Supervised Analysis Dictionary Learning: Application in Consumer Electronics Appliance Classification

P. Bhattacharjee
IIIT Delhi
protim1488@iiitd.ac.in

S. Banerjee
Xerox Research Center
Shisagnee.Banerjee@xerox.com

M. Gulati
IIIT Delhi
manojg@iiitd.ac.in

A. Majumdar
IIIT Delhi
angshul@iiitd.ac.in

S. S. Ram
IIIT Delhi
shobha@iiitd.ac.in

ABSTRACT
The objective of this paper is to estimate if an electrical appliance is 'ON' based on their common mode electromagnetic (CM EMI) emissions. The assumption being that, a user by knowing the state of the appliance can make an informed decision whether to keep it running or switch it off to save power. Here, state estimation of a single appliance is formulated as a classification problem. A new technique called *analysis dictionary learning* is proposed to generate features from CM EMI. The proposed method outperforms feature extraction based on deep learning techniques as well as a state-of-the-art information theoretic feature extraction technique based on Conditional Likelihood Maximization.

CCS Concepts
• Computing methodologies • Machine learning • Learning paradigms • Supervised Learning • Applied computing • Physical sciences and engineering • Electronics.

Keywords
Non-intrusive load monitoring; supervised learning; dictionary learning.

1.INTRODUCTION
Residential and commercial buildings account for 40% of the global energy consumption [1]. With the rapid growth in infrastructure, there is an urgent need to reduce energy consumption. The twin goals for long-term energy sustainability can be identified as (1) use of energy efficient appliances to reduce energy consumption (2) reducing energy consumption through appliance identification and optimizing appliance operations in non-working hours. In this work, we are focusing on the latter goal, where the objective is to accurately detect and classify appliances that are operational on a power line. Broadly, this work falls under the area of Non-intrusive load monitoring (NILM), which has been extensively researched in the past.

NILM techniques try to infer state and power consumption of individual appliances by electrical signals measured from a single sensing point (usually smart meter) on the power line. However, this is usually an overkill; in residential and commercial buildings, the power consumed is not of much importance, but the state of the appliance is. For example, users have a fair idea of how much power individual appliances like a heater or an AC consume. As long as the users have the information on which appliances are operational, they can take an informed decision whether to turn it off or not. Also, NILM techniques claim to be capable of disaggregating power consumption of several appliances using total power reading of the smart meter acquired at regular intervals of time. A few household appliances, such as electric toasters and irons, are simple on/off loads that show a sharp increase in the power consumption when switched on. These patterns can be identified with data gathered from smart meters [3]. However, most of the residential and commercial appliances are not simple on/off loads. They exhibit multi-state (refrigerator, AC, washer, etc.) or continuously time varying (CPU, printers, etc.) power consumption behavior [4]. Detecting the operation of these appliances has challenged researchers in the past few years. Multi-state loads can be modeled by stochastic finite state machines (Factorial Hidden Markov Model or Product of Experts), but the continuously varying loads are hard to model by such classical techniques.

In recent years, an alternate technique to smart meter sensing has emerged as a viable method for detecting time-varying power patterns in appliances. The method utilizes unique electromagnetic emissions, generated by the switched mode power supplies within the appliances, to infer appliance usage on the power lines [5-7]. The electromagnetic emissions are of two types: the differential signal between the phase and neutral power lines; and the common mode signal between these lines and the earth. The common-mode signal was demonstrated as a far more robust feature vector for classification in comparison to the differential signal in [6]. This is because of the primary power signal (110V/230V) and its harmonics, which interfere heavily with the differential signal [6], are not present in the common-mode signal measurements. Also, most appliances, today, are required to be fitted with high-quality differential mode filters that regulate their emissions. Therefore, Differential Mode Electromagnetic (DM EMI) emissions form an unreliable feature for detecting appliances, and classification techniques yield poor results.

All prior studies in this area [5-7] use empirical-analytic tools; they are not based on theoretically well-known machine learning approaches. Our work is based on the foundations of dictionary learning; a topic theoretically well understood and highly successful for similar research areas like computer vision and biometrics. We believe dictionary learning is specifically suited to this problem since it can be used to model the complex electrical signatures of the appliances. There are a plethora of papers on supervised dictionary learning, but these are all synthesis techniques, i.e. they learns a dictionary so that the features (when multiplied with the dictionary) can synthesize the data. The main limitation of this approach is that it has a relatively slow execution time since an iterative optimization problem needs to be solved while generating the features during operation.

In this work a whole new framework for dictionary learning – analysis dictionary learning is developed. The dictionary is learnt

CODS '17, March 09-11, 2017, Chennai, India
 ISBN 978-1-4503-4846-1/17/03…$15.00
DOI: 10.1145/3041823.3041825

such that it will generate features while analyzing the data. This is different from all prior dictionary learning methods; where the data is synthesized / re-generated given the dictionary and the features. One should not confuse the proposed method with analysis KSVD (to be discussed later). Aanalysis KSVD was designed for solving denoising problems; it can generate features only in a roundabout fashion. Furthermore it is extremely slow; running the algorithm on a standard desktop for a simple dataset like MNIST takes about a week. The main operational benefit of our proposed analysis dictionary learning is that the operation time is very fast. One does not need to solve a complex optimization problem to generate the features from the test samples. The features are generated, simply by multiplying the data with the learned dictionary.

To summarize, the contributions of this work are as follows:

1. A new signal for estimating the state is proposed based on CM EMI emission.

2. A new technique called analysis dictionary learning is proposed for data analysis.

The paper is organized as follows. In the following section, an overview of synthesis dictionary learning is presented. In Section 3, the proposed technique – analysis dictionary learning is described in detail. In Section 4, the detail of the sensor used to collect the CM EMI data from multiple appliances is given. Experimental results are presented in Section 5. The conclusions of this work and future directions are discussed in Section 6.

2.Literature Review

Early studies in dictionary learning wanted to learn a basis for representation. This is the synthesis formulation, where the training data (X) is represented as: $X=DZ$ where D and Z are the learned dictionary and the coefficients respectively. There were no constraints on the dictionary atoms or on the loading coefficients. The method of optimal directions [8] was used to solve the learning problem by alternately updating the dictionary and the coefficients via least squares.

$$\min_{D,Z} \|X - DZ\|_F^2 \tag{1}$$

Here $\|\cdot\|_F^2$ denotes the Frobenius norm. Today, this problem, (1) is known by the name of matrix factorization.

