Two-Class Decision Jungle

Updated: October 13, 2015

Creates a two-class classification model using the decision jungle algorithm

Category: Machine Learning / Initialize Model / Classification (https://msdn.microsoft.com/en-us/library/azure/dn905808.aspx)

Module Overview

You can use the **Two-Class Decision Jungle** module to create a machine learning model that is based on a supervised ensemble learning algorithm called decision jungles.

The **Two-Class Decision Jungle** module returns an untrained classifier that can be passed to another module, such as Train Model (https://msdn.microsoft.com/en-us/library/azure/dn906044.aspx) to Sweep Parameters (https://msdn.microsoft.com/en-us/library/azure/dn905810.aspx), for training on a labeled training data set. The trained model can then be used to make predictions. Alternatively, the untrained model can be passed to Cross-Validate Model (https://msdn.microsoft.com/en-us/library/azure/dn905852.aspx) for cross-validation against a labeled data set.

Understanding Decision Jungles

Decision jungles (http://go.microsoft.com/fwlink/?LinkId=403675) are a recent extension to decision forests (http://go.microsoft.com/fwlink/?LinkId=403677). A decision jungle consists of an ensemble of decision directed acyclic graphs (DAGs).

Decision jungles have the following advantages:

- By allowing tree branches to merge, a decision DAG typically has a lower memory footprint and better generalization performance than a decision tree, albeit at the cost of somewhat longer training time.

- Decision jungles are non-parametric models that can represent non-linear decision boundaries.

- They perform integrated feature selection and classification and are resilient in the presence of noisy features.

How to Configure a Decision Jungle Model

1. Drag the **Two-Class Decision** Forest module into your experiment.

2. For **Resampling method**, specify the method to use for creating multiple trees.

   See the Recommendations section for guidance.

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

   - **Single Parameter**

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

   - **Parameter Range**

     If you are not sure of the best parameters, you can find the optimal parameters by specifying multiple values and using a parameter sweep to find the optimal configuration.

     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.

4. Set other model parameters as required. For more information, see the Options section.

5. Connect a labeled dataset and train the model. For this model type, the label column in the dataset can contain no more than two outcomes.

   - If you set **Create trainer mode** option to **Single Parameter**, train the model by using a labeled dataset and the Train Model (https://msdn.microsoft.com/en-us/library/azure/dn906044.aspx) module.

   - If you set **Create trainer mode** option to **Parameter Range**, train the model using Sweep Parameters (https://msdn.microsoft.com/en-us/library/azure/dn905810.aspx) and a labeled dataset.

     You can then use the trained model, or you can make a note of the parameter settings to use when configuring a learner.

6. The trained model can then be used to make predictions. Alternatively, the untrained model can be passed to Cross-Validate Model (https://msdn.microsoft.com/en-us/library/azure/dn905852.aspx) for cross-validation against a labeled data set.

Options

This module supports extensive customization of the decision jungle model using these parameters:

Resampling method

Specify which method should be used to bootstrap graph creation.

- **Bagging**. Select this option to use bagging, also called bootstrap aggregating.

  Each tree in a decision forest outputs a Gaussian distribution by way of prediction. The aggregation is to find a Gaussian whose first two moments match the moments of the mixture of Gaussians given by combining all Gaussians returned by individual trees.

- ***Replicate***. In replication, each tree is trained on exactly the same input data. The determination of which split predicate is used for each tree node remains random and the trees will be diverse.

  For more information about the training process with the **Replicate** option, see Decision Forests for Computer Vision and Medical Image Analysis. Criminisi and J. Shotton. Springer 2013. (http://research.microsoft.com/en-us/projects/decisionforests/)

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.

Number of decision DAGs

Type a number to constrain the total number of graphs that can be created in the ensemble.

If you use a parameter sweep, you can either set a single value, or use the Range Builder to set multiple values to use when building each ensemble.

Maximum depth of the decision DAGs

Type a value to constrain the depth of the graphs.

If you use a parameter sweep, you can either set a single value, or use the Range Builder to set multiple values to use when building each ensemble.

Maximum width of the decision DAGs

Type a value to constrain the width of the graphs.

If you use a parameter sweep, you can either set a single value, or use the Range Builder to set multiple values to use when building each ensemble.

Number of optimization steps per decision DAG layer

Type a value that indicates the number of iterations to perform when building each DAG.

If you use a parameter sweep, you can either set a single value, or use the Range Builder to set multiple values to use when building each ensemble.

Allow unknown values for categorical features

Select this option to create a group for unknown values in testing or validation data.

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

Recommendations

If you have limited data or want to minimize the time spent training the model, try these settings:

Limited training set: If the training set contains a limited number of instances:

- Create the decision jungle by using a large number of decision DAGs (for example, more than 20).
- Use the **Bagging** option for resampling.
- Specify a large number of optimization steps per DAG layer (for example, more than 10,000).

Limited training time: If the training set contains a large number of instances and training time is limited:

- Create the decision jungle using a fewer number of decision DAGs (for example, 5-10).
- Use the **Replicate** option for resampling.
- Specify a smaller number of optimization steps per DAG layer (for example, less than 2000).

Example

For examples of how decision jungles are used in machine learning, see this sample experiment in the Model Gallery (http://gallery.azureml.net/):

- The Compare Binary Classifiers (https://gallery.azureml.net/Experiment/b2bfde196e604c0aa2f7cba916fc45c8) sample uses several algorithms and discusses their pros and cons.

Technical Notes

Module Parameters

Name	**Range**	**Type**	**Default**	**Description**
Resampling method	Any	ResamplingMethod	Bagging	Choose a resampling method
Number of decision DAGs	>=1	Integer	8	Specify the number of decision graphs to build in the ensemble
Maximum depth of the decision DAGs	>=1	Integer	32	Specify the maximum depth of the decision graphs in the ensemble
Maximum width of the decision DAGs	>=8	Integer	128	Specify the maximum width of the decision graphs in the ensemble

Number of optimization steps per decision DAG layer	>=1000	Integer	2048	Specify the number of steps to use to optimize each level of the decision graphs
Allow unknown values for categorical features	Any	Boolean	True	Indicate whether unknown values of existing categorical features can be mapped to a new, additional feature

Output

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

See Also

Machine Learning / Initialize Model / Classification (https://msdn.microsoft.com/en-us/library/azure/dn905808.aspx)
Multiclass Decision Jungle (https://msdn.microsoft.com/en-us/library/azure/dn905963.aspx)
A-Z List of Machine Learning Studio Modules (https://msdn.microsoft.com/en-us/library/azure/dn906033.aspx)