Ref. Ares(2017)5319172 - 31/10/2017

D3.3 FIRST ENERGY DISAGGREGATION ALGORITHMS

Initial prototype, with documentation, of the algorithms for energy disaggregation into end uses

Project title	**Collaborative Recommendations and Adaptive Control for Personalised Energy Saving**
Project acronym	**enCOMPASS**
Project call	**EE-07-2016-2017 Behavioural change toward energy efficiency through ICT**
Work Package	**WP3**
Lead Partner	**SUPSI**
Contributing Partner(s)	**CERTH**
Security classification	**Public**
Contractual delivery date	**31/10/2017**
Actual delivery date	**31/10/2017**
Version	**1.0**

enCOMPASS D3.3 First Energy Disaggregation Algorithms

Version 1.0

Reviewers	**PDX**
	C. Schneider - SHF

History of changes

Version	Date	Comments	Main Authors
0.1	02/07/2017	DDP (Deliverable Development Plan) – definition of the document structure and contributions expected from each partner	A.E. Rizzoli, C. Rottondi
0.2	01/09/2017	Initial drafting of the whole document	A.E. Rizzoli, C. Rottondi
0.3	11/10/2017	Full description of the disaggregation algorithm and its performance	C. Rottondi, D. Piga, A.E. Rizzoli,
0.4	25/10/2017	Final revision and conclusions	C.Rottondi
0.5	26/10/2017	Added link to software availability and modified the type of deliverable	C. Rottondi
0.6	27/10/2017	Review of deliverable	C.Schneider
1.0	27/10/2017	Final corrections	C. Rottondi

Disclaimer

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 723059.

enCOMPASS D3.3 First Energy Disaggregation Algorithms

Version 1.0

TABLE OF CONTENTS

Executive Summary 6
1 Introduction 7
2 State of the art on energy end-use characterization 8
3 enCOMPASS algorithm for energy end-use disaggregation 9
3.1 Background on Quadratic Programming 9
3.2 Assumptions 9
3.3 Problem Statement 9
3.4 Standard least square estimation 10
3.5 Formulation of the disaggregation problem in terms of Quadratic Programming 12
3.6 Disaggregation Algorithm 13
3.6.1 Training 13
3.6.2 Solving the Disaggregation Problem 14
4 First Disaggregation Results 16
4.1 Dataset 16
4.2 Performance Metrics 17
4.3 Testing and Validation 18
5 Conclusions 21
6 Bibliography 22

EXECUTIVE SUMMARY

"D.3.3 First energy disaggregation algorithms" is specified in the enCOMPASS Description of Action as defining "the initial prototype, with documentation, of the algorithms for energy disaggregation into end uses".

This deliverable is one of the outputs of the project task "T3.4 Disaggregation of energy use", which aims at developing algorithms to disaggregate the building's energy consumption to the level of individual users and appliances. This task is fed with the information provided by T 3.1 and T 3.2: it uses the aggregate energy consumption at building level at different temporal resolutions, and it also uses the so called "signatures" of specific appliances, to disaggregate the total consumption into its components. Accuracy will depend on the quality and resolution of the available input data, but the objective of this task is to study a set of algorithms able to gracefully degrade their performance, to provide useful results under a wide spectrum of situations. The output of the algorithms allows for adaptive, in-situ feedback on energy consumption and recommendations for energy saving actions.

For an overall description of the dependencies among the above task T3.4 and the other Project tasks and work packages, please refer to the section 3.1.2 "detailed work description" of the enCOMPASS proposal.

This deliverable presents the algorithms developed in the enCOMPASS project to derive, directly from metered energy consumption data, mathematical models describing the users' consumption behavior. Specifically, the algorithms focus on energy end-use characterization, which aims at decomposing the aggregate (i.e., whole household) high-resolution energy flow data collected from a single measurement point into energy end use categories (e.g., washing machine, dishwasher), in order to understand how, when and where energy is used. The developed disaggregation algorithms are tested against data available in the literature or synthetically generated by open source software emulators of residential energy consumption traces.

The Java Source code has bene released and it is available at this link:

https://drive.switch.ch/index.php/s/bCw7DrvPs8Ae4dZ

This document is structured as follows:

- Section 1 describes the goals and motivations of end-use energy disaggregation approaches.
- Section 2 provides a review on the state-of-the-art algorithms for energy end-use disaggregation.
- Section 3 describes the novel algorithms for energy end-use characterization developed within the enCOMPASS project.
- Section 4 discusses the performance assessment of the algorithm by testing it on real energy consumption data.
- Section 5 provides conclusions and sketches future enhancements of the disaggregation algorithm.

1 INTRODUCTION

High spatial (household) and temporal (up to few seconds) resolution energy consumption data gathered by smart meters provide a detailed user consumption profile. This enables an accurate characterization of the energy consumption share and patterns of end-uses, which, in turn, constitute the basis for the mathematical modeling of individual and collective user behaviors.

Within the enCOMPASS project, Work Package 3 addresses energy end-use disaggregation, which aims at decomposing the aggregate (i.e., whole household) energy consumption data collected from a single measurement point into energy end use categories, to understand how, when and where energy is used. Beside using this information for building mathematical models of the user behavior, the generated knowledge can be also directly provided to customers, municipalities and energy utilities, so that:

1. household's components have a detailed knowledge on their energy usage. For instance, through the enCOMPASS platform, customers can log into a web page to view their hourly consumption, as well as charts on their energy end-uses patterns across major end-use categories (e.g., washing machine, dishwasher, cloth dryer, fridge) and they can be alerted of occurring consumption anomalies. Furthermore, personalized hints for reducing energy consumption can be directly delivered by means of the recommender system, that will be described in Deliverable D4.2 First user behavior modeler and recommender;
2. customers can be informed on potential savings in differing the usage of some energy using appliances (e.g., washing machine and dishwasher) to peak-off hours, or in replacing low-efficient appliances into high-efficient ones, and personalized rewards schemes can be then proposed to stimulate customers to adopt energy saving actions.

