Low-rate Non-intrusive Load Monitoring approaches via Graph Signal Processing

PhD Thesis

Kanghang He

in

Electronic and Electrical Engineering

University of Strathclyde, Glasgow

May 20, 2020

To my parents

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Abstract

The large-scale roll-out of smart metering worldwide brings many new application possibilities. One promising application is appliance-level energy feedback based on identifying individual loads from aggregate measurements. Driven by high application potentials, the research in this area has intensified. In particular, non-intrusive load monitoring (NILM), that is, estimating appliance load consumption from aggregate readings, using software means only, has attracted a lot of attention, since it does not require any additional hardware to be installed. This thesis first proposes two Graph Signal Processing (GSP)-based approaches for disaggregation of total energy consumption down to individual appliances used. The first approach uses the Graph Laplacian Regularisation (GLR) minimiser results as a starting point, adding further refinement via Simulated Annealing (SA). The second approach applies data segmentation and associates data segments with graph nodes. A Dynamic Time Warping (DTW) distance is applied for evaluating weights between graph nodes. GLR minimiser is again used for clustering. Finally, a generic optimisation based approach is proposed for improving the accuracy of existing NILM by minimising the difference between the measured aggregate load and the sum of estimated individual loads with the difference from original

NILM approaches' results as regularisation. For all proposed methods, the competitive performance are demonstrated in terms of both accuracy and efficiency compared to state-of-the-art approaches, using the public Personalised Retrofit Decision Support Tools For UK Homes Using Smart Home Technology (REFIT) dataset and Reference Energy Disaggregation Dataset (REDD) electrical load datasets.

Contents

List of Figures

List of Tables

Glossary

$\mathbf{A}$ Graph weighted adjacency matrix.

a Cluster accuracy.

α Fraction parameter.

$\mathbf{c}$ A set of cluster labels.

CD Number of correctly detected ROIs.

$\mathbf{D}$ Graph degree matrix.

ϵ Minimum number of points required to form a cluster in DBSCAN.

η Distance threshold in I-DT.

$\mathcal{G}$ Graph.

$\mathbf{g}^m$ A graph signal for cluster m.

$\mathbf{g}^{\hat{m}}$ Graph Fourier Transformed $\mathbf{g}^m$.

ID Number of incorrectly detected ROIs.

$\mathbf{L}$ Graph Laplacian matrix.

$\mathbf{\Lambda}$ Eigenvalue matrix of graph laplacian matrix $\mathbf{L}$.

M Total number of clusters.

$\mathcal{M}$ Set of all known appliances in a house.

N Length of the vector signal $\mathbf{x}$.

ρ Number of neighbourhood samples of a point in Mean-shift.

$\mathbf{s}$ Graph signal.

σ A scaling factor of graph adjacency matrix $\mathbf{A}$.

σ_s Distance threshold in Mean-shift.

$\mathbf{s}^m$ Graph signal for appliance m.

$\mathbf{U}$ Eigenvector matrix of graph Laplacian matrix $\mathbf{L}$.

$\mathcal{V}$ Graph nodes.

ϱ Graph signal for GSP-based clustering.

$\mathbf{x}$ A vector dataset.

ζ Number of neighbourhood samples of a point in DBSCAN.

Acronyms

ANE Average Normalized Error.

FN False Negative.

FP False Positive.

F_M F-Measure.

PR Precision.

RE Recall.

TP True Positive.

AFAMAP Additive Factorial Approximate Maximum A Posteriori.

AFHMM Additive Factorial Hidden Markov Model.

ASP Algebraic Signal Processing.

BNB Branch and Bound.

DBSCAN Density-based Spatial Clustering of Applications with Noise.

DSP Discrete Signal Processing.

DT Decision Tree.

DTW Dynamic Time Warping.

FHMM Factorial Hidden Markov Model.

GBF Graph-based Bilateral Filter.

GFT Graph Fourier Transform.

GLR Graph Laplacian Regularisation.

GPCA Graph Principal components analysis.

GSP Graph Signal Processing.

GSP-P Graph Signal Processing based Post-processing.

GTV Graph Total Variation.

HMM Hidden Markov Model.

I-DT Distance-Threshold Identification.

ILM Intrusive load monitoring.

MR Manifold Regularisation.

NILM Non-intrusive load monitoring.

PCA Principal components analysis.

PSD Positive Semi-definite.

REDD Reference Energy Disaggregation Dataset.

REFIT Personalised Retrofit Decision Support Tools For UK Homes Using Smart Home Technology.

ROI Region of Interest.

SA Simulated Annealing.

SDR Semi-definite Programming Relaxation.

SVM Support Vector Machine.

Acknowledgement

I would like to first thank my supervisors Dr Vladimir Stankovic and Dr Lina Stankovic, for their long-term guidance and inspiration on my PhD work. Without their endless help and encouragement, I would never have this work done. I would also like to thank Dr Dusan Jakovetic, Dr Dragana Bajovic, Dr Dejan Vukobratovic, Dr Jing Liao and Dr Samuel Cheng, for their thoughtful advice and comments.

I would also like to thank Cheng Yang, Mingxiang Ye and Bochao Zhao, for their assistance given during my PhD work. It's my pleasure and honour to have the opportunity to work with them and learn from them.

This work might have never happened without the financial aid of the University of Strathclyde Faculty of Engineering Scholarship (2015-2018), and University of Strathclyde Department of Electronic and Electrical Engineering Scholarship (2015-2018). This work is also supported by the SENSIBLE project funded by the European Unions Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 734331.

I would especially like to give thanks to my friends and members of my band *The Bedtime*, for their long companionship and support.

Finally, I would like to thank my parents for all their help and support throughout this challenging work.

Chapter 1

Introduction

1.1 Smart Metering

Electromechanical electrical meters were widely used for the last century and measured only one value from each house, which is electric energy consumed. Nowadays, electricity meters (so-called smart meters) usually use electronic components and digital processing for measuring further readings, including voltage, current, active power and reactive power of each household in power distribution networks [1] and communicate the readings in (near) real-time to utilities. The wide implementation of smart metering systems introduces the possibility for new applications that exploit additional information from measured data to benefit both utilities and end-users [2]. Most of these new services are currently under development and are achieving remarkable growth since 2006 [3].

The key motivator of ongoing large-scale smart metering deployments worldwide is to maximise the benefits of the smart grid [4], [5]. Analysis of smart meter data has shown the ability to improve grid operation and maintenance. Main smart metering applications, including distributed generation and storage control [6], fault detection [7], non-technical loss detection [8], outage prediction [9], load forecasting [10], demand response [11] and improving customer satisfaction (including accurate billing and meaningful energy feedback) now play an important role in smart grid composition.

Another new type of smart metering applications that could be brought by real-time smart metering is related to appliance-level consumption analysis, as discussed in the next section.

1.2 Non-intrusive load monitoring (NILM)

Information analysis of individual residential loads is of major interest in smart home applications. Appliance level demand information significantly en-riches customer energy feedback and improves demand management measures via, for example, replacement programmes of inefficient appliances and appliance load shifting [12]. Indeed, it has been demonstrated that the appliance level energy feedback can lead to a significant reduction in domestic energy consumption waste [13].

According to the approach of monitoring appliance level consumption, corresponding methods can be divided into two categories: 1) Intrusive load monitoring (ILM) and 2) Non-intrusive load monitoring (NILM).

ILM is usually achieved by implementing and deploying a set of measurement devices, one for each appliance under interest. It is simple to measure and record the consumption of individual loads directly. However, the need for several measuring devices in the ILM ecosystem makes it expensive and hard to maintain, install and expand. The term intrusive means that the monitoring device is located in the habitation, close to the appliance being monitored and disturbing the daily life of the residents.

The above reasons lead to the introduction of NILM methods with much lower cost. NILM assumes the installation of a single measuring device at the panel level, which extracts aggregate loads measurements. The preference of using NILM techniques over ILM ones are mainly due to its easier and cheaper installation since it only uses one metering device for each energy entrance to a house instead of at least one metering device per socket.

The main motivations to study NILM are:

- Energy saving [14].
- Supporting retrofit appliance advice
- Demand response management [15].
- Smart home automation [16].
- Electricity theft detection [17].
- Power flow optimisation in micro-grids [18].
- Occupancy detection.
- Activity recognition [19], [20].
- Decision making for home-owners, utilities, appliance manufacturers and policy-makers.

NILM or energy disaggregation from the household aggregate measurements is a computational technique based purely on analytical tools for estimating the power demand of individual appliances.

Though NILM appeared in the 1980s [21], there has been a recent explosion in the NILM literature to tackle its practical challenges. The first NILM methods focused on formulating unique load signatures for appliances [22], [23]. The resulting methods are usually based on the assumption that every load has a distinct signature.

However, the NILM methods, as mentioned above, require high sampling frequency usually in the order of kHz or MHz. The lower rates measurements sampled between 1-60 seconds provided by smart metering [4], [5] are driving research into *low-rate* smart meter datasets [15] and low-rate NILM methods. Low-rate NILM is particularly challenging due to several reasons. Firstly, the accuracy of NILM algorithm is facing significant challenges because of various uncertainty factors, such as variable number, type, size, and using habits of appliances [24]. Noise from unknown appliances, abnormal transients and load fluctuations also make the low-rate NILM difficult. Thus, there is no universal model that can be used in all residences. In addition, no widely accepted load signature or model can be universally used for any appliance type. The power consumption of many appliances depends on their parameter settings which significantly vary.

The proposed low-rate NILM approaches in this thesis are motivated by the increasing availability of low-rate data from electrical smart meters that are being deployed at large scale, with an increasing penetration rate, in Europe, Australia and the USA. For example, in the UK [5] and the Netherlands [25], every household is planned to have access to 10-second active power readings. In the USA and Australia, smart meters providing readings at rates in the order of seconds and minutes, are massively deployed. Thus, NILM outputs can be accessible to the average household, without additional metering or monitoring hardware. This has prompted a recent trend in NILM literature tackling the NILM problem at low sampling rates, for example, [26], [27], [28], [29], [30], [31].

NILM methods can be divided into two groups: steady-state and transient-state methods. Steady-state NILM methods rely on features extracted under steady-state operation of appliances, e.g., changes in steady-state active power [26], [27], [28], [32], reactive power [21], voltage and current waveform [33], [34], steady-state current harmonics and total harmonic distortion [35], [36], or voltage-current trajectory [37]. At low sampling rates, steady-state features can be extracted reliably. However, due to the similarity of steady-state load signatures among many domestic appliances, the NILM problem is particularly challenging. Transient-state NILM methods identify appliances based on their transient signatures, including transient power [38], high frequency voltage noise [39], [40], harmonics of the transients [41], [42], [43], duration and shape of power/voltage/current transient waveform [44], [45]. Transient-state approaches provide more distinguishable features than steady-state approaches and hence, in general, lead to higher disaggregation accuracy. Transient methods require sampling rates in the order of kHz or MHz [46], unlike steady-state methods, which are more sensitive to power level fluctuations and at low sampling rates, require longer monitoring time to capture all operation cycles [47]. For a more detailed review of NILM, see review papers [44], [46].

Based on another classification criteria, low-rate NILM methods can be classified into two types: 1) state-based methods and 2) event-based methods.

State-based methods represent each appliance operation using a state machine with distinct state transitions, based on appliance usage patterns. Such probabilistic approaches are usually build on a Hidden Markov Model (HMM) and its variants (see [27], [29], [30], [31] [46] and references therein). Four state-based methods for low-rate NILM, using conditional Factorial Hidden Markov Model (FHMM) and Hidden semi-Markov graph models, are proposed in [29], but these methods have high computational complexity and are prone to converge to a local minimum. Another FHMM method is proposed by [27] for disaggregation of active power loads sampled at 1min, using expert knowledge to build initial models for states of known appliances, requiring correct setting a priori-values for each state for each appliance, which is in turn limited by or strongly dependent on the particular aggregate dataset on which NILM is being performed. The Hierarchical Dirichlet Process Hidden Semi-Markov Model factorial structure is used in [48], removing some limitations of the approach of [29], but at increased complexity. A sparse coding algorithm that discriminately trains sparse coding dictionaries is proposed in [49], to learn a probabilistic model for each appliance's load demand over a typical week. The HMM-based method of lower complexity, proposed in [50], reduces the execution time by 72.7 times, but still requires 94 minutes for disaggregating 11 appliances. The main drawback of state-based approaches is the need for expert knowledge to set a priori values for each appliance state via long periods of training and their high computational complexity, which makes them unsuitable for real-time applications [51].

Event-based NILM approaches have thus emerged [52], which are based on detecting events, usually via edge detection, when the load signal undergoes a statistically significant change indicating appliance use. After event detection, features (e.g., active power signature, increasing/falling edge [28], duration [53], uncorrelated power spectral components [26]) are extracted to classify the events into pre-defined categories, each corresponding to a known appliance. Different classification tools have been used, including Support Vector Machine (SVM) [54], neural networks [36], non-negative tensor factorization [32], k-means [53] and Decision Tree (DT) [28]. Challenges encountered by event detection tools include large measurement noise, including large variance of active power readings for common household appliances, and similarity among active power steady-state signatures of different appliances.

1.3 Datasets Used

This thesis, two publicly available datasets are used for evaluation. The first one is Reference Energy Disaggregation Dataset (REDD) that contains load data from 10 US houses [55] sampled at 1-sec resolution which can be downloaded at `http://redd.csail.mit.edu/`. The second dataset is Personalised Retrofit Decision Support Tools For UK Homes Using Smart Home Technology (REFIT) dataset [56], one of the largest UK datasets that contains active power measurements, sampled at 8-sec resolution, and collected over a continuous period of 2 years from 20 UK homes, available at `https://pureportal.strath.ac.uk/en/datasets/refit-electrical-load-measurements` and `https://pureportal.strath.ac.uk/en/datasets/refit-electrical-load-measurements-clean`

Each dataset is comprised of aggregate measurements and individual appliance measurements. REFIT has 9 appliance monitors per house and all other appliances are considered unknown. REDD has around 20 appliance monitors per house. Measurements of each appliance monitors are used as ground truth. The REDD dataset contains relative clean data and a small number of unknown appliances. On the other hand, the REFIT dataset contains many unknown appliances and high variations in the baseload.

1.4 Contribution of the Thesis

The main technical challenges of low-rate NILM are:

- Considerable amount of appliances inside each household.
- Various measurement noise including sensor noise, transient spikes and signal fluctuations.
- Similar operating load pattern for different appliances.
- Multiple appliances operating at the same time.
- Lack of recordings for short operating appliances.
- The infrequent use of certain appliances.

In addressing the above challenges and providing a reliable and accurate energy disaggregation results, the contributions of this thesis are: 1) We propose a Graph Signal Processing (GSP) based classification applied to a semi-supervised event-based NILM approach combined with the simulated annealing algorithm applied for further refinements. The proposed GSP-based NILM approach addresses the large training overhead and associated complexity of previous approaches. 2) We proposed a GSP-based clustering applied to an unsupervised state-based NILM approach for further improvements of the disaggregation accuracy. Dynamic Time Warping (DTW) distance is applied to generate a novel graph in the proposed approach. 3) We proposed novel post-processing approaches for improving the accuracy of existing NILM methods. This is posed as an optimisation problem to refine the final NILM result using regularisation, based on the level of confidence in the original NILM output. 4) We also provide a Graph Fourier Transform (GFT) based self parameter tuning method.

1.5 Organisation of the Thesis

The remainder of the thesis is organized as follows:

Chapter 2 firstly reviews the background, application and general definition about GSP. Then a case study about GSP applied for eye tracker data analysis is shown. We provide a GSP-based clustering algorithm with GFT included for self-parameter tuning. Experimental results are compared with benchmarks on public shape dataset and data recorded by eye tracker cameras.

Chapter 3 introduces the general background of event-based NILM and GSP, followed by a description of the proposed GSP and event based NILM algorithm. Then the performance of proposed methods is compared with state-of-the-art approaches.

Chapter 4 starts from the general review of state-based NILM. Then the proposed GSP and DTW applied to state-based NILM algorithm is introduced in the following order: 1) Pre-processing. 2) Signal segmentation. 3) GSP-based Clustering. 4) Labelling for clusters. Finally, the performance of proposed methods is compared with several benchmarking approaches.

Chapter 5 first introduces a novel modelling method of how one can systematically incorporate a first-pass NILM and post-processing into a common framework leading to a clear mathematical formulation. Then three approximate solutions to the formulated optimisation problem are proposed based on convex relaxation and convex optimization tools. In the next section, improvements of the proposed post-processing methods comparing with four state-of-the-art post-processing methods are demonstrated.

Chapter 6 concludes this thesis and justify the challenges in low-rate NILM that still remains.

1.6 Publications

Journal Articles

1. K. He, D. Jakovetic, B. Zhao, V. Stankovic, L. Stankovic and S. Cheng, “A generic optimisation-based approach for improving non-intrusive load monitoring,” *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6472-6480, Nov. 2019.

   Contribution: Experiments design, Algorithm implementation, Results analysis, Formal analysis, Draft writing.

2. K. He, L. Stankovic, J. Liao and V. Stankovic, “Non-intrusive load disaggregation using graph signal processing,” *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1739-1747, May. 2018.

Contribution: Experiments design, Algorithm improvement and implementation, Results analysis, Draft writing, Formal analysis.

3. B. Zhao, K. He, L. Stankovic and V. Stankovic, “Improving event-based non-intrusive load monitoring using graph signal processing,” *IEEE Access*, pp. 1-15, Sep. 2018.

   Contribution: Evaluation of DT-based NILM with proposed approach, Editing write-up.

4. D. Bajovic , K. He, L. Stankovic, D. Vukobratovi and V. Stankovic, “Optimal detection and error exponents for hidden semi-Markov models,” *IEEE Journal on Selected Topics in Signal Processing*, vol. 12, no. 5, pp. 1077-1092, Oct. 2018.

   Contribution: Algorithm implementation, Experiments design, Results analysis, Proofreading.

Conference Papers

1. K. He, D. Jakovetic, B. Zhao, V. Stankovic and L. Stankovic, “Post-processing for event-based non-intrusive load monitoring,” in *Proceeding of the 4th International Workshop on Non-Intrusive Load Monitoring*, Mar. 2018.

   Contribution: Experiments design, Algorithm implementation, Results analysis, Formal analysis, Draft writing.

2. K. He, C. Yang, V. Stankovic and L. Stankovic, “Graph-based clustering for identifying region of interest in eye tracker data analysis,” in *Proceeding of the 2017*

IEEE 19th International Workshop on Multimedia Signal Processing (MMSP), Oct. 2017.

Contribution: Experiments design, Data acquirement and analysis, Algorithm implementation, Results analysis, Formal analysis, Draft writing.

3. L. Stankovic, K. He, V. Stankovic, S. Lulic and S. Sladojevic, "Online accuracy estimation and improvement of event-based NILM algorithms without resorting to submetered individual loads," in *Proceeding of the EU NILM Workshop 2017*, Dec. 2017.

   Contribution: GFT application design, Experiments design, Results analysis, Conference presentation.

Submitted Paper

1. K. He, V. Stankovic, and L. Stankovic, "Non-Intrusive Load Disaggregation via Dynamic time warping based Graph Signal processing," *IEEE Transactions on Smart Grid*, Submitted, Apr. 2020.

   Contribution: Experiments design, Algorithm improvement and implementation, Results analysis, Formal analysis, Draft writing.

Chapter 2

Graph Signal Processing

2.1 Background

Graph Signal Processing (GSP) is a novel signal processing concept [57], [58] that effectively captures correlation among data samples in time and space by embedding the structure of signals onto a graph [58]. Typical graphs are used to represent common real-world data, including Erdos-Renyi graphs, ring graphs, random geometric graphs, small-world graphs, power-law graphs, nearest-neighbour graphs, scale-free graphs, and many others [59]. These graphs are connected by either random connections (ErdsRnyi graphs), brain neurons (small-world graphs), social networks (scale-free graphs), and many other ways.

GSP aims to extend the well-developed tools for Discrete Signal Processing (DSP) and Algebraic Signal Processing (ASP) theory of conventional signals to signals on graphs while exploiting the underlying connectivity information [57], [60]. Due to high practical potential, graph-based signal processing tools are widely researched, including shifting, filtering, classification, clustering, Fourier transform and interpolation of signals on graphs, etc. This leads to powerful, scalable and flexible approaches suitable for many data mining and signal processing problems, ranging from image denoising and data compression to classification, biomedical, and environmental data processing (see [57], [?], [61], [62], [63] and references therein).

Inspired by the initial success of GSP in many fields, e.g., compression of piecewise-smooth images [64], coding of point cloud attributes [65], coding in flexible representations [66], image bit-depth enhancement [67], and more applications in [57], [58], [60], [61], in this thesis, two GSP-based NILM approaches are proposed by extending [61] and [68] to perform low-complexity classification or clustering for the acquired active power readings. GSP is particularly suitable for data classification when training periods are short and insufficient to build appropriate class models [61].

The remaining of this chapter is organised as follows: Section 2.2 gives general definitions of GSP. Then Section 2.3 introduces thee regularisation on graph and it's application to classification/clustering. At last Section 2.4 introduces a case study that applies GSP on eye tracker data analysis.

2.2 General Definition

In this section, some basic concepts of GSP used in the remainder of the thesis are describe.

All matrices are denoted by upper-case bold letters, such as $\mathbf{X}$. $\mathbf{X}^T$ and $\mathbf{X}^\#$ are the transpose and pseudo-inverse matrices of $\mathbf{X}$, respectively. An element in the i-th row and j-th column of matrix $\mathbf{X}$ is denoted by $X_{i,j}$. $X_{i:j,a:b}$ represents a sub matrix consist of elements from i-th row to j-th row and a-th column to b-th column of matrix $\mathbf{X}$. Vectors are denoted by lower-case bold letters, such as $\mathbf{x}$, with the i-th element x_i, and $\mathbf{x}_{i:j}$ denotes a sub-vector $[x_i, x_{i+1}, \ldots, x_j]^T$, for $i < j$. A set is denoted using calligraphic bold-letters, such as $\mathcal{M}$, where $|\mathcal{M}|$ denotes its cardinality.

A dataset $\mathbf{x}$ of length N is represented by nodes of a graph $\mathcal{G} = (\mathcal{V},\mathbf{A})$, where $\mathcal{V}$ is the set of nodes and $\mathbf{A}$ is an $N \times N$ a weighted adjacency matrix of the graph that captures the correlation between graph nodes [57], [59]. Each element $x_i \in \mathbf{x}$ corresponds to a graph node $v_i \in \mathcal{V}$. The weight of the edge between nodes v_i and v_j reflects the similarity between x_i and x_j and is usually defined using a Euclidean distance based Gaussian kernel weighting function, which is one of the most used kernels in machine learning for expressing similarity between dataset elements [61]:

$$A_{i,j} = \exp\left\{-\tfrac{(x_i - x_j)^2}{\sigma^2}\right\}, \tag{2.1}$$

where σ is a scaling factor. Then, $\mathbf{s}$, often referred to as *graph signal*, is defined as a mapping from $\mathcal{V}$ to a set of complex numbers. For example, $\mathbf{s}$ can be a set of classification labels, where s_i is set to the label of the class that x_i belongs to.

An $N \times N$ *combinatorial graph Laplacian* matrix of graph $\mathcal{G}$ is defined as follows:

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \tag{2.2}$$

which is a real symmetric matrix and can be seen as a difference operator for the graph signal $\mathbf{s}$ [57]. In Eq. (2.2), $\mathbf{D}$ is an $N \times N$ diagonal matrix where for $k = 1, \ldots, N$,

$D_{k,k} = \sum_{j=1}^{N} A_{j,k}$. Eigenvalues of $\mathbf{L}$ carry the notion of frequency similar to classical Fourier analysis, where values of eigenvectors associated with low eigenvalues (low frequency) change less rapidly [57] across the nodes, which for example can be used to design graph signal filters.

There are two basic categories of approaches for signal processing on graphs. The first one uses the graph Laplacian matrix as its basic building block (see [57] and references therein). The second approach adopts the adjacency matrix of the underlying graph as its fundamental building block [58], [69], [70], [71]. Both frameworks define fundamental signal processing concepts on graphs, but the difference in their foundation leads to different definitions and techniques for signal analysis and processing. According to the two frameworks, Graph Laplacian Regularisation (GLR) and Graph Total Variation (GTV) minimisation, have been applied in the past in many applications, such as denoising, filtering, interpolation [57], [70], [72], [73]. Similar to the traditional DSP, GTV is defined as the sum of magnitudes of differences between two graph signal samples:

$$GTV(\mathbf{s}) = \frac{1}{||\mathbf{s}||_2^2} ||\mathbf{s} - \frac{1}{|\lambda_{max}|}\mathbf{A}\mathbf{s}||_2^2, \tag{2.3}$$

where λ_{max} is the largest-magnitude eigenvalue of $\mathbf{A}$ and $||.||_2^2$ is square of Euclidean norm.

For consistency of notation, the signal variation function GLR is defined based on the Laplacian matrix as [62]:

$$GLR(\mathbf{s}) = \mathbf{s}^\top \mathbf{L}\mathbf{s} = \sum_{i,j} (s_i - s_j)^2 A_{i,j}, \tag{2.4}$$

This formulation based on $\mathbf{L}$ can only be used for the undirected graphs where weights from node v_i to v_j and v_j to v_i are the same for $i = 1, ..., N$ and $j = 1, ..., N$, that is, $A_{i,j} = A_{j,i}$.

Similar to classical DSP, graph spectral analysis is often used to describe and analyse graph filtering. Graph spectral analysis can be done with both $\mathbf{L}$ and $\mathbf{A}$. For, since for undirected graphs, $\mathbf{L}$ and $\mathbf{A}$ are both diagonalizable matrix. However, eigenvalues of $\mathbf{L}$ carry information about connectivity of the graph and as such have been more popular in practice [57]. Another advantage of graph spectral analysis using the Laplacian matrix is the fact that eigenvalues are always non-negative where the smallest one $\lambda_0 = 0$ and $\lambda_0 <= \lambda_1 ... \leq \lambda_N$. We introduce spectral decomposition of a graph Laplacian $\mathbf{L}$ as the equation below:

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top, \tag{2.5}$$

where eigenvalues are listed on the diagonal of $\mathbf{\Lambda}$ and $\mathbf{U}$ is a complete set of eigenvectors. GFT of a graph signal $\mathbf{s}$ defined in [57] can be expressed as:

$$\hat{\mathbf{s}} = \mathbf{U}^\top\mathbf{s}. \tag{2.6}$$

In classical Fourier analysis, the Fourier basis functions carry a specific notion of frequency, for eigenvalues close to 0 (low frequency), the related eigenfunctions are smooth and oscillating less. In GFT, $\mathbf{\Lambda}$ and $\mathbf{U}$ provide a similar notion as classical Fourier analysis, which is eigenvectors related with the low eigenvalues vary across the graph slowly.

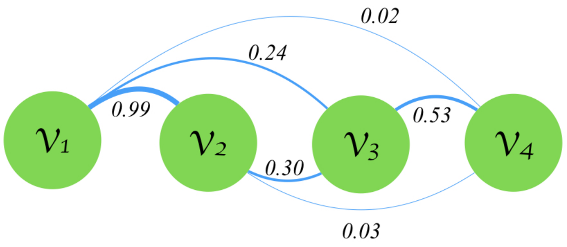


Figure 2.1: A GSP example with four nodes.

Example: Fig. 2.1 shows an example of a four-node graph constructed from $\mathbf{x} = [22, 21, 10, 2]$. With scaling factor σ setting to 10, the adjacency weight matrix of the example is calculated by Eq. (2.1) and shown as below:

$$\mathbf{A} = \begin{bmatrix} 1 & 0.99 & 0.24 & 0.02 \\ 0.99 & 1 & 0.30 & 0.03 \\ 0.24 & 0.30 & 1 & 0.53 \\ 0.02 & 0.03 & 0.53 & 1 \end{bmatrix}. \tag{2.7}$$

The example is a fully-connected undirected graph. So as shown above, $\mathbf{A}$ is a symmetric matrix with ones on its diagonal. We use lines instead of arrows with different thickness between each graph nodes namely edges to reflect the correlation between the graph nodes. The thickness of the line between v_i and v_j depends on the number entries in from adjacency matrix, that is, $A_{i,j} = A_{j,i}$.

Similarly, the graph Laplacian matrix is obtained by Eq. (2.2):

$$\mathbf{L} = \begin{bmatrix} 1.25 & -0.99 & -0.24 & -0.02 \\ -0.99 & 1.32 & -0.30 & -0.03 \\ -0.24 & -0.30 & 1.06 & -0.53 \\ -0.02 & -0.03 & -0.53 & 0.57 \end{bmatrix}. \tag{2.8}$$

Then the eigenvalues and the related eigenvectors of $\mathbf{L}$ are obtained by Eq. (2.5) and shown in Fig. 2.2 and 2.3.

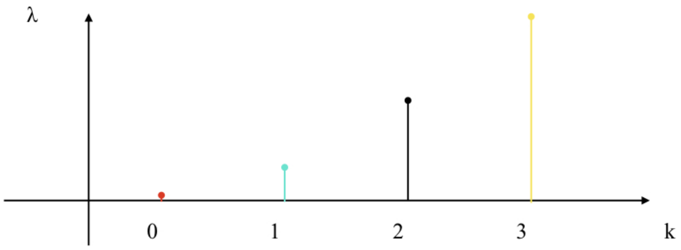


Figure 2.2: Eigenvalues of the graph Laplacian matrix $\mathbf{L}$ in GSP example.

