Social Network Recommendation System

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

1st Zheng Huang
zh4zn
University of Virginia
April 25 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 [2]to classify users' image posts and gets labels on images. By providing these image information, the model will make an initial recommendation decision.Our model not only consider the similarity of two users, but take our client's choice into consideration.**

***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. In the field of social network, people post large numbers of information, from texts, images to videos. Based on the 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 the image a user posted and recommend friends to him or her. Thus, the decision-maker will be our recommendation model which will make a decision on whether to match two users with given keywords. If the two users 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 a 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. Our model has two kinds of recommendation methodologies: (1)Accuracy base recommendation method (2)User-based method. According to these methods, our model will help users make the decision of selecting a friend he or she may be interested in the most. The procedure of making the final decision is as follows. (1)Our recommendation model will return a prior decision based on other users' image information. (2) Our model will update the decision by allowing a user to choose which friends recommended he or she would like to choose based on the similarity of different users. (4) The next step of the decision will focus on the user's choice then the model will be updated and return a new decision. (5) These two results of decisions will be compared. (6)Our final decision of friend recommendation can be made.

In terms of evaluation, we simulate the procedure of making a decision by introducing three users, each of them contains 4 images, which reflect their interest, extracted from the data set. The final decision is made based on this information.

II. Model of the decision problem

Our model is motivated by previous recommendation systems in various other tasks and domains. Some previous work only focuses on prediction making, which ignores clients' preferences. Our decision model allows clients' interaction as an input, and the model will be further updated than make a final decision.

The decision model can be composed with an action set, payoff function, prior belief, intermediate decision and final decision.

The final decision of our model will be a user from an action set: user list U, U $\in \{user1, user2, user3, ...\}$. The input of the model is a data set with images and labels. To make a better decision, we adopt and further develop the idea of neural networks, than combine it with our decision model. Based on the pictures one user upload on their social network, we implement neural networks, as shown in figure 1, over them to acquire keywords of the pictures. In my project, for example, we train a computer to understand an image. To be more specific, by providing labels and images as input, we can use computers to recognize objects, people, animals, places, etc. Some applications 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. According to the classification, a preference matrix $M_u, u \in \{user1, user2, user3, ...\}$is generated for each user. Given these matrices, the model will calculate the similarity of each two matrices. Previous efforts on achieving recommendation by Collaborative Filtering[2]. 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 in figure2. To be more specific, a row vector can be generated for one user. For each user, we combine each row vector and generate an interest matrix as shown in figure 3. Then we can do matrix decomposition. The result of it is shown in figure4. Inspired by this idea, our model acquire the similarity $S_{ij}(i, j \in \{1, 2, 3\})$of each user by calculation Euclidean distance of two matrices $m_i, n_j (i, j \in \{1, 2, 3, ...\})$:

$$d(m_i, n_j) = \sqrt{(m_{ab} - n_{xy})^2},$$

where m_{ab}, n_{xy} means each elements within matrix m_i and n_i. According to the similarity matrix, the initial decision will be made. However, this decision result did not focus on the users' interaction. For example, in real life, a user can accept or dismiss a recommendation did by a recommendation system. We posit a payoff function that amplifies this user-software interaction.

$$MatchingScore = w1 \times S + w2 \times M + w3 \times A, \quad (1)$$

where variable S means the similarity of two users. Besides, M means the matching rate of our clients, variable A means the accuracy of the prediction model, and w1, w2, w3 are weights. For S, this variable determines how many users our model will recommend to the client. For example, with a larger S, the users recommended will have a high similarity with respect to our client, thus fewer users are recommended. However, with a smaller S, the model will recommend more users to clients, which means our client will have more users to select. For matching rate

$$M = \frac{RS}{NR},$$

where RS means the recommendation a client selected, NR means the total number of users recommended. During each iteration, our client can choose to accept or dismiss the recommendation. If the client dismisses the recommendation, M will equal to 0. In the payoff function, variable A means the accuracy of the recommendation model, especially the accuracy of different neural network models that extract image labels. To update the payoff function and make a final decision, we save each matching score given every iteration, after all iteration finished, our model will return a final decision according to the highest matching score. Therefore, the utility function of our model is

$$U = argmax(MatchScore).$$

The detailed updating procedure will be discussed in Section: The Predictive Model.

