E-commerce Recommendation System with Reverse Image Search

Project report submitted in fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

ShivamRai (151208)

Under the supervision

of

Dr. Hari Singh

to

Department of Computer Science and Engineering

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Candidate's Declaration

I hereby declare that the work presented in this report entitled **"E-commerce Recommendation system with Reverse Image Search"** in fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2018 to May 2019 under the supervision of Dr.Hari Singh(Assistant Professor(Senior Grade),Department of CSE and IT).

The context in the report is not submitted for the award of any other degree or diploma.

ShivamRai (151208)

This is to certify that the above statement made by the candidates is true to the best of my knowledge.

Dr. Hari Singh

Assistant Professor (Senior Grade)

Department of CSE and IT

Dated: 13 May, 2019

Acknowledgement

It is my privilege to express my sincerest regards to my project supervisor Dr. Hari Singhfor his valuable inputs, able guidance, encouragement, whole-hearted cooperation and direction throughout the duration of our project.

I deeply express my sincere thanks to our Head of Department **Prof. Dr. Satya Prakash Ghrera** for encouraging and allowing me to present the project on the topic "E-commerce Recommendation System using Reverse Image Search" at my department premises for the award of my B. Tech degree.

I would also like to express my sincere thanks to all my friends and others who have helped me directly or indirectly during the project work.

Date: 13-05-2019 ShivamRai (151208)

E-Commerce Recommendation System with Reverse Image Search

Abstract

The business-to-consumer(B2C) part of Electronic commerce (e-commerce) is the most prominent and wide use of business over the World Wide Web. The essential objective of an e-commerce website is to sell goods and services online.

This project deals with developing an e-commerce website for Sale of Different types of products categorized under similar features.
In order to facilitate online purchase User Accounts and Guest Accounts are provided to the user.

Our system is implemented using a 3-tier approach, web browser as our front-end client, Node web server as a middle tier and with a backend database (MongoDB).

This project allows viewing various products available and tagging them, enables registered and Guest users to purchase desired products instantly via different payment platforms (Stripe Payment) or COD (Cash on Delivery).

Our Website also helps users, either Registered or Guest, to search products accurately by using Machine Learning Techniques like Text analysis, Text sentiment analysis, Computer Vision and Deep Learning.

Many unique features are also added in the project like Searching products by image using reverse image search, User can tag products to get updates on related products and every product will have a consumer blog where product related issues can be discussed.

Introduction

1. What is E-Commerce?

E-commerce, or Electronic commerce, is the used to denote any commercial transaction or trade that takes place between at least two or more people using the internet [1]. In a much broader sense we can define E-commerce as: “The marketing, promoting, buying & selling of goods electronically, particularly via the Internet”, which encompasses, *interalia*, “e-tailing (virtual shop fronts), EDI, which is B2B exchange of data; e-mail & computer faxing; [and] B2B buying and selling [2]”.

1.1. Features of an e-commerce are (all or few) [3].

E-commerce:

1.1.1. Is pervasive, available anyplace, anytime.

1.1.2. Has worldwide reach.

1.1.3. Operates as per universal standards shared by all countries around the globe.

1.1.4. Provides data extravagance (Plethora of options).

1.1.5. Is interactive – it permits two-way interaction between Buyer and Seller by means of common medium.

1.1.6. Increases information density. Information on the e-commerce, like products, services and users keeps on growing and thus increasing information density.

1.1.7. Permits personalization and customization. Sellers can target their marketing messages to specific individuals by using their past purchasing patterns, Ethnicity, Region etc.

1.1.8. Is easily accessible and is not limited to a set of people. Free, easy and regular accessibility to everyone.

2. Definition of Terms

2.1. Recommender: A recommender is a word used to portray someone who puts forward (something or someone) as being suitable for a particular purpose or role [4]. Recommender systems can be stated as functionalities or programs which try to recommend the most accurate suitable items (products or services) to specific users (individuals or businesses) by predicting a user's interest in an item based on related information (By comparing all the features) about the items, the users and the interactions between items and users [1]. The aim of developing a recommender system is to minimize the overloading of information in retrieving the most relevant information and services from huge data, thereby providing personalized recommnedation. The most valuable feature of a recommender system is its ability to "guess" user's preferences and interests accurately by analyzing the behavior of user and the behavior of other users to generate personalized recommendations [2].

2.2. Machine Learning: This is a major and critical sub field which comes under Artificial intelligence (AI), which is centered around algorithms and computation models that are trained and these models learn from data provided. Machine learning models learn from the dataset provided and then predict future results.

2.3. Algorithm: An algorithm is a detailed independent step by step progression of activities with a specific end goal to solve a particular problem. Algorithms are like step by step guidance to tackle a problem. Algorithms can be used to perform computations, handle information and solve other computer and mathematical operations.

2.4. Collaborative Filtering: Collaborative filtering (CF) is an algorithm that is mainly applied in recommendation systems to predict the interest of clients (say recommending a product a user) by finding common behavioral patterns from data of numerous clients, so called Collaboration. The underlying assumption of Collaborative Filtering is that similar users share the same interest and like similar items. For example, a collaborative filtering recommendation system for E-commerce website will forecast the product that user will probably like from the user's previous likes and dislikes and all the other users in their database with similar likes and dislikes.

2.5. **Reverse Image Search:** This is a technique that uses Deep Convolutional Neural Network (DCNN) architecture to get images with similar features from the database. First of all, an image is provided, and the DCNN architecture extracts all the features from the image and makes a property array of the image. All the images in the database are already processed and they have their features array prepared statically. The given image's property array is matched from all the images in the database and the best matching results are displayed.

3. Problem Statement

3.1. Products in the E-commerce India face problems such as:

3.1.1. E-mail marketing Faux pas:

Many of the existing E-commerce website flood their user's e-mail inbox with various advertisements and products that are irrelevant for the user. This decreases the reputation of an e-commerce website and the user ultimately un-subscribe from the website and the E-commerce business loses its valuable customer.

3.1.2. Product Suggestion problem:

Collaborating filtering ismostly used inrecommendation systems, but it also has some problems like sparsity (sparse data available) and cold start (no data available). Differentmodifications are applied to handle the CF problems, but there is no single algorithm whichcan predict the personalized needs of each user in an e-commerce website. Multiple algorithms are applied to suggest product for every user, even then suggestion problem cannot be completely solved, only accuracy can be improved over time.

3.1.3. Search Problem:

Search problem is the most prominent problem across every existing e-commerce business. Every user expects for the best product according to his/her taste. No-one wants to waste their precious time in searching for a product. Relevant and good products are the top priority of every customer. Every user can search for a product if the user knows what exactly it is called. Exact name must be known to search for a product. But at times user may come across certain product that may be completely unknown to him or may not know the exact name or description. For example, say a unique necklace that has embodied gems. Searching for term ‘necklace’ will show a lot of options that might confuse the user or even show entirely different product.

3.1.4. Product Ranking:

Ranking the product in searches and suggestions is a very critical task. Fulfilling this expectation of the customer is very hard for any E-commerce website. It is relatively more difficult for New Users (because of lack of data) and also for very old users (because of overfitting of data). There are many factors involved that determines the taste of any user and predicting that taste as well as balancing all the changes is very hard.

3.1.5. Security:

Most users are afraid of their personal info being leaked on these ecommerce websites. With every E-commerce business, security is a major concern and well-established and huge organizations are capable to spend a fortune to handle security issues but many of the small ecommerce business can’t afford such extravagant expenditure. This makes the user hesitant to use their services.

4. Methodology

4.1.1. E-mail marketing Faux pas:

We have considered the flooding of e-mails with advertisements in our project. In our E-commerce website, users receive e-mails only for those related products in which he/she has shown his/her interest by tagging the product. Suggesting products via e-mail based on the history of the user gets flooded over time and relevance and objective of suggesting product is lost over time.

4.1.2. Product Suggestion problem:

We have tried to solve product suggestion problem by giving significant weight to the product tagged by the user along with using Collaborative item-based filtering algorithm. For sparsity and cold start, we are also trying to suggest them product by asking (most probably a short form) them about their taste and interests. This can be a little tedious for new users but in this way, we will be able to serve them better. We are also working to improve suggestions for new users in other ways.

