Energy Disaggregation using Piecewise Affine Regression and Binary Quadratic Programming

Manas Mejari[1], Vihangkumar V. Naik[2], Dario Piga[3] and Alberto Bemporad[2]

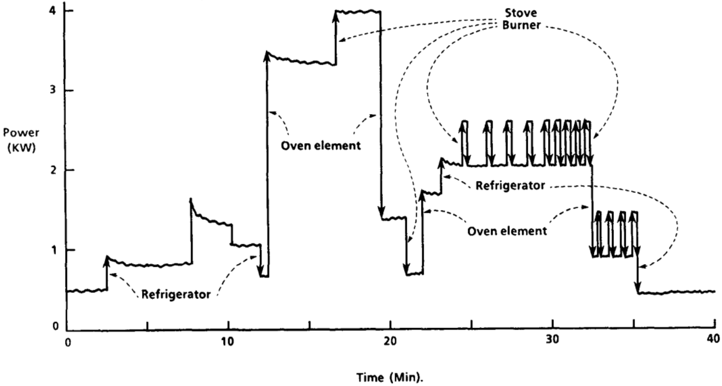
[1]University of Lille and École Centrale de Lille, France
manas-dilip.mejari@univ-lille.fr

[2]IMT School for Advanced Studies Lucca, Italy
{vihangkumar.naik,alberto.bemporad}@imtlucca.it

[3]The IDSIA Dalle Molle Institute for Artificial Intelligence, Manno, Switzerland
dario.piga@supsi.ch

Energy Disaggregation problem

Estimate the end-use power consumption profiles of individual household appliances using only aggregated power measurements.



Original figure from Hart 1992, first contribution on energy disaggregation problem.

Energy Disaggregation problem

Given an N-length data sequence $\{y(k)\}_{k=1}^{N}$ of aggregated power signals $y(k)$, estimate power consumption profiles $y_i(k)$ for each appliance

$$y(k) = \sum_{i=1}^{n} y_i(k) + e(k),$$

n denotes number of household appliances,
$e(k)$ is the measurement noise and unmodeled appliances.

Energy Disaggregation Algorithm

We propose a two-stage supervised disaggregation algorithm:

Stage S1:
- The power consumption profile of each appliance is described by PieceWise Affine AutoRegressive (PWA-AR) model.
- The PWA-AR model for each appliance is estimated via the moving-horizon PWA regression algorithm[1]

Stage S2:
- Using the PWA-AR models obtained from stage **S1**, a Binary Quadratic Programming (BQP) problem is solved iteratively.
- The BQP solution determines the active appliances contributing to the instantaneous total power.

[1] using a small set of disaggregated training data

Energy Disaggregation Algorithm

We propose a two-stage supervised disaggregation algorithm:

Stage S1:
- The power consumption profile of each appliance is described by PieceWise Affine AutoRegressive (PWA-AR) model.
- The PWA-AR model for each appliance is estimated via the moving-horizon PWA regression algorithm[1]

Stage S2:
- Using the PWA-AR models obtained from stage **S1**, a Binary Quadratic Programming (BQP) problem is solved iteratively.
- The BQP solution determines the active appliances contributing to the instantaneous total power.

[1] using a small set of disaggregated training data

Energy Disaggregation Algorithm

We propose a two-stage supervised disaggregation algorithm:

Stage S1:
- The power consumption profile of each appliance is described by PieceWise Affine AutoRegressive (PWA-AR) model.
- The PWA-AR model for each appliance is estimated via the moving-horizon PWA regression algorithm[1]

Stage S2:
- Using the PWA-AR models obtained from stage **S1**, a Binary Quadratic Programming (BQP) problem is solved iteratively.
- The BQP solution determines the active appliances contributing to the instantaneous total power.

[1] using a small set of disaggregated training data

Energy Disaggregation Algorithm

We propose a two-stage supervised disaggregation algorithm:

Stage S1:
- The power consumption profile of each appliance is described by PieceWise Affine AutoRegressive (PWA-AR) model.
- The PWA-AR model for each appliance is estimated via the moving-horizon PWA regression algorithm[1]

Stage S2:
- Using the PWA-AR models obtained from stage **S1**, a Binary Quadratic Programming (BQP) problem is solved iteratively.
- The BQP solution determines the active appliances contributing to the instantaneous total power.

[1] using a small set of disaggregated training data

Energy Disaggregation Algorithm

We propose a two-stage supervised disaggregation algorithm:

Stage S1:
- The power consumption profile of each appliance is described by PieceWise Affine AutoRegressive (PWA-AR) model.
- The PWA-AR model for each appliance is estimated via the moving-horizon PWA regression algorithm[1]

Stage S2:
- Using the PWA-AR models obtained from stage **S1**, a Binary Quadratic Programming (BQP) problem is solved iteratively.
- The BQP solution determines the active appliances contributing to the instantaneous total power.

