
Orange Model Maps Documentation

v0.2.8

Release 0.2.8

Miha Stajdohar, University of Ljubljana, FRI

January 30, 2014

1	Scripting Reference	3
1.1	Model Maps (<code>modelmaps</code>)	3
2	Installation	5
3	Examples	7
3.1	Model map on Zoo dataset	7
3.2	Model map on Wisconsin Breast Cancer	7
3.3	Random forest on Zoo dataset	8
4	Source Code and Issue Tracker	9
5	Indices and tables	11
	Python Module Index	13
	Python Module Index	15

Orange Model Maps is an add-on for Orange data mining software package. It extends Orange by providing modules for building model maps. It also provides widgets for Orange Canvas to enable users explore model maps.

Scripting Reference

1.1 Model Maps (modelmaps)

1.1.1 Model

```
class Orange.modelmaps.Model (type_, classifier, probabilities, class_values, attributes, in-  

stance_predictions=None, instance_classes=None, name=None,  

XAnchors=None, YAnchors=None)
```

```
get_instance (domain)
```

Return `Orange.data.Table` instance with model meta-data.

Parameters `domain` (`Orange.data.Domain`) – instance will match given domain

```
set_instance (instance)
```

Set `Orange.data.Table` instance with model meta-data.

1.1.2 Build Model Map

```
class Orange.modelmaps.BuildModelMap (fname, folds=10, model_limit=500, seed=42)
```

```
build_model (attributes, learner)
```

Build a classification meta-model.

Parameters

- **attributes** (*list of strings*) – subset of attributes
- **learner** (`Orange.classification.Learner`) – classification learner to wrap

```
build_model_data (models)
```

Return an `Orange.data.Table` of model meta-data.

```
build_model_matrix (models, dist=<function distance_manhattan at 0x41bdb90>)
```

Build a distance matrix of models given the distance measure.

```
build_projection_model (attributes, viz_method)
```

Build a projection meta-model.

Parameters

- **attributes** (*list of strings*) – attributes for projection
- **viz_method** (*enum*) – visualization method

build_rf_models (*trees=50, max_depth=4, min_instances=5*)

Build Random forest and return tree models.

Parameters

- **trees** (*int*) – number of trees in the forest
- **max_depth** (*int*) – maximal tree depth
- **min_instances** (*int*) – nodes with less than min_instances instances are not split further

select_representatives (*models, dist=<function distance_euclidean at 0x41bdc08>*)

Construct a network, detect communities and return representatives.

Parameters

- **models** (list of `modelmaps.Model`) – select representatives from this models
- **dist** (*func*) – distance function

1.1.3 Help Functions

`Orange.modelmaps.modelmap.load(file_name)`

Load model map.

Read compressed tuple containing model similarity matrix and data table.

`Orange.modelmaps.modelmap.save(file_name, smx, model_data, original_data)`

Save model map.

Model similarity matrix and data table tuple is pickled and compressed as a bz2 archive.

`Orange.modelmaps.modelmap.get_models_table()`

Return an empty data table for model meta data.

`Orange.modelmaps.modelmap.get_feature_subsets(domain, nsubsets=None, min_features=None, max_features=None, seed=None)`

Return random attribute subsets.

Parameters

- **domain** (`Orange.data.Domain`) – data set domain to extract features
- **nsubsets** (*int*) – number of attribute subsets

`Orange.modelmaps.modelmap.model_network(smx, table=None, knn=1)`

Build network from model distance matrix.

Parameters

- **smx** (`Orange.misc.SymMatrix`) – model distance matrix
- **table** (`Orange.data.Table`) – model meta data
- **knn** (*int*) – connect each model with knn neighbours

1.1.4 Examples

No examples are implemented yet. Write to the author for more information.

Installation

To install Model Maps add-on for Orange from [PyPi](#) run:

```
pip install orange-modelmaps
```

To install it from source code run:

```
python setup.py install
```

To build Python egg run:

```
python setup.py bdist_egg
```

To install add-on in [development mode](#) run:

```
python setup.py develop
```

Examples

3.1 Model map on Zoo dataset

Script `mixzoo.py` builds a model map on 7 kinds of models. 3 are classic classification models:

- Naive Bayes
- k-Nearest Neighbour
- Classification Tree

where Classification Trees are taken from Random forest as proposed by Breinman. We also include 4 projections in 2-dimensional plane:

- Supervised PCA
- Radviz
- Polyviz
- Scatter plot

Projections are wrapped into k-NN classifiers that predict on projected points. The reason behind is that good projections are those that separate points well (Leban et al. 2005 and 2006).

Run the scripy with:

```
python mixzoo.py -n 500 .
```

This will create a model map in the current folder.

3.2 Model map on Wisconsin Brest Cancer

Script `mixwbc.py` builds a model map on 8 kinds of models. 4 are classic classification models:

- Naive Bayes
- k-Nearest Neighbour
- Support Vector Machine
- Classification Tree

where Classification Trees are taken from Random forest as proposed by Breinman. We also include 4 projections in 2-dimensional plane:

- Supervised PCA
- Radviz
- Polyviz
- Scatter plot

Projections are wrapped into k-NN classifiers that predict on projected points. The reason behind is that good projections are those that separate points well (Leban et al. 2005 and 2006).

Run the scripy with:

```
python mixwbcd.py -n 500 .
```

This will create a model map in the current folder.

3.3 Random forest on Zoo dataset

Script `rfzoo.py` builds a model map of Random forest classifier. Use this model map to explore the ensemble.

Run the scripy with:

```
python rfzoo.py -n 1000 .
```

This will create a model map of random forest with 1000 trees in the current folder.

Source Code and Issue Tracker

Source code is available on [Bitbucket](#). For issues and wiki we use [Trac](#).

Indices and tables

- *genindex*
- *modindex*
- *search*

e

`examples.ensemble.rfzoo`, 8
`examples.modelmix.mixwbcd`, 7
`examples.modelmix.mixzoo`, 7

O

`Orange.modelmaps`, 4
`Orange.modelmaps.model`, 3
`Orange.modelmaps.modelmap`, 3

e

`examples.ensemble.rfzoo`, 8
`examples.modelmix.mixwbcd`, 7
`examples.modelmix.mixzoo`, 7

O

`Orange.modelmaps`, 4
`Orange.modelmaps.model`, 3
`Orange.modelmaps.modelmap`, 3