K-Means Clustering

Published: March 2, 2015

Updated: August 31, 2015

Configures and initializes a K-means clustering model

Category: Machine Learning / Initialize Model / Clustering (https://msdn.microsoft.com/en-us/library/azure/dn905908.aspx)

Module Overview

You can use the **K-Means Clustering** module to create an untrained K-means clustering model. K-means is one of the simplest and the best known *unsupervised* learning algorithms, and can be used for a variety of machine learning tasks, such as detecting abnormal data (https://msdn.microsoft.com/magazine/jj891054.aspx), clustering of text documents, and analysis of a dataset prior to using other classification or regression methods.

After you have configured the module parameters, you must pass the untrained model to the Train Clustering Model (https://msdn.microsoft.com/en-us/library/azure/dn905873.aspx) or the Sweep Clustering (https://msdn.microsoft.com/en-us/library/azure/mt484327.aspx) modules to train the model on a set of input data that you provide.

Because the K-means algorithm is an *unsupervised* learning method, the data you use to train the model does not need a label column. In other words, you don't need to know any of the cluster categories in advance; the algorithm will find possible categories based solely on the data.

If your training data already has labels, use one of the supervised classification (https://msdn.microsoft.com/en-us/library/azure/dn905808.aspx) methods provided in Azure Machine Learning. Or, you can use the label values to guide selection of the clusters.

Understanding K-Means Clustering

In general, clustering uses iterative techniques to group cases in a dataset into clusters that contain similar characteristics. These groupings are useful for exploring data, identifying anomalies in the data, and eventually for making predictions. Clustering models also can help you identify relationships in a dataset that you might not logically derive through casual observation. For this reason, clustering is often used in the early phases of machine learning task to explore the data and discover correlations that may have been unexpected.

When you configure the k-means model, you must specify a target number of *centroids* (the mean points that are representative of each cluster. The K-means algorithm places data points into the specified *k* number of clusters by minimizing the within-cluster sum of squares. The K-means algorithm begins with an initial set of centroids, which are like central starting points for each cluster, and then uses Lloyd's algorithm to iteratively refine the locations of the centroids.

The algorithm stops building and refining clusters when it meets one or more of these conditions:

- The centroids stabilize, meaning that cluster assignments for individual points no longer change and the algorithm has converged on a solution.
- The algorithm completed running the specified number of iterations.

If you have new cases that you need to assign to one of the groups or clusters after completing the training phase, use the Assign Data to Clusters (https://msdn.microsoft.com/en-us/library/azure/mt484313.aspx) module that computes the distance between the new case and the centroid of each cluster and then assigns that case to the cluster with the nearest centroid.

How to Configure K-Means Clustering

1. Add the **K-Means Clustering** module to your experiment.

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

    - **Single Parameter**

      If you know the exact parameters you want to use in the clustering 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.

      You then connect the Sweep Clustering (https://msdn.microsoft.com/en-us/library/azure/mt484327.aspx) module and configure its parameters, to iterate over all possible combinations of the settings you provided and determine the combination of settings that produces the optimal clustering results.

3. Specify the number of initial centroids to create.

    The model is not guaranteed to produce exactly this number of clusters, but it starts with this number of data points and iterates to find the optimal configuration, as described in the Technical Notes section.

4. Choose the metric used to compute the distance between new data points and the randomly chosen centroid.

5. Specify an initialization method, which is used to create the initial group of clusters.

This *seeding* process can significantly affect the model. For example, if the dataset contains many outliers, and an outlier is chosen to seed the clusters, no other data points would fit well with that cluster and the cluster could be a singleton -- that is, a cluster with only one point. There are various ways to avoid this problem:

- Use a parameter sweep to change the number of centroids and try multiple seed values.
- Create multiple models, varying the metric or iterating more.
- Use a method such as PCA to find variables that have a detrimental effect on clustering. See the Find similar companies (http://gallery.azureml.net/Experiment/60cf8e46935c4fafbf86f669121a24f0) sample for a demonstration of this technique.

6. Add the Train Clustering Model (https://msdn.microsoft.com/en-us/library/azure/dn905873.aspx) or Sweep Clustering (https://msdn.microsoft.com/en-us/library/azure/mt484327.aspx) module and connect it to the training dataset with the untrained model.

   - If you set the **Create trainer mode** option to **Single Parameter**, add a tagged dataset and train the model by using the Train Clustering Model (https://msdn.microsoft.com/en-us/library/azure/dn905873.aspx) module.
   - If you set the **Create trainer mode** option to **Parameter Range**, add a tagged dataset and train the model using Sweep Clustering (https://msdn.microsoft.com/en-us/library/azure/mt484327.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.

7. Run the experiment.

   It is possible that any given configuration will result in a locally optimum cluster configuration that suits only the current data points and is not generalizable. Therefore, if you use a different initial configuration, the K-means method might find a different, perhaps superior, configuration. Therefore, you should always experiment with the parameters, creating multiple models, and compare the resulting models.

Options

You can modify the way that the algorithm creates and refines clusters by using the following 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 Clustering Model (https://msdn.microsoft.com/en-us/library/azure/dn905873.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 the Sweep Clustering (https://msdn.microsoft.com/en-us/library/azure/mt484327.aspx) module, to find the optimum configuration.

⚠ **Warning**

- If you pass a parameter range to Train Clustering Model (https://msdn.microsoft.com/en-us/library/azure/dn905873.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 Clustering (https://msdn.microsoft.com/en-us/library/azure/mt484327.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 centroids

Type the number of clusters you want the algorithm to begin with.

