

A clustering algorithm to organize satellite hotspots data for the purpose of tracking bushfires remotely

by Weihao Li, Emily Dodwell, and Dianne Cook

Abstract An abstract of less than 150 words.

Introduction

Bushfires are a major problem for Australia, and many other parts of the globe. There is concern that as the climate becomes hotter, and drier, that the impact of fires becomes much more severe and extensive. In Australia, the 2019-2020 fires were the worst on record causing extensive ecological damage, as well as damage to agricultural resources, properties and infrastructure. The Wollemi pine, rare prehistoric trees, required special forces intervention to prevent the last stands in the world, in remote wilderness areas, from being turned into ash.

Contributing to the problem is that many fires started in very remote areas, locations deep into the temperate forests ignited by lightning, that are virtually impossible to access or to monitor. Satellite data provides a possible solution to this, particularly remotely sensed hotspot data, which may be useful in detecting new ignitions and movements of fires. Understanding fires in remote areas using satellite data may provide some help in developing effective strategies for mitigating bushfire impact.

This work addresses this topic. Using hotspot data, can we cluster in space and time, in order to determine (1) points of ignition and (2) track the movement of bushfires.

This paper is organised as follows. The next section provides an introduction to the literature on spatiotemporal clustering and bushfire modeling and dynamics. Section [Algorithm](#) describes the clustering algorithm, and section [Application](#) illustrates how the resulting data can be used to study bushfire ignition.

Background

literature review

Algorithm

Data source

This algorithm is initially developed in the research of 2019-2020 bushfires in Victoria, Australia. Therefore, the illustration of this algorithm will use hotspot data during 2019-2020 Australian bushfire season taken from Himawari-8 satellite ([P-Tree System, 2020](#)). This satellite hotspot dataset contains records of 1989572 hotspots for 6 months in the full disk of 140 °east longitude.

The data pre-processing procedure includes selecting hotspots within the boundary of Victoria and filtering hotspots with a threshold (irradiance over 100 watts per square metre) suggested by [Williamson \(2020\)](#) to reduce noise from the background.

The final dataset contains 75936 hotspots with ID, longitude, latitude and observed date as fields. The map of this dataset is shown in [Figure 1](#).

Steps

The spatiotemporal clustering algorithm is consist of 4 steps, (1) divide hotspots into intervals, (2) cluster hotspots spatially, (3) update the memberships and (4) compute ignition locations. They will be described in details in the rest of the section.

1. Divide hotspots into intervals

Despite hotspot data can be represented in the three-dimensional Euclidean space and clustered using ordinary algorithms, like K-means, the clustering results could be highly sensitive to the scaling of the temporal dimension ([Kisilevich et al., 2009](#)). Besides, one of the characteristics of the hotspot data is cloud cover could lead to missing observations of a bushfire in several hours. It suggests that

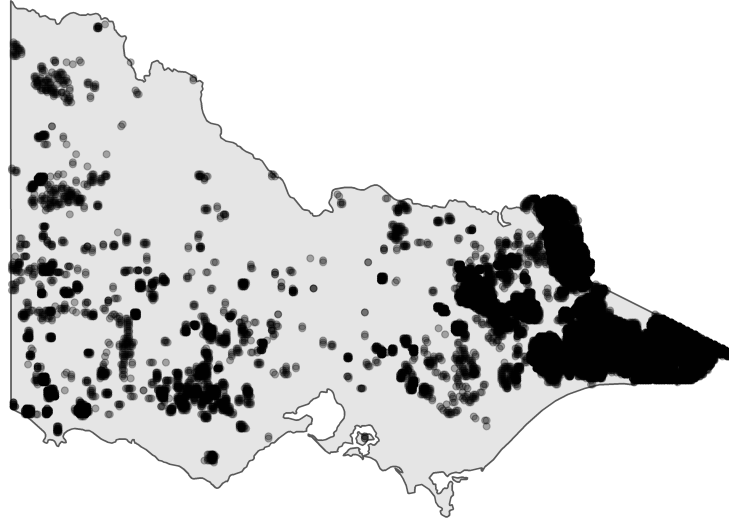


Figure 1: A map shows the distribution of hotspots in Victoria during 2019-2020 Australia bushfire season.

hotspots with long intervals may belong to the same bushfire. One possible solution to this issue is dividing hotspot data into intervals. In other words, the temporal dependence between hotspots is predetermined by a parameter *ActiveTime*. The interpretation of *ActiveTime* is the time a fire can stay smouldering but undetectable by satellite before flaring up again.

