Social Network Recommendation System

Project proposal for University of Virginia SYS 6014 Decision Analysis Spring 2020

1st Zheng Huang
zh4zn
University of Virginia
March 23 2020

***Abstract*—This project seeks to match 2 people on a social network based on their social network activities, especially on image posts. Our program uses neural networks to classify users' image posts and gets labels on images. By providing these image information, the model calculates the similarity of two users by collaborative filtering, and outputs a score representing if they can be matched as friends.**

***Index Terms*—Python, Neural networks, Collaborative filtering, Recommendation systems, Decision analysis**

I. Introduction

The match and recommendation of friends on social networks is a burgeoning field of machine learning with great potential for application. As hardware, which was previously prohibitively expensive, becomes cheaper, the ability to make use of GPU in novel and interesting ways becomes greater. On the field of social network, people post a large number of information, from texts, image to video. Based on text, we can recommend and match friends easily in that we can decompose the text and acquire keywords. However, if a user only upload pictures on their social network, it is almost to difficult to extract keywords by the same method as text decomposition. Therefore, we propose a method to get "keywords" from image a user posted and recommend friends to him or her. Thus, the decision maker will be our recommendation model which will make decision on whether to match two users with given keywords. If the two user accept the recommendation, we can say that the model does an ideal decision. If the recommendation is declined, we can say our model makes the decision badly. When a user receives friends recommendation that he or she does not like, our user may feel disappointing and leave our social network. This is a stake we may face. To lead to better decisions, we can optimize our recommendation algorithms and improve the performance of extracting image keywords. With this insight, we can create our tool by two part:(1) image classification process (2)Friends recommendation process.

II. Model of the decision problem

Based on the pictures one user upload on their social network, we implement neural network over them to acquire keywords of the pictures. In my project, for example, I want to train a computer to understand an image. To be more specific, by providing data and images as input, we can use computers to recognize objects, people, animals, places, etc. Some application of my model can be used by people who want to make friends, for example, users can upload an image or a photo to our model, then the model will classify and create a label for it. Moreover, it will match people who with the same label.

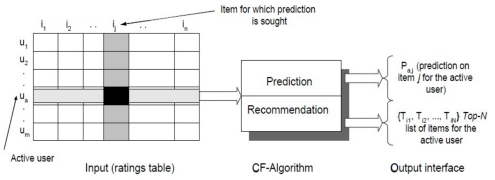

Fig. 1: Process of CF

By the label or keywords, we can use Collaborative Filtering(CF)[1] to implement friends recommendation algorithm. The basic idea of CF is to recommend items to users based on their previous preferences and the choices of other users with similar interests. The process of CF is shown on figure1. To be more specific, we generate a row vector for the user. For more users, we combine each row vector and generate a interest matrix as shown on figure 2. Then we can do matrix decomposition. The result of it is shown on figure3. For relevant payoffs, the model will match 2 people in terms of meeting successful or unsuccessful

To further improve the performance of the recommendation model, we can use three methods to calculate the similarity between two users. (1)Cosine-based similarity (2)Correlation-based similarity (3)Adjusted Cosine similarity.

III. The predictive model

From the predictive tools, we can learn about what will happen in the future based on historical data. In the case of images, our prediction model can potentially tell you the type, the label of a certain image. To generate a useful prediction, we need to know the process of classification. By comparing the result produced by our model, we can get feedback data. This is data you collect when the prediction machine is operating in real situations. Feedback data is often generated from a richer set of environments than training data. To some extent feedback data is more realistic than the data used for training.

User / Item	Batman	Star Wars	Titanic
Bill	3	3	
Jane		2	4
Tom		5	

Fig. 2: Interest matrix

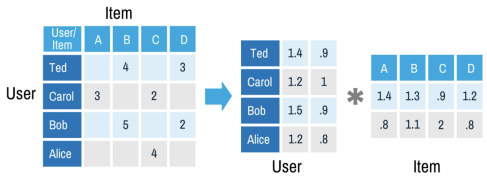

Fig. 3: Matrix decomposition

So, you can improve the accuracy of predictions further with continual training using feedback data.

The decision-maker's beliefs about uncertain parameter values can be sharpen with Gradient Descent where we computer derivatives of loss function with respect to W and b. Then we update the parameters according to the following rules:

$$w_{ij} = wij - \lambda \frac{\partial \ell}{\partial w_{ij}}$$

$$b_i = b_i - \lambda \frac{\partial \ell}{\partial b_i}$$

.

IV. Data generating process

I will use SUN20 dataset. The whole dataset was split into two parts: Training set and validation set. The validation set has 10k images. The classifier will take input vectors x of size 1x28x28=784 , which is the size of the images. The function representing the linear function is:

$$\hat{a} = softmax(Wx + b)$$

,

$$\hat{a}_i = \frac{exp(Wx + b)}{\sum_{k=1}^{10} exp(Wx + b)}$$

To be more specific

$$\hat{a}_i = W_{i1}x_{i1} + W_{i2}x_{i2} + ... + W_{ij}x_{ij} + b_i$$

W and b can be initialized with random values.

$$W_{ij} \in \mathbb{N}(\theta_1, \sigma_1), b \in \mathbb{N}(\theta_2, \sigma_2)$$

V. Results

I will use $\theta \in (0, 1)$ to represent the potential matches of people. If $\theta > 0.7$ we can make a prediction that the 2 people can be matched with this method.

References

[1] Francesco Ricci and Lior Rokach and Bracha Shapira, Introduction to Recommender Systems Handbook, Recommender Systems Handbook, Springer, 2011, pp. 1-35