Wildfires risk assessment using machine learning

A Preprint

Zharov Georgii
MIPT
zharov.g@phystech.edu

Abstract

The paper considers the problem of predicting extreme climatic phenomena, namely forest fires. The goal is to predict fires based on already available data on phenomena of this type using deep learning methods. It is proposed to study the time series for stationarity. Further, for short-term forecasting, stationary time series are studied in more detail. For the implementation of long-term forecasting, non-stationary time series are studied.

Keywords Time series · Extreme events

1 Introduction

Forecasting extreme climatic events is an important applied task, since natural disasters can cause significant harm to many areas of the human economy. The purpose of this work is to build a model for predicting the occurrence of forest fires. To solve the problem, it is proposed to study the behavior of time series, stationary - for predicting phenomena over a time interval of about $4-5$ years, and non-stationary - for predicting phenomena within $40-50$ years.

The main difficulty in solving the problem of predicting extreme events is that these events are random and occur quite rarely - the time interval between two neighboring events in the region under consideration can be very large. Due to this feature of the phenomenon under study, when solving the problem, we may encounter the problem of unbalanced data. This problem is as follows. When working with data, we introduce certain threshold values for the features under consideration, and events for which the feature values exceed the threshold values will be considered extreme. Since the events of interest to us occur quite rarely, only a small part of the events in the sample will be labeled as extreme. Thus, most of the data will be within the thresholds, and the most interesting events for us will be in the minority, and the sample will be unbalanced. Due to the imbalance of the input data, when training the model, we may encounter both the problem of underfitting and the problem of overfitting, examples of this are given in [1].

The main goal of the work is to find an architecture that will better cope with the task of predicting the behavior of time series and predicting extreme events than existing models. Classical models do not cope well with the task. Therefore, for the solution, it is proposed to use an architecture with elements of recurrent neural networks (RNNs) capable of storing information about previous events in memory. In work [1], examples of unsuccessful work of classical models are given, both cases of overfitting and cases of underfitting are demonstrated. Also in the paper [1] one can find an explanation why the use of a quadratic loss function in this problem leads to bad results. This is because most commonly used distributions, such as the Gaussian or Poisson distribution, do not describe heavy-tailed data. This is the name of the data, among which a small amount contains events with a small probability of origin. In our problem, extreme events can be interpreted as such data. To solve this, it is proposed to add the so-called Extreme Value Loss (EVL) [1] to the quadratic loss function. This solution allows you to take into account heavy-tailed data and correctly train the model on them.

As the main dataset, information about forest fires in the United States over the previous several decades from the Google Earth Data service is used. It also uses data from the Wildfire Risk Database and Severe Weather Dataset. Severe Weather Dataset

2 Problem statement

The input data are a set of two-dimensional geographical points (x_i^1, x_i^2), time t and vector of climatic parameters h. Our model's prediction for time o_t. The output for time t is y_t. $T-$ number of time points to consider. We can take the quadratic loss function and pose the minimization problem as follows

$$\min \sum_{t=1}^{T} \|o_t - y_t\|^2$$

But, as mentioned in the introduction, this approach will not be optimal. Instead, in accordance with the algorithm proposed in the article [1], we can choose a loss function and set the optimization problem as follows

$$\min \sum_{t=1}^{T} (\|o_t - y_t\|^2 + \lambda_1 EVL(w_t, u_t))$$

where λ_1 and w_t are parameters. Function EVL:

$$EVL(u_t) = -(1 - P(v_t = 1))[\log G(u_t)]v_t \log(u_t) - (1 - P(v_t = 0))[\log G(1 - u_t)](1-)v_t \log(1 - u_t) =$$

$$= \beta_0[1 - \frac{u_t}{\gamma}]^{\gamma} v_t \log(u_t) - beta_1[1 - \frac{1 - u_t}{\gamma}]^{\gamma}(1 - v_t) \log(1 - u_t)$$

where $\beta_0 = P(v_t = 0), \beta_1 = P(v_t = 1)$, $v_t = \{0, 1\}$ it is an indicator showing whether we consider the given event to be extreme or not. $G(y)$ it is the modified distribution:

$$G(x) = \begin{cases} \exp(-(1 - \frac{1}{\gamma}y)^{\gamma}) & \gamma \neq 0, 1 - \frac{1}{\gamma}y > 0 \\ \exp(-e^{-y}) & \gamma = 0 \end{cases}$$

3 Theory

It is offered to use memory network to memorize extreme events. This is done to find patterns in historical data. First, we define the concept of a historical window.