2 STATE OF THE ART ON ENERGY END-USE CHARACTERIZATION

There is a rich literature on automatic disaggregation methods (known as Non Intrusive Appliance Load Monitoring -NIALM- algorithms) aiming at decomposing the aggregate household energy consumption data collected from a single measurement point into device-level consumption data without requiring a limited interaction with the user. The first algorithm for NIALM was proposed by Hart in 1992 (Hart, 1992). Hart's approach is based on the segmentation of the aggregate power signal into successive steps, which are then matched to the appliance signatures. However, this method is not able to detect multistate appliances and it is neither able to decompose power signals made of simultaneous on/off events on multiple appliances. Since Hart's contribution, the problem of Nonintrusive Appliance Load Monitoring has been extensively studied in the literature. The survey papers (Zoha, Gluhak, Imran, & Rajasegarar, 2012) and (Zeifman & Roth, 2011) give a complete review on the state-of-the-art of NILAM methods, which can be classified into two main categories: optimization based and machine learning based approaches. The methods based on sparse coding (Figueiredo, Ribeiro, & de Almeida, 2013), (Dong, Ratliff, Ohlsson, & Sastry, 2013) and integer programming (Suzuki, Inagaki, Suzuki, Nakamura, & and Ito, 2008), (Camier, Giroux, Bouchard, & Bouzouane, 2013) belong the first category, while the approaches discussed in (Srinivasan, Ng, & Liew, 2006), (Zia, Bruckner, & Zaidi, 2011), (Parson, Ghosh, Weal, & Rogers, 2012), (Johnson & Willsky, 2013), which make use of Hidden Markov Models and Artificial Neural Networks belong to the second category. All of the aforementioned algorithms have generally shown good performance in estimating the fraction of energy consumed by each appliance, however most of them lack in skill in accurately reconstructing the power consumption trajectories over time. This represents a serious drawback, since: (i) no information on the time of use of each appliance can be derived, and so feedback on potential savings in differing the usage of some devices to peak-off hours cannot be provided; (ii) anomalous events, such as a device consuming an exceptional amount of power over an extended period, can be barely detected; (iii) it is not evident if the accuracy in the estimate of the fraction of energy consumed by each appliance is due to fortuitous balancing mechanisms.

Our proposed approach, fully described in (Piga, Cominola, Giuliani, Castelletti, & Rizzoli, 2016), exploits the assumption that the power demand profiles of each appliance are piecewise constant over time (as it is typical for energy use patterns of household appliances), and takes advantage of the information on the time-of-day probability in which a specific appliance is likely to be used. The disaggregation problem is treated as a least-square error minimization problem, with an additional (convex) penalty term aiming at enforcing the disaggregated signals to be piecewise constant over time. Besides being able to handle situations where multiple appliances are operating simultaneously, the proposed algorithm is able to reconstruct the consumption trajectories over time, thus overcoming the main drawback of the disaggregation methods available in the literature.

3 ENCOMPASS ALGORITHM FOR ENERGY END-USE DISAGGREGATION

This Section provides a description of the proposed disaggregation algorithm that can be used to decompose the aggregated energy consumption readings of a household in the consumption patterns of individual appliances. The Section is organized as follows: basic background notions and problem assumptions are reported in Sections 3.1 and 3.2, the problem of data disaggregation is formalized in Sections 3.3-3.5; the training procedure adopted to tune the algorithm parameters and the procedure used to solve the disaggregation problem are discussed in Section 3.6.

3.1 BACKGROUND ON QUADRATIC PROGRAMMING

Quadratic Programming aims at minimizing a (convex) quadratic cost function of several variables subject to linear constraints, i.e.,

$$\min_x \frac{1}{2} x^T H x + f^T x \qquad (3.1a)$$

$$s.t. Ax \leq b \qquad (3.1b)$$

$$A_{eq} x = b_{eq} \qquad (3.1c)$$

where $\mathrm{x} \in \mathbb{R}^{\mathrm{n}}$ is the set of optimization variables, $\mathrm{H} \in \mathbb{R}^{\mathrm{n,n}}$ is a positive semidefinite matrix and $\mathrm{f} \in \mathbb{R}^{\mathrm{n}}$ denote, respectively, the Hessian and the gradient of the objective function in 3.1a. The terms $\mathrm{A} \in \mathbb{R}^{\mathrm{m,n}}$ and $\mathrm{b} \in \mathbb{R}^{\mathrm{m}}$ define the linear inequality constraints on the variables x (with m being the number of inequality constraints). Similarly, $\mathrm{A_{eq}} \in \mathbb{R}^{\mathrm{m_{eq}},\mathrm{n}}$ and $\mathrm{b_{eq}} \in \mathbb{R}^{\mathrm{m_{eq}}}$ are used to define $\mathrm{m_{eq}}$ linear equality constraints on x.

3.2 ASSUMPTIONS

Our proposed disaggregation algorithm is based on sparse optimization and works under the following assumptions:

A. Each appliance can only operate at a single mode at each time instant.
B. A rough knowledge of the energy consumption of each appliance at each operating mode is supposed to be available. For instance, they can be estimated based on a training dataset by means of k-means clustering as proposed in (Likas, Vlassis, & and Verbeek, 2003).
C. The energy consumption profile of each appliance is piecewise constant over time (as it typically happens for many residential electrical appliances).

3.3 PROBLEM STATEMENT

Consider the situation where N different electrical appliances *i=1,..,N* are available in a house. Each appliance *i* is characterized by C_i operating modes. Let $B_i^{(j)}$ be the energy consumption of the i-th appliance at the *j*-th operating mode (with $j = 1,\dots, C_i$). Given the measure of the household-aggregated power consumption, $y(t)$, we define the consumption estimation of the *i*-th appliance $\hat{y}_i(t, \theta_i)$:

$$\hat{y}_i(t,\theta_i) = y_i(t) - e_i(t) = [B_i^{(1)} \ \dots \ B_i^{(C_i)}] \begin{bmatrix} \theta_i^{(1)}(t) \\ \dots \\ \theta_i^{(C_i)}(t) \end{bmatrix} \quad (3.2)$$

Where the vector θ_i contains the optimization binary variables related to each operating mode of the *i* -th appliance (their value should be either 0 or 1 and they must satisfy the constraint $\sum_{j=1}^{C_i} \theta_i^{(j)}(t) = 1$, i.e., each appliance works at a single operating mode at every instant *t*) and e_i models the noise that affects energy consumption measurements.

