	-0.0296
5	0	0	0.6700	0	0	0	0.3300	0	0	0	0.2624	0.1637
6	0.3330	0.3330	0	0	0	0	0	0	0.3330	0	0.0953	-0.0034
7	0	0	0	0	0	0	1	0	0	0	4.0254	3.9267
8	0.6700	0	0	0	0	0	0	0.3300	0	0	0.4055	0.5042
9	0	0	0.3300	0	0.6700	0	0	0	0	0	-1.7720	-1.8707
10	0.3330	0.3330	0	0	0	0.3330	0	0	0	0	0.9555	0.8568
11	0	0.6700	0	0	0	0	0.3300	0	0	0	8.7160	8.6173
12	0	0	0	0.6700	0	0	0	0.3300	0	0	0	-0.0987
13	0.3330	0	0	0	0.3330	0.3330	0	0	0	0	-0.5108	-0.4121
14	0	0.3330	0.3330	0	0	0	0.3330	0	0	0	1.8563	1.9550

Surface views:

Clicking this icon either in the solver or the viewer will revoke the surface view gui: "gui_surf.m", for model visualization in 2D or 3D surface. Load the model first if running it as standalone, or the model from the previous step will be passed over.



	✓ 1	✓ 2	✓ 3	✓ 4
Col1	0	0	0	0
Col2	0	0	0	0
Col3	0	0	0	0
Col4	0	0.5000	1	1
Col5	0	0	0	1
Col6	0	0	0	0
Col7	0	0	0	0
Col8	0	0	0	0
Col9	0	0	0	0
Col10	0	0	0	0

	min	max	grid
Col1	0	1	5
Col2	0	1	
			layers
Col3	0	1	10

2D Surface plot
The surface plot can display up to 4 surfaces as shown above. Each layer is defined by the corresponding column in the table for fixed variables and the first two drop boxes for **two** continuously varying variables along with their lower and upper bounds (min and max). "Grid" defines the sampling size along each axis.
In this example, Col1 is the first variable and Col2 is the second variable, both have bounds [0,1]. The Z axis is the model prediction (last column: Col11). Therefore, col3 to col10 are all fixed variables, whose values have to be defined in order to make the surface plots. Their values are defined by the table (initially they are all 0 by default and you can manually change the values). Since Col1 and Col2 are selected to be varying, their values in the table can be ignored (changing them won't do anything). Here the first surface plots [col4,col5]=[0,0]; second surface plots [col4,col5]=[0.5,0]; third surface plots [col4,col5]=[1,0]; forth surface plots [col4,col5]=[1,1]; while col3 and col6-10 are all fixed at 0;
The buttons above the surface plot is used for:
1): red toggle button 'o' , red dot for sampling points on/off
2): yellow toggle button 'c', colorbar on/off
3): green toggle buttons '1-4', change the transparency of each surface plot, so by toggling on/off, you can tell which surface corresponds to which number.
4): Click the 'P' button to make a copy of the surface plot in a new figure which can be saved as a '.png' file.

5): Autoscale changes how colormap is mapped to the predicted values.
6): Colormap Dropbox: variety of colormaps to choose from.
*Note: all the variables are rescaled to [0, 1] even if in modeling it was [-1, 1]. This is because, in most applications, the variables have physical meanings and make more sense only for non-negative values.

3D plots:
For 3D plots, we attempt to visualize the subspace with an extra dimension. Therefore, **three** continuously varying variables need to be defined. Here, the third dropbox along with its min and max is used to define the selection and bounds for the 3rd variable. Similarly, all the rest variables are fixed and their values are defined by the same table.