[1] using a small set of disaggregated training data

Stage S1: Training appliance models

The power $y_i(k)$ consumed by the i-th appliance at time k is modeled by

$$y_i(k) = \begin{cases} \Theta_{i,1}^\top \begin{bmatrix} 1 \\ x_i(k) \end{bmatrix} & \text{if } \delta_{i,1}(k) = 1, \\ & \vdots \\ \Theta_{i,s}^\top \begin{bmatrix} 1 \\ x_i(k) \end{bmatrix} & \text{if } \delta_{i,s}(k) = 1, \end{cases}$$

$\delta_{i,j}(k) \in \{0, 1\}$ operating mode, $x_i(k)$ regressor vector

$$x_i(k) = \left[y_i(k-1), \ldots, \; y_i(k-n_\mathrm{a})\right]^\top$$

Estimation of the PWA-AR appliance model consists of:

1. selecting the number of modes s
2. estimating the model parameter matrices $\Theta_{i,j}$
3. estimating the active operating mode $\delta_{i,j}(k)$

Stage S1: Training appliance models

The power $y_i(k)$ consumed by the i-th appliance at time k is modeled by

$$y_i(k) = \begin{cases} \Theta_{i,1}^\top \begin{bmatrix} 1 \\ x_i(k) \end{bmatrix} & \text{if } \delta_{i,1}(k) = 1, \\ & \vdots \\ \Theta_{i,s}^\top \begin{bmatrix} 1 \\ x_i(k) \end{bmatrix} & \text{if } \delta_{i,s}(k) = 1, \end{cases}$$

$\delta_{i,j}(k) \in \{0, 1\}$ operating mode, $x_i(k)$ regressor vector

$$x_i(k) = [y_i(k-1), \ldots, \; y_i(k-n_\mathrm{a})]^\top$$

Estimation of the PWA-AR appliance model consists of:

1. selecting the number of modes s
2. estimating the model parameter matrices $\Theta_{i,j}$
3. estimating the active operating mode $\delta_{i,j}(k)$

Stage S1: Training appliance models

The power $y_i(k)$ consumed by the i-th appliance at time k is modeled by

$$y_i(k) = \begin{cases} \Theta_{i,1}^\top \begin{bmatrix} 1 \\ x_i(k) \end{bmatrix} & \text{if } \delta_{i,1}(k) = 1, \\ & \vdots \\ \Theta_{i,s}^\top \begin{bmatrix} 1 \\ x_i(k) \end{bmatrix} & \text{if } \delta_{i,s}(k) = 1, \end{cases}$$

$\delta_{i,j}(k) \in \{0, 1\}$ operating mode, $x_i(k)$ regressor vector

$$x_i(k) = [y_i(k-1), \ldots, \; y_i(k-n_\mathrm{a})]^\top$$

Estimation of the PWA-AR appliance model consists of:

1. selecting the number of modes s
2. estimating the model parameter matrices $\Theta_{i,j}$
3. estimating the active operating mode $\delta_{i,j}(k)$

Stage S1: Training appliance models

The power $y_i(k)$ consumed by the i-th appliance at time k is modeled by

$$y_i(k) = \begin{cases} \Theta_{i,1}^\top \begin{bmatrix} 1 \\ x_i(k) \end{bmatrix} & \text{if } \delta_{i,1}(k) = 1, \\ & \vdots \\ \Theta_{i,s}^\top \begin{bmatrix} 1 \\ x_i(k) \end{bmatrix} & \text{if } \delta_{i,s}(k) = 1, \end{cases}$$

$\delta_{i,j}(k) \in \{0, 1\}$ operating mode, $x_i(k)$ regressor vector

$$x_i(k) = [y_i(k-1), \ldots,\ y_i(k - n_\mathrm{a})]^\top$$

Estimation of the PWA-AR appliance model consists of:

1. selecting the number of modes s
2. estimating the model parameter matrices $\Theta_{i,j}$
3. estimating the active operating mode $\delta_{i,j}(k)$

Stage S1: Training appliance models

The power $y_i(k)$ consumed by the i-th appliance at time k is modeled by

$$y_i(k) = \begin{cases} \Theta_{i,1}^\top \begin{bmatrix} 1 \\ x_i(k) \end{bmatrix} & \text{if } \delta_{i,1}(k) = 1, \\ & \vdots \\ \Theta_{i,s}^\top \begin{bmatrix} 1 \\ x_i(k) \end{bmatrix} & \text{if } \delta_{i,s}(k) = 1, \end{cases}$$

$\delta_{i,j}(k) \in \{0, 1\}$ operating mode, $x_i(k)$ regressor vector

$$x_i(k) = [y_i(k-1), \ldots,\ y_i(k-n_\mathrm{a})]^\top$$

Estimation of the PWA-AR appliance model consists of:

1. selecting the number of modes s (via cross-validation)
2. estimating the model parameter matrices $\Theta_{i,j}$
3. estimating the active operating mode $\delta_{i,j}(k)$

Stage S1: Training appliance models

- At each time sample k, a moving-horizon window of length N_p containing regressor/output pairs $\{x_i(k), y_i(k)\}$ from time $k - N_p + 1$ to time k is considered.
- A *mixed-integer programming problem* is solved to simultaneously estimate:
  1. the model parameters $\Theta_{i,j}$
  2. the active mode $\sigma_i(k) \in \{1, \ldots, s\}$

     $\boxed{\sigma_i(k) = j^* \Leftrightarrow \delta_{i,j^*}(k) = 1}$
- The training data samples $\{x_i(k), y_i(k)\}$ are processed iteratively by shifting the horizon window.

Stage S1: Training appliance models

- At each time sample k, a moving-horizon window of length N_p containing regressor/output pairs $\{x_i(k), y_i(k)\}$ from time $k - N_p + 1$ to time k is considered.
- A *mixed-integer programming problem* is solved to simultaneously estimate:
  1. the model parameters $\Theta_{i,j}$
  2. the active mode $\sigma_i(k) \in \{1, \ldots, s\}$

     $\boxed{\sigma_i(k) = j^* \Leftrightarrow \delta_{i,j^*}(k) = 1}$
- The training data samples $\{x_i(k), y_i(k)\}$ are processed iteratively by shifting the horizon window.

