

Package ‘Myrrix’

November 16, 2017

Version 1.2

Title Interface to Myrrix. Myrrix is a Complete, Real-Time, Scalable Clustering and Recommender System, Evolved from Apache Mahout

Description Recommendation engine based on 'Myrrix'. 'Myrrix' is a complete, real-time, scalable clustering and recommender system, evolved from 'Apache Mahout'. It uses Alternating Least Squares to build a recommendation engine.

Depends R (>= 2.6.0), Myrrixjars

Imports methods, rJava (>= 0.6-3)

SystemRequirements Java (>= 5.0)

License Apache License (== 2.0)

Author Jan Wijffels [aut, cre]

Maintainer Jan Wijffels <jwi.jffels@bnosac.be>

URL <https://github.com/jwijffels/Myrrix-R-interface>

Collate 'MyrrixClientConfiguration.R' 'ClientRecommender.R'
'ServerRecommender.R' 'RecommenderMethods.R'
'HyperParameters.R' 'utils.R' 'namespace_imports.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2017-11-16 13:31:07 UTC

R topics documented:

ClientRecommender-class	2
MyrrixClientConfiguration-class	2
MyrrixClientConfiguration-methods	3
RecommenderMethods-methods	4
ServerRecommender-class	7

Index	8
--------------	----------

ClientRecommender-class

Object of class ClientRecommender

Description

An implementation of MyrrixRecommender which accesses a remote Serving Layer instance over HTTP or HTTPS. This is like a local "handle" on the remote recommender.

Arguments

config An object of class MyrrixClientConfiguration

Slots

recommender: A java object of class net.myrrix.client.ClientRecommender

clientConfiguration: An object of class MyrrixClientConfiguration, which holds the java object with the connection settings to Myrrix

Examples

```
myconfig <- new("MyrrixClientConfiguration")
myconfig
recommendationengine <- new("ClientRecommender", config=myconfig)
recommendationengine
```

MyrrixClientConfiguration-class

Object of class MyrrixClientConfiguration

Description

An object of class MyrrixClientConfiguration describes how access options to the ClientRecommender.

Slots

config: A java object representing a MyrrixClientConfiguration

Examples

```
myconfig <- new("MyrrixClientConfiguration")
myconfig
```

MyrrixClientConfiguration-methods*Methods to apply on objects of class MyrrixClientConfiguration*

Description

Methods to apply on objects of class MyrrixClientConfiguration. These allow to define which Myrrix service to listen to. See the methods section of this doc for an enumerated list of function to apply.

More information on <http://myrrix.com/docs/client/java/javadoc/net/myrrix/client/MyrrixClientConfiguration.html>

Arguments

object	An object of class MyrrixClientConfiguration
host	host containing the Serving Layer, if not in distributed mode
port	port on which to access the Serving Layer, if not in distributed mode.
username	user name needed to access the Serving Layer, if any
password	password needed to access the Serving Layer, if any
allpartitionsspecification	specification for all servers that have partitions. Only applicable in distributed mode and returns null otherwise. May be specified as "auto", in which case <code>getHost()</code> and <code>getPort()</code> must be valid, since this host will be queried for partition details. Otherwise, Serving Layers are specified explicitly as "host:port" pairs. Replicas are specified as many Serving Layers, separated by commas, like "rep1:port1,rep2:port2,...". Finally, partitions are specified as multiple replicas separated by semicolon, like "part1rep1:port11,part1rep2:port12;part2rep1:port21,part2rep2:port22;...". Example: "foo:80,foo2:8080;bar:8080;baz2:80,baz3:80"
contextpath	the context path under which the target Serving Layer app is deployed (e.g. http://example.org/contextPath/...), or null if the default root context should be used.
keystorefile	the keystore file containing the server's SSL keys. Only necessary when accessing a server with a temporary self-signed key, which is not by default trusted by the Java SSL implementation
keystorepassword	password for keystorefile
secure	if true, this client is accessing the Serving Layer over HTTPS, not HTTP
...	other arguments passed on the the methods