Fig. 2.3 demonstrates that eigenvectors associated with larger eigenvalues oscillate more rapidly and are more likely to have dissimilar values on vertices connected by an edge with high weight. This oscillation can be related to the change in the sign of eigenvectors, that is $\mathbf{u_0}$: no change; $\mathbf{u_1}$: 1 change; $\mathbf{u_2}$: 2 changes; $\mathbf{u_3}$: 3 changes.

We define three graph signals: $\mathbf{s_a} = [1, 1, 0, 0]$, $\mathbf{s_b} = [1, 0, 0, 0]$ and $\mathbf{s_c} = [1, 0, 1, 0]$ which, for example, could be classification labels for identifying samples around 20 in

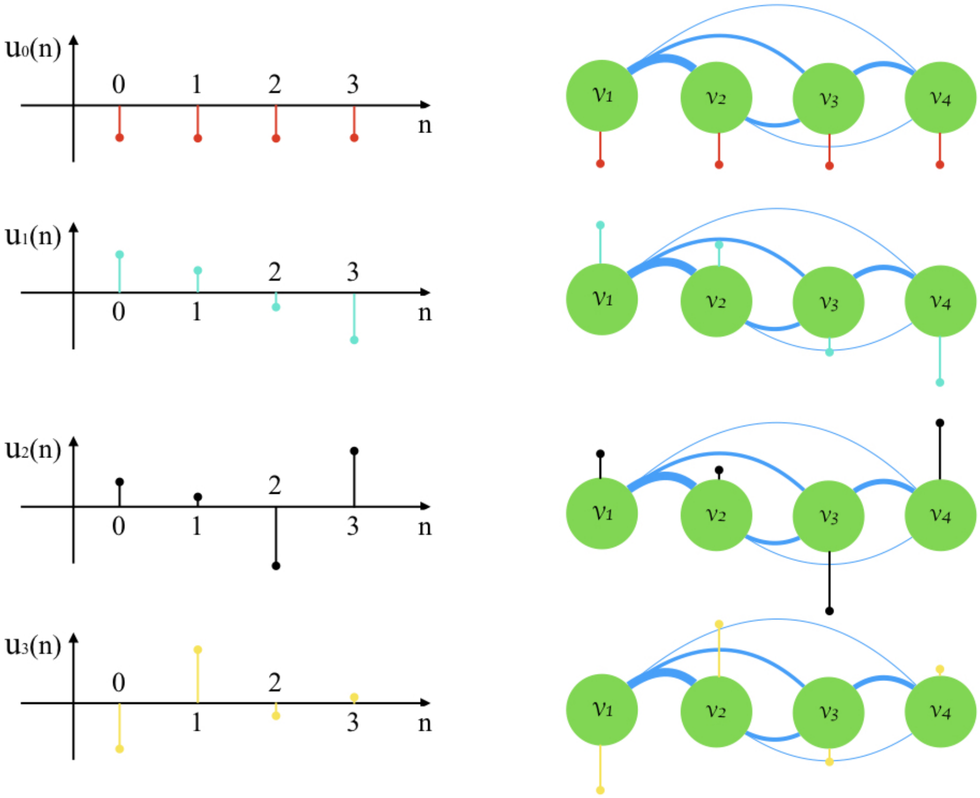


Figure 2.3: The distinct eigenvectors shown both on the vertex index axis, n, (left) and on the graph itself (right).

$\mathbf{x}$. GLR results obtained using Eq. (2.4) are $GLR(\mathbf{s_a}) = 0.58$, $GLR(\mathbf{s_b}) = 1.25$ and $GLR(\mathbf{s_c}) = 1.84$. The smallest GLR result for graph signal $\mathbf{s_a}$ indicates the smoothest graph signal and best classification labels.

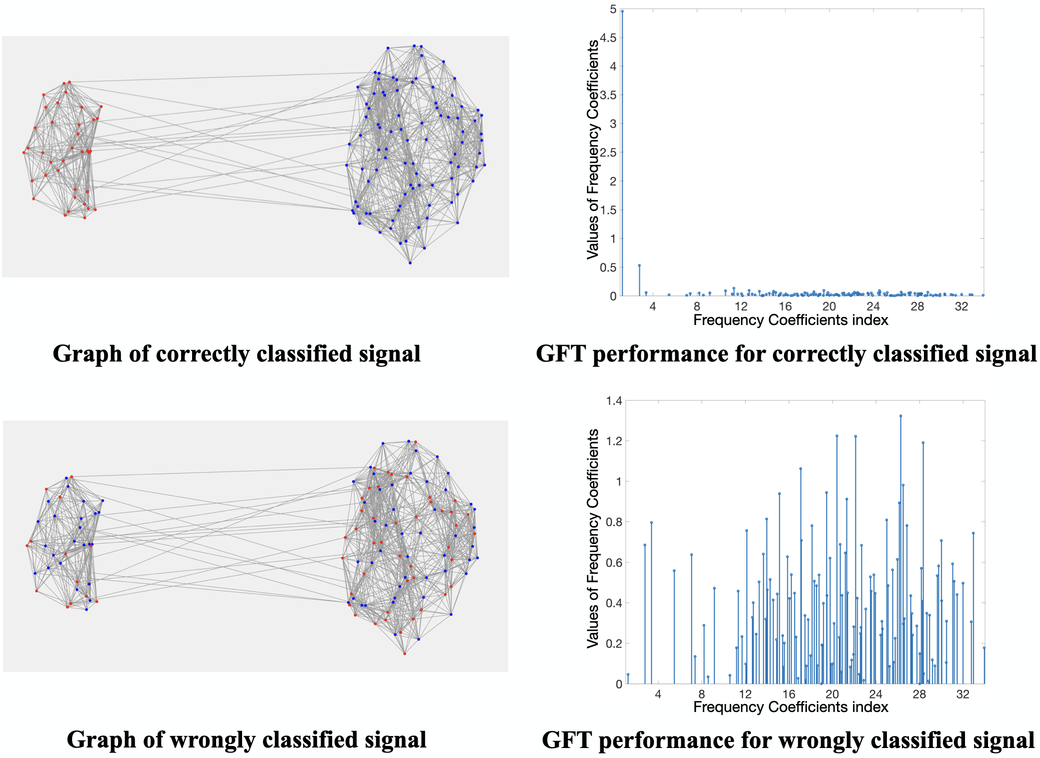
2.3 Classification and Clustering Based on Graph

Regularisation on graphs has emerged recently as a competitive model-based classification and clustering method. The approach is based on representing classification labels as a piece-wise smooth signal on graphs and then applying a graph signal smoothness prior, that is, either GTV or GLR.

For binary classification, $\mathbf{s} \in \{0, 1\}$ is introduced to define classification labels, where $s_i = 1$ means x_i belongs to the class, otherwise not. For $i < n$, where n is the training size, classification labels s_i are known and fixed. The remaining values of $\mathbf{s}$ are estimated by minimisation of GLR or GTV, that is Eq. (2.3) and (2.4). The main rationale behind the approach in [58] is that the graph nodes associated with elements within the same class are strongly correlated and will be connected via high-weight edges. Minimising GLR will assign graph signal s_i for $i > n$ to classification label 1 to make $\mathbf{s}$ vary smoothly across the connected nodes in the graph. This provides a powerful, scalable, and flexible data mining and signal processing approach, particularly suitable for data classification when training periods are short and insufficient to build appropriate class models [58].

Inspired by [71], GFT can be used to evaluate the quality of the classification results. As Fig. 2.4 displayed, if the given signal is correctly classified, the graph

Fourier transformed graph signal $\hat{\mathbf{s}}$ will have relatively high response associated with eigenvalues that reflect low frequency notions.



Graph of correctly classified signal

GFT performance for correctly classified signal

Graph of wrongly classified signal

GFT performance for wrongly classified signal

Figure 2.4: GFT performance comparison between correctly and wrongly classified signal. Red and blue dots refer to two different classes.

The next section will introduce a case study about a practical application of practical application, that is GSP-based clustering for identifying Region of Interest (ROI) in eye tracker data analysis. Inside the case study, GSP-based spatial clustering is achieved by GLR-based minimisation. GFT is also applied for self parameter tuning based on the idea above illustrated in Fig. 2.4.

2.4 Case Study: GLR-based Clustering Applied for Identifying Region of Interest in Eye Tracker Data Analysis

The previous sections introduced the background, including main definitions, on GSP and background on classification and clustering based on graphs. In this section, a practical application of GLR-based clustering approach for eye tracker data analysis is proposed as case study before GSP is applied to NILM. The remainder of this section is organised as following: Subsection 2.4.1 introduces the background of ROI detection in eye tracker data analysis, followed by Subsection 2.4.2 that describes the proposed method. Subsection 2.4.3 demonstrates the experimental results. The last subsection summaries the work of this case study. The material is published in [74].

2.4.1 Introduction

When an image or scene is viewed, the eye gaze tends to pause on small regions within the image, called fixation areas. On average, fixations last for around 200 ms during the reading of the linguistic text, and 350 ms during the viewing of a scene [75]. Existing approaches for detecting ROI in the viewed image first represent the centre of a fixation area as a fixation point [76], and then use clustering to group these fixation points from all fixation areas into spatial regions, identified as ROI. Various clustering approaches have been used to detect ROI, such as k-means and distance threshold [77], [78], Density-based Spatial Clustering of Applications with Noise (DBSCAN) [79], Distance-Threshold Identification (I-DT) [80] and Mean-shift [81]. The gaze data, acquired by commercial eye trackers, is usually affected by the high level of measurement noise and contains missing data due to eye blinks and occasional head movements.

To identify ROI from noisy eye tracker data, in this section, the eye tracker measurements are firstly pre-processed, that is comprising the time-stamped eye gaze locations in the viewed image, by filtering the data to ensure convergence to locations of higher density, similarly to [81], and then a data sample are cleverly chosen as a starting point for clustering. Then a GSP-based iterative clustering method is proposed, for spatial clustering of pre-processed eye tracker data to detect ROI. Clustering is performed on the graph, where each graph vertex is associated to one spatial gaze measurement, that defines horizontal and vertical position of the gaze, and weights of the edges reflect the spatial correlation between the measurements.

Note that, we focus on the detection of ROI in still images, where an ROI is a group of gaze measurement spatially concentrated regardless of the time information [81]. Having this in mind, and due to GSP's resilience to noise, the traditional step of first finding time-dependent fixation points prior to ROI detection is bypassed, making the proposed method robust to timing jitter and synchronization problems. Moreover, the method is inherently robust to measurement noise and eye blinks, and no denoising or data cleaning steps, common for eye tracker data processing, are needed.

2.4.2 Proposed Method

System Overview

Let N be the total number of samples from eye tracker gaze data, and $(\mathbf{x}, \mathbf{y}) = \{(x_1, y_1), ..., (x_N, y_N)\}$ be the spatial locations of the samples in the viewed image. The objective is to group samples into clusters $m \in \{1, \ldots, M\}$, where M is the total number of clusters which is unknown. Let the clustering label be $\mathbf{c} = \{c_1, \ldots, c_N\}$ where $c_i = m$ if sample (x_i, y_i) belongs to cluster m.

Fig. 2.5 shows the block diagram of the proposed GSP-based clustering method. First, similarly to [81], preprocessing is applied by shifting the input eye gaze data to make all samples move simultaneously towards locations of higher density. Then, one sample with the high density centre is chosen as the starting point to clustering. Next, binary GSP-based classification is performed, similarly to [62] and [82], to classify all data samples into one of two classes: belonging to the same class as the starting point or not. All samples classified to the class of the starting point starting from Cluster m =1, and are removed from the dataset, a new starting point is chosen, m is incremented, and the process is repeated until no samples remain unclustered.

The value of a (define later in subsection Self Parameter Tuning), is then used to evaluate the *quality* of the formed clusters. If the accuracy improved compared to the previous iteration, then the underlying graph used for GSP processing is adjusted, the clustering labels are reset, and the process is repeated until accuracy cannot be further improved.

The pseudocode of the proposed method is shown in Algorithm 1. The following

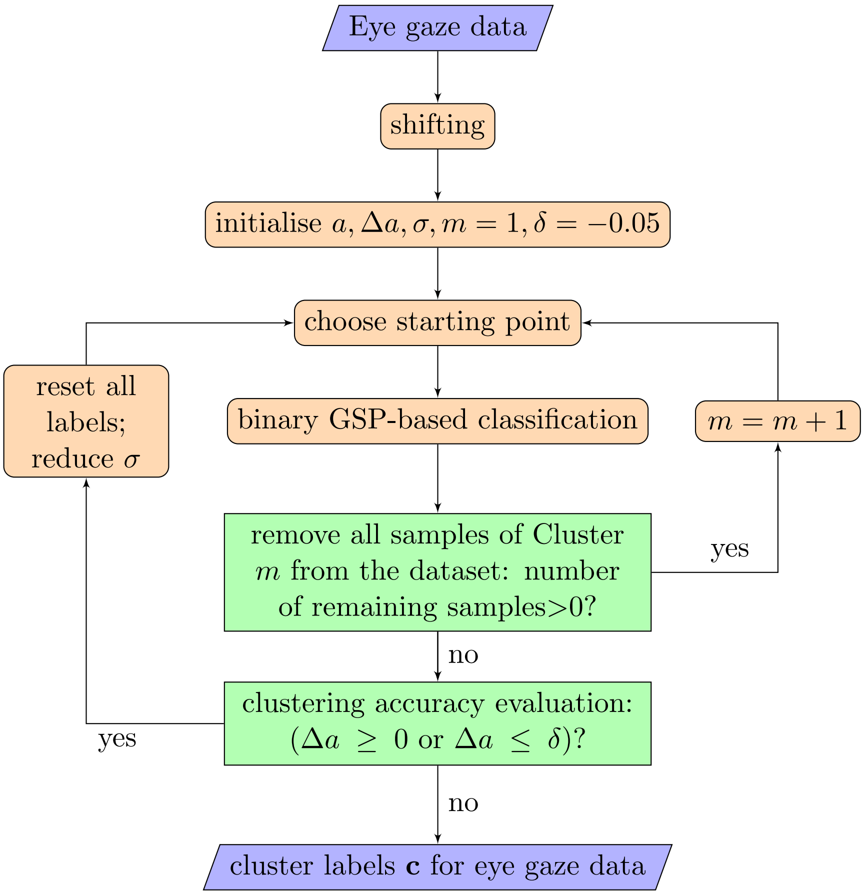

Eye gaze data
shifting
initialise $a, \Delta a, \sigma, m = 1, \delta = -0.05$
choose starting point
reset all labels; reduce σ
binary GSP-based classification
$m = m + 1$
remove all samples of Cluster m from the dataset: number of remaining samples>0?
yes
no
clustering accuracy evaluation: $(\Delta a \geq 0$ or $\Delta a \leq \delta)$?
yes
no
cluster labels **c** for eye gaze data

Figure 2.5: The proposed GSP-based clustering algorithm.

subsections provide a detailed description of each step.

Algorithm 1: Proposed GLR-based spatial clustering.

```
Input: $(\mathbf{x}, \mathbf{y})$;
Output: $\mathbf{c}$;
set $x_i^*$ via (2.9), $y_i^*$ via (2.10), $a = 0$, $\Delta a = 0$, $\sigma = 20$, $\beta = 0.99$, $\tau = 5$, $b = 10$, $\delta = -0.05$, $\alpha = 0.03$ ;
while $\Delta a \geq 0$ or $\Delta a \leq \delta$ do
    set $\sigma = \sigma - 1$, $m = 1$, $\mathbf{c} = 0$;
    while number of remaining samples $> 0$ do
        randomly set $(x_s, y_s)$;
        compute $(\hat{x}_s, \hat{y}_s)$ ;
        compute $\mathbf{A}$, $\mathbf{L}$ with $(\hat{x}_s, \hat{y}_s)$, $(\mathbf{x}^*, \mathbf{y}^*)$, (2.1), (2.2);
        compute $\mathbf{s}^{m*}$ via (2.13);
        set $\mathbf{c}$(find $\mathbf{s}_i^{m*} > \beta) = m$, $m = m + 1$;
        remove from the dataset samples $i$ with $\mathbf{s}_i^{m*} > \beta$;
    compute $\Delta a$ via (2.15), (2.16), (2.17);
return $\mathbf{c}$;
```

Shifting Samples

Before clustering, a preprocessing operation, *shifting*, is performed to make clustering robust to outliers, e.g., saccade points. This shifting step aims to move all samples to higher density locations, making samples in each cluster more compact. Let $(\mathbf{x}^*, \mathbf{y}^*) = \{(x_1^*, y_1^*), ..., (x_N^*, y_N^*)\}$ denote the shifted data. For each sample (x_i, y_i), (x_i^*, y_i^*) is defined as:

$$x_i^* = \frac{\sum_{n=1}^{b} x_n}{b}, \tag{2.9}$$

$$y_i^* = \frac{\sum_{n=1}^{b} y_n}{b}, \tag{2.10}$$

where b is the number of most correlated neighbours to the vertex v_i. Fig. 2.6 shows an example of the raw data samples from Shape Dataset and pre-processed, shifted samples. It is clear that the shifted points become more concentrated in each cluster.

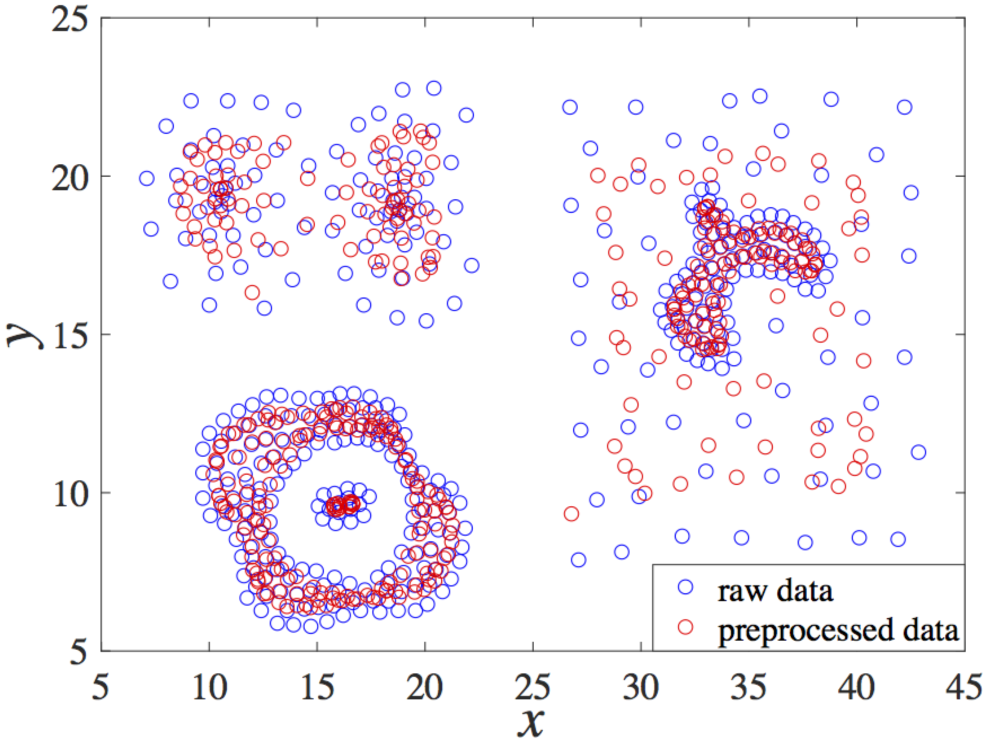


Figure 2.6: Comparison between shifted, preprocessed data and raw data.

Setting the Starting Point

Many ROI detection methods are based on randomly choosing the starting point (see [81] and references therein); hence bad initial positions (e.g., saccade points) will rapidly reduce the clustering accuracy. We thus propose a method to refine the initial, random selection of the starting point. Let (x_s, y_s) be a randomly chosen starting point. The shifted starting point x_s^* and y_s^* is updated using Eq. (2.9) and (2.10) until the difference between the points before and after shifting cannot be reduced any more. The refined starting point $(\hat{x}_s, \hat{y}_s)$ is the nearest sample to the final (x_s^*, y_s^*). Note that this way we ensure high point density around the selected starting point, hence the starting point is unlikely to be an outlier.

GLR-based Clustering

We use binary GLR-based classification to find samples that belong to the same class as $(\hat{x}_s, \hat{y}_s)$. We do this by first constructing a graph $\mathcal{G}$= ($\mathcal{V}$, $\mathbf{A}$), where for $i = 2, \dots, N+1$ each vertex $v_i \in \mathcal{V}$ is associated with one data sample (x_i^*, y_i^*), v_1 is associated with $(\hat{x}_s, \hat{y}_s)$, and $\mathbf{A}$ is the weighted adjacency matrix of $\mathcal{G}$ defined similarly as Eq. (2.1) using Euclidean distance with a Gaussian kernel:

$$A_{i,j} = \exp\left\{ - \frac{(x_i^* - x_j^*)^2 + (y_i^* - y_j^*)^2}{\sigma^2} \right\}, \tag{2.11}$$

where σ is a scaling factor. Next, a *graph Laplacian* is calculated using Eq. (2.2).

Starting from $m = 1$, let $\hat{x}_s$ and $\hat{y}_s$ belong to Cluster m and classify the remaining N samples as belonging to Cluster m or not. In particular, an $(N+1)$-length vector ϱ is initialized as graph signal: $\varrho = [\varsigma \;\; \mathbf{s}^m]^\top$, where $\varsigma = 1$ is associated with starting point $(\hat{x_s}, \hat{y_s})$, $\mathbf{s}^m$ is an N-length row vector initialised as all zeros.

We adopt GLR as introduced in the previous section $\varrho^\top \mathbf{L} \varrho$ to measure the variation in signal $\mathbf{s}^m$ with respect to the underlying graph, with the objective to find an $\mathbf{s}^{m*}$ that minimizes the variation in the graph signal. The optimization problem

$$\mathbf{s}^{m*} = \arg\min_{\mathbf{s}^m} ||\varrho^\top \mathbf{L} \varrho||_2^2, \tag{2.12}$$

has the following closed-form solution [83], [84]:

$$\mathbf{s}^{m*} = -\mathbf{L}^{\#}_{2:N+1,2:N+1} \varsigma \mathbf{L}^\top_{1,2:N+1}, \tag{2.13}$$

where $\mathbf{L}^{\#}_{2:N+1,2:N+1}$ is the pseudo-inverse of $\mathbf{L}_{2:N+1,2:N+1}$, and $s^{m*} \in [0,1]$. If s_i^{m*} is close to 1 (based on a heuristically set distance threshold), then (x_i^*, y_i^*) is designated to the same cluster m as $(\hat{x}_s, \hat{y}_s)$, which is labeled as $c_i = m$.

Next, all clustered points are removed, increment m and repeat the procedure, starting with selecting a new starting point until all samples are labelled with a cluster number. Finally, the clusters that contain small fractions of samples, where the fraction parameter is denoted as α, are assumed to be outliers and all grouped to Cluster 0. The cluster with coordinate $(0,0)$ is also labelled as 0 since it contains all lost data caused by eye blinks and noise.

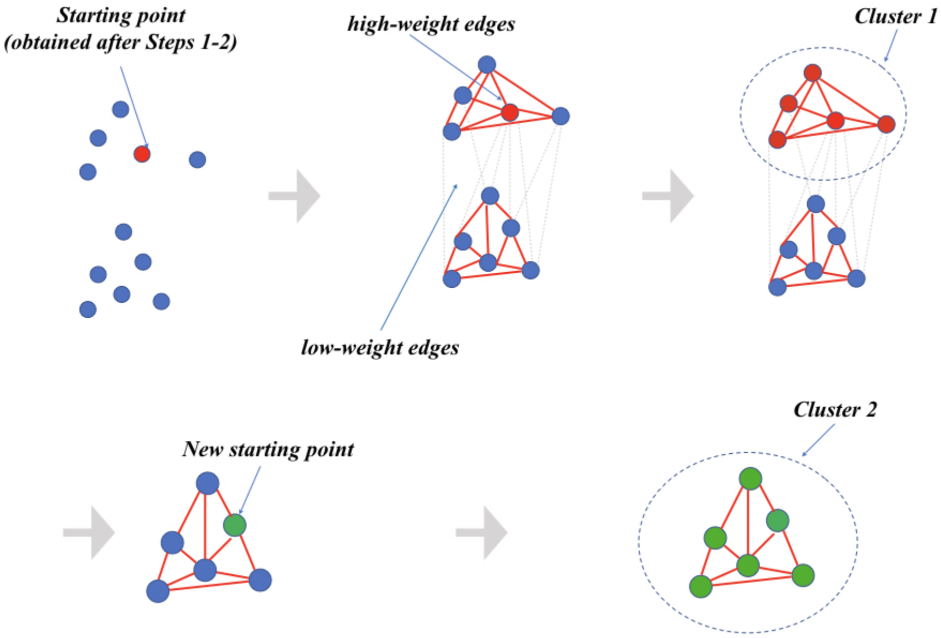


Figure 2.7: Example of GLR-based clustering for spatial point data.

Fig. 2.7 shows an example of how GLR-based clustering is applied for spatial point data. As the example displayed, graph nodes belong to the same cluster as the starting point are connected with high-weight edges. After all these nodes are assigned to a cluster m, they are removed together with the starting point. We increase m with $m = m + 1$ and repeat the above steps to find all clusters.

Self Parameter Tuning

Initial testing shows large dependency of the accuracy of the results on the scaling factor σ defined in (2.1) that weights the relationship between the data samples. Large σ leads to large $A_{i,j}$ indicating high correlation between the samples i and j, which would result in many sample points being clustered together. Low σ has the opposite effect: small clusters would be formed comprising only highly correlated samples.

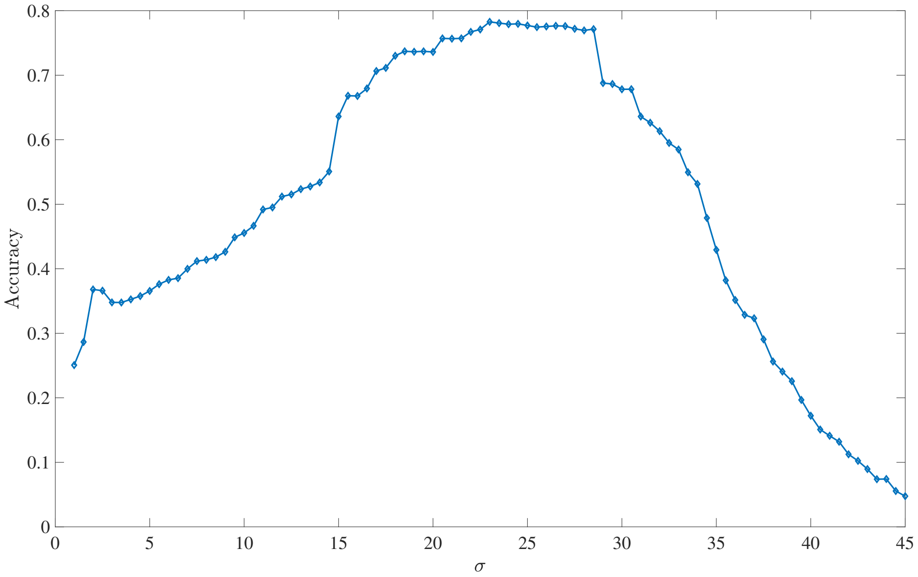


Figure 2.8: Example of accuracy varying in dependency of σ.

As Fig. 2.8 demonstrates, when σ is too large, weights between graph nodes can easily be too high. Clustering accuracy becomes low for many samples with relatively low similarities being clustered together. In the opposite, smaller σ will dramatically decrease the weights. Clustering accuracy again will be low because only samples with very high similarities will be regarded as the same cluster.

Since the best σ, that is, the one that maximises accuracy is signal-dependent, a method for finding the optimal σ is proposed based on the signal samples. First, σ is set to be a very high value, which leads to rough clustering (i.e., a small number of large clusters) for the giving dataset. After all samples are labelled as the procedure from the previous subsection, a graph signal $\mathbf{g}^m$ for Cluster m is defined as:

$$g_i^m = \mathbb{1}(c_i == m), \tag{2.14}$$

where $\mathbb{1}$ is an indicator function that returns 1 if the condition is true and 0 otherwise.

We then recalculate a graph Laplacian without the starting point using Eq. (2.2), then apply the spectral decomposition. The graph Fourier transformed signal $\mathbf{g}^{\hat{m}}$ can be obtained using Eq. (2.6).

As discussed in the Section 2.2, the eigenvalues of a graph Laplacian act as the graph frequencies and corresponding eigenvectors act as the graph harmonics [69], [71], [85]. Small λ's carry information about low frequency components of the signal, while high frequencies (details) are carried by large λ's. Motivated by the fact that high energy in the high frequencies indicates bad cluster quality, the frequency content of $\hat{g}_i^m$ is estimated as follows. Let $f = (\lambda_0 + \lambda_{N-1})/2$ and $j^* = \arg\min_j |f - \lambda_j|$, then λ_i for $i \leq j^*$ would carry information about the energy content in the lower half of the frequency spectrum.

