

TAO Proyecto Final

Camilo Mariño

Diciembre 2020

1. Introducción

1.1. NILM

Non Intrusive Load Monitoring (**NILM**) o Monitoreo no Intrusivo de Cargas Eléctricas es la tarea de descomponer el consumo eléctrico de un hogar en el de sus distintos electrodomésticos. Es decir, a partir de una medida centralizada, típicamente tomada en el tablero general, se busca predecir los consumos de cada uno de los electrodomésticos (ver Figura 1).

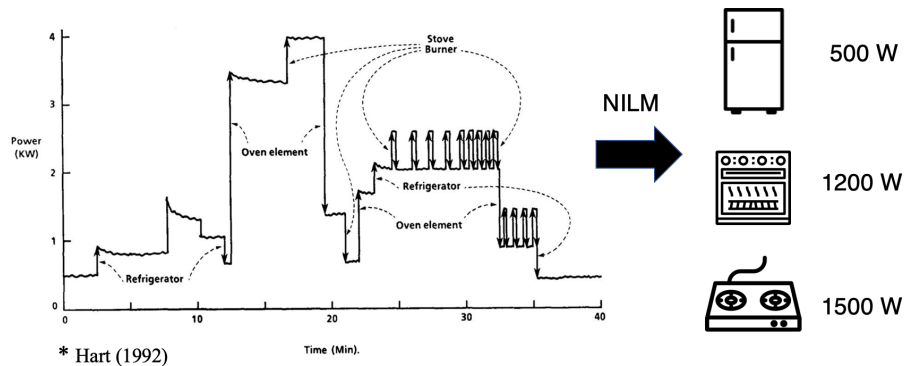


Figura 1: Esquema de desagregación de consumo. Imagen extraída de [1].

En el campo de NILM existen distintas variantes de las tareas a realizar. Algunas de ellas son:

- Desagregación de consumo: Asignar en una ventana de consumo agregado cuanta potencia consumió cada uno de los electrodomésticos.
- Detectar electrodomésticos: Determinar si un electrodoméstico está presente o no en una ventana.
- Reconstruir los consumos: Predecir la señal de potencia de cada uno de los electrodomésticos.

Muchas de estas de las tareas mencionadas están íntimamente relacionadas y no son independiente.

2. Objetivo

En este trabajo el objetivo fue desagregar y reconstruir los consumos. En las siguientes secciones se explica como se modela la realidad para tratarla como un problema de optimización, junto con sus supuestos y limitaciones. Luego, se presentan los métodos de optimización utilizados. Por último, se introduce los datos utilizados, métricas y resultados obtenidos.

3. Modelado

3.1. Definiciones previas

En primer lugar se definen algunos conceptos sencillos y comunes en cualquier problema NILM, pero que para un lector no familiarizado pueden ser confusos.

3.1.1. Señal agregada y desagregada

- Señal agregada: Señal de potencia/corriente que contiene la suma de todos los consumos. Típicamente esta señal es medida inmediatamente después del tablero general.
- Señal desagregada: Señal asociada a un único electrodoméstico.

3.1.2. Ventana de una señal

Dado que las señales de potencia/corriente son de largo arbitrario, la forma típica de abordar el problema es dividir estas en ventanas.

- Ventana: Tramo de la señal agregada o desagregada. Típicamente puede ser de un minuto, día, semana, etc.

Existen diversos criterios para determinar el tamaño de la ventana, además de que estos dependen fuertemente de la frecuencia de muestreo de las señales. Cuando se poseen datos a alta frecuencia ($1KHz$ o más) se suelen usar ventanas del orden de los segundos. Para tasas de muestreo intermedias (por ejemplo: una muestra cada 6 segundos) por lo general las ventanas son del orden de los minutos [2]. En cambio, si se trata de frecuencias más bajas, como lo son una muestra por minuto o más, se suele trabajar con ventanas de un día [3], o incluso una semana. En el contexto de este trabajo se utilizan **ventanas de un día**.

En el resto del documento, en el caso de tratarse de una ventana agregada, esta se denotará como \mathbf{x} . En tanto una ventana señal desagregada se denotará como \mathbf{d} ; en la literatura \mathbf{d} también es conocida como firma de un electrodoméstico, y cada electrodoméstico puede tener asociada varias firmas.

3.2. Modelado del problema

Como se mencionó en la introducción el problema de NILM consiste en dada una señal agregada \mathbf{x} obtener la señal desagregada para cada electrodoméstico. Una analogía típica a este problema es el *Cocktail Party*, en el que en una sala con un gran número de personas hablando simultáneamente, una persona pretende poder distinguir a quien corresponde cada una de las voces que escucha.

3.2.1. Enfoque básico

El enfoque más básico es modelar el problema como uno de mínimos cuadrados lineales.

$$\min_a ||x - Da||_2^2 \quad (1)$$

Donde \mathbf{x} es un vector que contiene la señal agregada. Se considera que $\mathbf{x} \in R^m$. Donde \mathbf{D} es llamada matriz de firmas o diccionario y tiene en cada columna un \mathbf{d} . Es decir, es una matriz en que cada columna es una ventana de consumo de un electrodoméstico. Se considera $\mathbf{D} \in R^{m \times T}$.

$$\mathbf{D} = \begin{pmatrix} | & | & & | \\ d_1 & d_2 & \dots & d_T \\ | & | & & | \end{pmatrix}$$

Mientras que \mathbf{a} , que es la variable de optimización, representa cuanto debería aportar cada \mathbf{d}_i para reconstruir la señal agregada.

Por lo tanto la Ecuación 1 pretende que cada consumo agregado se pueda descomponer como la suma ponderada de firmas de electrodomésticos.

Este enfoque presenta el gran problema de que algún elemento de \mathbf{a} sea negativo. Esto en la práctica representa que una firma aporte potencia negativa, por lo cual no es físicamente posible y se pretende esquivar en el modelado.

3.2.2. Enfoque con restricción

Por lo que se explicó, una restricción útil es forzar que \mathbf{a} sea no negativo. Es decir, que el problema de optimización sea:

$$\boxed{\min_{a \geq 0} ||x - Da||_2^2} \quad (2)$$

Este modelo permite una resolución más real, ya que la señal agregada es una suma ponderada positiva de firmas.

El inconveniente que presenta este enfoque es que no se limita cuanto puede aportar cada electrodoméstico a la señal. Esto es un problema porque \mathbf{D} está formado por más de una firma de cada electrodoméstico, generando que la señal pueda ser reconstruida exclusivamente por varias firmas de unos pocos electrodomésticos. Es decir, se podría reconstruir la señal agregada por medio únicamente de la suma de lámparas, ya que estas a priori pueden aportar la potencia/corriente que sea necesaria.

Para resolver este inconveniente, es posible añadir diversas regularizaciones sobre \mathbf{a} . En las siguientes secciones se abordan algunas de ellas.

