Model User Preferences for Location Based Recommendation

CS5100 – Final Project

Bhanu Pratap Jain

Rohit Begani

Ronny George Mathew

Table of Contents

1. [Abstract 1](#_Toc437859165)
2. [Methodology 1](#_Toc437859166)

# 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.

# Methodology

We have used concepts of supervised learning to come up with a prediction algorithm which is derived from KNN.

The KNN algorithm or the k-Nearest Neighbors Algorithm is a non-parametric lazy learning algorithm. [1]. K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970’s as a non-parametric technique. [2]The model for KNN is the entire training dataset. When a prediction is required for an unseen data instance, the KNN algorithm will search through the training dataset for the k-most similar instances. The prediction attribute of the most similar instances is summarized and returned as the prediction for the unseen instance. The similarity measure is dependent on the type of data. For real-valued data, the Euclidean distance can be used. Other types of data such as categorical or binary data, Hamming distance can be used. [3]

For our custom algorithm we have leveraged from the nearest neighbor measure from the KNN to formulate the prediction score for each sample. Our entire system is based on two models as below:

1. User Model
2. Business Model

## User Model

The user model represents the user interacting with the system. We have used passive learning over an existing data set to train the user model. The user model is has the following attributes:

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

# 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.saedsayad.com/k\_nearest\_neighbors.htm*](http://www.saedsayad.com/k_nearest_neighbors.htm)
3. [*http://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/*](http://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/)
4. [*http://www.ijera.com/papers/Vol3\_issue5/DI35605610.pdf*](http://www.ijera.com/papers/Vol3_issue5/DI35605610.pdf)