Given a sequence D_T of *T* observations of the aggregate energy consumption readings $y(t)$ (with *t=1,…,T*), our scope is the reconstruction of the individual energy usage patterns $y_i(t)$ (with *t=1,…,T*) of each electrical appliance based on the household aggregate energy consumption readings in the sequence D_T.

3.4 STANDARD LEAST SQUARE ESTIMATION

In order to estimate the energy consumption $y_i(t)$ of each appliance *i* at time *t*, the time varying parameters $\theta_i^{(j)}(t)$ can be computed by solving the standard least squares problem:

$$\min_{\substack{\theta_i^{(1)}(t),\dots,\theta_i^{(C_i)}(t) \\ i=1,\dots,N \\ t=1,\dots,T}} \left(\sum_{t=1}^{T} y(t) - \sum_{i=1}^{N} \hat{y}_i(t,\theta_i)\right)^2 \quad (3.3)$$

Where $\hat{y}_i(t,\theta_i)$ is defined by equation 3.2.

Unfortunately, the least-squares optimization problem (3.3) is overparametrized, since it involves more unknown parameters than measurements. As a consequence, overfitting occurs in computing the time varying variables $\theta_i^{(j)}(t)$ by means of the least squares approach. To overcome this problem, we must introduce regularization (or penalty) terms in (3.3) in order to impose that each appliance operates at a single mode at each time instant (assumption A) and that usage patterns are piecewise constant over time (assumption C).

Based on assumption A and on the fact that the variables $\theta_i^{(j)}(t)$ are binary, we can rewrite equation (3.3) as follows:

$$\min_{\substack{\theta_i^{(1)}(t),\dots,\theta_i^{(C_i)}(t) \\ i=1,\dots,N \\ t=1,\dots,T}} \left(\sum_{t=1}^{T} y(t) - \sum_{i=1}^{N} \hat{y}_i(t,\theta_i)\right)^2 + \lambda_1 \sum_{i=1}^{N} \sum_{t=1}^{T} \left\| \begin{bmatrix} \theta_i^{(1)}(t) \\ \dots \\ \theta_i^{(C_i)}(t) \end{bmatrix} \right\|_0 \quad (3.4)$$

s.t.

$$\sum_{j=1}^{C_i} \theta_i^{(j)}(t) = 1; \theta_i^{(j)}(t) \geq 0; \ i = 1, \dots, N; t = 1, \dots, T$$

Where $\| \quad \|_0$ indicates the 0-norm of a vector (i.e., number of nonzero elements). This way, the second term in formula (3.4) minimizes the number of nonzero elements in the vector $\left[\theta_i^{(1)}(t) \quad \dots \quad \theta_i^{(C_i)}(t)\right]$.

Moreover, constraint $\sum_{j=1}^{C_i} \theta_i^{(j)}(t) = 1$ imposes the presence of at least a nonzero element in the vector. The non-negative parameter λ_1 must be tuned (for instance by means of cross validation, see Section 3.6.1.3) to balance the tradeoff between minimizing the fitting error (when $\lambda_1$is low) and minimizing number of the nonzero elements in the vector $\left[\theta_i^{(1)}(t) \quad \dots \quad \theta_i^{(C_i)}(t)\right]$ (when λ_1 is high). Because of the 0-norm, problem (3.4) is nonconvex and thus difficult to solve with commercial numerical optimization solvers. However, an approximate solution of Problem (3.4) can be obtained by replacing the 0-norm with the (convex) 1-norm (i.e., sum of the absolute values of the elements of the vector). Furthermore, we can improve the estimation by multiplying the variables $\theta_i^{(j)}(t)$ by a vector of nonnegative weights $\left[w_i^{(1)}(t) \quad \dots \quad w_i^{(C_i)}(t)\right]$, thus leading to the following formula:

$$\min_{\substack{\theta_i^{(1)}(t),\dots,\theta_i^{(C_i)}(t) \\ i=1,\dots,N \\ t=1,\dots,T}} \left(\sum_{t=1}^{T} y(t) - \sum_{i=1}^{N} \hat{y}_i(t,\theta_i)\right)^2 + \lambda_1 \sum_{i=1}^{N} \sum_{t=1}^{T} \left\| \begin{bmatrix} w_i^{(1)}(t) \\ \dots \\ w_i^{(C_i)}(t) \end{bmatrix} \odot \begin{bmatrix} \theta_i^{(1)}(t) \\ \dots \\ \theta_i^{(C_i)}(t) \end{bmatrix} \right\|_1 \quad (3.5)$$

where $\odot$ indicates the element-wise multiplication. The choice of the weights $\left[w_i^{(1)}(t) \quad \dots \quad w_i^{(C_i)}(t)\right]$ will be discussed in Section 3.6.1.1.

Additionally, based on assumption C, we add a second regularization term as follows:

$$\min_{\substack{\theta_i^{(1)}(t),\dots,\theta_i^{(C_i)}(t) \\ i=1,\dots,N \\ t=1,\dots,T}} \left(\sum_{t=1}^{T} y(t) - \sum_{i=1}^{N} \hat{y}_i(t,\theta_i)\right)^2 +$$

$$\lambda_1 \sum_{i=1}^{N} \sum_{t=1}^{T} \left\| \begin{bmatrix} w_i^{(1)}(t) \\ \dots \\ w_i^{(C_i)}(t) \end{bmatrix} \odot \begin{bmatrix} \theta_i^{(1)}(t) \\ \dots \\ \theta_i^{(C_i)}(t) \end{bmatrix} \right\|_1 + \lambda_2 \sum_{i=1}^{N} \sum_{t=2}^{T} \left\| k_i \begin{bmatrix} \theta_i^{(1)}(t) - \theta_i^{(1)}(t-1) \\ \dots \\ \theta_i^{(C_i)}(t) - \theta_i^{(C_i)}(t-1) \end{bmatrix} \right\|_\infty \quad (3.6)$$

s.t.

$$\sum_{j=1}^{C_i} \theta_i^{(j)}(t) = 1; \theta_i^{(j)}(t) \geq 0; \; i = 1, \dots, N; t = 1, \dots, T$$

where λ_2 is a tuning parameter analogous to λ_1. The values k_i are a-priori specified nonnegative weights which can be set as described in Section 3.6.1.2. Note that the infinity norm of a vector (i.e., maximum absolute value among the elements of the vector) is used in the second regularization term. This way, only the largest time variation among the elements of the vector $\left[\theta_i^{(1)}(t) \quad \dots \quad \theta_i^{(C_i)}(t)\right]$ affects the cost function.