For problems in sparse representation, the objective is to learn a basis that can represent the samples in a sparse fashion, i.e. Z needs to be sparse. The KSVD [9, 10] is the most well-known technique for solving this problem. Fundamentally, it solves a problem of the form:

$$\min_{D,Z} \|X - DZ\|_F^2 \text{ such that } \|Z\|_0 \le \tau \tag{2}$$

Here $\|\cdot\|_0$ is defined as the number of non-zero elements. It is not exactly a norm in the true sense of the term, but is a diversity measure. Solving the l_0-norm minimization problem is NP hard [11]. KSVD employs the greedy (sub-optimal) orthogonal matching pursuit (OMP) [12] to solve the l_0-norm minimization problem approximately. The major disadvantage of KSVD is that, it is a relatively slow technique owing to its requirement of computing the SVD (singular value decomposition) in every iteration. There are alternate ways to formulate this problem such that the solution is more efficient. These variants [13], [14] usually propose solving the following problem instead.

$$\min_{D,Z} \|X - DZ\|_F^2 + \lambda \|Z\|_1 \tag{3}$$

where $\|\cdot\|_1$ is defined as the sum of absolute values and is the closest convex envelope of the NP hard l_0-norm. This formulation can be solved efficiently without resorting to SVD or other such computationally intensive steps. However, in every iteration, the columns of D need to be normalized. This is to prevent the degenerate solution where D is very large and Z is very small.

First a discussion on the differences between synthesis and analysis dictionary learning is in order. The discussion so far has been limited to synthesis dictionary learning. Analysis KSVD [15] assumes that the training data, when analyzed by the learned dictionary, will produce sparse coefficients. The analysis KSVD formulation solves the following:

$$\min_{D,X_0} \|X - X_0\|_F^2 \text{ such that } \|DX_0\|_0 \le \tau \tag{4}$$

This formulation looks for a 'clean' representation of the signal and learns a dictionary such that the clean representation, when analyzed by this dictionary, is sparse. Such a formulation appears restrictive and seems only suitable for solving inverse problems like denoising and restoration. This does not produce any features / coefficients that can be used for any classification problems.

It is not possible to review the plethora of different supervised synthesis dictionary learning techniques. Initial techniques proposed naïve approaches, which learnt specific dictionaries for each class [16, 17]. Later approaches incorporated discriminative penalties into the dictionary-learning framework. One such technique is to include softmax discriminative cost function [18], [19]; other discriminative penalties include Fisher discrimination criterion [20], linear predictive classification error [21] and hinge loss function [22].

To the best of our knowledge dictionary learning based techniques have not been employed for energy analytics. There are two papers on tensor factorization [23, 24] that are in a different context.

3.Proposed Analysis Dictionary Learning

In traditional dictionary learning, a dictionary is learnt such that it can synthesize the data from the learned coefficients (Figure 1b). Here an alternative is proposed – an analysis dictionary, which generates sparse coefficients when operated on by the data (Figure 1a).

Mathematically the analysis model is represented as:

$$DX = Z \tag{5}$$

where the symbols have their usual meaning.

The learning problem can be formulated as follows:

$$\min_{D,Z} \|DX - Z\|_F^2 + \lambda_1 \|D\|_2 + \lambda_2 \|Z\|_1 \text{ s.t. } \|d_i\| = 1 \tag{6}$$

The l_2-norm on D is for regularization and the l_1-norm on Z promotes sparsity. The columns of D (d_i) are normalized so as to prevent the trivial solution $D=0$ and $Z=0$. The framework in (6) can be solved easily using alternating minimization of D and Z.

$$D_k : \min_D \|DX - Z_{k-1}\|_F^2 + \lambda_1 \|D\|_2 \tag{7a}$$

$$Z_k : \min_Z \|D_k X - Z\|_F^2 + \lambda_2 \|Z\|_1 \quad (7b)$$

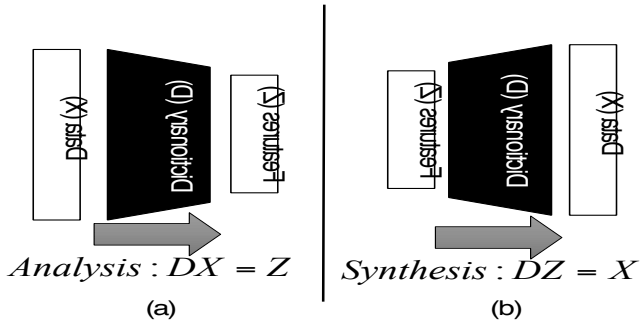

Figure 1. Analysis vs. synthesis.

Here '*k*' denotes iteration number. The first sub-problem (7a) is a least squares problem and has a closed form solution. In order to enforce normalization of the columns of *D*, all the elements are divided by the l_2-norm of the column it belongs to. The second sub-problem has a closed form update in the form of soft thresholding [25].

$$Z_k = signum(D_k X)\max(0, |D_k X| - \frac{\lambda_2}{\alpha}) \quad (8)$$

The problem (6) is non-convex and hence is not guaranteed to reach any global minima. However each of the sub-problems (7) can be proven to be convex. The algorithm proposed above converges and reaches a local solution. The empirical convergence plot is shown in Figure 2.

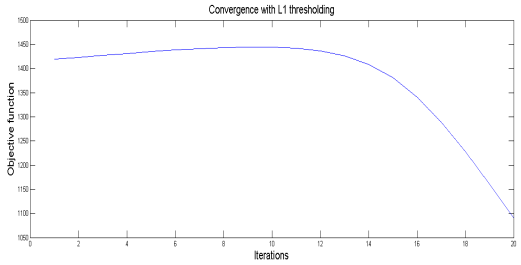

Figure 2. Convergence plot for analysis dictionary learning

3.1.Supervised Learning

The formulation proposed in the previous sub-section is not supervised. Nowhere in the learning algorithm is the class information required. In this sub-section three simple supervision techniques are put forth. The first one is a continuation of the previous idea and is based on the concept of row-sparsity.

