

Bursts in Streams of Geodata

1 Introduction

We seek to analyze time stamped geo-data, with the goal of describing the change in both the rate events occur and their distribution in space. We suppose the data is generated randomly by an automata with multiple states. The arrival times are assumed non-stationary Poisson Process, where the rate is determined by the state of the automata. Further, the location of each data point is given by some distribution parameterized by the state. We assume that the state changes over time according to some probability distribution. Using a dynamic programming algorithm, we employ a hidden Markov model to find the most likely state sequence given our observed data. The state sequences generated allows us to see a burst's intensity grow and shrink, or it's geographic radius expand and contract over time.

We begin supposing we know where the burst is located and show how to find the sequence of states the automata passes through, giving burst intensity and radius. We then build up increasingly powerful, but computationally expensive tools to find the burst center in a region, identify overlapping bursts, allow for moving burst centers, and ultimately to take raw data and without any screening and determine the number of, size of, and location of the bursts that are present. As these tools are increasingly expensive, they must be performed with decreasing precision. However, we can first run the coarse powerful algorithms to get an idea of about where the bursts are located and how large in diameter they are, then use this information to fulfill the assumptions of one the finer algorithms.

Prospective applications of this project include geo-tagged photos from Yahoo's photo sharing application Flickr, and cell phone call data.

2 A Simple Model

This initial model is for the case when there is a single burst centered about some known point \tilde{x} , where the data is assumed to be normally distributed about the burst center. The state in this model has two components, the variance of the normal distribution, and the arrival rate of the data, and state transitions occur independently for these two parameters. We seek the most likely state sequence given our observed data.

Formally, this model takes data points $(x_i, t_i) \in \mathbb{R}^2 \times \mathbb{R}$ of events occurring, where x_i gives the location and t_i gives the time, a center \tilde{x} that is assumed to be the only center of any burst occurring in the region throughout the time period, and radius R such that $R > \max_{i \in \{1, \dots, n\}} \{\|x_i - \tilde{x}\|\}$. The data points are supposed to be generated by a non-stationary Poisson process, with rate $\lambda(t)$. Further, we assume that the distribution of the location for each data point, while a function of t , was either normal¹ with mean \tilde{x} and variance $\sigma(t)^2$ or uniform over the region². Finally, we assume that for each data point, the location of the next data point and the inter-arrival time between the data points are independent. Thus, the joint density for the location and time of the next photo is

$$\begin{aligned} f_{X,T}(x,t) &= \frac{1}{Z} \cdot \exp\left(-\frac{\|x - \tilde{x}\|^2}{2\sigma(t)^2}\right) \cdot \lambda(t) \cdot \exp(-\lambda(t) \cdot t) \cdot \mathbb{1}_{\{\|x - \tilde{x}\| \leq R\}} \\ &= \frac{\lambda(t)}{Z} \cdot \exp\left(-\frac{\|x - \tilde{x}\|^2}{2\sigma(t)^2} - \lambda(t) \cdot t\right) \cdot \mathbb{1}_{\{\|x - \tilde{x}\| \leq R\}} \end{aligned}$$

¹We must normalize the distribution's density by $1/P[\|X - \tilde{x}\| > R]$, as we are not considering data that falls outside of this region

²Technically, if $X_\sigma = N(\mu, \sigma^2)$, then $X_\sigma/P[X_\sigma - \mu > R] \rightarrow U(\mu, R)$ uniformly on the compact space $\{x: |x - \mu| \leq R\}$ as $\sigma \rightarrow \infty$, making the uniform distribution over space a limiting case of the normal.

where Z is a normalizing constant, and $t = t_{i+1} - t_i$ is the interarrival time. For simplicity of notation we will suppress the indicator function in the future. Supposing $\bar{X} \sim N(\tilde{x}, \sigma(t))$, we have

$$Z = 2\pi\sigma(t)^2 P[\|\bar{X} - \tilde{x}\| > R] = \int_0^{2\pi} \int_R^\infty \exp\left(-\frac{r^2}{2\sigma(t)^2}\right) r dr d\theta = 2\pi\sigma(t)^2 \cdot \left(1 - \exp\left(-\frac{R^2}{2\sigma(t)^2}\right)\right).$$

We define S_1 to be the set of values that $\lambda(t)$ can take on, and S_2 to be the set of values $\sigma(t)$ can take. In our implementation, they were constructed in the following manner. Let $\lambda^* = (n+1)/(t_n - t_0)$ be the average rate photos are taken in the time period. Then S_1 was a finite truncation of $\{\infty, R^2, (R/2)^2, (R/4)^2, \dots\}$. Similarly, S_2 was a finite truncation of the set $\{\dots, \lambda^*\alpha^{-1}, \lambda^*\alpha^0, \lambda^*\alpha^1, \dots\}$ for some $\alpha > 1$.

