Machine Learning (ML) Framework

Create an ML-Experiment in MATLAB. An experiment consists of one or more trials, which represent a unique combination of data preparation steps, model specification, and model optimization. The experiment trials are built automatically from user defined parameters. The framework includes methods for automated machine learning (autoML), and supports supervised, semi-supervised, unsupervised and predictive maintenance (PdM) problems.

MathWorks Consulting and Application Engineering
Nick Howes and Sudheer Nuggehalli
2020

Getting started

Create a MATLAB Project to contain the experiment.

```
createProjectFromTemplate( "Name", "NewExperiment" )
```

Create a data preparation pipeline from a template. In this example, we create a template for a supervised analysis. The template contains instructions.

```
newPipelineTemplate( "byType", "Supervised" )
```

A user defines a data preparation pipeline/function. As a requirement, this function must accept a `table` type and return a `table` type. Any data preparation parameters defined in the arguments block can be explored in the experiment.

Configure an experiment. See `optimizeParameters` for valid syntax.

```
%List valid experiment parameters for specified data pipeline
validDataParameters = optimizeParameter.listPipeline( "thisPipeline" )

%List valid autoML parameters
validModelParameters = optimizeParameter.listModelParameters( "Classification" )

%Sweep Normalization parameter
```

```
dataparams(1) = optimizeParameter.new(...
    "Name", "Normalization", ...
    "Range", ["none" "range" "zscore"], ...
    "Type", "Discrete");

%Sweep frame (data processing buffer). Note each sample instance will be a vector.
dataparams(2) = optimizeParameter.new(...
    "Name", "Frame", ...
    "Range", {[200 100], [100 50], [50 25]}], ...
    "Type", "Set");

%Use a convenience method to return a default model parameters to include in
experiment. %Or define a custom set as above.
modelparams = optimizeParameter.defaultAutoMLOptimization()
```

Create and run with autoML. Supervised classification is shown.

```
    session = experiment.Classification( "Data", thisData, ...
        "DataFcn", @(data, param)thisPipeline(data, param{:}), ...
        "Model", "autoML", ...
        "DataConfiguration", dataparams, ...
        "ModelConfiguration", modelparams ...
        );

    session.run()
```

Evaluate trials. View model evaluation statistics against input parameters. In the case of classification, valid metrics for each data partition are: loss, precision, recall, F1score, and AUC.

```
session.describe( "full" )
session.sort( "errorOnCV" )
```

Select/save an experiment `item` (optimal model).

```
session.select( item_ID, "Metadata", true )

session.save( item_ID, "Metadata", true, ...
    "WriteDirectory", location);
```

Load an existing experiment item for prediction on new data. Note that this step requires defining a prediction pipeline.

```
[mdl, preparation] = loadExperimentItem(filename, ...
    "Pipeline", "thisPredictionPipeline");

predictions = thisPredictionPipeline( mdl, newdata, preparation );
```

Contents

ML Experiment

Please see doc `experiment.Type` for a list of learners available for each type / data character.

- **Supervised**
  - experiment.Classification
  - experiment.Regression
- **SemiSupervised**
  - experiment.SemiSupervised (Classification)
- **Unsupervised**
  - experiment.Cluster
- **Predictive Maintenance** (Degradation)
  - experiment.RUL

ML Utility

Convenience methods for ML data preparation, training, and evaluation/visualization. ML methods share standardized syntax and perform encoding if mixed dataset(e.g. categorical and continuous). Data preparation methods ingest and output tables and return feature names as a second output argument if features are created/removed.

Package/Utility

- ML Algorithm
  - fitc - Classification
  - fitr - Regression
  - fits - SemiSupervised
  - fitd - PdM degradation/remaining Useful Life
  - clst - Clustering
  - baseml - Base/shared ML utilities
- Data Preparation / Visualization
  - util - standard ML data preparation methods
  - viz - standard ML visualization methods