Stage S1: Training appliance models

- At each time sample k, a moving-horizon window of length N_p containing regressor/output pairs $\{x_i(k), y_i(k)\}$ from time $k - N_p + 1$ to time k is considered.
- A *mixed-integer programming problem* is solved to simultaneously estimate:
    1. the model parameters $\Theta_{i,j}$
    2. the active mode $\sigma_i(k) \in \{1, \ldots, s\}$

       $\boxed{\sigma_i(k) = j^* \Leftrightarrow \delta_{i,j^*}(k) = 1}$
- The training data samples $\{x_i(k), y_i(k)\}$ are processed iteratively by shifting the horizon window.

Stage S1: Training appliance models

- At each time sample k, a moving-horizon window of length N_p containing regressor/output pairs $\{x_i(k), y_i(k)\}$ from time $k - N_p + 1$ to time k is considered.
- A *mixed-integer programming problem* is solved to simultaneously estimate:
  1. the model parameters $\Theta_{i,j}$
  2. the active mode $\sigma_i(k) \in \{1, \ldots, s\}$

     $\boxed{\sigma_i(k) = j^* \Leftrightarrow \delta_{i,j^*}(k) = 1}$
- The training data samples $\{x_i(k), y_i(k)\}$ are processed iteratively by shifting the horizon window.

Stage S1: Training appliance models

- At each time sample k, a moving-horizon window of length N_p containing regressor/output pairs $\{x_i(k), y_i(k)\}$ from time $k - N_p + 1$ to time k is considered.
- A *mixed-integer programming problem* is solved to simultaneously estimate:
  1. the model parameters $\Theta_{i,j}$
  2. the active mode $\sigma_i(k) \in \{1, \dots, s\}$

     $\boxed{\sigma_i(k) = j^* \Leftrightarrow \delta_{i,j^*}(k) = 1}$
- The training data samples $\{x_i(k), y_i(k)\}$ are processed iteratively by shifting the horizon window.

Stage S1: Training appliance models

At time k, we solve,

$$\min_{\Theta_{i,j},\ \delta_{i,j}(k-t)} \sum_{j=1}^{s} \sum_{t=0}^{N_p-1} \left\| \left(y_i(k-t) - \Theta_{i,j}^\top \begin{bmatrix} 1 \\ x_i(k-t) \end{bmatrix} \right) \delta_{i,j}(k-t) \right\|_2^2$$

$$\text{s.t.} \quad \delta_{i,j}(k-t) \in \{0,1\}, \ \sum_{j=1}^{s} \delta_{i,j}(k-t) = 1, \ t = 0, \ldots, N_p - 1.$$

- Fitting error term optimized over N_p horizon samples
- $\delta_{i,j}(k)$ indicates active mode at time k, i.e., $\boxed{\delta_{i,j}(k) = 1 \Rightarrow \sigma_i(k) = j}$
- Computes both model parameters $\Theta_{i,j}$ and active modes $\sigma_i(k)$ at k simultaneously

Stage S1: Training appliance models

At time k, we solve,

$$\min_{\Theta_{i,j},\ \delta_{i,j}(k-t)} \sum_{j=1}^{s} \sum_{t=0}^{N_p-1} \left\| \left(y_i(k-t) - \Theta_{i,j}^\top \begin{bmatrix} 1 \\ x_i(k-t) \end{bmatrix} \right) \delta_{i,j}(k-t) \right\|_2^2$$

$$\text{s.t.} \quad \delta_{i,j}(k-t) \in \{0,1\}, \ \sum_{j=1}^{s} \delta_{i,j}(k-t) = 1, \ t = 0, \ldots, N_p - 1.$$

- Fitting error term optimized over N_p horizon samples
- $\delta_{i,j}(k)$ indicates active mode at time k, i.e., $\boxed{\delta_{i,j}(k) = 1 \Rightarrow \sigma_i(k) = j}$
- Computes both model parameters $\Theta_{i,j}$ and active modes $\sigma_i(k)$ at k simultaneously

Stage S1: Training appliance models

At time k, we solve,

$$\min_{\Theta_{i,j},\ \delta_{i,j}(k-t)} \sum_{j=1}^{s} \sum_{t=0}^{N_p-1} \left\| \left(y_i(k-t) - \Theta_{i,j}^\top \begin{bmatrix} 1 \\ x_i(k-t) \end{bmatrix} \right) \delta_{i,j}(k-t) \right\|_2^2$$

$$\text{s.t.} \quad \delta_{i,j}(k-t) \in \{0,1\}, \ \sum_{j=1}^{s} \delta_{i,j}(k-t) = 1, \ t = 0, \ldots, N_p - 1.$$

- Fitting error term optimized over N_p horizon samples
- $\delta_{i,j}(k)$ indicates active mode at time k,
  i.e., $\boxed{\delta_{i,j}(k) = 1 \Rightarrow \sigma_i(k) = j}$
- Computes both model parameters $\Theta_{i,j}$ and active modes $\sigma_i(k)$ at k simultaneously

Stage S1: Training appliance models

At time k, we solve,

$$\min_{\Theta_{i,j},\ \delta_{i,j}(k-t)} \sum_{j=1}^{s} \sum_{t=0}^{N_p-1} \left\| \left(y_i(k-t) - \Theta_{i,j}^\top \left[\begin{smallmatrix} 1 \\ x_i(k-t) \end{smallmatrix} \right] \right) \delta_{i,j}(k-t) \right\|_2^2$$

$$\text{s.t.}\ \ \delta_{i,j}(k-t) \in \{0,1\},\ \sum_{j=1}^{s} \delta_{i,j}(k-t) = 1,\ t = 0, \ldots, N_p - 1.$$

