Boosted Decision Tree Regression

Updated: July 13, 2015

Creates a regression model using the Boosted Decision Tree algorithm

Category: Machine Learning / Initialize Model / Regression (https://msdn.microsoft.com/en-us/library/azure/dn905922.aspx)

Module Overview

You can use the **Boosted Decision Tree Regression** module to create an ensemble of regression trees using boosting. *Boosting* means that each tree is dependent on prior trees, and learns by fitting the residual of the trees that preceded it. Thus, boosting in a decision tree ensemble tends to improve accuracy with some small risk of less coverage.

This regression method is a supervised learning method, and therefore requires a *tagged dataset*, which includes a label column. The label column must contain numerical values.

You can train the model by providing the model and the tagged dataset as an input to Train Model (https://msdn.microsoft.com/en-us/library/azure/dn906044.aspx) or Sweep Parameters (https://msdn.microsoft.com/en-us/library/azure/dn905810.aspx). The trained model can then be used to predict values for the new input examples.

Use this module only with datasets that use numerical variables.

Understanding Boosted Regression Trees

Boosting is one of several classic methods for creating ensemble models, along with bagging, random forests, and so forth. In Azure Machine Learning Studio, boosted decision trees use an efficient implementation of the MART gradient boosting algorithm. Gradient boosting is a machine learning technique for regression problems. It builds each regression tree in a step-wise fashion, using a predefined loss function to measure the error in each step and correct for it in the next. Thus the prediction model is actually an ensemble of weaker prediction models.

In regression problems, boosting builds a series of trees in in a step-wise fashion, and then selects the optimal tree using an arbitrary differentiable loss function.

For additional information, see these articles:

- Wikipedia article on boosted trees. In particular see the section about gradient tree boosting.

- Microsoft Research: From RankNet to LambdaRank to LambdaMART: An OverviewC.J.C. Burges.

  http://research.microsoft.com/apps/pubs/default.aspx?id=132652 (http://research.microsoft.com/apps/pubs/default.aspx?id=132652)

The gradient boosting method can also be used for classification problems by reducing them to regression with a suitable loss function. For more information about the boosted trees implementation for classification tasks, see Two-Class Boosted Decision Tree (https://msdn.microsoft.com/en-us/library/azure/dn906025.aspx).

How to Configure a Boosted Decision Tree Regression Model

1. Specify how you want the model to be trained, by setting the **Create trainer mode** option.

   When you create this model, you have two options for training the model: using a single set of parameters with Train Model (https://msdn.microsoft.com/en-us/library/azure/dn906044.aspx), or choosing a range of parameters and training the model with a parameter sweep.

   - **Single Parameter mode**

     If you know how you want to configure the regression model, you can provide a specific set of values as arguments. You might have learned these values by experimentation or received them as guidance.

   - **Sweep mode**

     If you are not sure of the best parameters, you can find the optimal parameters by using Sweep Parameters (https://msdn.microsoft.com/en-us/library/azure/dn905810.aspx) to train the model.

2. If you choose the **Single Parameter** option, type other values in the **Properties** pane that control the behavior of the regression model, such as leaves per tree and learning rate. See the section for details.Bookmark link 'bkmk_Options' is broken in topic '{"project_id":"37f8d135-1f1d-4e57-9b7d-b084770c6bf5","entity_id":"0207d252-6c41-4c77-84c3-73bdf1ac5960","entity_type":"Article","locale":"en-US"}'. Rebuilding the topic '{"project_id":"37f8d135-1f1d-4e57-9b7d-b084770c6bf5","entity_id":"0207d252-6c41-4c77-84c3-73bdf1ac5960","entity_type":"Article","locale":"en-US"}' may solve the problem.

Then, connect the model to the Train Model (https://msdn.microsoft.com/en-us/library/azure/dn906044.aspx) module, along with a labeled training dataset, and run the experiment

3. If you choose the **Parameter Range** option, use the **Range Builder** to set an upper and lower range for each numeric parameter.

   Then, connect the model to the Sweep Parameters (https://msdn.microsoft.com/en-us/library/azure/dn905810.aspx) module, along with a labeled training dataset, and run the experiment.

