WildfireDB: A Spatio-Temporal Dataset Combining Wildfire Occurrence with Relevant Covariates

Samriddhi Singla
Computer Science and Engineering
University of California, Riverside
ssing068@ucr.edu

Tina Diao
Stanford University
Stanford, CA 94305
tdiao@stanford.edu

Ayan Mukhopadhyay
Stanford University
Stanford, CA 94305
ayanmukh@stanford.edu

Ahmed Eldawy
Computer Science and Engineering
University of California, Riverside
eldawy@ucr.edu

Ross Shachter
Stanford University
Stanford, CA 94305
shachter@stanford.edu

Mykel Kochenderfer
Stanford University
Stanford, CA 94305
mykel@stanford.edu

Abstract

Modeling fire spread is critical in fire risk management. Creating data-driven models to forecast spread remains challenging due to the lack of comprehensive data sources that relate fires with relevant covariates. We present the first comprehensive dataset that relates historical fire data with relevant covariates extracted from satellite imagery. This open-source dataset contains over 2 million data points. We discuss an algorithmic approach based on large-scale raster and vector analysis that can be used to create similar dataset.

1 Introduction

Wildfires cause loss of life, economic damage, and pose indirect environmental and health threats [Doerr and Santín, 2016]. The November 2018 Camp Fire in Northern California resulted in losses worth $24 billion, including property destruction and firefighting costs [Bartz, 2019]. Occurrences of such extreme fire events are likely to increase [Joseph et al., 2019]. In the current wildfire season in California so far, more than four million acres have already burned due to more than 8,000 wildfires. At one point in August 2020, the entire northern half of the state had been instructed to prepare for evacuation.

Modeling the dynamics of fire spread is crucial to first responders. Responders need to allocate limited resources across large areas to combat fires and minimize loss of life and property. Traditionally, fire spread is modeled by tools that use *physics-based* modeling [Rothermel, 1972, Andrews, 1986, Finney, 1998]. While such models are widely used, prediction of fire spread is improved by a large set of covariates. It is difficult to model the exact effect of each covariate on fire in closed-form. Data-driven modeling can be used to estimate the effects of a diverse set of features on wildfire susceptibility (such as geographic and climate data) [Joseph et al., 2019, Ghorbanzadeh et al., 2019] and improve response to emergency incidents in general [Mukhopadhyay et al., 2020]. However, to the best of our knowledge, there is no complete and open-source data source that combines fire occurrences with geo-spatial features, fuel levels, and weather to allow the research community to develop approaches to manage wildfires.

Through this paper, we make available a spatio-temporal dataset, *WildfireDB*, that can be used to model how wildfires spread as a function of relevant covariates. We discretize space and time and integrate fire occurrence with corresponding vegetation, fuel, and topographic information. We use "cell" and "time-step" to denote the smallest units of spatial and temporal discretization, respectively. Each data point in our data source consists of information about a specific spatial cell (called reference

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

cell) on fire at a given time-step. Each data point also consists of information about neighbors of the reference cell at the subsequent time-step and whether fire spread from the reference cell to the neighboring cell or not. We define neighbors as spatial units bordering each other. Our data source can then be used to predict how fire will spread from an area to adjacent areas as a function of relevant covariates.

Generating a comprehensive dataset on fire spread dataset is complicated for the following two reasons. First, data regarding fire occurrence and covariates is often available in different data models. For example, the locations and sizes of historical fire occurrences are usually available in a vector model, while information about vegetation, fuel, and topographic features is available in a raster model. These two data models use different storage mechanisms and computational methods that makes it difficult to combine them. Second, fires spread through extremely large areas through which covariates can vary significantly. As an example, the raster data used in our data source has over a billion spatial units for the state of California alone. Mining large-scale feature data is a massive computational bottleneck. The large size of the data sources further complicates the fusion of raster and vector data.

