Spatio-Temporal Scan Statistics

Chance Haycock & Edward Thorpe-Woods

June 2020

Contents

Notation

Symbol	Description
$\mathcal{D} \subset \mathbb{R}^2$	Spatial Domain in
$T \subset \mathbb{N}$	Time index set
W	Length of Time Domain; $T = \{0, 1, \ldots, W-1\}$
$\mathbf{S}$	Set of space-time regions in $\mathcal{D} \times T$
S	Space-time region in $\mathcal{D} \times T$
$s_i \in \mathbb{R}^2$	Spatial locations of detectors
c_i^t	Count data at spatial location s_i at time t.
b_i^t	Baseline data found from time series analysis at spatial location s_i at time t.
H_0	Null Hypothesis
$H_1(S)$	Set of alternative hypothesis at space-time region $S \in \mathbf{S}$.
N	Number of partitions per spatial axis used to create grid.
M	Number of simulations used for randomisation testing.
D	Data

1 Introduction

1.1 Data and Applications

Here, we describe an implementation of the *Expectation-Based Scan Statistic* heavily based on work by Daniel Neill[Nei09]. It is hoped that this methodology can be successfully applied to the SCOOT data set, comprising of hourly vehicle count data of roughly 10,000 traffic sensors located in central London, enabling the detection of emerging space-time clusters in near real-time. As an extension use-case, the framework can also be applied to multi-variate *JamCam* data, comprising of spatio-temporally aggregated count data for cars, bikes, buses, trucks and individuals. With many variations of scan statistics in the literature, we aim to create a unique methodology; using ideas from a number of sources.

1.2 Variations of Scan Statistics

Several variations of the Scan Statistic have been published; the below four publications cover a large range of methodologies for real-monitoring of spatial time series data. In this work, we initially focus on implementing the approach as described in [Nei09], and generalise the model using other methods in the literature.

Acronym	Freq/Bayes	Description	Year	Citations
ESS [Nei09]	Freq	Expectation-Based Spatio-Temporal - TSA	2009	87
BSS[NMC06]	Bayes	Bayesian Spatial-Only Scan	2006	143
MBSS[NC10]	Bayes	Multi-Variate Bayesian Spatio-Temporal Scan	2010	89
MDGSS[NMPM05]	Freq	Generalised metrics - introduces Fast Scan	2005	69

The general set-up is as follows:

Suppose we are given hourly count data c_i^t at spatial locations $s_i \in \mathcal{D}$ over the period of a month. We can think of each location s_i having its own time series spanning the monthly period. In the SCOOT setting, each s_i is the location of a detector, and c_i^t is the number of vehicles at location s_i at time between hours $t-1$ and t. We impose a uniform grid on the to the spatial domain $\mathcal{D}$, partitioning the space in to N^2 equally sized rectangles. Additionally, we can think of the entire space on which the data lies as a 3D grid with hourly slices in time. Let $t = 0$ be the current day; the idea is to search the most recent data and detect emerging clusters. Suppose we are interested in determining whether the most recent day of data contained any clusters of count data; the expectation-based scan statistic uses the first 29 days of the month to forecast the time series for the most recent day, hence providing a way to compare actual counts and estimated counts for the most recent day. To avoid the *curse of the partition*, the scan involves searching over many different shapes and sizes of overlapping rectangles in the spatial domain. Although computationally intensive, it provides a good first step into understanding the applicability of such a model.

The above four publications differ due to the domain on which the scan is valid, the type of metric used to score regions and whether it generalises to multi-variate data. To summarise, the overall methodology in each case can be split into three disjoint parts: time series analysis, the spatial scan and estimating the significance of the results. The first of these relies on historical data to estimate baseline values of the vehicle count data. The model then compares these estimated *baseline* values with the actual recorded *count* data with respect to some metric. Our search is then tuned to find regions of space-time that are most likely to represent clustering.

There are four *parameters* of the model which are dependent on the scientific question at the heart of the research; these are

- Which Spatial Region to observe and how to partition the space?
- How far back in time to look for clusters/ how far ahead to forecast?
- Which method of time series analysis to use?
- Which metric to use?

