ANOMALY DETECTION

Lesson 7: Time Series Anomaly Detection

Learning objectives

You will be able to:

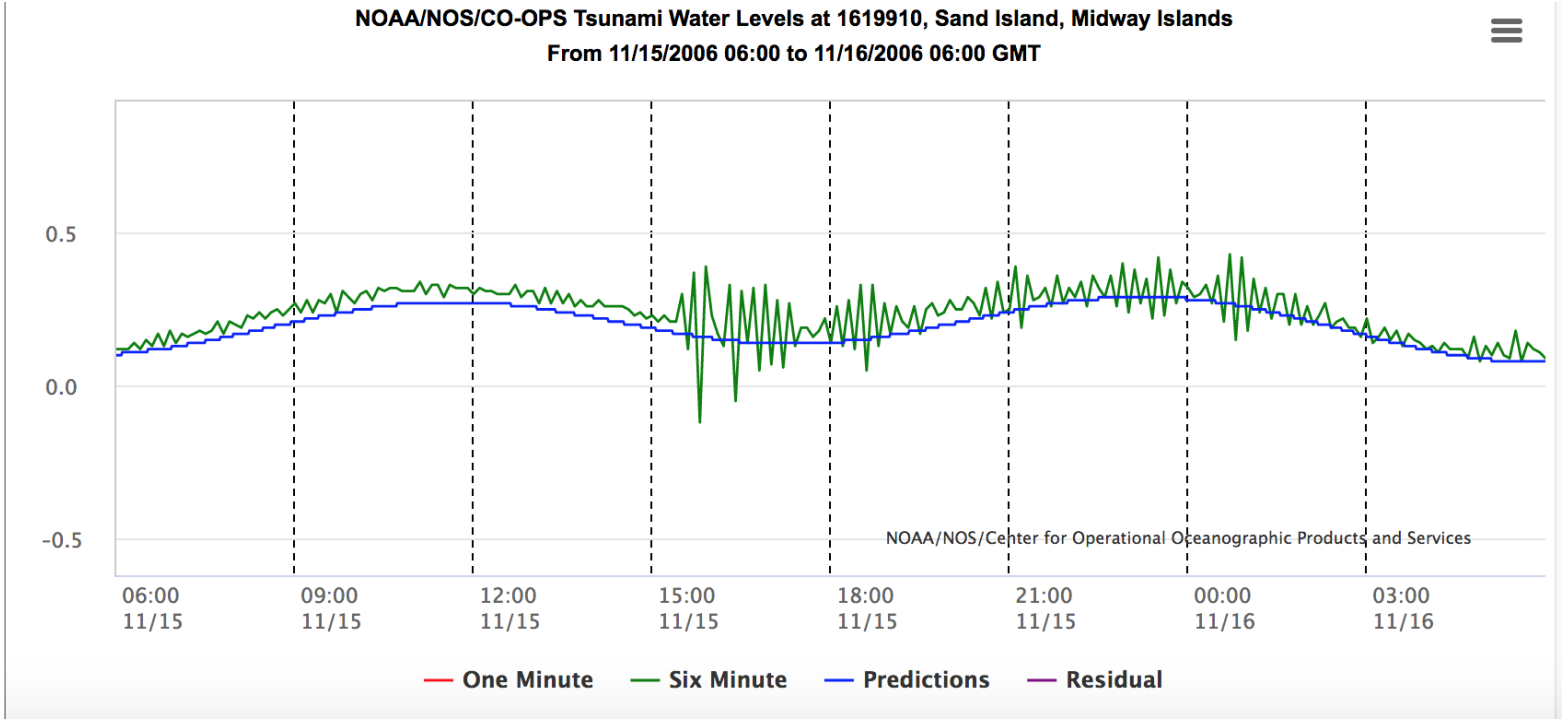
- Describe time series anomaly detection
- Apply statistical process control for anomaly detection
- Apply streaming anomaly detection using autoregressive models
- Use Python* to perform time series anomaly detection

