APPAREL RECOMMENDATION SYSTEM USING DEEP LEARNING

Alla Sri Swarna Devi*1, Dr. Marepalli Kamala Kumari2

*1M.Tech (CST), Adikavi Nannaya University, Rajamahendravaram, Andhra Pradesh, India
2Assoc.Professor, Adikavi Nannaya University, Rajamahendravaram, Andhra Pradesh, India

Abstract : Now a days, the internet has been well known as a big data repository consisting of a variety of data types as well as large amount of unseen informative knowledge which can be discovered via a wide range of data mining or machine learning models. Although the development of e-commerce markets results in likely development of search engines, users are still facing a problem with accurate results. Instead, to achieve this problem recommendation engines are mostly helpful. Most e-commerce sites are developing recommendation systems analyze a large amount of transaction data without having any idea of what the items in the transactions mean or what they say about the customers who purchased or browsed items. Apparel fashion recommendation engine that utilizes deep convolutional neural networks using Amazon API to suggest products and provide customers with information to help them find the products. Utilizing deep neural networks enable us to parse such images into a high dimensional feature representation that allows us to recommend a pair from user preference tensor.

***IndexTerms* - Neural networks, Recommendation systems, Apparel, Machine learning, Content-based recommendation system, Collaborative recommendation system, Hybrid recommendation system.**

I. INTRODUCTION

The idea of a recommendation system was first proposed in the mid of 1990's. Based on users' behavior the recommendation systems suggest interesting products. Due to the rapid growth of e-commerce, recommendation systems are playing a vital role and even arrested attention from e-business companies like Amazon.com. [1]

1.1 Recommendation Algorithms

Most recommendation algorithms start by finding a set of customers whose purchased and rated items overlap the user's purchased and rated items. The algorithm aggregates items from these similar customers, eliminates items the user has already purchased or rated, and recommends the remaining items to the user. Two popular versions of these algorithms are collaborative filtering and cluster models. Other algorithms including search-based methods and our own item-to-item collaborative filtering focus on finding similar items, not similar customers. For each of the user's purchased and rated items, the algorithm attempts to find similar items. It then aggregates the similar items and recommends them. [1]

1.2 Traditional Collaborative Filtering

A traditional collaborative filtering algorithm represents a customer as an N-dimensional vector of items, where N is the number of distinct catalog items. The components of the vector are positive for purchased or positively rated items and negative for negatively rated items. [1] To compensate for best-selling items, the algorithm typically multiplies the vector components by the inverse frequency (the inverse of the number of customers who have purchased or rated the item), making less well-known items much more relevant. For almost all customers, this vector is extremely sparse. The algorithm generates recommendations based on a few customers who are most similar to the user. It can measure the similarity of two customers, A and B, in various ways; a common method is to measure the cosine of the angle between the two vectors: [2]

$$similarity(\vec{A}, \vec{B}) = \cos(\vec{A}, \vec{B}) = \frac{\vec{A} \bullet \vec{B}}{\|\vec{A}\| * \|\vec{B}\|} \quad (1)$$

1.3 Cluster Models

To find customers who are similar to the user, cluster models divide the customer base into many segments and treat the task as a classification problem. The algorithm's goal is to assign the user to the segment containing the most similar customers. It then uses the purchases and ratings of the customers in the segment to generate recommendations. The segments typically are created using a clustering or other unsupervised learning algorithm, although some applications use manually determined segments. Using a similarity metric, a clustering algorithm groups the most similar customers together to form clusters or segments. Because optimal clustering over large data sets is impractical, most applications use various forms of greedy cluster generation. These algorithms typically start with an initial set of segments, which often contain one randomly selected customer each. They then repeatedly match customers to the existing segments, usually with some provision for creating new or merging existing segments. [3]

II. RELATED WORK

2.1 Exploration of Lukas et al. apparel recommendation system

Lukas et al. mostly focused on the task of classification which is the issue of describing what type of attire is worn in an image. The work is related to visual attributes which are also gained leading significance and their core method consists of a multi-class learner based on Random Forest for type classification and Super Vector Machines for attribute classification.

