Electric Load Disaggregation in Smart Metering Using a Novel Feature Extraction Method and Supervised Classification

Khaled Chahine*, Khalil El Khamlichi Drissi, Christophe Pasquier, Kamal Kerroum, Claire Faure, Thierry Jouannet, Michel Michou

*LASMEA, UMR 6602 UBP/CNRS, Université Blaise Pascal, Clermont-Ferrand II, 24 Avenue des Landais, 63177 Aubière Cedex, France

Landis+Gyr, 30 Avenue du Président Vincent Auriol, 03100 Montluçon, France

Abstract

Improving energy efficiency by monitoring household electrical consumption is of significant importance with the present-day climate change concerns. A solution for the electrical consumption management problem is the use of a nonintrusive appliance load monitoring system. This system captures the signals from the aggregate consumption, extracts the features from these signals and classifies the extracted features in order to identify the switched on appliances. This paper complements a novel feature extraction scheme presented in a previous work for load disaggregation with a comparative study of supervised classification methods. The objective of the current work is hence to make use of the feature extraction scheme to construct a database of signatures and then to compare different supervised learning methods for load classification. Preliminary results indicate high classification accuracy of all tested methods.

Keywords: Matrix Pencil Method, Nonintrusive Appliance Load Monitoring, Supervised Learning, Power Saving, Smart Meters

1. Introduction

The basic premise behind the drive for the installation of smart meters in homes and businesses is that they make it easier for consumers to monitor their energy consumption, thereby making it easier for them to save energy, carbon emissions and money. To help customers as well as utilities in the monitoring process, researchers have been studying load disaggregation schemes for almost two decades.

One method of load disaggregation is distributed direct sensing which requires a sensor at each device or appliance in order to measure consumption. Although conceptually straightforward and potentially highly accurate, direct sensing is often expensive due to time consuming installation and the requirement for one sensor for each device or...
appliance. In response to limitations with the direct sensing approach, researchers have explored methods to infer disaggregated energy usage via a single sensor. Pioneering work in this area is nonintrusive appliance load monitoring (NIALM), first introduced by George Hart in the late 1980s [1]. In contrast to the direct sensing methods, NIALM relies solely on single-point measurements of voltage and current on the power feed entering the household. NIALM consists of four steps: data acquisition, event detection, feature extraction, and event classification. The raw current and voltage waveforms are transformed into a feature vector, i.e., a more compact and meaningful representation that may include real power, reactive power, and harmonics. These extracted features are monitored for changes, identified as events (e.g., an appliance turning “on” or “off”), and classified down to the appliance or device category level using a classification algorithm, which compares the features to a preexisting database of signatures.

This work builds on the novel feature extraction scheme presented in [2, 3] to construct a database of signatures. The core of the feature extraction phase is the Matrix Pencil Method (MPM) which represents the electric current in terms of complex poles and residues (see figure 1) [4], and its output is a feature vector whose elements are the pole-residue products. For example, the signature of a purely resistive charge is characterized by two conjugate poles at the grid frequency (e.g., ±50 Hz), and thus its feature vector contains one element (the sum of two conjugate pole-residue products) since only one frequency contributes to the drawn current. The database of signatures is then used to train and compare three supervised classification methods: $k$-Nearest Neighbor, Naive Bayes, and Support Vector Machine.

2. Feature Extraction

The drawn electric current can be modeled as a linear combination of $M$ cisoids (complex-valued sinusoidal signals) weighted by complex residues according to the following equation:

$$i(t) \approx \sum_{m=1}^{M} r_m \exp \{(\alpha_m + j2\pi f_m)t\} + b(t) \quad (1)$$

where $r_m$ is the residue of the $m$th cisoid, $\alpha_m$ is its attenuation factor, $f_m$ is its frequency, and $b(t)$ is additive white Gaussian noise. After sampling, the time variable, $t$, is replaced by $t_k = kt_s$, where $t_s = 6.25 \times 10^{-4}$ is the sampling period. The discrete current signal becomes:

$$i(k) \approx \sum_{m=1}^{M} r_m z_m^k + b(k) \quad k = 1, 2, \ldots, N \quad (2)$$

where

$$z_m = \exp \{(\alpha_m + j2\pi f_m)t_s\} \quad m = 1, 2, \ldots, M. \quad (3)$$

The feature extraction problem can now be stated as follows. Given the electric current data sequence $\{i(k)\}_{k=1}^{N}$, use MPM to extract the complex poles $\{z_m\}_{m=1}^{M}$ and residues $\{r_m\}_{m=1}^{M}$ of the appliance.
3. Supervised Multi-Class Classification

Supervised multi-class classification algorithms aim at assigning a class label for each input example. Given a training data set of the form \((x_i, y_i)\), where \(x_i \in \mathbb{R}^n\) is the \(i^{th}\) example and \(y_i \in \{1, ..., K\}\) is the \(i^{th}\) class label, the algorithms aim at finding a learning model \(\Lambda\) such that \(\Lambda(x_i) = y_i\) for new unseen examples. The problem is simply formulated in the two-class case, where the labels \(y_i\) are just +1 or -1 for the two classes involved. Several algorithms have been proposed to solve this problem in the two-class case, some of which can be naturally extended to the multi-class case, and some that need special formulations to be able to solve the latter case. The first category of algorithms include decision trees, neural networks, \(k\)-Nearest Neighbor, Naive Bayes classifiers, and Support Vector Machines. The second category include approaches for converting the multi-class classification problem into a set of binary classification problems that are efficiently solved using binary classifiers.

