Movie Recommendation

by Grant Doan, Prayash Joshi, Reagan Orth, Ved Patel

Problem to Solve

Premise:

Movie recommendations on popular websites do not give deeply personalized recommendations

Problem:

a. Recommendations are often centered around one movie
b. In other cases, recommendations are centered around search history, not preference
c. Popular movies are far more highly considered
    i. *Sunset Blvd.* and *The Godfather* are often recommended together
d. Users are never asked for preferences
e. Recommendations cannot reliably improve

Current State-of-the-Art (Include Papers)

Collaborative filtering - consider n similar users
- Memory based vs model based
- Scaling issues, Sparse preferences, Privacy concerns
- Current Literature
  - Bo Yang Et. al deploying Trust Matrix Factorization creating Taxonomy between users in a network

Content-Based Filtering - consider n similar items
- Alleviates scaling issues by viewing data in a vacuum
- Is overspecialized due to detection of products that strongly resemble what the user prefers
- Current Literature
  - By Jieun Son and Seoung Bum Kim proposing multiattribute networks

Hybrid Filtering
- Combines the 2
- Different examples of 2 part filtering methods :
  - Yang et al considering social similarities and item attributes through Matrix Factorization and Personalized Bayesian Ranking models for personalized evaluations and TF-IDF for content similarities

Analysis of SoA methods per use case

Note:

- Popularity of Cosine and Pearson Similarity for non-NNs
- The only Hybrid made use of separate NNs for each type of filtering type

Table 1: Comparison of State-of-the-Art methods

Ref	CF/CB/H	Dataset	Method	Similarity Measures	Performance
[50]	H	Movie lens	PCA	Cosine Similarity	Cutoff = 0.5 will leave some useful tags Cutoff = 0.3 will lose some important tags
[51]	H	Movielens-1k	HFB-KNN, CFB-KNN, User KNN		Rating changes from 10 to 80 HFB-KNN, CFB-KNN performs better
[22]	CF	From, Yahoo Research Web scope database	NNUN, Log Likely hood	Pearson Correlation	AAD =0 .Prediction Accuracy = 100%
[23]	CF	Movie lens	PNN, SVD		The sparsity level of the Movie Lens dataset is 93.7% the sparsity level of the Yahoo! Web scope R4 dataset is 99.8%
[24]	CF	Movielens-10M	User Based CF, Item Based CF		HDFS makes the performance better.
[52]	H	Movie lens	BPR-MF,TF-IDF	Cosine similarity	MAE = 0.817 , RMSE= 1.037
[53]	H	Movielens 100k,Film Trust	Biased Matrix Factorization		For Movie Lens, Precision = 0.8704 Recall =0.107 For Film Trust, Precision= 0.7391 Recall = 0.0051
[54]	H	Movie lens	Map Reduce	Cosine Similarity, Centered Cosine Similarity	Using Hadoop Map Reduce Framework, movie recommendation is done fast though dataset is large.
[18]	CB	Movie lens -1M	CB-MN system, FW		Overspecialization and data sparsity is improved
[19]	CB	IMDB and BBC web server	SA		Overspecialization problem is solved and observed that when no. of cluster increases accuracy decreases.
[34]	User based CF	Movie lens with 6.3% rating available	K-means, SOM, Fuzzy clustering	Pearson Correlation	Fuzzy C-means and max average accuracy-80.44 and Fuzzy C-means and max Pearson accuracy-81.1
[35]	User based CF	Movielens-100K	Neo4j is NoSQL graph database	Euclidean distance	Larger radius of nodes and thicker edge then movie score is high
[37]	H	Movie lens	UBCF and IBCF	Pearson Correlation	Improvement of 13% over UBCF and 16% over IBCF
[44]	CF	Movie lens	EMF, NMF and PMF	Cosine	Average RMSE of EMF is 1.3411
[48]	CF	Movie lens	SVD with UBCF and IBCF	Demographic correlation	IBCF produces less error value than UBCF

Limitations of Current Models

Cold Start:

- Initially, websites like IMDB have no prior data on a user
- This makes recommendations difficult

Data Sparsity:

- Difficult to find users that have rated the same movies

Scalability:

- With the large amount of data being used on popular websites, inaccurate results are inevitable

Filter Bubble:

- Users are often diverged from having different perspectives

Our Approach

- Web scrape dataset using IMDb
  - Take relevant information like name, year, director, as well as reviews
- NLP preprocessing
  - Clean up punctuation
  - Sparse Matrix
  - DF/TDF to remove words that don't matter
    - Names mentioned in the reviews should be given special importance
- Long Short-Term Memory(LSTM) Neural Network
  - Extension of Recurrent Neural Networks (RNNs)
    - Better on non sequential data, considering old data
    - Helps with vanishing/exploding gradient issues
  - Input, Output, Forget gates
- Graph Neural Network
  - Great for modeling relationships

