Consumer Targeting Based on Historical path and Network Structure on Real Time

Dongguang(Nick) You
Computer Science Department

Cenying (Tracy) Yang
IROM Department

1 Motivation

Smart phone usage is expected to exceed 6.1 billion users worldwide by 2020. The proliferation of mobile and sensor technologies has contributed to the rise of mobile location-based advertising. such advertising can enable business to deliver information to mobile users in real time about offers in geographical proximity to them. Recent studies using randomized field experiments have causally shown that mobile advertisements based on static location and time information can significantly increase consumers' likelihood of redeeming a geo-targeted mobile coupon.

Beyond the real-time snapshot about the static geographical location , the mobile trajectory of each individual consumer and behavior of geographically near consumers can provide richer information about consumer preferences. In particular, "trajectory" refers to the offline physical movement of each consumer. Combining the movements of all consumers, we get a dynamic network in which consumers who are geographically close might share common preferences. The ***goal*** of this project is to come up with an advertising strategy using the information of historical trajectory of each consumer and the information of geographic relationship among consumers.

Consider in a big shopping mall, say Domain, consumers are walking around to search for products they like. Consumer A is in the Sports section and wants to buy a pair of shoes. He/She has visited Nike, Adidas, Air Jordan and is now in New Balance. Consumer B is also in the sports section, and has bought a pair of shoes in Puma. In that case, for consumer A at time t, a trajectory would be Nike to Adidas, to Air Jordan, to New Balance. His/Her geographically close "neighbor", consumer B, is likely to embrace similar preference for shoes as consumer A. The question of interest is how shall we make recommendations to consumer A based on his/her own trajectory and the behavior of other consumers near by.
In the end, we would like to come up with a dynamic recommendation system, i.e., sending mobile-based coupons to consumers on real time. We decompose the problem into two tasks.

- Based on a snapshot about the static geographic location,how shall we make recommendations and design advertising strategy?
- When taking dynamics into account, how shall we make recommendations and design advertising strategy given consumers' historical trajectory and purchasing behavior? The biggest challenge is the computational burden when there is a large number of consumers. Based on historical information, how to update the previously computed recommendations to the current recommendations?

In this project we aim to solve the first task, and leave the second task as future work. We apply label propagation and spatial smoothing to conquer the question.

2 Approach

Roughly speaking, we need to first initialize whether each consumer has bought a product at the current time. We do the initialization using label propagation and Bernoulli random draw. Then we use spatial smoothing to determine to whom shall we recommend the product. The key input, D, is computed based on distances among data points. Finally, we compare our proposed recommendation with another Bernoulli random draw to assess our approach. We now detail our approach as follows.

2.1 Step 1: derive utility score

We initialize whether each consumer has bought a product at the current time based on his/her utility. We derive such utility score using label propagation. We first randomly select N data points as our seeds. Among those N data points, we randomly initialize 20% of them as bought the product with value 1, and the rest as not bought yet with value 0. Then we use label propagation to propagate those N labeled values to other unlabeled data points, i.e., other data points that are not selected. The resultant values fall in the range of $(0, 1)$. We followed "Learning from Labeled and Unlabeled Data with Label Propagation"(Zhu 2002) 's approach for label propagation.

2.2 Step 2: initialization

Given the utility score, we add a random error term to it, and then initialize whether each consumer has bought a product at the current time using Bernoulli random draw. We call the resultant binary vector Y. Now, all the data have been initialized.

2.3 Step 3: spatial smoothing

Each consumer is associated with a binary indicator in Y that indicates whether they have bought this product or not. Our targets are those who haven't bought this product but have a high utility score. We perform graph fused lasso smoothing and identify those who have a zero indicator but high smoothed indicator as our targets. Specifically, we first transform the location distance into similarity through a exponential transformation $similarity = \exp \div -distancealpha$. Then we construct the D matrix. In particular, for each non-zero similarity s between points a and b where a ¡ b, we add a row to D that contains s at a and $-s$ at b. Next we find the vector X that minimizes $(X - Y)^2 + |DX|$. For the data points with initial binary value 0, we use their values in X to decide whether we should make recommendation to them at the current time based on smoothed utility. Spatial smoothing is appropriate here because we assume that consumers we are close to each in a shopping mall share common tastes for the product.

