Sweep Parameters

Updated: October 5, 2015

Performs a parameter sweep on a model to determine the optimum parameter settings

Category: Machine Learning / Train (https://msdn.microsoft.com/en-us/library/azure/dn905846.aspx)

Module Overview

You can use the module to build and test models using different combinations of settings, in order to determine the optimum parameters for the given prediction task and data.

There are several ways that you can use :

- **Integrated train and sweep**: You can train any model using . However, some machine learning algorithms provide the ability to specify which parameters to sweep over during the training process.

  See the Technical Notes section for a list of all algorithms that support a integrated parameter sweep.

- **Cross validation mode:** You can use to divide your data into some number of folds and then perform tests on the models for each fold, while performing a parameter sweep to identify the best parameters. This method provides the best accuracy but can take longer to train.

With both methods, you get an accuracy report describing the different models that were created, plus a trained model that you can save for re-use.

How to Configure Sweep Parameters

If you are not sure of the correct parameters for a given machine learning algorithm or task, you might conduct multiple parameter sweeps, as well as performing feature selection to determine the columns or variables that have the highest information value.

This section describes several scenarios:

- How to configure a parameter sweep for generic machine learning models
- How to configure a parameter sweep when the machine learning method or algorithm supports an integrated parameter sweep
- How to perform cross-validation using the module

> **Tip**
>
> If you are building a clustering model, you can also use a parameter sweep to determine the optimum number of clusters and other parameters. For more information, see Sweep Clustering (https://msdn.microsoft.com/en-us/library/azure/mt484327.aspx).

To train a model using a parameter sweep

1. Add the module to your experiment.

2. Connect an untrained learner to the leftmost input.

   If the model supports an integrated parameter sweep, set the **Create trainer mode** option to **Parameter Range** and use the **Range Builder** to specify a range of values to use in the parameter sweep.

   > **Tip**
   >
   > You don't need to specify a range for all values. You can fix one or more parameters at a certain value manually and then sweep over the remaining parameters. This might save some computation time.

3. Add the dataset you want to use for cross-validation and connect it to the middle input of **Sweep Parameters**.

   If you have a tagged dataset, you can connect it to the rightmost input port (**Optional validation dataset**) to use in measuring accuracy.

4. In the **Properties** pane of , set options that define the number of parameters used and the number of iterations.

   - If you select the **Random sweep** option, specify the maximum number of runs that you want the module to execute. The module will randomly select parameter values over a system-defined range.

     This option is useful for cases where you want to increase model performance using the metrics of your choice and conserve computing resources.

   - If you select the **Grid sweep** option, the module loops over a grid predefined by the system to try different combinations and identify the best learner.

     This option is useful for cases where you don't know what the best parameter settings might be and want to try many parameters.

     There are two types of grid sweeps: one that trains a model over all possible combination of values, and one that builds the matrix of all possible values, and then randomly samples from the matrix.

5. For **Label column**, launch the column selector to choose a single label column.

6. Choose a single metric to use when assessing the best model.

   Note that when the parameter sweep is complete, all applicable metrics for the model are returned in the **Sweep results** report. However, the models are ranked by using the metric you selected and if you save the trained model to use for scoring, that model is the one that is output as the trained model.

7. Run the experiment.

8. The module outputs a set of evaluation results, indicating the parameters that produced the best models, and the accuracy of all models on the available metrics.

   The right-hand output (**Trained best model**) contains a trained model that uses the optimal parameters. You can click **View Results** and review the parameters and accuracy values, or you can click the output and select **Save as Trained Model** to use it for scoring in other experiments.

To perform cross-validation with a parameter sweep

1. Add the module to your experiment.

2. Connect an untrained learner to the leftmost input.

   If the model supports an integrated parameter sweep, set the **Create trainer mode** option to **Parameter Range** and use the **Range Builder** to specify a range of values to use in the parameter sweep.

Tip
You don't need to specify a range for all values. You can fix one or more parameters at a certain value manually and then sweep over the remaining parameters. This might save some computation time.

3. Add the dataset you want to use for cross-validation. Do not connect it to **Sweep Parameters** yet; you need to divide the data into folds for cross-validation first.

4. Add the Partition and Sample (https://msdn.microsoft.com/en-us/library/azure/dn905960.aspx) module, and connect your dataset.

   Choose the **Assign to Fold** option and optionally specify some number of folds to divide the data into.

   By default, the parameter sweep performs 10-fold cross validation, with a random split. However, you can create a different number of folds. You can also specify that stratified random sampling be used. In that case, you must specify the column that contains the values for stratification.

5. Connect the output of Partition and Sample (https://msdn.microsoft.com/en-us/library/azure/dn905960.aspx) to the middle input of .

You don't need to connect a validation dataset to the rightmost input of – for cross-validation you just need a training dataset.