3.1.1.Row-Sparsity

Assume that the training data belongs to multiple classes and are represented by X_c where c denotes the class. This work postulates that samples in the same class should lead to similar sparsity patterns, i.e. features would have a common support. See Figure 3 – the dark circles denote non-zero values and the white ones denote zeroes. Some arbitrary sparsity patterns for each class are shown, but according to the assumption the pattern is consistent within the class. Thus the corresponding features (DX_c=)Z_c are supposed to be row-sparse within every class, i.e. they have non-zero values only in a few rows.

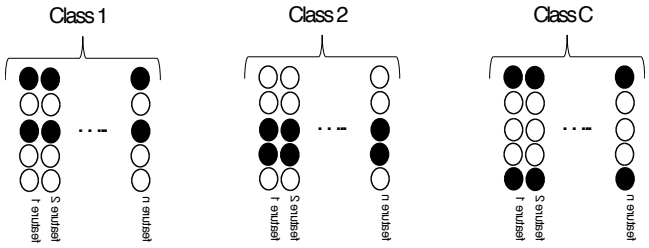

Figure 3. Illustration of row-sparsity

The optimization problem that accounts for row-sparsity is formulated as:

$$\min_{D,Z} \|DX - Z\|_F^2 + \lambda_1 \|D\|_2 + \lambda_2 \sum_c \|Z_c\|_{2,1} \text{ s.t. } \|d_i\| = 1 \quad (9)$$

The $l_{2,1}$-norm is defined as the sum of the l_2-norms of the rows. The l_2-norm on the row vectors promotes a dense solution, but the sum – of – the – l_2-norms promotes sparsity in the selection of rows. Such mixed norms for row-sparsity are widely used in signal processing [26, 27]. The step for dictionary update is the same as before (7a). But the update for the coefficients will be different. The problem can be expressed as,

$$\min_{Z_c's} \sum_c \|D_k X_c - Z\|_F^2 + \lambda_2 \|Z_c\|_{2,1} \quad (10)$$

The individual Z_c*'s* can be solved separately by solving the $l_{2,1}$-norm minimization problem. An efficient solution for (10) has been derived in [28]; it is based on modified soft thresholding. The step is:

$$Z_k = signum(D_k X_c)\max(0, |D_k X_c| - \frac{\lambda_2}{2\alpha}\Lambda) \quad (11)$$

where $\Lambda = diag(\|Z_{k-1}^{j\rightarrow}\|_2^{-1})|Z_{k-1}^{j\rightarrow}|$; $Z_{k-1}^{j\rightarrow}$ denotes j^{th} row.

Group-sparsity has been used in autoencoders [29] and restricted Botlzmann machines (RBM) [30] in the recent past to enforce supervision.

3.1.2.Rank Deficiency

The formulation described here is slightly different from the sparsity based techniques. Here a dictionary is learnt such that the features from the same class will be similar (linearly dependent) to each other, and hence if they are stacked as columns of a matrix, the corresponding matrix will be rank deficient. It means that even though the input raw samples for each class may be dissimilar (larger angle between samples), the features from the same class will be very similar to each other and will have a very small angle between them. Smaller angle means larger linear dependency, which in turn will lead to rank-deficiency of the feature matrix. The scheme is illustrated in Figure 4. The input training samples are shown as black lines; at the output the learned dictionary generates linearly dependent features (shown by different colors). This formulation can be expressed as,

$$\min_{D,Z} \|DX - Z\|_F^2 + \lambda_1 \|D\|_2 + \lambda_2 \sum_c \|Z_c\|_* \text{ s.t. } \|d_i\| = 1 \quad (12)$$

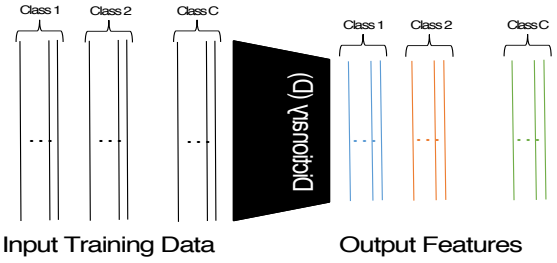

Figure 4. Scheme elucidating rank deficiency. Colors define linear dependence with each other.

Here $\|Z_c\|_*$ denotes the nuclear norm of the matrix. It is the closest convex surrogate of its rank; a small nuclear norm leads to a low-rank in most cases. Learning the dictionary from (12) is the same as before (7a). Learning the coefficients is expressed as,

$$\min_{Z_c's} \sum_c \|D_k X - Z\|_F^2 + \lambda_2 \|Z_c\|_* \tag{13}$$

This is a nuclear norm minimization problem and can be solved using singular value shrinkage [31]. The steps are:

$$D_k X_c = U \Sigma V^T$$

$$Z_k = U \; Soft_{\lambda/2}(\Sigma) \; V^T$$

$$\text{where } Soft_{\lambda/2}(\Sigma) = diag(\Sigma) \cdot \max(0, diag(\Sigma) - \frac{\lambda_2}{2})$$

3.1.3.Label Consistency

In this formulation, the analysis dictionary is learnt in a similar fashion as the unsupervised setting; but an additional supervision term is introduced that learns a linear map from the coefficients to the target class labels. The formulation is as follows:

$$\min_{D,Z,W} \|DX - Z\|_F^2 + \lambda_1 \|D\|_2 + \lambda_2 \|Z\|_1 + \lambda_3 \|Q - WZ\|_F^2 \tag{14}$$

Here Q consists of binary class labels, i.e., say there are three classes – the class label for class 1 will be $[1, 0, 0]^T$, for class 2 it will be $[0, 1, 0]^T$ and for class 3 it will be $[0, 0, 1]^T$. W is a projection that maps the features to the class labels. The full scheme is shown in Figure 5. Such a type of formulation has been used for synthesis dictionary learning in [32], [33], in RBM [34] and in autoencoders (single layer [35] and stacked [36]).

Note that one does not need to enforce the column normalization term in (14); the additional label consistency term discounts for the trivial solution.