We now assume our data were generated by an automata with state space $S_1 \times S_2$. After each photo, it would transition between states according to some distribution function $p_{(\bar{s}_1, \bar{s}_2)}: S_1 \times S_2 \rightarrow \mathbb{R}$, defined uniquely for each state.

To construct the transition function we used, for each dimension of the state space i , if $|S_i| = n_i$, we defined $\rho_{\bar{s}_i}: S_i \rightarrow \mathbb{R}$ by

$$\rho_{\bar{s}_i}(s_j) = \begin{cases} \beta^{|i-j|} & i \neq j, i \in \{1, n_i\} \\ \beta^{|i-j|}/2 & i \neq j, i \notin \{1, n_i\} \\ 1 - \sum_{k \neq i} \rho_{\bar{s}_i}(s_k) = 1 - \beta + \mathcal{O}(\beta^2) & i = j \end{cases} \quad (1)$$

for some $\beta < 1/2$. Using these functions, we constructed our transition function $p_{\bar{s}_i, \bar{s}_k}(s_i, s_j) = \rho_{\bar{s}_i}(s_i) \cdot \rho_{\bar{s}_j}(s_j)$.

Now the state forms a Markov Chain, and we need to determine the most likely state sequence given the data we have seen. This calculation is performed just as in Professor Kleinberg's paper. The prior probability of a given state sequence $\mathfrak{s} = s_0, s_1, \dots, s_n$ where $s_i \in S_1 \times S_2$ is $\prod_{i=1}^n p_{s_{i-1}}(s_i)$. Letting $\mathbf{u} = u_1, u_2, \dots, u_n$ where $u_i = t_i - t_{i-1}$ be the inter-arrival times, we have that $P[\mathbf{u}|\mathfrak{s}] = \prod_{i=1}^n f_{s_i}(x_i, u_i)$. Thus by Bayes' rule, we have the probability of a state sequence given our points is

$$\begin{aligned} P[\mathfrak{s}|\mathbf{u}] &= \frac{P[\mathfrak{s}]P[\mathbf{u}|\mathfrak{s}]}{\sum_{\mathfrak{s}' \in S_1 \times S_2} P[\mathfrak{s}']P[\mathbf{u}|\mathfrak{s}']} \\ &= \frac{1}{\sum_{\mathfrak{s}' \in S_1 \times S_2} P[\mathfrak{s}']P[\mathbf{u}|\mathfrak{s}']} \cdot \prod_{i=1}^n p_{s_{i-1}}(s_i) \cdot \prod_{i=1}^n f_{s_i}(x_i, u_i) \end{aligned}$$

To maximize this probability over all state sequences, we need only minimize the cost function

$$c(\mathfrak{s}|\mathbf{u}) = \sum_{i=1}^n -\ln(p_{s_{i-1}}(s_i)) + \sum_{i=1}^n -\ln(f_{s_i}(x_i, u_i))$$

as \ln preserves order and the missing term does change over state sequences.

Now we can use the same dynamic programming algorithm, with an extra dimension for our state space, to compute the most likely sequence of states. Suppose the initial state is $s_0 = (\infty, \lambda^*)$. In this state the data is distributed uniformly across the region and occurs at the average rate for the time period. Let $C(s, i)$ be the minimum cost sequence up to time i ending in state s . As we know the initial state, $C(s_0, 0) = 0$ and for all $s \neq s_0$, $C(s, 0) = \infty$. Then for all other i ,

$$C(s, i) = -\ln f_s(u_i) + \min_{\{s' \in S_1 \times S_2\}} \{C(s', i-1) - \ln(p_{s'}(s))\}.$$

Note that this algorithm differs from Professor Kleinberg's algorithm in that there is an extra dimension in the state space, and thus the dynamic programming table.

3 Eliminating the Assumption that the Center is Known

In Section 2, we assumed that we knew where the burst center was located. However, suppose we know there should be a burst in a region, but are not sure exactly where it is centered. Further, we don't know how large (in radius) the burst is, only that there is no burst in the region.

To solve this problem, we can run the Section 2 algorithm on a grid of centers, and choose the center that can produce the lowest cost. We may want to run the algorithm with a more limited state space, maybe only two or three possible rates and radii, using parameters that make transitions large and unlikely, as we want to run the algorithm on a fine grid to find the center. Once we know the burst center, we can then run the algorithm from Section 2 again with a larger state space for the radius and rate to get a more precise measure of the burst.