2 Time Series Analysis

The premise of expectation scanning is that there is an expected number of counts as forecast by some Time Series Analysis. This is an essential step, as spatial-temporal clusters will not be recognised unless we have a baseline for what the *should* be. Daniel Neil outlines some methods of TSA in his original paper, here we repeat those methods which are best suited for out problem with adjustments along with some extra methods which we intend to introduce

2.0.1 Hourly adjusted moving average

when considering TSA let $t = 0$ represent the hour at the start of our forecasting period, and $t > 0$ represent the historical counts, our moving average is defined as follows:

$$b_{i,0}^t = \frac{1}{L_b} \sum_{u=t+1 \ldots t+L_b} c_i^u \tag{1}$$

where b_i^t is out baseline forecast at time t at location i. L_b is the number of time intervals we want to look back on to create out moving average, for example if we wanted a daily moving average $L_b = 24$. A standard moving average is obviously a very poorly suited forecast for counts which vary so much depending on their time of day. For that reason we suggest an hourly adjusted moving average, following for the example of D. Neils day of the week adjustment. To do this we consider a local adjustment factor (MALD) β_i^j which is multiplied by the moving average to account for hourly fluctuation. Our hourly MALD factor is calculated as follows:

$$\text{MALD}: \quad \beta_i^j = \frac{\sum_{t=j,j+24,\ldots,j+24\times d} c_i^t}{\sum_{t=1,\ldots,L_b} c_i^t} \tag{2}$$

a simple proportion of counts that occurred at that hour in our look back period, where d is the number of days (or 24 hour periods) in our look back. Thus, our hourly adjusted baseline forecast for time t at detector i is $b_i^t = 24 \times b_{i,0}^t \times \beta_i^j$ this forecasting method accounts for hourly variation well, but misses out on any slower trends such as seasonal trends

2.0.2 Holt-Winters

The multiplicative Holt–Winters' (HW) method is a more dynamic forecasting method that takes into account smaller periodic trends, as well as long term trends (such as seasonal trends). The Holt–Winters' forecasts are determined by from the historical counts c_i^t by iterating the three equations given below. It has three main components: the smoothed value S_t , trend component Tt , and the periodic component I_t

$$\begin{aligned}
&S_t = \alpha \tfrac{c_i^t}{I_{t+24}} + (1-\alpha)\left(S_{t+1} + T_{t+1}\right) \\
&T_t = \beta\left(S_t - S_{t+1}\right) + (1-\beta)T_{t+1} \\
&I_t = \gamma \tfrac{c_i^t}{S_t} + (1-\gamma)I_{t+24}
\end{aligned} \tag{3}$$

α, β, and γ are out hyper parameters, which have to be fine tuned for increased accuracy. The baseline estimate for the next hour is then given by $\underline{t}_i = \left(S_{t+1} + T_{t+1}\right) I_{t+24}$. To forecast the whole period you then just continue the iteration forward of the above equations with the baseline estimates instead of counts. This method comes with the major limitation of a large number of hyper parameters (3 for each detector) which must be fine tuned effectively to provide a reliable forecast

2.0.3 LSTM

Taking forecasting to a more advanced and complex method, we can consider the use of recursive deep learning methods. This provides an extremely accurate forecast, which can be updated straightforwardly with new data. A standard deep learning method for time series prediction is the Long Short-Term Memory Network, with input nodes equal to the number of hours in the lookback period, and output nodes equal to the number of hours in the forecasting period. The LSTM is then trained by iterating through all the historical data. There are two possible approaches

2.1 Missing Values

3 The Scan

3.1 Temporal Subtleties

3.2 The Naive Approach

Given our spatial domain $\mathcal{D} \subset \mathbb{R}^2$, our first step is to partition the grid into N^2 equally sized rectangles; ideally so that at least one detector sits within each grid cell. We plan to search over all rectangles whose maximum length along a given axis does not span more than $N/2$ partitions. Hence, the spatial search alone is $\mathcal{O}(N^4)$. Combining this with a possible W forecasted time steps, the total run-time complexity of this search is $\mathcal{O}(N^4 W)$.