Time series anomaly detection

Introduction

- Time series: a series of data points arranged sequentially in time, usually at equally spaced intervals.
- Time series data is ubiquitous in weather, finance, medicine, engineering
- Examples: daily temperatures, stock prices, heart rate, power grid voltage

Example: tsunami warning



Time series anomaly detection

Nature does not make jumps

- Time is special a feature as it provides a natural ordering of the data.
- Furthermore, when looking at time series data we expect changes to be gradual. This assumption of temporal continuity is often used to detect anomalies in time series.
- Anomaly = sudden change in data values
  - Can be a jump at a single value or change in trend

Time series anomaly detection

What kind of data do you have?

- Offline: you have all the data you are interested in
  - For any point, you can use information from the past, present and future
- Streaming: all you know are the present and the past
  - To detect anomalies, you often have to predict the future and compare the new data with your prediction (see tsunami warning example)
- We will look at both approaches in this lesson

Statistical process control

Quality control of the data

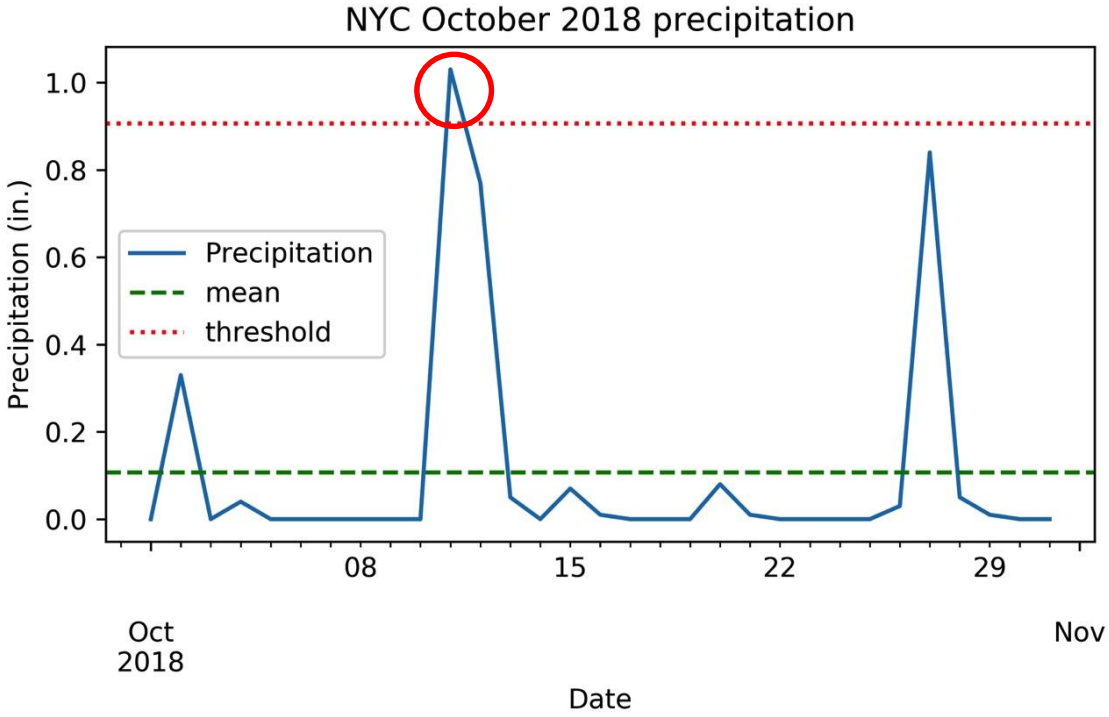
- Statistical methods used to monitor a process
- Anomaly = data exceeds the threshold for a pre-defined statistic
- Often carried out through visual display of the data: control charts
- Here we consider two sets of statistics:
  - Mean and standard deviation
  - Cumulative sum

Control chart

Mean and standard deviation

- Time series of N data points: x_i (temperature, price, etc. at time point $i=1,2,...N$)
- Decide on a threshold z for anomaly detection
- Calculate the mean (μ) and standard deviation (σ)
- A point is labeled as an anomaly if

$$\left| \frac{x - m}{s} \right| > z$$



Anomaly:

October 11, 2018: precipitation of 1.03 in.