III. The Predictive Model

A. Neural Network

Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must

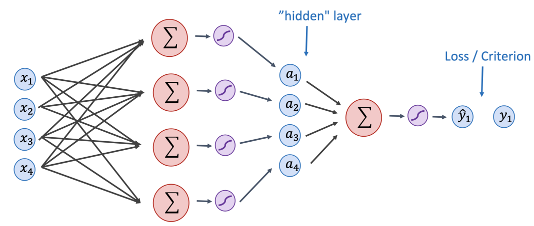

Fig. 1: Neural Network Structure

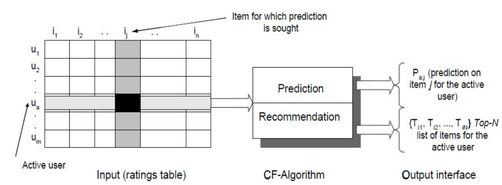

Fig. 2: Process of CF

be translated. The structure of neural networks composed of several layers. The layers are made of nodes. A node is just a place where computation happens, loosely patterned on a neuron in the human brain, which fires when it encounters sufficient stimuli. A node combines input from the data with a set of coefficients, or weights, that either amplify or dampen that input, thereby assigning significance to inputs with regard to the task the algorithm is trying to learn; e.g. which input is most helpful is classifying data without error? These input-weight products are summed and then the sum is passed through a node's so-called activation function, to determine whether and to what extent that signal should progress further

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

Fig. 3: Interest matrix

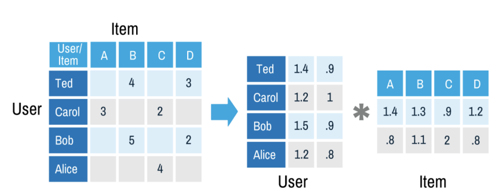

Fig. 4: Matrix decomposition

through the network to affect the ultimate outcome, say, an act of classification. If the signals pass through, the neuron has been "activated." A node layer is a row of those neuron-like switches that turn on or off as the input is fed through the net. Each layer's output is simultaneously the subsequent layer's input, starting from an initial input layer receiving your data.

In our model, we use neural networks to recognize images and distribute a label for each of them. A basic and effective neural I implement for our model is shown in figure 5. To achieve the goal of making a decision based on different prediction accuracy, we also introduced a more complicated model to finish the task of image recognition. The neural networks are trained by SUN20 data set. Firstly we extract images and labels from the data set and transfer each image as tensor which is a high-dimensional matrix than pass the tensor into our neural network. The output of the neural network is a vector [1]. The input and output structure is shown in figure 6. We can assume the training schedule as linear combinations and trying to find appropriate weights and biases to make the output and ground-truth as similar as possible. The procedure of training a neural network is shown in figure 7. To make a prediction, we implement a Softmax Classifier

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

where $\hat{y}$ is a prediction value which will be compared with ground-truth. This classifier is applied to each output element and makes a prediction that which label this input belongs to. After assume a = Wx + b, the Softmax function can be written as

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

Above all is the forward procedure of how the model attribute label for each input image.

To define a criterion for how is our prediction look like, we implement Cross Entropy Loss. Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label [5].

$$\ell(y, \hat{y}) = -\sum_i y_i log(\hat{y}_i).$$

To minimize the loss function, we introduce the methodology called gradient descend. Gradient descent is one of the most popular algorithms to perform optimization and by far the most common way to optimize neural networks. It is an iterative optimization algorithm used to find the minimum value for a function. Gradient Descent [3] The Algorithm helps us to make these decisions efficiently and effectively with the use of derivatives. Then pass the gradient or derivatives back to the model to update weights and bias, which called backpropagation. Within our neural network model, for each node, we save corresponding gradient and sharpen them by the equation:

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

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

where λ is a hyper-parameter which always assigned a small value. Each step of our training composed of the combination of the forward pass and backpropagation. By repeating the step with a certain epoch, we can get an ideal model for image prediction.

