CSE472 (Machine Learning Sessional)

Assignment 3: Matrix Factorization for Recommender System

Introduction

In this assignment, you'll be building a recommendation system to make predictions. You'll be given data that comprises Users, Items, and Ratings. We'll focus on the User-Item utility matrix and build an alternating least square (ALS) based recommendation systems.

Dataset

You are given a comma separated file titled "data.csv" to be used as the dataset for this assignment. The file contains 24983 rows and 101 columns. First column indicates the number of rated products by the user corresponding to row. Next 100 columns indicate the rating given by the user of that row for 100 items. A 99 indicates no rating.

Use the following guidelines.

1. Taking ratings data from the given data set, build an ALS model with a small number of latent factors, between 10-50 factors.
2. We strongly recommend that you first try your code on a smaller dataset.
3. Split the data set into 60-20-20 train-validate-test partitions. That is, the first 60% of the data is the training set. The next 20% is for validation and the remaining 20% is for test. You'll use the training set to learn your ALS model and use the validation set to choose the regularization parameter and the number of latent factors. Your splits will randomly select ratings so that every portion of your matrix is sampled uniformly.
4. You'll evaluate these systems via RMSE (root mean square error) metrics on Validation and Test sets. Make sure you try different regularization parameters $\lambda \in (0.01, 0.1, 1.0, 10.0)$ and several latent factor dimensions $K \in (5, 10, 20, 40)$ and select the model that gives you the best RMSE on the validation set.
5. Once you have finished choosing your model using the validation set, write a simple recommendation engine that will take the ALS model and test it on the test set and report that error as your final error metric.
6. You can use any data structure library for sparse matrix representation and any linear algebra library for matrix inversion.

ALS implementation

1. Initialize the latent factors $\mathbf{u}_1^T, \mathbf{u}_2^T, \dots, \mathbf{u}_N^T$ randomly
2. Iterate until converge
   a. Update each column latent factor $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_M$ (In parallel)

$$\mathbf{v}_m^* = \left(\sum_{n \in \Omega_{cm}} \mathbf{u}_n \mathbf{u}_n^T + \lambda_v \mathbf{I}_K \right)^{-1} \sum_{n \in \Omega_{cm}} x_{n,m} \mathbf{u}_n$$

   b. Update each row latent factor $\mathbf{u}_1^T, \mathbf{u}_2^T, \dots, \mathbf{u}_N^T$ (In parallel)

$$\mathbf{u}_n^* = \left(\sum_{m \in \Omega_{r_n}} \mathbf{v}_m \mathbf{v}_m^T + \lambda_u \mathbf{I}_K \right)^{-1} \sum_{m \in \Omega_{r_n}} x_{n,m} \mathbf{v}_m$$

c. Calculate empirical risk

Submission

1. Upload the codes in Moodle within **9:00 P.M. of 6th January, 2019 (Sunday)**. (Strict deadline)
2. You need to submit a report file in pdf format containing the tables shown in the performance evaluation section with your experimental results. No hardcopy is required.
3. Write code in a single *.py file, then rename it with your student id. For example, if your student id is 1405123, then your code file name should be "1405123.py" and the report name should be "1405123.pdf".
4. Finally make a main folder, put the code and report in it, and rename the main folder as your student id. Then zip it and upload it.

Evaluation

1. You have to reproduce your experiments during in-lab evaluation. Keep everything ready to minimize delay.
2. You are likely to give online tasks during evaluation which will require you to modify your code.
3. You will be tested on your understanding through viva-voce.
4. If evaluators like performance, efficiency or modularity of a particular code, they can give bonus marks. This will be completely at the discretion of evaluators.
5. You are encouraged to bring your computer in the sessional to avoid any hassle. But in that case, ensure an internet connection as you have to instantly download your code from the Moodle and show it.

Warning

1. Don't copy! We regularly use copy checkers.
2. First time copier and copyee will receive **negative** marking because of dishonesty. Their default is bigger than those who will not submit.
3. Repeated occurrence will lead severe departmental action and jeopardize your academic career. We expect Fairness and honesty from you. Don't disappoint us!