CSE515- MULTIMEDIA AND WEB DATABASES PROJECT PHASE I

Group members:
- Shubham Vipul Majmudar
- Chirag Bhansali
- Aravamuthan Lakshminarayan
- Yash Pande
- Manthan Bharat Bhatt

Submitted by:
Shubham Vipul Majmudar
1215200298
Prof Selcuk Candan
CSE515
September 16, 2018

ABSTRACT

The purpose of the project is to use vectors corresponding to various entities like user, image, or location and find a way to condense all this information in relative terms so as to find similar entities. The dataset given contains data pivotal to this task- textual (such as TF, DF, TF-IDF of the terms) as well as visual (CM, CN, HOG, etc). The textual information gives the context of the image and the visual aspect gives the contents of the image. In the initial 3 tasks, we have used the textual data to infer similar users/images/locations by numerically comparing them by calculating their relative distance from each other. This distances that are used, as will be discussed later, quantify the similarity of the entities which can now give us the scope of ranking them based on this measure. The terms associated to a particular entity comprise of a dimension and this leads to the vectors being very sparse in nature. In the final two tasks, visual descriptors have been used to identify similar images and locations. The image models provided measure a variety of features (visual) of an image- the color intensities, variations and even texture; each model serves a unique measure. When it comes to comparing locations, dimensions need to be reduced/summarized at certain points for computational efficiency; without significantly incurring loss of data. In the end, a combination of all the models have been used to find the most similar locations.

INTRODUCTION

Terminologies

- Cosine distance:
  It refers to the value obtained on dividing the summation of all the values at a particular dimension corresponding to the two entities by the modulus of the entities. This value gives the angular distance between two entities while normalizing their modulus value.

- Manhattan distance:
  This distance refers to the summation of the absolute positive values of the difference between the values belonging to the same dimension, between two entities. This measure treats positive and negative deviations equally.

- K-means clustering:
  K-means is a form of clustering in vector spaces, such that given the ‘k’ (which refers to the number of clusters that have been pre-assigned) it creates ‘k’ number of clusters. It arbitrarily initializes the cluster centers, and builds from there subsuming the nearest points at every iteration while also updating the cluster centers.

- Visual descriptors:
  Visual descriptors assign numerical values to the visual information stored as an image. Each descriptor conveys a unique characteristic of the image, and hence the usage of each of these is highly subjective on what needs to be found out. The characteristics can include edges, textures, variations, and intensities; although they may even be sometimes abstract.

Goal description

The first task requires us to find the top ‘k’ similar users based on their textual descriptors, and list the top 3 contributing terms. Each user contains a set of terms that have an associated TF, DF, and TF-IDF value; which are in turn later used to compare the users.

The second task requires us to find the top ‘k’ similar images based on their textual descriptors while also finding the top 3 contributing terms. The goal is akin to the previous one except that the superclass is now of an image instead. It can be noted that the dataset used is scraped from a single source but only rearranged in terms of images, and hence the terms in each of the entities involved in the initial 3 tasks remain the same.

The goal of the third task is to find the top ‘k’ similar locations based on their textual descriptors and further find the top 3 contributing terms. The user input however has to be pre-processed and the corresponding location name/id has to be located from an xml file.

The goal of the fourth task is to find the top ‘k’ locations based on a model provided by the user but this time using its visual descriptors. Also, top 3 images for each similar location have to be shown.

The fifth task requires us to take into consideration all the descriptor models of the images of a given location and find out the top ‘k’ similar locations to the one provided by the user. Each descriptors contribution is also to be shown.

Assumptions

It has been assumed that k-means clustered images, where ‘k’ is taken as 50, in a location are an accurate description of the same. This approximation has been done in order to make computation efficient and time-saving.

Further, for the fifth task, it has been assumed that each descriptor model has an equal say in the overall similarity of the locations.

DESCRIPTION OF THE PROPOSED SOLUTION

Owing to the equivalence of the initial three tasks, user/image/location will be referred to as ‘entity’ hereon.

Each entity consists of terms that each form a dimension. Hence the vector matrix for each entity is very sparse, and hence images oriented towards similar directions will correspond to more keywords in common. This can be mathematically quantified in the form of cosine distance, which measures the angle between two vectors. Once this has been done, the similarity score of each entity is sorted and a rank-list is prepared. From here, the top ‘k’ entities can be obtained. Now to get the top 3 terms, the contributing factor of each term is the value obtained on multiplying the terms (dot product of the particular dimension) and these are in turn ranked and displayed for each similar image.

For the fourth task, each location has unto 300 images, and each image in turn has several visual models of up to 81 dimensions. To compute for each image required significant computational power and time, and hence a trade-off had to be struck in order to simplify computations while also not losing the essence of the information. Clustering images was found to be useful here, which meant that a set of images belonging to a cluster is now represented by only the center of the cluster. Each location is aggregated to 50 image clusters instead of every image, and this also helped bring uniformity to each location as the dimension is now fixed and not dependent on the number of images within. Now, to calculate the distance between the locations (one fixed by the user, and the other iterated from the locations-list), an image cluster from the given location was taken and compared to every image cluster present in the juxtaposed location. The comparison/similarity metric used for comparing the images is Manhattan distance. Now for every image on the fixed location side, only the minimum Manhattan distance is stored for it, ie we are only storing the distance associated to nearest cluster pair of the image. This is done for all the image clusters on the fixed location side, which will gives us a 1x50 matrix. This list is appended to a super-list (no. of locations x 50) which stores the set of all such lists once all locations are iterated over. One point to be noted is that this min-pair distance is not symmetric as we are only looking at the minimum cluster on the

other side wrt to the one on the fixed location side. Hence, it is not a symmetric relation as the clusters on the other side might have different min-pairs. Once the super-list is ready, it is condensed to a 1d list by adding up the min-pair values stored at each image on the fixed location side, forming a (no of locations x 1) matrix. The above similarity measure can be interpreted as- we are taking summation of the distance of the best cluster pairs for each image.

Now that we have the numerical relation of the provided location with the other locations, we can prepare a rank-list and extract the top ‘k’ locations. The ‘k’ as well as the model is provided by the user. The min-pair distance was also found to be helpful in maintaining a consistency, wherein a location is always most similar to itself- which was not always the case while using aggregation with other measures like cosine and euclidian distance.

In case of the fifth task, the implementation to obtain a rank-list is as above, but now we have 10 individual rank-lists for the 10 models. To find the top ‘k’ locations out of these, the rank associated to each location was used as a weight and the overall rank-list was prepared on the basis of the cumulative weight of each location over the 10 rank-lists.

INTERFACE SPECIFICATIONS

The user has to run the python file associated to the task (eg task1.py for task 1). After that the entity id, textual descriptor model and the number of similar entities need to be entered separated by spaces. For the fourth task, input requires the location number, visual model and the top ‘k’ similar images separated by a space. The fifth task requires the same except the visual model.
The output will display the top ‘k’ similar entities with their scores and also the top 3 most contributing terms.

SYSTEM REQUIREMENTS

A basic Windows/Unix OS, with the system having at least
1.2 GHz processor;
8 GB RAM;
10 GM disk space

in order for it to run within 5 minutes of time

Installation requirements include Python 2.7 with the libraries numpy, BeautifulSoup, and scikit-learn