Methods

`show(object)`: Show method for a MyrrixClientConfiguration object: Prints configuration settings, indicating which server information Myrrix talks to

`getMyrrixOptions(object)`: Returns a list of configuration settings, indicating which information Myrrix talks to

`setHost(object, host)` Sets the host of the Myrrix Serving layer
`setPort(object, port)` Sets the port on which to access the Serving Layer
`setUserName(object, userName)` Sets the user name needed to access the Serving Layer
`setPassword(object, password)` Sets the password needed to access the Serving Layer
`setContextPath(object, contextPath)` Sets the context path under which the target Serving Layer app is deployed
`setKeystoreFile(object, keystoreFile)` Sets the the keystore file containing the server's SSL keys.
`setKeystorePassword(object, keystorePassword)` Sets the the password for keystorefile
`setSecure(object, secure)` Set if this client is accessing the Serving Layer over HTTPS, not HTTP
`setAllPartitionsSpecification(object, allPartitionsSpecification)` Sets the specification for all servers that have partitions

Examples

```

myconfig <- new("MyrrixClientConfiguration")
getMyrrixOptions(myconfig)
setHost(myconfig, "myhostname")
setPort(myconfig, 20L)
  
```

RecommenderMethods-methods

Methods to apply on objects of class ClientRecommender and ServerRecommender

Description

Methods to apply on objects of class ClientRecommender and ServerRecommender. Objects of ClientRecommender and ServerRecommender behave similarly for the user. Both are classes which provide the interface with the recommendation engine of Myrrix, which is either running locally or in a distributed fashion. The methods which can be applied on this recommendation engine are `await`, `getAllItemIDs`, `getAllUserIDs`, `estimatePreference`, `mostPopularItems`, `recommend`. If Myrrix is running locally, you can set the hyperparameters of the recommendation engine which are set in java system variables and are used by Myrrix. This can be done by using the provided methods `setMyrrixHyperParameters` and `getMyrrixHyperParameters`. A full description of these hyperparamters which influence the model are listed below.

Arguments

<code>object</code>	An object of class ClientRecommender or of class ServerRecommender
<code>userID</code>	a user id for which to make the recommendation
<code>itemIDs</code>	a vector of item id's for which to make the recommendation

<code>params</code>	a list of hyperparameters to set for building the recommendation engine. Where the names of the list elements need to be part of the specified hyperparameters below. See the examples.
<code>parameters</code>	a character vector of names of hyperparameters to obtain the values. See the examples.
<code>howMany</code>	an integer indicating how many popular items you want in the call to <code>mostPopularItems</code> and <code>recommend</code>
<code>...</code>	other arguments passed on to the methods

Methods

`setMyrrixHyperParameters(list)`: Set a list of hyperparameters for building and tuning the recommendation engine

`getMyrrixHyperParameters()`: Get a list of hyperparameters which is currently used for building and tuning the recommendation engine

`getMyrrixHyperParameters(parameters)`: Get a list of hyperparameters which is currently used for building and tuning the recommendation engine, limited to the parameters specified

`await(ClientRecommender/ServerRecommender)`: Wait until the model is finished

`getAllItemIDs(ClientRecommender/ServerRecommender)`: Get all item id's known to the model

`getAllUserIDs(ClientRecommender/ServerRecommender)`: Get all user id's known to the model

`estimatePreference(ClientRecommender/ServerRecommender, userID, itemIDs)`: Score a user for different items alongside the recommendation engine

`mostPopularItems(ClientRecommender/ServerRecommender, howMany)`: Get the most popular items

`recommend(ClientRecommender/ServerRecommender, userID, howMany)`: Recommend a number of items to a specific user

Hyperparameters

`model.iterations.max`: A hard limit of the number of iterations that will run in one build. Defaults to 30.

`model.features`: Number of features to use when creating the matrix factorization. Defaults to 30.

`model.als.iterations.convergenceThreshold`: Estimated strength values in the original matrix change a little after each iteration, and less over time. If average absolute change in estimates is below this threshold, iteration will stop. Defaults to 0.001.