Let r_m be the ratio of the total number of low/high frequency components in $\mathbf{g}^{\hat{m}}$ that are above/below a heuristically set threshold γ *i.e.*,

$$r_m = \frac{\sum_{i=1}^{j^*} \big(\mathbb{1}(|\hat{g}_i^m| > \gamma)\big)}{\sum_{i=j^*+1}^{N} \big(\mathbb{1}(|\hat{g}_i^m| > \gamma)\big)}. \tag{2.15}$$

$r_m > \tau$ indicates a good cluster, where τ is a chosen parameter. If not, all samples in this cluster are considered as incorrectly clustered. In addition, the samples with cluster label equal to 0 are also counted as incorrect samples. The estimated clustering accuracy a is calculated as:

$$a = 1 - \frac{\kappa}{N}, \tag{2.16}$$

where κ is the total number of incorrectly clustered samples that is given by:

$$\begin{aligned} \kappa = & \sum_{i=1}^{N} \big(\mathbb{1}(x_i^* = 0 \;\&\; y_i^* = 0)\big) \\ & + \sum_{i=1}^{N} \sum_{m=1}^{M} \Big(\mathbb{1}\big((c_i = m) \;\&\; (\frac{\sum_{i=1}^{N} \big(\mathbb{1}(c_i = m)\big)}{N} \leq \alpha)\big)\Big) \\ & + \sum_{i=1}^{N} \sum_{m=1}^{M} \Big(\mathbb{1}\big((c_i = m) \;\&\; (r_m \leq \tau)\big)\Big), \end{aligned} \tag{2.17}$$

where the first line captures all samples in Cluster 0, the second line includes clusters that have a very low number of samples (below α), and the third line comes from all clusters that give $r_m < \tau$, where τ is a heuristically found clustering quality threshold. The scaling factor σ is reduced by small decrements until there is no improvement in a any-more.

2.4.3 Experimental Results

In this section, the proposed spatial clustering algorithm is first validated on a public clustering dataset with known ground-truth labels, and show how the proposed clustering algorithm compares with DBSCAN, I-DT and Mean-shift algorithms. Then the results with true eye tracker data are presented to compare the accuracies of the four aforementioned methods in detecting ROI.

Parameter Setting

The parameters used in this section are set as shown in Table 2.1.

Table 2.1: Parameter settings for the proposed method used in all the experiments.

symbol	parameter	setting
a	estimated accuracy	initially set 0
b	neighbouring samples	10
τ	cluster quality	5
σ	scaling factor for $\mathbf{A}$	initially set 20
α	decreasing scale	0.03
β	labelling threshold	0.99
δ	accuracy difference threshold	-0.05
γ	frequency response	$\frac{1}{2}\max\{\lvert\hat{g_i^m}\rvert\}$

Estimated accuracy without ground truth, a, is initially set to 0 and updated by Eq. (2.16). b is the number of most correlated neighbours to a certain vertex v_i, i.e., we take the b highest-weight connections to vertex v_i which is set as hyper-parameter, heuristically. τ is a heuristically found clustering quality threshold. α is the decreasing scale of self tuned parameter σ. β is the threshold to decide the labelling results and is set to be 0.99 to ensure only the candidates that are correlated very well are included in a cluster. δ is the accuracy difference threshold which is heuristically set to be -0.05

to avoid abnormal disturbing in self parameter tuning. The proposed method is not sensitive to the parameters provided in Table 2.1 except labelling threshold β. Smaller β, for example 0.9, will include candidates that are not belong to the cluster. This will increase FPs and hence influence the final F-measure accuracy. β set to be over 0.99 but smaller than 1 will provide the same results.

Results with Shape Dataset

The proposed algorithm is first tested on the public *Shape dataset* [86], which is often used to assess the accuracy of spatial clustering methods. The images in this dataset are scatter diagrams with labels, indicating the cluster index for each point, where the points close to each other are assigned to the same cluster. The shape dataset is demonstrated as Fig. 2.9 bellow:

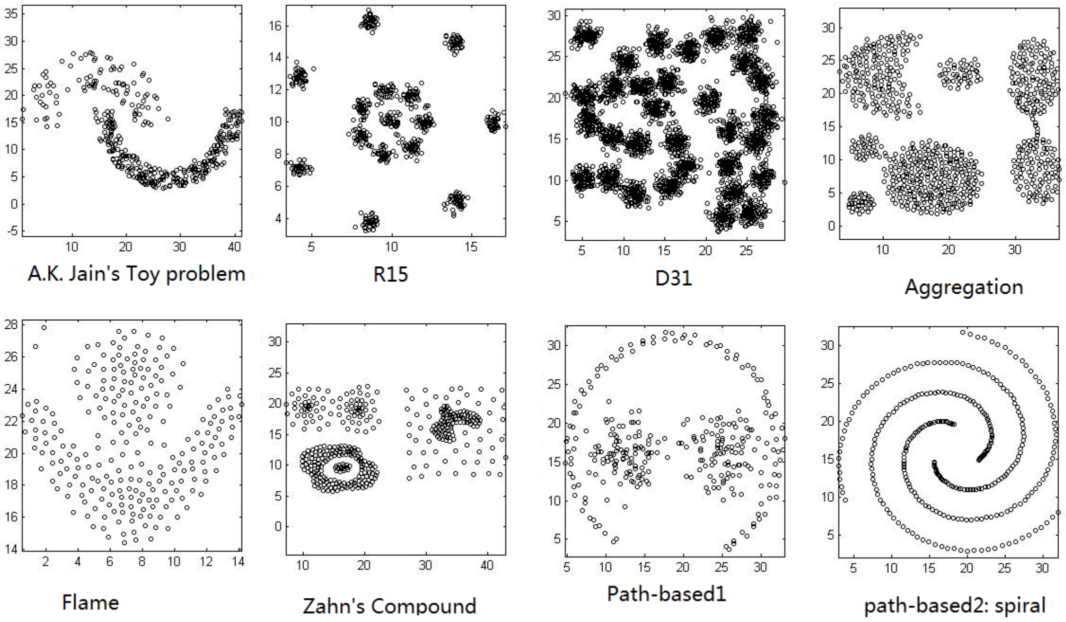


Figure 2.9: Shape data set.

Table 2.2: Clustering accuracy of the proposed method, DBSCAN, I-DT and Mean-shift without preprocessing.

	Proposed	DBSCAN	I-DT	Mean-shift
parameter	Self-adaptable σ	$\epsilon = 2$ $\zeta = 14$	$\eta = 5$	$\sigma_s = 7$ $\rho = 10$
R15	0.89	0.53	0.13	0.20
D31	0.76	0.06	0.30	0.35
Aggregation	0.79	0.88	0.87	0.89
Toy	0.30	0.91	0.46	0.48
Compound	0.84	0.90	0.69	0.78
Pathbased	0.80	0.85	0.67	0.66
Flame	0.95	0.35	0.95	0.91
Mean	**0.76**	**0.64**	**0.58**	**0.61**

Table 2.2 shows the clustering results of the proposed method on the Shape dataset comparing with three benchmarking methods. The accuracy is measured as a ratio of the number of correctly clustered samples to the total number of samples. In DBSCAN, the minimum number of points required to form a cluster and the number of neighbourhood samples of a point are denoted as ϵ and ζ, respectively. The distance threshold in I-DT is denoted as η. The distance threshold and the number of neighbourhood samples of a point in Mean-shift are denoted as σ_s and ρ, respectively. Parameters in DBSCAN, I-DT and Mean-shift are tuned and fixed using a greedy search scheme to get the best performance for the whole dataset. The proposed approach can self-tune itself to find the best parameters for each image pattern.

The performance of DBSCAN on some images, such as *Compound*, is good. There are many individual points that are isolated in *Compound* dataset. They are all clustered as a single cluster in ground truth which DBSCAN consider them as noise. However, since DBSCAN is a density-based spatial clustering, it cannot adapt well to different point density characteristics of the images, leading to poor performance in some cases. I-DT is a distance-based method, where results are highly influenced by the size of clusters in the image. In Mean-shift, all points are repetitively moved until converged to positions with high density [81]. Then, a distance threshold is applied to cluster the shifted points, while the size of the clusters is depended on σ_s. Thus, the overall performance is poor due to variations in cluster sizes across the images.

For some images, the ground-truth clusters are connected with consecutive points. In GLR-based clustering, these clusters will be treated as piecewise smooth signals since the weight is defined based on the distance between samples, and thus these clusters are incorrectly merged into the same cluster. Shift pre-processing can move these connecting points closer to their closest high density centres, disconnecting in this way the clusters, and leading to more effective clustering.

Table 2.3 shows the results of our proposed GLR-based clustering method compared with DBSCAN, I-DT and Mean-Shift after shift preprocessing is applied on the images prior to running all 4 clustering algorithms. We again use a greedy search scheme to get the optimal parameters for all competing schemes. Overall performance for all methods except Mean-shift has improved compared to clustering the raw data without pre-processing. This proves that the shift preprocessing can significantly improve the clustering accuracy. For Mean-shift, the preprocessing does not improve the performance, since the effect of the proposed shift preprocessing is similar to the operation that is already done in the Mean-shift. Indeed, Mean-shift uses the weighted mean of nearby points based on the kernel function to make the samples compact.

Table 2.3: Clustering accuracy of the proposed method, DBSCAN, I-DT and Mean-shift with preprocessing.

	Proposed	DBSCAN	I-DT	Mean-shift
parameter	Self-adaptable σ	$\epsilon = 1$ $\zeta = 5$	$\eta = 5$	$\sigma_s = 7$ $\rho = 10$
R15	0.99	0.53	0.52	0.20
D31	0.93	0.23	0.48	0.35
Aggregation	0.95	0.97	0.64	0.90
Toy	0.93	0.90	0.33	0.47
Compound	0.73	0.86	0.92	0.79
Pathbased	0.77	0.73	0.63	0.68
Flame	0.98	0.93	0.97	0.91
Mean	**0.90**	**0.74**	**0.64**	**0.61**

Results on the Eye Tracker Dataset

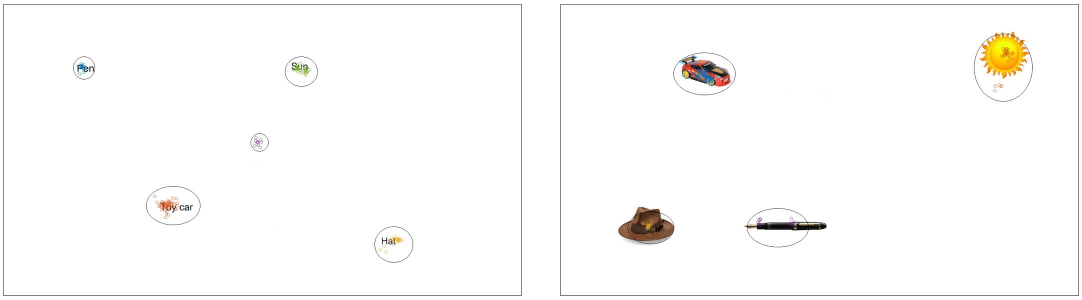
The algorithms are also tested on true eye tracker data recorded by Eye Tribe [87] at a sampling rate of 30Hz, to assess the accuracy of the methods in detecting ROI in the viewed images. Experiments are performed in a laboratory with moderate artificial light conditions, which remain unchanged for the duration of all trials. Ten subjects participated in the experiments, aged between 25 and 45 years old, both male and female, all with normal vision. The subjects were sitting in front of a DELL P2414 screen with a resolution of 1920x1080 pixels, at about 70 cm distance from the eye tracker, which is located under the screen. Calibration was performed using OGAMA [88] calibration process, whereby subjects are asked to follow a coloured dot moving in the corners and centre of the screen. This calibration process was included before each trial. Note that OGAMA is an open source software for recording and analysing eye gaze and mouse data for experiments with screen-based slide show stimuli. OGAMA does not generate ROI information.

Two different experiments are set. In Experiment 1, four objects are displayed on a blank white-coloured background and shown to viewers for 5 seconds. Two slides are shown:

1. A white/Blank background image with four words sparsely distributed.

2. A white/Blank background image with four small coloured icons spread out across the slide.

The viewers are asked to focus their attention on the four objects, one at the time. Hence, the clustering algorithms should result in four distinct clusters each pointing to one object. Examples of the ROI identification with the proposed approach overlapped with the displayed image is shown in Fig. 2.10. The ellipses are drawn to emphasise all samples of a cluster to make the visual clustering results clearer.



(a) Slide1.

(b) Slide2.

Figure 2.10: ROI estimation validation in Experiment 1 for Subject 2 with (a) words distributed, (b) icons distributed (Enlarge slightly in colour).

The comparison results between the four methods are shown in Table 2.4, where CD is the number of correctly detected ROIs and ID stands for the number of incorrectly detected ROIs. Both CD and ID are averaged over all subjects. $CD = 20$ if all ROIs are detected correctly. An ROI is correctly detected if at least half of the samples in the resulting cluster overlap with the target object, and there are no other clusters that overlap with the target.

One can see from the table that the proposed method leads to the highest CD and lowest ID indicating the highest accuracy. Generally, all methods perform well, since the objects are clear, the background is white, and the objects are far away from one another. In order to test the ROI detection accuracy in a more challenging scene, we set Experiment 2.

Table 2.4: Comparison of the ROI detection results between the proposed method, DBSCAN, I-DT and Mean-shift, for Experiment 1. CD is the number of correctly detected ROIs and ID is the number of incorrectly detected ROIs.

	Proposed		DBSCAN		I-DT		Mean-shift	
	CD	ID	CD	ID	CD	ID	CD	ID
Slide1	18	3	14	6	14	7	15	8
Slide2	19	2	15	5	16	7	13	6

In Experiment 2, 10 slides are shown to the subjects, all full of icons (around 70) [89]:

1. Slide1. Blank background; 2 sec on each 4 target icons.

2. Slide2. Blank background; 5 sec on each 4 target icons.

3. Slide3. Blank background and the target icons are very small (compared to other icons in the image).

4. Slide4. Blank background and the target icons are large.

5. Slide5. Blank background; and the whole slide is noisy.

6. Slide6. Nature image as background; 2 sec on each 4 target icons.

7. Slide7. Nature image as background; 5 sec on each 4 target icons.

8. Slide8. Nature image as background and the target icons are very small.

9. Slide9. Nature image as background and the target icons are very large.

10. Slide10. Nature image as background and the whole slide is noisy.

The subjects are informed about the positions of the target icons in the slides before the experiment. During the experiment, the subjects are told to focus their attention on those icons. The ROI will be the target icons that the subjects are asked to focus on. The saccades while finding the target icons are noise. The numerical comparison results are shown in Table 2.5. Figs. 2.11 and 2.12 show two examples obtained with the proposed method.

Table 2.5: Comparison of ROI detection results between the proposed method, DBSCAN, I-DT and Mean-shift, for Experiment 2. The results are averaged over all subjects.

	Proposed		DBSCAN		I-DT		Mean-shift	
	CD	ID	CD	ID	CD	ID	CD	ID
Slide1	28	1	24	12	21	5	25	5
Slide2	29	2	25	10	22	10	22	8
Slide3	26	1	20	12	18	19	20	7
Slide4	27	4	21	9	17	21	15	25
Slide5	29	2	20	5	19	6	21	3
Slide6	27	3	18	11	22	10	19	6
Slide7	30	1	20	13	21	9	19	7
Slide8	28	2	15	11	19	11	17	9
Slide9	28	3	14	10	21	7	11	23
Slide10	26	2	22	7	21	9	21	9
Mean	**27.8**	**2.1**	**19.9**	**10**	**20.1**	**10.7**	**19**	**10.2**

If all ROIs are correctly detected, and no redundant ROIs are found, $CD = 30$ and $ID = 0$. Table 2.5 indicates that the proposed GLR-based clustering method gives highly accurate ROI detection results in all situations. The incorrectly detected ROI are very few, which shows the competitiveness of the proposed method. DBSCAN and I-DT cannot adapt to the changes in the slides, producing often poor results. Only considering the density or distance is the main reason why DBSCAN and I-DT cannot provide as good results as the proposed method. As for Mean-shift, the results are relatively better than DBSCAN and I-DT for most slides except Slides 4 and 9. The target icons in these two slides are much larger than other icons. Therefore the size of ROIs is also relatively large. Mean-shift incorrectly breaks the ROI into more than one cluster.

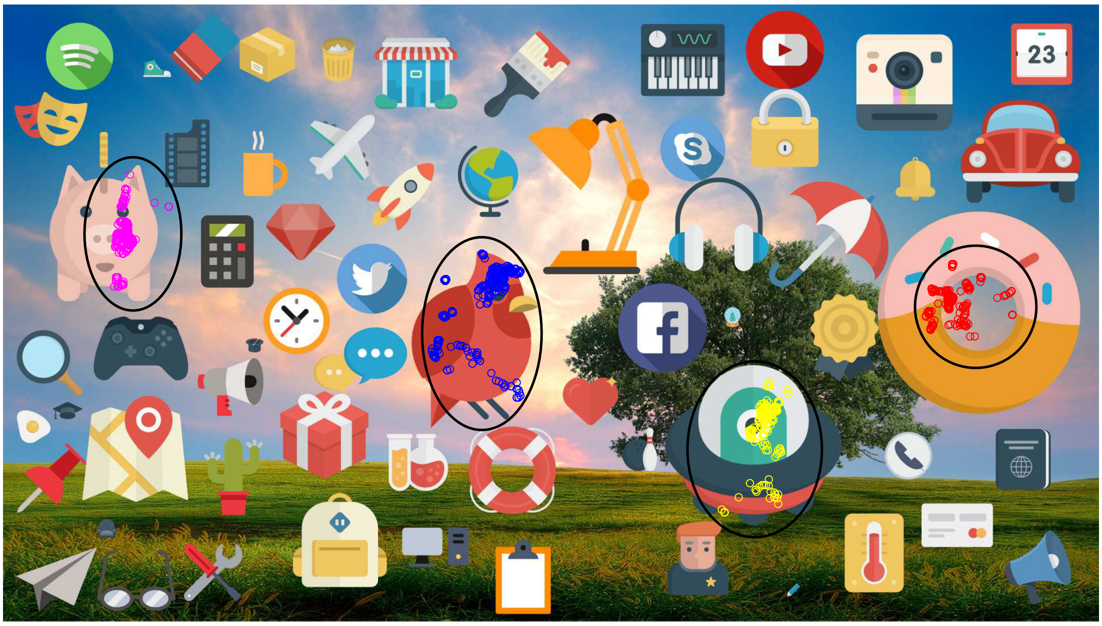


Figure 2.11: ROI estimation validation for Subject3 and Slide9.

2.4.4 Summary

In this section, a spatial clustering method is proposed for ROI detection using the emerging concept of GSP. A shift preprocessing approach is utilised to further improve the clustering accuracy. Graph Fourier Transform is applied to evaluate the cluster quality, and thereby adjust the GSP parameter. The proposed method can provide highly accurate clustering results on public shape clustering dataset. It also shows

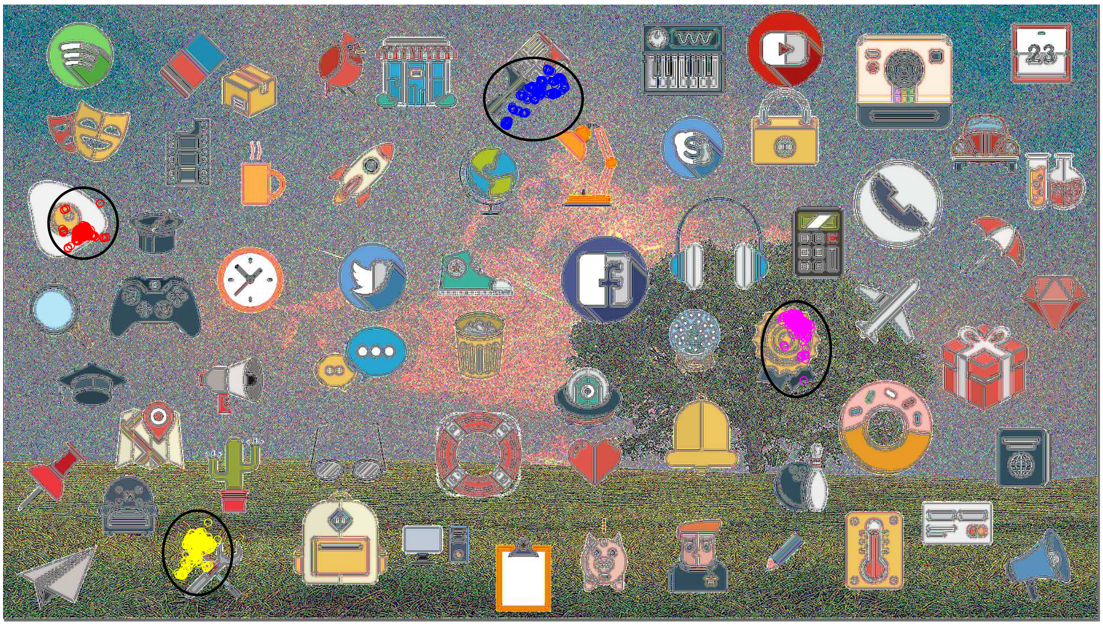


Figure 2.12: ROI estimation validation for Subject3 and Slide10.

excellent ROI detection performance for true eye tracker data in a range of challenging scenes. This successful application demonstrates the potential of GSP applied to practical applications.

2.5 Summary

This chapter reviewed the background of GSP and how GSP is developed to extend the traditional DSP and ASP. Initial successful applications of GSP and its potential are also introduced. General definitions of GSP are included to make the concept of GSP more intuitive. The related GSP extension including GTV, GLR and GFT are explained followed by a 4 nodes graph example. At the last of this chapter, a case study of applying GSP-based clustering to ROI detection in eye tracker data analysis is included.

Chapter 3

Event-based Non-Intrusive Load Disaggregation using GSP

Chapter 2 introduced the background on GSP, notation, and general definitions. Then the various GSP applications were mentioned followed by a case study on GSP classification/clustering specifically applied to identifying ROI in eye tracker data analysis. In this chapter, a GSP-based classification approach is designed for low-rate NILM. The experimental results using REDD and REFIT datasets demonstrate the competitive performance of the proposed GSP-based approaches with respect to traditionally used HMM-based and DT-based approaches.

3.1 Introduction

Despite the significant challenges reviewed in Chapter 1, low-rate NILM has attracted an increased number of researchers to contribute to its development. There has been a dramatic increase in the number of papers published on solving low-rate NILM problems in the recent decade [24], [44].

As introduced in Chapter 1, low-rate NILM approaches can be divided into two categories, state-based and event-based NILM. State-based NILM approaches commonly require good a priori information for initialisation of appliance state models or a large training dataset for good performance [27], [90], [91]. In contrast, event-based NILM approaches, are often easier to implement and deploy due to data reduction via event feature extraction [28].

The first step in an event-based NILM is to identify events. An *event* is defined as significant changes in the power signal, characterised by sharp edges, such as rising edges due to appliances being switched on, and falling edges detected when appliances are switched off. For a multi-state appliance that is appliance including more than one steady-state, a rising edge can also be transiting to a higher power state and a falling edge of being returning to a lower-power state. Features, such as rising/falling power edge magnitude, are extracted to identify events. Finally, classification built on the ground truth labels from the trained data is applied to the extracted features. An example is given in Fig. 3.1 where events are detected as the rising or falling edges when the appliance was switched on or off.

3.2 Related Work

After the work in this chapter was published in [92], follow-up research has applied GSP to solve NILM problems [82], [93] [94], [95], [96]. In [82], GSP-based clustering with

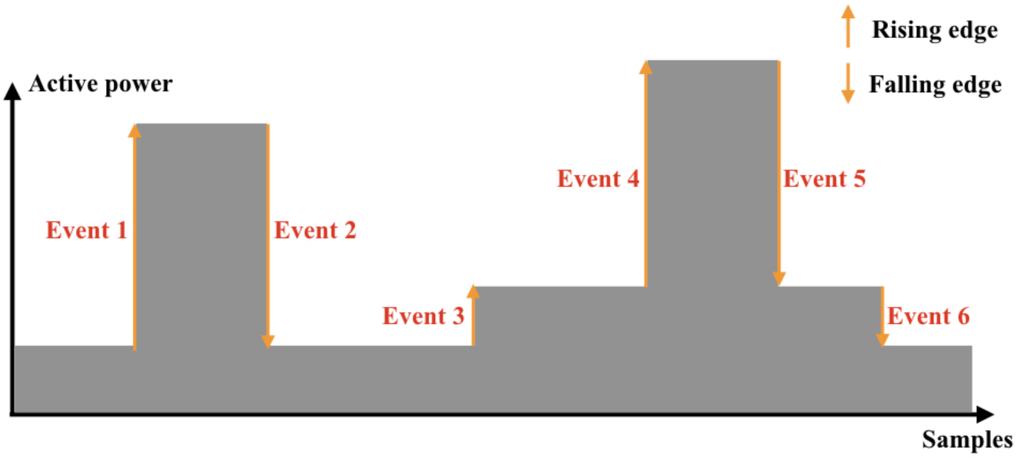


Figure 3.1: An example of aggregated power with events detected.

GLR is applied to cluster together events with similar features (namely, transient power changing values when operation state changes). In [94] and [95], different to classification/clustering labels defined as graph signal in Chapter 2 and [82], the authors directly use aggregate active power or power changing values as graph signals. [96] introduces a semi-supervised GSP-based approach for NILM, where multiple labels (that is a vector of {-1,1} where each element refers to a binary label) are applied as graph signal for classification. GLR and Manifold Regularisation (MR) are applied for predicting the classification results. To further improve NILM disaggregation results, [93] also provides GSP-based pre-processing and post-processing methods. Our proposed GSP-based method is also used as a benchmark in [97] for solving disaggregation problems of 15 minutes sampling rate smart meter data.

The proposed GSP-based method for NILM in this chapter poses the load disaggregation problem as a single-channel blind source separation problem [98] to perform a low-complexity classification of active power measurements. We treat active power changing values at event detected as a signal, indexed by the nodes of an undirected graph where each vertex corresponds to the signal sample, and the weights of the edges connecting the vertices reflect the degree of similarity between the nodes, i.e., the weights of the edges enable 'grouping' on/off events from the same appliance. Then, we define an optimisation problem that contains the regularisation term of the GLR as defined Eq. (2.4) in Chapter 2; that is, applying regularisation on the constructed graph signal to find the signal with minimum variation. However, unlike the approach in [62], which solves for a smooth graph signal using initially known labels as prior, to avoid over-smoothing, the GLR minimiser is used as a starting point to minimise the difference between the total measured power and the sum of the disaggregated loads, deviating from traditional NILM approaches.

In the following sections of this chapter, two GSP-based NILM approaches are proposed. Solution 1, via GLR minimisation, searches for a smooth graph signal under known label constraints. Solution 2, uses the GLR minimiser results as a starting point for further refinement via simulated annealing.

3.3 Methodology

3.3.1 Problem Formulation

Let $\mathcal{M}$ be the set of all known appliances in the house and $p(t_i)$ be the aggregate active power of the entire house measured by smart meter at time t_i. Without loss of generality, in the following, $p(t_i)$ is denoted as $p(t_i) = p_i \geq 0$. Let $p_i^m \geq 0$ be the power load of appliance $m \in \mathcal{M}$ at time instance t_i.

Then problem can be formulated as following:

$$p_i = \sum_{m=1}^{|\mathcal{M}|} p_i^m + n_i. \tag{3.1}$$

where, n_i is the measurement noise that also includes baseload and all unknown appliances running. The disaggregation task is, for $i = 1, \ldots, N$ and $m \in \mathcal{M}$, given p_i, to estimate p_i^m.

Let $\Delta p_i = p_{i+1} - p_i$ and $\Delta p_i^m = p_{i+1}^m - p_i^m$ to be the power load changing values between time instance t_{i+1} and t_i for aggregate and load m respectively. So problem can be reformulated as:

$$\Delta p_i = \sum_{m=1}^{|\mathcal{M}|} \Delta p_i^m + n_i^*, \tag{3.2}$$

where n_i^* is the noise without the baseload, since power variations of baseload is negligible.

3.3.2 GSP-based Disaggregation

To tackle the disaggregation problem using the GSP framework, a graph is constructed as $\mathcal{G}$= ($\mathcal{V}$, $\mathbf{A}$), where each vertex $v_i \in \mathcal{V}$ is associated to one sample Δp_i, $i = 1, \ldots, N$. For training, we assume availability of p_i and p_i^m, for $i = 1, \ldots, n < N$, for all $m \in\mathcal{M}$. Then, the task is to estimate Δp_i^m, for $n < i \leq N$.

Let $Thr_m \geq 0$ be a power threshold for appliance m which is set during training (see Sec. 3.4) in such a way that if the magnitude of the appliance active power change is larger than the power threshold, then we assume that the appliance changed its operation state (e.g., switched on/off). Then, an N-length graph signal $\mathbf{s}^m$ for Appliance m is defined as:

$$s_i^m = \begin{cases} +1, & \text{for } |\Delta p_i^m| \geq Thr_m \text{ and } i \leq n \\ -1, & \text{for } |\Delta p_i^m| < Thr_m \text{ and } i \leq n \\ 0, & \text{for } i > n. \end{cases} \tag{3.3}$$

Note that values of $\mathbf{s}^m$ can be seen as a set of classification labels, where during training ($i \leq n$) s_i^m is set to +1 (element i belongs to Appliance m class) or -1 (element i does not belong to the class), depending whether the appliance changed state or not. Since for the testing dataset ($i > n$) we do not know if the appliance was running, the corresponding values of s_i^m are initially set to 0.