Traditional approaches to geospatial data fusion are designed to work with either raster or vector data. Therefore, in order to combine data sources in different data models, they need to be converted to a uniform representation. This conversion is computationally expensive and increases the size of data quadratically with the spatial resolution since the data is two-dimensional. The raster-based approach [Baumann et al., 1998, Brown et al., 2013] rasterizes each polygon in the vector layer to a raster (mask) layer with the same resolution as the input raster layer. It then combines the two raster layers to compute the desired aggregate function. Systems that use this approach generally keep the mask layer in memory, making it difficult to use them when the mask layer becomes too large. This approach has a computational complexity of $O(n_p \cdot c \cdot r)$, where n_p is the number of polygons in the vector data, and c and r are the number of columns and rows in the raster data respectively. On the other hand, the vector-based approach [Zhang et al., 2015] converts each pixel in the raster to a point and then tests the point against each polygon in the vector data to find a match. This approach has a computational complexity of $O(n_p \log n_p \cdot c \cdot r)$.

The limitations of these systems in processing the combination of raster and vector data becomes more prominent when we need to process large amounts of data. Hence, we use a fully decentralized approach to data fusion to combine raster and vector data [Singla and Eldawy, 2018]. This approach does not require data to be converted from one form to another (vector or raster). Instead, it computes an intermediate data structure, called an *intersection file* between the raster and vector data. The *intersection file* serves as a mapping between the raster and vector dataset and can also be leveraged to allow parallel computation. This approach, with a computational complexity of $O(n_p \log n_p + c \cdot r)$, is scalable and efficient for large raster and vector datasets. This approach allows us to combine large raster and vector datasets and further process it to generate the *WildfireDB* dataset.

2 Data

WildfireDB contains the locations and sizes of historical fire occurrences in California, through the years 2012 to 2018. Each entry in the dataset consists of a specific cell that is observed to be on fire at a particular time-step along with spatially-associated vegetation descriptors, fuel levels, and topography information. Each entry also consists of fire occurrence and the same set of features in neighboring cells at the subsequent time-step.

The fire occurrence data was collected in vector form from the Visible Infrared Imaging Radiometer Suite (VIIRS) thermal anomalies/active fire database [Schroeder et al., 2014]. The data contains latitude and longitude values that correspond to the center of pixels representing 375 $\times$ 375 meter square cells. An incidence of fire is indicated by the fire radiative power (FRP) levels in the VIIRS dataset. The temporal granularity of the data is one day.

The vegetation, fuel, and topography data was collected in raster form from the “Landfire” project [Ryan and Opperman, 2013], which is based on satellite imagery. The raster files have a spatial resolution of 30 $\times$ 30 meter square cells and each file consists of over 1 billion pixels. This includes data categories like canopy base density, canopy cover, and vegetation type. We list all the data categories used and the years from which the data was collected in Table 1.

Table 1: LANDFIRE raster data categories

Name	Year(s)
Canopy Base Density	2012, 2014, 2016
Canopy Base Height	2012, 2014, 2016
Canopy Cover	2012, 2014, 2016
Canopy Height	2012, 2014, 2016
Existing Vegetation Cover	2012, 2014, 2016
Existing Vegetation Height	2012, 2014, 2016
Existing Vegetation Type	2012, 2014, 2016
Elevation	2016
Slope	2016

To reconcile the different spatial resolutions, we divide the spatial area under consideration (the state of California) into a grid of 375 $\times$ 375 meter cells, resulting in over 3 million polygons. The center of each fire pixel from the vector data can therefore overlap with exactly one cell in the grid. To compute the corresponding vegetation, fuel, and topographic information associated with each data point, we compute *zonal statistics* for the vector data using the raster data. The method of zonal statistics refers to calculating summary statistics using a raster dataset within zones defined by another dataset (typically in vector form).

