Geo-Spotting: Mining Online Location-based Services for Optimal Retail Store Placement

Atul Joshi (2018MSBDA005)
Divya Kumari(2018MSBDA020)
Vivek Shukla(2018MSBDA023)

CONTENTS:

PROBLEM STATEMENT

Identifying the optimal location for a new retail store among a given set of candidate locations .

Optimal Location:

- Becomes popular <=> Attracts customers <=> attracts large no. of checkins

Previous Approaches

Issue of practical application of the store location models has become extremely popular with the development of retail giants, such as Tesco, WalMart and others.

History of Optimal Location Store Placement Problem:

- “Stability in Competition” by Hotelling (1929) [2]
  - location in the competitive environment
- “Changing methods of location planning for retail companies” Clarke , 1998 [3]
  - measure the potential of existing centres and analysis of catchments area in terms of population structure

- “The Effect of Retail Store Environment on Retailer Performance, Kumar and Karande”, (2000) [4]
    - determining the appropriate retail environment
    - Geographical features
- Campo and Gijsbrechts (2004) [5]
    - Importance of Retail Format:
        - Independent, Chain owned, or Franchise owned
- Jensen (2006) [6]
    - approach uses a spatial network based formulation of the problem, using intertype attraction coefficient

Approach

Formulate the problem as an ML task, where, by extracting a set of features, we seek to exploit them to assess the retail quality of a geographic area .

- Features:
  - Geographic:
    - Based on types and density of nearby places
  - User Mobility:
    - transitions between venues
    - incoming flow of mobile users from distant areas.
- Target Variable:
  - No. of check-ins

Data

To get data describing user mobility and popularity of places, we used Foursquare.

- Foursquare
    - Local search-and-discovery app
    - Provides personalized recommendations of places to go near a user's current location based on users' previous browsing history and check-in history.
    - Provides API to get user check in data

Collecting Data from Foursquare API

- The Places API offers real-time access to Foursquare’s global database of rich venue data and user content to power your location-based experiences in your app or website.
- Steps:
  - Create Developer account
  - Create App : Get Client ID and Client Secret token
  - Choose Plan:
    - Personal
    - Premium
    - Enterprise

Foursquare in Python: A walkthrough

- Foursquare library
- In addition to library, we can access the data by using GET requests, with url (of foursquare resource), latitude, longitude and query as parameters
- Foursquare Access Restrictions:
    - Account-wise : 950 Regular API Calls per day
    - Authentication:
        - Userless Authentication:
            - used for server-side applications and others that don’t intend to require a Foursquare user’s permissions
        - User Authentication:
            - used when you require a Foursquare or Swarm user e.g. to get all the checkins for a given user.

Request using Userless Auth : Example

```
import json, requests
url = 'https://api.foursquare.com/v2/venues/explore'

params = dict(
  client_id = my_client_id,
  client_secret= my_client_secret,
  v='20191005',
  ll='40.7243,-74.0018',
  query='coffee',
  limit=1
)
resp = requests.get(url=url, params=params)
data = json.loads(resp.text)
```

- Returned Data object is a json format object consisting requested data, and metadata.

Request using User Auth

We need: Time-series of checkins by users for various venues, for different categories.

To get such data, we need to use the foursquare ‘time series’ Endpoint .

But, it requires User Auth with User as the venue’s manager.

```
url1 = 'https://api.foursquare.com/v2/venues/timeseries'
```

```
resp = requests.get(url=url1, params=params)
data = json.loads(resp.text)
```

```
data
```


```
{'meta': {'code': 403,
  'errorDetail': 'A user is required to call this endpoint.',
  'errorType': 'not_authorized',
  'requestId': '5dc18ba302a172002837afe6'},
 'response': {}}
```

Dataset used:

NYC Check-in Dataset :

- 227,428 check-ins in NYC collected for about 10 month
- collected from Foursquare
- from 12 April 2012 to 16 February 2013
- 227428 rows × 8 columns
- Each check-in is associated with its time stamp, its GPS coordinates, user ID, and venue category

We need to use this check-in dataset, to extract geographical and mobility features for each location to predict no. of check-ins.

Prediction Features

Geographic Features:

1. Density
2. Neighbors Entropy
3. Competitiveness
4. Quality by Jensen

Each feature metric returns a numeric score, denoted by $\hat{\chi}_l(r)$

Density

N(l,r): numbers of neighbours of a place within a radius r for candidate location l.

$$\hat{\chi}_l(r) = |\{p \in P : dist(p,l) < r\}|$$

Intuitively ,a denser area could imply higher likelihood for an opportunistic visit to a retail facility.

