1.5.3 Project 3: Traffic Monitoring

This project aims to provide helpful information about traffic in a given geographic area based on the history of traffic patterns, current weather, and time of the day. Such information can be used by automobile drivers in choosing the best time to travel a given route or the best route at a given time.

Most traffic information services (e.g., Yahoo! Traffic, Traffic.com) only provide current information about traffic conditions in a given area. While current information is essential, these reports are often incomplete because their sources frequently fail to report the current traffic conditions. Hence, the user cannot assume that there is no heavy traffic in a given location simply because that location was not reported by these services.

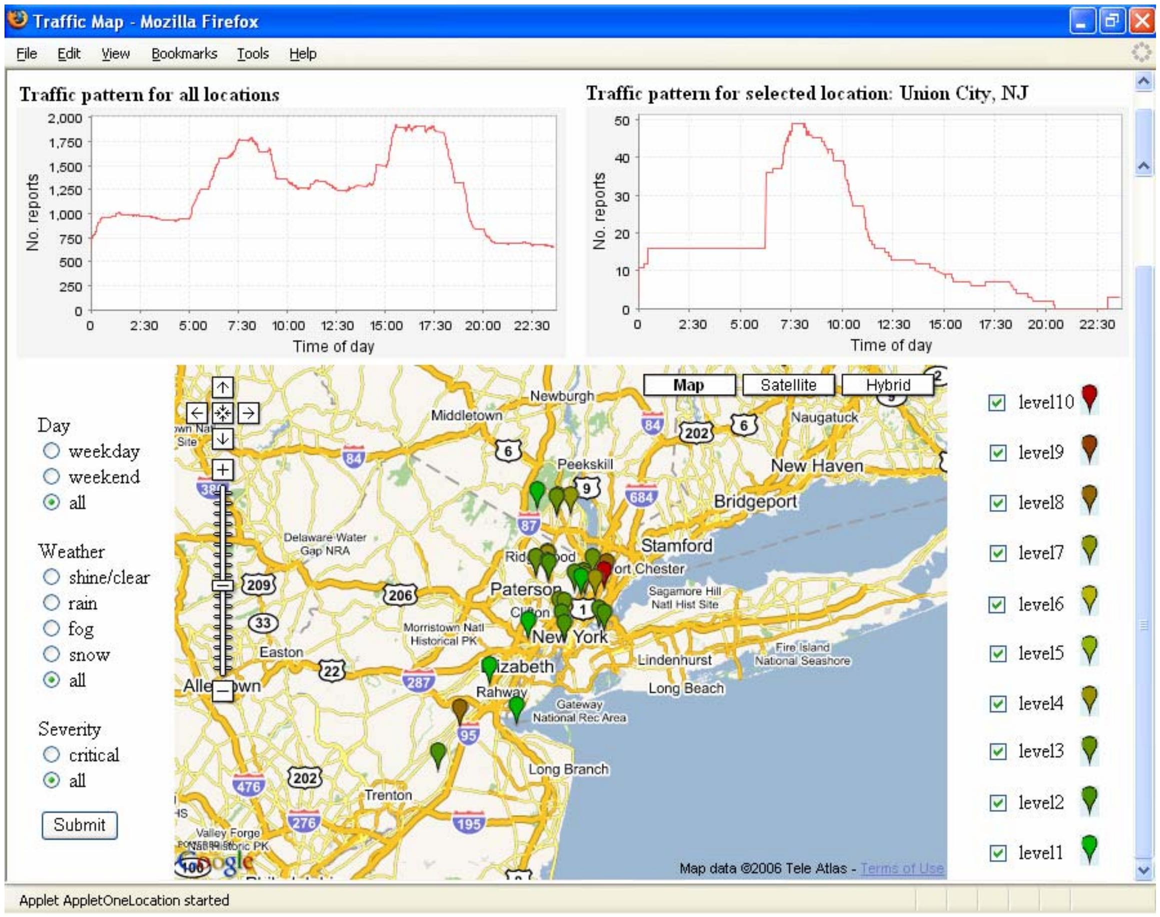
The idea is to analyze historic traffic information collected over a long period of time to highlight locations which appear most frequently in the traffic advisories and thus are likely to be subject to heavy traffic conditions even if they do not appear in the current traffic advisory. Here are specific examples of services that could be offered:

