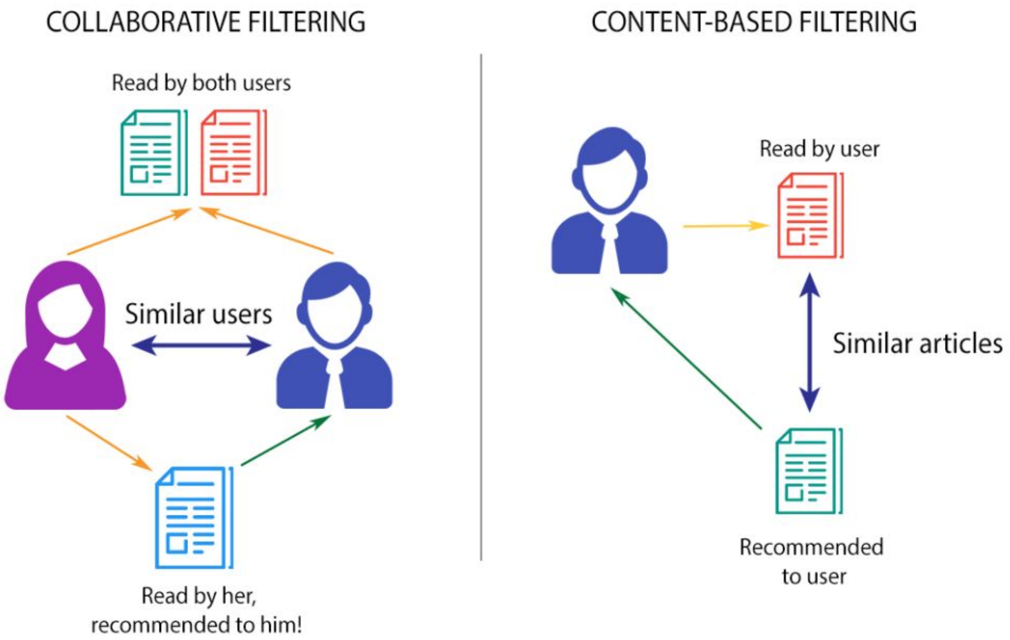
Northeastern

EECE 5698 Project:
Movie Recommender System using Collaborative Filtering

Emmanuel Ojuba
Iuliia Klykova

Types of recommender systems

- Demographic Filtering
- Content Based filtering
- Collaborative Filtering
- Hybrid Filtering



Northeastern

Dataset

The Movies Dataset from Kaggle.com:

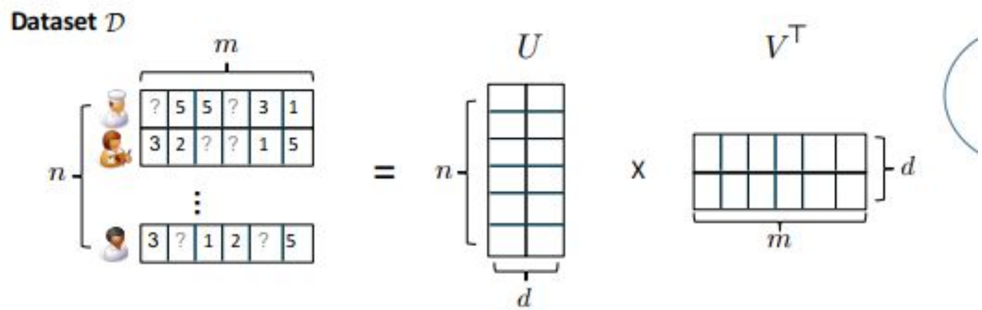
- The dataset contains: movies_metadata.csv, keywords.csv, credits.csv, links.csv, links_small.csv, ratings.csv, ratings_small.csv:

- **Ratings.csv** file that consists of 26 million ratings from 270,000 users for all 45,000 movies. Ratings are on a scale of 1-5.

- We picked 450,000 random samples and randomly divided them into 5 folds.
    - Fold 0: 121650 users and 8611 items.
    - Fold 1: 121573 users and 8613 items.
    - Fold 3: 121646 users and 8578 items.
    - Fold 3: 121695 users and 8610 items.
    - Fold 4: 121693 users and 8593 items

-
-

Collaborative Filtering (Matrix Factorization)



$$R \approx U \times V^{\top}$$

r_{ij} : rating by user i to item j.

$$r_{ij} = \langle u_i, v_j \rangle + \varepsilon_{ij}$$, where $u_i \in \mathbb{R}^d$, $v_j \in \mathbb{R}^d$.