Benefits and Drawbacks

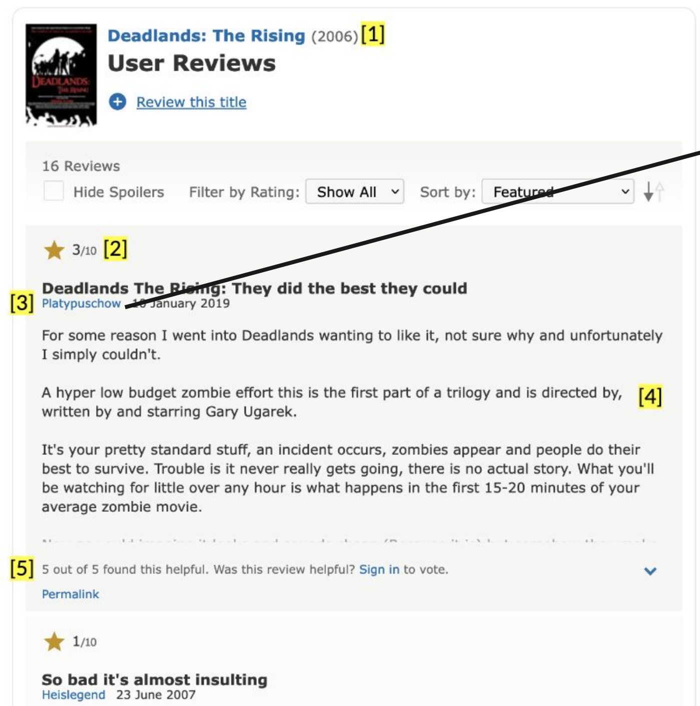
- LSTM shown to be successful in recommendation problems
  - \+ Ideally keeps and uses all relevant information
  - \- Can be very computationally expensive
  - \- Affected by random weight initialization
  - \- Prone to overfitting

- GNN also shown to be successful
  - \+ Better for showing relationships
  - \- Usually difficult to add new nodes/edges
  - \- Extremely computationally expensive

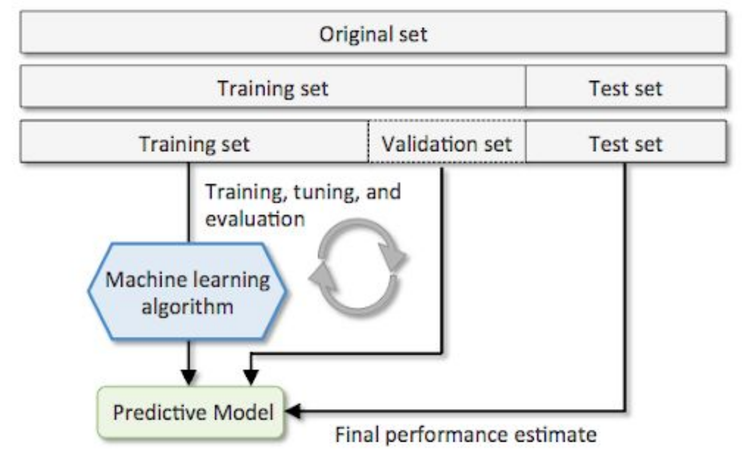
Our Data

review/

....
http://www.imdb.com/title/tt0406816/
http://www.imdb.com/title/tt0406816/
http://www.imdb.com/title/tt0406816/
http://www.imdb.com/title/tt0406816/
http://www.imdb.com/title/tt0085461/
http://www.imdb.com/title/tt0065611/
http://www.imdb.com/title/tt0065611/
http://www.imdb.com/title/tt0065611/
....



Testing

Midterm Milestone → Evaluating Classifications	Final Milestone → Evaluating Recommendations
Holdout set evaluation: 1. After web scraping and preprocessing → split training and testing sets. 2. Use LSTM Neural Network to train the model on training set. 3. Use training set to evaluate how we can classify the user ratings(including comment) 4. Evaluate accuracy, precision/recall metrics 5. Use results of the model to make changes for the next test. 	Option 1: Cold Start Problem Testing 1. Generate a subset of users with limited movie ratings 2. Assess the recommendation performance on original set 3. Evaluate what the appropriate filtering technique would be (Content-based vs. Collaborative) 4. Preform metric evaluations and tuning until happy with results. Option 2: Offline A/B Testing 1. Divide into two parts, control and test subsets. 2. Control subset → existing model, test subset → modified model 3. Evaluate how well it recommends a movie the user might like 4. Use various statistical tests(p-value, confidence interval, etc.) to check if modified model made a difference

Discussion

Any Questions?

References

Reddy, SRS, et al. “Content-Based Movie Recommendation System Using Genre Correlation.” *Smart Intelligent Computing and Applications*, 2018, pp. 391–397., https://doi.org/10.1007/978-981-13-1927-3_42.

Yang, Chen, et al. “A Hybrid Movie Recommendation Method Based on Social Similarity and Item Attributes.” *Lecture Notes in Computer Science*, 2018, pp. 275–285., https://doi.org/10.1007/978-3-319-93818-9_26.

Son, Jieun, and Seoung Bum Kim. “Content-Based Filtering for Recommendation Systems Using Multiattribute Networks.” *Expert Systems with Applications*, vol. 89, 2017, pp. 404–412., https://doi.org/10.1016/j.eswa.2017.08.008.

Sunilkumar, Chaurasia Neha. “A Review of Movie Recommendation System: Limitations, Survey and Challenges.” *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, vol. 19, no. 3, 2020, p. 18., https://doi.org/10.5565/rev/elcvia.1232.

Steffan Rendle, Li Zhang ,and Yehunda Koran. “On the Difficulty of Evaluating Baselines: A Study on Recommender Systems.” *Cornell University*. https://doi.org/10.5565/rev/elcvia.1232.

KumarI, Ajitesh. “Hold-out Method for Training Machine Learning Models.” *Data Analytics*, 27 Apr. 2022, https://vitalflux.com/hold-out-method-for-training-machine-learning-model/.