Visually Aided Restaurant Selection in Entree

Pankaj Kumar (CS10B042), Rajan Bhargava (CS10B049),
Shivam Srivastava (CS10B051)

CSE department,
Indian Institute of Technology, Madras

Abstract. The aim of the project is to visually help the user in the selection of his/her choice of restaurant, while navigating through the plethora of restaurants in the Entree dataset, each having several attributes. The complexity of the data has been reduced and displayed using MDS. For calculating dissimilarity, Gower's similarity measure has been used. A GUI has been built which helps the user navigate using the MDS graph and find the restaurant of his/her choice. The setup was finally tested on some available session trails in the dataset.

Keywords: Multidimensional Scaling (MDS), Gower's similarity measure, Entree, Graphical Shopping interface, compromise and critiquing in recommenders.

1 Motivation

Entree is a restaurant recommendation system developed for the city of Chicago. It was widely used from 1996 to 1999 and session trails were stored. The Chicago city had 676 restaurants (at the time of creation of the data), each having multiple features describing them. This data is available as well. The prime motivation behind this project was to use the prior information and present it to a user in the current scenario so that he/she can make the best choice and reach the desired restaurant quickly. MDS simplifies the choice by preserving the measure of distance between every two restaurants in a 2-D space (obtained from a higher dimensional space) and because of this a larger number of restaurants can be displayed at any step, reducing the number of steps in reaching the final restaurant. Also, a novel similarity measure has been used which attempts to capture better similarity between any two restaurants.

2 Background

To understand the project, one needs to understand some underlying concepts such as Multidimensional scaling, Gower's similarity measure, Graphical Shopping interface and compromise in recommenders. Some concepts have been explained in detail in the approach section. A brief description is as follows:

2.1 Multidimensional scaling

Takes an input of an NxN distance measure matrix and returns an Nx2 output in terms of coordinates that preserve the distance. The aim is to represent the high dimensional distance data in a 2-D space.

2.2 Gower's similarity measure

A measure which takes as input the real and categorical attributes of two entities and also the weights associated with each attribute, and return a similarity between the two entities.

2.3 Graphical Shopping interface

An interface allowing users to provide an ideal restaurant as input and navigate to their choice.

2.4 Compromise in recommenders

Out of several attributes, a few attributes may not be so important, so not taking them into consideration for fetching restaurants can be a choice. This lack of consideration is the compromise on this attribute.

3 Related work

The core ideas used in the project have been inspired from [1]. This paper builds a graphical shopping interface on gadgets and uses Gower's similarity measure to achieve it. A significant amount of their ideas have been used in our approach.

4 Approach

4.1 Overview

As stated earlier, our system takes three file inputs: restaurant-critiqueFeature mapping, restaurant-properties mapping and properties-description mapping. The restaurant-critiqueFeature mapping data has an (Nx6) data which describes the 6 critiquing properties:cheapness, niceness, creativity, whether it is traditional, liveliness and quietness. This data was provided to us and was generated using the ideas stated in [2]. The restaurant-properties mapping data has a map between each restaurant and its properties such as Mexican, American, Chinese food etc. And finally, the properties description mapping describes each of these properties. The latter have been used in displaying. Once the system starts, it asks the user to enter the attributes of the ideal restaurant that he/she is searching for and also the weights associated with each attribute indicating the importance of each attribute compared to the others. Given the above inputs, the system presents a user interface where the user can navigate to some restaurant. Either the user keeps navigating until a restaurant is found or the user leaves in between.

4.2 Details

Having taken the inputs, first some pre-processing is done. The input restaurant is compared with all the Chicago restaurants and a sorted list is created which stores the Chicago restaurant Ids sorted according to the similarity measure. Then a matrix (NxN) is created which stores the dissimilarity between each pair of restaurants in Chicago. The similarity measure used is Gower's similarity which is explained later.

Having calculated the most similar Chicago restaurants from the sorted list, a list of restaurants to be displayed is created. For the first display, top 'p' entries in the list are selected and displayed after getting their coordinates using MDS. The working of MDS is explained later. Having displayed the restaurants, user's input is waited for. The user can select any of the 'p' restaurants displayed. The distance between the output restaurants is indicated using the numbers marked on the X and Y axes. Now, the user's input is taken and the next set of 'p' nodes to display is calculated. For this calculation, various methods exist as suggested in [1].

We have implemented our own idea which takes top 'k-1' restaurants from the ranked list of restaurants that we first generated for the input restaurant and 'p-k' restaurants from the next most similar restaurants compared to the clicked restaurants. The clicked restaurant is displayed in the next step again, so that if the user wants to end his search having seen the next best recommendations, he can do so. Also, the other already displayed restaurants, are not displayed again. This approach was found to be feasible as one would want to get the most similar restaurants to the currently clicked restaurant and also the closest restaurant to the input parameters. So, our method is a trade-off between the two by choosing some from each of them.

An alternative way of getting the next restaurants is by compromising on one of the attributes and getting the most similar restaurant. This is an alternative we have implemented where each attribute is compromised on once and the most similar restaurant hence found is added to the list of restaurants to be displayed next. This compromise is done by setting the weight of that attribute to zero so that that attribute wont be considered while calculating the similarity. To compromise or not to compromise can be selected at the beginning and then the system can proceed.