4.1.3. Search Problem:

To overcome search problems, we offer 'Reverse Image Search' for products that will show visually similar product of the image uploaded on the website. This will save a lot of time of the customer and the customer can search for his/her favorite product with more ease and accuracy. This will also help to develop the taste of the user and the weightage of this image will be more as image describes a lot more than traditional text searching.

4.1.4. Product Ranking:

Whenever a user searches for a product, options showed to him are based on the text, or previous history of the user or some featured products that are being endorsed by some brands. But we are working on to show best products on based of mixture of Rating of the product and reviews of the products. We are trying to increase the quality of the products showed on top by considering more about the quality of reviews and ratings given by the users.

4.1.5. Security:

The hesitation among the customers for not using small E-commerce site can be reduced if user can buy products by providing minimum information on these websites. To overcome this issue, we have included a buy on the go feature which enables the user to buy products without signing-up for an account. Interested User can buy products and only their E-mail address will be saved with us and rest of the details will be filled by the user but we will not store the rest of the details and any further queries or problems can be solved via E-mails only.

Literature Review

RELATED WORK AND REVIEWS

1.1. Reviews

1.1.1. Recommender Systems (RSs) are tools and techniques which uses software to provide suggestions of useful items to a user [6]. The suggestions provided by a recommendation system depends on the user such as what things to buy, which song to listen, what news to read or which place to visit next etc. The recommender systems are concerned with learning about the user through his/her activities and then providing appropriate and favorable suggestions to the user. The main difference between a search engine and recommendation system is “individual interest” and “intriguing and helpful” results [5].

1.1.2. The term “Item” in recommendation systems generally refers to an object that is suggested by the recommendation system.

1.1.3. Recommender systems are useful for an e-commerce site as to increase the quality of products showed to the user and help them increase their deals, user engagement and growth. Today, most of the e-commerce site use some sort of recommendation technique. Most widely used are collaborative, content based, and knowledge-based filtering [5].

1.2. Related Work

1.2.1. Neal Lathiaxshowed the importance of diversity in recommendations through a survey given by users. Recommendations generated by CF algorithms produce similar results over time as they always pick the top-N recommendations. Generally, over time the need and interest of the user changes over time but none of the recommendation algorithm takes time into account. With time diversity of a user increases but the recommendation remains the same [6].

1.2.2. Linden et al. proposed Amazon.com recommendations algorithm: an item-to-item collaborative filtering. This approach was much more efficient than other collaborative filtering algorithm. The item-to-item CF is used by Amazon for generating recommendation. Item-to-item CF is easier to scale and can be applied to a very large system (like amazon with very large number of users and products) [8].

1.2.3. For Text-sentiment analysis, Support Vector Machines (SVMs) [9,10] and the Naive Bayes algorithm are the most common classification techniques. The accuracy of these two algorithms ranges from 63% and 82%, but these results depend on the features selected (like important positive and negative words). We have used Naïve Bayes algorithm as it performs well (greater than 75%) in all situations. In [7], they compared two methods for sentiment analysis: Lexical method and Machine Leaning methods. Lexical methods (which compares the words of a text with a database which has weights for each and every word and overall result is calculated by taking average of all the weights), Machine Learning methods like SVM, Naïve Bayes and ADTree). Below is the given comparison of three famous algorithms for Text sentiment analysis from [7].

Table: Machine Learning Accuracy Results (%)

Approach	**SVM-Light**	**Naïve Bayes**	**ADTree**
Unigram Integer	77.4	77.1	69.3
Unigram Binary	77.0	75.5	69.3
Unigram Integer + Aggregate	68.2	77.3	67.4
Unigram Binary + Aggregate	65.4	77.5	67.4

1.2.4. Image Similarity Models: There are various Image similarity models. Most popular and widely used are SIFT (Scale Invariant Feature Transform) and HOG (Histogram of Oriented Gradients). SIFT [11] extracts the image features and convert it into a large collection of local feature vectors. Stages of SIFT includes feature detection, Local image description, Indexing and Matching and Model Verification. HOG [12] breaks the image in blocks and then generates histogram from first order image gradients. HOG is Dense and Hand Engineered. These two models are good at finding similar images with high level similarities (car match car) but are not able to compare low level similarities (red car match red car). But on the other hand, DCNN architecture can extract and match high level as well as low level features.

1.2.5. Kiapour et al. tried to compare items (pictures) on the street and on online shopping websites, to get a match. They used a number of methods and their two-layer DCNN performed the best to find a correct match [14].

1.2.6. Jing et al. proposed a scalable and a cost-effective visual Search system using AWS and other various open source tools. They used a DCNN architecture to extract local features, deep features and salient color signatures. They used a Two-step object Detection and

Localization. But problem with their model was that their model showed both type (images with high level similarities and also images with low-level similarities) of similar images [13].

2. Existing Work

2.1. Existing Recommendation System - *Amazon.com:*

Amazon.com also uses a recommendation system to generate recommendation for each and every user uniquely. They use item-to-item collaborative filtering algorithm. They have tweaked the general item-to-item CF algorithm to increase scalability. Their algorithm works well with massive data sets for each user and also provides high quality recommendations in real time.

2.2. Flipkart.com Recommendation System:

Flipkart gives recommendation to the users based on multiple usage instances. Flipkart recommendation system focuses on user's search terms, history and wish list. Flipkart also gives more priority to the last viewed products by the user. They store user's history on the user's system locally in the form of cookies.

2.3. Existing Image Search

2.3.1. Google also provides a "Search by image" feature that uses reverse image search and allows users to search for similar images by uploading an image or image URL on their site. Google extracts image features and then construct a mathematical model of it using advanced algorithms [15]. It is then compared with billions of other images in Google's databases and returns similar images. It should be noted that when available, Google also uses metadata about the image such as image description, date created, size, resolution etc.

2.3.2. TinEye is a search engine that is specifically used for reverse image search process. First, we upload an image to TinEye, then TinEye creates a "unique and compact digital signature or fingerprint" of the uploaded image and then image is compared with the rest of the indexed image inside TinEye database [16]. The algorithm used by TinEye enables them to match even the heavily edited similar images, but results are not very accurate [17].

2.4. Recommendation techniques

2.4.1. Content-based recommendation techniques

2.4.1.1. Recommendations techniques on contents (CB) recommend items or commodities similar to the items previously preferred by a particular user[6]. The basic principles of CB recommender systems are: 1) to analyze a particular user's description of the preferred items to identify the main common attributes (preferences) that can be used to distinguish these items. In a user profile, these preferences are stored. 2) Compare the attributes of each item with the user profile in order to recommend only items with a high degree of similarity to the user profile[6]. Two methods were used in CB recommendation schemes to produce suggestions. One method produces heuristically suggestions using traditional techniques of retrieving data, such as measuring cosine resemblance. The other method uses statistical research and machine teaching techniques to generate suggestions, mainly constructing models capable of understanding the concerns of customers from users ' historical information (practice information).

2.4.2. Collaborative filtering-based recommendation techniques

2.4.2.1. Recommendation-based collaborative processing (CF) methods assist individuals create decisions oriented on the views of others who share comparable interests[19]. The CF method can be split into CF methods based on users and items[20]. A customer will obtain suggestions for products preferred by comparable customers in the user-based CF strategy. A customer will obtain suggestions for products comparable to those they have enjoyed in the past in the item-based CF strategy. Pearson correlation-based similarity[21], restricted Pearson correlation (CPC)-based similarity, cosine-based comparison, or adapted cosine-based measures can calculate the resemblance between customers or objects. Only people who have ranked both products will be regarded when calculating the resemblance between products using the above steps. This can affect the precision of resemblance when products receiving a very tiny amount of scores share a elevated amount of resemblance with other products. To improve similarity accuracy, the combination of the adjusted cosine approach with Jaccard metric as a weighting scheme presented an enhanced item-based CF approach. The Jaccard chart was used as a weighting system with the CPC to achieve a weighted CPC measure[22] to calculate the resemblance between customers. [23].