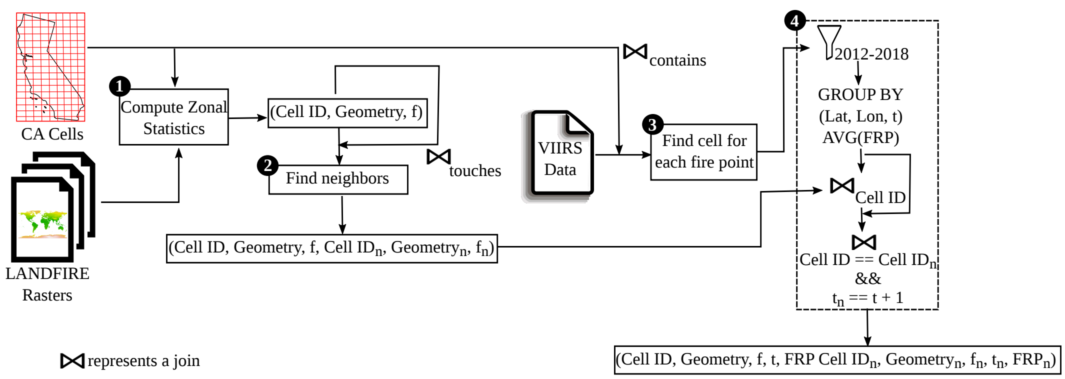

Figure 1: Data Generation Process

2.1 Data Generation

The data generation process as depicted in Figure 1 includes the following steps: 1. Compute zonal statistics for each spatial cell (in the form of a polygon in the vector data) using the Landfire rasters. 2. Find the geographical neighbors for each cell. 3. For each fire point in VIIRS data, find its corresponding cell. 4. For each fire observed in VIIRS data (denoted by x_i), generate tuples $\{x_i, t_i, f_i, x_j, t_{i+1}, f_j\}$, where x_i and x_j are neighbors, x_i is burning at timestamp t_i and x_j may or may not be burning at timestamp t_{i+1}. f_i and f_j are the respective feature vectors (zonal statistics and FRP) for the fire points. We describe each step below.

1. Compute Zonal Statistics: For each spatial cell in the 375m $\times$ 375m grid placed over California and for each raster dataset mentioned in Table 1, we want to compute aggregated feature vectors. To compute zonal statistics, we employ a fully distributed version of the system proposed in [Singla and Eldawy, 2018] on an Amazon AWS EMR cluster with one head node and 19 worker nodes of type m4.2xlarge with 2.4 GHz Intel Xeon $E5 - 2676$ v3 processor, 32 GB of RAM, up to 100 GB of SSD, and 2$\times$8-core processors. This system can work with data in their native formats by computing an intermediate data structure called *intersection file* that maps raster to vector data. The creation of *intersection file* also facilitates the use of distributed computing to compute zonal statistics. The system takes approximately two hours to compute zonal statistics for all the rasters mentioned in Table 1. It outputs a collection of tuples $(Cell\ ID, Geometry, f)$ where $Cell\ ID$ is a unique identifier for each cell in the spatial grid placed, $Geometry$ refers to the actual spatial geometry of

Table 2: *WildfireDB* dataset example. Each column is a data entry.

Cell ID	7234	7380	…
Date	2012-01-16	2012-01-06	…
FRP	3.2	5.1	…
Cell ID$_n$	7233	7233	…
FRP$_n$	0.0	0.0	…
Canopy Base Density max.	13.0	100.0	…
Canopy Base Density min.	0.0	0.0	…
Canopy Base Density median	9.0	8.0	…
…	…	…	…
Slope$_n$ **sum**	3109.0	3109.0	…
Slope$_n$ **mode**	24.0	24.0	…
Slope$_n$ **count**	169.0	169.0	…
Slope$_n$ **mean**	18.396450	18.396450	…

the cell, and f denotes the set of statistics calculated for each cell using all the Landfire rasters.
2. Find neighbors: The neighbors for each cell in the spatial grid are computed by doing a spatial self join using the predicate *touches* on the $Geometry$ values of the tuples generated in the previous step. The predicate *touches* returns true, if only the boundaries of the cells intersect. This spatial join is implemented using SpatialHadoop [Eldawy and Mokbel, 2015]. It outputs a collection of tuples $(Cell\ ID, Geometry, f, Cell\ ID_n, Geometry_n, f_n)$ where each tuple in the previous step is appended by the tuples of one of its neighbors (we use subscript n to denote variables that refer to the neighbors of the cell in consideration).
3. Find cell for each fire point: For specific points (latitude-longitude pairs) in VIIRS data and the cells in our spatial grid, a spatial join using the predicate *contains* is performed to find the cell that each fire point is contained in. The predicate *contains* returns true, if and only if the fire point lies in the interior of the cell. This step is implemented using SpatialHadoop [Eldawy and Mokbel, 2015].
4. Generate tuples: To generate the final tuples for *WildfireDB*, we start by filtering the tuples in the VIIRS data for the years 2012 – 2018. The VIIRS dataset may contain multiple tuples for the same fire point having the same timestamp yet different FRP values. We group all such tuples by the fire point and timestamp and average the FRPs to generate a single tuple with this average FRP. The resulting VIIRS tuples are then joined with tuples from Step 2 based on the $Cell\ ID$. This results in tuples of the form $(Cell\ ID, Geometry, f, t, FRP, Cell\ ID_n, Geometry_n, f_n)$, where t is the timestamp of the fire incidence from VIIRS data and FRP is the average FRP calculated previously. The next step is to perform a left join on these tuples with the VIIRS data based on the neighbor's cell identifier $Cell\ ID_n$ and on the condition that the neighbor's timestamp $t_n = t + 1$. This results in tuples of the form $(Cell\ ID, Geometry, f, t, FRP, Cell\ ID_n, Geometry_n, f_n, t_n, FRP_n)$. If the condition on timestamp is not satisfied, the value of FRP_n is set to zero, i.e. no fire.