3.2.3. Enfoque con regularización L-1

Consiste en añadir un término de penalización con la norma L-1 de \mathbf{a} .

$$\min_{a \geq 0} \|x - Da\|_2^2 + \beta \|a\|_1 \quad (3)$$

La intuición detrás de esta normalización, es generar que \mathbf{a} sea dispersa (varios elementos en 0). Esto evita que se reconstruya la señal por medio de pocas firmas y que no todas aporten.

3.2.4. Enfoque con regularización L-2

Igual que con L-1, pero con penalización L2 sobre \mathbf{a} .

$$\min_{a \geq 0} \|x - Da\|_2^2 + \beta \|a\|_2 \quad (4)$$

La idea detrás de esta regularización es que no haya elementos de \mathbf{a} demasiado elevados. Esta regularización hace que no se reconstruya la señal exclusivamente a partir de grandes aportes de algunas firmas.

3.2.5. Enfoque con regularización Elastic Net

Combina la regularización L-1 y L-2 agregando ambos términos en el problema de optimización.

$$\min_{a \geq 0} \|x - Da\|_2^2 + \beta_1 \|a\|_1 + \beta_2 \|a\|_2 \quad (5)$$

3.2.6. Enfoque con regularización Sum-to-k

Un enfoque más novedoso y específico para este problema es el que se propone en [4]. Para entenderlo es necesario recordar que \mathbf{D} está formado por T firmas de k electrodomésticos distintos, teniendo de esta forma $\frac{T}{k}$ firmas por cada aparato.

La idea de sum-to-k (**S2K**) es que la suma por grupo de electrodomésticos sea 1 o cercano a este valor. Cada grupo de firmas está compuesto por las firmas correspondientes al mismo electrodoméstico.

Esto elimina por completo el problema que un aparato aporte más potencia de la que suele aportar. Además exige que todos los electrodomésticos aporten alguna de sus firmas al consumo agregado, potencialmente 0 si alguna de sus firmas es nula.

En la práctica existen dos variantes de implementación de S2K. La primera de ellas, la cual es la que se propone en el artículo de referencia [4], añade S2K mediante un término que penaliza cuando la suma por grupo es distinta de 1. La formulación es la siguiente:

$$\min_{a \geq 0} \|x - Da\|_2^2 + \beta \|u - Qa\|_2^2$$

Con $u \in R^k$ un vector completo de 1 y $Q \in R^{k \times T}$ una matriz. A modo de ejemplo si se tienen 3 grupos de electrodomésticos, cada uno con 3, 1 y 2 firmas respectivamente, la matriz u es de tamaño 3 y Q es 3x6; y sus componentes son:

$$u = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, Q = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

De esta forma se tiene que el costo correspondiente al segundo termino será 0 solo en el caso de que se cumpla exactamente la S2K. El problema se puede reescribir para que tenga la misma forma que en la ecuación 2.

$$\min_{a \geq 0} \left\| \begin{bmatrix} x \\ \beta u \end{bmatrix} - \begin{bmatrix} D \\ \beta Q \end{bmatrix} a \right\|_2^2 \quad (6)$$

Sin embargo este problema tiene la complicación de la elección del β , el cual puede ser difícil de escoger y dado los largos tiempos de entrenamiento, el realizar una validación cruzada en busca del β óptimo es costoso.

Por lo anterior, se propone una variante que consiste en añadir S2K como una restricción, que impone que la suma por grupo sea 1. El problema asociado es:

$$\begin{array}{ll} \min_{a \geq 0} & \|x - Da\|_2^2 \\ \text{subject to} & \sum_{i \in \text{Group}} a_i = 1 \quad \forall \text{Group} \end{array} \quad (7)$$

3.2.7. Generalización

Todas las variantes planteadas pueden ser generalizadas a un problema que desagrega varios días en simultáneo. Esto se logra cambiando el vector \mathbf{x} por una matriz \mathbf{X} , que en cada columna posea un \mathbf{x} . Lo mismo ocurre con \mathbf{a} , que puede ser sustituido por una matriz \mathbf{A} .

3.3. Supuestos y Limitaciones

Los modelados propuestos se basan en que el consumo de un día puede ser desagregado mediante la suma de consumos de electrodomésticos en otro día. El gran problema que esto presenta es que para asignar que un electrodoméstico tenga determinado patrón de consumo, ya debe existir un patrón de consumo similar en la matriz \mathbf{D} .

A modo de ejemplo, si se quiere desagregar un consumo \mathbf{x} en el que una televisión fue encendida a las 4:00 AM y estuvo prendida por 2 horas, la única forma de conseguirlo es que exista en \mathbf{D} una firma de una TV en ese horario y con duración similar.

Por lo tanto, se supone que se cuenta con la suficiente cantidad de firmas \mathbf{d} , lo que es equivalente a tener una base de datos adquirida durante un largo

período de tiempo. Este supuesto es necesario para capturar todos los patrones de consumo de cada electrodoméstico, estos refieren al modo de uso, tiempo que estuvo encendido y momento del día.

4. Métodos de Optimización

Lo primero a destacar es que todos los problemas planteados en la sección anterior son convexos. Esto es porque la función a optimizar es convexa (se corresponde con la de mínimos cuadrados), mientras que las restricciones de desigualdad e igualdad son lineales.

En este trabajo se resolvió el enfoque básico con restricción (Ecuación 2) y las dos variantes de S2K (Ecuación 6 y 7). A continuación, se muestran dos métodos que permiten la resolución del problema.

Explorando como resolver el problema, se utilizó *CVXPY* [5]. Dado el gran tamaño de las matrices utilizadas y los métodos genéricos que implementa *CVXPY*, no fue posible resolver el problema por este medio en un tiempo razonable.

4.1. NNLS

Tanto la ecuación 2 como 6 tienen la misma forma, por lo cual se puede aplicar el mismo mecanismo de optimización. Existe para estos casos el algoritmo NNLS [6] de resolución **exacta**. En el Algoritmo 1 se presentan los pasos que propone NNLS.

A modo de resumen, este es un algoritmo iterativo que posee un conjunto de variables activas y no activas que se van actualizando en cada paso. Para esto en cada paso resuelve el problema de mínimos cuadrados solo con las restricciones no activas y verifica que el gradiente respecto a restricciones activas sea negativo. Por más detalle se sugiere ir a las referencias en las que se encuentra muy bien explicado.

Este algoritmo fue implementado por completo en *Python*. Esta implementación se encuentra disponible en el archivo *npls.py*.

El problema que presenta NNLS es que no es eficiente, en cada paso se debe resolver un problema de mínimos cuadrados, lo que implica una inversión matricial. Esto lleva a que sea un algoritmo que no escala a un problema con matrices grandes. De todas formas, el implementarlo sirvió como primera aproximación a la resolución del problema. Además, fue útil para luego utilizarlo como referencia en la etapa de desarrollo de métodos aproximados que se presentaran a continuación.

