fanova Documentation

Release 1.0

Frank Hutter, Aaron Klein

Sep 07, 2017

Contents

1 Fanova **3**

1.1 Documentation 3

2 Installation **5**

2.1 Requirements 5

2.2 Manually 5

3 Manual **7**

3.1 Quick Start 7

3.2 Advanced 7

3.3 More functions 8

3.4 pysmac 8

3.5 Visualization 8

3.6 How to load a CSV-file 10

4 Citing Fanova **11**

Contents:

CHAPTER 1

Fanova

Functional ANOVA: an implementation of the ICML 2014 paper “An Efficient Approach for Assessing Hyperparameter Importance” by Frank Hutter, Holger Hoos and Kevin Leyton-Brown.

Documentation

An ‘ever growing’ documentation for the Python bindings can be found at https://automl.github.io/fanova/

CHAPTER 2

Installation

Requirements

Fanova requires:

Numpy

matplotlib (Version 1.4.2)

pyrfr

Manually

To install fanova from command line type the following commands in your bash terminal:

```
git clone https://github.com/automl/fanova.git
cd fanova/
pip install -r requirements.txt
python setup.py install
```

CHAPTER 3

Manual

Quick Start

To run the examples, just download the data and start the python console. We can then import Fanova and start it by typing

```
>>> from fanova import fANOVA
>>> import csv
>>> path = os.path.dirname(os.path.realpath(__file__))
>>> X = np.loadtxt(path + '/example_data/online_lda/online_lda_features.csv',␣
↪delimiter=",")
>>> Y = np.loadtxt(path + '/example_data/online_lda/online_lda_responses.csv',␣
↪delimiter=",")
>>> f = fANOVA(X,Y)
```

This creates a new Fanova object and fits the Random Forest on the specified data set.

To compute now the marginal of the first parameter type:

```
>>> f.quantify_importance((0, ))
    5.44551614362
```

Fanova also allows to specify parameters by their names.

```
>>> f.quantify_importance(("Col0", ))
5.44551614362
```

Advanced

If you want the Fanova only a certain quantiles (let's say between 10% and 25%) of the data you can call it by:

```
>>> f = fANOVA(X,Y)
>>> f.set_cutoffs(quantile=(10, 25))
```

Furthermore fANOVA now supports cutoffs on the y values. These will exclude parts of the parameters space where the prediction is not within the provided cutoffs.

```
>>> f.set_cutoffs(cutoffs=(-np.inf, np.inf))
```

You can also specify the number of trees in the random forest as well as the minimum number of points to make a new split in a tree or your already specified configuration space by:

```
>>> f = fANOVA(X,Y, config_space=config_space, num_trees=30, min_samples_split=3)
```

More functions

- **f.get_most_important_pairwise_marginals(n)**

Returns the **n** most important pairwise marginals

- **fANOVA.marginal_mean_variance_for_values(p, v)**

Computes the mean and standard deviation of the parameter (or parameterlist) **p** for a certain value **v**

pysmac

In order to run fANOVA on pysmac output:

```
>>> import pysmac.utils.pysmac_fanova as pysmac_fanova
>>> fanova = pysmac_fanova.smac_to_fanova('path_to/pysmac_output_dir/out/scenario',
↪'path_to/merged_states')
```

Visualization

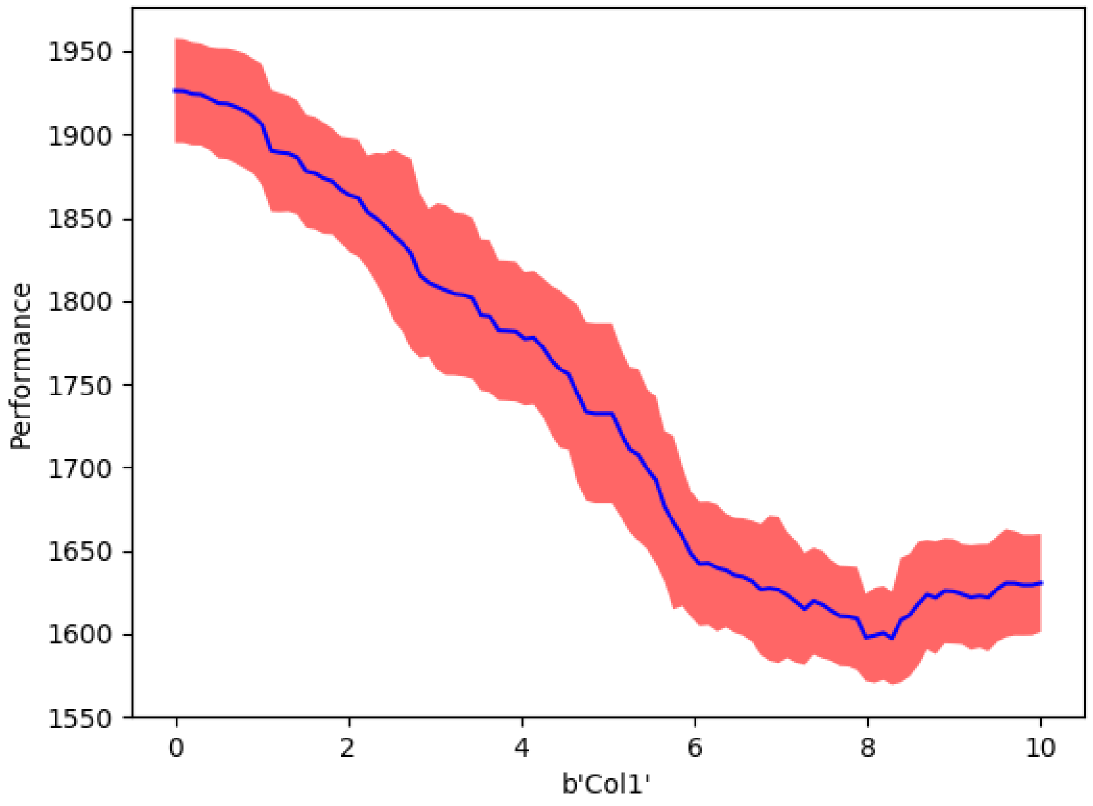
To visualize the single and pairwise marginals, we have to create a visualizer object first containing the fanova object and configspace

```
>>> import fanova.visualizer
>>> vis = fanova.visualizer.Visualizer(f, config_space)
```