Given a certain value of *ActiveTime* and the length of the time frame T , the algorithm will define several intervals,

$$S_t = [\max(1, t - \text{ActiveTime}), t], \quad t = 1, 2, \dots, T$$

,where T and t have the same unit as *ActiveTime*.

For example, if the dataset contains 48 hours of hotspot data and the *ActiveTime* = 24 hours, there will be 48 intervals, S_1, S_2, \dots, S_{48} , where

$$\begin{aligned} S_1 &= [1, 1] \\ S_2 &= [1, 2] \\ &\dots \\ S_{25} &= [1, 25] \\ S_{26} &= [2, 26] \\ &\dots \\ S_{47} &= [23, 47] \\ S_{48} &= [24, 48] \end{aligned}$$

2. Cluster hotspots spatially

The previous step breaks the temporal dimension. Hence, the following step only needs to address the hotspots spatially by introducing another parameter *AdjDist*. *AdjDist* represents the potential distance a fire can spread with respect to the temporal resolution of the data. For example, let *AdjDist* = 3000m and the temporal resolution of the data is 10-minute, then the potential speed of the bushfire is $3000m/10 \text{ min} = 18km/h$.

Given a fixed value of *AdjDist* and the interval S_t , the algorithm will:

- Append a randomly selected hotspot h_i to a empty list L , where h_i is the i th hotspot in the interval S_t , and let pointer P points to the first element of the list L .
- Visit every h_i where $h_i \notin L$. If $\text{geodesic}(h_i, P) \leq \text{AdjDist}$, append h_i to list L .
- Move pointer P to the next item of the list L .
- Repeat (b) and (c) till the pointer P reaches to the end of the list L .

- (e) For all hotspots $h_i \in L$, assign a new membership to them. Pop these hotspots from the interval S_t . Repeat (a) to (e) if interval S_t is not empty.
- (f) Recover the interval S_t and record the memberships.

Diagram 2 shows an example of this step.

3. Update the memberships

With clustering results for each interval, the next step is to update the memberships by bringing in information from earlier intervals.

This step starts from $t = 2$ till $t = T$. Given the interval S_t , the algorithm will,

- (a) Let h_i succeeds its membership from S_{t-1} , if h_i belongs to S_{t-1} , where h_i is the i th hotspot in the interval S_t . These hotspots are collected by a set $H_s = \{h_s^1, h_s^2, \dots\}$.
- (b) Set $H_c = \{h_c^1, h_c^2, \dots\}$, where h_c^i is the i th hotspot in set H_c . h_c^i belongs to S_t but does not belong to S_{t-1} . If h_c^i being clustered into the same component with h_s^j in interval S_t , h_c^i succeeds the membership from the nearest h_s^j , where h_s^j is the j th hotspot in set H_s .

4. Compute ignition locations

The previous step assigns all hotspots with updated memberships. Hence, the final step is to compute the ignition location for each cluster. If there are multiple earliest hotspots belong to the same cluster, the centroid of these hotspots is used as the ignition location. Otherwise, the earliest hotspot is used as the ignition location.

Effects of parameter choices

There are two parameters that can be tuned in this algorithm. They are `adj_dist`, which is the density distance and `active_time`, which is the .

Application

Determining the ignition point and time for individual fires

Show ignition points for a particularly heavy day and another for a particularly light day

Tracking fire movement

Display showing how a fire moves over time, maybe two or more fires

Allocating resources for future fire prevention

Merging data with camp sites, CFA, roads, ...