Control chart: mean/standard deviation

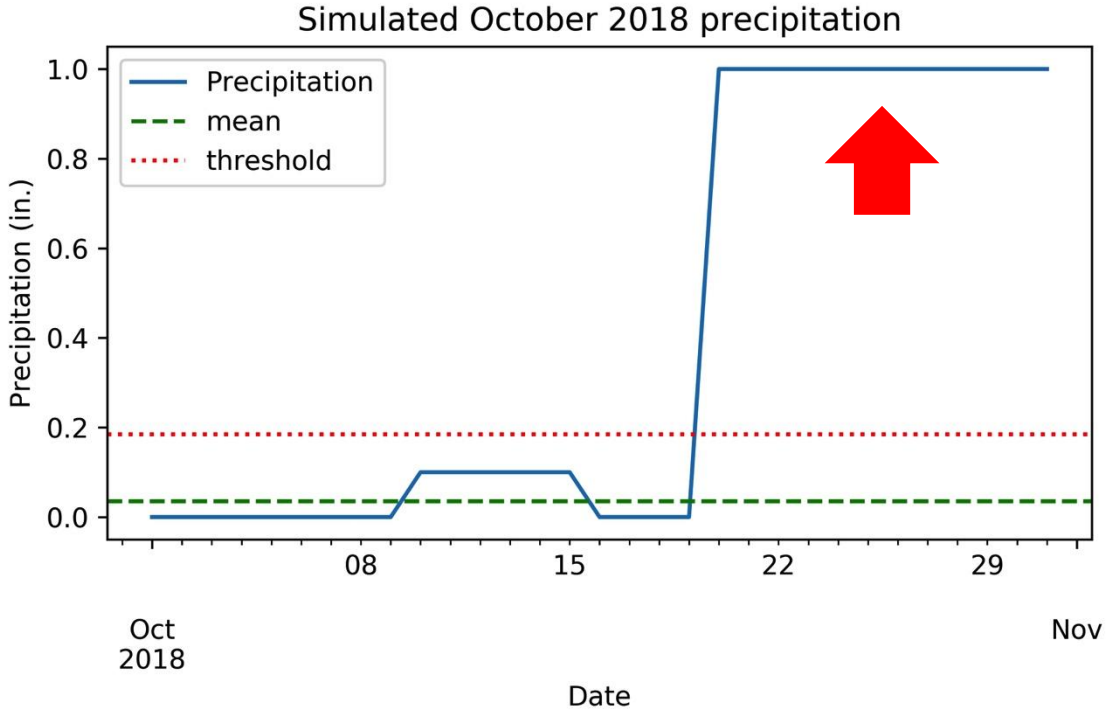
Strengths

- Simple to implement
- Easy to interpret
- Sensitive to point anomalies
- Can be used in both offline mode and with streaming data
- If underlying distribution is known, can relate threshold to probability of observing the event

Control chart: mean/standard deviation

Weakness

- Requires reasonable estimates of mean and standard deviation
  - Ideally, such estimates should come from uncontaminated normal data
- Susceptible to false positives
  - Given enough time, an event far from the mean should happen
- Cannot adapt to changing trend



Anomaly:

All of data from Oct. 20 to Oct. 31

A new trend?