4.2. PGD

Con el fin de tener un algoritmo que escale, se propone utilizar *Projected Gradient Descent*. Este enfoque nos permite resolver cualquiera de los tres pro-

Algorithm 1: NNLS.

Algoritmo extraído de [7] y [8], donde también se puede obtener una mayor intuición de cada uno de los pasos y porque este funciona.

A. Inicialización

- A1. $P \leftarrow \emptyset$
- A2. $R \leftarrow \{1, 2, \dots, M\}$
- A3. $\mathbf{a} \leftarrow \mathbf{0}$
- A4. $\mathbf{w} \leftarrow \mathbf{D}^T(\mathbf{x} - \mathbf{D}\mathbf{a})$

B. Loop Principal

- B1. Continuar si: $R \neq \emptyset \wedge \left[\max_{n \in R} (w_n) > \text{tolerancia} \right]$
- B2. $m \leftarrow \underset{n \in R}{\operatorname{argmax}}(w_n)$
- B3. Incluir el índice m en P y eliminar este de R
- B4. $\mathbf{s}^P \leftarrow \left[(\mathbf{D}^P)^T \mathbf{D}^P \right]^{-1} (\mathbf{D}^P)^T \mathbf{x}$

B. Loop interno

- C1. Continuar si $\min(\mathbf{s}^P) \leq 0$
 - C2. $\alpha \leftarrow -\min_{n \in P} \frac{a_n}{a_n - s_n}$
 - C3. $\mathbf{a} \leftarrow \mathbf{a} + \alpha(\mathbf{s} - \mathbf{a})$
 - C4. Actualizar R y P
 - C5. $\mathbf{s}^P \leftarrow \left[(\mathbf{D}^P)^T \mathbf{D}^P \right]^{-1} (\mathbf{D}^P)^T \mathbf{x}$
 - C6. $\mathbf{s}^R \leftarrow \mathbf{0}$
- Ir a B5*

- B5. $\mathbf{a} \leftarrow \mathbf{s}$
 - B6. $\mathbf{w} \leftarrow \mathbf{D}^T(\mathbf{x} - \mathbf{D}\mathbf{a})$
- Ir a B1*
-

blemas propuestos, a diferencia de NNLS que no sirve para resolver un problema cuya forma es de la Ecuación 7.

De forma general, PGD alterna un paso de gradiente con la proyección a la región factible. Considerando que la función que se quiere minimizar es $\min_{a \in C} f(a)$, un pseudocódigo de PGD se puede observar en el Algoritmo 2.

Algorithm 2: Projected Gradient Descent Algorithm

P_C es la proyección al conjunto C

```

 $\alpha \leftarrow \frac{1}{\|D^T D\|_F^2}$ 
 $a \leftarrow a_0$ 
while Stop Condition do
  |  $a \leftarrow P_C(a - \alpha \nabla f(a))$ 
end

```

En el caso particular de interés, se tiene que el gradiente de la función objetivo es $\nabla f(a) = 2a^T(Da - x)$.

Mientras que la proyección (P_C) depende del caso que tratemos. Si se desea proyectar a los al conjunto $C_1 = \{a \in R^T / a > 0\}$ alcanza con llevar a 0 todas aquellas componentes que sea menores a 0.

En cambio para proyectar a la restricción de S2K (Ecuación 7), es un poco más complejo. En este caso se debe proyectar a dos conjuntos, en primer lugar a C_1 y luego a $C_2 = \left\{ a / \sum_{i \in \text{Group}} a_i = 1 \ \forall \text{Group} \right\}$. Para proyectar a C_2 lo que se hace es dividir cada elemento de a por la suma de los elementos de su grupo.

Este algoritmo fue implementado por completo en *Python* y hace uso de la GPU para acelerar el entrenamiento mediante el framework *Pytorch*. El código correspondiente se encuentra en *pgd.py*.

Con PGD se logra que mejorar sustancialmente el tiempo respecto a NNLS. De todas formas, con el objetivo de mejorar aún más la velocidad de convergencia, en la siguiente sección se proponen una mejora llamada A-PGD.

4.3. A-PGD

Accelerated Projected Gradient Descent (A-PGD) es una variación de método PGD que pretende mejorar la velocidad de convergencia.

Para lograr esto, lo que hace A-PGD es agregar momentum (o inercia) a la actualización del gradiente. Esto implica que en cada paso se actualice el valor de \mathbf{a} con una componente del gradiente actual y otro del momentum, como se ilustra en la Figura 2¹.

¹Imagen extraída de: <https://cs231n.github.io/neural-networks-3/>

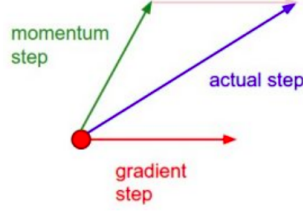


Figura 2: Nesterov Momentum

El pseudocódigo de APG se puede ver en el Algoritmo 3.

Algorithm 3: Accelerated Projected Gradient Descent Algorithm

α es el paso del algoritmo
 P_C es la proyección al conjunto C
 μ parámetro de momentum del problema

$a \leftarrow a_0$
while *Stop Condition* **do**
 $v \leftarrow \mu v - \alpha \nabla f(a)$
 $a \leftarrow P_C(a + v)$
end

La implementación consistió en una variante de Nesterov Momentum, que utiliza ADAM. El código se implementó en *Python* y se encuentra en el archivo *apg.py*.

4.4. Comparativa

Con fines comparativos se plantea el problema de la ecuación 2 con tamaños de las matrices pequeños.

En la Figura 3 se muestra la curva comparativa de la evolución de la loss en PGD y A-PGD. Ambos métodos llegan a la solución óptima dada por NNLS. Sin embargo, como se observa en la gráfica A-PGD llega en mucho menos iteraciones que PGD. Por esto en la sección de experimentos, se decidió utilizar **A-PGD** como método de resolución.

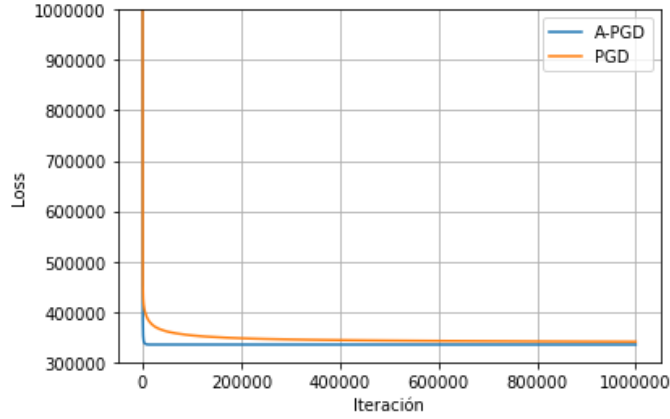


Figura 3: Comparativa de la evolución de la loss entre PGD y A-PGD con matrices pequeñas.