- A "traffic map" which reviews historic traffic "hotspots" across a certain area, given the choice of day(s) of the week, weather conditions, and severity of incidents (Figure 1)
- Historic traffic patterns along a path (route) between two end-points, given time of the day and weather conditions (Figure 2)
- If the desired route is anticipated to suffer heavy traffic for the specified travel period, the system could
    - Offer a detour
    - Suggest an alternative period to travel along the current route.

Traffic pattern could be defined as the function describing the number of reported traffic incidents with respect to time of day and weather conditions.

Statement of Requirements

The system should collect traffic and weather information over a given geographic area. The user will be able to view two types of statistics about the traffic incidents:


Traffic Map - Mozilla Firefox
File Edit View Bookmarks Tools Help
Traffic pattern for all locations
No. reports
Time of day
Traffic pattern for selected location: Union City, NJ
No. reports
Time of day
Map
Satellite
Hybrid
Map data ©2006 Tele Atlas - Terms of Use
Applet AppletOneLocation started

Day
- weekday
- weekend
- [✓ all]

Weather
- shine/clear
- rain
- fog
- snow
- [✓ all]

Severity
- critical
- [✓ all]

Submit

- ✓ level10
- ✓ level9
- ✓ level8
- ✓ level7
- ✓ level6
- ✓ level5
- ✓ level4
- ✓ level3
- ✓ level2
- ✓ level1

Figure 1: Traffic "hotspots" across a given area. Symbol color encodes traffic intensity.

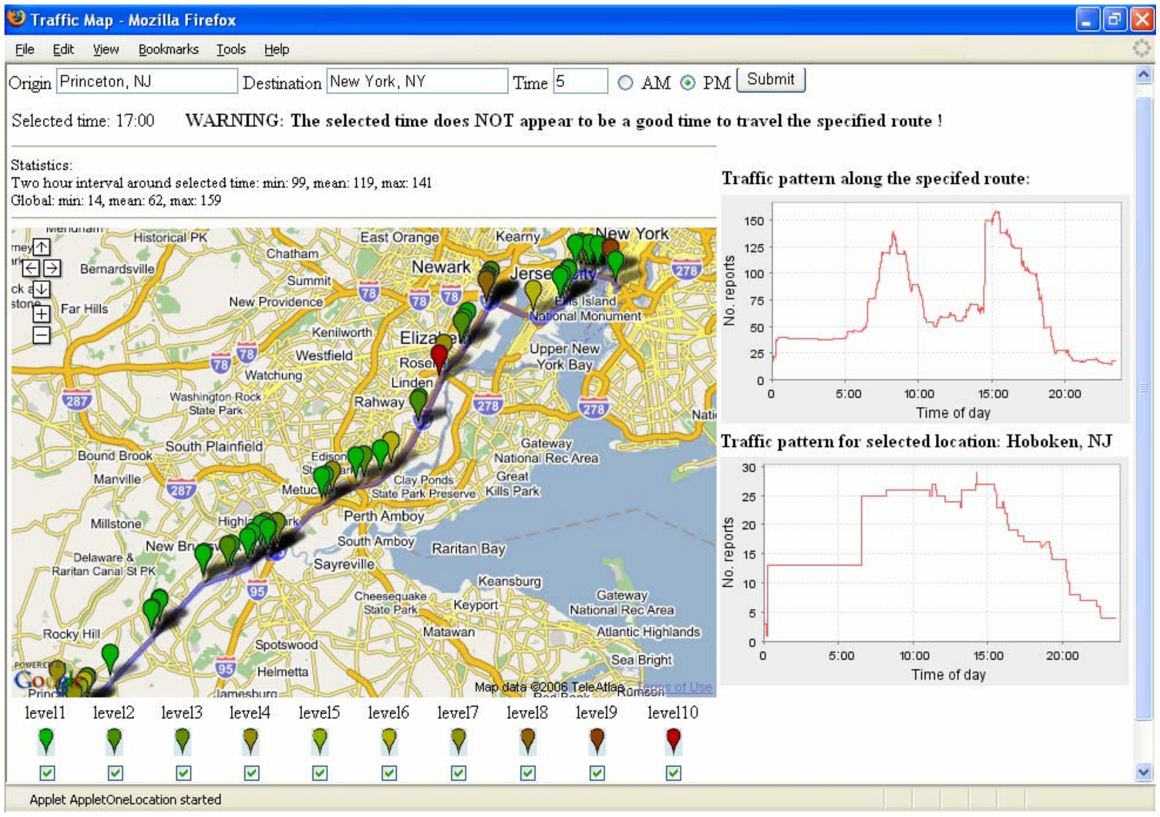
SERVICE 1: Statistics across the Entire Area