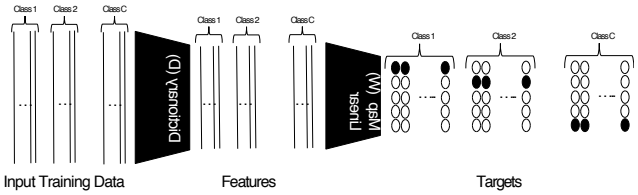

Figure 5. Illustration for Label Consistency Formulation

The above problem can solve for the variables (D, Z and W) using alternate minimization. The update for the dictionary remains as before (7a). The update for the feature (Z) is:

$$\min_Z \|DX - Z\|_F^2 + \lambda_2 \|Z\|_1 + \lambda_3 \|Q - WZ\|_F^2 \tag{15}$$

This can be compactly represented as:

$$\min_Z \left\| \begin{pmatrix} DX \\ Q \end{pmatrix} - \sqrt{\lambda_3} \begin{pmatrix} Z \\ WZ \end{pmatrix} \right\|_F^2 + \lambda_2 \|Z\|_1 \tag{16}$$

Note that, this is not a simple problem as (7b) or (9); the variable Z is coupled in the least squares term. Thus it does not have a closed form update but can be solved using iterative soft thresholding algorithm (ISTA) [37].

ISTA:

$\min_T \|Y - HT\|_F^2 + \lambda \|T\|_1$: Basic ISTA steps

Landweber Iteration: $$B = T_{k-1} + \frac{1}{\alpha} H^T (Y - HT_{k-1})$$

Soft Thresholding: $$T_k = signum(B) \max(0, |B| - \frac{\lambda}{2\alpha})$$

where α is the maximum eigen value of $H^T H$.

The final last task is to update the projection W. This is a simple least squares problem (17), having a closed form solution.

$$\min_W \|Q - WZ\|_F^2 \tag{17}$$

3.2.Advantage of Analysis Dictionary Learning

Once the analysis dictionary is learnt (supervised or unsupervised), using it for feature extraction on a test sample is easy – one just needs to apply the learned dictionary on the sample. If x_{test} is the test sample and D is the learned dictionary, the feature corresponding to the sample is obtained by:

$$z_{test} = Dx_{test} \tag{18}$$

A matrix vector multiplication has a complexity of $O(mn)$ – thus the feature generation during operation stage for the proposed analysis formulations is very fast.

On the other hand, synthesis dictionary learning methods (irrespective of supervised and unsupervised) require solving an l_1-norm minimization problem for feature generation from test samples.

$$\|x_{test} - Dz_{test}\|_2^2 + \lambda \|z_{test}\|_1 \tag{19}$$

This (19) needs to be solved iteratively. The computational complexity for every iteration is $O(mn^2)$. Approximately $O(n^{.5})$ iterations are required. Therefore the overall complexity of generating a feature from a test sample is $O(mn^{2.5})$. Thus the computational complexity of synthesis dictionary learning is significantly larger compared to analysis dictionary learning when in operation (after training).

3.3.Connection with Restricted Boltzmann Machine

The architecture for a Restricted Boltzmann Machine (RBM) is shown in Figure6. It is a generic architecture where there is an input layer fully connected with a hidden / latent layer.

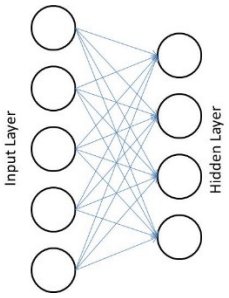

Figure 5. Restricted Boltzmann Machine

In RBM, the problem is to learn the connection weights (mapping) from the input layer to the output/hidden layer as well as the features at the hidden layer. In RBM the energy function used to find out the weights and the features is the Boltzmann function. Basically one tries to learn the network weight and the output features such that the similarity between the projected data (at the input) and the features (Z^TDX) is maximized.

In the proposed method, the cost function is modified– instead of maximizing similarity; the Euclidean distance between the projection of the data (DX) and the generated features (Z) is minimized.

The basic architecture of the RBM is the same as the analysis dictionary framework. They differ from each other in the cost functions. The main disadvantage of RBM is that (ideally) it only works with binary inputs; it can be modified to work with inputs that are real numbers between 0 and 1. But the proposed formulation does not have such constraints – it will work with any real or complex data.

4.Data Acquisition

In this work, four different consumer appliances (appliance under test – AUT) are considered – laptop charger, desktop computer, CFL and LCD monitor. While these are not the only commonly occurring consumer appliances, they are chosen since they are used both in residences and offices. Five instances of the same appliance make and model are considered, and individual measurements are made for each instance. A sensor similar to the one proposed in [7] is used for collecting both common-mode (CM) and differential mode (DM) EMI injected by an appliance under test (AUT) on the power line. The experimental setup is shown in the figure below. The sensor directly interfaces with the phase, neutral and earth power lines through an extension cord through which the AUT is powered. The DM EMI is measured from the differential across the phase and neutral lines. A high pass filter is introduced to remove the 230V, 50Hz power signal from the measurement. The CM EMI is measured directly from the earth currents. The measurements are stored in an internal buffer within the sensor and then uploaded to an external computer for further processing through the wired interface.

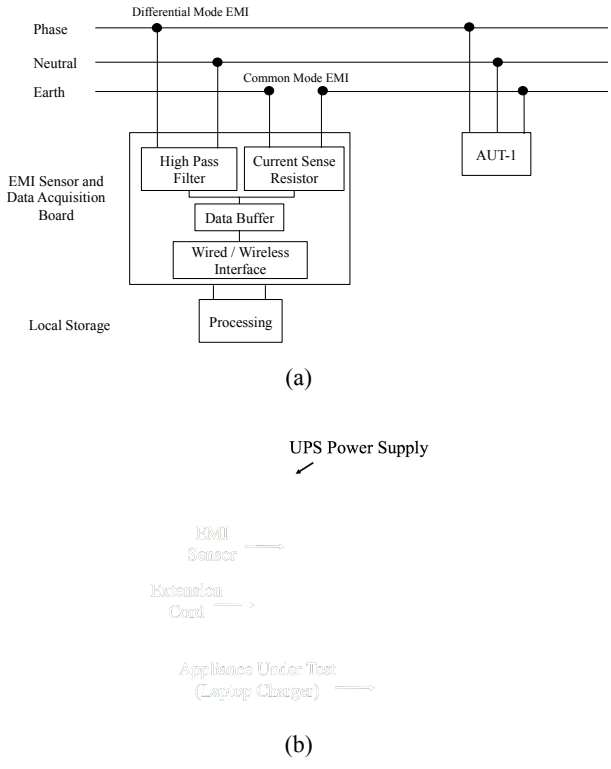

Figure 7. (a) shows the block view of EMI sensor and 7(b) shows the actual test setup used for EMI measurements, buffered to local storage using a wired interface. The AUT is also connected to the same power line through an extension cord.