- Fitting error term optimized over N_p horizon samples
- $\delta_{i,j}(k)$ indicates active mode at time k,
  i.e., $\boxed{\delta_{i,j}(k) = 1 \Rightarrow \sigma_i(k) = j}$
- Computes both model parameters $\Theta_{i,j}$ and active modes $\sigma_i(k)$ at k simultaneously

Stage S1: Training appliance models

$$\min_{\Theta_{i,j},\ \delta_{i,j}(k-t)} \sum_{j=1}^{s} \sum_{t=0}^{N_p-1} \left\| \left(y_i(k-t) - \Theta_{i,j}^\top \begin{bmatrix} 1 \\ x_i(k-t) \end{bmatrix} \right) \delta_{i,j}(k-t) \right\|_2^2$$

$$+ \boxed{\sum_{t=1}^{k-N_p} \left\| y_i(t) - \Theta_{i,\sigma(t)}^\top \begin{bmatrix} 1 \\ x_i(t) \end{bmatrix} \right\|^2} \quad \text{regularization on } \Theta_{i,j}$$

$$\text{s.t.}\quad \delta_i(k-t) \in \{0,1\},\ \sum_{i=1}^{s} \delta_i(k-t) = 1, \quad t = 0, \ldots, T-1.$$

- takes into account the time history, outside the considered time window i.e. from time 1 to time $k - N_p$
- Fixed estimates $\{\sigma_i(t)\}_{t=1}^{k-N_p}$ are used, to refine the estimates of the model parameters $\Theta_{i,j}$

Stage S1: Training appliance models

$$\min_{\Theta_{i,j},\ \delta_{i,j}(k-t)} \sum_{j=1}^{s} \sum_{t=0}^{N_p-1} \left\| \left(y_i(k-t) - \Theta_{i,j}^\top \begin{bmatrix} 1 \\ x_i(k-t) \end{bmatrix} \right) \delta_{i,j}(k-t) \right\|_2^2$$

$$+ \boxed{\sum_{t=1}^{k-N_p} \left\| y_i(t) - \Theta_{i,\sigma(t)}^\top \begin{bmatrix} 1 \\ x_i(t) \end{bmatrix} \right\|^2} \quad \text{regularization on } \Theta_{i,j}$$

$$\text{s.t.} \quad \delta_i(k-t) \in \{0,1\}, \ \sum_{i=1}^{s} \delta_i(k-t) = 1, \quad t = 0, \ldots, T-1.$$

- takes into account the time history, outside the considered time window i.e. from time 1 to time $k - N_p$
- Fixed estimates $\{\sigma_i(t)\}_{t=1}^{k-N_p}$ are used, to refine the estimates of the model parameters $\Theta_{i,j}$

Stage S1: Training appliance models

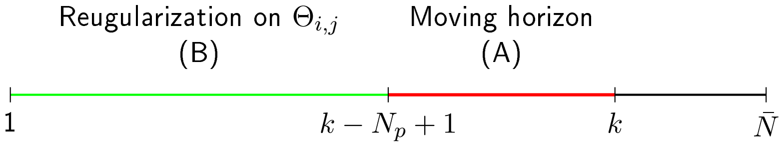
$$\min_{\Theta_{i,j},\ \delta_{i,j}(k-t)} \sum_{j=1}^{s} \sum_{t=0}^{N_p-1} \left\| \left(y_i(k-t) - \Theta_{i,j}^\top \begin{bmatrix} 1 \\ x_i(k-t) \end{bmatrix} \right) \delta_{i,j}(k-t) \right\|_2^2$$

$$+ \boxed{\sum_{t=1}^{k-N_p} \left\| y_i(t) - \Theta_{i,\sigma(t)}^\top \begin{bmatrix} 1 \\ x_i(t) \end{bmatrix} \right\|^2} \quad \text{regularization on } \Theta_{i,j}$$

$$\text{s.t.} \quad \delta_i(k-t) \in \{0,1\},\ \sum_{i=1}^{s} \delta_i(k-t) = 1, \quad t = 0, \ldots, T-1.$$

- takes into account the time history, outside the considered time window i.e. from time 1 to time $k - N_p$
- Fixed estimates $\{\sigma_i(t)\}_{t=1}^{k-N_p}$ are used, to refine the estimates of the model parameters $\Theta_{i,j}$

Stage S1: Training appliance models

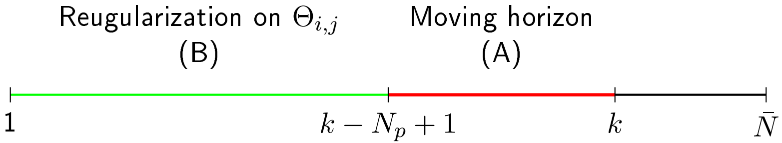


$$\min_{\Theta_{i,j},\ \delta_{i,j}(k-t)} \sum_{j=1}^{s} \sum_{t=0}^{N_p-1} \left\| \left(y_i(k-t) - \Theta_{i,j}^\top \begin{bmatrix} 1 \\ x_i(k-t) \end{bmatrix} \right) \delta_{i,j}(k-t) \right\|_2^2 \qquad \text{(A)}$$

$$+ \sum_{t=1}^{k-N_p} \left\| y_i(t) - \Theta_{i,\sigma(t)}^\top \begin{bmatrix} 1 \\ x_i(t) \end{bmatrix} \right\|_2^2 \qquad \text{(B)}$$

$$\text{s.t. } \delta_{i,j}(k-t) \in \{0,1\},\ \sum_{j=1}^{s} \delta_{i,j}(k-t) = 1,\ t = 0, \ldots, N_p-1.$$

