Machine Learning Engineer Nanodegree

Capstone Proposal

Kanja Saha
December 15th, 2017

Proposal

_Build a Product Recommendation Engine with item to item Collaborative Filtering technique using Matrix Factorization.

Domain Background

Some say customer is God and others say customer is King. They say, "Listen to your customer!". I say "Know thy customer!".

A company exists because of its customers; to be precise, its loyal customers. A loyal customer is a satisfied customer whose trust you have earned. And the word "trust" carries a lot of weight and is only earned through consistent positive value of product, service and experience and show that you "know your customer" . To reflect that you indeed know your customer, it is imperative that you recommend only the products that you believe will benefit the customer.

Hence a product recommendation engine is crucial for every company's success. In fact it is necessary for success of every department of the company. A recommender system is already in place for large market place such as Amazon, Google and other organizations, small or large, are inspired to build a near accurate recommender system. As I got introduced to Deep Learning in Udacity's Machine Learning nanodegree program, I think Deep learning will be able to help up build the most accurate recommender engine. Deep learning involves utilizing multiple-layered neural networks, which use mathematical properties to minimize losses from predictions vs. actuals to converge toward a final model, effectively learning as they train on data.

Problem Statement

With the advent of internet commerce, it has become convenient for the users to find the items of their interest without stepping out of their house. However with the consistent increase of number of online retailers it is also very difficult for the user to find the items that really meets their need the best. Users feel lost in the sea of products available to them and often fear of making a wrong purchasing decision. And once they make a purchase that is not ideal, customers shy away from making further online purchase investment or recommending others. This is a no win situation for the user as well as the company selling the product

Recommendation system is considered a semi supervised learning or a combination of supervised(ranking a recommended item) and unsupervised learning(forming clusters of similar groups of customers/items). But overall it is an information retrieval system, which is another large area of machine learning.

Datasets and Inputs

I plan to use Amazon Product Ratings Only Dataset. These datasets include no metadata or reviews, but only (user,item,rating,timestamp) tuples. Following are the column details in the dataset.

- reviewerID - ID of the reviewer, e.g. A2SUAM1J3GNN3B
- asin - ID of the product, e.g. 0000013714
- overall: rating of the product,
- reviewTime - time of the review (raw)

Sample Ratings Only Data: { "reviewerID": "A2SUAM1J3GNN3B", "asin": "0000013714", "overall": 5.0, "reviewTime": "09 13, 2009" }

Source for the data: http://jmcauley.ucsd.edu/data/amazon/links.html (http://jmcauley.ucsd.edu/data/amazon/links.html)

Solution Statement

To make a buyer feel confident about making frequent purchase decision, there is a need for a system which learns the user preferences, spending pattern and generate recommendations based on his interest and past buying habit, The Recommender System. I believe, one of the best ways to build the recommendation engine is by using Collaborative Filtering technique. Collaborative filtering is the technique of recommending items to users based on past interactions between users and items.

So, my goal is to build a recommendation system with item to item Collaborative Filtering Technique. I will implement the Matrix Factorization algorithms in sklearn Library: Non-Negative Matrix Factorization(NMF)

Benchmark Model

I will implement K-nearest neighbors as benchmark model and compare the results of the Non-Negative Matrix Factorization (NMF).

Evaluation Metrics

I will use Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) as the evaluation metrics.

Mean Absolute Error (MAE): MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It's the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.

Root mean squared error (RMSE): RMSE is a quadratic scoring rule that also measures the average magnitude of the error. It's the square root of the average of squared differences between prediction and actual observation.

Reference: https://en.wikipedia.org/wiki/Mean_absolute_error (https://en.wikipedia.org/wiki/Mean_absolute_error) https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d (https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d)

Project Design

- Programming Language: Python 3.6
- Libraries : Pandas,Numpy, Scikit-learn
- Goal: Implement NMF in sklearn to build a recommender system using item to item collaborative filtering technique
- Workflow :
  - Data Pre Processing:
    - Take the first three columns of the dataframe: reviewerID, asin, overall for my project.
    - Pivot the dataframe to give me "reviewerID" as Index, "asin" as Column and "overall" as value.
    - For KNN, I will transpose the above matrix to have "asin" as row and "reviewerID as Columns
    - Split the dataset into train and test set.
  - Baseline Model: Implement K-nearest neighbors using cosine similarities.

    - Build the Item to Item similarity matrix implementing sklearn.neighbors.NearestNeighbors
    - Generate Prediction Rating
    - Compare the True Rating with the predicted rating and find the MAE and RMSE values
- Implement NMF(Non Negative Matrix Factorisation).
    - Implement sklearn.decomposition.NMF with optimum n_components.
    - Generate the dot product of the two factored matrix to get the predicted ratings
    - Generate the optimum n_components iteritavely by minimizing the loss function.(MAE and RMSE values)
- Implement KNN, NMF on test dataset and compare MAE and RMSE score for KNN, NMF.

Reference:

https://datascienceplus.com/building-a-book-recommender-system-the-basics-knn-and-matrix-factorization/ (https://datascienceplus.com/building-a-book-recommender-system-the-basics-knn-and-matrix-factorization/)
https://blog.insightdatascience.com/explicit-matrix-factorization-als-sgd-and-all-that-jazz-b00e4d9b21ea (https://blog.insightdatascience.com/explicit-matrix-factorization-als-sgd-and-all-that-jazz-b00e4d9b21ea)
https://github.com/ArthurFortes/CaseRecommender/blob/master/caserec/recommenders/rating_prediction/itemknn (https://github.com/ArthurFortes/CaseRecommender/blob/master/caserec/recommenders/rating_prediction/itemkn
http://www.benfrederickson.com/matrix-factorization/ (http://www.benfrederickson.com/matrix-factorization/)
http://www.quuxlabs.com/blog/2010/09/matrix-factorization-a-simple-tutorial-and-implementation-in-python/ (http://www.quuxlabs.com/blog/2010/09/matrix-factorization-a-simple-tutorial-and-implementation-in-python/)