6. In the **Properties** pane of , set values for parameters that determine how the sweep is performed. See the Options section for details.

7. Run the experiment.

8. The module outputs a set of evaluation results, indicating the parameters that produced the best models, and their accuracy. The accuracy metrics are calculated from the cross-validation pass, and may vary slightly depending on how many folds you selected.

   The right-hand output (**Trained best model**) contains a trained model that uses the optimal parameters. You can click the output and select **Save as Trained Model** to use it for scoring in other experiments.

Options

Specify parameter sweeping mode

Select a **grid sweep** or **random sweep**.

- A **random sweep** performs a set number of training iterations. The parameter values in each iteration are chosen randomly from the specified range of values, with replacement.

- A **grid sweep** creates a matrix that includes every combination of the parameters in the value range you specify, and then trains multiple models using these parameters. Therefore a grid sweep can take a long time.

  For a grid sweep, there are two options: **Entire grid** and **Random grid**. When you use the entire grid, each and every combination is tested. This option can be considered the most thorough, but requires the most time.

  However, recent research has shown that random sweeps can perform better than grid sweeps. If you select the random grid approach, the matrix of combinations is created and you can then specify the number of iterations to use when sampling from the matrix.

Maximum number of runs on random sweep

Specify how many times the model should be trained on a random combination of parameter values.

This option is available only if you select the **Random sweep** option.

Maximum number of runs on random grid

Specify how many times the model should be trained on a random sampling of parameter values.

The values are not generated randomly from the specified range, but are sampled from a matrix of all possible combinations of parameter values that is created when you select the **Random grid** option.

Random seed

Type a number to use when initializing the parameter sweep.

If you are training a model that supports an integrated parameter sweep, you can also set a range of seed values to use and iterate over the random seeds as well. This can be useful for avoiding bias introduced by seed selection.

Label column

Use the column selector to specify a single column to be used as the label.

Metric for measuring performance for classification

Select a metric from the dropdown list. This metric will be used to measure the accuracy of each model that is generated during the parameter sweep. A report containing the accuracy for each model is presented at the end of the parameter sweep, in the **Sweep results** output.

Different metrics are used, depending on the model type. For classification models, the options are:

- **Accuracy** The proportion of true results to total cases.
- **Precision** The proportion of true results to positive results.
- **Recall** The fraction of all correct results over all results.
- **F-score** A measure that balances precision and recall.
- **AUC** A value that represents the area under the curve when false positives are plotted on the x-axis and true positives are plotted on the y-axis.
- **Average Log Loss** The difference between two probability distributions: the true one, and the one in the model.
- **Train Log Loss** The improvement provided by the model over a random prediction.

For more information, see Machine Learning / Evaluate (https://msdn.microsoft.com/en-us/library/azure/dn906026.aspx).

Metric for measuring performance for regression

Select a metric from the dropdown list. This metric will be used to measure the accuracy of each model that is generated during the parameter sweep. A report containing the accuracy for each model is presented at the end of the parameter sweep, in the **Sweep results** output.

Different metrics are used, depending on the model type. For regression models, the options are:

- **Mean absolute error** Averages all the error in the model, where error means the distance of the predicted value from the true value. Often abbreviated as **MAE**.
- **Root of mean squared error** Measures the average of the squares of the errors, and then takes the root of that value. Often abbreviated as **RMSE**

- **Relative absolute error** Represents the error as a percentage of the true value.

- **Relative squared error** Normalizes the total squared error it by dividing by the total squared error of the predicted values.

- **Coefficient of determination** A single number that indicates how well data fits a model. A value of 1 means that the model exactly matches the data; a value of 0 means that the data is random or otherwise cannot be fit to the model. Often referred to as **r^2**, **R^2**, or **r-squared**.

For more information, see Machine Learning / Evaluate (https://msdn.microsoft.com/en-us/library/azure/dn906026.aspx).

Examples

For examples of how parameter sweeping is used in machine learning, see these experiments in the Model Gallery (http://gallery.azureml.net/):

- The Prediction of student performance (https://gallery.azureml.net/Experiment/da44bcd5dc2d4e059ebbaf94527d3d5b) sample uses the Two-Class Boosted Decision Tree (https://msdn.microsoft.com/en-us/library/azure/dn906025.aspx) algorithm with different parameters to generate a model with the best possible root mean squared error (RMSE).

- The Learning with Counts: Binary Classification (https://gallery.azureml.net/Experiment/47deb75fc7bb428194e9d0d5713350c8) sample generates a compact set of features using count-based learning, and then applies a parameter sweep to find the best model parameters.

- The Binary Classification: Network intrusion detection (https://gallery.azureml.net/Experiment/e7fb30de726e4e02b034233ec6c34ce4) sample uses in cross-validation mode, with a custom split into five folds, to find the best hyperparameters for a Two-Class Logistic Regression (https://msdn.microsoft.com/en-us/library/azure/dn905994.aspx) model.