user profile   item profile

$$\mathrm{RSE}(U, V) = \sum_{(i,j,r_{ij}) \in \mathcal{D}} (u_i^{\top} v_j - r_{ij})^2 + \lambda \sum_{i=1}^{n} \|u_i\|_2^2 + \mu \sum_{j=1}^{n} \|v_j\|_2^2,$$

Assumption: Bilinear relationship between user profiles, ui, and item profiles vj

Goal: To learn ui and vj, given a sparse set of ratings

Objective Function: Least-Square Error with Parameter Regularization

Optimization: Stochastic Gradient Descent; Alternating Least Squares

Alternating Least Squares

$$\text{RSE}(U,V) = \sum_{(i,j,r_{ij})\in\mathcal{D}} (u_i^\top v_j - r_{ij})^2 + \lambda \sum_{i=1}^{n} \|u_i\|_2^2 + \mu \sum_{j=1}^{n} \|v_j\|_2^2,$$

We alternate between fixing $U \ni \{u_1, u_2, \dots, u_n\}$ and $V \ni \{v_1, v_2, \dots, v_n\}$

With either U or V fixed, the objective becomes convex least-squares estimation problem that can be optimized using the an analytic formulation:

$$v_j = \left[\sum_i u_i u_i^T + \mu I\right]^{-1} \left[\sum_i r_{ij} u_i\right]$$

This process is allowed to continue for a set number of iterations.

We utilize Spark's MLLib's implementation which offers these hyperparameters:
- rank - Iterations - numBlocks - nonNegative - lambda

Stochastic Gradient Descent

1. Randomly initialize the U and V matrices
2. Predict the ratings for the initialized U and V matrices
3. Update U and V iteratively until convergence using a stochastic estimate of the gradient, obtained by calculating the gradient over a subsampled set of the data:

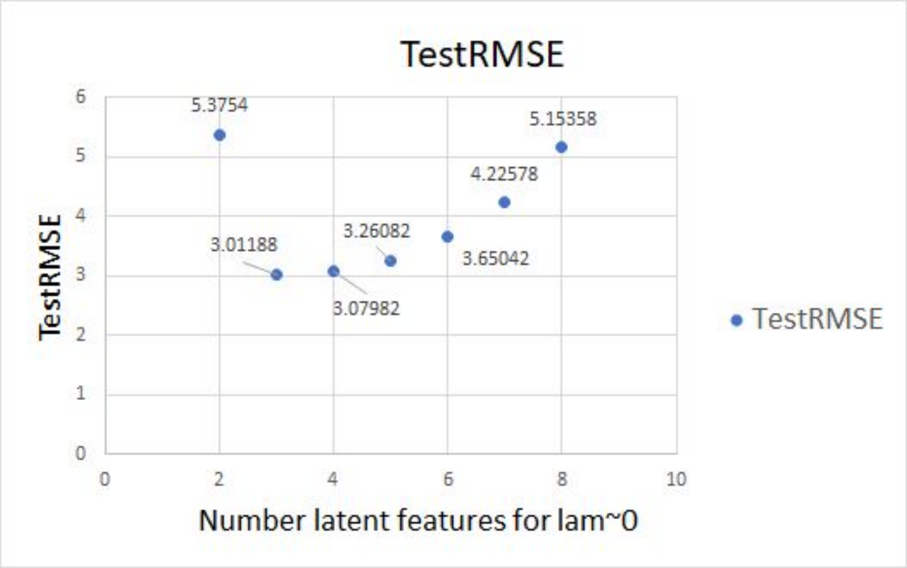
$$\nabla_{u_1} \widetilde{RSE} = 2 \sum_{v_j, subsampled} \delta_{ij} v_j + 2\lambda u_i$$

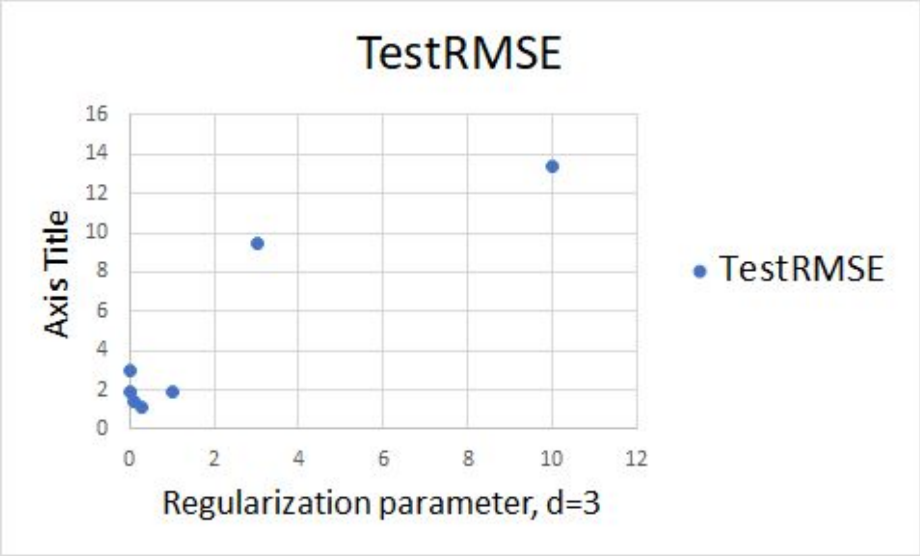
$$\nabla_{v_j} \widetilde{RSE} = 2 \sum_{u_i, subsampled} \delta_{ij} u_i + 2\mu v_j$$