Time domain measurements (CM EMI and DM EMI traces) of 1ms duration were made at a sampling frequency of 15.625MHz. A total of 1500 traces are measured for each instance of an AUT. In each case, the background noise on the power lines before connecting the AUT was also collected for comparison. This background component arises from the noise on the electrical infrastructure such as power harmonics from the supply side, transients from the high voltage loads like heating, ventilation and air-conditioning, and emissions from other appliances on the transmission line. The measured data shows some interesting features. First, the strength of the background component on DM EMI measurements are higher than on CM EMI due to the presence of harmonics from the power supply on the DM EMI measurements. Second, all the IT loads considered, were high-quality appliances that were fitted with filters specifically targeted towards reducing their DM emissions. As a result, most of the appliances showed very similar DM EMI signatures. The signal-to-noise ratio (EMI to background noise) ratio was substantially higher for CM EMI measurements than DM EMI measurements.

Figure 8. Left to Right: MNIST, CIFAR-10 and SVHN

5.Experimental Evaluation

5.1.Benchmarking

Since we are proposing a new representation-learning framework, it is imperative that we compare with existing techniques on benchmark problems.

5.1.1.Datasets

To test our formulation we used three datasets, MNIST [40], Street View House Numbers (SVHN) [41] and CIFAR-10 [42]. The MNIST dataset is a handwriting recognition dataset developed by Y. LeCun et al. using the larger NIST dataset. It has 60,000 images of handwritten digits, which were used as training images and 10,000 images were used as test images. The SVHN dataset is obtained from Google Street View Images dataset. It also involves recognition of digits, like the MNIST, however, it is significantly harder to do so because of clustering of nearby digits and variety of backgrounds. It is a real world problem of recognizing the digits from natural scene images. It is a coloured images database, with 73,257 images for training, and 26,032 images for the test. There are also 531,131 simpler training images; however, we do not use them. We use format 2 of the dataset, which is like the MNIST dataset. Alex Krizhevsky et al. compiled the CIFAR-10 dataset from the 80 million small images dataset. This dataset contains 50,000 32x32 training images with ten classes which are mutually exclusive. CIFAR-10 contains images from various categories such as ship, frog, truck and more. This dataset contains 10,000 test images.

5.1.2.Pre-processing

MNIST: No preprocessing was used on this dataset.

CIFAR: We used a similar preprocessing as was used by Zeiler et al. in [43]. From each pixel, we subtracted the mean of the image for all images in the dataset. This suppressed the brightness variation in the image. The entire three channel image after the mean suppression was used for training and testing. A similar pre-processing in the form of SVHN database was tried; however, we lost the clarity of features in the image by this process.

SVHN: We use similar preprocessing as used by Sermanet et al. [44]. We contrast normalize the Y channel of the YUV images of the dataset and use only the Y channel for training and classification. The Y channel is locally contrast normalized using a Gaussian neighborhood, with a 7x7 window. This made the images look more like the MNIST database. The resultant images reside in an R1024 space. From figure 5 we see that the Y channel contains the shape information in a clear and precise manner as compared to the U and V channels. Figure 5c shows the preprocessed Y channels of the SVHN dataset. We only use the Y channel for training. The same preprocessing is applied to the test set before the classification step.

5.1.3.Results

We have compared with Discriminative KSVD [32] and Label Consistent KSVD (LC-KSVD)[33] was used as the baseline for the various datasets. This method has shown to yield better results than other dictionary learning techniques for classification problems. After generating features by our proposed method, we used k-nearest neighbor (KNN) classifier; sparse representation classifier (SRC) [35] and artificial neural network (ANN). The no. of neighbors used for nearest neighbor classification (unless specified otherwise) is 100. The number of hidden layers in neural nets was fixed to 100. The SRC is non-parametric.

Now we describe the parameters for our proposed algorithm. For the sparse (unsupervised) and row-sparse (supervised) formulations, the number of atoms for using row sparse formulation is 200 with a regularization parameter of 0.5. For the low-rank formulation, the number of atoms are 500; the regularization parameter is 0.27. For the label consistency

formulation, the number of atoms are 200 with a regularization parameter of 0.2; the parameter for learning the linear map is unity.

Table 1. Unsupervised Analysis Deep Dictionary Learning

Method	MNIST	SVHN	CIFAR-10
Disc. KSVD [32]	82.6	25.7	17.8
LC-KSVD2 [33]	94.1	30.0	26.0
Proposed + KNN	91.1	55.0	31.9
Proposed + SRC	84.7	66.4	32.3
Proposed + ANN	95.3	76.6	46.0

Table 2. Row-sparse Analysis Deep Dictionary Learning

Method	MNIST	SVHN	CIFAR-10
Disc. KSVD [32]	82.6	25.7	17.8
LC-KSVD2 [33]	94.1	30.0	26.0
Proposed + KNN	91.1	55.6	31.4
Proposed + SRC	84.6	66.9	33.8
Proposed + ANN	95.8	74.5	42.3

Table 3. Low-rank Analysis Deep Dictionary Learning

Method	MNIST	SVHN	CIFAR-10
Disc. KSVD [32]	82.6	25.7	17.8
LC-KSVD2 [33]	94.1	30.0	26.0
Proposed + KNN	93.2	57.3	32.0
Proposed + SRC	86.2	73.1	32.9
Proposed + ANN	96.4	78.4	45.6

Table 4. Label-Consistent Analysis Deep Dictionary Learning

Method	MNIST	SVHN	CIFAR-10
Disc. KSVD [32]	82.6	25.7	17.8
LC-KSVD2 [33]	94.1	30	26
Proposed + KNN	93.2	59.2	32.6
Proposed + SRC	87.7	76.7	32. 7
Proposed + ANN	96.3	78.6	46.2

Discriminative KSVD and LC-KSVD are not dependent on the external classifier; hence their results remain the same. We have repeated them in different tables for the ease of comparison. We see that our proposed method is at par and even slightly better than these well-known synthesis dictionary learning techniques for MNIST. SRC does not perform well with our proposed framework, but KNN and ANN do. With KNN we always perform better than [32] and slightly worse then [33]; with the neural network, we always yield the best results.