`model.als.lambda`: Controls the lambda overfitting parameter in the ALS algorithm. Defaults to 0.01

`model.als.alpha`: Controls the alpha parameter in the ALS algorithm. Defaults to 40

`model.noKnownItems`: If true, does not store in memory items that each user is already associated to. This saves memory, but means that the recommender does not remember which items the user is already associated to. These can't be automatically removed from consideration as recommendations. This is desirable behavior in some contexts. To use this, the `considerKnownItems` argument to `recommend` must be true. `mostPopularItems` will also not work with this flag enabled. Not recommended in general. Defaults to false.

`model.local.writesBetweenRebuild`: Sets the number of new data points written to the model that will trigger a model rebuild. Only applies to stand-alone mode. Defaults to 10000.

`model.distributed.writesBetweenUpload`: Sets the number of new data points written to the model that will trigger an upload of local data to the distributed storage system. Only applies to distributed mode. Defaults to 50000

`model.lsh.sampleRatio`: Enables locality-sensitive hashing to speed up the `/recommend` method, at the cost of accuracy. Set to a value in (0,1]; LSH is disabled unless set to a value less than 1. Recommended values are less than 0.1. This feature is experimental. Defaults to 1.0

See Also

[ClientRecommender-class](#), [ServerRecommender-class](#)

Examples

```
##
## Set Hyperparameters to tune the Myrrix recommendation engine
##
x <- getMyrrixHyperParameters()
str(x)
setMyrrixHyperParameters(
  params=list(model.iterations.max = 10, model.features=30, model.als.lambda=0.1))
x <- getMyrrixHyperParameters(
  parameters=c("model.iterations.max", "model.features", "model.als.lambda"))
str(x)
##
## Build a recommendation model locally
##
## Not run:
inputfile <- file.path(tempdir(), "audioscrobbler-data.subset.csv.gz")
download.file(
  url="http://dom2bevkhre1.cloudfront.net/audioscrobbler-data.subset.csv.gz",
  destfile = inputfile)
## Set hyperparameters
setMyrrixHyperParameters(
  params=list(model.iterations.max = 2, model.features=10, model.als.lambda=0.1))
x <- getMyrrixHyperParameters(
  parameters=c("model.iterations.max", "model.features", "model.als.lambda"))
str(x)
## Build a model which will be stored in getwd() and ingest the data file into it
recommendationengine <- new("ServerRecommender", localInputDir=getwd())
ingest(recommendationengine, inputfile)
await(recommendationengine)
## Get all users/items and score
items <- getAllItemIDs(recommendationengine)
users <- getAllUserIDs(recommendationengine)
estimatePreference(recommendationengine, userID=users[5], itemIDs=items[1:20])
mostPopularItems(recommendationengine, howMany=10L)
recommend(recommendationengine, userID=users[5], howMany=10L)

## End(Not run)
```

ServerRecommender-class

Object of class ServerRecommender

Description

The core implementation of Recommender and furthermore MyrrixRecommender that lies inside the Serving Layer.

This is the recommendation engine class.

The ServerRecommender has a local model, allowing it to build recommendation models based on data which are locally stored on your disk.

Next to the local serving, it also allows to build recommendation models based on data which is distributed on Hadoop. Special out-of-the-box classes exists to let it run on CDH, AWS and on Hadoop clusters. If you run the ServerRecommender in a distributed mode, we assume that you have set up the Computation layer already. This R package allows to ingest new data, update the model, get recommendations and similarities based on the recommendation engine which is running.

Arguments

bucket	character string with the bucket that Serving Layer is using for instances
instanceID	character string with the instance ID that the Serving Layer is serving. May be 0 for local mode.
localInputDir	character string with the local input and model file directory
partition	integer with the partition number in a partitioned distributed mode. 0 if not partitioned.
allPartitions	reference to an object that can describe all partitions; only used to get their count (see http://myrrix.com/docs/serving/javadoc/index.html)

Slots

recommender: A java object of class `net.myrrix.online.ServerRecommender`

Examples

```
recommendationengine <- new("ServerRecommender", localInputDir=tempdir())
recommendationengine
```