2.4.3. Knowledge-based recommendation techniques

2.4.3.1. Knowledge-based (KB) suggestion provides customers with products oriented on user knowledge, objects and/or interactions. Usually, KB suggestions maintain a functional information base that defines how a particular product meets the needs of a specific user, which can be done on the basis of inferences about a user's need and a feasible recommendation[14]. Case-based argument is a popular manifestation of KB suggestion method in which case-based advice schemes portray objects as instances and produce suggestions by retrieving the user's request or profile in the most comparable cases[24]. Ontology reflects the domain ideas and the interactions between those ideas as a formal technique of depiction of information. It was used in recommender schemes to convey domain knowledge[25]. It is possible to calculate the linguistic resemblance between products based on the ontology of the domain[26].

2.4.4. Hybrid recommendation techniques

2.4.4.1. A hybrid recommendation method that incorporates the finest characteristics of two or more recommendation methods into one hybrid method was suggested to obtain greater efficiency and solve the drawbacks of traditional recommendation techniques[27]. According to Burke[27], there are seven fundamental combined hybridization methods used in hybrid building recommendation schemes:weighted[28], mixed[29], switching[30], mixture of features, increase of features[31,32], cascade[14] and meta-level[33]. In the existing hybrid recommendation techniques, the most common practice is to combine the CF recommendation techniques with the other recommendation techniques in an attempt to avoid problems of cold start, sparse and/or scalability[3,34]..

2.4.5. Computational intelligence-based recommendation techniques

2.4.5.1. Techniques of computational intelligence (CI) include Bayesian methods, artificial neural networks, methods of clustering, genetic models, and methods of blurring. These computational intelligence methods are commonly used in recommendation schemes to build recommendation models. A Bayesian classifier is a probabilistic approach to solve issues with classification. Bayesian classifiers are common with model-based recommendation systems[35] and are often used to obtain the CB recommendation system model.When implementing a Bayesian network in recommender schemes, each node refers to an object and the countries match each feasible ballot value. There will be a collection of sibling products in the network for each object representing its highest predictors. Also implemented as a structure for merging CB and CF methods was a hierarchical Bayesian network[36]. An artificial neural network (ANN) is an array of interconnected nodes and weighted connections influenced by biological brain design that can be used to build model-based recommendation systems[35].Hsu et al.[37] used ANN to build a TV recommender scheme to train a three-layered neural network using the neural network back-propagation technique. Christakou et al.[38] suggested a hybrid recommendation scheme combining CB and CF to produce accurate film suggestions. The system's content filtering portion is focused on a qualified ANN that represents personal customer preferences.Clustering involves assigning objects to organizations in order to make objects in the same group more comparable to products in distinct communities. Clustering can be used to decrease the expense of computing, for example in[35], to find the knearestneighbours. Xue et al.[39] introduced a typical use of clustering in recommendation schemes. Their technique utilizes clusters to smooth individual users ' unrated information. Using ranking data from a community of tightly associated consumers, the unrated products of an individual customer in a community can be anticipated.Furthermore, assuming that the closest neighbour should also be most comparable to the current consumer in the Top N clusters, only the closest neighbours in the Top N clusters should be chosen so that the scheme can be scalable. The clustering method is also used by combining items[40] to tackle the issue of cold starting in recommender schemes. Ghazanfar and Prügel-Bennett[41] used clustering models to determine and resolve the issue of gray-sheep consumers. Genetic algorithms (GA) are stochastic search methods appropriate for parameter optimization issues with an objective feature topic to difficult and smooth limitations[42].They were primarily used in two recommender systems aspects[43]: clustering[42] and 14 J. Lu et al./Systems 74 (2015) Decision Support 12–32 hybrid user models[44]. GA-based K-means clustering is implemented to a real-world internet industry segmentation situation for custom recommendation schemes in[42], leading in enhanced segmentation results. For ideal similarity features, a genetic algorithm technique is provided in[43]. Results indicate that the similarity features acquired provide greater performance and quicker outcomes than traditional metrics provide. Fuzzy set theory provides a wealthy range of non-stochastic uncertainty leadership techniques.It is well adapted for managing imprecise data, the unsharpness of object or situation categories, and the graduality of preferential profiles[45]. In[46], an element was depicted as a blurred set over an argument set in a recommender scheme. A feature or attribute valuation for an object is a blurred array over the feature-relevant sub-set of assertions. The deliberate choices of the user are depicted as a fundamental matrix of choice, the ordered weighted average of elements that can assess objects. Extensional opinions of the user are conveyed as a blurred collection over the skilled products of the user whose scores are the affiliation degrees. The user's preference for an object can be inferred based on the depiction. In[45,47], a range of products features and a set of values are described for each function. The objects are displayed by a function matrix as the blurred subgroup over the variables. Cao and Li[48] used language terms for field specialists to assess the characteristics of consumer digital goods and enable consumers to convey their requirements for product characteristics using language words.The customer preferences in[49] are portrayed as two blurred relationships, both beneficial and negative, from user set to object set. The resemblance of the product is calculated by incorporating CB resemblance, which is a blurred relationship within a collection of items, and CF resemblance based on items, which is calculated based on customer expectations. Based on the preferences and object resemblance relationships, the customer resemblance is produced by blurred relational calculus. Composing the above-mentioned blurred relationships generates the ultimate suggestions, which are the favorable and bad attitudes.Porcel et al.[50] created a Fuzzy Linguistic Recommendation System that combines CB filtering with the Fuzzy Linguistic Modeling Multigranular Technique, which is helpful for evaluating distinct qualitative ideas. Zhang et al.[9] used fuzzy set methods to address language scores and calculate the blurred CF similarity to provide a alternative to manage uncertainty in a recommendation system for telecom products/services.

2.4.6. Social network-based recommendation techniques

2.4.6.1. Due to the drastic development of social networking instruments in Web-based applications in latest years, social network assessment (SNA) has been used in recommendation schemes. Recommending schemes progressively enable people to participate in personal communication with other customers, such as internet friends, creating social remarks, social labels, etc., to assist enhance customer experience. These developments give possibilities to make suggestions by using the personal connections of customers, particularly for applications with ranking information that are too scarce for cooperative filtering. "Trust" in social network research is a commonly debated connection.Considering the true world scenario in which one's choice to buy is more probable to be affected by friends recommendations than internet ads, the social network of a user may be an significant source if it occurs in a recommender scheme. Similarly, because of the failure of conventional CF methods to locate enough comparable surroundings in scarce information collections, the personal interactions of customers are arising as another facet of enhancement for recommender schemes. Trust reflects another user's intuitive view. The term "trust" is generally described in a recommender scheme as "how well does Alice trust Bob with regard to the particular item or taste"[51].Online groups have been shown to have a favorable connection between confidence and user similarity[52]. Researchers performed sequence of research on trust integration into recommendation schemes. Usually, these trust-based frameworks are oriented on analyzes of users ' diffusion process of "the trust web." The undefined trust value was anticipated in the trust measurement module of Massa and Avesani[53], based on the premise that "consumers nearer to the origin consumer in the trust network have greater trust value."Golbeck[54] suggested a systematic method, TidalTrust, to solve the issue of trust-based classification forecast and is regarded to be efficient in the phase of building numeric trust networks across multiple structures. Ben-Shimon et al.[51] used a Breadth-First Search algorithm to build private social spaces for active customers and then calculated the distances between active customers and others, which can be seen as a sign of confidence, as the ultimate weights of forecast. In[55], in a recommender scheme, the writers evaluated the local trust matrix and the worldwide trust matrix.Their findings show that local trust awareness as well as worldwide trust awareness (also recognized as reputation) can boost increased visibility and precision of recommendations. It is typically assumed that trust-based methods can boost coverage of recommendations by keeping precision. A huge amount of other kinds of personal relationships are used for the creation of recommendations other than confidence. For instance, personal bookmarks[56], physical context[57], personal tags[58], "co-authorship" relationships[59], and more lately, confidence or similarity measurement replacements have been used to filter and predict a user's choice.Shiratsuchi et al.[56] created an internet data recommendation scheme centered on an internet bookmarking "co-citation" network that treats the amount of "co-cited" bookmarks as the weight of personal relationships. Woerndl and Groh[57]collected as a vector the full appropriate social background and incorporated it into ranking information to create a multi-dimensional user-item-context matrix to produce private suggestions in a specific setting. In[58], Ma et al. tried to combine a technique of probabilistic matrix factorization with data on cultural context / trust for recommendations.A cultural relationship is depicted by the concept of "co-authorship" in the job of[59] on the advice of scholarly operations: "the moments two scientists have co-authored articles." Researchers also carried out several research on the recommender systems ' social networks relying solely on the user-item ranking matrix. Palau et al.[60] have designed social networks to display cooperative interactions and suggested several steps to clarify how cooperation can be accomplished within the structure of recommendations. O'Donovan[61] argued that resemblance between users could be overemphasized. They presented a trust calculation model from rating data in their trust-based recommendation architecture to make the system more explainable without decreasing predictionaccuracy.

