Industry Paper Presentation On
Amazon.com recommendations

By Authors: G. Linden; B. Smith; J. York

Publisher: IEEE

by

Group 9:

Yogita Suryavanshi

Saroj Saran

Cheuk Hong Ip

Preeti Khatri

Hrushikesh Pokala

San Jose State University

Abstract

Product Recommendation plays a key role to boost the online sale of products on e-commerce websites. certain recommendations algorithms are used to efficiently recommend the products which might interest the customers based upon their purchase history and their ratings and reviews expressing their interests, including items viewed, subject interests, and favorite artists.

Amazon.com uses one such recommendation algorithm to personalize the online store for each customer. The store radically changes based on customer interests, Thus highlighting the products which might interest them based on their search pattern like showing programming titles to a software engineer and baby toys or related products to a new mother.

Several challenges to implementing the recommendation algorithms include: millions of customers contributing to huge data, real-time analysis for a high-quality recommendation, new customers have limited history and recommendation suggestions based on the limited available data, old customers have huge data history and browsing through all the data to better recommend as per the current interest.

Three common approaches to solving the recommendation problem involve traditional collaborative filtering, cluster models, and search-based methods, we will compare Amazon's Item to item collaborative algorithm with these methods. This algorithm is efficient enough to handle the number of customers and the number of items in the product catalog. Thus producing recommendations in real-time, scales to massive data sets, and generates high-quality recommendations

Recommendation Algorithms

Recommendation algorithms mostly start with :

- Finding a set of customers whose purchased and rated items overlap the user's purchased and rated items.
- The algorithm then aggregates items from these similar customers, eliminates items the user has already purchased or rated
- Then recommends the remaining items to the user.
- Some algorithms might focus on finding similar items, not similar customers. It then aggregates similar items and recommends them.

Traditional Collaborative Filtering:

Traditional collaborative filtering algorithms represent a customer in the form of an N-dimensional vector of items, where N is the Number of items in the catalog items. The components which are positively rated or purchased are positive and the components which are negatively rated or not purchased are filled as negative.

The algorithm then uses the filtering criteria for recommendations to list down the items vector component values to recommend the current customer, hence it uses the similarity between the current customer and other customer's details from the database for a recommendation.

There are several ways to do it:
a) A common method is to measure the cosine of the angle between the two vectors:

$$similarity(\vec{A}, \vec{B}) = \cos(\vec{A}, \vec{B}) = \frac{\vec{A} \bullet \vec{B}}{\|\vec{A}\| * \|\vec{B}\|}$$

b)Another common technique is to rank each item according to how many similar customers purchased it.

This recommendation algorithm gets expensive at times, because of the complexity of calculations. In the worst case, the Big O is O(MN), if the algorithm examines M number of customers with N number of items.

Since most customers will not buy all the products, and all products are not owned by all customers, hence the customer vector will be sparse. Hence the complexity will fall to O(M+N) scanning the customers for the products they bought. But considering millions of customers and millions of products the algorithm encounters severe performance and scaling issues.

It is possible to partially address these scaling and performance issues:
a) By discarding the customers with who purchased very few items

b) Or by discarding the least popular or items which are rated low.
These dimensionality reduction techniques such as clustering and principal component analysis can reduce M or N by a large factor.

However the above can help improve the performance but can also reduce the recommendation quality in several ways.
a) Examining only a smaller scale of customers may not match current customers.
b) If the algorithm discards unpopular or never purchased items it will never appear in the recommendation list.

Cluster Models :

The cluster Model follows the algorithm of finding the customers with the most similarity with the current customer and assigning it to the segment of users. It then uses the purchases and ratings of the customers in the segment to generate recommendations.

Segment creation also uses certain algorithms to sort out similar customers like similarity matrix or clustering algorithms groups the most similar algorithms in segments forming clusters. This might be a suitable approach for a smaller group of customers but when large retail companies are involved like amazon the customer's count will be in millions and clustering algorithms may not be as efficient as expected hence they end up using a greedy cluster generation. Where the algorithm typically starts with an initial set of segments, which contain a randomly selected customer and they repeatedly match all customers with existing segments, following this logic the algorithm ends up creating a new segment or updating the existing segments with more similar customers.

The algorithm compares the user similarity with the vector for user segment details summarized and finds the vector with strong similarity and adds the user to the segment. Some algorithms add users to different segments based on the similarity of the items.

The clusters method performs better than the collaborative clustering algorithm, as it compares the customer with a segment rather than comparing with the entire customer list.

However recommendation quality is still low as the recommendation is made on matching against a segment formed grouping several customers, The recommendation will mostly be not that relevant as expected. This can be improved by using more finer segments but may get expensive and the approach may end up similar to collaborative filtering.

Search-Based Methods :

Search-based algorithms mostly search for the similar item as searched, i.e the algorithm constructs a searching look based on the user's purchased and rated items, algorithm construct query to find the other similar items based on the keyword search. Like books by similar authors, or similar products by different brands.

This kind of algorithm may perform well for customers with few purchases and ratings but may get expensive when customers purchase is high and rated vastly for more similar products.

In any case, the recommendation quality is low because either the search is too narrow(specific author search) or general(movie DVD for drama). As the recommendation goal is to help with a much better search with a more relevant search.

