Model User Preferences for Location Based Recommendations

*Bhanu Pratap Jain, Rohit Begani, Ronny George Mathew*

**Abstract**

In this project we will create an artificial agent that can recommend eating choices to a user. Our agent will run through an existing dataset, representing user’s eating behavior (cuisines, category, etc.) and learn the user’s preference model for food choices. Based on the learned user preferences, current location and time, the agent will recommend an eating choice for the user in terms of food type and the optimal place to eat. Additionally, through user’s feedback, the agent will evolve its preference model and recommendation function. We have used KNN algorithm.

**Outline**

1. Problem description: What problem are you solving? Describe the problem from a computational perspective. What are the inputs and outputs (exactly)?
2. Algorithms: What algorithms do you use? Why are these algorithms appropriate? How are these algorithms typically used, and how are you using them?
3. Results: Quantitatively characterize how your algorithm worked. Under what circumstances does the algorithm solve your problem successfully? When does it fail? You should quantify performance somehow. For example, if you are using a search algorithm, you might report the number of nodes expanded for different problem scenarios. If you are using a learning algorithm, you might report accuracy. If you are using reinforcement learning, you might report average utility as a function of trial number.

Objective:

**KNN Algorithm**

We have used KNN. The KNN algorithm or the k-Nearest Neighbors Algorithm is a non parametric lazy learning algorithm. [1] The KNN algorithm shows the k nearest neighbor. We are using this algorithm to match the feature values of the user model with the feature values of the Business Model. Then using KNN the nearest neighbor would be selected and sent as an output.

**Uses of KNN:**

1. Text Mining\*
2. Agriculture\*
3. Finance \*
4. Medicine\*

\*All based on [2]

Why KNN?

**The Data**

We are separating the data into:

* **Training Data:** The training data consists of the user data from a particular location.
* **Test Data:** This data consists of the user data from a new location.

**Training Stage**

In this stage, we will iterate through the data and create the weight vector of a particular user, which will have corresponding weights for all the features in the business.

Initially, all the weights are initialized to 0.

Pick the first business from the data and iterate through each feature updating its weights using the weight update function.

**Testing Stage**

Here, we will use the test data to apply KNN on the test data and predicted user weights as input.

If the rating for the particular prediction is greater than a threshold value, the prediction was good.

**Error Correction**

We have not implemented an error correction mechanism, but future work can be done to integrate NLP into the project, where the user review can be read to find out the particular feature, the user did not like about the predicted business and update the appropriate feature vector accordingly.

**Inputs**

* **User Model:**
  + This model, we are trying to create a preference model of the user for the different features of the businesses.
  + The User Model contains the following features:

|  |  |
| --- | --- |
| User id | Name |
| Music | Attire |
| Ambience | Price range |
| Good for | Parking |
| Category | Miscellaneous |
| Location | Wi-Fi |
| Dietary Restrictions | Stars |
| Alcohol | Current Time |

* **Business Model**
  + The Business Model also contains some features. Almost of these features are same as that of the User Model.
  + The Business Model contains the following features:

|  |  |
| --- | --- |
| Business id | Name |
| Music | Attire |
| Ambience | Price range |
| Good for | Parking |
| Category | Miscellaneous |
| Location | Wi-Fi |
| Dietary Restrictions | Stars |
| Alcohol | Current Time |

* + **The weight update function:**
    - is used in the weight calculation for the user model. We provide the features of the business and the weight is updated as below:

wi 🡨 wi + trainingFactor \* (ru) + (1 - trainingFactor)\* ra

where:

* + - * wi is the weight for the feature in the user model.
      * ru is the user rating for the particular business.
      * ra is the average rating of the business by the all the users.
      * trainingFactor is the factor that computes the amount of training between the user rating and the average rating.
  + It consists of a number of features about the user which will be used to create a preference model for the user.

**Reference:**

1. [*https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/*](https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/)
2. [*http://www.ijera.com/papers/Vol3\_issue5/DI35605610.pdf*](http://www.ijera.com/papers/Vol3_issue5/DI35605610.pdf)