The user should make the following choices:

1. Day of the week (weekday vs. weekend vs. all)
2. Weather conditions (shine/clear vs. rain vs. fog vs. snow vs. all)
3. Incident severity (critical vs. all)

For example, the user may be interested to know the traffic statistics of critical severity incidents for rainy weekdays across the given area. Given the user-selected input parameters, the system shall extract the historic statistics and visualize the results on top of a geographic map of the area, similar to Figure 1. The numeric values of the mean and variance can be visually encoded by color and size of the graphical markers shown on the map. For example, marker color can range from *green* (level 1) for the smallest mean[1] number of reports to *red* for the highest mean number of reports (level 10). The marker size can be used to encode the variance.

[1] As discussed in Extensions below, computing the means may not be the best way to express the statistical properties of traffic incidents, i.e., it is not known whether these follow a normal or skewed distribution.


Origin Princeton, NJ Destination New York, NY Time 5 AM [✓ PM] Submit

Selected time: 17:00 WARNING: The selected time does NOT appear to be a good time to travel the specified route !

Statistics:
Two hour interval around selected time: min: 99, mean: 119, max: 141
Global: min: 14, mean: 62, max: 159

Traffic pattern along the specifed route:

Traffic pattern for selected location: Hoboken, NJ

Applet AppletOneLocation started

Figure 2: Traffic history along a given path. Symbol color encodes traffic intensity.

The user should be able to choose to show/hide the markers corresponding to any of the 10 different levels of traffic severity (mean number of reports) by checking/un-checking the check boxes shown below the map.

The system should also show a plot of the traffic pattern with respect to time of day, i.e., the mean number of traffic incidents as a function of time of day. Example is shown on top of Figure 1.

SERVICE 2: Statistics along a Given Route

The route is to be specified by giving the starting and destination addresses, similar to Figure 2.

Data Collection and Preprocessing

Data collection should commence early on in the semester so to have adequate statistics by the end of the project[2]. Ideally, data collection should be continuous over several years, but several months should give reasonable results. To initiate the data collection process in the soonest possible time, here I provide a tentative design for the data collection program. The readers are welcome to adopt their own data collection procedure.

[2] Normally, this description would not be part of the problem statement. The reason it is included here is to facilitate data collection as early as possible in the course of the project.