The problem formulated in equation (3.6) is a convex optimization problem, which can be solved by means of numerical optimization algorithms.

3.5 FORMULATION OF THE DISAGGREGATION PROBLEM IN TERMS OF QUADRATIC PROGRAMMING

It is necessary to convert the optimization problem (3.6) in a formulation analogous to that reported in equations 3.1. In order to obtain this matrix form, we proceed step by step with each part of the objective function. First, we rewrite the aggregation error as follows:

$$\left(\sum_{t=1}^{T} y(t) - \sum_{i=1}^{N} \hat{y}_i(t,\theta_i)\right)^2 = \|Y - \Phi\theta\|_2^2$$

Where the L_2 norm is used to compute the sum over time T; the vector θ (of size $\theta \cdot T$) contains the optimization variables (as explained before; the matrix Φ (of size $T \times \theta \cdot T$) contains the bases for every appliance at each time:

$$\Phi = \begin{bmatrix} \underbrace{\begin{matrix} B_1^T & 0 & 0 & 0 \\ 0 & B_1^T & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & B_{i=1}^T \end{matrix}}_{T} & \underbrace{\begin{matrix} B_1^T & 0 & 0 & 0 \\ 0 & B_1^T & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & B_{i=1}^T \end{matrix}}_{T} & \begin{matrix} \cdots \\ \cdots \\ \\ \cdots \end{matrix} \end{bmatrix}$$

Multiplying this matrix with the optimization variables vector θ, we obtain exactly $\sum_{i=1}^{N} \hat{y}_i(t,\theta_i)$.

The vector *Y* is the aggregated consumption $Y = [y_{(1)} \quad y_{(2)} \quad \cdots \quad y_{(T)}]$.

By means of the following algebraic steps:

$$\begin{aligned} \|Y - \Phi\theta\|_2^2 &= (Y - \Phi\cdot\theta)^T \cdot (Y - \Phi\cdot\theta) = \left(Y^T - \underbrace{(\Phi\cdot\theta)^T}_{(\Phi\cdot\theta)^T = \Phi^T\cdot\theta^T}\right)\cdot(Y - \Phi\cdot\theta) \\ &= (Y^T - \Phi^T\cdot\theta^T)\cdot(Y - \Phi\cdot\theta) \\ &= \theta^T\cdot\Phi^T\cdot\Phi\cdot\theta - \underbrace{\theta^T\cdot\Phi^T\cdot \mathrm{Y}}_{\theta^T\cdot\Phi^T\cdot\mathrm{Y} = (\theta^T\cdot\Phi^T\cdot\mathrm{Y})^T = \mathrm{Y}^T\cdot\Phi\cdot\theta} - \mathrm{Y}^T\cdot\Phi\cdot\theta + \mathrm{Y}^T\cdot\mathrm{Y} \\ &= \underbrace{\theta^T\cdot\Phi^T\cdot\Phi\cdot\theta}_{quadratic\ term} - \underbrace{2\mathrm{Y}^T\cdot\Phi\cdot\theta}_{linear\ term} + \underbrace{\mathrm{Y}^T\cdot\mathrm{Y}}_{constant\ term} \end{aligned}$$

It follows that $H = 2\Phi^T\cdot\Phi$ and $f = -2\mathrm{Y}^T\cdot\Phi$, whereas the constant term can be neglected for the minimization.

Then, we consider the sparsity term:

$$\lambda_1 \sum_{i=1}^{N}\sum_{t=1}^{T} \left\| \begin{bmatrix} w_i^{(1)}(t) \\ \cdots \\ w_i^{(C_i)}(t) \end{bmatrix} \odot \begin{bmatrix} \theta_i^{(1)}(t) \\ \cdots \\ \theta_i^{(C_i)}(t) \end{bmatrix} \right\|_1$$

The L_1 norm of a vector is the sum of the absolute value of its components. For this reason, we can ''substitute'' this part of the objective function with some constraints.

Assume we have the vector *v* with *n* elements: $\|v\|_\infty \Leftrightarrow \max \ (|v_1|, |v_2|, \ldots, |v_n|)$. We may write an equivalent constraint by introducing a slack variable *s* which, after the minimization, will assume the max value.

$$|v_i| \leq s \begin{cases} v_i \leq s \Leftrightarrow v_i - s \leq 0 \\ v_i \geq -s \Leftrightarrow -v_i - s \leq 0 \end{cases}$$

These inequalities must be added into matrices *A* and *b*, thus leading to the following expression of the objective function:

$$\min_{\substack{\theta_i^{(1)}(t),\dots,\theta_i^{(C_i)}(t) \\ i=1,\dots,N \\ t=1,\dots,T \\ s_1,\dots,s_{N(T-1)} \\ r_1,\dots,r_{N\cdot T}}} \left(\sum_{t=1}^{T} y(t) - \sum_{i=1}^{N} \hat{y}_i(t,\theta_i)\right)^2 + \lambda_1 \cdot w_1 \cdot r_1 + \dots + \lambda_1 \cdot w_{N\cdot T} \cdot r_{N\cdot T} + \lambda_2 \cdot k_1 \cdot s_1 + \dots + \lambda_2 \cdot k_N \cdot s_{N\cdot(T-1)}$$

The matrices *A* and *b* now contain $N \cdot \sum C_i \cdot (T-1)$ additional rows to include the L_1 norm inequalities, whereas the vector *f* contains $N \cdot (T-1)$ additional elements. The optimization variables are now θ, *r* and *s*.

Moreover, we rewrite the non-negativity constraints of variables θ as follows $-\theta_i^{(j)}(t) \leq 0$, thus leading to

$A = \begin{bmatrix} -1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -1 \end{bmatrix}$ and $b = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$.