Stage S1: Training appliance models



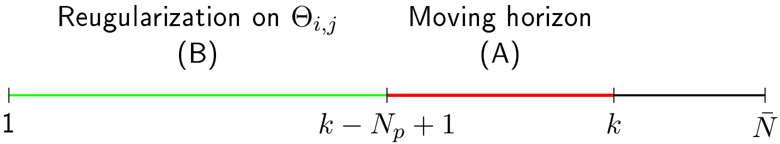
Mixed-Integer Problem

$$\min_{\Theta_{i,j},\ \delta_{i,j}(k-t)} \sum_{j=1}^{s} \sum_{t=0}^{N_p-1} \left\| \left(y_i(k-t) - \Theta_{i,j}^\top \begin{bmatrix} 1 \\ x_i(k-t) \end{bmatrix} \right) \delta_{i,j}(k-t) \right\|_2^2 \qquad \text{(A)}$$

$$+ \sum_{t=1}^{k-N_p} \left\| y_i(t) - \Theta_{i,\sigma(t)}^\top \begin{bmatrix} 1 \\ x_i(t) \end{bmatrix} \right\|_2^2 \qquad \text{(B)}$$

$$\text{s.t. } \delta_{i,j}(k-t) \in \{0,1\}, \sum_{j=1}^{s} \delta_{i,j}(k-t) = 1,\ t = 0, \ldots, N_p-1.$$

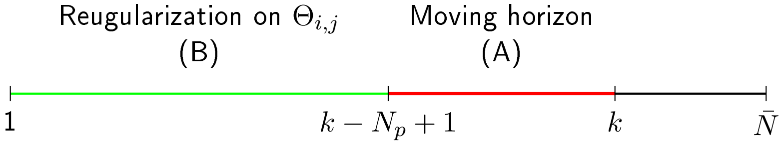
Stage S1: Training appliance models



Mixed-Integer Problem

- Only the active mode $\sigma(k)$ at time k is kept
- the N_p-length time window is shifted forward to estimate the next mode $\sigma(k+1)$ solving the MIQP problem
- the signature of the i-th appliance is captured by the estimated model parameters $\Theta_{i,j}$ for all modes $j = 1, \ldots, s$

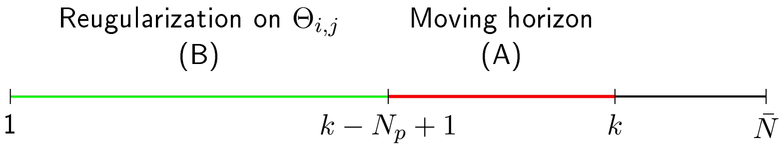
Stage S1: Training appliance models



Mixed-Integer Problem

- Only the active mode $\sigma(k)$ at time k is kept
- the N_p-length time window is shifted forward to estimate the next mode $\sigma(k+1)$ solving the MIQP problem
- the signature of the i-th appliance is captured by the estimated model parameters $\Theta_{i,j}$ for all modes $j = 1, \ldots, s$

Stage S1: Training appliance models



Mixed-Integer Problem

- Only the active mode $\sigma(k)$ at time k is kept
- the N_p-length time window is shifted forward to estimate the next mode $\sigma(k+1)$ solving the MIQP problem
- the signature of the i-th appliance is captured by the estimated model parameters $\Theta_{i,j}$ for all modes $j = 1, \ldots, s$

Stage S2: Energy Disaggregation

- Energy disaggregation problem reduces to determining the operating mode $\delta_{i,j}(k)$ of each *PWA-AR* appliance model.
- We solve the following *Binary Quadratic Program* (BQP)

$$\min_{\{\delta_{i,j}(k)\}_{i,j=1}^{n,s}} \left\| y(k) - \sum_{i=1}^{n} \sum_{j=1}^{s} \Theta_{i,j}^{\top} \begin{bmatrix} 1 \\ \hat{x}_i(k) \end{bmatrix} \delta_{i,j}(k) \right\|_2^2,$$

$$\text{s.t. } \delta_{i,j}(k) \in \{0,1\},\ \sum_{j=1}^{s} \delta_{i,j}(k) = 1,$$

Stage S2: Energy Disaggregation

$$\min_{\{\delta_{i,j}(k)\}_{i,j=1}^{n,s}} \left\| y(k) - \sum_{i=1}^{n} \sum_{j=1}^{s} \Theta_{i,j}^{\top} \begin{bmatrix} 1 \\ \hat{x}_i(k) \end{bmatrix} \delta_{i,j}(k) \right\|_2^2,$$

$$\text{s.t. } \delta_{i,j}(k) \in \{0,1\},\ \sum_{j=1}^{s} \delta_{i,j}(k) = 1,$$

- At each time instance k the BQP is solved iteratively using an estimate $\hat{x}_i(k)$ of the regressor obtained from the previous iterations.
- the active operating mode j^* of each appliance is determined by the solution of the BQP namely

$$j^* : \delta_{i,j^*}(k) = 1.$$

Stage S2: Energy Disaggregation

$$\min_{\{\delta_{i,j}(k)\}_{i,j=1}^{n,s}} \left\| y(k) - \sum_{i=1}^{n}\sum_{j=1}^{s} \Theta_{i,j}^{\top} \begin{bmatrix} 1 \\ \hat{x}_i(k) \end{bmatrix} \delta_{i,j}(k) \right\|_2^2,$$

$$\text{s.t. } \delta_{i,j}(k) \in \{0,1\}, \ \sum_{j=1}^{s} \delta_{i,j}(k) = 1,$$