2.4.7. Context awareness-based recommendation techniques

2.4.7.1. One of the most cited definitions of context is the definition of Dey et al. [62] that defines context as “any information that can be used to characterize the situation of an entity. An entity could be a person, a place, or an object that is considered relevant to the interaction between a user and an application, including the user and the application themselves.” The context information such as time, geometrical information, or the company of other people (friends, families or colleagues for example) has been recently considered in existing recommender systems; for example, the information obtained with the rapid growth of mobile handset use [63]. The contextual information provides additional information for recommendation making, especially for some applications in which it is not sufficient to consider only users and items, such as recommending a vacation package, or personalized content on a website. It is also important to incorporate the contextual information in the recommendation process to be able to recommend items to users in specific circumstances. For example, using the temporal context, a travel recommender system might make a very different vacation recommendation in winter compared to summer [64]. The contextual information about users in technology enhanced learning environments is also incorporated into the recommendation process [65]. J. Lu et al. / Decision Support Systems 74 (2015) 12–32 15 In the review of Adomavicius and Tuzhilin [11], context in the recommender system field is a multifaceted concept used across various disciplines, with each discipline adopting a certain angle and putting its “stamp” on this concept. With context awareness, the rating function is no longer a two-dimensional (2D) function (R: User × Item → Rating) but becomes a multi-dimensional function (R: User × Item × Context → Rating), where User and Item are the domains of users and items respectively, Rating is the domain of ratings, and Context specifies the contextual information associated with the application. To incorporate the contextual information in recommender systems, Adomavicius and Tuzhilin [11] proposed a three-step process to make such information computable and valuable: Contextual PreFiltering, Contextual Post-Filtering, and Contextual Modeling. By processing all three steps, the system can detect the contextual information that is useful and compliable for making suggestions.

2.4.8. Group recommendation techniques

2.4.8.1. Group recommendation schemes (GRS) are suggested to generate a community of customer recommendations when set participants are unable to meet for face-to-face bargaining or their views are not evident despite meeting each other[66,67]. Also known as e-group activity recommendation schemes, GRS has been implemented to many fields including films, music, web pages, activities, and complicated problems such as travel schedules. Many approaches are used to aggregate all participants into a community, influenced by the concept of personal selection and decision-making process.As the most prevalent in GRS, Masthoff[12] described 11 approaches, including the least poverty, median, most enjoyment, and their modifications. Quijano-Sanchez et al.[68] used average strategy; PolyLens[69] used the least poverty strategy[70]; MusicFX used an average non-misery strategy version; and Popescu[71] embraced the electoral system. Other tactics are also used in aggregation, such as consent voting and amount. Asynchronous and synchronous applications are also engaged in GRS for multi-user assistance, with the exception of aggregating techniques. In[72], consumers have created an asynchronous interaction system in which consumers in a community can view (and duplicate) the decisions of other participants. A synchronous conversational scheme was introduced by McCarthy et al.[73] to generate group recommendations for ski holidays. Group participants can criticize the characteristics predefined in this scheme, both for resorts and lodging. All responses from the employees can be aggregated and suggestions are eventually produced that fulfill the group as a whole.

System Design

1. **Overall description**

1.1. Basic Features

1.1.1. Any member can register and view available products.

1.1.2. Registered Users and also Guest Users can purchase multiple products.

1.1.3. ContactUs page is available to contact Admin for queries.

1.1.4. There are three roles available: Guest, User and Admin.

1.1.4.1. Guest can view available products and buy products.

1.1.4.2. User can view, purchase products, save products and tag products.

1.1.4.3. Admin has some extra privilege including all privilege of visitor and user.

1.1.4.4. Admin can add categories, edit products and add/remove product.

1.1.4.5. Admin can add user, edit user information and can remove user.

1.2. Advanced Searching options

1.2.1.1. Search based on text analysis to get accurate results

1.2.1.2. Use of Computer Vision to suggest related products.

1.2.1.3. Text sentiment analysis of comments to show highest rated products on top.

1.3. Search products by Image

1.3.1. Enables Users to search products by **Image**.

1.4. Extra Features

1.4.1. Personal blogs for customers

1.4.2. Product Discussion boards

1.4.3. Tag the products to receive notification of related new products.

2. Data Model:

Data Model for a project represents the data structure required by the database. Our project uses MongoDB, which is a schema-less NoSQL document database. That is, data is stored in the form of JSON document, and the structure of these documents can vary from one another.

But without a data model, operations (CRUD) becomes an overhead for multiple entries. And our Project uses approximately 500 entries, so to simplify the operations we use Mongoose which is an Object Data Modelling (ODM) library for MongoDB and Node.js. Mongoose manages data translation between Objects in code and then saving the Objects inside MongoDB. Mongoose helps in creating schema and Models for JSON objects. Schema in Mongoose is used to define a structure, default values, validators, etc. for JSON documents so that we don't have to worry about the structure of similar entries. Schema is also used to create models for our JSON objects. Model in Mongoose is like a wrapper for the Mongoose schema. Mongoose model provides an interface for us to connect to our database for CRUD operations.

2.1.Database Design

Our database consists of three MongoDB collections which is comprised of User, Product and Cart. A collection in MongoDB is a set of JSON documents that consists of possibly similar type of JSON documents. Type and properties can be varied but it increases the complexity and reduces flexibility. User collection saves the user information like email, name, profile, address and history. Product collection saves information about the products, that is name, price and image. Cart collection saves information about the owner, total and items. These collections comprise the whole dataset. Each collection is filled using the data extracted from 'Flipkart data set' which is available on Kaggle.com to use for general purpose.

We have used only 500 entries out of 20000 entries to reduce the computation time.

2.1.1. User Schema

User-schema creates a reference for the fields to be stored for a user. Our user has fields like email, password, profile, address and history of the user. Each user is provided a unique "_id" which is generated by the MongoDB automatically. This userSchema is used to define out User data model which is responsible for operations on our data like adding a user, deleting a user and retrieve the user during a session.

```
var mongoose=require('mongoose');
var bcrypt=require('bcrypt-nodejs');
var Schema=mongoose.Schema;
var crypto=require('crypto');
/* The User schema attributes / characteristics / fields */
var userSchema=new Schema({
    email:{
        type:String,unique:true,lowercase:true},
    password:String,
    profile:{
        name:{
            type:String,default:''},
        picture:{
            type:String,default:''}
        },
    address:String,
    history:[{
        date:Date,
        paid:{
            type:Number,default:0
        },
        //item: {type: Schema.Types.ObjectId,ref:'}
    }]
});
```

2.1.2. ProductSchema

productSchema creates a schema required by our products. Our product has fields like category, name, price and image to store the respective details. Here category is an Object reference from the data model "category". That is, each product is stored with a category reference to make information retrieval and updating feasible. This productSchema is used to define our Product data model which is responsible for operations on our data like adding a product, deleting products and retrieve products when searched or when a particular category is visited.

```
var mongoose=require('mongoose');
var Schema=mongoose.Schema;
var mongoosastic=require('mongoosastic');

var productSchema=new Schema({
    category:{type:Schema.Types.ObjectId,ref:'Category'},
    name:String,
    price:Number,
    image:String
});
productSchema.plugin(mongoosastic,{
    hosts:[
        'localhost:9200'
    ]
});
```

2.1.3. CartSchema

cartSchema creates a schema for our Cart data model. cartSchema has fields like owner, total and items, where “items” is an array that contains all the items stored by the user. Here owner and items: [item] is a reference type as cart is owned by a user and only registered users can have cart and [item] is a reference from the product as product is one of the “product” picked from the database and also to get the price and image of the product. cartSchema is used to define our Cart data model that is used to retrieve user’s cart information from the database and update cart items and price.