We can now calculate adjacency matrix $\mathbf{A}$ according to Eq. (2.1) in Chapter 2, where $x_i = \Delta p_i$. Similarly, the graph Laplacian matrix is obtained as Eq. (2.2). Then the graph smoothness can be represented using GLR as Eq. (2.4) in Chapter 2.

$$GLR(\mathbf{s}) = \mathbf{s}^{\top}\mathbf{L}\mathbf{s} = \sum_{i,j}(s_i - s_j)^2 A_{i,j}.$$

Let $\mathbf{r}_i = \left[\Delta p_i^1, \Delta p_i^2, \cdots, \Delta p_i^{|\mathcal{M}|}\right]$ and let the difference between the actual aggregate power and the sum of disaggregated appliance powers be:

$$f(\mathbf{r}_i) = \left\|\Delta p_i - \textstyle\sum_{m=1}^{|\mathcal{M}|} \Delta p_i^m\right\|_2^2. \tag{3.4}$$

We pose the disaggregation optimisation problem as minimisation of $\sum_{i=n+1}^{N} f(\mathbf{r}_i)$ using piecewise smoothness as a prior by introducing Eq. (2.4) as a regularisation term, i.e.,

$$\min_{[\mathbf{r}_{n+1},\cdots,\mathbf{r}_N]} \sum_{i=n+1}^{N} f(\mathbf{r}_i) + \omega \sum_{m\in\mathcal{M}} \left\|\mathbf{s}^{m^T}\mathbf{L}\mathbf{s}^m\right\|_2^2. \tag{3.5}$$

Note that Eq. (3.5) defines an optimal solution as the smoothest solution that minimises Eq. (3.4), where ω is a parameter that trades off smoothness and the minimisation Eq. (3.4). We name the first term in Eq. (3.5) as fidelity term and the second term to be smoothness term. The fidelity term reflects the difference between aggregated measurements and the sum of estimated power loads. The smoothness term represents the piecewise smoothness of the graph signal with each sample assigned to be a classification label.

Eq. (3.5) is a hard optimisation problem especially since $|\mathcal{M}|$ and $N - n$ can be large. Thus, optimisation is processed separately for two terms. Solution 1 minimises only the smoothness term to predict classification label for testing samples. The other solution minimises smoothness term first and using simulated annealing algorithm to optimize fidelity term.

3.3.3 Solution 1: GLR Classifier

If as in [62], that the true classification labels form a low-frequency graph signal $\mathbf{s}^m$, then for each Appliance m, an individual classifier can be defined that minimises $\left\| \mathbf{s}^{m\top} \mathbf{L} \mathbf{s}^m \right\|_2^2$, i.e., one that finds the smoothest signal.

We call this classifier, a GLR classifier. The intuition behind it is that the labeled training samples for $i \leq n$ that are close in value to the unknown samples, $j > n$, will have large edge weights $A_{i,j}$, and so a smooth graph-signal prior will ensure that those testing samples have similar classification label as these training samples.

Since $\mathbf{s}^m_{1:n}$ is known, determined during training, the smoothness term can be simplified as [63]:

$$\begin{aligned} \mathbf{s}^{m\top} \mathbf{L} \mathbf{s}^m = & \mathbf{s}^{m\top}_{1:n} \mathbf{L}_{1:n,1:n} \mathbf{s}^m_{1:n} + \\ & \mathbf{s}^{m\top}_{1:n} \mathbf{L}_{1:n,n+1:N} \mathbf{s}^m_{n+1:N} + \\ & \mathbf{s}^{m\top}_{n+1:N} \mathbf{L}_{n+1:N,1:n} \mathbf{s}^m_{1:n} + \\ & \mathbf{s}^{m\top}_{n+1:N} \mathbf{L}_{n+1:N,n+1:N} \mathbf{s}^m_{n+1:N}. \end{aligned} \tag{3.6}$$

Since matrix $\mathbf{A}$ is symmetric, $\mathbf{D}$ is diagonal, the graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is symmetric. Therefore, Eq.(3.6) follows that:

$$\mathbf{s}^{m\top}_{1:n} \mathbf{L}_{1:n,n+1:N} \mathbf{s}^m_{n+1:N} = \mathbf{s}^{m\top}_{n+1:N} \mathbf{L}_{n+1:N,1:n} \mathbf{s}^m_{1:n}. \tag{3.7}$$

Using Eq. (3.7), and since $\mathbf{s}^m_{1:n}$ is constant, the first term does not affect minimisation, Eq. (3.6) can be minimised as:

$$\begin{aligned} \arg\min \left\| \mathbf{s}^{m^\top} \mathbf{L} \mathbf{s}^{m} \right\|_2^2 = \\ \arg\min\{2\mathbf{s}_{n+1:N}^{m^\top} \mathbf{L}_{n+1:N,1:n} \mathbf{s}_{1:n}^{m} + \\ \mathbf{s}_{n+1:N}^{m^\top} \mathbf{L}_{n+1:N,n+1:N} \mathbf{s}_{n+1:N}^{m}\}. \end{aligned} \tag{3.8}$$

This is an unconstrained quadratic programming problem with a closed form solution [63], [84]:

$$\mathbf{s}^{m^*} = \mathbf{L}_{n+1:N,n+1:N}^{\#} \left(-\mathbf{s}_{1:n}^{m^T} \mathbf{L}_{1:n,n+1:N} \right)^\top . \tag{3.9}$$

where $(.)^{\#}$ denotes the pseudo-inverse matrix and $(.)^\top$ denotes the transpose of a matrix. $\mathbf{s}^{m^*}$ is predicted s_i^m values for testing samples $i \in [n+1, N]$.

Once $\mathbf{s}^{m^*}$ is determined, if $s_i^{m^*} > T_s$, then, appliance m changed state, $\Delta p_i^{m^*}$ is set to $\overline{\Delta p^m}$ the mean of power changing values for appliance m, which is calculated through the training process; otherwise, appliance m did not change its state, and $\Delta p_i^{m^*} = 0$. We use $\mathbf{r}_i^* = \left[\Delta p_i^{1^*}, \Delta p_i^{2^*}, \cdots, \Delta p_i^{|\mathcal{M}|^*}\right]$ to represent the output of GLR classifier for NILM, that is estimation of active power changes for each appliance over all testing samples.

In contrast to [62], where the threshold T_s is set to zero, the classification threshold in this chapter is set to T_s =0.5, which imposes that only samples that are highly correlated with the training samples will be assigned to the same class. The value of 0.5 was found heuristically to yield the fewest false positives without increasing the number of false negatives.

We repeat minimisation of the smoothness term for all appliances $m \in \mathcal{M}$, where after each appliance has been disaggregated, its contribution to the total load is removed by subtracting its mean from the aggregate. Note that the same nodes are used for all appliances, but matrix $\mathbf{A}$ changes with updated $\Delta p_i, i = n+1, ..., N$.

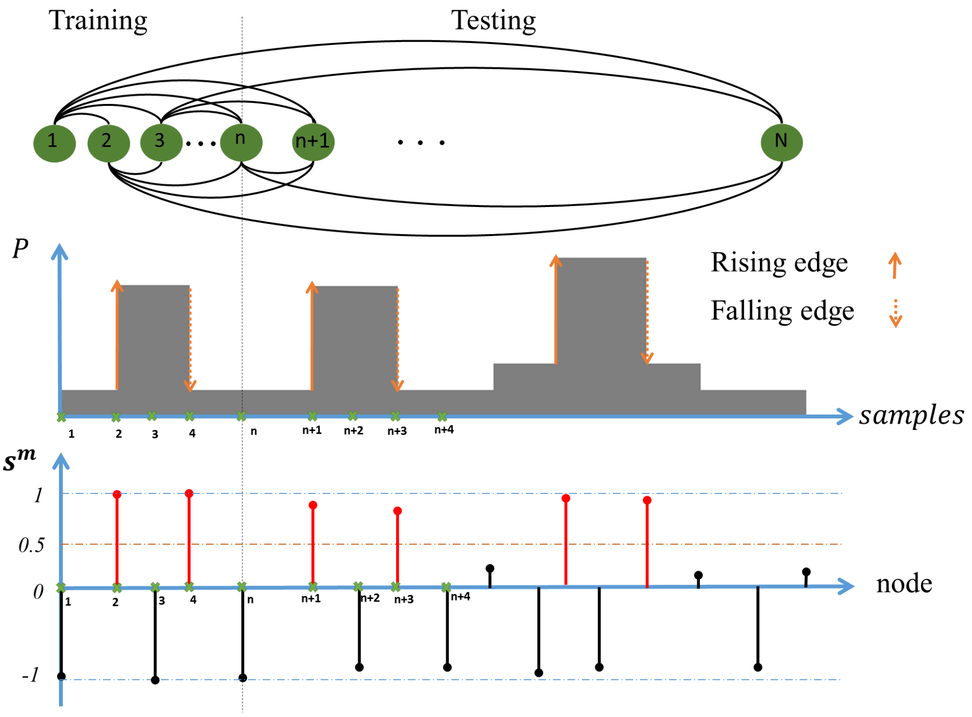


Figure 3.2: An example of GSP-based disaggregation.

An example is given in Fig. 3.2. The top figure shows the generated graph nodes and connections between the nodes. Note that each node corresponds to one active power reading shown in the middle graph. The graph signal $\mathbf{s}$ (shown in the bottom graph) contains classification labels for each power edge. During testing $(i > n)$ all calculated values of $\mathbf{s}^m$ above threshold $T_s = 0.5$ will be classified to the Appliance m class, i.e., that is appliance m state changed.

The complexity of the approach depends on $(N-n)$ since it is necessary to find the pseudo-inverse of an $(N-n) \times (N-n)$ real-valued matrix, which can be done in

$O((N-n)^3)$ time [99], [100]. The pseudocode of the GLR classifier for NILM is shown in Algorithm 2.

```
Algorithm 2: GLR Classifier for NILM.
Input: $\mathbf{\Delta p}_{1:N}$, $\mathbf{\Delta p}^m_{1:n}(n < N)$, $|\mathcal{M}|$, $Thr_m$, $T_s$, $\sigma$;
Output: $\mathbf{\Delta p}^{m^*}_{n+1:N}$ for all $m$;
set $m = 1$ ;
while $m < |\mathcal{M}|$ do
    compute $\mathbf{A}$, $\mathbf{L}$ with Eq. (2.1) and (2.2), respectively;
    set $\mathbf{s}^{m^*}_{1:N}$ as in Eq. (3.3) using $Thr_m$;
    compute $\mathbf{s}^m_{1:N}{}^*$ via Eq. (3.9) and determine $\mathbf{\Delta p}^{m^*}_{n+1:N}$;
    remove $\mathbf{\Delta p}^m_{1:n}$ and $\mathbf{\Delta p}^{m^*}_{n+1:N}$ from $\mathbf{\Delta p}_{1:N}$, $m = m + 1$;
return $\mathbf{\Delta p}^{m^*}_{n+1:N}$;
```

The flow chart of the algorithm is shown in Fig. 3.3.

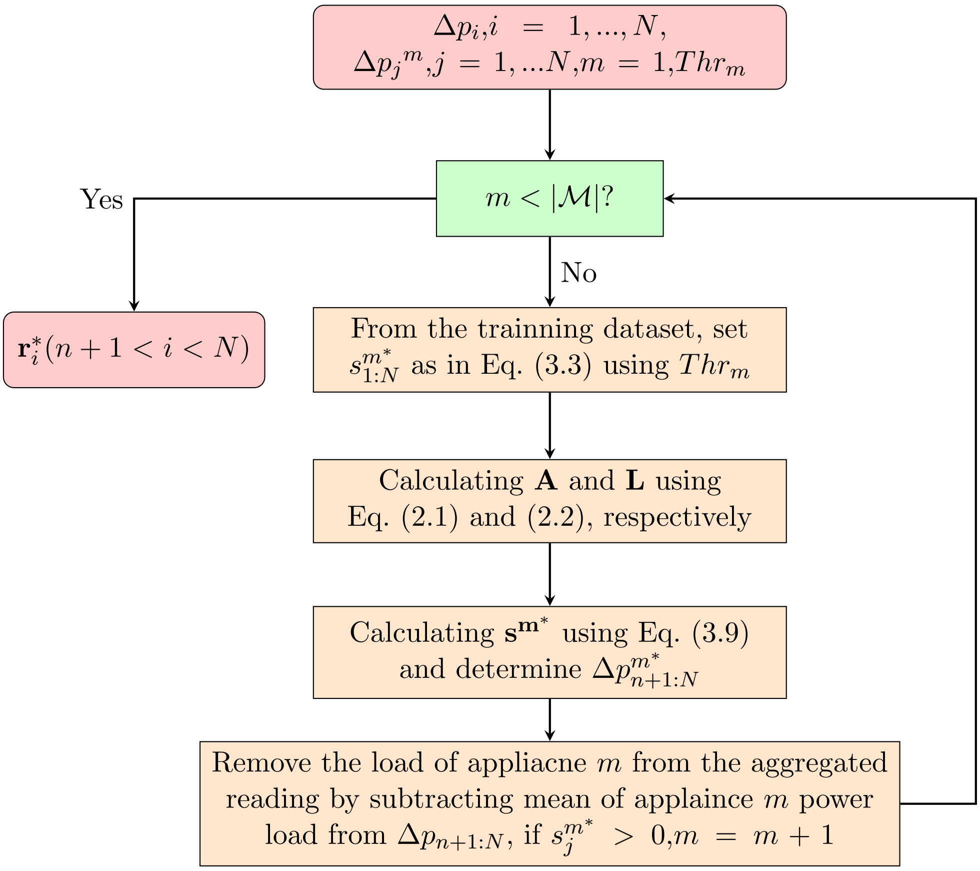

$\Delta p_i, i = 1, ..., N,$
$\Delta p_j{}^m, j = 1, ...N, m = 1, Thr_m$
$m < |\mathcal{M}|?$
Yes
No
$\mathbf{r}^*_i(n+1 < i < N)$
From the trainning dataset, set $s^{m^*}_{1:N}$ as in Eq. (3.3) using Thr_m
Calculating $\mathbf{A}$ and $\mathbf{L}$ using Eq. (2.1) and (2.2), respectively
Calculating $\mathbf{s}^{\mathbf{m}^*}$ using Eq. (3.9) and determine $\Delta p^{m^*}_{n+1:N}$
Remove the load of appliacne m from the aggregated reading by subtracting mean of applaince m power load from $\Delta p_{n+1:N}$, if $s^{m^*}_j > 0, m = m + 1$

Figure 3.3: Flow chart for Solution1: GLR Classifier.

3.3.4 Solution 2: GSP + Simulated Annealing (SA) Refinement

While Solution 1 finds the smoothest graph signal under given constraints, it may over-smooth the result. To avoid this, a sub-optimal solution is introduced for solving Eq. (3.5) based on minimising both fidelity and smoothness terms.

First, Solution 1 is applied based on Eq. (3.9), which is the starting point for the simulated annealing method [101] that will attempt to refine the result by minimising the first term of Eq. (3.5). We demonstrate in the next section that simulated annealing provides a solution identical or close to a greedy full-search method, which has exponential complexity in $|\mathcal{M}|$.

The flow chart of Solution 2 is shown in Fig. 3.4. Note that the input is, $\mathbf{r}_i^*, i = n+1, \cdots, N$, which is a solution found by Solution 1. This solution is refined via iterative simulated annealing, where $iter_{num}$ denotes the number of iterations, T_0 is the initial temperature and Y stands for terminate temperature. In each iteration, a candidate solution $\mathbf{q}_i$ is formed by randomly setting appliances on/off. Step $\exp\left\{\frac{f(\mathbf{r}_i)-f(\mathbf{q}_i)}{T}\right\} > rand()$ ensures that when the "temperature" T is high, the algorithm does not accept a worse solution, where $rand()$ is a function that returns a random number in the interval $(0,1)$. We heuristically demonstrate the convergence of the algorithm in the next section.

3.4 Results and Discussion

In this section, the following results are presented: (1) Relative performance of using SA-based optimisation only vs. Full Search applied on minimisation of Eq. (3.4), (2)

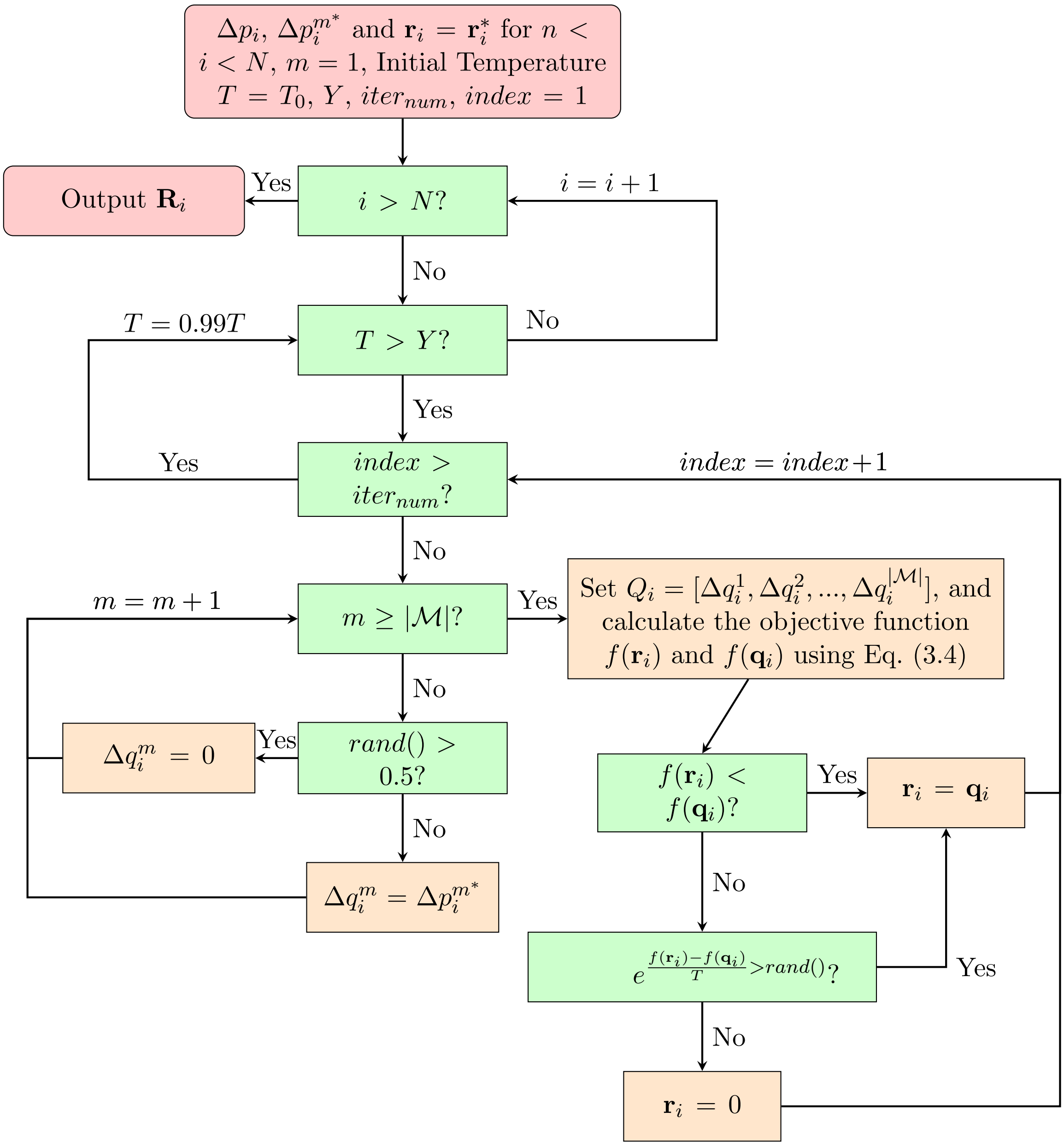

Δp_i, $\Delta p_i^{m^*}$ and $\mathbf{r}_i = \mathbf{r}_i^*$ for $n < i < N$, $m = 1$, Initial Temperature $T = T_0$, Y, $iter_{num}$, $index = 1$
Output $\mathbf{R}_i$
Yes
$i > N$?
$i = i + 1$
No
$T = 0.99T$
$T > Y$?
No
Yes
Yes
$index >$ $iter_{num}$?
$index = index + 1$
No
$m = m + 1$
$m \geq |\mathcal{M}|$?
Yes
Set $Q_i = [\Delta q_i^1, \Delta q_i^2, ..., \Delta q_i^{|\mathcal{M}|}]$, and calculate the objective function $f(\mathbf{r}_i)$ and $f(\mathbf{q}_i)$ using Eq. (3.4)
No
$\Delta q_i^m = 0$
Yes
$rand() >$ 0.5?
$f(\mathbf{r}_i) <$ $f(\mathbf{q}_i)$?
Yes
$\mathbf{r}_i = \mathbf{q}_i$
No
$\Delta q_i^m = \Delta p_i^{m^*}$
No
$e^{\frac{f(\mathbf{r}_i) - f(\mathbf{q}_i)}{T}} > rand()$?
Yes
No
$\mathbf{r}_i = 0$

Figure 3.4: Flow chart for Solution 2. $\mathbf{r}_i^*$, $i = n + 1, \cdots, N$ is the solution found by Solution 1.

Relative performance of Solution 1, SA only and Solution 2, (3) Comparison with state-of-the-art methods of [28] and [27], (4) Computational complexity.

For validation purposes, household appliances, for which timestamped individual power consumption is available via sub-metering, are treated as known appliances. They are: Dishwasher (DW), Refrigerator (REFR), Microwave (MW), Washer dryer (WD), Kitchen outlet (KO), Stove (ST), air-conditioning high (AC$_H$) and low (AC$_L$) state, Electronics (EL), Washing machine (WM), Kettle (KE), Electric shower (ES), Electric heater (EH), Freezer (FRZ), Fridge-freezer (FFRZ). Baseload and Unknown appliances are abbreviated as BL and UN, respectively.

For training of the proposed methods, for each appliance, a period of aggregate measurements (normally 2 days) is applied with here including the occurrence of only that appliance is running (together with the base-load). If a new appliance is introduced in the household, the training dataset is updated with that particular appliance's signature, comprising samples representing a full run from on to off. No retraining is needed to be performed for other appliances. At least 14 days worth of data is used for testing.

3.4.1 Parameter Setting

The parameter used in this section is set as shown in Table 3.1.

Thr_m is always set to one half of the difference between mean values of Appliance m's consecutive states. For example, for a two-state appliance, on and off states only, Thr_m would be half of the power value in the on-state. The scaling factor σ is selected during training in the area from the first non-zero value of the smoothness term to

Table 3.1: Parameter settings for the proposed method.

symbol	parameter	setting
T_s	classification label threshold	0.5
Y	temperature threshold	0.01
T_0	initial temperature	$100\|\mathcal{M}\|$
$iter_{num}$	iteration times	1000
Thr_m	appliance power threshold	$\frac{1}{2}\|\overline{P^m}\|$

the inflexion point, which was shown to provide the highest performance. For SA, temperature threshold $Y = 0.01$, $T_0 = 100|\mathcal{M}|$ and $iter_{num} = 1000$ which trades off performance and complexity. Classification label threshold T_s is set to be the half of its maximum. The proposed method is not sensitive to the parameters provided in Table 3.1. Deviating from these values by 25% would not distinctly affect the accuracy of results.

3.4.2 Performance Metrics

The evaluation metrics used are Precision (PR), Recall (RE) and F-Measure (F_M) [102] defined as:

$$PR = TP/(TP + FP) \tag{3.10}$$

$$RE = TP/(TP + FN) \tag{3.11}$$

$$F_M = 2 * (PR * RE)/(PR + RE), \tag{3.12}$$

where True Positive (TP) is recorded when the detected appliance actually switched state, False Positive (FP) is when the detected appliance was not changing state, and False Negative (FN) indicates that the appliance operation was not detected. Precision captures the correctness of detection - the higher the PR, the fewer the FPs. On the other hand, high RE means a low number of FNs, which implies that a higher percentage of appliance state changes are detected correctly. F_M balances PR and RE.

In addition, the Average Normalized Error (ANE) is used to measure the energy difference between total power estimated by the NILM algorithm across all known appliances and the actual power consumed, which is defined as:

$$ANE = \frac{\left| \sum_{i=1}^{N} \bar{p}_i - \sum_{i=1}^{N} \hat{p}_i \right|}{\sum_{i=1}^{N} \bar{p}_i}, \tag{3.13}$$

where $\hat{p}_i$ is sum of the power estimated by the NILM algorithm from all disaggregated appliances $m \in \mathcal{M}$ at time i and $\bar{p}_i$ is the actual power consumed by all known appliances at time i. $\mathbf{p}^{m^*}$ is recovered after $\boldsymbol{\Delta}\mathbf{p}^{m^*}$ is predicted, according to the average working power of appliance m obtained from training. This measure is useful, e.g., for appliance-itemized billing, when quantifying, across a fixed period of time, the error incurred by the NILM algorithm in estimating the total power consumed by individual appliances.

3.4.3 Full-Search vs SA

Fig. 3.5 shows an example of the convergence of the SA algorithm. It can be seen that the method converges after less than 300 iterations. Similar results are obtained for different datasets.

A full-search method can be used to minimise Eq. (3.4) off-line when $|\mathcal{M}|$ is small. Assuming only two-state appliances (i.e., ON/OFF), for each sample i, each appliance can either be switched on or off, or does not change state. Thus, with full-search, there

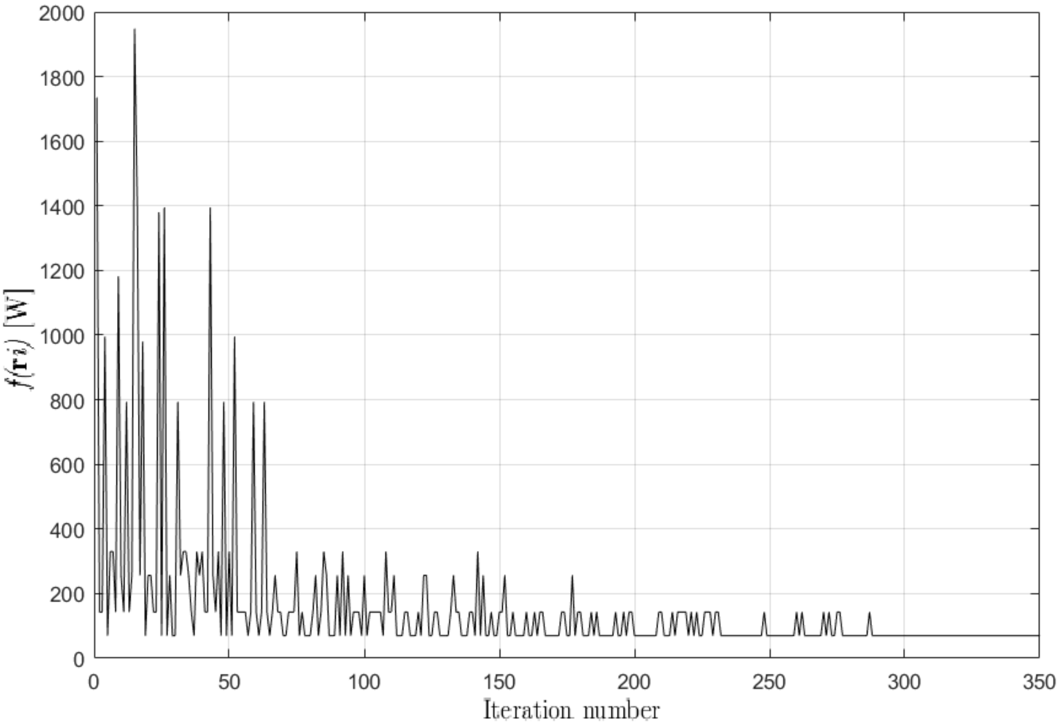


Figure 3.5: Convergence of the simulated annealing method for House 2 from the REDD dataset.

are $3^{|\mathcal{M}|}$ possible combinations that should be inspected for each sample. Table 3.2 shows the F_M value comparison between the full-search method and SA approach applied only on minimising Eq. (3.4) for two houses from the REDD dataset. One can see that the proposed sub-optimal SA approach finds a solution that is either identical to or very close to the full-search method.

Table 3.2: SA vs. full-search (FS) for REDD House (H) 2 and 6.

	REFR	ST	MW	KO	EL	AC_H	AC_L	DW
FS H2	0.84	0.31	0.91	0.83	-	-	-	0.62
SA H2	0.84	0.30	0.91	0.83	-	-	-	0.62
FS H6	0.77	0.75	0.77	0.55	0.48	0.92	0.56	-
SA H6	0.77	0.75	0.77	0.55	0.48	0.92	0.53	-

3.4.4 SA, Solution 1 and Solution 2 Comparison

In this subsection, three approaches are compared: (1) GLR (Solution 1); (2) minimising the first term of Eq. (3.5) only, i.e., Eq. (3.4), using SA; (3) incorporating the latter two approaches (Solution 2). As shown in Tables 3.3 and 3.4, SA and Solution 1, lead to significantly worse F_M performance for some appliances than treating them jointly (Solution 2).

When minimising Eq. (3.4), SA only uses known mean values of appliance m and thus does not account for small fluctuations in actual instantaneous. SA results in Table 3.3 are poor for Stove, since Stove (mean p_i^m = 408W) is often confused with the low-power state of Dishwasher (mean p_i^m=349W). Similarly, in Table 3.4, Kitchen Outlet and Electronics have similar operating power states, hence they are often incorrectly labelled.

SA does not outperform Solution 1, but the advantage of Solution 2 (integrating

Solution 1 and SA) is consistent across all appliances and especially significant for appliances such as Electronics and AC.

Table 3.3: The F_M results for REDD House 2.

Appliance	REFR	ST	MW	KO	DW
Avg.Power [W]	171	408	1840	1056	1201 (349)
Solution1	0.81	0.81	0.91	0.85	0.83
SA	0.84	0.30	0.91	0.83	0.62
Solution2	0.84	0.86	0.93	0.88	0.84

Table 3.4: The F_M results for REDD House 6.

Appliance	REFR	ST	MW	KO	EL	AC_H	AC_L
Avg.Power [W]	149	1724	1352	946	815	2376	357
Solution1	0.77	0.92	0.91	0.86	0.28	0.94	0.53
SA	0.77	0.75	0.77	0.55	0.48	0.92	0.56
Solution2	0.78	0.92	0.91	0.88	0.66	1.00	0.79

3.4.5 Comparison with State-of-the-art

A comparison of the performance of proposed Solution 2 with the state-of-the-art NILM approaches, namely DT approach [28] and HMM-based approach [27] is shown in Tables 3.5, 3.6, and 3.7 for REDD Houses 1, 2, 6, respectively.