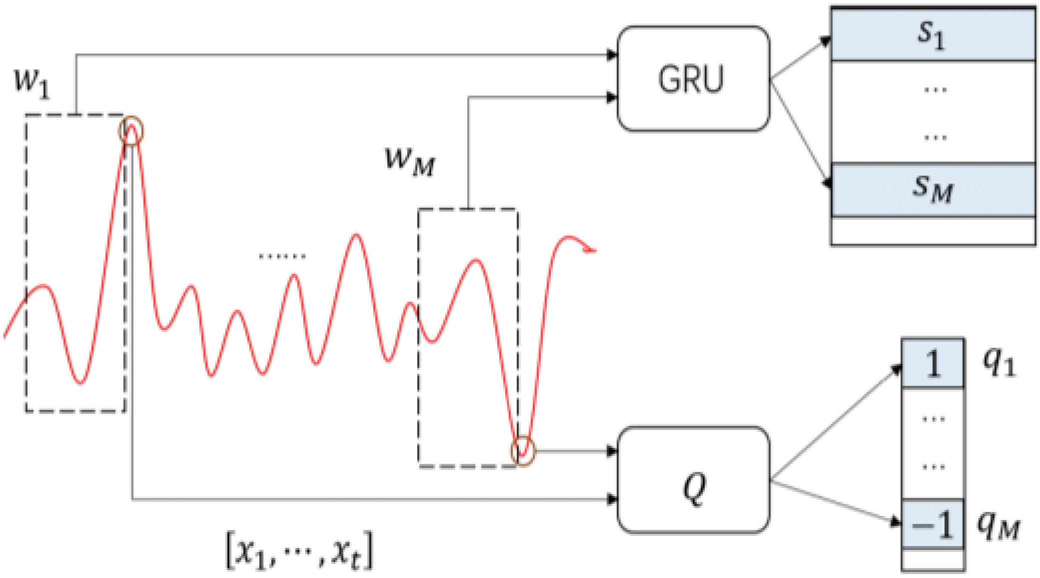
3.1 Historical window

For each time step t, we first randomly sample a sequence of windows by $W = \{w_1, \ldots, w_M\}$, where M is the size of the memory network. Each window w_j defined as $w_j = [x_{t_j}, x_{t_j+1}, \ldots, x_{t_j+\Delta}]$, where Δ as the size of the window satisfying $0 < t_j < t - \Delta$. Then we apply GRU module to embed each window into feature space. Specifically, we use w_j as input, and regard the last hidden state as the latent representation of this window, denoted as $s_j = GRU([x_{t_j}, x_{t_j+1}, \ldots, x_{t_j+\Delta}]) \in R^H$. We apply memory module to memorize whether there is a extreme event in $t_j + \Delta + 1$ for each window w_j. Also, we propose to feed memory module bu $q_j = v_{t_j+\Delta+1} \in \{0, 1\}$. In summary, at each time step t, the memory of our proposed architecture consists of the following two parts:

1) Embedding Module $S \in R^{M \times H}$: s_j is the latent representation of history window j.

2) History Module $Q \in \{0, 1\}^M$: q_j is the label of whether there is a extreme event after the window j.