Their apparel classification mechanism is composed of two parts:

1. One part recounts the overall style/variety of clothing.

2. Another be made up of the attributes of the style such as “black”,” synthetic”.

By amalgamating the outputs of these parts the system comes up with the comprehensive elucidation of the clothing style. This paper did not focus on similarity searching or clothing segmentation, but only on classification, i.e., the task of describing what type of clothing is worn in an image. Their work is only related to learning visual attributes which have been applied in color and pattern naming. [4]

2.2 Exploration of ziwei et al. recommendation system

Ziwei et al. proposed a new deep model, namely fashion-Net, which learns clothing characteristics by jointly predicting landmarks and clothing attributes. Their large-scale experiments reveal the usefulness of deep fashion and effectiveness of fashion-Net. They investigated multiple building blocks of proposed fashion-Net and also summarized the performance of various methods on category classification and attribute prediction. The performance drops of 6~9 percent are observed in this model when the clothing landmarks in fashion-Net are replaced with human joints and pose lets. This model only achieves an accuracy of 76.4%. Using human landmarks instead of clothing landmarks could have helped in increasing the accuracy of the model. [4]

2.3 Exploration of xiaosong et al. recommendation system

Xiaosong et al. Introduced a hybrid recommender system for uncomplicated clothes shopping. This system recommends clothes in terms of clothing features and user ratings. Experiments in their simulation environment show that the recommender well pleases the needs of customers.

- Firstly, this system applies human detection techniques to perceive the clothes field in an image and then analyzes and calculates the percentage of each color.

- Secondly, by basing on the extended rating matrix it targets the product to be endorsed are selected.

As the recommendation system based on users past preferences, and ratings, the cold start problem occurs in this type of systems when a new user or item enters the system. In hybridization strategies, there is an adaptive switching of weights based on the user model, context and meta-features which is a complex task. [4]

2.4 Exploration of Wenhui et al. aesthetic based recommendation system

Wenhui et al. proposed a novel model trained with coupled matrices. Then combine it with additional image features and term the method as Dynamic Collaborative Filtering model with Aesthetic features. Aesthetic information which is most pertinent with user preference, into clothing recommender systems. They extracted the aesthetic features by a pre-trained neural network trained for the aesthetic assessment task which is a brain-inspired deep structure. They investigated the serviceability of aesthetic features for personalized recommendation and feedback data set. The extremely large and sparse matrix used for filtering could bring out the challenges in the performance of recommendation. In this model, we first propose a dynamic collaborative filtering model and then a dynamic collaborative filtering model with aesthetic features, these extensive experiments consumes more time. [4]

2.5 Performance analysis

The recommendation systems have been a favored topic ever since the omnipresence of the internet made it comprehensible that people from various backgrounds would be able to query and access the underlying data. The various machine learning technologies used for filtering and can predict that whether the user like a recommended resource.

- One of the major aspects of a recommendation system is that according to the users" preference it can personalize the website for a user by proposing the items and suggests the top products to the user by predicting the rating that a user would give to a product. This has a number of engrossing applications which includes online advertising, helps e-commerce websites by attracting more customers to buy more products.

Various algorithms have been executed and explored to drive peak conversion rate versus non-personalized product recommendations. There exist more advanced and traditional methods to boost up recommendation process. Namely, deep learning, machine learning, neural networks, and social learning. These Cognitive computing methods can take the worth of your recommendation systems to next level. [4]

AUTHOR	METHODS CONSIDERED	DATA SET	ACCURACY
Lukas et al. Recommendation system	SVM's, Random Forest	80,000	41.36%
Ziwei et al. Recommendation system	Neural Networks	8,00,000	76.40%
Wenhui et al. Recommendation system	Neural Networks	2,50,000	80%
Xiaosong et al. Recommendation system	Group Rating Matrix	1783	82%