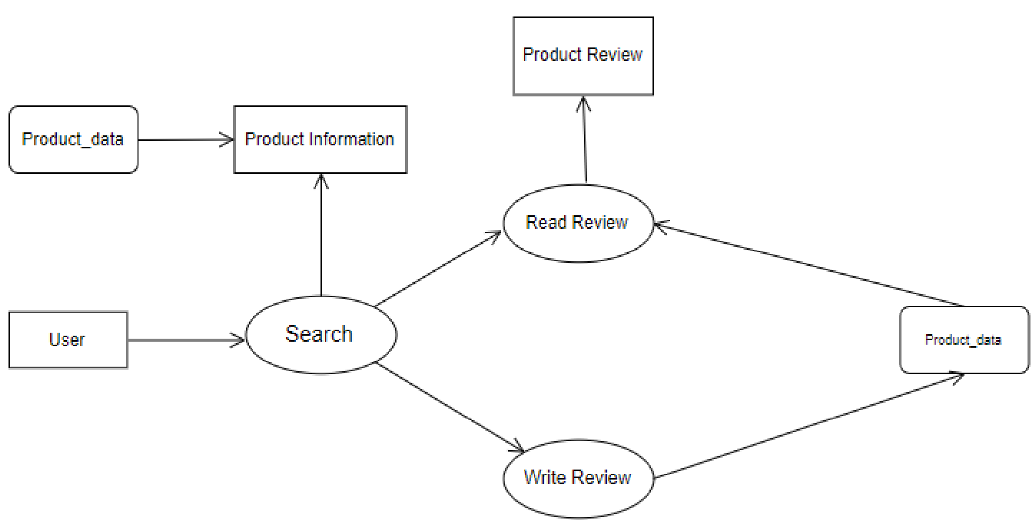
```
var mongoose=require('mongoose');
var Schema=mongoose.Schema;

var cartSchema=new Schema({
    owner:{ type: Schema.Types.ObjectId,ref:'User'},
    total:{type:Number,default:0},
    items:[{
        item:{type:Schema.Types.ObjectId,ref:'Product'},
        quantity:{type:Number,default:1},
        price:{type:Number,default:0}
    }]
});
```

2.2.Data Flow Diagram

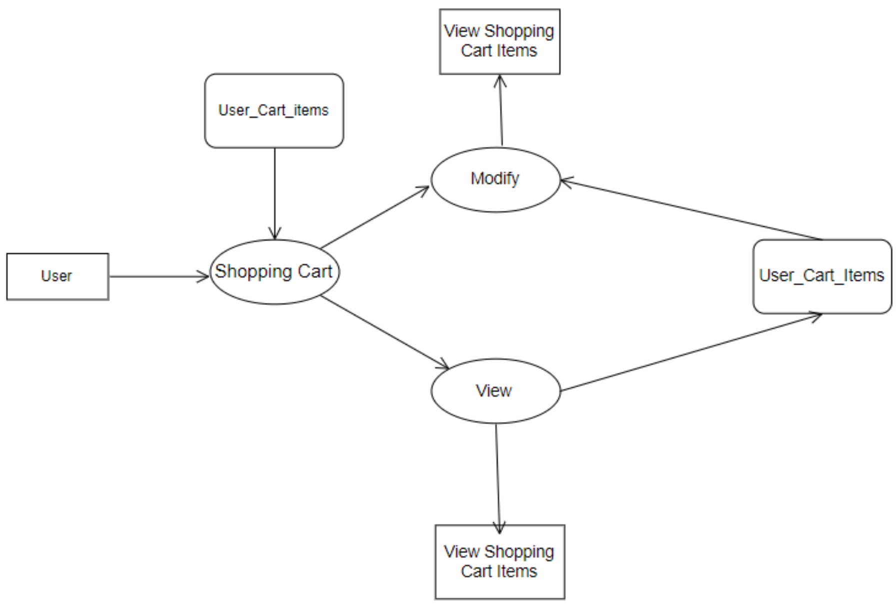
2.2.1. Search DFD

Search DFD represents the transfer of data in between the user (External entity) and search(process). Search process interacts with product_data (data store) to display search results and further if user views a product, product information is fetched from product_data(database) and showed to the user.



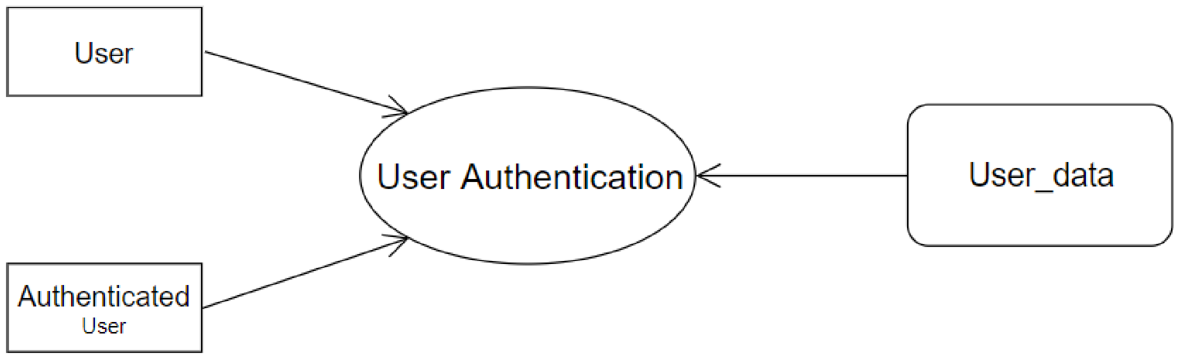
2.2.2. Shopping Cart DFD

Shopping Cart DFD represents the transfer of data in between user (External entity) and Shopping cart(process). Shopping Cart process then fetches data from User_cart_items(database) and displays the result to the user. There are two cases: user may view the cart or modify the cart. While viewing data is fetched from the User_cart_items and displayed to the user and when user modifies the shopping cart, then User_cart_items is updated.



2.2.3. Customer Authentication DFD

Customer Authentication represents interaction of data between user (External entity) and User Authentication(process). User Authentication process then interacts with User_data(database) to authenticate the existence of the user and returns the result.



2.3.Implementation Details

We have developed our E-commerce website with the objective to develop an online store that sells different type of products. As soon as the user types the URL of our website in the address bar of the browser, our Node Server is contacted to get the requested information. In our project, Node.Js server acts as the Web Server. The task of a web server is to accept incoming HTTP requests and to return the response to the front end client. The first thing Server does when a request comes in is to decide how to handle the request. Its decision is based upon the query field of the request. For example, if the requested file query = page-2 then Node’s Express via find the route to the page. If page exists, then that page is rendered else error page is rendered.

2.3.1. Node.JS is

2.3.1.1. JavaScript runtime that uses the v8(written in c++, used by chrome browser) JavaScript engine.

2.3.1.2. Allow us to run JS on the server

2.3.1.3. Used to build very fast and scalable real-time applications

2.3.1.4. Uses an event driven, non-blocking I/O model, asynchronous. Version: 10.11.0

2.3.2. MongoDB

2.3.2.1. MongoDB is a schema-less NoSQL document database. That is, data is stored in the form of JSON document, and the structure of these documents can vary from one another. This is one of the advantages of using NoSQL database. **Version: 4.0**

Algorithms

1. ITEM-BASED COLLABORATIVE FILTERING ALGORITHM [18]

1.1. Item recommendations provide can be possibly based on the history of the user, demographics, rating of the items, bestselling items (products) or history, likes, dislikes and buying pattern of neighbors (similar minded people). Collaborative filtering is one of the most successful recommendation techniques that is used in variety of situations. CF-based algorithms give recommendations based on the likes and dislikes (behavior) of like-minded users (users with similar taste).