Cumulative sum (CUSUM)

A control chart to detect small shifts in the process mean

- Calculate the cumulative sum of the values in the time series
- When the cumulative sum exceeds a threshold, a change has been found
- Can detect changes missed by standard control chart (based on z-score)

A tool for change point detection

CUSUM

The algorithm

- Track the cumulative sums of the positive deviation from the mean (“the high sum”) and the negative deviation from the mean (“the low sum”)
- Consider the time series described previously:
  - x_i with $i=1,2,...N$
  - mean μ and standard deviation σ
- Denote size of the shift to be registered by λ
- Denote threshold by w (often a multiple of λ or σ)

CUSUM

The high sum

- Given by the following recursive sequence:

$$S_i^+ = \max\left[0, S_{i-1}^+ + x_i - m - l\right] \qquad i = 1, 2, \ldots N$$

- The initial condition is $S_0^+ = 0$
- The shift l is a measure of the leeway in the process
  - The high sum increases only if $x_i > m + l$
- If $S_i^+ > w$ then a change point/anomaly is reported

CUSUM

The low sum

- Given by the following recursive sequence:

$$S_i^- = \min\left[0, S_{i-1}^+ + x_i - m + l\right] \qquad i = 1, 2, \ldots N$$

- The initial condition is $S_0^- = 0$
- As before, the shift l is a measure of the leeway in the process
  - The low sum decreases only if $x_i < m - l$
- If $S_i^+ < -w$ then a change point/anomaly is reported

CUSUM in action

Example

- Data is drawn from a normal distribution
- $\mu = 0$ and $\sigma = 1$
  - Value of data is value of z-score
- Control chart with z=3 threshold: no anomalies detected
- High sum of CUSUM detects anomaly at last point
  - $\lambda = \sigma$ and $w = 5\sigma$
  - Suggests change in process at end of time series

x	high sum
-1.178	0.000
0.322	0.000
-0.985	0.000
-0.351	0.000
-0.859	0.000
1.105	0.105
-0.770	0.000
1.046	0.046
-0.621	0.000
1.456	0.456
1.097	0.553
2.501	2.055
1.980	3.035
1.601	3.636
2.714	5.350

Autoregressive models

Detecting anomalies through modeling

- The idea: make a model to predict the value of the data at a given time
- Compare the predicted value with the actual value
- If the difference exceeds a threshold, label the point as an anomaly
- In Lesson 3, we applied this idea to regression-based modeling
  - Dependent variable vs. independent variable(s)
- Here we use autoregressive models
  - Present value depends on past values

Autoregressive models: *AR(p)* model

Univariate time series

- Data: $X_1, X_2, \ldots X_t$
- Assume value at any given time depends on preceding *p* points [AR(p) model]:

$$X_t = \sum_{i=1}^{p} a_i X_{t-i} + c + e_i$$

- Coefficients $\{a_i\}$ and *c* to be determined from (training) data
- Residuals ε_t are the unexplained behavior—use as an anomaly score

Additional models

- Include **m**oving **a**verage (past deviations): **ARMA** model

$$X_t = \sum_{i=1}^{p} a_i X_{t-i} + \sum_{i=1}^{q} b_i e_{t-i} + c + e_t$$

- Further modeling may be necessary to account for other factors:
  - Trends (differencing of data; “integrated model”)
  - Seasonality (periodic variation of the data)
  - Exogenous variables (allows external variables to be considered)

CONCLUSION

Use Python* for anomaly detection

Next up is a look at applying these concepts in Python*

- See notebook entitled *Time_Series_Anomaly_Detection_student.ipynb*

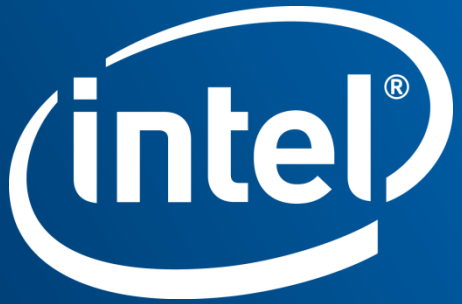
Learning objectives recap

In this session you learned how to:

- Describe time series anomaly detection
- Apply statistical process control for anomaly detection
- Apply streaming anomaly detection using autoregressive models
- Use Python* to perform time series anomaly detection

References

- *Introductory Overview of Time-Series-Based Anomaly Detection* by A. Moore
- *Detection of Abrupt Changes: Theory and Application* by M. Basseville and I.V. Nikiforov (Prentice Hall, 1993)
- *Time Series Analysis course on the Intel Developer Zone*


intel®