2.2 WildfireDB Description

Our data has a total of 2,367,209 data points. Each data entry of *WildfireDB* corresponds to a specific 375-meter $\times$ 375-meter polygon at a given point in time, and the status of one of its neighboring cells at the subsequent time step. Relevant covariates of both the cells are also available. The vegetation, fuel, and topography information consists of summary statistics (maximum, minimum, median, sum, mode, count and mean of each of the data categories). The data is available at URL (suppressed for blind review). A data example is shown in Table 2.

3 Discussion

Wildfires affect large areas and are expected to grow in frequency and severity [Joseph et al., 2019]. To better analyze and study fires, we created *WildfireDB*, the first comprehensive dataset on wildfires that combines historical fire data with relevant covariates fused from heterogeneous data sources. Our dataset, with over a million data points for California, is open-source for the research community to use. Forecasting the spread of wildfires is crucial to develop models of resource allocation and suppression. Although our data set is the first of its kind, there are some limitations that we highlight. A crucial determinant of how wildfires spread is wind. Our data does not include information about wind. We are currently incorporating hourly wind data from National Centers for Environmental Information (NCEI).[1] We are also augmenting the data set by adding data points of all fire occurrences in the contiguous United States.

Broader Impact

Wildfires have caused massive damage to lives and property in the last decade. In the four years between 2014 and 2018, the U.S. wildfire acreage increased from 3.6 million to 8.8 million acres [Stacker, 2020]. In order to mitigate and suppress wildfires, it is important to understand how fires originate and spread. We created the first open-source comprehensive data source that links fire occurrence with relevant covariates extracted from satellite imagery. We hope that our data set will help first responders and researchers better model and fight wildfires. Our dataset can be used to build generative models for fire spread, which in turn can be used to create principled response strategies against wildfires [Diao et al., 2020].

Acknowledgments and Disclosure of Funding

This work is supported in part by Agriculture and Food Research Initiative Competitive Grants no. 2019-67022-29696 and 2020-69012-31914 from the USDA National Institute of Food and Agriculture, Department of Management Science & Engineering at Stanford University, and the Center of Automotive Research at Stanford (CARS).

References

Stefan H Doerr and Cristina Santín. Global trends in wildfire and its impacts: perceptions versus realities in a changing world. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 371(1696):20150345, 2016.

Kelsey Bartz. Record wildfires push 2018 disaster costs to $91 billion., 2019. URL `www.c2es.org/2019/02/record-wildfires-push-2018-disaster-costs-to-91-billion/`.

Maxwell B Joseph, Matthew W Rossi, Nathan P Mietkiewicz, Adam L Mahood, Megan E Cattau, Lise Ann St. Denis, R Chelsea Nagy, Virginia Iglesias, John T Abatzoglou, and Jennifer K Balch. Spatiotemporal prediction of wildfire size extremes with bayesian finite sample maxima. *Ecological Applications*, 29(6), 2019.

Richard C Rothermel. *A mathematical model for predicting fire spread in wildland fuels*, volume 115. Intermountain Forest & Range Experiment Station, Forest Service, 1972.

Patricia L Andrews. *BEHAVE: fire behavior prediction and fuel modeling system: BURN subsystem, Part 1*, volume 194. US Department of Agriculture, Forest Service, Intermountain Research Station, 1986.

Mark A Finney. *FARSITE, Fire Area Simulator–model development and evaluation*. Number 4. US Department of Agriculture, Forest Service, Rocky Mountain Research Station, 1998.

