Matrix Factorization using ALS

NAME

Matrix Factorization using ALS - A matrix factorization algorithm commonly used for recommender systems.

SYNOPSIS

`#include <frovedis/ml/recommendation/als.hpp>`

`matrix_factorization_model<T>`
matrix_factorization_using_als::train (`crs_matrix<T>`& data,
    size_t factor,
    int numIter = 100,
    T alpha = 0.01,
    T regParam = 0.01,
    size_t seed = 0)

DESCRIPTION

Collaborative filtering is commonly used for recommender systems. These techniques aim to fill in the missing entries of a user-item association matrix. Frovedis currently supports model-based collaborative filtering, in which users and products are described by a small set of latent factors that can be used to predict missing entries. Frovedis uses the alternating least squares (ALS) algorithm to learn these latent factors. The algorithm is based on a paper "Collaborative Filtering for Implicit Feedback Datasets" by Hu, et al.

Detailed Description

matrix_factorization_using_als::train()

Parameters
data: A `crs_matrix<T>` containing the sparse rating matrix
factor: A size_t parameter containing the number of latent factors (also known as rank)
numIter: A size_t parameter containing the maximum number of iteration count (Default: 100)
alpha: A parameter of T type containing the learning rate (Default: 0.01)
regParam: A parameter of T type containing the regularization parameter (also known as lambda) (Default: 0.01)
seed: A size_t parameter containing the seed value to initialize the model structures with random values (Default: 0)

Purpose
It trains a matrix factorization model with alternating least squares (ALS) algorithm. It starts with initializing

the model structures of the size MxF and NxF (where MxN is the dimension of the input rating matrix and F is the latent factors count) with random values and keeps updating them until maximum iteration count is reached. After the training, it returns the trained output model.

Return Value
After the successful training, it returns a trained model of the type `matrix_factorization_model<T>` which can be used for predicting user choices or making recommendation.

SEE ALSO

matrix_factorization_model