There are two types of 3D plots available: 3D-iso and 3D-slice: See example below. They are plotting the exact same data. The number of layers is defined under "layers" (default = 10)

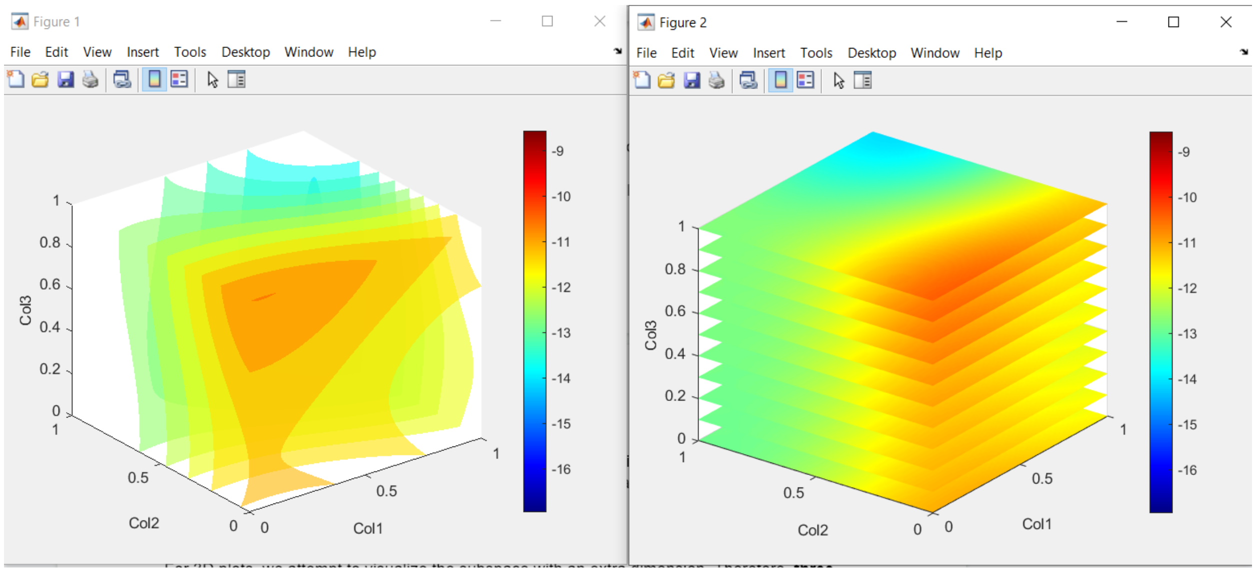


Figure 1
File Edit View Insert Tools Desktop Window Help
Col3
Col2
Col1
Figure 2
File Edit View Insert Tools Desktop Window Help
Col3
Col2
Col1

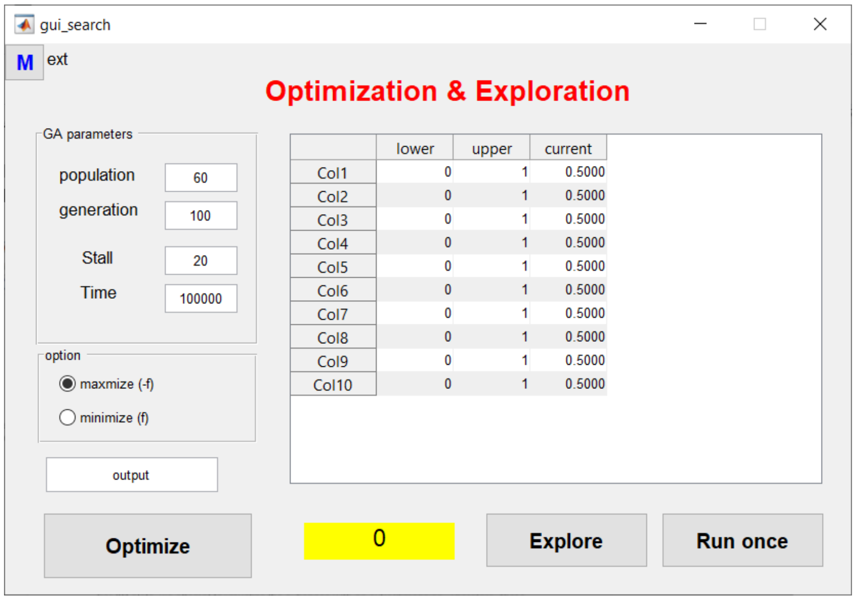
Optimal region exploration:

Clicking this icon either in the solver or the viewer will revoke the optimal region exploration gui: "gui_search.m". If running it as standalone, you need to load in model first.