1.2. Overview:

CF algorithm suggests new items for a particular user based on the user's past behavior (likes and dislikes) and also based on the behavior (history, rating, etc.) of similar users.

Generally, In CF algorithms, there are m users U= {u1, u2, u3…, u_m} and a list of n items I = {i1, i2, i3…, i_n}. Each user u_i has expressed his/her opinions for items list I_{U_i}($I_{U_i} \subseteq I$ and I_{U_i} can be null set). These opinions given by the user can be rating given to a particular item, or previous history of the user (say browsing record) etc.

Item-based collaborative filtering computes the similarities between the item i rated by the active user (u_a) and then selects k most similar items {i_1,i_2, i_3…., i_k} and also computes their corresponding similarities {s_1, s_2, s_3…, s_k}. After finding the list of most similar products, CF predicts the new suggestions by computing the weighted average of the active user's ratings on these similar items.

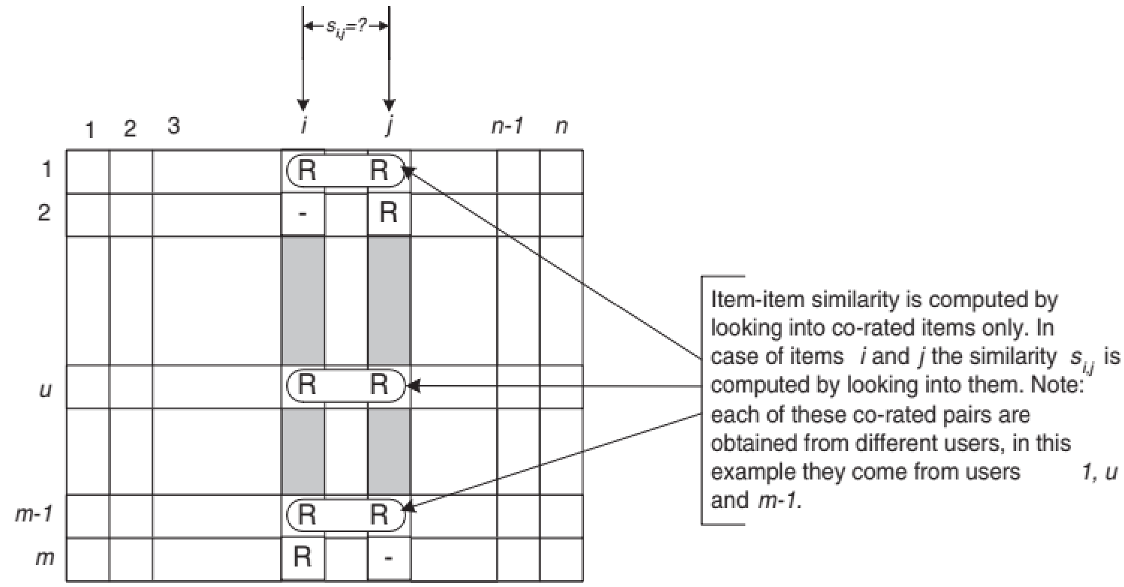
1.2.1. Item Similarity Computation

The most critical step of Collaborative filtering algorithm is to compute similarities between items and then to select the most similar items.

There are many different ways to compute similarities between items. These three methods are cosine-based similarity method, correlation-based similarity method and adjusted-cosine similarity method.

In Figure-1, Rows of matrix represent users and columns represent items.

Figure 1: Isolation of the co-rated items and similarity computation



1.2.1.1. Cosine-based Similarity method:

In this method, two items are considered as m-dimensional vectors in user-space. Cosine of the angle between these two vectors gives the similarity between these two items.

$sim(i, j)$ is given by

$$sim(i,j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{||\vec{i}||_2 * ||\vec{j}||_2}$$

where "·" denotes the dot-product of the two vectors.

1.2.1.2. Correlation-based Similarity method:

In this method, similarity between two items is calculated using Pearson-r correlation $corr_{i,j}$. Let U be the set of users who rated both items (i and j), then the correlation-based similarity is given by:

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R_i})(R_{u,j} - \bar{R_j})}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R_i})^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R_j})^2}}.$$

Here $R_{u,i}$ denotes the rating of user u on item i, $\bar{R_i}$ is the average rating of the i-th item.

1.2.1.3. Adjusted-Cosine Similarity method:

The adjusted cosine similarity works similarly to cosine-based similarity method, but Adjusted-cosine similarity method also takes the differences in rating scale between different users into account. Adjusted-Cosine similarities are calculated by:

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R_u})(R_{u,j} - \bar{R_u})}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R_u})^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R_u})^2}}.$$

Here $\bar{R_u}$ is the average of the u-th user's ratings.

1.2.2. Prediction Computation:

After generating a set of similar items based on similarity measure, the next step is to generate predictions for active user (u_a).
Two methods to generate predictions from the set of generated similar items are:

1.2.2.1. Weighted Sum Method:

This method computes the sum of the ratings given by the user on similar items for any given item i to generate prediction. Each rating between item i and j is computed as per the weighted similarities between these items. The prediction $P_{u,i}$can be denoted as:

$$P_{u,i} = \frac{\sum_{\text{all similar items, N}}(s_{i,N} * R_{u,N})}{\sum_{\text{all similar items, N}}(|s_{i,N}|)}$$

1.2.2.2. Regression:

Regression method computes the sum of approximated ratings instead of directly computing the ratings of similar items. Let $R_i$and R_n be the target item(i) vector and similar item vector respectively, then the linear regression model can be expressed as:

$$\bar{R}'_N = \alpha\bar{R}_i + \beta + \epsilon$$

whereα and β are calculated from R_i and R_N and ε is the error of the regression model.

1.3.Performance:

The CF algorithm, because of its performance bottleneck, is unsuitable for real-time. The CF needs modification to be used in real-time. To scale the CF algorithm, one way is to use a model-based approach. We can do so, by separating neighborhood generation and prediction steps. Generally, items are static for any E-commerce site and thus, item similarities can be precomputed and then perform a quick table lookup to retrieve similarities between items. This method requires $O(n^2)$ space for n items but it saves a lot of time and is fast.

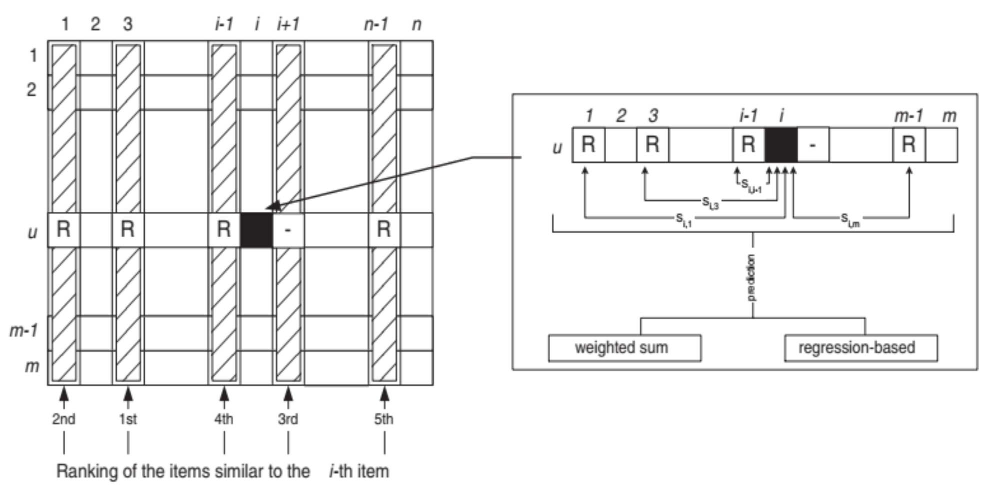


Figure-2: Item-based Collaborative filtering algorithm. The prediction generation process of 5 neighbors is illustrated.

