Usage instructions for as-auto-sklearn

This document elaborates on the command line options for each step in the as-auto-sklearn pipeline for runtime prediction and algorithm selection.

- Training
- Testing
- Validation

Training

The `train-as-auto-sklearn` script is used to train auto-sklearn models for runtime prediction (and, later, algorithm selection). The script creates a pipeline based on configuration options to first clean the input (ASlib scenario) data; the early steps in the pipeline include things like taking the logarithm of the runtime predictions and removing forbidden features. The final step in the pipeline is always the auto-sklearn regressor.

The command line options for the `train` script control things like the number of CPUs and file paths. Please see the configuration options for more details about the preprocessing pipeline behavior.

```
train-as-auto-sklearn <scenario> <out> [--config <config>] [--solvers <solver_1> ...] [--fol
```

Command line options

- `scenario`. The ASlib scenario. This must be the path to the folder which includes the various ASlib files. For example, if the path to the description file is `/path/to/my/aslib_scenario/description.txt`, then this value should be `/path/to/my/aslib_scenario`.
- `out`. A template string for the filenames for the learned models. They are written with joblib.dump, so they need to be read back in with joblib.load. `${solver}` and `${fold}` are the "template" part of the string. It is probably necessary to surround this argument with single quotes in order to prevent shell replacement of the template parts.
- [`--config`]. A (yaml) config file which specifies options controlling the learner behavior. Please see the configuration options for more details.
- [`--solvers`]. The solvers for which models will be learned. By default, models for all solvers are learned. The names must match those in the ASlib scenario. Default: all

- [`--folds`]. The outer-cv folds for which a model will be learned. By default, models for all folds are learned. The total set of models learned is the cross-product of `solvers` and `folds`. Default: all
- [`-p/--num-cpus`]. The number of CPUs to use for parallel (solver, fold) training. Default: 1
- [`--num-blas-threads`]. The number of threads for parallelizing BLAS operations, which are used by many of the models included in auto-sklearn. The total number of CPUs will be `num_cpus * num_blas_cpus`. This option is implemented in a "best guess" approach. Currently, it is only expected to affect OpenBLAS and MKL. Please see the source code for more details. Default: 1
- [`--do-not-update-env`]. **N.B.** This flag is mostly used to control internal behavior, and it should not be used by external users.
  By default, `num-blas-threads` requires that relevant environment variables are updated. Likewise, if `num-cpus` is greater than one, it is necessary to turn off python assertions due to an issue with multiprocessing. If this flag is present, then the script assumes those updates are already handled. Otherwise, the relevant environment variables are set, and a new processes is spawned with this flag and otherwise the same arguments. This flag is not intended for external users.

Testing

The `test-as-auto-sklearn` script tests the models learned during the training phase. Importantly, the training-testing strategy uses an "outer" cross-validation strategy, so testing data is never seen during training.

Command line options

- `scenario`. The ASlib scenario. This must be the path to the folder which includes the various ASlib files. For example, if the path to the description file is `/path/to/my/aslib_scenario/description.txt`, then this value should be `/path/to/my/aslib_scenario`.
- `model_template`. A template string for the filenames for the learned models. This must match the `out` parameter used in training. It is probably necessary to surround this argument with single quotes in order to prevent shell replacement of the template parts.
- `out`. The predictions, in gzipped csv format. They are in the form of a "long" data frame with fields described below.
- [`--config`]. A (yaml) config file which specifies options controlling the learner behavior. Please see the configuration options for more details.

Output format

The following fields are present in the output file.

- `instance_id`. The name of the instance, from the ASlib scenario
- `solver`. The solver for which the prediction is made
- `fold`. The fold in which the instance is assigned in the ASlib scenario
- `actual`. The true runtime of the solver on the instance
- `predicted`. The predicted runtime of the instance on the solver using the learned models.

Validation

The `validate-as-auto-sklearn` script uses the predictions from the `test` script and the `Validator` class from `autofolio` to evaluate the algorithm selection behavior based on the predictions.

Currently, the `validate` script only writes the evaluation to the screen, but this may change depending on the `Validator` implementation.

Command line options

- `scenario`. The ASlib scenario. This must be the path to the folder which includes the various ASlib files. For example, if the path to the description file is `/path/to/my/aslib_scenario/description.txt`, then this value should be `/path/to/my/aslib_scenario`.
- `predictions`. The predictions from the `test` script.
- [`--config`]. A (yaml) config file which specifies options controlling the learner behavior. Please see the configuration options for more details. This is primarily used to calculate the time for feature extraction.