Table –1: Performance Analysis

III. HOW RECOMMENDATION WORKS

Rather than matching the user to similar customers, item-to-item collaborative filtering matches each of the user's purchased and rated items to similar items, then combines those similar items into a recommendation list. To determine the most-similar match for a given item, the algorithm builds a similar-items table by finding items that customers tend to purchase together. [1] We could build a product-to-product matrix by iterating through all item pairs and computing a similarity metric for each pair. However, many product pairs have no common customers, and thus the approach is inefficient in terms of processing time and memory usage. The following iterative algorithm provides a better approach by calculating the similarity between a single product and all related products:

For each item in product catalog, I_1

For each customer C who purchased I_1

For each item I_2 *purchased by customer C*

Record that a customer purchased I_1 *and* I_2

For each item I_2

Compute the similarity between I_1 *and* I_2

It's possible to compute the similarity between two items in various ways, but a common method is to use the cosine measure we described earlier, in which each vector corresponds to an item rather than a customer, and the vector's M dimensions correspond to customers who have purchased that item. This offline computation of the similar-items table is extremely time intensive, with O (N2 M) as worst case. In practice, however, it's closer to O (NM), as most customers have very few purchases. Sampling customers who purchase best-selling titles reduces runtime even further, with little reduction in quality. Given a similar-items table, the algorithm finds items similar to each of the user's purchases and ratings, aggregates those items, and then recommends the most popular or correlated items. This computation is very quick, depending only on the number of items the user purchased or rated. [1][5]

IV. DATASET AND FEATURES

For building the recommendation system, we use Amazon product image data, spanning May 1996 July 2014, which includes 9.4 million products. Excluding the ones that lack images, we collected a dataset of 1,80,000 products, with different brands. The detailed information of each image contains:

S.NO	FEATURE	DESCRIPTION
1	Asin	Amazon standard identification number- ID of the product, ex: 0000031852
2	Brand	Brand to which the product belongs to
3	Colour	Colour information of apparel, it can contain many colorus as a Values ex: red and black stripes
4	Product_type_name	Type of the apparel ex: SHIRT/TSHIRT
5	Medium_image_url	url of the image
6	Title	Title of the product
7	Formatted_price	Price of the product

Table -2: Dataset Features

Fig -1. Examples of the data.

V. WORKING METHODOLOGY

5.1 Working Procedure

There are two major problems that we want to solve in our project. First, determine the category that a given image belongs to; second, find and recommend the most similar products according to the given image. Since our project is mainly based on convolutional neural network, we would first introduce common used convolutional neural network layers.

a) CNN Layers

The most important step of CNN is Convolutional (Conv) layer. As we can see from Figure 2 that conv layer would translate small rectangle of input layer into a number of output layer using matrix multiplication.

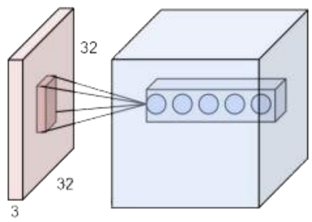

Fig -2. Conv layer

Pooling layers are similar to convolutional layers except that it would use non-parameter method to transform small rectangle into a number. Max pooling are commonly used in CNN, which would output the maximum number in the rectangle of input layer. [6]

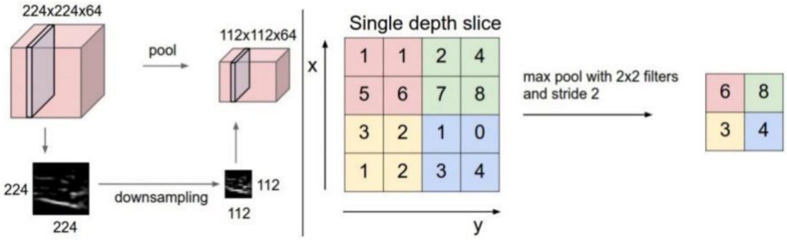

Fig -3. Pooling layer

b) Classification

In this step, we would like to classify an input image into one of the 20 categories. We construct Alex Net and VGG model for the classification task.