3.3 The Fast Spatial Scan

The ideas is to compute some metric for each space-time region of the search, and then return to the user regions of space-time which have the most significant value of the metric. Here, we use a Poisson likelihood metric.

3.4 The Hypotheses and their Metrics

Fundamentally, for each space-time region S that is searched, we are interested in finding the posterior probabilities

$$\mathbb{P}[H_1(S) \mid D] \tag{4}$$

and

$$\mathbb{P}[H_0 \mid D] \tag{5}$$

For the frequentist approaches, we must come up with metrics that estimate these quantities with certain assumptions. First we define for a space-time region S, $C_S := \sum_S c_i^t$ and $B_S := \sum_S b_i^t$. In our setting, the values of B_S and C_S, represent the number of cars that passed a certain region of space-time, S. For some areas, these can become too large to compute (6); we therefore scale B_S and C_S down by a factor of 1 million.

3.4.1 Poisson Likelihood Metric

As in [Nei09], we begin by setting the null and alternative hypotheses as follows

$$\begin{gathered} H_0 : c_i^t \sim \mathrm{Po}(b_i^t) \ \forall \ s_i^t \in \mathcal{D} \times T \\ H_1(S) : c_i^t \sim \mathrm{Po}(qb_i^t) \ \forall \ s_i^t \in S \text{ and } c_i^t \sim \mathrm{Po}(b_i^t) \ \forall \ s_i^t \notin S \text{ for some } q > 1. \end{gathered} \tag{6}$$

The Poisson likelihood ratio metric $F(S)$ of our hypotheses is defined and simplified as follows:

$$\begin{aligned} F(S) &:= \frac{\mathbb{P}(D|H_1(S))}{\mathbb{P}(D|H_0)} \\ &= \frac{\max_{q>1} \prod_{c_i^t \in S} \mathbb{P}(c_i^t \sim \mathrm{Po}(qb_i^t))}{\prod_{c_i^t \in S} \mathbb{P}(c_i^t \sim \mathrm{Po}(b_i^t))} \\ &= \max_{q>1} \prod_{c_i^t \in S} e^{(1-q)b_i^t} q^{c_i^t} \\ &= \max_{q>1} e^{(1-q)B_S} q^{C_S} \\ &= \left(\frac{C_S}{B_S}\right)^{C_S} e^{B_S - C_S}. \end{aligned} \tag{6}$$

Above we use that the maximum value of $F(S)$ is achieved for $q = \max(1, C_S/B_S)$. In the case of $C_S < B_s$, the value of this metric defaults to 1.

3.4.2 Kulldorf's Original Metric

In the previous metric, the null hypothesis makes the assumption that the counts are generated from a Poisson random variable with mean *equal* to their associated baselines. Kulldorf's original metric differs to this by assuming counts are generated from Poisson random variables with means *proportional* to the estimated baselines. To encode this, we introduce 3 proportionality constants, q_{in}, q_{out} and q_{all}. Mathematically, the hypotheses are as follows:

$$\begin{gathered} H_0 : c_i^t \sim \mathrm{Po}(q_{\mathrm{all}} b_i^t) \ \forall \ s_i^t \in \mathcal{D} \times T \\ H_1(S) : c_i^t \sim \mathrm{Po}(q_{\mathrm{in}} b_i^t) \ \forall \ s_i^t \in S \text{ and } c_i^t \sim \mathrm{Po}(q_{\mathrm{out}} b_i^t) \ \forall \ s_i^t \notin S \end{gathered} \tag{7}$$

Using maximum likelihood estimates for the free parameters q_{in}, q_{out} and q_{all}, the metric in this case is:

$$F_{\mathrm{kul}}(S) = \left(\frac{C_{\mathrm{in}}}{B_{\mathrm{in}}}\right)^{C_{\mathrm{in}}} \left(\frac{C_{\mathrm{out}}}{B_{\mathrm{out}}}\right)^{C_{\mathrm{out}}} \left(\frac{C_{\mathrm{all}}}{B_{\mathrm{all}}}\right)^{-C_{\mathrm{all}}} \tag{8}$$