P = set of venues

Distance Metric Used: Haversine Distance

The central angle Θ between any two points on a sphere :

$$\Theta = \frac{d}{r}$$

d:distance between two points of the sphere

r:radius of sphere

Haversine Formula: $\mathrm{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2)\,\mathrm{hav}(\lambda_2 - \lambda_1)$

$$\mathrm{hav}(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1 - \cos(\theta)}{2}$$

- φ_1, φ_2: latitude of point 1 and latitude of point 2 (in radians),
- λ_1, λ_2: longitude of point 1 and longitude of point 2 (in radians).

Neighbors Entropy

Specifies the influence of the spatial heterogeneity of the area on the popularity of a place.

$$\hat{\chi}_l(r) = -\sum_{\gamma \in \Gamma} \frac{N_\gamma(l,r)}{N(l,r)} \times \log \frac{N_\gamma(l,r)}{N(l,r)}$$

$N\gamma(l,r)$ = No. of neighbors of category γ .

Γ= set of all venue categories

high entropy values => diverse in terms of types of places

least entropic areas => less diverse => number of check-ins is biased towards a specific category

Competitiveness

Measures the proportion of neighboring places of the same type with respect to the total number of nearby places.

$$\hat{\chi}_l(r) = -\frac{N_{\gamma_l}(l, r)}{N(l, r)}$$

More Competitiveness ⇔Customers must be shared

Quality by Jensen

Provides spatial interactions between different place categories.

Used to access retail quality of an area.

$$\hat{\chi}_l(r) = \sum_{\gamma_p \in \Gamma} \log(\kappa_{\gamma_p \to \gamma_l}) \times (N_{\gamma_p}(l,r) - \overline{N_{\gamma_p}(l,r)})$$

$$\kappa_{\gamma_p \to \gamma_l} = \frac{N - N_{\gamma_p}}{N_{\gamma_p} \times N_{\gamma_l}} \sum_p \frac{N_{\gamma_l}(p,r)}{N(p,r) - N_{\gamma_p}(p,r)}$$

$\overline{N_{\gamma_p}(l,r)}$	Venues of type γ_p are observed on average around the places of type γ_l
$\kappa_{\gamma_p \to \gamma_l}$	inter-type attractiveness coefficients

Mobility Features

It depicts knowledge about user movements and transitions between places.

1. Area Popularity
2. Transition Density
3. Incoming Flow

Area Popularity

$$\hat{\chi}_l(r) = |\{(m,t) \in C : dist(m,l) < r\}|$$

Where (m,t) =check-in recorded in place m ∈ P within time t

C =set of all check-ins in the dataset.

P=set of venues

Transition Density

Measures the density of transitions between the venues inside the area.

$$\hat{\chi}_l(r) = |\{(m,n) \in T : dist(m,l) < r \wedge dist(n,l) < r\}|$$

where T : the total set of transitions

Incoming Flow

- accounts for external user traffic towards the area of the place

Considers transitions between places denoted by a tuple, (m, n) ∈ T , such that first place m is located outside and second place n inside the area under prediction.

$$\hat{\chi}_l(r) = |\{(m,n) \in T : dist(m,l) > r \wedge dist(n,l) < r\}|$$

Supervised Learning Approach

How feature vectors x can be associated with the check in scores y of the areas under prediction?

Two different ranking methodologies used are-

- Supervised Regression for Ranking
- Supervised Learning to Rank

Supervised Regression for Ranking

Three algorithms used -

- Support Vector Regression
- Decision Tree Regressor
- Linear Regression with Regularisation

Supervised Learning to Rank

- Ranking objects for some locations
- Gives the optimal ordering
- Application >> Search engine ranking.

Approaches for LTR

- Pairwise approaches-
  a. Training samples: pairs
  b. Learning task: classification of object pairs into 2 categories (correctly ranked or incorrectly ranked)
- Listwise approaches-Listwise approaches directly look at the entire list of documents and try to come up with the optimal ordering for it.

RankNet

- A pairwise learning to rank approach
- Originally developed using neural nets
- The cost function for RankNet aims to minimize the number of inversions in ranking
- Optimizes the cost function using Stochastic Gradient Descent.
- Developed by Chris Burges and his colleagues at Microsoft Research.