Item to Item Collaborative filtering :

Existing recommendation algorithms may not be efficient and suitable for tens of millions of customers at amazon, so at Amazon, they developed an item to item collaborative filtering, which scales to their massive datasets and produces high-quality recommendations. Amazon uses recommendations as a targeted marketing tool.

"Your recommendation" link in the Amazon webpage, take the customers to the recommended list which can be filtered based on products and subject area.

Below is the shopping cart recommendation based on items in the cart, it offers customers product suggestions. With the Item to Item filtering method rather than matching the user to a similar customer, it matches each of the customers purchased and rated item to similar items and combines those similar items into a recommendation list. This algorithm builds a similar item table by finding items that customers tend to purchase together.

Following iterative algorithm provides a better approach by calculating the similarity between a single product and all related products:

A common method used to compute the similarity between two items is to use the cosine measure, in which each vector corresponds to an item and not customer, vectors M dimension corresponds to the customers who purchased the items, In practice, the time complexity for this algorithm is O(NM), as most customers will have very few purchased or rated items.

```
For each item in product catalog, Item1
    For each customer C who purchased Item1
        For each item I2 purchased by customer C
            Record that a customer purchased I1 and I2
    For each item I2
        Compute the similarity between I1 and I2
```

With a similar item table, the algorithm finds items similar to each of the users' purchases and rated items and aggregates them, to recommend the most popular products. This commutation is quick as it only iterates through the items purchased and rated.

Scalability: A Comparison

As a brief comparison shows, existing methods fall short:

- Traditional collaborative filtering does little or no offline computation, and its online computation scales with the number of customers and catalog items. The algorithm is impractical on large data sets except it uses dimensionality reduction, sampling, or partitioning. However, it will reduce recommendation quality.

- Cluster models can perform much of the computation offline, but recommendation quality is relatively poor.

- Search-based models build keyword, category, and author indexes offline, but fail to provide recommendations with interesting, targeted titles and scale poorly for customers with numerous purchases and ratings.

  The key to item-to-item collaborative filtering's scalability and performance is that it creates the expensive similar-items table offline. The online component looks up similar items for the user's purchases and ratings as well as scaling independently of the catalog size or the total number of customers. The algorithm is fast even for extremely large data sets since the algorithm recommends highly correlated similar items and recommendation quality is excellent.

Conclusion:

Recommendation algorithms provide an effective form of targeted marketing by creating a personalized shopping experience for each customer, a recommendation algorithm for a large retailer like Amazon.com should be scalable over large customers and product catalog, which should require only milliseconds of processing time to generate online recommendations regardless of the user's data and purchase rate. Where Item to Item collaborative filtering is able to meet these criteria.

Key learnings: salient points:

1. Every detail/information provided or related to the user is stored for analysis by the retail company to mine through the information to better understand the needs, thus helping in the business intelligence to serve the customer better and also to favor the company's profit.
2. Most recommendation algorithms use dimensional aggregation and data storage for fast and easy analysis of heavy incoming data to serve online customers better.
3. We could relate the key learning from the course to the paper presentation, as it talked about multidimensional data storage with the aggregate calculation based on the ratings for further recommendations.
4. We could understand that the recommendation algorithms take input structured data such as users, items and historical preferences that could be adequately stored and accessed using a database system.
5. The recommendation algorithm also takes advantage of the DBMS inherent features such as query optimization,materialized views and indexing.

What can be enhanced ?

Since the paper presentation focuses mostly on the algorithm used to filter out the recommendations, it could focus on including the details about how the data used for recommendations is collected and accessed.

Motivation for Project :

While working on the paper presentation, it gave us an insight into the back-end of the big retailer's recommendation system like amazon.com, since most of the recommendation systems rely on the customer's reviews and ratings to serve the online customers better by creating more personalized shopping experience and thus helping with better quality product recommendations, customer reviews and ratings being a key player in the recommendation systems it helped us narrow down our focus on the databases handling the customer's reviews and ratings as a project objective.

Sources

https://ieeexplore.ieee.org/abstract/document/1167344?casa_token=nbY3WL2heqoAAAAA:LmQ7G5TPffD0toVB0BLxP5c1Nlu6pFSl7Ixh9ypaR7CYHiZV_BEPZ5z9wOIoT24nOc-_iXXd3RQ

1 Appendix

Criteria	Remarks
This criterion is linked to a Learning Outcome Source of the Significant Paper Is it from a reputed journal or conference?	Yes,published by IEEE, with 1747 paper citation and 144 patent citation
This criterion is linked to a Learning Outcome Key Learnings Students summarized what they learned and the learnings are substantial	Summary is provided under Key learnings: salient points,click to view
This criterion is linked to a Learning Outcome Topic and contents of the paper Are the contents relevant and significant to the course work?	Summary is provided under Key learnings: salient points above,click to view
This criterion is linked to a Learning Outcome Report Report captures salient points and is of good quality?	Yes, Paper presentation discuss on how the vectors are created is represented in multidimensional form for the users and items,further explaining how multidimensional and their aggregates are handled in recommendation algorithm
use of unique tools	we used Latex to create this report
This criterion is linked to a Learning Outcome Relation to Project Did the project build upon the significant paper?	Yes, our paper presenation topic inspired us to work on a database handling customers review