Table 3.5: Comparison between the proposed Solution 2 (P), HMM and DT-based methods for REDD House 1.

Appliance	REFR	MW	DW	KO	WD
F_{M_p}	0.88	0.70	0.57	0.39	0.89
$F_{M_{HMM}}$	0.97	0.50	0.13	0	0
$F_{M_{DT}}$	0.88	0.85	0.39	0.19	0.88

The proposed method was also tested using the noisy REFIT dataset [56]. The REFIT households were monitored remotely and uninterruptedly, while they were going about their usual domestic routines. Each house contains numerous appliances that

Table 3.6: Comparison between the proposed Solution 2 (P) and HMM and DT-based methods for REDD House 2.

Appliance	ST	REFR	KO	MW	DW
F_{M_p}	0.86	0.84	0.88	0.93	0.84
$F_{M_{HMM}}$	0.21	0.90	0.68	0.47	0.04
$F_{M_{DT}}$	0.33	0.97	0.92	0.97	0.32

Table 3.7: Comparison between the proposed Solution 2 (P), HMM and DT-based methods for REDD House 6.

Appliance	ST	REFR	KO	MW	AC	EL
F_{M_p}	0.92	0.77	0.88	0.91	0.88	0.66
$F_{M_{HMM}}$	0	0.88	0	0	0.12	0.03
$F_{M_{DT}}$	0.67	0.99	0	0	0.89	0

were not monitored, including oven, lights, chargeable devices, small electronics, etc., which are considered unknown and contribute significantly towards the noise. Tables 3.8 and 3.9 show results for REFIT Houses 2 and 17, respectively.

Table 3.8: Comparison between the proposed Solution 2 (P), HMM and DT-based methods REFIT House 2.

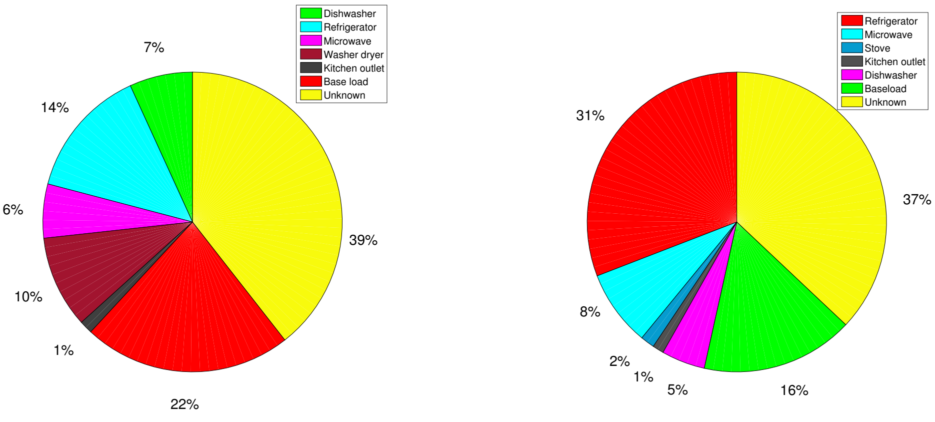
Appliance	FRZ	WM	DW	TV	MW	KE
F_{M_p}	0.77	0.55	0.62	0.49	0.95	0.88
$F_{M_{HMM}}$	0.49	0.26	0	0.06	0.01	0.01
$F_{M_{DT}}$	0.33	0.73	0.36	0	0.95	0.58

Fig. 3.6 and 3.7 show the relative contribution of known appliances to the total aggregate load, for REDD Houses 1 and 2 over a period of two weeks and REFIT Houses 2 and 17 over a period of one month (October 2015). The proposed Solution 2 can disaggregate over 60% of the household's total load, demonstrating its effectiveness in accounting for individual appliance demand.

With respect to disaggregation accuracy, the ANE measure given by Eq. (3.13), which measures the discrepancy between the true consumption and the disaggregated values, is, for REDD Houses 1 and 2, 7.33% and 6.91%, respectively, and for REFIT

Table 3.9: Comparison between the proposed Solution 2 (P), HMM and DT-based methods for REFIT House 17.

Appliance	FRZ	FFRZ	KE	MW	WM
F_{M_p}	0.64	0.76	0.96	0.81	0.76
$F_{M_{HMM}}$	0.32	0.19	0.01	0.22	0.01
$F_{M_{DT}}$	0.81	0.79	0.97	0.71	0.50



(a) House1.The total demand for two weeks was 158 kWh. ANE=7.33%

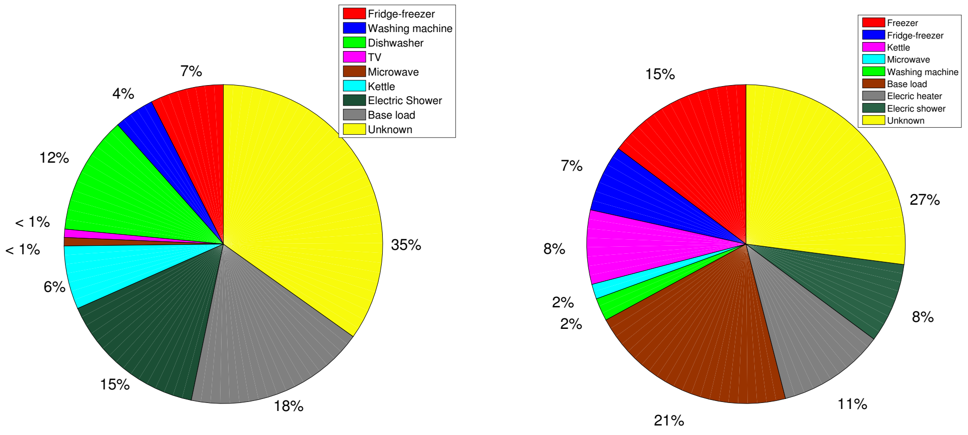
(b) House2. he total demand for two weeks was 77 kWh. ANE=6.91%

Figure 3.6: Relative contribution of appliances to the total load for two REDD houses.

Houses 2 and 17, the ANE is 8.97% and 9.24%, respectively. These results demonstrate that a very small percentage of the total load is wrongly disaggregated.

3.4.6 Discussion

As can be seen from Tables 3.5, 3.6, and 3.7, the proposed method significantly outperforms the HMM-based approach for all appliances in the REDD dataset except the refrigerator. HMM usually performs well disaggregating the refrigerator due to continuous and sole operation (i.e., without any other appliances running) during the night and hence large data availability for learning and improving initial models [27,28]. The poor performance for other appliances can be attributed to the short training period.



(a) House2. Total monthly demand was 372 kWh. ANE=8.97%

(b) House17. Total monthly demand was 341 kWh. ANE=9.24%

Figure 3.7: Relative contribution of known appliances to the total load over one month (Oct 2015) for two REFIT houses.

The proposed method shows better or similar performance to the DT method in [28] except for microwave in Houses 1 and 2, and refrigerator in most houses. These appliances have very small power fluctuations during operation, and hence the decision tree classifier based on the rising and falling power edge works especially well.

Tables 3.8 and 3.9 show poor HMM results for the REFIT dataset due to the noise and many unknown appliances. The proposed solution is again better than or equal to the benchmark methods for most of the appliances except washing machine in House 2, where DT performs the best due to very distinctive high-state washing machine power edges.

The results for both REDD and REFIT dataset demonstrate that the proposed method provides more accurate disaggregation than the benchmark methods for most appliances. The difference is especially pronounced for the kitchen appliances, namely

Kettle, Microwave, and Stove. This is due to the fact that Stove and other kitchen appliances normally have a short operation time and relatively high power. Thus machine-learning based approaches cannot generate probabilistic models that accurately capture appliance operation. The HMM-based approach is sensitive to noise and suffers from a short training period. The DT-based method works well for appliances that have small fluctuations in load during their steady-state operation.

The results for multi-state appliances (dishwasher, washing machine) are generally worse for all tested algorithms. This is due to first, the fact that low-power operating states are often difficult to detect since they are 'hidden' in the baseload and noise. Second, since these appliances are on for a very long period, many appliances are likely to run in parallel, adding to the noise. Finally, multi-state appliances are not used frequently, thus it is more difficult to isolate good training periods.

On the other hand, cold appliances, refrigerator, freezer and fridge-freezer, are frequently used and have regular periodic signatures; thus the algorithms show good accuracy. However, due to the higher level of noise from many unknown appliances, slightly worse results are obtained for the REFIT dataset with a higher false positive rate.

The TV in REFIT House 2 is hard to identify since it has relatively low operating power and thus it is often hidden by noise and baseload. In addition, TV usually runs for a long period of time, and thus many appliances will run in parallel. Still, the proposed method is more successful than the benchmark methods, since it is less sensitive than HMM to the training dataset that does not have any instances of TV running alone and it is more successful in resolving the cases when multiple appliances run in parallel than DT. Electronics in REDD House 6 include different electronics equipment which produces complex power signatures that lead to worse results. The proposed algorithm again shows more robustness than the benchmark methods in this situation, since it is less sensitive to fluctuations in steady-state power signature during training and testing.

Fig. 3.6, and 3.7 show that unknown appliances, including lighting, whose consumption could not be validated, make up only under 40% of the total load. The ANE results indicate that the discrepancy between the actual load due to known appliances and NILM-estimated load is very low. However, the ANE results of REFIT houses are slightly worse than in the case of REDD dataset, mostly due to additional noise from unknown appliances and multi-state appliances, i.e., washing machine and dishwasher. Note that, each REFIT house contained over 40 operational appliances (see [56] for monitoring details), many of which could not be validated via a time-diary or appliance-level load measurements. Moreover, considering that lighting, which contributes towards the 'Unknown' category in Fig. 3.7, accounts for about 20% of a household's domestic consumption in October in the UK (see [56] and references therein), the results demonstrate the potential of the proposed technology in effectively disaggregating smart meter aggregate loads.

3.4.7 Computational Complexity

The proposed algorithm was implemented in Matlab2015b and executed on an Intel(R) Core(TM) i5-3470 CPU @ 3.20GHz machine running Windows 7 64-bit. Table 3.10 shows the computational time needed to disaggregate the refrigerator, which is periodically on, in the REDD House 1 for various testing set sizes $N - n$. The shaded row shows the execution time and the accuracy when the testing dataset of size $N - n$ is split into windows of size 1000. Each window is independently processed, but the whole testing dataset is used. This reduces the dimension of matrix $\mathbf{L}$ and hence lowering the complexity of calculating Eq. (3.9). The bottom unshaded row shows the accuracy when the entire testing dataset is processed in one go - it also shows indirectly the effect of using only one 'window' of size 1000, 2000, 3000 etc. Similar F_M values in all cases confirm that splitting the testing dataset in smaller manageable 'windows' significantly reduces execution time without a loss in performance. The table also shows that the proposed method performs well with a small training overhead, i.e., increasing the number of samples in the testing dataset does not improve performance. Note that a window size of less than 1000 would not capture a full run of appliances with long operation period, such as washing machines or AC.

Table 3.10: Computation time of the proposed Solution 2 for REFR in REDD House 1. The shaded rows show results obtained using separate windows of size 1000. The bottom rows show results where the testing dataset is not split. All execution time is for the whole testing dataset with size $N - n$.

$N - n$	1000	2000	3000	4000	5000	6000	7000
Time [s]	1.33	3.22	3.86	4.41	4.99	5.52	6.01
F_M	0.89	0.89	0.90	0.90	0.90	0.89	0.89
Time [s]	1.33	16.85	44.81	90.96	153.99	243.88	406.69
F_M	0.89	0.89	0.90	0.90	0.90	0.90	0.90

When two weeks of data are used, comprising just over $N = 20,000$ samples, the computation time on average for REDD House 1, 2 and 6, is between 10 to 12 sec for disaggregating one appliance. This is faster than the HMM-based method, which disaggregates the same amount of data in 40-50 sec, as reported in [53] using the similar implementation platform.

3.5 Summary

This chapter presented two NILM algorithms based on the GLR classification concept of GSP. The first approach performs GLR assuming that classification labels form a piece-wise smooth graph signal. The second approach further refines the GLR solution using simulated annealing. Previous GLR solutions are applied as the starting of simulated annealing. Experimental results show the competitiveness of the methods in either accuracy or efficiency with respect to two benchmarking NILM methods and were demonstrated over two datasets with a range of appliances. We also discuss the relative performance of the proposed methods for different appliances and how robust the methods are to short training periods, and how fast this can be implemented through effective windowing without performance loss. The proposed methods can work with conventional smart meters, e.g., accessing 10 second data via the Consumer Access Device, and do not require any additional hardware installation.

Chapter 4

NILM via DTW-based GSP

In Chapter 3, an event-based NILM algorithm via GLR-based classification combined with SA based refinement is introduced. Though the proposed methods outperform the benchmarks, HMM-based and DT-based NILM algorithms, they highly rely on the quality of the 'edges' as they ignore all measured values between the start and the end of the event. To improve the performance, in this chapter, a novel unsupervised NILM method is proposed using DTW-embedded GLR-based clustering that considers the entire event data, from the detected start of the event until the end.

4.1 Introduction

As shown in Chapter 3, although GSP-based designs in [82], [93] [94], [95] and [96] demonstrate good potentials in solving NILM problem, there are still limitations.

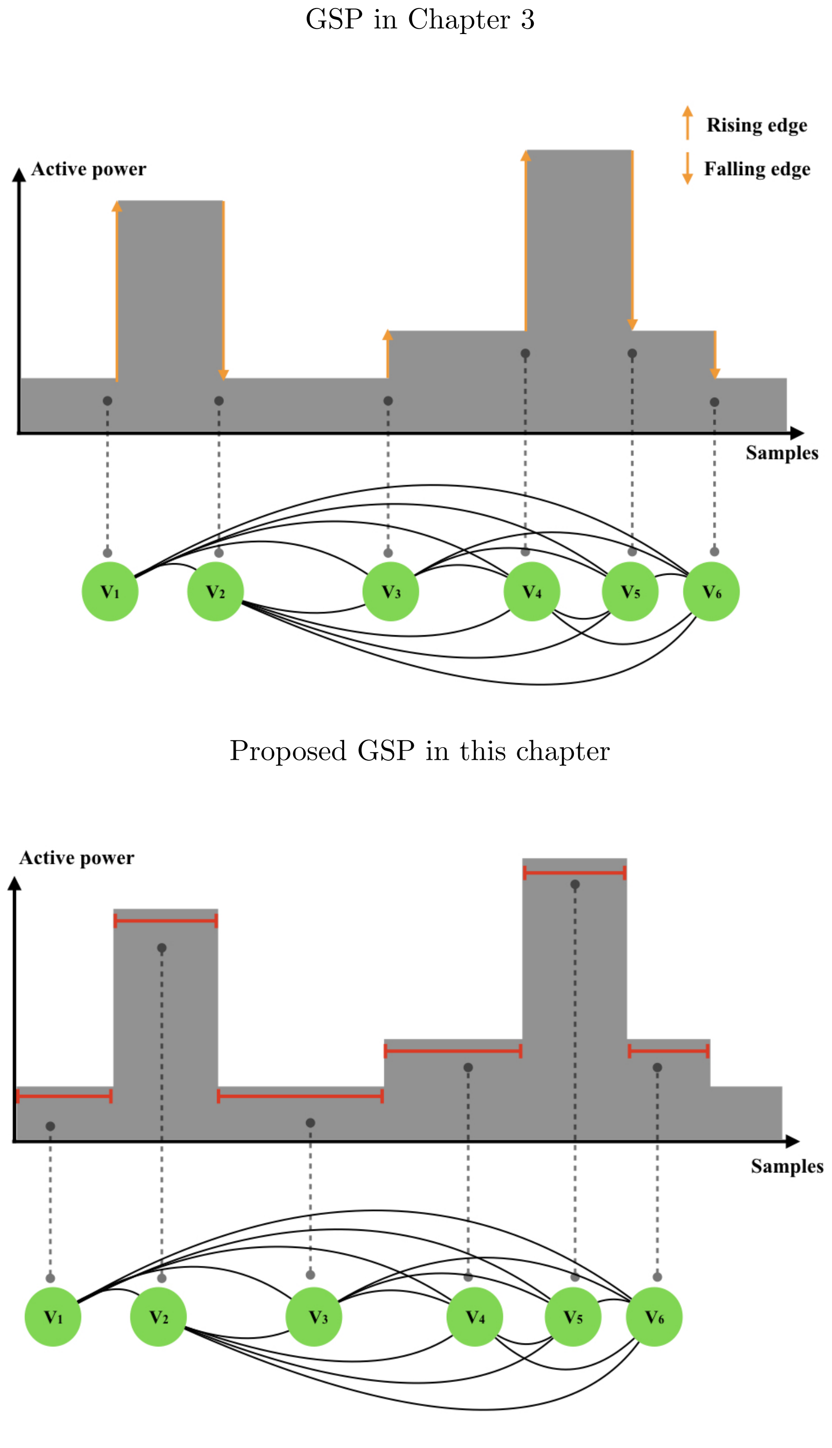
In Chapter 3 and [82], GSP is applied as a classification/clustering tool based exclusively on transit features, that is, on the change of power values when an event starts and ends. Graph signal $\mathbf{s}$ is defined as classification labels of events detected. These above methods throw away measurements between the start and end of an event, hence lose relevant information and surfer from noise, including unknown appliance operations and high load fluctuations.

In [94] and [95], aggregated power measurements $\mathbf{p}$ and active power changing values, that is $\mathbf{\Delta p}$, the difference between two consecutive power measurements, are used as the GSP-based classification features, respectively. Different from Chapter 3 and [82], methods in [94] and [95] directly use the classification features as the graph signal, that is $\mathbf{s} = \mathbf{p}$ and $\mathbf{s} = \mathbf{\Delta p}$, respectively. Since each sample from aggregated measurements or the difference between two consecutive samples, is related to a graph node, the above two methods show high computation complexity. The method in [94] also shows poor performance when multiple appliances are operating at the same time when only active power measurements are used as GSP-based classification features.

The semi-supervised GSP-based classification method in [96], applies multiple labels as graph signal. Different from methods in Chapter 3 and [82] that use binary GSP-based classification on each appliance, this approach defines each graph signal s_i as a vector of classification labels for all appliances and only applied classification algorithms once. Optimisation on this multiple-label based graph signal is less accurate and more complex. This semi-supervised approach can only show competitive results for relative 'clean' datasets. In fact, as the distance between classes is not captured by GLR, GLR-based classification is ineffective when used directly in the multi-class problem. Instead, the multi-class problems are solved using multiple binary classifiers, for example, via the one-against-all approach. All methods above except [96] associate single feature to each graph node that is, p_i or Δp_i at sample i. To address the above limitations of current designs, in this chapter, we first attempt to increase the number of features related to graph nodes. Weights are introduced to each additional features which are the neighbouring measurements of p_i, that is p_{i-k} to p_{i+k}. Then a novel unsupervised method is proposed for NILM disaggregation using DTW. The proposed method extends the previous GSP clustering [74], [82] approach, by considering the whole data segment, instead of working with the edges of detected events. Since extracted events are of different length, DTW is used to define a distance measure used to design a graph.

The novelty of the proposed method is to assign a time-series sequence to a graph node based on multiple extracted features (including power values of samples inside each data segment, length of the data segment and variation tendency) to facilitate GSP-based classification or clustering. The proposed approach segments aggregated power data into windows, where each window contains the complete set of measurements between two contiguous events. These windows are associated with graph nodes. Then DTW distance instead of Euclidean distance, used in previous work in Chapter 3 and [82], is applied to set the weights between graph nodes. If two aggregated data segments are comparable in values and have similar variation tendency, the DTW distance between them will be relatively small. Then GSP based clustering [82] is used to cluster different data segments. GFT-based self parameter tuning, as introduced in Chapter 2, is also applied for choosing the scaling factor σ for adjacency matrix $\mathbf{A}$.

Fig. 4.1 compares the GSP method proposed in Chapter 3 and the GSP method proposed in this chapter based on how the aggregated measurements are associated with the graphs nodes. As displayed in the figure, for event-based GSP algorithms (in


GSP in Chapter 3

Proposed GSP in this chapter

Figure 4.1: Comparison of association between graph node and raw data for GSP method in Chapter 3 and proposed GSP method in this chapter.

Chapter 3), [82], each rising and falling edge is related to a graph node, which are aggregated active power changes at the beginning and the ending of events. Graph signals are defined as the classification labels. In [94], [95], each graph node is still associated with one sample. Different from the above methods, the proposed GSP solution in this chapter associates data segments consist of the entire aggregate measurements between two contiguous events to a graph node.

The remaining of the chapter is organised as follows: Section 4.2 describes the proposed DTW embedded GLR-based clustering applied to NILM. Section 4.3 provides a short background on benchmarking methods, followed by results discussed in Section 4.4.

4.2 Methodology

The main limitation of the prior NILM approaches is the fact that they ignore fluctuation of power values between the event edges, that is, the approaches rely only on power value when the appliance is switching on/off or changing state. This is in contract to state-based approaches that track the power values but are difficult to train [27], [29], [103]. To address these limitations, an improved GSP-based NILM design is proposed and described next. The proposed NILM-based system comprises a pre-processing block, data segmentation, clustering, and finally labelling the clusters. Each of the blocks is discussed one by one, in the following subsections.

4.2.1 Pre-processing

The smart meter measurements are accompanied by unavoidable measurement noise. Furthermore, transient power values act as noise in the stead-state analysis. The fluctuation of power while appliances are at their steady on states can also be regarded as interference. Pre-processing has been commonly used to reduce these effects [30], [93]. First, signal edges are sharpened by applying median filtering to isolate better potential events. That is, if $|\Delta p_i| > Thr$, then the ith sample potentially belongs to the start/end of an event, that is, to a state change. We update such samples using:

$$\Delta p_i = \sum_{j=i-k}^{j=i+k} \Delta p_j, \tag{4.1}$$

where k is the size of averaging window.

Next, Graph-based Bilateral Filter (GBF) is used as in [93] to smooth signal fluctuations. For a time series signal $\mathbf{y}$, a graph filtering can be realised by the optimisation proposed in [104]. By applying the bilateral filter operator $\mathbf{D^{-1}A}$ [105], an optimisation problem for GBF to signal $\mathbf{y}$ can be formulated as:

$$\min_{\mathbf{x}} \frac{1}{2}||\mathbf{x}-\mathbf{y}||_2^2 + \frac{1}{2}\alpha||\mathbf{x}-\mathbf{D}^{-1}\mathbf{A}\mathbf{x}||_2^2, \tag{4.2}$$

where α is a heuristically chosen trade-off parameter.

Eq. (4.2) can be solved by calculating the first derivative of the cost function with respect to the filter input [106] as:

$$\tilde{\mathbf{x}} = (\mathbf{I} + \alpha(\mathbf{I} - \mathbf{D^{-1}A}) * (\mathbf{I} - \mathbf{D^{-1}A}))^{-1}\mathbf{y}, \tag{4.3}$$

where $\tilde{\mathbf{x}}$ is the pre-processed signal.

Though GBF effectively smooths the signal, it may also split the sharp changes into multiple segments. Therefore, GBF is only applied after signal segmentation (introduced in the next section) is completed.

In the following to simplify exposition with a slight abuse of notation, we refer to $\mathbf{p}$ and $\mathbf{\Delta p}$ as pre-processed aggregate power and pre-processed power change value, respectively.

4.2.2 Signal Segmentation

The next step in the proposed method is to segment the aggregate power data and associate each segment to one graph node.

The idea is to group aggregate power measurements together if they belong to the same steady state. Data segmentation is achieved by comparing the value of the aggregate power change $|\Delta p_i| = p_{i+1} - p_i$ with a preset threshold $Thrs$. If the power change $|\Delta p_i| > Thrs$, p_{i+1} will be the first sample of a new segment. Otherwise, p_i belongs to the same segment as the previous sample.

We use $\mathcal{S}$ to represent the set of all segments, $\mathbf{S}_i$ the ith segment, which consists by a continues series of aggregated power measurements for the i-th steady state. The preset threshold $Thrs$ should be larger than the envisaged maximum variation during

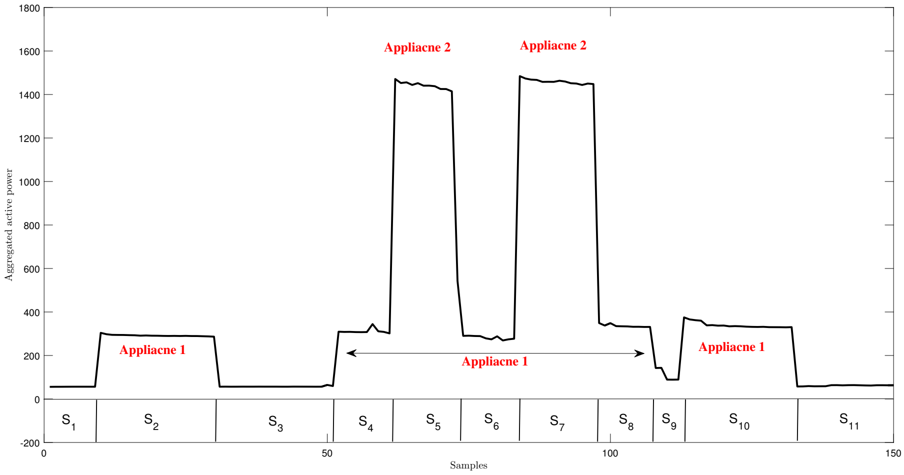


Figure 4.2: Example of signal segmentation for aggregate power measurement.

the same steady state but smaller than the minimum value of the state changes.

Starting from $i = 1$ and $j = 1$, signal segmentation then, assigns p_i to $\mathbf{S_j}$ and compares $|\Delta p_i|$ with $Thrs$. p_{i+1} is assigned to $\mathbf{S_j}$ if $|\Delta p_i| < Thrs$ or it is assigned to $\mathbf{S_{j+1}}$ otherwise. If p_{i+1} is assigned to $\mathbf{S_{j+1}}$, j is increased by $j = j + 1$. Then i is increased with $i = i+1$ and the above steps are repeated until all samples from aggregate power measurements $\mathbf{p}$ are assigned to a signal segment. Algorithm 3 demonstrates the pseudocode of data segmentation.

Algorithm 3: Data segmentation.

```
Input: $\mathbf{p}$, $\mathbf{\Delta p}$, $Thrs$, $N$, $p_1$ assigned to $\mathbf{S_1}$;
Output: $\mathcal{S}$;
set $i = 1$, $j = 1$ ;
while $i < N - 1$ do
    if $|\Delta p_i| < Thrs$ then
        assign $p_{i+1}$ to $\mathbf{S_j}$;
    else
        assign $p_{i+1}$ to $\mathbf{S_{j+1}}$, $j = j + 1$;
    update $i = i + 1$;
return $\mathbf{S}$;
```

Figure 4.2 shows an example of the signal segmentation. Two different appliances operate at the same time. The signal is segmented into 11 segments: $\mathbf{S}_1, \dots, \mathbf{S}_{11}$. Within each segment, all power samples are in the same steady state.

After the signal is segmented, clustering is applied to assign the segments into different clusters as explained next.

4.2.3 GLR-based Clustering

In this section, three unsupervised GSP (UGSP) clustering methods that differ in the way segments are defined and consequently the way the underlying graph is designed are introduced. The first approach segments the data and then resorts to DTW to build a graph. The other two methods do not use the segmentation approach described in Subsection 4.2.2, but instead rely on comparing fixed-length segments. Following the notation introduced in Chapter 2, a graph $\mathcal{G}= (\mathcal{V},\mathbf{A})$ is defined by graph nodes $\mathcal{V}$ and adjacency matrix $\mathbf{A}$.

Method 1 (UGSP-DTW)

Each node $v_i \in \mathcal{V}$ is assigned to a segment $S_i \in \mathcal{S}$. Since, $\mathbf{S_i}$ and $\mathbf{S_j}$ are vectors of different lengths, it is not appropriate to use Eq. (2.1) directly. Instead, DTW distance $DTWdist(\mathbf{a}, \mathbf{b})$ is used to evaluate the distance between sequences $\mathbf{a}$ and $\mathbf{b}$. DTW is one of the common used algorithms for measuring similarity between two sequences. Thus, the adjacency matrix of graph is defined as:

$$A_{i,j} = \exp\left\{ \frac{-DTWdist(\mathbf{S_i},\mathbf{S_j})^2}{\sigma^2} \right\}, \tag{4.4}$$

where σ is the scaling factor and is self tuned as proposed in Chapter 2, Section 2.4.

Similar to Chapter 3, the graph signal $\mathbf{s}$ is defined as the classification labels of data segments $\mathbf{S}$. Starting from the first sample of the graph signal $\mathbf{s}$, binary GSP-based classification is adopted to find all samples s_i, where $i > 1$, that belong to the same cluster as s_i. That is, first set $s_1 = 1$, and all remaining samples of graph signals are set to zero. Then GLR is adopted to measure the variation in graph signal $\mathbf{s}$ with respect to the underlying graph, with the objective to find values of all preset zero samples. The optimisation problem now is:

$$\mathbf{s}^* = \min_{\mathbf{s}} ||\mathbf{s}^\top \mathbf{L}\mathbf{s}||. \tag{4.5}$$

The closed-form solution to the optimization in (4.5) is [84], [83]:

$$\mathbf{s}^* = -\mathbf{L}^{\#}_{2:N+1,2:N+1} s_1 \mathbf{L}^\top_{1,2:N+1}, \tag{4.6}$$

where N is the number of samples in the set, $\mathbf{L}^{\#}_{2:N+1,2:N+1}$ is the pseudo-inverse of $\mathbf{L}_{2:N+1,2:N+1}$ and ${s_i}^* \in [0,1]$. If $s_i^* > T_s$, the i-th sample is assigned to the same cluster as the starting sample. Similarly to Chapter 3, set $T_s = 0.5$. Then, all clustered segments including the first one are removed and the above steps are repeated until all segments are assigned to a cluster.

