Nonparametrics and kernels (Part 1, Transforms)

Daniel Lawson University of Bristol

Lecture 04.1.1 (v1.0.1)

Signposting

- We have looked at clustering methods, based on **algorithms**, **distances** or **models**.
- Clustering links to non-parametric statistics, which provides features that can be clustered.
- The **dimensionality reduction** session was one example of non-parametric statistics.
- This is part 1 of Lecture 4.1, which is split into:
  - 4.1.1 covers Transforms
  - 4.1.2 covers Density estimation
  - 4.1.3 covers the Kernel Trick.

Intended Learning Outcomes

- ILO1 Be able to **access and process cyber security data** into a format suitable for mathematical reasoning
- ILO2 Be able to **use and apply basic machine learning** tools
- ILO3 Be able to make and report appropriate inferences from the results of applying basic tools to data

Non-parametric statistics

- Non-parametric statistics come in several flavours:
  1. Parameter-free hypothesis tests
  2. **Zero-parameter** representations which can be thought of as a **data transformation**.
     - examples include: Time-Frequency transforms, Kernel methods
  3. **Infinite-parameter** representations which can be thought of as generalisations of parametric models.
     - examples include: Hierarchical Dirichlet Process, the Stochastic Block Model for graphs
- We covered 1 in testing. We touch on 3 later. This lecture is about 2.
- Most methods are **parametric nonparametrics**: it is rare that a data transformation method isn't naturally thought of with a parameter!

Transforming data

- In previous practical problems we've used simple transforms to make the data easier to model:
  - log-transform
  - square-root/power transform
- Some data simplify greatly when transformed appropriately:
  - periodic data are simpler after taking a frequency transform
- Bring in expertise on such transforms if you have it.
- Transformed data can be seen as feature augmentation, or latent embedding, depending on use.

The Basis Expansion

- Most transforms we consider are designed to exactly reproduce the data.
- These are **basis expansions** and are typically invertible.
- They make good feature sets if they result in a **dimensionality reduction**;
  - that is, they lead to a useful approximation using only a few features.
- PCA is one example of this.
- There are many others…

Fourier transform

- The Fourier transform is written:

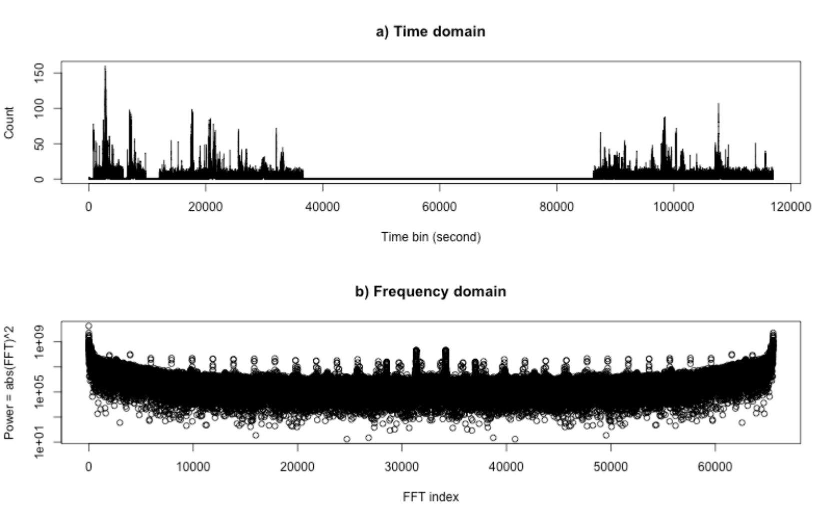
$$\hat{f}(\eta) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \eta} dx$$

  - The **Discrete Fourier Transform (DFT)** is used in practice as datasets typically have a minimum sampling rate δ.
  - It is usually computed using the **Fast Fourier Transform (FFT)**.
  - Consider using it for periodic data, or to look for periodicity.
- The **power** in any frequency i is proportional to $|\eta_i|^2$.
  - High power means this frequency is present in your data.
  - There are formal tests for “significance” of high power.

