
searchgrid Documentation

Release 0.1.1

Joel Nothman

Oct 25, 2017

Contents

1	Quick Start	3
2	Motivating examples	5
3	API Reference	7
4	Changelog	9
4.1	v0.2	9
	Python Module Index	11

Helps building parameter grids for `scikit-learn` grid search.

Specifying a parameter grid for `GridSearchCV` in Scikit-Learn can be annoying, particularly when:

- you change your code to wrap some estimator in, say, a `Pipeline` and then need to prefix all the parameters in the grid using lots of `__`s
- you are searching over multiple grids (i.e. your `param_grid` is a list) and you want to make a change to all of those grids

`searchgrid` allows you to define (and change) the grid together with the estimator, reducing effort and sometimes code. It stores the parameters you want to search on each particular estimator object. This makes it much more straightforward to specify complex parameter grids, and means you don't need to update your grid when you change the structure of your composite estimator.

It provides two main functions:

- `searchgrid.set_grid()` is used to specify the parameter values to be searched for an estimator or GP kernel.
- `searchgrid.make_grid_search()` is used to construct the `GridSearchCV` object using the parameter space the estimator is annotated with.

CHAPTER 1

Quick Start

If scikit-learn is installed, then, in a terminal:

```
pip install searchgrid
```

and use in Python:

```
from search_grid import set_grid, make_grid_search
estimator = set_grid(MyEstimator(), param=[value1, value2, value3])
search = make_grid_search(estimator, cv=..., scoring=...)
search.fit(X, y)
```

Or search for the best among multiple distinct estimators/pipelines:

```
search = make_grid_search([estimator1, estimator2], cv=..., scoring=...)
search.fit(X, y)
```

Motivating examples

Let's look over some of the messy change cases. We'll get some imports out of the way.:

```
>>> from sklearn.pipeline import Pipeline
>>> from sklearn.linear_model import LogisticRegression
>>> from sklearn.feature_selection import SelectKBest
>>> from sklearn.decomposition import PCA
>>> from searchgrid import set_grid, make_grid_search
>>> from sklearn.model_selection import GridSearchCV
```

Wrapping an estimator in a pipeline. You had code which searched over parameters for a classifier. Now you want to search for that classifier in a Pipeline. With plain old scikit-learn, you have to insert `__`s and change:

```
>>> gs = GridSearchCV(LogisticRegression(), {'C': [.1, 1, 10]})
```

to:

```
>>> gs = GridSearchCV(Pipeline([('reduce', SelectKBest()),
...                               ('clf', LogisticRegression())]),
...                   {'clf__C': [.1, 1, 10]})
```

With `searchgrid` we only have to wrap our classifier in a Pipeline, and do not have to change the parameter grid, adding the `clf__` prefix. From:

```
>>> lr = set_grid(LogisticRegression(), C=[.1, 1, 10])
>>> gs = make_grid_search(lr)
```

to:

```
>>> lr = set_grid(LogisticRegression(), C=[.1, 1, 10])
>>> gs = make_grid_search(Pipeline([('reduce', SelectKBest()),
...                                   ('clf', lr)]))
```

You want to change the estimator being searched in a pipeline. With scikit-learn, to use PCA instead of `SelectKBest`, you change:

```
>>> pipe = Pipeline([('reduce', SelectKBest()),
...                  ('clf', LogisticRegression())])
>>> gs = GridSearchCV(pipe,
...                   {'reduce__k': [5, 10, 20],
...                    'clf__C': [.1, 1, 10]})
```

to:

```
>>> pipe = Pipeline([('reduce', PCA()),
...                  ('clf', LogisticRegression())])
>>> gs = GridSearchCV(pipe,
...                   {'reduce__n_components': [5, 10, 20],
...                    'clf__C': [.1, 1, 10]})
```

Note that `reduce__k` became `reduce__n_components`.

With `searchgrid` it's easier because you change the estimator and the parameters in the same place:

```
>>> reduce = set_grid(SelectKBest(), k=[5, 10, 20])
>>> lr = set_grid(LogisticRegression(), C=[.1, 1, 10])
>>> pipe = Pipeline([('reduce', reduce),
...                  ('clf', lr)])
>>> gs = make_grid_search(pipe)
```

becomes:

```
>>> reduce = set_grid(PCA(), n_components=[5, 10, 20])
>>> lr = set_grid(LogisticRegression(), C=[.1, 1, 10])
>>> pipe = Pipeline([('reduce', reduce),
...                  ('clf', lr)])
>>> gs = make_grid_search(pipe)
```

Searching over multiple grids. You want to take the code from the previous example, but instead search over feature selection and PCA reduction in the same search.

Without `searchgrid`:

```
>>> pipe = Pipeline([('reduce', None),
...                  ('clf', LogisticRegression())])
>>> gs = GridSearchCV(pipe, [{'reduce': [SelectKBest()],
...                                     'reduce__k': [5, 10, 20],
...                                     'clf__C': [.1, 1, 10]},
...                        {'reduce': [PCA()],
...                                     'reduce__n_components': [5, 10, 20],
...                                     'clf__C': [.1, 1, 10]}])
```

With `searchgrid`:

```
>>> kbest = set_grid(SelectKBest(), k=[5, 10, 20])
>>> pca = set_grid(PCA(), n_components=[5, 10, 20])
>>> lr = set_grid(LogisticRegression(), C=[.1, 1, 10])
>>> pipe = set_grid(Pipeline([('reduce', None),
...                           ('clf', lr)]),
...                 reduce=[kbest, pca])
>>> gs = make_grid_search(pipe)
```

`searchgrid.build_param_grid(estimator)`

Determine the parameter grid annotated on the estimator

Parameters `estimator` : scikit-learn compatible estimator

Should have been annotated using `set_grid()`

Notes

Most often, it is unnecessary for this to be used directly, and `make_grid_search()` should be used instead.

`searchgrid.make_grid_search(estimator, **kwargs)`

Construct a GridSearchCV with the given estimator and its set grid

Parameters `estimator` : (list of) estimator

When a list, the estimators are searched over.

kwargs

Other parameters to the `sklearn.model_selection.GridSearchCV` constructor.

`searchgrid.set_grid(estimator, **grid)`

Set the grid to search for the specified estimator

Overwrites any previously set grid.

Parameters `grid` : dict (str -> list of values)

Keyword arguments define the values to be searched for each specified parameter.

Returns estimator

Useful for chaining

CHAPTER 4

Changelog

v0.2

- Fixed a bug where the grid of the default estimator in a Pipeline step was attributed to alternatives for that step. [#10](#).

S

searchgrid, [7](#)

B

`build_param_grid()` (in module `searchgrid`), [7](#)

M

`make_grid_search()` (in module `searchgrid`), [7](#)

S

`searchgrid` (module), [7](#)

`set_grid()` (in module `searchgrid`), [7](#)