- At each time instance k the BQP is solved iteratively using an estimate $\hat{x}_i(k)$ of the regressor obtained from the previous iterations.
- the active operating mode j^* of each appliance is determined by the solution of the BQP namely

$$j^* : \delta_{i,j^*}(k) = 1.$$

Stage S2: Energy Disaggregation

$$\min_{\{\delta_{i,j}(k)\}_{i,j=1}^{n,s}} \left\| y(k) - \sum_{i=1}^{n} \sum_{j=1}^{s} \Theta_{i,j}^{\top} \begin{bmatrix} 1 \\ \hat{x}_i(k) \end{bmatrix} \delta_{i,j}(k) \right\|_2^2,$$

$$\text{s.t. } \delta_{i,j}(k) \in \{0,1\}, \ \sum_{j=1}^{s} \delta_{i,j}(k) = 1,$$

- The power of each individual appliance is thus given by

$$\hat{y}_i(k) = \Theta_{i,j^*}^{\top} \begin{bmatrix} 1 \\ \hat{x}_i(k) \end{bmatrix},$$

which is used to construct the regressor $\hat{x}_i(k+1)$.

Application to Real Data

- Test on a benchmark AMPds dataset of house located in Canada.
- We consider the aggregate power consumption by 1) fridge; 2) dish washer; 3) heat pump; 4) clothes dryer.
- The aggregated power is corrupted by a fictitious white noise $e(k) \in \mathcal{N}(0, \sigma_{\mathrm{e}}^2)$ where $\sigma_{\mathrm{e}} = 4$ W.
- **Stage S1**: PWA-AR models of each appliance are tranined using only 500 min data of the day 19.
- **Stage S2**: The power measurements of one month are disaggregated by solving the binary quadratic program.

Application to Real Data

- Test on a benchmark AMPds dataset of house located in Canada.
- We consider the aggregate power consumption by $1)$ fridge; $2)$ dish washer; $3)$ heat pump; $4)$ clothes dryer.
- The aggregated power is corrupted by a fictitious white noise $e(k) \in \mathcal{N}(0, \sigma_{\mathrm{e}}^2)$ where $\sigma_{\mathrm{e}} = 4$ W.
- **Stage S1**: PWA-AR models of each appliance are tranined using only 500 min data of the day 19.
- **Stage S2**: The power measurements of one month are disaggregated by solving the binary quadratic program.

Application to Real Data

- Test on a benchmark AMPds dataset of house located in Canada.
- We consider the aggregate power consumption by $1)$ fridge; $2)$ dish washer; $3)$ heat pump; $4)$ clothes dryer.
- The aggregated power is corrupted by a fictitious white noise $e(k) \in \mathcal{N}(0, \sigma_{\mathrm{e}}^2)$ where $\sigma_{\mathrm{e}} = 4$ W.
- **Stage S1**: PWA-AR models of each appliance are tranined using only 500 min data of the day 19.
- **Stage S2**: The power measurements of one month are disaggregated by solving the binary quadratic program.

Application to Real Data

- Test on a benchmark AMPds dataset of house located in Canada.
- We consider the aggregate power consumption by $1)$ fridge; $2)$ dish washer; $3)$ heat pump; $4)$ clothes dryer.
- The aggregated power is corrupted by a fictitious white noise $e(k) \in \mathcal{N}(0, \sigma_{\mathrm{e}}^2)$ where $\sigma_{\mathrm{e}} = 4$ W.
- **Stage S1**: PWA-AR models of each appliance are tranined using only 500 min data of the day 19.
- **Stage S2**: The power measurements of one month are disaggregated by solving the binary quadratic program.

Application to Real Data

- Test on a benchmark AMPds dataset of house located in Canada.
- We consider the aggregate power consumption by $1)$ fridge; $2)$ dish washer; $3)$ heat pump; $4)$ clothes dryer.
- The aggregated power is corrupted by a fictitious white noise $e(k) \in \mathcal{N}(0, \sigma_{\mathrm{e}}^2)$ where $\sigma_{\mathrm{e}} = 4$ W.
- **Stage S1**: PWA-AR models of each appliance are tranined using only 500 min data of the day 19.
- **Stage S2**: The power measurements of one month are disaggregated by solving the binary quadratic program.

Application to Real Data

- **Stage S1**: Supervised traning phase.
- PWA-AR models with $s = 3$ and $n_a = 2$ are estimated.
- The moving-horizon mixed-integer quadratic programming problem is solved with horizon length $N_p = 5$
- The average computation time is 90 ms using GUROBI.
- For comparison with PWA-AR models, we also consider static device models for energy disaggregation:
  a. fridge: $[\Theta_{1,1}\ \Theta_{1,2}\ \Theta_{1,3}] = [0\ 128\ 200]$ W;
  b. dish washer: $[\Theta_{2,1}\ \Theta_{2,2}\ \Theta_{2,3}] = [0\ 120\ 800]$ W;
  c. heat pump: $[\Theta_{3,1}\ \Theta_{3,2}\ \Theta_{3,3}] = [0\ 39\ 1900]$ W;
  d. clothes dryer: $[\Theta_{4,1}\ \Theta_{4,2}\ \Theta_{4,3}] = [0\ 260\ 4700]$ W.