AlexNet: A deep convolutional neural network classification model proposed by. As we can see, (Figure 3) AlexNet model first contains 2 convolutional layers with max pooling and batch normalization; then there are 3 convolutional layers with separated feature; one max pooling before three fully connected layers.

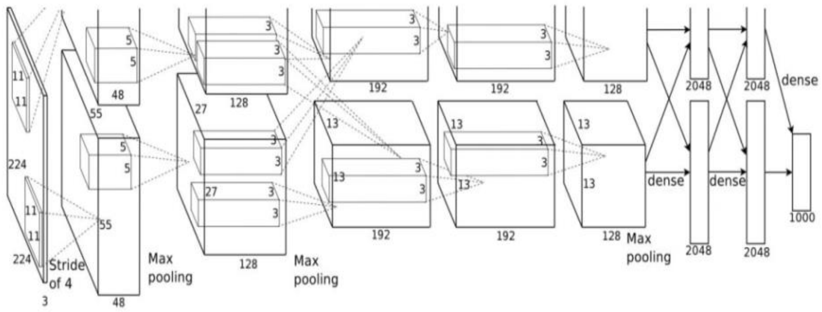

Fig -4. AlexNet Model

The original model was trained to classify images in the ImageNet LSVRC-2010 content, where there were 1000 categories. Since our problem only contains 20 categories, we change the last fully connected layer to 4096×20. To save time, we use the pre-trained weights of the first 5 neurons and train the last three fully connected layers. VGG: a deep convolutional neural network classification model proposed by. As showed below (Figure 4), VGG contains 13 convolutional layers with max pooling every 2 or 3 convolutional layers; then 3 fully connected layers and softmax as the final layer. The original model was trained to classify images in the ImageNet ILSVRC-2014 content, where there were 1000 categories. We change the last fully connected layer to 4096 × 20. We also utilize the pretrained weights as initialization of parameters and to train the last three fully connected layers. We also add batch normalization layers after the activation functions in the first two fully connected layers. [6]

5.2 Feature Extraction using BOW: TF-IDF

Term frequency-Inverse document frequency uses all the tokens in the dataset as vocabulary. Frequency of occurrence of a token from vocabulary in each document consists of the term frequency and number of documents in which token occurs determines the Inverse document frequency. What this ensures is that, if a token occurs frequently in a document that token will have high TF but if that token occurs frequently in majority of documents then it reduces the IDF ,so stop words like an, the, i which occur frequently are penalized and important words which contain the essence of document get a boost. Both these TF and IDF matrices for a particular document are multiplied and normalized to form TF-IDF of a document.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right) \tag{2}$$

For the recommendation step, we use the last fully connected layer in our classification model as feature vectors of images. For any images in the dataset, there will be one corresponding feature vector. And this feature vector will be the input for our recommendation model. The work flow of this step is shown in the following bullets. [6]

Fig -4. VGG model

• **Feature extraction:** the classification model is used to identify which category the target image belongs to. Then we extract the input from last fully connected layer of classification model as features.

• **Input of the model:** the feature vector of the target image extracted in the above.

• **Similarity calculation:** using different measures to calculate similarity scores between feature vector of target image and feature vectors of all images in the target category to measure similarity between image pairs.

We have tried L2 distance, cosine distance and neural network models to compute the similarity scores. For two different images i and j, the L2 distance score is defined as

$$s_{L2} = ||v_i - v_j||_2 \quad (3)$$

Where vi, vj ∈ R l are the two corresponding feature vectors, and l = 4096 is the length of feature vectors. The smaller the score sL2 is, the more similar the two images are.

