A NOVEL PRODUCT FILTERING AND PRODUCT RECOMMENDATION SYSTEM TO OPTIMIZE THE SEARCH SPACE AND SPARSITY IN REAL TIME ECOMMERCE ENVIRONMENT

MR.T.SUBHA MASTAN RAO, DR.G.RAMA KRISHNA

1Asstt Prof., Department of Computer Science and Engineering, KL University
2Professor., Department of Computer Science and Engineering, KL University

E-mail: 1mastan@kluniversity.in, 2ramakrishnag_cse@kluniversity.in

ABSTRACT

Online Product recommended system is the most effective prediction system in the e-commerce websites. Customized/Automated recommendation systems can assist the users to find relevant products within short time in large e-commerce databases. Several recommendation techniques have been proposed to filter the user interested products and to optimize e-commerce sales from different vendors. With the tremendous increase in the products and customers in e-commerce systems, the time taken to search the required product using traditional recommended techniques increases due to the large number of products features and its specifications. With the tremendous growth in the products and customers information however, the traditional systems encounter two key issues, enhanced response time and minimized recommending quality. New Qualitative recommended systems are essential to handle. The traditional techniques can’t implement search space optimization when the item-space changes; on the other hand, the overall efficiency of the recommendation system will decrease as the items grow into a large amount. Sparsity of the product is the most significant basis which minimizes the inadequate quality. In this paper, a new product filtering and product recommended system is proposed to optimize the search space and sparsity. Product filtering can be used to filter the relevant products from the web using the conditional probability and similarity values. Product prediction based recommended system is proposed to predict the products ranking based on the products features as well as calculated predicted probabilities. Experimental results show that proposed filtering based recommended system outperforms well on web applications like e-bay in terms of search space and ranking are concern.

Keywords: Online, Product Filtering, Qualitative, Sparsity, Product Filtering

1. INTRODUCTION

In the e-commerce world everything goes on the web and depends on the internet, the purchasers and the vendors never meet one another but transaction deal can be finished through online via internet. Vendors design their webpages to maintain their product’s information. Purchasers use the web-navigator to check out the website and they will admit the deal when those items meet their needs. Recommended filtering (RF) is one of the most potentially/promising recommending approaches. The major advantages of RF system are below: it possesses no special restriction on the highly recommended objects; it cannot only suggest text-based information, but additionally movies , music etc, whose data is hard to be assessed[4]; and it may also choose the users’ new desires and priorities. But, recommending systems utilizing conventional recommended filtering are encountering two kinds of major issues: one is the processing workload; and the other is the forecasting precision. With the increase in large number of users and the items; the scale of the system becomes larger in size. Consequently, on one side, the processing workload boosts more and more rapidly; on the other side, the database of user ratings seems to be much more sparse. It is ever more difficult to determine the users’ similarities precisely, which allows the decreases in the recommending quality. In some ways these two issues are in conflict[1-2].
The system that predicts relevant products using robust filtering technique is known as Recommendation System (RS). The main objective of Recommendation System is to perform task to filter relevant products from the large databases. It much relies upon similarity prediction. In [1], Recommendation System (RS) can be categorized into three different types based upon application used; they are content based technique, collaboration technique, and hybrid technique.

**Content Based Technique**
Content-Based Technique filters products to the user from products favored by that user previously. It is carried out by first creating interaction between product and its features in term of Similarity Matrix then select among the most equivalent products to the target product by computing similarity from different mathematical functions. The most popular similarities like Cosine, Adjusted Cosine and Pearson coefficient. Good similarity techniques will result a high quality filtered items[4].

**Collaborative technique**
This is the most effective technique in offline Recommendation System (ORS). Collaborative technique tries to suggest products for target consumer from consumers with similar priorities. Most typical process of CF performs similarities computation on selection of user favorites that ecommerce websites typically gather rating towards products [3-6].

**Hybrid technique**
This technique starts from problems of above two approaches. In case, a customer purchases product X, it's not practical to suggest an entirely varying product Y to that buyer by Content Based technique. The prediction used in this system is same as Content Based approach but focus on peer experiences.

The conventional recommended filtering techniques include Item based, User based and Model based techniques. In this technique, we predict the user’s behavior towards a particular item using the measured sum of variations from mean scores of consumers that formerly rated this item and the user mean-rate. Using clusters to eliminate the number of products and end-users was suggested to restore the large computation challenge; yet, this will reduce the good-quality of recommendations. Also, partitioning products to item space will restrict the recommendations to exact types of products. Moreover, if the cluster doesn't include the popular or unpopular products, they will never be suggested to users. Search based procedure depends on discovering related popular products to the one that the consumer has purchased or graded highly. This technique works effectively by creating a search-query that looks for items with the same capabilities that the consumer has graded extremely [2-7].

