Traffic Event Prediction

Task Overview

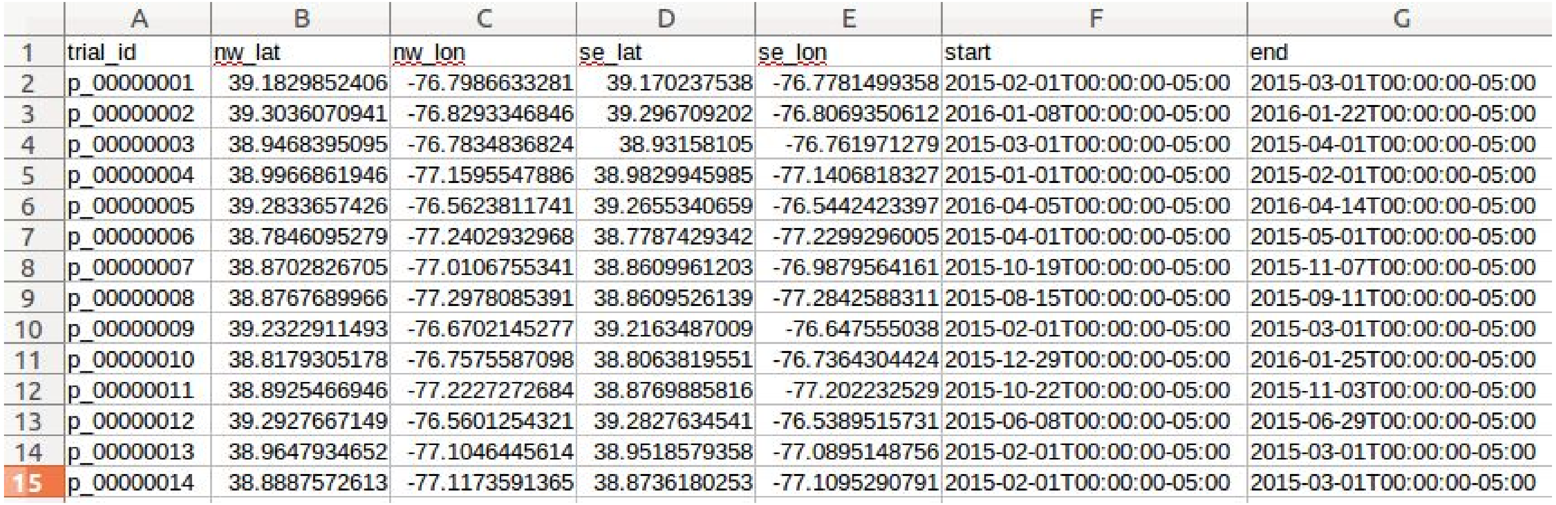
- Description

  Given event information in the past years, you are asked to predict future event occurrence.

- Event Data
  - event_id: uniquely identify an event, e.g. “MDOT_CHART_4aff02b300110095003f0be8b3035daa”
  - event_description: a text description about an event, e.g. “Disabled Vehicle Event @ I-495 AT MD 187”
  - Timestamps: times the event was created, confirmed, and closed (some are missing). You should use **closed_timestamp** in this project.
  - event_type: the type of an event, e.g. “accidentsAndIncidents”.
  - geographical location: (latitude, longitude)
  - There are 9 other less important fields.