Method 2 (UGSP_PCA-FIX and UGSP_GPCA-FIX)

An alternative approach is to reduce dimensionality of the segments leading to shorter fixed-length segments and estimate correlation of the newly generated segments using Euclidean distance. Principal Component Analysis (PCA) is a commonly used statistical procedure that uses an orthogonal transformation to convert a set of possibly correlated variables into a set of linearly uncorrelated variables called principal components [107], and maximise the sample variance. After data segmentation (Subsection 4.2.2), PCA is performed on each segment of samples to obtain the principle components. Since components of all segments are of the same length, the graph adjacency matrix is defined using the sum of Euclidean distances, akin to Eq. (2.1), by replacing DTW distance with Euclidean distance in Eq. (4.4). The remaining clustering and labelling steps are the same as in the proposed scheme.

We also provide results for Graph Principal Component Analysis (GPCA) inspired by [108]. In GPCA, for each data segment, we build a graph $\mathcal{G} = (\mathcal{V}, \mathbf{A})$, where each graph node $v_i \in \mathcal{V}$ is associated with a sample inside the data segment. The adjacency matrix $\mathbf{A}$ and the graph Laplacian matrix $\mathbf{L}$ are defined using Eq. (2.1) with Euclidean distance measure and Eq. (2.2), respectively. As with PCA, GPCA finds the new coordinates (i.e., principal components) of each data segment. The solution to the generalised eigenvalue decomposition problem shown in (4.7) provides the new parametrised data with the same dimension [108], [109].

$$(\mathbf{D} - \mathbf{A})\psi_k = \lambda_k \mathbf{D} \psi_k, \tag{4.7}$$

where ψ_k and λ_k are the kth eigenvector and eigenvalue of the graph Laplacian matrix, respectively. This is similar to the traditional PCA, but has an advantage that provides a more general model of the raw data. The GPCA method shares the same remaining steps as the PCA method.

Method 3 (UGSP-FIX and UGSP_OPT-FIX Methods)

The segmentation approach proposed in Algorithm 3, relies on a manually set threshold $Thrs$. To avoid that, UGSP-FIX and UGSP_OPT-FIX methods use fixed-length segments of duration T samples. Then, the adjacency matrix $\mathbf{A}$ is defined as:

$$A_{i,j} = \exp\left\{\frac{-\sum_{t=1}^{T}\omega_t(p_{i+t-l}-p_{j+t-l})^2}{\sigma^2}\right\}, \tag{4.8}$$

where $l = \frac{T+1}{2}$, T, the length of the segment, is assumed to be an odd number, and σ is the scaling factor, tuned as in [74]. ω_t, with $\sum_{t=1}^{T}\omega_t = 1$, is the weight of the tth sample, enabling the ability to assign different levels of importance to different segments. Note that $A_{i,j}$ is the averaged weighted sample-by-sample Euclidean distance between Segments i and j. In the low-complexity UGSP-FIX method ω_t is set to 1 for $t = \frac{T+1}{2}$, and zero otherwise. In UGSP_OPT-FIX method, values of ω_t are optimised via a full-search algorithm for $t = 1, \ldots, T$ [110], which is possible since T is kept small. The remaining GSP-based classification steps are the same as in Method UGSP-DTW.

Table 4.1 compares GSP-based NILM algorithms proposed in the last chapter and this chapter with four GSP-based NILM methods in [82], [94], [95], [96]. We use S, U and S-S to represent Supervised, Unsupervised and Semi-supervised methods respectively. In all cases, each feature, a scalar or a vector, is assigned to one graph node.

As the example in Fig. 4.2 shows, after GLR-based clustering, similar data segments are corrected clustered. For this idealised example, three clusters are obtained, that

Table 4.1: Comparison of the GSP-based NILM algorithms.

Method	Feature	S/U	Classification method	**A**
Chapter 3	Δp_i	S-S	GLR	Eq. (2.1)
Chapter 4 Method 1	$\mathbf{S_i} = p_t, ..., p_j$	U	GLR	Eq. (4.4)
Chapter 4 Method 2	Principal Components	U	GLR	Eq. (2.1)
Chapter 4 Method 3	$\mathbf{S_i} = p_{i-k}, ..., p_{i+k}$	S-S	GLR	Eq. (4.8)
[82]	Δp_i	U	GLR	Eq. (2.1)
[94]	p_i	S	GLR	Eq. (2.1)
[95]	Δp_i	S	GLR	Eq. (2.1)
[96]	$\mathbf{S_i} = p_t, ..., p_j$	S-S	GLR and MR	Eq. (2.1)

are cluster 1 including S_1, S_3, S_9, S_{11}, cluster 2 including $S_2, S_4, S_6, S_8, S_{10}$ and cluster 3 including S_5, S_7.

4.2.4 Cluster Labelling

After all data segments are grouped in to clusters, the clusters need to be assigned to correct labels, that is, appliances. For each cluster $\mathcal{C}_m$, calculate the average length $\overline{H_m}$ of all segments within the cluster is calculated. If $\overline{H_m}$ is smaller than Thr_H, the corresponding cluster $\mathcal{C}_m$ is assigned to be noise and belong to $\mathcal{L}_1$. Thr_H is the heuristically decided threshold of picking up noise clusters with short data segments.

Then $\overline{P_{\mathcal{C}_m}}$, the mean of active power values over all measurements inside each cluster $\mathcal{C}_m$, are compared with Thr_l and Thr_h, which are the lower and higher bound thresholds. Clusters with $\overline{P_{\mathcal{C}_m}}$ higher than Thr_h or $\overline{P_{\mathcal{C}_m}}$ lower than Thr_l are also labelled as $\mathcal{L}_1$. For the remaining clusters, $\overline{P_{\mathcal{C}_m}}$ are compared with the $\overline{P_n^l}$ for $n \in [1, K]$ where K is the total number of labels. For each preset label $\mathcal{L}_n$, $\overline{P_n^l}$ is the expected power value of that label which can be obtained using manufacturer information. These labels include each appliance is on individually and any multiple number of appliances are on at the same time. For multiple state appliances, each state is treated as an individual appliance. Cluster m is assigned to the label $\mathcal{L}_n$ if $|\overline{P_{\mathcal{C}_m}} - \overline{P^l_n}|$ is the minimum over all $n \in [1, K]$. Clusters with $|\overline{P_{\mathcal{C}_m}} - \overline{P^l_n}|$ larger than the preset threshold Thr_d will be assigned to unknown appliance which is also $\mathcal{L}_1$. The pseudocode of the cluster labelling method is shown in Algorithm 4.

Algorithm 4: Cluster labelling.

```
Input: $\mathcal{C}$, $\mathcal{S}$, $\mathcal{M}$, $Thr_H$, $Thr_h$, $Thr_l$, $Thr_d$, $\overline{\mathbf{P}^l}$, $K$;
Output: $\mathcal{L}$;
set $m = 1$ ;
while $m < |\mathcal{M}|$ do
    compute $\overline{H_m}$ as average length of all $\mathbf{S_j} \in C_m$;
    if $\overline{H_m} < Thr_H$ then
        assign $\mathcal{C}_m$ to $\mathcal{L}_1$;
    else
        compute $\overline{P_{\mathcal{C}_m}}$ as average power of all measurements in $\mathbf{S_j} \in C_m$;
        if $\overline{P_{\mathcal{C}_m}} > Thr_h$ or $\overline{P_{\mathcal{C}_m}} < Thr_l$ then
            assign $\mathcal{C}_m$ to $\mathcal{L}_1$;
        else
            find $n$ that minimises $|\overline{P_{\mathcal{C}_m}} - \overline{p^l_n}|$, for $n \in [1, K]$;
            if $|\overline{P_{\mathcal{C}_m}} - \overline{P^l_n}| > Thr_d$ then
                assign $\mathcal{C}_m$ to $\mathcal{L}_1$;
            else
                assign $\mathcal{C}_m$ to $\mathcal{L}_n$;
    update $m = m + 1$;
return $\mathcal{L}$;
```

After all clusters are assigned a label, the coherence of multiple states appliances is considered to further refine the labelling results. Any cluster with single occurred state from multiple states appliances that can not appear alone is assigned to the label n,only if n satisfies that $|\overline{P_{\mathcal{C}_m}} - \overline{P^n_l}|$ is the second smallest and is within threshold Thr_d. The average duration $\overline{H_m}$ of cluster $\mathcal{C}_m$ is also used as additional information to help improve the labelling accuracy. This is only applied for clusters that are labeled as single appliance is on. If $\overline{H_m}$ is much longer than the normal working duration of corresponding appliance, second minimal difference is used for revise the label.

4.3 Benchmarks

This section describes the schemes used as benchmarks.

4.3.1 Semi-supervised GSP

Semi-supervised GSP approach shares the same pre-processing and data segmentation steps as the proposed unsupervised GSP method. The difference comes after data segmentation. A graph-based binary classification (Chapter 3) is applied with Eq. (4.4) replacing Eq. (2.1). Training with a small amount of the ground truth labels known in advance is included in the semi-supervised GSP approach. GLR optimiser and SA are implemented as in Chapter 3. All data segments that are similar to the training data can be correctly detected. However, multiple appliances working at the same time will usually be regarded as noise or confused with other appliances.

4.3.2 DBSCAN

DBSCAN is a common used data clustering algorithm which does not require the number of clusters to be specified. DBSCAN is usually used for spacial point set clustering [111], [112]. Similar to Chapter 2, DBSCAN is applied instead of GSP based clustering as a benchmark. Data segments are regarded as points to be clustered. DTW distance is used to compute the distance between each data segment and further obtain the density of the segments. However, DBSCAN highly relies on tuning hyper-parameters. DBSCAN is not efficient for clustering appliances with low frequency usage.

4.3.3 Event-based GSP

The proposed methods are also compared with event-based unsupervised GSP of [93]. The same pre-processing steps are applied to the aggregated data. To make the comparison fair, the on and off states of each appliance are recovered after the rising and falling edge are detected. Then the same evaluation metrics are usedto compare the performance with proposed methods and all other benchmarks.

4.4 Results and Discussion

In this section, the experimental results are presented starting with Subsection 4.4.2 introduces performance measures and datasets. The next Subsection 4.4.3 demonstrates the performance of pre-processing followed by Subsection 4.4.4 compares three methods introduced in Section 4.2. The last Subsection 4.4.5 compares the proposed DTW-based unsupervised GSP method with benchmarks.

4.4.1 Parameter Setting

The parameter used in this section is set as shown in Table 4.2.

Table 4.2: Parameter settings for the proposed method.

symbol	parameter	setting
Thr_H	noise cluster threshold	5
Thr_d	maximum different threshold	$200W$
Thr_l	lower bound threshold	base-load+$10W$
Thr_h	higher bound threshold	$2\max\{p_m\}$

The parameters of the proposed method are set as following: $Thr_H = 5$ samples is the heuristically decided threshold of picking up noise clusters with short data segments. $Thr_d = 200W$ is the maximum difference allowed between the average power of a cluster and the expectation of labels. Thr_l is set slightly larger (by $10W$) than the base-load. Thr_h is the double of the maximum power of all known appliances in the house. The proposed method is not sensitive to the parameters provided in Table 4.2. The parameters are set to trade off performance and complexity. Changing these parameters by10-20% will not influence the accuracy of the results.

4.4.2 Performance Metrics and Datasets

As in the previous chapter, our proposed DTW based unsupervised GSP method and benchmarking methods are applied on house 1 & 2 in REDD and house 2 & 6 in REFIT datasets. These 4 houses are selected to include typical appliances and various noise levels [103].

We use F-measure over all samples to evaluate NILM disaggregation methods which is defined similar as Eq. (3.10), (3.11) and (3.12) in Chapter 3. However, in contrast to Chapter 3, here, for each appliance, TP is recorded when any sample is estimated to be the on state of the corresponding appliance and the appliance is actually on. FP is the number of samples that the off state of the appliance is misestimated as on. FN is the number of samples that the on state of the appliance is misestimated as off.

In the results tables, BGFI is used to label bathroom GFI, DW dishwasher, R refrigerator, KO kitchen outlet, MW microwave, WD wash dryer, S stove, F-F fridge freezer, WM washing machine, T toaster and K kettle. The results with the best performance are highlighted with bold numbers.

4.4.3 Pre-processing Gain Evaluation

The proposed pre-processing (Subsection 4.2.1) introduces additional complexity and delay. Hence, in this subsection, the proposed schemes with and without pre-processing are compared to estimate the gain obtained via the proposed pre-filtering. We denote R-SGSP-DTW and R-UGSP-DTW (where R = raw measurements) to represent DTW-based semi-supervised and unsupervised GSP methods, respectively, without pre-processing applied on the raw measurements.

Table 4.3 demonstrates the comparison of results provided by DTW-based semi-supervised and unsupervised GSP methods with and without pre-processing applied. We use bold red numbers to highlight the better results for each approach with and without pre-processing. It is clear that pre-processing provides great improvements for both methods over all appliances. Some appliance even doubles the accuracy after pre-processing is included. Appliance like KO already has relatively high accuracy and suffer less from the noise, but it shows slight improvements.

Table 4.3: Comparison of F-measure for two GSP-based methods with and without pre-processing applied for REFIT House 2.

	F-F	WM	DW	TV	MW	T	KO
R-SGSP-DTW	0.53	0.47	0.34	0.04	0.37	0.48	0.77
SGSP-DTW	**0.75**	**0.64**	**0.63**	**0.11**	**0.68**	**0.58**	**0.87**
R-UGSP-DTW	0.52	0.44	0.23	0.01	0.45	0.33	0.81
UGSP-DTW	**0.83**	**0.70**	**0.61**	**0.38**	**0.79**	**0.72**	**0.90**

4.4.4 Comparison Between Segmentation Methods

To quantify the gains from the proposed data segmentation, in this section the five proposed methods: UGSP-DTW, UGSP-FIX, UGSP_OPT-FIX, UGSP_PCA-FIX and

UGSP_GPCA-FIX as introduced in Subsection 4.2.3 are compared.

Table 4.4 shows the F-measure results of the five segmentation approaches for House 1 in the REDD dataset. For the fixed-segment methods, $T = 7$ to include 6 neighbouring samples. First, when comparing UGSP_OPT-FIX and UGSP-FIX, we can see that optimising weights provides negligible performance gain across all six appliances. Secondly, exploiting correlation and reducing dimensionality via PCA-based and GPCA-based methods, provides an additional performance gain over both fixed-segmentation methods. However, PCA-based or GPCA-based methods often do not capture well the structure of raw data leading to a large difference between two data segments with the same label. Finally, there is a noticeable and significant performance gain of the proposed variable-length segmentation UGSP-DTW over both fixed-segmentation methods and PCA-based and GPCA-based methods, indicating that DTW preserves some key information in the segments which PCA and GCPA removes during the transformation. Similar observations are made for the three other houses, as shown in Tables 4.6, 4.7 and 4.8.

Table 4.4: Comparison of F-measure for three methods introduced this chapter.

	BGFI	DW	R	KO	MW	WD
UGSP-FIX	0.33	0.41	0.37	0.39	0.45	0.24
UGSP_OPT-FIX	0.33	0.42	0.49	0.40	0.45	0.25
UGSP_PCA-FIX	0.45	0.62	0.61	0.55	0.67	0.61
UGSP_GPCA-FIX	0.55	0.69	0.70	0.61	0.72	0.63
UGSP-DTW	**0.65**	**0.77**	**0.92**	**0.86**	**0.74**	**0.73**

4.4.5 Comparison Against Benchmarks

In this section, we evaluate the robustness of the proposed UGSP-DTW NILM method against the benchmark algorithms described in Section 4.3, namely SGSP-DTW (DTW-based semi-supervised GSP method (Subsection 4.3.1)), UGSP_PCA-FIX (PCA-based unsupervised GSP method (Subsection 4.2.3)), UGSP_GPCA-FIX (GPCA-based unsupervised GSP method (Subsection 4.2.3)), DBSCAN-DTW (DTW-based DBSCAN method (Subsection 4.3.2)), and [82]'s UGSP. The results are given in Tables 4.5 and 4.6, for two REDD houses, and Tables 4.7 and 4.8 for two houses from the REFIT dataset.

Table 4.5: Comparison of F-measure for REDD House 1.

	BGFI	DW	R	KO	MW	WD
UGSP-DTW	**0.65**	**0.77**	**0.92**	**0.86**	0.74	**0.73**
SGSP-DTW	0.53	0.67	0.89	0.81	0.73	0.66
UGSP_PCA-FIX	0.45	0.62	0.61	0.55	0.67	0.61
UGSP_GPCA-FIX	0.55	0.69	0.70	0.61	0.72	0.63
DBSCAN-DTW	0.60	0.71	0.88	0.82	**0.81**	0.69
UGSP [82]	0.41	0.62	0.85	0.78	0.61	0.68

Table 4.6: Comparison of F-measure for REDD House 2.

	MW	WD	DW	R	S	KO
UGSP-DTW	**0.94**	**0.85**	**0.75**	0.88	**0.83**	**0.78**
SGSP-DTW	0.88	0.68	0.53	0.89	0.71	0.68
DBSCAN-DTW	0.81	0.68	0.69	**0.92**	0.70	0.63
UGSP_PCA-FIX	0.56	0.62	0.51	0.43	0.24	0.41
UGSP_GPCA-FIX	0.62	0.67	0.61	0.37	0.38	0.14
UGSP [82]	0.69	0.75	0.57	0.43	0.61	0.44

It is obvious that the proposed UGSP_state(DTW) method provides the best disaggregation accuracy for most appliances. DTW distance used for evaluating the similarity of two data segments provides convincing outcomes as shown by good performance of

Table 4.7: Comparison of F-measure for REFIT House 2.

	F-F	WM	DW	TV	MW	T	K
UGSP-DTW	**0.83**	**0.70**	0.61	**0.38**	**0.79**	**0.72**	**0.90**
SGSP-DTW	0.75	0.64	0.63	0.11	0.68	0.58	0.87
DBSCAN-DTW	0.77	0.61	0.62	0.20	0.45	0.61	0.82
UGSP_PCA-FIX	0.61	0.53	0.44	0.17	0.51	0.49	0.68
UGSP_GPCA-FIX	0.66	0.7	0.34	0.25	0.63	0.57	0.73
UGSP [82]	0.62	0.58	**0.64**	0.14	0.69	0.55	0.78

Table 4.8: Comparison of F-measure for REFIT House 6.

	F	K	T	DW	MW
UGSP-DTW	**0.88**	0.77	0.44	**0.69**	**0.70**
SGSP-DTW	0.81	**0.79**	**0.45**	0.57	0.63
DBSCAN-DTW	0.76	0.69	0.30	0.45	0.65
UGSP_PCA-FIX	0.72	0.67	0.39	0.58	0.54
UGSP_GPCA-FIX	0.75	0.66	0.34	0.52	0.59
UGSP [82]	0.46	0.74	0.37	0.43	0.57

all DTW-based approaches. However, Principal components analysis (PCA) or Graph Principal components analysis (GPCA) sometime lose the key structure information of raw data which makes the distance between two data segments with the same label large. That is why three DTW applied methods perform better than PCA and GPCA based methods. As introduced, since GPCA keeps more model information of the raw data, the F-measure results of GPCA based approach are slightly better than PCA based.

DBSCAN is a density based clustering which is highly depended on the size of each cluster and the minimum number of neighbour samples. Appliance with low working frequency, such as kettle, dish washer and microwave are harder to cluster together with DBSCAN. But appliances like refrigerator can be correctly clustered through DBSCAN.

Normally, the supervised and semi-supervised methods will have better performance than unsupervised methods. However, in this work, UGSP_s(DTW) provides slightly better results than SGSP_s(DTW). SGSP_s(DTW) applies binary classification appliance by appliance. Each sample is classified as on or off for all appliances. SGSP_s(DTW) is weak when multiple appliances are on at the same time. Also supervised methods highly rely on the quality of training dataset, while the proposed UGSP_s(DTW) method can self tuned parameter and cleverly label all the clusters, using further information obtained from each cluster. UGSP_e is established by detecting the correct ascription of each event. After recovering the on and off state after event-based method, any wrong estimation samples will cause serious dislocation of the state estimation, which dramatically increase FP and FN numbers.

As explained in Subsection 3.4.6, some appliances like TV in REFIT house 2 provides poor performance due to several reasons. Toaster (T) in REFIT house 6, is a short operating appliance, usually with an entire working duration to be no more than 3 minutes. In REFIT house 6, there is an unknown appliance which has similar working power and just a little longer duration compared to toaster. As a kitchen appliance, the toaster is also usually used together with other appliances. All these make the toaster in REFIT house 6 hard to identify.

4.5 Summary

In this chapter, a DTW-based unsupervised GSP method for NILM is proposed. The proposed method introduces signal segmentation on smart meter measured active power data. GSP-based clustering with GFT applied for self parameter tuning is used for load disaggregation. Comparing to GSP applied to NILM in Chapter 3, [82], [93], [94],

[95] and [96], the proposed method is the first one to link segments of smart meter measurements with graph nodes. Data segments contain more information than a single sample used in benchmarks. The experiments of load disaggregation on REDD and REFIT datasets show that the proposed method has better performance. Also, the proposed approach is an unsupervised method that does not need training which might have interference on residents.

Chapter 5

A Generic Optimisation-based Approach for Improving NILM

The previous two chapters introduced two specific GSP-based approaches for solving the low-rate NILM problem. Motivated by minimising the difference between estimations of NILM algorithms and the measured aggregated load without losing the benefits of existing NILM algorithms, in this chapter, novel post-processing approaches are proposed for improving the accuracy of existing NILM algorithms. The NILM solution post-processing is posed as an optimisation problem to refine the final NILM result using regularisation, based on the level of confidence in the original NILM output.

5.1 Introduction

Prior work on post-processing NILM output has focused on manually checking if the disaggregated result is within acceptable limits, e.g., [82], [95], or is proposed as part of a specific NILM algorithm, for example, in Chapter 3 (the SA algorithm improves the GLR-based approach), as well as in [30], [93], [113]. In contrast, in this chapter, a *generic* method is proposed to improve the accuracy after conventional NILM is applied, that does not require any manual intervention, by casting our post-processing problem as an optimisation problem that aims to minimise the distance between the sum of the disaggregated loads and the total measured consumption. A regularisation term is added to assign a weight based on confidence in the accuracy of the initial disaggregation result for each appliance.

The resulting optimisation problem is a boolean quadratic problem (combinatorial in nature) with zero-one type constraints that belongs to the class of NP-hard problems [114], [115]. By adapting recent convex optimisation methods [114], [116], three approaches are provided to solve the posed optimisation problem which trade off complexity and accuracy. The first two methods are based on relaxation which changes the constraint of the optimisation problem to a soft real value between zero and one: the first, more complex approach, uses norm-2 minimisation, while the second, reduces the complexity, by casting the problem as a norm-1-type minimisation (see, e.g., [116]). After the optimal solution is found, the result is projected back to 0 or 1, entry-wise. The third approach is based on Semi-definite Programming Relaxation (SDR) [114].

Our motivation for using SDR comes from its demonstrated high accuracy in many similar problems, such as electricity theft detection with smart meters [117], power flow optimisation in microgrids [118], and finding an approximate solution with FHMM NILM [119], where the quadratic programming relaxation of [30] is replaced by SDR to boost performance.

Optimisation methods have been used to address the NILM problem by minimising the difference between the power consumption of the detected events and all possible combinations of loads [22], [33], [120]. Though various optimisation methods have been proposed, e.g., integer programming [121] and genetic algorithms [122], noise, unknown loads, and similar appliance signatures render these approaches ineffective. Our proposed methods differ from the aforementioned approaches since optimisation is used to refine or *post-process* the NILM result by regularising the cost function with our *confidence* in the initial NILM estimate.

To the best of our knowledge, this is the first attempt to develop a generic approach that can be used to improve the output of NILM, yielding the following contributions:

- a novel modelling method of how one can incorporate systematically a first-pass NILM and post-processing into a common framework leading to a clear mathematical formulation;

- a novel adaptive parameter selection based on the level of confidence in the first-pass NILM output for each appliance;

- three proposed approximate solutions to the formulated optimisation problem based on convex relaxation and convex optimisation tools [114], [116];

- demonstrating improvements of the proposed generic post-processing methods on two state-of-the-art first-pass NILM methods using two public electrical load measurement datasets;

- demonstrating improvements of the proposed post-processing methods with four state-of-the-art post-processing methods, including the methods proposed in Chapter 3, as well as methods of [30] and [93];

- a detailed analysis, in terms of accuracy and complexity, of the suitability of the proposed and existing post-processing NILM methods for different appliance combinations contributing to the aggregate load.

The remaining of this chapter is organised as: Section 5.2 introduces the state of the art clustering methods followed by the problem formulation described in Section 5.3. The next Section 5.4 describes three proposed post-processing methods. Section 5.5 presents the experimental results of the proposed methods compared with state-of-the-art methods. The last Section 5.6 summaries the work of this chapter.

5.2 State-of-the-art

In this section, the state-of-the-art post-processing approaches are reviewed in order to identify the research gap and how our contributions address this.

Despite significant progress made in recent years to crack the low-rate NILM problem suited for widespread national smart metering programmes, in Chapter 3 and 4, and in prior work [53], [82], [103], [123], [124], [125], state-of-the-art low-rate NILM methods still do not demonstrate acceptable levels of accuracy, scalability and complexity necessary for widespread deployment.

Therefore most approaches have revolved around improving low-rate NILM accuracy and complexity. Event-based NILM, as discussed in the previous section, has gained significant traction for low-rate NILM due to its relatively lower complexity and robustness to noise compared to state-based methods.

Only recently has post-processing of NILM, that is, improving the accuracy of NILM output, been gaining attention as a way of leveraging on NILM advantages and improving accuracy in a targeted manner by observing the results of the first-pass of NILM.

While [82] and [95] manually check the appliance power level and operation time after disaggregation and retain only the estimates that are within expected limits, they do not apply any post-processing method to improve the estimated result. In Chapter 3, SA is used to minimise the difference between the sum of the estimated power of appliances and actual measurements. SA is a probabilistic technique for approximating the global optimum, which randomly searches around the starting points obtained from NILM and updates the result when improvements are observed in the objective function. Though SA can find the optimal result, it usually requires many iterations to converge to the global optimum. Moreover, often SA results in a local minimum, away from the globally optimal result.

Additive Factorial Approximate Maximum A Posteriori (AFAMAP) in Additive Factorial Hidden Markov Model (AFHMM) [30], [113] compares the observation (aggregate measurement) with the sum of disaggregated loads. Additionally, in [30], Branch and Bound (BNB) is used to refine the results of AFAMAP where the difference between the sum of estimated and aggregate power is minimised. BNB is a well-investigated method for discrete and combinatorial optimisation problems that partitions the solution into two branches and recalculates the objective function; depending on the obtained value, one branch is chosen to continue the partition until the optimal result is found. BNB finds the global optimum within an accuracy of ϵ, but is often slow. Indeed, the worst case complexity is comparable to the full search.

More recently, in [93], a Graph Signal Processing based Post-processing (GSP-P) method is proposed to refine the disaggregation results by matching the falling and rising edges obtained by edge detection. For each rising edge, a graph is generated and according to minimisation of the graph Laplacian regularisation over all possible candidate rising edges are picked. Due to high complexity, the method is used only for appliances that tend to be confused with other appliances with similar power levels.

Note that, the above post-processing methods are designed for specific NILM algorithms, whereas in this chapter, generic post-processing methods are proposed for improving the result of *any* event-based NILM approach. We cast the NILM post-processing problem as an optimisation problem with a regularisation term that depends on the output of the original NILM, and propose solutions to optimally and automatically tune the regularisation parameters by adapting three state-of-the-art convex optimisation methods: two relaxation-based methods and a third on SDR. These convex-based relaxed optimisation solutions (polynomial) are drastically smaller than solving the original NP hard problem. Complexity constraints are addressed through the proposed norm-2 and norm-1 approaches. We analyse how the different relaxations compare in terms of both complexity and accuracy for our NILM post-processing problem.

5.3 Problem Formulation

In this section, the optimisation problem is formulated with task to improve the disaggregation result already obtained by any NILM method.

The task of NILM is to estimate individual loads contributing to the aggregate meter data. Focusing on the most common case when the meter measures only active power, the aggregate reading of the meter at time sample j can be expressed as:

$$P_j = \sum_{m=1}^{M} P_j^m + n_j, \tag{5.1}$$

where $j = 1, \ldots, N$ and $m = 1, \ldots, M$ with $M = |\mathcal{M}|$ and $\mathcal{M}$ is the set of all known appliances in the house. In Eq. (5.1), P_j and P_j^m are the total aggregate power and power consumption of Appliance m at time sample j, respectively; n_j is the noise that includes measurement errors, base-load and all unknown appliances in the on state at time sample j. The NILM task is to estimate all P_j^m, given P_j.

Let $\Delta P_j = P_{j+1} - P_j$ be the change of the aggregate power signal. Let w be a threshold, such that if $|\Delta P_j| > w$ an event is detected, i.e., an appliance changed its state, e.g., switched on or off. Let E_i denote the i-th event, where $i = 1 \cdots, N_E$ with N_E being the total number of events in a processing window. We set the value of E_i to j if Event i is detected at time instant j. Once events are identified, these events are classified and appliance consumption determined.

