Master's Degree in Mechatronic Engineering

Politecnico di Torino

Master's Degree Thesis

UNSUPERVISED MACHINE LEARNING ALGORITHMS FOR EDGE NOVELTY DETECTION

Supervisors

Prof. Marcello CHIABERGE

Dott. Umberto ALBERTIN

Dott. Gianluca DARA

Candidate

Ariel PRIARONE

April 2024


positive proposed healthy normal RUL faulty
tree Now IMS dataset
framework value time first possible
known novel measure way file 42 solution
linear signal used collection classification
two application python
no see standard outlier Fourier plot also shown vector output
snapshot data use profile microcontroller
10 Novelty detection
forest edge type train phase called defined library
Model case dataset
selecting 100 technique purpose xi
decision 1.0 space clustering Results
function done new cluster Wavelet
sampling Transform set sample database
feature system K-mean
able cost Let's last may Previous
even implemented result Fault
section just machine agent
better device float state ML vibration
number comparison metric Figure FD it's described
problem detection parameter
GMM work axis false sensor example
algorithm frequency
12 task LOF degradation
trained line centroid score threshold
Chapter training DBSCAN structure end perform
distance select MLA Test evaluation
configuration one event Table ND
common learning approach large instance
point extraction evaluate Maintenance FFT user unsupervised
novelty procedure method field validation condition
silhouette Bearing class prediction applied different
implementation Since detect another mean based behaviour considered

Acknowledgements

I would like to thank the PoliTO Interdepartmental Centre for Service Robotics (PIC4Ser) for giving me the opportunity to work on this project. The guidance and infrastructure provided by the centre have been invaluable during the development of this work.

To my parents, who have given me everything.
Thank you for always making me believe that anything is possible.
Ai miei genitori, che mi hanno dato tutto.
Grazie per avermi sempre fatto credere che tutto sia possibile.

Ariel

Table of Contents

List of Tables

List of Figures

Appendix A

Fourier Transform

This appendix aims to provide a brief and non-exhaustive introduction to the Fourier Transform. The various types of Fourier Transform are key tools in a vast range of fields in engineering, physics and mathematics. In computer science, the Fourier Transform is used in signal processing, image processing, data compression, and many other applications.

A.1 Continuous Fourier Transform

Most of the modern signal processing techniques find their roots in the Fourier Transform, which is a mathematical tool that allows the decomposition of a non-periodic signal into its frequency components. For periodic signals, The transform is a train of impulses, whose amplitudes are linked to the Fourier series coefficients.

The Continuous Fourier Transform (CFT) of a signal $x(t)$ is defined as:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt, \qquad \forall f \in \mathbb{R}$$

where $X(f)$ is the Fourier Transform of $x(t)$, f is the frequency variable and j is the imaginary unit. The quantity $2\pi f$ is the angular frequency.

A.2 Discrete Fourier Transform

The Discrete Fourier Transform (DFT) is a sampled version of the Continuous Fourier Transform. From an engineering perspective, the DFT is of particular interest because most of the signals are sampled in time and therefore, the DFT is the most commonly used form of the Fourier Transform.

The DFT is used to transform a sequence of N complex numbers $x(n)$ into another sequence of N complex numbers $X(k)$. The DFT is defined as:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}, \qquad \forall k \in \{0, 1, \ldots, N-1\} \tag{A.1}$$

where $X(k)$ is the DFT of $x(n)$, k is the frequency index and N is the number of samples in the sequence.

The output spectrum given by the DFT is also discrete and finite, with the same number of samples as the input signal. The sampling frequency of the signal $x(n)$ determines the upper frequency limit of the DFT output, as the faster the sampling rate, the higher the frequencies that can be represented. The length of the signal $x(n)$ determines the frequency resolution of the DFT output, as the longer the signal, the finer the frequency resolution is.

Computational complexity Examining the definition of the DFT in Equation A.1, we can notice that it requires N complex multiplications and $N(N)$ complex additions. Therefore, the computational complexity of the DFT is $\mathcal{O}(N^2)$, which does not scale well for large values of N.

A.3 Fast Fourier Transform

To overcome the computational complexity of the DFT, the Fast Fourier Transform (FFT) algorithm was developed in the 1960s by Cooley and Tukey [1]. Choosing a data length N that is a power of 2, the FFT algorithm reduces the computational complexity of the DFT from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log_2 N)$. This becomes particularly important for large values of N, where the FFT algorithm is significantly faster than the DFT.