4 Allowing for More Than One Center

If by glancing at the data, it is clear that there are two or more distinct bursts, but they are close enough together that there is some region where both bursts are contributing to the rate at which data occurs, we cannot just separate the region into parts and run our algorithm on each part.

Suppose at first that we can clearly identify where the centers are. Then instead, we should have an automata with a rate parameter and a dispersion parameter in its state for each center. If we let $f_{X,T}^i$ denote the density function for the location and time of the next data point generated by the burst centered about burst center i , then for n centers, the joint density of data inter-arrival time in space could be model by

$$\frac{1}{n} \sum_{i=1}^n f_{X,T}^i(x, t) \tag{2}$$

which integrates to 1 as each $f_{X,T}^i$ integrates to one.

As a special case of this technique, it may frequently be useful to always have a burst centered in the middle of the region that is fixed in uniformly distributing data over the space, so it can reflect changes in the overall rate of photos being taken, such as for time of day, if you wanted to remove that kind of cyclic noise from the other bursts.

Interestingly, notice that this technique could allow for multiple bursts with the same center.

If we do not know the precise locations of the centers, but we know there is some small number, we can create a grid of possible centers as in section 3, and run the algorithm from this section on all combinations of possible centers.

5 Moving Burst Centers

For simplicity, we will only consider a single burst, but this will generalize to any number of bursts as in section 4.

Now the state space will be the Cartesian product of a grid of possible burst centers, a set of points denoted C , some set of possible rates, and some set of possible dispersions. Then we can construct a probability distribution for moving from state to state by taking our fixed center density function and multiplying it by some function $\rho_c: C \rightarrow \mathbb{R}$ for each $c \in C$ with the property that

$$\sum_{c' \in C} p_c(c') = 1 \qquad \forall c \in C.$$

We may want to assume that the burst center moves continuously, for example if we are looking at a hurricane region, and we suspect the hurricane is generating a burst of photo activity. In this case, we might want ρ to be decreasing rapidly with $\|c' - c\|$, and construct a function similar to equation 1. Alternatively, we may have no reason to believe that successive burst centers should be near one another. Our data could be all photos tagged with the word “Yankees,” with a region of the United States, and then we might expect the burst center to move as the team moves from stadium to stadium. In this case, taking

$$\rho_c(c') = \begin{cases} \gamma & c = c' \\ \frac{1-\gamma}{|C|-1} & c \neq c' \end{cases}$$

for some $\gamma \in (0, 1)$ such that $\gamma > (1 - \gamma)/(|C| - 1)$ would be more appropriate.

In the first case, where we expect the motion of the burst centers to be continuous, it may be more convenient for computational reasons to only allow for the burst center to move a small, fixed distance each time it moves (have probability zero of moving further), as then in the dynamic programming you can greatly reduce the number of states you must iterate through.

6 When We Know Nothing At All

In the previous sections, we have always assumed we had some idea about the number of bursts there should be. However, this would require looking at the data, and if there are many burst centers it may be difficult to identify them even after plotting the data. In this section we will give a method to identify the burst centers in a region when we know neither how many there are or how large they are in diameter. In this section, we do not allow the burst centers to move, but there is no reason why we could not do this as well (at greater computational expense). Further, we could also allow for multiple burst centers at a given location, but for simplicity we will not consider this either.

The automata will have a single burst that uniformly distributes the data points in space, and a variable rate parameter, similar to the set of rates for the automata in Section 2. The automata will seek to use as few bursts as possible to explain the observed data, by assuming some distribution of the number of states that decreases quickly as the number of states increases, such as $p_Q(n) = \delta^n/C$, where $\delta > 1$ and C is a normalizing constant, and we arbitrarily impose some maximum number of centers m . Here n indicates the number of states in addition to the previously described base state, and p_Q is non-zero for $n = 0, \dots, m$. Increasing δ will clearly make a solution using a fewer number of states seem more likely.

Now we will run our algorithm from section 4 for each number of centers, for $n = 0, \dots, m$, but multiplying the probability of the optimal state sequence by $p_Q(n)$, and then take the most likely state sequence.

Clearly, this will be quite computationally intensive, so it will probably be best to use a more coarse grid, fewer radii, and fewer rates. Also we can use shapes other than Gaussians, as something simpler like a circle or square with a radius and a rate would be computationally much cheaper.

7 Data Issues

Maybe these things should just be talked about in each application, or maybe this section should be an appendix. Anyway we need to talk about all the issues with data precision, in both time and location, for photos using upload time/place when other data is missing is appropriate, if ever, and how we only allow one photo per user per day (or whatever time period we deem appropriate).