$$u^{k+1} = u^k - \gamma \nabla_u \widetilde{RSE}(U,V)$$

$$v^{k+1} = v^k - \gamma \nabla_v \widetilde{RSE}(U,V)$$

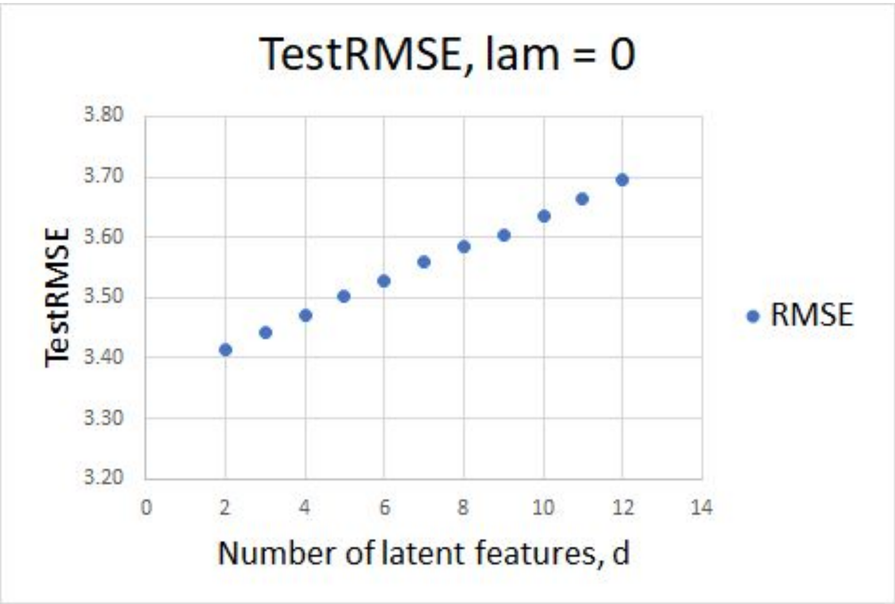
Results: Alternating Least Squares (ALS)

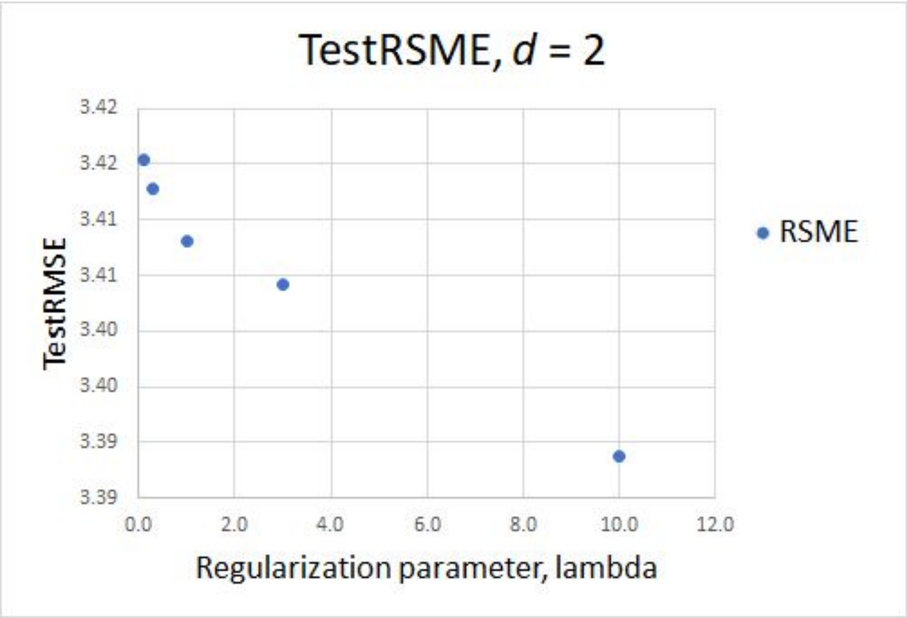




Number of latent features *d* = 2
Regularization parameters λ = μ=0.3
The smallest *TestRMSE*=1.1248