Omid Ghorbanzadeh, Khalil Valizadeh Kamran, Thomas Blaschke, Jagannath Aryal, Amin Naboureh, Jamshid Einali, and Jinhu Bian. Spatial prediction of wildfire susceptibility using field survey gps data and machine learning approaches. *Fire*, 2(3):43, 2019.

Ayan Mukhopadhyay, Geoffrey Pettet, Sayyed Vazirizade, Di Lu, Hiba Baroud, Alex Jaimes, Yevgeniy Vorobeychik, Mykel Kochenderfer, and Abhishek Dubey. A review of emergency incident prediction, resource allocation and dispatch models, 2020.

Peter Baumann, Andreas Dehmel, Paula Furtado, Roland Ritsch, and Norbert Widmann. The multidimensional database system rasdaman. pages 575–577, Seattle, WA, June 1998.

Paul Brown, Donghui Zhang, and Jacek Becla. SciDB: A Database Management System for Applications with Complex Analytics. *Computing in Science and Engineering*, 15(3):54–62, 2013.

[1]https://ncei.noaa.gov/

Jianting Zhang, Simin You, and Le Gruenwald. Efficient parallel zonal statistics on large-scale global biodiversity data on gpus. In *Proceedings of the 4th International ACM SIGSPATIAL Workshop on Analytics for Big Geospatial Data*, pages 35–44, 2015.

Samriddhi Singla and Ahmed Eldawy. Distributed zonal statistics of big raster and vector data. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 536–539, 2018.

Wilfrid Schroeder, Patricia Oliva, Louis Giglio, and Ivan A Csiszar. The new VIIRS 375 m active fire detection data product: Algorithm description and initial assessment. *Remote Sensing of Environment*, 143:85–96, 2014.

Kevin C Ryan and Tonja S Opperman. Landfire–a national vegetation/fuels data base for use in fuels treatment, restoration, and suppression planning. *Forest Ecology and Management*, 294:208–216, 2013.

Ahmed Eldawy and Mohamed F Mokbel. Spatialhadoop: A mapreduce framework for spatial data. In *2015 IEEE 31st international conference on Data Engineering*, pages 1352–1363. IEEE, 2015.

Stacker. Largest wildfires of the decade. `https://stacker.com/stories/3688/largest-wildfires-decade`, 2020.

Tina Diao, Samriddhi Singla, Ayan Mukhopadhyay, Ahmed Eldawy, Ross Shachter, and Mykel Kochenderfer. Uncertainty aware wildfire management. *arXiv preprint arXiv:2010.07915*, 2020.

A Appendix

This appendix presents an analysis of the computational complexity of raster-based approach, vector-based approach and the EMI approach [Singla and Eldawy, 2018] used in this paper.

A.1 Raster Approach (RA)

The raster-based approach requires to create a separate raster layer for each polygon in the vector dataset. It then scans each pixel in this rasterized (mask) layer and the corresponding pixels in the input raster layer in order to compute the desired aggregate function. It takes at most T_{RA} time computed as

$$T_{RA} = n_p \cdot c \cdot r \tag{1}$$

$c \cdot r$ represents the time taken to scan each pixel in the raster layer and n_p is the number of polygons in the vector layer.

A.2 Vector Approach (VA)

The vector-based approach converts each pixel in the raster to a point and then test the point against each polygon in the vector data to find a match. This approach can be optimized by creating an index for the vector dataset. However, it would still require scanning the whole raster dataset and converting each pixel to a point. It takes at most T_{VA} time computed as

$$T_{VA} = n_p \log n_p \cdot c \cdot r \tag{2}$$

$n_p \log n_p$ represents the time taken for the index lookup for each pixel in the raster layer with c columns and r rows.

A.3 EMI Approach

The EMI approach computes an intermediate data structure called *intersection file* using the vector layer and the metadata from raster layer. The *intersection file* serves as a mapping between the raster and vector layer, and can be used to compute the desired aggregate function in one scan over the raster layer.

It takes at most T_{EMI} time computed as

$$T_{EMI} = n_p \log n_p + c \cdot r \tag{3}$$

$n_p \log n_p$ represents the time taken to compute the *intersection file* while $c \cdot r$ is the time taken to complete one scan of the raster layer with c columns and r rows.