Optimization:
The optimization uses GA as the optimizer with parameter settings available in the groupbox.

The option allows to set the fitness to be either maximize or minimize. Table defines the lower, upper bounds and current value for each variable. By Clicking “Run once” the fitness value will be calculated based on the current values and display in yellow highlighted.

“Optimize” will start an optimization searching for the best output value within the space defined by the table. By default, it is full space with each variable bounded [0, 1]. You can change these values to reduce the space. Making lower=upper will turn that variable to be a fixed value. It is possible to make upper>1 or lower<0, which will extend the original space, but should use that for caution.



population	60
generation	100
Stall	20
Time	100000

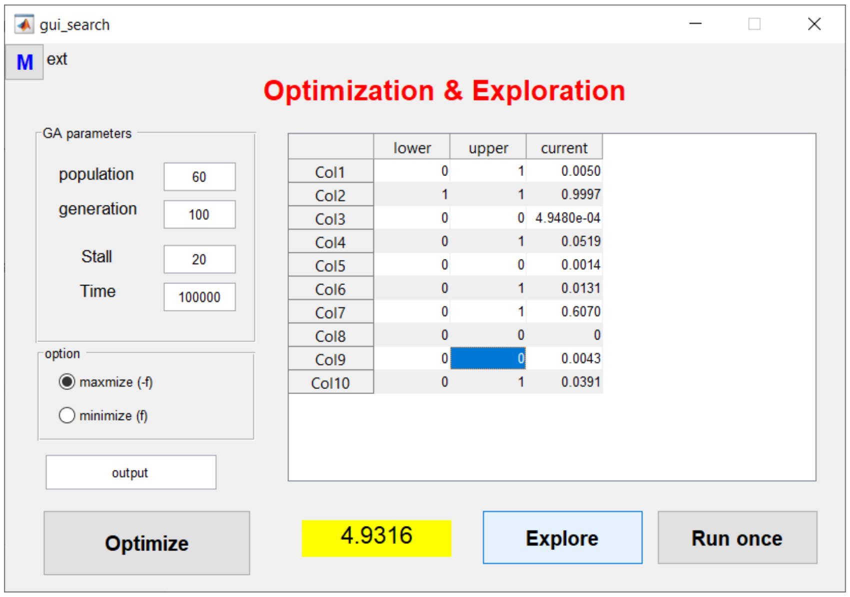
option: [● maxmize (-f)] ○ minimize (f)

	lower	upper	current
Col1	0	1	0.5000
Col2	0	1	0.5000
Col3	0	1	0.5000
Col4	0	1	0.5000
Col5	0	1	0.5000
Col6	0	1	0.5000
Col7	0	1	0.5000
Col8	0	1	0.5000
Col9	0	1	0.5000
Col10	0	1	0.5000

0

Region exploration:
The drawback for optimization is that it always returns a single point on the landscape instead of a region. Even though one can run optimization multiple times to obtain more than one optimal solution, they may or may not be diversified and people usually do not care whether it is optimal or suboptimal. Region exploration can be used in conjunction with optimization to explore a “good” region and show the statistics.