Let $\alpha_i^m = 1$ if, after NILM, it is predicted that Event i is caused by Appliance m changing its state, and $\alpha_i^m = 0$, otherwise. When $\alpha_i^m = 1$, if $\Delta P_{E_i} > 0$, the detected edge is a rising edge, otherwise, it is a falling edge. Based on this rule, we set $S_j^m = 1$, if Appliance m is running at time sample j, or $S_j^m = 0$ otherwise. Note that, given α_i^m (which are results of NILM), the state S_j^m of Appliance m is predicted. For example, $\alpha_2^1 = 1$ and $\alpha_3^1 = 1$ indicate that Appliance 1 changed its state during Events 2 and 3. Suppose that $\Delta P_{E_2} > 0$ and $\Delta P_{E_3} < 0$, then the power change at Event 2 corresponds to a rising edge and Event 3 to a falling edge, indicating the time when the appliance was most likely switched on and off, leading to $S_j^1 = 1$, for time interval $j \in [E_2, E_3]$.

Given the average working power $\overline{P^m}$ of Appliance m, obtained by training on submetering data or using the appliance manual, the power consumed by this appliance at each time sample j can be estimated as $\hat{P}_j^m = \overline{P^m} S_j^m$. We can also express similarly estimated power signal change $\Delta \hat{P}_i^m = \overline{\Delta P^m} \alpha_i^m$ with $\overline{\Delta P^m}$ being the average of power change value for appliance m. Since the sampling rate of two datasets are all below 10 seconds/sample, only a single appliance ia assumed to trigger an event, that is an appliance changing the operating state.

If NILM is successful, the sum of estimated power consumption of all appliances should be close to the aggregate power, that is,

$$\sum_{j=1}^{N} |P_j - P_j^0 - \sum_{m=1}^{M} S_j^m \overline{P^m}|^2, \tag{5.2}$$

should be close to zero, where P_j^0 is the estimated base-load at time sample j. Following terminology of Chapter3, the above term are named as the fidelity term, which represents the difference between aggregate power without the base-load, i.e., $P_j - P_j^0$, and the sum of the loads estimated by NILM, i.e., $\sum_{m=1}^{M} S_j^m \overline{P^m}$.

The next logical step is to minimise Eq. (5.2) over all possible S_j^m. However, there are several reasons why minimising Eq. (5.2) is not a good idea. First, two appliances with similar working power $\overline{P^m}$ are difficult to be distinguished by minimising the fidelity term alone. Secondly, the fluctuations of power values around the mean $\overline{P^m}$ during the appliance operation is ignored. Thirdly, the sum of multiple appliance loads might be close to another load, leading to wrong minimisation. Finally, noise including measurement errors and unknown appliances is not taken into account.

Instead of minimising Eq. (5.2) over all possible solutions, we assume that a NILM method has been applied to lead to a solution S_i^{m*}, for which, the fidelity term is fixed as:

$$\sum_{j=1}^{N} |P_j - P_j^0 - \sum_{m=1}^{M} S_j^{m*}\overline{P^m}|^2. \tag{5.3}$$

Starting from S_i^{m*}, (see Chapter3), SA is used to minimise the fidelity term by updating S_i^{m*}. Note that SA usually updates only several appliance states (appliances with unique high working power) to correct misclassification of the employed NILM.

To improve the reliability and accuracy of this *post-processing* step, the disaggregation results are introduced as a regularisation term. Then the optimisation problem becomes:

$$\min_{S_j^m \in \{0,1\}} \sum_{j=1}^{N} |P_j - P_j^0 - \sum_{m=1}^{M} S_j^m\overline{P^m}|^2 + \sum_{j=1}^{N}\sum_{m=1}^{M} \lambda_m |S_j^m - S_j^{m^*}|^2 \tag{5.4}$$

where MN optimisation variables, S_j^m, take values from a discrete set (0 or 1), and $\lambda_m \geq 0$ is the weight of the regularisation term for Appliance m. (Here, again, S_j^{m*}, $j = 1, \cdots, N$, is the estimate obtained by an initial NILM method used.) In this optimisation set-up, the fidelity term shows how far the result is from the observation, while the regularisation term weights our confidence in the original NILM output. Large λ_m means we have more confidence in the results of the original NILM for Appliance m. Small λ_m means that we have less confidence in the NILM result, and put more weight in minimising the fidelity term. Note that λ_m is appliance dependent, to reflect the case that a NILM method has different accuracy for different appliances.

To reduce the computational complexity and considering that the NILM algorithm will provide edge detection results α^*, the objective function is modified as:

$$\min_{\alpha_i^m \in \{0,1\}} \sum_{i=1}^{N_E} ||\Delta P_{E_i}| - |\sum_{m=1}^{M} \alpha_i^m \overline{\Delta P^m}||^2 + \sum_{i=1}^{N_E} \sum_{m=1}^{M} \lambda_m |\alpha_i^m - \alpha_i^{m^*}|^2 \tag{5.5}$$

to only optimise for sample i when the events are detected. The minimisation here is with respect to MN_E discrete variables α_i^m taking values 0 or 1. Since N is usually large, while number of events N_E is much smaller (i.e., appliances are rarely switched on/off), this significantly reduces complexity, and largely eliminates noise and fluctuations during appliance operation.

Besides testing the original SA, as proposed in Chapter3, in Section 5.5, applying SA and BNB to find the values of α_i^m that minimise Eq. (5.5) starting from $\alpha_i^{m^*}$ is also included.

5.4 Proposed Solutions

In this section, three solutions to the optimisation problem Eq. (5.5) or a related problem (see Eq. (5.6)) are provided.

5.4.1 Problem Relaxation

Eq. (5.5) is a boolean (combinatorial) quadratic problem that is known to be hard to solve exactly [114]. To solve efficiently this optimisation problem, relaxation is introduced, that is, instead of being one or zero, α_i^m in Eq. (5.5) takes soft real-number values in the set [0, 1]. This way, the minimisation problem in (5.5) can be converted to a convex optimisation problem, which enables the use of known convex optimisation tools (a problem with convex quadratic cost and box constraints).

To solve Eq. (5.5), the infeasible path-following algorithm [126] is used based on two Newton steps per iteration, which always finds a non-negative solution, and is implemented in CVX, a package for specifying and solving convex programs [127], [128]. For large-scale problems, one can also implement other efficient methods such as [129].

After the above method is applied, and a solution $\alpha_i^m \in [0, 1]$ to the relaxed version of the problem in Eq. (5.5) is obtained, the obtained α_i^m is replaced with $\alpha_{i,final}^m = 1$ if $\alpha_i^m > 0.5$, and with $\alpha_{i,final}^m = 0$, otherwise. In other words, the solution is projected back to the discrete set $\{0, 1\}$.

We also consider an optimisation problem with a modified, ℓ_1-norm type regularisation; see [116] for a similar regularisation in a different, compressed sensing, context. The main motivation for this is that both the output of the NILM and post-processing NILM are usually expected to be sparse, i.e., they have many zeros and a few non-zero entries. (Appliances generally rarely change state.) It is well known that ℓ_1-type regularisation yields sparse solutions. This provides a motivation to attempt to improve the NILM post-processing by moving from ℓ_2-norm-type regularisation in Eq. (5.5) to

ℓ_1-norm-type regularisation in:

$$\min_{\alpha_i^m \in \{0,1\}} \sum_{i=1}^{N_E} ||\Delta P_{E_i}| - |\sum_{m=1}^{M} \alpha_i^m \overline{\Delta P^m}||^2 + \sum_{i=1}^{N_E} \sum_{m=1}^{M} \lambda_m |\alpha_i^m - \alpha_i^{m^*}|. \tag{5.6}$$

Note that the optimisation variable in Eq. (5.6) is the same as in Eq. (5.5), and the regularisation coefficients λ_m's remain the same.

Note that each time the objective function in Eq. (5.6) is calculated, $N_E M$ fewer multiplication operations are needed compared to Eq. (5.5).

5.4.2 Semi-definite Programming Relaxation (SDR)

SDR is a powerful, computationally efficient approximation technique for a class of combinational optimisation problems that finds a wide range of applications [114], [117], [118], [119]. To make our optimisation suitable for SDR, Eq. (5.5) are adopted as follows. First the new optimisation variable $z_i^m = 2\alpha_i^m - 1$ is introduced. Then the optimisation problem becomes:

$$\min_{z_i^m \in \{-1,+1\}} \sum_{i=1}^{N_E} |g_i - \sum_{m=1}^{M} z_i^m r_i^m|^2 + \sum_{i=1}^{N_E} \sum_{m=1}^{M} \lambda'_m |z_i^m - z_i^{m^*}|^2, \tag{5.7}$$

where $g_i = |\Delta P_{E_i}| - \frac{1}{2} \sum_{m=1}^{M} |\overline{\Delta P_i^m}|$, $r_i^m = \frac{\overline{\Delta P_i^m}}{2}$, $z_i^{m^*} = 2\alpha_i^{m^*} - 1$ and $\lambda'_m = \lambda_m/4$. Eq. (5.7) has the optimisation variable z_i^m, where z_i^m takes either value -1 or +1. Note that Problem Eq. (5.7) is equivalent to Eq. (5.5). For example, once the optimal value of z_i^m is obtained, α_i^m can be recovered as: $\alpha_i^m = 0.5(z_i^m + 1)$.

We now express Eq. (5.7) in the standard format of a boolean quadratic program, so that the SDR method [114] can be applied. To this end, a new $N_E M$ length vector is constructed with all elements from z_i^m in a fixed order $\mathbf{z} = [z_1^1\ ,\ z_2^1\ ,\ \ldots\ ,\ z_{N_E}^1\ ,\ z_1^2\ ,$ $z_2^2\ ,\ \ldots\ ,\ z_{N_E}^2\ ,\ \ldots\ ,\ z_1^M\ ,\ z_2^M\ ,\ \ldots\ ,\ z_{N_E}^M]^\top$. Then Eq. (5.7) is equivalently expressed as:

$$\min_{\mathbf{z}\in\{-1,+1\}^{N_E\times M}} \|\mathbf{Bz}-\mathbf{d}\|_2^2\,, \tag{5.8}$$

where $\mathbf{d} = \left[\mathbf{g}, \sqrt{\lambda_1'}\mathbf{z^1}^*, \sqrt{\lambda_2'}\mathbf{z^2}^*, ..., \sqrt{\lambda_M'}\mathbf{z^M}^*\right]^\top$ with $\mathbf{g} = [g_1, g_2, ..., g_{N_E}]$ and $\mathbf{z^{m*}} = \left[z_1^{m*}, z_2^{m*}, ..., z_{N_E}^{m\ *}\right]$, and

$$\mathbf{B} = \begin{bmatrix} \mathbf{r^1}\circ\mathbf{I} & \mathbf{r^2}\circ\mathbf{I} & \cdots & \mathbf{r^M}\circ\mathbf{I} \\ \sqrt{\lambda_1'}\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \sqrt{\lambda_2'}\mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \sqrt{\lambda_M'}\mathbf{I} \end{bmatrix} \tag{5.9}$$

where $\mathbf{r^m}$ is an $N_E \times N_E$ matrix with all elements in the ith row equal to r_i^m, $\mathbf{I}$ is an $N_E \times N_E$ identity matrix, and $\mathbf{0}$ is an $N_E \times N_E$ zero matrix. Here, $\circ$ is the Hadamard product operation of two matrices.

Following [114] (see [114] for more details on the derivation of the involved algorithmic steps), the problem of Eq. (5.8) is relaxed into the following problem:

$$\begin{aligned} &\min_{\mathbf{y}\in\mathbb{R}^{N_E M+1}} \{\mathbf{y}^\top\mathbf{G}\mathbf{y}\} \\ &s.t. \qquad y_j^2 = 1, j = 1, ..., N_E M + 1 \end{aligned} \tag{5.10}$$

where matrix $\mathbf{G}$ is given by:

$$
\mathbf{G} = \begin{bmatrix} \mathbf{B}^\top\mathbf{B} & -\mathbf{B}^\top\mathbf{d} \\ -\mathbf{d}^\top\mathbf{B} & ||\mathbf{d}||_2^2 \end{bmatrix}.
$$

Then the semi-definite relaxation of Eq. (5.10) is solved as:

$$
\begin{aligned}
\min \quad & tr\left(\mathbf{GY}\right) \\
s.t. \quad & \mathbf{Y} \geq 0, \quad \mathbf{Y} = \mathbf{Y}^\top \\
& \mathbf{Y}_{i,i} = 1, \forall i = 1, ..., N_E M + 1.
\end{aligned}
\tag{5.11}
$$

where $\mathbf{Y} = \mathbf{y}\mathbf{y}^\top$. Here, notation $\mathbf{Y} \geq 0$ means that matrix $\mathbf{Y}$ is Positive Semi-definite (PSD). Eq. (5.11) is a semi-definite program, and hence it can be efficiently solved. We solve it here numerically using CVX. Compared to the previous convex relaxation solving Eq. (5.5) or (5.6) with respect to $\alpha_i^m \in [0, 1]$, the optimisation variable size in Eq. (5.11) is squared. Hence, from a computational point of view, Eq. (5.5) or (5.6) is preferred for higher dimensions of $N_E \times M$. On the other hand, SDR may exhibit higher accuracy.

Once $\mathbf{Y}^*$ is obtained, the discrete $-1/+1$ variables z_i^m need to be recovered. This is achieved via the eigenvalue decomposition:

$$
\mathbf{Y}^* = \sum_{i=1}^{N_E M+1} \mu_i \mathbf{u_i} \mathbf{u_i}^\top, \tag{5.12}
$$

where, μ_i is the i-th largest eigenvalue of $\mathbf{Y}$, and $\mathbf{u_i}$ is the unit-norm eigenvector of $\mathbf{Y}^*$ that corresponds to the eigenvalue μ_i.

Next, an $N_E M + 1$ length vector y^* is set to $\sqrt{\mu_1}\mathbf{u_1}$, and an intermediate solution variable $\hat{y}$ is calculated as:

$$\hat{y_j} = \begin{cases} +1, & \text{if} \quad y_j^* > 0 \\ -1, & \text{otherwise} \end{cases} \quad \text{for} \quad j = 1, ..., N_E M + 1. \tag{5.13}$$

Since only the leading eigenvalue-eigenvector pair (u_1, μ_1) is needed, it is not necessary to perform the full eigenvalue decomposition, thus significantly reducing the computational cost of this algorithmic step. Then, $\hat{\mathbf{z}}$ is calculated as:

$$\hat{z_j} = \begin{cases} \hat{y_j}, & \text{if } \hat{y}_{N_E M+1} = 1 \\ -\hat{y_j}, & \text{if } \hat{y}_{N_E M+1} = -1 \end{cases} \quad \text{for} \quad j = 1, ..., N_E M. \tag{5.14}$$

and change $\hat{z}$ back to the event classifier $\hat{\alpha}$:

$$\hat{\alpha} = \frac{\hat{\mathbf{z}} + 1}{2}. \tag{5.15}$$

such that the final NILM output is:

$$\alpha_i^m = \hat{\alpha}_{(m-1) \cdot N_E + i}. \tag{5.16}$$

5.4.3 Adaptive Calculation of the Regularisation Term Weight

We recommend the following heuristic choice for tuning parameters λ_m:

$$\lambda_m = \frac{\theta^2}{\overline{\Delta P^m}^2} \frac{\beta}{\min\limits_{n \in [1,M], n \neq m} ||\overline{\Delta P^n}| - |\overline{\Delta P^m}||}. \tag{5.17}$$

Two parameters θ and β are used to balance the weight of the regularisation term.

We set $\theta = [\overline{\overline{\Delta P_{app}}}]$, where operand $[x]$ rounds a positive number x to the nearest power of 10 and $\overline{\overline{\Delta P_{app}}} = \frac{\sum_{m=1}^{M} |\overline{\Delta P^m}|}{M}$. Similarly, we set $\beta = [\overline{\overline{\Delta P_{all}}}]$, where $\overline{\overline{\Delta P_{all}}}$ is the average value of $|\Delta P_i|$. Note that the accuracy of the final result is not very sensitive to the choice of λ_m, hence θ is rounded to the nearest power of 10.

λ_m is inversely proportional to the appliance mean power, which implies that for high loads, more weight is put on the fidelity term, since these loads contribute to the total aggregate the most. For high loads m, the first term of Eq. (5.17) is smaller or very close to 1. Thus, the fidelity term is given more weight. On the other hand, if an appliance has very small $\overline{|\Delta P_m|}$, a larger value of λ_m indicates that the optimisation weight is on the original NILM approach, i.e., the NILM outputs are reliable.

If there is another appliance that has average power fluctuation similar to Appliance m, the denominator of the second term of Eq. (5.17) is small, which suggests larger λ_m. Then, more weight is put on the original NILM result, since optimising the fidelity term would not be able to separate these two appliances. Note that β is set to be close to the average value of aggregate power change for all events detected, including noise and unknown high load appliances, which are usually larger than θ. The second term of Eq. (5.17) will be much larger than 1, and hence ensures that the second term of Eq. (5.5) is of comparable size to the fidelity term.

5.4.4 Summary of the Proposed Algorithms

All three proposed solutions consist of two steps: 1) solving a convex optimisation problem; and 2) "projecting" the solution back to the corresponding 0-1 set. For Step 1, the algorithms guarantee to converge and to a global solution of the convex problem [114]. Step 2, a simple projection step, can always be performed. Hence, all our methods are guaranteed to be *stable* and always *converge* to the global solution of the corresponding convex relaxed problem and to a sub-optimal solution of the overall NP-Hard combinatorial problem [114].

Table 5.1 summarises the three proposed methods and compares their computational complexity per iteration [130]. The average number of iterations per window is obtained by averaging the number of iterations needed to get optimal solutions for all testing windows. SDR has the largest computational complexity, so in practice, higher computational cost and longer execution time are expected.

Table 5.1: Summary of the proposed post-processing algorithms.

Proposed NILM Post-processing	Objective Function	Computational Complexity	Average Num. Iterations
Norm2	Eq. (5.5)	$O(N_E M)$	25
Norm1	Eq. (5.6)	$O(N_E M)$	15
SDR	Eq. (5.11)	$O((N_E M + 1)^3)$	8

5.5 Results and Discussion

We apply the proposed methods to the output of two state-of-the-art event-based NILM approaches, GLR in Chapter3 and DT in [28]. We also use REFIT and REDD datasets for testing the methods. For the REDD dataset, the first week are used for training and the rest for testing. For the REFIT dataset, one month of data (April 2014) are chosen to test the performance and use the previous month's measurements for training. Two houses with typical appliances, and different 'noise level' [103], are chosen from each dataset. The total number of considered appliances M depends on the house and is between 6 and 10.

F_M is used as evaluation metric to assess the appliance classification accuracy, as in [82], [93], (in Chapter3), [29]. To assess energy disaggregation accuracy, Accuracy ($Acc.$) [55] is used and defined as:

$$Acc. = 1 - \frac{\sum_{j=1}^{N} |\hat{P}_j^m - P_j^m|}{2\sum_{j=1}^{N} P_j^m}. \tag{5.18}$$

To reduce complexity, the datasets are split into windows, which are independently processed. For NILM norm2 and norm1 methods, window size $N_E = 1000$ is used, and for NILM SDR $N_E = 100$ (due to the larger size of the optimisation variable, which is a 101×101 matrix in this case).

For benchmarking, BNB method [30] applied to Eq. (5.5), GLR method [93] (denoted by GSP-P) and two different SA methods are used: SA1 denotes the method of (in Chapter3), i.e., using SA to optimise the fidelity term only, and SA2 refers to the method where SA is applied to Eq. (5.5).

To compare the relative complexity of different post-processing approaches, the execution time of the proposed methods are displayed as SA1, SA2, BNB and GSP-P, in Table 5.2. For SA (Chapter3), Eq. (5.5) is optimised sample by sample to improve calculation efficiency, and each sample needs more than 300 iterations to converge to the minimum. It is clear that the proposed NILM norm1 converges faster than other methods. All three proposed post-processing approaches are faster than SA, BNB and GSP-P (except SDR for REDD houses). The average execution time of BNB is over 0.5sec per sample, which implies that roughly half an hour is needed to process one month of data from a single REFIT house (with 3000 events per month). The fastest proposed method, Norm1, only needs few seconds and the slowest of the proposed methods, SDR, needs no more than 3min to complete the same task.

Table 5.2: Average execution time per sample in [sec] for two REDD and two REFIT houses for seven post-processing methods.

	REDD House1	REDD House2	REFIT House2	REFIT House6
SA1	0.19	0.18	0.20	0.17
SA2	0.21	0.22	0.23	0.18
Norm2	0.009	0.008	0.008	0.008
Norm1	**0.001**	**0.001**	**0.002**	**0.003**
SDR	0.065	0.053	0.072	0.063
BNB	0.55	0.61	0.49	0.54
GSP-P	0.037	0.043	0.082	0.075

Next, the accuracy of the methods are compared. We use F to label fridge, BG bathroom GFI, K kettle, T toaster, DW dishwasher, MW microwave, WM washing machine, WD washer dryer, KO kitchen outlet, S stove, and AVG for the average accuracy across all these appliances.

Tab. 5.3 shows per-appliance results for House 2 from the REFIT dataset. Fig. 5.1 shows results for House 6, averaged over all considered appliances. In tab. 5.3, the best results are highlighted with bold red numbers. It can be seen that all post-processing methods improve the disaggregation result for all listed appliances with respect to the original NILM. The three proposed methods are more accurate than SA1 and SA2. NILM SDR has the highest accuracy for most appliances compared to the other methods. Since BNB finds the globally optimal solution of the optimisation problem it solves (to within an accuracy ϵ), it is not surprising that it has similar, or occasionally slightly better, performance than the proposed methods. Note that the optimal solution to the problem of Eq. (5.5) does not necessarily correspond to the most accurate result for the NILM problem. The optimisation problem Eq. (5.5) itself is part of the modelling approach and makes certain inherent assumptions. For example, each appliance is assumed to operate at its average power with no fluctuation, which may not ideally match actual sub-metered load measurements. This is an explanation of why NILM BNB does not always provide the best result in terms of *Acc*. However, the main problem of the BNB approach is its high execution time. GSP-P [93] provides certain improvements for some appliances, but its average is always worse than any of the three proposed methods.

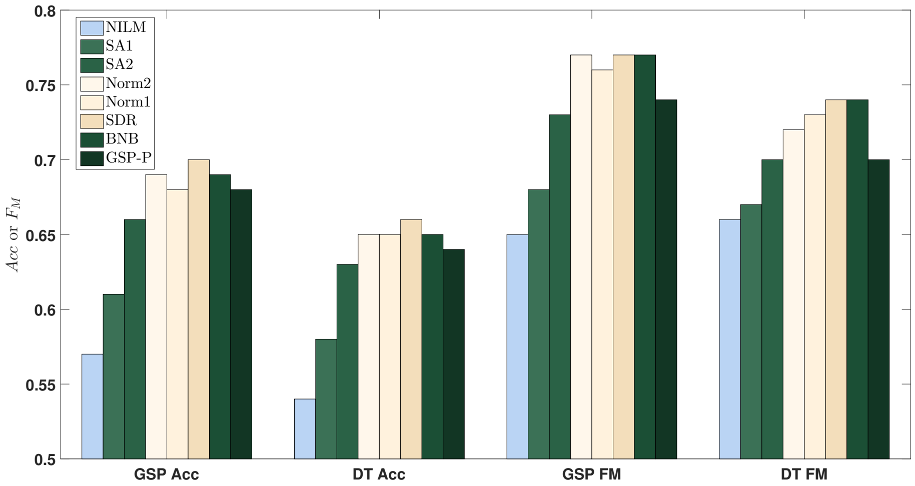


Figure 5.1: REFIT House 6 results: Average performance across all appliances.

Tab. 5.4 and Fig. 5.2 show the results for REDD Houses 1 and 2, respectively. Similarly to REFIT results, NILM SDR and BNB provide the best post-processing results for most appliances. For some appliances, such as KO in House 2, WD and F in House 1, NILM approaches without post-processing already have fairly accurate results

Table 5.3: Comparison of accuracy for REFIT House 2. NILM denotes the result without any post-processing, i.e., after GSP-based or DT-based NILM.

Appliances		F	K	T	DW	MW	AVG
Acc GSP	NILM	0.64	0.61	0.53	0.78	0.71	0.65
	SA1	-	0.79	0.68	0.78	0.71	0.72
	SA2	0.78	0.81	0.64	0.74	0.81	0.76
	Norm2	0.79	0.81	0.74	**0.79**	**0.83**	**0.79**
	Norm1	0.77	0.82	**0.76**	**0.79**	0.81	**0.79**
	SDR	**0.80**	**0.84**	0.75	0.77	0.81	**0.79**
	BNB	0.79	0.83	0.75	0.78	0.80	**0.79**
	GSP-P	**0.80**	0.67	0.60	0.78	0.73	0.72
Acc DT	NILM	0.55	0.63	0.44	0.70	0.72	0.61
	SA1	-	0.74	0.54	0.70	0.73	0.65
	SA2	0.69	**0.81**	0.64	0.74	0.81	0.74
	Norm2	0.72	0.79	0.67	0.70	**0.80**	0.74
	Norm1	0.71	**0.81**	0.70	0.70	**0.80**	0.74
	SDR	**0.73**	0.80	**0.72**	**0.73**	**0.80**	**0.76**
	BNB	**0.73**	**0.81**	0.71	**0.73**	**0.80**	**0.76**
	GSP-P	0.72	0.70	0.56	0.72	0.78	0.71
F_M GSP	NILM	0.61	0.80	0.60	0.75	0.79	0.71
	SA1	-	0.81	0.64	0.75	0.80	0.72
	SA2	0.61	0.81	0.64	0.74	0.81	0.72
	Norm2	0.63	**0.92**	0.71	**0.80**	0.85	0.78
	Norm1	**0.65**	**0.92**	0.72	**0.80**	0.83	0.78
	SDR	0.64	0.91	**0.73**	**0.80**	**0.86**	**0.79**
	BNB	0.63	**0.92**	**0.73**	**0.80**	0.85	**0.79**
	GSP-P	0.66	0.82	0.66	0.71	0.83	0.76
F_M DT	NILM	0.53	0.76	0.58	0.74	0.81	0.68
	SA1	-	0.81	0.64	0.74	0.81	0.71
	SA2	0.62	0.81	0.64	0.74	0.81	0.72
	Norm2	0.64	0.88	0.69	0.78	**0.83**	0.76
	Norm1	0.64	0.89	**0.70**	0.77	**0.83**	**0.77**
	SDR	0.64	**0.90**	**0.70**	**0.80**	0.82	**0.77**
	BNB	0.63	0.89	**0.70**	0.79	**0.83**	**0.77**
	GSP-P	**0.71**	0.77	0.69	0.73	0.81	0.75

and thus post-processing cannot improve much. For appliances such as BG and MW, which have unique and relatively high loads, significant improvement can be observed.

Table 5.4: Comparison of accuracy for REDD House 1. NILM denotes the result without any post-processing.

Appliances		BG	DW	F	KO	MW	WD	AVG
Acc GSP	NILM	0.51	0.71	0.93	0.84	0.69	0.86	0.77
	SA1	0.63	0.73	-	-	0.71	0.89	0.79
	SA2	0.73	0.78	0.93	0.85	0.72	0.95	0.83
	Norm2	0.72	**0.83**	**0.94**	**0.88**	0.71	**0.96**	0.84
	Norm1	0.74	0.82	0.93	0.86	0.74	0.95	**0.85**
	SDR	**0.75**	**0.83**	**0.94**	0.87	**0.76**	**0.96**	**0.85**
	BNB	**0.75**	0.82	0.93	0.87	**0.76**	**0.96**	**0.85**
	GSP-P	0.72	0.80	0.91	0.85	0.70	0.95	0.83
Acc DT	NILM	0.31	0.56	0.88	0.80	0.62	0.89	0.68
	SA1	0.57	0.68	-	-	0.58	0.90	0.74
	SA2	0.62	0.71	0.88	0.82	0.63	0.91	0.76
	Norm2	**0.69**	0.73	**0.89**	**0.83**	0.64	0.91	0.78
	Norm1	0.67	0.73	0.88	**0.83**	0.64	0.92	0.78
	SDR	**0.69**	**0.74**	**0.89**	**0.83**	**0.66**	**0.93**	**0.79**
	BNB	0.68	**0.74**	**0.89**	**0.83**	**0.66**	**0.93**	**0.79**
	GSP-P	0.56	0.63	0.85	0.80	0.63	0.90	0.72
F_M GSP	NILM	0.61	0.55	0.94	0.82	0.87	**0.96**	0.79
	SA1	0.62	0.56	-	-	0.87	**0.96**	0.80
	SA2	0.63	0.58	0.94	0.83	0.87	**0.96**	0.80
	Norm2	0.64	0.59	0.95	0.84	0.87	**0.96**	0.81
	Norm1	0.65	0.59	0.94	0.84	0.86	**0.96**	0.81
	SDR	**0.67**	**0.60**	**0.95**	**0.85**	**0.88**	**0.96**	**0.82**
	BNB	**0.67**	**0.60**	0.94	**0.85**	0.87	**0.96**	**0.82**
	GSP-P	0.63	0.58	0.93	0.84	0.87	**0.96**	0.80
F_M DT	NILM	0.43	0.44	**0.93**	0.78	0.87	0.63	0.68
	SA1	0.51	0.48	-	-	0.81	0.66	0.70
	SA2	0.54	0.49	**0.93**	0.79	0.86	0.68	0.72
	Norm2	0.55	0.52	**0.93**	0.80	0.85	0.67	0.72
	Norm1	0.55	0.51	**0.93**	0.80	0.85	0.67	0.72
	SDR	0.57	**0.53**	**0.93**	**0.82**	**0.88**	**0.69**	**0.74**
	BNB	0.56	**0.53**	**0.93**	0.80	0.87	0.68	0.73
	GSP-P	**0.59**	0.50	**0.93**	0.80	0.85	0.60	0.71

Based on the above results, the following conclusions can be made: 1. The gain with post-processing is lager for the noisier, REFIT dataset, since for the relatively clean,

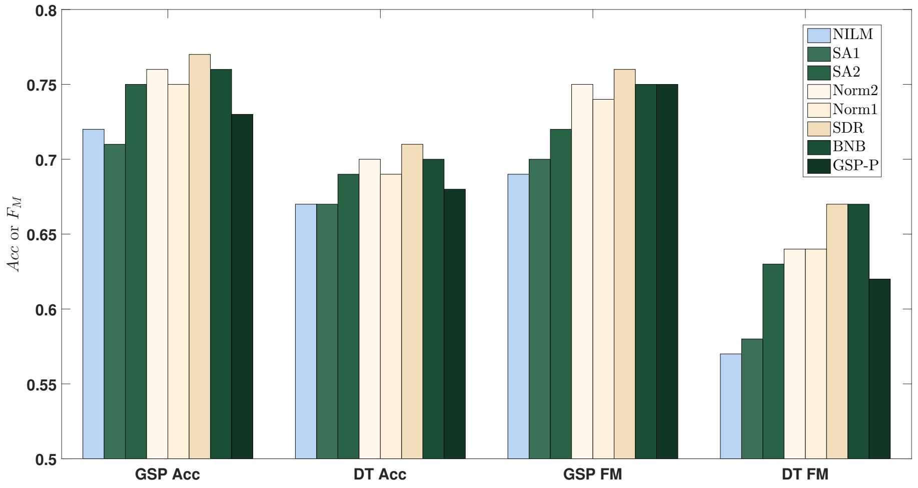


Figure 5.2: REDD House 2 results: Average performance across all appliances.