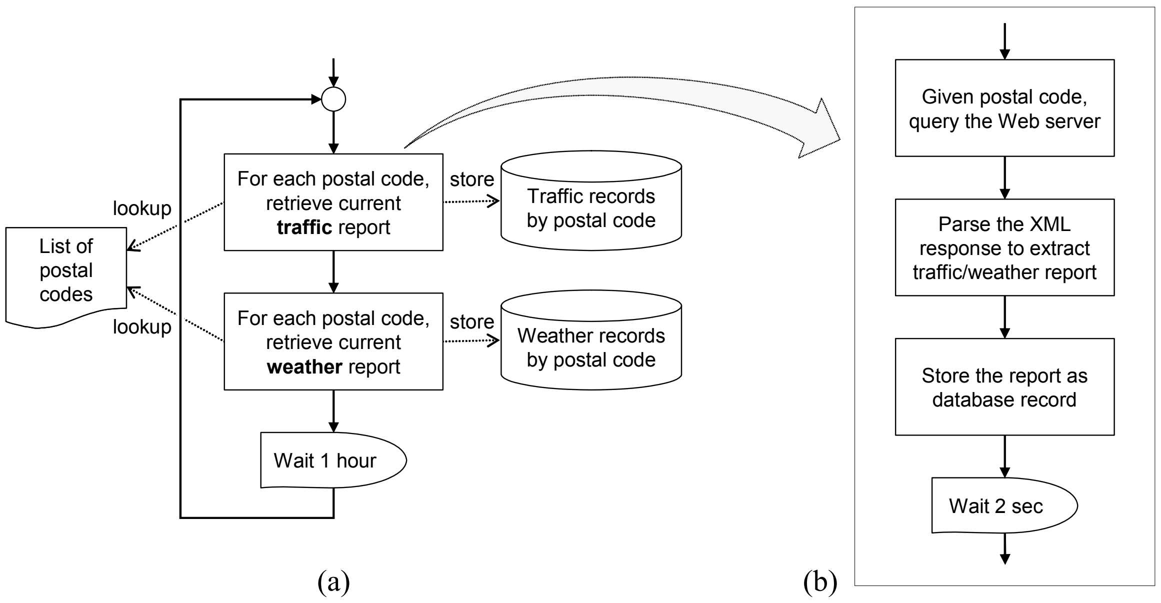

lookup
List of postal codes
lookup
For each postal code, retrieve current **traffic** report
store
Traffic records by postal code
For each postal code, retrieve current **weather** report
store
Weather records by postal code
Wait 1 hour
Given postal code, query the Web server
Parse the XML response to extract traffic/weather report
Store the report as database record
Wait 2 sec
(a)
(b)

Figure 3: Data collection algorithm: (a) overall design; (b) web report retrieval module.

The data collection algorithm design is shown in Figure 3. The program runs in an infinite loop, periodically retrieving the traffic and weather reports for the specified cities. The student should prepare a list of postal zip codes to cover the area for which the traffic monitoring tool will be used. This need not include every zip code in the area because the web traffic reports are relatively coarse and cover a significantly larger area surrounding the zip code. After retrieving all the reports and storing them in a local database, the program waits for a given period of time, shown as one our in Figure 3(a), and repeats the procedure.

The web report retrieval modules for traffic and weather reports are very similar, as shown in Figure 3(b). Notice that there is a 2-second waiting period inserted at the end of each zip code's data retrieval to avoid the web server mistaking our program for a denial-of-service attack, which would happen if a large number of requests were posted in a short time period.

Current traffic conditions reports can be downloaded from the Yahoo! Traffic service: http://maps.yahoo.com/traffic. This website provides traffic information for a target zip code in the RSS XML format. The service is accessed using HTTP GET request (see Appendix C) with the following URL:

http://maps.yahoo.com/traffic.rss?csz=⟨*zipCode*⟩&mag=⟨*magnification*⟩&minsev=⟨*minimumSeverity*⟩

The parameters are as follows:

- csz=⟨*zipCode*⟩ The target area for which to retrieve traffic information. You can provide a zip code, a city name, or an address. Here we assume that zip codes are used.
- mag=⟨*magnification*⟩ The level of "magnification" for the displayed map. Allowed values are 3, 4, and 5, corresponding to 4 miles, 10 miles, and 40 miles, respectively.
- minsev=⟨*minimumSeverity*⟩ The minimum severity of the reported traffic incidents. Allowed values are 1 through 5, with the following significance: 1 ≡ Minor, 2 ≡ Moderate, 4 ≡ Major, 5 ≡ Critical. If such a value is specified, only those incidents with a