In the example below, after running an optimization, the optimal solution is shown as the current values(X) and yellow highlighted(Y), then you can manually narrow the windows (lower/upper bounds) for some selected variables to reduce the space but still containing the optimal solution. And then, clicking explore, it will do a random sampling within the new region(subspace), make y predictions, and show the statistics as compared to the one in full space. A popup dialog will show up asking for #sampling and histogram bins. Do this iteratively allows one to find a region of interest (there might be a trade off between subspace size and the mean value or distribution, which highly depends)



GA parameters

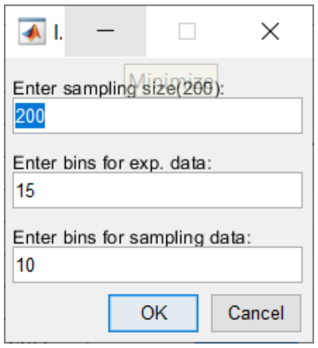
population	60
generation	100
Stall	20
Time	100000

option
[✓ maxmize (-f)]
minimize (f)

output

	lower	upper	current
Col1	0	1	0.0050
Col2	1	1	0.9997
Col3	0	0	4.9480e-04
Col4	0	1	0.0519
Col5	0	0	0.0014
Col6	0	1	0.0131
Col7	0	1	0.6070
Col8	0	0	0
Col9	0	0	0.0043
Col10	0	1	0.0391

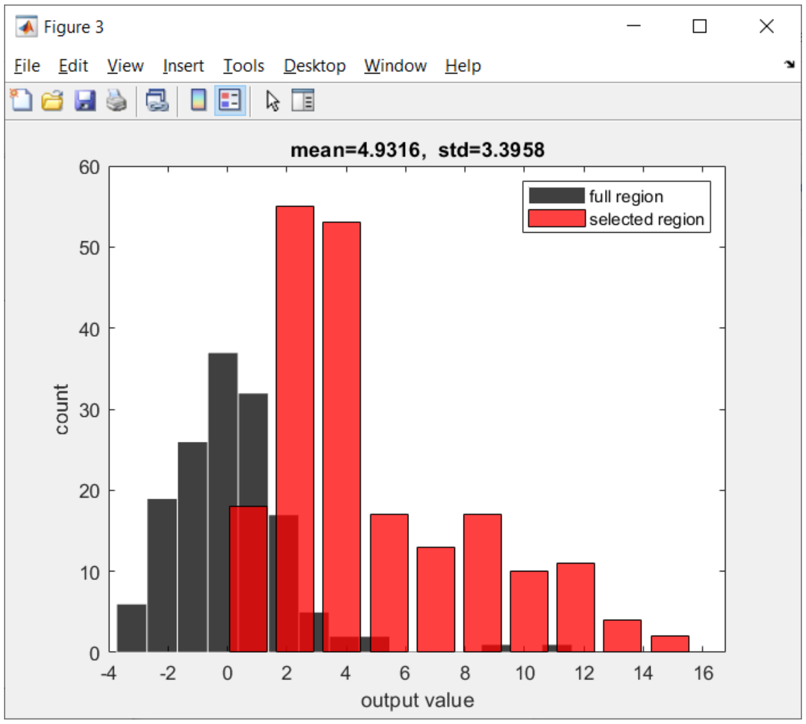
4.9316



Enter sampling size(200):
200

Enter bins for exp. data:
15

Enter bins for sampling data:
10



Predictions of new dataset:

There are two ways to use the model and make predictions on additional testing data:

1. In the model viewer, after loading in the model, Click “load data” to import new testing data (.hdd format, X only), which will show in the table. Then click “predict”, the predicted values will show as an extra column in the table. Click ‘Save data’ to export.
2. In the solver, instead of load data “all-in-one”, load modeling data and testing data separately (note: modeling data = training + validating, so it doesn’t matter how modeling data is divided into training and validating, you can leave validating as 0). Select the kernel and load in kernel parameters and click Run(manual).

Important note: extra testing data has to be processed together with the modeling data using “data preparation” to create the ‘.hdd’ file, because the normalization scaling factor could be different if processed separately. is a slightly different version of Data preparation, which allows to process one dataset and save into two split .hdd files (either by absolute numbers or percentage)