Similarly, from the constraint $\sum_{j=1}^{C_i} \theta_i^{(j)}(t) = 1$, we obtain:

$A_{eq} = \begin{bmatrix} -1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -1 \end{bmatrix}$ and $b = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$.

3.6 Disaggregation Algorithm

In this chapter we explain how the disaggregation algorithm is trained and how the disaggregation problem is solved.

3.6.1 Training

According to assumption B, a training dataset $D'_{T'}$ is available. The training set consists of the energy consumption profiles of each appliance available in the house. An intrusive monitoring period is required to build the set $D'_{T'}$. During this period, the energy consumption pattern of each appliance is observed, and information on time-of-day probability characterizing the usage of each appliance/fixture can be also gathered.

Moreover, the training dataset is used to estimate the optimization weights $w_i^{(j)}(t)$ and k_i, as well as the weights λ_1 and λ_2, as discussed in the following subsections.

3.6.1.1 *Choosing the weights* $\mathbf{w}_i^{(j)}(\boldsymbol{t})$

The main criterion for the choice of the weights $w_i^{(j)}(t)$ is the following: if the *i*-th appliance is likely to operate at mode *j* at time *t*, then the variable $\theta_i^{(j)}(t)$ is likely to be equal to 1, whereas the remaining variables $\theta_i^{(g)}(t)$ (with $g \neq j$) are likely to be equal to 0. This means that the scaling weights $w_i^{(g)}(t)$ (with $g \neq j$) should be higher than $w_i^{(j)}(t)$. The information on time-of-day probability of the usage of each appliance can be inferred from the training dataset $D'_{T'}$. More in detail, for given *i* and *t*, the weights $w_i^{(1)}(t)$, …, $w_i^{(C_i)}(t)$ can be set as follows:

- compute the number of occurrences of the *i*-th fixture/appliance operating at mode *j* at the time samples $t+n\cdot 24h$, where $n=0,1,-1,2,-2,\ldots$ Denote the resulting integer as $q_i^{(j)}(t)$.
- If $q_i^{(j)}(t) \neq 0$, set the weight $w_i^{(j)}(t) = \frac{1}{q_i^{(j)}(t)}$ Otherwise, set$w_i^{(j)}(t) = 0$

Note that the weights $w_i^{(j)}(t)$ could be also computed in a more sophisticated way, e.g. by considering not only the observations at time t, t+24h, t-24h, t+48h, t-48h, … but also the observations (possibly weighted) within given time intervals [t+n·24h+Δ, t+n·24h-Δ].

3.6.1.2 *Choosing the weights* $\boldsymbol{k_i}(\boldsymbol{t})$

The weights k_i (with i=1,…,N) can be chosen as follows: if the i-th appliance changes its operating mode rarely over the time, than the time variation of the parameters $\theta_i^{(j)}(t)$ should be more penalized w.r.t. the time variation of the parameters characterizing another appliance which frequently changes its operating mode. The weight k_i can be then inversely proportional to the number of mode changes of the *i*-th appliance observed in the training dataset.

3.6.1.3 *Choosing the weights* $\boldsymbol{\lambda_1}$*and* $\boldsymbol{\lambda_2}$

In order to set the weights λ_1 and λ_2, a subset D'_{Tc} of length $T_c < T'$ is extracted from the original training dataset $D'_{T'}$ and used as calibration dataset. The values of λ_1 and λ_2, are then defined by means of a cross-validation procedure by minimizing with a grid search the Total Relative Mean Square Error (TRMSE) over the calibration dataset D'_{Cal}, where the TRMSE is defined as:

$$TRMSE = \sum_{i=1}^{N} \frac{\sum_{t=1}^{Tc}(y_i(t) - \hat{y}_i(t))^2}{\sum_{t=1}^{Tc} y_i^2(t)}$$

3.6.2 Solving the Disaggregation Problem

After computing the weights, we have to define the matrices *H*, *f*, *A*, *b*, A_{eq} and b_{eq} to be provided to the solver, by means of the equations reported in Section 3.5. Note that their size exhibit a linear dependency on the time horizon *T* and on the number of appliances (and their operational bases).

As the problem has been modelled as a convex function, in order to incorporate the inequality constraints in the objective function we adopt a logarithmic barrier function: such function forces the objective function in the space regions where constraints are not satisfied. Therefore, the objective function is modified as follows:

$$\min_{\substack{\theta_i^{(1)}(t),\dots,\theta_i^{(C_i)}(t)\\ i=1,\dots,N\\ t=1,\dots,T\\ s_1,\dots,s_{N(T-1)}\\ r_1,\dots,r_{N\cdot T}}} \left(\sum_{t=1}^{T} y(t) - \sum_{i=1}^{N} \hat{y}_i(t,\theta_i)\right)^2 + \lambda_1 \cdot w_1 \cdot r_1 + \dots + \lambda_1 \cdot w_{N\cdot T} \cdot r_{N\cdot T} + \lambda_2 \cdot k_1 \cdot s_1 + \dots + \lambda_2 \cdot k_N \cdot s_{N\cdot(T-1)} - \frac{1}{t}\sum \log(-Ax+b)$$

The problem is then solved by means of the Newton's method, which works as follows:

1. Start with a feasible point
2. Compute the Newton step Δx_{nt} by solving the following equation:

$$\begin{bmatrix} \nabla^2 j(x^{(k)}) & A_{eq}^T \\ A_{eq} & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta x_{nt} \\ \mu_{NS} \end{bmatrix} = \begin{bmatrix} -\nabla j(x^{(k)}) \\ 0 \end{bmatrix}, \text{ with } \lambda\left(x^{(k)}\right) = \sqrt{-\nabla j(x^{(k)})^T \cdot \Delta x_{nt}}$$

3. Perform backtracking line search (see details below)
4. Update $x^{(k+1)} = x^{(k)} + t \cdot \Delta x_{nt}$
5. Repeat steps 1-4 until Δx_{nt} matches the stopping criterion.

