Two-Class Averaged Perceptron

Updated: July 13, 2015

Creates an averaged perceptron binary classification model

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

Module Overview

You can use the **Two-Class Averaged Perceptron** module to create a machine learning model that uses the averaged perceptron algorithm.

Classification is a supervised learning method, and therefore requires a *tagged dataset*, which includes a label column.

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.

Understanding the Averaged Perceptron Model

The *averaged perceptron method* is an early and very simple version of a neural network.

In this supervised learning method, inputs are classified into several possible outputs based on a linear function, and then combined with a set of weights that are derived from the feature vector —hence the name "perceptron."

The simpler perceptron models are suited to learning linearly separable patterns, whereas neural networks (especially deep neural networks) can model more complex class boundaries. However, perceptrons are faster, and because they process cases serially, perceptrons can be used with continuous training.

How to Configure an Averaged Perceptron Model

1. 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 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.

2. Specify properties that affect the behavior of the two-class averaged perceptron model, such as the learning rate and random number seed.

   See the Options section for details.

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

   If you set the **Create trainer mode** option to **Parameter Range**, add a tagged dataset and train the model using Sweep Parameters (https://msdn.microsoft.com/en-us/library/azure/dn905810.aspx). 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 modify the behavior of this learner by using the following settings:

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.

Learning rate

Specify a value for the learning rate.

The learning rate values controls the size of the step that is used in stochastic gradient descent each time the model is tested and corrected. By making the rate smaller, you test the model more often, with the risk that you will get stuck in a local plateau. By making the step larger, you can converge faster, at the risk of overshooting the true minima.

Maximum number of iterations

Specify a number to control the maximum iterations over the training data.

Stopping early often provides better generalization. Increasing the number of iterations improves fitting, at the risk of overfitting.

Random number seed

Type a value to use as the seed.

Always use a seed if you want to ensure reproducibility of the experiment across runs.

Allow unknown categorical levels

Select this option to create a group for unknown values in the training and validation sets.

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

For this model type, it is a best practice to normalize datasets before using them to train the classifier.

The averaged perceptron model is an early and simplified version of neural networks. As such, it works well on simple data sets when your goal is speed over accuracy. However, if you are not getting the desired results, you can try one of these models:

- Two-Class Neural Network (https://msdn.microsoft.com/en-us/library/azure/dn905947.aspx) or Multiclass Neural Network (https://msdn.microsoft.com/en-us/library/azure/dn906030.aspx)
- Two-Class Logistic Regression (https://msdn.microsoft.com/en-us/library/azure/dn905994.aspx) or Multiclass Logistic Regression (https://msdn.microsoft.com/en-us/library/azure/dn905853.aspx)
- Two-Class Boosted Decision Tree (https://msdn.microsoft.com/en-us/library/azure/dn906025.aspx)

Example

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

- The Cross Validation for Binary Classifiers sample (http://go.microsoft.com/fwlink/?LinkId=525734) compares multiple classification models.

Module Parameters

Name	Range	Type	Default	Description
Learning rate	>=double.Epsilon	Float	1.0	The initial learning rate for the Stochastic Gradient Descent optimizer.
Maximum number of iterations	>=1	Integer	10	The number of Stochastic Gradient Descent iterations to be performed over the training dataset.
Random number seed	Any	Integer		The seed for the random number generator used by the model. Leave it blank for the default.
Allow unknown categorical levels	Any	Boolean	True	If True, creates an additional level for each categorical column. Any levels in the test dataset that are not

				available in the training dataset are mapped to this additional level.

Output

Name	Type	Description
Untrained model	ILearner interface (https://msdn.microsoft.com/en-us/library/azure/dn905938.aspx)	An untrained binary classification model that can be connected to the One-vs-All Multiclass (https://msdn.microsoft.com/en-us/library/azure/dn905887.aspx), Train Model (https://msdn.microsoft.com/en-us/library/azure/dn906044.aspx), or Cross-Validate Model (https://msdn.microsoft.com/en-us/library/azure/dn905852.aspx) modules.

See Also

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