MNIST is the easiest dataset; for more complex ones like CIFAR-10 and SVHN we always perform significantly better.

The results are not at par with those of deep learning techniques like convolutional neural network, deep belief network or stacked denoising autoencoder. However, it would not be fair to compare our new yet simple technique to these complex tools. We are proposing a new framework for analysis dictionary learning, and we have compared it against well-known synthesis dictionary learning techniques.

5.2.Appliance Classification

Experiments were carried out on the toughest possible scenario. The training set consisted of samples from only one of the five instances for each appliance while the remaining four instances of each appliance constituted the test set. This is also the most practical scenario. In the office environment, it is not possible to train on every possible instance of the same appliance – one should be able to identify multiple instances of the same appliance after being trained on a single instance.

The proposed techniques were compared with [6] – which was specifically developed for appliance classification problems. Deep learning techniques like stacked autoencoders and deep belief networks were also used. The results from these were as good as random labelling – basically, all the samples were getting mapped to one class, hence the accuracy was always $= \frac{1}{\text{num of appliances}}$. The results showed no difference with the type of data; raw data or Fourier magnitudes or cepstrum features. Even though thorough experiments were carried out on these, results for deep learning are not reported in this study, as they yield poor results.

Both CM and DM EMI data were collected. Analysis on the DM EMI yields poor results (same as random label assignment), no matter what technique is used – this supports the prior discussion regarding the disadvantages of DM EMI for appliance detection and classification. Since this signal is filtered by most of the today's sophisticated appliances, there is no distinguishing information left. Since DM EMI results are poor, they are not presented here. The results shown here are for CM EMI.

Traditional feature selection methods like Principal Component Analysis, Linear Discriminant Analysis and their kernelized versions also failed to produce any improvements. Finally, one of the most recent information theoretic feature selection techniques [38] based on conditional likelihood maximization was used. These features were input to the neural network (NN) and support vector machine (SVM) for classification.

The proposed techniques, described in the previous sections, were implemented on the raw time-domain data, but it yielded poor classification results. This is because the time domain data is not synchronized, i.e. the samples are shifted versions of each other. Operating on the Fourier frequency magnitudes yields somewhat superior results. However, the best results are obtained for cepstrum features. The motivation for using cepstrum features follow from [39] – where it was used for feature extraction on an energy disaggregation problem. Some representative cepstrum features for different appliances are shown in Fig. 8. This figure shows that the features look similar for different instances of the same appliance but are different across different appliances. The

last category, background noise, is the data measured when there are no appliances connected to the power lines. The dictionary learning techniques were applied to the cepstrum features.

The features generated by the dictionary learning process are used to train a neural network for classification. For the test data, the features are simply generated by multiplying it with the learned dictionary. These features are input to the learned neural network for classification. The overall classification accuracy across various methods is shown in Table 13.

However, the overall classification accuracy does not yield insight. Therefore the confusion matrices are shown for all the appliances in following Tables 5 through 12. The diagonal values demonstrate the percentage of correct classification. The off-diagonal elements show the proportion of a device being misclassified as some other device.

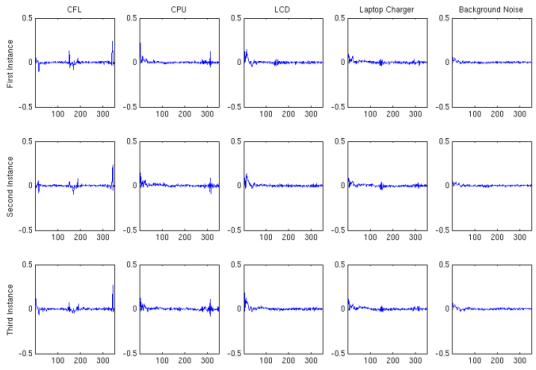

Figure 9. Cepstrum features – horizontal axis – frequency in kHz and vertical axis – Volt

Table 5. Confusion Matrix - Unsupervised ADL

Confusion Matrix: Unsupervised ADL					
	CFL	CPU	LCD	LC	Noise
CFL	65.6%	0.8%	0.4%	3.1%	5.8%
CPU	1.7%	88.3%	5.2%	0.1%	0.0%
LCD	3.3%	9.4%	79.9%	26.2%	0.0%
LC	3.7%	1.6%	14.5%	68.4%	0.0%
Noise	25.7%	0.0%	0.0%	2.2%	94.2%

Table 6. Confusion Matrix – Row-sparse ADL

Confusion Matrix: Row-sparse ADL					
	CFL	CPU	LCD	LC	Noise
CFL	69.2%	1.1%	0.1%	1.0%	4.7%
CPU	1.7%	88.1%	3.5%	0.0%	0.0%
LCD	2.7%	8.7%	79.9%	25.1%	0.0%
LC	3.4%	2.1%	16.4%	73.0%	0.0%
Noise	22.9%	0.0%	0.0%	0.9%	95.3%

Table 7. Confusion Matrix – Low-rank ADL

Confusion Matrix: Low-rank ADL					
	CFL	CPU	LCD	LC	Noise
CFL	72.0%	0.9%	0.6%	2.1%	5.1%
CPU	0.4%	87.2%	2.3%	0.1%	0.0%
LCD	4.1%	10.2%	79.9%	24.7%	0.0%
LC	2.1%	1.8%	17.2%	70.3%	0.0%
Noise	21.5%	0.0%	0.0%	2.8%	94.9%

Table 8. Confusion Matrix – Label-Consistent ADL

Confusion Matrix: LC ADL					
	CFL	CPU	LCD	LC	Noise
CFL	52.8%	5.1%	0.0%	0.9%	0.2%
CPU	4.7%	71.0%	6.8%	0.9%	0.0%
LCD	4.2%	18.6%	79.7%	29.5%	0.0%
LC	1.3%	5.3%	13.0%	60.3%	0.0%
Noise	37.0%	0.0%	0.5%	8.4%	99.8%