In the classic algorithm, a single multiplication happens many times. The general idea of this fast algorithm is to exploit the periodicity of the complex exponentials in the DFT to divide the computation into smaller sub-problems and reuse the already computed multiplication.

Breaking the exponential into its sine and cosine components, it becomes intuitive that the same result appears multiple times in the computation. The FFT algorithm takes advantage of this redundancy to reduce the number of operations, recursively dividing the problem into half-sized subproblems up to the point that the subproblems have only one sample.

Appendix B

Wavelets

B.1 Introduction

In the previous chapter, the FFT has been briefly described. It maps N time-domain datapoints into N frequency-domain datapoints. The FFT retain the information about the original signal, expressing it in the frequency domain. In other words, the FFT does not retain any knowledge about the time evolution of the signal and preserves the number of datapoints.

Another tool for signal processing is the wavelet. There are two main categories:

- Wavelet Transform (WT), which maps the datapoints of a 1D signal into a 2D plane, giving information about the frequency content of a signal at a given time.
- Wavelet Packet Decomposition (WPD), which is a tool that allows to decompose a signal into a tree of sub-bands, down-sampled signals.

A mother wavelet is a function that complies with the following conditions:

$$\int_{-\infty}^{\infty} \psi(t)dt = 0 \qquad \text{i.e. it has a mean value of zero.} \tag{B.1}$$

$$\int_{-\infty}^{\infty} |\psi(t)|^2 dt = 1 \qquad \text{i.e. it has a unitary energy.} \tag{B.2}$$

The Morlet wavelet is a complex wavelet, which complies with the above conditions. It is defined as:

$$\psi(t) = \frac{1}{\sqrt{\pi f_b}} e^{i2\pi f_c t} e^{-t^2/f_b} = \frac{1}{\sqrt{\pi f_b}} [\cos(2\pi f_c t) + j \cdot \sin(2\pi f_c t)] e^{-t^2/f_b} \tag{B.3}$$

where f_c is the central frequency and f_b is the bandwidth. The Morlet wavelet is a complex wavelet, which means that it has a real and an imaginary part, as shown in Figure B.1.

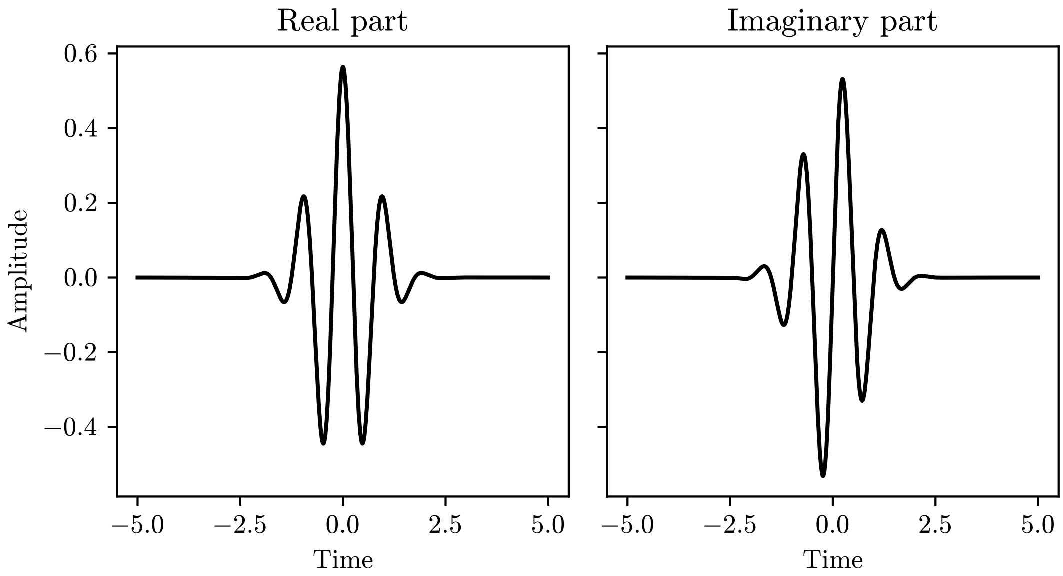

Figure B.1: *Real and imaginary part of the Morlet wavelet*

Bibliography

[1] James W Cooley and John W Tukey. «An algorithm for the machine calculation of complex Fourier series». In: *Mathematics of computation* 19.90 (1965), pp. 297–301 (cit. on p. 2).