5. Experimentos

5.1. Base de datos: AMPds

AMPds [3] es un dataset público de consumo eléctrico recolectado en Canadá. Cuenta con 2 años de **series corriente** eléctrica a una frecuencia de una muestra por minuto. De los 21 medidores, 20 de estos se encargaron de recolectar los consumos desagregados y el restante para adquirir la señal del tablero general.

Las 20 señales desagregadas con las que se cuenta son:

- | | |
|------------------------------|--|
| ■ Whole-House Meter | ■ Forced Air Furnace: Fan and Thermostat |
| ■ North Bedroom | ■ Garage |
| ■ Master and South Bedroom | ■ Heat Pump |
| ■ Basement Plugs and Lights | ■ Instant Hot Water Unit |
| ■ Clothes Dryer | ■ Home Office |
| ■ Clothes Washer | ■ Outside Plug |
| ■ Dining Room Plugs | ■ Rental Suite |
| ■ Dishwasher | ■ Entertainment: TV, PVR, AMP |
| ■ Electronics Workbench | ■ Utility Room Plug |
| ■ Security/Network Equipment | ■ Wall Oven |
| ■ Kitchen Fridge | |

Notar que no todas las señales desagregadas corresponden a un único electrodoméstico. Por ejemplo, algunas corresponden a enchufes o habitaciones. Sin embargo esto no afecta al modelado, ya que pueden considerarse a estas señales como provenientes de un súper-electrodoméstico, cuya señal es la suma de las señales de un conjunto de electrodomésticos.

En la Figura 4 se muestra una de las señales agregadas, mientras que en la Figura 5 se observan las señales desagregadas.

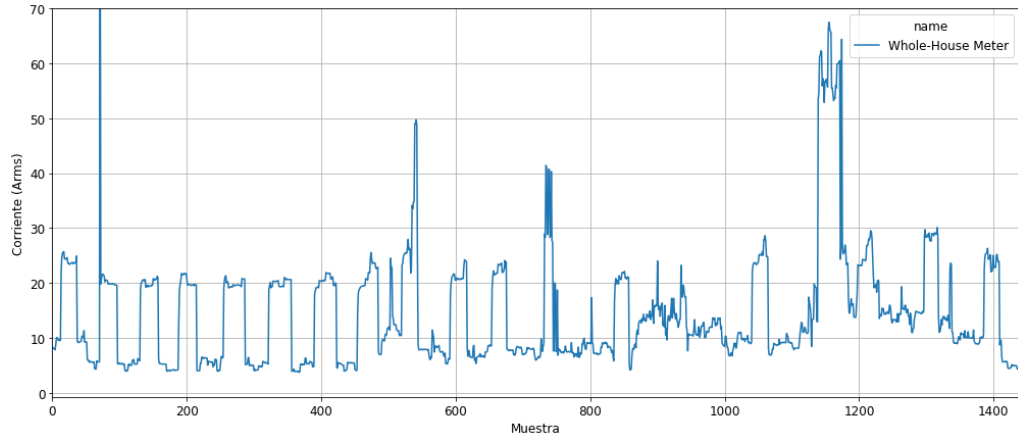


Figura 4: Curva de corriente agregada correspondiente al primer día de consumo de la base AMPds.

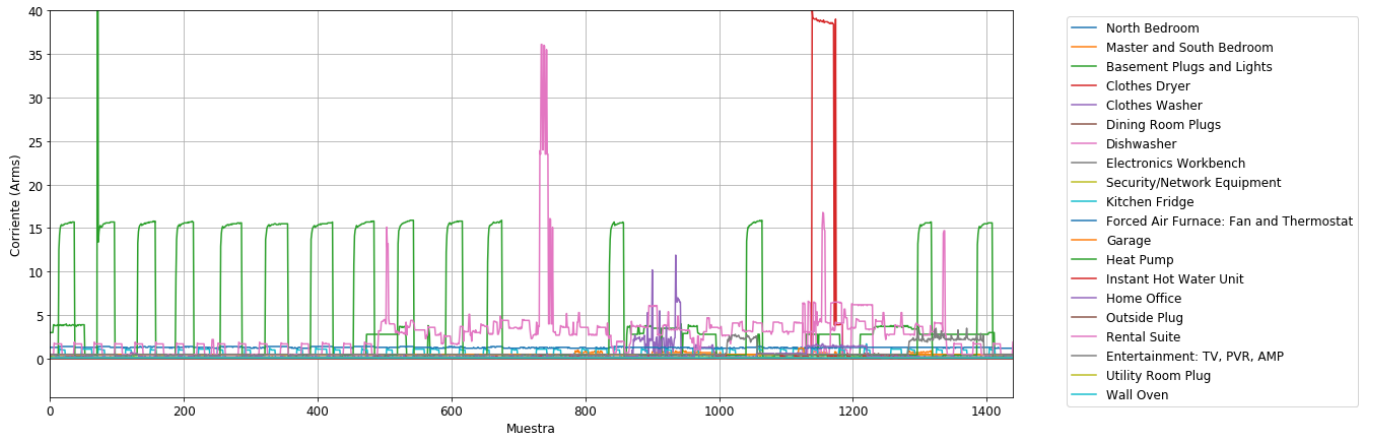


Figura 5: Curva de corriente correspondiente a cada electrodoméstico del primer día de la base AMPds.

Todo el preprocesamiento y post-procesamiento de los datos se realizó me-

diente el Toolkit de *Python* llamado *NILMTK* [9].

5.2. Métricas de desempeño

Se presentan dos métricas que son de interés para este problema de NILM.

5.2.1. Disaggregation Error

Corresponde en el error medio muestra a muestra en corriente entre la señal real y la reconstruida.

$$\sum_{i=1}^k \frac{1}{km} \|x_i - \hat{x}_i\|_1^2$$

En este trabajo solo se para la señal agregada. DE mide cuan bien se reconstruye la señal agregada y se debe notar que no aporta información sobre que tan bien se desagrega el consumo.

5.2.2. PCEC

Indica el porcentaje que ha sido asignado a cada electrodoméstico sobre el total de consumo estimado.

Esta es una métrica de desagregación y no aporta información de sí se reconstruye bien la señal agregada.

5.3. Resultados

Como ya se mencionó, se resuelven tres problemas mediante A-PGD.

Se realizó una división 80 %-20 % en los datos. Es decir, como se tenían exactamente 715 días de datos, 580 fueron utilizados como entrenamiento (en la matriz **D**) y 135 fueron utilizados para evaluar (en el vector **x**).

Los algoritmos fueron entrenados en *Cluster Uy* [10], con el fin de poder hacer uso de las GPU disponibles y así disminuir el tiempo de cómputo.