REDD dataset, NILM approaches without post-processing already provide good results. 2. If an appliance is disaggregated accurately with original NILM, then post-processing does not help much (e.g., F and WD in REDD House 1, K and MW in REFIT House 2). 3. The post-processing gains are similar for the two NILM algorithms (GSP and DT). 4. All the three proposed methods (as in Table 5.1) outperform SA and GSP-P methods, and exhibit (at least) comparable accuracy with respect to BNB, while significantly reducing the computational time.

5.6 Summary

In this chapter, three post-processing methods based on convex optimisation tools are introduced to improve the accuracy of NILM algorithms. The proposed methodology involves, as an intermediate step, a heuristic approach to solve a (combinatorial) boolean quadratic problem through relaxing zero-one constraint sets to compact zero-one intervals. SDR is applied to solve boolean quadratic problems with zero-one constraint sets. The three proposed approaches provide different trade-offs between performance and computational efficiency. The performance is compared with several post-processing NILM methods including the method of Chapter 3, methods of [30] and [93]. The experiments show that the proposed methods have better or similar performance to the benchmarks, but at much lower complexity.

Regarding the overall combinatorial (NP hard) problem, there is no estimate of 'quality' of the sub-optimal solution, i.e., there is no guarantee on how far is the solution from the global solution to the combinatorial problem, e.g., in terms of approximation accuracy. Note that, approximation accuracy guarantees have been established for some related problems [114]. The proposed methods, as add-ons to the existing NILM, slightly increase the overall complexity of the disaggregation module. If the employed NILM has a very poor result, it is unlikely that the proposed methods will lead to improvements. An interesting area of research is assessing the quality of the NILM output without relying on ground-truth to decide whether post-processing should be applied or not.

Chapter 6

Conclusion

6.1 Summary

As introduced in Chapter 1, low-rate NILM is still a very challenging problem. The field of NILM has seen a tremendous amount of research and novel approaches since NILM can provide individual residential loads information that improves customer energy feedback and reduces energy consumption waste. This thesis proposed several GSP-based approaches for solving NILM problems in smart meter data analysis.

In Chapter 2, general definitions of GSP are defined as an emerging signal processing concept that shows good potential in analysing data in time and space. Inside the chapter, the concepts of GTV, GLR, GFT and their potential applications are also introduced. The remaining part of the chapter demonstrates a case study that applied GLR-based clustering for eye tracker data analysis. In the case study, GFT is also applied as a visualisation tool to assess the quality of clustering.

In Chapter 3, a GSP-based classification that applies GLR with further refinements using SA is presented for NILM. Performance of the proposed method compared with HMM-based and DT-based methods on REDD and REFIT datasets are discussed. The proposed approach showed competitive results in both accuracy and efficiency. This NILM method addressed the large training overhead and associated complexity of conventional graph-based methods through a novel event-based GLR approach.

Then in Chapter 4, a novel GLR-based clustering method for NILM is demonstrated, which included data segmentation and DTW distance. Aggregated measurements of smart meters are first divided into data segments. Data segmentation is applied to make sure each data segment contains a complete steady-state. DTW distance is applied to evaluate the similarity of two data segments. Experimental results again show the competitiveness of the proposed method with respect to state-of-the-art approaches, and are demonstrated over the above two datasets with a range of appliances.

To further improve the disaggregation accuracy, in Chapter 5, three post-processing methods based on convex optimisation tools are proposed. The proposed methodology involves, as an intermediate step, a heuristic approach to solve a (combinatorial) boolean quadratic problem through relaxing zero-one constraint sets to compact zero-one intervals. SDR is applied in the third approach to solve boolean quadratic problems with zero-one constraint sets. The three proposed approaches provide different trade-offs between performance and computational efficiency. The experimental results show that the proposed methods have better or similar performance to the benchmarks, but at much lower complexity.

6.2 Future Work

Future work includes identifying in more detail the weaknesses of the proposed methods and providing an efficient real-time implementation of those algorithms integrating them into smart home decision support systems for demand response as well as designing advanced energy feedback mechanisms.

Another interesting area of research is assessing the quality of the NILM output without relying on ground-truth. As attempted, GFT is applied to evaluate the classification/clustering quality. In the future, this work can be extend to evaluate the NILM algorithms accuracy to further provide confidence of our results to customers.

Future work also includes expanding the proposed GSP-based approaches to very-low rate NILM, usually sampled at a rate of 10-60 minutes. Very-low rate NILM is attracting interest for the lower amount of data recording and exchanging required and smart meters deployed at scale in most countries tend to provide extremely low-rate measurements.

Another direction of future work is enlarging the set of features used for power disaggregation. These features could be reactive power, voltage/current and other measurements that might influence the using habit of the appliances, such as date, time and weather.

Bibliography

[1] M. Sira and V. N. Zachovalov, “System for calibration of nonintrusive load meters with load identification ability,” *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 6, pp. 1350–1354, June 2015.

[2] N. Uribe-Prez, L. Hernndez-Callejo, D. Vega, and I. Angulo, “State of the art and trends review of smart metering in electricity grids,” *Applied Sciences*, vol. 6, pp. 68–92, 02 2016.

[3] M. Lee, O. Aslam, B. Foster, D. Kathan, J. Kwok, L. Medearis, R. Palmer, P. Sporborg, and M. Tita, “Assessment of demand response and advanced metering,” *Federal Energy Regulatory Commission, Tech. Rep*, 2013.

[4] *My Energy Data*, Smart Grids Task Force. Ad hoc group of the Expert Group 1 Standards and Interoperability., Nov 2016.

[5] *Smart Metering Equipment Technical Specifications Version 2*, Department of Energy and Climate Change, UK, 2013.

[6] S. J. Pappu, N. Bhatt, R. Pasumarthy, and A. Rajeswaran, “Identifying topology of low voltage distribution networks based on smart meter data,” *IEEE*

Transactions on Smart Grid, vol. 9, no. 5, pp. 5113–5122, Sep. 2018.

[7] S. Chakraborty and S. Das, "Application of smart meters in high impedance fault detection on distribution systems," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 3465–3473, May 2019.

[8] M. M. Buzau, J. Tejedor-Aguilera, P. Cruz-Romero, and A. Gmez-Expsito, "Detection of non-technical losses using smart meter data and supervised learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2661–2670, May 2019.

[9] R. Moghaddass and J. Wang, "A hierarchical framework for smart grid anomaly detection using large-scale smart meter data," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 5820–5830, Nov 2018.

[10] B. P. Hayes and M. Prodanovic, "State forecasting and operational planning for distribution network energy management systems," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 1002–1011, March 2016.

[11] Y. Huang, L. Wang, W. Guo, Q. Kang, and Q. Wu, "Chance constrained optimization in a home energy management system," *IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 252–260, Jan 2018.

[12] C. Fischer, "Feedback on household electricity consumption: a tool for saving energy?," *Energy Efficiency*, vol. 1, no. 1, pp. 79–104, 2008.

[13] B. Neenan and J. Robinson, "Residential electricity use feedback: A research synthesis and economic framework," *technical report, Electric Power Research Institute*, 2009.

[14] K. C. Armel, A. Gupta, G. Shrimali, and A. Albert, "Is disaggregation the holy grail of energy efficiency? the case of electricity," *Energy Policy*, vol. 52, pp. 213 – 234, 2013, Special Section: Transition Pathways to a Low Carbon Economy.

[15] Y. Huang, L. Wang, W. Guo, Q. Kang, and Q. Wu, "Chance constrained optimization in a home energy management system," *IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 252–260, Jan 2018.

[16] D. Murray, "A data management platform for personalised real-time energy feedback," *Proc. 8th Int. Conf. Energy Efficiency Domestic Appl. Lighting (EEDAL)*, pp. 1–15, 08 2015.

[17] C. Su, W. Lee, and C. Wen, "Electricity theft detection in low voltage networks with smart meters using state estimation," in *Proc. 2016 IEEE International Conference on Industrial Technology (ICIT)*, March 2016, pp. 493–498.

[18] T. Erseghe and S. Tomasin, "Power flow optimization for smart microgrids by sdp relaxation on linear networks," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 751–762, June 2013.

[19] L. Stankovic, C. Wilson, J. Liao, V. Stankovic, R. H. Baldwin, D. Murray, and M. Coleman, "Understanding domestic appliance use through their linkages to common activities," in *Proc. 8th international conference on energy efficiency in domestic appliances and lighting*, 2015.

[20] H. Rashid, P. Singh, V. Stankovic, and L. Stankovic, "Can non-intrusive load monitoring be used for identifying an appliance's anomalous behaviour?," *Applied*

Energy, vol. 238, pp. 796–805, March 2019.

[21] G. W. Hart, Massachusetts Institute of Technology. Energy Laboratory, and Electric Power Research Institute, *Nonintrusive Appliance Load Data Acquisition Method: Progress Report*, MIT Energy Laboratory, 1984.

[22] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec 1992.

[23] S. B. Leeb, S. R. Shaw, and J. L. Kirtley, "Transient event detection in spectral envelope estimates for nonintrusive load monitoring," *IEEE Transactions on Power Delivery*, vol. 10, no. 3, pp. 1200–1210, July 1995.

[24] A. U. Haq and H. Jacobsen, "Prospects of appliance-level load monitoring in off-the-shelf energy monitors: A technical review," *Energies*, vol. 11, no. 1, 2018.

[25] H. Van Elburg, "Dutch energy savings monitor for the smart meter," *Rijksdienst voor Ondernemend Nederland*, 2014.

[26] C. Dinesh, B. W. Nettasinghe, R. I. Godaliyadda, M. P. B. Ekanayake, J. Ekanayake, and J. V. Wijayakulasooriya, "Residential appliance identification based on spectral information of low frequency smart meter measurements," *IEEE Transactions on Smart Grid*, vol. 7, no. 6, pp. 2781–2792, Nov 2016.

[27] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types," *Proceedings of the National Conference on Artificial Intelligence*, vol. 1, 01 2012.

[28] J. Liao, G. Elafoudi, L. Stankovic, and V. Stankovic, "Non-intrusive appliance load monitoring using low-resolution smart meter data," in *Proc. 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Nov 2014, pp. 535–540.

[29] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements," in *Proc. the 2011 SIAM Int. Conf. Data Mining*. SIAM, 2011, pp. 747–758.

[30] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," *J. Machine Learning Research - Workshop and Conf.e Proc.*, vol. 22, pp. 1472–1482, 2012.

[31] D. Egarter, V. P. Bhuvana, and W. Elmenreich, "Paldi: Online load disaggregation via particle filtering," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 2, pp. 467–477, 2014.

[32] M. Figueiredo, B. Ribeiro, and A. de Almeida, "Electrical signal source separation via nonnegative tensor factorization using on site measurements in a smart home," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 2, pp. 364–373, Feb 2014.

[33] J. Liang, S. K. K. Ng, G. Kendall, and J. W. M. Cheng, "Load signature studypart i: Basic concept, structure, and methodology," *IEEE Transactions on Power Delivery*, vol. 25, no. 2, pp. 551–560, April 2010.

[34] W. Wichakool, Z. Remscrim, U. A. Orji, and S. B. Leeb, "Smart metering of variable power loads," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 189–198, Jan 2015.

[35] C. Laughman, Kwangduk Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong, "Power signature analysis," *IEEE Power and Energy Magazine*, vol. 1, no. 2, pp. 56–63, March 2003.

[36] D. Srinivasan, W. S. Ng, and A. C. Liew, "Neural-network-based signature recognition for harmonic source identification," *IEEE Transactions on Power Delivery*, vol. 21, no. 1, pp. 398–405, Jan 2006.

[37] H. Y. Lam, G. S. K. Fung, and W. K. Lee, "A novel method to construct taxonomy electrical appliances based on load signaturesof," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 653–660, May 2007.

[38] S. R. Shaw, S. B. Leeb, L. K. Norford, and R. W. Cox, "Nonintrusive load monitoring and diagnostics in power systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 7, pp. 1445–1454, July 2008.

[39] S. N. Patel, T. Robertson, J. A. Kientz, M. S. Reynolds, and G. D. Abowd, "At the flick of a switch: Detecting and classifying unique electrical events on the residential power line (nominated for the best paper award)," in *Proc. UbiComp*, 2007.

[40] M. Hazas, A. Friday, and J. Scott, "Look back before leaping forward: Four decades of domestic energy inquiry," *IEEE Pervasive Computing*, vol. 10, no. 1,

pp. 13–19, Jan 2011.

[41] W. Wichakool, A. Avestruz, R. W. Cox, and S. B. Leeb, "Modeling and estimating current harmonics of variable electronic loads," *IEEE Transactions on Power Electronics*, vol. 24, no. 12, pp. 2803–2811, Dec 2009.

[42] K. D. Lee, S. B. Leeb, L. K. Norford, P. R. Armstrong, J. Holloway, and S. R. Shaw, "Estimation of variable-speed-drive power consumption from harmonic content," *IEEE Transactions on Energy Conversion*, vol. 20, no. 3, pp. 566–574, Sep. 2005.

[43] Y. Lin and M. Tsai, "Development of an improved timefrequency analysis-based nonintrusive load monitor for load demand identification," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 6, pp. 1470–1483, June 2014.

[44] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 76–84, February 2011.

[45] Y. Lin and M. Tsai, "Non-intrusive load monitoring by novel neuro-fuzzy classification considering uncertainties," *IEEE Transactions on Smart Grid*, vol. 5, no. 5, pp. 2376–2384, Sep. 2014.

[46] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey," *Sensors*, vol. 12, no. 12, pp. 16838–16866, 2012.

[47] J. M. Gillis, S. M. Alshareef, and W. G. Morsi, "Nonintrusive load monitoring using wavelet design and machine learning," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 320–328, Jan 2016.

[48] M. J. Johnson and A. S. Willsky, "Bayesian nonparametric hidden semi-markov models," *Journal of Machine Learning Research*, vol. 14, no. Feb, pp. 673–701, 2013.

[49] J. Z. Kolter, S. Batra, and A. Y. Ng, "Energy disaggregation via discriminative sparse coding," in *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, Red Hook, NY, USA, 2010, NIPS10, p. 11531161, Curran Associates Inc.

[50] S. Makonin, I. V. Bajic, and F. Popowich, "Efficient sparse matrix processing for nonintrusive load monitoring (nilm)," in *Proc. 2nd International Workshop on Non-Intrusive Load Monitoring*, 2014.

[51] S. Barker, S. Kalra, D. Irwin, and P. Shenoy, "Nilm redux: The case for emphasizing applications over accuracy," in *Proc. NILM-2014 workshop*. Citeseer, 2014.

[52] S. Giri and M. Bergés, "An energy estimation framework for event-based methods in non-intrusive load monitoring," *Energy Conversion and Management*, vol. 90, pp. 488–498, 2015.

[53] H. Altrabalsi, V. Stankovic, J. Liao, and L. Stankovic, "Low-complexity energy disaggregation using appliance load modelling," *Aims Energy*, vol. 4, no. 1, pp.

884–905, 2016.

[54] G. Lin, S. Lee, J. Y. Hsu, and W. Jih, "Applying power meters for appliance recognition on the electric panel," in *Proc. 2010 5th IEEE Conference on Industrial Electronics and Applications*, June 2010, pp. 2254–2259.

[55] J. Kolter and M. Johnson, "Redd: A public data set for energy disaggregation research," in *Proc. Workshop on data mining applications in sustainability (SIGKDD), San Diego, CA*, 2011, vol. 25, pp. 59–62.

[56] D. Murray, L. Stankovic, and V. Stankovic, "An electrical load measurements dataset of united kingdom households from a two-year longitudinal study," *Scientific Data*, vol. 4, pp. 160122, 01 2017.

[57] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, May 2013.

[58] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, April 2013.

[59] A. Ortega, P. Frossard, J. Kovaevi, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.

[60] A. Sandryhaila and J. M. F. Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular

structure," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 80–90, Sep. 2014.

[61] A. Sandryhaila and J. M. F. Moura, "Classification via regularization on graphs," in *Proc. 2013 IEEE Global Conference on Signal and Information Processing*, Dec 2013, pp. 495–498.

[62] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the 20th International conference on Machine learning (ICML-03)*, 2003, pp. 912–919.

[63] C. Yang, Y. Mao, G. Cheung, V. Stankovic, and K. Chan, "Graph-based depth video denoising and event detection for sleep monitoring," in *2014 IEEE 16th International Workshop on Multimedia Signal Processing (MMSP)*, Sep. 2014, pp. 1–6.

[64] W. Hu, G. Cheung, A. Ortega, and O. C. Au, "Multiresolution graph fourier transform for compression of piecewise smooth images," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 419–433, Jan 2015.

[65] C. Zhang, D. Florncio, and C. Loop, "Point cloud attribute compression with graph transform," in *Proc. 2014 IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 2066–2070.

[66] T. Maugey, Y. Chao, A. Gadde, A. Ortega, and P. Frossard, "Luminance coding in graph-based representation of multiview images," in *Proc. 2014 IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 130–134.

[67] P. Wan, G. Cheung, D. Florencio, C. Zhang, and O. C. Au, "Image bit-depth enhancement via maximum-a-posteriori estimation of graph ac component," in *Proc. 2014 IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 4052–4056.

[68] V. Stankovic, J. Liao, and L. Stankovic, "A graph-based signal processing approach for low-rate energy disaggregation," in *2014 IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES)*, Dec 2014, pp. 81–87.

[69] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Graph fourier transform," in *Proc. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 6167–6170.

[70] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Graph filters," in *Proc. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 6163–6166.

[71] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, June 2014.

[72] J. Pang and G. Cheung, "Graph laplacian regularization for image denoising: Analysis in the continuous domain," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1770–1785, April 2017.

[73] B. Motz, G. Cheung, A. Ortega, and P. Frossard, "Re-sampling and interpolation of dibr-synthesized images using graph-signal smoothness prior," in *Proc. 2015*

Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), Dec 2015, pp. 1077–1084.

[74] K. He, C. Yang, V. Stankovic, and L. Stankovic, "Graph-based clustering for identifying region of interest in eye tracker data analysis," in *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, Oct 2017, pp. 1–6.

[75] X. Chen and Z. Chen, "Visual attention identification using random walks based eye tracking protocols," in *2015 IEEE Global Conf. Signal and Inform. Proc. (GlobalSIP)*, Dec 2015, pp. 6–9.

[76] D. D. Salvucciand and J. H. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," in *The 2000 ACM Symp. Eye Tracking Research & Applications*, 2000, pp. 71–78.

[77] C. M. Privitera and L. W. Stark, "Algorithms for defining visual regions-of-interest: comparison with eye fixations," *IEEE Trans. Pattern Analysis and Machine Intell.*, vol. 22, no. 9, pp. 970–982, Sep 2000.

[78] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*, John Wiley & Sons, 2012.

[79] M. Ester, H. Kriegel, J. Sander, and X.Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *KDD*, 1996, vol. 96, pp. 226–231.

[80] O. Špakov and D. Miniotas, “Application of clustering algorithms in eye gaze visualizations,” *Inform. Techn. and Control*, vol. 36, no. 2, 2015.

[81] A. Santella and D. DeCarlo, “Robust clustering of eye movement recordings for quantification of visual interest,” in *The 2004 ACM Symp. Eye Tracking Research & Applications*, 2004, pp. 27–34.

[82] B. Zhao, L. Stankovic, and V. Stankovic, “On a training-less solution for non-intrusive appliance load monitoring using graph signal processing,” *IEEE Access*, vol. 4, pp. 1784–1799, 2016.

[83] C. Yang, Y. Mao, G. Cheung, V. Stankovic, and K. Chan, “Graph-based depth video denoising and event detection for sleep monitoring,” in *Proc. IEEE Int. Workshop Multimedia Signal Proc. (MMSP)*, Sept 2014, pp. 1–6.

[84] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge university press, 2004.

[85] R. Singh, A. Chakraborty, and B. S. Manoj, “Graph fourier transform based on directed laplacian,” in *2016 Int. Conf. Signal Processing and Commun. (SPCOM)*, June 2016, pp. 1–5.

[86] *Clustering datasets*, `https://cs.joensuu.fi/sipu/datasets/`.

[87] *The eye tribe*, `https://theeyetribe.com/`.

[88] *OGAMA open gaze and mouse analyzer*, `http://www.ogama.net/`.

[89] *Flaticons*, `https://www.flaticon.com/`.

[90] F. Paradiso, F. Paganelli, D. Giuli, and S. Capobianco, "Context-based energy disaggregation in smart homes," *Future Internet*, vol. 8, no. 1, pp. 4, 2016.

[91] M. Aiad and P. H. Lee, "Non-intrusive load disaggregation with adaptive estimations of devices main power effects and two-way interactions," *Energy and Buildings*, vol. 130, pp. 131–139, 2016.

[92] K. He, L. Stankovic, J. Liao, and V. Stankovic, "Non-intrusive load disaggregation using graph signal processing," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1739–1747, May 2018.

[93] B. Zhao, K. He, L. Stankovic, and V. Stankovic, "Improving event-based non-intrusive load monitoring using graph signal processing," *IEEE Access*, vol. 6, pp. 1–1, 2018.

[94] K. Kumar, R. Sinha, M. G. Chandra, and N. K. Thokala, "Data-driven electrical load disaggregation using graph signal processing," in *Proc. 2016 IEEE Annual India Conference (INDICON)*, Dec 2016, pp. 1–6.

[95] K. Kumar and M. G. Chandra, "An intuitive explanation of graph signal processing-based electrical load disaggregation," in *2017 IEEE 13th Int. Coll. Sig. Proc. its App. (CSPA)*, March 2017, pp. 100–105.

[96] D. Li and S. Dick, "Residential household non-intrusive load monitoring via graph-based multi-label semi-supervised learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 4615–4627, July 2019.

[97] J. Holweger, M. Dorokhova, L. Bloch, C. Ballif, and N. Wyrsch, "Unsupervised algorithm for disaggregating low-sampling-rate electricity consumption of households," *Sustainable Energy, Grids and Networks*, vol. 19, pp. 100244, 2019.

[98] J. Z. Kolter, S. Batra, and A. Y. Ng, "Energy disaggregation via discriminative sparse coding," in *Advances in Neural Information Processing Systems*, 2010, pp. 1153–1161.

[99] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*, MIT press, 2009.

[100] R. Raz, "On the complexity of matrix product," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, 2002, pp. 144–151.

[101] L. Ingber, "Simulated annealing: Practice versus theory," *Mathematical and computer modelling*, vol. 18, no. 11, pp. 29–57, 1993.

[102] D. L. Olson and D. Delen, *Advanced data mining techniques*, Springer Science & Business Media, 2008.

[103] S. Makonin, F. Popowich, I. V. Baji, B. Gill, and L. Bartram, "Exploiting hmm sparsity to perform online real-time nonintrusive load monitoring," *IEEE Trans. Smart Grid*, vol. 7, pp. 2575–2585, Nov 2016.

[104] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovacevic, "Signal denoising on graphs via graph filtering," in *Proc. 2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec 2014, pp. 872–876.

[105] A. Gadde, S. K. Narang, and A. Ortega, "Bilateral filter: Graph spectral interpretation and extensions," in *Proc. 2013 IEEE International Conference on Image Processing*, Sep. 2013, pp. 1222–1226.

[106] S. Chen, R. Varma, A. Singh, and J. Kovačević, "Signal representations on graphs: Tools and applications," *arXiv preprint arXiv:1512.05406*, 2015.

[107] H. Hotelling, "Analysis of a complex of statistical variables into principal components.," *Journal of educational psychology*, vol. 24, no. 6, pp. 417, 1933.

[108] K. Taylor, M. Procopio, C. Young, and F. Meyer, "Estimation of arrival times from seismic waves: A manifold-based approach," *Geophysical Journal International*, vol. 185, pp. 435 – 452, 04 2011.

[109] F. R. K. Chung and F. C. Graham, *Spectral graph theory*, Number 92. American Mathematical Soc., 1997.

[110] M. Brunig and W. Niehsen, "Fast full-search block matching," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 2, pp. 241–247, Feb 2001.

[111] P. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*, Pearson Education India, 2016.

[112] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*, Elsevier, 2011.

[113] M. Zhong, N. Goddard, and C. Sutton, "Latent bayesian melding for integrating individual and population models," in *Advances in Neural Inform. Processing Sys.*, 2015, pp. 3618–3626.

[114] Z. Q. Luo, W. K. Ma, A. M. C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Sig. Proc. Magazine*, vol. 27, pp. 20–34, May 2010.

[115] D. E. Knuth, "Postscript about np-hard problems," *SIGACT News*, vol. 6, no. 2, pp. 15–16, Apr. 1974.

[116] J. F. C. Mota, N. Deligiannis, and M. R. D. Rodrigues, "Compressed sensing with prior information: Optimal strategies, geometry, and bounds," *IEEE Trans. Inform. Theory*, vol. 63, July 2017.

[117] C.-L. Su, W.-H. Lee, and C.-K. Wen, "Electricity theft detection in low voltage networks with smart meters using state estimation," in *Proc. IEEE Int. Conf. Industrial Technology (ICIT)*, 2016, pp. 493–498.

[118] T. Erseghe and S. Tomasin, "Power flow optimization for smart microgrids by sdp relaxation on linear networks," *IEEE Trans. Smart Grid*, vol. 4, pp. 751–762, 2013.

[119] K. Shaloudegi, A. György, C. Szepesvari, and W. Xu, "Sdp relaxation with randomized rounding for energy disaggregation," in *Advances in Neural Inform. Proc. Sys. 29*, pp. 4978–4986. Curran Associates, Inc., 2016.

[120] A. Zoha, A. Gluhak, I. Muhammad, and S. Rajasegarar, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey," *Sensors*, vol. 12, pp. 16838–16866, 2012.

[121] K. Suzuki, S. Inagaki, T. Suzuki, H. Nakamura, and K. Ito, "Nonintrusive appliance load monitoring based on integer programming," in *SICE Annual Conf., 2008*. IEEE, 2008, pp. 2742–2747.

[122] M. Baranski and J. Voss, "Genetic algorithm for pattern detection in nialm systems," in *Proc. 2004 IEEE Int. Conf. Sys., Man and Cybernetics*, Oct 2004, vol. 4, pp. 3462–3468 vol.4.

[123] C. Zhang, M. Zhong, Z. Wang, N. Goddard, and C. Sutton, "Sequence-to-point learning with neural networks for nonintrusive load monitoring," *arXiv:1612.09106*, Dec. 2016.

[124] L. Mauch and B. Yang, "A novel dnn-hmm-based approach for extracting single loads from aggregate power signals," in *2016 IEEE Int. Conf. Acoustics, Speech and Sig. Proc. (ICASSP)*, March 2016.

[125] M. Pchacker, D. Egarter, and W. Elmenreich, "Proficiency of power values for load disaggregation," *IEEE Trans. Instrum. and Measurement*, vol. 65, pp. 46–55, Jan 2016.

[126] P. Tseng, "An infeasible path-following method for monotone complementarity problems," *SIAM J. Optimization*, vol. 7, pp. 386–402, 1997.

[127] M. Grant, S. Boyd, and Y. Ye, "Cvx: Matlab software for disciplined convex programming," 2008.

[128] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," *Recent advances in learning and control*, pp. 95–110, 2008.

[129] Y. H. Dai and R. Fletcher, "Projected barzilai-borwein methods for large-scale box-constrained quadratic programming," *Numerische Mathematik*, vol. 100, pp. 21–47, 2005.

[130] J. Wimp, "Pi and the agm: A study in analytic number theory and computational complexity (jonathan m. borwein and peter b. borwein)," *SIAM Review*, vol. 30, pp. 530–533, 1988.