3.4.3 Generalised Likelihood Metric

In real world applications, we expect some fluctuation in the underlying baseline; thus, we do not want to detect all deviations from baseline, but only those where the amount of deviation is greater than some threshold ε. This motivates the following generalised hypotheses and metric:

$$\begin{array}{ll} H_0(S) : c_i^t \sim \mathrm{Po}(q_{\mathrm{in}} b_i^t) \ \forall \ s_i^t \in S \text{ and } c_i^t \sim \mathrm{Po}(q_{\mathrm{out}} b_i^t) \ \forall \ s_i^t \notin S, & q_{\mathrm{in}} \leq (1+\varepsilon) q_{\mathrm{out}} \\ H_1(S) : c_i^t \sim \mathrm{Po}(q_{\mathrm{in}} b_i^t) \ \forall \ s_i^t \in S \text{ and } c_i^t \sim \mathrm{Po}(q_{\mathrm{out}} b_i^t) \ \forall \ s_i^t \notin S, & q_{\mathrm{in}} > (1+\varepsilon) q_{\mathrm{out}} \end{array} \tag{9}$$

Taking the maximum over the ratio of these likelihoods over all regions S, we arrive at (taking logs for convenience)

$$D_\varepsilon(S) = \log \frac{\sup_{q_{\mathrm{in}} > (1+\varepsilon) q_{\mathrm{out}}} \prod_{s_i \in S} \mathbb{P}[c_i^t \sim \mathrm{Po}(q_{\mathrm{in}} b_i^t)] \prod_{s_i \notin S} \mathbb{P}[c_i^t \sim \mathrm{Po}(q_{\mathrm{out}} b_i^t)]}{\sup_{q_{\mathrm{in}} \leq (1+\varepsilon) q_{\mathrm{out}}} \prod_{s_i \in S} \mathbb{P}[c_i^t \sim \mathrm{Po}(q_{\mathrm{in}} b_i^t)] \prod_{s_i \notin S} \mathbb{P}[c_i^t \sim \mathrm{Po}(q_{\mathrm{out}} b_i^t)]} \tag{10}$$

$$= (\mathrm{sgn}) \left(C_S \log \frac{C_S}{(1+\varepsilon) B_S} + (C_{\mathrm{tot}} - C) \log \frac{C_{\mathrm{tot}} - C_S}{B_{\mathrm{tot}} - B_S} - C_{\mathrm{tot}} \log \frac{C_{\mathrm{tot}}}{B_{\mathrm{tot}} + \varepsilon B_S} \right) \tag{11}$$

where $\mathrm{sgn}(B_S, C_S, B_{\mathrm{tot}}, C_{\mathrm{tot}}) = 1$ if

$$\frac{C_S}{B_S} > (1+\varepsilon) \frac{C_{\mathrm{tot}} - C_S}{B_{\mathrm{tot}} - B_S}$$

and -1 otherwise.

3.4.4 Bayesian Metrics

Within the Bayesian frameworks, instead of assigning scores, we can directly calculate (assuming priors) the two posterior probabilities (4) and (5). [NC10] introduces a Gamma-Poisson model to estimate q_{in} and q_{out} from historical data.

4 Significance Testing

Now that we have calculated metrics for each searched space-time region S, we need a way to report on regions which have a significantly large score. [Nei09] explain how there are two main methods to find the value of F_{thresh} such that we report on all regions with $F(S) > F_{\mathrm{thresh}}$. In the Bayesian framework, this step is not required since the output is probabilities.