Technical Notes

The following machine learning algorithms support an integrated parameter sweep. That is, when you are creating the model, you can specify that a parameter sweep be used during training, and set the range of parameters to use while configuring the model. You can also specify that some parameters use unchanging values and set a range of values for others.

Anomaly detection models

- One-Class Support Vector Machine (https://msdn.microsoft.com/en-us/library/azure/dn913103.aspx)
- PCA-Based Anomaly Detection (https://msdn.microsoft.com/en-us/library/azure/dn913102.aspx)

Classification models

- Multiclass Decision Forest (https://msdn.microsoft.com/en-us/library/azure/dn906015.aspx)
- Multiclass Decision Jungle (https://msdn.microsoft.com/en-us/library/azure/dn905963.aspx)
- Multiclass Logistic Regression (https://msdn.microsoft.com/en-us/library/azure/dn905853.aspx)
- Multiclass Neural Network (https://msdn.microsoft.com/en-us/library/azure/dn906030.aspx)
- Two-Class Averaged Perceptron (https://msdn.microsoft.com/en-us/library/azure/dn906036.aspx)
- Two-Class Boosted Decision Tree (https://msdn.microsoft.com/en-us/library/azure/dn906025.aspx)
- Two-Class Decision Forest (https://msdn.microsoft.com/en-us/library/azure/dn906008.aspx)
- Two-Class Decision Jungle (https://msdn.microsoft.com/en-us/library/azure/dn905976.aspx)
- Two-Class Locally Deep Support Vector Machine (https://msdn.microsoft.com/en-us/library/azure/dn913070.aspx)
- Two-Class Logistic Regression (https://msdn.microsoft.com/en-us/library/azure/dn905994.aspx)
- Two-Class Neural Network (https://msdn.microsoft.com/en-us/library/azure/dn905947.aspx)
- Two-Class Support Vector Machine (https://msdn.microsoft.com/en-us/library/azure/dn905835.aspx)

Clustering models

- K-Means Clustering (https://msdn.microsoft.com/en-us/library/azure/dn905944.aspx)

For clustering models, you must use the separate Sweep Clustering (https://msdn.microsoft.com/en-us/library/azure/mt484327.aspx) module.

Regression models

- Boosted Decision Tree Regression (https://msdn.microsoft.com/en-us/library/azure/dn905801.aspx)
- Decision Forest Regression (https://msdn.microsoft.com/en-us/library/azure/dn905862.aspx)
- Fast Forest Quantile Regression (https://msdn.microsoft.com/en-us/library/azure/dn913093.aspx)
- Linear Regression (https://msdn.microsoft.com/en-us/library/azure/dn905978.aspx)

  The integrated parameter sweep is available only if you choose the online gradient descent method.
- Neural Network Regression (https://msdn.microsoft.com/en-us/library/azure/dn905924.aspx)
- Poisson Regression (https://msdn.microsoft.com/en-us/library/azure/dn905988.aspx)

Expected Inputs

Name	Type	Description
Untrained model	ILearner interface (https://msdn.microsoft.com/en-us/library/azure/dn905938.aspx)	Untrained model for parameter sweep
Training dataset	Data Table (https://msdn.microsoft.com/en-us/library/azure/dn905851.aspx)	Input dataset for training
Validation dataset	Data Table (https://msdn.microsoft.com/en-us/library/azure/dn905851.aspx)	Input dataset for validation (for Train/Test validation mode). This input is optional.

Module Parameters

Name	Range	Type	Default	Description
	List	Sweep Methods		

Specify parameter sweeping mode			Random sweep	Sweep entire grid on parameter space, or sweep with using a limited number of sample runs
Maximum number of runs on random sweep	[1;10000]	Integer	5	Execute maximum number of runs using random sweep
Random seed	any	Integer	0	Provide a value to seed the random number generator
Label column	any	ColumnSelection		Label column
Metric for measuring performance for classification	List	Binary Classification Metric Type	Accuracy	Select the metric used for evaluating classification models
Metric for measuring performance for regression	List	RegressionMetric Type	Mean absolute error	Select the metric used for evaluating regression models

Outputs

Name	Type	Description
Sweep results	Data Table (https://msdn.microsoft.com/en-us/library/azure/dn905851.aspx)	Results metric for parameter sweep runs
Trained best model	ILearner interface (https://msdn.microsoft.com/en-us/library/azure/dn905938.aspx)	Model with best performance on the training dataset

See Also

A-Z List of Machine Learning Studio Modules (https://msdn.microsoft.com/en-us/library/azure/dn906033.aspx)
Machine Learning / Train (https://msdn.microsoft.com/en-us/library/azure/dn905846.aspx)
Cross-Validate Model (https://msdn.microsoft.com/en-us/library/azure/dn905852.aspx)