Se muestra la evolución de la loss, el DE y el PCEC. Más adelante, en la Sección 5.4, se presentan algunos ejemplos de la señal agregada y desagregada estimada para cada uno de los tres problemas resueltos.

5.3.1. Enfoque con restricción ($a > 0$)

Este es el enfoque en el que no se añade regularización, en la Figura 6 se observa la evolución de la loss a lo largo de las iteraciones. Se aprecia que esta logra la convergencia, ya que se estabiliza.

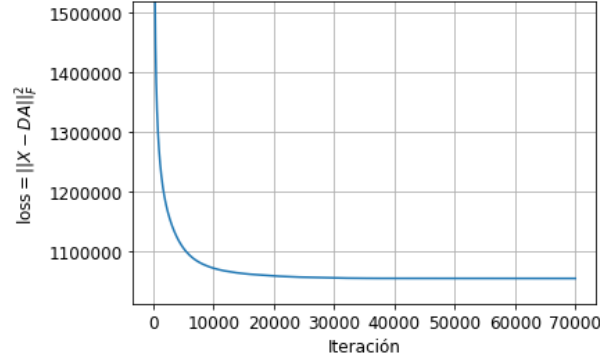


Figura 6: Evolución de la loss al resolver $\min_{a \geq 0} \|x - Da\|_2^2$ mediante A-PGD.

El DE obtenido es de 1,314, lo que representa que se ha reconstruido la señal con poco error.

A continuación, en la Figura 7 se presenta la comparativa del PCEC estimado con el PCEC real. Se ve que estos difieren, por ejemplo se le estima 0 % para *Forced Air Furnace*, siendo este uno de los que más consume en verdad.

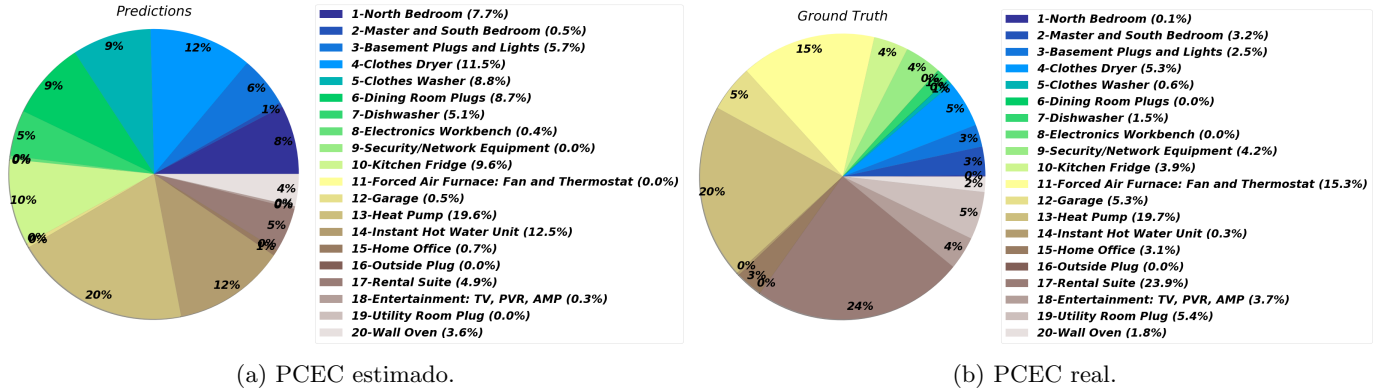


Figura 7: PCEC problema sin restricción.

Que se tenga un buen DE, pero mal PCEC se explica porque el algoritmo se concentra en reconstruir la señal lo más preciso que se pueda, sin tener en cuenta ninguna estructura específica del problema. Por esto fue que se propuso agregar S2K.

5.3.2. Regularización con S2K

En este caso, para resolver el problema, es necesario escoger el valor del parámetro β , para esto se usó una búsqueda en grilla. Se escogió $\beta = 10$, ya que

este mejoraba los resultados de PCEC sin generar que en el óptimo obtenido el término de regularización tenga mucho peso.

La evolución de la loss se presenta en la Figura 8.

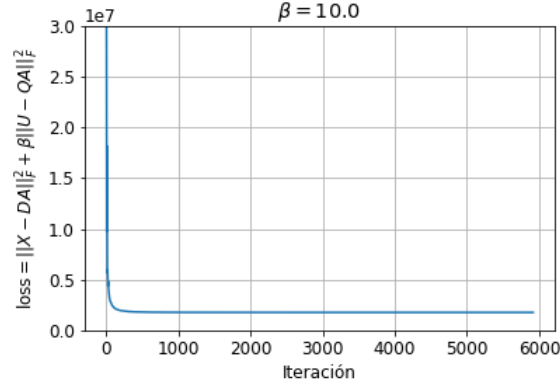


Figura 8: Evolución de la loss al resolver $\min_{a \geq 0} \left\| \begin{bmatrix} x \\ \beta u \end{bmatrix} - \begin{bmatrix} D \\ \beta Q \end{bmatrix} a \right\|_2^2$ mediante A-PGD.

El DE obtenido es 1,74, siendo este inferior al caso sin S2K. Los resultados de PCEC se encuentran en la Figura 9.

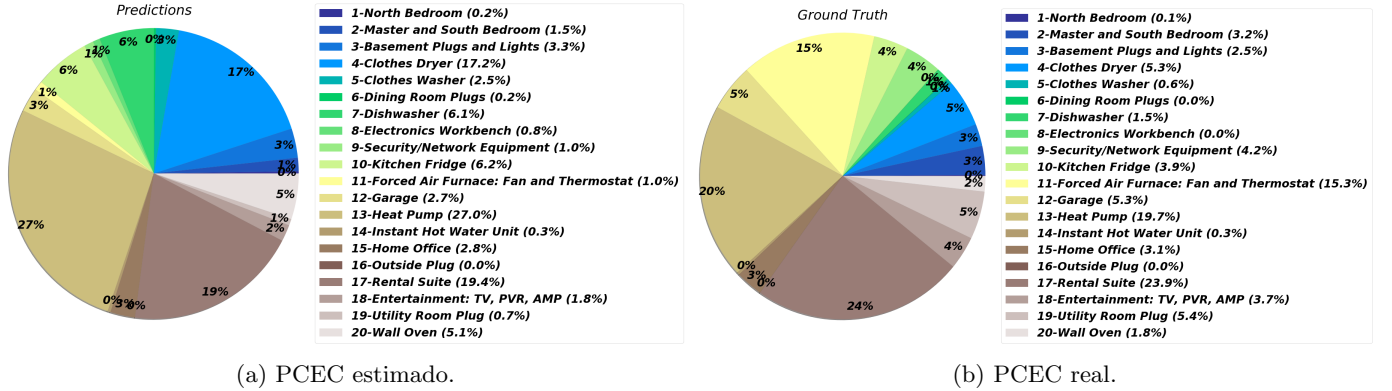


Figura 9: PCEC problema con regularización S2K.