Table 1: Database schema for traffic reports.

```
+-----------------+---------------------+------+-----+---------------------+----------------+
| Field           | Type                | Null | Key | Default             | Extra          |
+-----------------+---------------------+------+-----+---------------------+----------------+
| id              | int(10) unsigned    |      | PRI | NULL                | auto_increment |
| zipCode         | varchar(5)          |      |     |                     |                |
| latitude        | int(11)             |      | MUL | 0                   |                |
| longitude       | int(11)             |      |     | 0                   |                |
| titleHash       | int(11)             |      |     | 0                   |                |
| descriptionHash | int(11)             |      |     | 0                   |                |
| startTime       | datetime            |      |     | 0000-00-00 00:00:00 |                |
| endTime         | datetime            |      |     | 0000-00-00 00:00:00 |                |
| updateTime      | datetime            |      |     | 0000-00-00 00:00:00 |                |
| severity        | tinyint(3) unsigned |      |     | 0                   |                |
| title           | varchar(255)        | YES  |     | NULL                |                |
| description     | varchar(255)        | YES  |     | NULL                |                |
+-----------------+---------------------+------+-----+---------------------+----------------+
```

severity larger than the one requested will be included in the response. For example if we set this value to 4, only major and critical incidents will be reported.

For our purposes, it is suggested that the following parameters are used: csz= one of the zip codes from the list prepared for the region of interest; mag= 4 and minsev= 1. This allows us to capture all the incidents regardless of their severity even if some zip codes are missing. (The magnification may need to be adjusted to reflect the density of the towns in the region under consideration.)

The reader should consult the Yahoo! website for the format of the RSS XML response. Once the response is parsed, the extracted data are stored in a local database. The schema for the traffic database table is shown in Table 1, where the fields have the following meaning (`id` is the primary key and `zipCode` is explained above):

- `latitude` and `longitude` coordinates give the position of the particular incident
- `title` is a short title for the reported event
- `description` provides a short description of the reported event
- `titleHash` and `descriptionHash` can be used for quick equality comparisons to detect duplicate reports; they are hash codes of the contained strings
- `startTime` is the date and time when this event was reported to Yahoo! and published at their website
- `endTime` is the estimated date and time when this event will end (example: the accident will be cleared or the construction will end)
- `updateTime` is the date and time of the last update for this particular event

Data duplication of traffic incidents is common as many traffic incidents are current for more than one hour. Because we are retrieving a full list of incidents every hour, we are likely to collect the same information multiple times. Furthermore, as noted earlier, traffic reports for a given zip code usually cover a much larger area around that zip code, which results in data duplication because of overlapping regions covered by the reports for different zip codes. Data duplication represents a problem for our project because we compute the traffic pattern for a location based on the number of traffic incidents reported for that location. Data duplication would artificially inflate these scores and grossly skew the results.

Table 2: Database schema for weather reports.

```
+-------------+---------------------+------+-----+---------------------+----------------+
| Field       | Type                | Null | Key | Default             | Extra          |
+-------------+---------------------+------+-----+---------------------+----------------+
| id          | int(10) unsigned    |      | PRI | NULL                | auto_increment |
| zipCode     | varchar(5)          |      | MUL |                     |                |
| time        | datetime            |      |     | 0000-00-00 00:00:00 |                |
| description | varchar(40)         |      |     |                     |                |
| temperature | tinyint(3) unsigned |      |     | 0                   |                |
+-------------+---------------------+------+-----+---------------------+----------------+
```

Before inserting a new traffic incident into the database, we first check if a report already exists in the database with the same latitude, longitude, title and start time. If such a report does not exist, the incident is added to the table. If a report with these parameters is found, we assume it refers to the same incident and we check the update times of the two reports. If the update time of the current report is later than the one in the database, the record in the database is updated with the new information; otherwise, the newly reported incident is discarded.

We may also wish to remove reports of road construction from the analysis. These reports are not really indicative of the traffic patterns for a specific location. They do affect traffic for the duration they are in effect but, because they do not occur regularly, they are not good predictors for future traffic situations. Construction-related items can be identified by parsing the title or description of the item for keywords as "construction," "work," and "maintenance."