Table 9. Confusion Matrix [5]

Confusion Matrix: GMM + kNN (ElectriSense)					
	CFL	CPU	LCD	LC	Noise
CFL	0.0%	75.0%	0.0%	25.0%	0.0%
CPU	0.0%	50.0%	0.0%	50.0%	0.0%
LCD	0.0%	25.0%	0.0%	75.0%	0.0%
LC	0.0%	25.0%	0.0%	75.0%	0.0%
Noise	0.0%	25.0%	0.0%	50.0%	25.0%

Table 10. Confusion Matrix – CLM [38] + Nearest Neighbor

Confusion Matrix: CLM + NN					
	CFL	CPU	LCD	LC	Noise
CFL	50.5%	38.4%	20.4%	31.2%	20.0%
CPU	9.3%	51.3%	0.0%	0.5%	0.0%
LCD	17.9%	0.1%	79.4%	0.3%	0.0%
LC	13.0%	10.1%	0.2%	68.0%	0.0%
Noise	9.2%	0.0%	0.0%	0.0%	80.0%

Table 11. CLM [38] + Support Vector Machine

Confusion Matrix: CLM + SVM					
	CFL	CPU	LCD	LC	Noise
CFL	29.3%	25.4%	1.1%	15.7%	0.1%
CPU	9.4%	49.4%	6.5%	6.7%	6.5%
LCD	27.9%	5.2%	84.2%	8.2%	5.0%
LC	23.0%	20.0%	8.2%	69.4%	8.4%
Noise	10.4%	0.0%	0.0%	0.0%	80.0%

Table 12. Confusion Matrix [6]

Confusion Matrix: [6]					
	CFL	CPU	LCD	LC	Noise
CFL	89.90%	4.90%	13.30%	5.60%	5.20%
CPU	0%	64.20%	0.60%	11.90%	0%
LCD	5.60%	8.90%	67.10%	1.10%	4.30%
LC	0%	11.10%	0%	69.40%	3.70%
Noise	4.50%	10.90%	19%	12%	86.80%

* Please note that the results from Table 12 cannot be directly compared with [6]. This is because of the random splits used in [6] are different from ours.

Table 13. Overall Classification Results

[5]	[6]	CLM + NN	CLM + SVM	ADL	R S ADL	L R ADL	L C ADL
30	75.5	65.87	62.46	79.29	81.12	80.85	80.20

*RS – row-sparse; LR – low-rank; LC – label-consistent

6.Conclusion

In this work, the problem of identifying consumer appliances, used in most office environments, by their EMI signatures was addressed. This problem has gained interest since the publication of ElectriSense [5] in 2010. The main difference between [5] and [6] is that here the common mode (CM) EMI signature is acquired whereas the previous one used differential mode (DM) EMI signature. The shortcomings of DM EMI are discussed – the power signal and its harmonics interfere with the DM EMI; hence analyses based on such signatures are not reliable. CM EMI

measurements are unaffected by the power signal and hence the CM EMI carries more discerning information about appliances.

Prior techniques [5, 6] for appliance identification based on EMI signature were largely heuristic. In this work a new technique for feature extraction is developed – analysis dictionary learning. The basic formulation is unsupervised; three supervised variations are also proposed. The proposed formulation generates features, which are further employed to train a neural network for classification. The results show that the proposed method yields a significant improvement over [5, 6]

There are two benefits of analysis dictionary learning. It has a faster operation compared to prior synthesis dictionary learning. This is because, in synthesis dictionary learning, one needs to solve a convex optimization problem iteratively – this is time-consuming. Analysis dictionary learning just requires a matrix-vector multiplication. Therefore, the feature extraction time during testing is drastically reduced. This makes the technique suitable for real-time processing.

The other benefit of this approach is for the future. We would like to address the scenario where multiple appliances are running simultaneously. This is the disaggregation problem. It is possible to extend the dictionary learning based approach to solve this – synthesis dictionary learning has already achieved this for smart-meter data. Also, there are several applications based on energy disaggregation / non-intrusive load monitoring; currently they are based on power meter readings. We would like to explore if such problems can be solved in a better fashion using CM EMI signatures and analysis dictionary learning.

7.ACKNOWLEDGMENTS

Authors acknowledge the support provided by ITRA project, funded by DEITy, Government of India, under grant with Ref. No. ITRA/15(57)/Mobile/HumanSense/01.