Index

await (RecommenderMethods-methods), 4
await, ClientRecommender-method
 (RecommenderMethods-methods), 4
await, ServerRecommender-method
 (RecommenderMethods-methods), 4
ClientRecommender-class, 2
estimatePreference
 (RecommenderMethods-methods), 4
estimatePreference, ClientRecommender, numeric, numeric-method
 (RecommenderMethods-methods), 4
estimatePreference, ServerRecommender, numeric, numeric-method
 (RecommenderMethods-methods), 4
getAllItemIDs
 (RecommenderMethods-methods), 4
getAllItemIDs, ClientRecommender-method
 (RecommenderMethods-methods), 4
getAllItemIDs, ServerRecommender-method
 (RecommenderMethods-methods), 4
getAllUserIDs
 (RecommenderMethods-methods), 4
getAllUserIDs, ClientRecommender-method
 (RecommenderMethods-methods), 4
getAllUserIDs, ServerRecommender-method
 (RecommenderMethods-methods), 4
getMyrrixHyperParameters
 (RecommenderMethods-methods), 4
getMyrrixHyperParameters, character-method
 (RecommenderMethods-methods), 4
getMyrrixHyperParameters, missing-method
 (RecommenderMethods-methods), 4
getMyrrixOptions
 (MyrrixClientConfiguration-methods),
 3
getMyrrixOptions, MyrrixClientConfiguration-method
 (MyrrixClientConfiguration-methods),
 3
ingest (RecommenderMethods-methods), 4
ingest, ClientRecommender, character-method
 (RecommenderMethods-methods), 4
ingest, ServerRecommender, character-method
 (RecommenderMethods-methods), 4
mostPopularItems
 (RecommenderMethods-methods), 4
mostPopularItems, ClientRecommender, integer-method
 (RecommenderMethods-methods), 4
mostPopularItems, ServerRecommender, integer-method
 (RecommenderMethods-methods), 4
MyrrixClientConfiguration-class, 2
MyrrixClientConfiguration-methods, 3
recommend (RecommenderMethods-methods),
 4
recommend, ClientRecommender, numeric, integer-method
 (RecommenderMethods-methods), 4
recommend, ServerRecommender, numeric, integer-method
 (RecommenderMethods-methods), 4
RecommenderMethods-methods, 4
ServerRecommender-class, 7
setAllPartitionsSpecification
 (MyrrixClientConfiguration-methods),
 3
setAllPartitionsSpecification, MyrrixClientConfiguration, ch
 (MyrrixClientConfiguration-methods),
 3
setContextPath
 (MyrrixClientConfiguration-methods),
 3
setContextPath, MyrrixClientConfiguration, character-method
 (MyrrixClientConfiguration-methods),
 3
setHost
 (MyrrixClientConfiguration-methods),
 3
setHost, MyrrixClientConfiguration, character-method
 (MyrrixClientConfiguration-methods),
 3

setKeystoreFile
(MyrrixClientConfiguration-methods),
3

setKeystoreFile,MyrrixClientConfiguration,character-method
(MyrrixClientConfiguration-methods),
3

setKeystorePassword
(MyrrixClientConfiguration-methods),
3

setKeystorePassword,MyrrixClientConfiguration,character-method
(MyrrixClientConfiguration-methods),
3

setMyrrixHyperParameters
(RecommenderMethods-methods), 4

setMyrrixHyperParameters,list-method
(RecommenderMethods-methods), 4

setPassword
(MyrrixClientConfiguration-methods),
3

setPassword,MyrrixClientConfiguration,character-method
(MyrrixClientConfiguration-methods),
3

setPort
(MyrrixClientConfiguration-methods),
3

setPort,MyrrixClientConfiguration,integer-method
(MyrrixClientConfiguration-methods),
3

setSecure
(MyrrixClientConfiguration-methods),
3

setSecure,MyrrixClientConfiguration,logical-method
(MyrrixClientConfiguration-methods),
3

setUserName
(MyrrixClientConfiguration-methods),
3

setUserName,MyrrixClientConfiguration,character-method
(MyrrixClientConfiguration-methods),
3