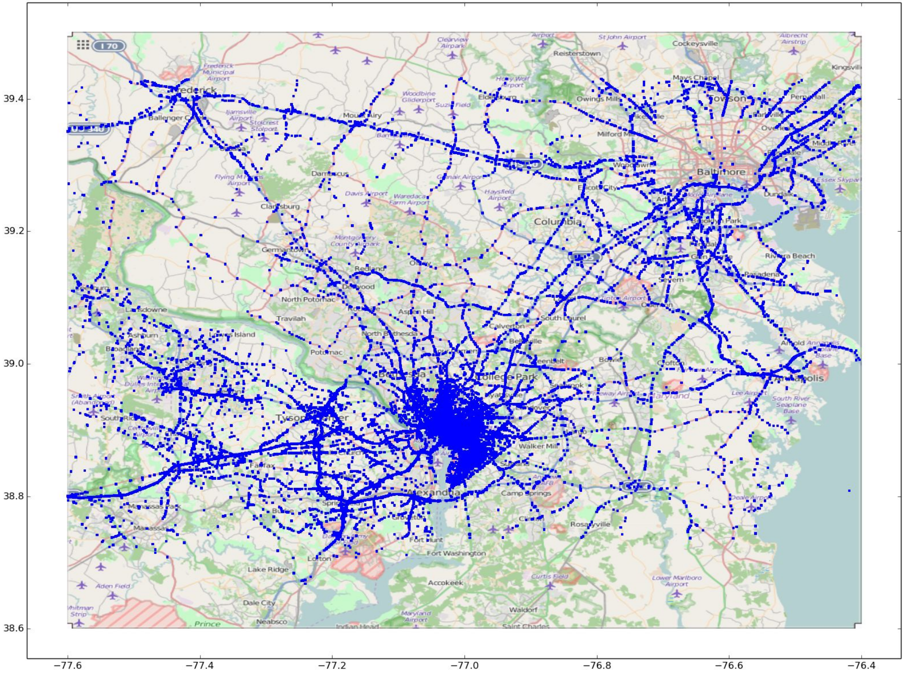
Task Overview

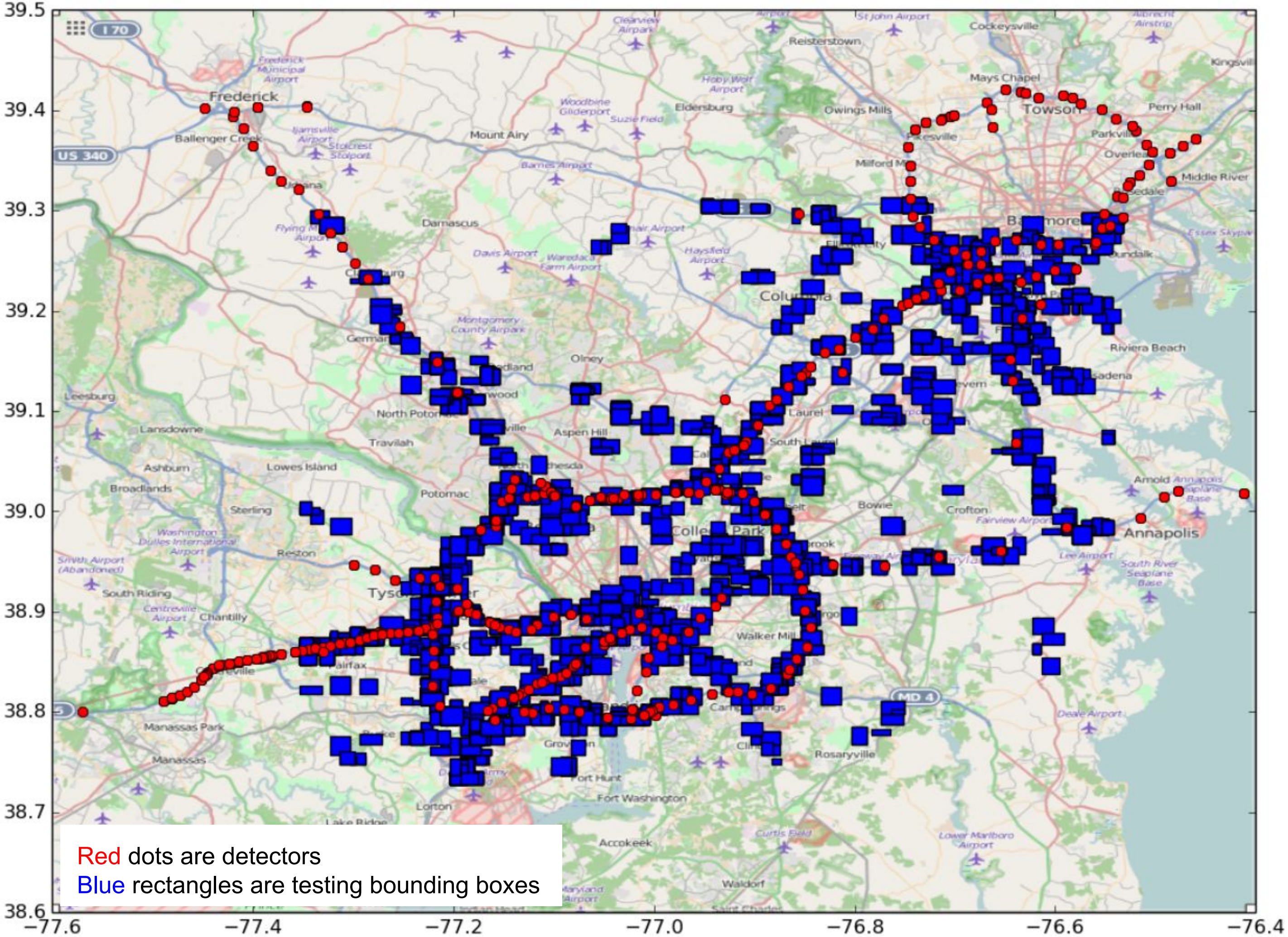
- Testing Trials
    - Each trail specifies a rectangle bounding box and time interval of one month between 2015 and 2016.