   Sweep Parameters (https://msdn.microsoft.com/en-us/library/azure/dn905810.aspx) will iterate over all possible combinations of the settings you provided and determine the combination of settings that produces the optimal results. You can use the model trained using those parameters, or you can make a note of the parameter settings to use when configuring a learner.

Options

You can customize the behavior of the regression model by using these parameters:

Create trainer mode

Choose the method used for configuring and training the model:

- ***Single Parameter***

  Select this option to configure and train the model with a single set of parameter values that you supply.

  If you choose this option, you should train the model by using the Train Model (https://msdn.microsoft.com/en-us/library/azure/dn906044.aspx) module.

- ***Parameter Range***

  Select this option to use the Range Builder and specify a range of possible values. You then train the model using a parameter sweep, to find the optimum configuration.

Warning

- If you pass a parameter range to Train Model (https://msdn.microsoft.com/en-us/library/azure/dn906044.aspx), it will use only the first value in the parameter range list.
- If you pass a single set of parameter values to the Sweep Parameters (https://msdn.microsoft.com/en-us/library/azure/dn905810.aspx) module, when it expects a range of settings for each parameter, it ignores the values and using the default values for the learner.

- If you select the **Parameter Range** option and enter a single value for any parameter, that single value you specified will be used throughout the sweep, even if other parameters change across a range of values.

Maximum number of leaves per tree
Type a value to constrain the number of leaves in the tree.

Maximum number of samples per leaf node
Type a value that defines the number of cases required to form any single terminal node, or leaf.

Learning rate
Type a value that defines the step taken at each iteration, before correction.

A larger value for learning rate can cause the model to converge faster, but it can overshoot local minima.

Total number of trees constructed
Type a value to constrain the number of total trees in the ensemble.

Random number seed
Type a value to use as the seed.

Specifying a seed value is useful when you want to ensure repeatability across runs of the same experiment.

Allow unknown categorical levels
When this option is selected, the model will create a grouping for Unknown values in the training validation sets.

If you deselect it, the model can accept only the values contained in the training data. In the former case, the model might be less precise on known values but provide better predictions for new (unknown) values.

Recommendations

In general, decision trees yield better results when features are somewhat related. If features have a large degree of entropy (that is, they are not related), they share little or no mutual information, and ordering them in a tree will not yield a lot of predictive significance.

Examples

For examples of how boosted trees are used in machine learning, see these sample experiments in the Model Gallery (http://gallery.azureml.net/):

- The Demand estimation (http://go.microsoft.com/fwlink/?LinkId=525271) sample uses **Boosted Decision Tree Regression** to predict the number of rentals for a particular time.

- The Twitter sentiment analysis (http://go.microsoft.com/fwlink/?LinkId=525274) sample uses regression to generate a predicted rating.

Technical Notes

- The ensemble of trees is produced by computing, at each step, a regression tree that approximates the gradient of the loss function, and adding it to the previous tree with coefficients that minimize the loss of the new tree.
- The output of the ensemble produced by MART on a given instance is the sum of the tree outputs.
- For binary classification problem, the output is converted to probability by using some form of calibration.
- For regression problems, the output is the predicted value of the function.
- For ranking problem, the instances are ordered by the output value of the ensemble.

Module Parameters

Name	Range	Type	Default	Description
Maximum number of leaves per tree	>=1	Integer	20	Specify the maximum number of leaves per tree
Minimum number of samples per leaf node	>=1	Integer	10	Specify the minimum number of cases required to form a leaf node
Learning rate	[double.Epsilon;1.0]	Float	0.2	Specify the initial learning rate
Total number of trees constructed	>=1	Integer	100	Specify the maximum number of trees that can be created during training
Random number seed	any	Integer		Provide a seed for the random number generator used by the model. Leave blank for default.

Allow unknown categorical levels	any	Boolean	true	If true, create an additional level for each categorical column. Levels in the test dataset not available in the training dataset are mapped to this additional level.

Outputs

Name	Type	Description
Untrained model	ILearner interface (https://msdn.microsoft.com/en-us/library/azure/dn905938.aspx)	An untrained regression model

See Also

A-Z List of Machine Learning Studio Modules (https://msdn.microsoft.com/en-us/library/azure/dn906033.aspx)
Machine Learning / Initialize Model / Regression (https://msdn.microsoft.com/en-us/library/azure/dn905922.aspx)