The well-advised solution is to use a sampling of transactions for recommendations which will certainly influence the recommendation good-quality. Discovering popular products that impart one or more capabilities is not a personalized system. Alternatively, Amazon uses Item to item Recommended Filtering. This technique, as mentioned in [5] is scalable and generates high good-quality recommendations. Also, they provide a characteristic where you will see what is highly recommended to you and revise your filter of feedback by product-line and subject. In addition they can get access to those items along with their prior purchases to rate them.

**Issues of recommendation techniques**
The issues for recommendation techniques increase to two core dimensions, noted as scalability and sparsity.

1. Sparsity: As the most of the recommendation techniques are dependent on similarities calculated over the co-occurred rated set of items with large intensity of sparsity.
2. Scalability: Recommendation techniques appear to be highly effective in streaming items
that are interesting to consumers. Yet, they will require computations that are quite expensive and grow nonlinearly with the number of items and users in a database. Thus, in order to bring recommendation techniques efficiently on the web, and are successful in featuring recommendations with satisfactory delay, refined data structures and enhanced, scalable architectures are essential[8-14].

2. PROPOSED APPROACH

Product based recommended system predicts filtered based products list to the target users according to the view of other users. Assumption of the traditional recommended systems include, “if the ratings of the web based products rated by one or more users are similar, then the rating of the other products rated by same users will also be similar”. Proposed workflow is defined in the figure 2. In this proposed approach, web based products along with features are taken as dataset to predict the relevant top-most filtered list to the users. In this work flow, real time web application like e-bay is used to extract products information along with different features like hit rate, associated information, category, title, image, link etc. The entire products lists along with features are stored in a file for preprocessing. Product filtering technique is proposed to find the relevant items from the large set of items along with high dimensional features. Finally, Products recommended system is proposed to predict the sparsity value and then predict the rank based items using conditional probability and product similarity values.

![Proposed Workflow](image)

**Definitions:**
User based item interaction matrix can be defined as “user interaction with the item along with rating”. Product based features interaction matrix is defined as “User interaction with the product along with the features list”. The general format of the product and its features matrix is tabulated below.

<table>
<thead>
<tr>
<th>Features</th>
<th>F₁</th>
<th>F₂</th>
<th>F₃</th>
<th>……</th>
<th>….</th>
<th>Fₙ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
<td>P₁</td>
<td>P₁F₁</td>
<td>P₁F₂</td>
<td>P₁F₃</td>
<td>……</td>
<td>….</td>
</tr>
<tr>
<td>P₂</td>
<td>P₂F₁</td>
<td>P₂F₂</td>
<td>P₂F₃</td>
<td>P₂F₄</td>
<td>……</td>
<td>….</td>
</tr>
<tr>
<td>P₃</td>
<td>P₃F₁</td>
<td>P₃F₂</td>
<td>P₃F₃</td>
<td>P₃F₄</td>
<td>……</td>
<td>….</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>Pₙ</td>
<td>PₙF₁</td>
<td>PₙF₂</td>
<td>PₙF₃</td>
<td>PₙF₄</td>
<td>……</td>
<td>….</td>
</tr>
</tbody>
</table>

Each product rating can be defined by collaborating the features hit rate as:

\[ P_{i,j}F_j = \text{Rate}_{ij} \text{; exists} \]
\[ = 0; \text{otherwise.} \]
Where $P_i$ is the product in the web, $F_j$ is the features list of the product and $P_iF_j$ is the product feature rate.

$$Rate_j = \frac{\text{(Number of hits or Access rate in each category)}}{\text{Total category hit rate}}.$$  