The cosine distance score is defined as

$$s_{cosine} = \frac{v_i^\top v_j}{||v_i||||v_j||} \quad (4)$$

The larger the score s_{cosine} is, the more similar the two images are.

$$\mathbf{h1 = f(v \cdot W1 + b1)}$$
$$\mathbf{h2 = f(h1 \cdot W2 + b2)}$$
$$\mathbf{s_{model} = sigmoid(h2 \cdot W3 + b3)} \quad (5)$$

Where v = [v1, v2] ∈ R l×2 is obtained by concatenating two feature vectors. f(x) = max(0.01x, x) is the leaky ReLU function. The first layer can be treated as a 1-d convolution layer with Leaky ReLU as the activation function and W1 ∈ R 2 and b1 ∈ R as parameters. The second layer is a fully-connected layer with Leaky ReLU as the activation function and W2 ∈ R l and b2 ∈ R as parameters. The output layer is a linear transformation with the sigmoid function as activation function. The larger the score smodel is, the more similar the two images are. There is no easy way to define similarity score purely based on the image pixels. Fortunately, the input images has a corresponding title describing the product. To characterize how similar two images are, we use the Jaccard similarity of two sets of tokens in the titles of two images as the similarity of two images. The Jaccard similarity of two set A and B is defined as

$$s_{Jaccard} = \frac{|A \cap B|}{|A \cup B|} \quad (6)$$

• **Output:** Top k images (products) that are most similar to the target image.

For the recommendation task, we trained a neural network model described in above. In our outfit recommendation system, we give the input image to the system. Then the system generates several outfits randomly, and these generated outfits are put into our model to get the judgments ranging from 0 to 1. After sorting these judgments we can get the top 3 good outfits.

Fig -4. Input

Fig -5. Recommendations (Output)

VI. CONCLUSION

Recommendation algorithms provide an effective form of targeted marketing by creating a personalized shopping experience for each customer. For large retailers like Amazon.com, a good recommendation algorithm is scalable over very large customer bases and product catalogs, requires only sub-second processing time to generate online recommendations, is able to react immediately to changes in a user's data, and makes compelling recommendations for all users regardless of the number of purchases and ratings. Unlike other algorithms, item-to-item collaborative filtering is able to meet this challenge.

In the future, we expect the retail industry to more broadly apply recommendation algorithms for targeted marketing, both online and offline. While e-commerce businesses have the easiest vehicles for personalization, the technology's increased conversion rates as compared with traditional broad-scale approaches will also make it compelling to offline retailers for use in postal mailings, coupons, and other forms of customer communication.

REFERENCES

[1] G. Linden, B. Smith and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," in *IEEE Internet Computing*, vol. 7, no. 1, pp. 76-80, Jan.-Feb. 2003.

[2] Co-selling recommendations from an eCommerce dataset. Pawan Rewari – SUID 01421653 – dadrewari@gmail.com CS224W – Project Milestone Update

[3] Gaudani, Hetal. (2014). A Review Paper on Machine Learning Based Recommendation System. Development. 2. 3955-3961. J. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," Proc. 14th Conf. Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1998, pp. 43-52.

[4] Devi, Swarna & Raghuvamshi, Archana. (2018). "Exploration of apparel recommendation systems using deep learning". 5. 41-44.

[5] https://www.analyticsindiamag.com/know-will-buy-next-power-ai-machine-learning/ P.S. Bradley, U.M. Fayyad, and C. Reina, "Scaling Clustering Algorithms to Large Databases," Knowledge Discovery and Data Mining, Kluwer Academic, 1998, pp. 9-15.

[6] Image-based Product Recommendation System with Convolutional Neural Networks Luyang Chen Stanford University 450 Serra Mall, Stanford, CA lych@stanford.edu Fan Yang Stanford University 450 Serra Mall, Stanford, CA fanfyang@stanford.edu Heqing Yang Stanford University 450 Serra Mall, Stanford, CA heqing@stanford.edu