4.3 Modified Gower's dissimilarity measure

We have used the Gower's measure with some modification to find the dissimilarity between any two restaurants. The following formula is used in Gower's dissimilarity.

$$\delta_{ij} = \sqrt{\frac{\sum_{k\in C} w_k m_{ik} m_{jk} \delta^C_{ijk} + \sum_{k\in N} w_k m_{ik} m_{jk} \delta^N_{ijk}}{\sum_{k=1}^{K} w_k m_{ik} m_{jk}}}.$$

Gower's measure considers both categorical and numerical attributes together to find the dissimilarity. The summation on C indicates the dissimilarity on the k^{th} categorical attribute between entities i and j. The same holds for the numerical (with superscript N) attribute. Ignore the 'm' attributes as they only stand for missing values which our dataset does not have. So 'm' takes 1 everywhere and 'w' are the weights of associated with each attribute. The numerator is normalized using the weights.

The following formula is for the numerical attribute. It simply calculates the distance between the attribute values normalized over all attributes.

$$\delta^N_{ijk} = \frac{|x_{ik} - x_{jk}|}{\left(\sum_{i<j} m_{ik} m_{jk}\right)^{-1} \sum_{i<j} m_{ik} m_{jk} |x_{ik} - x_{jk}|}.$$

The following is the formula for calculating dissimilarity on the categorical attributes:

$$\delta^C_{ijk} = \frac{1(x_{ik} \neq x_{jk})}{\left(\sum_{i<j} m_{ik} m_{jk}\right)^{-1} \sum_{i<j} m_{ik} m_{jk} 1(x_{ik} \neq x_{jk})}.$$

It is the normalized value if the values match, else it is zero. We have modified this formula to suit our needs by using the Jaccard's distance coefficient:

$$d_J(A,B) = 1 - J(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$$

Here, the A and B correspond to the feature description of each restaurant which takes values such as Asian, American, Chinese, Continental etc. The more these features match the lesser the distance between the two. Also, we have not associated any weight with the categorical dissimilarity.

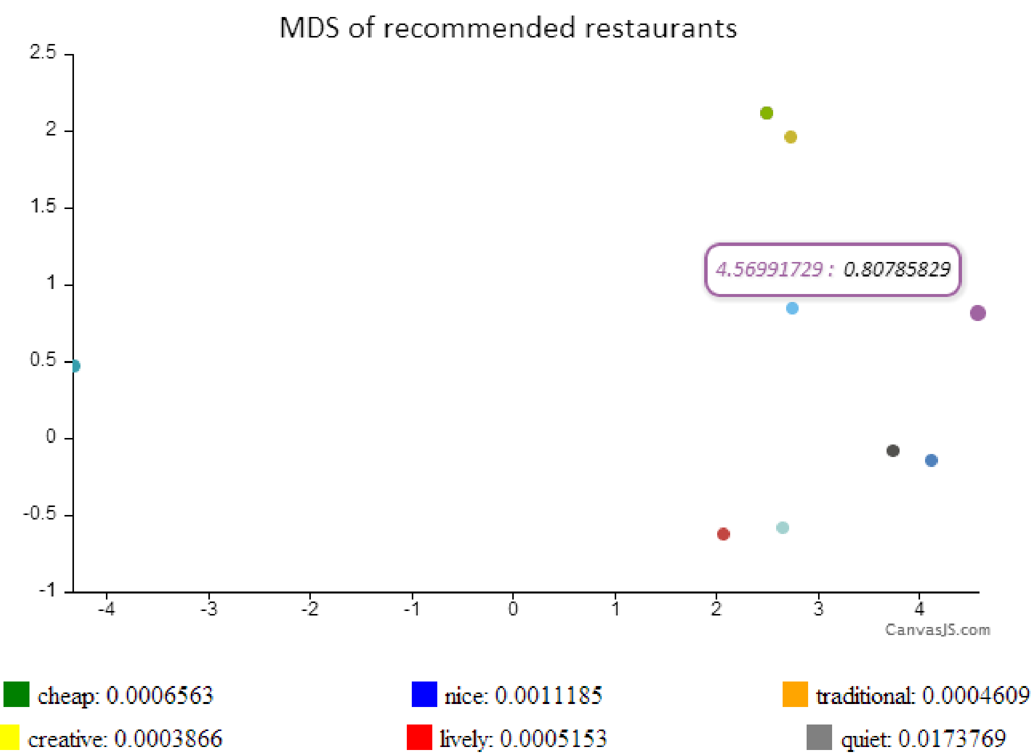
So, the above values are plugged into the Gower's formula and a dissimilarity matrix is obtained between each pair of restaurants. This matrix is given as an input to MDS for display. MDS minimizes the following stress function to arrive at coordinates for each restaurant in the 2-D space:

$$\sigma_r(\mathbf{Z}) = \sum_{i<j}(\delta_{ij} - d_{ij}(\mathbf{Z}))^2$$

Here d is the Euclidean distance between i and j restaurants in the 2-D space and delta is the dissimilarity found in the previous step between restaurants i and j. The above is solved as an optimization problem and hence yields different values depending on convergence.
In our system, at each step 'p' restaurants are calculated (p as 10 was used in experiments) and MDS is used to calculate the coordinates of these 10 restaurants in the 2-D space by giving the dissimilarity values between these 10 restaurants.