Application to Real Data

- **Stage S1**: Supervised traning phase.
- PWA-AR models with$s = 3$ and $n_a = 2$ are estimated.
- The moving-horizon mixed-integer quadratic programming problem is solved with horizon length $N_p = 5$
- The average computation time is 90 ms using GUROBI.
- For comparison with PWA-AR models, we also consider static device models for energy disaggregation:
  a. fridge: $[\Theta_{1,1}\ \Theta_{1,2}\ \Theta_{1,3}] = [0\ 128\ 200]$ W;
  b. dish washer: $[\Theta_{2,1}\ \Theta_{2,2}\ \Theta_{2,3}] = [0\ 120\ 800]$ W;
  c. heat pump: $[\Theta_{3,1}\ \Theta_{3,2}\ \Theta_{3,3}] = [0\ 39\ 1900]$ W;
  d. clothes dryer: $[\Theta_{4,1}\ \Theta_{4,2}\ \Theta_{4,3}] = [0\ 260\ 4700]$ W.

Application to Real Data

- **Stage S1**: Supervised traning phase.
- PWA-AR models with$s = 3$ and $n_a = 2$ are estimated.
- The moving-horizon mixed-integer quadratic programming problem is solved with horizon length $N_p = 5$
- The average computation time is 90 ms using GUROBI.
- For comparison with PWA-AR models, we also consider static device models for energy disaggregation:
  a. fridge: $[\Theta_{1,1}\ \Theta_{1,2}\ \Theta_{1,3}] = [0\ 128\ 200]$ W;
  b. dish washer: $[\Theta_{2,1}\ \Theta_{2,2}\ \Theta_{2,3}] = [0\ 120\ 800]$ W;
  c. heat pump: $[\Theta_{3,1}\ \Theta_{3,2}\ \Theta_{3,3}] = [0\ 39\ 1900]$ W;
  d. clothes dryer: $[\Theta_{4,1}\ \Theta_{4,2}\ \Theta_{4,3}] = [0\ 260\ 4700]$ W.

Application to Real Data

- **Stage S1**: Supervised traning phase.
- PWA-AR models with$s = 3$ and $n_a = 2$ are estimated.
- The moving-horizon mixed-integer quadratic programming problem is solved with horizon length $N_p = 5$
- The average computation time is 90 ms using GUROBI.
- For comparison with PWA-AR models, we also consider static device models for energy disaggregation:
  a. fridge: $[\Theta_{1,1}\ \Theta_{1,2}\ \Theta_{1,3}] = [0\ 128\ 200]$ W;
  b. dish washer: $[\Theta_{2,1}\ \Theta_{2,2}\ \Theta_{2,3}] = [0\ 120\ 800]$ W;
  c. heat pump: $[\Theta_{3,1}\ \Theta_{3,2}\ \Theta_{3,3}] = [0\ 39\ 1900]$ W;
  d. clothes dryer: $[\Theta_{4,1}\ \Theta_{4,2}\ \Theta_{4,3}] = [0\ 260\ 4700]$ W.

Application to Real Data

- **Stage S2**: Energy Disaggregation.
- Once PWA-AR models are estimated, *Binary Quadratic Program* is solved to disaggregated the power measurements of one month.
- The average CPU time taken is 8 ms using GUROBI.
- The quality of the energy disaggregation results is assessed via:

  a. *Energy Fraction Index* (EFI)
  b. *Relative Square Error (RSE) and R^2 coefficient*
  c. *Total Energy Correctly Assigned* (TECA)

Application to Real Data

- **Stage S2**: Energy Disaggregation.
- Once PWA-AR models are estimated, *Binary Quadratic Program* is solved to disaggregated the power measurements of one month.
- The average CPU time taken is 8 ms using GUROBI.
- The quality of the energy disaggregation results is assessed via:

  a. *Energy Fraction Index* (EFI)
  b. *Relative Square Error (RSE) and R^2 coefficient*
  c. *Total Energy Correctly Assigned* (TECA)

Application to Real Data

- **Stage S2**: Energy Disaggregation.
- Once PWA-AR models are estimated, *Binary Quadratic Program* is solved to disaggregated the power measurements of one month.
- The average CPU time taken is 8 ms using GUROBI.
- The quality of the energy disaggregation results is assessed via:

  a. *Energy Fraction Index* (EFI)
  b. *Relative Square Error (RSE) and R^2 coefficient*
  c. *Total Energy Correctly Assigned* (TECA)

Application to Real Data

Energy Fraction Index (EFI)

$$\hat{h}_i = \frac{\sum_{k=1}^{N} \hat{y}_i(k)}{\sum_{i=1}^{n} \sum_{k=1}^{N} \hat{y}_i(k)}$$

quantifies the estimated fraction of total energy consumed by the i-th appliance.

	PWA-AR models $\hat{h}_i$	static models $\hat{h}_i$	ground truth h_i
Fridge	19.6 %	14.9 %	21.3 %
Dish washer	6.8 %	11.4 %	5.1 %
Heat pump	41.6 %	42.0 %	42.3 %
Clothes dryer	31.9 %	31.6 %	31.3 %

Application to Real Data

Relative Square Error (RSE) and R^2 coefficient

$$\mathrm{RSE}_i = \frac{\sum_{k=1}^{N} \left(y_i(k) - \hat{y}_i(k)\right)^2}{\sum_{k=1}^{N} y_i^2(k)}$$

$$R_i^2 = 1 - \frac{\sum_{k=1}^{N} \left(y_i(k) - \hat{y}_i(k)\right)^2}{\sum_{k=1}^{N} \left(y_i(k) - \bar{y}_i\right)^2}$$

RSE_i and R_i^2 measure the match between the actual and the estimated power profiles over time.