Summary

Acknowledgements

- The code and files to reproduce this work are at XXX
- Data on hotspots can be downloaded from XXX

Bibliography

- S. Kisilevich, F. Mansmann, M. Nanni, and S. Rinzivillo. Spatio-temporal clustering. In *Data mining and knowledge discovery handbook*, pages 855–874. Springer, 2009. [p1]
- P-Tree System. JAXA Himawari Monitor - User's Guide, 2020. URL <https://www.eorc.jaxa.jp/pmtree/userguide.html>. [p1]
- G. Williamson. Example code to generate animation frames of Himawari-8 hotspots, 2020. URL <https://gist.github.com/ozjimbo/80254988922140fec4c06e3a43d069a6>. [p1]

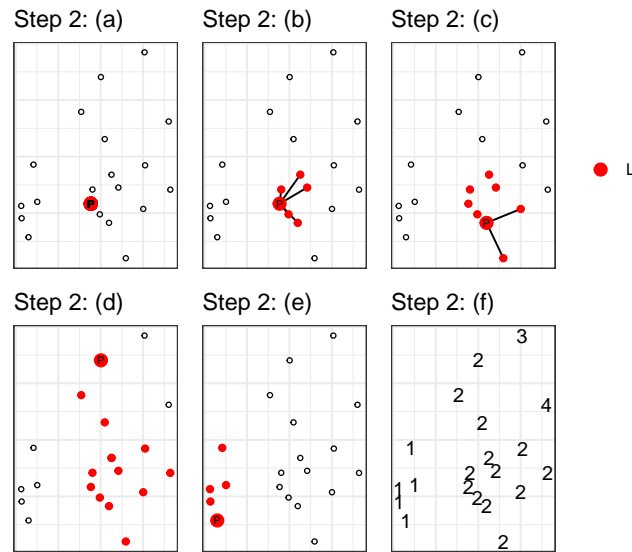


Figure 2: An example of step 2 given 20 hotspots in interval S_t . (a) A hotspot is selected randomly as the first item of list L and the pointer P . Hotspots in list L are in red. Pointer P is drawn with larger marker size. (b) Nearby hotspots of the pointer P are appended to the list L . (c) Move pointer P to the next item of list L and append the nearby hotspots to list L . (d) The first cluster is identified via repeating substep (c). (e) Clear the list L , then randomly select an unassigned hotspot to identify another cluster. (f) The final clustering result is produced via repeating substep (d). The labels show the cluster each hotspot belongs to.