Fourier transform example

```
conndata_ts=data.frame(t=seq(min(conndata$ts),
            max(conndata$ts),by=1),x=0)
for(i in 1:dim(conndata)[1]){
    conndata_ts[ceiling(conndata[i,"ts"]-
                    conndata_ts[1,"t"]),"x"] =
            conndata_ts[ceiling(conndata[i,"ts"]-
                    conndata_ts[1,"t"]),"x"] + 1
}
# Not fast unless length(x)=2^k
myx=1:(2^16) # Largest valid choice
conndata_fft=fft(conndata_ts[myx,"x"])
```

Fourier transform example



Walsh-Hadamard transform

- The Walsh-Hadamard transform is a version of the Fourier Transform that is useful for **Binary data**.
- It is defined recursively via the **Hadamard Matrix**:

$$H_0 = 1,$$

$$H_m = \frac{1}{\sqrt{2}} \begin{pmatrix} H_{m-1} & H_{m-1} \\ H_{m-1} & -H_{m-1} \end{pmatrix}$$

- For N total bits, the whole matrix is of size $2^N \times 2^N$.
- The transform is $\mathrm{w} = \mathbf{H}\mathrm{x}$.
- w can be computed efficiently with the fast Walsh-Hadamard transform in complexity $O(N\log(N))$.
- It was developed in encryption & signals processing but is useful to generate features in many contexts.

Walsh-Hadamard matrices

$$H_1 = \frac{1}{\sqrt{2}} \left(\begin{array}{cc} 1 & 1 \\ 1 & -1 \end{array} \right)$$

Walsh-Hadamard matrices

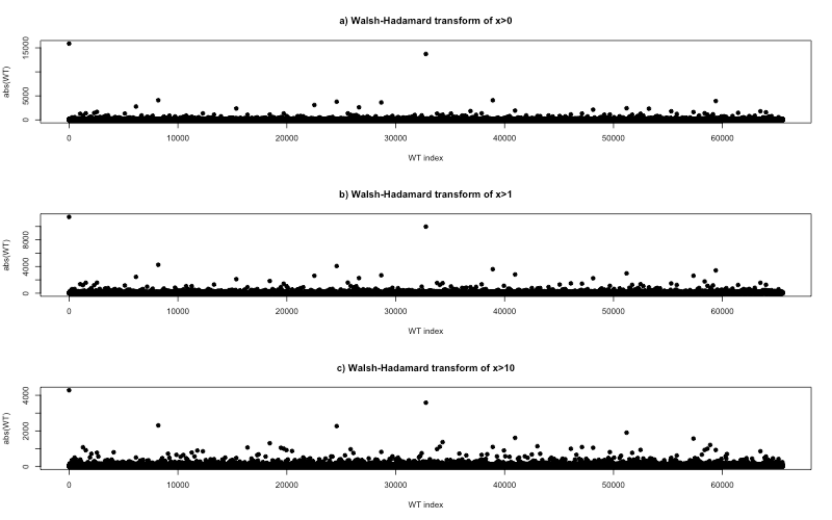
$$H_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$H_2 = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

Walsh-Hadamard transform examples

- Examples:
  - 00000… -> 00000…
  - 11111… -> +0000…
  - 01010… -> +-000…
  - 10101… -> ++000…
  - 10001000… -> +–+000…
  - 00010001… -> ++++000….
- i.e. the i-th bit is activated by a periodicity of length i
- The details are sensitive to the “phase”, i.e. exactly where in the sequence the periodicity lies.

Walsh-Hadamard transform example



Other transforms

- Other transforms exist and could be useful. For example:
  - Wavelets (time and space decomposition)
  - Laplace transform
  - Sine/ Cosine transforms
  - Hankel transform (radial basis function)
  - Polynomials
  - … etc
- All you need is a **basis function** and you have a **transform**.

Reflection

- What role could transforms play in classification?
- What other uses could you put them to? How do you know if they are working?
- Can you think of other classes of transform that could be useful? How would you test whether they were?
- How do these transforms generalise? What parameters does this introduce?
- By the end of the course, you should:
  - Be able to use transforms in practical cyber security questions
  - Be able to make appropriate judgement of whether a transform is worth trying
  - Be able to work with the Walsh-Hadamard tranform

Signposting

- Transforms are clearly linked to PCA from Block 03
- Further reading:
  - Nonparametric Statistics by Eduardo García Portugués
  - Basis Expansions: Chapter 5 of The Elements of Statistical Learning: Data Mining, Inference, and Prediction (Friedman, Hastie and Tibshirani).