Location-based Predictive User Activity Models

Proposal for machine learning nano-degree course

October 2017

Proposal

In this project, a predictive model is developed that can power users' intelligent personal assistants. The model enables use cases which need to predictively anticipate the user's desired activities and preferences at each point in time. Effectively, such activities constitute the user's personal *habits*. Furthermore, the model is context-aware and is trained based on the user's past activities. The model can be continuously trained as future training data becomes available. Finally, the predictive model offers a simple and standard interface allowing it to be plugged programmatically to a variety of "intelligent" systems. Such systems (not covered in this proposal) could use the model's knowledge of user's habits for scenarios including social planning (e.g., when a group of friends are available for a social gathering), or opportunistic marketing (e.g., identifying the needs of a user given his/her anticipated future activity/needs) and alike.

Domain Background

Intelligent agents will be an integral part of our lives in the future. These agents will be able to observe our day-to-day life style and activities, sense and gather the necessary contextual information, and offer proactive suggestions and predictions based on our recurring habits.

Much of these "intelligent" functionality will be based on building a predictive model of our activities and preferences. For example, an agent familiar with our lifestyle might be able to predict our commute from work back home, and suggest or organize recreational or social activities with our fiends (through their agents). Such predictions could be done automatically and particularly without requiring the user to explicitly define the habits. The space of such possibilities is expansive and broad, and can be individualized or group-oriented, depending on the scenarios.

The key enabling technology for such intelligent agent is the ability to learn from our past activities. This proposal addresses this by developing a model capable of supporting such scenarios.

Problem Statement

The problem addressed in this proposal is to determine what activity a user is inclined to do at a given point in time. This determination is based on the user's past behavior and their context at that particular point in time. We rely on the user's location as an indication of what the activity is. For example, if at work, the user is presumed to be working. Likewise, if at a restaurant, the user is presumed to be eating.

As such, the user data conveying the following pieces of information is presented to the model for training purposes:

```
UserId: list of <DateTimeBag, LocContext>
```

The `UserId` identifies the user; `TimeDateBag` is the time the activity took place; `LocContext` identifies the user location and the duration of stay.

For simplicity, we focus on only one user (`UserId=0000`). Also, we use a decomposition of the `DateTimeBag` element into `'year', 'month', 'hour', 'partofday', 'weekday'`. Finally, we use a simplified `LocContext` that is based on latitude, longitude. We use an online service (offered by Google) to translate location to landmarks, e.g., `Home, Office, Central-Park`.

For predictive purposes, a separate mode is trained for each user, and is subsequently presented that following data pertaining to that user:

Mode for `UserId`:
`<cluster, year, month, hour, weekday>`

where ‘`cluster`’ is a constructed feature based on the unsupervised clustering of user location (using KMeans clustering), and the remaining features are from user’s `TimeDateBag`.

The model then performs a multi-label classification where each predicated label is a previous user `Activity` (in our case, location).

