(Supplementary Material) SmartD: Smart Meter Data Analytics Dashboard *

Aylin Jarrah Nezhad, Tri Kurniawan Wijaya, Matteo Vasirani, and Karl Aberer
School of Computer and Communication Sciences
École Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland
{aylin.jarrahnezhad, tri-kurniawan.wijaya, matteo.vasirani, karl.aberer}@epfl.ch

1. SUPPLEMENT FOR SECTION 2.1

Consumer aggregations.

A simple example of a grammar which expresses consumer selection/aggregation: string "$1; 2; 3, 4, 5$" corresponds to the visualization of the energy data of consumer 1 alone, consumer 2 alone, and a cluster composed of consumers 3, 4, and 5.

2. SUPPLEMENT FOR SECTION 2.2

Estimating typical load profile.[1]

SmartD is able to estimate consumer typical load profile given her demographics and contextual information. Let D be the set of demographic information, and C be the context that we are interested in. The set of demographic information, D, can be, for example: a family with two children, live in 2000 sq. ft. apartment, and own a dishwasher. The context, C, can be, for example: weekdays in January, or Monday in the summer. In addition, let N be the set of $k \in \mathbb{N}$ consumers with the closest demographics to D. Thus, $|N| = k$. Then, the estimated load profile of consumers with demographics D on context C is the average of (hourly) load profile of consumers in N.

A question remains, however, to decide the best k. Should k be 1, 2, 3, or something else? To answer this, for each k under consideration, we perform *leave-one-out-cross-validation*. See Algorithm 2.1 for details. For load profiles L_i and L_j, function $dist(L_i, L_j)$ return the distance between L_i and L_j. It can be computed, for example, using the difference between the norm of L_i and L_j.

Algorithm 2.1: Find the best k

Input: a set of consumers $\mathcal{A}$, a set of k under consideration $\mathcal{K} = \{k_1, \dots k_n\}$, contextual information C
Output: the best $k \in \mathcal{K}$

```
foreach k ∈ K do
    δ_k ← 0
    foreach i ∈ A do
        A' ← A \ i
        Let N be the set of k consumers in A' having the
        closest demographics to i
        L_i ← (hourly) load profile of i
        L_N ← average (hourly) load profile of consumers in
        N on context C
        δ_k ← δ_k + dist(L_i, L_N)
return arg min_k(δ_k)
```

Discovering significant demographic characteristics.

SmartD is also able to infer demographic information that significantly influences energy consumption on a specific context, e.g., weekdays in January, or Monday in the summer. For this purpose, we use a supervised feature selection algorithm, namely *correlation-based feature selection*.[2] We refer to this algorithm as *cfs*.

Let an *instance* be a tuple (F, l), where $F = \{f_1, \dots, f_{|F|}\}$ is a feature set and l is a target attribute. Given a set of instances $\mathcal{I}$, applying *cfs* to $\mathcal{I}$ results in the set R of indexes of the features that are deemed to be relevant to the target attributes. Formally $cfs(\mathcal{I}) = R = \{r_1, \dots, r_{|R|}\}$ such that $r_i \subseteq \{1, \dots, |F|\}$.

Next, we explain how to infer top-q demographic characteristics which are relevant to the energy consumption for a context C. To make it clearer, we also illustrate the steps in Figure 1. Let $D = \{d_1, \dots, d_{|D|}\}$ be the set of consumer demographics. We define F_i as the feature set of consumer i, where each of its element is consumer i's demographic information. Thus $|F| = |D|$. Let l_i^h be the average of hourly energy consumption of consumer i, on context C, at hour $1 \leq h \leq 24$. Further, let $\mathcal{A}$ be the set of consumers, and $\mathcal{I}^h$ be the set of instances for hour h, consist of tuples (F_i, l_i^h) for all consumers $i \in \mathcal{A}$.

For $1 \leq h \leq 24$, let $cfs(\mathcal{I}^h) = R^h$. Then, we define $score(r) = |\{R^h \mid r \in R^h, 1 \leq h \leq 24\}|$, for $1 \leq r \leq |F|$. The top-q demographic characteristics of the set of consumers $\mathcal{A}$ on context C is the q demographics $d_{r_1^*}, \dots, d_{r_q^*}$ with the highest scores. That is, the top-q demographics are $d_{r_1^*}, \dots, d_{r_q^*}$, where $score(r^*) \geq score(r)$ for all $r^* \in \{r_1^*, \dots, r_q^*\}$ and $r \in \{1, \dots, |F|\} \setminus \{r_1^*, \dots, r_q^*\}$.

[1] We use the terms *energy consumption* and *load* interchangeably.

[2] See the bibliographic information for this method in the main paper.

3. POST E-ENERGY SUBMISSION

After we submit SmartD for a demo to e-Energy 2014, we still continue to develop it. Below are the functionalities that we added after the submission. SmartD's source code is available at `https://github.com/LSIR/smartd`.