The memory module diagram is shown below



3.2 Attention mechanism

This module is incorporated for imbalanced time-series predictions. At each time step t, we use GRU to produce the output value:

$$\widetilde{o}_t = W_o^T h_t + b_o,\ where\ h_t = GRU([x_{t_j}, x_{t_j+1}, \ldots, x_{t_j+\Delta}])$$

Prediction of $\widetilde{o}_t$ may lack the ability of recognizing extreme events in the future. Therefore we also require our model to retrospect its memory to check whether there is a similarity between the target event and extreme events in history. To achieve this, we propose to utilize attention mechanism:

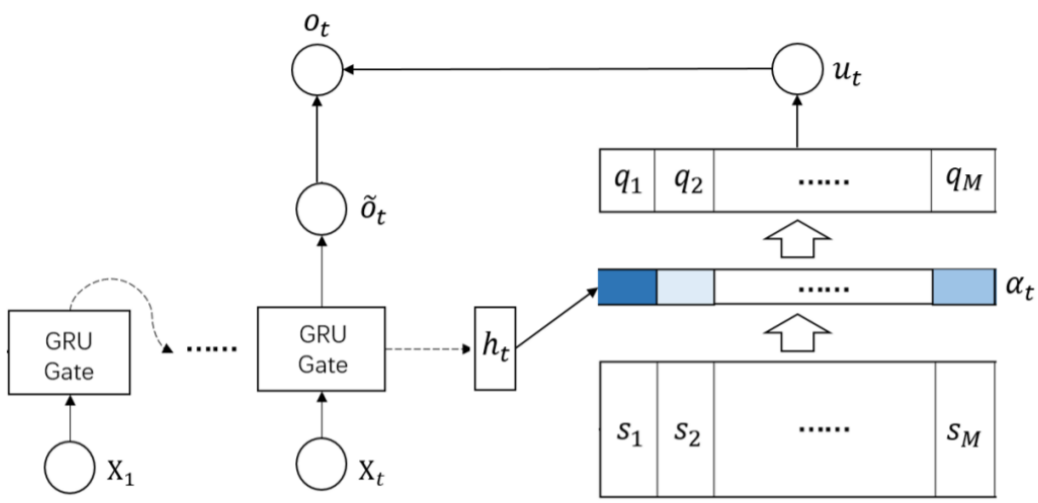
$$\alpha_{tj} = \frac{\exp(c_{tj})}{\sum\limits_{j=1}^{M} \exp(c_{tj})},\ where\ c_{tj} = h_t^T s_j$$

Finally, the prediction of whether an extreme event would happen after referring historical information can be measured by imposing attentive weights on q_j. The output of our model at time step t is calculated as

$$o_t = \widetilde{o}_t + b^T u_t,\ where\ u_t = \sum_{j=1}^{M} \alpha_{tj} q_j$$

In the definition $u_t \in [1, 1]$ is the prediction of whether there will be an extreme event after time step t, and $b \in R_+$ is the scale parameter.

The main advantage of this model is the ability to switch between yielding predictions of normal values and extreme values. But in addition to these architectural features, to work with extreme events, it is necessary to modify the loss function. More about it below.



3.3 Extreme Value Loss

As mentioned above, the common square loss function is not suitable for working with heavy-tailed data. To do this, as a correction, the Extreme Value Loss is added to it.

$$EVL(u_t) = -(1 - P(v_t = 1))[\log G(u_t)]v_t \log(u_t) - (1 - P(v_t = 0))[\log G(1 - u_t)](1-)v_t \log(1 - u_t) =$$
$$= \beta_0[1 - \frac{u_t}{\gamma}]^\gamma v_t \log(u_t) - beta_1[1 - \frac{1 - u_t}{\gamma}]^\gamma (1 - v_t) \log(1 - u_t)$$

3.4 Optimization

First, in order to incorporate EVL with the proposed memory network, a direct thought is to combine the predicted outputs o_t with the prediction of the occurrence of extreme events:

$$L_1 = \sum_{t=1}^{T} (\|o_t - y_t\|^2 + \lambda_1 EVL(u_t, v_t))$$

Furthermore, in order to enhance the performance of GRU units, is proposed to add the penalty term for each window j, which aims at predicting extreme indicator q_j of each window:

$$L_2 = \sum_{t=1}^{T} \sum_{j=1}^{M} EVL(p_j, q_j)$$

, where $p_j \in [1, 1]$ is calculated through s_j, which is the embedded representation of window j, by a full connection layer

Список литературы

[1] Daizong Ding, Mi Zhang, Xudong Pan, Min Yang 0002, and Xiangnan He 0001. Modeling extreme events in time series prediction. In Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis, editors, KDD, pages 1114–1122. ACM, 2019.