Major functions of the spark als api

CONFIDENTIAL - NOT FOR USE AND/OR DISTRIBUTION TO THE GENERAL PUBLIC

HELLO!

We are Group 5

Cathy

Jiayue

Michael

Xinyi

1.

PYSPARK.MLLIB.RECOMMENDATION MODULE

1 MatrixFactorizationModel

class pyspark.mllib.recommendation.MatrixFactorizationModel

A matrix factorisation model trained by regularized alternating least-squares.

1 MatrixFactorizationModel

- ✘ classmethod load(sc, path)
- ✘ predict(user, product)
- ✘ predictAll(user_product)
- ✘ productFeatures( )
- ✘ userFeatures( )
- ✘ *property* rank
- ✘ recommendProducts(user, num)
- ✘ recommendProductsForUsers(num)
- ✘ recommendUsers(product, num)
- ✘ recommendUsersForProducts(num)

2 ALS

class pyspark.mllib.recommendation.ALS

Alternating Least Squares matrix factorization

2 ALS

Parameters

- ratings – RDD of Rating or (userID, productID, rating) tuple.
- rank – Number of features to use (also referred to as the number of latent factors).
- iterations – Number of iterations of ALS. (default: 5)
- lambda – Regularization parameter. (default: 0.01)
- blocks – Number of blocks used to parallelize the computation. A value of -1 will use an auto-configured number of blocks. (default: -1)
- nonnegative – A value of True will solve least-squares with nonnegativity constraints. (default: False)
- seed – Random seed for initial matrix factorization model. A value of None will use system time as the seed. (default: None)

2 ALS

- ✘ classmethod train(ratings, rank, iterations=5, lambda_=0.01, blocks=-1, nonnegative=False, seed=None)

  Train a matrix factorization model given an RDD of ratings by users for a subset of products. The ratings matrix is approximated as the product of two lower-rank matrices of a given rank (number of features). To solve for these features, ALS is run iteratively with a configurable level of parallelism.

2 ALS

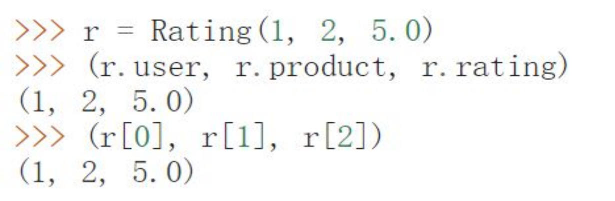
- ✘ classmethod trainImplicit(ratings, rank, iterations=5, lambda_=0.01, blocks=-1, alpha=0.01, nonnegative=False, seed=None)

  Train a matrix factorization model given an RDD of ‘implicit preferences’ of users for a subset of products. The ratings matrix is approximated as the product of two lower-rank matrices of a given rank (number of features). To solve for these features, ALS is run iteratively with a configurable level of parallelism.

3 Rating

class pyspark.mllib.recommendation.Rating

Represents a (user, product, rating) tuple.



```
>>> r = Rating(1, 2, 5.0)
>>> (r.user, r.product, r.rating)
(1, 2, 5.0)
>>> (r[0], r[1], r[2])
(1, 2, 5.0)
```

“

THANK YOU!

ANY QUESTION?