	A	B	C	D	E	F	G
1	trial_id	nw_lat	nw_lon	se_lat	se_lon	start	end
2	p_00000001	39.1829852406	-76.7986633281	39.170237538	-76.7781499358	2015-02-01T00:00:00-05:00	2015-03-01T00:00:00-05:00
3	p_00000002	39.3036070941	-76.8293346846	39.296709202	-76.8069350612	2016-01-08T00:00:00-05:00	2016-01-22T00:00:00-05:00
4	p_00000003	38.9468395095	-76.7834836824	38.93158105	-76.761971279	2015-03-01T00:00:00-05:00	2015-04-01T00:00:00-05:00
5	p_00000004	38.9966861946	-77.1595547886	38.9829945985	-77.1406818327	2015-01-01T00:00:00-05:00	2015-02-01T00:00:00-05:00
6	p_00000005	39.2833657426	-76.5623811741	39.2655340659	-76.5442423397	2016-04-05T00:00:00-05:00	2016-04-14T00:00:00-05:00
7	p_00000006	38.7846095279	-77.2402932968	38.7787429342	-77.2299296005	2015-04-01T00:00:00-05:00	2015-05-01T00:00:00-05:00
8	p_00000007	38.8702826705	-77.0106755341	38.8609961203	-76.9879564161	2015-10-19T00:00:00-05:00	2015-11-07T00:00:00-05:00
9	p_00000008	38.8767689966	-77.2978085391	38.8609526139	-77.2842588311	2015-08-15T00:00:00-05:00	2015-09-11T00:00:00-05:00
10	p_00000009	39.2322911493	-76.6702145277	39.2163487009	-76.647555038	2015-02-01T00:00:00-05:00	2015-03-01T00:00:00-05:00
11	p_00000010	38.8179305178	-76.7575587098	38.8063819551	-76.7364304424	2015-12-29T00:00:00-05:00	2016-01-25T00:00:00-05:00
12	p_00000011	38.8925466946	-77.2227272684	38.8769885816	-77.202232529	2015-10-22T00:00:00-05:00	2015-11-03T00:00:00-05:00
13	p_00000012	39.2927667149	-76.5601254321	39.2827634541	-76.5389515731	2015-06-08T00:00:00-05:00	2015-06-29T00:00:00-05:00
14	p_00000013	38.9647934652	-77.1046445614	38.9518579358	-77.0895148756	2015-02-01T00:00:00-05:00	2015-03-01T00:00:00-05:00
15	p_00000014	38.8887572613	-77.1173591365	38.8736180253	-77.1095290791	2015-02-01T00:00:00-05:00	2015-03-01T00:00:00-05:00