2. Deep Convolutional Neural Networks for Image Classification [19]

2.1. Introduction

Machine learning methods can also be used for image recognition. ML models can also be trained with huge datasets and overfitting can also be reduced using appropriate techniques. For smaller datasets (tens of thousands of images), it performs well. For example, the error rate on the MNIST digit-recognition task is less than 0.3% [20]. But objects in real life are much more complex, so to learn to recognize them requires a very large data set. Their DCNN architecture used LabelMe [21], which consists of hundreds of thousands of fully-segmented images, and ImageNet [22], which consists of over 15 million labeled high-resolution images in over 22,000 categories to train their model.

To learn about this huge (millions of images) dataset, very large learning capacity model is required. DCNN are suitable for such tasks. They can be trained well by varying their depth and breadth. They are also better with making assumptions about images. Therefore, instead of standard feedforward neural networks, similarly sized layered CNN's are easier to train.

2.2. Architecture

Their network consists of 5 convolutional and 3 fully-connected layers and all these layers are important. Removing any layer resulted in drop in performance.

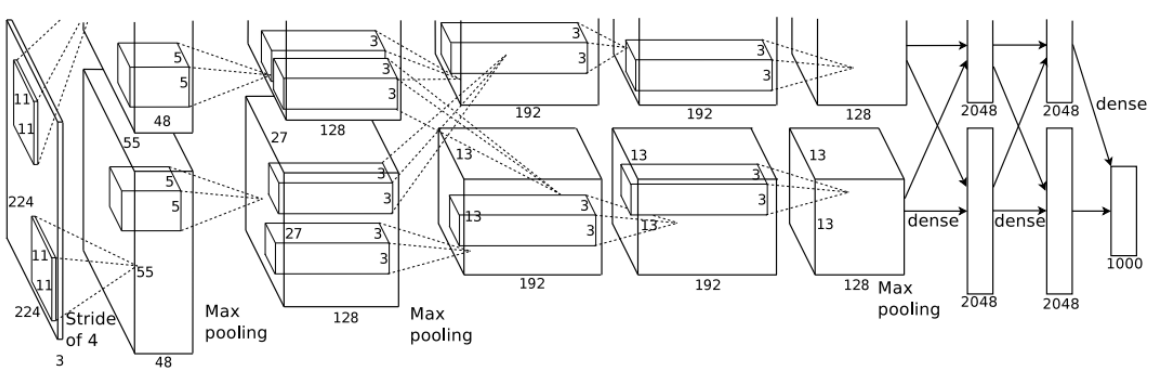


Figure 2: An illustration of the architecture used CNN showing the delineation of responsibilitiesbetween the two GPUs. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

2.3.Features of the Architecture

2.3.1. ReLU Nonlinearity

Their architecture used non-saturating nonlinearity f(x)=max (0, x) along with gradient descent to train their model to reduce their training time. Training time for saturating nonlinearity with gradient descent is much slower than non-saturating nonlinearity. They referred neurons with this non-saturating non-linearity as Rectified Linear Units (ReLUs). Faster Learning rate (Reduced Training time) is very crucial for the performance of the large models which are trained with huge datasets.

2.3.2. Local Response Normalization

RelUs generally don't require normalization to avoid saturation but using Local response normalization was beneficial for their architecture. Their Local response normalization is called "brightness normalization" and reduced their top-1 and top-5 error rates by 1.4% and 1.2% respectively.

Let $a^i_{x,y}$ be the neuron activity at position(x, y) for kernel i. Applying ReLU non-linearity , the Local response normalization activity $b^i_{x,y}$ is given by:

$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^{\beta}$$

where N=total number of kernels, n=adjacent kernel maps, k=2, n=5, $\alpha=10^{-4}$, and $\beta=0.75$.

This normalization makes neurons compete with each other for big activities like real neurons.

2.3.3. Overlapping pooling

Pooling is used for sub-sampling (reduce the dimension of input) and make input features transition from one kernel to another kernel independent. They overlapped pooling and that reduced their top-1 and top-5 error rates by 0.4% and 0.3% respectively.

Test Plan

We have used two datasets for our project. These two datasets are used to provide data to our E-commerce website, Machine Leaning model and DCNN architecture. We used 80-20 methodology to split our data. 80% to train our models and architecture and 20% for testing and analysis process.

1. Data Set

1.1. Flipkart.com Product Dataset

This sub dataset is taken from a pre-crawled extracted data from flipkart.com and is provided by PromptCloud. This dataset is available on Kaggle.com for general public use. We have used this dataset to populate our E-commerce website with products. This dataset has 20,000 products and we took around 500 products to populate our E-commerce to reduce load and faster operations. This dataset has following fields:

- url
- name
- category
- product_id
- retail_price
- discounted_price
- image_url
- description
- rating
- brand
- specifications

uniq_id	crawl_tim	product_url
c2d766ca9	2016-03-2	http://www.flp
7f7036a6	2016-03-2	http://www.flp
f449ec65d	2016-03-2	http://www.flp

1.2. Amazon.com Customer Reviews Dataset

This dataset is extracted from Amazon.com and it contains 5,000 Amazon products reviews and is provided by Datafiniti. This dataset is available on data.world for general public use. We have used this dataset to train our Machine learning model for sentiment analysis that is used by our E-commerce website to classify reviews.
Important fields from this dataset are:

- id
- dateAdded
- dateUpdated
- name
- brand
- categories
- keys
- reviewsDateAdded
- reviewsRating
- reviewsTitle
- reviewsText

id	dateAddec	dateUpdat	name
AVqVGZNv	2017-03-0	2018-10-2	Amazon Ki
AVqVGZNv	2017-03-0	2018-10-2	Amazon Ki
AVqVGZNv	2017-03-0	2018-10-2	Amazon Ki

Conclusion

E-commerce business is growing very rapidly all around the globe. There are a lot of E-commerce websites over the internet but only very few are successful. Success of any E-commerce business website depends on a lot of factors like advertisement, SEO techniques, Product Richness etc. but most important is the features provided by the E-commerce website. Every user expects a very simple but elegant UI design, good recommendations so that he has good options to choose from, better search results and accurate product details. We have taken one step forward by embedding “Reverse Image Search” feature for our E-commerce Business to improve user experience and ease of access.

Future Scope: “Reverse Image Search” architecture can be improved to generate accurate results to improve User Experience. “Reverse Image Search” is a very unique and good feature for any E-commerce website and its proper use can lead to a very good User Experience.

The recommendation algorithm used in this project struggles with sparse data or “cold start”. We have tried our best to improve recommendations for “cold users” but still recommendations given to them largely depends on the responses provided by the user to us through a form but it can be improved by considering the region, ethnicity, and neighbor users. Currently many algorithms are being developed to improve “cold start” problem. We are looking for algorithms being developed and as soon as any algorithm comes, we will try to implement that algorithm in our project.

REFERENCES AND SUGGESTED READINGS

[1]. P. Carey, "The Internet and E-Commerce", ThoroGood (2009).

[2]. US Small Business Administration. Office of Advocacy. E-commerce: Small

Business Venture Online (1999).

[3]. J. Turban and C. King “Introduction to ECommerce”, Prentice Hall (2001).

[4] Simpson, J., & Weiner, E. Recommender. Oxford English Dictionary. Oxford: Oxford University Press (2016).

[5] Akshita, J., and Smita, A. Recommender system: review. International Journal of computer application, 71(24), 38-42 (2013).

[6] Neal Lathiax, Stephen Hailesx, LiciaCaprax, Xavier Amatriainy, “Temporal Diversity in Recommender Systems”, SIGIR’10, Geneva, Switzerland (2010).

[7] A Comparison of Sentiment Analysis Techniques: Polarizing Movie BlogsMichelle Annett and GrzegorzKondrak Department of Computing Science, University of Alberta {mkannett,kondrak}@cs.ualberta.ca (2011).

