COMP90051 Assignment1 Report

M&Y: Qingyang Hong(629379), Anchalee Laiprasert(617544)

September 4, 2014

1 Overview

The project focuses on predicting test users' geographical locations with access to posts by their friends, part of test users' posts information and relationship graph. Since we would face several key challenges to get locations of test users, in this report, we will get insightful observations from data, select relevant features and apply various learning models to make the most reasonable prediction.

2 Implementation and Analysis

According to the properties of the task, the intuitive attempt is to find test user's friends and try to predict based on friends' information. As friendship represents close relationship, test user and his/her friends should have more nature in common. And one of the most relevant nature is the location, which we can directly retrieve from training data set.

2.1 Data Sampling

Before selecting relevant features and applying machine learning models, the underlying information in the data sets should be discovered. By counting the friends of each test user and concluding the number of users having $'$N$'$ direct friends, we can have a table as follows:

friendsNumber(N)	1	2	3	4	5	6	7	8	9	>10
CntUsrFrdN	357	143	106	67	51	29	28	25	18	176

Table 1: Summation of usrs who has $'$N$'$ direct friends

Such conclusion gives an intuitive look into the first layer of user relationship in the social graph, which indicates that users who have limited number of friends account for large percentage of the test users(e.g. (N1+N2+N3)/NN = 606/1000). Further thought into this discovery implies that prediction based on limited friends information might not be precise.

Subsequently, by studying the training data set, some exceptional data records are found to have both its latitude=0, longitude=0. Such training user records should be seen as "invisible" users, for hiding their locations. When we predict test users' locations, the location information of these users aren't involved.

Further traversing through the training data set shows that some other users have few posts, which might indicate that they aren't reliable in our first point of view. The assumption could lead to lower weight of such users in location prediction. While later experiments show that this assumption doesn't seem to be correct. For example, ignoring those records with only 1 post during sampling data doesn't help to improve preciseness in prediction.

2.2 Features Selection

Features selection is the most important step in getting a good prediction. At the most general level, we consider the friendship graph, graph.txt, to infer relationship between users. By combining data records from training data set, posts-train.txt, and the social graph, features which are suitable for the project context are produced:

- Friendship: In our intuitive thinking, the friendship is the most influential feature. Friends' information should be the direct resources for further feature expanding and it is also confirmed by the theory that geographically nearby users are more probable to create friendship relation. Based on the analysis in the data sampling, if a test user has a limited number of friends, then the accuracy of predicted location is relatively low. Regarding the result, the 2nd-tier friends (friends of friends) are involved into our experiments. The rationality of such process is based on indirect relationship between 2nd-tier friends and test users.
- Location: Latitude and longitude of friends are the main direct features in the prediction since they can be seen as in relevance to the test users' location. By doing a simple experiment, calculating average location of each of test user's friends and feed those locations to related test users, we get 38.88 in RMSE score, which we later found is a relatively precise prediction.
- Region: After locating all training data points in a world map plot, most data points are located in US, Europe, Asia(especially Japan).
- Most Frequent Hours: With the perception that the friend who has an active hour close to the test user, they might live close to each other. Actually, numbers of users on hours from 0 to 23 in Hour1, Hour2, Hour3 are mostly evenly distributed from 12 to 23 and very few users post between 00 and 12. It still seems to be reasonable to use this feature to classify users from different time zones based on exactly modern people's social habit that they tend to post in the afternoon and night rather in the early morning. While it's a very general regular parttern, which means that it doesn't fit in small amount of people. e.g. try to infer a test user's time zone based on 10 friends' hour1 information. In a word, this feature could deviate the prediction on test users with limited friend and leads to bad prediction overall.
- Distance: Not all friends are geographically close to a test user. Some of the friends could be in a quite far location, who would actually deviate the prediction and are seen as outliers in the data. So distance feature is chosen to represent the distance from one friend data point to the average centre of all friends' locations. Outliers shoud be eliminated before learning.

2.3 Models Analysis

2.3.1 Friend Similarity

When taking the friends into account, our method counts number of common friends between a test user and each of his/her friend into account and set a weight based on number of common friends(e.g. a friend has 5 common friends with larger weight in prediction than that has 3). The setting of weight is a main issue in this approach, which means that a too large weight on certain friends might lead to biased prediction. Our method can be described as follows: if test user T has a friend N, T has T_s friends and N has N_s friends, they have some friends in common, as SUM, the weight for N should be 1+SUM/$(T_s+N_s)^2$. By having this setting, the weight is based on the proportion of common friends in both direct and indirect friends sets of test user, which would not lead to much bias.

2.3.2 Learning Models

Several learning models have been applied to try to do the predictions:

Model	SVR	k-NN	K-Means	DecisionTree
RMSE	62.80	27.13	27.46	65.96

Table 2: Performance of Models

Rationality based on advantages/disadvantages:

- With the distance of friends, we adopted the clustering with K-Means algorithm. In this scenario, it aims to find the all elements within a group which are more similar among others. Dataset of latitude and longitude of friends as input; together with a parameter K that specify the number of cluster. The cluster that has the greatest number of members is chosen to compute the locations of test users assuming that a smaller cluster is outliers set. The clustering methodology gives clear division between useful friends points and outliers. Disadvantage of this approach is the distance between clusters was not scaled well and the number of cluster could not assigned properly.
- Similarly, k-NN, which focuses on making use of nearer friends as inputs of predictions. Objective of this method is also trying to minimize influence of outliers by ignoring those friends points that are too far and performance of this method is similar to k-Means, for the property of using nearer friends fits the relationship feature. In terms of disadvantage, it's hard to decide how many neighbours should be involved, and with improper k neighbours, the model might give bias in prediction.
- SVR gives relatively a bad performance. According to the region feature, most users are located in three regions. On the contrast, SVR fits more in a scenario whose outputs are in two categories, which doesn't quite fit our scenario.
- Useful features are limited to feed a decision tree model. After training the model, predictions of both latitude, longitude have discrete values(e.g. prediction of latitude's value could be in the set (A,B,C)). Limited features only help to produce a tree with limited branches, which would not be a good decision tree.

2.3.3 Overfitting

Basic steps to do a machine learning project are: Prepare Data− >Choose an Algorithm− >Fit a Model− >Choose a Validation Method− >Examine Fit and Update Until Satisfied− >Use Fitted Model for Predictions.

In our experiments, we've ignored the validation and examination step, which later gives an important lesson. In real world machine learning problem, we would hardly know how ”fitting” our current model is by only examining it based on limited test data. In real world, data is growing in training data set and test data could also probably change. The insight is that a model fits well in some data might have overfitted and wouldn't describe the real world. Examining resubstitution error, cross-validation error or other errors with current precise data(training data) is necessary and a must-do, in order to predict well in the unknown(test data).