	PWA-AR models		static models	
	RSE_i	R_i^2	RSE_i	R_i^2
Fridge	15.5 %	76.5 %	35.9 %	45.5 %
Dish washer	12.4 %	87.3 %	38.0 %	61.4 %
Heat pump	0.6 %	99.3 %	4.0 %	95.6 %
Clothes dryer	0.1 %	99.9 %	0.3 %	99.7 %

Application to Real Data

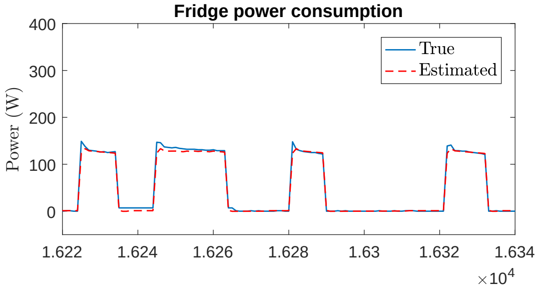
Total Energy Correctly Assigned (TECA)

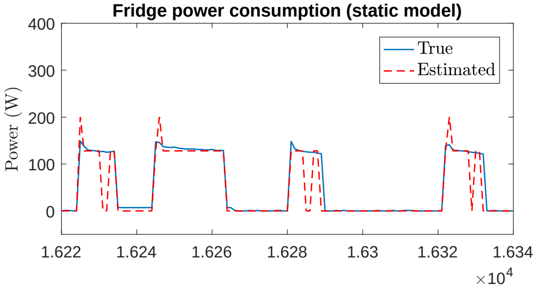
$$\text{TECA} = 1 - \frac{\sum_{k=1}^{N} \sum_{i=1}^{n} |\hat{y}_i(k) - y_i(k)|}{2 \sum_{k=1}^{N} y(k)},$$

quantifies the percentage of energy correctly classified.

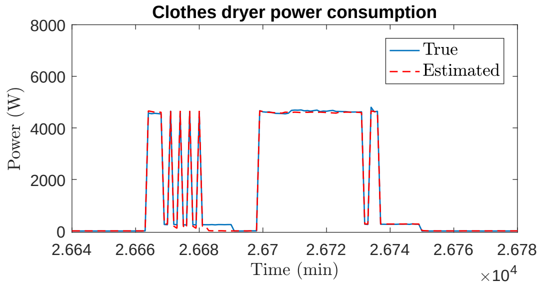
	PWA-AR models	static models
TECA	95.3 %	89.4 %

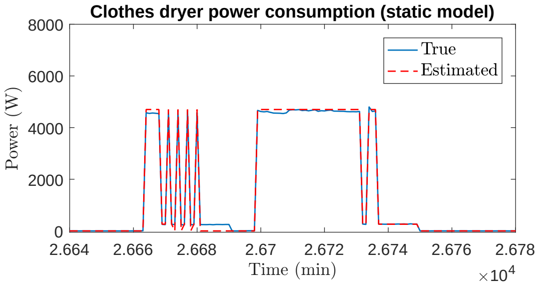
Application to Energy Disaggregation



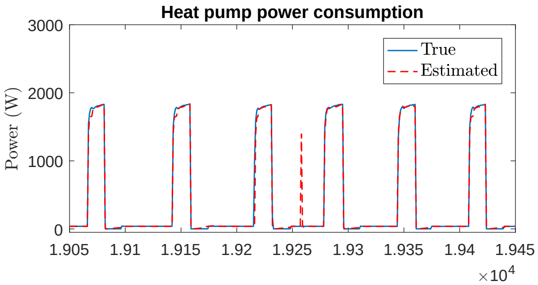


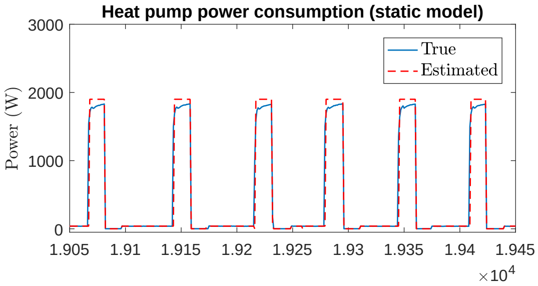
Application to Energy Disaggregation





Application to Energy Disaggregation





Conclusions

- A two-stage supervised algorithm for energy disaggregation is proposed.
- The dynamic PWA-AR modeling of the power profiles of individual appliances leads to better energy disaggregation results compared to the same approach relying on static models.
- The proposed method is computationally efficient as the appliance models can be estimated off-line only once, while energy disaggregation is performed online with low computational complexity.

Conclusions

- A two-stage supervised algorithm for energy disaggregation is proposed.
- The dynamic PWA-AR modeling of the power profiles of individual appliances leads to better energy disaggregation results compared to the same approach relying on static models.
- The proposed method is computationally efficient as the appliance models can be estimated off-line only once, while energy disaggregation is performed online with low computational complexity.

Conclusions

- A two-stage supervised algorithm for energy disaggregation is proposed.
- The dynamic PWA-AR modeling of the power profiles of individual appliances leads to better energy disaggregation results compared to the same approach relying on static models.
- The proposed method is computationally efficient as the appliance models can be estimated off-line only once, while energy disaggregation is performed online with low computational complexity.

Thank You