Results: Stochastic Gradient Descent





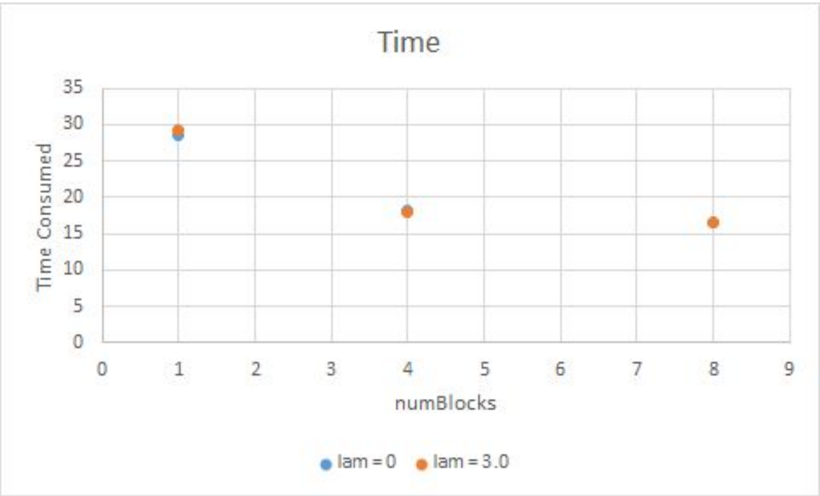
Number of latent features d = 3

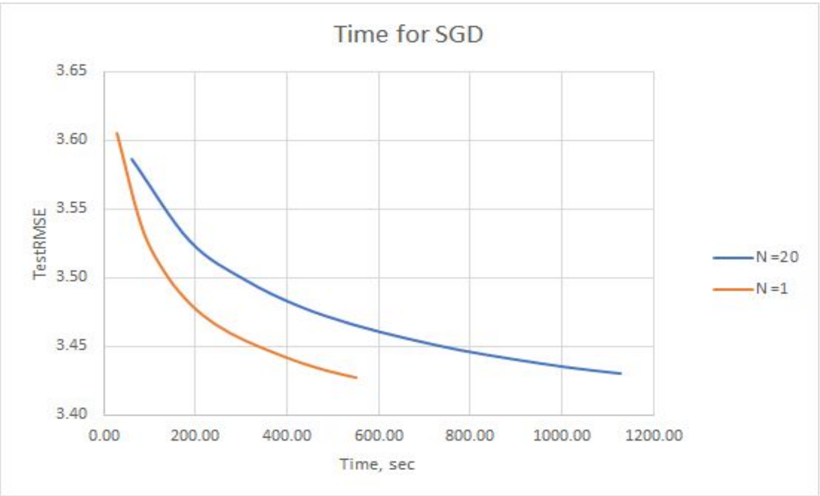
Regularization parameters $\lambda = \mu$=10.0

The smallest *TestRMSE*=3.388762

Northeastern

Parallelism





ALS

- almost 2x speed up between 1 and 8 numbers of blocks used to parallelize computation

SGD

- no speed up observed

Conclusions

- Best performing model is the ALS-optimized MF with parameters of $d = 2$, $\lambda = \mu = 0.3$. This yielded and RMSE of 1.125
- Unexpectedly, there is huge discrepancy between the best performing ALS and SGD models, with SGD yielding RMSE of 3.389
- Parallelism yields a speedup in the ALS computation, although not in our implementation of SGD.

Future Work

- Hybrid Approaches
- Deep Learning Methods

References

[1] - Ahmed I. Getting Started with a Movie Recommendation System. https://www.kaggle.com/ibtesama/getting-started-with-a-movie-recommendation-system/notebook

[2] - The Movies Dataset https://www.kaggle.com/rounakbanik/the-movies-dataset

[3] - Koren Y., Bell R., Volinsky C. Matrix Factorization Techniques for Recommender Systems

[4] - Ioannidis, E. Homework 4 Handout

[5] - Matrix Completion via Alternating Least Squares - http://stanford.edu/~rezab/classes/cme323/S15/notes/lec14.pdf

[6] - Various Implementations of Collaborative Filtering - https://towardsdatascience.com/various-implementations-of-collaborative-filtering-100385c6dfe0