2.4 Step 4: assessment

For the data points with initial value 0, we redo Bernoulli random draws as our true realization of purchasing behavior. From spatial smoothing, we get smoothed utility, and then propose recommendation based on a specific threshold. Finally, we compare the true value with our proposed recommendation.

We use a simple K-nearest-neighbor method as our benchmark, and assess the effectiveness in terms of receiver operating characteristic or **ROC curve**. ROC curve is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting the true positive rate against the false positive rate at various threshold settings. The bigger the area under ROC curve is, the more effective the algorithm is in terms of classification.

3 Implementation

The data is collected on UT campus in terms of latitude and longitude. Since many data points have very similar values, we first group data based on latitude and longitude to make distance between two data points more meaningful, which is the key in our recommendation. We round the latitudes and and longitudes to 4 digits, and end up with 27,750 unique data points. We will use this data set as our input geographic location.

3.1 Step 1: derive utility score

We choose a number of $N = 500$ as our seeds, which is approximately 20% of the total data points. Then, among those 500 data points, 100 of them have value 1, and 400 of them have value 0. We implemented Label Propagation using Scikit's LabelPropagation method with 'rbf' kernel and gamma = 10000.

3.2 Step 2: initialization

We add an error with standard normal distribution of zero mean and 0.05 standard deviation to the utility score derived in the first step, and set the total utility value at 0 if it becomes negative and 1 if it becomes larger than 1. The Bernoulli random draw is implemented using the in-built function in R.

3.3 Step 3: spatial smoothing

The biggest challenge arises when we try to build the D matrix. As mentioned before, the distance is transformed into similarity through an inverse exponential transformation with parameter δ. Since we have 27,750 data points, the total number of pairwise similarities is round 385 million. To reduce the storage memory, we only store similarity with value larger than 0.01. That reduces the number of pairs to roughly 4 million. We implemented this part in Rcpp for its efficiency. After obtaining the D matrix, we run the lasso fused graph smoothing using the ADMM algorithm in exercise 8 (Tansey et al. 2014).

3.4 Benchmark: K-nearest-neighbor

We benchmark spatial smooth method with a simple K-nearest neighbor method for generating recommendations. We used the knn method in package "class" of R.

4 Results

We assess the algorithms in terms of AUC (area under ROC curve). If we randomly make recommendation, the AUC should be 0.5. In the benchmark model, i.e., k-nearest neighbor model, the highest AUC is achieved at $k = 300$ with a value of 0.5552. Put it another way, when we use 300 nearest neighbors to make recommendation, the accuracy is slightly better than random recommendation.

There are two key parameters in our approach: (1) λ in running ADMM, and (2) δ that transforms distance to 0 when calculating D matrix. We tried different combinations of these two parameters with λ ranging from 1 to 300, and δ ranging from 0.0001 to 0.0012. The highest AUC is obtained at $\lambda = 30$ and $\delta = 0.0005$ with a value of 0.56772.

Overall, the results are not satisfactory. Both methods barely outperforms random recommendation, with spatial smoothing obtaining a slightly higher AUC. One possible reason is that the geographic locations are to0 close to each other so that the similarity is not that meaningful. In reality, we could use more rich data to represent locations to calculate distance similarity. For instance, in a shopping mall, we could use the store location rather than geographic coordinates. It could also be that our proposed approach does not fit our research goal here.

Reference

1. Zhu, Xiaojin, and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University, 2002.

2. Tansey, Wesley, et al. "False discovery rate smoothing." arXiv preprint arXiv:1411.6144 (2014).