A Probabilistic Ranking Cost Function

- Given a set of pairs of samples [A,B] in R^d, together with target probabilities P_{AB} .
- Consider model f : $R^d \longrightarrow R$, such that the rank order of a set of test samples is specified by the real values that f takes, specifically, f(x1) > f(x2).
- Denote the modeled posterior p(xi>xj) by p_{ij} , i,j =1...m.
- Let Pij be the desired target values for those posteriors.

Cross Entropy Cost Function

- Define $o_i \equiv f(xi)$ and $o_{ij} \equiv f(xi) - f(xj)$. We will use the cross entropy cost function,

$$C_{ij} \equiv C(o_{ij}) = -P_{ij} \log p_{ij} - (1 - P_{ij}) \log (1 - p_{ij})$$

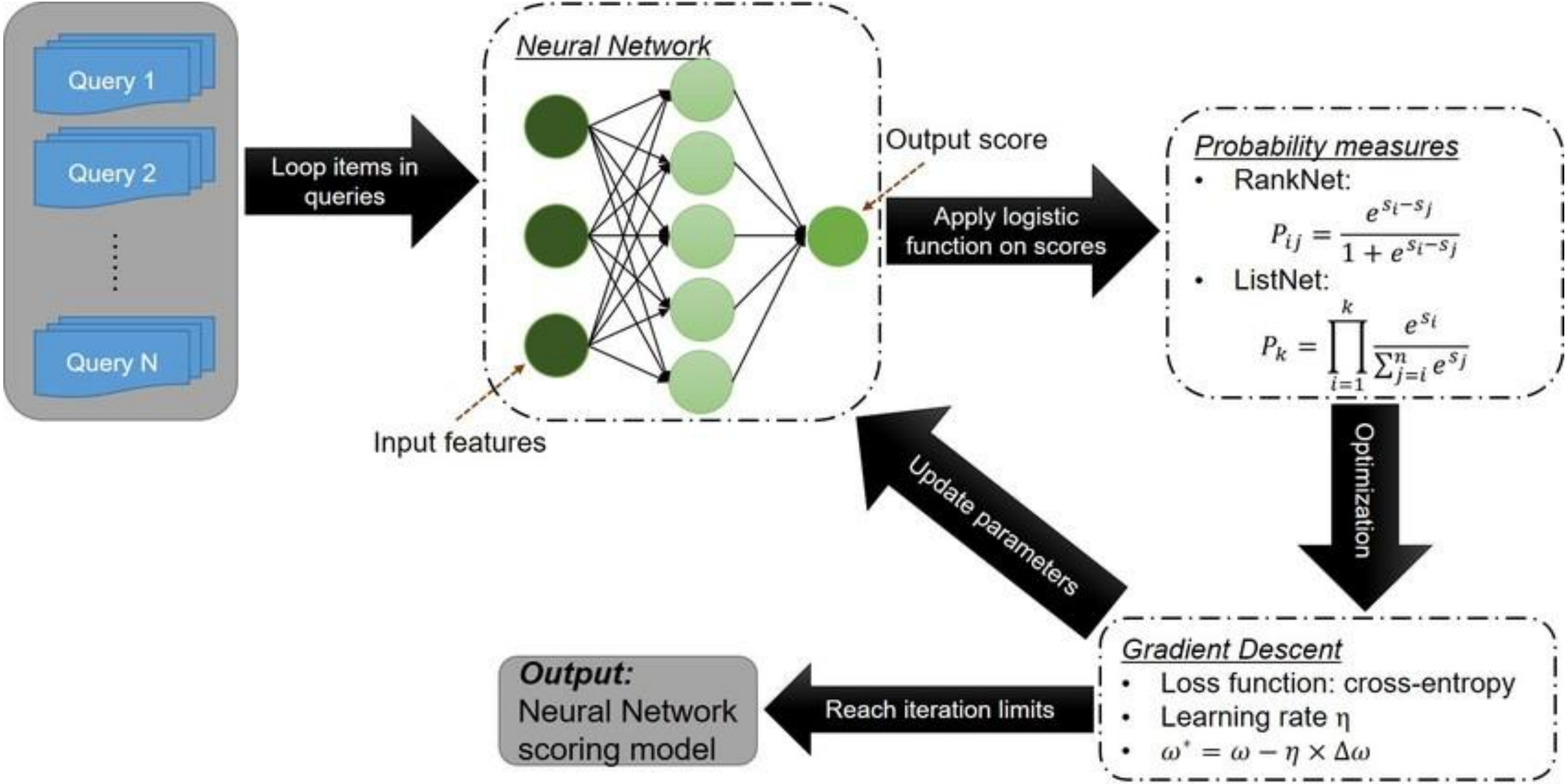
where the map from outputs to probabilities are modeled using a logistic function,