Where $x^{(k+1)}$ is the new candidate point, μ_{NS} is the dual optimal value, $\lambda\left(x^{(k)}\right)$ is the stopping criterion, $j(x)$ is our objective function, $\nabla j(x)$ is its first derivative, i.e.:

$$\nabla j(x) = H \cdot x + f - \mu \cdot \sum_i \frac{-1}{-A_i^T \cdot x + b_i} \cdot A_i$$

And $\nabla^2 j(x)$ is its second derivative, i.e.:

$$\nabla^2 j(x) = H - \frac{1}{t} \cdot \sum_i \begin{bmatrix} a_{i,1}^2 & a_{i,2} \cdot a_{i,1} & \cdots & a_{i,n} \cdot a_{i,1} \\ a_{i,1} \cdot a_{i,2} & a_{i,2}^2 & \cdots & a_{i,n} \cdot a_{i,2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i,1} \cdot a_{i,n} & a_{i,2} \cdot a_{i,n} & \cdots & a_{i,n}^2 \end{bmatrix} \cdot \frac{-1}{-A_i^T \cdot x + b_i^2}$$

In (unconstrained) minimization, a backtracking line search is a search scheme based on the Armijo–Goldstein condition, aimed at determining the maximum step size for a move along a given search direction. It starts with a relatively large estimate of the step size and iteratively shrinks it (i.e., "backtracking") until a decrease of the objective function is observed that approximates the local gradient of the objective function.

The search method works as follows: it initializes *t=1*, then given a descent direction Δx, while the inequality $f(x + t \cdot \Delta x_{nt}) > f(x) + \alpha \cdot T \cdot \nabla f(x) \cdot \Delta x_{nt}$ holds, t is updated as $t \leftarrow \beta \cdot t$. The parameters α and β are chosen within the intervals (0,0.5) and (0,1), respectively.

4 FIRST DISAGGREGATION RESULTS

The first disaggregation algorithm has been implemented in Java and a version is available here:

https://drive.switch.ch/index.php/s/bCw7DrvPs8Ae4dZ

4.1 DATASET

To assess the performance of our proposed disaggregation algorithm we use the AMPds dataset (Makonin, Popowich, Bartram, Gill, & and Bajic, 2013). It is available online and includes the energy consumption readings at one minute resolution collected from 21 breakers/loads of a single house located in the Vancouver region in British Columbia, Canada from April 1, 2012 to March 31, 2013, with sampling frequency of one sample per minute.

For the sake of analysis, we consider only the aggregate power consumption given by the sum of the power consumption readings of the following five electrical appliances: air conditioner (HPE), air heater (FRE), clothes dryer (CDE), fridge (FRE), and electronic workbench (EQE). These five appliances account for the largest contribution to the total energy consumption.

To evaluate the robustness of our algorithm to measurement noise, the aggregate power signal $y(t)$ has been corrupted by means of an additive zero-mean random Gaussian noise $e(t)$ with standard deviation σ=4 W. Note that, because of such added noise, the aggregate power consumption signal may become negative. When this happens, the power consumption signal is replaced by the value 0 W.

The AMPds dataset has been divided into two disjoint datasets:

- A training dataset $D'_{T'}$ containing the data for the days 1-15 June 2012, used to estimate the power demand of each appliance at each operating mode (i.e., the terms $B_i^{(j)}$) as well as the weights $w_i^{(j)}(t)$ and k_i through the procedure discussed in Sections 3.6.1.1 - 3.6.1.3. Moreover, in order to tune the parameters λ_1 and λ_2, a calibration dataset D'_{Cal} has been extracted from the original training dataset. Such a calibration dataset consists of the power readings from December 1, 2012 to December 15, 2012. Note that the sub-metered power consumptions of each appliance are supposed to be available in the training and calibration phase.
- The algorithm is validated on a portion of dataset extracted from the summer period and a portion from the winter period, since we expect seasonality to impact on the consumption pattern of the different end uses. In particular, a validation dataset D_T, which consists of the aggregate power readings from July 1, 2012 to July 31, 2012 (plotted in Figure 1) and from January 1, 2013 to January 31, 2013 (plotted in Figure 2) has been considered for the validation. In the validation phase, the sub-metered power consumption measurements are not supposed to be available and the aggregate power consumption signal is decomposed into the power consumption of each appliance by means of the proposed disaggregation algorithm. The sub-metered power consumption measurements are only used to assess the algorithm performance.

Note that, in order to evaluate the impact of different data granularities on the disaggregation performance, we also resampled the consumption patterns using coarser sampling frequencies (i.e., one sample every 2, 5, 10 or 15 minutes) thus including the data temporal resolution that will be available in the en-COMPASS pilot case studies (15minutes).

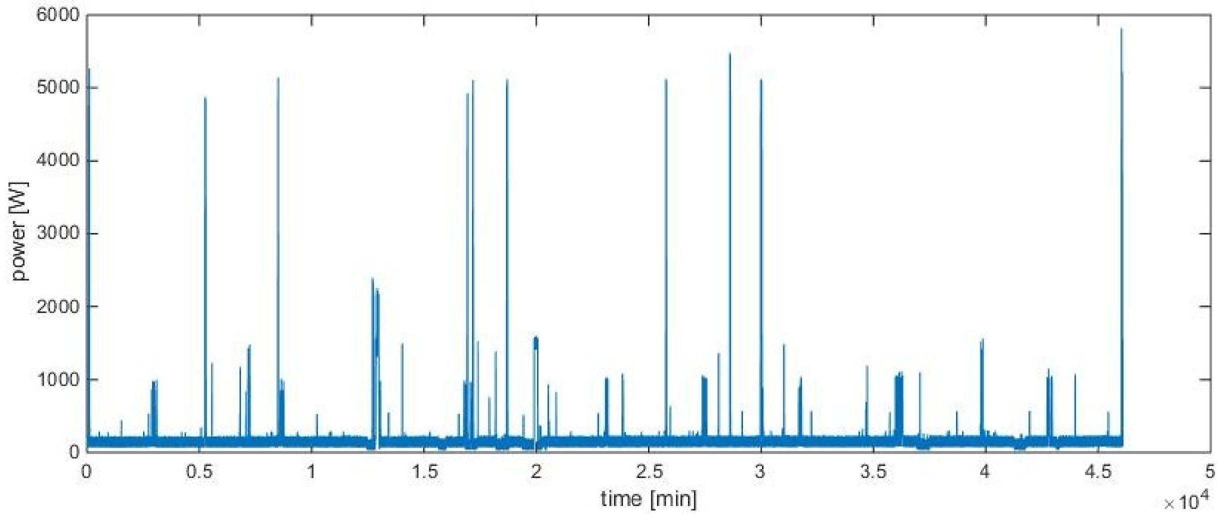

Figure 1: Electric power consumption from July 1, 2012 to July 31, 2012.