8.REFERENCES

1. Perez-Lombard L., Ortiz J., and Pout, C. 2008. A review on buildings energy consumption information. *Energy and buildings*. 40, 394-398.
2. Hart G.W. 1992. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*. 80, 1870-1891.
3. Koutitas, G. C., and Tassiulas L. 2016. Low Cost Disaggregation of Smart Meter Sensor Data. *IEEE Sensors Journal*. 16 (6), 1665-1673.
4. Xu, Y., and Milanovi, J. V. 2015. Artificial-Intelligence-Based Methodology for Load Disaggregation at Bulk Supply Point. *IEEE Transactions on Power Systems*. 30 (2), 795-803.
5. Gupta, S., Reynolds, M. S., and Patel, S.N. 2010. ElectriSense: single-point sensing using EMI for electrical event detection and classification in the home. *12th ACM international conference on Ubiquitous computing*.
6. Gulati, M., Ram, S. S., Majumdar, A., and Singh, A. 2016. Single Point Conducted EMI Sensor With Intelligent Inference for Detecting IT Appliances. *IEEE Transactions on Smart Grid (accepted)*.
7. Kulkarni, A. S., Harnett, C. K., and Welch, K. C. 2015. EMF Signature for Appliance Classification. *IEEE Sensors Journal*. 15 (6), 3573-3581.
8. Engan, K., Aase, S., and Hakon-Husoy, J. 1999. Method of optimal directions for frame design. *IEEE International Conference on Acoustics, Speech, and Signal Processing*.
9. Aharon, M., Elad, M., and Bruckstein, A. 2006. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing*. 54 (11), 4311-4322.
10. Elad, M., and Aharon, M. 2006. K-SVD: Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries. *IEEE Transactions on Image Processing*. 15 (12), 3736-3745.
11. Natarajan, B. K. 1995. Sparse approximate solutions to linear systems. *SIAM Journal on computing*. 24, 227-234, 1995.
12. Pati, Y., Rezaiifar, R., and Krishnaprasad, P. 1993. Orthogonal Matching Pursuit : recursive function approximation with application to wavelet decomposition. *Asilomar Conference on Signals, Systems and Computing.*
13. Yaghoobi, M., Blumensath, T., and Davies, M. E. 2009. Dictionary Learning for Sparse Approximations With the Majorization Method. *IEEE Transacyions on Signal Processing*.57 (6), 2178-2191.
14. Rakotomamonjy, A. 2013. Applying alternating direction method of multipliers for constrained dictionary learning. *Neurocomputing*. 106 (15), 126-136.
15. Rubinstein, R., Peleg, T., Elad, M. 2013. Analysis K-SVD: A Dictionary-Learning Algorithm for the Analysis Sparse Model. *IEEE Transactions on Signal Processing*. 61 (3), pp. 661-677.
16. Mairal, J., Bach, F., Ponce, J., Sapiro, G., and Zisserman, A. 2008. Discriminative learned dictionaries for local image analysis. *IEEE International Conference on Computer Vision and Pattern Recognition*.
17. Zhang, W., Surve, A., Fern, X., Dietterich, T. 2009. Learning non-redundant codebooks for classifying complex objects. *International Conference on Machine Learning*.
18. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A. 2009. Supervised dictionary learning. *Neural Information Processing Systems*.
19. Mairal, J., Leordeanu, M., Bach, F., Hebert, M., and Ponce, J. 2008. Discriminative sparse image models for class-specific edge detection and image interpretation. *European Conference on Computer Vision*.
20. Huang, K., Aviyente, S. 2007. Sparse representation for signal classification 2007. *Neural Information Processing Systems*.
21. Pham, D., and Venkatesh, S. 2008. Joint learning and dictionary construction for pattern recognition. *IEEE International Conference on Computer Vision and Pattern Recognition*.
22. Yang, J., Yu, K., and Huang, T. 2010. Supervised translation-invariant sparse coding. *International Conference on Computer Vision and Pattern Recognition*.
23. Figueiredo, M., Ribeiro, B., and de Almeida, A. 2014. Electrical Signal Source Separation Via Nonnegative Tensor Factorization Using On Site Measurements in a Smart Home. *IEEE Transactions on Instrumentation and Measurement*. 63 (2), 364-373.
24. Figueiredo, M., Ribeiro, B., and de Almeida, A. 2015. Analysis of trends in seasonal electrical energy consumption via non-negative tensor factorization. *Neurocomputing*. 170, 318-327.
25. Donoho, D. L. 1995. De-noising by soft-thresholding. *IEEE Transaction on Information Theory*. 41 (3), 613-627.

26. Van den Berg, E., and Friedlander, M. P. 2010 .Theoretical and empirical results for recovery from multiple measurements. *IEEE Transactions on Information Theory*. 56 (5), 2516-2527.
27. Cotter, S. F., Rao, B. D., Engan, K., and Kreutz-Delgado, K. 2005. Sparse solutions to linear inverse problems with multiple measurement vectors. *IEEE Transactions on Signal Processing*. 53 (7), 2477-2488.
28. Majumdar, A., and Ward, R. K. 2012. Synthesis and Analysis Prior Algorithms for Joint-Sparse Recovery. *IEEE International Conference on Acoustics, Speech, and Signal Processing*.
29. Majumdar, A., Vatsa, M., and Singh, R. 2016. Face Recognition via Class Sparsity based Supervised Encoding. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Epub).
30. Sankaran, A., Sharma, G., Singh, R., Vatsa, M., and Majumdar, A. 2016. Class Sparsity Signature based Restricted Boltzmann Machines. *Pattern Recognition* (Epub).
31. Majumdar, A., and Ward, R. K. 2011. Some Empirical Advances in Matrix Completion. *Signal Processing*. 91 (5), 1334-1338.
32. Zhang, Q., and Li, B. 2010. Discriminative K-SVD for dictionary learning in face recognition. *IEEE International Conference on Computer Vision and Pattern Recognition*.
33. Jiang, Z., Lin, Z., and Davis, L. S. 2013. Learning A Discriminative Dictionary for Sparse Coding via Label Consistent K-SVD. *IEEE Trans. on Pattern Analysis and Machine Intelligence*. 35, 2651-2664.
34. Larochelle, H., Bengio, Y. 2008. Classification using Discriminative Restricted Boltzmann Machines. *International Conference on Machine Learning*.
35. Gogna, A, and Majumdar, A. 2016. Semi Supervised Autoencoder. *International Conference on Neural Information Processing*.
36. Majumdar, A., Gogna, A., Ward, R. K. 2016. Semi-supervised Stacked Label Consistent Autoencoder for Reconstruction and Analysis of Biomedical Signals. *IEEE Transactions on Biomedical Engineering*. (accepted).
37. Daubechies, I., Defrise, M., and De Mol, C. 2014. An iterative Thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*. 57, 1413–1457.
38. Brown, G., Pocock, A., Zhao, M. J., Luján, M. 2012. Conditional Likelihood Maximisation: A Unifying Framework for Information Theoretic Feature Selection. *Journal of Machine Learning Research*. 13, 27−66.
39. Kong, S., Kim, Y., Joo, S. K., and Kim, J. H. 2015. Home appliance load disaggregation using cepstrum-smoothing-based method. *IEEE Transactions on Consumer Electronics*. 61 (1), 24-30.
40. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. 86(11), 2278-2324.
41. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A. Y. 2011. Reading Digits in Natural Images with Unsupervised Feature Learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.
42. Krizhevsky, A., and Hinton, G. 2009. Learning Multiple Layers of Features from Tiny Images. Technical Report, University of Toronto.
43. Zeiler, M. D., and Fergus, R. 2013. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*.
44. Sermanet, P., Chintala, S., & LeCun, Y. 2012. Convolutional neural networks applied to house numbers digit classification. *International Conference on Pattern Recognition*.