Two-Class Support Vector Machine

Updated: July 13, 2015

Creates a binary classification model using the Support Vector Machine 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 Support Vector Machine** module to create a model that is based on the support vector machine algorithm. The classifier that this module initializes is useful for predicting between two possible outcomes that depend on continuous or categorical predictor variables.

This model 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 Support Vector Machines

Support vector machines (SVMs) are supervised learning models that analyze data and recognize patterns. They can be used for classification and regression tasks.

The classifier that is created by this module is useful for predicting between two possible outcomes that depend on continuous or categorical predictor variables.

Given a set of training examples labeled as belonging to one of two classes, the SVM algorithm assigns new examples into one category or the other. The examples are represented as points in space, and they are mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

The feature space that contains the training examples is sometimes called a *hyperplane*, and it may have many dimensions.

How to Configure an SVM 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 neural network, 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.

      During training, all combinations of the settings you provided will be tested to determine the combination of settings that produces the optimal results.

2. Set other parameters that control the behavior of the support vector model, such as.

   For more information, see the Options section.

3. Connect a tagged dataset and train the model.

    - If you set **Create trainer mode** option to **Single Parameter**, train the model by using a tagged 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 tagged dataset. You can then use the trained model, or you can make a note of the parameter settings to use when configuring a learner.

Options

The module can be customized 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.

Number of iterations

Type a number that denotes the number of iterations used when building the model.

This parameter can be used to control trade-off between training speed and accuracy.

Lambda

Specify a value to use as the weight for L1 regularization.

This regularization coefficient can be used to tune the model. Larger values penalize more complex models.

Normalize features

Select this option to normalize features before training.

If you apply normalization, before training, data points are centered at the mean and scaled to have one unit of standard deviation.

Project to the unit sphere

Select this option to normalize coefficients by projecting values to unit space.

Before training, data points are centered at the mean and scaled to have one unit of standard deviation.

Random number seed

Specify a seed if you want to ensure reproducibility across runs.

Allow unknown category

Select this option to create a group for unknown values in the training or 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 recommended that you normalize the dataset before using it to train the classifier.

Support vector machines are among the earliest of machine learning algorithms, and SVM models have been used in many applications, from information retrieval to text and image classification. Although recent research has developed algorithms that have higher accuracy, this algorithm can work well on simple data sets when your goal is speed over accuracy. If you do not get the desired results by using **Two-Class Support Vector Model**, try one of these classification methods:

- 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)

Examples

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

- The Direct marketing (http://go.microsoft.com/fwlink/?LinkId=525168) sample uses an SVM model to classify customers by appetency.
- The Credit risk prediction (http://go.microsoft.com/fwlink/?LinkId=525270) sample uses SVM for assessing credit risk.
- The Compare Multiclass Classifiers (http://go.microsoft.com/fwlink/?LinkId=525730) sample uses an SVM model for handwriting recognition.

Module Parameters

Name	Range	Type	Default	Description
Number of iterations	>=1	Integer	1	The number of iterations

Lambda	>=double.Epsilon	Float	0.001	Weight for L1 regularization. Using a non-zero value avoids overfitting the model to the training dataset.
Normalize features	Any	Boolean	True	If True, normalize the features.
Project to the unit-sphere	Any	Boolean	False	If True, project the features to a unit circle.
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	Data Table (https://msdn.microsoft.com/en-us/library/azure/dn905851.aspx)	An untrained binary classification model.

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)