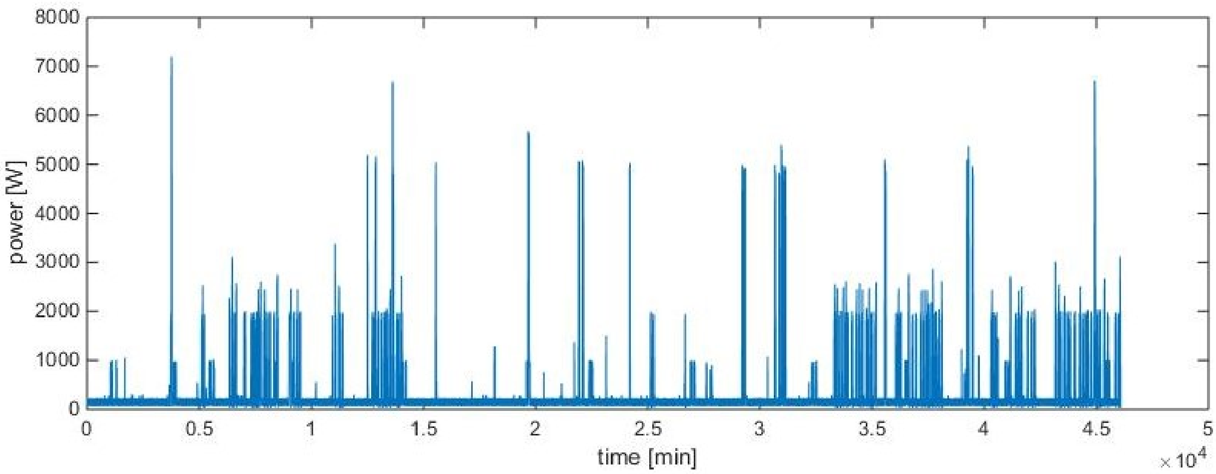

Figure 2: Electric power consumption from January 1, 2013 to January 31, 2013.

4.2 Performance Metrics

We consider the following performance metrics have been used to assess the performance of the proposed disaggregation algorithm:

The *Estimated Energy Fraction Index (EEFI)*, defined as:

$$\hat{h}_i = \frac{\sum_{t=1}^{T'} \hat{y}_i(t)}{\sum_{i=1}^{N} \sum_{t=1}^{T'} \hat{y}_i(t)}$$

The index $\hat{h}_i$ provides the fraction of energy assigned to the i-th appliance and should be compared to the *Actual Energy Fraction Index (AEFI)*, defined as:

$$h_i = \frac{\sum_{t=1}^{T'} y_i(t)}{\sum_{i=1}^{N} \sum_{t=1}^{T'} y_i(t)}$$

which computes the actual fraction of energy consumed by the *i*-th appliance.

The Relative Square Error (RMSE), defined as:

$$RSE_i = \frac{\sum_{t=1}^{Tc} (y_i(t) - \hat{y}_i(t))^2}{\sum_{t=1}^{Tc} y_i^2(t)}$$

which provides a normalized measure of the difference between the actual and the estimated power consumption of the *i*-th appliance.

Moreover, we compare the average AEFI computed over a yearly basis, since we consider yearly averaged data as benchmark on the disaggregation accuracy.

4.3 Testing and Validation

The proposed disaggregation approaches have been tested against the validation dataset D_T (i.e., July 2012 and January 2013). The performance metrics introduced in Section 4.2 and the estimated disaggregate power profiles are computed in order to assess the performance of the algorithms. Specifically:

- Table 3 shows the Relative Square Error (RSE) for each appliance for the two validation periods and energy consumption granularities ranging from 1 to 15 mins;
- Table 4 shows the Estimated Energy Fraction Index (EEFI) for each appliance, along with the Actual Energy Fraction Index (AEFI) for the two validation periods and energy consumption granularities ranging from 1 to 15 mins;
- Table 5 shows the Estimated Energy Fraction Index (EEFI) for each appliance, along with the Actual Energy Fraction Index (AEFI) computed over yearly basis, for the two validation periods and energy consumption granularities ranging from 1 to 15 mins.

The reported results show that the algorithm ensures a graceful performance degradation when coarsening the consumption data granularity. Moreover, the obtained disaggregation percentages are very close to the actual ones, with errors always below 6% when comparing EEFIs and AAFIs. Moreover, when comparing EEFIs to yearly-based AAFIs, the proposed disaggregation method shows consistently higher accuracy with respect to yearly averaged data (which reach errors up to 30% in the summer period).

Table 3: RSE values for each appliance computed for the two testing periods

Sampling rate (min)	winter					summer				
	HPE	FRE	CDE	FGE	EQE	HPE	FRE	CDE	FGE	EQE
1	0.1012	0.6394	0.0746	0.4076	0.0496	0.0557	0.1833	0.0142	0.1746	0.0498
2	0.1001	0.0594	0.1003	0.0594	0.0498	0.4323	0.2043	0.0750	0.1854	0.0591
5	0.2266	0.3102	0.2781	0.3114	0.0632	0.6837	0.2249	0.1289	0.1332	0.0581
10	0.2414	0.2657	0.2939	0.3394	0.063	1.553	0.2364	0.2865	0.149	0.0634
15	0.1945	0.3343	0.2679	0.3457	0.0641	1.4266	0.217	0.279	0.1684	0.0607

Table 4: EEFI vs AEFI (percentages) of each appliance computed for the two testing periods