[8] Linden, G., Smith, B., and York, Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 7(1), 76–80 (2003).

[9] Akshay, J.: A Framework for Modeling Influence, Opinions and Structure in Social Media. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, Vancouver, BC, pp. 1933–1934 (2007).

[10] Durant, K., Smith, M.: Mining Sentiment Classification from Political Web Logs. In: Proceedings of Workshop on Web Mining and Web Usage Analysis of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (WebKDD-2006), Philadelphia, PA (2006).

[11] David G. Lowe. Object Recognition from Local Scale-Invariant Features. In Proc. ICCV. 1150–1157 (1999).

[12] NavneetDalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In Proc. CVPR. 886–893 (2005).

[13] Yushi Jing, David Liu, Dmitry Kislyuk, Andrew Zhai, JiajingXu, Je Donahue, and Sarah Tavel. Visual Search at Pinterest. In Proc. KDD. 1889–1898 (2015).

[14] M. HadiKiapour, Xufeng Han, Svetlana Lazebnik, Alexander C. Berg, and Tamara L. Berg. Where to Buy It: Matching Street Clothing Photos in Online Shops. In Proc. ICCV (2015).

[15] "How does Google's reverse image search work? - Quora". quora.com

[16] "FAQ - TinEye - How does TinEye work?". TinEye.

[17]"FAQ - TinEye - Can TinEye find similar images??". TinEye.

[18] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th international conference on World Wide Web. ACM: 285-295 (2001).

[19]Alex Krizhevsky, IlyaSutskever, and Geofrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In Proc. NIPS. 1106–1114 (2012).

[20] D. Cire¸san, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. Arxiv preprint arXiv:1202.2745 (2012).

[21] B.C. Russell, A. Torralba, K.P. Murphy, and W.T. Freeman. Labelme: a database and web-based tool for image annotation. International journal of computer vision, 77(1):157–173 (2008).

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09 (2009).

[23] M. Nilashi, O.B. Ibrahim, N. Ithnin, Multi-criteria collaborative filtering with high

accuracy using higher order singular value decomposition and Neuro-Fuzzy system, Knowledge-Based Systems 60 (2014) 82–101.

[24] B. Smyth, Case-based recommendation, in: P. Brusilovsky, A. Kobsa, W. Nejdl

(Eds.), The Adaptive Web, Springer, Berlin Heidelberg 2007, pp. 342–376.

[25] S. Middleton, D. Roure, N. Shadbolt, Ontology-based recommender systems, in: S.

Staab, R. Studer (Eds.), Handbook on Ontologies, Springer, Berlin Heidelberg 2009, pp. 779–796.

[26] I. Cantador, A. Bellogín, P. Castells, A multilayer ontology-based hybrid recommendation model, AI Communications 21 (2008) 203–210.

[27] R. Burke, Hybrid web recommender systems, in: P. Brusilovsky, A. Kobsa, W. Nejdl (Eds.), The Adaptive Web, Springer-Verlag, Berlin Heidelberg 2007, pp. 377–408.

[28] B. Mobasher, X. Jin, Y. Zhou, Semantically enhanced collaborative filtering on the web, in: B. Berendt, A. Hotho, D. Mladenič, M. Someren, M. Spiliopoulou, G. Stumme (Eds.), Web Mining: From Web to Semantic Web, Springer, Berlin Heidelberg 2004, pp. 57–76.

[29] B. Smyth, P. Cotter, A personalised TV listings service for the digital TV age, Knowledge-Based Systems 13 (2000) 53–59.

[30] D. Billsus, M. Pazzani, User modeling for adaptive news access, User Modeling and User-Adapted Interaction 10 (2000) 147–180.

[31] D.C. Wilson, B. Smyth, D. O'Sullivan, Sparsity reduction in collaborative recommendation: a case-based approach, International Journal of Pattern Recognition and Artificial Intelligence 17 (2003) 863–884.

[32] D. O'Sullivan, B. Smyth, D. Wilson, Preserving recommender accuracy and diversity in sparse datasets, International Journal on Artificial Intelligence Tools 13 (2004) 219–235.

[33] M. Pazzani, A framework for collaborative, content-based and demographic filtering, Artificial Intelligence Review 13 (1999) 393–408.

[34] A. Bellogin, I. Cantador, F. Diez, P. Castells, E. Chavarriaga, An empirical comparison of social, collaborative filtering, and hybrid recommenders, ACM Transactions on Intelligent Systems and Technology (TIST) 4 (2013) 1–29.

[35] X. Amatriain, A. Jaimes, N. Oliver, J. Pujol, Data mining methods for recommender systems, in: F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Eds.), Recommender Systems Handbook, Springer, US 2011, pp. 39–71.

[36] K. Yu, V. Tresp, S. Yu, A nonparametric hierarchical bayesian framework for information filtering, Proceedings of the 27th Annual International ACM SIGIR

[37] S. Hsu, M.-H. Wen, H.-C. Lin, C.-C. Lee, C.-H. Lee, AIMED — a personalized TV recommendation system, in: P. Cesar, K. Chorianopoulos, J. Jensen (Eds.), Interactive TV: a

Shared Experience, Springer, Berlin Heidelberg 2007, pp. 166–174.

[38] C. Christakou, S. Vrettos, A. Stafylopatis, A hybrid movie recommender system

based on neural networks, International Journal on Artificial Intelligence Tools 16

(2007) 771–792.

[39] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, Z. Chen, Scalable collaborative filtering using cluster-based smoothing, Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,

ACM, Salvador, Brazil 2005, pp. 114–121.

[40] S.K. Shinde, U. Kulkarni, Hybrid personalized recommender system using

centering-bunching based clustering algorithm, Expert Systems with Applications

39 (2012) 1381–1387.

[41] M.A. Ghazanfar, A. Prügel-Bennett, Leveraging clustering approaches to solve the

gray-sheep users problem in recommender systems, Expert Systems with Applications 41 (2014) 3261–3275.

[42] K.-j. Kim, H. Ahn, A recommender system using GA K-means clustering in an online

shopping market, Expert Systems with Applications 34 (2008) 1200–1209.

[43] J. Bobadilla, F. Ortega, A. Hernando, J. Alcalá, Improving collaborative filtering recommender system results and performance using genetic algorithms, KnowledgeBased Systems 24 (2011) 1310–1316.

[44] M.Y.H. Al-Shamri, K.K. Bharadwaj, Fuzzy-genetic approach to recommender systems based on a novel hybrid user model, Expert Systems with Applications 35

(2008) 1386–1399.

[45] A. Zenebe, A.F. Norcio, Representation, similarity measures and aggregation

methods using fuzzy sets for content-based recommender systems, Fuzzy Sets

and Systems 160 (2009) 76–94.

[46] R.R. Yager, Fuzzy logic methods in recommender systems, Fuzzy Sets and Systems 136 (2003) 133–149.

[47] J. Zhan, H. Chia-Lung, I.C. Wang, H. Tsan-Sheng, L. Churn-Jung, W. Da-wei, Privacy-preserving collaborative recommender systems, IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews 40 (2010) 472–476.

[48] Y. Cao, Y. Li, An intelligent fuzzy-based recommendation system for consumer electronic products, Expert Systems with Applications 33 (2007) 230–240.

[49] C. Cornelis, J. Lu, X. Guo, G. Zhang, One-and-only item recommendation with fuzzy logic techniques, Information Sciences 177 (2007) 4906–4921.

[50] C. Porcel, A.G. López-Herrera, E. Herrera-Viedma, A recommender system for research resources based on fuzzy linguistic modeling, Expert Systems with Applications 36 (2009) 5173–5183.

[51] D. Ben-Shimon, A. Tsikinovsky, L. Rokach, A. Meisles, G. Shani, L. Naamani, Recommender system from personal social networks, Advances in Intelligent Web Mastering, Springer, 2007. 47–55.

[52] C.-N. Ziegler, G. Lausen, Analyzing correlation between trust and user similarity in online communities, Trust Management, Springer, 2004. 251–265.