**Algorithm 1:**  
**Input:** User Specified Product/Category, weight $\omega$.  
**Output:** Retrieves all relevant products/Category list.  
**Procedure:**  
**Step 1:**  
Check internet connection or web access service.  
**Step 2:**  
Connect web service using the specified key pair as authentication.  
**Step 3:**  
List all the available products from the web.  
For each item/product in the list.  
Do  
Get product features list as item link, rating, related products etc.  
Done  
Map each item to the available features.  
**Step 4:**  
Find the conditional probability to each pair of features in the product mapping.  
For each pair $(p_i, f_j)$ in the list.  
Do  
**Conditional probability C.P**  
$(p_i, f_j) = \frac{\text{prob}(p_i \cap f_j)}{\text{prob}(p_i)} \times \omega_j$  
where $\omega_j$ is the weight of the each product feature.  
Similarity of the product and its feature can be calculated using eq(2)  
$$Sim(p_i, f_j) = \frac{C.P(p_i, f_j) \times (\sum_{i,j} (\alpha_i - \beta_i)(\alpha_j - \beta_j)) / \left( \sqrt{\sum_{i} (\alpha_i - \beta_i)^2} \sqrt{\sum_{j} (\alpha_j - \beta_j)^2} \right)}{ \sum_{i} (\alpha_i - \beta_i)^2 \sum_{j} (\alpha_j - \beta_j)^2 }$$  
**Step 5:**  
Sort product list based on product feature similarity relationship.  
**Step 6:**  
Find the conditional probability to each pair of features in the product mapping.  
For each pair of products in the list.  
**Conditional probability C.P**  
$(p_i[p_i', f_j]) = \frac{\text{prob}(p_i \cap p_i' \cap f_j)}{\text{prob}(p_i \cap p_i') \times \omega_j}$  
where $\omega_j$ is the weight of the each product feature.  
Similarity of the pair of products can be calculated using eq(4)  
$$SimList(p_i, p_j) = \frac{C.P(p_i[p_i', f_j]) \times (\sum_{i,j} (\alpha_i - \beta_i)(\alpha_j - \beta_j)) / \left( \sqrt{\sum_{i} (\alpha_i - \beta_i)^2} \sqrt{\sum_{j} (\alpha_j - \beta_j)^2} \right)}{ \sum_{i} (\alpha_i - \beta_i)^2 \sum_{j} (\alpha_j - \beta_j)^2 }$$  
**Algorithm 2:**  
**Input :**  
$\lambda$: Product weight threshold  
P: product to be predicted in the list of highest similarity list.  
$U_i$: User Access  
**Output:** Predicted items.
Procedure:

ProductKPredict(U_i, P_i, λ)

If Simlist!=null
Then
   If P_i!=null
   Then
      λ = randomize(0,1);
      List(k):= Selectkitems(Simlist)// select top k items from Simlist.
      x:=0, y:=0;
      for each product in List(k)
         do
            if C.P(P_i, F_rate) not exist  //F_rate feature rating
               then
                  x:=x_i+ (α_i - β_i)
                  y:=y_i+ |Sim(P_i, F_rate)|
                  end if
               else
                  x:=x_i+ λ * ÷/ \sum f_i / N
                  y:=y_i+ λ * |Sim(P_i, F_rate)|
                  end if
      end if
      // calculating items similarity between product features.

Top item in the list is predicted using the equation (6) as

\[ \text{TopSim}\: i, j = \psi \: C.P\: (i, j) + (1 - \psi) \: \text{sim}(i, j) \]

Where \( \psi \) is correlation coefficient between the two product features.

Predicted Probability:
Pr\(ob(P_k,i) := \sum \{TopSim(i,k) * C.P(P_i,k)\}\)

\(P_k\) is the k recommended products \(i=1\ldots n\) features

Finally sort the list of items according to the probability values.

3. EXPERIMENTAL RESULTS:

![Item Prediction Login Page](image1)

![Items Prediction Home Page](image2)

![Parameters Setting For Item Prediction](image3)
Table 1: Average Ranking Of Items In Proposed Approach And Ebay Website

<table>
<thead>
<tr>
<th>Items</th>
<th>Avg-EbayRank</th>
<th>Avg-ProposedRank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobiles</td>
<td>76.34</td>
<td>95.76</td>
</tr>
<tr>
<td>Jewellery</td>
<td>82.45</td>
<td>96.32</td>
</tr>
<tr>
<td>Books</td>
<td>77.23</td>
<td>94.54</td>
</tr>
<tr>
<td>Cloths</td>
<td>79.34</td>
<td>97.45</td>
</tr>
</tbody>
</table>
4. CONCLUSION

In this work, EBay web services are used to analyze ranking and popular items from the large set of database. Sparsity of the product is the most significant basis which minimizes the inadequate quality. In this paper, a new product filtering and product recommended system is proposed to optimize the search space and sparsity. Product filtering can be used to filter the relevant products from the web using the conditional probability and similarity values. Product prediction based recommended system is proposed to predict the products ranking based on the products features as well as calculated predicted probabilities. Proposed approach gives better accuracy in terms of item prediction and filtering infrequent items from the large set of categorical products. Probability estimators used in this method can optimize the product importance based on the features. Experimental results on the web services gives better accuracy in terms of item prediction and accuracy factors are concerned.

REFERENCES