En este caso se observan mejores resultados de PCEC, a costa de disminuir el DE. De todas formas el PCEC obtenido en algunos casos tiene fallas.

5.3.3. Con restricción S2K

La evolución de la loss para este problema se muestra en la Figura 10.

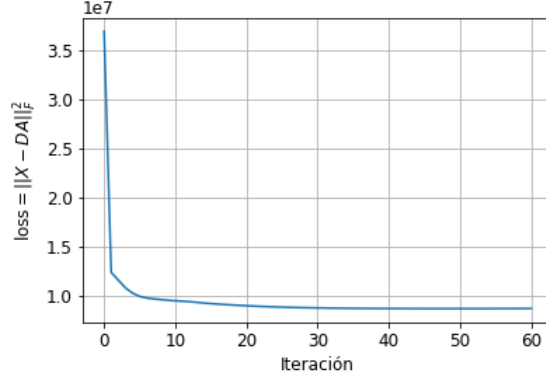


Figura 10: Evolución de la loss al resolver de la Ecuación 7 mediante A-PGD.

El DE para este problema es 3,664, el más alto de los 3, ya que se impone S2K como una restricción dura.

En tanto, el PCEC correspondiente a este problema es el de la Figura 11.

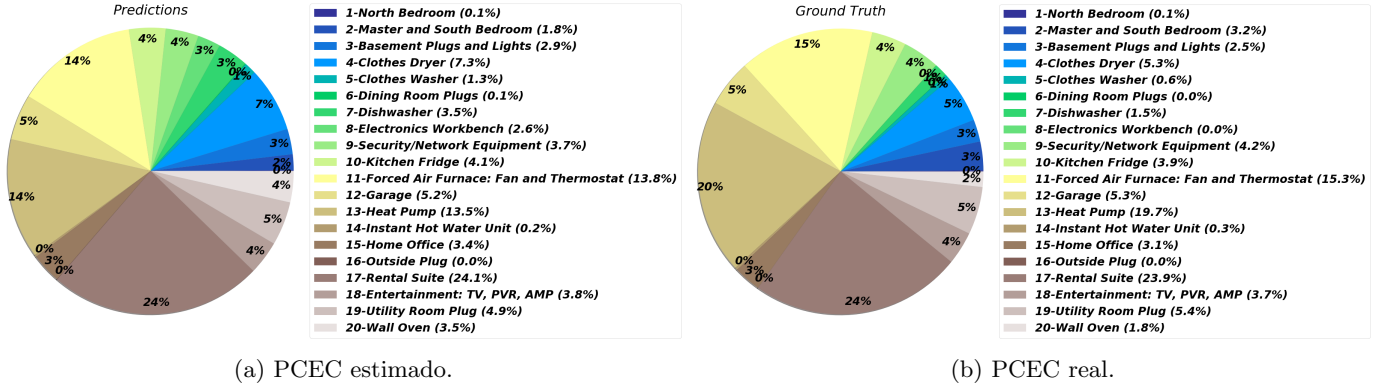


Figura 11: PCEC problema con restricción S2K.

Para este modelado, se obtienen excelentes resultados de PCEC, mientras que el peor DE de los tres.

5.3.4. Comparativa

Con fines de poder contrastar más claramente el DE y PCEC, se presentan los siguientes cuadros comparativos.

	Regularización S2K	Restricción S2K	Sin regularización
DE	1.7 A	3.66 A	1.31 A

Cuadro 1: Tabla comparativa de DE.

	Ground truth	Regularización S2K	Restricción S2K	Sin regularización
North Bedroom	0.12 %	0.23 %	0.08 %	7.69 %
Master and South Bedroom	3.25 %	1.50 %	1.80 %	0.53 %
Basement Plugs and Lights	2.51 %	3.26 %	2.87 %	5.66 %
Clothes Dryer	5.33 %	17.23 %	7.34 %	11.52 %
Clothes Washer	0.62 %	2.55 %	1.34 %	8.83 %
Dining Room Plugs	0.04 %	0.23 %	0.09 %	8.67 %
Dishwasher	1.48 %	6.12 %	3.49 %	5.10 %
Electronics Workbench	0.01 %	0.79 %	2.60 %	0.39 %
Security/Network Equipment	4.17 %	1.04 %	3.73 %	0.00 %
Kitchen Fridge	3.92 %	6.23 %	4.14 %	9.58 %
Forced Air Furnace: Fan and Thermostat	15.33 %	1.04 %	13.84 %	0.00 %
Garage	5.28 %	2.67 %	5.15 %	0.45 %
Heat Pump	19.74 %	26.95 %	13.53 %	19.58 %
Instant Hot Water Unit	0.29 %	0.32 %	0.24 %	12.50 %
Home Office	3.08 %	2.76 %	3.41 %	0.69 %
Outside Plug	0.00 %	0.01 %	0.04 %	0.01 %
Rental Suite	23.93 %	19.42 %	24.07 %	4.92 %
Entertainment: TV, PVR, AMP	3.75 %	1.84 %	3.80 %	0.30 %
Utility Room Plug	5.40 %	0.72 %	4.93 %	0.00 %
Wall Oven	1.76 %	5.09 %	3.54 %	3.58 %

Cuadro 2: Tabla comparativa de PCEC.

A modo de resumen, cuanto más se impone que se cumpla S2K más se mejora el PCEC, sin embargo se disminuye el DE. Esto se explica debido a que el DE indica que tan bien se reconstruye la señal agregada, no siendo esto lo que intenta hacer S2K. El objetivo de S2K es que se reconstruya mejor a las señales desagregadas a costa de una peor reconstrucción de la señal agregada.

5.4. Resultados visuales

Por ultimo, se presentan los resultados de desagregación de algunos electrodomésticos. Para cada uno de los problemas, se muestra la serie agregada obtenida y la serie desagregada para un *Lavavajillas*, *Heladera* y *Bomba de calor*.