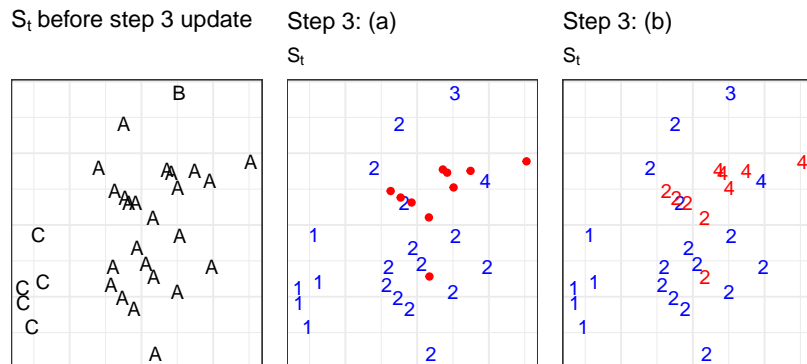
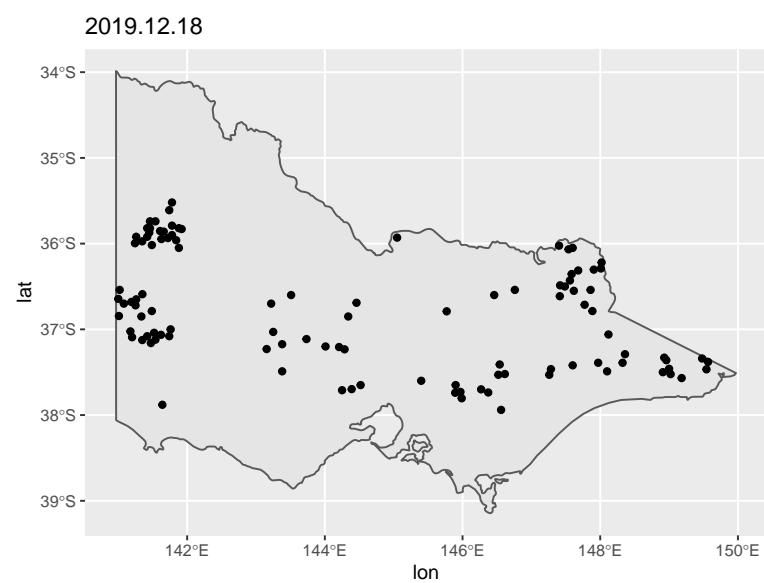
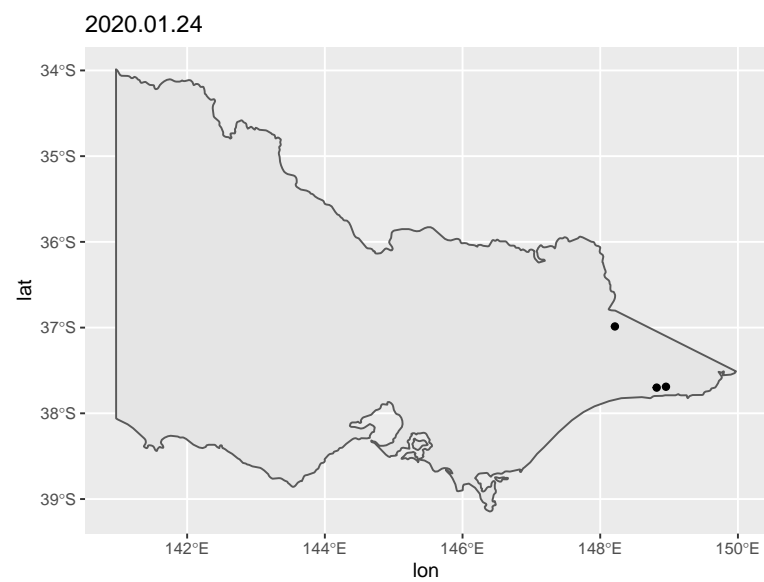
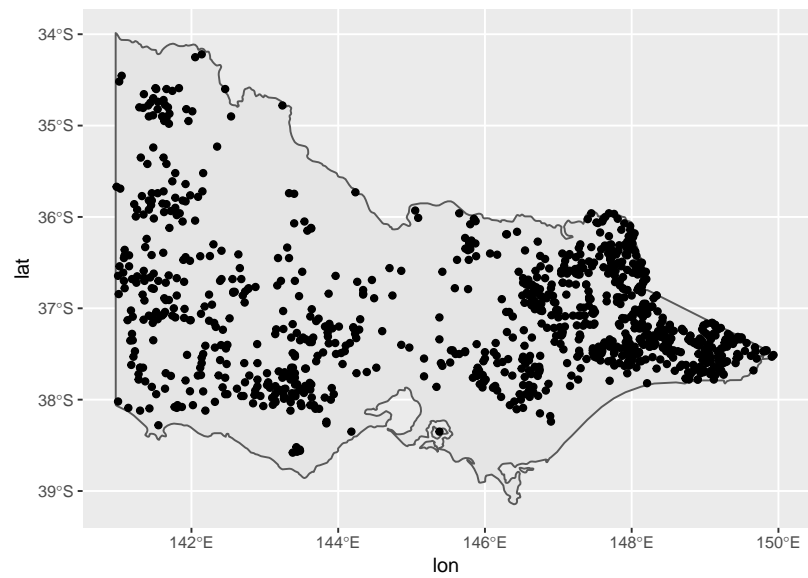


Figure 3: An example of step 3. In this example, there are 30 hotspots in interval S_t (a) 20 out of 30 hotspots belong to both interval S_t and interval S_{t-1} . These hotspots succeed their memberships from S_{t-1} . They are annotated in blue with membership labels. Points in red are the rest 10 hotspots only belong to interval S_t . (b) For each red point, succeeds the nearest blue label that shares the same component (according to the left plot) with that red point in interval S_t .



Weihao Li
Monash University
line 1
line 2

wlii0039@student.monash.edu

Emily Dodwell
AT&T
line 1
line 2

emdodwell@gmail.com

Dianne Cook
Monash University
line 1
line 2

dicook@monash.edu