Sampling rate (min)	HPE		FRE		CDE		FGE		EQE	
	AEFI	EEFI	AEFI	EEFI	AEFI	EEFI	AEFI	EEFI	AEFI	EEFI
winter										
1	54.84	53.79	5.15	5.53	16.06	15.88	12.59	15.54	11.37	9.26
2	54.82	55.82	5.18	4.30	16.09	15.24	12.55	15.79	11.36	8.84
5	55.07	62.40	5.16	3.26	15.74	10.42	12.66	15.22	11.37	8.70
10	54.77	62.38	5.08	3.06	16.20	10.75	12.63	15.16	11.32	8.65
15	55.19	61.58	5.30	3.20	15.11	11.11	12.85	15.31	11.55	8.79
Summer										
1	24.98	26.44	8.68	6.18	12.24	14.67	32.32	35.03	21.78	17.68
2	24.54	27.18	8.32	5.59	13.60	14.91	31.97	35.55	21.56	16.77
5	24.67	28.91	8.33	5.08	13.57	14.21	31.82	34.95	21.60	16.86
10	24.79	31.76	8.36	5.06	13.61	11.78	31.60	34.91	21.64	16.49
15	24.58	31.38	8.60	5.53	13.28	11.99	32.09	34.53	21.46	16.57

Table 5: EEFI vs yearly AEFI (percentages) of each appliance computed for the two testing periods

Sampling rate (min)	HPE		FRE		CDE		FGE		EQE	
	AEFI	EEFI	AEFI	EEFI	AEFI	EEFI	AEFI	EEFI	AEFI	EEFI
winter										
1	54.84	55.05	5.15	4.12	16.06	15.29	12.59	14.19	11.37	11.35
2	54.82		5.18		16.09		12.55		11.36	
5	55.07		5.16		15.74		12.66		11.37	
10	54.77		5.08		16.20		12.63		11.32	
15	55.19		5.30		15.11		12.85		11.55	
summer										
1	24.98	55.05	8.68	4.12	12.24	15.29	32.32	14.19	21.78	11.35
2	24.54		8.32		13.60		31.97		21.56	
5	24.67		8.33		13.57		31.82		21.60	
10	24.79		8.36		13.61		31.60		21.64	
15	24.58		8.60		13.28		32.09		21.46	

5 CONCLUSIONS

This document describes the disaggregation approaches implemented in the algorithm adopted by the enCOMPASS platform to decompose the overall energy consumption pattern of a building in single end-uses. The proposed approach has been validated over real energy consumption traces retrieved from a publicly available database and its performance has been evaluated for different granularities of the aggregated energy consumption measurements, showing that graceful degradation of the disaggregation results is achieved and that still accurate results can be obtained also in the case of data with 15-mins resolution (i.e. the data temporal resolution that will be available in the enCOMPASS pilot case studies).

As future work, the algorithm will be refined by leveraging the sensor data gathered by at the users' premises by means of the encompass sensor kit. More in detail, the following sources of information will be exploited to improve the accuracy of disaggregation results:

- Luminance measurements will be used to infer the on/off patterns of artificial lightening
- Internal temperature measurements will be exploited to make inferences about the use of air conditioning/heating plants
- Information gathered from motion sensors will be relied upon to make inferences about the use of certain categories of appliances (e.g., if the house is supposed to be empty, it is very unlikely that electric appliances such as microwave oven or kettles are active)
- Smart plugs capable of detecting on/off events (where available) will be used to identify the activity periods of single appliances
- The energy consumption curves of single appliances measured via smart plugs (where available) will be incorporated in the training process of the algorithm

6 BIBLIOGRAPHY

Camier, T. R., Giroux, S., Bouchard, B., & Bouzouane, A. (2013). Designing a NIALM in smart homes for cognitive assistance. *Procedia Computer Science, 19*, 524-532.

Dong, R., Ratliff, L., Ohlsson, H., & Sastry, S. (2013). A dynamical systems approach to energy disaggregation. *IEEE 52nd Annual Conference on Decision and Control.*

Figueiredo, M., Ribeiro, B., & de Almeida, A. (2013). On the regularization parameter selection for sparse code learning in electrical source separation. *Adaptive and Natural Computing Algorithms.*

Hart, G. (1992). Nonintrusive appliance load monitoring. *Proceedings of the IEEE, 80*(12).

Jama : A java matrix package. (2017, 09). Retrieved from http://math.nist.gov/javanumerics/jama/

Johnson, M., & Willsky, A. (2013). Bayesian nonparametric hidden semi Markov models. *The Journal of Machine Learning Research,, 14*(1).

Likas, A., Vlassis, N., & and Verbeek, J. (2003). The global k-means clustering algorithm. *Pattern recognition, 36*(2), 451-461.

Makonin, S., Popowich, F., Bartram, L., Gill, B., & and Bajic, I. (2013). AMPds: a public dataset for load disaggregation and eco-feedback research. *IEEE Electrical Power and Energy Conference (EPEC)*, (pp. 1-6).

Mysql: The world's most popular open source database. . (2017, 09). Retrieved from http://dev.mysql.com/

Parson, O., Ghosh, S., Weal, M., & Rogers, A. (2012). Non-Intrusive Load Monitoring using prior models of general appliance types. *AAAI Conference on Artificial Intelligence.*

Piga, D., Cominola, A., Giuliani, M., Castelletti, A., & Rizzoli, A. E. (2016, May). Sparse Optimization for Automated Energy End Use Disaggregation. *IEEE Transactions on Control Systems Technology, 24*(3), 1044-1051.

Srinivasan, D., Ng, W., & Liew, A. (2006). Neural-network-based signature recognition for harmonic source identification. *IEEE Transactions on Power Delivery, 21*(1).

Suzuki, K., Inagaki, S., Suzuki, T., Nakamura, H., & and Ito, K. (2008). Nonintrusive appliance load monitoring based on integer programming. *Proceedings of SICE Annual Conference.*

Zeifman, M., & Roth, K. (2011). Nonintrusive appliance load monitoring: Review and outlook. *IEEE Transactions on Consumer Electronics,, 57*(1).

Zia, T., Bruckner, D., & Zaidi, A. (2011). Hidden Markov Model based procedure for identifying household electric loads. *37th Annual Conference on IEEE Industrial Electronics Society,.*

Zoha, A., Gluhak, A., Imran, M., & Rajasegarar, S. (2012). Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey",. *Sensors, 12*(12).