Training Events Distribution (based on 2014)





Red dots are detectors
Blue rectangles are testing bounding boxes

Event Types

We are interested in 6 different types of events (though there are more in the given data), they are:

1. Accidents and Incidents (A).
2. Roadwork (R).
3. Precipitation (P).
4. Device Status (D).
5. Obstruction (O).
6. Traffic Conditions (T).

Task Description

- Foreach each of the testing trial **T** = (Geo_Box, Time_Interval) given in prediction_trials.tsv:
  - Foreach event type **et** in {A, R, P, D, O, T}:
    - Count number of occurred events of type **et** within Geo_Box by years, resulting in **D** = {(year, count)}.
    - Train a regression model **M** (either linear or polynomial or others) based on **D**.
    - Extract the year **Y** (e.g. 2015 or 2016) from Time_interval of **T**.
    - Use the M to predict #events (denoted as **NUMe**) will occur in year **Y**.
    - Since **NUMe** is for the entire year **Y**, but we want to predict #events within Time_Interval (all intervals are of length 1 month), so the final predicted #events within Time_Interval can be estimated as averaged event of that year within 1 month: predicted = **NUMe**/12.

- The above algorithm generates 6 predicted numbers for each testing trial, since we are interested in 6 types of events {A, R, P, D, O, T}. There are around 300K trials, so you may want to parallelize the computation of each trial (e.g. with distributed mapreduce or Python multiprocessing).

- You are encouraged to use your own methods. The above algorithm is a baseline one, and it is also for the purpose of clarifying the problem.

Model Validation

You may have several models in your mind, and want to test how individual model performs. Here the model validation is used to roughly estimate the quality of your models. To do this, you can use event data until 2013 as training, and then use your model to predict 2014. Since we have groundtruth event occurrence for 2014, we can calculate the prediction errors.

- Foreach each of the testing trial **T** = (Geo_Box, Time_Interval) given in prediction_trials.tsv:
    - Foreach event type **et** in {A, R, P, D, O, T}:
        - Count number of occurred events of type **et** within Geo_Box by years, resulting in **D** = {(year, count)}.
        - Split **D** into training and validation sets:

          D_Train = {(year,count) | year <= 2013}, D_Val = {(**2014**, **gt_count**)}.
        - Train a regression model **M** (either linear or polynomial or others) based on **D_Train**.
        - Use the M to predict #events (denoted as **predicted_count**) will occur in year **2014**.
        - Calculate the error of prediction: E = (**gt_count** - **predicted_count**)^2
- Calculate the averaged error AvgE for E over all trials and event types. The final model validation error is:

  ValError = sqrt(AvgE).

Submission

For each testing trial given in prediction_trials.tsv file, predict number of events will occur in given specified time periods (1 month). Submit a file named “prediction.tsv”, which the same number of lines as prediction_trials.tsv file. Each line should have exactly 6 floating numbers, splitted by a TAB, like:

```
10    15    9    2    49    6
```

Each line in prediction.tsv should correspond to testing trail in prediction_trials.tsv. The order of number should always be: Accidents and Incidents, Roadwork, Precipitation, Device Status, Obstruction, Traffic Conditions.

- Only one member needs to submit the results.
- Also submit a report (pdf file) describing all details.
- At the beginning of your report, you should report the validation error (**ValError**) of your models (you need to document all the models you have tested, but only submit one prediction.tsv with best validation performance).
- We only accept two files: **prediction.tsv** and **report.pdf**