We can then plot single marginals by

```
>>> vis.plot_marginal(1)
```

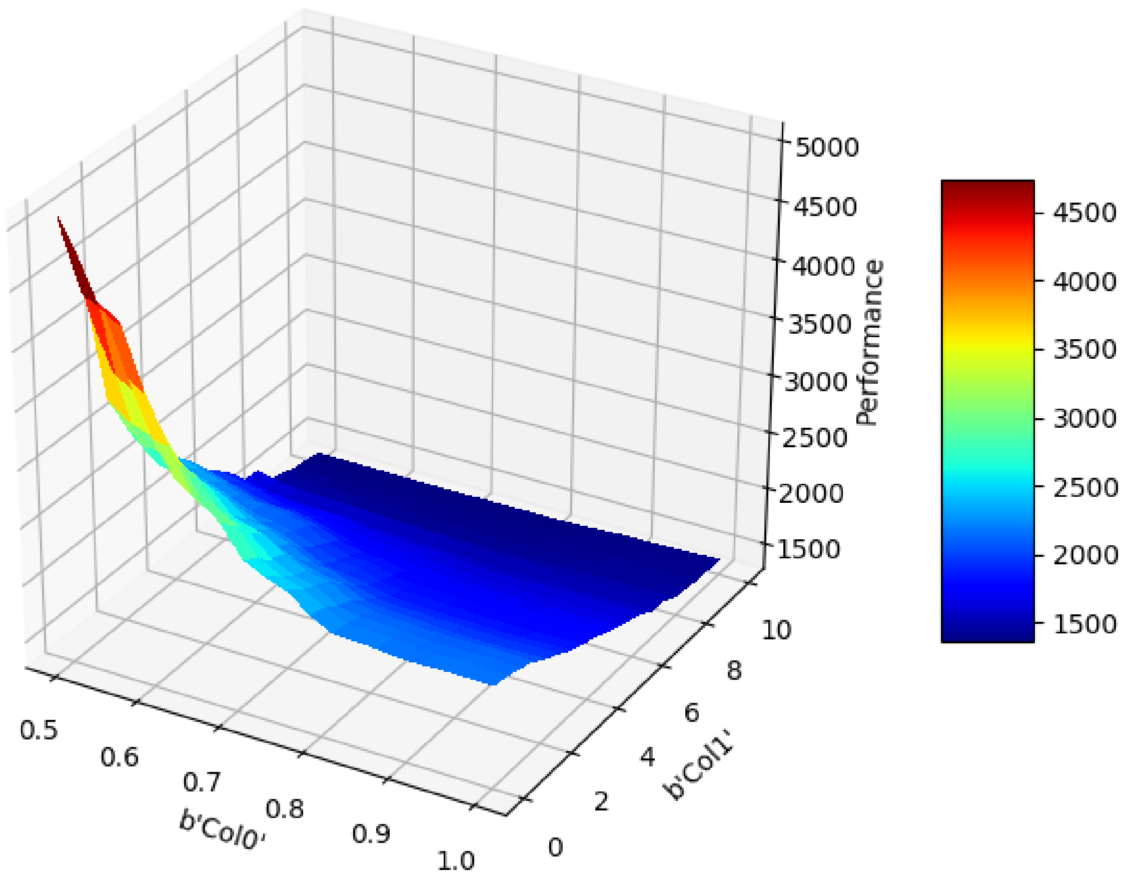
what should look like this



NOTE: For categorical values use the function plot_categorical_marginal(parameter) instead.

The same can been done for pairwise marginals

```
>>> vis.plot_pairwise_marginal([0,2])
```



If you are just interested in the N most important pairwise marginals you can plot them through:

```
>>> create_most_important_pairwise_marginal_plots(dir, n)
```

and Fanova will save those plot in dir. However, be aware that to create the plots Fanova needs to compute all pairwise marginal, which can take awhile!

If you're not interested in the plot itself, but want to extract the values for your own plots, simply call

```
>>> vis.generate_marginal(0)
```

The same for generate_pairwise_marginal([0,2]) and get_categorical_marginal().

At last, all plots can be created together and stored in a directory with

```
>>> vis.create_all_plots("./plots/")
```

How to load a CSV-file

import numpy as np

data = np.loadtxt('your_file.csv', delimiter=",")

CHAPTER 4

Citing Fanova

If you use the Fanova for your research, please cite the ICML 2014 paper “An Efficient Approach for Assessing Hyperparameter Importance” by Frank Hutter, Holger Hoos and Kevin Leyton-Brown.

with the following Bibtex file:

> @inproceedings{HutHooLey14, lauthor = {Frank Hutter and Holger Hoos and Kevin Leyton-Brown}, author = {F. Hutter and H. Hoos and K. Leyton-Brown}, title = {An Efficient Approach for Assessing Hyperparameter Importance}, booktitle = {Proceedings of International Conference on Machine Learning 2014 (ICML 2014)}, year = {2014}, pages = {754–762}, month = jun, }