3.1. \(k\)-Nearest Neighbor

\(k\)-Nearest Neighbor (\(k\)-NN) is considered among the oldest non-parametric classification algorithms. To classify an unknown example, the distance (using some distance measure, e.g. Euclidean) from that example to every other training example is measured [5]. The \(k\) smallest distances are identified, and the most represented class in these \(k\) classes is considered the output class label. The value of \(k\) is generally determined using a validation set or using cross-validation.

3.2. Naive Bayes

Naive Bayes is based upon the principle of maximum a posteriori (MAP) [6]. Given a problem with \(K\) classes \(\{C_1, ..., C_K\}\) with so-called prior probabilities \(\{P(C_1), ..., P(C_K)\}\), we can assign the class label \(c\) to an unknown example with features \(x = (x_1, ..., x_N)\) such that \(c = \max_c P(C = c|x_1, ..., x_N)\), that is choose the class with the maximum
a posteriori probability given the observed data. This a posteriori probability can be formulated, using Bayes theorem, as follows:

$$P(C = c| x_1, \ldots, x_N) = \frac{P(C = c)P(x_1, \ldots, x_N| C = c)}{P(x_1, \ldots, x_N)}.$$  

As the denominator is the same for all classes, it can be dropped from the comparison. Now, we should compute the so-called class conditional probabilities of the features given the available classes. This can be quite difficult taking into account the dependencies between features. The naive Bayes approach is to assume class conditional independence, i.e. $x_1, \ldots, x_N$ are independent given the class. This simplifies the numerator to be $P(C = c)P(x_1| C = c) \ldots P(x_N| C = c)$, and then choosing the class $c$ that maximizes this value over all the classes $c = 1, \ldots, K$. Clearly this approach is naturally extensible to the case of having more than two classes, and was shown to perform well despite the underlying simplifying assumption of conditional independence.

### 3.3. Support Vector Machine

Support Vector Machines (SVM’s) are a relatively new learning method used mainly for binary classification [7]. The basic idea is to find a maximum-margin hyperplane which separates the $n$-dimensional data perfectly into its two classes. However, since example data is often not linearly separable, SVM’s introduce the notion of a kernel-induced feature space which casts the data into a higher-dimensional space where the data is separable. Typically, casting into such a space would cause problems computationally and with overfitting. The key insight used in SVM’s is that the higher-dimensional space does not need to be dealt with directly (as it turns out, only the formula for the dot-product in that space is needed), which eliminates the above concerns.
Table 1: Classification accuracy of k-NN, Naive Bayes, and SVM.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>k-NN</th>
<th>Naive Bayes</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incandescent</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Halogen</td>
<td>100%</td>
<td>97%</td>
<td>100%</td>
</tr>
<tr>
<td>Economy</td>
<td>100%</td>
<td>99%</td>
<td>94%</td>
</tr>
<tr>
<td>Water Heater</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Convector</td>
<td>96%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Oven</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Hot Plate</td>
<td>97%</td>
<td>50%</td>
<td>73%</td>
</tr>
<tr>
<td>Television</td>
<td>100%</td>
<td>100%</td>
<td>98%</td>
</tr>
<tr>
<td>Computer</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Overall Accuracy</td>
<td>99.22%</td>
<td>94%</td>
<td>96.11%</td>
</tr>
</tbody>
</table>

4. Performance Evaluation

In this section, a performance comparison of the previously discussed classification methods is presented. The training set contains 900 examples uniformly representing nine classes: {Incandescent Lamp, Halogen Lamp, Economy Lamp, Water Heater, Electric Convector, Oven, Hot Plate (one and two burners), Television, Computer}. As shown in figure figure 2, each example (represented by a point in the feature space) is characterized by three pole-residue products corresponding to the fundamental, third and fifth harmonic currents.

Optimal classifier parameters, if they exist, are determined by means of a grid search. For example, there are two parameters for the RBF kernel of SVM: $C$ and $\gamma$. It is not known beforehand which $C$ and $\gamma$ are best for a given problem. Consequently, some kind of model selection (parameter search) must be done. The goal is to identify good $(C, \gamma)$ so that the classifier can accurately predict unknown data (i.e. testing data). This can be achieved by separating the training set into two parts, of which one is considered unknown. The prediction accuracy obtained from the “unknown” set more precisely reflects the performance on classifying an independent data set. An improved version of this procedure is known as cross-validation. In $\nu$-fold cross-validation, we first divide the training set into $\nu$ subsets of equal sizes. Sequentially, one subset is tested using the classifier trained on the remaining $\nu - 1$ subsets. Thus, each instance of the whole training set is predicted once so the cross-validation accuracy is the percentage of data which are correctly classified.

Table 1 shows the classification accuracy evaluated from 900 cross-validations. The obtained results indicate that three classification methods are comparably successful with $k$-NN having the highest classification accuracy. The significantly lower accuracy obtained by Naive Bayes and SVM for the hot plate is due to overlapping feature vectors (refer to figure 2).

5. Summary and Conclusion

In this paper, we compared three supervised classification methods for electric load disaggregation. Owing to the ability of MPM to distinguish the contribution of each existing frequency to the total drawn current, all three methods performed comparably
well. The final choice of a method, however, should take into consideration the technological limitations of the micro-controller (processing power and memory size). The reason is that for most methods the speed of learning is in negative correlation with the speed of classification, i.e., the slower the speed of learning, the faster the speed of classification, and vice versa. Consequently, unless the learning step of SVM is conducted offline, it is impossible for this method to be integrated in a smart energy meter, and attention should be drawn to alternative methods such as Naive Bayes and $k$-NN.

References