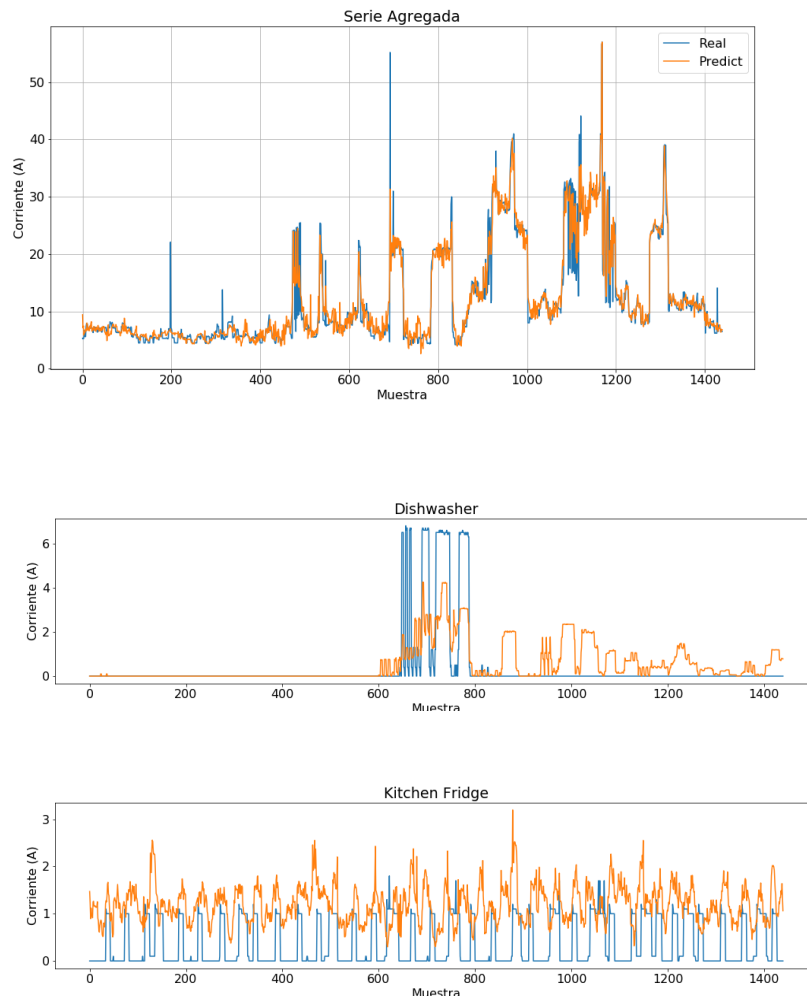
En el caso del problema sin regularización y con regularización S2K se observa una reconstrucción de la señal agregada razonable, empeorando mucho en el caso de agregar S2K como restricción.

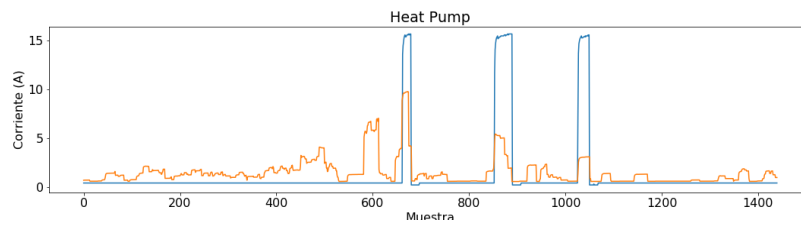
Mientras que para las señales desagregadas no parecen reconstruirse correctamente en ninguno de los casos. De todas maneras, al introducir S2K se aprecia una leve mejoría.

Las estimaciones de las señales desagregadas encontradas no son buenas, no lográndose mejorar la reconstrucción al añadir regularización.

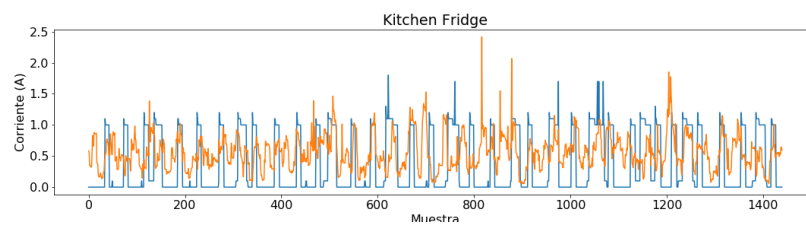
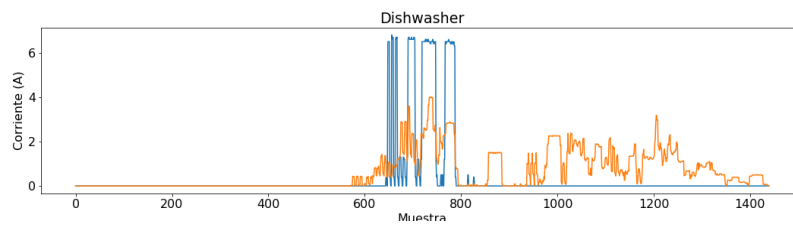
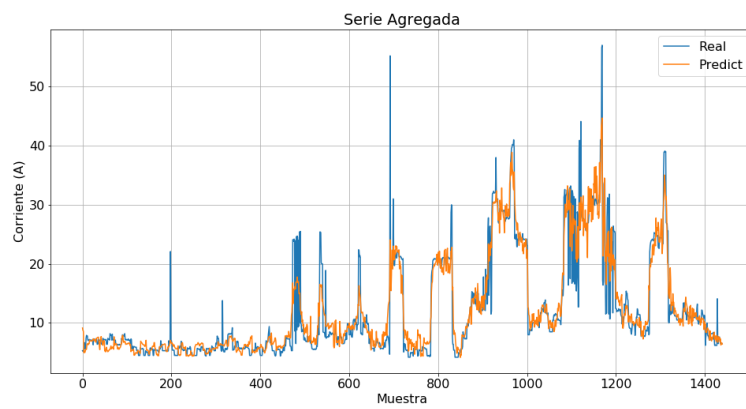
A continuación, se presentan las figuras que muestran las figuras que se mencionaron.