B. Reducing Model Uncertainty

According to the prediction of neural networks, a similarity matrix will be derived by calculating the Euclidean Distance between two preference matrices. An initial prediction will be made given the similarity matrix. However, this is not enough to make a final decision. In an effort to generate an ideal decision, we introduce the term Matching Rates M, Accuracy of Model A and Similarity of Users S. The goal of introducing this variable is to reduce the bias of making a decision only by similarity. Combined with weights assigned to each variable, we acquire our payoff function as shown on formula (1). For the Similarity of Users $S \in \{s_1, s_2, s_3\}$, this variable can be seen as a threshold that determines how many users the model will recommend to a client. A lower value of S means the model will recommend more users compared with a high value of S. From high to low values of S, the client can choose which recommended users he or she likes the most, which introduces the variable Matching Rate M. This variable counts the percentage of selected users with respect to all recommended users. The idea behind the variable M is to encourage the client to interact with our model. During each step of recommendation, the client is free to choose which user he or she wants to select or even give up this recommendation can check the next recommendation. According to the client's input, if a recommendation is declined, which means the model made a wrong or bad decision, the Matching Score will be severely influenced and get a low value, which means this recommendation will be a low priority within final decision list. To achieve that, among weights $W \in \{w_1, w_2, w_3\}$, we assign the highest weight as the coefficient of M. Beyond client's interaction, a variable representing model accuracy A is taken into account. In the sense that we cannot guarantee the similarity matrices generated by different neural networks on the validation set is unmistakable, and the similarity matrices will influence the recommendation result, thus we should take variable A into consideration. A is derived by the number of right labels over the number of all labels. After iterations of updating the payoff function with respect to each neural model, our model will return a final decision given the highest Matching Score.

IV. Data generating process

We extract the training set and validation set with 20 categories from SUN database. The goal of the SUN database project is to provide researchers in computer vision, human perception, cognition and neuroscience, machine learning and data mining, computer graphics and robotics, with a comprehensive collection of annotated images covering a large variety of environmental scenes, places and the objects within [4].

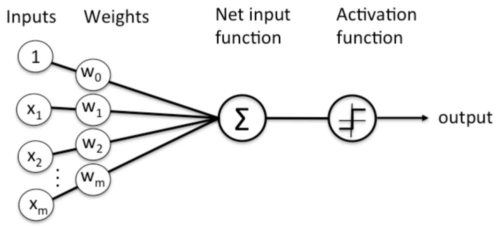

Fig. 5: Neural network structure

Training Data

inputs	targets / labels / ground truth	predictions
$x_1 = [x_{11} \;\; x_{12} \;\; x_{13} \;\; x_{14}]$	$y_1 = [1 \;\; 0 \;\; 0]$	$\hat{y}_1 = [0.85 \;\; 0.10 \;\; 0.05]$
$x_2 = [x_{21} \;\; x_{22} \;\; x_{23} \;\; x_{24}]$	$y_2 = [0 \;\; 1 \;\; 0]$	$\hat{y}_2 = [0.20 \;\; 0.70 \;\; 0.10]$
$x_3 = [x_{31} \;\; x_{32} \;\; x_{33} \;\; x_{34}]$	$y_3 = [1 \;\; 0 \;\; 0]$	$\hat{y}_3 = [0.40 \;\; 0.45 \;\; 0.15]$
$\vdots$		
$x_n = [x_{n1} \;\; x_{n2} \;\; x_{n3} \;\; x_{n4}]$	$y_n = [0 \;\; 0 \;\; 1]$	$\hat{y}_n = [0.40 \;\; 0.25 \;\; 0.35]$

Fig. 6: Input and output structure

V. Experiments

According to the first neural network model, we extract client preference as "User", recommended friends as "User1", "User2" and "User3". Given the result of validation accuracy, variable A, which means the accuracy of the prediction model, can be assigned to 0.2. The result is shown as figure8. The list under each user and the client shows the class of images they post on their social network. The similarity is shown at the bottom of this figure. Each of the similarities between a certain user to the client is 10, which is the highest score, thus we do not need to set variable S in payoff function. The number in front of each prediction is the index of the image's true label value. We can see from the figure that the prediction accuracy is not an ideal one. From the result, intuitively, as a client, we can assume that the neural network may make a wrong prediction in that the neural network predicts each image as an airport terminal. Moreover, as a client, we do not want to accept the recommendation result based on the airport terminal because it cannot reflect client's interests. Therefore, our model made a wrong decision. Thus, we can dismiss this recommendation by selecting "Give up" as an input. As seen

$$x_i = [x_{i1} \;\; x_{i2} \;\; x_{i3} \;\; x_{i4}] \qquad y_i = [1 \;\; 0 \;\; 0] \qquad \hat{y}_i = [f_c \;\; f_d \;\; f_b]$$

$$g_c = w_{c1}x_{i1} + w_{c2}x_{i2} + w_{c3}x_{i3} + w_{c4}x_{i4} + b_c$$
$$g_d = w_{d1}x_{i1} + w_{d2}x_{i2} + w_{d3}x_{i3} + w_{d4}x_{i4} + b_d$$
$$g_b = w_{b1}x_{i1} + w_{b2}x_{i2} + w_{b3}x_{i3} + w_{b4}x_{i4} + b_b$$

$$f_c = e^{g_c}/(e^{g_c}+e^{g_d} + e^{g_b})$$
$$f_d = e^{g_d}/(e^{g_c}+e^{g_d} + e^{g_b})$$
$$f_b = e^{g_b}/(e^{g_c}+e^{g_d} + e^{g_b})$$