The weather data collection process is quite similar to traffic data collection. Weather reports by zip code can be obtained from Weather.com. Their RSS page is available at http://www.weather.com/weather/rss/subscription?from=footer&ref=/index.html. The weather reports can be accessed from the following URL using a HTTP GET request:

http://rss.weather.com/weather/rss/local/⟨*zipCode*⟩?cm_ven=LWO&cm_cat=rss&par=LWO_rss

The ⟨*zipCode*⟩ parameter specifies the area of interest. The significance of the other parameters (cm_ven, cm_cat and par) is not disclosed, so we just use default values. The database schema is shown in Table 2, where the `time` field contains the time when this report was collected and `description` is description of the weather conditions, such as sunny, heavy rain, cloudy, etc.

Data duplication for weather data does not pose a problem because weather is auxiliary information that is being added to the traffic reports. Having duplicate items here should not affect the traffic reports accuracy.

It should be noted that traffic and weather reports are not synchronized, because of using different providers for traffic and weather data. This means that for a traffic incident reported at time t we may not have a weather report with the same time stamp. Precise synchronization is not critical because, usually, weather does not change very rapidly. So it is acceptable to match a traffic report with time stamp t with a weather report with time stamp $t \pm x$, where x is a short time interval. Because we are collecting both traffic and weather data every hour, we can find a matching weather report within an hour of the reported traffic incident.

Extensions

At present, the user can view only the past data statistics but not the current traffic or weather information. Develop the facilities for the user to view the current information. This can be useful when viewing the statistics along a given route, so the user can inspect the current conditions.

We do not really know what could be useful to drivers in terms of historic characterization of traffic. This description only provides initial suggestions and the student should be creative and interview drivers to discover with other ways to process and present traffic information.

There are also user-interface issues: how to design the system so that it can be quickly and safely used (via a built-in, touch-sensitive screen) while driving on a congested road.

Studying traffic statistics is interesting in its own right. This tool can be repositioned to assist in studying traffic-related statistics, from incidents to intensity.

Domain Fieldwork

Developing a system to meet the above requirements may be the easiest thing in this project. A key question is: who should benefit from this system and in what way? At this point I am not sure whether the system would be valuable to local commuters, people who are new to the area (tourists?), or someone else. Perhaps it could be useful to a person moving to a new area, to observe how traffic has changed over the past five years? Or, when getting a new home, to consider driving distances and the average rush-hour commute times for the past six months? The developers should interview different potential users and ask for suggestions. Brainstorming or focus groups could be employed, as well. Perhaps the most effective way to get help is to demonstrate a working prototype of the system.

Additional Information

See also the book, Problem 2.7 at the end of Chapter 2, the solution of which can be found at the back of the text.

Yahoo! Maps, online at: http://maps.yahoo.com/

Yahoo! Maps Web Services – Introducing the Yahoo! Maps APIs: http://developer.yahoo.com/maps/

Google Maps, online at: http://maps.google.com/

Google Maps API, online at: http://www.google.com/apis/maps/

GoogleMapAPI – A library used for creating Google maps:
http://www.phpinsider.com/php/code/GoogleMapAPI/

No jam tomorrow? Transport: New techniques are being developed to spot existing traffic jams, predict future ones, and help drivers avoid both kinds. From *The Economist* print edition, Sep 15th 2005, Online at: http://www.beatthetraffic.com/aboutus/economist.htm

A system called Beat-the-Traffic, developed by Triangle Software of Campbell, California. A traffic-prediction system called JamBayes developed by Eric Horvitz of Microsoft Research.

E. Horvitz, J. Apacible, R. Sarin, and L. Liao, "Prediction, expectation, and surprise: Methods, designs, and study of a deployed traffic forecasting service," *Proceedings of the Conference on Uncertainty and Artificial Intelligence 2005*, AUAI Press, July 2005. Online at: http://research.microsoft.com/~horvitz/jambayes.htm