$$p_{ij} = e^{o_{ij}} / (1 + e^{o_{ij}})$$

C_{ij} then becomes,

$$C_{ij} = -P_{ij} o_{ij} + \log(1 + e^{o_{ij}})$$

Rank net Flowchart



RankNet: Learning to Rank with Neural Nets

Using back-propagation equations for a two layer net with q output nodes,

- Let the cost function be $\sum_{i=1}^{q} f(o_i, t_i)$.
- Here outputs of net are o_i, targets by t_i
- If α_k are the parameters of the model, then a gradient descent step amounts to $\delta\alpha_k = -\eta_k * \partial f / \partial \alpha_k$, η_k are positive learning rates.
- The net embodies the function,

$$o_i = g^3\left(\sum_j w_{ij}^{32} g^2\left(\sum_k w_{jk}^{21} x_k + b_j^2\right) + b_i^3\right) \equiv g_i^3$$

Contd.

- g^j activation function of each node in the jth layer of node
- weights w, bias b, upper indices index the node layer and the lower indices index the nodes within each corresponding layer
- Taking derivatives of f with respect to the parameters,

$$\frac{\partial f}{\partial b_i^3} = \frac{\partial f}{\partial o_i} g_i'^3 \equiv \Delta_i^3 \qquad \frac{\partial f}{\partial b_m^2} = g_m'^2 \left(\sum_i \Delta_i^3 w_{im}^{32} \right) \equiv \Delta_m^2$$

$$\frac{\partial f}{\partial w_{in}^{32}} = \Delta_i^3 g_n^2 \qquad \frac{\partial f}{\partial w_{mn}^{21}} = x_n \Delta_m^2$$

where x_n is the nth component of the input.

Final Phase

Turning now to a net with a single output, the above is generalized to the ranking problem as follows,

- Cost function becomes a function of the difference of the outputs of two consecutive training samples: $f(o_2 - o_1)$
- The gradient of the cost is then $\partial f/\partial \alpha = (\partial o_2/\partial \alpha - \partial o_1/\partial \alpha)\, f'$
- Assumption, the first pattern is known to rank higher than, or equal to, the second.

Contd.

- Again taking the derivatives for the back propagation,

$$
\begin{aligned}
\frac{\partial f}{\partial b^3} &= f'(g_2'^3 - g_1'^3) \equiv \Delta_2^3 - \Delta_1^3 \\
\frac{\partial f}{\partial w_m^{32}} &= \Delta_2^3 g_{2m}^2 - \Delta_1^3 g_{1m}^2 \\
\frac{\partial f}{\partial b_m^2} &= \Delta_2^3 w_m^{32} g_{2m}'^2 - \Delta_1^3 w_m^{32} g_{1m}'^2 \\
\frac{\partial f}{\partial w_{mn}^{21}} &= \Delta_{2m}^2 g_{2n}^1 - \Delta_{1m}^2 g_{1n}^1
\end{aligned}
$$

- Thus training RankNet is accomplished by a straightforward modification of back-prop

Result & Conclusion:

- We got the best result from SVR (R2 Score = 0.74).
- Knowledge on the semantics of geo-tagged venues can provide effective data representations to model the commercial value of urban area.
- Combination of geographic and mobility features in the yields superior performance suggesting that the dynamics of human movement matter in understanding the retail quality of an area.

References

1. Geo-Spotting: Mining Online Location-based Services for Optimal Retail Store Placement, Karamshuk et. al
2. “Stability in Competition” by Hotelling (1929)
3. “Changing methods of location planning for retail companies” Clarke , 1998
4. “The Effect of Retail Store Environment on Retailer Performance, Kumar and Karande”, (2000)
5. Should retailers adjust their micro-marketing strategies to type of outlet? An application to location-based store space allocation in limited and full-service grocery stores, Katia Campo, Gijsbrechts (2004)
6. Pablo Jensen. Network-based predictions of retail store commercial categories and optimal locations. Phys. Rev. E, 74:035101, Sep 2006.