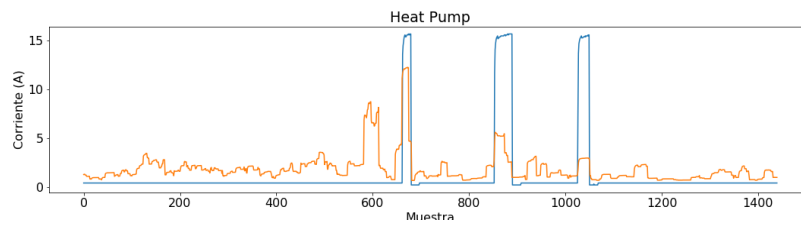
Sin restricción



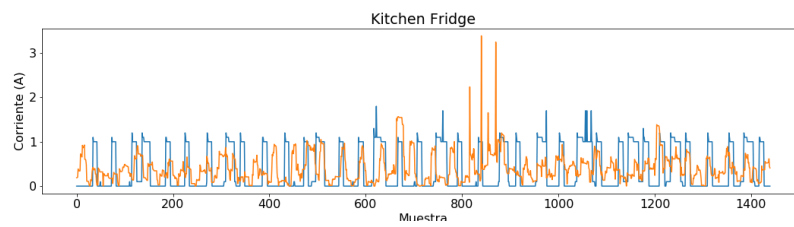
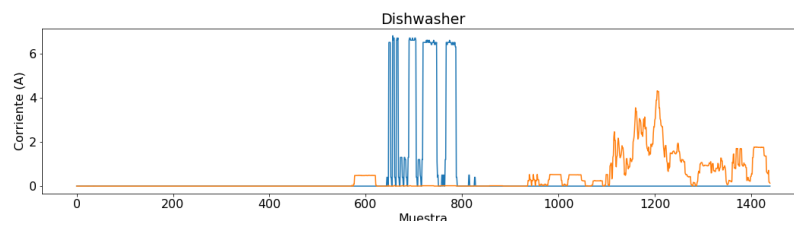
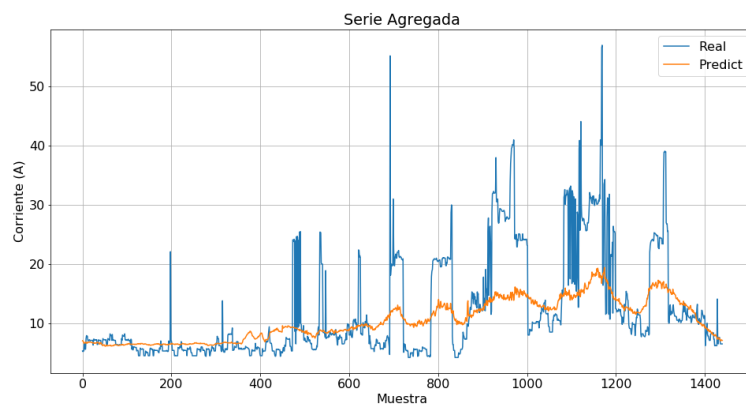


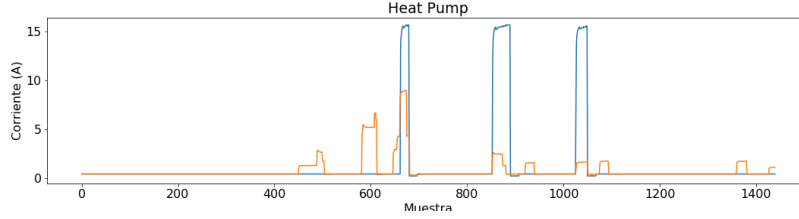
Regularización S2K





Restricción S2K





6. Conclusiones

Se consiguió modelar NILM como un problema de optimización. Para esto se utilizó el enfoque novedoso S2K y se propuso una variante del mismo.

Se logró resolver el problema de optimización, lo cual fue un gran desafío, ya que utilizando software tradicional (CVXPY) y métodos clásicos (NNLS), el problema no era resoluble en un tiempo adecuado. Se implementaron 2 algoritmos y con estos se resolvieron 3 problemas de optimización. Todo el código se encuentra disponible en GitHub ².

Se probó el enfoque con una restricción sencilla y sin regularización, obteniéndose malos resultados de PCEC. En cambio se tuvieron buenos resultados de DE. Esto se explica sencillamente porque se cuenta con varias bases **d** de electrodomésticos que sus consumos son de tiempos muy pequeños (similares a deltas de valor finito), y a partir de estos se puede reconstruir cualquier señal.

Al agregar regularización se logró mejorar los resultados de PCEC, lo cual es deseado. Sin embargo, las Figuras correspondientes a las señales desagregadas estimadas no son nada alentadoras. Se obtienen muy malos resultados de reconstrucción en todos los casos, mejorando levemente al utilizar cualquiera de las 2 variantes de S2K.

Para que el modelado planteado sea útil, se necesita una gran cantidad de datos, lo que conlleva a un costo computacional grande en la resolución del problema de optimización. Con 2 años de datos de una única casa no fue posible desagregar correctamente, lo que sugiere que este enfoque necesita aún más datos para poder capturar más firmas y así contemplar más patrones de consumo.

Como se mencionó en la Sección 3.3, que consumos de un electrodoméstico en horarios distintos correspondan a una firma distinta es una gran limitante. Por lo que se concluye que seria de gran ayuda incluir en el modelado alguna forma de tener en cuenta este corrimiento temporal. A la fecha, no existe ningún trabajo que aborde esta problemática desde el punto de vista de la resolución de un problema de optimización.

Lo implementado no logra reconstruir la señal desagregada. De todas formas, no se puede concluir que sea un enfoque que deba ser descartado. Para descartarlo se deberían explorar alternativas, algunas de ellas:

- Realizar búsqueda detallada del hiper-parámetro β .

²Los tres algoritmos de optimización, junto con el pre y post procesamiento de los datos se encuentran disponibles en un repositorio GitHub público. Este se encuentra en https://github.com/camilomarinero/Obligatorios_TA0_2020/tree/master/ProyectoFinal

- Filtrar la base de datos utilizada. Por ejemplo, eliminar los consumos tipo *delta* a la hora de entrenar.
- Modificar el largo de ventana. Se podría utilizar en lugar de un día, menos o más días.

Bibliografía

- [1] G. W. Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, 1992.
- [2] Jack Kelly and William Knottenbelt. The uk-dale dataset, domestic appliance-level electricity demand and whole-house demand from five uk homes. *Scientific data*, 2(1):1–14, 2015.
- [3] S. Makonin, F. Popowich, L. Bartram, B. Gill, and I. V. Bajić. Ampds: A public dataset for load disaggregation and eco-feedback research. In *2013 IEEE Electrical Power Energy Conference*, pages 1–6, 2013.
- [4] A. Rahimpour, H. Qi, D. Fugate, and T. Kuruganti. Non-intrusive energy disaggregation using non-negative matrix factorization with sum-to-k constraint. *IEEE Transactions on Power Systems*, 32(6):4430–4441, 2017.
- [5] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.
- [6] Charles L Lawson and Richard J Hanson. *Solving least squares problems*. SIAM, 1995.
- [7] Rasmus Bro and Sijmen De Jong. A fast non-negativity-constrained least squares algorithm. *Journal of Chemometrics*, 11(5):393–401, 1997.
- [8] Wikipedia. Non-negative least squares — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Non-negative%20least%20squares&oldid=993145058>, 2020. [Online; accessed 23-December-2020].
- [9] Nipun Batra, Jack Kelly, Oliver Parson, Haimonti Dutta, William Knottenbelt, Alex Rogers, Amarjeet Singh, and Mani Srivastava. Nilmtk: An open source toolkit for non-intrusive load monitoring. In *Proceedings of the 5th International Conference on Future Energy Systems*, e-Energy ’14, page 265–276, New York, NY, USA, 2014. Association for Computing Machinery.
- [10] Sergio Nesmachnow and Santiago Iturriaga. Cluster-uy: Collaborative scientific high performance computing in uruguay. In Moisés Torres and Jaime Klapp, editors, *Supercomputing*, pages 188–202, Cham, 2019. Springer International Publishing.