3.1 Multilingual Support

When Smart first developed, the (only) language of the application is English. However, we expect SmartD to be used world-

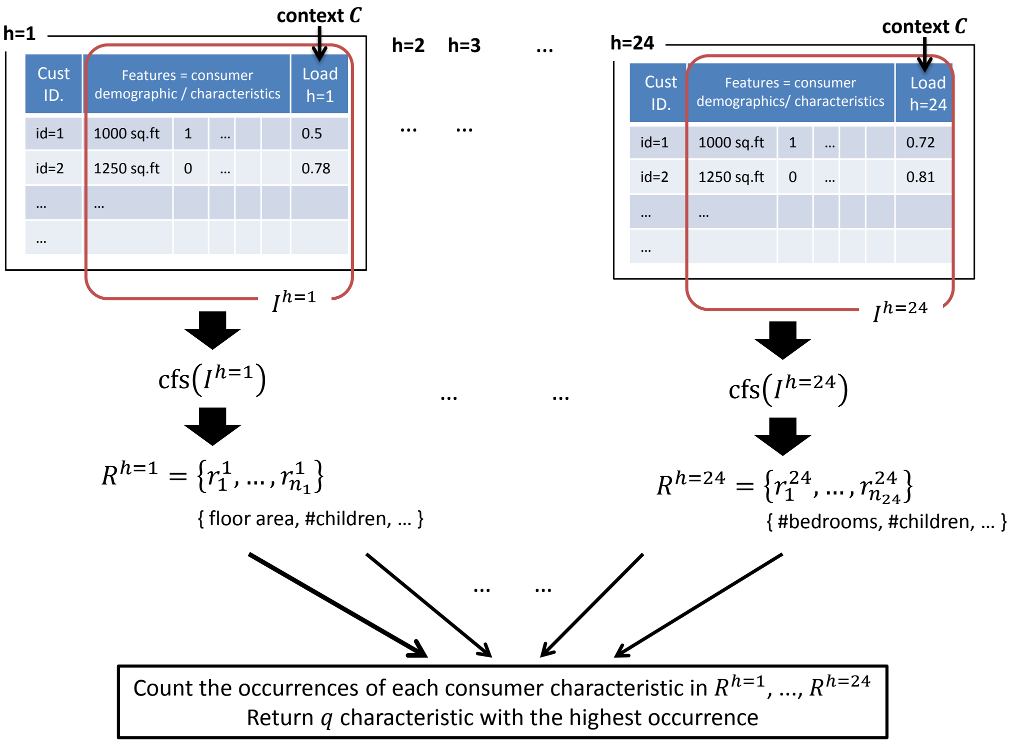

Figure 1: Illustration on how SmartD discovers consumers' characteristics which define their energy consumption profile.

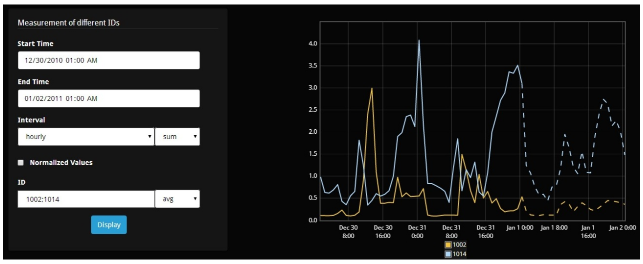

Figure 2: SmartD energy consumption analysis extended with forecasting. The forecasted values are shown by the dashed lines.

wide (since smart meters are also deployed world-wide), it might help the user to interact with SmartD in their own mother tongue. As the first step, we add French language.[3] Developers can add other languages easily by providing the translation of the SmartD's user interface label in the targeted languages in `gsn/webapp/js/smartd-languages` and providing the function to load the language in `gsn/webapp/smartd*.html`.

3.2 Forecasting

Electricity load forecasting is another important analytics task in the smart energy domain. See, for example, our previous work and its bibliographic information [1]. Although in principle, any forecasting algorithm can be used, we extend SmartD by incorporating a simple (and interpretable) forecasting method. In some application domain, e.g. demand response (DR), simplicity and interpretability of the models is required (in addition to accuracy). For instance, since DR baseline (or the forecasted demand) plays a key role in determining consumer's incentive/payment, the method to produce DR baseline should be comprehensible by the stakeholders (the utility company/DR providers and the consumers). In this extension, we use one of the methods to compute DR baseline (which is essentially a forecasting algorithm): ISONE. [4] A more thorough discussion about DR baseline and the performance can be found in [2]. Currently, SmartD is able to display the forecasted demand up to x days ahead the latest day of the measurements, where x is a user predefined parameter.[5] Figure 2 shows the consumer load profiles with the forecasted values in dashed lines.

[3] French is the language spoken in Lausanne, Switzerland, where our university is located.

[4] We use the implementation in `https://github.com/tritritri/baselines`. Since this repository also contains various other methods, we can easily change ISONE with other methods as well. This can be done by changing the method call parameter in `smartd-forecasting/src/ch/epfl/lsir/smartd/forecast/Forecast.java`.

[5] See variable `forecast-horizon` in the configuration file `system.config` in directory `smartd-forecasting`.

4. REFERENCES

[1] S. Humeau, T. K. Wijaya, M. Vasirani, and K. Aberer. Electricity load forecasting for residential customers: Exploiting aggregation and correlation between households. In *Sustainable Internet and ICT for Sustainability (SustainIT), 2013*, pages 1–6, Oct 2013.

[2] T. Wijaya, M. Vasirani, and K. Aberer. When bias matters: An economic assessment of demand response baselines for residential customers. *IEEE Transactions on Smart Grid*, 2014. doi:10.1109/TSG.2014.2309053.