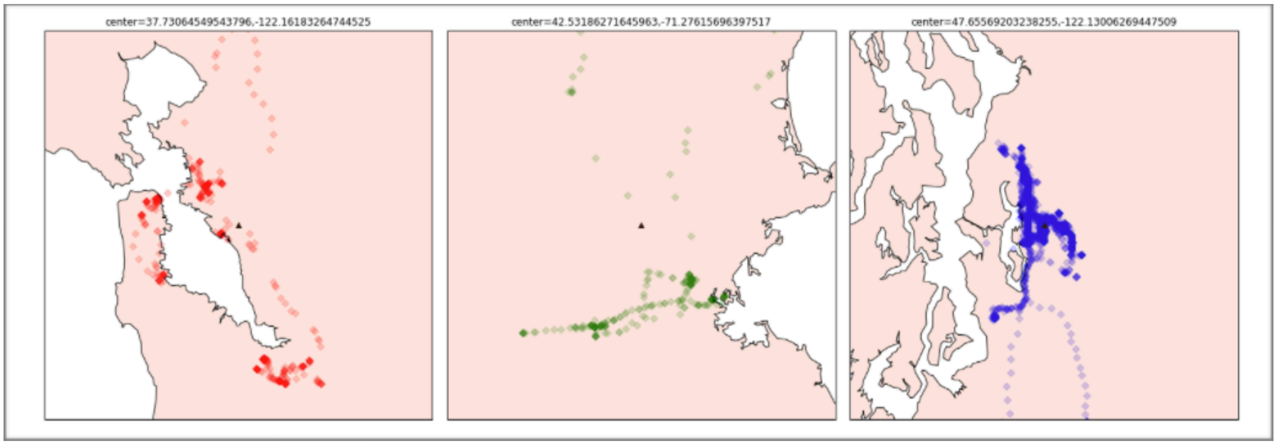
Datasets and Inputs

The raw datasets are in json format (a separate json file per user). The dataset is in the following format:

```
{
  "locations" : [ {
    "timestampMs" : "1509141279901",
    "latitudeE7" : 476373620,
    "longitudeE7" : -1221356482,
    "accuracy" : 65,
    "altitude" : 110,
    "verticalAccuracy" : 10
  },
  {
  ...
  }]
}
```

The present dataset captures a single user’s real world activities over 3 months and contains over 9000 datapoints. As evident in the diagram below, a user’s movements across different geographical locations is highly clustered. For example a main cluster of movements is formed around the work and home location of the user. Another cluster of movements is formed when the user goes on vacation in different parts of the world.

The data is preprocessed to extract the user’s duration of stay between successive movements (based on difference between successive datapoints), as well as `year, month, hour, weekday`. Additionally a new field is derived indicating whether the user has been stationary in one location. The stationary criteria is to stay in one location for more than 5 minutes (this removes data points that capture user’s commute, for example, which is not significant for our


center=37.73064549543796,-122.16183264744525
center=42.53186271645963,-71.27615696397517
center=47.65569203238255,-122.13006269447509

predictive purposes). The post-processed data has over 3000 data points where the user is stationary. These data points are further filtered to capture frequently visited places where the user has visited at least 10 times. The final outcome consists of 12 frequently visited locations.

Anonymization of data:
The user identifier already mask the identity of users (the data does not belong to me but has been received from a friend by his permission). However, further anonymization can still be done by concealing the latitude/longitude of the location data to hide the locations visited by users (this can prevent reverse-engineering of user identity based on movements). However, with data owner's explicit permission this step has been deemed unnecessary.

Solution Statement

The solution consists of three steps:

1. Pre-processing of data to extract frequently visited locations by the user and augment the input data with duration of stay and other features not in the raw data format (e.g., `year, month, hour, weekday`).
2. Clustering of user location using Means clustering to identify concentration of movements for the user pertaining to normal daily movements vs. movements pertaining to occasional travels and vacations.
3. Training of a decision tree classifier based a variety of feature sets available in the original or post-processed dataset. The output of the classifier indicates the predicted location of the user (and implicitly the activity associated with that location).

The choice of decision tree classifier has been to facilitate development of simple models that do not overfit the data (given the relatively small number of data points per user).

The model is based on a pre-processing and classification pipeline. The preprocessing step will convert the `Timestamp` to `DateTimeBag`, and `LocGPS` to `LocContext` (landmarks pre-defined by user will be used and missing ones will be queried from Google Maps API). Finally, the classification pipeline uses the processed data points for training a classification model (likely a random forest).

Benchmark Model

The benchmark model predicts user location based on the distribution of user's visits to frequently visited location. As such, this benchmark model ignores user's current location and predicts the user's next location using the frequency of previous visits as weights. We refer to this benchmark as naive model.

The initial evaluation of the naive model over 100 runs shows the accuracy of the predictions is on average 7% ranging from min of 0% to max of 23%.

```
Number of runs: 100
Min accuracy: 0.0
Avg accuracy: 0.07615384615384616
Max accuracy: 0.23076923076923078
```

Evaluation Metrics

The benchmark is most likely based on the accuracy score on a test dataset not used for training. This metric is chosen since the false predictions are important impediment to user experience. As such it is most important to ensure accuracy of predictions by thoroughly evaluating the performance of the predictive models (and naive model as benchmark).

Project Design

The model is exposed as a python module that can load raw datasets in csv format, train over the loaded data, and be queried for predictions on future data. The implementation uses `sklearn` library and digests the

Optionally, the trained model can be saved and restored from file.
Optionally, the model can be trained inline.
Optionally, the model can produce a set of predictions over a probability distribution.