4.1 Randomisation Testing

A naive approach is to first consider all 'actual' counts c_i^t as a Poisson random variable with mean b_i^t. By doing this for all events in $\mathcal{D} \times T$, we can re-run the spatial-scan and find the region S^* such that

$$S^* = \arg\max_{S \in \mathbf{S}} F(S) \tag{12}$$

By simulating this M times, we can build up a distribution of of $F(S^*)$ scores and compare our realised scores with this distribution. This approach is computationally intensive ($\mathcal{O}(N^4 W M)$) and prone to high false-positive rates as explained in [Nei09].

4.2 Historical Data Testing

Perhaps a better approach is to exploit historical data, given that it is available. Suppose we are interested in the $W = 24$ case. Neill explains that we can take a year's worth of historical data, and compute the maximum $F(S)$ score for each of the 365 days. Although initially intensive, each day's maximum $F(S)$ can be stored in memory; the current day's score can then be compared against this empirical distribution in relatively little time.

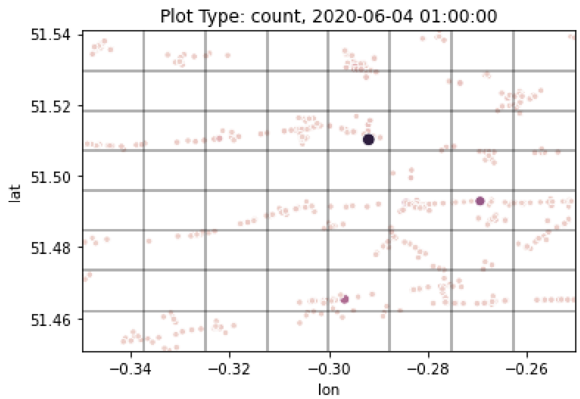

Figure 1: Detector locations in the Ealing area of London. Darker/larger dots signify higher counts of vehicles.

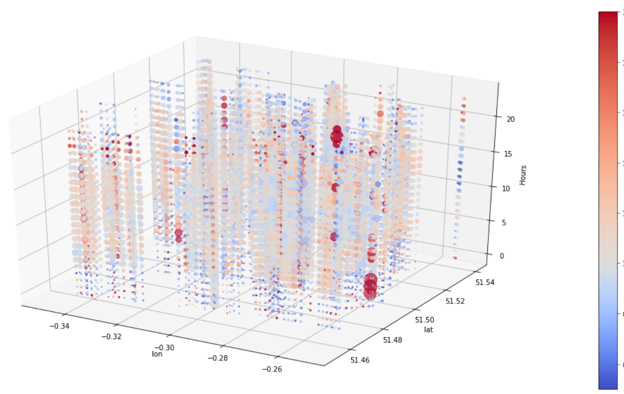

Figure 2: The global space-time region to be searched. The flat plane represents the locations of the detectors in space (lon/lat), and time is displayed upwards. Colour/size represents the ratio of the actual count to the predicted baseline value; i.e. c_i^t/b_i^t.

5 Ealing Example

5.1 Setup

Here, we look at an explicit example using data from SCOOT sensors in the area of Ealing between the 4th June 2020 and 5th June 2020. In longitude and latitude co-ordinates, we define

$$\mathcal{D} = (-0.3497, -0.2504) \times (51.4508, 51.5409)$$

and we set $W = 24$; i.e. looking back at the last 24 hours of data for space-time clusters. For a first example, we set $N = 8$ yielding 64 grid cells on the Ealing area. An example plot is shown in Figure 1. Here, $\mathcal{D}$ contains 596 detectors and 14238/14304 valid readings.

5.2 Time Series Analysis

For this particular example, we train a LTSM model to forecast baseline values for the most recent 25 hours. Figure 2 shows how the actual counts and baseline estimates compare over the 24 hours of interest. This procedure takes roughly 30 minutes for this amount of detectors.