This does not necessarily define the number of clusters in the final model.

Range of number of centroids

If you are performing a parameter sweep, use the **Range Builder** to specify a range, or type some number of clusters to create when iterating over parameters.

This does not necessarily define the number of clusters in the final model.

Initialization and ***Initialization for sweep***

Choose the algorithm that is used to define the initial cluster configuration. If you leave this parameter blank, the module will generate points using the ***K-Means++*** method.

- ***First N***. Some initial number of data points are chosen from the data set and used as the initial means.

  Also called the Forgy method.

- **Random**. The algorithm randomly places a data point in a cluster and then computes the initial mean to be the centroid of the cluster's randomly assigned points.

  Also called the *random partition* method.

- ***K-Means++***. K-means++ improves upon K-means by using a better method for choosing the initial cluster centers

  The K-means ++ algorithm was proposed in 2007 by David Arthur and Sergei Vassilvitskii to avoid poor clustering by the standard k-means algorithm.

- ***K-Means++Fast***. A variant of the K-means++ algorithm optimized for faster clustering.

- ***Evenly***. Centroids are located equidistant from each other in the D-Dimensional space of N data points.

- ***Use label column***. The values in the label column are used to guide the selection of centroids.

Random number seed

Type a value to use as the seed for the cluster initialization. This value can have a significant effect on cluster selection.

Tip

If you use a parameter sweep on the clustering model, you can specify that multiple initial seeds be created, to look for the best initial seed value.

Number of seeds to sweep

Type the total number of random seed values to use as starting points.

This option is available only if you are using a parameter sweep to create the clustering model.

Metric

Choose the function to use when measuring the distance between cluster vectors. Azure Machine Learning supports the following metrics:

- ***Euclidean***. The Euclidean distance is commonly used as a measure of cluster scatter for K-means clustering. This metric is preferred because it minimizes the mean distance between points and the centroids.

- ***Cosine***. Alternatively, you can use the cosine function to measure cluster similarity. Cosine similarity is useful in cases where you do not care about the length of a vector, only its angle.

Iterations

Type the number of times for the algorithm to iterate over the training data before finalizing the selection of centroids.

You can adjust this parameter to balance accuracy vs. training time.

Assign label mode

Choose an option that specifies how a label column, if present in the dataset, should be handled.

Because K-means clustering is an unsupervised machine learning method, labels are optional. However, if your dataset already has a label column, you can use those values to guide selection of the clusters, or you can specify that the values be ignored.

- ***Ignore label column***. The values in the label column are ignored and are not used in building the model.
- ***Fill missing values***. The label column values are used as features to help build the clusters. If any rows are missing a label, the value is imputed by using other features.
- ***Overwrite from closest to center***. The label column values are replaced with predicted label values, using the label of the point that is closest to the current centroid.

Examples

For examples of how K-means clustering is used in Azure Machine Learning, see these experiments in the Model Gallery (http://gallery.azureml.net/):

- The Group iris data (http://gallery.azureml.net/Experiment/a7299de725a141388f373e9d74ef2f86) sample compares the results of **K-Means Clustering** and Multiclass Logistic Regression (https://msdn.microsoft.com/en-us/library/azure/dn905853.aspx) for classification,
- The Color Quantization sample (http://go.microsoft.com/fwlink/?LinkId=525272) builds multiple K-means models with different parameters to find the optimum image compression.
- The Clustering: Similar Companies (http://go.microsoft.com/fwlink/?LinkId=525164) sample uses K-means with different numbers of centroids to find groups of similar companies in the S&P500.

Technical Notes

Given a specific number of clusters (*K*) to find for a set of *D*-dimensional data points with *N* data points, the K-means algorithm builds the clusters as follows:

1. The module initializes a *K*-by-*D* array with the final centroids that define the *K* clusters found.
2. By default, the module assigns the first *K* data points in order to the *K* clusters.

3. Starting with an initial set of *K* centroids, the method uses Lloyd's algorithm to iteratively refine the locations of the centroids.

4. The algorithm terminates when the centroids stabilize or when a specified number of iterations are completed.

5. A similarity metric (by default, Euclidean distance) is used to assign each data point to the cluster that has the closest centroid.

Module Parameters

Name	**Range**	**Type**	**Default**	**Description**
Number of Centroids	>=2	Integer	2	Number of Centroids
Metric	List (subset)	Metric	Euclidean	Selected metric
Initialization	List	Centroid initialization method	K-Means++	Initialization algorithm
Iterations	>=1	Integer	100	Number of iterations

Outputs

Name	**Type**	**Description**
Untrained model	ICluster interface (https://msdn.microsoft.com/en-us/library/azure/dn906016.aspx)	Untrained K-Means clustering model

Exceptions

For a list of all exceptions, see Machine Learning REST API Error Codes (https://msdn.microsoft.com/en-us/library/azure/dn913081.aspx).

Exception	Description
Error 0003 (https://msdn.microsoft.com/en-us/library/azure/dn906003.aspx)	Exception occurs if one or more of inputs are null or empty.

See Also

Machine Learning / Initialize Model / Clustering (https://msdn.microsoft.com/en-us/library/azure/dn905908.aspx)
Assign Data to Clusters (https://msdn.microsoft.com/en-us/library/azure/mt484313.aspx)
Train Clustering Model (https://msdn.microsoft.com/en-us/library/azure/dn905873.aspx)
Sweep Clustering (https://msdn.microsoft.com/en-us/library/azure/mt484327.aspx)