Fig. 7: Training procedure

in figure9, the prior belief is "User1" according to similarity. After the client select "Give up" each time, we can see in the figure that the matching score is pretty low.

To make a better decision, our model uses another neural network, and the result can be seen as figure10. We can see the model generates a reasonable prediction result. According to the neural network performance on the validation set, variable A equals 2. According to the similarity, we divide the decision update process into three parts. Given the lowest to the highest similarity, the model will recommend from 3 to 1 users. We assign w_1 = 0.2 and w_2 = 0.8 in that we want to focus on client's selection. Then, to concentrate on the accuracy of a model, we make $w_3 = 2$. As seen in figure 11, the model's prior belief is user1 in that it has the highest similarity with respect to our client. However, our client can choose whether to accept the recommendation or not. After viewing the images User2 has, he may have an idea that User2 is a better choice. Thus, on each decision making step, he chooses User2 and gives up the last recommendation made by the highest similarity. From figure11, we can see the matching score is changing each time. By the utility function, the model makes a final decision of User2, which means even though User1 seems like a good choice according to similarity, our model can still strike a balance between the client's input and the attributes within the model. These two experiments show that our model not only considers the model attributes, such as similarity matrices and model accuracy but takes the client's interaction into account.

VI. Conclusion

We present a new decision-making model to combine image classification and friend recommendation. Our approach enables an image, uploaded by a user, to be classified by a neural network. Our proposed decision-making model takes users' similarity, model accuracy and client's interaction into consideration and makes a reasonable final decision. I expect that the setup, methods and results in this report will be useful for further studies.

References

[1] Hagan Demuth Beale, Howard B Demuth, and MT Hagan. Neural network design. *Pws, Boston*, 1996.

[2] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *2014 IEEE conference on computer vision and pattern recognition workshops*, pages 512–519. IEEE, 2014.

[3] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[4] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492. IEEE, 2010.

[5] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in neural information processing systems*, pages 8778–8788, 2018.

```
User:
6 airport_terminal
16 airport_terminal
11 airport_terminal
18 airport_terminal
_____
User1:
6 airport_terminal
16 airport_terminal
11 airport_terminal
18 airport_terminal
_____
User2:
11 airport_terminal
18 airport_terminal
13 airport_terminal
1 airport_terminal
_____
User3:
6 airport_terminal
16 airport_terminal
11 airport_terminal
18 airport_terminal
10.0 10.0 10.0
```

Fig. 8: Preference Matrices

```
Decision beliefs: User 1
input the user you want to choose: 1 = user1, 2 = user2, 3 = user 3, 4 = give up>? 4
Matching score: 0.0800
input the user you want to choose: 1 = user1, 2 = user2, 3 =  give up>? 3
Matching score: 0.1600
input the user you want to choose: 1 = user1, 2 = give up>? 2
Matching score: 0.2000
New Decision: User:  1
```

Fig. 9: Decision Making Procedure

```
User:
dentists_office
skyscraper
lighthouse
tree_house
_____
User1:
dentists_office
skyscraper
lighthouse
tree_house
_____
User2:
lighthouse
tree_house
sauna
arch
_____
User3:
arch
sauna
lecture_room
classroom
_____
```

Fig. 10: Preference Matrices of New Model

```
similarity 1:  3.5968757625671515
similarity 2:  3.3909014280999132
similarity 3:  3.097425255226831
Decision beliefs: User 1
input the user you want to choose: 1 = user1, 2 = user2, 3 = user 3, 4 = give up>? 2
Matching Score: 0.1600
input the user you want to choose: 1 = user1, 2 = user2, 3 =  give up>? 2
Matching score: 0.6400
input the user you want to choose: 1 = user1, 2 = give up>? 2
Matching score: 0.4000
New Decision: User:  2
Compared with previous recommendation, Final Decision: User:  2
```

Fig. 11: Fianl Decision