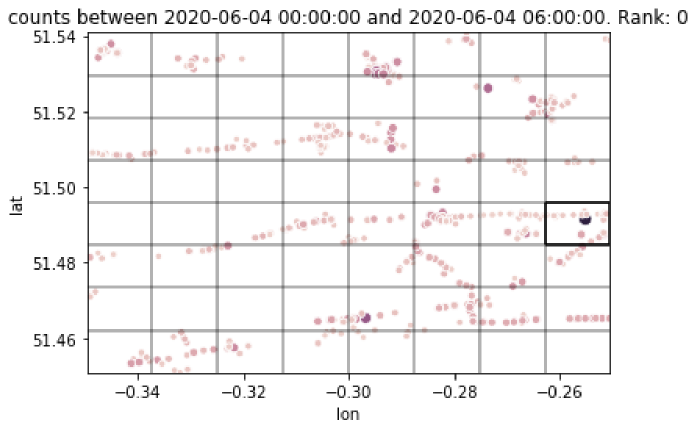

Figure 3: The best scoring region is highlighted in bold. A shown from the title, this occurs during the 12am and 4am on 4th June 2020. With such a large count value, it maybe hat we're looking at an anomoly instead.

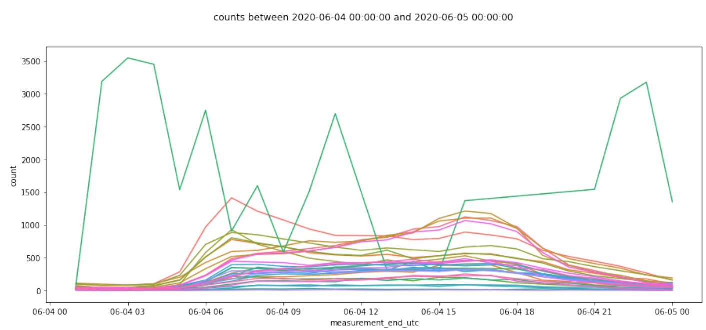

Figure 4: Time series data for the detectors in the highest scoring region highlighted in Figure 3 over the whole prediction period ($W = 24$ hours). Clearly, the scan is flagging an anomalous count from the detector.)

5.3 The Scan

An 8x8 grid over 24 hours yields 16224 space-time regions to scan over and compute $F(S)$. This takes roughly 45 seconds on modern laptop in serial. Figure 3 shows the location and time of the highest scoring region $F(S)$. Figure 4 shows the behaviour of these detectors in this spatial region over the time window of interest. It is indeed clear that the scan is flagging an anomalous count from the detector; this is an area of improvement which is currently being updated in the time series analysis stage of the method.

5.4 Significance Testing

As this part of the process is computationally intensive, we present example timings for only 5 simulations.

```
Found a grid partition = 8
Searching over the region spanning 24 hours

====================
Beginning Simulation
====================
Performing simulation 5 of 5.
Time Elapsed: 398.434152576 seconds
```

Using 100 simulations results in a run-time of approximately 2.5 hours for this particular region. For comparison, a larger area of London ($\mathcal{D} = (-0.3497, -0.1500)x(51.35067, 51.5499)$) containing 2665 detectors takes roughly 60 seconds to compute the $F(S)$ scores in a scan. Simulating these counts gives the following output:

```
Found a grid partition = 8
Searching over the region spanning 24 hours

====================
Beginning Simulation
====================
Performing simulation 5 of 5.
Time Elapsed: 867.489024811 seconds
```

Extrapolating to 100 simulations gives an estimated run-time of 5 hours. In addition, the time series forecasting for this region takes roughly 4 hours in serial.

References

[NC10] Daniel B. Neill and Gregory F. Cooper. A multivariate bayesian scan statistic for early event detection and characterization. *Mach. Learn.*, 79(3):261–282, June 2010.

[Nei09] Daniel Neill. Expectation-based scan statistics for monitoring spatial time series data. *International Journal of Forecasting*, 25:498–517, 07 2009.

[NMC06] Daniel B Neill, Andrew W Moore, and Gregory F Cooper. A bayesian spatial scan statistic. In *Advances in neural information processing systems*, pages 1003–1010, 2006.

[NMPM05] Daniel B Neill, Andrew W Moore, Francisco Pereira, and Tom M Mitchell. Detecting significant multidimensional spatial clusters. In *Advances in Neural Information Processing Systems*, pages 969–976, 2005.