5 Navigation system

The following shows what the user will see once he/she enters the system.



ID: 302 Characteristics: Cab, Parking/Valet, Weekend Brunch, Romantic, Quiet for Conversation, Private Parties, Private Rooms Available, Dining Outdoors, Italian, Excellent Food, Excellent Service, Excellent Decor, $15-$30,

The set of points are the restaurants displayed in the MDS graph. On hovering over a point, the user gets all the details about that restaurant. The details include the attribute values: cheapness, niceness, creativity, traditional, lively and quiet.
It also shows the properties of the restaurant such as Parking, weekend brunch, romantic/quiet etc. The points are click sensitive, so on clicking, the user will be taken to a set of new points which are chosen from the top 'p' restaurants as described earlier.

6 Empirical Evaluation

For testing, the session trails were used which had a Chicago restaurant as their starting point and had a valid end restaurant. To test how well our system performs, we used a measure of how quickly a user can reach the end restaurant in the best case given our system. That is, if a user starts with a restaurant 'r' in our system, where 'r' matches with the starting point in the session trail then how quickly (in the best scenario) can the user reach the end restaurant 'er', say. We calculate the distance between the starting restaurant and ending restaurant by finding the shortest set of selections that can help us reach there. We compare our results on 10 sample session trails and the results are as follows:

Table 1. Results from the System

Trail-From-To	Distance-From-Session	Shortest-Distance-from-Model
123 to 501	5	3
464 to 581	7	3
573 to 541	13	3
197 to 60	7	3
38 to 216	14	2
178 to 260	6	3
640 to 148	4	1
162 to 603	3	3
44 to 192	3	3
294 to 259	11	3

The above results show that in our system, the user can reach the choice of his/her restaurant in 3 or less steps. But this should be taken with a pinch of salt because at each step we generate 10 restaurants and display it in the user interface. So within 3 steps all restaurants are covered as the tree grows exponentially. Within 1000 restaurants all the Chicago restaurants (676) are covered. But the results aren't disappointing as we compensate for the possible lack of a good similarity measure by using a good interface which allows user to reduce his/her search length. Had there been more data more thorough tests could have been conducted. Also, as what selections a user makes between the starting and the ending restaurant is completely user dependent and cannot be simulated, nothing more can be analyzed from these results. But still, on an average it is expected that the user will reach the end result quicker in this interface compared to the prior-art.

7 Discussion

More approaches were considered which involved using experience based critiquing, critiques based on graphs and approaches to improve the user interface, but these were rejected because of lack of sufficient data in the Entree dataset. The system built in 1996 did not incorporate displaying the suggested restaurants such that a notion of distance between them exists, to the best of our knowledge. By using MDS and displaying it in a 2-D interface, the notion of distance has been covered. Also, given what we know, the recommendations used in the older system did not incorporate similarity between two restaurants using their categorical attributes as well. Our system captures this by finding the intersection between the properties (categorical variable) of two restaurants. Also, because we can display a large number of restaurants at a time in a 2-D space, the user is expected to reach the end restaurant faster.

There can be improvements to the above system. The user can be given the handle to modify weights at each step, which can be a way of critiquing the attributes. Another modification can be by experimenting with the dissimilarity measure and deciding whether weights should be used for the categorical attributes or not. Lastly, the user interface can always be improved in a countless ways.

8 Conclusion

The system provides a user interface to the user to carry out a search for the restaurant by giving more flexibility and better visual capability compared to the prior art to reach the end restaurant in the Entree dataset. The results also show considerable improvements, though the results cannot be taken as proof enough to show that the system outperforms the prior-art due to lack of data. But empirically, it has been shown that the distance between the starting and the end restaurant has been reduced significantly. These ideas can not only be used for Entree, but can also be used for any kind of dataset which involve critiquing, hence the approach has a wide scope for implementation and improvement.

References

1. Martijn Kagie, Michiel van Wezel, Patrick J.F. Groenen - A graphical shopping interface based on product attributes
2. Mining User Trails in Critiquing Based Recommenders (yet to be published)

3. McCarthy, Kevin; Salem, Yasser; Smyth, Barry - Experience-based critiquing : reusing critiquing experiences to improve conversational recommendation
4. Robin Burke, Kristian Hammond, Benjamin Young - The find me approach to assisted browsing
5. Dietmar Jannach - An introduction to recommender systems
6. Quang Nhat Nguyen, Francesco Ricci, Dario Cavada - Critique based Recommendations for Mobile Users: GUI Design and Evaluation
7. Tarik Hadzic, Barry OSullivan - Critique Graphs for Catalogue Navigation
8. Jiyong Zhang, Nicolas Jones, Pearl Pu - A Visual Interface for Critiquing-based Recommender Systems